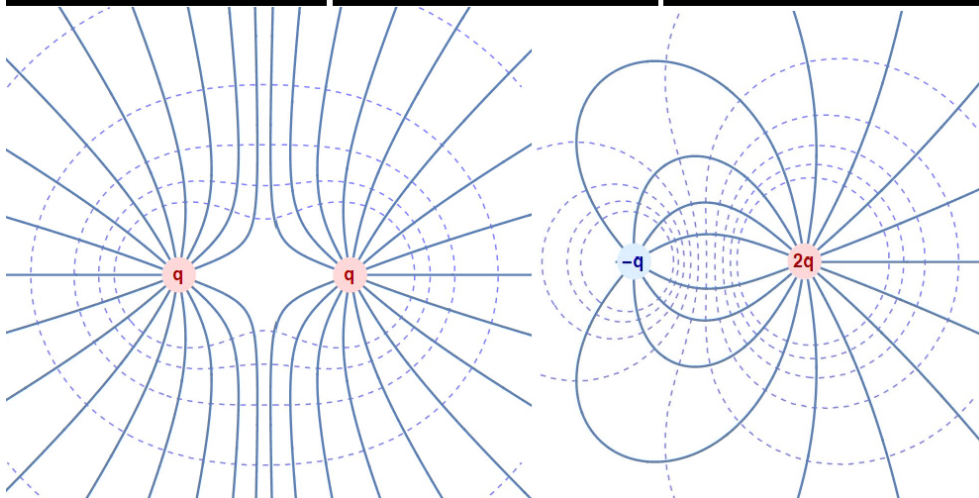
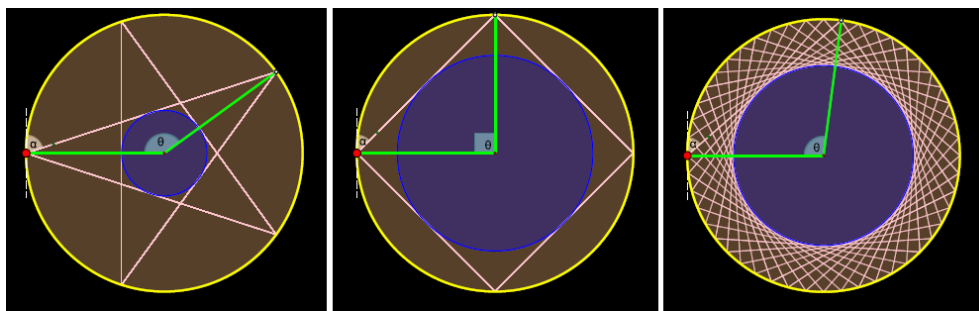

RAČUNARI U NASTAVI FIZIKE

Wolfram Mathematica + Cinderella

Zoran P. Popović
Saša Dmitrović



UNIVERZITET U BEOGRADU
FIZIČKI FAKULTET

RAČUNARI U NASTAVI FIZIKE

elektronsko izdanje

Autori: Doc. dr Zoran P. Popović
Prof. dr Saša Dmitrović

Izdavač: Univerzitet u Beogradu,
Fizički fakultet
Studentski trg 16, Beograd
<http://www.ff.bg.ac.rs>

Recenzenti: Prof. dr Zoran Nikolić
Prof. dr Božidar Nikolić

ISBN 978-86-84539-41-2

Predgovor

Ovaj udžbenik je nastao na osnovu kursa Racunari u nastavi fizike, koje autori već duži niz godina drže na Fizičkom fakultetu Univerziteta u Beogradu. Navedeni kurs je namenjen studentima smera Opšta fizika (A smer). Okosnica kursa je software Wolfram Mathematica, na koju se odnosi prvi deo knjige i Cinderella, na koji se odnosi drugi deo knjige. Izbor navedenih programskih paketa nije slučajan. Mathematica je jedan od najpoznatijih primera tzv. kompjuterskog algebarskog sistema (CAS) opšte namene: računarskog sistema za manipulaciju simboličkih (algebarskih) izraza, koji je istovremeno prilagođen za izvršavanje složenih numeričkih proračuna, kao i za računarsku grafiku visokog nivoa. Sam naziv ukazuje na suštinu samog sistema: manipulacija različitih tipova izraza koji se na izlazu transformišu u traženi tip podataka, izraze, numeričke podatke ili grafiku. Ostali poznati primeri CAS su Maple, SageMath, Maxima... Cinderella je verovatno najmoćniji predstavnik softvera (okruženja) za dinamičku (interaktivnu) geometriju (DGE). To su programi za kreiranje i manipulaciju geometrijskih konfiguracija. Ulazni podaci su najčešće kreirani direktno crtanjem (pomoću računarskog miša) i koji zatim postaju interaktivni. Manipulacija konfiguracijama posredstvom softvera ovog tipa omogućava primenu i razvoj geometrijske intuicije pri rešavanju problema iz matematike, fizike i tehnike. Poznati primeri DGE sistema, pored programa Cinderella su GeoGebra, The Geometer's Sketchpad, Cabri Geometry itd. Jasno je, već iz samog naziva i definicije da su programi za dinamičku geometriju komplementarni kompjuterskim algebra sistemima. Jedan od glavnih zadataka ove knjige je i da omogući studentima da na osnovu brojnih primera sami utvrde tipove problema kod kojih je primena jednog od ova dva tipa softvera efikasnija.

Prvi deo knjige sadrži rešene zadatke iz matematike i fizike koristeći funkcije programskog jezika Wolfram sa uputstvima za izradu. Zadaci su podeljeni i smešteni u poglavlja, izuzev uvodnog, nazvana prema upotrebljenim alatima za rešavanje: grafičke, simboličke i funkcije za numeričko rešavanje i implementacija numeričkih algoritama. Prvo poglavlje namenjeno je opisu funkcija jezika Wolfram implementiranih tokom rešavanja primera. Pisanju programa kojim se rešava zadatak prethodi teorijski opis, definisanje fizičkih zakona ili važniji koraci u izvođenju relacija koje se programiraju. Nakon teorijske pripreme prelazi se na pisanje programa, koji je neretko podeljen u više ćelija, prvenstveno iz pedagoških razloga da bi se korak po korak video tok rešavanja. Linije koda su pisane u istaknutim plavim boksovima, da bi se razlikovali od ostatka opisnog teksta i prostim kopiranjem mogu se preneti, nalepiti u ćeliju otvorene sveske programa Mathematica a potom pokrenuti. Programski jezik Wolfram je visokog nivoa, koji sadrži mnoštvo ugrađenih funkcija i podržava različite stilove programiranja kao što su funkcionalno, zasnovano na pravilima ili proceduralno. Interfejs programa Mathematica prilagođen je korisniku u vidu sveske, gde se unosi ulazni kod, dobijaju izlazi ili grafici. Napredne funkcije programa Mathematica optimizuju vreme rada korisnika (studenata, istraživača), umanjuju verovatnoću pravljenja grešaka te omogućavaju produblјivanje intuitivnog razumevanja fizike kroz zapanjujuće interaktivne mogućnosti i animacije dvodimenzionalne i trodimenzionalne grafike. Umesto opterećenosti pisanjem velikog broja linija komplikovanog koda u

programskom jeziku nižeg nivoa, upotrebom programa Mathematica student može da se fokusira na unapređenje veština rešavanja problema i fizičku intuiciju.

Rešeni su zadaci iz različitih oblasti fizike kako bi se uvežbale tehnike simboličkog i numeričkog programiranja i da bi korisnici drugačije sagledali teorijski već rešene probleme. Neki obrađeni, na prvi pogled prosti ili uobičajeni zadaci kao što su: linearni sistem više kuplovanih oscilatora, nalaženje stanja kvantne čestice vezane u konačno dubokoj potencijalnoj jami rešavanjem transcendentne jednačine ili ispitivanje dinamike matematičkog klatna u nelinearnom režimu pri konačnim otklonima, suviše je komplikovano proučavati i vizualizovati *ručno*. Naizgled malo složeniji problemi postaju računski veoma zahtevni, a mnogi nemaju analitička rešenja, pri čemu usvojene tehnike programiranja olakšavaju njihovo rešavanje ili simulaciju i vizualizaciju.

Kurs Računari u nastavi fizike sluša se na smeru Opšta fizika i sledi nakon pređenih kurseva savremene fizike. Kroz ovaj kurs se uvežbava vizuelizacija rešenja zadataka te mogućnost da se kroz veoma lepe i sadržajne ilustracije ili dinamične animacije stekne jasnija predstava o pojavama koje se proučavaju teorijski, a sa tim i njihovo dublje intuitivno razumevanje. Rešavanjem problema na programskom jeziku Wolfram tokom pripreme ispita, stiču se znanja primenjiva i na drugim poljima, a ne samo u nastavi fizike. Analitičke metode se uglavnom bave modelima, aproksimacijama ili razmatranju posebnih ograničenih slučajeva. Očekivano je da studenti nakon ovog kursa mogu sami programerski doći do rezultata, ispitati i vizualizovati ih, te posmatrati efekte variranjem parametara problema kako bi se razvilo bolje intuitivno razumevanje fizičkih pojava. Razrađene tehnike mogu biti dopunski tekst kurseva teorijske fizike, obogaćujući ih u vidu integrisanog dela, nikako ne zadirući u nastavni plani i program.

Neretko, u složenijim primerima programu prethode teorijska objašnjenja, definicije ili izvođenja veličina koje se obrađuju. Ponuđena programska rešenja uglavnom nisu optimalna po pitanju vremena izvršenja, niti su pisana kondenzovano, već u skladu sa pedagoškom praksom, pogodna za rad pri prvom suretu sa programskim kodom. Ubeđen sam da će studenti tokom rada razviti sopstveni stil programiranja te zameniti ponuđena rešenja vlastitim i kreativnijim, ako ne celih zadataka onda nekih njihovih segmenata. Cilj izbora i prezentacije sadržaja knjige je savladavanje programa Mathematica do stepena ne samo korisnog kao sredstvo za učenje i podučavanje fizike, već i usvajanje korisnog alata u istraživanju i primeni, te da stečene veštine i znanja budu dopuna analitičkim.

Drugi deo knjige sadrži rešene zadatke iz matematike i fizike primenom programa Cinderella. Rad u dinamičkom geometrijskom okruženju je potpuno različit od rada u kompjuterskom algebra sistemu i to je ovde naglašeno i promenom samog tekstualnog stila (Tufte Latex). Naime, dok je u programu Mathematica komunikacija dijaloškog tipa, mi pitamo - računar odgovara, u programu Cinderella mi kreiramo geometrijsku konfiguraciju ili fizički sistem, korak po korak, a računar zatim obezbeđuje interaktivnost. Zbog toga su slike integralni i verovatno važniji deo teksta. Tufte Latex stil obezbeđuje slobodnu marginu na svakoj stranici sa primerima, a njena uloga je dvojaka, da obezbedi sve komentare ili pomoćne slike koje su neophodne za praćenje integralnog dela teksta. Takođe, njena uloga je i da sami čitaoci dodaju svoje komentare i primedbe. Konstrukcije se načelno sastoje iz dva dela: geometrijskih elemenata koji se crtaju i unese neposredno, obično računarskim mišem, i malog koda - skripta, koji obezbeđuje potrebnu interaktivnost (ponekad su dovoljni ugrađeni alati, pa i nema potrebe za skriptom).

Kreativni doprinos autora, ukoliko ga ima, se odnosi na izbor pogodnih primera i njihov raspored. Konkretnije, osnovna ideja je, da se polazeći od geometrijskih konstrukcija preko kinematike, statike i geometrijske optike, dođe do dinamičkih problema. Neophodno znanje i matematike i fizike je na nivou osnovnih fakultetskih kurseva, mada je veliki deo primera moguće uraditi na osnovu znanja stečenog u srednjoj školi, a jedan deo čak i na osnovu gradiva osnovne škole. Kao i u prvom delu, i ovde je cilj da se

studentima ponudi još jedan osvrt na poznato gradivo, ali sada iz novog ugla, ugla dinamičke geometrije i da nakon kursa pokušaju sami da proanaliziraju probleme koji ovde nisu obuhvaćeni.

Važano je na kraju napomenuti, da cilj ove knjige nije samo da služi kao udžbenik za predmet Računari u nastavi fizike, već i kao koristan priručnik za sve one koji pohađaju nastavu iz ovde obuhvaćenih oblasti matematike i fizike. Nadamo se da će biti od koristi i nastavnicima fizike i matematike i pomoći im da programe za komjutersku algebru i/ili dinamičku geometriju samostalno primenjuju u svom radu. Takođe, knjigu treba koristiti uporedo sa ukjučenim programom na računaru: primere treba implementirati i, poželjno je, dalje razraditi.

Autor prvog dela (Mathematica) je Zoran Popović, a drugog (Cinderella) Saša Dmitrović. Autori su unapred zahvalni za sve primedbe, korekcije i komentare.

Sadržaj

	<i>I Wolfram programski jezik i Mathematica</i>	1
1	<i>Programski jezik Wolfram</i>	2
2	<i>Grafički potencijal Mathematica</i>	26
3	<i>Simboličke kalkulacije u programskom jeziku Wolfram</i>	52
4	<i>Numerički potencijal i kalkulacije funkcijama programskog jezika Wolfram</i>	117
5	<i>Primeri implementacije numeričkih metoda u rešavanju zadataka</i>	172
	<i>II Cinderella</i>	207
6	<i>Osnovni pojmovi i tehničke napomene</i>	208
7	<i>Pravolinijske i kružne konfiguracije i kretanja</i>	223
8	<i>Složena kretanja: cikloida i epiciklične krive</i>	238
9	<i>Složena kretanja: traktrise i neholonomni sistemi</i>	257

10	<i>Polinomijalne aproksimacije proizvoljnih krivih</i>	271
11	<i>Statičke konfiguracije</i>	277
12	<i>Dinamika jednodimenzionalnih sistema</i>	292
13	<i>Diferencijalne jednačine i dinamički sistemi</i>	304
14	<i>Iteracije i diskretni dinamički sistemi</i>	338
15	<i>Iteracije, fraktali i diskretna dinamika</i>	356
16	<i>Geometrijska optika i bilijari</i>	367
1	<i>Dodatak: Matematička analiza u slikama</i>	381
2	<i>Dodatak: Zglobni i ostali važni mehanizmi</i>	404
3	<i>Bibliografija</i>	419

Deo I

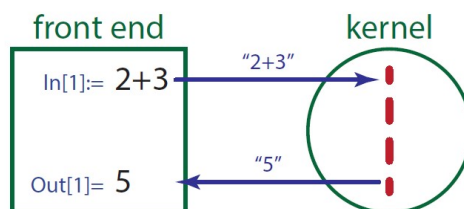
**Wolfram programski jezik i
Mathematica**

1 Programski jezik Wolfram

1.1 Uvod

Programski jezik Wolfram čine podaci, kod, grafika, dokumenti, simbolički izrazi, koji omogućavaju poseban nivo fleksibilnosti i snage programiranja. Novije verzije, pored dosadašnjih nude nove alate za rad kao na primer sa neuronskim mrežama ili sa slikama i video zapisima. Wolfram Mathematica je interaktivni program, koji omogućava različite matematičke kalkulacije. Program Mathematica dozvoljava korisniku da usmeri pažnju na ono čime se bavi umesto na to kako izvršava posao, svrstavajući programiranje u njenom okruženju na visoki nivo. Ona podržava širok spektar programerskih paradigmi, što znači da svaki korisnik može negovati vlastiti stil. Bez oštre granice prelaza, istovremeno je za proračun moguće kombinovati numeričke i analitičke objekte. Biblioteke programa Mathematica imaju bogatu riznicu algoritama za analitičke i numeričke kalkulacije. Zbog toga račun je moguće dovesti do krajnjih granica analitičkih mogućnosti, a potom uz minimalne promene moguće je preći na numeričke.

Prosto gledano Mathematica se sastoji od dve komponente. Jedna je ona u prvom redu, koju korisnik vidi, poznatija kao *front end*. Namenjena je unosu naredbi na programskom jeziku Wolfram, daje izvršne naredbe a tu se vide i izlazi odnosno rezultati izvršenog programa. Druga komponenta je jezgro ili *kernel*, gde se izvršavaju kalkulacije u vidu pokrenutih naredbi.



Jezgro pamti redom sve unesene operacije. One mogu biti u drugačijem redosledu od onih prikazanih u vidljivoj komponenti. Pokretanjem programa Mathematica korisnik dobija praznu svesku (*Notebook*), u koju dalje upisuje komande u vidu linija programa, sačinjene od mešavine teksta, matematičkih simbola i struktura. Dozvoljen je interaktivni pristup, tokom kog je moguće istovremeno razvijati novi program, analizirati dobijene rezultate, gledati grafike i testirati probni program. Moguće je sve pisati u tekstualnom režimu kao na primer

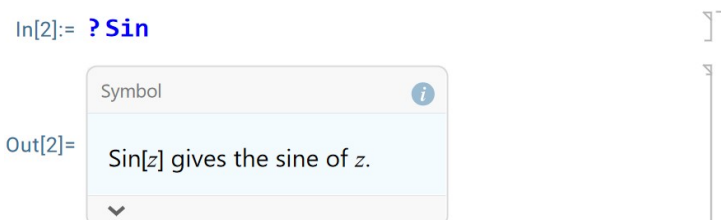
```
In[1]:= 2 + 3
```

```
Out[1]= 5
```



Tekst se piše u otvorenu ćeliju, oivičenoj zdesna, a pokreće se *evoluira* tako što se kursor postavi bilo gde unutar ćelije čije se izvršenje očekuje, a potom izvrši uzastopni pritisak tastera *shift+enter*. Ovim se sadržaj ćelije šalje u jezgro, koje izvršava naredbu a rezultat prikazuje u sledećoj ćeliji koju otvara po izvršenju programa. Matematika automatski generiše numeracije linija u vidu oznaka *In[1]:=* i *Out[1]=*. Nakon izvršene kalkulacije rezultat se vraća u ćeliju neposredno ispod ulaza. Komentari se pišu u obliku *(* - - - komentar - - - *)*, a sadržaj napisan između zvezdica *(* *)* se ne učitava kao deo programa. Pisanje komentara se preporučuje jer za programera a i za korisnika programa

moгу biti od velike koristi. Komentar obično sadrže smernice ili opis koji olakšava naknadno čitanje programa ili njegove izmene u cilju daljeg razvoja. Tokom formiranja promenljivih posebnu pažnju potrebno je obratiti na velika i mala slova zbog toga što se mnoge ključne reči sastoje upravo samo od velikih slova (E , I , N , D itd.), uostalom sa kojim počinju sve naredbe. Ukoliko sveska sadrži više ćelija, one se redom izvršavaju odozgo nadole nakon što se u meniju izabere opcija *Evaluation*, a potom klikne na *Evaluate Notebook*. Izvršavanje programa iz različitih ćelija neke sveske moguće je pokrenuti proizvoljnim redom, prosto postavljanjem kursora u određenu ćeliju, a potom pritiskom na *shift-enter*. Tokom pisanja programa, ma koliko iskustvo imao korisnik, poželjno znati kako naći definiciju ili opis neke određene naredbe. Postoji više načina za nalaženje definicije proizvoljnog simbola (ili funkcije), a najprostiji je postavljanje znaka $?$ ispred simbola (ili funkcije), što je ilustrovano u narednom primeru.



Klikom na strelicu na kraju objašnjenja otvara se opširnija dokumentacija sa detaljnim objašnjenjima i prostim primerima. Do pomenute dokumentacije moguće je doći i sledećim postupkom. Kursor se postavi na nepoznatu funkciju (simbol) a zatim se pritisne taster $F1$, ili izborom menija *Help* a potom klikom na *Find Selected Function*.

Primeri:

i) Prikazati broj π sa prvih 100 cifara.

```
N[Pi,100]
```

ii) Prikazati broj π sa prvih 100 cifara, ovaj put koristeći grčko slovo $Pi = \pi$.

```
N[Pi,100]
```

iii) Izračunati integral $\int_0^{\infty} x e^x dx$.

```
Integrate[x Exp[-x],{x,0,Infinity}]
```

iv) Izračunati limes funkcije $\lim_{x \rightarrow 0} \frac{\sin x}{x}$.

```
Limit[Sin[x]/x,x->0]
```

v) Nacrtati grafik funkcije čiji je limes tražen u prethodnom zadatku na intervalu $x \in [-5\pi, 5\pi]$.

```
Plot[Sin[x]/x,{x,-5 Pi,5 Pi}]
```

1.2 Promenljive i dodeljivanje vrednosti

Varijable u programskom jeziku Wolfram mogu biti slova ili reči sa velikim ili malim početnim slovom, uključujući grčke simbole. Pridruživanje vrednosti varijabli vrši se uz pomoć znaka $=$. Ukoliko se izvrši ćelija

```
In[3]:= a = 5
```

```
Out[3]= 5
```

izlaz je pridružena vrednost varijable a , dok sa druge strane njenu vrednost pamti jezgro. Pošto program Mathematica razlikuje velika i mala slova, promenljive koje se razlikuju do na veliko ili malo slovo su različite kao na primer:

```
vreme=1;
Vreme=2;
vReme=3;
VREME=4;
Print["vreme=", vreme, " Vreme=", Vreme,
" vReme=", vReme, " Vreme=", VREME]
```

Rešenje: $vreme = 1, Vreme = 2, vReme = 3, VREME = 4$.

Zabrana ispisa vrednosti varijable ili nekog izraza nakon izvršenja programa date ćelije postiže se postavljanjem tačke zarez (;) na kraju linije u kojoj je varijabla definisana. Ugrađene naredbe (funkcije) Matematike počinju velikim slovom. Stoga, da bi se izbegla zabuna, imena promenljivih koje definiše korisnik uputno je da počinju malim slovom. Jednom dodeljena vrednost nekoj promenljivoj, nakon izvršenja ćelije u kojoj je definisana, ostaje na snazi sve dok se toj promenljivoj u nastavku programa (u bilo kojoj ćeliji sveske) ne dodeli druga vrednost ili dok se ona ne ukloni. Praktično je da se vrednost promenljive ukloni čim joj vrednost postaje nepotrebna u nastavku programa. Uklanjanje vrednosti, primera radi gore definisane varijable a radi se na sledeći način

$$a = .$$

ili uz pomoć funkcije *Clear* čiji je argument se briše iz memorije kao

$$\text{Clear}[a].$$

```
In[4]:= a = 5;
```

Varijabla kojoj je pridružena određena vrednost može se upotrebiti u bilo kom izrazu programa koji se razvija. Izvršavanjem programa koji u izrazu sadrži varijablu a , dobija se vrednost izraza u kom je uračunata (zamenjena) njena vrednost.

```
In[5]:= a + 8
a - 2
Out[5]= 13
Out[6]= 3
```

1.3 Neposredno i odloženo dodeljivanje

Neposredno ($a =$) i odloženo ($b :=$) dodeljivanje vrednosti varijablama a i b obrazloženo je sledećim primerom.

```
In[7]:= a = Random[]
b := Random[]
Out[7]= 0.75973
```

Prva linija sadrži zapis $a =$, pri čemu je neposredno varijabli a dodeljen slučajni realni broj u opsegu od 0 do 1. Desna strana ove komande izvršena je klikom na *shift+enter*, pri čemu je generisan slučajni broj pridružen varijabli a i ispisan. Ukoliko se ova ćelija ponovo ne izvrši, vrednost varijable će ostati sve vreme ista i jednaka onoj dobijenoj nakon prvog pokretanja nezavisno od toga u kojoj ćeliji ili u kom izrazu ona bude korišćena. Ovo se može proveriti upitom tražeći vrednost varijable a .

Definicijom $b :=$ vrši se odloženo pridruživanje, što znači da desna strana nije izvršena trenutno, već je deponovana kao definicija varijable b . Od tada, svaki put kada se koristi ova promenljiva, desna strana biće iznova pokrenuta i izvršena, svaki put generišući novi slučajni realan broj između 0 i 1.

In[9]:= ? a
? b

Out[9]=

Symbol
Global`a
Assignment a = 0.75973
Full Name Global`a

Out[10]=

Symbol
Global`b
Assignment b := Random []
Full Name Global`b

Primera radi, ako se izvrši ćelija u kojoj se traži b više puta uzastopno, svaki put iznova se dobije novi slučajni realan broj.

In[6]:= b
b
b

Out[6]= 0.367789

Out[7]= 0.254182

Out[8]= 0.547527

1.4 Vrste zagrada

Programski jezik Wolfram poznaje četiri vrste zagrada:

- Oblim zagradama $()$ grupiše se deo matematičkog izraza kome se daje prednost tokom računanja.

```
100/(8-3)
```

- Unutar uglastih zagrada $[]$ se definišu argumenti funkcije tokom njene konstrukcije ili mesto gde se unose kada je potrebno da se ona pozove.

```
Log[1000]
```

- Vitičarske zagrade rezervisane su za definisanje lista. Tako, primera radi, varijabli lst dodeljena je lista prvih 7 prirodnih brojeva.

```
lst={1,2,3,4,5,6,7}
```

- Dvostrukom uglastom zagradom vrši se ekstrakcija odgovarajućeg elementa liste, koja može biti različitog nivoa složenosti. Unutar dvostruke zagrade pišu se brojevi koji određuju lokaciju traženog elementa.

i) $lst[[i]]$ daje i -ti element liste lst .

ii) $lst[[{i,j,k,...}]]$ daje listu sačinjenu od i -tog, j -tog, k -tog,... elementa liste lst .

iii) $lst[[i,j]]$ daje j -ti element od i -te podliste složene liste.

Primera radi, peti element iz prethodno definisane liste uzima se na sledeći način:

```
lst[[5]]
```

1.5 Prefiksi i dodaci

Postoji nekoliko načina pokretanja funkcija pozivanih u programskom jeziku Wolfram . Primera radi, vrednosti funkcije jednog argumenta $\sin \pi/3$ i $\sqrt{3+9}$ može se računati na sledeći način:

- Standardni način izračunavanja vrednosti funkcije realizuje se pisanjem argumenta u uglastoj zagradi, koja se nalazi na kraju funkcije.

```
In[11]:= Sin[Pi / 3]
          Sqrt[3 + 9]
Out[11]=  $\frac{\sqrt{3}}{2}$ 
Out[12]=  $2\sqrt{3}$ 
```

- Računanje vrednosti funkcije, realizovano prefiks notacijom uz pomoć znaka @ zadaje se tako što se iza napisane funkcije stavlja znak @ nakon čega dolazi vrednost argumenta.

```
In[13]:= Sin@(Pi / 3)
          Sqrt@(2 + 3)
          Sqrt@2 + 3
Out[13]=  $\frac{\sqrt{3}}{2}$ 
Out[14]=  $\sqrt{5}$ 
Out[15]=  $3 + \sqrt{2}$ 
```

Slika 1.1: Prefiks notacija računanja vrednosti funkcije.

- Postfiks notacijom računanje funkcije realizuje se tako što se prvo zada vrednost argumenta, nakon čega sledi znak //, a potom funkcija.

```
In[16]:= (Pi / 3) // Sin
          (2 + 3) // Sqrt
          2 + 3 // Sqrt
Out[16]=  $\frac{\sqrt{3}}{2}$ 
Out[17]=  $\sqrt{5}$ 
Out[18]=  $\sqrt{5}$ 
```

Osim za računanje vrednosti funkcija, u različitim situacijama postfiks notacija se neretko koristi i za transformaciju izlaza.

- Dodavanjem //N na kraju komandne linije, dovešće do prevođenja izlaza rezultata u decimalnu reprezentaciju, naravno ukoliko je moguće.
- Ukoliko se na kraju promenljive, čija je vrednost matrica, doda //MatrixForm, na izlazu biće prikazana matrica u formi tabele.
- Ako se na kraju programa nađe //Timing, po izvršenju programa, biće prikazano ukupno vreme trajanja njegovog izvršavanja do te linije.

- Prikazati Ojlerovu konstantu e u decimalnom obliku koristeći standardni zapis funkcije, prefiks i postfiks notaciju.

```
N[E]
N@E
E // N
```

Rezultat u sva tri slučaja je 2.71828.

1.6 Pravila i zamene

Prilikom rešavanja zadataka, uglavnom tokom simboličkih kalkulacija, koriste se pravila zamene. To je instrukcija koja se u programu realizuje na sledeći način

$$/.x \rightarrow y$$

kojim se svako pojavljivanje simbola (ili obrasca) x menja simbolom ili nekom konkretnom vrednošću y . Transformaciono pravilo se primenjuje na izraze uz pomoć operatora $/.$, koji se piše bez razmaka između $/$ i $.$, ili operatorskom naredbom. *ReplaceAll*.

```
In[5]:= b + 4 /. b -> 10
        ReplaceAll[b + 4, b -> 3]
Out[5]= 14
Out[6]= 7
```

Strelica koja se koristi za zamenu varijable datom vrednošću tokom pisanja u Matematiki – \rightarrow (minus+znak veće bez razmaka između) automatski se transformiše u specijalni karakter \rightarrow . Prednost zemene u odnosu na pridruživanje vrednosti varijabli x je u tome da se zamena odnosi samo na specifičan izraz, pri čemu ova varijabla u nastavku programa i dalje može biti formalno korišćena kao opšta.

Pravila zamene mogu da sadrže obrasce, što se realizuje kao u sledećem primeru.

```
In[7]:= a + b /. x_ -> x^2
Out[7]= (a + b)^2
```

Treba primetiti da se obrazac $x_$ odnosi na ceo izraz $a + b$ a ne samo za pojedinačne podizraze a i b . Ukoliko je neophodno biti više specifičan i izvršiti zamenu na dubljem nivou (nivou 1 ovog izraza) ili izvršiti zamenu celog izraza na nivou 0, to se realizuje na sledeći način.

```
In[8]:= Replace[a + b, x_ -> x^2, {1}]
Out[8]= a^2 + b^2
In[9]:= Replace[a + b, x_ -> x^2, {0}]
Out[9]= (a + b)^2
```

Da bi se postigao željeni rezultat moguće je implementirati ograničene obrasce. Ovim se u izrazu, po datom pravilu menjaju isključivo oni njegovi delovi koji zadovoljavaju postavljeni uslov, dok ostali delovi ostaju nepromenjeni.

```
In[10]:= a + 3 /. x_Integer -> x^2
Out[10]= 9 + a
```

Mnoge funkcije u jeziku Wolfram kao što su *Solve*, *DSolve*, *NSolve*, ... kao rezultat daju pravila zamene. Tako na primer, rešenje jednačine $x^2 - 9 = 0$ uz pomoć funkcije *Solve* je lista pravila. Upotrebom operatora zamene $/.$ moguće je proveriti ili koristiti na izlazu dobijena rešenja.

```
In[3]:= sol = Solve[x^2 - 9 == 0, x]
Out[3]= {{x -> -3}, {x -> 3}}

In[4]:= x^2 - 9 /. sol
Out[4]= {0, 0}
```

Kao kod dodeljivanja vrednosti varijablama i pravila koja se izvršavaju operatorima zamene mogu biti neposredno $x \rightarrow y$, tako što se vrednost y izračunava jednom definisanjem pravila. Tokom odloženog pravila, koje se tehnički realizuje kao $x : - > y$, vrednost y se ponovo računa svaki put kada se pravilo primenjuje. Ovo je posebno važno kada pravilo treba da izvrši radnju.

```
In[1]:= {a, d, c, l, c, r, c} /. c -> Random[]
          {a, d, c, l, c, r, c} /. c -> Random[]
Out[1]= {a, d, 0.000812323, l,
          0.000812323, r, 0.000812323}

Out[2]= {a, d, 0.83074, l, 0.729088, r, 0.645449}
```

Ukoliko, kao u prethodno prikazanom primeru, u listi varijabli c se menja slučajnim realnim brojem između 0 i 1, primenom trenutne zamene $sa \rightarrow$ generator slučajnih brojeva se poziva jednom i sve varijable c date liste se menjaju generisanim slučajnim brojem. Odloženom zamenom $: - > random$ generator se evoluira onoliko broj puta koliko se c javlja u listi i svaka od promenljivih c se menja novim generisanim slučajnim brojem. Ako se prilikom primene operatora $/.$ javi lista pravila zamene, najpre se izvršava prvo među njima. Sa druge strane, pravila zadata listom primenjivače se uzastopno ukoliko je zadan operator $//.$ sve dok se ne izvrši i poslednji element liste zamena.

```
In[11]:= a /. {a -> b, b -> c, c -> d}
          a //. {a -> b, b -> c, c -> d}
Out[11]= b

Out[12]= d
```

1.7 Definisane funkcije

Umesto očekivane definicije $ImeFunkcije[argument_1, argument_2, \dots] := IzrazFunkcije$, funkciju je ispravno definisati na sledeći način $ImeFunkcije[argument_1, argument_2, \dots] := IzrazFunkcije$. Izostanak ovakve definicije ilustrovan je sledećim primerom funkcije $f(x) = x^2 - 1$.

```
In[1]:= f[x] := x^2 - 1;
          f[x]
          f[2]
          f[y]
Out[2]= -1 + x^2

Out[3]= f[2]

Out[4]= f[y]

In[5]:= f[x_] := x^2 - 1;
          f[x]
          f[2]
          f[y]
Out[6]= -1 + x^2

Out[7]= 3

Out[8]= -1 + y^2
```

Znak $_$ označava bilo koji izraz, tako da simbol $argument_$ znači bilo koji izraz koji se imenuje kao $argument$ u telu funkcije. Funkcija se pored operatora za odloženo definisanje ($:=$), to jeste kada se poziva, može definisati i upotrebom operatora neposrednog pridruživanja ($=$) ukoliko je potrebna neposredna evolucija funkcije. Dobra je praksa ukloniti sve nepotrebne definicije funkcije što je pre moguće. Neposredno definisanje funkcije $g(x) = \frac{\sin x}{x}$ vrši se na sledećin način, sa vrednostima u o i 1.

```

In[9]:= g[x_] = Sin[x] / x;
        g[1.]
        g[0]

Out[10]= 0.841471

Power: Infinite expression  $\frac{1}{0}$  encountered. ⓘ
Infinity: Indeterminate expression 0 ComplexInfinity
encountered. ⓘ

Out[11]= Indeterminate

```

Očigledno da nije moguće izračunati vrednost zadate funkcije za $x = 0$, što se može korigovati njenim dodatnim definisanjem u toj vrednosti promenljive. Treba primetiti da definisanje vrednosti funkcije za konkretnu promenljivu ne iziskuje dodavanje donje crtice sa leve strane, te stoga nema definicije obrasca. Definicija funkcije ispitana je postavljanjem operatora `?g` ispred njenog naziva, nakon čega se ispisuju puna definicija funkcije. Pozivanjem funkcije `g` da izračuna zadatu vrednost, njene definicije se proveravaju po ispisanom redosledu pojavljivanja. Na primer, ako se traži `g[0]`, onda se prvi uslov poklapa i vraća se vrednost `1`, dok traženjem vrednosti `g[0.3]`, prvi uslov se ne poklapa, te se izvršava onaj preostali.

```

In[12]:= g[0] = 1;
         ?g

Out[13]=

```

Symbol
Global`g
Definitions
<code>g[0] = 1</code>
<code>g[x_] = $\frac{\text{Sin}[x]}{x}$</code>
Full Name Global`g

Poput odloženog pridruživanja vrednosti varijablama, slično se može definisati funkcija sa odloženim pozivom. Sledeći primeri ilustruju rad funkcija sa odloženim i neposrednim pozivom.

```

In[5]:= f1[x_] = x + Random[];
        f2[x_] := x + Random[];
        {f1[3], f2[3]}
        {f1[3], f2[3]}

Out[7]= {3.2424, 3.39168}

Out[8]= {3.2424, 3.23256}

```

Funkcija f_1 je izvršena nakon pokretanja programa i ima specifičnu vrednost slučajnog broja koji je dodeljen evolucijom, dok se f_2 svaki put iznova izvršava u potpunosti uključujući i evoluciju slučajnog broja. Ukoliko nova isprogramirana funkcija zahteva dugo vreme izvršavanja, moguće je jednom evoluirane vrednosti upamtiti, a potom ih koristiti bez novog izvršavanja funkcije ukoliko se ponovo pozove ta vrednost funkcije. Nije potrebno evoluirati funkciju iznova jer je zapamćen već izračunat rezultat. Da bi se konstruisala takva funkcija koja pamti svoje vrednosti iz prethodnih poziva koristi se interakcija između obrazaca i vrednosti, kao i između neposrednih i odloženih dodela.

```

In[1]:= F[x_] := F[x] = x^3
? F
Out[2]=
Symbol
Global`F
Definitions
F[x_] := F[x] = x^3
Full Name Global`F

In[3]:= F[3]
? F
Out[3]= 27
Out[4]=
Symbol
Global`F
Definitions
F[3] = 27
F[x_] := F[x] = x^3

```

Rezultati funkcije čija evolucija traje dugo, mogu se deponovati u fajl, a potom kasnije ponovo čitati i tako skratiti vreme njenog budućeg korišćenja.

Funkcije sa uslovnim i opcionim argumentima

U programskom jeziku Wolfram moguće je koristiti šablone kojima se definišu funkcije koje prihvataju određene argumente. Postoji više tipova obrazaca ovih funkcija.

- a) *Sve prolazi*, je funkcija koja se može pozvati sa bilo kojom vrstom argumenta pošto obrazac $x_.$ može odgovarati bilo čemu.

```

In[1]:= f[x_] := x^2
{f[4], f[3 - 2 I], f[{1, 2, 3, 4}], f[y^3]}
Out[2]= {16, 5 - 12 i, {1, 4, 9, 16}, y^6}

```

- b) *Funkcije ograničene po tipu*, gde obrazac kao što je x . *Integer* odgovara samo argumentima celobrojnog tipa. Ako je funkcija pozavana sa nepodudarnim argumentom tada se ona ne izvršava.

```

In[3]:= g[x_Integer] := x + 7
{g[2], g[y], g[3.4]}
Out[4]= {9, g[y], g[3.4]}

```

- c) *Uslovna funkcija*, pri čemu se komplikovani uslovi mogu pozivati operatorom $/;$

```

In[5]:= h[x_ /; x <= 4] := x^2
h[x_ /; x > 4] := x + 9
{h[3], h[5]}
Out[7]= {9, 14}

```

1.8 Relacije i logički operatori

Relacioni operatori upoređuju dva izraza i vraćaju *True* ili *False* kad god je poređenje moguće. Matematika ima nekoliko relacionih operatora ili funkcija kojima je moguće vršiti upoređivanje

- i) $lhs == rhs$ ili *Equal*[lhs, rhs] daje *True* ako su izrazi lhs i rhs jednaki.
- ii) $lhs != rhs$ ili *Unequal*[lhs, rhs] daje *False* ako su lhs i rhs identični.
- iii) $lhs > rhs$ ili *Greater*[lhs, rhs] daje *True* ako je lhs veće od rhs .

- iv) $lhs < rhs$ ili $Less[lhs, rhs]$ daje $True$ ukoliko je lhs manje od rhs .
- v) $lhs \geq rhs$ ili $GreaterEqual[lhs, rhs]$ daje $True$ ukoliko je lhs veće ili jednako rhs .
- vi) $lhs \leq rhs$ daje $True$ ukoliko je lhs manje ili jednako rhs .

```
In[1]:= 8 - 3 == 5
```

```
4 > 1 ≤ 2
```

```
Out[1]= True
```

```
Out[2]= True
```

Matematika podržava logičke operatore ili funkcije. Ukoliko nije u mogućnosti da uporedi izraze, vraća postavljene ulaze.

```
In[3]:= 1 + 7 x < 3 z
```

```
Out[3]= 1 + 7 x < 3 z
```

- i) $!expr$ ili $Not[expr]$ daje $False$ ukoliko je iskaz $expr$ tačan, a $True$ ukoliko je netačan.
- ii) Sa $expr_1 \&\&expr_2 \&\& \dots$ ili $And[expr_1, expr_2, \dots]$ evoluiraju se izrazi sleva na desno, a rezultat je $False$ ukoliko je neki od izraza netačan ili $True$ ukoliko su svi tačni.
- iii) Sa $expr_1 || expr_2 || \dots$ ili $Or[expr_1, expr_2, \dots]$ argumenti se evoluiraju sleva na desno, a rezultat je $True$ ukoliko je neki od izraza tačan, a $False$ ukoliko su svi netačni.
- iv) $Xor[expr_1, expr_2, \dots]$ daje $True$ ukoliko je neparan broj izraza $expr_i$ tačan a ostali netačni, ili je izlaz $False$ ukoliko je paran broj izraza $expr_i$ tačno dok su ostali netačni.

```
In[4]:= 4 > 3 || 10 ≥ 5
```

```
Out[4]= True
```

```
In[5]:= Xor[4 > 3, 10 ≥ 5, 2 > 5]
```

```
Out[5]= False
```

Operacija $||$ (ili) je inkluzivna dok je Xor isključiva. Drugim rečima, $||$ daje $True$ kada je jedan ili su oba izraza tačna, dok Xor daje $True$ samo kada je jedan izraz u prethodnom primeru tačan a ostali netačni. Izraz $expr_1 \&\&expr_2 \&\& \dots$ Matematika evoluirira sleva na desno i zaustavlja se kada prvi put naiđe na netačan izraz dajući $False$. Ukoliko Matematika ne može utvrditi tačnost iskaza, izlaz je jednak ulazu.

```
In[6]:= a || b || c
```

```
Out[6]= a || b || c
```

Upotreba logičkih operatora biće uvežbana na sledećem primeru. Za česticu mase m , koja se kreće u jednodimenzionalnoj potencijalnoj jami sa nepropusnim zidovima postavljenim u $x = 0$ i $x = a$, svojstvene energije i odgovarajuće svojstvene funkcije se mogu zapisati u obliku:

$$E = \frac{\hbar^2 k^2}{2m}$$

$$\phi(x) = A \sin kx + B \cos kx,$$

gde je \hbar redukovana Plankova konstanta. Granični uslovi su:

$$\phi(0) = 0$$

$$\phi(a) = 0$$

Vrednosti od k i A biće diskutovane uz pomoć sledećeg programa.

```
\[Phi][x_] := A*Sin[k*x] + B*Cos[k*x];
Reduce[{\[Phi][0] == 0, \[Phi][a] == 0, a != 0}, {A, k}]
```

$$B = 0 \wedge \left((a \neq 0 \wedge A = 0) \vee \left(c_1 \in Z \wedge a \neq 0 \wedge \left(k = \frac{2\pi c_1}{a} \vee k = \frac{\pi + 2\pi c_1}{a} \right) \right) \right)$$

Funkcija `Reduce[eqns, vars]` koristi se da bi se dobila potpuna informacija o rešenju, pri čemu se jednačina `eqns` redukovana uz zadržavanje svih mogućih rešenja varijabli `vars`. Rezultat je izraz koji sadrži logičke operatore po opadajućem prioritetu: `==`, `!=`, `∈`, `∉`, `&&` `||`. Iz rezultata se vidi da je netrivialno rešenje $A \neq 0$ i $k_n = n\pi/a$.

1.9 Kompleksni brojevi

Ulazi svih matematičkih funkcija u Wolfram programskom jeziku mogu imati vrednost kompleksnog broja. Najčešće primenljive funkcije na kompleksne brojeve su

- i) `Conjugate[z]` rezultat ove funkcije je konjugovana vrednost kompleksnog broja z ,
- ii) funkcija `Re[z]` daje realni deo kompleksnog broja z ,
- iii) `Im[z]` daje imaginarni deo kompleksnog broja z ,
- iv) `Abs[z]` daje apsolutnu vrednost realnog ili kompleksnog broja z ,
- v) `Arg[z]` daje argument kompleksnog broja z .

```
In[1]:= z = 4 + 7 I;
Conjugate[z]
Im[z]
Re[z]
Abs[z]
Arg[z]
```

```
Out[2]= 4 - 7 i
```

```
Out[3]= 7
```

```
Out[4]= 4
```

```
Out[5]=  $\sqrt{65}$ 
```

```
Out[6]= ArcTan[ $\frac{7}{4}$ ]
```

Primenom na promenljive, kojima nije dodeljena konkretna vrednost, gore navedene funkcije se ne izvršavaju pošto i promenljive mogu biti kompleksni brojevi.

```
In[7]:= z0 = x + I * y
Conjugate[z0]
Im[z0]
```

```
Out[7]= x + i y
```

```
Out[8]= Conjugate[x] - i Conjugate[y]
```

```
Out[9]= Im[x] + Re[y]
```

Ukoliko varijablama nisu dodeljene konkretne vrednosti za *Matematiku* varijable x i y ostaju neodređene prema tome da li su realni ili kompleksni brojevi. Tek kada je izvesno da će im se dodeliti realni brojevi, na ove funkcije može se primeniti `ComplexExpand`, čime se pretpostavlja da su sve varijable realne.

```
In[10]:= ComplexExpand[Conjugate[z0]]
ComplexExpand[Im[z0]]
```

```
Out[10]= x - i y
```

```
Out[11]= y
```

Drugi način je da se lokalno eksplicitno pretpostavi, što može biti više specifično nego pretpostavljati da su sve varijable realni brojevi.

```
In[12]:= Assuming[Element[x, Reals] &&
Element[y, Reals],
FullSimplify[Conjugate[z0]]]
```

```
Out[12]= x - i y
```

1.10 Programske konstrukcije

Postoji više različitih modela programiranja u programskom jeziku Wolfram, pri čemu se može koristiti svaki od njih ponaosob, ali takođe mogu biti kombinovani i primenjeni istovremeno različiti koncepti. Od primenjenog stila programiranja zavisi efikasnost izvršenja napisanog programa prilikom rešavanja određenog zadatka. Osnovna podela metoda programiranja je na proceduralno i funkcionalno.

Proceduralno programiranje

Deo programskog jezika Wolfram veoma je sličan jezicima C, Python, Java ili drugim proceduralnim programskim jezicima. Primenjujući ovaj metod neophodno je razlikovati kolone, koje razdvajaju komande u blokove koda, od zareza koji razdvajaju različite blokove.

Konstrukcija petlji u programskom jeziku Wolfram slična je kao i kod drugih.

```
In[19]:= For[i = 3, i ≤ 7, i = i + 1, Print[i]]
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
In[20]:= ? i
```

```
Out[20]=
Symbol
Global`i
Assignment
i = 8
```

Slika 1.2: For petlja kojom se vrši ispis prirodnih brojeva od 3 do 7, sa definicijom varijable i .

Brojač For petlje i je globalno definisana varijabla, čija je vrednost u gore datom primeru nakon izvršenja programa $i = 8$. Osim u petlji For, brojač je kao globalna varijabla definisan i u petlji While.

Brojač While petlje definisan je globalno, što se vidi u prethodnom primeru, gde nakon izvršenja programa i zaustavljanja petlje on ima vrednost $k = 6$.

Funkcijom Do ponavlja se izvršavanje neke procedure onoliko puta koliko je to zadato iterativnom konstrukcijom. Brojač petlje Do nije definisan globalno, stoga je on lokalna varijabla.

```
In[21]:= k = 2;
While[k ≤ 5,
  Print[k];
  k = k + 1]
2
3
4
5
```

```
In[23]:= ? k
```

Symbol

Global`k

Assignment

k = 6

Full Name Global`k

^

```
Out[23]=
```

Slika 1.3: *While* petlja kojom se ispisuju prirodni brojevi od 2 do 5 (levo). Definicija brojača u *While* petlji.

```
In[24]:= s = 0;
Do[s = s + j;
  Print[j], {j, 1, 3}];
Print["s=", s]
1
2
3
s=6
```

```
In[27]:= ? j
```

Symbol

Global`j

Full Name Global`j

^

```
Out[27]=
```

Slika 1.4: *Do* petlja koja ispisuje sve brojeve od 1 do 3 a potom i njihovu sumu.

U datom primeru za *Do* petlju brojač j uzima vrednosti od 1 do 3. Za svaku vrednost lokalne varijable j vrši se njen ispis, a istovremeno se vrši i sumiranje ovih brojeva. Po završetku petlje ispisana je dobijena suma svih vrednosti od j .

Nakon izvršenja dela programa u kom se nalazi *Do* petlja, njen brojač ostaje lokalno definisan, što znači da mu nakon izvršenja programske linije nije dodeljena konkretna vrednost.

Uslovne naredbe daju vrednost jednog od nekoliko alternativnih izraza ukoliko je ispunjen postavljeni uslov. Uslovne funkcije u programu Matematika su *If*, *Which*, *Switch* i *Piecewise*. Uslovna naredba

$$\text{If}[\text{condition}, t, f]$$

je slična kao i kod većine programsikih jezika, pri čemu je rezultat t ukoliko je ispunjen uslov na mestu varijable condition , a u protivnom rezultat je f .

```
In[28]:= a = "g";
If[a > 100, Print["vece"],
  Print["manje"], Print["?"]];
?
```

Ukoliko je naredba zadata kao

$$\text{If}[\text{condition}, t],$$

rezultat je t ukoliko je uslov ispunjen, a ukoliko nije izlaz je $Null$. Izvršavanje naredbe

$$If[condition, t, f, u]$$

biće u ukoliko uslov nije ni tačan niti netačan.

```
In[30]:= a = 3;
         If[a > 1000, Print["tacno"]]
```

Uslovnom naredbom

$$Which[condition_1, value_1, condition_2, value_2, ...]$$

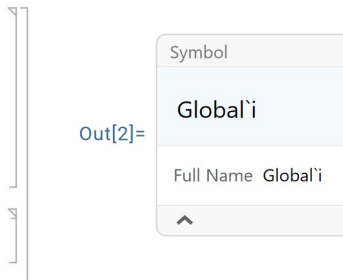
redom se razmatraju uslovi $condition_1, condition_2, \dots$ a izbacuje se ona vrednost ponuđena nakon prvog tačnog uslova u datom nizu. Izvršavanjem

$$Switch[expr, form_1, value_1, form_2, ...]$$

izvršava izraz pod varijablom $expr$ i upoređuje sa svakom $form_i$, a čim se nađe podudaranje sa prvom $form_i$ kao izlaz dobija se odgovarajuća vrednost $value_i$.

U okviru naredbe *Module* programski kod koristi lokalne varijable.

```
In[1]:= Module[{i},
           i = 1;
           While[i > 1 / 200, i = i / 2];
           i]
Out[1]= 1/256
```



Out[2]=

Symbol
Global`i
Full Name Global`i

Izvršavanjem programa pod okruženjem *Module*, premda korišćen u petlji *While* kao brojač, i je u globalnom kontekstu nedefinisan.

- i) Napisati program kojim se sumiraju svi celi brojevi od 123 do 9968, koristeći jedino lokalne varijable.

```
Module[{sum,i},
sum=0;
i=123;
While[i<=9968,
sum=sum+i;
i=i+1];
sum]
```

Rezultat: 49 677 993. Uprkos tome što je brojač i petlje *While*, zajedno sa promenljivom sum ostaje lokalna varijable zbog izvršenja programa u okviru *Module*.

- ii) Napisati program koji sumira uzastopne cele brojeve počevši od 123 sve dok suma ne postane veća od 10 000. Tada ispisati najveći ceo broj u ovoj sumi. Zadatak rešiti koristeći samo lokalne varijable.

```
Module[{i,sum},
i=123;
sum = 0;
While[sum+i<10000,
sum=sum+i;
i=i+1];
i-1]
```

Rešenje: 186.

Funkcionalno programiranje

Funkcionalno programiranje je veoma važna i korisna tehnika, karakteristična za Wolfram -ov jezik, koja zbog mogućnosti paralelizovanja i raspodele posla na više procesora može dati veliko ubrzanje tokom izvršavanja programa. U ovom kontekstu obično se misli na rad sa listama (vekotrima ili matricama) pri čemu se specifične funkcije primenjuju na svaki njihov element. Najčešće konstrukcije funkcionalnog programiranja su: obične matematičke ili bezimene funkcije, mapiranje na listu sa *Map* ili primena na listu sa *Apply*.

Bezimene funkcije se mogu brzo definisati uz pomoć parametara #1,#2,#3,... a završavaju se simbolom & . Ilustracija programiranja i izvršavanja bezimene funkcije biće realizovana kreiranjem kvadratne funkcije $f(x) = x^2$ uz pomoć funkcionalnog programiranja.

```
In[1]:= f = #^2 &;
        f[7]
Out[2]= 49
```

U drugom primeru isprogramirana je funkcija dve promenljive $g(x,y) = x - y$.

```
g=#1-#2 &;
g[100, 88]
```

Rešenje: 12.

Matematičke kao i bezimene funkcije su objekti tzv. prve klase poput brojeva ili matrica. Mogu se dodeliti promenljivim kao u prethodna dva primera. Takođe se mogu koristiti direktno kao argumenti drugih funkcija, što je ilustrovano narednim primerom.

```
l={1, 2, 3, ##, 4, 5, 6} &;
l[10, a, b]
```

Rešenje: {1,2,3,10,a,b,4,5,6}.

ovde se jasno vidi da simbol ## predstavlja niz parametara funkcije. Simbol #0 označava samu funkciju, što može biti korisno kod definisanja rekurzivne bezimene funkcije.

Sa *Map* ili */@* realizuje se dejstvo funkcije na svaki element zadate liste pojedinačno. Narednim primerom, konstrukcijom liste kvadrata elemenata iz prethodno zadate, biće ilustrovano dejstvo oba operatora. Treba primetiti da je upotreba bezimene funkcije #^2& moguća iako joj nije

```
In[1]:= lst = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        Map[#^3 &, lst]
Out[2]= {1, 8, 27, 64, 125, 216, 343, 512, 729, 1000}

In[3]:= #^3 & /@ lst
Out[3]= {1, 8, 27, 64, 125, 216, 343, 512, 729, 1000}
```

pridruženo ime.

Naredbom *Apply* ili njoj odgovarajućim simbolom @@ funkcija ili operacija deluje na sve elemente liste generišući jedan rezultat. Tako, sa *Plus@@lista* ili *Apply[Plus,lista]* rezultat je suma svih elemenata od *lista*, dok sa *Times@@lista* ili *Apply[Times,lista]* rezultat je jedan broj jednak

```

In[1]:= lst = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        Apply[Plus, lst]
Out[2]= 55

In[3]:= Plus @@ lst
Out[3]= 55

In[4]:= Apply[Times, lst]
Out[4]= 3 628 800

In[5]:= Times @@ lst
Out[5]= 3 628 800

```

proizvodu svih elemenata liste. Ovo će u narednom primeru biti demonstrirano primenom operacije sabiranja i množenja na listu prirodnih brojeva. Može se reći da operacija *Apply* redukuje listu brojeva u jedan.

Primeri za vežbu:

- i) Napisati bezimenu funkciju sa tri argumenta čijim se izvršavanjem dobija njihov proizvod.

```

f3=#1*#2*#3 &;
f3[x, y, z]
f3[2, 5, 8]

```

Rešenje: xyz i 80.

Napomena. Množenje $x * y$, pored alternativne forme operatorom $*$ ili eksplicitne koristeći *Times*[$x, y,$] može se realizovati razdvajanjem faktora tasterom *space*.

```

In[2]:= 4 × 7
Out[2]= 28

```

- ii) Izračunati vrednost funkcije $\sin(x^2)$ svakog elementa liste $a = \{0.1, 0.9, 2.25, -1.9\}$.

```

a={0.1,0.9,2.25,-1.9};
sma=Map[Sin[#^2] &, a]
Sin[#^2] & /@ a

```

Rešenje: $\{0.00999983, 0.724287, -0.939335, -0.451466\}$.

- iii) Izračunati sumu elemenata liste dobijene kao rezultat prethodnog zadatka.

```

Plus@@sma

```

Rešenje: -0.656513.

1.11 Vektori, matrice i tenzori

Kolone i matrice su često upotrebljavani objekti prilikom rešavanja zadataka iz fizike, kojima se na primer reprezentuju stanja i operatori u primerima iz kvantne mehanike.

Vektori

Vektor reprezentovan kolonom u programskom jeziku Wolfram zadaje se u vidu liste objekata, realnih ili kompleksnih brojeva. Broj elemenata vektora (kolone) v ili dužina liste nalazi se primenom naredbe

$$\text{Length}[v].$$

```
In[1]:= v = {1, 2, 3, 2 + I, 7};
      Length[v]
Out[2]= 5
```

Pristup bilo kom elementu liste, prema njemu odgovarajućem indeksu vrši se dvostrukom uglastom zagradom. Prvi element liste v ima indeks 1 ($v[[1]]$ je prvi element od liste v). Zadavanje negativnog indeksa brojanje počinje sa kraja liste, tako je $v[[-1]]$ poslednji element zadate liste.

```
In[3]:= v[[-2]]
      v[[4]]
Out[3]= 2 + i
Out[4]= 2 + i
```

Elementi lista (skupova) mogu da budu različiti objekti, kao na primer: stringovi, grafici, izrazi, nove liste (podliste), funkcije ili brojevi. Za vektore a i b , liste iste dužine čiji su elementi realni ili kompleksni brojevi, standardni skalarni proizvod računa se uz pomoć operatora *Dot* odnosno sa $.$ kao

$$a.b \text{ ili } \text{Conjugate}[a].b \quad (1.1)$$

redom ukoliko je prostor nad realnim odnosno poljem kompleksnih brojeva.

```
In[1]:= a = {1, I, 3 + I, 0};
      b = {5, 6, 7 + I, 9};
      Conjugate[a].b
Out[3]= 27 - 10 i
```

Vektori istih dužina mogu da se zbrajaju, oduzimaju, množe ili dele primenom standardnih operacija.

```
In[4]:= a + b
      a - b
      a * b
      a / b
Out[4]= {6, 6 + i, 10 + 2 i, 9}
Out[5]= {-4, -6 + i, -4, -9}
Out[6]= {5, 6 i, 20 + 10 i, 0}
Out[7]= {1/5, i/6, 11/25 + 2i/25, 0}
```

Pored standardnih računskih operacija primenjenih na dva vektora, množenje skalarom se izvrši tako da se svaka komponenta njime multiplikuje. Ako je vektor argument neke matematičke funkcije, rezultat je vektor čije su komponente vrednost funkcije odgovarajuće komponente ulaznog argumenta.

```
In[8]:= 3 a
         a^2
Out[8]= {3, 3 i, 9 + 3 i, 0}
Out[9]= {1, -1, 8 + 6 i, 0}
```

Matrice

Struktura matrica u programskom jeziku Wolfram je u skladu sa njihovom definicijom da su to dvodimenzionalni skupovi brojeva (simbola) soritirani u pravougaoni oblik vertikalnih i horizontalnih linija koje čine kolone odnosno vrste. U programu Mathematica matrice se zadaju kao skupovi (liste) čiji su elementi podliste, koje su zapravo vrste matrice. Sve podliste su iste dužine, a ukoliko je ona različita od broja podlista reč je o pravougaonoj matrici. Format matrice A_{mn} traži se naredbom

$$\text{Dimensions}[A_{mn}],$$

čiji je argument naredbe matrica a izlaz je lista elemenata koji predstavljaju broj vrsta i kolona $\text{Dimensions}[A_{mn}] = \{m, n\}$.

```
In[1]:= Mat = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}};
         Dimensions[Mat]
Out[2]= {3, 4}
```

Napisana u liniji programa, matrica ima strukturu liste sa podlistama brojeva a njena struktura nije uvek očigledna posebno ako je velikog formata. Prikazivanje matrice A_{mn} u klasičnom matematičkom obliku vrši se uz pomoć naredbe

$$\text{MatrixForm}[A_{mn}].$$

```
In[3]:= MatrixForm[Mat]
Out[3]//MatrixForm=
  ( 1  2  3  4
   5  6  7  8
   9 10 11 12 )
```

Pristup elementima matrice analogna je kao kod liste, u par otvorenih i zatvorenih uglastih zagrada upisuju se indeksi elementa. Osim elemenata, mogu se izdvajati i pojedine vrste, direktno zadajući jedan indeks, što se svodi na izdvajanje podliste kao elementa cele složene liste koja ima funkciju matrice.

```
In[4]:= Mat[[3, 4]]
         Mat[[2]]
Out[4]= 12
Out[5]= {5, 6, 7, 8}
```

Transponovanje matrice $A_{mn}^T = [a_{ij}]_{mn}^T = [a_{ji}]_{nm}$ vrši se naredbom

$$\text{Transpose}[A_{mn}].$$

Množenje matrica kao i množenje vektora matricom realizuje se opeacijom *Dot*. Množenje matrica ili matrica i vektora je moguće uz uslov da su matrice $A_{mn} \cdot B_{nl}$ odnosno matrica i vektor $A_{mn} \cdot v_n$, $v_n \in F^n$ ($F = \mathbb{R}, \mathbb{C}$) odgovarajućih dimenzija.

```

In[6]:= Transpose[Mat]
        MatrixForm[%]

Out[6]= {{1, 5, 9}, {2, 6, 10}, {3, 7, 11}, {4, 8, 12}}

Out[7]//MatrixForm=

$$\begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix}$$


In[8]:= v = {a, b, c, d};
        Mat.v
        Mat.Transpose[Mat]

Out[9]= {a + 2 b + 3 c + 4 d,
        5 a + 6 b + 7 c + 8 d, 9 a + 10 b + 11 c + 12 d}

Out[10]= {{30, 70, 110}, {70, 174, 278}, {110, 278, 446}}

```

Matematika poseduje funkciju kojom se kreira jedinična matrica formata $n \times n$, koja na dijagonali ima n jedinica, dok su svi ostali elementi nule

IdentityMatrix[n].

```

IdentityMatrix[4]
MatrixForm[%]

```

Rešenje:
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Dijagonalna matrica zadatih dijagonalnih elemenata listom *list* kreira se naredbom

DiagonalMatrix[*list*].

```

DiagonalMatrix[{1, 2, 3, 4}]
MatrixForm[%]

```

Rešenje:
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}.$$

Dijagonalne matrice pogodne su radi efikasnog izvršavanja računarskih operacija sa njima, a o postupku dijagonalizacije matrica uz pomoć funkcija iz Wolfram Mathematica biće više diskusije u nastavku.


Retke matrice i vektori

Retke ili često korišćen naziv *sparse* matrice ili vektori su skupovi brojeva kod kojih je većina elemenata nula. Ne postoji striktna definicija u kom odnosu stoje brojevi nultih i nenultih elemenata pa da se ta matrica kvalifikuje kao retka. One se pojavljuju u teoriji grafova i numeričkoj analizi, tipično u oblastima gde postoji mala gustina relevantnih podataka i veza. Istraživački računi obično operišu sa retkim matricama ili vektorima velikog formata odnosno dimenzije. Prednost njihove upotrebe je što se tokom ovih kalkulacija koriste specijalizovani adaptirani algoritmi te se sa njima lako barata, a osim toga moguće ih je efikasnije kompresovati i upisivati u memoriju. Za velike formate retkih matrica standardni algoritmi, korišćeni za guste matrice postaju neupotrebljivi jer velike matrice uzimaju veliki

deo računarske memorije. Efikasan način čuvanja retkih matrica je formiranje liste samo nenultih elemenata, naravno uz poznavanje njihovih pozicija. *Sparse* matrica M se zbog toga u programu Mathematica zadaju funkcijom


$$\text{SparseArray}[M].$$

```
In[1]:= M = { {0, 0, 0, 0, 0, 9, 0, 0, 0},
              {2, 0, 0, 0, 0, 0, 0, 3, 0},
              {0, 0, 0, 5, 0, 0, 0, 0, 0} };
Ms = SparseArray[M]
```

```
Out[2]= SparseArray [  Specified elements: 4
Dimensions: {3, 9}
Default: 0
Density: 0.148
Elements:
{1, 6} → 9
{2, 1} → 2
{2, 8} → 3
{3, 4} → 5 ]
```

Izlaz ove funkcije definiše zadatu retku matricu, a opis sadrži format matrice, odnosno broj vrsta i kolona, zatim broj nenultih elmenata, njihovu udeo u ukupnom broju i poziciju u vidu indeksa svakog nenultog elementa. Svaka *sparse* matrica može biti definisana uz pomoć navedene funkcije *SparseArray* tako što se kao njeni argumenti unesu nenulti elementi a potom i format. Ovako zadata matrica prevodi se u standardni oblik naredbom *Normal*. Retke matrice i vektori mogu biti korišćeni kao i gusti objekti, a lako se konvertuju po potrebi. Međutim, za neke operacije linearne algebre, kao na primer množenje dve retke matrice, operiše se samo sa nenultim elementima, što se izvršava neuporedivo brže nego standardno množenje. Pored optimizovanja izvršenja operacije zadavanje i memorisanje u *sparse* obliku je računarski isplativije.

```
In[3]:= Msnew = SparseArray [
          { {1, 6} → 9, {2, 1} → 2, {2, 8} → 3, {3, 4} → 5 },
          {3, 9} ]
Normal [Msnew]
```

```
Out[3]= SparseArray [  Specified elements: 4
Dimensions: {3, 9}
Default: 0
Density: 0.148
Elements:
{1, 6} → 9
{2, 1} → 2
{2, 8} → 3
{3, 4} → 5 ]
```

```
Out[4]= { {0, 0, 0, 0, 0, 9, 0, 0, 0},
          {2, 0, 0, 0, 0, 0, 0, 3, 0},
          {0, 0, 0, 5, 0, 0, 0, 0, 0} }
```

Dijagonalizacija matrice

Rešavanje vremenski nezavisne Šredingerove jednačine svodi se na rešavanje svojstvenog problema, odnosno nalaženje svojstvenog vektora i svojstvenih vrednosti Hamiltonovog operatora. Svojstveni problem se nalazi u mnogim oblastima primene linearne algebre. Nenulti vektor v je svojstveni za

operator H ako važi da je

$$Hv = \lambda v,$$

pri čemu je skalar λ odgovarajuća svojstvena vrednost. Primera radi biće rešen svojstveni problem hermitske matrice

$$H = \begin{pmatrix} 0 & 0.3 & i & 0 \\ 0.3 & 1 & 0 & 0 \\ -i & 0 & 1 & -0.2 \\ 0 & 0 & -0.2 & 3 \end{pmatrix}.$$

Izvršavanjem naredbe

$$\text{Eigenvalues}[H]$$

dobija se spektar σ_H kvadratne matrice H .

```
H={
  {0,0.3,I,0},
  {0.3,1,0,0},
  {-I,0,1,-0.2},
  {0,0,-0.2,3}};
\Sigma=Eigenvalues[H]; (*spektar*)
Sort[\Sigma](*sortiranje liste svojstvenih vred.*)
```

Računanje spektra zadate matrice naredbom *Eigenvalues* vrši se numerički ukoliko su elementi matrice realni brojevi, odnosno ukoliko je jedan od matricnih elemenata realan što znači da je zadat u decimalnom obliku. Za velike matrice, guste ali i retke, računski je neefikasno računati sve svojstvene vrednosti. Obično se traži najmanja ili najveća, a za taj postupak razvijeni su efikasni numerički algoritmi računanja ekstremalnih svojstvenih vrednosti. Ovi algoritmi su ugrađeni i u funkcije programskog jezika Wolfram, a mogu se pozvati u vidu opcije *Method* → {···}. Svojstveni vektori kvadratne matrice H nalaze se naredbom

$$\text{Eigenvectors}[H],$$

gde je matrica argument funkcije, a izlaz je lista svojstvenih vektora koji redom odgovaraju dobijenim svojstvenim vrednostima izvršenjem funkcije *Eigenvalues*.

```
eigv = Eigenvectors[H]
Conjugate[eigv[[1]]].eigv[[1]]
Conjugate[eigv[[1]]].eigv[[2]]
Conjugate[eigv[[1]]].eigv[[3]]
Conjugate[eigv[[1]]].eigv[[4]]
```

Rezultat primene funkcije *Eigenvectors*[H] su 4 ortonormirana vektora,

```
{{0.-0.0394613 I,0.- 0.00584989 I,-0.117564+0. I,0.992264+0. I},
 {0.+0.533642 I, 0.+0.250762 I,0.799103+0. I,0.117379+0. I},
 {0.- 0.0053472 I,0.+0.955923 I,-0.292115+0. I,-0.029187+0. I},
 {0.-0.844772 I,0.+0.152629 I,0.512134+0. I, 0.0279821+0. I}}
```

što je provereno primenom jednačine (1.1). Prvi svojstveni vektor skalarno pomnožen sam sa sobom daje jedinicu, a sa svim ostalim vektorima daje nulu. Odnos svih svojstvenih vektora može se proveriti tako što se skalarno pomnoži svaki sa svaki i sa samim sobom. Lista *eigv* svojstvenih vektora čini matricu, koju kada konjugujemo i matricno pomnožimo sa njom transponovanom, dobije se matrica A

$$eigv^* . eigv^T = A \quad (1.2)$$

čiji element a_{ij} odgovara skalarnom proizvodu i -tog i j -tog svojstvenog vektora $a_{ij} = \text{Conjugate}[eigv[[i]]].eigv[[j]]$.

```
Conjugate[eigv].Transpose[eigv]
MatrixForm[Chop[%]]
```

Rezultat matičnog množenja konjugovane i njoj transponovane matrice $eigv$ je matrica čiji su elementi kompleksni brojevi po modulu veoma bliski nuli, te se primenom naredbe

$$\text{Chop}[expr],$$

koja realne brojeve (realni i imaginarni deo kompleksnog broja) bliske nuli zamenjuje nulom, dobije jedinična matrica. Potom je rezultat ispisan u obliku matrice

$$\begin{pmatrix} 1. & 0 & 0 & 0 \\ 0 & 1. & 0 & 0 \\ 0 & 0 & 1. & 0 \\ 0 & 0 & 0 & 1. \end{pmatrix}.$$

Za unitarni operator U prema definiciji važi

$$U^\dagger U = U U^\dagger = I.$$

Ukoliko operator U reprezentujemo matično, iz date definicije sledi da kolone (vrste) reprezentacione matrice čine skup ortonormiranih vektora. To znači, slaganjem kolona dobijenog svojstvenog ortonormiranog bazisa, dobije se unitarna matrica. Primena dobijene unitarne matrice u relaciji sličnosti, može se preći iz zadate matrice H u njoj dijagonalnu formu

$$H_{diag} = U^\dagger H U,$$

pri čemu konstrukcijom U od ortonormiranog svojstvenog bazisa, njegove kolone (i vrste) se ponašaju kao ortonormirani vektori, odakle se adjungovan operator U može naći preko $U^\dagger = (U^*)^T$.

```
Hdiag=(Conjugate[eigv].H.Transpose[eigv]);
MatrixForm[Chop[Hdiag]]
```

Rezultat primene relacije sličnosti je dijagonalizovana matrica hamiltonijan reprezentovana u svojstvenom bazisu, pri čemu su na dijagonali svojstvene vrednosti (energije) redom koji odgovara redosledu po kom su sortirani svojstveni vektori u $eigv$

$$\begin{pmatrix} 3.0237 & 0 & 0 & 0 \\ 0 & 1.63842 & 0 & 0 \\ 0 & 0 & 0.998322 & 0 \\ 0 & 0 & 0 & -0.660442 \end{pmatrix}.$$

Primeru radi, biće izračunate svojstvene vrednosti i svojstveni vektori Paulijevih matrica. Paulijeve matrice čine skup od 3 kompleksne matrice σ_i , $i = 1, 2, 3$ (ili $i = x, y, z$) formata 2×2 , istovremeno ermitske, involutivne, unitarne i traga nula. Veoma su važne za kvantnu mehaniku, pojavljuju se u Paulijevoj jednačini koja uzima u obzir interakciju spina sa spoljašnjim elektromagnetnim poljem. Jedinična matrica 2×2 uzima se kao nulta Paulijeva matrica σ_0 i zajedno sa ostalim Paulijevim matricama čine bazis u prostoru matrica 2×2 nad poljem realnih brojeva. Tako proizvoljna ermitska matrica formata 2×2 može biti prikazana kao jedinstvena linearna kombinacija Paulijevih matrica sa realnim koordinatama. Ovo znači da Paulijeve matrice razapinju prostor opservabli u kompleksnom dvodimenzionalnom Hilbertovom prostoru pošto su opservable u kvantnoj mehanici reprezentovane ermitskim operatorima. Najpre će biti generisana lista Paulijevih matrica koristeći naredbu

$$\text{PauliMatrix}[k], \quad k = 1, 2, 3$$

```
list=Range[3];
sigma=PauliMatrix[#]&/@ list;
Map[MatrixForm[#]&,sigma]
```

odakle se generiše lista

$$\left\{ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \right\}.$$

Potom će biti izračunate svojstvene vrednosti, a zatim njima odgovarajući svojstveni vektor za svaku matricu ponaosob upotrebom naredbe

$$\text{Eigensystem}[\text{matrix}],$$

čiji je izlaz lista, koja na prvom mestu sadrži podlistu svojstvenih vrednosti a na drugom podlistu odgovarajućih svojstvenih vektora.

```
eigprob=Map[Eigensystem[#]&,sigma]
```

Dobijeni rezultat dat je u obliku složene liste, čiji elementi su podliste u kojima je na prvom mestu dat spektar a na drugom skup svojstvenih vektora koji redom odgovaraju svojstvenim vrednostima spektra:

$$\left(\begin{array}{cc} \{-1,1\} & \{-1,1\},\{1,1\} \\ \{-1,1\} & \{I,1\},\{-I,1\} \\ \{-1,1\} & \{0,1\},\{1,0\} \end{array} \right).$$

Koristeći relaciju (1.2) na sledeći način može se proveriti da li je dobijeni skup vektora ortonormiran.

Primeru radi za svojstvene vektore matrice σ_1

```
Conjugate[eigprob[[1,2]]].Transpose[eigprob[[1,2]]];
MatrixForm[%]
```

odakle se dobija

$$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \text{ što znači da su vektori uzajamno ortogonalni jer su vandijagonalni elementi jednaki nuli.}$$

Međutim, na dijagonali se ne nalaze jedinice već 2, što znači da dobijeni svojstveni vektori nisu normirani. Uopštenjem ovog testa na listu sve tri Paulijeve matrice redom σ_1, σ_2 i σ_3

```
Map[MatrixForm[Conjugate#[[2]].Transpose#[[2]]]&,eigprob]
```

dobije se lista matrica

$$\left\{ \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\} \text{ odakle se vidi da su svi svojstveni vektori uzajamno ortogonalni, a samo su svojstveni vektori matrice } \sigma_3 \text{ normirani.}$$

Tenzorske operacije

Vektor reprezentovan u programskom jeziku Wolfram ima strukturu liste brojeva, a matrica je lista čije su komponente liste brojeva istih dužina. Analogno, tenzori višeg reda imaju strukturu listi od listi od ... listi brojeva. Na sledećem primeru biće ilustrovani neki elementarni alati za rad sa tenzorima, kreiranje ili kontrakciju. Koristeći naredbu

$$\text{Range}[n],$$

generiše se lista brojeva, vektor iz prostora dimenzije n , čije su komponente prvih n prirodnih brojeva redom.

Na početku primera kojim će se ilustrovati rad sa tenzorima biće generisana lista prvih 24 prirodna broja

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24\}.$$

```
vect=Range[24];
Dimensions[vect]
```

čija je dimenzija {24}. Generisan je vektor dimenzije 24 zbog raznolike mogućnosti kreiranja tenzora različitog reda, a osim toga izbor elemenata je niz prirodnih brojeva radi lakšeg uočavanja složene listabilne strukture. Zadana lista *vect* biće preoblikovana u tenzor formata $2 \times 3 \times 4$ naredbom

$$\text{ArrayReshape}[vect, 2, 3, 4]$$

```
tns1=ArrayReshape[vect,{2,3,4}]
Dimensions[tns1]
```

čime se generiše lista dimenzije $\{2,3,4\}$ sazdana od 2 podliste, pri čemu svaka od njih sadrži 3 podliste dužine 4 na sledeći način

$$\left(\begin{array}{ccc} \{1,2,3,4\} & \{5,6,7,8\} & \{9,10,11,12\} \\ \{13,14,15,16\} & \{17,18,19,20\} & \{21,22,23,24\} \end{array} \right).$$

Pri tom, redosled elemenata nije promenjen, samo je rearanžirana struktura, tako da su oni sada elementi podlista od podlista od liste. Slično, zadata lista *vect* preoblikovana je u tenzor $2 \times 2 \times 3 \times 2$

```
tns2=ArrayReshape[vect,{2,2,3,2}]
Dimensions[tns2]
```

dimenzije $\{2,2,3,2\}$ čija se struktura složene liste može prikazati na sledeći način

$$\left(\begin{array}{cc} \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} & \begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix} \\ \begin{pmatrix} 13 & 14 \\ 15 & 16 \\ 17 & 18 \end{pmatrix} & \begin{pmatrix} 19 & 20 \\ 21 & 22 \\ 23 & 24 \end{pmatrix} \end{array} \right).$$

Suprotno od regrupisanja postoji operacija ravnjanja date složene liste *tns* realizovana njenim ravnjanjem,

$$\text{Flatten}[tns],$$

čime se operativno vrši brisanje svih ili definisanih slojeva vitičarskih zagrada koje definišu podliste unutar spoljašnje liste. Postavljanjem upita identičnosti poravnatih, gore definisanih tenzora i početnog vektora

```
Flatten[tns1]===Flatten[tns2]===vect
```

dobije se rezultat *True*. Ovaj primer ilustruje ravnjanje datog tenzora do proste liste. Primena naredbe *Flatten* može biti specifičnija, upotrebom drugog, opcionog argumenta ove funkcije, što ne mora da rezultuje strukturom proste liste. Primera radi ako se u tenzoru *tns2* izravnaju liste na nivou indeksa 1 i 2 i indeksa 3 i 4, dobiće se matrica formata 4×6 .

```
tns21=Flatten[tns2,{1,2},{3,4}]
tns3=ArrayReshape[vect,{4,6}]
tns21===tns3
```

Ravnjanjem liste *tns2* na zadatom nivou dobija se matrica

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \end{pmatrix}.$$

Preuređivanje tenzora može se realizovati transpozicijom indeksa naredbom *Transpose*.

```
tns12=Transpose[tns1, {2, 3, 1}]
Dimensions[tns12]
MatrixForm[tns12]
```

Rezultat je tenzor dimenzije $\{4,2,3\}$

$$\left(\begin{array}{cc} \{1,5,9\} & \{13,17,21\} \\ \{2,6,10\} & \{14,18,22\} \\ \{3,7,11\} & \{15,19,23\} \\ \{4,8,12\} & \{16,20,24\} \end{array} \right).$$

2 Grafički potencijal programa Mathematica

Vizuelizacija podataka, proizvedenih programskim kodovima (ili eksperimentalno) kao što su talasne funkcije, gustine naelektrisanja, ima neizmerno važnu ulogu u analizi rezultata i razumevanju pojava. Program Mathematica poseduje raznolike grafičke mogućnosti što podrazumeva interaktivnu grafiku, manipulacije i animacije, osnovno crtanje u dve i tri dimenzije. Generisanje adekvatnog grafika u programu Mathematica podrazumeva upotrebu različitih opcija kao što su odnos visine i širine crteža, opseg koordinata u kom se vrši prikaz, broj tačaka u kojima se uzorkuje funkcija. Drugim rečima, specificiraju se vrednosti opcija u funkcijama za prikaz. Tema ovog poglavlja je programiranje grafičkog prikaza i pregled grafičkih mogućnosti programskog jezika Wolfram . Biće uvedeni osnovni pojmovi grafičkih objekata sa njihovim elementima i opcijama, potom dvodimenzionalni i trodimenzionalni grafički prikazi. Jedan od glavnih razloga atraktivnosti grafike u programu Mathematica je dostupnost ilustrovanja u boji, što je demonstrirano kroz različite ilustracije u knjizi.

2.1 Dvodimenzionalni obični i višestruki grafici, primitivni objekti i opcije

Ugrađene funkcije programa Mathematica generišu grafikone lista podataka ili matematičkih funkcija. Da bi napravila dijagram, funkcija napravi izraz koji predstavlja grafički objekat, a potom poziva na prikaz njegove grafičke slike. Stavljanjem funkcije za crtanje pod argument *InputForm* može se videti njen izraz

```
InputForm[ListPlot[Table[{i,i},{i,0,4}],PlotStyle->PointSize[0.01]]]
```

```
Graphics[{{},{}, {Hue[0.67, 0.6, 0.6],  
Directive[RGBColor[0.368417, 0.506779,0.709798],  
AbsoluteThickness[1.6],PointSize[0.01]],  
Point[{{0.,0.},{1.,1.},{2.,2.},{3.,3.},{4.,4.}]},{}},{}},{}},{}},  
{DisplayFunction->Identity, PlotRangePadding->  
{Scaled[0.02],Scaled[0.02]},{Scaled[0.02],Scaled[0.05]}},  
AxesOrigin->{0,0},PlotRange->{{0, 4.},{0,4.}},PlotRangeClipping->True,  
...
```

Izraz se sastoji od okruženja *Graphics* koji se sastoji od grafičkih elemenata i opcija. Grafički elementi su *Point*[{0.,0.},{1.,1.},{2.,2.},{3.,3.},{4.,4.}] i direktive *Hue*[0.67,0.6,0.6], dok opcije počinju pozivom *OptionName* → *OptionValue* kao što je *PlotRange* → {{0,4.},{0,4.}}. Da bi se proizveo složeniji grafički oblik potrebno je primeniti grafičko programiranje, konstruisati izraze koji predstavljaju grafičke objekte. Forme grafičkih izraza:

Graphics[*elemets*,*options*] predstavljaju dvodimenzionalne grafičke objekte koji se u standardnoj formi prikazuju kao slike.

Graphics3D[*elements*,*options*] prikazuju trodimenzionalne grafičke objekte, koji se u standardnoj formi prikazuju kao slike.

Pored jedne, prvi argument funkcija *Graphics*(3D) može da sadrži prostu ili ugnježdenu listu osnovnih jedinica (*Point*, *Line*, *Polygon*, *Text*) i direktive (*GrayLevel*, *RGBColor*, *PointSize*, *Thicnkness*) u obliku liste

$\{prim_1, dir_1, dire_2, sublist_1, prim_3, dir_3, \dots\}$, gde su $prim_i$ osnovne (primitivne) jedinice, dir_j direktive, dok podliste $sublist_i$ mogu imati isti sadržaj kao i liste. Naredbama

`Show[g,options]` ili `Show[g1,g2,...,options]`

prikazuje se jedan g ili nekoliko kombinovanih g_1, g_2, \dots grafičkih objekata sa opcijama u opcionom argument. Osnovne jedinice koje se koriste u dvodimenzionalnom prikazu:

Strela koja potiče iz $\{x_1, y_1\}$ i završava se u $\{x_2, y_2\}$ poziva se sa `Arrow[{{x1, y1}, {x2, y2}]`.

Kružnica radijusa r sa centrom u tački $\{x, y\}$ zadaje se sa `Circle[{x,y},r]`, dok je kružni luk između uglova $[\theta_1, \theta_2]$ zadat sa `Circle[{x,y},r,{θ1,θ2}]`. Elipsa sa centrom u datoj tački i poluosama r_a i r_b poziva se naredbom `Circle[{x,y},{ra,rb}]`, dok je slično kružnom, eliptični luk zadat sa `Circle[{x,y},{ra,rb},{θ1,θ2}]`.

Sa `Disk[{x,y},r]` poziva se ispunjen disk radijusa r sa centrom u $\{x, y\}$. Eliptični disk sa poluosama $\{r_a, r_b\}$ ili njegov luk redom se zadaju sa `Disk[{x,y},{ra,rb}]` i `Disk[{x,y},r,{θ1,θ2}]`.

Linija koja spaja tačke $\{x_1, y_1\}, \{x_2, y_2\}, \dots$ zadaje se naredbom `Line[{{x1, y1}, {x2, y2}, \dots]`, dok se istom naredbom može se zadati više nezavisnih linija koje spajaju podskupove tačaka ugnježdene liste `{{{x11, y11}, {x12, y12}, \dots}, \dots}`.

Skup tačaka čije su koordinate $\{x_1, y_1\}, \{x_2, y_2\}, \dots$ zadaje se naredbom `Point[{{x1, y1}, {x2, y2}, \dots]`.

Ispunjen poligon čija su temena $vertices = \{x_1, y_1\}, \{x_2, y_2\}, \dots$ zadaje se izvršenjem naredbe `Polygon[vertices]`.

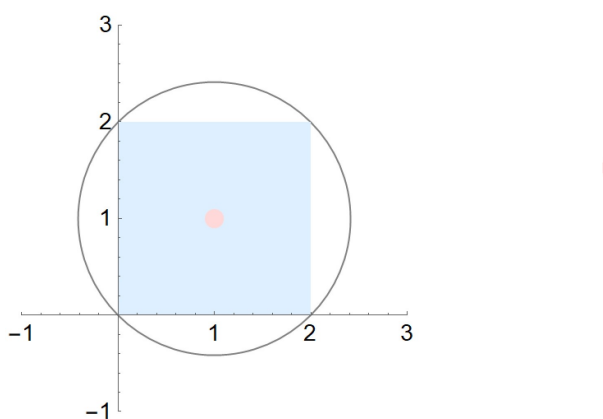
`Rectangle[{xmin, ymin}, {xmax, ymax}]` daje ispunjen pravougaonik, stranica paralelnih osama.

Sa `Text[expr, coo]` tekstualni izraz $expr$ pozicionira se na $coo = \{x, y\}$.

Ilustracije radi, biće prikazani neki osnovni grafički elementi: kvadrat i tačka u centru njemu opisanog kruga.

```
Graphics[{{Gray, Circle[{1, 1}, Sqrt[2]]},
{LightBlue, Rectangle[{0, 0}, {2, 2}]},
PointSize[0.05], LightRed, Point[{1, 1}]},
Axes -> True, PlotRange -> {{-1, 3}, {-1, 3}}
```

Navedeni objekti prikazani na grafiku sa iscrtanim osama $Axes \rightarrow True$ u opsegu zadatom sa $PlotRange \rightarrow \{-1, 3\}, \{-1, 3\}$ prikazani su na slici 2.1.



Slika 2.1: Ilustracija nekih osnovnih grafičkih objekata.

Program podržava mnoge direktive za dvodimenzionalni grafički prikaz. Neke od njih su:

`GrayLevel[level]` definiše nivo sive između 0 (crno) i 1 (belo).

`RGBColor[r, g, b]` određuje boju sa specificiranim vrednostima komponenti r – crvena, g – zelena i b – plava, svake između 0 i 1.

`Hue[h]` je grafička direktiva kojom se specificiraju boje određene promenljivom h između 0 i 1.

`Opacity[a]` definiše nivo neprozirnosti između 0 (potpuno transparentna) do 1 (neprovidna).

`PointSize[d]` zadaje se veličina tačke d , merena u odnosu na ukupnu širinu grafika.

`Thickness[d]` određuje debljinu linije d prema ukupnoj širini grafika. Moguće je izabrati *Thick* (podebljano) i tanko *Thin*.

Sa `Arrowheads[spec]` zadaje se veličina *spec* vrha strelice vektora u odnosu na ukupnu širinu grafika.

Primerom iz geometrijske optike, konstrukcijom dijagrama svetlosnih zraka tankog sabirnog sočiva biće ilustrovano korišćenje osnovnih grafičkih elemenata. Jedan po jedan element, zraci, predmet i lik, sočivo, biće programski nezavisno konstruisani, a potom i prikazani pomoću naredbi za grafičke objekte. Zraci su crtani sa *Line* i pridruženi promenljivoj g_1 . Strelicama u boji pomoću *Arrow* prikazani su predmet i lik. Sa dva kružna luka zadata sa *Circle* formirano je sočivo, a sa diskovima malog radijusa označene su žižne daljine.

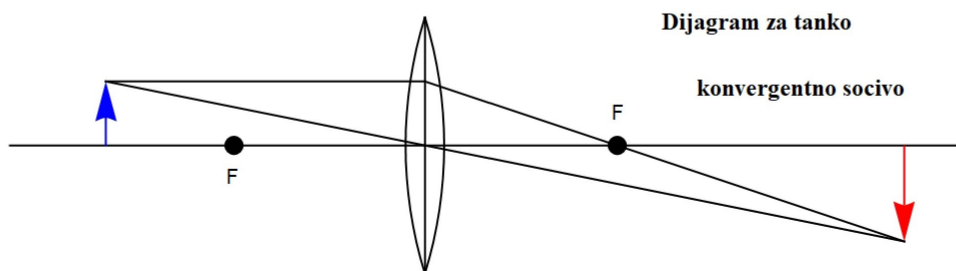
```
(*zraci-linije*)
g1=Line[
  {{-13,0},{17,0}},{0,-4},{0,4},
  {{-10,2},{0,2},{15,-3}},{-10,2},{15,-3}}];
(*predmet (plava strelica)+lik (crvena strelica)*)
g21={Blue,Arrow[{{-10,0},{-10,2}}]};
g22={Red,Arrow[{{15,0},{15,-3}}]};
(*tanko sabirno socivo-dva kruzna luka*)
gcr1=Circle[{13,0},13.6,{2.843,3.44}];
gcr2=Circle[{-13,0},13.6,{5.985, 6.582}];
(*tacke leve i desne zize- puni diskovi*)
pts={Disk[{-6,0},0.3],Disk[{6,0},0.3]};
(*oznake ziza+opis-tekst*)
txt={Text["F",{-6,-1}],Text["F",{6.05, 1}],
Text[Style["Dijagram za tanko \n
konvergentno socivo",
FontFamily -> "Times New Roman", Bold,12],
{10.5, 2.8}]}];
Graphics[{g1,g21,g22,gcr1,gcr2,pts,txt}]
```

Oznake žiža i naslov slike ispisani su naredbom *Text*. Zadavanje karakteristika naslova grafika, veličina i oblik slova 12 zatim i podebljanje izvršeno je u okruženju *Style* opcijom `FontFamily -> "TimesNewRoman", Bold, 12`. Nakon pojedinačne pripreme osnovnih objekata, a potom objedinjeni svi zajedno su kao elementi jedne liste (skupa) stavljani u argument naredbe *Graphics*.

Deo skupine osnovnih grafičkih prikaza spada i crtanje funkcije jedne promenljive $f(x)$ od x_{min} do x_{max} . Ovo se svrstava u dvodimenzionalne grafike i realizuje se naredbom

$$\text{Plot}[f, \{x, x_{min}, x_{max}\}, \text{OptionName}_1 \rightarrow \text{OptionValue}_1, \dots].$$

Funkcija i domen promenljive x su obavezni argumenti, dok su preostali opcioni. Tokom generisanja grafika u programu Mathematica na raspolaganju su različite opcije, a neke od njih, često upotrebljavane, su odnos visine i širine, izgled i oznake koordinatnih osa, linije grida, opseg koordinata i mnoge druge, koje se specificiraju nakon obaveznih argumenata. Ukoliko opcije nisu eksplicitno zadate, program Mathematica automatski uključuje one koje se podrazumevaju. Ilustracije radi, biće prikazana



Slika 2.2: Dijagram zraka kod sabirnog sočiva.

normalizovana svojstvena funkcija prvog pobuđenog stanja jednodimenzionalnog harmonijskog oscilatora

$$\psi_1(x) = \left(\frac{4}{\pi}\right)^{1/4} x e^{-x^2/2}.$$

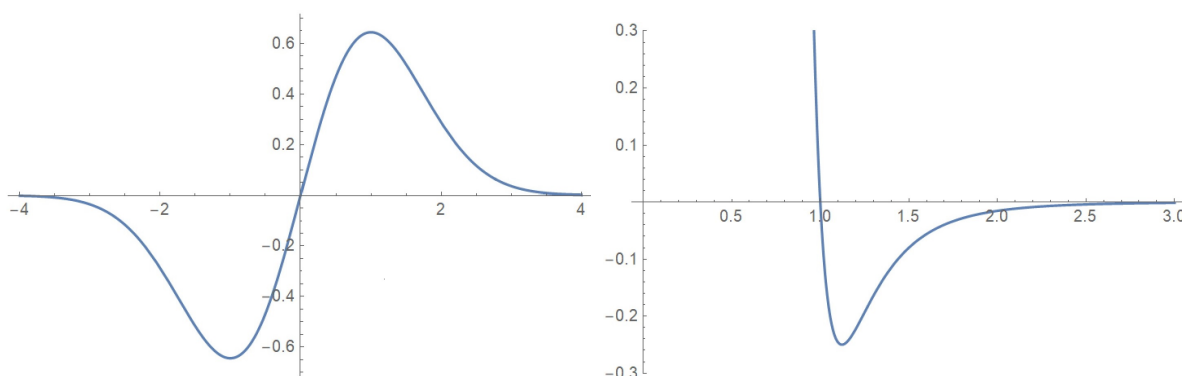
```
Plot[(4/Pi)^(1/4)x Exp[-x^2/2], {x, -4, 4}]
```

U sledećem primeru biće ilustrovan funkcionalni oblik potencijala Lenard-Džons

$$V(r) = 4\epsilon \left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right)$$

zadat u jedinicama od $\sigma = 1$ i $4\epsilon = 1$, na opsegu zadatom opcijom $PlotRange \rightarrow \{-0.3, 0.3\}$.

```
Plot[1/r^12-1/r^6, {r, 0.001, 3},  
PlotRange->{-0.3, 0.3}]
```



Slika 2.3: Rešenja postavljenih zadataka, levo: talasna funkcija prvog pobuđenog stanja 1 D LHO, a desno: Lenard-Džons-ov potencijal.

Grafički prikaz podataka

Naredbom

```
ListPlot[{{x1, y1}, {x2, y2} ...}]
```

prikazuju se parovi tačaka $\{\{x_1, y_1\}, \{x_2, y_2\} \dots\}$ na XY ravni. Ilustracije radi grafički će biti prikazan skup parova, kod kojih je prva koordinata vreme a druga dužina.

```
data = {{0, 0}, {1, 1.5}, {2, 6.}, {3, 13.5}, {4, 24.0}, {5, 37.5},  
{6, 54.0}, {7, 73.5}, {8, 96.0}, {9, 121.5}};
```

U nastavku dati skup podataka je fitovan na kvadratnu funkciju, a potom je dobijeni fit prikazan na istom grafiku zajedno sa setom datih eksperimentalnih tačaka.

```
g1=ListPlot[data,PlotStyle->PointSize[0.03],
AxesLabel->{"t(s)", "l(m)"}];
```

Tamno plavim tačkicama veličine 0.03 prikazani su podaci *data*, a ceo grafički prikaz pridružen je promenljivoj *g1*. Naredbom

$$\text{FindFit}[data, expr, param, vars]$$

nalaze se parametri *param*, skup zadan na mestu treće varijable, zadate funkcije *expr* (druga varijabla) promenljive *vars* (četvrti ulaz) kojom se fituje zadan set podataka *data* (prvi ulaz). Izvršenjem naredbe dobija se skup zamena koje odgovaraju funkciji koja najbolje fituje zadate podatke.

```
ftpmt=FindFit[data, a0+a1*x+a2*x^2,{a0,a1,a2},x];
```

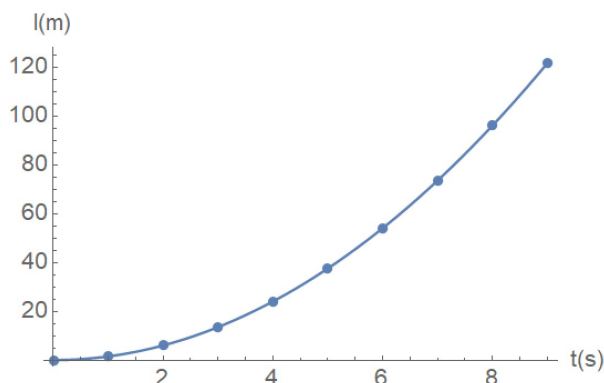
Lista zamena parametara kvadratne funkcije, dobijena primenom *FindFit* na zadati set podataka *dat* je

$\{a_0 \rightarrow -0.649091, a_1 \rightarrow -0.0128788, a_2 \rightarrow 1.51288\}$. Nađena pravila zamene su iskorišćena te su zamenjeni opšti parametri kvadratne funkcije $(a_0 + a_1x + a_2x^2)/.ftpmt$, nakon čega je ona nacrtana na opsegu u kom su date *x* koordinate parova a grafik je pridružen promenljivoj *g2*. Naredbom

$$\text{Show}[g_1, g_2, \dots, option]$$

istovremeno se prikazuju na zajedničkoj skali formirani grafici g_1, g_2, \dots sa zadatim opcijama *option*, recimo opseg prikaza. Ovom prilikom funkcija *Show* omogućava prikaz skupa tačaka i njemu odgovarajućeg fita na istom grafiku.

```
g2=Plot[(a0+a1*x+a2*x^2) /. ftpmt, {x, 0, 9}];
Show[g1,g2]
```



Slika 2.4: Podaci (plave tačke) prikazani zajedno sa parabolom koja ih fituje (puna plava linija).

Naredbom

$$\text{ListLinePlot}[\{\{x_1, y_1\}, \{x_2, y_2\} \dots\}]$$

crtaju se linije koje spajaju svake dve susedne tačke seta iz argumenta funkcije. Isti grafik bi se dobio ukoliko bi tačke bile interpolirane krivom interpolacionog reda 1. Interpolacija funkcije zadanih vrednosti f_1, f_2, \dots nezavisno promenljive x_1, x_2, \dots vrši se naredbom

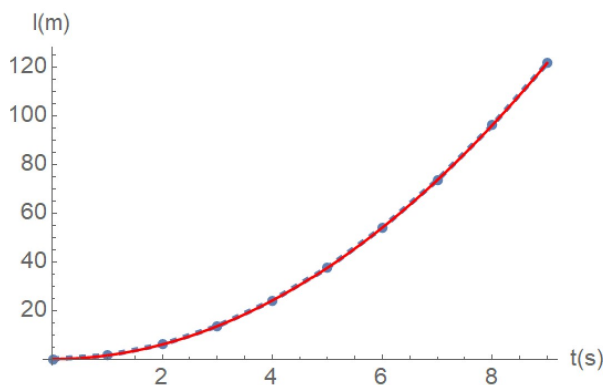
$$\text{Interpolation}[\{\{x_1, f_1\}, \{x_2, f_2\} \dots\}].$$

Najpre će biti primenjena naredba *ListLinePlot* na zadati skup tačaka *data* iz prethodnog primera. Dobijeni grafik biće nacrtan podebljanom isprekidanom linijom $\text{PlotStyle} \rightarrow \{\text{Dashed}, \text{Thickness}[0.01]\}$ i pridružen promenljivoj *g3*.

```
g3 = ListLinePlot[data, PlotStyle -> {Dashed, Thickness[0.01]}];
```

Interpolacijom podataka dobijena je funkcija $fdat$, nacrtana na opsegu koordinata nezavisno promenljive x_i te grafik pridružen promenljivoj g_4 , a potom prikazana preko prethodno dobijenog grafika spajanjem redom susednih tačaka zajedno sa graficima g_1 i g_3 .

```
fdat=Interpolation[data, InterpolationOrder->3];
g4=Plot[fdat[x], {x, 0, 9}, PlotStyle->Red];
Show[g1, g3, g4]
```



Slika 2.5: Susedne tačke spajane pravim isprekidanim linijama sa *ListLinePlot* zajedno sa interpolacionom funkcijom 3. reda prikazanom crvenom linijom.

Parametarski grafici

Funkcije $f_x(t), f_y(t)$ parametra t koje čine par x i y koordinata, ilustruju se parametarskim graficima uz pomoć naredbe

$$\text{ParametricPlot}[\{f_x, f_y\}, \{t, t_{min}, t_{max}\}].$$

Crtanjem slike koja se formira na osciloskopu, formirana udarom snopa elektrona na ekranu, ilustruje se prikaz parametarskih grafika. Elektroni u cevi osciloskopa su ubrzavani uz pomoć dva ortogonalna električna polja tako da je u svakom trenutku t njihov položaj zadat sa

$$x = A_x \cos(\omega_x t + \phi_x), \quad y = A_y \cos(\omega_y t + \phi_y),$$

gde su amplituda, ugaona frekvencija i faza redom A_x, ω_x, ϕ_x horizontalne, a A_y, ω_y, ϕ_y vertikalne defleksije. Nacrtati parametarske grafike za:

- i) $A_x = A_y = A, \quad \omega_x = \omega_y = \omega, \quad \phi_x = 0, \phi_y = \pi/2;$
- ii) $A_x = A_y = A, \quad \omega_x = \omega_y/3 = \omega, \quad \phi_x = 0, \phi_y = \pi/2;$
- iii) $A_x = A_y = A, \quad \omega_x = \omega_y/3 = \omega, \quad \phi_x = 0, \phi_y = -\pi/4;$
- iv) $A_x = A_y = A, \quad \omega_x/5 = \omega_y/4 = \omega, \quad \phi_x = 0, \phi_y = \pi/2.$

Mere dužine odnosno pomeraja biće izražene u jedinicama A , a vremena u $1/\omega$.

- i) $A_x = A_y = 1, \quad \omega_x = \omega_y = 1, \quad \phi_x = 0$ i $\phi_y = \pi/2$

```
ParametricPlot[{Cos[t], Cos[t+Pi/2]}, {t, 0, 2Pi},
AxesLabel -> {"x(A)", "y(A)"}]
```

Grafici sistema parametarskih jednačina u narednim primerima su Lisažuoove figure.

ii) $A_x = A_y = 1$, $\omega_x = \omega_y/3 = 1$, $\phi_x = 0$ i $\phi_y = \pi/2$

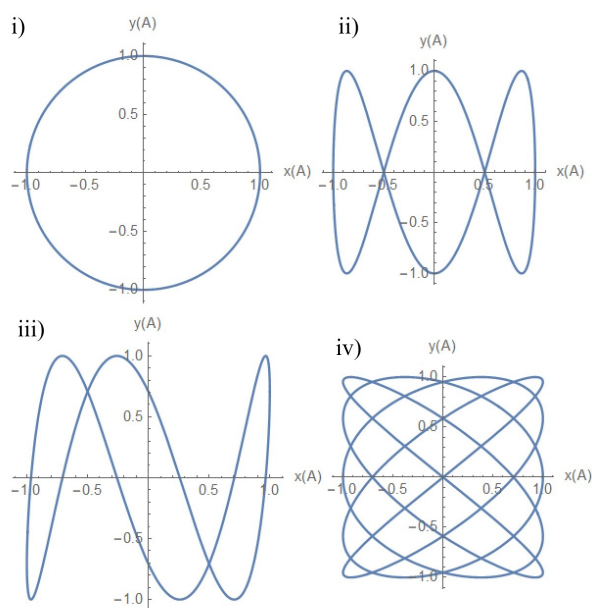
```
ParametricPlot[{Cos[t], Cos[3t+Pi/2]},{t,0,2Pi},
AxesLabel->{"x(A)", "y(A)"}]
```

iii) $A_x = A_y = 1$, $\omega_x = \omega_y/3 = 1$, $\phi_x = 0$ i $\phi_y = -\pi/4$

```
ParametricPlot[{Cos[t], Cos[3t-Pi/4]},{t,0,2Pi},
AxesLabel->{"x(A)", "y(A)"}]
```

iv) $A_x = A_y = 1$, $\omega_x/5 = \omega_y/4 = 1$, $\phi_x = 0$ i $\phi_y = \pi/2$

```
ParametricPlot[{Cos[5t], Cos[4t+Pi/2]},{t,0,2Pi},
AxesLabel->{"x(A)", "y(A)"}]
```



Slika 2.6: Grafici sistema parametarskih funkcija koji se javljaju na osciloskopu za različite zadate parametre zadate pod i), ii), iii) i iv).

2.1.1 Električni dipol. Konture ekvipotencijala i linije električnog polja.

Nacrtati ekvipotencijalne linije oko dva tačkasta naelektrisanja q_1 i q_2 na međusobnom rastojanju d ako je

a) $q_1 = q_2 = +q$,

b) $q_1 = +2q$ i $q_2 = -q$.

Skalarno polje u tački \vec{r} , koje potiče od sistema n tačkastih naelektrisanja q_1, q_2, \dots, q_n pozicionirani u $\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n$ redom je

$$\varphi(\vec{r}) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^n \frac{q_i}{|\vec{r} - \vec{r}_i|}. \quad (2.1)$$

Ekvipotencijalne linije su oblasti (tačke) u prostoru na istom električnom potencijalu. Za sistem tačkastih naelektrisanja raspoređenih u prostoru, ekvipotencijali čine površine. Izuzev tačaka u ravnotežnom položaju gde je električno polje nula, ekvipotencijalne površi se ne seku i silnice

električnog polja su svuda ortogonalne na ove površi. Prema uslovu zadatka problem se svodi na prikaz skalarnog polja oko dva tačkasta naelektrisanja. Shodno tome, radi što lakše izrade, bira se odgovarajući koordinatni sistem. Radi lakšeg programskog rešavanja, na sredini između dve tačke postavljen je koordinatni početak, a duž pravca koji određuju dve čestice od q_1 ka q_2 položena je x osa. Oblik ekvipotencijalnih krivih ovog sistema tačkastih naelektrisanja je isti u svakoj ravni koja sadrži x osu nezavisno od rotacije oko nje. Zbog toga će grafički prikaza ekvipotencijala biti dat u XY ravni. Za dve naelektrisane čestice, shodno relaciji 2.1 prostorna zavisnost električnog potencijala je data sa

$$\varphi(x,y) = \frac{1}{4\pi\epsilon_0} \left(\frac{q_2}{\sqrt{(x-d/2)^2 + y^2}} + \frac{q_1}{\sqrt{(x+d/2)^2 + y^2}} \right). \quad (2.2)$$

Električno polje se iz skalarnog potencijala dobije

$$\vec{E} = -\nabla\varphi(x,y) = -\frac{\partial\varphi(x,y)}{\partial x}\vec{e}_x - \frac{\partial\varphi(x,y)}{\partial y}\vec{e}_y, \quad (2.3)$$

gde su \vec{e}_x , \vec{e}_y jedinični vektori duž x i y ose.

Smer u kom deluje električna sila na probno naelektrisanje postavljeno u električnom polju pokazuju električne silnice. Linije električnog polja su ortogonalne na ekvipotencijalne linije skalarnog polja. Električno polje je veličina koja opisuje jačinu delovanja električne sile na naelektrisanje. Ilustruje se električnim silnicama kao krive u prostoru, čije tangente u svakoj tački predstavljaju smer električnog polja. Gustina silnica u svakoj oblasti prostora srazmerna je jačini polja u njoj. Pozitivna naelektrisanja su izvori električnih silnica a negativna ponori. Ovde će biti prikazan prost metod kojim se mogu generisati silnice električnog polja kolinearnih naelektrisanja (naelektrisanja poredana u niz), koji se uz male izmene može generalizovati na nekolinearna naelektrisanja. Zadate su tačke (ukupno n), raspoređene ravnomerno po krugu, iz kojih izvire odnosno u koje poniru silnice polja. Ako je u tački (x,y) vektor električnog polja $\vec{E}(x,y) = E_x(x,y)\vec{e}_x + E_y(x,y)\vec{e}_y$, pomeranjem iz nje duž silnice električnog polja za neku malu infinitezimalnu dužinu ds stiže se u njoj susednu definisanu sa

$$\left(x + \frac{E_x}{E} ds, y + \frac{E_y}{E} ds \right). \quad (2.4)$$

```

\{CurlyPhi}[x_,y_]:=
(1/(4 Pi \[Epsilon])) (q2/Sqrt[(x-d/2)^2+y^2]+q1/Sqrt[(x+d/2)^2+y^2]);
value={\[Epsilon]->1,d->1,q1->-1,q2->2};
ds=0.001;
r=0.1;
n=18;(*broj izvora silnica elektricnog polja*)
(*polozaji naelektrisanja q1 i q2*)
rq1={-d/2,0} /. value;
rq2={d/2,0} /. value;
(*izrazi za elektricni potencijal i polje*)
(*zamenjene zadate vrednosti u funkciji potencijala*)
fi=\{CurlyPhi}[x,y] /.value;
(*Vektor elektricnog polja*)
efld=-Grad[fi,{x,y}];
(*komponente elektricnog polja*)
Ex=efld[[1]];
Ey=efld[[2]];
(*Intenzitet elektricnog polja*)
Eintnst=Sqrt[Ex^2+Ey^2];
(*=====*)
(*inicijalne tacke iz kojih idu linije polja*)
init1=Table[r{Cos[f],Sin[f]}+rq1,{f,2Pi/n,2Pi-2Pi/n,2Pi/n}];
init2=Table[init1[[i]]{-1,1},{i,Length[init1]}];
(*ilustracija tackastog naelektrisanja obojenim diskovima*)
chrg1=Graphics[{LightBlue,Disk[rq1,r],
Text[Style["2q",Medium,Bold,Darker[Red],rq2,Automatic,{1,0}]}];
Text[Style["-q",Medium,Bold,Darker[Blue],rq1, Automatic,{1,0}]}];
chrg2=Graphics[{LightRed,Disk[rq2,r],

```

Program počinje definicijom funkcije električnog potencijala, potom numeričkog parametra pomeraja duž silnice ds , pravila zamene vrednosti ϵ_0, d, q_1, q_2 i broja izvora silnica koji se homogeno raspoređuju po disku radijusa r , pozicioniranog u tački naelektrisanja $rq_1/2$. Zamenom vrednosti u funkciji potencijala formira se izraz fi u zavisnosti od prostornih koordinata. Dejstvom nabra operatora na izraz fi po datim koordinatama (x, y) ,

$$\text{Grad}[\varphi(x, y), \{x, y\}]$$

nalazi se vektor polja u vidu liste $efld$, čije su komponente izrazi za $E_x(x, y)$ i $E_y(x, y)$ redom, iz kojih se dalje nalazi izraz intenziteta polja $E = \sqrt{E_x^2 + E_y^2}$. Potom su definisane tačke $init1/2$, po obodu ilustracija naelektrisanja u vidu obojenih diskova sa oznakama u centru $chrg1/2$, inicijalne za konstrukciju silnica polja.

```
(*silnice elektricnog polja*)
lst=Table[
(*otvoreni skupovi sa pocetnim tackama na
krugovima oko tackastih naelektrisanja*)
setcoo1={init1[[i]]};
coo1=init1[[i]];
setcoo2={init2[[i]]};
coo2=init2[[i]];
Do[
(*iz (x,y) u susednu tacku (x+Ex/E ds,y+Ey/E ds) duz silnice polja*)
coo1=coo1+((ds*{Ex,Ey}/Sqrt[Ex^2+Ey^2])/Thread[{x,y}->coo1]);
AppendTo[setcoo1,coo1];
coo2=coo2+((ds*{Ex,Ey}/Sqrt[Ex^2+Ey^2])/Thread[{x,y}->coo2]);
AppendTo[setcoo2,coo2],{4000}];
{setcoo1, setcoo2},{i,n}];

eflines1 = Show[
Table[ListLinePlot[lst[[j, 1]],{j,Length[lst]}],
PlotRange -> All];
eflines2=Show[
Table[ListLinePlot[lst[[j, 2]],{j,Length[lst]}],
PlotRange->All];
```

Jednak broj silnica polja izvire iz svakog naelektrisanja. Svaka silnica je data u vidu niza tačaka $setcoo1/2$, dobijenih prelaskom iz jedne u drugu prema jednačini (2.4). Elementi liste lst sadrže n parova $\{setcoo1, setcoo2\}$, iz kojih se posle selektuju one koje izvire na određenom naelektrisanju i nezavisno crtaju za svaki izvor $eflines1/2$.

```
(*funkcija*)
ContPot=ContourPlot[fi,{x,-1.5,1.5},{y,-1.5,1.5},
(**)ContourShading->False,
PlotRange->{-0.2,0.3},
Contours->10,
ContourStyle->{{Blue,Dashing[{0.01,0.01}]}}],
Frame->False];
(*konturni plot+silnice polja+oznake naelektrisanja*)
Show[ContPot, eflines1, eflines2, chrg1, chrg2]
```

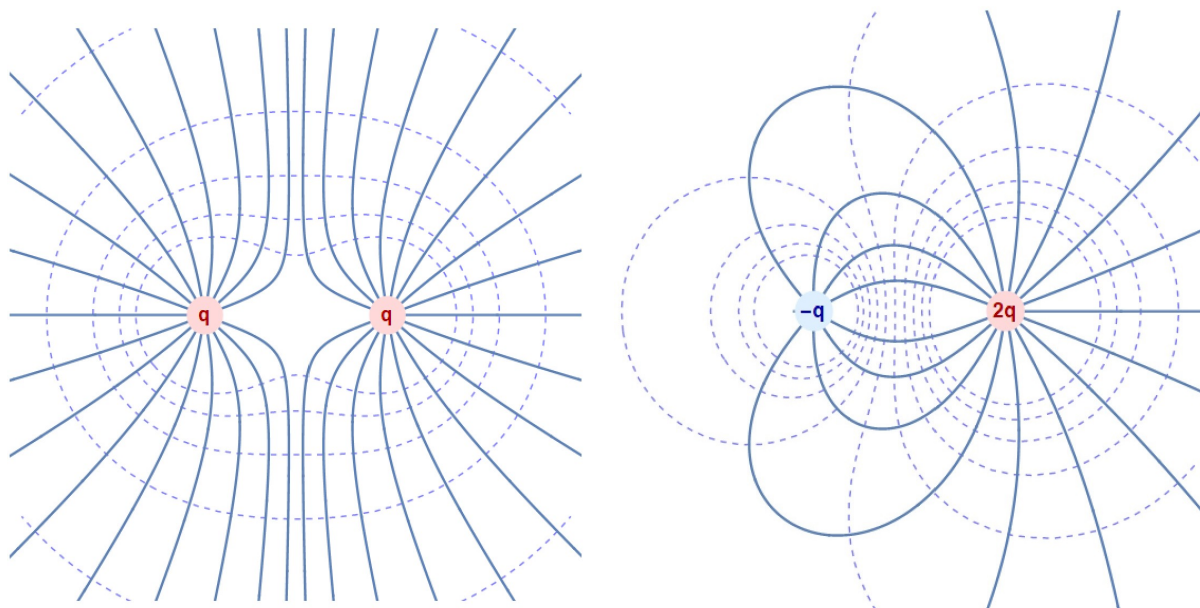
Pored linija sila električnog polja, konturni plot u zadatom opsegu sa 10 kontura $PlotRange \rightarrow \{-0.2, 0.3\}$, $Contours \rightarrow 10$, prikazan je bez obojenih oblasti između kontura $ContourShading \rightarrow False$ isprekidanim plavim linijama $ContourStyle \rightarrow \{\{Blue, Dashing[\{0.01, 0.01\}]\}$. Ovim su prikazane ekvipotencijalne krive, a grafik je dodeljen promenljivoj $ContPlot$. Svi nezavisno generisani grafički elementi su prikazani na istom grafiku.

2.1.2 Dejstvo linearnih transformacija.

i) Animirati dejstvo linearnih transformacija na vektore u dvodimenzionalnoj ravni. Linearna transformacija zadata je matricom

$$A_{mat} \in R^{2 \times 2}, \quad A_{mat} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Njeno dejstvo na tačke jediničnog kruga ilustrovati kontinualnom promenom matričnih elemenata.



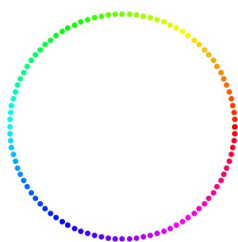
Slika 2.7: Ekvipotencijalne krive (isprekidana plava linija) i silnice električnog polja (puna plava linija) za sistem naelektrisanja sačinjen od $q_1 = q_2 = +q$ (levo) i $q_1 = -q$, $q_2 = +2q$ (desno).

- ii) Animirati dejstvo linearnih transformacija u dvodimenzionalnoj ravni delujući linearnim operatorom na geometrijsku figuru u ravni (trougao).

Najpre će biti ilustrovan set tačaka na jediničnom krugu, čije se boje kontinualno menjaju duž kružnice. Delujući matricom A_{mat} na vektor položaja svake tačke $(x_i, y_i)^T$ on se menja u ravni, a sa njim i oblik početne figure koju tačkice formiraju.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} ax_i + by_i \\ cx_i + dy_i \end{pmatrix}.$$

```
dots=Table[{Cos[x], Sin[x]}, {x, 0, 2Pi, 2Pi/100}];
coldots=Table[{Hue[i/Length[dots]], Point[dots[[i]]]},
{i, Length[dots]}];
g0=Graphics[coldots]
```



Slika 2.8: Tačke raspoređene po obodu kruga, obojene da bi se lakše pratila njihova pozicija tokom promene njenog položaja pod dejstvom linearnog operatora.

Opšti elementi matrice kojom se deluje na vektore zamenjeni su brojnim vrednostima, a potom je njome delovano na svaku tačku kruga. Krug zadat na početku zajedno sa novodobijenim rasporedom tačaka prikazuju se istovremeno. Da bi se bolje razumelo dejstvo, istom transformacijom su preslikani i prikazani i bazisni vektori pri čemu početni bojeni crvenom i plavom, delovanjem transformacije redom prelaze u tamno crveni i tamno plavi.

```

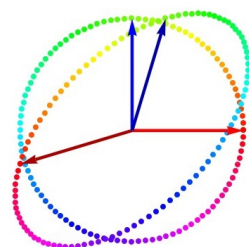
Amat={{a,b},{c,d}};
amat=Amat/.{a->-1,b->0.3,c->-0.3,d->1};
bazis=Transpose[{0IdentityMatrix[2],IdentityMatrix[2]}];
colrs={Red,Blue};
colrs1={Darker[Red],Darker[Blue]};
gbaz=Graphics[
Table[{colrs[[i]],Thickness[0.013],Arrowheads[0.07],
Arrow[bazis[[i]]},{i,Length[bazis]}]];
gbaz1=Graphics[
Table[{colrs1[[i]],Thickness[0.013],Arrowheads[0.07],
Arrow[{bazis[[i,1]],amat . bazis[[i,2]]},{i,Length[bazis]}]];
dots1=
Table[(amat.dots[[j]]),{j,Length[dots]}];
coldots1=
Table[{Hue[j/Length[dots1]],Point[dots1[[j]]}],{j,Length[dots1]}];
Show[g0,Graphics[coldots1],gbaz,gbaz1

```

Dejstvo transformacije kojom se preslikavaju vektori u ravni R^2 reprezentovana matricom

$$A_{mat} = \begin{pmatrix} -1 & 0.3 \\ -0.3 & 1 \end{pmatrix}$$

ilustrovano je na slici 2.9. Tu je prikazan raspored tačaka, obojenih redom kontinualnim spektrom boja, koje se dejstvom operatora preslikavaju u niz kontinualno obojenih tačkica koje formiraju elipsu. Transformacijom su povezane tačkice istih boja.



Slika 2.9: Tačkice na krugu premeštene su dejstvom transformacije u elipsu. Dejstvo transformacije ilustrovano je i preko vektora bazisa crveni i plavi, koji su prešli u tamno crveni i tamno plavi.

```

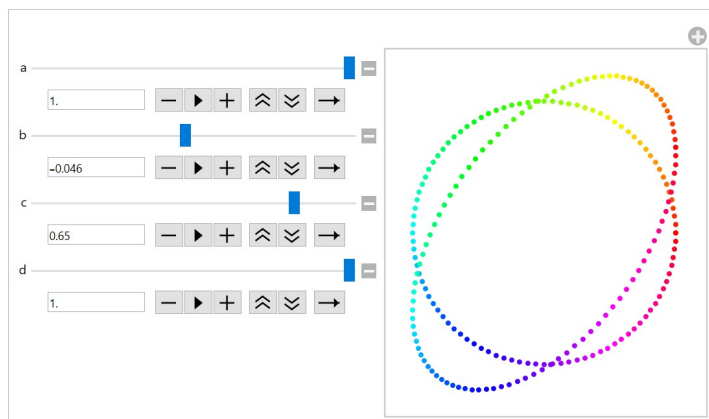
Manipulate[
Amat={{a,b},{c,d}};
dots1=
Table[(Amat . dots[[j]]),{j,Length[dots]}];
coldots1=
Table[{Hue[j/Length[dots1]],Point[dots1[[j]]}],{j,Length[dots1]}];
Show[g0,Graphics[coldots1],
{a,-1,1},{b,-1,1},{c,-1,1},{d,-1,1}]

```

Naredbom *Manipulate* omogućena je kontinualna promena matričnih elemenat, a sa njom istovremeno animacija dejstva dobijene matrice na vektore položaja, crtajući istovremeno početni raspored tačaka (krug) kao i novi oblik

2.1.3 Preslikavanje geometrijske figure u ravni.

Ilustracija dejstva linearnog operatora na vektore u realnoj ravni, realizovano je preslikavanjem



Slika 2.10: Istovremeno su prikazane inicijalne tačke koje formiraju krug, koji se nakon delovanja matrice zadatih matričnim elementima na kontrolnom panelu sa klizačima, transformiše u elipsu. Položaj tačke određene boje na elipsi dobijen je dejstvom matrice na položaj tačke odgovarajuće boje na krugu.

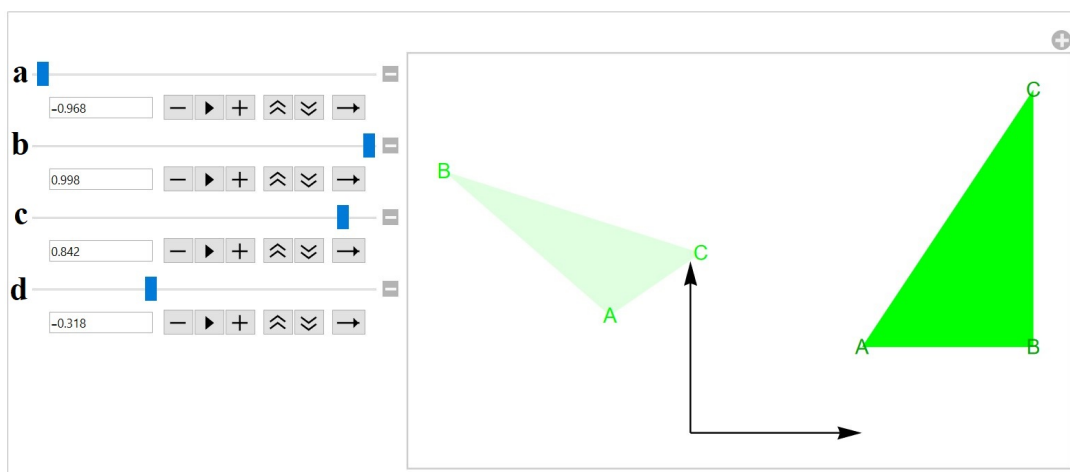
trougla. Trougao, crtan u okruženju *Graphics* zadaje se uz pomoć naredbe

$$\text{Triangle}[\{\{x_1, y_1\}, \{x_2, y_2\}, \{x_3, y_3\}\}],$$

gde su parovi $\{x_i, y_i\}$, $i = 1, 2, 3$ koordinate temena trougla, čineći listu koja je argument funkcije. Sa trouglom i njegovim likom, nastalim pod dejstvom linearne transformacije, istovremeno su prikazani bazisni vektori \mathcal{R}^2 ravni zajedno sa oznakama temena trouglova.

```
lmlin={Text[Style["A",Darker[Green],Medium],{1,0.5}],
Text[Style["B",Darker[Green],Medium],{2,0.5}],
Text[Style["C",Darker[Green],Medium],{2,2}]};
cooin={{1,0.5},{2,0.5},{2,2}};
coosys=Transpose[{0IdentityMatrix[2],IdentityMatrix[2]}];
coosys=Graphics[Arrow[#]&/@coosys];
gin=Show[Graphics[{Green,Triangle[cooin]}],Graphics[lmlin],coosys];
Manipulate[
Amat={{a,b},{c,d}};
lbl={Text[Style["A",Green,Medium],Amat.{1,0.5}],
Text[Style["B",Green,Medium],Amat.{2,0.5}],
Text[Style["C",Green,Medium],Amat.{2,2}]};
Show[
Graphics[
{LightGreen,Triangle[Amat.#&/@cooin]}],gin,Graphics[lbl]],
{a,-1,1},{b,-1,1},{c,-1,1},{d,-1,1}]
```

Najpre su definisani početni trougao i oznake njegovih temena zadavanjem liste koordinata *cooin* i tekst sa određenim sadržajem, veličinom, bojom i koordinatama *lmlin*. Od ovih elemenata zajedno sa koordinatnim sistemom napravljen je grafik *gin*. Naredbom *Manipulate*, menjani su matrični elementi transformacije koja preslikava zadati trougao, tako da je svaki element moguće menjati u rasponu od -1 do 1. Matrica deluje na koordinate oznaka u listi *lbl* kao i na koordinate inicijalnog trougla, što je realizovano funkcionalnim programiranjem, nakon čega se pravi grafik koji se istovremeno prikazuje sa *gin*. Trougao obojen zelenom bojom čija su temena označena sa $\triangle ABC$ i koordinatni sistem su nepokretni. Na vektore u ravni, kojima je određen položaj temena ovog trougla deluje se zadatim operatorom pri čemu se dobijaju temena njegove slike, svetlo zelenog trougla.



2.2 Trodimenzionalni grafički prikaz, objekti i opcije

Grafički prikaz površina i tačka gledišta

Naredba

$$\text{Plot3D}[f, \{x, x_{\min}, x_{\max}\}, \{y, y_{\min}, y_{\max}\}]$$

generiše trodimenzionalni (3D) grafik, površinu funkcije dve promenljive $f(x, y)$ u opsegu nezavisno promenljivih $x \in [x_{\min}, x_{\max}]$ i $y \in [y_{\min}, y_{\max}]$. Zavisnost električnog potencijala u YZ ravni u okolini dva tačkasta naelektrisanja Q i $-Q$, redom pozicionirana u tačkama $(0, 0, d)$ i $(0, 0, -d)$ je površina koja se može ilustrovati kao 3D grafik. Ukupan električni potencijal u proizvoljnoj tački prostora jednak je zbiru potencijala ove dve čestice. Primenom Kulonovog zakona dobija se prostorna zavisnost električnog potencijala

$$V(x, y, z) = kQ \left(\frac{1}{\sqrt{x^2 + y^2 + (z - d)^2}} - \frac{1}{\sqrt{x^2 + y^2 + (z + d)^2}} \right), \quad (2.5)$$

gde je k kulonova konstanta. Potencijal će biti prikazan u YZ ravni $x = 0$ u jedinicama kQ/d gde su jedinice koordinata date u d , a funkcija (2.5) koja se ilustruje postaje

$$V = \left(\frac{1}{\sqrt{x^2 + y^2 + (z - 1)^2}} - \frac{1}{\sqrt{x^2 + y^2 + (z + 1)^2}} \right).$$

Najpre je definisana funkcija potencijala u zavisnosti od prostornih koordinata, a potom pri određenim uslovima crtana je funkcija na YZ ravni na domenu definisanom kao argumenti naredbe *Plot3D*.

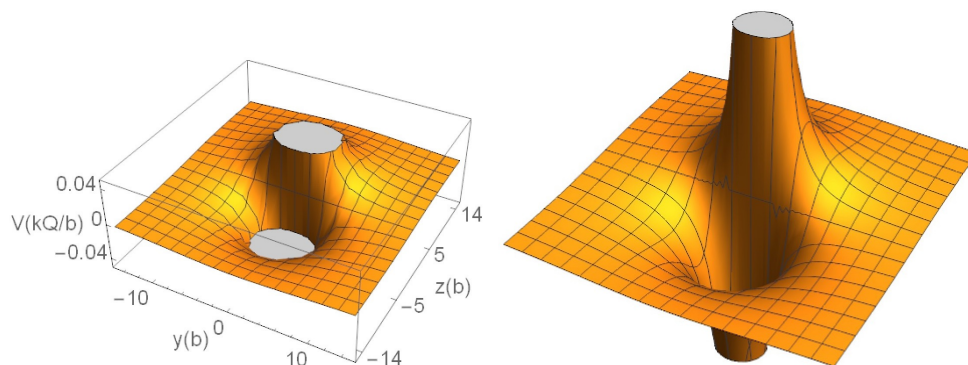
```
v[x_, y_, z_] := 1/Sqrt[x^2 + y^2 + (z - 1)^2] - 1/Sqrt[x^2 + y^2 + (z + 1)^2];
Plot3D[v[0, y, z], {y, -15, 15}, {z, -15, 15},
Ticks -> {Automatic, {-14, -5, 5, 14}}, {-0.04, 0, 0.04}},
AxesLabel -> {"y(b)", "z(b)", "V(kQ/b)"}]
```

Skala na osama definisana je naredbom *Ticks* → *None* bez zareza, sa tankim zarezima *Ticks* → *Automatic* ili sa istaknutim pozicijama *Ticks* → $\{x_1, x_2, \dots\}$.

Da bi se što bolje istakle karakteristike dipolnog potencijala biće upotrebljene opcije naredbe *Plot3D* : definisanje opsega u kom se prikazuje funkcija $V[0, y, z]$ sa *PlotRange* → $\{V_{\min}, V_{\max}\}$, broj inicijalnih tačaka *npts* u svakom smeru nezavisno promenljivih zahteva se sa *PlotPoints* →

npts, izmena odnosa ivica kutije (duž kojih su postavljene ose) zadana je sa *BoxRatios* $\rightarrow \{r_x, r_y, r_z\}$ pri čemu je u datom primeru odnos $\{1, 1, 1\}$ i prikaz bez osa zahtevan sa *Axes* \rightarrow *False*.

```
g3d=Plot3D[v[0,y,z],{y,-15,15},{z,-15,15},
PlotRange->{-0.12,0.12},
PlotPoints->50,
BoxRatios->{1,1,1},
Boxed->False,
Axes->False]
```



Slika 2.11: Levo: 3D grafik površine potencijala diopola u YZ ravni sa definisanom i označenom skalom. Desno: 3D grafik potencijala sa drugačijim odnosom razmere osa, bez kutije koja ga okružuje i koordinatnih osa.

Jedna od opcija naredbe *Plot3D* je *ViewPoint*, kojom se definišu koordinate tačke posmatranja trodimenzionalnog grafika funkcije. Ona se može koristiti za definisanje tačke posmatranja bilo kog trodimenzionalnog grafičkog objekta. Koordinate iz kojih se posmatra grafik određuju se u odnosu na koordinatni početak smešten u centru kutije koja okružuje grafički objekat ili grafik funkcije. Ose ovog desnog koordinatnog sistema u kom se definiše tačka gledanja paralelne su ivicama kutije, a jedinica na skali jednaka je najdužoj ivici kutije. Za neke specijalne vrednosti pogled seže: frontalno iz pozitivnog/negativnog duž negativnog/ pozitivnog dela *y* ose za *ViewPoint* $\rightarrow \{0, \pm y, 0\}$, odozgo/odozdo ka dole/gore pri izboru *ViewPoint* $\rightarrow \{0, 0, \pm z\}$, frontalno i odozgo/odozdo *ViewPoint* $\rightarrow \{0, y, \pm z\}$.

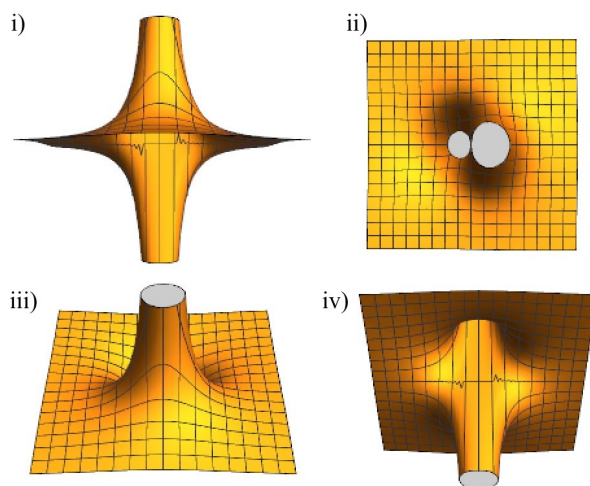
```
Show[g3d,ViewPoint->{0, 2, 0}]
Show[g3d,ViewPoint->{0, 0, 2}]
Show[g3d,ViewPoint->{0, 2, 2}]
Show[g3d,ViewPoint->{0, 2, -2}]
```

Električno polje dipola se može izračunati kao negativan gradijent potencijala $\vec{E} = -\nabla V$. Program Mathematica poseduje naredbe kojima se simbolički računaju dejstva vektorskih, diferencijalnih operatora na skalarne i vektorske funkcije. Za nalaženje gradijenta skalarne funkcije koristi se

$$\text{Grad}[f, \{x, y, \dots\}]$$

čiji su argumenti funkcija $f(x, y, \dots)$ a izlaz je vektorsko polje odnosno gradijent ∇f , u vidu liste funkcija od zadatih koordinata na mestu drugog argumenta. Primera radi, gradijent opšte funkcije $f(x, y, z)$ u Dekartovim koordinatama se traži sa

```
Grad[f[x,y,z],{x,y,z}]
```



Slika 2.12: 3D prikaz potencijala dipola u YZ ravni: i) frontalni prikaz sa $ViewPoint \rightarrow \{0, 2, 0\}$, pogledom odozgo $ViewPoint \rightarrow \{0, 0, 2\}$, pogledom sa strane i odozgo $ViewPoint \rightarrow \{0, 2, 2\}$ i pogledom sa strane i odozdo $ViewPoint \rightarrow \{0, 2, -2\}$.

čije je rešenje $\left\{ f^{(1,0,0)}(x, y, z), f^{(0,1,0)}(x, y, z), f^{(0,0,1)}(x, y, z) \right\}$. Osim u Dekartovim, dejstvo nabra, vektorskog i diferencijalnog operatora, na skalarnu (kao i vektorsku) funkciju može se primeniti u ortonormiranom bazu cilindričnih koordinata

```
Grad[f[r, \[Theta], z], {r, \[Theta], z}, "Cylindrical"]
```

odakle se dobija $\left\{ f^{(1,0,0)}(r, \theta, z), \frac{f^{(0,1,0)}(r, \theta, z)}{r}, f^{(0,0,1)}(r, \theta, z) \right\}$, ili sfernih koordinata

```
Grad[f[r, \[Theta], \[Phi]], {r, \[Theta], \[Phi]}, "Spherical"]
```

pri čemu je izlaz iz programa $\left\{ f^{(1,0,0)}(r, \theta, \phi), \frac{f^{(0,1,0)}(r, \theta, \phi)}{r}, \frac{\csc(\theta) f^{(0,0,1)}(r, \theta, \phi)}{r} \right\}$.

Komponente električnog polja u okolini dipola iz

```
Efld == -Grad[v[x, y, z], {x, y, z}]
```

Izlaz prethodne linije je vektorska funkcija

$\vec{E} = \{E_x(x, y, z), E_y(x, y, z), E_z(x, y, z)\}$, lista funkcija komponenti vektorskog polja u zavisnosti od prostornih koordinata, gde su funkcije komponenti date u arbitrarnim jedinicama

$$E_x = \frac{x}{(x^2 + y^2 + (z-1)^2)^{3/2}} - \frac{x}{(x^2 + y^2 + (z+1)^2)^{3/2}},$$

$$E_y = \frac{y}{(x^2 + y^2 + (z-1)^2)^{3/2}} - \frac{y}{(x^2 + y^2 + (z+1)^2)^{3/2}},$$

$$E_z = \frac{z-1}{(x^2 + y^2 + (z-1)^2)^{3/2}} - \frac{z+1}{(x^2 + y^2 + (z+1)^2)^{3/2}}.$$

Dobijena vektorska funkcija upotrebljena je za crtanje vektorskog polja naredbom

```
VectorPlot3D[{E_x, E_y, E_z}, {x, x_min, x_max}, {y, y_min, y_max}, {z, z_min, z_max}],
```

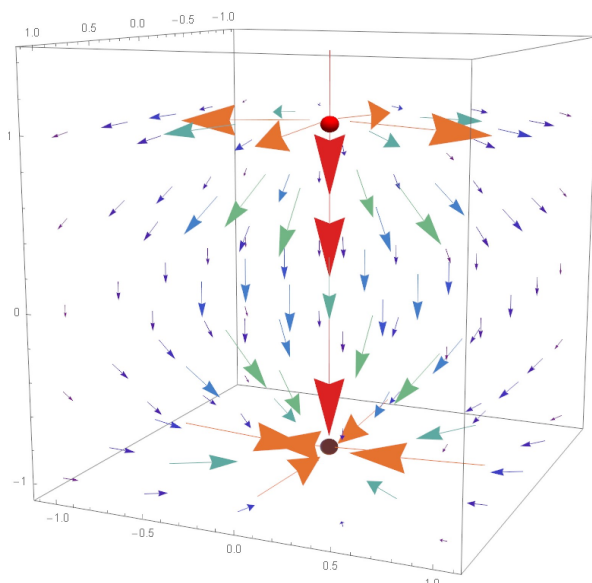
gde je prvi argument lista komponenti polja u funkciji koordinata na intervalima zadatim u nastavku.

```
DipCh=Graphics3D[{Sphere[{0,0,-1},-0.05],
Red,Sphere[{0,0,1},0.05],Darker[Blue]}];
EvecFld=VectorPlot3D[Efld,{x,-1,1},{y,-1,1},{z,-1,1},PlotRange->All,
VectorPoints->5,VectorStyle->"Arrow3D",
VectorColorFunction->"Rainbow"];
Show[EvecFld, DipCh]
```

Pozitivno naelektrisanje, izvor električnog polja, prikazano je crvenom sferom

$$Sphere\{x,y,z\},r\},$$

gde su $\{x,y,z\}$ koordinate centra sfere radijusa r , dok je ponor polja u negativnom naelektrisanju prikazanom plavom sferom. Strelice dužine srazmerne jačini polja, od većeg ka manjim intenzitetima bojene su od crvene ka plavoj, opcionom naredbom *VectorColorFunction* → "Rainbow".



Slika 2.13: Električno vektorsko polje u okolini naelektrisanja dipola. Osim dužinom strelica, intenzitet polja raste od plave ka crvenoj boji.

Vektorsko polje analizirano je i predstavljeno u XZ ravni za $y = 0.4$, a dobija se izvršavanjem ćelije

```
Exz=-Grad[v[x,0.4,z],{x,z}];
VectorPlot[Exz,{x,-1,1},{z,-1,1},VectorPoints->9,
VectorScale->Medium,VectorColorFunction->"Rainbow"]
StreamDensityPlot[Exz,{x,-1,1},{z,-1.2,1.2}]
```

gde su dobijene funkcije komponenti

$$E_x = \frac{x}{(x^2+(z-1)^2+0.16)^{3/2}} - \frac{x}{(x^2+(z+1)^2+0.16)^{3/2}},$$

$$E_y = \frac{z-1}{(x^2+(z-1)^2+0.16)^{3/2}} - \frac{z+1}{(x^2+(z+1)^2+0.16)^{3/2}}.$$

Vektorsko polje u ravni nacrtano je funkcijom

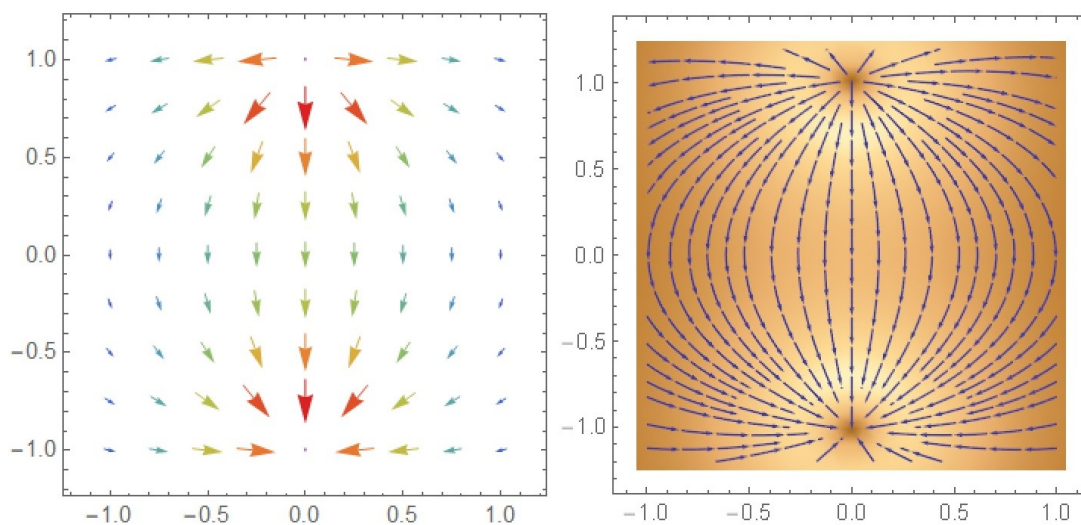
$$VectorPlot[\{E_x, E_z\}, \{x, -1, 1\}, \{z, -1, 1\}],$$

pri čemu je opciono definisano 9 redova strelica sa *VectorPoints* → 9. Na ovom prikazu intenzitet polja prikazan je dužinom i bojom strelica. Tok električnog polja u zadatoj ravni ilustruje

se pomoću

```
StreamDensityPlot[Exz, {x, -1, 1}, {z, -1.2, 1.2}]
```

gde je prvi argument lista zavisnosti komponenti od prostornih koordinata čiji intervali su zadati u nastavku.



Slika 2.14: Vektorsko polje (levo) i njegov tok (desno) u XZ ravni za $y = 0.4$.

2.2.4 Elektromagnetni talas.

U ovom zadatku biće prikazana propagacija linearno polarizovanog, sinusoidalnog i ravnog elektromagnetnog talasa koji putuje duž x -ose. Elektromagnetni talas je periodična promena električnog i magnetnog polja, koja putuje kroz prostor. Pravci oscilacija polja su uzajamno ortogonalni, a oba su okomiti na pravac prostiranja. Ukoliko se oscilacije polja odvijaju u jednoj ravni, radi se o linearno polarizovanom talasu. Kod ravnog talasa koji se prostire duž x -ose, električno (i magnetno) polje funkcije su od položaja x i vremena t ,

$$\vec{E} = \vec{E}(x, t) = E(x, t)\vec{e}_z, \quad \vec{B} = \vec{B}(x, t) = B(x, t)\vec{e}_y$$

te zadovoljavaju diferencijalne jednačine

$$\frac{\partial^2 \vec{E}}{\partial x^2} - \frac{1}{v^2} \frac{\partial^2 \vec{E}}{\partial t^2} = 0,$$

$$\frac{\partial^2 \vec{B}}{\partial x^2} - \frac{1}{v^2} \frac{\partial^2 \vec{B}}{\partial t^2} = 0,$$

gde je v brzina prostiranja elektromagnetnog talasa. Elektromagnetno zračenje čine elektromagnetni talasi, koji predstavljaju oscilacije električnog i magnetnog polja. Primer talasnih funkcija koje opisuju promene polja elektromagnetnog talasa

$$\vec{E} = \vec{E}_0 \sin(\omega t - kx), \quad \vec{B} = \vec{B}_0 \sin(\omega t - kx) \quad (2.6)$$

i zadovoljavaju gore date talasne jednačine.

Ilustracija promene električnog i magnetnog polja u zavisnosti od prostornih koordinata u fiksnom vremenskom trenutku biće realizovana naredbom

```
ParametricPlot3D[{{f_x, f_y, f_z}, {g_x, g_y, g_z}, ...}, {x, x_min, x_max}].
```

Ovom naredbom generišu se parametarske krive u trodimenzionalnom prostoru, parametrizovane varijablom x u opsegu od x_{min} do x_{max} . Radi jednostavnosti smatra se da je trenutak u kom se posmatra prostorna promena $t = 0$. Zadatak je rešen tako da su u prvoj fazi pripremljene parametarske linije, plava koja opisuje vrhove strelica električnog polja

```
ParametricPlot3D[{x,0,Amplt*Sin[x]},{x,0,lengt},PlotStyle->Darker[Blue]]
```

i crvena

```
ParametricPlot3D[{x,Amplt*Sin[x],0},{x,0,lengt},PlotStyle->Darker[Red]];
```

u kojoj se završavaju strelice magnetnog polja. Slika talasa u trenutak $t = 0$, kada je talasna funkcija $\sin(\omega t - kx) = \sin(-kx) = \sin x$, ako se uzme $k = -1$. Dužina prikazanog talasa, koji se prostire duž x -ose, zadata je promenljivom $lengt$, a amplituda sa $Amplt$. Ose Y i Z koordinatnog sistema $coosys$ imaju dužine amplituda dok je dužina x -ose $lengt$, prikazane su sa oznakama osa lbl radi lakšeg praćenja. Ispisivanje oznaka osa kao i polja na grafiku, kao na primer električno polje, realizovane su unutar okruženja *Graphics3D* koristeći naredbu

```
Text[Style["E",16,Bold,Italic,Darker[Blue]},{x,0,1.4Amplt*Sin[x]},{x->Pi/2}],
```

pri čemu je položaj oznake praćen oblikom talasa i postavljen na rastojanju $\pi/2$ od koordinatnog početka.

Opcije kojima se uređuje tekst date su u *Style*, a opcije za polje E su redom

, *Bold, Italic, Darker[Blue]* i veličina 16. Položaj teksta je zadat sa

```
{x,0,1.4Amplt*Sin[x]},{x->Pi/2}
```

Za konstrukciju koordinatnog sistema korišćena je naredba *DiagonalMatrix[lengt, Amplt, Amplt]*, čime su dobijene tri liste, odnosno koordinate vrhova vektora, dok su njihovi počeci definisani sa *0IdentityMatrix[3]*. Ovi setovi koordinata čine jedan skup, koji nakon transponovanja daje tri para koordinata u 3D prostoru. Svaki od ovih parova definiše početak i kraj vektora, uređenih opcijama

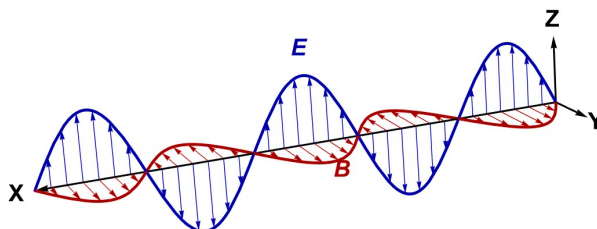
```
Graphics3D[{Thickness[0.003],Arrowheads[0.02],Arrow[coosys]}].
```

```
lengt = 5 Pi;(*duzina x ose*)
Amplt = 2;(*amplituda oscilacija*)
(*talasni broj je k=-1*)
(*talasi- poremećaj polja E (plavo) i B (crveno)*)
gEfld =
ParametricPlot3D[{x,0,Amplt*Sin[x]},{x,0,lengt},
PlotStyle->Darker[Blue]];
gBfld=
ParametricPlot3D[{x,Amplt*Sin[x],0},{x,0,lengt},
PlotStyle->Darker[Red]];
(*koordinatni sistem*)
coosys={0IdentityMatrix[3],DiagonalMatrix[{lengt,Amplt,Amplt]}};
coosys=Transpose[coosys];
coosys=
Graphics3D[{Thickness[0.003],Arrowheads[0.02],Arrow[coosys]}];
(*oznake osa*)
lbl=Show[
Graphics3D[Text[Style["X",16,Bold,Black],{lengt+0.5,0,0}],
Graphics3D[Text[Style["Y",16,Bold,Black],{0,Amplt+0.5,0}],
Graphics3D[Text[Style["Z",16,Bold,Black],{0,0,Amplt+0.5}],
Boxed->False];
(*oznake polja*)
lblFlds=Show[
Graphics3D[
Text[Style["E",16,Bold,Italic,Darker[Blue]],
{x,0,1.4Amplt*Sin[x]},{x->Pi/2}],
Graphics3D[Text[Style["B",16,Bold,Italic,Darker[Red]],
{x,1.4Amplt*Sin[x],0},{x->Pi/2}],Boxed->False];
```

Predstoji crtanje strelica, koje oslikavaju prostorno promenljivo vektorsko polje. Tehnički, polja

su realizovana kao strelice čiji su počeci nizani duž x -ose na rastojanju dz jedna od druge. Vektori električnog polja E usmereni su u pravcu z -ose, prostorno promenljive dužine $Amplt * \{0,0,1\} \sin i * dz$, tako da vrh strelice prati plavu konturu $Arrow[\{\{i * dz, 0, 0\}, \{i * dz, 0, 0\} + Amplt * \{0, 0, 1\} \sin i * dz\}]$. Ortogonalni na električno, tamno crveni vektori magnetnog polja su strelice veličine $Arrowheads[0.015]$ potiču iz x -ose sa istom učestalošću pojavljivanja i završavaju u crvenoj konturi prateći njenu prostornu promenu $Arrow[\{\{i * dz, 0, 0\}, \{i * dz, 0, 0\} + Amplt * \{0, 1, 0\} \sin (i * dz)\}]$. Svi dobijeni trodimenzionalni grafički objekti: koordinatni sistem *coosys* sa pripadajućim oznakama *lbl*, konture koje opisuju promenu električnog i magnetnog polja *gEfld* i *gBfld* njima odgovarajući vektori *gEvectrs* i *gBvectrs* sa oznakama *lblFlds* prikazani su istovremeno na jednom grafiku *g0* primenom naredbe *Show*, bez graničnog okvira *Boxed* \rightarrow *False*.

```
dz=Pi/7;(*koraci vektora*)
(*vektori elektricnog polja E duz x-ose*)
gEvectrs=
Graphics3D[Table[{Darker[Blue],Arrowheads[0.015],
Arrow[{{i*dz,0,0},{i*dz,0,0}+Amplt*{0,0,1}Sin[i*dz]}]},
{i,lengt/dz}]];
(*vektori magnetnog polja B duz x-ose*)
gBvectrs=Graphics3D[Table[{Darker[Red],Arrowheads[0.015],
Arrow[{{i*dz,0,0},{i*dz,0,0}+Amplt*{0,1,0}Sin[i*dz]}]},
{i,lengt/dz}]];
(*istovremeni prikaz grafika na jednom mestu*)
g0=Show[
coosys,
gEfld,
gBfld,
gEvectrs,
gBvectrs,
lbl,
lblFlds,
Boxed -> False]
```



Slika 2.15: Propagacija ravnog elektromagnetnog talasa.

2.3 Interaktivne animacije

Naredbom

$$Manipulate[expr, \{v, v_{min}, v_{max}, dv\}]$$

generiše se verzija izraza (grafika) *expr* sa kontrolnim panelom koji omogućava interaktivnu manipulaciju vrednosti promenljive v u opsegu između v_{min} i v_{max} sa korakom dv , pri čemu *expr* može biti primera radi dvodimenzionalni ili trodimenzionalni grafik ili izraz. Dozvoljene su diskretna i kontinualna promena vrednosti v , a moguća je i promena više promenljivih koje

figurišu u izrazima u argumentu naredbe. Program Mathematica simulira kretanje prikazujući sekvence grafika, poput brze promene uzastopnih slika filma.

=====

2.3.5 Propagacija elektromagnetnog talasa.

Cilj je da se svi grafički elementi prikazani na slici 2.15, sinusne konture, strelice kojima su reprezentovana vektorska polja, menjaju sa parametrom t (vremenom), čime se animira propagacija talasa. Svuda gde figuriše talasna funkcija (2.6), uneta je promenljiva t

$$\sin x \rightarrow \sin(t - x),$$

a potom je u vidu argumenta od *Manipulate* zadat i njen opseg, pri čemu je uzeto $\omega = 1$. Promenljive koje se menjaju sa vremenom, unose se kao argument funkcije *Manipulate*[..., { $t, 0, 5\pi$ }], Rezultat su slike koje se menjaju u zavisnosti od t , animirajući vremenski promenljivo vektorsko polje u vidu sinusnog talasa.

```
Manipulate[
(*konture promene intenziteta vektora*)
gEfld=ParametricPlot3D[{x,0,Amplt*Sin[t-x]},{x,0,lengt},
PlotStyle -> Darker[Blue]];
gBfld=ParametricPlot3D[{x,Amplt*Sin[t-x],0},{x,0,lengt},
PlotStyle -> Darker[Red]];
(*vektori elektricnog polja E duz x-ose*)
gEvectrs=
Graphics3D[Table[{Darker[Blue],Arrowheads[0.015],
Arrow[{{i*dz, 0, 0}, {i*dz, 0, 0}+Amplt*{0,0,1}Sin[t-i*dz]}]},
{i, lengt/dz}]];
(*vektori magnetnog polja B duz x-ose*)
gBvectrs=Graphics3D[Table[{Darker[Red],Arrowheads[0.015],
Arrow[{{i*dz,0,0},{i*dz,0,0}+Amplt*{0,1,0}Sin[t-i*dz]}]},
{i, lengt/dz}]];
(*istovremeni prikaz grafika na jednom mestu*)
g0=Show[
coosys,
gEfld,
gBfld,
gEvectrs,
gBvectrs,
lbl,
lblFlds,
Boxed->False], {t, 0, 5 Pi}]
```

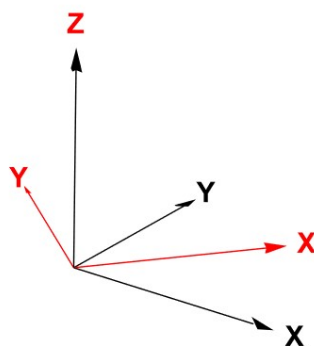
2.3.6 Koordinatni sistem i dejstvo rotacija na vektore.

Uz pomoć grafičkih objekata u trodimenzionalnom prostoru biće prikazan koordinatni sistem zajedno sa oznakama osa. Potom, koristeći naredbu *Manipulate* konstruisan koordinatni sistem moći će se rotirati oko z -ose.

```
(*definisavanje vektora, parovi koordinata*)
lst1=Transpose[{0IdentityMatrix[3],IdentityMatrix[3]}];
lbls={
Text[Style["X",Large,Bold],{1.1,0,0}],
Text[Style["Y",Large,Bold],{0,1.1,0}],
Text[Style["Z",Large,Bold],{0,0,1.1}]};
```

Za crtanje koordinatnog sistema korišćeni su vektori u trodimenzionalnom prostoru dobijeni naredbom `Arrow[coo1,coo2]`, definisani sa 2 para koordinata, početkom *coo1* i vrhom vektora *coo2*. Početak svih vektora je u koordinatnom početku, određen sa $(0,0,0)$, dok su vrhovi strelica na pravcima x, y i z -ose redom zadati $(1,0,0)$, $(0,1,0)$, $(0,0,1)$. Podaci, parovi u vidu početna i krajnja koordinata vektora nepokretnog sistema, nalaze se u listi *lst1*, dok su nazivi koordinatnih osa zadati naredbom `Text`, kao na primer za x -osu `Text[Style["X", Large, Bold], {1.1,0,0}]` i čine elemente liste *lbls*.

```
Manipulate[
MRot=RotationMatrix[\[CurlyPhi],{0,0,1}];(*matrica rotacije*)
g0=Graphics3D[
Table[{lbls[[i]],Arrow[lst1[[i]]}],{i,3}],Boxed->False];
lbls2={
Text[Style["X",Large,Bold,Red],MRot.{1.1,0,0}],
Text[Style["Y",Large,Bold,Red],MRot.{0,1.1,0}],
Text[Style["Z",Large,Bold,Red],MRot.{0,0,1.1}]];
g=Graphics3D[
Table[{lbls2[[i]],Red,Arrow[{lst1[[i,1]],MRot.lst1[[i,2]]}],
{i,3}],Boxed->False];
Show[g0,g],{\[CurlyPhi],0,Pi}]
```



Matrica rotacije $\mathcal{R}_{\vec{n}}(\varphi)$ za ugao φ oko ose \vec{n} , korišćena je kao transformacija kojom se rotiraju vektori, dobijena je naredbom

$$\text{RotationMatrix}[\varphi, \vec{n}].$$

Vektori kojima su reprezentovane ose sa svojim vizuelnim karakteristikama crtani su pod okruženjem `Graphics3D`, zajedno sa odgovarajućim oznakama. Matricom rotacije *MRot* delovano je na vektor položaja vrhova strelica (na primer `MRot.lst1[[i,2]]`) i koordinata teksta kojim se označavaju ose (`MRot.{1.1,0,0}`- pomeranje oznake *X*). Pokretni sistem, ose zajedno sa oznakama obojene su crvenom dok je početni crnom i sve je prikazano na istom grafiku kako bi se uočio efekat rotacije.

2.3.7 Interferencija.

U elastičnom medijumu koji se prostire po XY -ravni nalaze se dva tačkasta izvora S_1 i S_2 na međusobnom rastojanju d postavljena duž x -ose. Svaki izvor emituje talas koji se širi koncentrično. Talasi su iste frekvencije i amplitude i u istim fazama na izvorima (koherentni izvori). Poremećaj medijuma z_1 u tački P koja se nalazi na rastojanju r_1 od S_1 i r_2 od S_2 je

$$z_1 = \frac{Y}{r_1} \sin(kr_1 - \omega t),$$

a poremećaj z_2 sredine usled talasa emitovanog sa izvora S_2 je

$$z_2 = \frac{Y}{r_2} \sin(kr_2 - \omega t),$$

gde su t vreme, Y konstanta, k talasni broj, a ω kružna frekvencija. Biće simulirana superpozicija talasa emitovanih iz dva izvora (pri rastojanju $d = 30/k$). Rezultat poremećaja z u tački P na osnovu principa superpozicije je

$$z = z_1 + z_2.$$

Ako su poremećaj, udaljenost i vreme merenja, radi jednostavnosti implementacije u programskom kodu, dati redom u sledećim jedinicama Yk , $1/k$ i $1/\omega$, funkcije poremećaja koje se programiraju postaju

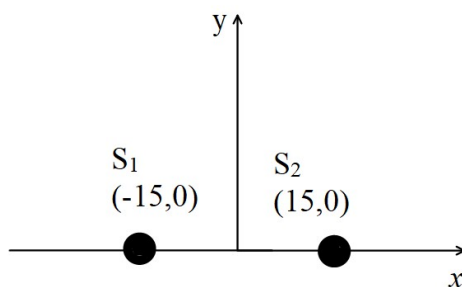
$$z_1 = \frac{1}{r_1} \sin(r_1 - t)$$

$$z_2 = \frac{1}{r_2} \sin(r_2 - t),$$

dok je rastojanje d između izvora 30. Koordinatni sistem je postavljen tako da su koordinate izvora S_1 i S_2 redom $(-15, 0)$ i $(15, 0)$. Rastojanje tačke P čije su koordinate (x, y) od izvora S_1 i S_2 su redom:

$$r_1 = \sqrt{(x + 15)^2 + y^2}$$

$$r_2 = \sqrt{(x - 15)^2 + y^2}.$$



Najpre su isprogramirane funkcija gore definisanih veličina $r_1(x, y)$, $r_2(x, y)$, $z_1(r_1, t)$, $z_2(r_2, t)$, $z(x, y, t)$.

```
r1[x_, y_] := Sqrt[(x + 15)^2 + y^2]
r2[x_, y_] := Sqrt[(x - 15)^2 + y^2]
z1[r1_, t_] := Sin[r1 - t]/r1
z2[r2_, t_] := Sin[r2 - t]/r2
z[x_, y_, t_] := z1[r1[x, y], t] + z2[r2[x, y], t]
```

Kako bi konstruktivna i destruktivna superpozicija bile što vidljivije u ravni $[-30, 30] \times [-30, 30]$, vrednosti funkcija poremećaja z dobijene superpozicijom su prikazane u opsegu $z \in [-0.7, 0.7]$, čime su odsečene vrednosti u neposrednoj okolini izvora gde bi divergirale.

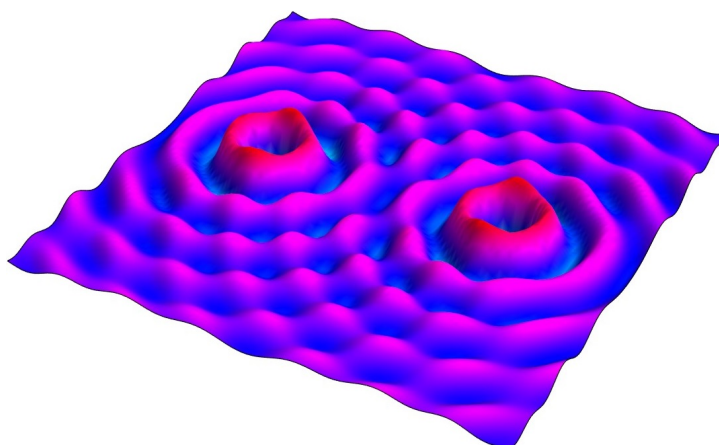
```
PlotRange->{{-30, 30}, {-30, 30}, {-1, 0.7}}
```

Radi prikaza površina grafika, korišćena je opcija za grafičke funkcije koja specificira funkciju koja se primenjuje za bojenje elemenata *ColorFunction*. Bojenje površine prema njenoj visini, kako bi se ilustrovala superpozicija, realizovano je sa

ColorFunction → *Function*[{*x*, *y*, *z*}, *Hue*[*z*]]. Interaktivnom manipulacijom, realizovanom naredbom *Manipulate* generisani su 3D grafici funkcije $z(x, y, t)$, crtani sa *Plot3D* u različitim vremenskim trenucima $t \in [0, 3\pi]$ sa korakom $\Delta t = 2\pi/100$.

```
Manipulate[
Plot3D[z[x,y,t],{x,-30,30},{y,-30,30},
Boxed->False,Axes->False,Mesh->None,PlotPoints->100,
ColorFunction->Function[{x,y,z},Hue[z]],
PlotRange->{{-30,30},{-30,30},{-1,0.7}}],
{t,0,3Pi,2. Pi/100}]
```

Osim navedenog, u cilju lakšeg vizuelnog praćenja izbrisane su granice trodimenzionalne kutije u kojoj je smešten grafik *Boxed* → *False*, zatim ose grafika *Axes* → *False*, a na posletku i mreža *Mesh* → *None*.



Slika 2.16: Interferencija talasa.

Rezultat je animacija na kojoj je prikazana propagacija talasa od izvora i oblasti u kojoj se koncentrično šire formirajući interferencione minimume i maksimume.

2.3.8 Superpozicija dva pulsa koja se kreću u suprotnim smerovima.

Princip superpozicije može biti primenjen na dva ili više talasa koji istovremeno putuju kroz isti medijum. Ukoliko se talasi ukrste, nastavljaju propagaciju bez smetnji. Ukupni poremećaj medijuma u bilo kojoj tački u prostoru ili vremenu je prosto zbir doprinosa poremećaja od pojedinih talasa. Ovo važi za kontinuirane kao i za talase konačne dužine, talasne impulse. U nastavku biće animirano kretanje dva Gausova talasna pulsa, koja se u istom medijumu kreću u susret jedan drugom. Pri tom, talasi prolaze jedan kroz drugog bez smetnji uzrokujući deformaciju medijuma u obliku sume doprinosa pojedinih poremećaja. Gausovi pulsevi ne menjaju oblik tokom propagacije, što znači da su nedisperzivni. Koristeći interaktivne manipulacije, realizovane uz pomoć naredbe *Manipulate*, biće demonstriran princip superpozicije simulirajući interferenciju dva impulsa koji imaju oblik Gausove funkcije i putuju duž istog pravca. Poremećaj y opisan je funkcijom (Gausijanom) koja zavisi od položaja x i vremena t na sledeći način

$$y = Ae^{-\alpha(x+vt)^2},$$

gde je A amplituda poremećaja, α faktor koji opisuje širinu Gausijana a v brzina kretanja impulsa duž x ose.

Impulsima su pridružene funkcije zavisne od položaja i vremena

$$y_1 = A_1e^{-\alpha_1(x+v_1t+x_1)^2}, \quad y_2 = A_2e^{-\alpha_2(x+v_2t+x_2)^2},$$

gde će A_i, α_i, v_i biti parametri koje menja korisnik, kao i inicijalni položaji Gausijana x_i . Prema principu superpozicije, ukupan poremećaj u posmatranoj tački prostora, gde se u datom trenutku talasi prepliću, jednak je sumi doprinosa poremećaja pojedinih talasa. Funkcija koja opisuje oblik poremećaja nastalog superpozicijom jednaka je zbiru Gausijana koji opisuju impulse koji se sudaraju. Ukoliko su brzine suprotnog smera, poremećaji putuju jedan ka drugom.

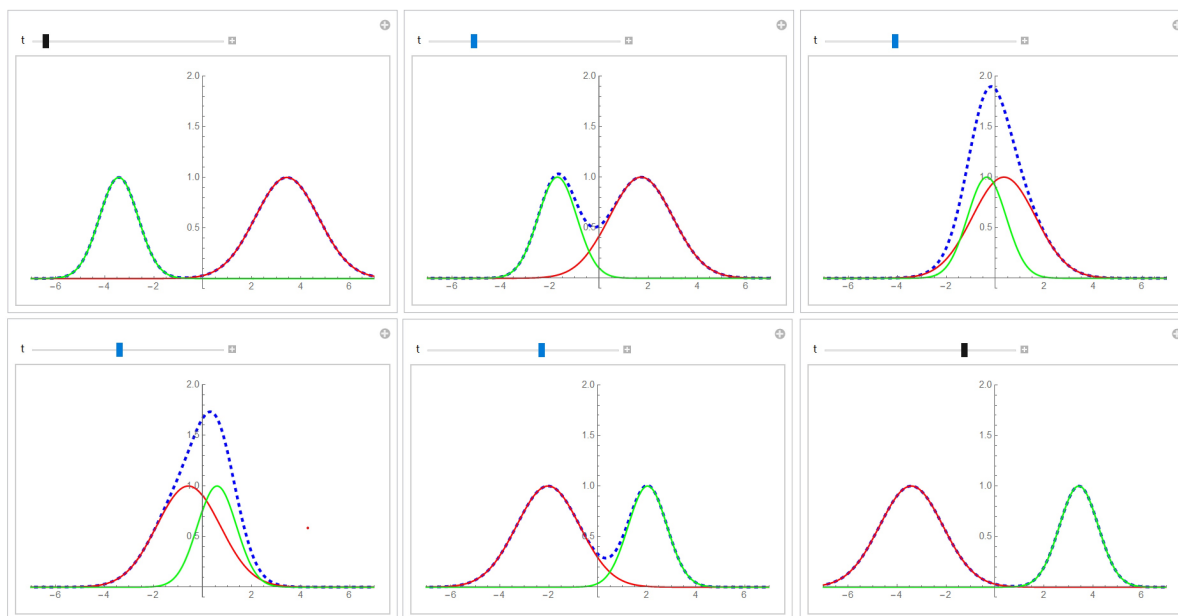
```
y1[x_, t_] := A1*Exp[-\[Alpha]1 (x + v1*t+x1)^2] (*1. poremecaj*)
y2[x_, t_] := A2*Exp[-\[Alpha]2 (x + v2*t+x2)^2] (*2. poremecaj*)
y[x_, t_] := y1[x, t] + y2[x, t] (*superpozicija*)
```

Opseg grafika zadat je promenljivom *range*. Varijablama koje definišu realne parametre nisu pridružene vrednosti već se vrši njihova zamena odgovarajućim vrednostima, definisanih u listi zamena *pmts*, koja sadrži amplitude, širine i početni položaji Gausijana kao i njihove brzine.

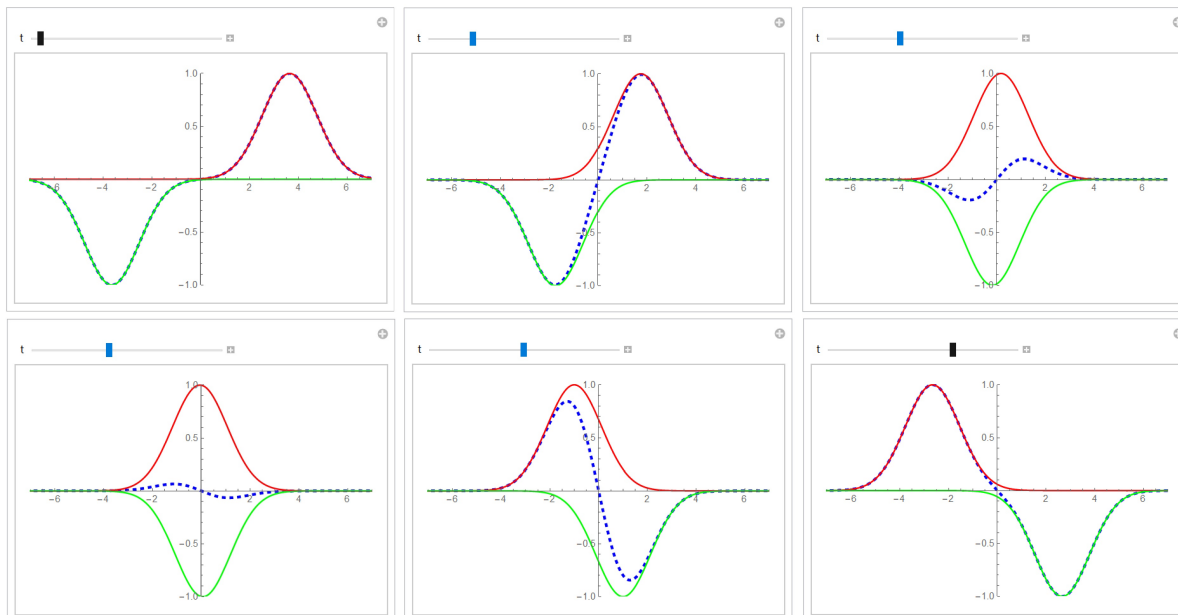
```
range={{-7,7},{-2,2}}; (*opseg grafika*)
pmts={\[Alpha]1->0.3, \[Alpha]2->0.8, A1->1,
A2->-1, v1->1, v2->-1, x1->-4, x2->4};
```

Ukoliko se izaberu iste širine Gausijana $\alpha_1 = \alpha_2$, a amplitude suprotnog znaka odnosno suprotnih faza, istih po apsolutnoj vrednosti, u trenutku sudara doći će do destruktivne superpozicije kada poremećaji budu jedan iznad drugog. Plava linija, kojom je prikazana zbir doprinosa poremećaja postaje ravna u trenutku susreta impulsa. Istovremeno se naredbom *Show* crtaju grafici plavom i crvenom bojom redom propagirajućih Gausijana 1 i 2, kao i njihova superpozicija (suma) prikazana plavom bojom. Parametar koji se menja naredbom *Manipulate*, u kojoj se istovremeno prikazuju grafici crtani u zavisnosti od koordinate $x \in [-10, 10]$, je vreme $t \in [0, 10]$. Slično prethodnom primeru, pogodnim izborom amplituda, može se animirati konstruktivna superpozicija. Ako su impulsi koji se kreću u susret jedan drugom iste faze, u trenutku sudara dolazi do konstruktivne superpozicije.

```
Manipulate[
  Show[Plot[y[x, t]/.pmts,{x,-10,10},PlotRange->range,
    PlotStyle->{Thickness[0.02],Blue}],
    Plot[y1[x, t]/.pmts,{x,-10,10},PlotRange->range,
    PlotStyle->Red],
    Plot[y2[x, t]/.pmts,{x,-10,10},PlotRange->range,
    PlotStyle -> Green]
  ,{t,0,10}]
```



Slika 2.17: Sudar Gausovih pulseva amplituda $A_1 = A_2 = 1$ i širina $\alpha_1 = 0.3$ (zeleni) i $\alpha_2 = 0.8$ (crveni). Superponirani poremećaj sredine označen je isprekidanom plavom linijom.



Slika 2.18: Animacija destruktivne interferencije Gausovih pulseva amplituda $A_1 = A_2 = -1$ i širina $\alpha_1 = \alpha_2 = 0.4$. Superponirani poremećaj sredine označen je isprekidanom plavom linijom.

3 Simboličke kalkulacije u programskom jeziku Wolfram

Ovo poglavlje bavi se mogućnostima simboličkog računa u programskom jeziku Wolfram Mathematica, u kom je moguće raditi sa opštim izrazima, varijablama ili simbolima podjednako operativno kao sa brojevima. Pri tome, tokom izvršavanja raznih operacija, izrazima ne moraju biti pridružene konkretne brojne vrednosti, a rezultat su novi izrazi, funkcije. Mnoge tehnike, razvijene za numerička računanja mogu se uopštiti i primeniti na simbolička. Simbolički softverski programi redukuju složene algebarske kalkulacije.

3.1 Transformacija algebarskih, trigonometrijskih izraza i ekstrakcija njihovih delova

Matematika poseduje mnogo funkcija kojima se vrše transformacije algebarskih izraza. Sledi pregled bazičnih standardnih algebarskih transformacija. Generalno, da bi se dobio izraz željene forme, često je neophodno eksperimentisati sa različitim transformacionim funkcijama.

Naredbom `Apart[expr, var]` upućuje se nalog za racionalizaciju izraza u vidu zbira članova sa minimalnim imeniocima, tretirajući sve osim varijable `var` kao konstante. Upotreba funkcije `Apart` ilustrovana je na sledećem primeru nalaženja parcijalne dekompozicije racionalnog izraza

$$\frac{5x^2 - 4x + 16}{(x^2 - x + 1)^2(x - 3)}$$

```
Apart[(5 x^2 - 4 x + 16)/((x^2 - x + 1)^2 (x - 3))]
```

Rešenje: $\frac{-2x-3}{(x^2-x+1)^2} + \frac{-x-2}{x^2-x+1} + \frac{1}{x-3}$.

Naredbom `Cancel[expr]` redukuje se zadati razlomak `expr` na najniže članove. Primenom `Cancel` na razlomak

$$\frac{x^2 - 5x + 4}{x - 1},$$

redukuje ga do na najniži član.

```
Cancel[(x^2 - 5 x + 4)/(x - 1)]
```

Rešenje: $-4 + x$.

`Collect[expr, x]` skuplja članove koji imaju isti stepen od `x`.

`Expand[expr]` će razviti proizvode i celobrojne pozitivne stepene izraza `expr`.

`ExpandAll[expr]` se primenjuje na razvoj svih podizraza (imenilac, brojilac) zadatog izraza `expr`.

Posebno imenilac ili brojilac izraza $expr$ razvijaju se redom naredbama $ExpandDenominator[expr]$ ili $ExpandNumerator[expr]$. Demonstracija rada funkcija kojima se razvija zadati izraz ili neki njegovi delovi biće primenjena na izrazu

$$\frac{(x+3)(x-1)^2}{(x^2+1)(x+5)^2}.$$

```
expr = ((x + 3) (x - 1)^2)/((x^2 + 1) (x + 5)^2);
Expand[expr]
```

Rešenje: $\frac{x^2}{(x+5)^2(x^2+1)} - \frac{5x}{(x+5)^2(x^2+1)} + \frac{3}{(x+5)^2(x^2+1)} + \frac{x^3}{(x+5)^2(x^2+1)}$.

Rezultat primene naredbe $ExpandAll$ na $expr$

```
ExpandAll[expr]
```

je izraz rastavljen na sabirke i u svakom od njih imenilac takođe rastavljen na sabirke

$\frac{x^3}{x^4+10x^3+26x^2+10x+25} + \frac{x^2}{x^4+10x^3+26x^2+10x+25} - \frac{5x}{x^4+10x^3+26x^2+10x+25} + \frac{3}{x^4+10x^3+26x^2+10x+25}$. Rastavljanje samo imenioca vrši se sa

```
ExpandDenominator[expr]
```

ili samo rastavljanje brojioca

```
ExpandNumerator[expr]
```

a njihovim izvršavanjem dobiju se redom izrazi

$$\frac{(x-1)^2(x+3)}{x^4+10x^3+26x^2+10x+25}, \quad \frac{x^3+x^2-5x+3}{(x+5)^2(x^2+1)}.$$

Sa $Factor[expr]$ izraz $expr$ se faktoriše do proizvoda podizraza koji se dalje ne mogu faktorisati. Tako, ilustracije radi primenom ove naredbe na izraz $16cx^2 - 25x^4$

```
Factor[16 x^2 - 25 x^4]
Expand[%]
```

čijim izvršenjem se dobije $-x^2(5x-4)(5x+4)$. Razvijanjem rezultata dobijenog primenom $Factor$ dobija se početni izraz.

Izvlačenje faktora izvan zadatog izraza $expr$, nezavisnih od x vrši se pomoću $FactorTerms[expr, x]$

Naredbom $Together[expr]$ zbir sabiraka svodi se na jedan sa najmanjim zajedničkim imeniocem.

Primenom na izraz

$$\frac{1}{x^2-16} - \frac{x+4}{x^2-3x-4}$$

ilustrovaće se funkcija $Together$

```
expr0 = 1/(x^2 - 16) - (x + 4)/(x^2 - 3 x - 4);
Together[expr0]
```

koja primenom na zadati izraz $expr0$ daje $\frac{-x^2-7x-15}{(x-4)(x+1)(x+4)}$. Ukoliko je preferenti oblik takav da imenilac ne bude rastavljen na faktore, može se primeniti

```
ExpandDenominator[%]
```

što rezultuje izrazom $\frac{-x^2-7x-15}{x^3+x^2-16x-16}$.

Konverzije stepena tipa $(a^b)^c$ u a^{bc} ili $(ab)^c$ u $a^c b^c$ mogu se izvršiti funkcijom $PowerExpand[expr]$, kojom se razvijaju proizvodi ili logaritmi stepena. Primera radi, primenom na izraz $\sqrt{x^2 y^5}$

```
PowerExpand[Sqrt[(x^2) (y^5)]]
```

dobije se $xy^{5/2}$.

Funkcijom *Simplify*[*expr*] vraća se najjednostavniji oblik izraza *expr* primenom različitih standardnih algebarskih transformacija. Izvršenjem naredbe *FullSimplify*[*expr*] vraća se najprostiji oblik izraza *expr* primenom širokog spektra različitih algebarskih ili trigonometrijskih transformacija. *FullSimplify* poznaje širok spektar transformacije, stoga može pojednostaviti izraze koje nije moguće sa *Simplify*. Primena ovih funkcija ilustrovana je kroz dva naredna primera, uprošćavanjem izraza

$$\sec^2(x)(\sec(2x) + 1)$$

```
Simplify[Sec[x]^2 (1 + Sec[2 x])]
```

čiji je rezultat: $2 \sec(2x)$ i uprošćavanjem

$$\sec x \tan x(\sec x + \tan x) + (\sec x - \tan x)$$

```
FullSimplify[Sec[x] Tan[x] (Sec[x] + Tan[x]) + (Sec[x] - Tan[x])]
```

nakon čega se dobije $\tan^3(x) + \sec^3(x)$.

Za pomenute algebarske transformacije, izuzev *ComplexExpand*, *Simplify* i *FullSimplify* trigonometrijske funkcije su nevidljivi objekti i ostavljaju ih nepromenjenima što ilustruju sledeći primeri.

```
ComplexExpand[Sin[x + I y]]
Expand[Sin[x + I y]]
```

odakle se dobiju redom $\sin(x) \cosh(y) + i \cos(x) \sinh(y)$ i $\sin(x + iy)$.

```
Simplify[Sin[x]^2 + Cos[x]^2]
```

Rešenje je 1.

Pored toga, neretko kada funkcija *Simplify* vrati nepromenjen izraz, istom *FullSimplify* napravi znatno uprošćenje. Primenom naredbi za uprošćavanje na trigonometrijski izraz $\tan^2(x) (\tan^4(x) + 3 \tan^2(x) + 3)$

```
Simplify[Tan[x]^2(3+3Tan[x]^2+Tan[x]^4)]
FullSimplify[Tan[x]^2 (3+3Tan[x]^2+Tan[x]^4)]
```

dobije se redom $\tan^2(x) (\tan^4(x) + 3 \tan^2(x) + 3)$ i $\sec^6(x) - 1$. Povremeno primenom na isti algebarski ili trigonometrijski izraz, izvršenje *FullSimplify* može da traje duže nego upotrebom *Simplify* zbog toga što *FullSimplify* varira veći izbor transformacija.

Sa *TrigExpand*[*expr*] trigonometrijski izraz *expr* razvija se u sumu članova. Primenom ove funkcije na $\cos(x + y + z)$

```
TrigExpand[Cos[x + y + z]]
```

izlaz je suma proizvoda trigonometrijskih funkcija $\cos(x) \cos(y) \cos(z) - \sin(x) \sin(y) \cos(z) - \sin(x) \cos(y) \sin(z) - \cos(x) \sin(y) \sin(z)$.

TrigReduce[*expr*] primenjuje se za trigonometrijske identitete višestrukih uglova. Ovom funkcijom proizvod i stepeni trigonometrijskih funkcija prevode se u sumu trigonometrijskih funkcija kombinovanog argumenta. Primer redukovanja trigonometrijskog izraza

$$\sin^2(x) \cos(y) \sin(z)$$

```
TrigReduce[Sin[2x]Cos[y]Sin[z]]
```

odakle se dobije $\frac{1}{4}(\cos(2x - y - z) + \cos(2x + y - z) - \cos(2x - y + z) - \cos(2x + y + z))$.

TrigFactors[expr] služi za faktorisanje trigonometrijskog izraza *expr* u proizvod članova. Ilustracija primene ove funkcije na gore dobijeni izraz, generisan funkcijom redukovanja

```
TrigFactor[%]
```

Rešenje je početni izraz.

TrigToExp[expr] služi za pisanje trigonometrijskih izraza ili funkcija *expr* preko eksponencijalnih članova.

ExpToTrig[expr] služi za konvertovanje eksponencijalnih u trigonometrijske funkcije. Na primer njenom primenom na e^{iz} verifikuje se Ojlerova formula $e^{iz} = \cos(z) + i \sin(z)$.

```
ExpToTrig[Exp[I z]]
```

Sa *FunctionExpand[expr]* vrši se proširenje određenih trigonometrijskih funkcija. Ova naredba služi za razvijanje trigonometrijskih funkcija čiji argumenti su racionalni multipli od π , celobrojnih ili polu-celobrojnih multiplata inverznih trigonometrijskih funkcija, ili izraza oblika $i \cdot expr$, gde je $i = \sqrt{-1}$. Razvojem izraza uz pomoć funkcije *FunctionExpand*

$$-\frac{1}{48} (1 + \sqrt{5})^3 \operatorname{sech}^2(x) \operatorname{csch}^3(y) \tanh(x + iy) \tan^3\left(\frac{1}{2} \cos^{-1}(z)\right)$$

```
FunctionExpand[Tan[I (x+I*y)] ((Sec[I*x]/Sin[5 Pi/3])^2 / (Sin[I*y]/(Cos [Pi/5] Tan[ArcCos [z]/2)))^3]
```

dobije se $-\frac{(1+\sqrt{5})^3(1-z)^{3/2}\operatorname{sech}^2(x)\operatorname{csch}^3(y)\tanh(x+iy)}{48(z+1)^{3/2}}$.

Trigonometrijski identiteti se mogu proveriti prostim metodom njihovim uprošćavanjem *Simplify[lhs == rhs]*, na primer:

i) $(1 - \tan(x))^2 + (1 - \cot(x))^2 = (\sec(x) - \csc(x))^2$

```
Simplify[(1-Cot[x])^2+(1-Tan[x])^2==(Sec[x]-Csc[x])^2]
```

True,

ii) $2 \cot(4x) + 2 \csc(4x) = \cot(x) - \tan(x)$

```
Simplify[2Csc[4x]+2Cot[4x]==Cot[x]-Tan[x]]
```

True,

iii) $\tan\left(\frac{\alpha}{2}\right) \cot\left(\frac{\beta}{2}\right) - \cot\left(\frac{\alpha}{2}\right) \tan\left(\frac{\beta}{2}\right) = 2 \csc(\alpha) \csc(\beta) (\cos(\beta) - \cos(\alpha))$

```
Simplify[Tan[\[Alpha]/2]Cot[\[Beta]/2]-Cot[\[Alpha]/2]Tan[\[Beta]/2]==2(Cos[\[Beta]]-Cos[\[Alpha]])/(Sin[\[Alpha]] Sin[\[Beta]])]
```

True.

Osim trigonometrijskih funkcija, neke od transformacija kao *FunctionExpand* kad je to moguće proširuje i specijalne funkcije, svodeći složene argumente na proste kao na primer

```
FunctionExpand[Zeta[1 - z]]
```

pri čemu se nakon izvršenja dobije $2^{1-z} \pi^{-z} \zeta(z) \sin\left(\frac{1}{2} \pi(1-z)\right) \Gamma(z)$, gde je $\zeta(z)$ Rimanova zeta funkcija, a $\Gamma(z)$ je Ojlerova gama funkcija. Pored razvoja na prostije funkcije, sa *FunctionExpand* mogu se ispitati odnosi između izraza koje se sastoje od specijalnih funkcija:

```
FunctionExpand[Beta[a, b]==Gamma[a]Gamma[b]/Gamma[a+b]]
```

gde je $Beta[a, b]$ Ojlerova beta funkcija $B(a, b)$. Ukoliko nije moguće ispitati relaciju sa *Simplify*, odnos između izraza se može ispitati sa *FullSimplify*. Testiranjem izraza koji sadrži specijalne funkcije

```
Simplify[Beta[a, b + 1] == (b/(a + b)) Beta[a, b]]
```

izlaz nije odgovor na test već $B(a, b + 1) = \frac{bB(a, b)}{a+b}$, dok je rezultat dobijen izvršenjem linije

```
FullSimplify[Beta[a, b + 1] == (b/(a + b)) Beta[a, b]]
```

True.

Tokom manipulacije izrazima program Mathematica pravi što je moguće manje pretpostavki o promenljivim i simbolima bez dodeljenih vrednosti kako bi rezultati bio što opštiji. U slučaju da je neophodno programski zadati konkretnije informacije o određenim promenljivim i simbolima tokom transformacije izraza, moguće je dati informacije uključivanjem pretpostavki u vidu dodatnog argumenta za nekoliko navedenih funkcija transformacije. Pretpostavke mogu biti postavljene u obliku jednakosti, nejednakosti, specifičnih domena ili njihovih logičkih kombinacija. Veličine koje se pojavljuju u nejednakostima smatraju se realnim. Specificiranje domena se može realizovati naredbom

Element[elem, dom]

čime se tvrdi da je promenljiva $elem = x$ ili skup $elem = \{x_1, x_2, \dots\}$ iz domena $dom = Complexes, Reals, Rationals, Integers$.

Sa *Refine[expr, assum]* vrši se prečišćavanje izraza *expr* koristeći pretpostavku *assum*. Izlaz funkcije *Refine* je ono što bi se dobilo kada bi se u izrazu *expr* simboli ili izrazi zamenili numeričkim vrednostima koji zadovoljavaju pretpostavke *assum*.

```
Refine[Sqrt[-x], x > 0]
```

Rešenje: $i\sqrt{x}$.

Funkcije kojima se uprošćava *Simplify[expr, assum]* i *FullSimplify[expr, assum]* ili razvija *FunctionExpand[expr, assum]* izraz *expr* su transformacione funkcije koje se izvršavaju pod pretpostavkom *assum*, zadatom na mestu drugog argumenta. Bezuslovnom primenom *Simplify* na izraz $\log x^r$ neće doći do njegove promene

```
Simplify[Log[x^r]]
```

dok uz pretpostavku da je x pozitivan a r realan broj

```
Simplify[Log[x^r], (x > 0) && Element[r, Reals]]
```

izvršeno je uprošćavanje logaritamskog izraza $r \log x$.

Primenom *Simplify* nepromenjen ostaje izraz $\cos(\frac{n\pi}{2} - x)$

```
Simplify[Cos[n Pi/2-x]]
```

dok uz uzlov da je $(n - 1)/4$ celobrojno, izraz se pojednostavljuje do $\sin(x)$.

```
Simplify[Cos[n Pi/2-x], Element[(n-1)/4, Integers]]
```

3.1.9 Izvršiti delimičnu dekompoziciju sledećih izraza.

i) $\frac{x^2+2x-1}{2x^3+3x^2-2x}$

$$(x^2 + 2x - 1)/(2x^3 + 3x^2 - 2x)$$

Rešenje: $-\frac{1}{10(x+2)} + \frac{1}{5(2x-1)} + \frac{1}{2x}$.

ii) $\frac{4x^3-27x^2+5x-32}{30x^5-13x^4+50x^3-286x^2-299x-70}$

$$\text{Apart}[(4x^3-27x^2+5x-32)/(30x^5-13x^4+50x^3-286x^2-299x-70)]$$

Rešenje: $\frac{22098x+48935}{260015(x^2+x+5)} - \frac{668}{323(2x+1)} - \frac{9438}{80155(3x-7)} + \frac{24110}{4879(5x+2)}$.

3.1.10 Faktorirati sledeće izraze.

i) $x^4 + 2x^3 - 3x - 6$

$$\text{Factor}[x^4 + 2x^3 - 3x - 6]$$

Rešenje: $(2+x)(-3+x^3)$.

ii) $6t^3 + 9t^2 - 15t$

$$\text{Factor}[6t^3 + 9t^2 - 15t]$$

Rešenje: $3(-1+t)t(5+2t)$.

iii) $ax^2 + ay + bx^2 + by$

$$\text{Factor}[a*x^2 + a*y + b*x^2 + b*y]$$

Rešenje: $(a+b)(x^2+y)$.

iv) $2x^4y^6 + 6x^2y^3$

$$\text{Factor}[2x^4*y^6 + 6x^2*y^3]$$

Rešenje: $2x^2y^3(3+x^2y^3)$.

3.1.11 Grupisanjem uprostiti izraze.

i) $\frac{1}{x^2-16} - \frac{x+4}{x^2-3x-4}$

$$\text{Together}[1/(x^2 - 16) - (x + 4)/(x^2 - 3x - 4)]$$

Rešenje: $\frac{-x^2-7x-15}{(x-4)(x+1)(x+4)}$.

ii) $\frac{\frac{x^2}{\sqrt{4-x^2}} + \sqrt{4-x^2}}{4-x^2}$

$$\text{Simplify}[\text{Together}[(\text{Sqrt}[4 - x^2] + x^2/\text{Sqrt}[4 - x^2])/(4 - x^2)]]$$

Rešenje: $\frac{4}{(4-x^2)^{3/2}}$.

iii) $\frac{\frac{1}{x^2} - \frac{1}{y^2}}{\frac{1}{x} + \frac{1}{y}}$

$$\text{Together}[(1/x^2 - 1/y^2)/(1/x + 1/y)]$$

Rešenje: $\frac{y-x}{xy}$.

$$\text{iv) } \frac{3}{x^3-x} + \frac{4}{x^2+4x+4}$$

```
ExpandDenominator[Together[3/(x^3 - x) + 4/(x^2 + 4 x + 4)]]
```

$$\text{Rešenje: } \frac{4x^3+3x^2+8x+12}{x^5+4x^4+3x^3-4x^2-4x}$$

$$\text{v) } \frac{\sqrt[3]{x+5}\sqrt{\frac{x+2}{x+7}}\sqrt[3]{\frac{3-x^2}{25-x^2}}}{\sqrt[3]{3-x^2}\sqrt{4-x^2}}$$

```
FullSimplify[(Sqrt[(2+x)/(7+x)] ((3-x^2)/(25-x^2))^(1/3) (5+x)^(1/3)) / (Sqrt[4-x^2] (3-x^2)^(1/3))]
```

$$\text{Rešenje: } \frac{1}{\sqrt[3]{5-x}\sqrt{-(x-2)(x+7)}}$$

3.1.12 Koristeći razvoj po stepenima uprostiti sledeće izraze.

$$\text{i) } 3\sqrt[3]{2}\sqrt[3]{x^3y^6z}$$

```
PowerExpand[(54*x^3*y^6*z)^(1/3)]
```

$$\text{Rešenje: } 3\sqrt[3]{2}xy^2\sqrt[3]{z}$$

$$\text{ii) } \frac{\sqrt{2}\sqrt{xy^2}}{\sqrt{x^5}}$$

```
PowerExpand[Sqrt[6*x*y^2]/Sqrt[3*x^5]]
```

$$\text{Rešenje: } \frac{\sqrt{2}y}{x^2}$$

$$\text{iii) } \log\left(\sqrt[3]{e}\sqrt[3]{x^2}\right) + \log\left(\frac{1}{\sqrt{x}}\right) \text{ pretpostavljajući da je } x > 0$$

```
Assuming[x>0, PowerExpand[Log[Sqrt[x]/x]+Log[(E*x^2)^(1/4)]]]
```

$$\text{Rešenje: } \frac{1}{4}$$

3.1.13 Pokazati da važe sledeće jednakosti.

$$\text{i) } \frac{1-\cos(x)}{1-\csc(x)} - \frac{\cos(x)+1}{\csc(x)+1} = 2 \tan(x)(\sin(x) - \sec(x))$$

```
Simplify[(1-Cos[x])/(1-Csc[x])-(1+Cos[x])/(1+Csc[x])]==2Tan[x] (Sin[x]-Sec[x])]
```

$$\text{Rešenje: } \text{True.}$$

$$\text{ii) } \frac{\sec(x)-\csc(x)}{\tan(x)+\cot(x)} = \frac{\tan(x)-\cot(x)}{\csc(x)+\sec(x)}$$

```
Simplify[(Sec[x]-Csc[x])/(Tan[x]+Cot[x])==(Tan[x]-Cot[x])/(Sec[x]+Csc[x])]
```

$$\text{Rešenje: } \text{True.}$$

$$\text{iii) } (2 \sin(2x) - \sin(3x)) \sec(x) = \sin(x) \left(8 \sin^2\left(\frac{x}{2}\right) + \sec(x)\right)$$

```
Simplify[(2 Sin[2x]-Sin[3x])/Cos[x]==Sin[x] (8Sin[x/2]^2+Sec[x])]
```

$$\text{Rešenje: } \text{True.}$$

3.2 Simboličko rešavanje algebarskih jednačina

Egzaktno rešenje jednačine $lhs == rhs$ u programu Mathematica traži se sa

$$\text{Solve}[lhs == rhs, var]$$

čime se zadaje njeno rešavanje po varijabli var . Dobijeni rezultat je u vidu ugnježdene liste pravila transformacija.

```
eq=x^2-3x-10==0;
sol=Solve[eq,x]
eq /. sol
```

Primeru radi rešenje jednačine

$$x^2 - 3x - 10 = 0$$

je set pravila $\{\{x \rightarrow -2\}, \{x \rightarrow 5\}\}$. Proveru rešenja moguće je izvršiti njihovom zamenom pomoću operatora $/$. čime se dobija izveštaj da je jednakost tačna ukoliko se x zameni sa vrednostima -2 ili 5. Premda je naredba *Solve* namenjena prvenstveno za rešavanje polinomijalnih jednačina njome je moguće rešiti i neke druge. Primeru radi, odrediti brzinu v relativističke čestice mase m i impulsa p za koju važi da je

$$p = \frac{mv}{\sqrt{1 - v^2/c^2}}$$

a c je brzina svetlosti.

```
Solve[p==m*v/Sqrt[1-v^2/c^2], v]
```

Rešenja ove jednačine su $\left\{ \left\{ v \rightarrow -\frac{cp}{\sqrt{c^2m^2+p^2}} \right\}, \left\{ v \rightarrow \frac{cp}{\sqrt{c^2m^2+p^2}} \right\} \right\}$, gde je prvo, sa znakom minus, fizički neprihvatljivo jer bi u tom slučaju predznak odnosno smer brzine i impulsa bio suprotan. Matematički komplikovanije jednačine kao što su transcendentne¹ ili polinomijalne stepena većeg od 4 nemaju uvek egzaktno rešenje. Rešavanjem transcendentne jednačine

$$x^2 e^{-x} + x - 5 = 0$$

```
Solve[x^2Exp[-x]+x-5==0, x]
```

Program Mathematica ne može da nađe egzaktno rešenje i vraća ulaz, eventualno rearanžiran $\text{Solve}[e^{-x}x^2 + x - 5 = 0, x]$. Odgovor programa Mathematica prilikom pokušaja traženja egzaktnog rešenja polinomijalne jednačine stepena većeg od 4

$$x^5 - 3x^4 + 2x^3 + x = 4$$

```
Solve[x^2Exp[-x]+x-5== 0, x]
```

je da ne može naći eksplicitna rešenja

$$\left\{ \left\{ x \rightarrow \text{Root} \left[\#1^5 - 3\#1^4 + 2\#1^3 + \#1 - 4\&, 1 \right] \right\}, \right. \\ \left\{ x \rightarrow \text{Root} \left[\#1^5 - 3\#1^4 + 2\#1^3 + \#1 - 4\&, 2 \right] \right\}, \\ \left\{ x \rightarrow \text{Root} \left[\#1^5 - 3\#1^4 + 2\#1^3 + \#1 - 4\&, 3 \right] \right\}, \\ \left\{ x \rightarrow \text{Root} \left[\#1^5 - 3\#1^4 + 2\#1^3 + \#1 - 4\&, 4 \right] \right\}, \\ \left. \left\{ x \rightarrow \text{Root} \left[\#1^5 - 3\#1^4 + 2\#1^3 + \#1 - 4\&, 5 \right] \right\} \right\}.$$

¹Jednačina oblika $F(x) = f(x)$ u kojoj barem jedna funkcija nije algebarska.

Rešenja zadate jednačine se mogu naći aproksimativno njenim numeričkim rešavanjem. Istom naredbom *Solve* program Mathematica proverava postoji li skup var_1, var_2, \dots koji zadovoljava sistem jednačina $lhs_1 == rhs_1, lhs_2 == rhs_2, \dots$. Primera radi, rešenje sistema algebarskih jednačina

$$\begin{aligned}x + y &= z \\10 - 6x - 2z &= 0 \\6x - 24 - 4y &= 0\end{aligned}$$

```
Solve[{x+y==z,10-6x-2z==0,6x-24-4 y==0},{x,y,z}]
```

je skup rešenja $\{\{x \rightarrow 2, y \rightarrow -3, z \rightarrow -1\}\}$.

3.2.14 Rešiti sistem jednačina

$$\begin{aligned}4x + 5y &= 5 \\6x + 7y &= 7\end{aligned}$$

```
Solve[{4 x+5y==5,6x+7y==7}, {x, y}]
```

Rešenje: $\{\{x \rightarrow 0, y \rightarrow 1\}\}$.

3.2.15 Rešiti jednačinu $z^2 + 4z - 8 = 0$

```
Solve[z^2+4z-8==0, z]
```

Rešenje: $\left\{ \left\{ z \rightarrow 2 \left(-1 - \sqrt{3} \right) \right\}, \left\{ z \rightarrow 2 \left(\sqrt{3} - 1 \right) \right\} \right\}$.

3.2.16 Rešiti jednačinu $\sqrt{2x-3} - \sqrt{x+7} = 2$

```
Solve[Sqrt[2x-3]-Sqrt[x+7]==2, x]
```

Rešenje: $\{\{x \rightarrow 42\}\}$.

3.2.17 i) Rešiti sistem jednačina

$$\begin{aligned}y + z &= x \\by + ax &= f \\-g + cz - by &= 0.\end{aligned}$$

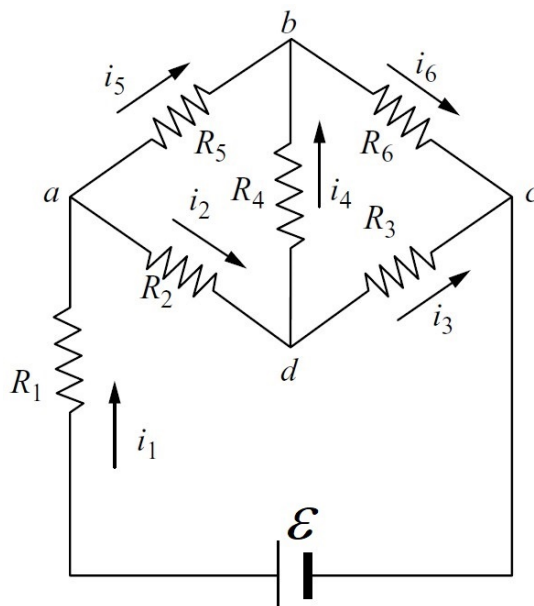
```
eqs={y+z==x,b*y+a*x==f,-g+c*z-b*y== 0};
sol = Solve[eqs, {x, y, z}]
```

Rešenje: $\left\{ \left\{ x \rightarrow -\frac{bf-bg-cf}{ab+ac+bc}, y \rightarrow -\frac{ag-cf}{ab+ac+bc}, z \rightarrow -\frac{-ag-bf-bg}{ab+ac+bc} \right\} \right\}$.

ii) Dobijena rešenja u obliku pravila zamene vratiti u jednačine i verifikovati rezultat.

```
pom=eqs/.sol[[1]]
Simplify[pom]
```

Izvršena je zamena vrednosti *sol* rešavanja zadanog sistema jednačina, nakon čega je dobijena lista jednakosti. Uproščavanjem izraza u listi sa *Simplify* dobijena je lista $\{\text{True}, \text{True}, \text{True}\}$.



Slika 3.1: Složeno električno kolo.

3.2.18 **Strujno kolo.** Za složeno električno kolo, sastavljeno od više kontura prikazano na slici ispod, naći jačine struje kroz sve otpornike. Kolo se sastoji od elektromotorne sile zanemarljivog unutrašnjeg otpora i otpornika u granama koje se stiču u čvorove: a , b i d , čineći petlje: $abda$, $dbcd$ i $adca$.

Na slici 3.1 je označen pretpostavljen tok struja. Ukoliko je on suprotan, dobiće se ista jačina samo suprotnog predznaka. Zadatak se rešava primenom Kirhofovih pravila i Omovog zakona. Jednačine dobijene na osnovu 1. Kirhofovog pravila, primenjenog na čvorove su

$$a - \text{čvor: } i_1 - i_2 - i_5 = 0$$

$$b - \text{čvor: } i_5 + i_4 - i_6 = 0$$

$$d - \text{čvor: } i_2 - i_4 - i_3 = 0.$$

Primenom 2. Kirhofovom pravila na zatvorene kružne konture dobiju se sledeće algebarske jednačine

$$abda \text{ kontura: } -i_5 R_5 + i_4 R_4 + i_2 R_2 = 0$$

$$dbcd \text{ kontura: } -i_4 R_4 - i_6 R_6 + i_3 R_3 = 0$$

$$adca \text{ kontura: } -i_1 R_1 - i_2 R_2 - i_3 R_3 + \varepsilon = 0.$$

```
(*linearne algebarske jednačine*)
eqs={i1-i2-i5==0,i5+i4-i6==0,
i2-i4-i3==0,-i5*R5+i4*R4+i2*R2==0,
-i4*R4-i6*R6+i3*R3==0,
-i1*R1-i2*R2-i3*R3+\[Epsilon]==0};
(*nepoznate*)
vars={i1,i2,i3,i4,i5,i6};
sol=Solve[eqs,vars][[1]];
sol=vars/.sol;
sol=Table[FullSimplify[sol[[i]],{i,Length[sol]}]
```

Provere radi može se izvršiti sledeći test. Ukoliko su otpornici R_2, R_3, R_5, R_6 jednaki $R_2 = R_3 = R_5 = R_6 = R$, na osnovu Omovog zakona očekivana razlika potencijala na grani bd je 0 i struja $i_4 = 0$. Ukoliko je tačno opšte rešenje zadanog sistema linearnih algebarskih jednačina, zamenom otpornika R_i , $i = 2, 3, 5, 6$ istim otpornikom, očekivano je da struje kroz njih budu iste, dok su prema Kirhofovom pravilu primenjenom na čvor a struje u granama jednake $i_1/2$.

```
soltest=sol/.{R2->R,R3->R,R5->R,R6->R};
Table[FullSimplify[soltest[[j]]], {j, Length[soltest]}]
```

Najpre je pravilima $R_2 \rightarrow R, R_3 \rightarrow R, R_5 \rightarrow R, R_6 \rightarrow R$ izvršena zamena vrednosti otpora $R_i = R$, $i = 2, 3, 5, 6$, čime je umesto opšteg rešenja dobijeno novo. Međutim izrazi za struje su i dalje veoma složeni, stoga je primenjeno uprošćavanje na svaku dobijenu struju pojedinačno $FullSimplify[soltest[[j]]]$ u listabilnoj funkciji $Table$

$$\left\{ i_1 = \frac{\epsilon}{R+R_1}, i_2 = \frac{\epsilon}{2(R+R_1)}, i_3 = \frac{\epsilon}{2(R+R_1)}, i_4 = 0, i_5 = \frac{\epsilon}{2(R+R_1)}, i_6 = \frac{\epsilon}{2(R+R_1)} \right\}.$$

3.2.19 Električno polje u okolini naelektrisane kugle.

Grafički će biti ilustrovana zavisnost električnog polja, unutar ($r < R$) i oko ($r > R$) zapreminski homogeno naelektrisane kugle radijusa R . Gustina naelektrisanja kugle ρ je konstantna po svim njenim delovima. Prema Gausovom zakonu

$$\Phi = \oint_{\partial V} \vec{E} d\vec{S} = \frac{Q}{\epsilon_0}, \quad (3.1)$$

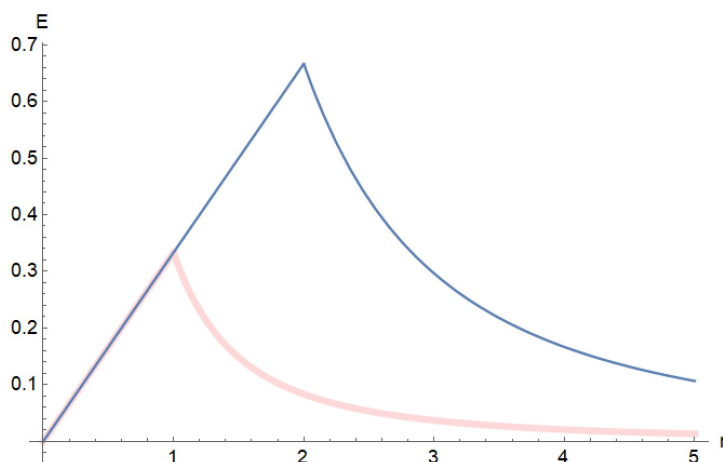
važi da je fluks električnog polja kroz zatvorenu površinu proporcionalan količini naelektrisanja koju ta površina obuhvata. Ukupno naelektrisanje kugle iznosi $Q = \rho \frac{4}{3} R^3 \pi$. Zbog simetrične raspodele naelektrisanja polje će biti usmereno radijalno, ravnomerno u svim pravcima. Odavde jednačina (3.1) postaje

$$4\pi r^2 E = \frac{q(r)}{\epsilon_0}, \quad q(r) = \begin{cases} Q, & r \geq R \\ \rho \frac{4}{3} r^3 \pi, & r < R \end{cases} \quad (3.2)$$

Dobijen je sistem algebarskih jednačina čije rešenje se traži analitički.

```
Clear[Eout,Ein];
(*sistem algebarskih jedna\{c\}ina*)
eq={4Pi*r^2Eout==(4/3R^3 Pi)\[Rho]/\[Epsilon],
4Pi r^2 Ein==\[Rho] (4/3 r^3 Pi)/\[Epsilon]};
(*zadate zamene velicina i konstante*)
value={\[Epsilon]->1,\[Rho]->1};
(*resavanje sistema jednacina po Eout i Ein*)
sol=Solve[eq,{Eout,Ein}][[1]];
(*konacni izrazi za Eout i Ein*)
{Eout,Ein}={Eout,Ein}/.sol;
{Eout,Ein}={Eout,Ein}/.value
```

```
(*zavisnost E od rastojanja od centra*)
(*drugi argument je opcioni*)
EField[rr,RR,1]:=If[rr<RR,Ein,Eout] /. {r->rr,R->RR};
Show[
Plot[EField[r], {r, 0, 5},
PlotStyle -> {Thickness[0.01], LightRed}],
Plot[EField[r, 2], {r, 0, 5}], PlotRange -> All]
```



Slika 3.2: Intenzitet električnog polja u zavisnosti od rastojanja od centra ravnomerno naelektrisane kugle radijusa $R = 1$ (svetlo crvena linija) i $R = 2$ (plava linija).

3.2.20 Borov model atoma vodonika.

Borov model atoma vodonika uzima u obzir zakone klasične fizike i Plankove ideje o kvantovanoj energiji elektrona u atomu vodonika. Ovaj model zasnovan je na tri postulata:

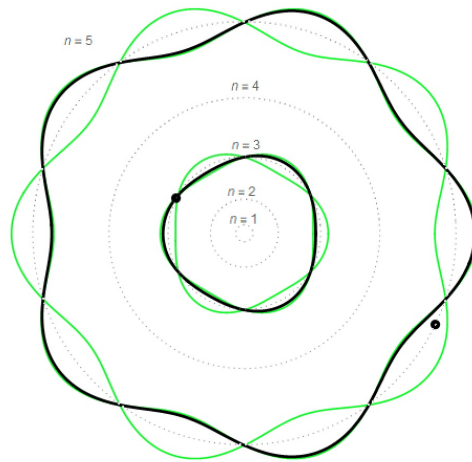
- i) Elektron se u atomu kreće oko jezgra po stabilnim kružnim orbitama bez promene energije. Ovaj postulat govori o diskretnim stacionarnim stanjima atoma.
- ii) Dozvoljena stacionarna stanja su ona čiji je angularni moment kvantovan $l = mvr = n\hbar$.
- iii) Prilikom prelaska elektrona sa početne orbite više/niže energije E_i u finalno stanje niže/više energije E_f praćeno je emisijom/apsorbcijom energije tako da je $h\nu = E_{i/f} - E_{f/i}$.

Smatrajući da je elektron vezan za jezgro privlačnom elektrostatičkom silom, opisanom Kulonovim zakonom kZe^2/r^2 , gde je e elementarno naelektrisanje elektrona, r udaljenost jezgra i elektrona, Z naelektrisanje jezgra ili broj protona u jedinicama e , a k je konstanta $k = 1/4\pi\epsilon_0$. Pretpostavlja se da jezgro miruje, zbog mnogo veće mase od elektrona, te da elektron kruži oko jezgra.

- i) Koristeći Borov metod kvantizacije, izvesti izraz za energiju nivoa u zavisnosti od glavnog kvantnog broja n , gde n označava n -tu Borovu orbitu.
- ii) Izvesti izraz za frekvenciju ω emitovanog zračenja, koje je posledica prelaska između Borovih energijskih nivoa označenih sa n_i i n_f .

Raspisane su jednačine u cilju povezivanja parametara, prva koja opisuje jednakost centrifugalne sile i Kulonove interakcije, a druga zadata na osnovu kvantizacionog pravila Borovih postulata. Njihovim rešavanjem nalaze se izrazi kvantovanih brzina v_n i radijusa r_n . Ukupna energija svakog nivoa E_n je suma potencijalne i kinetičke energije, koja se dalje nalazi iz dobijenih diskretnih parametara. U klasičnoj slici, primenom drugog Njutnovog zakona $F = ma$ gde je F privlačna Kulonova sila a $a = v^2/r$ normalno ubrzanje, dobije se veza između radijusa i brzine rotacije elektrona oko jezgra.

```
coulomb = k Z*e^2/r^2;
force = m*a;
a = v^2/r;
eq1 = coulomb == force
```

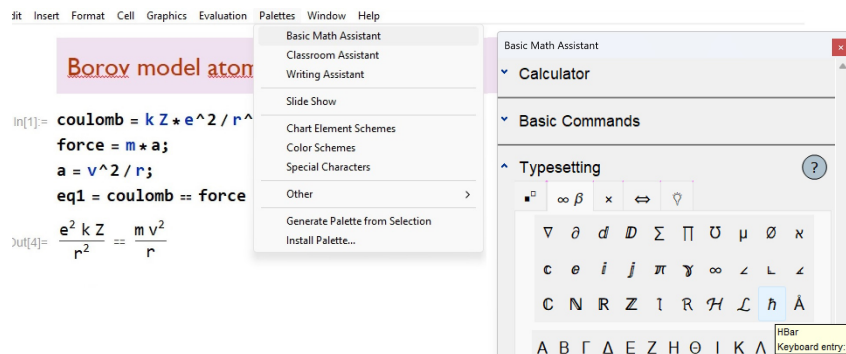


Slika 3.3: Borov model atoma.

Jednačina označena promenljivom *eq1* je oblika

$$\frac{e^2 k Z}{r^2} = \frac{m v^2}{r}$$

i daje klasičnu vezu između radijusa orbite i brzine kruženja elektrona po njoj.

Slika 3.4: Umetanje slova \hbar iz palete specijalnih simbola.

Ugaoni moment prema zakonima klasične mehanike definisan je sa $\vec{L} = m\vec{v} \times \vec{r}$. Kvantizacijom, intenzitetu angularnog momenta elektrona u atomu H izraženom u jedinicama redukovane Plankove konstante $\hbar = h/(2\pi)$, pridružuje se izraz koji zavisi od kvantnog broja $L = n\hbar$, $n = 1, 2, 3, \dots$

```
(*angularni moment cestice koja se krece po krugu radijusa r \
tangencijalnom brzinom v*)
anglmomnt = m*v*r;
(*kvantnovanje angularnog momenta*)
eq2 = anglmomnt == n*\[HBar];
sol = Solve[{eq1, eq2}, {v, r}][[1]]
```

Jednačina označena sa *eq2* je druga veza radijusa i brzine nakon kvantizacije. Rešavanjem sistema jednačina $\{eq1, eq2\}$ dobiju se diskretne vrednosti radijusa orbita kao i brzina kojima elektron kruži po orbitama $sol = Solve[\{eq1, eq2\}, \{v, r\}][[1]]$. Rešenje ove jednačine je skup pravila

koja nose informaciju o zavisnostima r_n i v_n

$$\left\{ v_n \rightarrow \frac{e^2 k Z}{n \hbar}, r_n \rightarrow \frac{n^2 \hbar^2}{e^2 k m Z} \right\}.$$

U nastavku su definisane kinetička T , potencijalna V i ukupna energija E_{tot} . Nakon zamene diskretizovanih zavisnosti v_n i r_n prema definiciji kao zbir kinetičke i potencijalne traži se ukupna energija elektrona u zavisnosti od kvantnog broja n

```
(*kinetička energija*)
T = m*v^2/2;
(*potencijalna energija*)
V = -k*Z*e^2/r;
(*ukupna energija elektrona*)
Etot = T + V
(*ukupna energija elektrona u H atomu u zavisnosti od n *)
Enrgy[n_] = Etot /. sol
```

odakle se dobije izraz za ukupnu energiju

$$E_{tot}(n) = -\frac{e^4 k^2 m Z^2}{2 n^2 \hbar^2}.$$

Energija emitovanog fotona zavisi od kvantnih brojeva početnog n_i i krajnjeg energijskog nivoa n_f uz uslov da je $n_i > n_f$

$$h\nu = \hbar\omega = E_{tot}(n_i) - E_{tot}(n_f),$$

```
(*prelaz iz nivoa sa ni na nf*)
eq3 = \[HBar] \[Omega] == Enrgy[ni] - Enrgy[nf];
omega = Solve[eq3, \[Omega]][[1]];
omega = \[Omega] /. omega;
(*ugaona frekvencija fotona *)
omega = Simplify[omega]
```

pri čemu se nakon sređivanja izraza dobije rezultat

$$\omega = \frac{e^4 k^2 m Z^2 (n_i^2 - n_f^2)}{2 n_i^2 n_f^2 \hbar^3}.$$

3.3 Razvoj u stepene redove i limesi

Razvoj funkcije $f(x)$ u stepeni red oko tačke x_0 po $(x - x_0)$ do na stepen $(x - x_0)^n$ realizuje se naredbom

$$\text{Series}[f, \{x, x_0, n\}]$$

gde je prvi argument funkcija $f(x)$, a na drugom lista koja sadrži oznaku promenljive x , tačku razvoja x_0 i stepen n do kog se razvoj vrši. Primer razvoja funkcije

$$f(x) = e^{\sin x}$$

oko $x = 0$, do na stepen x^7 .

```
pwsrs=Series[Exp[Sin[x]],{x,0,7}]
```

Rezultat Izvršenja gornje linije koda je stepeni red

$$1 + x + \frac{x^2}{2} - \frac{x^4}{8} - \frac{x^5}{15} - \frac{x^6}{240} + \frac{x^7}{90} + O(x^8)$$

gde je $O(x^n)$ član stepena x^n .

Rešenje funkcije `Series[f, {x, x0, n}]` se naredbom

`Normal[series]`

konvertuje u običan izraz, bez poslednjeg člana koji sadrži granično ponašanje ili stepen aproksimacije funkcije $O(x - x_0)^n$. Ilustracije radi, primenom funkcije `Normal` na prethodno dobijen red `pwsrs`

```
Normal[pwsrs]
```

dobija se izraz $\frac{x^7}{90} - \frac{x^6}{240} - \frac{x^5}{15} - \frac{x^4}{8} + \frac{x^2}{2} + x + 1$.

Naredbom

`Limit[f(x), x → x0]`

traži se limes funkcije $f(x)$ kada x teži x_0

$$\lim_{x \rightarrow x_0} f(x).$$

Primenom naredbe `Limit` naći limes funkcije

$$\lim_{x \rightarrow \infty} \frac{\sqrt{9x^2 + 2}}{3 - 4x}$$

```
Limit[Sqrt[9 x^2+2]/(3-4x), x->Infinity]
```

Rešenje: $-\frac{3}{4}$. Ukoliko funkcija ima diskontinuitet u nekoj tački, prilikom traženja njenog limesa u njoj neophodno je istaći sa koje strane se prilazi prekidu. Nalaženje jednostranog limesa funkcije $f(x)$ kada se argument x približava x_0 sleva $x \rightarrow x_0 -$ ili sdesna $x \rightarrow x_0 +$ što se označava sa

$$\lim_{x \rightarrow x_0 \pm} f(x) = f(x \pm)$$

u programu Mathematica traži se naredbom

`Limit[f(x), x → x0, Direction - > ±1],`

gde naredba `Limit` sadrži još jedan argument kojim se definiše sa koje strane argument funkcije x prilazi x_0 , da li sdesna (od veće vrednosti) `Direction → -1` ili sleva (od manje vrednosti) `Direction → 1`. Ova opcija biće uvežbana na primeru računanja limesa funkcije

$$f(x) = \frac{2x^2}{9 - x^2}$$

kada argument funkcije x prilazi $x_0 = 3$ sleva $x \rightarrow 3 -$ i sdesna $x \rightarrow 3 +$.

```
Limit[2 x^2/(9 - x^2), x -> 3, Direction -> 1]
Limit[2 x^2/(9 - x^2), x -> 3, Direction -> -1]
```

Rešenje: ∞ i $-\infty$.

3.3.21 Naći aproksimaciju funkcije $\sec(x)$ u vidu stepenog reda oko $x = \pi/3$ tačnosti do na $(x - \pi/3)^3$.

```
Series[Sec[x], {x, Pi/3, 3}]
```

Rešenje: $2 + 2\sqrt{3}\left(x - \frac{\pi}{3}\right) + 7\left(x - \frac{\pi}{3}\right)^2 + \frac{23\left(x - \frac{\pi}{3}\right)^3}{\sqrt{3}} + O\left(\left(x - \frac{\pi}{3}\right)^4\right)$.

3.3.22 Aproksimirati funkciju

$$\ln\left(\frac{1+x}{1-x}\right)$$

do na član x^{11} .

```
Series[Log[(1+x)/(1-x)], {x, 0, 11}]
```

Rešenje: $2x + \frac{2x^3}{3} + \frac{2x^5}{5} + \frac{2x^7}{7} + \frac{2x^9}{9} + \frac{2x^{11}}{11} + O(x^{12})$.

3.3.23 Izračunati limes funkcije

$$\frac{\log(x-a)}{(a-b)(a-c)} + \frac{\log(2(x-b))}{(b-a)(b-c)} + \frac{\log(x-c)}{(c-a)(c-b)}$$

kad x teži beskonačnosti.

```
Limit[Log[10, (x-a)]/((a-b)(a-c))+Log[10, 2(x-b)]/((b-a)(b-a))+Log[10, (x-c)]/((c-a)(c-b)), x->Infinity]
```

Rešenje: $-\frac{\log(2)}{\log(10)(a-b)(b-c)}$.

3.3.24 Izračunati limese.

i) $\lim_{x \rightarrow \infty} \frac{1}{e^x - e^{(x-x^2)}}$

```
Limit[1/(Exp[x]-Exp[x-x^(-2)]), x->Infinity]
```

Rešenje: 0.

ii) $\lim_{x \rightarrow \pi^-} \frac{\sin x}{1 - \cos x}$

```
Limit[Sin[x]/(1-Cos[x]), x->Pi, Direction->1]
```

Rešenje: 0.

iii) $\lim_{x \rightarrow 0^+} (1 + \sin(4x))^{\cot x}$

```
Limit[(1 + Sin[4 x])^(Cot[x]), x -> 0, Direction -> -1]
```

Rešenje: e^4 .

iv) $\lim_{x \rightarrow 1} \frac{\ln x}{x-1}$

```
Limit[Log[x]/(x - 1), x->1]
```

Rešenje: 1.

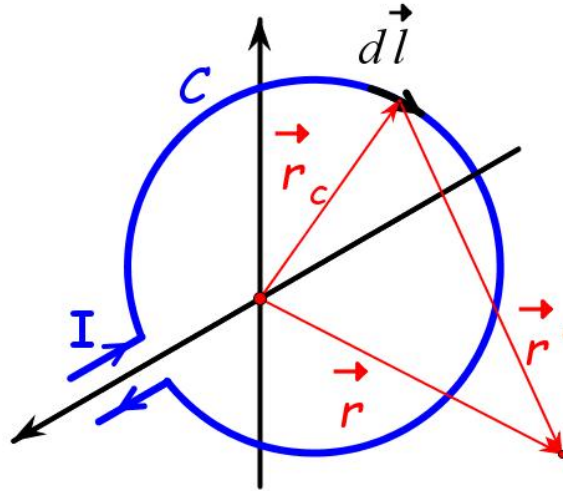
v) $\lim_{x \rightarrow 0} \frac{\sqrt{x^2+9}-3}{x^2}$

```
Limit[(Sqrt[x^2+9]-3)/x^2, x->0]
```

Rešenje: $\frac{1}{6}$.

3.3.25 Magnetno polje Helmholtcovih kalemova.

Primenom Bio-Savarovog zakona naći magnetno polje u razvoju do četvrtog stepena na sredini zajedničke ose dva identična paralelna kalemova kroz koja teče struja istog smera i jačine I . Kalemovi radijusa R_0 postavljeni su na međusobnom rastojanju d .



Slika 3.5: Strujna kontura C .

Prema Bio-Savarovom zakonu delić provodnika dužine dl kroz koji teče struja jačine I , usmerena duž vektora $d\vec{l}$, u tački čiji je radijus vektor \vec{r}' mereći od posmatranog dela provodnika prikazano na slici 3.5, generiše magnetno polje $d\vec{B}$ dato prema sledećem izrazu

$$d\vec{B} = \frac{\mu_0}{4\pi} I \frac{d\vec{l} \times \vec{r}'}{r'^2} \quad (3.3)$$

gde je $\mu_0 = 4\pi \cdot 10^{-7} \text{ Tm/A}$ magnetna permeabilnost vakuuma.

Ukupno polje u posmatranoj tački prostora radijus vektora \vec{r} kada kroz konturu C teče struja I računa se prema

$$\vec{B} = \frac{\mu_0}{4\pi} I \int_C \frac{d\vec{l} \times \vec{r}'}{r'^2}.$$

Radi što jednostavnije postavke zadatka koordinatni sistem će biti postavljen na sredini između kalemova. Kroz sredinu paralelnih kalemova, normalno na njihovu površinu, prolazi z -osa. Koordinate tačaka duž kalemova se zadate su na sledeći način

$$\vec{r}_c = (R_0 \cos \varphi, R_0 \sin \varphi, \pm d/2).$$

Neka je \vec{r} radijus vektor tačke u kojoj se traži magnetno polje $\vec{r} = (x, y, z)$, vektor relativnog položaja te tačke od malog dela konture kalemova određena je parametrom φ

$$\vec{r}' = \vec{r} - \vec{r}_c = (x, y, z) - (R_0 \cos \varphi, R_0 \sin \varphi, \pm d/2).$$

Vektor elementarnog dela konture nalazi se iz izraza

$$d\vec{l} = (R_0 \cos(\varphi + d\varphi), R_0 \sin(\varphi + d\varphi), \pm d/2) - (R_0 \cos \varphi, R_0 \sin \varphi, \pm d/2)$$

$$d\vec{l} = (-R_0 \sin \varphi, R_0 \cos \varphi, 0)d\varphi$$

što će biti dobijeno programskim kodom

```
(*R0- radijus kalema,
Ic- stacionarna struja kroz kalem *)
(*kalem 1*)
(*koordinate tacaka na kalemu 1*)
rc1 = {R0*Cos[\[CurlyPhi]], R0*Sin[\[CurlyPhi]], d/2};
(*vektor infinitezimalnog delica konture /d\[CurlyPhi]*)
dl1 = D[rc1, \[CurlyPhi]];
(*radijus vektor tacka u kojoj se racuna polje-delic konture*)
rp1 = {x, y, z} - rc1;
(*Biot-Savart-ov zakon*)
dB1 = FullSimplify[(\[Mu]/(4 Pi)) Ic*Cross[dl1, rp1]/(rp1.rp1)^(3/2)];
B1 = Integrate[dB1 /. {x -> 0, y -> 0}, {\[CurlyPhi], 0, 2 Pi}];

(*kalem 2*)
rc2 = {R0*Cos[\[CurlyPhi]], R0*Sin[\[CurlyPhi]], -d/2};
dl2 = D[rc2, \[CurlyPhi]];
rp2 = {x, y, z} - rc2;
dB2 = FullSimplify[(\[Mu]/(4 Pi)) Ic*Cross[dl2, rp2]/(rp2.rp2)^(3/2)];
B2 = Integrate[dB2 /. {x -> 0, y -> 0}, {\[CurlyPhi], 0, 2 Pi}];
```

Ukupno magnetno polje u posmatranoj tački računato je kao doprinos svakog kalema nezavisno. Vektorsko polje duž z- ose je

$$\vec{B}_2 = \left(0, 0, \frac{4I_c\mu_0 R_0^2}{((d+2z)^2 + 4R_0^2)^{3/2}} \right)$$

$$\vec{B}_1 = \left(0, 0, \frac{4I_c\mu_0 R_0^2}{((d-2z)^2 + 4R_0^2)^{3/2}} \right).$$

Na osnovu principa superpozicije vektorskog polja, ukupno polje nalazi se kao zbir vektora doprinosa pojedinih kalemova

$$\vec{B}(x, y, z) = \vec{B}_1(x, y, z) + \vec{B}_2(x, y, z),$$

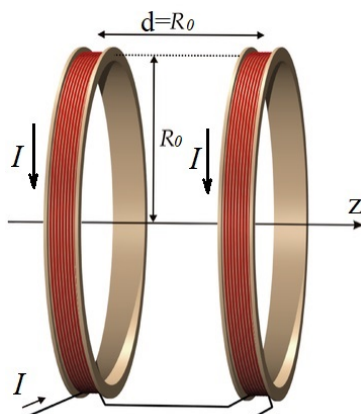
gde su vektori u tački čije su koordinate (x, y, z) , generisani kalemovima 1 i 2 redom \vec{B}_1 i \vec{B}_2 .

```
Btot=B1+B2;
rez=Normal[Series[Btot[[3]],{z,0,6}]]
```

Dalje, razvojem funkcije komponente $B_z(z)$ magnetnog polja u red oko koordinatnog početka (sredine dva paralelna kalema), dobije se razvoj čiji je najniži član nultog stepena.

$$B_z(z) \approx \frac{8I_c\mu_0 R_0^2}{(d^2 + 4R_0^2)^{3/2}} + \frac{192I_c\mu_0 R_0^2 z^2 (d^2 - R_0^2)}{(d^2 + 4R_0^2)^{7/2}} + \frac{1920I_c\mu_0 R_0^2 z^4 (d^4 - 6d^2 R_0^2 + 2R_0^4)}{(d^2 + 4R_0^2)^{11/2}}.$$

Helmholcovi kalemovi čine sistem od dva identična kružna paralelna kalema na rastojanju jednakom njihovom radijusu, kroz koje protiče struja istog intenziteta i smera. Kada se rastojanje d u prethodnoj relaciji zameni radijusom R_0 prema pravilima transformacija $BHlmh = rez/.\{d \rightarrow R0\}$, izraz za intenzitet magnetnog polja na sredini između Helmholtcovih kalemova duž ose koja prolazi kroz njihov centar se znatno pojednostavi. Uprošćenje se svodi na to da član koji zavisi kao z^2 iščezava, dok je sledeći koji preostaje z^4 . Što ukazuje da se magnetno polje na sredini između paralelnih kalemova duž ose koja prolazi normalno kroz njihove centre menja veoma sporo.

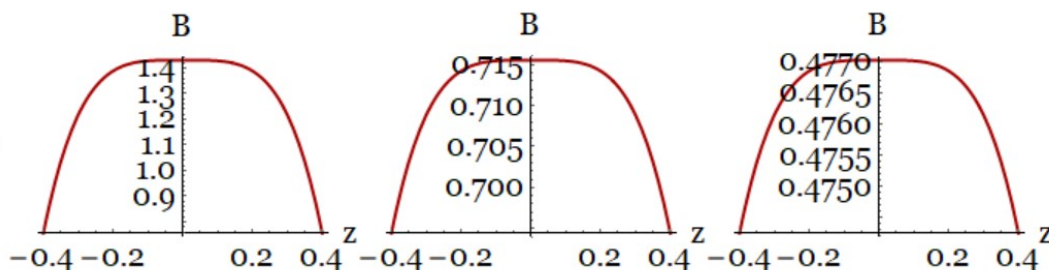


Slika 3.6: Helmholtzovi kalemovi radijusa R_0 koji se nalaze na međusobnom rastojanju R_0 .

```
BHlhm=rez/.{d->R0};(*magnetno polje Helmholtzovih kalemova*)
Table[Plot[BHlhm/.{Ic->1,\[Mu]->1,R0->ri},{z,-0.4,0.4},
AxesStyle->Directive[Black, 16],ImageSize->Small,
AxesLabel->{"z", "B"}],{ri,{0.5,1,1.5}}];
```

$$B(z) = \frac{8I_c\mu}{5\sqrt{5}\sqrt{R_0^2}} - \frac{1152I_c\mu_0R_0^6}{625\sqrt{5}(R_0^2)^{11/2}}z^4 \tag{3.4}$$

Zamenom rastojanja između kalemova d sa R_0 dobija se izraz (3.4). Funkcija magnetnog polja u bliskoj okolini centra Helmholtzovih kalemova ima plato gde se sporo menja. Polje sporo opada i u bliskoj okolini koordinatnog početka (između kalemova) i može se smatrati konstantnim.



Slika 3.7: Promena magnetnog polja duž z - ose oko centra Helmholtzovih kalemova različitih radijusa 0.5, 1.0 i 1.5 sleva nadesno. (Sve vrednosti date u arbitrarnim jedinicama.)

3.4 Sume

Naredbom

$$Sum[f_i, \{i, i_{min}, i_{max}\}]$$

vrši se sumiranje po relaciji

$$\sum_{i_{min}}^{i_{max}} f_i.$$

Funkcija *Sum* je jedna od funkcija sa iteratorom *i*, definisanim od i_{min} do i_{max} , koji se menja sa korakom 1, osim ako nije drugačije specificirano

$$Sum[f_i, \{i, i_{min}, i_{max}, di\}],$$

gde je *di* korak promene *i*. Pored jednostruke, suma može biti i višestruka, sa više brojača

$$Sum[f_{i,j,\dots}, \{i, i_{min}, i_{max}\}, \{j, j_{min}, j_{max}\}, \dots],$$

koja odgovara matematičkom izrazu

$$\sum_{i_{min}}^{i_{max}} \sum_{j_{min}}^{j_{max}} \dots f_{i,j,\dots}$$

Ukoliko konvergiraju naredbom *Sum* simbolički se mogu računati sume konačnih ali i beskonačnih redova.

3.4.26 Generisati opštu sumu oblika

$$1 + \frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \frac{1}{x^4} + \frac{1}{x^5} + \frac{1}{x^6}.$$

`Sum[1/x^i, {i, 0, 6}]`

3.4.27 Izračunati konačnu sumu

$$1 + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \frac{1}{5^4} + \frac{1}{6^4} + \frac{1}{7^4} + \frac{1}{8^4}.$$

`Sum[1/i^4, {i, 1, 8}]`

3.4.28 Generisati opštu sumu

$$1 + \frac{1}{x^2} + \frac{1}{x^4} + \frac{1}{x^6} + \frac{1}{x^8} + \frac{1}{x^{10}} + \frac{1}{x^{12}}.$$

`Sum[1/x^i, {i, 0, 12, 2}]`

Rešenje: $\frac{1}{x^{12}} + \frac{1}{x^{10}} + \frac{1}{x^8} + \frac{1}{x^6} + \frac{1}{x^4} + \frac{1}{x^2} + 1$.

3.4.29 Generisati opštu dvostruku sumu

$$\sum_{i=1}^3 \sum_{j=1}^i \frac{1}{x^i + y^j}.$$

`Sum[1/(x^i+y^j), {i, 1, 3}, {j, 1, i}]`

Rešenje: $\frac{1}{x^3+y^3} + \frac{1}{x^3+y^2} + \frac{1}{x^3+y} + \frac{1}{x^2+y^2} + \frac{1}{x^2+y} + \frac{1}{x+y}$.

3.4.30 Izračunati dvostruku sumu

$$\sum_{i=1}^5 \sum_{j=0}^{i+1} \frac{1}{(i^2 + j^3)^3}.$$

Rezultat prikazati u obliku približne numeričke vrednosti.

`Sum[1/(i^2 + j^3)^3, {i, 1, 5}, {j, 0, i + 1};
N[%]`

Rešenje: 1.1539.

3.4.31 U narednim primerima izračunati sume beskonačnih redova

$$\sum_{n=1}^{\infty} \frac{1}{(4n+1)(4n-3)}$$

`Sum[1/((4 n + 1) (4 n - 3)), {n, 1, Infinity}]`

Rešenje: 1/4.

3.4.32

$$\sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{2^{2n} (n!)^2}$$

`Sum[(-1)^i x^(2 i)/(2^(2 i) (i!)^2), {i, 0, Infinity}]`

Rešenje: $J_0(x)$.

3.4.33 Izračunati beskonačnu sumu

$$\sum_{i=1}^{\infty} \frac{1}{i(i+1)}$$

a potom i parcijalnu

$$\sum_{i=1}^n \frac{1}{i(i+1)}$$

`Sum[1/(i (i + 1)), {i, 1, Infinity}]`
`FullSimplify[Sum[1/(i (i + 1)), {i, 1, n}]]`

Rešenje beskonačne sume je 1, a parcijalne $n/(n+1)$.

3.4.34 Izračunati sume redova.

i) $\sum_{n=0}^{\infty} \frac{(-1)^n}{n!}$

`Sum[(-1)^n/n!, {n, 0, Infinity}]`

Rešenje: $\frac{1}{e}$.

ii) $\sum_{n=0}^{\infty} \frac{\cos^n(x)}{2^n}$

`Sum[Cos[x]^n/2^n, {n, 0, Infinity}]`

Rešenje: $-\frac{2}{\cos(x)-2}$.

iii) $\sum_{n=1}^{\infty} \frac{n^2+3n+1}{(n^2+n)^2}$

`Sum[(n^2+3n+1)/(n^2+n)^2, {n, 1, Infinity}]`

Rešenje: 2.

3.4.35 Izračunati degeneraciju energetskeg nivoa

i) atoma vodonika;

ii) trodimenzionalnog izotropnog harmonijskog oscilatora.

Rešenje:

- ii) Za označavanje stanja elektrona u atomu vodonika $\psi_{n,l,m_l,m_s}(r, \theta, \varphi)$ koriste se 4 kvantna broja: glavni kvantni broj n , orbitalni ili azimutalni l , magnetni kvantni broj m_l i spinski magnetni kvantni broj m_s . Dozvoljene vrednosti navedenih kvantni brojeva su

$$\begin{aligned} n &= 1, 2, 3, \dots \\ l &= 0, 1, 2, \dots, n-1 \\ m_l &= -l, -l+1, \dots, 0, 1, 2, \dots, l \\ m_s &= -1/2, 1/2 \end{aligned}$$

Spektar atoma vodonika dat je izrazom

$$E_n = \frac{-13.6eV}{n^2},$$

odakle se vidi da energija zavisi samo od glavnog kvantnog broja n , što znači da postoji najmanje 2 ili više stanja sa istom energijom. Ova stanja se nazivaju degenerisanim. Vrednost m_s je nezavisna od glavnog kvantnog broja i broji dva stanja. Vrednost $m_l = -l, \dots, 0, \dots, l$ broji $2l + 1$ različitih vrednosti, pri čemu za svako n , angularni kvantni broj uzima vrednosti $l = 0, \dots, n-1$. Odavde, ukupan broj stanja koja imaju vrednost energije E_n je

$$\sum_{l=0}^{n-1} 2(2l+1) = 2n^2.$$

Sum[2 (2 l + 1), {1, 0, n - 1}]

- ii) Spektar hamiltonijana trodimenzionalnog izotropnog harmonijskog oscilatora je

$$E_n = \hbar\omega \left(n + \frac{3}{2} \right), \quad n = n_x + n_y + n_z,$$

gde je $\hbar = h/2\pi$ redukovana plankova konstanta a $\omega = \sqrt{k/m}$ kružna učestalost. Stanja sa različitim n_x, n_y, n_z čija suma je ista i iznosi $n = n_x + n_y + n_z$ imaju istu energiju E_n i degenerisana su. Za svako n , broj n_x može da ima vrednosti od 0 do n tj. ukupno $n + 1$ vrednost. Potom za svako n_x vrednost n_y može da ima $n + 1 - n_x$ vrednosti, dok jedina vrednost za n_z ostaje $n - n_x - n_y$. Degeneracija nivoa E_n je

$$\sum_{n_x=0}^n (n - n_x + 1) = (n + 1)(n + 2)/2.$$

Sum[(n - i + 1), {i, 0, n}]

3.4.36 Kruti rotator- ilustracija kvadrata modula talasne funkcije.

Najpre će biti dobijen izraz za evoluciju kvantnog stanja $\psi(\theta, \varphi, t)$ krutog rotatora (u prostoru), koje je linearna kombinacija 9 najnižih svojstvenih stanja Hamiltonijana

$$H|J, M\rangle = E_J|J, M\rangle$$

čiji je spektar

$$E_J = J(J+1)b,$$

gde je $b = \frac{\hbar}{8\pi^2 I c}$ rotaciona konstanta. Potom će biti ilustrovan kvadrat modula dobijene talasne funkcije.

Angularni kvantni broj J može imati vrednosti $J = 0, 1, \dots$, dok je magnetni kvantni broj određen sa J i ima vrednosti $m = -J, \dots, J$. To znači da $2J + 1$ vektora opisuju stanja sa energijom E_J . U koordinatnoj reprezentaciji vektori stanja su sferni harmonici

$$Y_J^M(\theta, \phi) = \sqrt{\frac{2J+1}{4\pi} \frac{(J-|M|)!}{(J+|M|)!}} P_J^{|M|}(\cos \theta) e^{iM\phi},$$

gde su specijalne funkcije $P_J^{|M|}$ asocirani Ležandrovi polinomi. Stanje $|\psi\rangle$ možemo izraziti kao linearnu kombinaciju vektora čiji su orbitalni angularni kvantni brojevi $J = 0, 1, 2$ i ukupno ih ima 9. Zadato stanje u početnom trenutku $t = 0$, dato kao linearna kombinacija sfernih harmonika je

$$\psi(\theta, \phi, 0) = \sum_{J=0}^2 \sum_{M=-J}^J c_{J,M} Y_J^M(\theta, \phi).$$

Evolucija stanja stacionarnog sistema, čija energija ne zavisi od vremena $H \neq f(t)$ zadata je preko operatora evolucije

$$U(t) = e^{-\frac{i}{\hbar} H t}.$$

Zapisano u Dirakovoj notaciji, stanje sistema u zavisnosti vremena može se napisati preko operatora evolucije

$$\begin{aligned} |\psi(\theta, \phi, t)\rangle &= U(t) |\psi(\theta, \phi, 0)\rangle \\ |\psi(\theta, \phi, t)\rangle &= \sum_{J=0}^2 \sum_{M=-J}^J c_{J,M} U(t) |J, M\rangle. \end{aligned}$$

Hamiltonijan je hermitski (uz to i normalan) operator, koji se može prikazati spektralnom formom

$$H = \sum_{J'=0}^2 E_{J'} \sum_{M'=-J'}^{J'} |J', M'\rangle \langle J', M'|,$$

odakle se, u skladu sa relacijom za funkcije normalnih operatora, može napisati i unitarni operator evolucije, kao analitička funkcija Hamiltonovog operatora

$$U(t) = e^{-\frac{i}{\hbar} H t} = \sum_{J'=0}^2 e^{-\frac{i}{\hbar} E_{J'} t} \sum_{M'=-J'}^{J'} |J', M'\rangle \langle J', M'|.$$

Kada se operator evolucije izražen u spektralnoj formi unese u izraz za evoluciju stanja dobije se funkcija

$$|\psi(\theta, \phi, t)\rangle = \sum_{J=0}^2 \sum_{M=-J}^J c_{J,M} \sum_{J'=0}^2 e^{-\frac{i}{\hbar} E_{J'} t} \sum_{M'=-J'}^{J'} |J', M'\rangle \langle J', M'| J, M\rangle.$$

Iz uslova $\langle J', M'| J, M\rangle = \delta_{JJ'} \delta_{MM'}$, gornja relacija postaje

$$|\psi(\theta, \phi, t)\rangle = \sum_{J=0}^2 \sum_{M=-J}^J c_{J,M} e^{-\frac{i}{\hbar} E_J t} |J, M\rangle.$$

Pošto je reč o stanju kvantnog sistema, vektor koji ga opisuje normiran je na 1

$$1 = \langle \psi(\theta, \phi, t) | \psi(\theta, \phi, t) \rangle$$

odakle se dobija relacija

$$\begin{aligned} \langle \psi(\theta, \phi, t) | \psi(\theta, \phi, t) \rangle &= \left(\sum_{J'=0}^2 \sum_{M'=-J'}^{J'} c_{J',M'}^* e^{i\hbar E_{J'} t} \langle J', M' | \right) \left(\sum_{J=0}^2 \sum_{M=-J}^J c_{J,M} e^{-i\hbar E_J t} |J, M\rangle \right) = \\ &= \sum_{J'=0}^2 \sum_{M'=-J'}^{J'} \sum_{J=0}^2 \sum_{M=-J}^J c_{J',M'}^* e^{i\hbar E_{J'} t} c_{J,M} e^{-i\hbar E_J t} \langle J', M' | J, M \rangle = \\ &= \sum_{J=0}^2 \sum_{M=-J}^J c_{J,M}^* c_{J,M} e^{i\hbar E_J t} e^{-i\hbar E_J t} = \sum_{J=0}^2 \sum_{M=-J}^J |c_{J,M}|^2 = 1, \end{aligned}$$

odnosno suma kvadrata modula koeficijenata u linearnoj kombinaciji jednaka je 1. Koeficijenti početnog stanja $|\psi(\theta, \phi, 0)\rangle$ su generisani kao lista dužine $2J + 1$ za $J = 2$ slučajni brojeva između 0 i 1 $coeff = Table[Random[], \{Length[QnumStates]\}]$. Potom je ova lista normirana na jedinicu $norm = Sqrt[coeff.coeff]$; $coeff = coeff/norm$. Izraženo u koordinatnoj reprezentaciji, gde je $\langle \theta, \phi | J, M \rangle = Y_J^M(\theta, \phi)$, evolucija stanja se može napisati kao

$$\psi(\theta, \phi, t) = \sum_{J=0}^2 \sum_{M=-J}^J c_{J,M} e^{-i\hbar E_J t} Y_J^M(\theta, \phi)$$

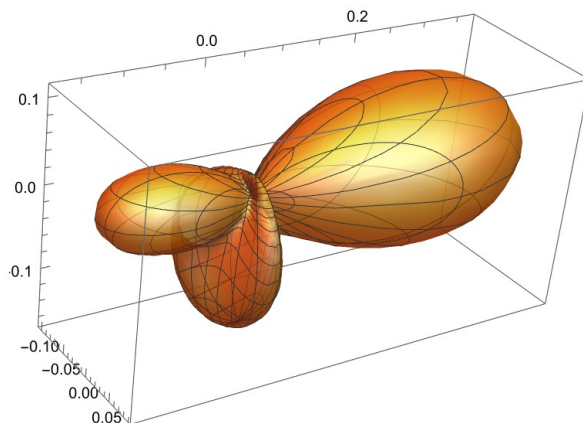
```
J = 2;
QnumStates=Table[{1, m},{1, 0, J},{m, -1, 1, 1}];
QnumStates=Flatten[QnumStates,1];
NumOfState=Length[Flatten[QnumStates,1]];
coeff0=Table[c["j","m"] /. {"j"->QnumStates[[i, 1]],
"m" -> QnumStates[[i, 2]]}, {i, Length[QnumStates]}];
coeff=Table[Random[], {Length[QnumStates]}];
norm=Sqrt[coeff.coeff];
coeff=coeff/norm;
coeff.coeff
pom=Transpose[{QnumStates, coeff}];
\[Psi]=Sum[pom[[i, 2]]
Exp[-I*2 Pi*pom[[i, 1, 1]] (pom[[i, 1, 1]] + 1) t]
SphericalHarmonicY["1","m",\[Theta],\[CurlyPhi]]/.
{"1" -> pom[[i, 1, 1]],"m" -> pom[[i, 1, 2]]},
{i, Length[pom]}
```

U nastavku će dobijena talasna funkcija $\psi(\theta, \phi, t)$ biti stavljena u modul, podignuta na kvadrat i prikazana kao trodimenzionalni grafik. Naredbom

$$SphericalPlot3D[r, \{\theta, \theta_{min}, \theta_{max}\}, \{\phi, \phi_{min}, \phi_{max}\}]$$

generiše se trodimenzionalni grafik sfernog radijusa r u funkciji sfernih koordinata θ i ϕ .

```
Manipulate[SphericalPlot3D[Abs[\[Psi]]^2 /. t->t1, {\[Theta], 0, Pi},
{\[CurlyPhi], 0, 2Pi},
PlotStyle->
Directive[Orange, Opacity[0.7], Specularity[White, 10]], PlotRange->All],
{t1, 0, 1, 0.001}]
```



3.5 Diferenciranje

Izvod funkcije $f(x)$ ili parcijalni izvod funkcije više promenljivih

$$\frac{\partial f}{\partial x}$$

u odnosu na x u programu Mathematica realizuje se sa

$$D[f, x].$$

Uvežbavanje primene ove naredbe biće izvršeno na prostom primeru diferenciranja funkcije

$$\frac{\cos(4x)}{1 - \sin 4x}.$$

Dobijeni izraz uprostiti, a potom naći njegovu vrednost za $x = 0$.

```
expr = D[Cos[4 x]/(1 - Sin[4 x]), x]
expr=FullSimplify[expr]
expr /. {x -> 0}
```

Nakon izvršenja naredbe dobija se izvod zadate funkcije

$$\frac{4 \cos^2(4x)}{(1 - \sin(4x))^2} - \frac{4 \sin(4x)}{1 - \sin(4x)},$$

a potom njegovim uprošćavanjem izraz

$$-(4/(-1 + \sin[4x])).$$

Zamenom promenljive x vrednošću 0 u diferenciran i uprošten izraz dobije se 4.

Izvodi višeg reda, na primer računanje n -tog izvod funkcije f

$$\frac{\partial^n f}{\partial x^n}$$

u programu Mathematica realizuje se sa

$$D[f, \{x, n\}].$$

Primer radi, četvrti izvod funkcije

$$4x^2 - 5x + 8 - \frac{3}{x}$$

dobija se izvršavanjem linije

```
D[4x^2-5x+8-3/x, {x, 4}]
```

čije je rešenje $-\frac{72}{x^5}$. Višestruki sukcesivni parcijalni izvodi funkcije više promenljivih f po x_1, x_2, \dots označen sa

$$\frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \dots f,$$

što će biti ilustrovano traženjem drugog parcijalnog izvoda funkcije dve promenljive

$$f(x, y) = x^3y^2 - 2x^2y + 3x$$

```
D[x^3*y^2-2x^2*y+3x, x, y]
```

čije je rešenje $6x^2y - 4x$. Naredbom

$$Dt[f]$$

nalazi se diferencijal df funkcije f . Njegova primena biće ilustrovana primerom računanja greške procene zapremine kutije izmerenih dimenzija $75cm \times 60cm \times 40cm$, pri čemu je greška merenja svake dimenzije jednaka $0.2cm$.

```
V=x y z
dV=Dt [V]
dV=dV/.{x->75, y->60, z->40, Dt [x]->0.2, Dt [y]->0.2, Dt [z]->0.2}
```

Rezultat primene funkcije Dt na V je $yzdx + xzdy + xydz$, nakon čega je izvršena zamena zadatih vrednosti dimenzija kutije i njihovih grešaka.

3.5.37 Izračunati prvi izvod datih funkcija

i) $f(x) = \sin(e^{x^2})$

```
D[Sin[Exp[x^2]], x]
```

Rešenje: $2e^{x^2} x \cos(e^{x^2})$.

ii) $f(x) = \frac{x^{3/4}\sqrt{x^2+1}}{(3x+2)^5}$

```
sol=D[x^(3/4)Sqrt[x^2+1]/(3 x+2)^5, x]
Simplify[sol]
```

Rešenje: $\frac{-39x^3+14x^2-51x+6}{4\sqrt[4]{x}(3x+2)^6\sqrt{x^2+1}}$.

iii) $f(x) = x \sinh^{-1}(x/3) - \sqrt{9+x^2}$

```
sol = D[(x*Sinh[x/3]^(-1)) - Sqrt[9 + x^2], x]
Simplify[sol]
```

Rešenje: $\left(1 - \frac{1}{3}x \coth\left(\frac{x}{3}\right)\right) \operatorname{csch}\left(\frac{x}{3}\right) - \frac{x}{\sqrt{x^2+9}}$.

3.5.38 Naći drugi izvod funkcije

$$f(x) = e^x \sin x$$

D[Exp[x] Sin[x], {x, 2}]

Rešenje: $2e^x \cos(x)$.3.5.39 Naći parcijalne izvode $\frac{\partial^3 f}{\partial y^2 \partial x}$ funkcije

$$f(x, y) = \ln(x^2 + y)$$

D[Log[x^2 + y], x, y]

Rešenje: $-\frac{2x}{(x^2+y)^2}$.3.5.40 **Srednja energija i toplotni kapacitet statističkog sistema.**

Biće razmotren sistem koji ima samo tri energijska stanja nenegativnih energija $\varepsilon_1 = 0, \varepsilon_2 = \varepsilon_3 = \varepsilon$. Stanje energije ε je degenerisano sa degeneracijom $g_2 = 2$. Biće izveden izraz za srednju energiju sistema i nacrtana njena zavisnost od temperature. Potom će biti izvedena i nacrtana funkcija toplotnog kapaciteta u zavisnosti od temperature. Particiona funkcija opisuje statističke osobine sistema u termodinamičkoj ravnoteži, ona je funkcija temperature T i energije mikrostanja $\varepsilon_1, \varepsilon_2, \dots$. Odnosno, opisuje sa kojom verovatnoćom su mikrostanja kao specifične konfiguracije u odnosu na položaj i impuls svake individualne čestice ili komponente, partitionisana prema svojim energijama. U modelu mikroskopskih konstituenata sistema, mogu se naći energije mikrostanja, potom partitiona funkcija koja dalje omogućava nalaženje drugih termodinamičkih karakteristika sistema. Zavisnost partitione funkcije od doprinosa pojedinih energijskih nivoa je

$$Z = \sum_i g_i e^{-\beta \varepsilon_i}, \quad (3.5)$$

što znači da je partitiona funkcija zadatog sistema

$$Z = 1 + 2E^{-\frac{\varepsilon}{kT}},$$

gde je k Bolcmanova konstanta a T apsolutna temperatura. Verovatnoća p_j da je sistem u mikrostanju j je

$$p_j = \frac{e^{-\beta \varepsilon_j}}{Z},$$

što omogućava partitionoj funkciji ulogu normalizacione konstante pri čemu važi

$$\sum_j p_j = \frac{1}{Z} \sum_j e^{-\beta \varepsilon_j} = \frac{1}{Z} Z = 1.$$

Ukupna energija, kao termodinamička veličina, dobije se usrednjavanjem u vidu očekivane vrednosti, što je suma energija mikrostanja otežinjena njihovim verovatnoćama

$$\langle E \rangle = \sum_i \varepsilon_i p_i = \frac{1}{Z} \sum_i \varepsilon_i e^{-\beta \varepsilon_i} = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = -\frac{\partial \ln Z}{\partial \beta}, \quad (3.6)$$

ili ekvivalentno

$$\langle E \rangle = kT^2 \frac{\partial \ln Z}{\partial T}, \quad (3.7)$$

čiji izraz je simbolički dobijen sledećim programom.

```
(*particiona funkcija Z*)
Zpt=Sum[g[i] Exp[-\[Beta]*\[Epsilon][i]],{i,1,n}]
(*izvod Z po beta *)
D[Zpt, \[Beta]]/Zpt
(*izvod Z po T, nakon prethodne zamena beta sa 1/kT*)
Expand[kb*T^2 D[Log[Zpt]/.\[Beta]->1/(kb*T)],T]]
```

Najpre je definisan skup svojstvenih energija σ , pri čemu se neke ponavljaju u skladu sa degeneracijom stanja. Potom se prema (3.5) računa particiona funkcija, iz koje se dalje prema relaciji (3.6) nalazi srednja energija.

```
(*srednja energija sistema *)
(*spektar energija*)
\[Sigma] = {0, \[Epsilon], \[Epsilon]};
(*particiona funkcija sistema*)
Z = Sum[Exp[-\[Beta] en], {en, \[Sigma]}]
ave=FullSimplify[-D[Log[Z], \[Beta]]]
```

Izvršavanjem prethodnog programa dobije se particiona funkcija

$$Z = 2e^{-\beta\varepsilon} + 1$$

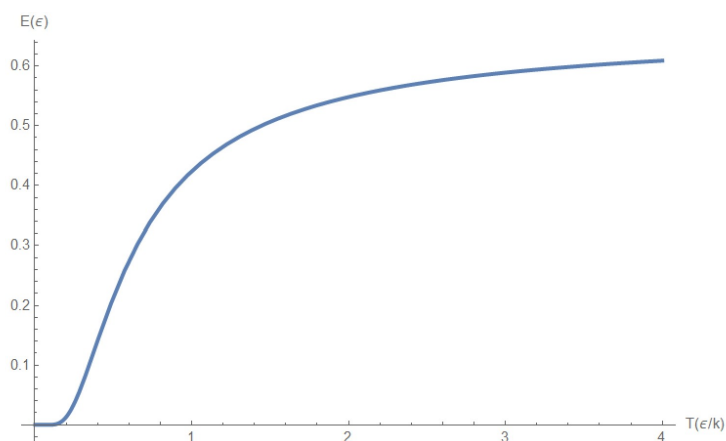
i srednja energija sistema

$$\langle E \rangle = \frac{2\varepsilon}{2 + e^{\beta\varepsilon}}.$$

Radi lakšeg prikaza zavisnosti srednje energije od temperature uvodi se smena $y = \langle E \rangle / \varepsilon$ i $t = kT / \langle E \rangle$, nakon čega se dobije funkcija zavisnosti od bezdimenzionih veličina

$$y(t) = \frac{2}{2 + e^{\frac{1}{t}}},$$

```
Plot[2/(2+Exp[1/t]),{t,0,4},
AxesLabel->{"T(\[Epsilon]/k)", "E(\[Epsilon])"}]
```



Slika 3.8: Srednja vrednost energije sistema $\langle E \rangle$ prikazana u zavisnosti od temperature T , datih u jedinicama ε i ε/k redom.

Izraz za toplotni kapacitet C , definisan kao

$$C = \frac{\partial \langle E \rangle}{\partial T}, \quad (3.8)$$

biće simbolički dobijen prema datoj definiciji

```
(*u izrazu za srednju energiju zameni se beta sa 1/kT *)
(*potom se izvrši diferenciranje *)
Cheat = D[avE /. {\[Beta] -> 1/(k*T)}, T]
Cheat = FullSimplify[Cheat];
(*smena velicina u izrazu za toplotni kapacitet*)
ch = (Cheat/k) /. {\[Epsilon]/(k*T) -> (1/t)};
ch = ch /. {\[Epsilon]^2/(k^2*T^2) -> (1/t^2)}
```

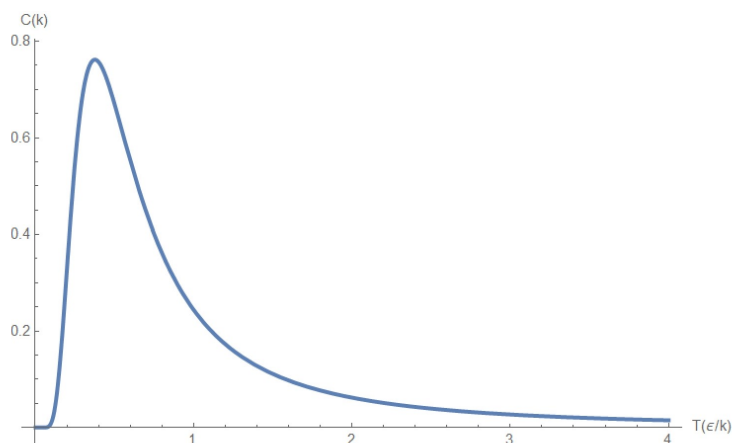
Toplotni kapacitet u zavisnosti od temperature i energije stanja je

$$C(T, \varepsilon) = \frac{2\varepsilon^2 e^{\frac{\varepsilon}{kT}}}{kT^2 \left(e^{\frac{\varepsilon}{kT}} + 2 \right)^2}.$$

Smenom promenljivih $c = C/k$, $t = kT/\varepsilon$ dobije se funkcija

$$c(t) = \frac{2e^{\frac{1}{t}}}{\left(e^{\frac{1}{t}} + 2 \right)^2 t^2},$$

odakle se može nacrtati grafik zavisnosti C od T u jedinicama k i ε/k redom.



Slika 3.9: Toplotni kapacitet sistema od tri stanja energija $\{0, \varepsilon, \varepsilon\}$, $\varepsilon > 0$.

Prema definiciji, entropija je očekivana vrednost logaritma verovatnoće popunjenosti stanja ili mera nedovoljnog poznavanja karakteristika sistema.

$$S = -k \sum_i p_i \ln p_i = k(\ln Z + \beta \langle E \rangle) = \frac{\partial(kT \ln Z)}{\partial T} \quad (3.9)$$

Prema definiciji (3.9) sledećom linijom programa računa se entropija zadatog sistema

```
S=FullSimplify[k Log[Z]+avE/T]
```

odakle se dobija uprošćen izraz

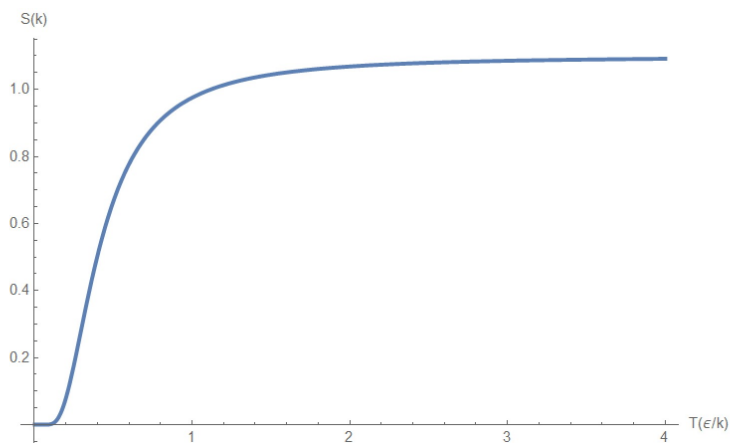
$$k \log \left(2e^{-\beta\varepsilon} + 1 \right) + \frac{2\varepsilon}{T \left(e^{\beta\varepsilon} + 2 \right)},$$

a ako se u gore datim bezdimenzionim veličinama izraze temperatura i entropija $s = S/k$ dobije se funkcija

$$s(t) = \frac{2}{(e^{1/t} + 2)t} + \log(2e^{-1/t} + 1)$$

nacrtna izvršavanjem sledeće linije.

```
Plot[2/((2 + E^(1/t)) t)+Log[1+2 E^(-1/t)], {t,0,4},
AxesLabel -> {"T(\[Epsilon]/k)", "S(k)"}, PlotRange -> All]
```



Slika 3.10: Entropija sistema.

Koristeći naredbe diferenciranja programskog jezika Wolfram biće primenjene za rešavanje problema vezanih male oscilacije. Diferenciranjem kinetičke i potencijalne energije, računace se matricni elementi matrice iz koje će se dalje rešavanjem karakteristične jednačine tražiti svojstvene frekvencije sistema.

Male oscilacije.

Sile koje se javljaju u velikom broju fizičkih sistema, gde su uključena istezanja, kompresije, savijanje ili torzija, su samo približno linearne funkcije za mala pomeranja, koje dovode do prostih harmonijskih oscilacija. Najpogodnije za opisivanje linearnih sistema sa mnogo stepeni slobode, poput spregnutih oscilatora, jeste pronalaženje normalnih modova. U nastavku biće razmotreni oni sistemi kojima je oscilatorno kretanje opisano potencijalom i odvija se u okolini ravnotežnog stanja.

Male oscilacije spregnutog sistema čestica, frekvencije zajedno sa odgovarajućim pomerajima, biće dobijene iz Lagranžijana

$$L = T - U, \quad T = \sum_{i=1}^N \frac{m_i \dot{\mathbf{r}}_i^2}{2}$$

koji odgovara sistemu u kom čestice intereaguju preko centralnog potencijala

$$U = \sum_{i < j} U(|\mathbf{r}_i - \mathbf{r}_j|),$$

koji zavisi samo od međučestičnog rastojanja. Najpre će biti izvršena transformacija iz Dekartovih $\{\mathbf{r}_i\}$ u nove, generalisane koordinate $\{q_\alpha\}$, preko odgovarajućih funkcija:

$$q_\alpha = f_\alpha(\{\mathbf{r}_i\}),$$

odakle se dobija linearna zavisnost $\dot{\mathbf{r}}_i$ i \dot{q}_α

$$\dot{q}_\alpha = \sum_i \frac{\partial f_\alpha}{\partial \mathbf{r}_i} \dot{\mathbf{r}}_i.$$

U skladu sa izvršenom transformacijom koordinata, Lagranžijan u generalisanim koordinatama postaje

$$L = \frac{1}{2} \sum_{\alpha, \beta} m_{\alpha\beta} \dot{q}_\alpha \dot{q}_\beta - U(\{q\}),$$

gde su $m_{\alpha\beta}$ matricni elementi koji zavise od q , ali ne i od \dot{q} . Dinamički sistem sa N stepeni slobode koji se nalazi u blizini lokalnog minimuma potencijalne energije $U(q_1, \dots, q_N) = U(\{q_i\})$. Neka se u konfiguracionom prostoru lokalni minimum nalazi u $q_i = q_i^{(0)}$, $i = 1, 2, \dots, N$. Tada je u tački $\{q_1^{(0)}, \dots, q_N^{(0)}\}$ sistem u stabilnoj ravnoteži, a sile koje deluju na njega jednake su nuli. Smatra se da pri malom izvođenju iz ravnoteže nije moguće trajno izvesti sistem daleko od ravnotežne stanja. Dinamika ovog sistema može se opisati uz pomoć Tejlorovog razvoja potencijala u okolini ravnotežne konfiguracije $x_i = q_i - q_i^{(0)}$, $\dot{x}_i = \dot{q}_i$

$$U(q_1, \dots, q_N) = U(\{q_i^{(0)}\}) + \sum_i \left. \frac{\partial U}{\partial q_i} \right|_0 + \frac{1}{2} \sum_{i,j} \left. \frac{\partial^2 U}{\partial q_i \partial q_j} \right|_0 x_i x_j \dots$$

Prvi član u razvoju potencijala $U(\{q_i^{(0)}\})$ je konstantan i može se postaviti da je jednak nuli. Drugi član je takođe jednak nuli zbog toga što sile ne deluju na sistem u ravnotežnoj konfiguraciji $\partial U / \partial q_i|_0 = 0$. Zbog toga je drugi član dominantan u razvoju. Ako je matricni element matrice \mathcal{U} zadat kao

$$U_{ij} = \left. \frac{\partial^2 U}{\partial q_i \partial q_j} \right|_0$$

tada se promena potencijala može zadati kao

$$U(\{q_i\}) = \frac{1}{2} \sum_{ij} U_{ij} q_i q_j,$$

gde je u okolini ravnotežnog položaja \mathcal{U} konstantna realna i simetrična matrica. Član od kog zavisi kinetička energija, matricni elementi

$$m_{\alpha\beta}(\{q\}) = m_{\alpha\beta}(\{q^{(0)}\})$$

ne zavise od položaja. Dobijen je bilinearni Lagranžijan

$$L = \frac{1}{2} \sum_{\alpha\beta} (m_{\alpha\beta} \dot{q}_\alpha \dot{q}_\beta - U_{\alpha\beta} q_\alpha q_\beta), \quad (3.10)$$

gde su $(\mathcal{M})_{\alpha\beta} = m_{\alpha\beta}$ i $(\mathcal{U})_{\alpha\beta} = U_{\alpha\beta}$ matricni elementi konstantnih matrica. U vektorskoj notaciji, uvođenjem kolone $Q = (q_1, q_2, \dots, q_N)^T$ Lagranžijan se može napisati u vektorsko-matricnoj notaciji

$$L = \frac{1}{2} (\dot{Q}^T \mathcal{M} \dot{Q} - Q^T \mathcal{U} Q).$$

Tako se jednačine kretanja sistema spregnutih oscilatora mogu izraziti preko matrice potencijalne i kinetičke energije. Jedino, za određene početne uslove sistem će oscilovati prirodnim frekvencijama, takozvanim normalnim modama. Iz Lagranževih jednačina kretanja

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_\alpha} - \frac{\partial L}{\partial q_\alpha} = 0, \quad \alpha = 1, 2, \dots, N \quad (3.11)$$

dobija se

$$\sum_{\beta} m_{\alpha\beta} \ddot{q}_{\beta} + U_{\alpha\beta} q_{\beta} = 0, \quad \alpha = 1, 2, \dots, N \quad (3.12)$$

u matricnoj formi zapisano kao

$$\mathcal{M}\ddot{Q} + \mathcal{U}Q = 0. \quad (3.13)$$

Rešenje diferencijalne jednačine (3.13) je oblika

$$Q = Amp \sin \omega t,$$

gde je $Amp = (a_1, a_2, \dots, a_N)$ lista pomeraja sa odgovarajućim fazama. Unošenjem rešenja u diferencijalnu jednačinu (3.13) rezultat je

$$(-\omega^2 \mathcal{M} + \mathcal{U}) Amp = 0, \quad (3.14)$$

koja ima netrivialno rešenje pod uslovom da je

$$|-\omega^2 \mathcal{M} + \mathcal{U}| = 0. \quad (3.15)$$

Definisana determinanta je karakteristični polinom čije su nule svojstvene frekvencije ω_{α}^2 , $\alpha = 1, 2, \dots, N$. Iz uslova da potencijalna energija ima lokalni minimum oko ravnotežnog položaja, dobijene vrednosti su pozitivne, a svojstveni vektori Amp su realni (imaju sve komponente realne). Sve čestice sistema koji se kreće prema obrascu normalnih moda, osciluju istom frekvencijom. Ukupno rešenje kretanja sistema uključuje superpoziciju normalnih moda u odgovarajućim fiksiranim faznim pomacima.

3.5.41 Funkcija kojom se traže normalne mode malih oscilacija.

Prvi zadatak je da se kreira korisnička funkcija kojom se nalaze svojstvene frekvencije i svojstveni vektori pomeraja čestica oscilatornog sistema na osnovu jednačine (3.14), čija će struktura biti oblika

$$SmallOsc[T, U, \{q_1[t], q_2[t], \dots\}, \{q_{10}, q_{20}, \dots\}],$$

gde su q_i , $i = 1, 2, \dots$ koordinate merene od ravnotežnog položaja q_{i0} . Dobijenu funkciju primeniti na dva tela iste mase m spojena oprugom koeficijenta elastičnosti k , koja mogu slobodno da osciluju duž pravca koji čine sa oprugom.

```

SmallOscill[T_, U_, coord_, equilb_] :=
Module[{dim, movvec, p, equilbr, Vij, Tij,
  mtrx, \[Omega]2, \[Omega]2sol, freq, eqs, a, vec},
  dim = Length[coord];
  (*komponente vektora pomeraja*)
  movvec = Array[a, dim];
  p = D[coord, t];
  equilbr = Thread[coord -> equilb];
  Vij = Table[D[U, i, j], {i, coord}, {j, coord}];
  Tij = Table[D[T, i, j], {i, p}, {j, p}];
  mtrx = Vij - \[Omega]2*Tij;
  (*izjednacavanje determinante sa 0,
  trazenje kvadrata frekvencije normalne mode*)
  \[Omega]2sol = Solve[Det[mtrx] == 0, \[Omega]2];
  (*lista dobijenih frekvencija*)
  freq = \[Omega]2 /. \[Omega]2sol;
  mtrx = mtrx /. equilbr;
  eqs = FullSimplify[mtrx.movvec];
  eqs == Table[0, {Length[eqs]}];
  eqs = ExpandAll[Thread[eqs == Table[0, {Length[eqs]}]]];
  (*svojtveni vektori*)
  vec=Table[
  movvec /.
  Solve[Join[(eqs /. {\[Omega]2 -> freq[[i]]}),
  {movvec.movvec==1}],movvec][[2]], {i, Length[freq]}];
  {Sqrt[freq], vec}]

```

Kinetička i potencijalna energija za dati primer dva tela masa m duž x pravca vezana oprugom konstante elastičnosti k koja osciluju su

$$T = \frac{1}{2}mq_1'(t)^2 + \frac{1}{2}mq_2'(t)^2, \quad U = \frac{1}{2}k(q_2(t) - q_1(t))^2.$$

```

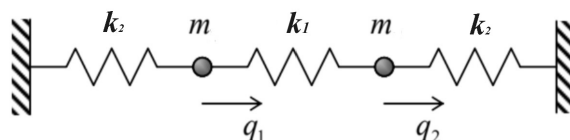
(*kineticka energija oscilatora*)
T = (m*q1'[t]^2 + m*q2'[t]^2)*1/2;
(*potencijalna energija oscilatora*)
V = k*(q2[t] - q1[t])^2/2;
(*koordinate:odstupanja od ravnoteznih poloza*)
coord = {q1[t], q2[t]};
(*ravnotezni poloza*)
equilb = {0, 0};
SmallOscill[T, V, coord, equilb]

```

Rešenje datog primera je $\left(\begin{array}{cc} 0 & \sqrt{\frac{2k}{m}} \\ \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\} & \left\{ \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right\} \end{array} \right)$ pri čemu su frekvencije date u prvoj a odgovarajući pomeraji čestica odnosno svojstveni vektori u drugoj vrsti.

- 3.5.42 Biće razmotrene slobodne oscilacije dve čestice masa m međusobno vezane oprugom elastičnosti k_1 i svaka od njih duž zadatog pravca vezana za nepokretni zid oprugom konstante elastičnosti k_2 . Biće analizirane male oscilacije pri pomerajima iz ravnotežnog položaja q_1 i q_2 . Potom, koristeći definisanu korisničku funkciju u prethodnom zadatku biće nađene svojstvene frekvencije i vektori pomeraja.

Reč je o jednodimenzionalnom oscilatornom sistemu sastavljenom od dva linearna harmonijska oscilatora koja osciluju bez disipacija. Kinetička energija je separabilna te se može napisati



Slika 3.11: Oscilator od dva oprugom spregnuta tela vezanih za zidove.

u obliku

$$T = T_1 + T_2 = \frac{1}{2}m\dot{q}_1^2 + \frac{1}{2}m\dot{q}_2^2.$$

Ukupna potencijalna energija, koja se sastoji od elastičnih energija tri opruge nije separabilna i data je kao

$$U = \frac{1}{2}k_2q_1^2 + \frac{1}{2}k_1(q_1 - q_2)^2 + \frac{1}{2}k_2q_2^2.$$

```
(*kineticka energija oscilatora*)
T = (m*q1'[t]^2)/2 + (m*q2'[t]^2)/2;
(*potencijalna energija oscilatora*)
U = (k2*q1[t]^2 + k1*(q1[t] - q2[t])^2 + k2*q2[t]^2)/2;
(*koordinate: odstupanje tela od ravnoteznih poloazaja*)
coord = {q1[t], q2[t]};
(*ravnotezni poloazaji*)
equilb = {q10, q20};
w2vect = Small0scill[T, U, coord, equilb]
```

Opšte rešenje je linearna kombinacija dobijenih članova oblika

$$c_1Q_1 \sin(\omega_1 t + \phi_1) + c_2Q_2 \sin(\omega_2 t + \phi_2).$$

```
(*izdvajanje svojstvenih frekvencija*)
{w1, w2} = w2vect[[1]];
(*izdvajanje svojstvenih vektora*)
{Q1, Q2} = w2vect[[2]];
(*pravila zamene koordinata*)
q12rule =
  Thread[{q1[t], q2[t]} ->
    Total[w2vect[[2]] {c1*Cos[w1*t + fi1], c2*Cos[w2*t + fi2]}];
```

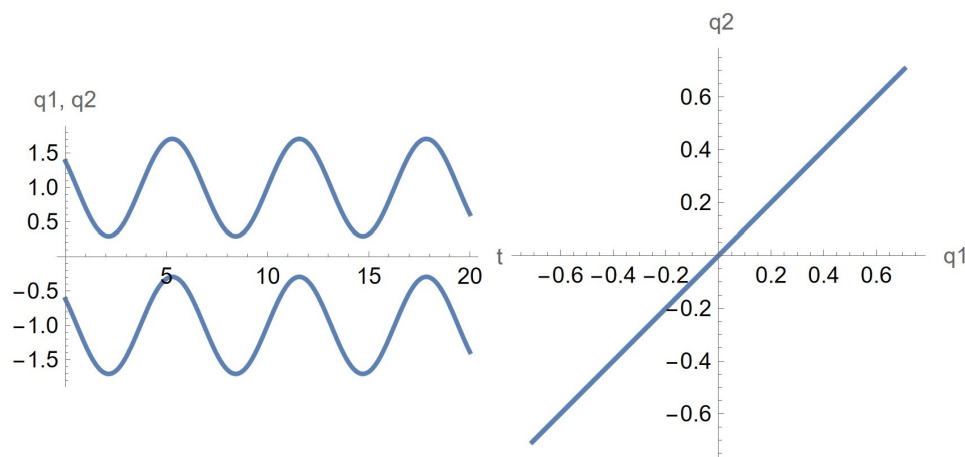
Izvršenjem gore datog programa dobiju se pravila zamene odstupanja od ravnotežnog položaja oscilatora ($q_1(t), q_2(t)$) u zavisnosti od vremena sledećim funkcijama

$$\left(\frac{c_1 \cos\left(\varphi_1 + t\sqrt{\frac{k_2}{m}}\right)}{\sqrt{2}} + \frac{c_2 \cos\left(\varphi_2 + t\sqrt{\frac{2k_1+k_2}{m}}\right)}{\sqrt{2}}, \frac{c_1 \cos\left(\varphi_1 + t\sqrt{\frac{k_2}{m}}\right)}{\sqrt{2}} - \frac{c_2 \cos\left(\varphi_2 + t\sqrt{\frac{2k_1+k_2}{m}}\right)}{\sqrt{2}} \right)$$

gde su c_1, c_2 relativne amplitude normalnih moda. Biće izdvojene i nacrtane koordinate u specijalnim slučajevima, a prva normalna moda dobijena je izborom $c_1 = 1, c_2 = 0$.

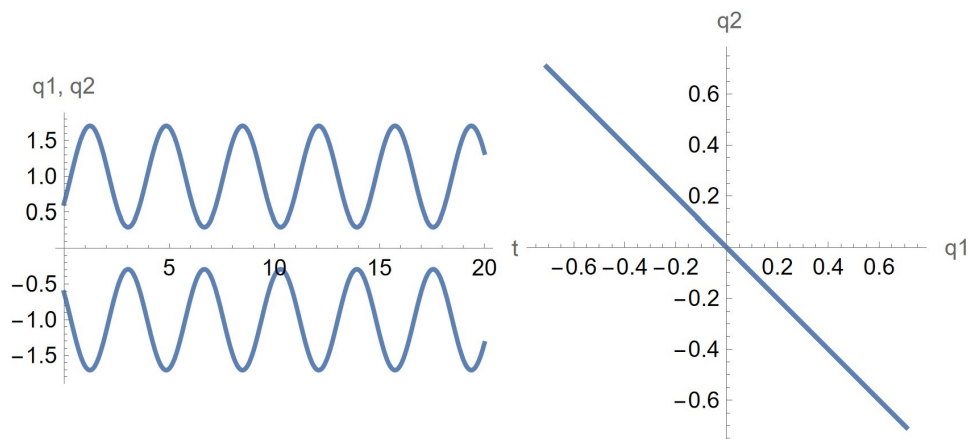
```
(*pravila zamene vrednosti konstanti*)
values1 = {c1->1,c2->0,k1->1,k2->1,m->1,fi1->1, fi2->1};
(*crtanje elongacija i faze*)
q12 = {q1[t], q2[t]} /. q12rule;
Plot[(q12 /. values1)+{-1, 1}, {t, 0, 20},
AxesLabel -> {"t", "q1, q2"}]
ParametricPlot[(q12 /. values1), {t, 0, 10},
AxesLabel -> {"q1", " q2"}]
```

Da bi se jasno uočile, elongacije oscilatora prikazane su međusobno rastavljene, tako je od prvog pomerena za -1 a drugog za $+1$ ($q12/.values1$) + $\{-1,1\}$.



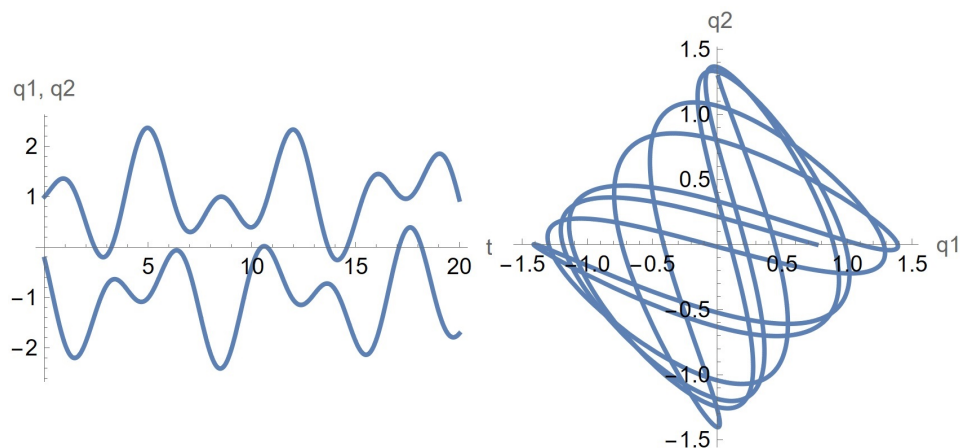
Slika 3.12: Prosto harmonijsko kretanje oscilatora u prvom modu oscilovanja koji su u fazi.

Sledeći primer je druga normalna moda dobijena izborom $c_1 = 0$, $c_2 = 1$, a grafici amplituda su realizovani slično kao u prethodnom primeru



Slika 3.13: Prosto harmonijsko kretanje oscilatora u drugom modu oscilovanja koji su u suprotnim fazama.

Karakteristika druge normalne mode je da u trenutku maksimuma jednog javlja se minimum drugog oscilatora i suprotno, odnosno u suprotnim su fazama. Mešavina (kombinacija) moda dobijena je izborom $c_1 = 1$, $c_2 = 1$.

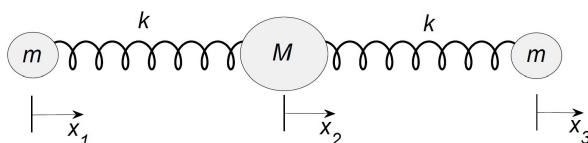


Slika 3.14: Prosto harmonijsko kretanje oscilatora kombinacije modova oscilovanja različitih frekvencija.

Položaj oscilatora tokom složenog oscilovanja, kombinacija modova različitih frekvencija, je složena kriva koja ispunjava prostor četverougla na slici 3.14.

3.5.43 Tri kuplovana oscilatora duž pravca.

Koristeći funkciju za traženje normalnih moda spregnutih oscilatora, biće analizirane vibracije tri tela međusobno povezana oprugama. Dva tela istih masa m , simetrično postavljena sleva i sdesna u odnosu na telo mase M . Sva tri tela postavljena su duž jednog pravca, vezana sa dve opruge istog koeficijenta elastičnosti k . Ako se oscilacije vrše samo na pravcu duž kog su položena tela, najpre će biti nađeni izrazi za kinetičku i potencijalnu energiju sistema, a potom svojstvene frekvencije sa odgovarajućim pomerajima (vektorima). Napisati opšte rešenje kao linearnu kombinaciju sve tri normalne mode sistema a neke slučajeve animirati. Model koji se opisuje u ovom primeru ekvivalentan je linearnom troatomskom molekulu poput ugljen-dioksida, gde su dva identična molekula kiseonika (mase m) vezana identičnim oprugama konstante elastičnosti k za jedan atom ugljenika mase M .



Slika 3.15: Klasični analogon linearnog troatomskog molekula, sličnog ugljen-dioksidu.

Neka su $x_1[t], x_2[t], x_3[t]$ odstupanja čestica prikazanih na slici 3.15 od ravnotežnog položaja. Tada su kinetička i potencijalna energija date sa

$$T = \frac{1}{2}m(x_1'[t]^2 + x_2'[t]^2) + \frac{1}{2}Mx_2'[t]^2, \quad U = \frac{1}{2}k((x_2[t] - x_1[t])^2 + (x_3[t] - x_2[t])^2). \quad (3.16)$$

```
(*koordinate koje opisuju vibracije u zavisnosti od t*)
coord = {x1[t], x2[t], x3[t]};
(*kineticka (T) i potencijalna energija (U)*)
T = (1/2) m (x1'[t]^2 + x3'[t]^2) + (1/2) M*x2'[t]^2;
U = (1/2) k ((x2[t] - x1[t])^2 + (x2[t] - x3[t])^2);
(*ravnotezni položaj*)
equilb = {0, 0, 0};
(*traženje svojstvenih frekvencija i vektora*)
eignwVec = SmallOscill[T, U, coord, equilb]
```

Promenljiva *eignwVec* sadrži frekvencije

$$\left\{ 0, \sqrt{\frac{k}{m}}, \sqrt{\frac{k(2m+M)}{mM}} \right\}$$

i njima odgovarajuće vektore moda zadanog sistema

$$X = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ -\frac{M}{\sqrt{2}\sqrt{2m^2+M^2}} & \frac{\sqrt{2}m}{\sqrt{2m^2+M^2}} & -\frac{M}{\sqrt{2}\sqrt{2m^2+M^2}} \end{pmatrix}$$

zadate redom kao vrste matrice. Opšte rešenje je linearna kombinacija normalnih moda

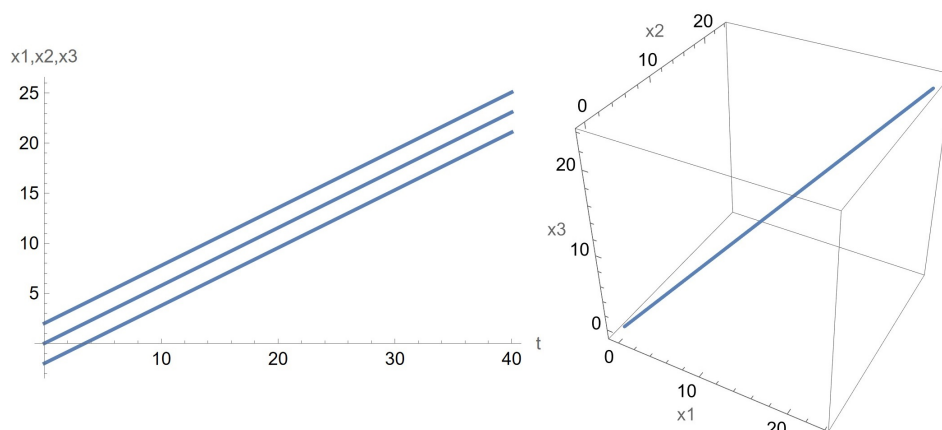
$$(c_1 t + \varphi_1)X[[1]] + c_2 \cos(\omega_2 t + \varphi_2)X[[2]] + c_3 \cos(\omega_3 t + \varphi_3)X[[3]].$$

Programski, opšte rešenje dobijeno je množenjem svojstvenih vektora odgovarajućim funkcijama $x_{123} = \text{Total}\{c_1 * t + f_{i1}, c_2 * \text{Cos}[w_2 * t + f_{i2}], c_3 * \text{Cos}[w_3 * t + f_{i3}]\} \text{eignwVec}[[2]]$; koje u sebi imaju kontribucije pojedinih moda c_1, c_2, c_3 . Prva moda, nulte frekvencije, je čista translacija dok preostale dve, nenultih frekvencija imaju harmonijske pomeraje. Kako bi se istakle sličnosti i razlike u kretanju na istom grafiku biće prikazani pomeraji čestica tokom vremena, pri čemu prva i treća čestica su pomerene duž y -ose redom za -2 i 2, $(x_{123}/.values) + \{-2, 0, 2\}$ da bi se postigla što bolja preglednost prikazanih funkcija elongacija od vremena.

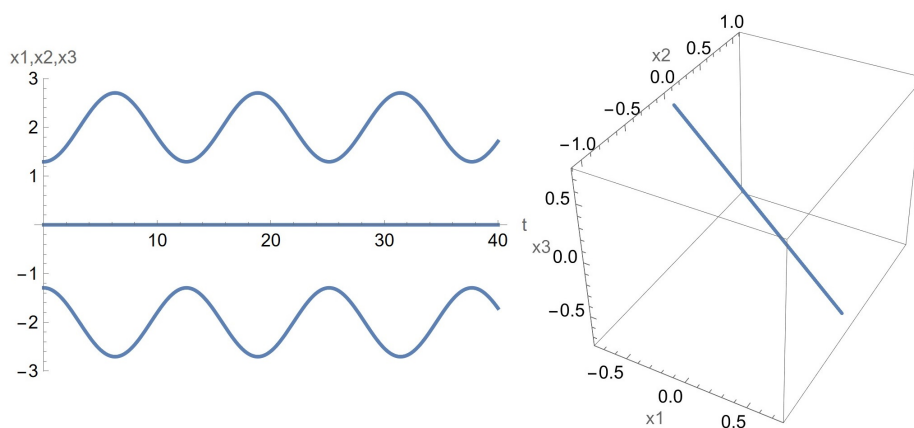
```
(*iz resenja izdvojene frekvencije*)
{w1, w2, w3} = eignwVec[[1]];
(*priprema opsteg resenja iz svojstvenih vektora moda*)
x123=
Total[{c1*t+f_{i1}, c2*Cos[w2*t+f_{i2}], c3*Cos[w3*t + f_{i3}]} eignwVec[[2]]];
(*izbor doprinosa moda preko c1, c2 i c3 i ostalih konstanti*)
values={c1->0, c2 ->1, c3->1, k->1, m->4, M->3, f_{i1}->0, f_{i2}->0, f_{i3}->0};
(*crtanje elongacije cestica oscilatora*)
Plot[(x123 /. values) + {-2, 0, 2}, {t, 0, 40},
AxesLabel -> {"t", "x1,x2,x3"}]
(*crtanje *)
ParametricPlot3D[x123 /. values, {t, 0, 40},
AxesLabel -> {x1, x2, x3}]
```

Trivijalna, prva normalna moda frekvencije o ilustrovana je zajedno sa funkcijama položaja čestica tokom ovog kretanja na slici 3.16. Sve čestice kreću se u istom smeru i u istoj fazi ne deformišući opruge, što odgovara čistoj translaciji sistema tri tela.

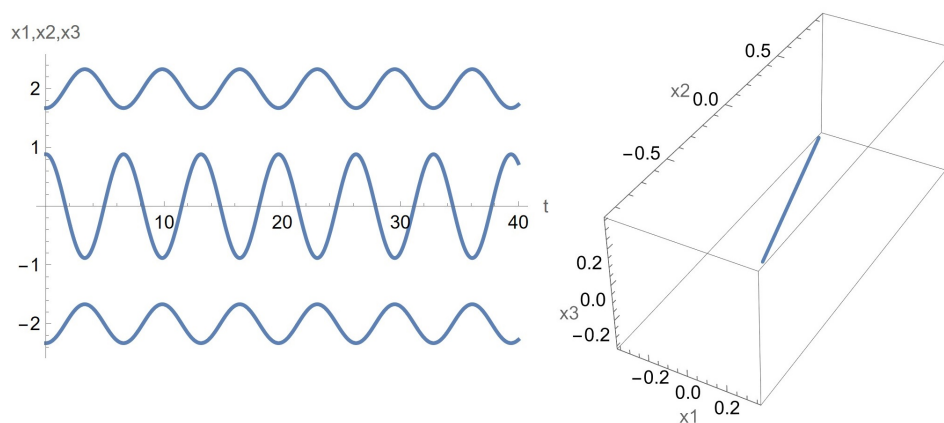
Sledi prikaz pomeraja čestica koji odgovaraju oscilacijama druge mode zadate sa $c_1 = 0, c_2 = 1$ i $c_3 = 0$. Na grafiku 3.17 vidi se da su dve identične čestice 1 i 3 suprotnih faza, dok 2 miruje u odnosu na susedne. Pomeraji tela treće normalne mode, dobijene izborom $c_1 = 0, c_2 = 0, c_3 = 1$,



Slika 3.16: Elongacije (levo) i prikaz parametarske jednačine prve mode $c_1 = 1, c_2 = 0, c_3 = 0$.



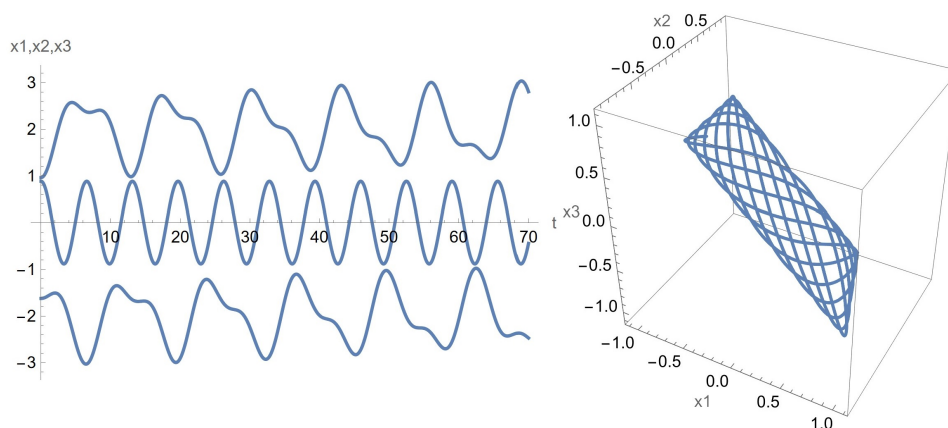
Slika 3.17: Elongacije (levo) i prikaz parametarske jednačine druge mode $c_1 = 0, c_2 = 1, c_3 = 0$.



Slika 3.18: Elongacije (levo) i prikaz parametarske jednačine treće mode $c_1 = 0, c_2 = 0, c_3 = 1$.

prikazani su na 3.18. Čestice 1 i 3, istih masa osciluju u fazi, dok lakša čestica 2 (ugljenik) ima suprotnu fazu u odnosu na njih i veću amplitudu. Mešavina moda prikazana na slici 3.19, zadata sa nultim doprinosom translacije $c_1 = 0$ i podjednakim doprinosom ostalih moda

$c_1 = c_2 = 1$. Animacija vibracija molekula CO_2 za mešavinu 2. i 3. mode pri $c_2 = c_3 = 1$ realizo-

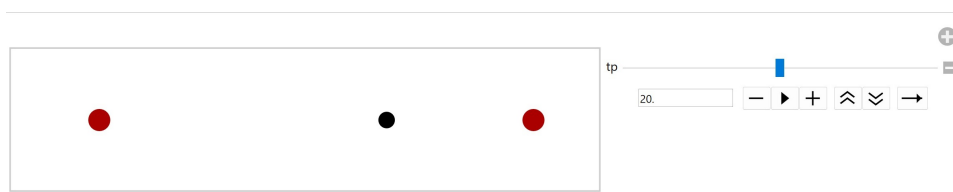


Slika 3.19: Elongacije (levo) i prikaz parametarske jednačine kombinacije druge i treće mode $c_1 = 0, c_2 = 1, c_3 = 1$.

vana je pomeranjem duž x -ose dva crvena kružića i crnog između njih, koji redom predstavljaju atome O i C. Programski, animacija je realizovana najpre zadavanjem zavisnost položaja centara čestica od vremena

$xt = (x123 /. values) + \{-3, 0, 3\}$, a potom su dobijene parametarske funkcije koordinata upotrebene kao centri tačaka (crvenih za kiseonik i crne za ugljenik) prikazanih u animaciji kreiranoj preko *Manipulate*.

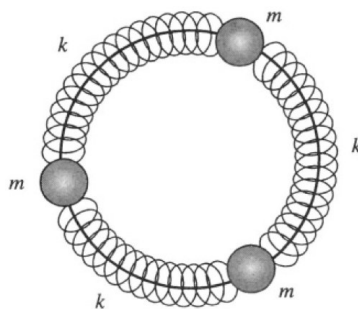
```
xt = (x123 /. values) + {-3, 0, 3};
Manipulate[
Graphics[{Darker[Red], PointSize[0.04], Point[{xt[[1]], 0}], Black,
  PointSize[0.03], Point[{xt[[2]], 0}], Darker[Red],
  PointSize[0.04], Point[{xt[[3]], 0}]} /. t -> tp,
PlotRange -> {{-5, 4}, {-0.8, 0.8}}, {tp, 0, 40}]
```



Slika 3.20: Animacija oscilacija kombinacije modova $c_1 = 0$ i $c_2 = c_3 = 1$ molekula CO_2 .

3.5.44 Tri kuplovana oscilatora- oscilacije po krugu.

Tri kuglice masa $m_1 = m_2 = m$ i $m_3 = M$ vezane su istim elastičnim oprugama koje se mogu slobodno kretati po krugu. U ravnotežnom položaju tela formiraju temena jednakos-traničnog trougla. Potencijalna energija ovog sistema određena je istezanjem opruga, koja se mere dužinom kružnih lukova između susednih tela. Koristeći korisničku funkciju *SmallOscill* biće nađene frekvencije i normalni modovi malih oscilacija u ravni kruga. Potom će biti ispitane karakteristike moda u limesu $M \rightarrow m$ nakon čega će biti prikazane funkcije pomeraja tokom vremena. Na osnovu simetrije problema izabrane su polarne koordinate za rešavanje zadatka, pri čemu je koordinatni početak u centru kruga radijusa R . Uzajamni pomeraji tela opisani su ugaonim pomerajima $\theta_1[t], \theta_2[t]$ i $\theta_3[t]$ u odnosu na ravnotežne. Kinetička i potencijalna energija



Slika 3.21: Tri tačkasta oscilatora iste mase, vezani oprugom koja se kreću po krugu radijusa R .

date su izrazima

$$T = \frac{1}{2}R^2(m\theta_1'[t]^2 + m\theta_2'[t]^2 + M\theta_3'[t]^2), \quad U = \frac{1}{2}kR^2((\theta_1[t] - \theta_2[t])^2 + (\theta_1[t] - \theta_3[t])^2 + (\theta_3[t] - \theta_2[t])^2).$$

Koristeći korisničku funkciju *SmallOscill*

```
(*koordinate*)
coord = {\[Theta]1[t], \[Theta]2[t], \[Theta]3[t]};
(*kineticka i potencijalna energija tela*)
T = (1/2)R^2(m*\[Theta]1'[t]^2+m*\[Theta]2'[t]^2+M*\[Theta]3'[t]^2);
U = (1/2)*k*
R^2((\[Theta]2[t]-\[Theta]1[t])^2+(\[Theta]3[t]-\[Theta]2[t])^2
+(\[Theta]1[t]-\[Theta]3[t])^2);
(*ranvnotezni položaj*)
equilb = {0, 0, 0};
eignwVec = SmallOscill[T, U, coord, equilb]
```

dobijene su svojstvene frekvencije

$$\left\{ 0, \sqrt{\frac{3k}{m}}, \sqrt{\frac{k(2m+M)}{mM}} \right\}$$

i njima odgovarajući pomeraji oscilatora

$$\begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{M}{\sqrt{4m^2+2M^2}} & \frac{M}{\sqrt{4m^2+2M^2}} & -\frac{\sqrt{2}m}{\sqrt{2m^2+M^2}} \end{pmatrix}.$$

Nulta frekvencija odgovara čistim translacijama svih čestica u istom smeru, dok su ostale nenulte različite. U limesu $M \rightarrow m$, nalaženje svojstvenih frekvencija moda i njima odgovarajućih vektora pomeraja realizuje se na sledeći način

```
(*frekvencije*)
w123 = FullSimplify[Limit[eignwVec[[1]], M->m], m>0]
(*pomeraji koji odgovaraju
redom dobijenim frekvencijama*)
X123 = FullSimplify[Limit[eignwVec[[2]], M->m], m>0]
```

odakle se uz limes primeni i uprošćavanje, uz uslov da je masa veća od nule, nakon čega se dobiju ponovo jedna nulta i dve iste frekvencije

$$\left\{ 0, \sqrt{\frac{3k}{m}}, \sqrt{\frac{3k}{m}} \right\},$$

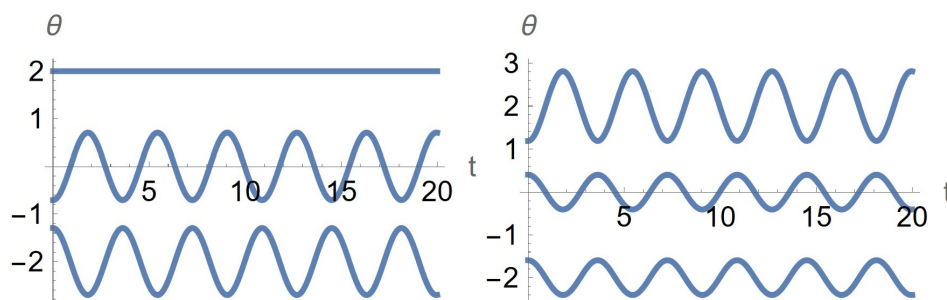
i njima redom odgovarajući pomeraji

$$\begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\sqrt{\frac{2}{3}} \end{pmatrix}.$$

Doprinosi moda c_1, c_2 i c_3 kretanju oscilatornom sistemu su kao i svojstvene frekvencije w_{123} zajedno sa pomerajima X_{123} promenljive korisničke funkcije $ThetaPlot[c_1, c_2, c_3, w_{123}, X_{123}]$ čiji je izlaz grafik zavisnosti pomeraja oscilatora tokom vremenskog intervala $t \in [0, 20]$.

```
(*pravila zamene velicina*)
values = {fi1 -> 0, fi2 -> 0, fi3 -> 0, k -> 1, R -> 1, m -> 1};
(*korisnicka funkcija za crtanje moda*)
ThetaPlot[c1_,c2_,c3_,w123_,X123_]:=Module[{w1, w2, w3, t},
  {w1, w2, w3} = w123;
  theta123 =Total[{c1*t+fi1,c2*Cos[w2*t fi2],c3*Cos[w3*t+fi3]}X123];
  Plot[(theta123 /. values) + {-2, 0, 2}, {t, 0, 20},
  AxesLabel -> {"t", "\[Theta]"}]];
(*crtanje netrivialnih moda*)
ThetaPlot[0, 1, 0, w123, X123]
ThetaPlot[0, 0, 1, w123, X123]
```

Aktivacijom drugog moda oscilovanja, jedno telo miruje dok druga dva harmonijski osciluju u suprotnoj fazi. Harmonijske oscilacije dva tela u istoj fazi i trećeg u suprotnoj karakteristika je trećeg moda slika 3.22.



Slika 3.22: Pomeraji druge (levo) i treće (desno) normalne mode tri jednaka tela na krugu.

3.6 Integracija

Rešavanje neodređenog integrala

$$\int f(x)dx$$

se u okviru simboličkog računa u programui Mathematica postavlja naredbom

$$Integrate[f, x].$$

Izlaz iz programa Mathematica ne uključuje arbitrarnu konstantu kao deo rešenja neodređenog integrala. Diferenciranjem dobijenog rešenja rezultuje integrandom.

3.6.45 Naći

$$\int \frac{x^4}{a^2 + x^2} dx,$$

a potom diferenciranjem rezultata vratiti se u integrand.

```
int=Integrate[(x^4)/(a^2+x^2), x]
```

Rešenje izraza $a^3 \tan^{-1}\left(\frac{x}{a}\right) - a^2x + \frac{x^3}{3}$ pridruženo je promenljivoj *ind*. Njenim diferenciranjem po *x*, a zatim uprošćavanjem dobije se integrand.

```
D[int,x]
FullSimplify[%]
```

$$\frac{\frac{a^2}{\frac{x^2}{a^2}+1} - a^2 + x^2}{\frac{x^4}{a^2+x^2}}$$

3.6.46 Izračunati neodređene integrale.

i) $\int \frac{4}{x^2+1} dx$

```
Integrate[4/(1+x^2), {x, 0, 1}]
```

Rešenje: π .

ii) $\int \frac{x}{a^3+x^3} dx$, a potom diferenciranjem rezultata proveriti da li se dobije integrand.

```
int=Integrate[x/(a^3+x^3), x]
Simplify[D[int, x]]
```

Rezultat integracije je: $\frac{\log(a^2-ax+x^2)-2\log(a+x)+2\sqrt{3}\tan^{-1}\left(\frac{2x-a}{\sqrt{3}a}\right)}{6a}$, a njegovim diferenciranjem, a potom i uprošćenjem izraza dobije se integrand $\frac{x}{a^3+x^3}$.

iii) Izračunati $\int \frac{2x^2-x+4}{x^3+4x} dx$, a potom diferencirati dobijeni rezultat.

```
int=Integrate[(2x^2-x+4)/(x^3+4x), x]
Simplify[D[int, x]]
```

Rezultat integracije je $\frac{1}{2} \log(x^2+4) + \log(x) - \frac{1}{2} \tan^{-1}\left(\frac{x}{2}\right)$, a njegovim diferenciranjem dobije se integrand $\frac{2x^2-x+4}{x^3+4x}$.

iv) $\int \sin^2(x) \cos(x) \log(\sin(x)) dx$

```
Integrate[Sin[x]^2Cos[x]Log[Sin[x]], x]
```

Rešenje: $\frac{1}{9} \sin^3(x)(3 \log(\sin(x)) - 1)$.

Umesto samo definisanja promenljive po kojoj se vrši integracija, dodavanjem argumenta u vidu intervala na kom je određena promenljiva $\{x, x_{min}, x_{max}\}$ formira se naredba

$$\text{Integrate}[f, \{x, x_{min}, x_{max}\}]$$

kojom se računa određen integral funkcije $f(x)$ na intervalu $[x_{min}, x_{max}]$

$$\int_{x_{min}}^{x_{max}} f(x) dx.$$

3.6.47 Izračunati $\int_0^1 \frac{1}{1+x+x^2} dx$

```
Integrate[1/(1 + x + x^2), {x, 0, 1}]
```

Rešenje: $\frac{\pi}{3\sqrt{3}}$.

Neretko, podintegralne funkcije mogu da sadrže parametre, od čijih vrednosti zavisi rešenje zadanog integrala. Kod naredbe konačnog integrala može se uključiti završni ili opcioni argument, oblika

Assumptions → *assumption*.

Struktura zadate pretpostavke može biti raznolika: jednačina, nejednakost, specifičan domen ili lista navedenih logičkih operacija. Za veličine koje se algebarski javljaju u nejednakostima uvek se smatra da su realne. Pored uslova zadanog kroz argument funkcije određenog integrala, uslovi se mogu zadati i sa naredbom *Assuming*.

3.6.48 Naći

$$\int_0^{\pi} \sin x^p dx.$$

Primenom naredbe bez uslova

```
Integrate[Sin[x]^p, {x, 0, Pi}]
```

dobije se uslovno rešenje

$$\text{ConditionalExpression} \left[\frac{\sqrt{\pi} \Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{p}{2} + 1\right)}, \text{Re}(p) > -1 \right]$$

u kom rezultat zavisi od parametra p , gde se kaže da je rešenje

$$\frac{\sqrt{\pi} \Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{p}{2} + 1\right)}$$

pod uslovom da je realni deo parametra p veći od -1. Upotrebom uslovnog argumenta

```
Integrate[Sin[x]^p, {x, 0, Pi}, Assumptions->Re[p]>-1]
```

dobija se rešenje $\frac{\sqrt{\pi} \Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{p}{2} + 1\right)}$.

3.6.49 Pod uslovom da su m i n celi brojevi i $m = n$ izračunati

$$\int_0^{\pi} \sin(mx) \sin(nx) dx.$$

Pod okruženjem *Assuming* kao prvi argument najpre se zadaju uslovi kojima su određeni parametri m i n , a potom se zadaje određeni integral. Uslovi mogu biti povezani u formulu logičkim operatorima ili samo nabrojani kao elementi liste.

```
Assuming[{Element[m, Integers] && Element[n, Integers] && m==n},
Integrate[Sin[n x] Sin[m x], {x, 0, Pi}]]
Assuming[{Element[m, Integers], Element[n, Integers], m==n},
Integrate[Sin[n x] Sin[m x], {x, 0, Pi}]]
```

Oba načina zadavanja uslova daju isti rezultat $\pi/2$.

3.6.50 Izračunati integrale pod zadatim uslovima.

i) $\int_{-\infty}^{\infty} x^2 e^{-2ax^2} dx$ ako je a realna pozitivna konstanta.

```
Integrate[x^2*Exp[-2*a*x^2],{x,-Infinity,Infinity},Assumptions->(a>0)]
```

Rešenje: $\frac{\sqrt{\pi}}{4a^{3/2}}$.

ii) $\int_0^{\infty} \frac{\cos^2(mx)}{a^2+x^2} dx$ ako su $a > 0$ i $m \geq 0$.

```
Integrate[Cos[m*x]^2/(a^2+x^2),{x,0,Infinity},
Assumptions->(a>0)&&(m>0)]
```

Rešenje: $\frac{\pi(e^{-2am}+1)}{4a}$.

iii) $\int_0^{\pi} x \sin^p(x) dx$ ako je $p > -1$.

```
Integrate[x*Sin[x]^p,{x,0,Pi},Assumptions->(p>-1)]
```

Rešenje: $\frac{\pi^{3/2}\Gamma(\frac{p+1}{2})}{2\Gamma(\frac{p}{2}+1)}$.

Višestruki integral

$$\int_{x_{min}}^{x_{max}} dx \int_{y_{min}}^{y_{max}} dy f(x, y)$$

nalazi se primenom iste naredbe kao i jednostruki, samo su kao argumenti dodate definisane promenljive i njihovi intervali

$$\text{Integrate}[f, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\}].$$

3.6.51 Naći

$$\int_0^1 dx \int_0^{2-2x} dy (x^2 + y^2 + 1).$$

```
Integrate[(x^2+y^2+1),{x,0,1},{y,0,2-2x}]
```

Rešenje: 11/6.

3.6.52 Maksvelova distribucija po brzinama molekula idealnog gasa na temperaturi T je

$$f(v) = 4\pi \left(\frac{m}{2\pi kT} \right)^{3/2} v^2 e^{-\frac{mv^2}{2kT}}$$

gde je m masa molekula a k Bolcmanova konstanta. Verovatnoća da molekul ima brzinu u opsegu $[v, v + dv]$ jednaka je $f(v)dv$. Koristeći definiciju Maksvelove raspodele simbolički izračunati tipične brzine: srednju i srednju kvadratnu brzinu.

i) Srednja ili očekivana brzina molekula definisana je kao

$$\langle v \rangle = \int_0^{\infty} v f(v) dv.$$

```
(*Maksvelova distribucija:funkcija od v*)
f[v_]:=4Pi(m/(2 Pi k T))^(3/2) v^2 Exp[-m v^2/(2 k T)];
Integrate[f[v]v,{v,0,Infinity},Assumptions->(m>0)&&(k T) > 0]
```

Rešenje: $\sqrt{3}\sqrt{\frac{kT}{m}}$.

ii) Koren srednje kvadratne brzine definiše se kao

$$\sqrt{\langle v^2 \rangle} = \left(\int_0^\infty v^2 f(v) dv \right)^{1/2}.$$

(*srednja kvadratna brzina molekula*)
`Sqrt[Integrate[f[v]v^2,{v,0,Infinity},Assumptions->m/(k T) > 0]]`

Rešenje: $2\sqrt{\frac{2}{\pi}}\sqrt{\frac{kT}{m}}$.

3.6.53 Funkcija spektralne radijance zračenja crnog tela u zavisnosti od apsolutne temperature T i talasne dužine λ data je kao

$$R(\lambda, T) = \frac{2\pi hc^2}{\lambda^5} \frac{1}{e^{hc/(\lambda kT)} - 1},$$

gde su k Bolcmanova konstanta, c brzina svetlosti a h Plankova konstanta. Iz R će biti izvedena Štefan-Bolcmanov zakon, koji opisuje intenziteta zračenja u zavisnosti od temperature

$$R(T) = \sigma T^4.$$

Veličina $R(\lambda, T)d\lambda$ je definisana kao energija izračena u intervalu talasnih dužina između λ i $\lambda + d\lambda$ po jedinici vremena kroz jedinicu površine. Tako, intenzitet zračenja $R(T)$, je ukupna energija na svim talasnim dužinama izračena po jedinici vremena kroz jediničnu površinu

$$R(T) = \int_0^\infty R(\lambda, T)d\lambda.$$

Prilikom rešavanja određenog integrala potrebno je zadati uslov da je rezultat ch/kT pozitivna veličina.

`Integrate[(2 Pi h c^2/\[Lambda]^5) (1/(Exp[h c/(\[Lambda] k T)] -1)),
 {\[Lambda], 0, Infinity}, Assumptions -> ((c h)/(k T)) > 0]`

Rešenje integrala je

$$R(T) = \frac{2\pi^5 k^4 T^4}{15c^2 h^3},$$

odakle se nalazi Štefan-Bolcmanova konstanta

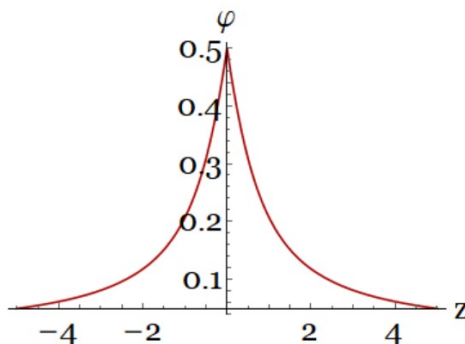
$$\sigma = \frac{2\pi^5 k^4}{15c^2 h^3}.$$

3.6.54 **Električno polje i potencijal duž ose ravnomerno naelektrisanog diska.**

Simbolički naći izraz za potencijal $\varphi(z)$ i električno polje $E(z)$ u zavisnosti od položaja duž z ose ravnomerno površinski naelektrisanog diska. Radijus diska, naelektrisanog površinskom gustinom σ je R . Osa z je okomita na disk i prolazi kroz njegov centar. Električni potencijal dat je kao

$$\varphi(z) = \frac{\sigma}{2\epsilon_0} \int_0^R \frac{r}{\sqrt{r^2 + z^2}} dr$$

`\[CurlyPhi]=\[Sigma]/(2\[Epsilon]0)Integrate[r/Sqrt[r^2+z^2],{r,0,R},
 Assumptions->R>0&&z>0];
 \[CurlyPhi]=\[CurlyPhi]/.z2->z^2];
 \[CurlyPhi]=FullSimplify[\[CurlyPhi]];
 Plot[\[CurlyPhi]/.\[Sigma]->1,\[Epsilon]0->1,R->1},{z,-5,5},
 AxesStyle->Directive[Black, 24],AxesLabel->{"z", "\[CurlyPhi]"}]`



Električno vektorsko polje je potencijalno i može da se izrazi kao gradijent skalarnog potencijala

$$\vec{E} = -\nabla\varphi,$$

odakle se može naći polje duž z ose u datom primeru, $E(z) = -\partial\varphi/\partial z$.

```
Ez=-D[\[CurlyPhi],z]
g1=Plot[Ez/.{\[Sigma]->1,\[Epsilon]0->1,R->1},{z,-5,5},
PlotRange->All];
EzLmt=Limit[Ez,R->Infinity];
g2=Plot[EzLmt/.{\[Sigma]->1,\[Epsilon]0->1,R->1},{z,-5,5},
PlotStyle->Blue];
Show[g1,g2,
AxesStyle->Directive[Black,24],
AxesLabel->{"z","E"}]
```

Naredbom $-D[\varphi, z]$ izračunata je z komponenta električnog polja u zavisnosti od položaja duž z ose

$$E(z) = -\frac{\sigma \left(\frac{z}{\sqrt{R^2+z^2}} - \frac{z}{\sqrt{z^2}} \right)}{2\epsilon_0},$$

što je na slici 3.25 predstavljeno tamnocrvenom linijom. U limesu kada $R \rightarrow \infty$ dobija se dobro poznat izraz za jačinu električnog polja iznad i ispod beskonačne, ravnomerno površinski naelektrisane ravni

$$\lim_{R \rightarrow \infty} E(z) = \frac{\sigma \operatorname{sgn}(z)}{2\epsilon_0},$$

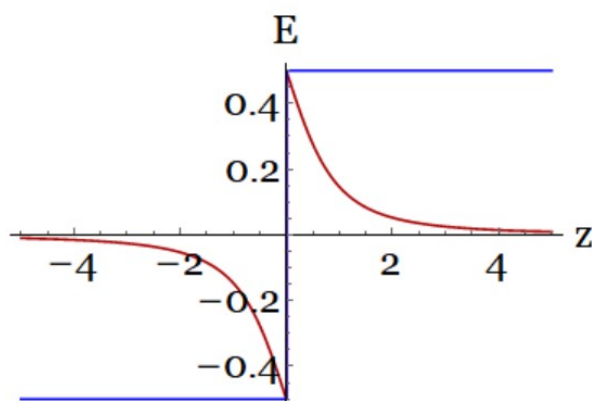
čija je zavisnost na slici 3.25 prikazana punom plavom linijom. Vidi se da intenzitet polja u okolini beskonačne ravnomerno površinski naelektrisane ravni ne zavisi od rastojanja od nje. Intenzitet polja iznad i ispod razlikuje se do na predznak. U zavisnosti od vrste naelektrisanja, vektor polja orijentisan je sa obe strane od ravni ukoliko je pozitivno ili ka njoj ukoliko je naelektrisanje negativno.

3.7 Rešavanje diferencijalnih jednačina

Naredbom

$$DSolve[eqn, y[x], x]$$

simbolički se rešava diferencijalna jednačina eqn za funkciju $y[x]$, nezavisno promenljive x . Ovom naredbom moguće je rešiti sistem diferencijalnih jednačina $\{eqn_1, eqn_2, \dots\}$ za funkcije $y_1[x_1, x_2, \dots], y_2[x_1, x_2, \dots], \dots$ nezavisnih varijabli x_1, x_2, \dots . Pored diferencijalnih jednačina, lista



Slika 3.23: Električno polje duž ose z koja prolazi kroz centar ravnomerno površinski naelektrisanog diska i normalna je na njega.

prvog argumenta može da sadrži i granične uslove, čime se smanjuje broj proizvoljnih konstanti u rešenju. Rešavanje diferencijalnih jednačina u programu Mathematica biće ilustrovano rešavanjem zadatka:

$$\frac{d^2y}{dx^2} - 3\frac{dy}{dx} - 18y = xe^{4x}.$$

```
DSolve[y''[x]-3y'[x]-18y[x]==x*Exp[4x], y[x], x]
```

Izvršenjem zadate linije dobije se opšte rešenje diferencijalne jednačine oblika

$\left\{ \left\{ y(x) \rightarrow c_1 e^{-3x} + c_2 e^{6x} - \frac{1}{196} e^{4x} (14x + 5) \right\} \right\}$, gde su $c[1], c[2]$ proizvoljne konstante.

Ista naredba koristi se za simboličko rešavanje sistema $\{eqn_1, eqn_2, \dots\}$ diferencijalnih jednačina, postavkom na sledeći način

```
DSolve[{eqn1, eqn2, ...}, {y1[x1, x2, ...], ...}, {x1, x2, ...}]
```

pri čemu se traže funkcije $y_1[x_1, x_2, \dots], y_2[x_1, x_2, \dots], \dots$ nezavisno promenljivih x_1, x_2, \dots . Umesto jednačina lista prvog argumenta može da sadrži granične uslove, čime se umanjuje opštost rešenja. Rešiti diferencijalnu jednačinu

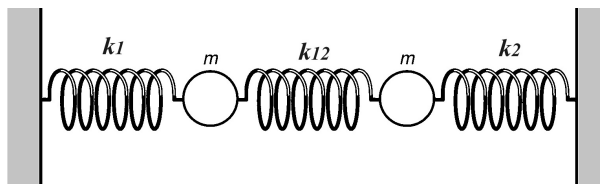
$$\frac{d^2y}{dx^2} + 8\frac{dy}{dx} + 16y = 0$$

uz uslove $y(0) = 2$ i $y'(0) = 1$.

```
eq={y''[x]+8y'[x]+16y[x]==0};
init={y[0]==2, y'[0]==1};
eqs=Union[eq, init];
DSolve[eqs, y[x], x]
```

Rešenje: $\left\{ \left\{ y(x) \rightarrow e^{-4x} (9x + 2) \right\} \right\}$.

Primena naredbe *DSolve* na rešavanje zadatka iz fizike biće ilustrovano na primeru iz klasične mehanike. Dva prosta harmonijska oscilatora spregnuta su međusobno i sa zidovima oprugama kao na slici 3.24. Telo mase m_1 se nalazi u ravnotežnom položaju, a m_2 je od ravnotežnog položaja pomeren za a . Ako su $k_1 = k_2 = k_{12} = k$ i $m_1 = m_2$ naći zavisnost položaja tela od vremena.



Slika 3.24: Dva spregnuta oscilatora.

Neka su $x_1(t)$ i $x_2(t)$ pomeraji tela m_1 i m_2 redom, merena od ravnotežnog položaja, pri čemu je usmerenje udesno pozitivno. Jednačine kretanja tela su

$$m_1 a_1 = -kx_1 - k(x_1 - x_2)$$

$$m_2 a_2 = -kx_2 - k(x_2 - x_1).$$

Diferencijalne jednačine kretanja oscilatora su

$$\begin{aligned} m \frac{d^2 x_1}{dt^2} + 2kx_1 - kx_2 &= 0 \\ m \frac{d^2 x_2}{dt^2} + 2kx_2 - kx_1 &= 0, \end{aligned} \quad (3.17)$$

pri definisanim početnim uslovima. U trenutku $t = 0$ oba tela miruju

$$v_1(0) = x_1'(0) = 0, \quad v_2(0) = x_2'(0) = 0, \quad (3.18)$$

a njihovi pomeraji u odnosu na ravnotežni položaj prema uslovima zadatka iznose

$$x_1(0) = 0, \quad x_2(0) = a. \quad (3.19)$$

Sistem diferencijalnih jednačina (3.17) zajedno sa početnim uslovima za brzinu (3.18) i položaj (3.19) su elementi liste prvog argumenta naredbe *DSolve*, dok su tražene funkcije $\{x_1[t], x_2[t]\}$. Deljenjem jednačina (3.17) sa m i zamenom sa $\omega^2 = k/m$ dobije se sistem praktičniji za programsko rešavanje

$$\frac{d^2 x_1}{dt^2} + 2\omega^2 x_1 - \omega^2 x_2 = 0, \quad \frac{d^2 x_2}{dt^2} + 2\omega^2 x_2 - \omega^2 x_1 = 0.$$

```
(*2 kuplovana oscilatora*)
(*sistem diferencijalnih jednacina*)
eqs={x1''[t]+2 w^2*x1[t]-w^2*x2[t]==0,
x2''[t]+2 w^2*x2[t]-w^2*x1[t]==0};
(*pocetni uslovi*)
init={x1'[0]==0,x2'[0]==0,x1[0]==0,x2[0]==a};
(*prvi argument DSolve: objedinjene jednacine i pocetni uslovi*)
eqs=Union[eqs,init];
sol=DSolve[eqs,{x1[t],x2[t]},t];
ExpToTrig[FullSimplify[sol]]
```

Rezultat je lista transformacija

$$\left\{ \left\{ x_1(t) \rightarrow \frac{1}{2}a \cos(tw) - \frac{1}{2}a \cos(\sqrt{3}tw), x_2(t) \rightarrow \frac{1}{2}a \cos(tw) + \frac{1}{2}a \cos(\sqrt{3}tw) \right\} \right\},$$

čijom zamenom se nalaze parametarske funkcije položaja.

3.7.55 Rešiti diferencijalne jednačine koristeći naredbu *DSolve*.

i) $\frac{d^2y}{dx^2} - 6\frac{dy}{dx} + 13y = e^x \cos x$

```
sol=DSolve[y''[x]-6y'[x]+13y[x]==Exp[x]Cos[x],y[x],x];
sol=sol[[1]];
FullSimplify[sol]
```

Rešenje: $\left\{ y(x) \rightarrow e^{3x} (c_1 \sin(2x) + c_2 \cos(2x)) + \frac{1}{65} e^x (7 \cos(x) - 4 \sin(x)) \right\}$.

ii) $y'' - 2y' = \sin(4x)$

```
DSolve[y''[x]-2y'[x]==Sin[4x],y[x],x]
```

Rešenje: $\left\{ \left\{ y(x) \rightarrow \frac{1}{2} c_1 e^{2x} + c_2 - \frac{1}{20} \sin(4x) + \frac{1}{40} \cos(4x) \right\} \right\}$.

iii) $y'' + y = e^x + x^3$ uz početne uslove $y(0) = 2$ i $y'(0) = 0$.

```
eq={y''[x]+y[x]==Exp[x]+x^3};
init={y[0]==2,y'[0]==0};
eq=Union[eq,init];
sol=DSolve[eq,y[x],x];
sol=FullSimplify[sol[[1]]]
```

Rešenje: $\left\{ y(x) \rightarrow \frac{1}{2} (2x(x^2 - 6) + e^x + 11 \sin(x) + 3 \cos(x)) \right\}$.

3.7.56 Prigušeni harmonijski oscilator.

Odrediti položaj prigušenog harmonijskog oscilatora u zavisnosti od vremena. Sile koje deluju na oscilator su elastična $-kx$ i sila prigušenja, koja deluje u smeru suprotnom od smera kretanja i pri malim brzinama njen intenzitet iznosi $-bv$. Pri tome k je konstanta elastičnosti opruge, x odstupanje od ravnotežnog položaja, $v = \dot{x}$ brzina oscilatora, a b je konstanta prigušenja. Jednačina kretanja ovog oscilatora je

$$m\ddot{x} = -bv - kx. \quad (3.20)$$

Napisana preko izvoda $a = \ddot{x}$ deljenjem jednačine (3.20) sa masom oscilatora m ona postaje postaje

$$\frac{d^2x}{dt^2} + \beta \frac{dx}{dt} + \omega_0^2 x = 0,$$

gde je $\beta = b/m$, a $\omega_0 = \sqrt{k/m}$ je sopstvena frekvencija oscilatora.

```
(*priguseni oscilator*)
eq=x''[t]+\[Beta]*x'[t]+w^2*x[t]==0;
sol=DSolve[eq, x[t], t];
sol=sol[[1]]
```

Rezultat je lista koja sadrži transformaciju

$$\left\{ x(t) \rightarrow c_1 e^{\frac{1}{2}t(-\beta - \sqrt{\beta^2 - 4\omega_0^2})} + c_2 e^{\frac{1}{2}t(\sqrt{\beta^2 - 4\omega_0^2} - \beta)} \right\},$$

pravilo zamene iz kog se dobija položaj tela u zavisnosti od vremena određen do na dve konstante c_1 i c_2 . Zadavanjem početnih uslova oscilatora, položaja i brzine, konstante c_1 i c_2 su u potpunosti određene i dobija se jedinstvena funkcija elongacije u zavisnosti od vremena.

3.7-57 **Rešavanje Laplasove jednačine i separacija varijabli u Kartezijanskim koordinatama.** Mnogi problemi iz elektrostatičke uključuju granične površine pod određenim potencijalom ili specifične površinske gustine naelektrisanja. Rešenja ovih problema se mogu tražiti rešavanjem Laplasove jednačine. Rešenja Laplasove jednačine

$$\Delta f(x, y, z) = 0, \quad (3.21)$$

gde je $\Delta = \nabla^2$ Laplasov operator, mogu se izraziti kao proizvod tri člana, svaki funkcija samo jedne od koordinata, a rešenje postaje superpozicija ovih proizvoda. Koeficijenti u ekspanziji linearne kombinacije nalaze se iz graničnih uslova i osobina ortogonalnosti izdvojenih funkcija. U nastavku biće razmotrena separacija varijabli Laplasove jednačine u Kartezijanskim koordinatama. Pretpostavimo da se potencijal može izraziti u formi $V_x(x)V_y(y)V_z(z)$. Laplasova jednačina dobijena je primenom Laplasovog operatora na ovaj proizvod i izjednačavanjem sa nulom

$$\Delta V_x(x)V_y(y)V_z(z) = 0.$$

Biće izvršeno razdvajanje funkcija grupisanjem članova sa istom varijablom. Potom će svaki od dobijenih članova biti izjednačen separacionom konstantom, pri čemu se dobije obična diferencijalna funkcija drugog reda. Za svaku razdvojenju funkciju rešava se diferencijalna jednačina.

Jednačina potencijala sledi izjednačavanjem dejstva operatora Laplasijana na potencijal *pot* sa nulom. Najpre će biti nađeno dejstvo ovog diferencijalnog operatora, a potom razvijen njegov količnik sa potencijalom.

```
pot=Vx[x]*Vy[y]*Vz[z];
lapeq=Laplacian[pot, {x, y, z}]
eq1=Expand[lapeq/pot]
```

Izvršenjem napisanih linija dobija se

$$Vy[y]Vz[z]Vx''[x] + Vx[x]Vz[z]Vy''[y] + Vx[x]Vy[y]Vz''[z] \\ \frac{Vx''[x]}{Vx[x]} + \frac{Vy''[y]}{Vy[y]} + \frac{Vz''[z]}{Vz[z]}.$$

Poslednji izraz razdvojen je u listu članova koji zavise od samo jedne varijable

```
eq2=eq1/.{Plus->List}
```

$$\left\{ \frac{Vx''[x]}{Vx[x]}, \frac{Vy''[y]}{Vy[y]}, \frac{Vz''[z]}{Vz[z]} \right\}.$$

Odavde se formiraju tri diferencijalne jednačine, grupisanjem u grupe članova sa istom varijablom izjednačenih odgovarajućoj separacionoj konstanti

```
fieldEq=Thread[eq2=={-cx^2,-cy^2,cz^2}]
```

što daje

$$\left\{ \frac{Vx''[x]}{Vx[x]} == -cx^2, \frac{Vy''[y]}{Vy[y]} == -cy^2, \frac{Vz''[z]}{Vz[z]} == cz^2 \right\}, \text{ pri čemu za separacione konstante mora da važi uslov } cx^2 + cy^2 = cz^2 \text{ kako bi zadovoljile Laplasovu jednačinu.}$$

Rešavanje dobijenih diferencijalnih jednačina drugog reda vrši se na sledeći način

```
sol=DSolve[#[[1]],#[[2]],#[[3]]]&/@Thread[
{fieldEq,{Vx[x], Vy[y], Vz[z]}, {x, y, z}}];
rez={};
Do[
AppendTo[rez, sol[[i, 1, 1]]], {i, Length[sol]};
rez
```

Rešenja postavljene Laplasove jednačine je superpozicija proizvoda članova

$$V_x(x) \rightarrow C_1 \cos[c_x x] + C_2 \sin[c_x x],$$

$$V_y(y) \rightarrow C_1 \cos[c_y y] + C_2 \sin[c_y y],$$

$$V_z(z) \rightarrow e^{c_z z} C_1 + e^{-c_z z} C_2,$$

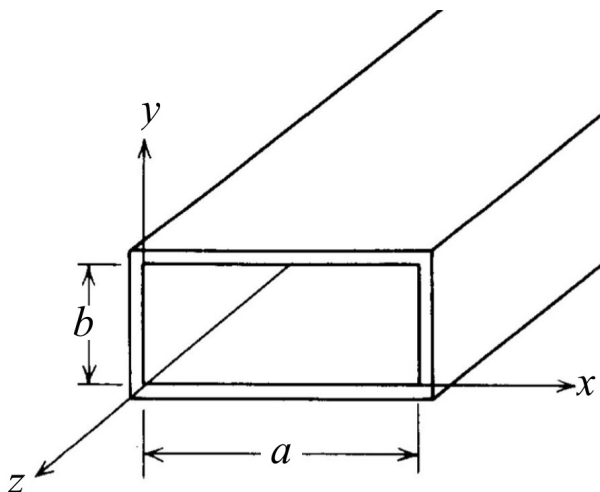
dok se nepoznate konstante nalaze iz graničnih uslova.

3.7.58 Potencijal pravougaonog provodnog talasovoda.

Razmatramo pravougaoni provodnik ograničen zidovima postavljenim u $x = 0, x = a, y = 0$ i $y = b$. Strana u $x = a$ je na potencijalu $V(a, y) = V_0$ a na strani u $x = 0$, potencijal zadovoljava uslov $V'(x) = 0$.

- Potencijal će biti izražen kao razvoj u red u Kartezijanskim koordinatama, pri čemu će biti određeni koeficijenti razvoja.
- Nakon nalaženja funkcije potencijala nacrtati je na zidu $x = a$, eksperimentišući zadržavanjem većeg broja članova u razvoju $m = 2, 4, 6, 8$, kao i trodimenzionalni grafik potencijala $V(x, y)$.
- Nacrtati ekvipotencijalne linije i električno polje u XY ravni.

Simetrija duž z -ose ukazuje na to da je rešenje nezavisno od z varijable. Prema tome, rešenje će u Kartezijanskim koordinatama biti proizvod funkcija $V_x(x)V_y(y)$. Granični uslovi zadati na zidovima zahtevaju da ovaj proizvod bude oblika $\cosh\left(\frac{m\pi x}{b}\right) \sin\left(\frac{m\pi y}{b}\right)$, gde je $m = 1, 2, 3, \dots$. Ortogonalnost funkcija $\sin\left(\frac{m\pi y}{b}\right)$ omogućava da se koeficijent u razvoju izrazi kao integral po granici na $x = a$.



Slika 3.25: Pravougaoni provodni talasovod.

Promenljivim *term* dodeljen je član i suma po njima *pot*

```
term=A[m] Cosh[m Pi x/b] Sin[m Pi y/b];
pot=Sum[term, {m, 1, n}]
```

odakle se dobije $\sum_{m=1}^n A(m) \cosh\left(\frac{\pi m x}{b}\right) \sin\left(\frac{\pi m y}{b}\right)$. Izvršiće se provera graničnog uslova $V'(x) = 0$ u $x = 0$. Nalazimo prvi izvod $D[pot, x]$, a potom se promenljiva x menja sa 0.

```
check=D[pot, x];
check/.{x->0}
```

Izvršenjem prethodnih linija koda dobija se 0, dok je vrednost potencijala na granici $x = a$ dobijena sa

```
pot/.{x->a}
```

što rezultuje $\sum_{m=1}^n A(m) \cosh\left(\frac{\pi am}{b}\right) \sin\left(\frac{\pi my}{b}\right)$.

Koeficijent $A(m)$ u razvoju se određuje iz uslova ortogonalnosti sinusne funkcije. Koristeći granični uslov u $x = a$ formira se jednačina $eq1$

```
eq1=V0==term/.{x->a}
```

$V_0 == A(m) \cosh\left(\frac{\pi am}{b}\right) \sin\left(\frac{\pi my}{b}\right)$.

Na desnoj strani jednačine $eq1$ je podrazumevana suma

$$V_0 = \sum_{m=1}^n A(m) \cosh\left(\frac{\pi am}{b}\right) \sin\left(\frac{\pi my}{b}\right),$$

odakle se $A(m)$ dobija množenjem obe strane jednačine $eq1$ sa $\sin\left(\frac{\pi ny}{b}\right)$, a potom integri. Kada se ovo primeni na desnoj strani rhs , zbog ortogonalnosti sinusne funkcije,

$$\int_0^b \sin\left(\frac{\pi ny}{b}\right) \sin\left(\frac{\pi my}{b}\right) dy = 0, \text{ za } n \neq m,$$

jedini član koji će preživeti je onaj sa $n = m$. Promenljiva rhs se definiše kao

```
rhs=Integrate[eq1[[2]] Sin[m Pi y/b], {y, 0, b}]/Simplify;
rhs=rhs/.{Sin[2m \[Pi]]->0}
```

odakle se dobije $\frac{1}{2}bA(m) \cosh\left(\frac{\pi am}{b}\right)$. Slično, primenom istog postupka na levu stranu jednačine $eq1$ postavlja se sa

```
lhs=Integrate[eq1[[1]] Sin[m Pi y/b], {y, 0, b}]/Simplify;
lhs=Simplify[lhs/.{Cos[m \[Pi]]->(-1)^m}]
```

što daje $-\frac{b((-1)^m - 1)V_0}{\pi m}$. Izjednačavanjem dobijenih promenljivih lhs i rhs formira se algebarska jednačina, iz koje se nalazi $A(m)$ primenom naredbe *Solve*. Zamenom

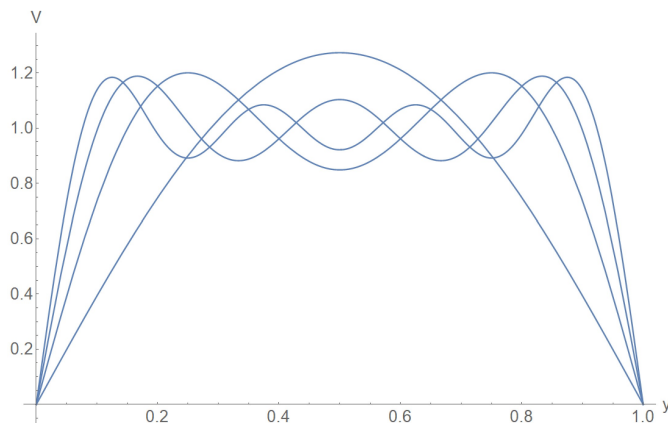
```
Asol=Solve[lhs==rhs, A[m]] [[1]];
c=A[m]/.Asol
Pot[n_] := Sum[c*Cosh[m Pi x/b] Sin[m Pi y/b], {m, 1, n}];
```

dobije se $A(m) = -\frac{2((-1)^m - 1)V_0}{\pi m} \operatorname{sech}\left(\frac{\pi am}{b}\right)$. Izraz ovog koeficijenta iskoristi se za kreiranje funkcije potencijala $Pot[n]$, u zavisnosti od stepena razvoja, reda n .

Doprinos ukupnom potencijalu može se videti koristeći različit broj članova n u ekspanziji potencijala. Kroz *Do* petlju brojačem je menjan broj članova i za svaki je nacrtan grafik. Nacrtani grafik potencijala za zadato n dodat je praznoj listi *grphs*, koji su nakon petlje svi zajedno prikazani.

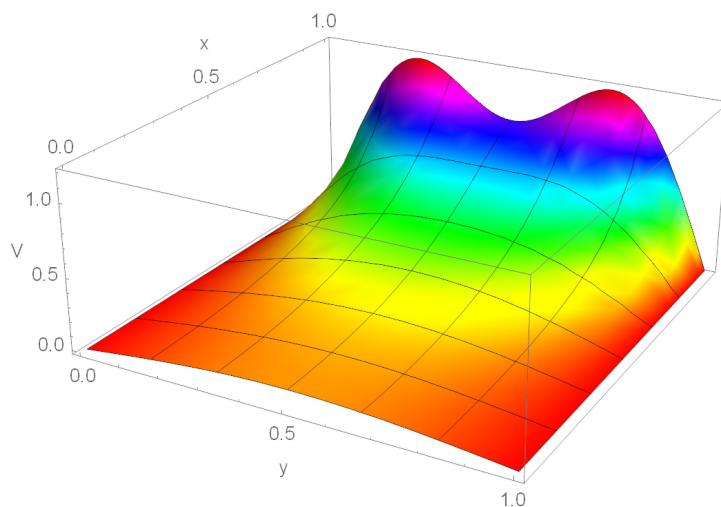
```
grphs={};
Do[
gr=Plot[Pot[n]/.{b->1, V0->1, x->a}, {y, 0, 1},
PlotRange -> All];
AppendTo[grphs, gr], {n, 2, 8, 2}];
Show[grphs, AxesLabel->{"y", "V"}]
```

Krive sa više tačaka uvijanja sadrže veći broj članova (n) i bolje aproksimiraju tačno rešenje. Za $n = 4$ nacrtana je funkcija potencijala $V(x, y)$.



Slika 3.26: Potencijal na granici $x = a$ za različit doprinos članova $m = 2, 4, 6, 8$.

```
Plot3D[Pot[4]/.{b->1,V0->1,a->1},{x,0,1},{y,0,1},
MeshStyle->Black,Mesh->5,
ColorFunction->Function[{x,y,z},Hue[z]],
AxesLabel->{"x","y","V"},
LabelStyle->Directive[18]]
```



Slika 3.27: Potencijal pravougaonog talasovoda u XY ravni za $n = 4$.

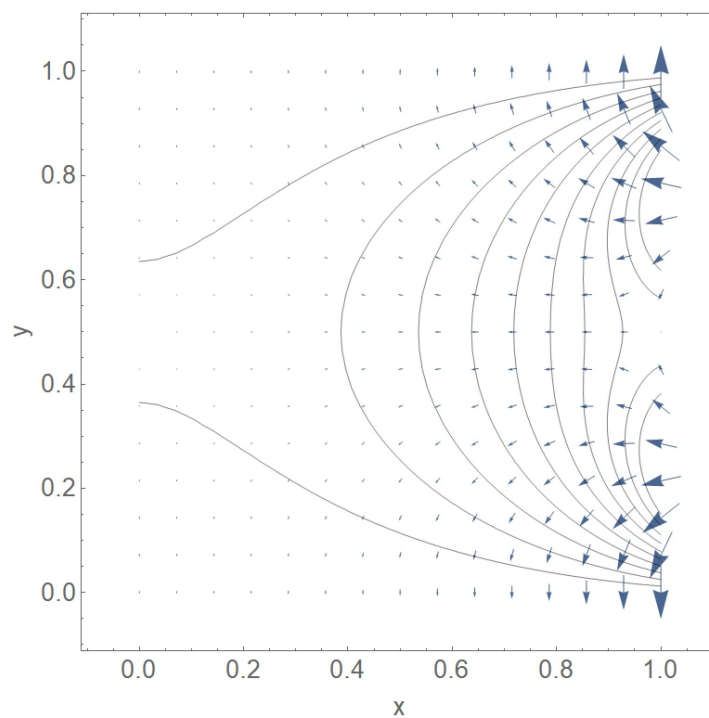
Veličinama V_0, a i b su pridružene vrednosti 1. Funkcijom *ContourPlot* nacrtane se izolinije potencijala u opsegu $x \in [0, 1], y \in [0, 1]$, a grafik pridružen promenljivoj *cont*. Vektorsko električno polje dobija se kao gradijent skalarnog potencijala prema relaciji $\vec{E} = -\nabla V(x, y)$. Da bi ovo bilo realizovano u programskom kodu, nađena je lista parcijalnih izvoda po Dekartovim koordinatama $D[\text{Pot}[4]/.exchange, \{x, y\}]$, koja reprezentuje funkcije komponenti električnog polja. Zamenom konstanti koristeći pravila data u *exchange*, dobije se funkcija vektorskog polja, nacrtana uz pomoć naredbe *VectorPlot* na opsegu kao potencijal i pridružen promenljivoj *vectf*.

```

exchange={b->1,V0->1,a->1};
vectf=VectorPlot[-D[Pot[4]/.exchange,{x,y}],{x,0,1},{y,0,1}];
cont=ContourPlot[Pot[4]/.exchange,{x,0,1},{y,0,1},
ContourShading->False];
Show[vectf,cont,FrameLabel->{"x","y"},LabelStyle->Directive[18]]

```

Vektorsko polje $vectf$ zajedno sa konturama skalarnog polja prikazano je na istom grafiku.



Slika 3.28: Ekvipotencijalne linije skalarnog potencijala i njemu odgovarajućeg električnog vektorskog polja po XY ravni pravougaonog provodnog talasovoda koristeći razvoj do $n = 4$.

3.7-59 **Let projektila.**

Projektil ispaljen kao kosi hitac pogađa metu, telo koje se pusti da pada slobodno sa visine h , na udaljenosti d duž X - ose od tačke lansiranja projektila. Hitac se ispaljuje iz koordinatnog početka XY ravni u trenutku kada meta počinje slobodno da pada. Vektor brzine kosog hica u početku je usmeren ka meti. Zadatak je da rešiti jednačine kretanja, da se nađu parametarske funkcije koordinata od vremena i vreme kretanja do sudara. Jednačine kretanja za kosi hitac ispaljen pod uglom α početnom brzinom v_0 je

$$m_1 \ddot{x}_1 = 0, \quad m_1 \ddot{y}_1 = -mg$$

zadata je kao varijabla $eq1$ sa početnim uslovima $x_1(0) = 0$, $\dot{x}_1(0) = v_0 \cos \alpha$, $y_1(0) = 0$, $\dot{y}_1(0) = v_0 \sin \alpha$ sadržanim u listi $incond1$. Njutnove jednačine kretanja mete (tela 2) sadržane u listi $eq2$ su

$$m_2 \ddot{x}_2 = 0, \quad m_2 \ddot{y}_2 = -mg,$$

dok su granični uslovi, sadržani u listi $incond2$, sledeći $x_2(0) = d$, $y_2(0) = h$, $\dot{x}_2(0) = 0$, $\dot{y}_2(0) = 0$. Unija diferencijalnih jednačina i početnih uslova, kao na primer za kosi hitac $Union[eq1, incond1]$ prvi je argument funkcije $DSolve$, koja se rešava za funkcije $\{x1[t], y1[t]\}$ po vremenu t . Nezavisno od projektila, postavljene su jednačine sa početnim uslovima i rešene za telo 2 (metu). Iz uslova da je vektor brzine projektila u početnom trenutku usmeren ka meti, te je shodno tome izvršena smena $\cos \alpha \rightarrow \frac{h}{\sqrt{h^2+d^2}}$, $\sin \alpha \rightarrow \frac{d}{\sqrt{h^2+d^2}}$.

```
(*diferencijalne jednacine za projektil*)
eq1={m1*x1''[t]==0,m1*y1''[t]==-m1*g};
(*diferencijalne jednacine za metu*)
eq2={m2*x2''[t]==0,m2*y2''[t]==-m2*g};
(*pocetni uslovi za projektil*)
incond1={x1[0]==0,x1'[0]==v0*Cos[\[Alpha]], y1[0]==0,
y1'[0]==v0*Sin[\[Alpha]]};
(*pocetni uslovi za metu*)
incond2={x2[0]==d, x2'[0]==0, y2[0]==h, y2'[0]==0};
(*resavanje diferencijalnih jednacina *)
sol1=DSolve[Union[eq1, incond1], {x1[t], y1[t]}, t][[1]];
sol2=DSolve[Union[eq2, incond2], {x2[t], y2[t]}, t][[1]];
xy1={x1[t], y1[t]} /. sol1;
xy2={x2[t], y2[t]} /. sol2;
(*iz uslova da je projektil usmeren ka meti*)
cond={Cos[\[Alpha]] -> d/Sqrt[d^2 + h^2],
Sin[\[Alpha]] -> h/Sqrt[d^2 + h^2]};
xy1=xy1/.cond;
(*uslov pogadjanja mete projektilom*)
Thread[FullSimplify[xy1==xy2]];
tflight=Solve[Thread[xy1 == xy2], t][[1]];
tflight=t /. tflight
```

Vektori položaja projektila i mete su redom

$$\vec{r}_1(t) = \left(\frac{dtv_0}{\sqrt{d^2+h^2}}, \frac{htv_0}{\sqrt{d^2+h^2}} - \frac{gt^2}{2} \right) \quad \vec{r}_2(t) = \left(d, h - \frac{gt^2}{2} \right).$$

Da bi bile ilustrovane trajektorije potrebno je zadati konkretne brojne vrednosti. Promenljiva *value* sadrži pravila zamene relevantnih veličina za potpunu analizu kretanja: visinu mete h i njeno rastojanje d od projektila u početnom trenutku, brzinu ispaljivanja hica v_0 i gravitaciono ubrzanje g . Iz datih podataka dobija se numerička vrednost vremena leta $tflight0$. Rešavanjem

algebarske jednačine $y_{1/2}(t) = 0$ računata su vremena leta projektila i mete iznad horizontalne podloge $t_{minflight}$. Uz pomoć naredbe *ParametricPlot* na intervalu $\{t, 0, t_{flight0}\}$ nacrtane su trajektorije oba tela čije su koordinate $\vec{r}_i(t) = \{x_i(t), y_i(t)\}$, $i = 1, 2$ parametarske funkcije vremena, date u $xy_{1/2}/value$. Dobijene trajektorije do sudara tela prikazane su na jednom grafiku *g12*. Projektil i meta zajedno sa vektorima brzina prikazani su u trenutku *tpt* (kraćem od $t_{flight0}$, u programu $tpt = t_{flight0}/2$). Najpre su dobijeni opšti izrazi vektora brzina tela $\vec{v}_i = d\vec{r}_i(t)/dt$ što je realizovano sa $v_i = D[xy_i, t]$, $i = 1, 2$, gde su potom zamenjene brojne vrednosti ulaznih parametara u trenutku crtanja traženih elemenata $v_i = (v_i.value)/\{t \rightarrow tpt\}$, $i = 1, 2$. Tela su ilustrovana kružićima radijusa 2 u koordinatama $coo_i = (xy_i.value)/\{t \rightarrow tpt\}$, $i = 1, 2$ pri tom svaki je nezavisno ilustrovan $bdy_i = Graphics[\{Darker[Green/Blue], Thickness[0.01], Circle[coo_i, 2]\}]$. Strelice vektora brzina počinju u koordinatama položaja tela, a vrhovi strelica određeni su dobijenim vektorima brzina $Arrow[\{coo_i, coo_i + v_i\}]$, $i = 1, 2$. Svi grafički objekti nacrtani su na istom grafiku, sa označenim koordinatnim osama i specifičnim opcijama za njihovu izmenu

```
Show[g12, bdy1, bdy2,
vects, AxesLabel → {"x", "y"},
LabelStyle → Directive[Black, Bold, 16],
AxesStyle → Thick].
```

```

(*brojne vrednosti velicina*)
value = {d -> 120, h -> 100, g -> 10, v0 -> 40};
(*vreme leta*)
tflight0 = 1. tflight /. value;
(*vreme leta projektila*)
tz01 = Solve[xy1[[2]] == 0, t];
tz01 = t /. tz01[[2]];
(*vreme leta mete*)
tz02 = Solve[xy2[[2]] == 0, t];
tz02 = t /. tz02[[2]];
(*najkrace vreme leta tela*)
tminflgt = Min[1. {tz01, tz02} /. value];
If[tminflgt < tflight0,
  Print["Pogodak se nece desiti iznad horizontalne ravni."]];
(*trajektorije kretanja tela projektila (1) i mete (2)*)
g1 = ParametricPlot[xy1 /. value, {t, 0, tflight0},
  PlotStyle -> Darker[Green]];
g2 = ParametricPlot[xy2 /. value, {t, 0, tflight0},
  PlotStyle -> Darker[Blue]];
g12 = Show[g1, g2, PlotRange -> All];
(*tpt-trenutak u kom se crtaju tela i vektori*)
tpt = tflight0/2;
(*=====*)
(*ctanje tela*)
bdy1 = Graphics[{Darker[Green], Thickness[0.01],
  Circle[coo1 = (xy1 /. value) /. {t -> tpt}, 2]};
bdy2 = Graphics[{Darker[Blue], Thickness[0.01],
  Circle[coo2 = (xy2 /. value) /. {t -> tpt}, 2]};
(*vektori brzina tela*)
v1 = D[xy1, t];
v2 = D[xy2, t];
v1 = (v1 /. value) /. {t -> tpt};
v2 = (v2 /. value) /. {t -> tpt};
vects = Graphics[{Darker[Green], Thick, Arrow[{coo1, coo1 + v1}],
  Darker[Blue], Arrow[{coo2, coo2 + v2}]}];
(*istovremeni prikaz svih grafickih elemenata*)
Show[g12, bdy1, bdy2,
  vects,
  AxesLabel -> {"x", "y"},
  LabelStyle -> Directive[Black, Bold, 16],
  AxesStyle -> Thick]

```

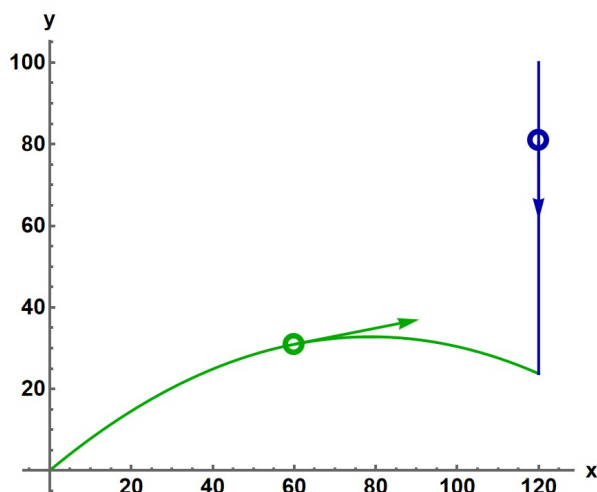
3.7.60 Kretanje naelektrisanе čestice u homogenom, stacionarnom magnetnom i električnom polju.

Razmatra se kretanje naelektrisanе čestice u vremenski nezavisnom, homogenom magnetnom i električnom polju. Najpre će biti postavljene jednačine kretanja iz Njutnovog zakona $\vec{F} = m\vec{a}$. Na naelektrisanu česticu u ovim poljima deluje Lorencova sila definisana kao

$$\vec{F} = q\vec{E} + q\vec{v} \times \vec{B},$$

gde je q naelektrisanje čestice, \vec{E} vektor električnog polja, \vec{B} vektor magnetnog polja usmerenog duž z - ose i \vec{v} vektor brzine čestice.

- i) Parametarske jednačine koordinata biće dobijene rešavanjem jednačina kretanja, uz pretpostavku da je čestica u početnom trenutku u koordinatnom početku $\{x, y, z\} = \{0, 0, 0\}$ i



Slika 3.29: Položaji projektila (zeleni kružić) i mete (plavi kružić) na trajektorijama pre kontakta.

imala brzinu \vec{v}_0 .

- ii) Dobijenu trajektoriju naelektrisane čestice u električnom i magnetnom polju biće ilustrovana, a potom animirano njeno kretanje, pri tom u svakoj tački putanje prikazivaće se vektor brzine.

Jednačine kretanja dobiju se iz komponenti vektorske diferencijalne jednačine

$$m\vec{a} = q\vec{E} + q\vec{v} \times \vec{B},$$

gde su $\vec{a} = \ddot{\vec{r}}$ i $\vec{v} = \dot{\vec{r}}$. Vektorska diferencijalna jednačina drugog reda rastavi se na sistem diferencijalnih jednačina po komponentama

$$mx''(t) = qB_z y'(t) + qE_x, \quad my''(t) = qE_y - qB_z x'(t), \quad mz''(t) = qE_z. \quad (3.22)$$

```
vE={Ex,Ey,Ez};(*vektor elektricnog polja*)
vB={0,0,Bz};(*vektor magnetnog polja*)
vr={x[t],y[t],z[t]};(*vektor poloazaja*)
vv={x'[t],y'[t],z'[t]};(*vektor brzine*)
av={x''[t],y''[t],z''[t]};
eq=q*vE+q*Cross[vv,vB]==m*av;(*II Njutnov zakon*)
eq=Thread[eq];
(*pocetni uslovi*)
initial={
x[0]==0,x'[0]==1,
y[0]==0,y'[0]==0,
z[0]==0,z'[0]==0.15};
eq=Union[eq,initial];
values={m->1,q->1,
Ex->0.5,Ey->0,Ez->-0.01,
Bz -> 5};
(*vrednosti parametara*)
sol=DSolve[eq /.values,vr,t];(*resavanje dif. jednacina*)
xyz=Chop[vr/.sol[[1]]];
```

Naredbom *DSolve* dobije se rešenje diferencijalne jednačine (3.22) gde su početni uslovi, položaj čestice u početnom trenutku $\vec{r}_0 = (x(0), y(0), z(0))^T = (0, 0, 0)^T$ i njena brzina $\vec{v}_0 = (x'(0), y'(0), z'(0))^T = (v_{x0}, v_{y0}, v_{z0})^T$. Nađene parametarske funkcije koordinata su

$$x(t) = \frac{-m(B_z v_{y0} + E_x) \cos\left(\frac{B_z q t}{m}\right) + m(B_z v_{x0} - E_y) \sin\left(\frac{B_z q t}{m}\right) + B_z E_y q t + B_z m v_{y0} + E_x m}{B_z^2 q},$$

$$y(t) = \frac{-B_z(E_x q t + m v_{x0}) + m(B_z v_{y0} + E_x) \sin\left(\frac{B_z q t}{m}\right) + m(B_z v_{x0} - E_y) \cos\left(\frac{B_z q t}{m}\right) + E_y m}{B_z^2 q},$$

$$z(t) = \frac{E_z q t^2}{2m} + t v_{z0}.$$

Jedinstvene parametarske funkcije, zavisnosti pozicije čestice od vremena, nalaze se zadavanjem početnih uslova, a u datom primeru položaj i brzina u početnom trenutku su $x(0) = 0$, $y(0) = 0$, $z(0) = 0$, $x'(0) = 1$, $y'(0) = 0$, $z'(0) = 0.15$, pri čemu su uzeti proizvoljni parametri.

Na osnovu zadatih uslova dobijene su parametarske funkcije položaja čestice u zavisnosti od vremena t

$$x(t) = 0.02 \left(1. \sin^2(5.t) + 10. \sin(5.t) + 1. \cos^2(5.t) - 1. \cos(5.t) \right),$$

$$y(t) = -0.1t + 0.02 \sin(5.t) + 0.2 \cos(5.t) - 0.2,$$

$$z(t) = 0.15t - 0.005t^2.$$

Vektor trenutne brzine v računa se kao prvi izvod položaja po vremenu $\vec{v}(t) = \dot{\vec{r}}(t)$, što je programski realizovano kao $v = D[xyz, t]$. Na osnovu gore zadatih početnih uslova i parametara dobijene komponente brzine u zavisnosti od vremena su sledeće parametarske funkcije:

$$v_x = 0.02(5. \sin(5.t) + 50. \cos(5.t)),$$

$$v_y = -1. \sin(5.t) + 0.1 \cos(5.t) - 0.1,$$

$$v_z = 0.15 - 0.01t.$$

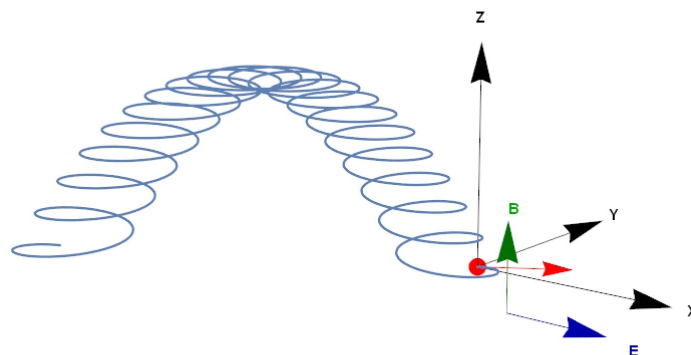
Vektor brzine prikazan je crvenom strelicom zadato sa

`Graphics3D[{Red, Arrowheads[0.03], Arrow[{xyz, xyz + 0.5v} /. t -> t1]}` čiji je početak u centru čestice (tačka) u datom trenutku xyz , a vrh strelice određen sa $xyz + 0.5v$, gde je faktor 0.5 uzet zbog korekcije dužine radi poboljšanja ilustracije.

```
(*konstrukcija koordinatnog sistema*)
coosys = IdentityMatrix[3];
coosys = Graphics3D[Arrow[#] & /@ Transpose[{0 coosys, coosys}]];
(*oznake osa koordinatnog sistema*)
lblcoosys=
Graphics3D[Text[Style#[[1]], Medium, Bold, Black], #[[2]]]
& /@Transpose[{"X", "Y", "Z"}, IdentityMatrix[3] 1.1]];
coosys = Show[coosys, lblcoosys];

(*izvod parametarskih funkcija koordinata po vremenu*)
v = D[xyz, t];
(*animacija tokom koje se menja vreme t*)
Manipulate[
  Show[coosys,
  Graphics3D[{Red, Arrowheads[0.03],
  Arrow[{xyz, xyz + 0.5 v} /. t -> t1]}],
  Graphics3D[{Red, PointSize[0.02], Point[xyz] /. {t -> t1}}],
  ParametricPlot3D[xyz, {t, 0, 25}], Boxed -> False], {t1, 0, 22}]
```

Brzina ima nenultu komponentu u ravni normalnoj na magnetno polje \vec{B} zbog čega će doći do kružnog kretanja. Međutim, istovremeno čestica je imala nenultu komponentu brzine u pozitivnom smeru z -ose, koja će se usled delovanja elektrostatičke sile qE_z usporavati (pozitivno) naelektrisanu česticu do brzine $v_z = 0$, a potom će je ubrzavati u suprotnom smeru. Ovo je ilustrovano na grafiku slika 3.30, gde se vidi da putanja u obliku deformisane helikoide koja postaje sve gušće namotavana prilikom usporavanja duž z -ose, a potom ponovo biva razvučena ubrzavanjem u suprotnom smeru. Naravno, oblik putanje zavisice od izbora početnih uslova i parametara, a na slici je dat i potom opisan samo jedan slučaj.



Slika 3.30: Crvena čestica sa vektorima početne brzine (crveni), električnog (plavi) i magnetnog (zeleni) polja i njena trajektorija (puna plava linija).

3.7.61 Kretanje naelektrisanе čestice u homogenom, stacionarnom magnetnom i vremenski promenljivoj električnom polju.

Naelektrisanа čestica se kreće u homogenom i stacionarnom magnetnom i vremenski zavisnom električnom polju

$$\vec{B} = B_0 \vec{e}_z, \quad \vec{E} = E_0 \cos \omega t \vec{e}_x,$$

gde je ω frekvencija promene električnog polja. Magnetno polje je vremenski nezavisno i usmereno duž z -ose.

- Rešiti jednačine kretanja za česticu koja kreće iz koordinatnog početka iz stanja mirovanja.
- Nacrtati trajektorije za različite vrednosti parametara $\{\omega, E, B\}$.

Program počinje definisanjem vektora položaja \vec{r} , električnog polja $\vec{E} = E_x \cos \omega t \vec{e}_x$, magnetnog polja $\vec{B} = B_z \vec{e}_z$ gde je komponenta z zbog pojednostavljenja konačnog izraza definisana kao $B_z = \frac{m\omega_0}{q}$ i vektor trenutne brzine definisan kao $\vec{v} = \dot{\vec{r}}$.

```
values={\[\Omega]0->1,q->1,m->1,Ex->1};
r={x[t],y[t],z[t]};
Ef={Ex,Ey,Ez};
Ef=Ef/.Thread[Ef->{Ex Cos[\[\Omega] *t],0,0}];
Bf={Bx,By,Bz};
Bf=Bf/.Thread[Bf->{0,0,Bz}];
v=D[r,t];
```

Potom je formulisana diferencijalna jednačina $eq1$, koja sadrži primenu 2. Njutnovog zakona na naelektrisanu česticu, na koju deluje Lorencova sila

$$m\vec{a} = q\vec{E} + q\vec{v} \times \vec{B}.$$

Iz ove vektorske jednačine dobije sistem diferencijalnih jednačina drugog reda po komponentama

$$\begin{aligned} mx''(t) &= B_z q y'(t) + E_x q \cos(t\omega), \\ my''(t) &= -B_z q x'(t), \\ mz''(t) &= 0. \end{aligned}$$

Da bi parametarske funkcije putanja čestice bile jednoznačno određene, za svaku osu potrebna su po 2 uslova. Ovde su dati početni položaj i početna brzina duž sva tri pravca. Diferencijalne jednačine zajedno sa uslovima čini novi skup jednačina *eq2*, glavni argument naredbe *DSolve[eq2, r, t]*. Po izvršenju programa ove linije rezultat je lista transformacija *sol*, iz koje se formira vektor (lista) *xyz*. Komponente ovog vektora su parametarske funkcije koordinata položaja čestice, čiji su izrazi prethodno uprošćeni sa *FullSimplify*

$$\dot{\vec{r}} = \left(\frac{E_x q (\cos(t\omega_0) - \cos(t\omega))}{m(\omega^2 - \omega_0^2)}, \frac{E_x q \omega_0 \sin(t\omega) - E_x q \omega \sin(t\omega_0)}{m\omega^3 - m\omega\omega_0^2}, 0 \right).$$

```
eq1 = Thread[q Ef + q Cross[v, Bf] - m D[r, {t, 2}] == 0];
initial = {
  x[0] == 0, x'[0] == 0,
  y[0] == 0, y'[0] == 0,
  z[0] == 0, z'[0] == 0};
eq2 = Union[eq1, initial];
eq2 = eq2 /. {Bz -> m * \[Omega]0/q};
sol = DSolve[eq2, r, t];
xyz = FullSimplify[r /. sol[[1]]]
xyz1 = xyz /. values
```

Zamenom proizvoljnih vrednosti veličina m, q, ω i E_x zadatim pravilima transformacije u listi *values*, dobija se vektor položaja čestice u saglasnosti sa zadatim parametrima

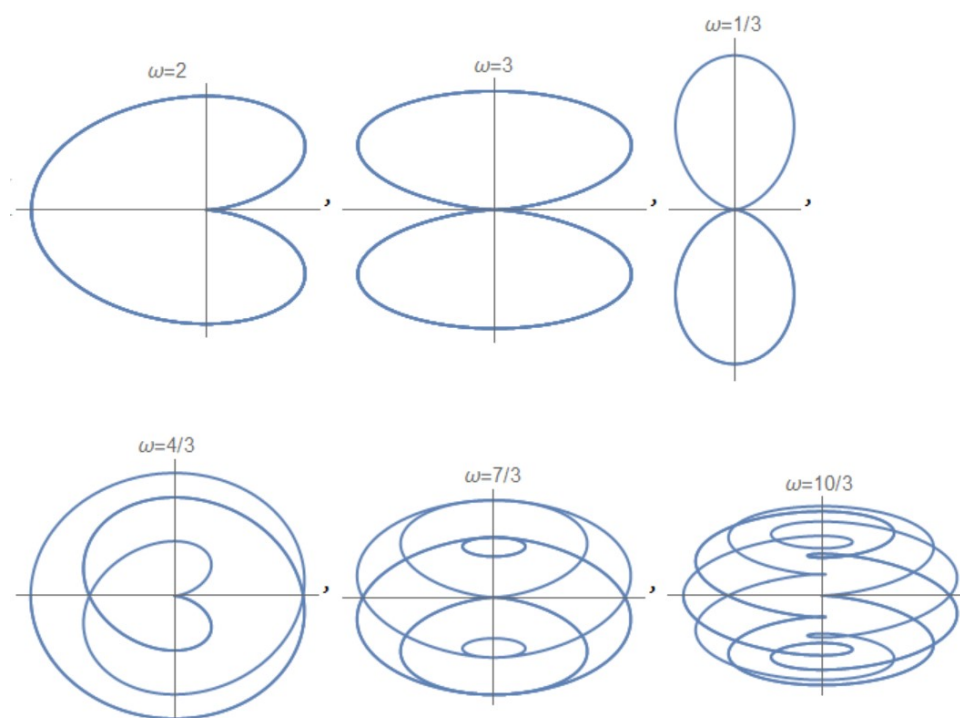
$$\vec{r}(t) = \left(\frac{\cos(t) - \cos(t\omega)}{\omega^2 - 1}, \frac{\sin(t\omega) - \omega \sin(t)}{\omega^3 - \omega}, 0 \right).$$

Paleta orbita u XY ravni za vrednosti kružnih frekvencija $\omega = 2, 3, 1/3, 4/3, 7/3, 10/3$ (različitih od 0 i ω_0) crtana je uz pomoć *ParametricPlot* sa odgovarajućom frekvencijom u oznaci grafika *ParametricPlot[posit1[i, t][[1, 2]], {t, 0, 9 Pi}*, *PlotLabel -> "\omega = " <> ToString[InputForm[i]]*, *Ticks -> None*].

```
orbits = Table[
  ParametricPlot[posit1[i, t][[1, 2]], {t, 0, 9 Pi},
  PlotLabel -> "\[Omega]=" <> ToString[InputForm[i]], Ticks -> None],
  {i, {2, 3, 1/3, 4/3, 7/3, 10/3}}
```

Drugi set parametaraskih funkcija dobijen u limesu kada $\omega \rightarrow \omega_0$ označen je sa *xyz2*. Zamenjene su brojne vrednosti veličina sadržane u listi transformacija *values* i potom kreirana funkcija zavisnosti položaja od vremena *posit2[t]*, čija je orbita nacrtana.

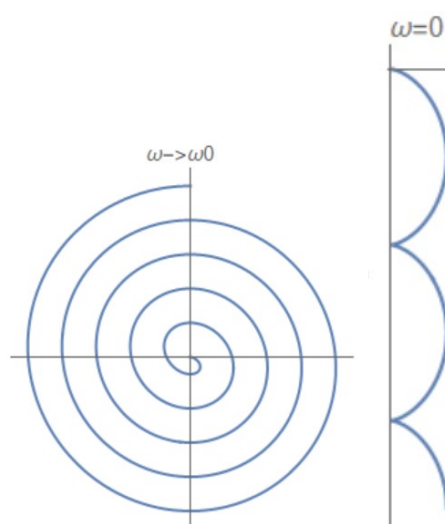
```
xyz2 = Limit[xyz, \[Omega] -> \[Omega]0];
xyz2 = xyz2 /. values;
posit2[t_] := xyz2[[1, 2]];
ParametricPlot[posit2[t], {t, 0, 10 Pi}, Ticks -> None,
  PlotLabel -> "\[Omega]->\[Omega]0"]
```



Slika 3.31: Orbite naelektrisane čestice u homogenom magnetnom i vremenski zavisnom električnom polju za različite frekvencije.

Vektor položaja u limesu kada $\omega \rightarrow \omega_0$

$$\lim_{\omega \rightarrow \omega_0} \vec{r}(t) = \left(\frac{1}{2} t \sin(t), \frac{1}{2} (t \cos(t) - \sin(t)), 0 \right)$$



Slika 3.32: Levo: Orbite naelektrisane čestice u homogenom magnetnom i vremenski zavisnom električnom polju kada $\omega \rightarrow \omega_0$ (levo) i $\omega \rightarrow 0$ (desno).

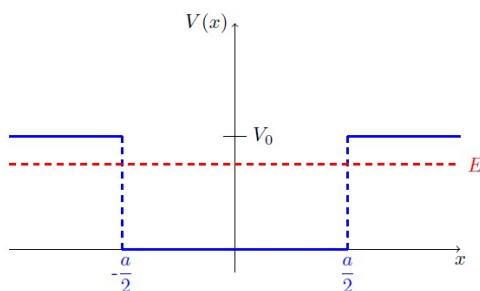
```
xyz3 = Limit[xyz, \[Omega] -> 0];
xyz3 = xyz3 /. values;
ParametricPlot[xyz3[{{1, 2}}], {t, 0, 5 Pi}, Ticks -> None,
PlotLabel -> "\[Omega]=" <> ToString[0]]
```

U limesu kada $\omega \rightarrow 0$ vektor položaja naelektrisane čestice, dobijen u skladu sa zadatim vrednostima promenljivih i početnih uslova je

$$\lim_{\omega \rightarrow 0} \dot{\vec{r}}(t) = (1 - \cos(t), \sin(t) - t, 0).$$

3.7.62 Vezano stanje čestice u pravougaonoj potencijalnoj jami.

Čestica se nalazi u konačno dubokoj potencijalnoj jami širine a i dubine V_0 . Analitički predvideti prirodu energijskog spektra čestice vezanih stanja.



Traži se rešenje vremenski nezavisne Šredingerove jednačine

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} + V(x)\psi = E\psi,$$

za česticu koja se nalazi u potencijalu oblika

$$V(x) = \begin{cases} 0 & |x| < \frac{a}{2} \\ V_0 & |x| > \frac{a}{2} \end{cases}.$$

Čestica je u vezanom stanju $E < V_0$, što znači da je gustina stanja izvan jame mala i brzo opada sa rastojanjem od ivica jame. Diferencijalna jednačina se rešava tako da se prostor podeli na oblasti $x \in (-a/2, a/2)$ i $x \in (-\infty, -a/2) \cup (a/2, \infty)$, te se potom u svakoj od njih nezavisno rešava Šredingerova jednačina, a dobijena rešenja povezuju uslovima neprekidnosti talasne funkcije i njenog prvog izvoda u tačkama konačnog skoka potencijala. Potencijal je vremenski nezavisan, reč je o stacionarnom problemu. U oblasti $x \in (-\infty, -a/2) \cup (a/2, \infty)$ Šredingerova jednačina postaje

$$\psi'' - \frac{2m}{\hbar^2}(V_0 - E)\psi = 0,$$

odnosno dobija se diferencijalna jednačina drugog reda

$$\psi'' - \kappa^2\psi = 0$$

gde je $\kappa = \sqrt{\frac{2m(V_0 - E)}{\hbar^2}}$. Za $x \in (-a/2, a/2)$ rešavanjem Šredingerove dobije se diferencijalna jednačina oblika

$$\psi'' + \frac{2m}{\hbar^2}E\psi = 0,$$

odnosno

$$\psi'' + k^2\psi = 0,$$

gde je $k = \sqrt{\frac{2mE}{\hbar^2}}$. Simboličko rešavanje dobijenih diferencijalnih jednačina uz pomoć *DSolve* realizuje se kao

```
eq1=\[Psi]''[x]+(k^2)\[Psi][x]==0;
eq2=\[Psi]''[x]-(\[Kappa]^2)\[Psi][x]==0;
values1={k->Sqrt[2mE/h],\[Kappa]->Sqrt[2m*(V0-E)/h]};
sol1=DSolve[eq1,\[Psi][x],x][[1]]
sol2=DSolve[eq2,\[Psi][x],x][[1]]
```

odakle se dobije rešenje *sol1* za jednačine *eq1* u obliku

$$\psi(x) \rightarrow A_0 \sin(kx) + B_0 \cos(kx),$$

dok je transformacija *sol2* rešenje jednačine *eq2* u obliku izlaza

$$\psi(x) \rightarrow A_1 e^{\kappa x} + B_2 e^{-\kappa x}.$$

Ispitujući fizičko rešenje talasne funkcije u oblasti izvan jame, kada vrednosti x teži $\pm\infty$.

```
psiout=\[Psi][x] /. sol2
Assuming[\[Kappa] > 0, Limit[psiout, x->-Infinity]]
```

Ova analiza pomaže da se odbace nefizički delovi talasne funkcije. Tako, talasna funkcija u oblasti $x \in (-a/2, a/2)$ ima oblik $\psi_0(x) = A_0 \sin kx + B_0 \cos kx$, za $x < -a/2$ zbog divergencije funkcije $\lim_{x \rightarrow -\infty} e^{-\kappa x} = \infty$, talasna funkcija je oblika $\psi_1(x) = A_1 e^{\kappa x}$, dok je za $x > a/2$ zbog $\lim_{x \rightarrow \infty} e^{\kappa x} = \infty$ talasna funkcija oblika $\psi_2(x) = B_2 e^{-\kappa x}$.

```
\[Psi]0[x_]:=A0*Sin[k x]+B0*Cos[k x];
\[Psi]1[x_]:=A1*E^(x \[Kappa]);
\[Psi]2[x_]:=B2*E^(-x \[Kappa]);
conditions ={\[Psi]1[0]-\[Psi]0[0], \[Psi]1'[0]-\[Psi]0'[0],
\[Psi]0[a]-\[Psi]2[a], \[Psi]0'[a]-\[Psi]2'[a]};
```

Nakon definisanja funkcija u različitim oblastima, postavljeni su uslovi njihovih neprekidnosti kao i neprekidnosti prvih izvoda u tačkama konačnog skoka potencijala $x = \pm a/2$, koji su deponovani u listu *conditions*. Uslovi dati u listi čine sistem algebarskih jednačina, koji se može napisati matricno,

$$\mathcal{A}\tilde{\psi} = \begin{pmatrix} e^{-\frac{ak}{2}} & \sin\left(\frac{ak}{2}\right) & -\cos\left(\frac{ak}{2}\right) & 0 \\ \kappa e^{-\frac{ak}{2}} & -k \cos\left(\frac{ak}{2}\right) & -k \sin\left(\frac{ak}{2}\right) & 0 \\ 0 & \sin\left(\frac{ak}{2}\right) & \cos\left(\frac{ak}{2}\right) & -e^{-\frac{ak}{2}} \\ 0 & k \cos\left(\frac{ak}{2}\right) & -k \sin\left(\frac{ak}{2}\right) & \kappa e^{-\frac{ak}{2}} \end{pmatrix} \begin{pmatrix} A_1 \\ A_0 \\ B_0 \\ B_2 \end{pmatrix} = 0. \quad (3.23)$$

Fizički zahtev da je vektor stanja nenulti, odnosno $(A_1, A_0, B_0, B_2)^T \neq 0$ ekvivalentan je uslovu $\det \mathcal{A} = 0$. Da bi se dobila matrica \mathcal{A} potrebno je ekstrahovati koeficijente uz koordinate vektora $\tilde{\psi}$. Uslovi, dati preko izraza linearno zavisnih od koeficijenata A_1, A_0, B_0, B_2 , redom se diferenciraju $D[\text{conditions}[[i]], \{A_1, A_0, B_0, B_2\}]$.

```
conditions ={\[Psi]1[-a/2]-\[Psi]0[-a/2],
\[Psi]1'[-a/2]-\[Psi]0'[-a/2], \[Psi]0[a/2]-\[Psi]2[a/2],
\[Psi]0'[a/2]-\[Psi]2'[a/2]};
Amat =Table[D[conditions[[i]], {A1, A0, B0, B2}],
{i, Length[conditions]}];
finsol = Solve[Det[Amat] == 0, \[Kappa]];
\[Kappa]1 =\[Kappa] /. finsol[[2]]
```

Diferenciranjem svakog uslova ponaosob po promenljivim koje su istovremeno koordinate vektora $\tilde{\psi}$, dobiju se vrste matrice \mathcal{A} . Uslov netrivialnosti rešenja $\tilde{\psi} \neq 0$ jeste singularnost matrice \mathcal{A} , što je ekvivalentno uslovu da je njena determinanta nula. Na osnovu definisanog zahteva dobija se jednačina $\det \mathcal{A} = 0$ čije rešenje se traži traženjem korena polinoma $\text{Solve}[\text{Det}[\mathcal{A}] == 0, \kappa]$. Rešenje dobijene algebarske jednačine je vrednost $\kappa(k)$. Dobijena rešenja izvršenjem naredbe Solve , potrebno je srediti i selektovati fizičko $\kappa_1 = \kappa/\text{finsol}[[2]]$ rešenje dobijene algebarske jednačine iz uslova da su $k > 0$ i $\kappa > 0$.

Sređivanjem izraza nastalog iz uslova (3.23) dobije se transcendentna jednačina

$$\kappa = k \tan\left(\frac{ak}{2}\right) \quad (3.24)$$

čija su rešenja dozvoljene energije E čestice vezane u konačno dubokoj potencijalnoj jami dubine V_0 i širine a i nalaze se numerički.

4 Numerički potencijal i kalkulacije funkcijama programskog jezika Wolfram

4.1 Mašinska i kontrolisana preciznost

Mathematica tretira približne relane brojeve na nivou mašinske preciznosti ili brojeve proizvoljne preciznosti. Brojevi mašinske preciznosti imaju fiksiran broj značajnih cifara, dok brojevi proizvoljne preciznosti mogu imati bilo koji veći broj cifara. Za brojeve mašinske preciznosti naredbom `Precision[x]` dobija se izlaz `MachinePrecision`, čija je numerička vrednost mašinska preciznost. Za računarski operativni sistem *Windows* ova preciznost je data na 16 cifara, dok brojevi proizvoljne preciznosti prikazuju se sa preciznošću većom od mašinske.

```
In[1]:= Precision[3.14]
```

```
Out[1]= MachinePrecision
```

Osim ako nije drugačije zadato, aproksimirani realni brojevi iako učitani (uneti) sa manjom, Mathematica uzima u obzir sa mašinskom preciznošću. Brojni rezultati dobijeni u programu Mathematica po definiciji ispisuju se samo na šest cifara.

```
In[2]:= Sqrt[5.0]
```

```
Out[2]= 2.23607
```

Ukoliko je potrebno ispisati ostale cifre, koje određuju rezultat do na mašinsku preciznost, koristi se naredba `InputForm`.

```
In[3]:= InputForm[Sqrt[5.0]]
```

```
Out[3]//InputForm=
```

```
2.23606797749979
```

Dok se drugačije ne naglasi program Mathematica izvršava funkciju simbolički. Jedan od načina da se funkcija (izraz) *expr* izvrši tako da je rezultat realan broj do na mašinsku preciznost koristi se `N[expr]` ili do na neku zadatu preciznost do na *n* cifara `N[expr, n]`. Primera radi biće izračunat obim kruga radijusa 3 m prikazan na 50 cifara.

```
In[5]:= N[2 * 3 * Pi, 50]
```

```
Out[5]= 18.849555921538759430775860299677017305 \
183016396251
```

4.2 Rešavanje polinomijalnih jednačina numerički

Ne postoji uopšteno analitički metod za nalaženje rešenja polinomijalne jednačine stepena većeg od četiri, zbog čega se za rešavanje obično koriste numerički metodi. Numerička aproksimacija korena polinomijalne jednačine ili sistema jednačina *eqs* u programu Mathematica nalazi se uz pomoć

$$NSolve[eqns, vars].$$

Jednačine se u programu Mathematica postavljaju u obliku *lhs == rhs*, povezujući izraze operatorom *==*, dok je *=* rezervisan za pridruživanje vrednosti nekoj promenljivoj. Primer rešavanja polinomijalne jednačine stepena 5

$$x^5 + 4x^3 + 3x^2 + 2x = 10.$$

```
eq=x^5+4x^3+3x^2+2x==10;
sol=NSolve[eq,x]
```

Izlaz iz programa je skup transformacionih pravila oblika $x \rightarrow x_i$, $i = 1, 2, \dots, 5$ u obliku liste $\{\{x \rightarrow -1. - 1.i\}, \{x \rightarrow -1. + 1.i\}, \{x \rightarrow 0.5 - 2.17945i\}, \{x \rightarrow 0.5 + 2.17945i\}, \{x \rightarrow 1.\}\}$ gde su x_i približni koreni. Primer rešavanja sistema polinomijalnih jednačina, koji se zadaje u obliku liste

$$\begin{aligned}x^2 - xy - y^2 &= 1 \\x^3 + 3xy - y^3 &= 9.\end{aligned}$$

```
eqs={x^2-x*y-y^2==1,x^3+3x*y-y^3== 9};
NSolve[eqs,{x,y}]
```

Rezultat numeričkog rešavanja sistema jednačina je skup pravila transformacija:

```
{x -> 1.76268, y -> -2.57952},
{x -> -0.43195 + 0.816763i, y -> 0.552813 - 1.71762i},
{x -> -0.43195 - 0.816763i, y -> 0.552813 + 1.71762i},
{x -> -2.07194 + 1.87447i, y -> -1.16444 + 1.26905i},
{x -> -2.07194 - 1.87447i, y -> -1.16444 - 1.26905i},
{x -> 1.74511, y -> 0.802781}
```

čiji su elementi parovi rešenja $\{x \rightarrow x_i, y \rightarrow y_i\}$.

4.2.63 Primenom naredbe *NSolve* rešiti jednačine.

i) $x^7 + x^5 + 2x^3 + x^2 + 1 = 0$

```
eq = x^7 + x^5 + 2 x^3 + x^2 + 1 == 0;
NSolve[eq, x]
```

Rešenje: $\{\{x \rightarrow -0.812432\}, \{x \rightarrow -0.640787 - 1.07931i\}, \{x \rightarrow -0.640787 + 1.07931i\}, \{x \rightarrow 0.254825 - 0.700968i\}, \{x \rightarrow 0.254825 + 0.700968i\}, \{x \rightarrow 0.792178 - 0.881387i\}, \{x \rightarrow 0.792178 + 0.881387i\}\}$.

ii) $4\beta^7 - 16\beta^4 + 17\beta^3 + 6\beta^2 - 21\beta + 10 = 0$

```
eq=4\ [Beta] ^7-16 \ [Beta] ^4+17\ [Beta] ^3+6\ [Beta] ^2-21\ [Beta]+10==0;
NSolve[eq, \ [Beta]]
```

Rešenje: $\{\{\beta \rightarrow -1.05209 - 1.54511i\}, \{\beta \rightarrow -1.05209 + 1.54511i\}, \{\beta \rightarrow -1.\}, \{\beta \rightarrow 0.648515 - 0.68252i\}, \{\beta \rightarrow 0.648515 + 0.68252i\}, \{\beta \rightarrow 0.807146\}, \{\beta \rightarrow 1.\}\}$.

4.2.64 Funkcijama za numeričko rešavanje naći rešenje sistema jednačina.

$$\begin{aligned}x^2 + xy + y^2 &= 1 \\x^3 + x^2y + xy^2 + y^3 &= 4.\end{aligned}$$

```
eqs={x^2+x*y+y^2==1,x^3+x^2*y+x*y^2+y^3== 4};
NSolve[eqs, {x, y}]
```

Rešenje: $\{\{x \rightarrow -1. + 1.41421i, y \rightarrow -1. - 1.41421i\},$
 $\{x \rightarrow -1. - 1.41421i, y \rightarrow -1. + 1.41421i\},$
 $\{x \rightarrow 1.6838 + 0.133552i, y \rightarrow -0.683802 - 1.13355i\},$
 $\{x \rightarrow 1.6838 - 0.133552i, y \rightarrow -0.683802 + 1.13355i\},$
 $\{x \rightarrow -0.683802 + 1.13355i, y \rightarrow 1.6838 - 0.133552i\},$
 $\{x \rightarrow -0.683802 - 1.13355i, y \rightarrow 1.6838 + 0.133552i\}\}.$

4.2.65 Svojtveni problem.

Svojtveni problem, nalaženje spektra a potom i odgovarajućih svojtvenih vektora zadate simetrične matrice, biće rešen postupno, prema navedenom redosledu, najpre nalazeći spektar tražeći koren karakterističnog polinoma uz pomoć *NSolve* a potom i svojtvenih vektora koristeći *NullSpace*. Dobijeni rezultat biće proveren upotrebom naredbe *Eigenvalues[]*. Budući da je reč o simetričnoj matrici, nju je moguće razložiti na spektralnu formu. Na osnovu dobijenih rezultata biće izvršena njena spektralna dekompozicija. Transformacijom sličnosti, zadatu matricu prevesti u njoj sličnu, dijagonalnu pomoću ortogonalnog operatora formiranog iz dobijenih svojtvenih vektora.

Prema definiciji, vektor $x \in U_n$, $x \neq 0$ je svojtveni vektor operatora $A \in L(U_n, U_n)$, ako važi da je $Ax = \lambda x$, gde je λ svojtvena vrednost, koja pripada spektru operatora A . Za vektor x kažemo da pripada svojtvenom potprostoru za svojtvenu vrednost λ , $x \in V_\lambda$. Ako je zadat bazis $\beta = \{v_1, v_2, \dots, v_n\}$ prostora U_n , operator A se u njemu reprezentuje matricom \mathcal{A} , a vektor x reprezentovan je kolonom $\mathcal{X} \in \mathcal{F}^n$ gde je \mathcal{F} polje ($\mathcal{F} = \mathcal{R}, \mathcal{C}$) pri čemu važi da je

$$\mathcal{A}\mathcal{X} = \lambda\mathcal{X}.$$

Iz prethodne jednakosti može se dobiti $(\mathcal{A} - \lambda I)\mathcal{X} = 0$, odakle se vidi da svojtveni vektor \mathcal{X} pripada nul-potprostoru operatora $\mathcal{X} \in N(\mathcal{A} - \lambda I)$. Svojtveni potprostor nalazi se iz uslova da je $V_\lambda = N(\mathcal{A} - \lambda I)$. Spektar matrice \mathcal{A} se dobija iz uslova singularnosti matrice $(\mathcal{A} - \lambda I)$ što je ekvivalentno sa $\det(\mathcal{A} - \lambda I) = 0$. Zapravo, traže se koreni karakterističnog (ili svojtvenog) polinoma n -tog stepena $p(\lambda) = \det(\mathcal{A} - \lambda I)$. Program počinje zadavanjem prirodnog broja n , kojim je određen format matrice. Zatim je generisana matrica slučajnih realnih brojeva od 1 do 3 formata $n \times n$, koja je u nastavku simetrizovana dodavanjem njoj transponovane, kako bi se dobila simetrična matrica čiji je spektar realan. Da bi se demonstrirao rad algoritma kojim se rešava opštiji problem, potrebno je pronaći matricu čiji su svojtveni potprostori degenerisani. Zbog toga, rešen je svojtveni problem matrice dobijene direktnim proizvodom realne simetrične i jedinične formata 2×2 . Ovim se izvesno dobija matrica u čijem spektru se nalaze degenerisane svojtvene vrednosti. Direktni proizvod matrica \mathcal{A} i \mathcal{B} u kodu realizovan funkcijom

$$KroneckerProduct[\mathcal{A}, \mathcal{B}]$$

čiji su argumenti matrice, a izlaz je njihov direktni proizvod.

```

Clear[n];
n=5;
Amat=Table[Random[Integer,{1,3}],{n},{n}];
Amat=Amat+Transpose[Amat];
(*direktni proizvod matrica*)
Amat=KroneckerProduct[Amat,{{1,0},{0,1}}];
Sort[Chop[Eigenvalues[1.Amat]]]
n=2n;
MatrixForm[Amat]

```

Nakon opisanog postupka generisana je realna simetrična matrica formata 10×10 .

$$\mathcal{A} = \begin{pmatrix} 6 & 0 & 4 & 0 & 4 & 0 & 5 & 0 & 5 & 0 \\ 0 & 6 & 0 & 4 & 0 & 4 & 0 & 5 & 0 & 5 \\ 4 & 0 & 4 & 0 & 3 & 0 & 5 & 0 & 4 & 0 \\ 0 & 4 & 0 & 4 & 0 & 3 & 0 & 5 & 0 & 4 \\ 4 & 0 & 3 & 0 & 4 & 0 & 6 & 0 & 4 & 0 \\ 0 & 4 & 0 & 3 & 0 & 4 & 0 & 6 & 0 & 4 \\ 5 & 0 & 5 & 0 & 6 & 0 & 4 & 0 & 4 & 0 \\ 0 & 5 & 0 & 5 & 0 & 6 & 0 & 4 & 0 & 4 \\ 5 & 0 & 4 & 0 & 4 & 0 & 4 & 0 & 6 & 0 \\ 0 & 5 & 0 & 4 & 0 & 4 & 0 & 4 & 0 & 6 \end{pmatrix}.$$

Karakteristični polinom $p(\lambda) = (-\lambda^5 + 24\lambda^4 - 28\lambda^3 - 134\lambda^2 + 257\lambda - 118)^2$ date matrice do-
bijen je prema definiciji, kao determinanta razlike zadate i jedinične matrice istog formata
pomnožene skalarom λ . Dalje, primenom naredbe *NSolve* čiji je argument jednačina $p(\lambda) = 0$
izračunate su nule karakterističnog polinoma. Dobijen je spektar matrice zamenom pravila do-
bijenih izvršenjem programa a potom zamenom nađenih pravila
{22.514, 22.514, -2.58969, -2.58969, 2.12183, 2.12183, 1., 1., 0.95383, 0.95383}. Da bi se otklonile više-
strukosti korena svojstvenog polinoma na skup svojstvenih vrednosti u nastavku je primenjena
naredba *Union*, zajedno sa zadatim uslovom
SameTest $\rightarrow (Abs[\#1 - \#2] \leq 10^{-(15)} \&)$ u opcijama naredbe, kojim se svojstvene vrednosti raz-
likuju do na 10^{-15} . Sve svojstvene vrednosti čija je razlika manja ili jednaka 10^{-15} se primenom
ovog uslova smatraju istim.

```

n=Length[Amat];(*duzina liste Amat*)
Bmat=Amat-[Lambda]*IdentityMatrix[n];
pol=Det[Bmat];(*karakteristicni polinom*)
sol=NSolve[pol==0,[Lambda]];
(*Nadjena determinanta od B, izjednacena sa 0, zatim
se reseva jednačina numericki*)
[Sigma]=Sort[[Lambda]/.sol];
(*spektar sortiran od manje ka vecoj vrednosti*)
[Sigma]=Union[[Sigma],SameTest->(Abs[#1-#2]<=10^{-15}&)]

```

Svojstveni potprostori V_{λ_i} koji odgovara svojstvenoj vrednosti λ_i , prema gore datoj definiciji,
dobijen je kao jezgro operatora $\mathcal{A} - \lambda_i I$ primenom naredbe

$$\text{NullSpace}[\mathcal{A} - \lambda I]$$

čiji je argument matrica $\mathcal{A} - \lambda I$, a izlaz skup vektora koji formiraju bazis jezgra matrice u argu-
mentu.

```

decomp={};
Do[
l=\[Sigma][[j]];(*izdvojena svojstvena vrednost*)
Bop=Amat-l*IdentityMatrix[n];
(*za ovu matricu trazi se nul-potprostor*)
eigvec=NullSpace[Bop];
eigvec=Chop[eigvec];
Prj=Total[KroneckerProduct[#,#]&/@eigvec];
AppendTo[decomp,{l,eigvec,Prj}},{j,Length[\[Sigma]]}];

```

Potom se iz dobijenih, ortnormiranih vektora $\{e_i | i = 1, \dots, k\}$ konstruiše projektor na svojstveni potprostor za datu svojstvenu vrednost. Prazan skup *decomp* puni se redom trojkama koje sadrže svojstvenu vrednost, zatim odgovarajući svojstveni vektor i projektor $\{\lambda_i, x_{\lambda_i}, P_{\lambda_i}\}$. Izvršena je suma svih projektoru da bi se proverila dekompozicija jedinice.

```

unt=Total[decomp[[All,3]]];
MatrixForm[Chop[unt]]

```

Implementirajući operatorsku jednakost spektralne forme,

$$A = \sum_i \lambda_i P_i,$$

gde su λ_i svojstvene vrednosti, a

$$P_i = \sum_j x_j x_j^\dagger, \quad x_j \in V_{\lambda_i}$$

svojstveni projektori za odgovarajuću svojstvenu vrednost gde brojač j prebrojava degenerisane vektore $j = 1, \dots, \dim V_{\lambda_i}$, dobijena je matrica *Aspct*. Budući da se zadatak rešava u prostoru nad poljem realnih brojeva, množenje kolone sa njom adjungovanom, svodi se na množenje kolone i vrste, bez konjugovanja

$$\mathcal{X}_{\lambda_i} \mathcal{X}_{\lambda_i}^\dagger = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \end{pmatrix} (a_1 \quad a_2 \quad \dots).$$

Razlika početne i matrice dobijene spektralnom formom daje nultu matricu, čime je izvršena provera računa.

```

(*spektralna forma normalnog operatora*)
Aspct=Sum[decomp[[j,1]]decomp[[j,3]},{j,Length[decomp]};
MatrixForm[Aspct];
MatrixForm[Chop[Amat-Aspct]]
(*razlika date i matrice dobijene spektralnom dekompozicijom*)

```

Dobijeni svojstveni vektori su međusobno ortnormirani u odnosu na standardni skalarni proizvod, što se lako može proveriti njihovim međusobnim skalarnim množenjem. Izdvoje se svi dobijeni bazisi svojstvenih potprostora *Flatten[decomp[[All,2]],1]*, a potom se listi bazisa skine jedna zagrada (izravna se za jedan sloj), čime se dobija lista vektora. Svaki vektor se skalarno, standardnm skalarnim proizvodom koristeći *.* ili *Dot[v1,v2]* pomnoži sa svim ostalim. Rezultat skalarnog proizvoda je argument funkcije *Table*, čiji brojači prođu kroz ceo skup svojstvenih vektora. Rezultat opisane procedure je jedinična matrica, što znači da je skalarni proizvod istih vektora jednak 1 a različitih 0.

```

eigvcs=Flatten[decomp[[All,2]],1];
testON=Chop[Table[eigvcs[[i]].eigvcs[[j]],
{i,Length[eigvcs]},{j,Length[eigvcs]}]];
Chop[testON]//MatrixForm

```

Lista svih bazisa svojstvenih potprostora se poravnava, da bi se dobila lista svojstvenih vektora. Iz uslova da se radi sa standardnim skalarnim proizvodom, ova lista čini ortogonalnu matricu

$$U = (\mathcal{X}_{\lambda_{1,1}} \quad \mathcal{X}_{\lambda_{1,2}} \quad \cdot \quad \cdot \quad \cdot),$$

čije su kolone ortnormirane $(\mathcal{X}_{\lambda_{i,j}}, \mathcal{X}_{\lambda_{k,l}}) = \delta_{i,k} \delta_{j,l}$ i prestavlja matricu prelaska iz apsolutnog u svojstveni bazis, gde indeksi i, k odgovaraju različitim svojstvenim vrednostima, a indeksi j, l prebrojavaju vektore unutar istog svojstvenog potprostora. Transformacijom sličnosti iz početne dobija se dijagonalna matrica

$$A_{diag} = U^T A U$$

čiji su dijagonalni elementi svojstvene vrednosti matrice A_{mat} (a svojstveni bazis apsolutni)

$$A_{diag} = \begin{pmatrix} \lambda_1 & 0 & 0 & \dots \\ 0 & \lambda_2 & 0 & \dots \\ 0 & 0 & \lambda_3 & \dots \\ \vdots & & & \end{pmatrix}.$$

```
Umat=Transpose[decomp[[All, 2]]];(*selekcija svojstvenih vektora*)
Umat=Flatten[Umat, 1];
Adiag=Umat.Amat.Transpose[Umat];
MatrixForm[Chop[Adiag]]
TeXForm[%]
```

$$A_{diag} = \begin{pmatrix} -2.58969 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2.58969 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.95383 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.95383 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.12183 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.12183 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22.514 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22.514 \end{pmatrix}$$

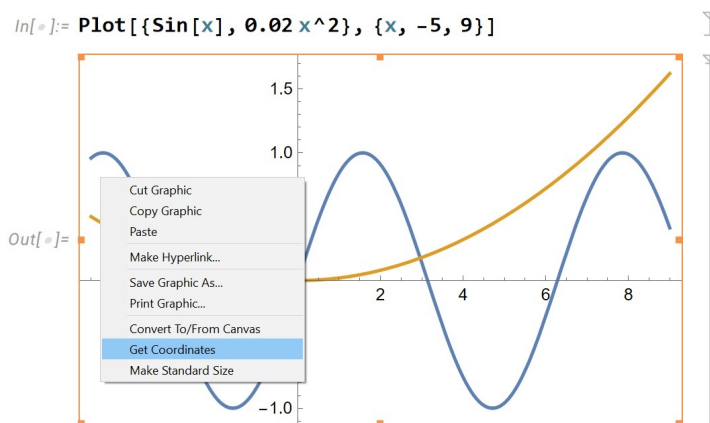
4.3 Nalaženje korena, minimuma i maksimuma nepolinomijalnih funkcija

Naredbom *NSolve* numerički se traže koreni polinomijalnih funkcija. Ovom naredbom nije moguće naći rešenje jednačine u opštem slučaju ukoliko nije polinomijalna. Primera radi, traženje rešenja jednačine $\sin x = x$ uz pomoć naredbe *NSolve* je bezuspešno i po izvršenju naredbe biće vraćen neizvršen ulaz. Funkcijom *FindRoot* moguće je naći numeričke rešenja proizvoljne jednačine

$$FindRoot[lhs == rhs, \{x, x_0\}],$$

kojom se traži numeričko rešenje proizvoljne jednačine $lhs == rhs$ nepoznate x , počevši od $x = x_0$. Uspešnost nalaženja korena neke funkcije u velikoj meri zavisi od izbora početne vrednosti od koje algoritam započinje potragu. Ukoliko je početna vrednosti dovoljno blizu rešenju, izvršenjem ove naredbe dobije se traženi rezultat. Preporučuje se određena procedura tokom traženja rezultata, koja podrazumeva grafičku analizu. Ona prethodi izvršenju funkcije, nakon čega se otprilike detektuju rešenja u čijoj okolini se nastavlja numerička potraga za preciznim rezultatom. Sledećim primerom biće ilustrovan algoritam traženja korena nepolinomijalne funkcije $\sin x = 0.02x^2$, gde postoji više rešenja. U ovom slučaju neophodno je naći nekoliko početnih tačaka od kojih se pokreće potraga za rešenjem.

- i) Kao prvi korak u detekciji rešenja je grafički prikaz složenih funkcija i njihovog preseka, geometrijski gledano čije se koordinate traže.
- ii) Drugi korak zahteva određivanje konkretnih, brojnih po vrednosti, startnih pozicija u okolini vizuelno detektovanih korena i njihovo deponovanje u otvoren skup $xycoo$. Ovo će biti realizovano upotrebom funkcije *ClickPane*, dok je u novijim verzijama moguće koristiti alate za crtanje iz grafičkog menija, u kom postoji opcija očitavanja koordinata sa 2D grafika.



Desnim klikom otvori se meni u kom se bira *GetCoordinates*. Potom levim klikom selektuju se tačke na grafiku. Izabrane tačke se dalje deponuju u prethodno otvoren prazan skup $xycoo$ naredbom *Paste*.

- iii) Za svaku selektovanu tačku $xycoo[[i]]$ (ukoliko je formiran skup), koja je potom dodata skupu $xycoo$, izvršava se naredba za traženje korena

$$FindRoot[\sin x == 0.02x^2, \{x, xycoo[[i, 1]]\}],$$

pri čemu se uzima početna x koordinata $xycoo[[i, 1]]$. Ukoliko je korišćena naredba *ClickPane*, tada je $xycoo$ lista od dve komponente, pri čemu se uzima prva

$$FindRoot[\sin x == 0.02x^2, \{x, xycoo[[1]]\}].$$

```
ClickPane[Plot[{Sin[x], 0.02 x^2}, {x, -5, 9}], (xycoord = #) &]
```

Rezultat funkcije *FindRoot* je transformacija kojom se zadaje vrednost promenljive $x \rightarrow solution$. Provere radi, promenljiva x je zamenjena vrednošću dobijenom nakon izvršenja funkcije.

```
root = FindRoot[Sin[x] == 0.02 x^2, {x, xycoord[[1]]}]
(Sin[x] - 0.02 x^2) /. root
```

Traženje lokalnih minimuma i maksimuma numerički pomoću funkcija programa Mathematica moguće je upotrebom naredbi *FindMinimum* i *FindMaximum* :

- *FindMinimum* $[f, \{x, x_0\}]$ je funkcija kojom se numerički traži najbliži lokalni minimum od f , počevši od $x = x_0$,
- *FindMaximum* $[f, \{x, x_0\}]$ se koristi za nalaženje najbližeg lokalnog maksimuma od $x = x_0$.

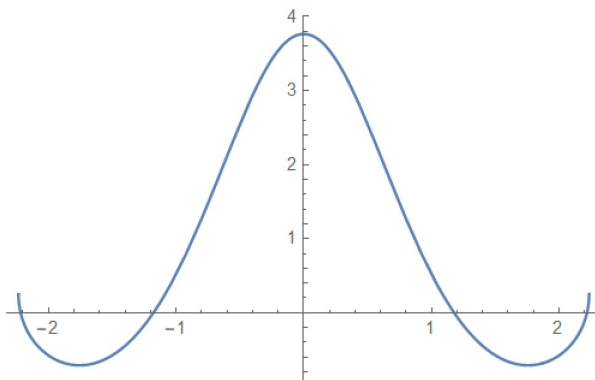
Vrednost x_0 može poslužiti kao početna tačka od koje počinje potraga minimuma/maksimuma ukoliko je $f(x_0)$ dovoljno blizu lokalnog minimuma odnosno maksimuma. Određivanje startnih tačaka za nalaženje lokalnog ekstremuma slična je onoj za određivanje korena.

4.3.66 Koristeći naredbu *FindRoot* rešiti jednačinu

$$6\operatorname{sech}(x)^2 - \sqrt{5 - x^2} = 0.$$

Najpre je nacrtana funkcija zadata sa leve strane jednakosti. Nakon čega se odredi interval u kom ova funkcija seče osu nezavisno promenljive x .

```
Plot[6 Sech[x]^2-Sqrt[5-x^2],{x,-3,3},PlotRange->All]
```



Slika 4.1: Grafik funkcije čije se nule traže na intervalu od -3 do 3.

Na dobijeni grafik postavi se kursor i desnim klikom pojavi se meni na kom se izabere opcija *GetCoordinates*. Kursor postavimo na grafik što bliže nulama funkcije, a potom se klikom na tačke pokupe njihove koordinate. Nakon toga, prikupljene koordinate iz memorije nalepe se u novu ćeliju sveske. Iz liste koordinata pročitanih sa grafika u blizini nula funkcije izdvoje se vrednosti nezavisno promenljive, odnosno x komponente.

```
(*tačke prikupljene sa gornjeg grafika*)
set=
{{-2.265400870474698, 0.08513955965804776},
{-1.106419121779983,0.0051083411161336745'},
{1.111099290722572, 0.0051083411161336745'},
{2.285534129399884,0.0051083411161336745'}};
(*izdvajanje x komponente iz parova*)
set=set[[All,1]];
rez=Table[
ini=set[[i]];
FindRoot[6 Sech[x]^2 - Sqrt[5 - x^2], {x, ini}], {i, Length[set]};
Chop[x /. rez]
```

Lista prikupljenih tačaka, gore opisanom metodom u blizini nula zadate funkcije, pridružena je promenljivoj *set*. Iz nje su izdvojene prve koordinate svih parova *set*[[*All*,1]], odnosno x -komponente odakle će se inicirati traženje nula. Promenljivoj *ini* = *set*[[*i*]] redom se dodeljuju početne tačke, a potom od njih počinje traganje za nulom

$\text{FindRoot}[6\operatorname{Sech}[x]^2 - \operatorname{Sqrt}[5 - x^2], \{x, \text{ini}\}], i = 1, \dots, \text{Length}[\text{set}].$

Nađene su sve četiri nule zadate funkcije
 $\{-2.21882, -1.17705, 1.17705, 2.21882\}.$

4.3.67 Sferna Beselova funkcija reda n , označava se sa $j_n(x)$.

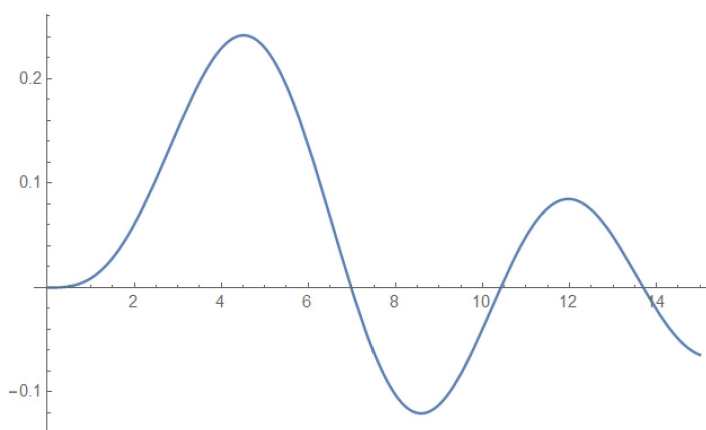
i) Nacrtati $j_3(x)$ na intervalu $x \in [0, 15]$.

- ii) Odrediti prve tri nule na pozitivnom delu x - ose koristeći dobijeni grafik za očitvanjanje inicijalnih pozicija.

Sferna beselova funkcija reda n u programu Mathematica , pogodna kako za numerički tako i za simbolički račun, poziva se funkcijom

$$\text{SphericalBesselJ}[n, x].$$

```
Plot[SphericalBesselJ[3, x], {x, 0, 15}]
```



Slika 4.2: Sferna Beselova funkcija reda 3 na intervalu od 0 do 15.

```
set = {{6.78508641870714, -0.00010882584073812485'},
{10.236009912734097, -0.0022717078832897863'},
{13.63503982038471, -0.0011902668620140666'}};
set=set[[All,1]];
rez = Table[
ini = set[[i]];
FindRoot[SphericalBesselJ[3, x], {x, ini}], {i, Length[set]};
rez = x /. rez
```

Prve tri nule Beselove funkcije 3. reda na pozitivnom delu x - ose su $\{6.98793, 10.4171, 13.698\}$.

4.3.68 Za Beselovu funkciju prve vrste i reda 0, označena sa $J_0(x)$

- i) naći prvih 5 nula od koordinatnog početka za $x > 0$,
ii) prva dva lokalna minimuma u pozitivnom delu x - ose.

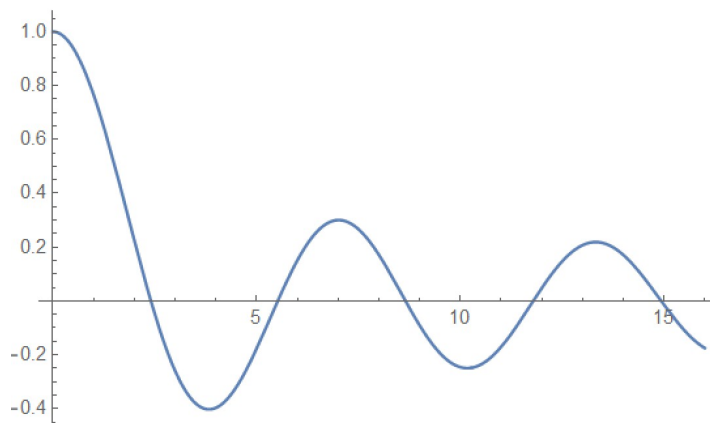
Beselova funkcija prve vrste reda n u programu Mathematica poziva se naredbom

$$\text{BesselJ}[n, x],$$

što će biti korišćeno u rešavanju zadatka. Funkcija je najpre predstavljena grafički na domenu $x > 0$, šireći opseg od x dok se ne pojavi 5 nula funkcije i dva lokalna minimuma.

```
Plot[BesselJ[0, x], {x, 0, 16}]
```

Variirajući opseg grafika prikazanog na slici 4.3 dobijena je gornja granica intervala koja zadovoljava uslove zadatka. Potom je sa njega očitani skup tačaka, bliskih nulama, kako bi one bile određene iz ovih inicijalnih pozicija.



Slika 4.3: Beselova funkcija prve vrste reda 0 na intervalu od 0 do 16.

```
(*skup tacaka inicijalnih za trazenje nula funkcije*)
set0={{2.261612491407254, -0.0057464007473573275'},
{5.405414252512371, -0.0008923023563278676},
{8.452977184195905, -0.0057464007473573275'},
{11.62885855510822, -0.0057464007473573275'},
{14.804739926020531, -0.010600499138386343'}};
set0 = set0[[All, 1]];
(*trazenje nula funkcije*)
roots=Table[
ini = set0[[i]];
FindRoot[BesselJ[0,x], {x, ini}], {i, Length[set0]}]
```

Prvih 5 nula zadate funkcije na $x > 0$, date kao skup *roots* pravila zamene $\{\{x \rightarrow 2.40483\}, \{x \rightarrow 5.52008\}, \{x \rightarrow 8.65373\}, \{x \rightarrow 11.7915\}, \{x \rightarrow 14.9309\}\}$. Slično postupku traženja nula funkcije, za nalaženje lokalnih minimuma potrebno je najpre sa grafika očitati vrednosti promenljive x u neposrednoj blizini traženog ekstremuma da bi proces traženja počeo iz što povoljnijeg položaja, čineći efikasnijim izvršenje naredbe.

```
(*selektovan skup tacaka inicijalnih za trazenje lokalnih minimuma*)
setminm={{3.3843988346590823, -0.3617429960627241},
{9.54368391764054, -0.19991033638851974'}};
setminm = setminm[[All, 1]];
(*trazenje minimuma funkcije*)
minms=Table[
ini = setminm[[i]];
FindMinimum[BesselJ[0, x], {x, ini}], {i, Length[setminm]}]
```

Dobijen je rezultat u obliku liste parova vrednost lokalnog minimuma i njegovog položaja

$$\left(\begin{array}{cc} -0.402759 & \{x \rightarrow 3.83171\} \\ -0.249705 & \{x \rightarrow 10.1735\} \end{array} \right).$$

4.3.69 Ajnštajnov model.

U Ajnštajnovom modelu podrazumeva se da je kristal ekvivalentan ansamblu nezavisnih jednodimenzionalnih harmonijskih oscilatora, koji vibriraju istom frekvencijom ν . Toplotni kapacitet

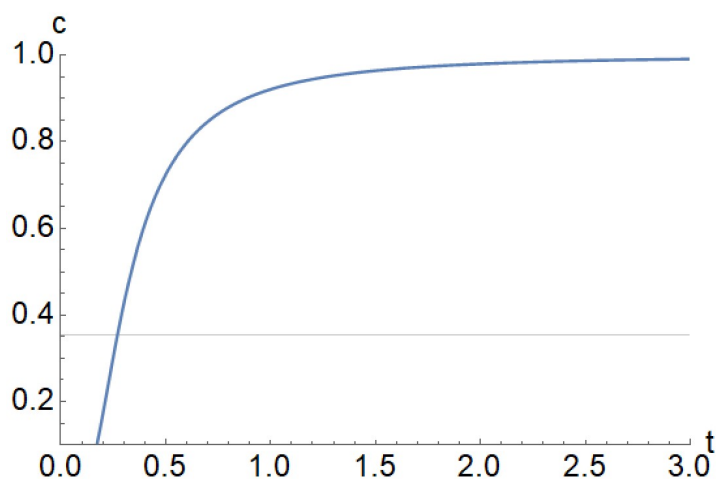
ovakvog sistema na konstantnoj zapremini C_v je zadat kao

$$C_v = 3Nk \left(\frac{\theta}{T} \right)^2 \frac{e^{\theta/T}}{(e^{\theta/T} - 1)} \quad (4.1)$$

gde je N broj atoma u kristalu, k je Bolcmanova konstanta, a θ je Ajnštajnova temperatura definsana jednačinom $k\theta = hv$, u kojoj je h Plankova konstanta. Eksperimentalna vrednost $C_v/3Nk$ iznosi 0.3552 na temperaturi $T = 358.5\text{K}$. Koristeći naredbu *FindRoot* odrediti θ za dijamant. Prilikom izrade koristiti smene: $c = C_v/3Nk$ i $t = T/\theta$. Potom nacrtati c u funkciji od t da bi se grafički odredila početna tačka od t , neophodna za definisanje inicijalne pozicije za traženje korena, argumenta od *FindingRoot*. Nakon izvršene smene promenljivih funkcija (4.1) postaje

$$c(t) = \frac{1}{t^2} \frac{e^{1/t}}{(e^{1/t} - 1)^2}.$$

```
Plot[(1/t^2)Exp[1/t]/(Exp[1/t]-1)^2, {t,0,20},
PlotRange->{{0,3},{0.3552,1}},AxesLabel->{"t", "c"},
LabelStyle->Directive[Black,14]]
```



Slika 4.4: Toplotni kapacitet u Ajnštajnovom modelu.

Sa grafika prikazanog na slici 4.4 uočava se da traženje nule funkcije $f(t) = \frac{e^{1/t}}{(e^{1/t}-1)^2} - 0.3552$,

može početi variranjem t od 0.2 (na grafiku je prikazana horizontalna linija 0.3552 koja seče grafik u traženoj tački).

```
(*trazenje nule funkcije f(t)*)
sol=FindRoot[(1/t^2)Exp[1/t]/(Exp[1/t]-1)^2-0.3552,{t,0.2}];
(*promenljivoj t0 dodeljena nula funkcije f(t)*)
t0=(t /. sol);
(*Debajeva temperatura iz uslova zadatka*)
358.5/t0
```

Rešenje: Debajeva temperatura za dijamant iznosi 1327.72 K.

4.3.70 Spektar energije vezanog stanja čestice u konačno-dubokoj potencijalnoj jami. Numeričko rešavanje transcendentne jednačine.

Čestica energije E se nalazi u konačno dubokoj potencijalnoj jami širine a i dubine V_0 , ($E < V_0$). Za ovaj sistem analitički je nađena trascendentna jednačina (3.24), čija rešenja pripadaju diskretnom energijskom spektru čestice u vezanom stanju. Brojne vrednosti dozvoljenih energija kvantnog sistema nalaze se rešavanjem ove jednačine numerički. Uvođenjem smene $z = \frac{\kappa a}{2}$ i njenom zamenom u (3.24) dobija se

$$\kappa = z \frac{2}{a} \tan z,$$

dok je sa druge strane κ izražena preko V_0, E i a dato kao

$$\kappa = \sqrt{\frac{2m(V_0 - E)}{\hbar^2}} = \sqrt{\frac{2mV_0}{\hbar^2} - k^2}.$$

Kombinovanjem gornjih jednačina nalazi se

$$\begin{aligned} z \tan z &= \frac{a}{2} \sqrt{\frac{2mV_0}{\hbar^2} - k^2}, \\ &= \sqrt{\frac{a^2 m V_0}{2\hbar^2} - \frac{(ak)^2}{4}} \\ &= \sqrt{\frac{a^2 m V_0}{2\hbar^2} - z^2}, \\ &= \sqrt{z_0^2 - z^2}, \end{aligned}$$

gde je $z_0^2 = \frac{a^2 m V_0}{2\hbar^2}$. Delenjem leve i desne strane gornje jednačine sa z dobija se

$$\tan z = \sqrt{\left(\frac{z_0}{z}\right)^2 - 1}. \quad (4.2)$$

Rešenja jednačine (4.2) biće nađena numerički uz pomoć *FindRoot*. Najpre se nacrtaju funkcije sa leve i desne strane jednakosti

$$y = \tan z \text{ i } y = \sqrt{\left(\frac{z_0}{z}\right)^2 - 1}$$

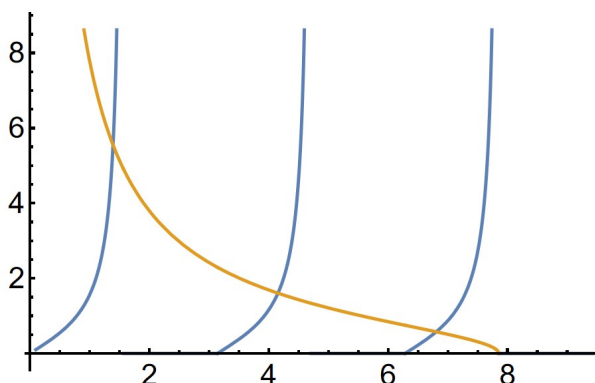
čiji su preseki rešenja trascendentne jednačine. Nula druge funkcije se nalazi u $z = z_0$, a budući da fizičko rešenje zahteva samo pozitivne vrednosti ovih funkcija, u toj oblasti desiće se konačan broj njihovih preseka koje treba naći.

Za zadatu vrednost z_0 kreirane su i nacrtane funkcije funkcije $y_1 = \sqrt{(z_0/z)^2 - 1}$ i $y_2 = \tan z$, a potom prikazan panel grafika sa kog se prikupljaju tačke bliske njihovim presecima uz pomoć naredbe *ClickPane*.

```
z0 = 5 Pi/2;
y1[z_] := Sqrt[(z0/z)^2 - 1];
y2[z_] := Module[{y},
  y = Tan[z];
  If[y > 0, y, 0]];
gr = Plot[{y2[z], y1[z]}, {z, 0.2, 3 Pi}];
ClickPane[gr, (xycoord = #) &]
```

Tačke prikupljene na panelu, klikom na grafik, moraju da budu blizu rešenja trascendentne jednačine, odnosno u okolini preseka funkcija. Proces se odvija dinamički zahvaljujući naredbi

$$\text{Dynamic}[xcoord],$$



Slika 4.5: Grafik preseka funkcija y_1 i y_2 koji služi na kom se prikupljaju početne tačke od kojih se traži presek.

što omogućava da se uzastopnim klikovima na željene tačke istovremeno prikazuju njihove koordinate, koje se dalje prikupljaju i ubacuju u listu *xycoords*.

```
Dynamic[xycoord]
```

```
xycoords = {{1.2881162269921902, 5.9294508286957175},
{3.838468512753291, 1.7145075157213547},
{6.657278933857666, 0.7021848859676867}};
```

Iz svakog para liste *xycoords* za početnu tačku uzme se prva koordinata, kao inicijalna u traženju rešenja

$FindRoot[Sqrt[(z0/z)^2 - 1] == Tan[z], \{z, xycoords[[i, 1]]\}]$. Dobijene transformacije iz funkcije *FindRoot* su zamenjene brojnim vrednostima $root = z /. root$, koja se dalje ubacuju $AppendTo[res, root]$ u prazan skup *res* formiran pre *Do* petlje.

```
res = {};
Do[
root = FindRoot[Sqrt[(z0/z)^2 - 1] == Tan[z], {z, xycoords[[i, 1]]}];
root = z /. root;
AppendTo[res, root],
{i, Length[xycoords]}];
```

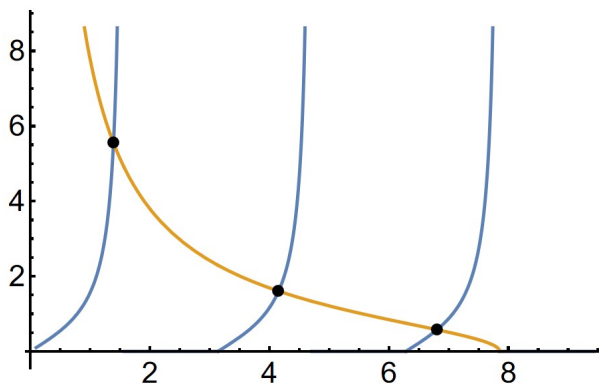
Dobijena rešenja transcendentne jednačine redom su uvršćena u funkciju y_1 , te je napravljena lista $\{\{x_1, y_1(x_1)\}, \dots\}$, koja je potom, provere radi nacrtana na istom grafiku zajedno sa funkcijama na datom opsegu.

```
Show[gr,
ListPlot[{\#, y1[\#]} & /@ res, PlotStyle -> {Black, PointSize[0.02]}]]
```

4.4 Numerička integracija

Postoje funkcije čiji se određeni integrali ne mogu rešiti analitički, što se upotrebom programa Mathematica prevazilazi upotrebom

$$NIntegrate[f, \{x, x_{min}, x_{max}\}],$$



Slika 4.6: Funkcije y_1 i y_2 prikazane sa tačkama njihovog preseka, rešenjima transcendentne jednačine.

koja daje numeričku aproksimaciju konačnog integrala

$$\int_{x_{min}}^{x_{max}} f(x) dx$$

funkcije $f(x)$ na intervalu $[x_{min}, x_{max}]$. Ovako se mogu pronaći aproksimacije nepravilnih integrala, pod uslovom da konvergiraju. Određen integral je nepravilan ako su jedna ili obe granice integracije beskonačne ili ako je integrand beskonačan u nekim izolovanim tačkama na intervalu integracije. Rešavanje nepravilnih integrala biće ilustrovano kroz nekoliko primera.

4.4.71 Izračunati određeni integral kada je jedna njegova granica beskonačna

$$\int_1^{\infty} \sin\left(\frac{1}{1+x^2}\right) dx.$$

```
NIntegrate[Sin[1/(1 + x^2)], {x, 1, Infinity}]
```

Rešenje: 0.778031.

4.4.72 Izračunati integral sa obe beskonačne granice

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2+x^3+x^8}} dx.$$

```
NIntegrate[1/Sqrt[2+x^3+x^8], {x, -Infinity, Infinity}]
```

Rešenje: 1.96609.

4.4.73 Izračunati integral čiji je integrand beskonačan na gornjoj granici

$$\int_{-1}^1 \sqrt{\frac{1+x^5}{1-x}} dx.$$

```
NIntegrate[Sqrt[(1 + x^5)/(1 - x)], {x, -1, 1}]
```

Rešenje: 3.07224.s

4.4.74 Izračunati integral čiji je integrand beskonačan na donjoj granici

$$\int_0^1 \frac{e^{-x}}{\sqrt{x}} dx.$$

```
NIntegrate[Exp[-x]/Sqrt[x], {x, 0, 1}]
```

Rešenje: 1.49365.

Naredbom

```
NIntegrate[f, {x, xmin, xmax}, {y, ymin, ymax}, ...]
```

traži se numerička aproksimacija višestrukog integrala

$$\int_{x_{min}}^{x_{max}} dx \int_{y_{min}}^{y_{max}} dy \dots f(x, y, \dots).$$

4.4.75 Ravnomerno površinski naelektrisan disk radijusa a nalazi se u XY ravni sa centrom u koordinatnom početku. Električni potencijal u tački (x, y, z) dat je kao rešenje integrala

$$V(x, y, z) = \frac{\sigma}{4\pi\epsilon_0} \int_0^a dr \int_0^{2\pi} d\varphi \frac{r}{\sqrt{(x-r\cos\varphi)^2 + (y-r\sin\varphi)^2 + z^2}}$$

gde je ϵ_0 konstanta permitivnosti. Izračunati V u tački $(a, 2a, 5a)$. U jedinicama $\sigma a/4\pi\epsilon_0$, zadati integral postaje

$$V(x, y, z) = \int_0^1 dr \int_0^{2\pi} d\varphi \frac{r}{\sqrt{(x-r\cos\varphi)^2 + (y-r\sin\varphi)^2 + z^2}},$$

dok je pri zadatoj zameni tačka u prostoru u kojoj se meri potencijal $(1, 2, 5)$.

```
x = 1;
y = 2;
z = 5;
NIntegrate[r/Sqrt[(x-r*Cos[fi])^2+(y-r*Sin[fi])^2+z^2],
{r,0,1},{fi,0,2 Pi}]
```

Rešenje: $0.570012\sigma a/4\pi\epsilon_0$.

4.4.76 **Aproksimacija određenog integrala.**

U narednom primeru biće upoređeni rezultati dobijeni aproksimacijom integrala

$$\int_0^{0.3} (1+x^4)^{1/3} dx$$

računatom tako što je integrand razvijen u red oko $x = 0$ do na stepen x^8 , a potom izvršena integracija (*Integrate*) na zadatom intervalu i njegove vrednosti numerički izračunate sa *NIntegrate*. Najpre je definisana funkcija $f(x) = (1+x^4)^{1/3}$ i granice integracije. Potom se funkcija razvija u red sa *Series*, nakon čega se naredbom *Normal* uzima njegov izraz bez $O(x^9)$. Dobijena funkcija *pwsrs* = $1 + \frac{x^4}{3} - \frac{x^8}{9}$ od x se dalje integrirali na intervalu od a do b sa *Integrate*[*pwsrs*, { x, a, b }]. Pored toga izvrši se i numerička integracija na istom intervalu.

```
a = 0;
b = 0.3;
f[x] :=(1+x^4)^(1/3);
pwsrs=Series[f[x],{x,0,8}]
pwsrs=Normal[pwsrs]
intapprx=Integrate[pwsrs,{x,a,b}]
intnum=NIntegrate[f[x],{x,a,b}];
Abs[intapprx-intnum]
```

Vrednost integracije datim metodama se razlikuje 7.53928×10^{-10} i iznose približno 0.300162.

- 4.4.77 **Moment inercije nehomogenog tela poznate geometrije.** Zapremina nehomogenog čvrstog tela opisana je jednačinama $4x^2 - y^2 + z^2 = 0$ i $y = 3a$. Ako je gustina u nekoj tački tela proporcionalna njenom rastojanju od koordinatnog početka i opisana sa $\rho(x, y, z) = k\sqrt{x^2 + y^2 + z^2}$, moment inercije tela oko y - ose se može izraziti preko integrala

$$I_y = \int_0^{3a} \int_{-y/2}^{y/2} \int_{-\sqrt{y^2-4x^2}}^{\sqrt{y^2-4x^2}} (x^2 + z^2)k\sqrt{x^2 + y^2 + z^2} dx dy dz.$$

Dati integral će pre rešavanja biti uprošćen. Naime zbog parnosti podintegralne funkcije, ovaj integral se može izraziti na sledeći način

$$I_y = 4 \int_0^{3a} dy \int_0^{y/2} dx \int_0^{\sqrt{y^2-4x^2}} dz (x^2 + z^2)k\sqrt{x^2 + y^2 + z^2}.$$

Zamenom promenljivih $x \rightarrow ax$, $y \rightarrow ay$, $z \rightarrow az$, moment inercije I_y postaje

$$I_y = 4ka^6 \int_0^{3a} dy \int_0^{y/2} dx \int_0^{\sqrt{y^2-4x^2}} dz (x^2 + z^2)\sqrt{x^2 + y^2 + z^2}$$

```
4 k a^6 NIntegrate[(x^2+z^2)Sqrt[x^2+y^2+z^2],
{y,0,3},{x,0,y/2},{z,0,Sqrt[y^2 - 4 x^2]}]
```

Rešenje: $72.5991a^6k$.

- 4.4.78 **Normiranje talasne funkcije vezanog stanja čestice u 1D konačno-dubokoj potencijalnoj jami.**

Za vezanu česticu mase m koja se nalazi u konačno dubokoj potencijalnoj jami širine $a = 1$ i dubine V_0 ($z_0 = \sqrt{\frac{a^2 m V_0}{2\hbar^2}} = 5\pi/2$) iskoristiti nađeni spektr z_n , dobijen numeričkim rešavanjem transcendentne jednačine (4.2), a potom nacrtati normirane talasne funkcije $\psi(x)$ za 3 nađena vezana stanja

$$\psi(x) = \begin{cases} A_1 e^{x\kappa} & x < -\frac{a}{2} \\ B_2 e^{-x\kappa} & x > \frac{a}{2} \\ A_0 \sin kx + B_0 \cos kx & x \in \left(-\frac{a}{2}, \frac{a}{2}\right) \end{cases}.$$

Na osnovu oblika potencijala može se zaključiti da će talasna funkcija biti parna, to jeste važiće $\psi(x) = \psi(-x)$, $\forall x \in \mathbb{R}$. Odavde sledi da su $B_2 = A_1$ i $A_0 = 0$. Najpre će u prvoj ćeliji programa, uz dobijenu listu vrednosti z_n , biti definisane talasne funkcije na pojedinim intervalima x , određenih u odnosu na tačke konačnog skoka potencijala

$$\psi(x) = \begin{cases} A_1 e^{x\kappa} & x < -\frac{a}{2} \\ A_1 e^{-x\kappa} & x > \frac{a}{2} \\ B_0 \cos kx & x \in \left(-\frac{a}{2}, \frac{a}{2}\right) \end{cases}.$$

```
z={1.3925490876084483, 4.154939709055357, 6.805761140682553};
\[\Psi]0[x_]:=B0*Cos[k x];
\[\Psi]1[x_]:=A1*E^(x\[Kappa]);
\[\Psi]2[x_]:=A1*E^(-x\[Kappa]);
```

Sa n je selektovano odgovarajuće stanje, odnosno energija iz dobijenog diskretnog spektra za date vrednosti z_0 i a . Potom se računaju odgovarajuće vrednosti za $k = 2z_n/a$ i $\kappa = (2/a)\sqrt{z_0^2 - z_n^2}$.

Iz uslova neprekidnosti talasne funkcije $\psi_0(-a/2) = \psi_1(-a/2)$ nalazi se veza između koeficijenta A_1 i B_0 , rešavajući algebarsku jednačinu po B_0 na sledeći način
 $pom = Solve[\psi_0[-a/2] == \psi_1[-a/2], B_0][[1]]$. Uz odgovarajuće uslove, povežu se funkcije ψ_0, ψ_1, ψ_2 u jednu ψ , čiji grafik se u nastavku prikazuje.

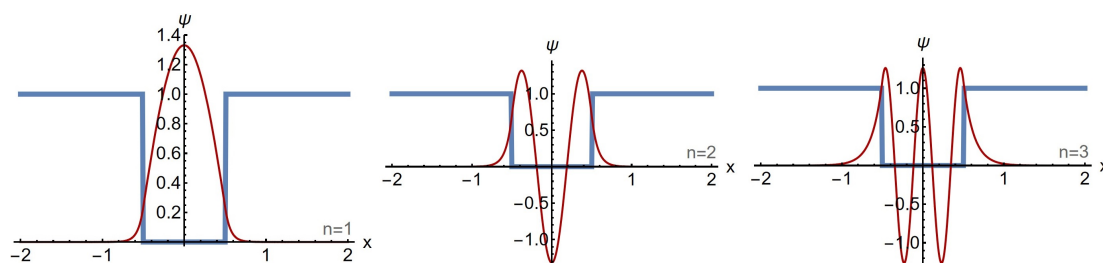
```
n=3;(*izbor stanja*)
a=1;(*sirina jame*)
z0=5 Pi/2;
\Kappa=Sqrt[(z0^2 - z[[n]]^2)] 2/a;
k=2 z[[n]]/a;
(*uslov neprekidnosti- daje vezu izmedju konstanti*)
pom=Solve[\Psi0[-a/2]==\Psi1[-a/2],{B0}][[1]];
\psi[x_]:=Which[x < (-a/2),\Psi1[x],x >a/2,\Psi2[x],
True,\Psi0[x]] /. pom;
```

U funkciji ψ_0 na osnovu uslova neprekidnosti, izvršena je zamena, pri čemu je B_0 izražena preko A_1 u liniji $psi_0/.pom$. Potom je talasna funkcija normirana. Najpre je na svim segmentima talasne funkcije izvršena zamena $A_1 \rightarrow 1$, a potom je numerički izračunat integral kvadrata dobijene funkcije na pojedinim intervalima definisanih u odnosu na tačke skoka potencijala.

```
(*normiranje*)
norm=NIntegrate[\Psi1[x]^2 /.{A1->1},{x,-Infinity,-a/2}]+
NIntegrate[\Psi2[x]^2 /.{A1->1},{x,a/2,Infinity}]+
NIntegrate[(\Psi0[x]/. pom)^2 /.{A1->1},{x,-a/2,a/2}];
norm = Sqrt[norm];
```

Za svaku energiju vezanog stanja crvenom bojom nacrtana je normirana talasna funkcija
 $gpsi = Plot[(\psi[x]/norm)/. {A1->1}, {x,-2,2}, PlotRange->All, PlotStyle->Darker[Red]]$. Radi što bolje ilustracije talasne funkcije i predikcije gustine stanja, gde je očekivana vrednost nalaženja čestice mala, ψ je prikazana zajedno sa potencijalnom jamom crtanom kao uz pomoć $ListLinePlot$, kojim se povezuju tačke $welli$ duž linije potencijala
 $gwell = ListLinePlot[welli, PlotStyle->Thickness[0.015]]$;

```
(*funkcija pravougaone jame*)
well[x_]:=Which[x<-1/2,1,x>1/2,1,True,0];
welli=Table[{x,well[x]},{x,-2,2,0.01}];
gwell=ListLinePlot[welli,PlotStyle->Thickness[0.015]];
gpsi=Plot[(\Psi[x]/norm)/. {A1->1},{x,-2,2},PlotRange->All,
PlotStyle->Darker[Red]];
Show[gwell,gpsi,PlotRange->All,
AxesStyle->Directive[Black,Thickness[0.005],18],
AxesLabel->{"x","\Psi"},PlotLabel->Style["n="<>ToString[n],18]]
```



Slika 4.7: Talasne funkcije osnovnog, prvog i drugog pobuđenog stanja čestice u 1D konačno-dubokoj pravougaonoj potencijalnoj jami pri zadatim parametrima $a = 1$, $z_0 = 5\pi/2$.

4.4.79 **Toplotni kapacitet kristala.**

Prema Debajevoj teoriji, toplotni kapacitet čvrstih tela dat je relacijom

$$C_v = 9kN \left(\frac{T}{\theta} \right)^3 \int_0^{\theta/T} \frac{x^4 e^x}{(e^x - 1)^2} dx,$$

gde je θ Debajeva temperatura, N broj atoma, k Bolcmanova konstanta, a T apsolutna temperatura. Nacrtati $C_v/3Nk$ u funkciji od T/θ u opsegu od 0.001 do 1.4 i adekvatno označiti ose.

Veza između energije fonona ε , ugaone frekvencije i talasnog vektora \vec{q} je

$$\varepsilon = \hbar\omega, \quad \omega = v_s |\vec{q}| \quad (4.3)$$

gde je v_s brzina zvuka. Kao bozoni, fononi su opisani Boze-Ajnštajnovom (BA) statistikom, te je očekivan broj popunjenosti stanja energije E

$$n_E = \frac{1}{e^{E/kT} - 1} = \frac{1}{e^{\hbar\omega/kT} - 1} \quad (4.4)$$

pri čemu je $k = 1.3806504 \cdot 10^{-23} \text{ J/K}$ Bolcmanova konstanta. Gornja frekvencija, poznata kao Debajeva određena je na sledeći način. Iz zapremine elementarne ćelije u trodimenzionalnom recipročnom prostoru

$$\left(\frac{2\pi}{L} \right)^3 = \frac{8\pi^3}{V} \quad (4.5)$$

gde postoji gornja granica talasnog vektora $q_D = \omega_D/v_s$ sve mode se nalaze u ograničenom delu prostora, sferi radijusa q_D , te je broj moda

$$N = \left(\frac{4}{3} \pi q_D^3 \right) / (8\pi^3/V), \quad q_D = \left(6\pi^2 \frac{N}{V} \right)^{1/3}, \quad \omega_D = v_s \left(6\pi^2 \frac{N}{V} \right)^{1/3}. \quad (4.6)$$

Prema definiciji Debajeva temperatura je

$$T_D = \frac{\hbar\omega_D}{k} = \frac{\hbar v_s}{k} \left(6\pi^2 \frac{N}{V} \right)^{1/3}. \quad (4.7)$$

Debajeva aproksimacija podrazumeva konstantnu brzinu zvuka v_s svake polarizacije bilo da je longitudinalna ili transferzalna, što odgovara klasičnom elastičnom kontinuumu. Gustina stanja prema definiciji i u skladu sa rezultatom pod (4.6) je

$$D(\omega) = \frac{dN}{d\omega} = \frac{V\omega^2}{2\pi^2 v_s^3}. \quad (4.8)$$

Izraz za termalnu enegiju, koja uključuje mode svih mogućih frekvencija jedne fononske grane (polarizacije) (do Debajeve) (4.6) uz prethodnu relaciju za gustinu stanja (4.8) i (BA) distribuciju (4.4) postaje

$$U = \int D(\omega) n(\omega) \hbar\omega d\omega = \int_0^{\omega_D} \frac{V\omega^2}{2\pi^2 v_s^3} \frac{\hbar\omega}{e^{\hbar\omega/kT} - 1} d\omega. \quad (4.9)$$

Iz pretpostavke da se brzine 3 akustičke grane (1 longitudinalne i 2 transferzalne) ne razlikuju, ukupna unutrašnja energija je (4.9) multiplikovana je za 3

$$U = \frac{3V\hbar}{2\pi^2 v_s^3} \int_0^{\omega_D} d\omega \frac{\omega^3}{e^{\hbar\omega/kT} - 1} \quad (4.10)$$

Simbolički nađe se izvod podintegralne funkcije po T .

```
D[\[Omega]^3/(Exp[h \[Omega]/(k*t)] - 1), t]
```

Prema definiciji toplotni kapacitet je

$$C_v = \frac{\partial U}{\partial T} = \frac{3V\hbar}{2\pi^2v_s^3} \int_0^{\omega_D} d\omega \frac{\hbar\omega^4 e^{\frac{\hbar\omega}{kT}}}{kT^2 \left(e^{\frac{\hbar\omega}{kT}} - 1 \right)^2}. \quad (4.11)$$

Nakon uvođenja smene $x = \hbar\omega/kT$ u izraz za toplotni kapacitet (4.11) dobije se

$$C_v = \frac{3V}{2\pi^2v_s^3} \frac{k^4 T^3}{\hbar^3} \int_0^{x_D} dx \frac{x^4 e^x}{(e^x - 1)^2}, \quad (4.12)$$

gde je $x_D = T_D/T$. Ako se iz (4.7) izrazi zapremina V a potom zameni u izrazu (4.12) ispred integrala

```
sol1=Solve[TD==(h vs/k) (6 Pi^2 Nn/V)^(1/3), V][[1]];
expr=(3 V k^4 T^3)/(2 Pi^2 vs^3 h^3);
FullSimplify[expr /. sol1[[1]]]
```

i dalje uproščavanjem nastalog izraza dobije se $\frac{9kNnT^3}{TD^3}$. Prethodnim simboličkim sređivanjem nalazi se konačan izraz za toplotni kapacitet

$$C_v = 9Nk \left(\frac{T}{T_D} \right)^3 \int_0^{x_D} dx \frac{x^4 e^x}{(e^x - 1)^2}, \quad (4.13)$$

Smenom $t = T/T_D$, $t \in [0.001, 1.4]$ i $c = C_v/(3Nk)$ dobija se funkcija koja se prikazuje grafički

$$c = t^3 \int_0^{1/t} dx \frac{x^4 e^x}{(e^x - 1)^2},$$

što je isprogramirano u sledećoj ćeliji.

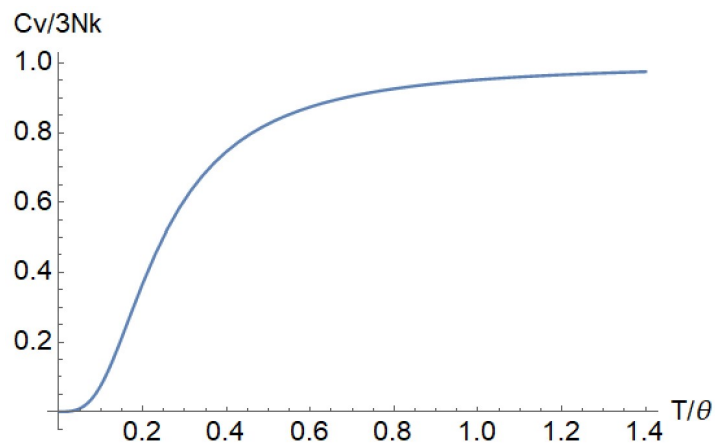
```
Plot[
  3t^3 NIntegrate[(x^4) Exp[x]/(Exp[x]-1)^2,
    {x, 0, 1/t}],
  {t, 0.001, 1.4},
  AxesLabel -> {"T/\[Theta]", "Cv/3Nk"},
  LabelStyle -> Directive[Black, 14]]
```

4.5 Rešavanje diferencijalnih jednačina numerički

Naredbe programa Mathematica kojima se analitički rešavaju diferencijalne jednačine nisu primenjive na sve tipove jednačina koje se javljaju prilikom rešavanja zadataka. Zbog toga se neretko pribegava upotrebi funkcija kojima se diferencijalne jednačine rešavaju numeričkim metodama, čime se dobijaju funkcije potpuno određene na datom intervalu. Prilikom zadanja, pored jednačine potrebno je zadati odgovarajuće uslove kao i interval promenljive na kom se traži rešenje. Numeričko rešavanje jednačine zajedno sa početnim uslovima eq u programu Mathematica radi se pomoću funkcije

$$NDSolve[eq, y, \{x, x_{min}, x_{max}\}],$$

gde je tražena funkcija $y(x)$, a x nezavisna varijabla, čija se vrednost uzima od x_{min} do x_{max} .



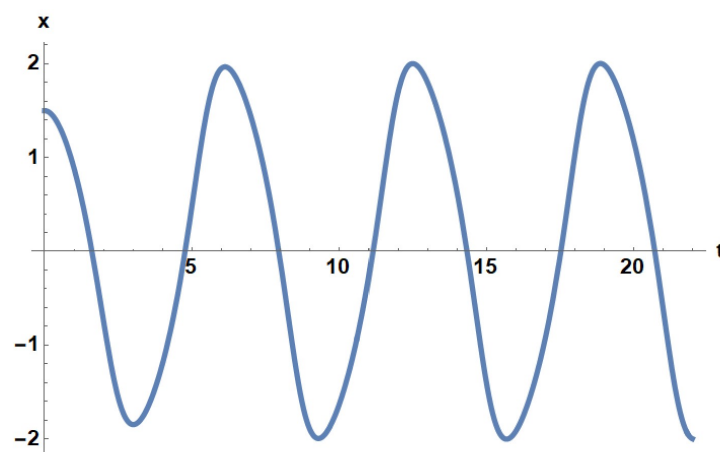
Slika 4.8: Zavisnost toplotnog kapaciteta od temperature u Debajevom modelu.

Primera radi biće rešena, analitički nerešiva jednačina kretanja Polovog oscilatora

$$\frac{dx^2}{dt^2} - \varepsilon(1 - x^2)\frac{dx}{dt} + x = 0,$$

gde je $\varepsilon > 0$. Diferencijalnu jednavčinu rešiti uz početne uslove $v(0) = 0$ i $x(0) = 1.5$, kada je $\varepsilon = 0.5$ za $t \in [0, 7\pi]$.

```
(*pocetni uslovi diferencijalne jednacine*)
init={x[0]==1.5,x'[0]==0};
(*parametar*)
\[Epsilon]=0.5;
(*diferencijalna jednacina zadata kao element skupa*)
eq={x'[t]-\[Epsilon](1-x[t]^2)x'[t]+x[t]==0};
(*resavanje diferencijalne jednacine*)
sol=NDSolve[Union[eq,init],x,{t,0,7Pi}];
(*pravilom zamene pridruzi se interpolirana funkcija*)
xt=x/.sol[[1]];
Plot[xt[t],{t,0,7Pi},AxesLabel->{"t","x"}]
```



Slika 4.9: Položaj Polovog oscilatora za početne uslove $v(0) = 0$, $x(0) = 1.5$ i parametar $\varepsilon = 0.5$.

Da bi se našlo jedinstveno rešenje diferencijalne jednačine drugog reda potrebno je zadati dva početna uslova, datih kao elemente skupa *init*. Nakon zadavanja parametra ε , zadata je i jednačina kao element skupa *eq*, a njegova unija sa *init* čini prvi argument naredbe *NDSolve*, gde se definišu jednačine i početni uslovi. Drugi argument je funkcija $x(t)$ koja se traži, a treći sadrži promenljivu i interval na kom se traži nepoznata funkcija. Rezultat *sol* je dvostruka lista, čiji element je pravilo zamene funkcije x , numerički dobijenom interpolacionom funkcijom.

Numeričko rešenje diferencijalne jednačine je lista parova (t_i, x_i) , $i = 1, 2, \dots$ na zadanom intervalu t . Mathematica daje rešenje u vidu interpolirane funkcije, koja je aproksimacija prave funkcije, a rezultat je interpolacije skupa liste parova

$$\{\{x \rightarrow \text{InterpolatingFunction}[\]\}\}.$$

Vrednost funkcije, dobijene kao rešenje diferencijalne jednačine, može se tražiti u bilo kojoj tački intervala zadanog prilikom postavljanja, te se na njemu može prikazati grafički.

Istom naredbom u programu Mathematica numerički se na sličan način može rešiti sistem diferencijalnih jednačina dat u vidu skupa *eqs*, koji je prvi argument naredbe *NDSolve*. Drugi argument je skup funkcija $\{y_1, y_2, \dots\}$ koje se traže, a treći argument sadrži definiciju nezavisno promenljive zajedno sa intervalom na kom se rešava sistem $\{t, t_{min}, t_{max}\}$

$$\text{NDSolve}[\text{eqs}, \{y_1, y_2, \dots\}, \{t, t_{min}, t_{max}\}].$$

4.5.80 Koristeći naredbe *NDSolve* rešiti diferencijalne jednačine, a dobijena rešenja funkcija nacrtati na zadanom opsegu.

- i) $\frac{d^2y}{dx^2} = 2x + y + 3\frac{dy}{dx}$ sa graničnim uslovima $y(2) = 1$ i $y'(2) = -1$ na opsegu $x \in [2.0, 2.3]$

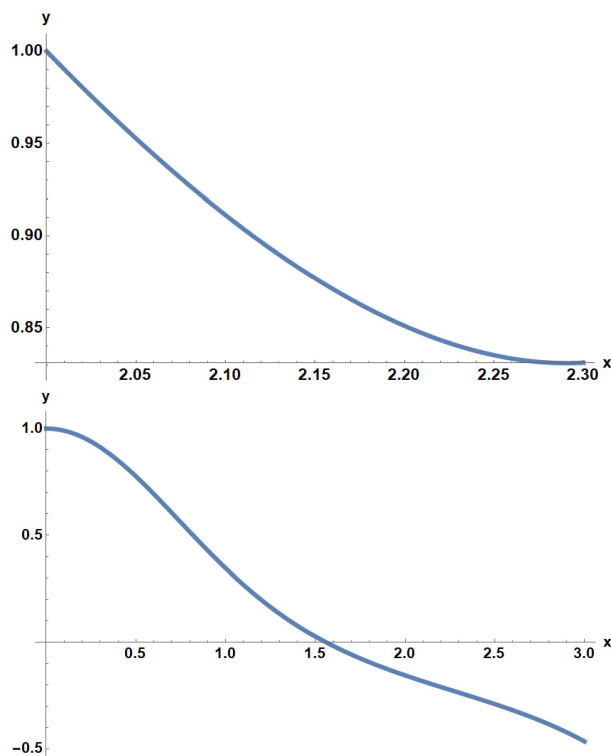
```
(*postavljena jednacina element skupa eqi*)
eqi={y''[x]==2x+y[x]+3y'[x]};
(*granicni uslovi elementi skupa initi*)
initi={y[2]==1,y'[2]==-1};
(*resavanje diferencijalne jednacine na zadanom intervalu*)
soli=NDSolve[Union[eqi, initi],y,{x,2,2.3}];
yi=y/.soli[[1]];
Plot[yi[x],{x,2,2.3},PlotStyle->Thickness[0.008],
AxesLabel->{"x","y"},LabelStyle->Directive[Black, 14, Bold]]
```

- ii) $\frac{d^2y}{dx^2} + \sin^2(x)\frac{dy}{dx} + 3y^2 = e^{-x^2}$ sa $y(0) = 1$ i $y'(0) = 0$. Naći $y(1)$, $y(2)$, $y(3)$ i nacrtati $y(x)$ na intervalu od 0 do 3.

```
(*postavljena jednacina element skupa eqii*)
eqii={y''[x]+Sin[x]^2y'[x]+3y[x]^2==Exp[-x^2]};
(*granicni uslovi elementi skupa initii*)
initii={y[0]==1,y'[0]==0};
(*resavanje diferencijalne jednacine na zadanom intervalu*)
solii=NDSolve[Union[eqii,initii],y,{x,0,3}];
yii=y/.solii[[1]];
{yii[1],yii[2],yii[3]}
Plot[yii[x],{x,0,3},PlotStyle->Thickness[0.008],
AxesLabel->{"x","y"},LabelStyle->Directive[Black,14,Bold]]
```

Rešenje: vrednosti funkcije $y(x)$ u tačkama $x=1, 2$ i 3 su $\{0.345837, -0.155485, -0.465162\}$.

- iii) $\frac{d^2y}{dx^2} + 5x\frac{dy}{dx} - (1-x^3)y = 0$ sa graničnim uslovima $y(0) = 0$ i $y'(0) = 1$. Odrediti $y(1.8)$ i nacrtati funkciju $y(x)$ na intervalu od 0 do 4.



Slika 4.10: Grafici funkcija na zadatim intervalima dobijenih rešavanjem diferencijalnih jednačina pod i) levo i pod ii) desno.

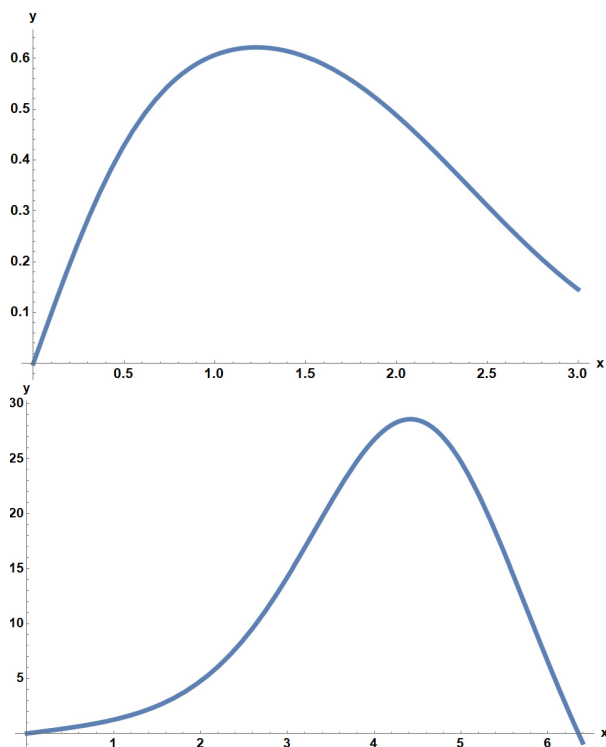
```
(*postavljena jednačina element skupa eqiii*)
eqiii={y''[x]+5 x*y'[x]-(1-x^3)*y[x]==0};
(*granicni uslovi elementi skupa initiii*)
initiii={y[0]==0,y'[0]==1};
(*resavanje diferencijalne jednačine na zadatom intervalu*)
soliii=NDSolve[Union[eqiii, initiii], y, {x, 0, 4}];
yiii=y /.soliii[[1]];
yiii[1.8]
Plot[yiii[x],{x,0,3},PlotStyle -> Thickness[0.008],
  AxesLabel->{"x","y"}, LabelStyle->Directive[Black, 14, Bold]]
```

Rešenje: $y(1.8) = 0.544786$.

- iv) $y'' = y \sin(t) + t$ sa početnim uslovima $y(0) = 0$ i $y'(0) = 1$. Narctati $y(t)$ u opsegu od 0 do 6.4.

```
(*postavljena jednačina element skupa eqiv*)
eqiv={y''[t]==y[t] Sin[t]+t};
(*granicni uslovi elementi skupa initiv*)
initiv={y[0]==0,y'[0]==1};
(*resavanje diferencijalne jednačine na zadatom intervalu*)
soliv=NDSolve[Union[eqiv,initiv],y,{t,0,6.4}];
yiv=y /.soliv[[1]];
Plot[yiv[x],{x,0,6.4}, PlotStyle -> Thickness[0.008],
  AxesLabel -> {"x", "y"}, LabelStyle -> Directive[Black, 14, Bold]]
```

4.5.81 Planeta i Sunce.



Slika 4.11: Grafici funkcija dobijenih rešavanjem diferencijalnih jednačina na zadatim intervalima pod iii) levo i pod iv) desno.

Intenzitet sile kojom Sunce privlači neku planetu dat je sa

$$F = \gamma \frac{Mm}{r^2},$$

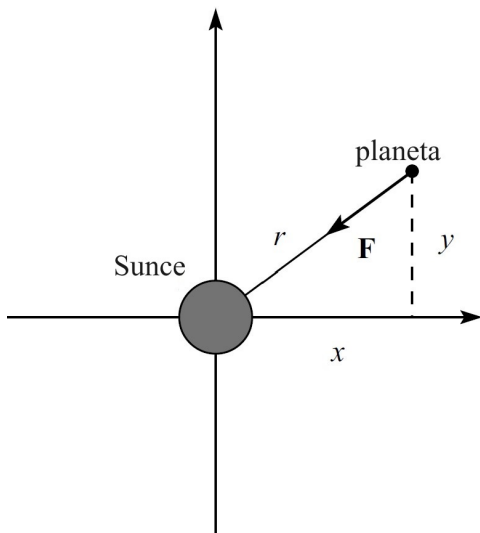
gde je γ gravitaciona konstanta, a r rastojanje između Sunca i planete masa M i m redom. Dinamičke jednačine kojima je opisano kretanje nebeskih tela u XY ravni prema Njutnovom zakonu su

$$\frac{d^2x}{dt^2} + \frac{\gamma Mx}{(x^2 + y^2)^{3/2}} = 0$$

$$\frac{d^2y}{dt^2} + \frac{\gamma My}{(x^2 + y^2)^{3/2}} = 0$$

pri čemu se smatra da je Sunce miruje i nalazi se u centru koordinatnog sistema. Za izradu zadatka koristi se astronomska jedinica (AU) dužine, jednaka srenjem rastojanju Zemlje i Sunca i jedinice za vreme jednake 1 godini, a $\gamma M = 4\pi^2$. Diferencijalne jednačine kretanja rešiti za $x(t)$ i $y(t)$ za parametar $t \in [0, 1.6]$. Narctati trajektoriju kretanja planete, pod uslovima $x(0) = 1$, $y(0) = 0$, $x'(0) = -\pi$ i $y'(0) = 2\pi$.



Najpre je definisana lista *eqs* diferencijalnih jednačina po osama iz vektorske diferencijalne jednačine drugog reda, a potom i lista početnih uslova, koordinata i komponenti brzina u trenutku $t = 0$. Sistem jednačina zajedno sa početnim uslovima čineći jedinstvenu listu *eqs*, formiran je naredbom *Union*.



```

eqs={x''[t]+4Pi^2x[t]/(x[t]^2+y[t]^2)^(3/2) == 0,
y''[t]+4Pi^2y[t]/(x[t]^2+y[t]^2)^(3/2)==0};
initial={x[0]==1,x'[0]==-Pi,y[0]==0,y'[0]==2Pi};
r={x[t], y[t]};
eqs=Union[eqs, initial];
sol=NDSolve[eqs,{x, y},{t,0,1.6}][[1]];
xy=r/.sol;
ParametricPlot[xy,{t,0,1.6},AxesLabel->{"x(AU)", "y(AU)"}]
    
```

Dobijena rešenja su transformaciona pravila, koja vode u interpolirane funkcije zavisnosti koordinata tokom vremena zadanog intervala.

{ InterpolatingFunction [ Domain: {{0., 1.6}} Output: scalar] [t],
 InterpolatingFunction [ Domain: {{0., 1.6}} Output: scalar] [t] }

Slika 4.12: Lista interplacionih funkcija, dobijenih kao rezultat numeričkog rešavanja sistema diferencijalnih jednačina na zadanom intervalu vremena.

```

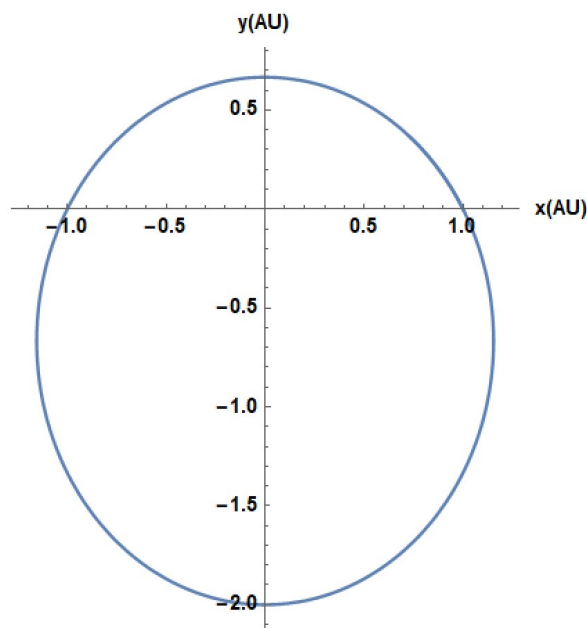
ParametricPlot[xy, {t, 0, 1.6}, AxesLabel -> {"x(AU)", "y(AU)"}]
    
```

4.5.82 **Realno matematičko klatno.**

Kuglica malih dimenzija mase m okačena o laku neistegljivu nit dužine l nalazi se u gravitacionom polju Zemlje g . Grafički uporediti vremenske zavisnosti elongacije matematičkog klatna radeći u aproksimaciji malih uglova i potpunim numeričkim rešavanjem diferencijalne jednačine kretanja. Ilustrovati zavisnost ugla otklona kuglice od vremena i u slučaju kada postoji otpor vazduha.

Jednačina dinamike rotacije koja važi za matematičko klatno

$$I\alpha = -mgl \sin \varphi,$$



Slika 4.13: Orbita planete oko sunca dobijena numeričkim rešavanjem jednačina kretanja.

gde su m, g, l masa gravitaciono ubrzanje i dužina klatna, $I = ml^2$ moment inercije klatna, $\alpha = \ddot{\varphi}$ ugaono ubrzanje a φ ugao otklona od vertikalne. Njenim sređivanjem dobije se diferencijalna jednačina, čije rešenje će biti traženo numerički

$$\ddot{\varphi} + \frac{g}{l} \sin \varphi = 0.$$

Za male uglove otklona važi aproksimacija $\sin \varphi \approx \varphi$, te gornja jednačina postaje

$$\ddot{\varphi} + \frac{g}{l} \varphi = 0,$$

čije je rešenje za klatno koje je iz mirovanja ($\dot{\varphi} = 0$) otklonjeno za $\varphi = \varphi_0$ pušteno da osciluje

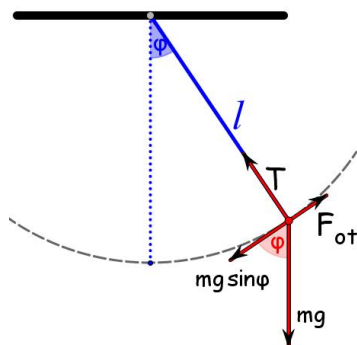
$$\varphi(t) = \varphi_0 \cos \omega t.$$

```
tint = 20; (*vreme oscilovanja u arbitrarnim jedinicama*)
eq = {\[CurlyPhi]''[t] + (g/l) Sin[\[CurlyPhi][t]] == 0};
initial = {\[CurlyPhi][0] == \[CurlyPhi]0, \[CurlyPhi]'[0] == 0};
eqs = Union[eq, initial];
values = {m -> 0.03, g -> 10, \[CurlyPhi]0 -> Pi/4, l -> 1.9};
eqs = eqs /. values;
sol = NDSolve[eqs, \[CurlyPhi], {t, 0, 20}];
fi = \[CurlyPhi] /. sol[[1]]
```

Promenljivoj $fi0$ dodeljen je izraz $\varphi_0 \cos \omega t$, gde je ω zamenjeno pravilom transformacije $\omega \rightarrow \sqrt{g/l}$. Na zadatom intervalu $t \in [0, tint]$ nacrtane su plavom zavisnosti elongacije od vremena u aproksimaciji malih otklona

`Plot[fi/.values, {t,0,tint}, PlotStyle -> Darker[Blue]]` i obojeno crvenom numerički dobijene elongacije, bez aproksimacije,

`Plot[fi[t], {t,0,tint}, PlotStyle -> Darker[Red]]`. Da bi se što lakše uočilo odstupanje numeričkog od aproksimativnog rešenja, zavisnosti otklona od vremena prikazani su na istom grafiku.

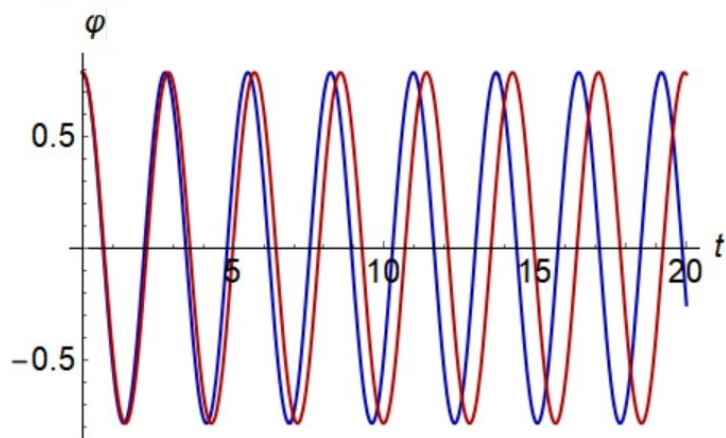


Slika 4.14: Model matematičkog klatna.

```

fi0 = \[CurlyPhi]0*Cos[\[Omega] t] /. {\[Omega] -> Sqrt[g/l]};
Show[Plot[fi0 /. values, {t, 0, tint}, PlotStyle -> Darker[Blue]],
Plot[fi[t], {t, 0, tint}, PlotStyle -> Darker[Red]],
AxesLabel -> {t, \[Omega]}, AxesStyle -> Directive[Black, 16]]

```



Slika 4.15: Ugao otklona matematičkog klatna u zavisnosti od vremena dobijeno u aproksimaciji malih otklona (plava linija) i numerički bez aproksimacije (crvena linija).

Dinamička jednačina matematičkog klatna $v = l\dot{\varphi}$

$$I\alpha = -mgl \cos \varphi - F_{ot}l,$$

na koje deluje sila otpora vazduha srazmerna brzini kretanja kuglice $F_{ot} = \beta l\dot{\varphi}$ i usmerena suprotno od smera kretanja. Kad se gornji izraz za silu otpora zameni u jednačini kretanja klatna dobije se konačna diferencijalna jednačina drugog reda

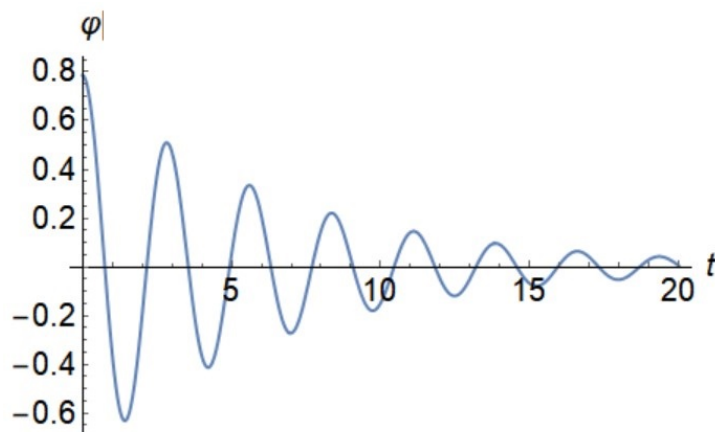
$$\ddot{\varphi} + \gamma\dot{\varphi} + \frac{g}{l} \sin \varphi = 0,$$

gde je $\gamma = \frac{\beta}{m}$ a rešenje se traži numerički.

```

eq ={\[CurlyPhi]''[t] +\[Gamma]*\[CurlyPhi]'[t]+
(g/l) Sin\[CurlyPhi][t] == 0};
initial={\[CurlyPhi][0]==\[CurlyPhi]0,\[CurlyPhi]'[0]==0};
eqs=Union[eq, initial];
values={m->0.03,g->10,\[CurlyPhi]0->Pi/4,l->1.9,\[Gamma]->0.3};
eqs=eqs /. values;
sol=NDSolve[eqs, \[CurlyPhi], {t, 0, 20}];
fi =\[CurlyPhi] /. sol[[1]];
Plot[fi[t],{t,0,tint},AxesLabel->{t,InputForm\[CurlyPhi][rad]}],
AxesStyle -> Directive[Black, 16]]

```



4.5.83 Stanja 1D kvantnog harmonijskog oscilatora.

Za jednodimenzionalni kvantni harmonijski oscilator, rešavanje vremenski nezavisne Šredingerove jednačine se svodi na rešavanje

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} + \frac{1}{2}m\omega^2 x^2\psi = E\psi.$$

Smenom promenljivih, menjajući varijablu x bezdimenzionom $\zeta = \sqrt{\frac{m\omega}{\hbar}}x$, Šredingerova jednačina postaje

$$\frac{d^2\psi}{d\zeta^2} = (\zeta^2 - 2n - 1)\psi,$$

gde je $n = \frac{E}{\hbar\omega} - \frac{1}{2}$, $n = 0, 1, 2, \dots$. Za nalaženje jedinstvenog rešenja ove diferencijalne jednačine potrebna su dva uslova. U slučaju kada je n neparno vrednosti talasne funkcije i njenog prvog izvoda u $x = 0$ iznose $\psi(0) = 0$ i $\psi'(0) = 1$, dok za parne vrednosti n uslovi iz kojih se dobiju jedinstvena rešenja talasne funkcije su $\psi(0) = 1$ i $\psi'(0) = 0$.

- i) Za $n = 6$ rešiti datu jednačinu po ζ u opsegu od -5 do 5. Nacrtati grafik talasne funkcije na opsegu $\zeta \in [-5, 5]$ i naći njenu vrednost $\psi(-5)$ i $\psi(5)$.
- ii) Za $n = 1, 2, 3, 4, 5$ nacrtati talasne funkcije $\psi(\zeta)$ i gustine verovatnoće nalaženja čestice na intervalu $\zeta \in [-5, 5]$

Primeru radi, za $n = 6$ kada su rešenja parne funkcije, definiše se diferencijalna jednačina drugog reda *eqLHO*, a potom i odgovarajući uslovi u $\zeta = 0$, vrednost funkcije i njen prvi izvod kao elemente liste *incond*. Formira se lista uslova zajedno sa diferencijalnom jednačinom *eqLHO*. Ova lista je prvi argument od *NDSolve*, funkcije programa Mathematica za numeričko rešavanje običnih diferencijalnih jednačina

$$NDSolve[eqLHO, \psi, \{\zeta, \zeta_{min}, \zeta_{max}\}]$$

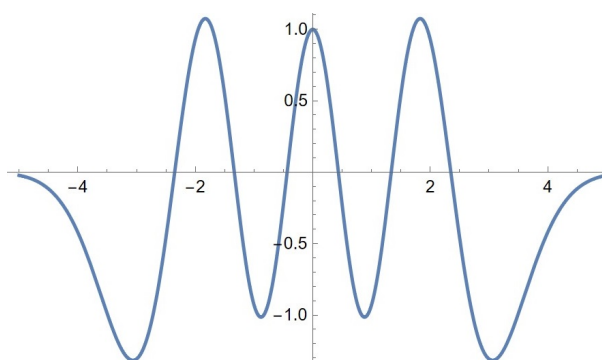
čije je rešenje funkcija $\psi(x)$ promenljive x na intervalu $\zeta \in [\zeta_{min}, \zeta_{max}]$. Dobijeni rezultat je lista *sol*, koja sadrži interpolirane vrednosti tražene funkcije ψ na zadatom intervalu.

```

eqLHO = \[Psi]''\[Xi] - (\[Xi]^2 - 2 n - 1) \[Psi][\[Xi]] == 0
(*diferencijalna jednacina II reda*)
incond = {\[Psi][0] == 1, \[Psi]'[0] == 0};
(*pocetni uslovi za talasnu funkciju*)
eqLHO = Union[{eqLHO}, incond] /. {n -> 6};
sol = NDSolve[eqLHO, \[Psi], {\[Xi], -7, 7}];
psi = \[Psi] /. sol[[1]];
Print["\[Psi](-5)=", psi[-5]];
Print["\[Psi](5)=", psi[5]];
Plot[psi[x], {x, -5, 5}]

```

Po izvršenju programa i nalaženja rešenja postavljene diferencijalne jednačine, interpolaciona funkcija je izdvojena i ispisane su njene vrednosti na krajevima zadatog opsega.



Slika 4.16: Talasna funkcija $\psi_6(x)$ za 1D LHO.

Daljim razvojem ovog programa, umetanjem njegovih linija u funkciju *Do* sa celobrojnim brojačem *m* umesto broja *n*, napisan je program kojim se generiše lista interpoliranih talasnih funkcija.

```

functions = {};
Do[
eqLHO = \[Psi]''\[Xi] - (\[Xi]^2 - 2 n - 1) \[Psi][\[Xi]] == 0;
incond = If[Mod[m, 2] == 0, {\[Psi][0] == 1, \[Psi]'[0] == 0},
{\[Psi][0] == 0, \[Psi]'[0] == 1}];
eqLHO = Union[{eqLHO}, incond] /. {n -> m};
sol = NDSolve[eqLHO, \[Psi], {\[Xi], -5, 5}];
psi = \[Psi] /. sol[[1]];
AppendTo[functions, psi], {m, 1, 5}];

```

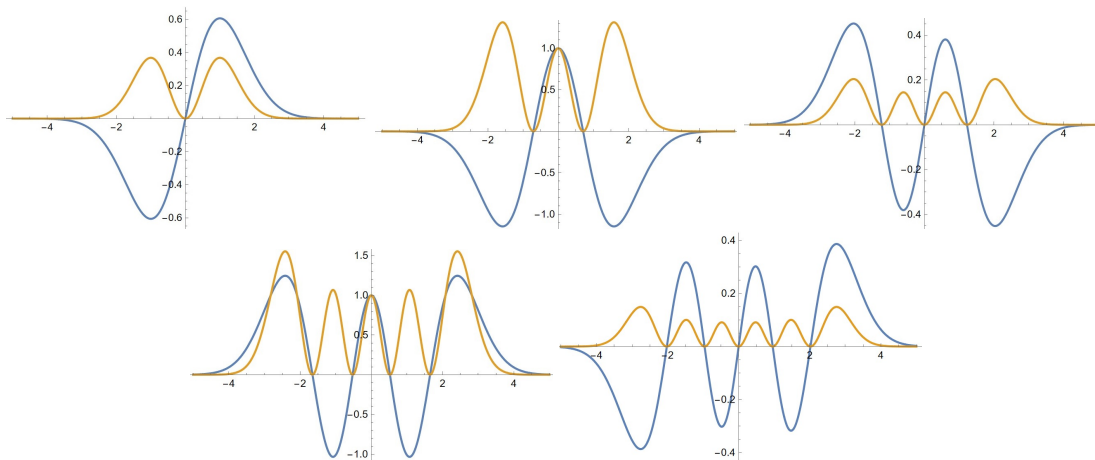
Nađene talasne funkcije $\psi_n(x)$ jednodimenzionalnog kvantnog harmonijskog oscilatora za $n = 1, 2, 3, 4, 5$ deponovane su u skupu *functions*. Svaka od njih je nacrtana za vrednosti x na intervalu od -5 do 5 u paru zajedno sa kvadratom njene vrednosti, što predstavlja gustinu verovatnoće nalaženja čestice

$Plot[\{functions[[i]][x], functions[[i]][x]^2\}, \{x, -5, 5\}]$. Uz pomoć naredbe *Table* napravljena je lista ovih grafika ($i = 1, \dots, Length[functions]$), sortirana naredbom

GraphicsColumn

te čini kolonu.

```
GraphicsColumn[
Table[Plot[{functions[[i]][x], functions[[i]][x]^2}, {x, -5, 5}],
{i, Length[functions]}]]
```



Slika 4.17: Talasne funkcije za 1D LHO (plava linija) i njihove gustine verovatnoće (žuta linija) za $n = 1, 2, 3, 4, 5$.

4.6 Upotreba lista u računu

Liste, opšta definicija

Česti su primeri gde su vektori ili matrice reprezentovani listama ili listama čiji su elementi liste. Primena lista u programskom jeziku Wolfram je raznolika. Postoje različiti načini njihovog definisanja, prikazivanja, konstruisanja. Najprostije rečeno, liste su nizovi objekata, razdvojenih zarezom i ograđenih vitičarskim zagradama. Česti objekti su brojevi, čineći nizove brojeva. U opštem slučaju, elementi lista mogu biti bilo koji izrazi kao što su: simboli, stringovi karaktera, funkcije, jednačine ili transformaciona pravila koja su izaz naredbi *Solve*, *DSolve*. Prvi argument funkcije *Solve* je lista jednačina, a drugi lista varijabli. Izlaz funkcije je lista transformacionih pravila kojim se nepoznatim dodeljuju varijable. Takođe, lista najčešće, ali ne nužno sadrži isti tip objekata. Pored najčešće upotrebljavane funkcije *Table* za generisanje lista, postoje i druge funkcije ove namene.

Naredbom $\text{Range}[i_{max}]$ generiše se lista $\{1, 2, \dots, i_{max}\}$.

Sa $\text{Range}[i_{min}, i_{max}]$ generiše se lista $\{i_{min}, i_{min} + 1, \dots, i_{max}\}$, dok sa varijantom $\text{Range}[i_{min}, i_{max}, di]$ se generiše lista $\{i_{min}, i_{min} + di, \dots, i_{max}\}$.

Naredbom $\text{Array}[a, n]$ kreira se opšta lista oblika $\{a[1], a[2], \dots, a[n]\}$,

dok se naredbom $\text{Array}[a, \{n_1, n_2, \dots\}]$ kreira složena lista elemenata $a[i_1, i_2, \dots]$.

Za prikazivanje lista koriste se funkcije:

U formatu kolone elementi od *list* prikazuju se naredbom $\text{ColumnForm}[list]$.

Naredbom $\text{MatrixForm}[list]$ prikazuju se elementi od *list* u njenom regularnom pravougaonom (kvadratnom) poretku.

$\text{TableForm}[list]$ prikazuje elemente od *list* u nizu pravougaonih ćelija koje ne moraju biti nužno iste.

Prebrojavanjem elemenata listi najčešće se vrši uz pomoć naredbi:

`Length[list]` daje broj elemenata od `list`,

`Count[list, elem]` daje broj elemenata `elem` od `list`,

`Dimensions[list]` je lista koja sadrži dimenzije strukture `list`.

Pored već prikazanog izdvajanja elemenata liste `list` uz pomoć uglaste zagrade, za ekstrakciju elemenata i podlisti koriste se i sledeće funkcije:

`First[list]` izdvaja prvi element liste `list[[1]]`,

`Last[list]` izdvaja poslednji element liste `list[[-1]]`,

`Part[list, n]` ili `list[[n]]` daje n -ti element liste,

`Part[list, {n1, n2, ...}]` ili `list[[{n1, n2, ...}]]` izdvajaju se elementi liste na mestima n_1, n_2, \dots ,

`Take[list, n]` izdvaja podlistu prvih n elemenata od `list`,

`Take[list, -n]` izdvaja podlistu poslednjih n elemenata od `list`,

`Take[list, n, m]` izdvaja podlistu od n -tog do m -tog elemenata od `list`,

`Drop[list, n]` daje podlistu od `list` bez n prvih elemenata,

`Drop[list, -n]` daje podlistu od `list` bez n poslednjih elemenata,

`Drop[list, {m, n}]` daje podlistu elemenata od `list` koja sadrži sve njene elemente izuzev elemenata od m -tog do n -tog.

Ilustracije radi, biće kreiran skup elemenata $\{x^i \cos(ix), \dots\}$, $i = 1, \dots, 6$. Potom će iz ove liste biti izdvojen četvrti element, formirana lista sačinjena od prvog, trećeg i petog elementa, izdvojena lista od prva tri elementa i generisana lista izuzev elemenata od drugog do četvrtog.

Zadavanje liste izvršeno je sledećom linijom.

```
set=Table[x^i Cos[i*x], {i, 1, 6}]
```

Rešenje: $\{x \cos(x), x^2 \cos(2x), x^3 \cos(3x), x^4 \cos(4x), x^5 \cos(5x), x^6 \cos(6x)\}$. Izdvajanje četvrtog elementa

```
Part[set, 4]
```

daje $x^4 \cos(4x)$.

Generisanje podliste čiji su elementi prvi, treći i peti od `set`:

```
Part[set, {1, 3, 5}]
```

dobija se $\{x \cos(x), x^3 \cos(3x), x^5 \cos(5x)\}$.

Kreiranjem listi sačinjenih od prva 3 elementa, a potom od drugog do četvrtog elementa

```
Take[set, 3]
Drop[set, {2, 4}]
```

dobije se $\{x \cos(x), x^2 \cos(2x), x^3 \cos(3x)\}$ i $\{x \cos(x), x^5 \cos(5x), x^6 \cos(6x)\}$.

Do sada su prikazane funkcije koje se odnose na pozicije elemenata u listama. Postoje funkcije uz pomoć kojih se elementi liste ne biraju prema pozicijama već na osnovu njihovih svojstava:

`Select[list, criterion]` izdvaja sve elemente e_i iz `list` prema zadatom kriterijumu, odnosno ako je tačno `criterion[ei]`,

`Select[list, criterion, n]` izdvaja prvih n elemenata e_i iz `list` prema zadatom kriterijumu, odnosno ako je tačno `criterion[ei]`.

Primeri za vežbu:

- (i) Iz liste prvih 10 prirodnih brojeva izdvojiti listu parnih.

```
lst1 = Range[10];
Select[lst1, EvenQ]
```

Rešenje: $\{2, 4, 6, 8, 10\}$.

Korišćena je naredba

$$\text{EvenQ}[num]$$

čiji je izlaz *True* ako je *num* paran ili *False* u suprotnom.

- (ii) Iz liste $\{a, 4 + 5i, x^7, -4, 6, \frac{7}{8}\}$ selektovati sve pozitivne brojeve.

```
lst2={a,4+5 I,x^7,-4,6,'7/8};
Select[lst2, Positive]
```

Rešenje: $\{6, \frac{7}{8}\}$.

Upotrebljena je predikatska funkcija

$$\text{Positive}[num],$$

čiji je argument broj *num*, a izlaz *True* ukoliko je on pozitivan odnosno *False* u suprotnom.

- (iii) Iz liste $\left\{ \{a, 3 + 6i\}, \{a, \{5, 9\}\}, \frac{5}{4}, x, \{1, 2, 3\}, \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \right\}$ selektovati skup svih objekata koji imaju strukturu vektora.

```
lst3={{a,3+6 I},{a,{5,9}},5/4,x,{1,2,3},{1,2},{3,4}};
Select[lst3, VectorQ]
```

Dobija se $\{\{a, 3 + 6i\}, \{1, 2, 3\}\}$, a koristi se funkcija

$$\text{VectorQ}[expr],$$

čiji je izlaz *True* ukoliko argument *expr* ima strukturu vektora odnosno *False* u suprotnom.

Za izmene elemenata zadate liste *list* ili podliste koriste se naredbe:

Prepend[*list*, *elem*] vrši se umetanje objekta *elem* na prvo mesto od *list*,

Append[*list*, *elem*] element *elem* umeće se na poslednje mesto od *list*,

Insert[*list*, *elem*, *i*] vrši umetanje *elem* na *i*-tu poziciju od *list*,

Insert[*list*, *elem*, {*i*, *j*, ...}] vrši umetanje *elem* na pozicije {*i*, *j*, ...} u *list*,

Delete[*list*, *i*] briše *i*-ti element u *list*,

Delete[*list*, {*i*, *j*, ...}] briše elemente u *list* na pozicijama {*i*, *j*, ...}

ReplacePart[*list*, *elem*, *i*] umesto elementa na poziciji *i* u *list* umeće se *elem*,

ReplacePart[*list*, *elem*, {*i*, *j*, ...}] umesto elemenata na poziciji {*i*, *j*, ...} u *list* umeće se *elem*.

Ilustracije radi izvršiće se izmena elemenata liste

$\{a, \{a, b, a\}, b, \{\{a, b\}, c\}, d\}$.

Umetanjem elementa *x* na početak zadate liste:

```
yourlist={a,{a,b,a},b,{a,b,c},d};
Prepend[yourlist,x]
```

dobije se $\{x, a, \{a, b, a\}, b, \{\{a, b\}, c\}, d\}$.

Brisanjem poslednjeg elementa iz *yourlist*

```
Delete[yourlist, -1]
```

izlaz je $\{a, \{a, b, a\}, b, \{\{a, b\}, c\}\}$. Brisanje elementa c zahteva brisanje po dubini iz 4 podliste na 2. mestu

```
Delete[yourlist, {4, 2}]
```

a rezultat je $\{a, \{a, b, a\}, b, \{\{a, b\}\}, d\}$. Umetanje elementa y na pozicije: $\{2\}$, $\{2, 2\}$ i $\{4, 1, 2\}$

```
Insert[yourlist, y, {{2}, {2, 2}, {4, 1, 2}}]
```

rezultuje listom $\{a, y, \{a, y, b, a\}, b, \{\{a, y, b\}, c\}, d\}$.

Neke od funkcija za preuređivanje liste $list$:

Sort[list] sortira listu prema kanonskom redu (brojevi od manjeg ka većem ili simboli prema abecednom redu),

Union[list] daje sortiranu verziju od $list$, bez duplikata elemenata,

Reverse[list] daje listu obrnutog redosleda od zadatog,

RotateLeft[list] daje listu ciklično zarotiranu za 1 mesto ulevo, odnosno rezultujuća lista počinje od drugog elementa zadate a završava se njenim prvim,

RotateRight[list] daje listu ciklično zarotiranu za 1 mesto udesno, odnosno rezultujuća lista počinje od poslednjeg elementa zadate a završava se njenim pretposlednjim,

Permutation[list] generiše listu svih mogućih permutacija elemenata liste zadate u argumentu.

Opisane naredbe biće uvežbane sledećim primerima.

(i) Soritirati elemente liste

$$\left\{abd, abc, ab, a, d, 10, a, 10, 5.5, \frac{1}{4}, \frac{23}{2}, ada\right\}$$

```
hislist={abd, abc, ab, a, d, 10, a, 10, 5.5, 1/4, 23/2, ada};
Sort[hislist]
```

Rešenje: $1/4, 5.5, 10, 10, 23/2, a, a, ab, abc, abd, ada, d$. Rezultat je lista brojeva i simbola redom raspoređena po rastućem i abecednom redu respektivno. Treba istaći da brojevi prethode simbolima.

(ii) Uklanjanje višestrukih elemenata liste uz njihovo po opisanom poretku, primenjeno na zadanu listu:

```
Union[hislist]
```

dobije se izlaz: $1/4, 5.5, 10, 23/2, a, ab, abc, abd, ada, d$.

(iii) Izvršiti rotaciju liste prvih 10 prirodnih brojeva ulevo za dva i udesno za tri mesta navedenim funkcijama.

```
lst=Range[10];
RotateLeft[lst, 2]
RotateRight[lst, 3]
```

Rešenje: $\{3, 4, 5, 6, 7, 8, 9, 10, 1, 2\}$,
 $\{8, 9, 10, 1, 2, 3, 4, 5, 6, 7\}$.

(iv) Dati sve moguće permutacije liste $\{a, b, c\}$.

```
Permutations[{a, b, c}]
```

$$\text{Rešenje: } \begin{pmatrix} a & b & c \\ a & c & b \\ b & a & c \\ b & c & a \\ c & a & b \\ c & b & a \end{pmatrix}.$$

Neke funkcije kojima se restrukturiraju liste:

Partition[list, n] izvrši podelu list na podliste dužine n koje se ne preklapaju,

Split[list] izvrši podelu list na podliste istih uzastopnih elemenata,

Transpose[list] izvrši transponovanje prva dva nivoa liste,

Flatten[list, n] izvrši ravnjanje prvih n nivoa liste.

Sledećim primerima ilustrovana su reorganizovanja listi koristeći navedene naredbe.

- (i) Listu $\{a, b, c, d, e\}$ podeliti na podliste od po dva elementa, a potom formirati podliste od 2 elementa sa pomakom od jednog.

```
lst1={a,b,c,d,e};
Partition[lst1, 2]
Partition[lst1, 2, 1]
```

Rešenje: $\{\{a, b\}, \{c, d\}\},$
 $\{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}\}.$

- (ii) Podeliti listu $\{a, a, a, b, c, c, c, d, e, e\}$ na podliste koje sadrže iste elemente.

```
Split[{a, a, a, b, c, c, c, d, e, e}]
```

Rešenje: $\{\{a, a, a\}, \{b\}, \{c, c, c\}, \{d\}, \{e, e\}\}.$

- (iv) Oznaka talasne funkcije atoma vodonika sadrži četiri kvantna broja: glavni kvantni broj n , azimutalni ili orbitalni kvantni broj l , magnetni kvantni broj m_l i spinski kvantni broj m_s . Dozvoljene vredosti ovih kvantnih brojeva su:

$$n = 1, 2, 3, \dots$$

$$l = 0, 1, 2, \dots, n - 1$$

$$m_l = -l, -l + 1, \dots, 0, 1, 2, \dots, l$$

$$m_s = -1/2, 1/2.$$

Za $n \leq 3$ generisati listu svih mogućih kvantnih brojeva u vidu $\{n, l, m_l, m_s\}$, a potom izabrati sve one sa $l = 1$.

```
qnums=Table[{n, 1, ml, ms}, {n, 3}, {1, 0, n-1}, {ml, -1, 1}, {ms, -1/2, 1/2}];
qnums = Flatten[qnums, 3]
```

Rešenje: $\{\{1, 0, 0, -1/2\}, \{1, 0, 0, 1/2\}, \{2, 0, 0, -1/2\}, \{2, 0, 0, 1/2\},$
 $\{2, 1, -1, -1/2\}, \{2, 1, -1, 1/2\}, \{2, 1, 0, -1/2\}, \{2, 1, 0, 1/2\}, \{2, 1, 1, -1/2\},$
 $\{2, 1, 1, 1/2\}, \{3, 0, 0, -1/2\}, \{3, 0, 0, 1/2\}, \{3, 1, -1, -1/2\}, \{3, 1, -1, 1/2\},$
 $\{3, 1, 0, -1/2\}, \{3, 1, 0, 1/2\}, \{3, 1, 1, -1/2\}, \{3, 1, 1, 1/2\}, \{3, 2, -2, -1/2\},$
 $\{3, 2, -2, 1/2\}, \{3, 2, -1, -1/2\}, \{3, 2, -1, 1/2\}, \{3, 2, 0, -1/2\}, \{3, 2, 0, 1/2\},$
 $\{3, 2, 1, -1/2\}, \{3, 2, 1, 1/2\}, \{3, 2, 2, -1/2\}, \{3, 2, 2, 1/2\}\}.$ Linija koda kojom se nalaze svi setovi kvantnih brojeva čiji je angularni kvantni broj $l = 1$:

```
Select[qnums, #[[2]]===1 &]
```

a rezultat je: $\{\{2, 1, -1, -1/2\}, \{2, 1, -1, 1/2\}, \{2, 1, 0, -1/2\}, \{2, 1, 0, 1/2\}, \{2, 1, 1, -1/2\},$
 $\{2, 1, 1, 1/2\}, \{3, 1, -1, -1/2\}, \{3, 1, -1, 1/2\}, \{3, 1, 0, -1/2\}, \{3, 1, 0, 1/2\},$
 $\{3, 1, 1, -1/2\}, \{3, 1, 1, 1/2\}\}.$

Često korišćene funkcije za kombinovanje listi:

Join[*list*₁, *list*₂, ...] spaja liste *list*₁, *list*₂, ..., u jednu,

Union[*list*₁, *list*₂, ...] daje sortiranu listu svih različitih elemenata koji se pojavljuju u *list*₁, *list*₂, ...,

Intersection[*list*₁, *list*₂, ...] nalazi sortiranu listu svih zajedničkih elemenata od *list*₁, *list*₂, ...,

Complement[*listall*, *list*₁, *list*₂, ...] nalazi sortiranu listu elemenata iz *listall* koji ne pripadaju listama *list*₁, *list*₂, ...

Navedene funkcije biće uvežbane na primerima kombinovanja skupova *listA* = {1, 2, 3}, *listB* = {2, 3, 4, 5} i *listC* = {1, 2, 3, 4, 5, 6, 7, 8}.

(i) Spojiti liste *listA* i *listB*.

```
listA = Range[3];
listB = Range[2, 5];
listC = Range[8];
Join[listA, listB]
```

Rešenje: {1, 2, 3, 2, 3, 4, 5}.

(ii) Naći uniju i presek od *listA* i *listB*.

```
Union[listA, listB]
```

Rešenje: unija skupova *listA* i *listB* je {1, 2, 3, 4, 5}, a njihov presek je {2, 3}.

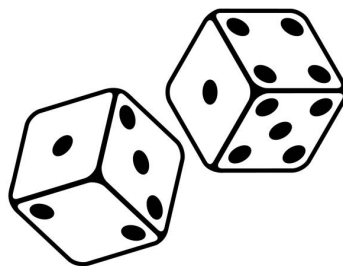
(iii) Naći elemente koji pripadaju skupu *listC* a ne pripadaju listama *listA* i *listB*.

```
Complement[listC, listA, listB]
```

Rešenje: {6, 7, 8}.

4.6.84 Simulacija bacanja kockice.

Na osnovu velikog broja bacanja kockice izračunati srednju vrednost pojavljivanja. Grafički prikazati srednju vrednost u zavisnosti od broja bacanja i na dobijenom grafiku horizontalnom linijom grida istaći teorijsku vrednost.



Za rešavanje zadatog problema koristiće se generator slučajnih brojeva od 1 do 6.

```
Random[Integer, {1, 6}]
```

Verovatnoća dobijanja bilo kog od 6 brojeva prilikom bacanja kockice iznosi $p_i = 1/6$. Srednja vrednost dobijena prilikom ovog slučajnog događaja iznosi $\langle i \rangle = \sum_{i=1}^6 ip_i = 3.5$. Simulacija bacanja kockice veliki broj puta *N*hts izvršena je kreiranjem skupa (liste) uz pomoć naredbe

Table dužine jednake broju bacanja, čiji su elementi slučajni celi brojevi od 1 do 6, generisani prikazanom naredbom. Nakon generisanja ove liste, traži se aritmetička sredina njenih elemenata, najpre zbrajanjem svih vrednosti naredbom *Total*, a potom deljenjem sa brojem elemenata liste.

```
Nhts=1000;(*broj bacanja*)
lista=Table[Random[Integer,{1,6}],{Nhts}];
1.Total[lista]/Nhts
```

Umesto usrednjavanja prikazanim metodom, moguće je upotrebom samo jedne funkcije

$$\text{Mean}[lista],$$

čiji je argument lista slučajno generisanih brojeva.

```
1. Mean[1]
```

Slučajno generisani elementi skupa *lista* su celi brojevi, te će rezultat u oba data slučaja usrednjavanja biti racionalan broj, a njegov prikaz u decimalnom zapisu realizuje se množenjem sa realnim brojem 1.

Za grafički prikaz zavisnosti srednje vrednosti od broja bacanja potrebno je dati listu čiji su elementi podliste dužine 2 strukture $\{broj\ bacanja, srednjavrednost\}$. Na početku, pre *Do* petlje definisan je prazan skup *setrez*. Unutar petlje sa brojačem *Nhts* koji definiše brojeve bacanja i uzima vrednosti između 10 i 10000, najpre se generiše skup celih brojeva između 1 i 6, dužine *Nhts*, a zatim naredbom *AppendTo* za svaku vrednost *Nhts* par $\{Nhts, srednjavrednost\}$ se ubacuje u skup *setrez*.

```
setrez = {};(*skup rezultata*)
Do[
lista=Table[Random[Integer, {1, 6}], {Nhts}];
AppendTo[setrez, {Nhts, 1. Mean[lista]}], {Nhts, 10, 10000}];
ListPlot[setrez,
AxesLabel->{"Broj bacanja", "Srednja vrednost"},
LabelStyle -> Directive[18],
GridLines -> {{}, {{3.5, Thickness[0.01]}}},
GridLinesStyle -> Directive[Blue]]
```

```
LabelStyle->Directive[Blue,Bold,18]
```

Lista *setrez* je nacrtana sa *ListPlot*. Teorijski dobijena srednja vrednost istaknuta je punom plavom linijom, što je realizovano opcijama naredbe *ListPlot*.

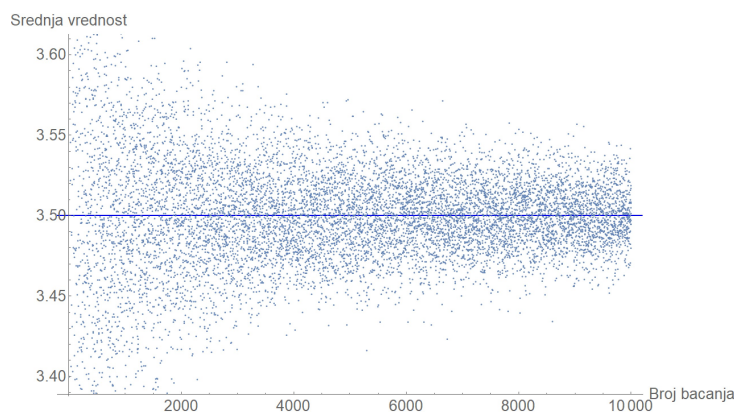
```
GridLines->{{},{{3.5,Thickness[0.01]}}},
GridLinesStyle->Directive[Blue]]
```

Kada je broj bacanja kockice mali očekuje se veliki rastur usrednjenih rezultata oko teorijske vrednosti 3.5, dok se sa povećanjem broja bacanja odstupanje od srednje vrednosti smanjuje.

- 4.6.85 Simulirajući veliki broj bacanja 3 kockice izračunati verovatnoću da se tom prilikom dobiju sve tri vrednosti iste.

Za rešavanje ovog zadatka biće korišćena naredba *Union*, koja daje sortiranu listu iz koje prethodno obriše sve duplirane elemente. Simulira se istovremeno bacanje 3 kockice veliki broj puta. Tako na primer, 5 bacanja 3 kockice se realizuje uz pomoć naredbe *Table* na sledeći način.

Prvi korak simulacije je generisanje skupa *lista1* trojki slučajnih brojeva (liste slučajnih brojeva dužine $Ndcs = 3$) koji broji onoliko elemenata koliki je broj bacanja *Nbc*.



Slika 4.18: Tačkice predstavljaju usrednju vrednost rezultata dobijenih bacanjem kockice u zavisnosti od broja simuliranih bacanja. Plava linija seče y osu na egzaktnoj srednjoj vrednosti.

```
In[1]:= Table[Random[Integer, {1, 6}], {5}, {3}]
Out[1]= {{5, 4, 6}, {5, 1, 4}, {3, 6, 5}, {6, 2, 6}, {4, 5, 2}}
```

```
Clear[Ndcs,Nbc];
Ndcs=3;(*broj kockica*)
Nbc=100000;(*broj bacanja*)
lista1=Table[Random[Integer,{1,6}],{Nbc},{Ndcs}];
```

Koristeći klasično proceduralno programiranje implementirano uz pomoć naredbe *Do*, najpre se formira brojač *br*, kojim se broje sve one podliste trojki generisane liste *lista1* u kojima su sva tri broja ista. Svaka trojka *lista1[[i]]* se proverava i kada je njena unija lista koja ima samo jedan element, brojaču se dodaje +1. Na taj način se prebroje sve trojke istih slučajno generisanih elemenata. Konačan broj *br* podeljen sa ukupnim brojem bacanja daje verovatnoću (učestalost) pojavljivanja 3 iste kockice.

```
br=0;
Do[
If[Length[Union[lista1[[i]]]]===1,br=br+1,br=br+0],{i,1,Nbc}];
1.br/Nbc
```

Predloženo rešenje uz pomoć funkcionalnog programiranja počinje formiranjem liste *lista2* sa elementima definisanim kao dužina unije elemenata liste *lista1*. Lista *lista2* se sortira od manjeg ka većem elementu, a zatim se iz nje generiše lista *lista4* selekcijom svih elemenata jednakih 1. Dužina liste *lista4* predstavlja broj trojki slučajnih brojeva čiji su elementi isti. Ovaj broj podeljen sa brojem bacanja daje njihovu učestalost pojavljivanja.

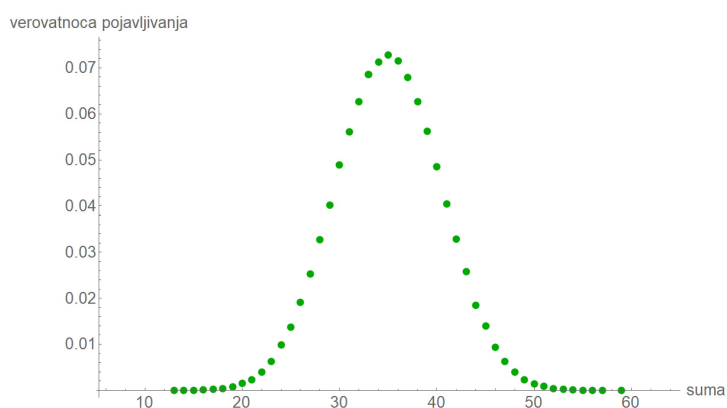
```
lista2=Length[Union[#]]&/@lista1;
lista3=Sort[lista2];
lista4=Select[lista3,#===1 &];
1.Length[lista4]/Nbc
```

- 4.6.86 Grafički prikazati raspodelu verovatnoće pojavljivanja sume dobijenih vrednosti prilikom istovremenog bacanja 10 kockica. Na istom grafiku prikazati raspodele ukoliko se bacaju 5, 6, 7, 8, 9 i 10 kockica.

```
Ndcs=10;(*broj kockica koji se baca*)
Nbc=100000;(*broj bacanja seta kockica*)
lista1=Table[
lista2=Table[Random[Integer,{1,6}],{Ndcs}];
Total[lista2],{Nbc}];
lista3=Sort[lista1];
lista3=Split[lista3];
```

```
setrez=
Table[{lista3[[i,1]],1.Length[lista3[[i]]]/Nbc},{i,Length[lista3]}];
```

```
graph=
ListPlot[setrez,PlotStyle->Darker[Green],
AxesLabel->{"suma", "verovatnoca pojavljivanja"},
LabelStyle->Directive[16], PlotRange -> {{4, 70}, All}]
```



Slika 4.19: Distribucija verovatnoća pojavljivanja suma prilikom istovremenog bacanja 10 kockica.

Zna se da za 10 kockica sume uzimaju vrednost od 10 do 60 tj. ukupno 51 različitih vrednosti, a zbir verovatnoća svih pojavljivanja iznosi 1. Lista *setrez* sadrži parove, na prvom mestu suma a na drugom verovatnoća njenog javljanja. Sledećom linijom sumirane su verovatnoće pojavljivanja svih suma prilikom bacanja 10 kockica.

```
Total[setrez[[All,2]]]
```

Drugi deo zadatka, prikazivanje distribucija pojavljivanja suma za različit broj kockica (od 5 do 10), rešen je neznom izmenom prethodnog primera. Najpre je uveden novi brojač koji prebrojava kockice. Umesto promenljive *Ndcs*, broj kockica je sada brojač *id* spoljašnje *Do* petlje, u kojoj se izvršava prethodno napisan program za računanje verovatnoća pojavljivanja suma prilikom bacanja zadatog broja kockica i njihov prikaz. Pre *Do* petlje formiran je prazan skup *graphs* u koji je za svaku vrednost *id* ubačen grafik na kom su tačkice spojene linijama iste boje. Ovakav prikaz, dat radi lakšeg praćenja, realizovan je kombinovanjem grafika sa tačkicama i linijama dobijenih redom naredbama *ListPlot* i *ListLinePlot*. Grafici koji odgovaraju različitom broju kockica prikazani su različitim bojama što je realizovano kroz opciju *PlotStyle* -> *Hue[id/6]* u zavisnosti od iteratora koji prebrojava broj kockica.

```
PlotStyle -> Hue[id/6]
```

Tokom izvršavanja *Do* petlje puni se skup *graphs*, koji se nakon završenog ciklusa prikazuje kombinovano naredbom *Show*.

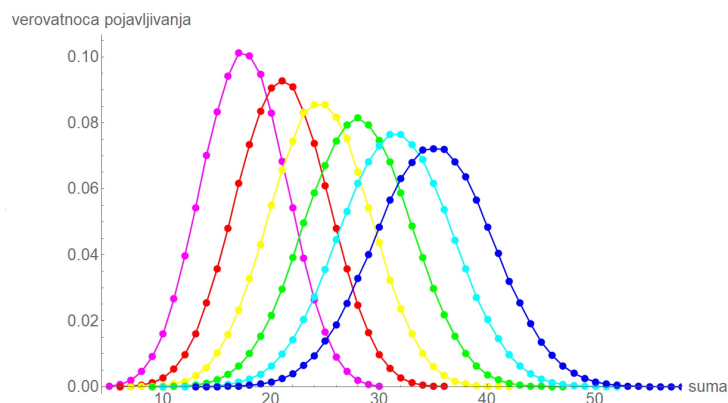
```

graphs = {};
Nbc = 500000;(*broj bacanja seta kockica*)
Do[
lista1 = Table[lista2 = Table[Random[Integer, {1, 6}], {id}], {id}],
Total[lista2],{Nbc}];
lista3=Sort[lista1];
lista3=Split[lista3];
setrez=
Table[{lista3[[i, 1]], 1. Length[lista3[[i]]]/Nbc},
{i,Length[lista3]}];

graph=Show[
ListLinePlot[setrez, PlotStyle -> Hue[id/6],
LabelStyle->Directive[16],
GridLines->{{}}, {{3.5, Thickness[0.01]}}},
GridLineStyle->Directive[16],
PlotRange->{{4, 70}, All}],
ListPlot[setrez, PlotStyle -> Hue[id/6],
LabelStyle->Directive[Blue, Bold, 18],
GridLines->{{}},{{3.5,Thickness[0.01]}}},
GridLineStyle -> Directive[Blue],
PlotRange->{{4, 70}, All}],
AxesLabel->{"suma", "verovatnoca pojavljivanja"}];
AppendTo[graphs, graph], {id, 5, 10}];

Show[graphs, PlotRange -> All]

```



Slika 4.20: Distribucije verovatnoća pojavljivanja suma prilikom istovremenog bacanja 5,6,7,8,9 i 10 kockica.

Tačkicama roze boje slika 4.20 odgovaraju verovatnoće pojavljivanja suma prilikom bacanja 5 kockica. Od svih računatih ova distribucija ima najveću maksimalnu vrednost koja je pomena ka manjim sumama. Sa brojem kockica broj različitih vrednosti suma raste (sa tim i broj tačkica) šireći raspodelu se dok maksimumi verovatnoća postaju sve manji i pomeraju se ka većim sumama.

4.6.87 Računanje broja π .

Računanje broja π Monte Karlo aproksimacijom predstavlja jedan od osnovnih primera Monte Karlo integracije. Krug radijusa 1, sa centrom u koordinatnom početku, upisan je u kvadrat čije su stranice paralelne koordinatnim osama. Izdvoji se četvrt kvadrata i kruga u prvom kvadrantu,

površine P_{sq} oivičene koordinatnim osama sa temena u tačkama $A = (0,0)$, $B = (0,1)$, $D = (1,0)$ i $C = (1,1)$. Površina kružnog isečka P_{crc} jednaka je četvrtini od ukupne površine kruga $P_{crc} = 1^2\pi/4$. Po izdvojenom kvadratu uniformno se razbacuju tačke, što znači da ukoliko je njihov broj veliki $N \rightarrow \infty$, verovatnoća p da tačka padne recimo na površinu kružnog isečka srazmena je njenoj veličini

$$p = \lim_{N \rightarrow \infty} \frac{n}{N},$$

gde je n broj tačaka koji padne na četvrt kruga. Sa druge strane, verovatnoća da tačke, koje se uniformno generišu u kvadratu $\square ABCD$ padnu na površinu četvrtini kruga jednaka je odnosu površina

$$p = \frac{P_{crc}}{P_{sq}} = \frac{\pi/4}{1}$$

Iz datih jednakosti dobije se veza između broja π i verovatnoće

$$\pi = 4 \lim_{N \rightarrow \infty} \frac{n}{N},$$

na osnovu koje se može razviti strategija za njegovo računanje. Slučajno generisana i -ta tačka sa $\{Random[], Random[]\}$ koja je uniformno distribuisana na domenu $[0, 1] \times [0, 1]$, određena je sa dve koordinate (x_i, y_i) , $i = 1, 2, \dots, N$. Ovo se ponavlja veliki broj puta N pri čemu počinje da važi aproksimacija

$$\pi \approx 4 \frac{n}{N} \quad (4.14)$$

a potom se izbroje sve one tačke koje pripadaju četvrtini kruga da bi se ova aproksimacija primenila. Na tačnost izračunate vrednosti prema (4.14) veliki uticaj ima broj generisanih tačaka $Npts$.

```
Clear[Npts,pts];
Npts=10000>(*broj tacaka*)
pts=Table[{Random[],Random[]},{Npts}];(*generisane tacke *)
(*lista parova slucajnih brojeva izmenju 0 i 1*)
grsq=ListPlot[pts,PlotStyle->{Blue,PointSize[0.01]}];
setpts={};(*tacake u krugu*)
Do[
If[pts[[j]].pts[[j]]<=1,AppendTo[setpts,pts[[j]]],{j,Npts}];
grc=ListPlot[setpts,PlotStyle->{Red,PointSize[0.005]}];
Show[grsq,grc,AspectRatio->Automatic,LabelStyle->Directive[Black,Large]]
(*istovremeno prikazane generisane tacke i one u krugu*)
4.*Length[setpts]/Npts
```

Rešenje je realizovano sledećim postupkom, nakon generisanja liste tačaka pts uniformno razbacanih po kvadratu $\square ABCD$, selektovane su one unutar četvrtine kruga. Pre Do petlje formira se prazna lista $setpts$ u koju se ubacuju tačke ukoliko ipunjavaju uslov da je kvadrat njihove udaljenosti od koordinatnog početka manji od kvadrata radijusa kruga $x_i^2 + y_i^2 \leq r^2$. Selektovane tečke iz skupa $setpts$, dužine n , ilustrovane su crvenom zajedno sa svim generisanim tačkama, čime je proverena procedura njihovog izbora.

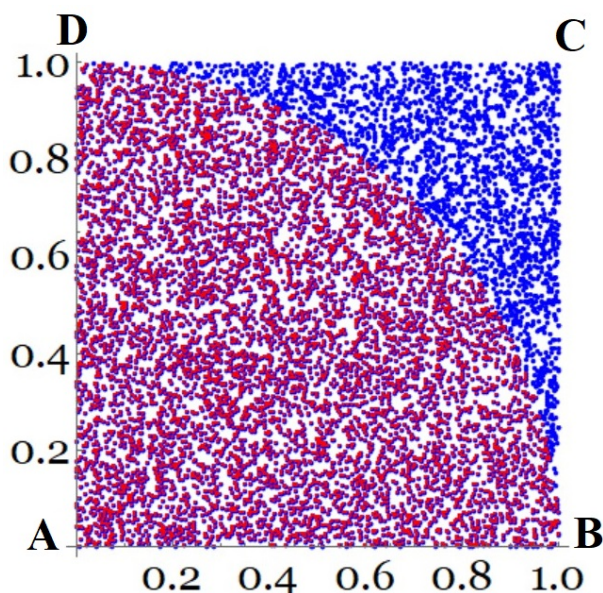
Umesto gore datog proceduralnog, biće razrađen i programski kod traženja skupa $setpts$ baziran na funkcionalnom programiranju. Kao što je naglašeno u uvodnom poglavlju, ovakav način programiranja u programu Mathematica je znatno efikasniji.

```
setpts=Select[pts, #.#<=1 &];
```

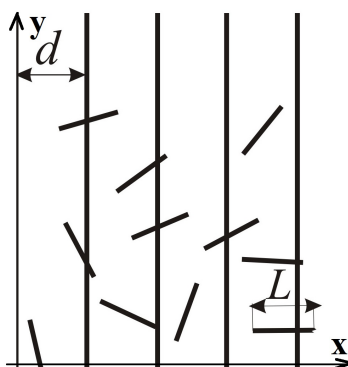
4.6.88 Bufonove igle i broj π .

Tabela 4.1: Aproximativne vrednosti π_a dobijene Monte Karlo metodom za različit broj slučajno generisanih tačaka N .

N	π_a	$ \pi_a - \pi $
100	3.3600	0.2184
1000	3.264	0.1224
100008	3.1348	0.0067
100000	3.1434	0.0019



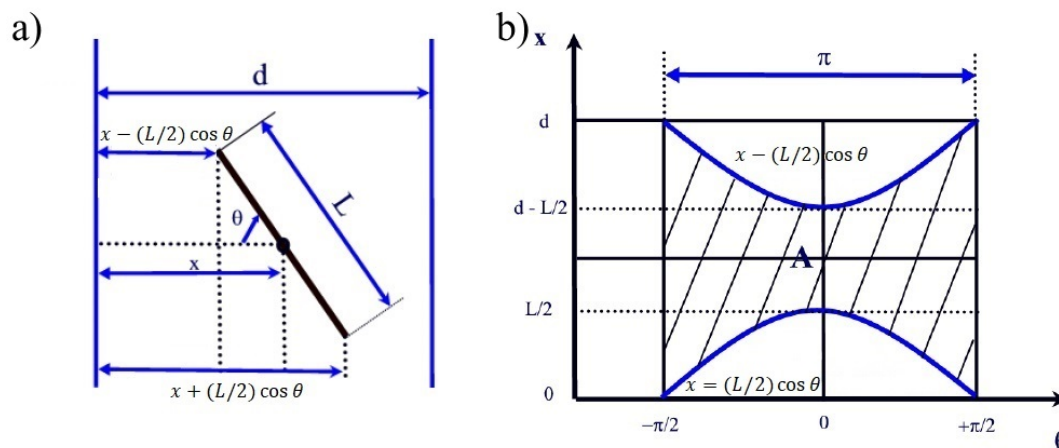
Određivanje broja π pomoću geometrijske verovatnoće jedna je od prvih Monte Karlo simulacija nastala 1777. godine. Eksperiment se sastoji u tome da se bacaju igle (šibice) dužine L na ravnu površinu gde su uniformno iscrtane paralelne linije na jednakom međusobnom rastojanju d . Traži se verovatnoća da igla kada padne na površinu preseče zadatu liniju. Igle se nakon pada na ovu površinu orijentišu u slučajnom pravcu.



Slika 4.21: Štapići rasuti po ravnoj površini iscrtanoj linijama paralelnih y -osi.

Prilikom bacanja igle u eksperimentu, javljaju se dve stohastičke veličine. Jedna je položaj centra mase igle a druga ugao za koji je zarotirana oko svog centra mase u odnosu na pravac pružanja linija. Ugao orijentacije θ igle uzima vrednosti unutar intervala $[-\pi/2, \pi/2]$. Položaj centra

igle između paralelnih pravih može biti bilo koja vrednost uniformno raspoređena na intervalu $x \in [0, d]$. Da li će igla preseći liniju ili ne, nije zavisno od y koordinate slika 4.21. Stoga, prostor parametara je određen sa dve slučajne promenljive x i θ . Zadatak je moguće pojednostaviti na osnovu periodičnih uslova. Pošto se linije pravilno periodično javljaju u jednom pravcu, dovoljno je gledati samo jedan njihov par i x koordinatu razrađivati u intervalu $x \in [0, d]$.



Slika 4.22: a) Determinisanje prostornih koordinata u odnosu na položaj igle. b) Prostor slučajnih parametara sa označenim oblastima kada dolazi do preseka igle sa linijom.

Verovatnoća dobijena sledećom analizom ima smisla uz uslov da je $d > L$! Prema slici 4.22 b) do presecanja igle i linije neće doći kada je ispunjeno $x - L/2 \cos \theta > 0$ i $x + L/2 \cos \theta < d$ pri čemu su paralelne (plave linije na slici 4.22 a)) postavljene u $x = 0$ i $x = d$. Kada slučajno generisani parametri x i θ pripadaju išrafiranoj površini neće doći do preseka. Analizom površine domena parametara nalazi se verovatnoća da ne dođe do preseka ($p_{noncross}$), jednaka odnosu išrafirane površine A i ukupne površine u ravni domena dozvoljenih parametara πd .

$$A = d\pi - 2 \int_{-\pi/2}^{\pi/2} \frac{L}{2} \cos \theta d\theta = d\pi - 2L/2 \int_{-\pi/2}^{\pi/2} \cos \theta d\theta,$$

$$A = d\pi - L \sin \theta \Big|_{-\pi/2}^{\pi/2} = d\pi - 2L.$$

Oдавde se dobije verovatnoća da šibica neće preseći liniju $p_{noncross} = 1 - \frac{2L}{d\pi}$, odnosno verovatnoća da će doći do preseka šibice i linije

$$p_{cross} = \frac{2L}{d\pi}. \quad (4.15)$$

```
d=5;(*rastojanje izmedju paralelnih crta*)
l=3;(*duzina sibica*)
nsib=200000;
br=0;
Do[
\[Theta]=Random[Real,{-Pi/2, Pi/2}];
x=Random[Real,{0, d}];
If[(x+Cos\[Theta])/2>d || (x-Cos\[Theta])/2<0),
br=br+1, {nsib}];
p=br/nsib;
2.1/(p*d)
```

Simulacija pada igle na površinu rađena je slučajnim izborom ugla θ i koordinate x centra mase igle, uz pomoć naredbe

$$\text{Random}[\text{Real}, \text{interval}],$$

na sledeći način.

```
\[Theta]=Random[Real,{-Pi/2, Pi/2}];
x=Random[Real,{0, d}];
```

Nakon što se prebroje sve igle br koje preseku paralelne linije, nađe se verovatnoća preseka $p_{cross} = br/nsib$, gde je $nsib$ broj igala u simulaciji. Potom, primenom relacije (4.15), odakle se vrednost broja π može izraziti preko verovatnoće i geometrijskih parametara

$$\pi = \frac{2L}{p_{cross}d}.$$

4.6.89 Monte Karlo integracija pogodak i promašaj.

Uopštenjem iznetih ideja u prethodnim zadacima, ukoliko je zadata funkcija $f(x) \geq 0$ za $x \in [a, b] \subset R$, prikazanom tehnikom može se izračunati površina

$$A = \int_a^b f(x)dx.$$

Neka je ζ definisana na sledeći način

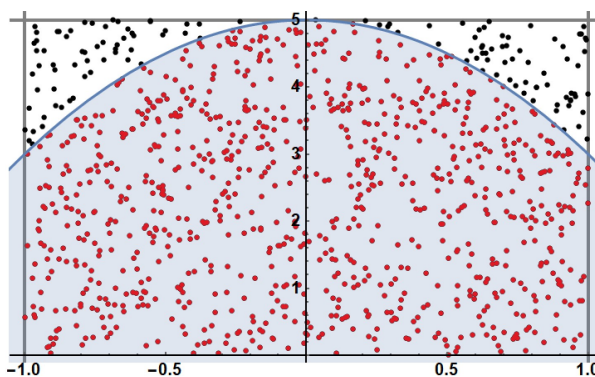
$$\zeta = \max_{x \in [a, b]} f(x),$$

a sa A_s definiše se površina pravougaonika $A_s = (b - a)\zeta$. Tada će odnos površina $A/A_s = \lim_{N \rightarrow \infty} \frac{n}{N}$, gde je N ukupan broj tačaka uniformno raspoređen $(x_i, y_i) \in [a, b] \times [0, \zeta]$, dok je n broj tačaka ispod krive zadate funkcije. Prema receptu predloženog algoritma Monte Karlo integracije, izračunati

$$\int_{-1}^1 (-2x^2 + 5)dx.$$

```
f[x_]:= -2x^2+5;(*zadavanje funkcije*)
gplt=Plot[f[x],{x,-1.8,1.8}];
Npts=1000;
As=2*5;
pts=Table[{Random[Real,{-1,1}],Random[Real,{0,5}]},{Npts}];
(*generisanje tacaka*)
setpts=Select[pts,f#[[1]]>#[[2]] &];
n=Length[setpts];
int2=As*n/Npts;
Show[
ListPlot[pts,PlotStyle->{PointSize[0.009],Black}],
ListPlot[setpts,PlotStyle->{PointSize[0.0081],Red}],
gplt,PlotRange->All,GridLines->{{-Sqrt[1],Sqrt[1]},{5}},
GridLinesStyle->Directive[Thick,Gray],
AxesStyle->Directive[Black,Bold,20]]
```

Za zadatu funkciju nađen je njen maksimum na zadatom intervalu i izračunata površina njene određenog pravougaonika. Unutar njega, uniformno su generisane tačke pts , a potom funkcionalnim programiranjem izvršena selekcija tačaka ispod dela funkcije na zadatom intervalu, na osnovu čega je dalje primenom izvedenih relacija, izračunat određeni integral.



4.6.90 Monte Karlo integracija srednjom vrednošću.

Ova tehnika integracije zasnovana je na teoremu o srednjoj vrednosti, koja kaže da ako je $f(x)$ neprekidna funkcija na intervalu $x \in [a, b]$, tada postoji $z \in [a, b]$ takvo da je

$$\int_a^b f(x)dx = f(z)(b - a).$$

Vrednost funkcije $f(z) \equiv \langle f \rangle$ još se naziva očekivana ili srednja vrednost od $f(x)$. Iz teorije verovatnoće poznato je da očekivana vrednost može biti aproksimirana aritmetičkom sredinom \bar{f} sa standardnom greškom kao

$$f \pm \sqrt{\frac{\bar{f}^2 - \bar{f}^2}{N}} \simeq \frac{1}{b-a} \int_a^b f(x')dx'.$$

Aritmetička sredina se takođe, na drugi način može definisati kao

$$\bar{f} = \frac{1}{N} \sum_{i=1}^N f(x_i),$$

dok je srednja vrednost kvadrata funkcije definisana sa

$$\bar{f}^2 = \frac{1}{N} \sum_{i=1}^N f^2(x_i).$$

Gornje relacije važe uz uslov da su x_i , $i = 1, 2, \dots, N$ uniformno distribuisani slučajni brojevi iz intervala $[a, b]$. Za velike vrednosti N gornja aproksimacija postaje sledeća jednakost

$$\frac{1}{b-a} \int_a^b f(x')dx' = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(x_i),$$

odakle se dobije krajnja formula Monte Karlo integracije

$$\int_a^b f(x)dx = (b-a) \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (4.16)$$

Na osnovu prethodno razrađenog algoritma, napisan je program gde je implementirana formula (4.16) za računanje integrala zadate neprekidne funkcije.

```
f[x_]:=Sin[x]^2-Cos[x];
a=0;
b=3Pi/2;
Npts=100000;
pts=Table[Random[Real,{a,b}],{Npts}];
int=(b-a)Total[f[#]&/@pts]/Npts
(*integral funkcije na datom intervalu*)
```

Efikasnosti radi primenjeno je funkcionalno programiranje za nalaženje vrednosti funkcije $f(x)$ u slučajno generisanim tačkama pts na zadatom intervalu $[a, b]$. Aritmetička sredina je dobijena sumirajući sve nađene vrednosti funkcija sa $Total$, a potom je dobijeni rezultat podeljen brojem generisanih slučajnih tačaka $Npts = 100000$. Komparacije radi, vrednosti izračunatog integrala date funkcije dobijene iz (4.16) na zadatom intervalu, upoređene su sa egzaktom vrednošću, a dobijena relativna greška je ispod 1%.

```
Abs[1-Integrate[f[x],{x,a,b}]/int]
(*relativna greska racunanja integrala*)
```

4.6.91 Traženje normalnih moda rešavanjem svojstvenog problema.

Problem traženja modova oscilovanja sistema više tela (čestica) može se svesti na rešavanje dinamičkih, spregnutih diferencijalnih jednačina drugog reda. Ovo predstavlja jedan od osnovnih zadataka fizike. U predstojećem primeru traže se pomeraji tokom oscilacija sistema tela istih masa (m) koji čine lanac. Tela su u nizu duž jednog pravca, a svaka dva susedna su vezana oprugama istih konstanti elastičnosti (k), čineći sistem spregnutih linearnih harmonijskih oscilatora. Najpre će biti dat pregled opšte teorije raspredanja sistema n diferencijalnih jednačina drugog reda

$$\begin{aligned}\ddot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n, \\ \ddot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n, \\ &\vdots \\ \ddot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n,\end{aligned}$$

gde su $a_{ij} \in \mathbb{R}$ koeficijenti. Ukoliko se funkcije, koje se traže rešavanjem sistema diferencijalnih jednačina, predstavljaju kao vektor

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix},$$

ceo sistem diferencijalnih jednačina dobija oblik

$$\ddot{x}(t) = \mathcal{A}x(t), \quad (4.17)$$

gde je \mathcal{A} matrica definisana na sledeći način

$$\mathcal{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}.$$

Ako je \mathcal{A} dijagonalizabilna matrica čiji je spektar $\sigma_{\mathcal{A}} = (\lambda_1, \lambda_2, \dots, \lambda_n)$ a svojstveni bazis $\{v_1, v_2, \dots, v_n\}$, pri čemu važi da je $\mathcal{A}v_i = \lambda_i v_i$. Kolonu $x(t)$ moguće je predstaviti kao linearnu kombinaciju svojstvenih vektora

$$x(t) = \sum_{i=1}^n c_i(t)v_i,$$

u svakom trenutku t , gde su $c_i(t)$ vremenski zavisni koeficijenti u razvoju. Zamenom ove linearne kombinacije u jednačini 4.17 dobije se sledeći izraz

$$\ddot{x}(t) = \sum_{i=1}^n \ddot{c}_i(t)v_i = \mathcal{A}x(t) = \sum_{i=1}^n c_i(t)\mathcal{A}v_i$$

odakle se nalazi jednakost

$$\sum_{i=1}^n \ddot{c}_i(t)v_i = \sum_{i=1}^n c_i(t)\lambda_i v_i,$$

odnosno

$$\sum_{i=1}^n (\ddot{c}_i(t) - c_i(t)\lambda_i)v_i = 0.$$

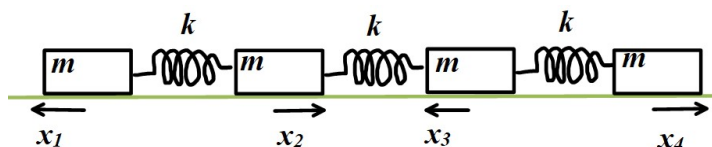
Iz uzlova da v_i čine svojstveni bazis dobija se $\ddot{c}_i(t) = c_i(t)\lambda_i$, $i = 1, \dots, n$. Rešenja ovako raspregnutih diferencijalnih jednačina zavise od predznaka λ_i . Tako, za $\lambda_i > 0$, $\mu_i = \sqrt{\lambda_i}$ rešenje je

$$c_i(t) = A_i e^{\mu_i t} + B_i e^{-\mu_i t}.$$

Kada je $\lambda_i < 0$ uvodi se $\omega_i = \sqrt{-\lambda_i}$, a rešenja jednačine je

$$c_i(t) = A_i \sin \omega_i t + B_i \cos \omega_i t,$$

gde se koeficijenti A_i, B_i u oba slučaja nalaze iz početnih uslova.



Slika 4.23: Sistem tela istih masa, vezanih istim oprugama, koja mogu oscilovati duž pravca.

Za svako telo u linearnom sistemu tela vezanih elastičnim oprugama, položenih na glatkoj ravni, prema II Njutnovom zakonu piše se dinamička jednačina

$$\begin{aligned} m\ddot{x}_1(t) &= k(x_2(t) - x_1(t)) \\ m\ddot{x}_2(t) &= k(x_3(t) - x_2(t)) + k(x_1(t) - x_2(t)) \\ &\vdots \\ m\ddot{x}_n(t) &= k(x_{n-1}(t) - x_n(t)). \end{aligned}$$

Opšte rešenje za $x_i(t)$ dato je u obliku $x_i(t) = A_i e^{i\omega t}$, gde je A_i amplituda pomeraja i -tog tela tokom oscilovanja sistema frekvencijom ω . Zamenom opšteg rešenja u sistem diferencijalnih jednačina, a zatim njegovim sređivanjem, dobija se sledeći sistem linearnih algebarskih jednačina

$$\begin{aligned} -\omega^2 A_1 &= k/m(A_2 - A_1) \\ -\omega^2 A_2 &= k/m(A_3 - A_2) + k(A_1 - A_2) \\ &\vdots \\ -\omega^2 A_n &= k/m(A_{n-1} - A_n), \end{aligned}$$

sa n nepoznatih amplituda i frekvencijom oscilovanja ω . Ovo je sistem n linearnih algebarskih jednačina sa $n + 1$ nepoznatom. Njegovim rešavanjem nalaze se sopstvene frekvencije oscilovanja sistema spregnutih oscilatora u određenog moda i odgovarajućih pomeraja tela. Ovaj algebarski sistem napisan je matrično u sledećem obliku:

$$\omega^2 \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix} = \frac{k}{m} \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \dots \\ -1 & 2 & -1 & 0 & 0 \dots \\ \vdots & & & & \dots \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix},$$

a rešavnja sistema algebarskih jednačina svodi se na rešavanje svojstvenog problema matrice

$$\mathcal{D}\mathcal{A} = \omega^2\mathcal{A},$$

gde je ω^2 svojstvena vrednosti, a pomeraji tela komponente svojstvenog vektora \mathcal{A} . Očekivano je n svojstvenih moda linearnog oscilatora, što je jednako broju spregnutih tela. Takođe, za sistem na slici 4.23 očekivano je da jedna moda bude čista translacija, to jeste da jedan svojstveni vektor ima sve komponente iste. Pri ovakvom pomeranju tela opruge su nedeformisane, a svojstvena frekvencija je nula.

Program počinje definisanjem matrice \mathcal{D} čiji se svojstveni problem rešava. Najpre joj se dodeli nulta matrica formata $n \times n$. Potom se *Do* petljom prelistavaju njeni elementi i definišu oni nenulti, dijagonalni $\mathcal{D}_{ii} = \begin{cases} 2, & i \neq 1 \wedge i \neq n \\ 1 & i = 1 \vee i = n \end{cases}$ i elementi $\mathcal{D}_{ij} = -1, |i - j| = 1$.

```
n = 7; (*broj tela u spregnutom linearnom oscilatoru*)
(* radi se u jedinicama m=k=1*)
Dmat = 0*IdentityMatrix[n];
Do[If[i === j, Dmat[[i, j]] = 2,
If[Abs[i - j] === 1, Dmat[[i, j]] = -1]], {i, n}, {j, n}];
(*formiranje matrice D*)
Dmat[[1, 1]] = Dmat[[n, n]] = 1;
MatrixForm[Dmat]
```

Sledi rešavanje svojstvenog problema matrice \mathcal{D} uz pomoć naredbe *Eigensystem*, čiji je izlaz spektar i skup svojstvenih vektora, koji redom odgovaraju elementima spektra. Ovi skupovi su transponovani, pri čemu je formiran skup parova svojstvena vrednost- svojstveni vektor. Svojtvene vrednosti se korenuju da bi se dobila kružna frekvencija dalje upotrebljiva u simulaciji i formirana je lista, sortirana po rastućim frekvencijama.

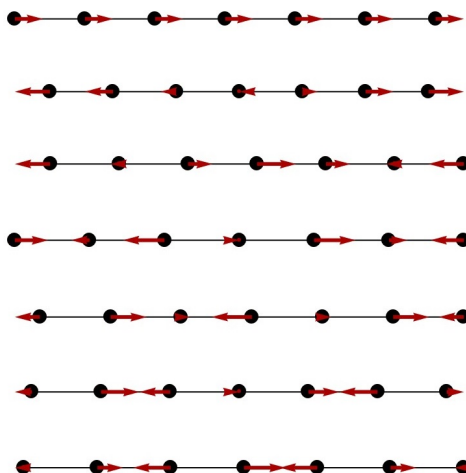
```
eig = Eigensystem[1. Dmat]; (*resavanje svojstvenog problema*)
eig = Transpose[eig];
(*grupisanje {svojstvena vrednost,vektor}*)
eig = Table[{Sqrt[eig[[i, 1]]], eig[[i, 2]]}, {i, Length[eig]}];
eig = Sort[eig]; (*lista slozena prema rastucim frekvencijama*)
```

Mode spregnutih tela ilustrovane su uz pomoć prostih grafičkih objekata u ravni. Izabere se moda i vektori pomeraja se postave u centre 7 tačaka, definisanih listom *coo*. Potom je formirana lista parova koji sadrže koordinatu i amplitudu pomeraja duž x - ose.

```
j=7; (*redni broj mode koja se ilustruje*)
w=eig[[j,1]]; (*frekvencija mode*)
coo=Table[{3i,0},{i,n}];
(*koordinate tela postavljjenih duz X ose*)
vecs=Transpose[{Table[coo[[i]], {i,Length[eig[[j,2]]}], eig[[j,2]]}];
```

Pomeraji su reprezentovani crvenim strelicama, crtanim iz centra tačaka pozicioniranih u (x_i, y_i) i usmerenih duž x - ose. Zadati su sa $norm\{vecs[[j,2]], 0\}$, gde je *norm* promenljiva kojom se sistematski menjaju dužine vektora radi što bolje ilustracije a *vecs[[j,2]]* su pomeraji tela j -te mode.

```
norm=2.>(*faktor normalizacije radi poboljšanja vizuelnog efekta*)
Show[Graphics[
Table[{Darker[Green],PointSize[0.025],Point[coo[[j]]]},
{j,Length[coo]}]],
Graphics[Table[{Arrowheads[0.025],
Arrow[{coo[[j]],(coo[[j])+norm{vecs[[j, 2]], 0}]}]},
{j,Length[coo]}]]]
```



Slika 4.24: Pomeraji tela normalnih moda sistema 7 kuplovanih oscilatora, odozgo nadole od nižih ka višim frekvencijama.

Na slici 4.24 su prikazani pomeraje tela duž poravca oscilovanja različitih moda, od najmanje (čiste translacije) ka višim frekvencijama. Prva moda odgovara nultoj frekvenciji dok su svi pomeraji tela u istoj fazi i iste amplitude, a tela se kreću translatorno i opruge su nedeformisane.

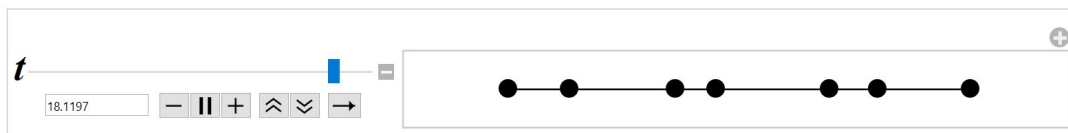
Animirano je kretanje tela (tačaka) promenom njihovog položaja u zavisnosti od kontinualnog parametra t (vremena). Svakoju tački je umesto strelice dodata promena položaja srazmerna amplitudi A_i i multiplikovana funkcijom od vremena

$(x_i, y_i) + norm A_i \sin \omega t$, $i = 1, \dots, n$, gde je ω svojstvena frekvencija. Vreme je nezavisno promenljiva kojom se interaktivno manipuliše u naredbi *Manipulate*, a kao rezultat animirani su kontinualni pomeraji tačaka.

```
(*animacija*)
pairs=Table[{i,i+1},{i,n-1}];
Manipulate[
coodyn =
Table[coo[[j]]+Sin[w*t]norm{vecs[[j,2]],0},{j,Length[vecs]}];
Show[Graphics[
Table[{Black, PointSize[0.03], Point[coodyn[[j]]]},
{j,Length[coodyn]}]],
Graphics[
Table[{Black, Line[coodyn[[pairs[[i1]]]]}], {i1, Length[pairs]}]],
PlotRange -> {{0, 24}, {-0.5, 0.5}}, {t, 0, 20}]
```

4.6.92 Digitalna obrada slika.

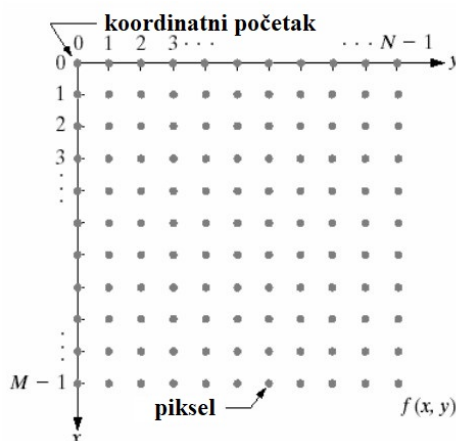
Crno bela digitalna slika reprezentuje se matrično, kao složena dvostruka lista brojeva (piksela). Svaki element te liste na slici čini tačku čije su koordinate određene indeksima matričnog elementa, dok njegova vrednost određuje nijansu od belog do crnog piksela. Slika u boji može biti



Slika 4.25: Animacija izabranog moda oscilovanja sa kontrolnim panelom.

zadata u takozvanom RGB modelu, u kom je umesto matričnog elementa zadata lista koja sadrži tri komponente (broja) $\{r, g, b\}$. Komponentama ove liste određen je doprinos crvene r , zelene g i plave b boje. Sve tri boje mogu da se kombinuju u različitim razmerama kako bi se dobila boja vidljivog spektra. Tako u RGB modu svaki piksel obuhvata podpiksele koji prikazuju crvenu, zelenu i plavu različitim intenzitetima.

- i) Učitati RGB sliku, zatim napraviti algoritam za njeno filtriranje. Filtriranjem slike izdvojiti komponente slike: crvenu, zelenu i plavu, a potom ih izvesti i snimiti ih kao posebne slike u istu fasciklu odakle je učitana početna.
- ii) Učitano RGB sliku konvertovati u crno-belu i snimiti u fasciklu u kojoj se nalazi inicijalna.
- iii) Učitati sliku, potom izvršiti njenu vertikalnu i horizontalnu refleksiju, a dobijene slike snimiti u fasciklu odakle je povučena inicijalna slika.



Slika 4.26: Reprzentacija digitalne slike, čije koordinate odgovaraju položaju piksela u dvostrukoj listi.

- i) Slika koja se obrađuje smeštena je u fascikli zajedno sa sveskom u kojoj je pisan dati programski kod. Najpre je na početku programa učitana putanja do sveske nazvana *filepath* uz pomoć naredbe

$$\text{NotebookDirectory}[].$$

Potom su prelistana imena svih fajlova ekstenzije *.jpg* iz fascikle u koju vodi izdvojena putanja

$$\text{files} = \text{FileNames}["*.jpg", \text{filepath}],$$

u kom je smeštena samo slika koja se obrađuje. Ona se učitava iz selektovanog fajla i dodjeljuje promenljivoj

$$\text{img} = \text{Import}[\text{files}[[1]]].$$

Složena, dvostruka lista piksela, koja odgovara učitanoj slici *img*, u vidu RGB trojki pridružena je

$$\text{imgdat} = \text{ImageData}[\text{img}],$$

a potom nacrtana u maloj veličini sa

Image].

```
(*ucitavanje putanje foldera gde je smestena sveska*)
filepath=NotebookDirectory[];
(*izlistavanje svih fajlova .jpg u fascikli ucitane putanje*)
files=FileNames["*.jpg",filepath];
(*ucitavanje prve slike na spisku*)
img=Import[files[[1]]];
(*uzimanje podataka slike*)
imgdat=ImageData[img];
(*format matrice u kojoj je slika zapisana*)
dim=Dimensions[imgdat];
Image

```



Slika 4.27: Učitana slika sa strukturom RGB, iz fascikle gde je smeštena zajedno sa sveskom.

Izdvajanjem neke od komponenti može se realizovati intervencijom na RGB trojku svakog piksela. Tako recimo, ukoliko je cilj primeniti zeleni filter, proizvoljni piksel $\{r, g, b\} \rightarrow \{0, g, 0\}$. Tehnički ovo je realizovano na sledeći način $\{0, g, 0\} = \{r, g, b\}\{0, 1, 0\}$. Slično, crveni filter dobijen je sa $\{r, 0, 0\} = \{r, g, b\}\{1, 0, 0\}$ dok je plavi $\{0, 0, b\} = \{r, g, b\}\{0, 0, 1\}$. Realizujući bilo koji od filtera, gornje operacije potrebno je primeniti na svaki piksel $imgdat[[i, j]]$, gde su brojači definisani $i = 1, \dots, dim[[1]]$, $j = 1, \dots, dim[[2]]$. Promenljiva $dim = Dimensions[imgdat]$ sadrži dimenzije matrice koja čini strukturu slike $imgdat$. Slike dobijene filtriranjem eksportovane su u vidu malog formata naredbom *Export* u fasciklu sa odgovarajućim nazivom. Tako je, na primer putanja za plavi filter $filepath <> "BlueFilter.jpg"$.

```
(*crvena komponenta*)
redimg=Table

```

ii) Crno-bela slika ima strukturu matrice, čiji su matrični elementi između 0 i 1. Trojka RGB



Slika 4.28: Sleva nadesno dobijene slike primenom crvenog zelenog i plavog filtera.

svakog piksela od učitane slike konvertovane u digitalne podatke *imgdat* je preslikana u 1 broj skalarnim množenem $\{r, g, b\} \cdot \{0.2989, 0.5870, 0.1140\}$.

```
(*koristiti NTSC standard
{r,g,b}·{0.2989,0.5870,0.1140}*)
imggray=
Table[imgdat[[i,j]]·{0.2989,0.5870,0.1140},{i,dim[[1]]},{j,dim[[2]]}];
imggray=Image[imggray];
Export[filepath<>"Gray_NTSC_Filter.jpg",imggray,ImageSize->Small];
```



Slika 4.29: Rezultat konvertovanja učitane fotografije u crno-belu.

- iii) Najlakši način da se realizuju transformacije slike poput vertikalne ili horizontalne refleksije je da na vektore položaja piksela, koje možemo poistovetiti sa indeksima matrice, delujemo odgovarajućim operatorom. Tako, horizontalno okretanje slike biće izvršeno slaganjem svake vrste $i = 1, \dots, \text{dim}[[1]]$ unazad $\text{Reverse}[\text{imgdat}[[i]]]$. Za vertikalno okretanje slike korišćene su dodatne manipulacije. Najpre je izvršeno transponovanje, pri čemu su kolone postale vrste $\text{imgdatTrns} = \text{Transpose}[\text{imgdat}]$, a potom su pikseli svih novodobijenih vrsta redom posloženi unazad $\text{Reverse}[\text{imgdatTrns}[[i]]]$. Zatim se ponovo primeni zamena vrsta i kolona $\text{Transpose}[\text{imgdatTrns}]$, nakon čega je dobijena slika vertikalno okrenuta u odnosu na početnu.

```

(*vertikalna refleksija*)
imgVref=Table[Reverse[imgdat[[i]]],{i,dim[[1]]}];
imgVref=Image[imgVref]
Export[filepath<>"Slika_Vref1.jpg",imgVref,ImageSize->Small];
(*horizontalna refleksija*)
imgdatTrns=Transpose[imgdat];
imgdatTrns=Table[Reverse[imgdatTrns[[i]]],{i,dim[[2]]}];
imgdatTrns=Transpose[imgdatTrns];
imgdatTrns=Image[imgdatTrns]
Export[filepath<>"Slika_Vref1.jpg",imgdatTrns,ImageSize->Small];

```



Slika 4.30: Sleva nadesno, slike dobijene refleksijom u odnosu na vertikalnu odnosno horizontalnu liniju.

4.6.93 Ortonormalizacija.

- i) Skup vektora $\{t^i | i = 0, 1, 2, \dots, n\}$ ortonormirati u odnosu na zadati skalarni proizvod

$$(f(x), g(x)) = \int_a^b f(x)g(x)dx.$$

- ii) Zadatu funkciju $f(x)$, definisanu na intervalu $[a, b]$, aproksimirati njenim razvojem u linearnu kombinaciju po dobijenim ortonormiranim polinomima. Potom $f(x)$ i njenu aproksimaciju prikazati zajedno, na istom grafiku na zadatom intervalu $x \in [a, b]$.
- i) Postupkom ortonormalizacije omogućava se rad sa ortonormiranim vektorima u prostoru sa zadatim skalarnim proizvodom. Njime se iz proizvoljnog bazisa $X = \{x_1, x_2, \dots, x_n\}$ prostora V_n konstruiše ortonormirani bazis $on\beta = \{e_1, e_2, \dots, e_n\}$, $(e_i, e_j) = \delta_{ij}$. Prvi ort dobija se direktno normalizacijom prvog vektora

$$e_1 = \frac{x_1}{\|x_1\|}, \quad \|x_1\| = \sqrt{(x_1, x_1)}.$$

Ukoliko je nađeno k -ortonormiranih vektora ($k \leq n$) e_1, e_2, \dots, e_k , sledeći se bira tako da je ortogonalan na prethodno nađene ortove

$$y_{k+1} = x_{k+1} - \sum_{l=1}^k \alpha_l e_l.$$

Uzevši u obzir da je $0 = (e_i, y_{k+1}) = (e_i, x_{k+1}) - \sum_{l=1}^k \alpha_l (e_i, e_l) = (e_i, x_{k+1}) - \sum_{l=1}^k \alpha_l \delta_{il} = (e_i, x_{k+1}) - \alpha_i = 0$, $i = 1, 2, \dots, k$ dobiju se koeficijenti linearne kombinacije određenih ortova

$$\alpha_i = (e_i, x_{k+1}).$$

Konačno, nalazi se vektor ortogonalan na prethodno određene

$$y_{k+1} = x_{k+1} - \sum_{l=1}^k (e_l, x_{k+1}) e_l, \quad i = 1, 2, \dots, k,$$

a nakon normalizacije dobije se ort $e_{k+1} = \frac{y_{k+1}}{\|y_{k+1}\|}$. Ukratko, svaki naredni ort dobija se najpre razlikom predstojećeg vektora od njegove projekcije na prethodno određene ortove, a zatim se dobijena razlika normira. Ovaj postupak se ponavlja sve dok se ne iscrpe svi vektori zadatog skupa a rezultat je ortonormiran bazis $\{e_1, e_2, \dots, e_n\}$ sa svojstvom $(e_i, e_j) = \delta_{ij}$. Zadatak je da se u odnosu na zadati skalarni proizvod ortonormira set polinoma. Najpre je zadat maksimalni stepen polinoma iz kog je generisan apsolutni bazis polinoma *inset* koji se ortonormira u odnosu na zadati skalarni proizvod. Potom je definisan interval određenog integrala preko kog je zadat skalarni proizvod *SclProd*. Prva dva argumenta ove funkcije su polinomi, a treći varijabla preko koje su zadati.

```
(*zadavanje intervala i bazisa*)
n=3;
inset=Table[t^i,{i,0,n}];
a=-1;
b=1;
(*zadavanje skalarnog proizvoda*)
SclProd[f_,g_,var_]:=NIntegrate[f*g,{var,a,b};
```

Prethodno objašnjen teorijski algoritam nalaženja ortonormiranog bazisa pri zadatom skalarnom proizvodu implementiran je na ortonormalizaciju definisanog apsolutnog bazisa polinoma. Pre *Do* petlje, koja se dopunjava redom nađenim bazisnim vektorima formiran je prazan skup *onbaz*. Tokom iteracije od svakog vektora zadatog *inset[[i]]* oduzima se projekcija na do tada određene ortove $\text{Sum}[SclProd[onbaz[[j]], inset[[i]], t] * onbaz[[j]]$, a dobijena razlika dalje normira $e = y / \text{Sqrt}[SclProd[y, y, t]]$. Nađeni ort se dodaje skupu *onbaz* sa *AppendTo[onbaz, e]*.

```
onbaz = {};
Do[
(*od vektora se oduzima njegova projekcija na ortove*)
y=inset[[i]]-Sum[SclProd[onbaz[[j]],inset[[i]], t]*onbaz[[j]],
{j,Length[onbaz]};
e=y/Sqrt[SclProd[y,y,t]];
AppendTo[onbaz,e],{i,Length[inset]};
```

Drugачije, koristeći funkcije programa Mathematica zadatak se može rešiti uz pomoć naredbe

$$\text{Orthogonalize}[\{x_1, x_2, \dots, x_n\}, SclProd],$$

gde je rešenje ortonormiran skup dobijen ortonormalizacijom zadatog skupa vektora $\{x_1, x_2, \dots, x_n\}$ u odnosu na zadati skalarni proizvod *SclProd*. Drugi argument ove naredbe, funkcionalnim programiranjem zadaje se skalarno množenje. Potom je izvršena provera, svaki vektor je sa svakim kao i sam sa sobom pomnožen skalarno, realizovano naredbom *Table* preko dvostrukog brojača, za svaki faktor skalarnog proizvoda po jedan.

```
onbaz=Orthogonalize[inset,SclProd[#1,#2,t] &];
(*provera*)
Table[SclProd[onbaz[[i]], onbaz[[j]],t],{i,Length[onbaz]},
{j,Length[onbaz]} // Chop
```

- ii) Drugi deo zadatka je da se na intervalu na kom je definisan skup ortonormiranih funkcija, data funkcija aproksimirala njenim razvojem u linearnu kombinaciju \tilde{f} po dobijenim ortogonalnim polinomima prema relaciji

$$\tilde{f} = \sum_{i=1}^n (e_i, f) e_i.$$

Norma kompleksne funkcije f iz prostora funkcija definisanih na intervalu $[a, b]$ definisana je kao

$$\|f\|_\omega = \sqrt{\int_a^b \omega(x)|f(x)|^2 dx},$$

gde je težina $\omega(x) > 0$ pozitivna funkcija, a standardni skalarni proizvod zadat je na sledeći način

$$(f, g)_\omega = \int_a^b f(x)g(x)^* \omega(x) dx.$$

Treba zapaziti da u slučaju kad se radi o realnom prostoru, konjugacija funkcije pod integralom se izostavlja. Zadatak se svodi na traženje aproksimacione funkcije u otežinjenom L^2 prostoru definisanom na sledeći način

$$L_\omega^2([a, b]) = \{f : [a, b] \rightarrow \mathcal{C}, \int_a^b |f(x)|^2 < \infty\},$$

koji praktično sadrži sve funkcije. Skup funkcija $\{\varphi_i | i = 1, 2, \dots\}$ prostora $L_\omega^2([a, b])$ je takav da se proizvoljna funkcija $f \in L_\omega^2([a, b])$ može jedinstveno izraziti kao

$$f = \sum_{i=1}^{\infty} c_i \varphi_i,$$

u smislu da pracijalna suma $\sum_{i=1}^N c_i \varphi_i$ konvergira ka funkciji f , odnosno

$$\lim_{N \rightarrow \infty} \|f - \sum_{i=1}^N c_i \varphi_i\|_\omega = 0.$$

U operativnom smislu, rad sa ortogonalnim funkcijama je praktičan sa aspekta teorijskog ali ili numeričkog pristupa. Ako je $\{\varphi_i\}$ ortogonalan bazis za koji važi $(\varphi_i, \varphi_j) = 0$, $i \neq j$, da bismo proizvoljnu funkciju $f \in L^2([a, b])$ izrazili kao linearnu kombinaciju ovog bazisa

$$f = \sum_{j=1}^{\infty} a_j \varphi_j$$

potrebno je naći koeficijente a_i , $i = 1, 2, \dots$. Ako se leva i desna strana gornje jednakosti skalarno pomnoži sa φ_k

$$(\varphi_k, f) = (\varphi_k, \sum_{j=1}^{\infty} a_j \varphi_j) = \sum_{j=1}^{\infty} (\varphi_k, \varphi_j) a_j = (\varphi_k, \varphi_k) a_k,$$

nalazi se

$$a_k = \frac{(\varphi_k, f)}{(\varphi_k, \varphi_k)}.$$

Linearna kombinacija prvih N članova iz razvoja od f

$$f_N = \sum_{j=1}^N a_j \varphi_j$$

je najbolja aproksimacija funkcije f u potprostoru $S_N < L^2([a, b])$, obrazovanog vektorima $\{\varphi_1, \varphi_2, \dots, \varphi_N\}$, u smislu da $g = f_N$ minimizuje $\|f - g\|_\omega$ za $g \in S_N$. Iz uslova da je gore data, konačna linearna kombinacija najbolja aproksimacija, što znači da $f_N = \sum_{i=1}^N a_i \varphi_i$

minimizuje normu $\|f - g\|_\omega$ na celom S_N , moguće naći koeficijente a_i u razvoju. Treba imati u vidu da se minimum funkcije $E(c)$, gde je

$$E(c) = \|f - \sum_{i=1}^N a_i \phi_i\|^2 = \int_a^b (f - \sum_{i=1}^N a_i \phi_i)^2 \omega(x) dx$$

nalazi u tački u kojoj je prema gore datom uslovu minimuma gradijent od E nula. Primenujući ovaj kriterijum

$$0 = \frac{\partial E}{\partial a_j} = -2 \int_a^b (f - \sum_{i=1}^N a_i \phi_i) \phi_j \omega(x) dx, \quad j = 1, \dots, N$$

nakon sređivanja gornjeg izraza dobija se

$$\int_a^b f(x) \phi_j \omega(x) dx = \sum_{i=1}^N a_i \int_a^b \phi_i \phi_j \omega(x) dx, \quad j = 1, \dots, N.$$

Ukoliko se definišu matrični elementi matrice \mathcal{H}

$$h_{ij} = \int_a^b \phi_i \phi_j \omega(x) dx = (\phi_i, \phi_j),$$

a komponente vektora \mathbf{f}

$$f_i = \int_a^b f(x) \phi_i \omega(x) dx = (f(x), \phi_i),$$

pri čemu važi da je

$$\mathcal{H}\mathbf{a} = \mathbf{f}.$$

Da bi rešavanje zadatka bilo računski što stabilnije i efikasno, potrebno je da operator \mathcal{H} bude što jednostavnije reprezentovan. Najprostiji i računski najefikasniji je dijagonalni oblik matrice reprezentovanja, a redukcija na njega postiže se izborom ortogonalnog bazisa za koji važi da je $(\phi_i, \phi_j) = \delta_{ij}$. Izborom ortogonalnog bazisa $\{\phi_i\}$ dobija se sledeća matrična jednačina

$$\begin{pmatrix} h_{11} & 0 & 0 & 0 \dots \\ 0 & h_{22} & 0 & 0 \dots \\ \vdots & & & \dots \\ 0 & \dots & 0 & h_{NN} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}$$

iz koje se nalaze koeficijenti

$$c_i = \frac{f_i}{h_{ii}}, \quad i = 1, 2, \dots, N,$$

odnosno

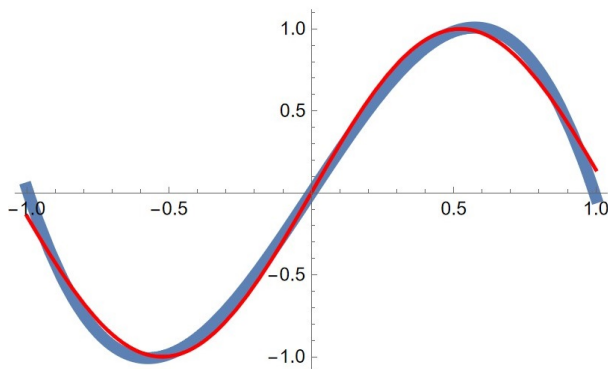
$$c_i = \frac{(f(x), \phi_i)}{(\phi_i, \phi_i)}.$$

Dati bazis *inset* je ortonormiran, te pored ortogonalnosti karakteriše ga i normiranost, odnosno $(\phi_i, \phi_i) = 1$, $i = 1, 2, \dots, N$. Što znači da se koeficijenti u razvoju najbolje aproksimacije zadate funkcije f računaju iz sledeće relacije

$$c_i = (f, \phi_i), \quad i = 1, 2, \dots, N.$$

```
f[x_]:=Sin[3x];
coeff=Table[SclProd[f[t],onbaz[[i]],t],{i,Length[onbaz]};
Show[Plot[{coeff.onbaz},{t,a,b},PlotStyle->Thickness[0.02]],
Plot[f[t],{t,a,b},PlotStyle->Red]]
```

Lista koeficijenata dobijena je uz pomoć listabilne funkcije *Table*, čiji su argumenti koeficijenti u linearnoj kombinaciji aproksimacije računati sa *SclProd*[*f*[*t*], *onbaz*[[*i*]], *t*]. Dobijena linearna kombinacija nađena je skalarnim množenjem liste koeficijenata i liste ortonormiranih polinoma *coeff.onbaz*, što je prikazano podebljano na intervalu $t \in [a, b]$ kao *Plot*[{*coeff.onbaz*}, {*t, a, b*}, *PlotStyle* → *Thickness*[0.02]]. Zadana funkcija *f* nacrtana je crvenom *Plot*[*f*[*t*], {*t, a, b*}, *PlotStyle* → *Red*] i sve zajedno prikazano je na istom grafiku.



Slika 4.31: Funkcija čija se aproksimacija traži prikazana je crvenom, a njena najbolja aproksimacija preko datog ortonormiranog bazisa, dobijena kao linearna kombinacija funkcija ortonormiranog bazisa, plavom podebljanom linijom.

Aproksimacija se poboljšava povećanjem broja polinoma n , što se vidi kao sve manje odstupanje zadate funkcije f od njene aproksimacije, dobijene kao linearna kombinacija ortonormiranog bazisa na definisanom intervalu $t \in [a, b]$.

5 Primeri implementacije numeričkih metoda u rešavanju zadataka

5.1 Dinamika višestičnog sistema- integracioni metod rešavanja Njutnovih jednačina kretanja.

Za mnoge oblasti istraživanja u fizici cilj je opisati makroskopska svojstva materije, rešavanjem dinamičkog problema na mikroskopskom nivou. Međutim, opis kretanja velikog broja intereagujućih čestica u većini slučajeva nije izvodljiv postojećim analitičkim metodama. Štaviše, opis je posebno težak ukoliko je interakcija između čestica jaka. Postoji pokušaj uklanjanja ove poteškoće primenom metoda statističke fizike uz pojednostavljujuće pretpostavke, tretiranjem sistema iz statističke tačke gledišta. Većina pojednostavljenih pretpostavki je opravdana samo pod određenim uslovima, kao što je na primer uslov slabe hemijske veze. Često nije jednostavno utvrditi kako neuračunati uslovi utiču na dobijena rešenja, te kako se izvan njih mogu sagledati pojave. Zbog toga se javlja potreba za numeričkim rešenjima koja obuhvataju sve relevantne uslove. Jedna od metoda određivanja fizičkih veličina iz ograničenog skupa stanja je *molekularna dinamika*. U nastavku biće obrađena podoblast molekularne dinamike koja tretira klasične fizičke sisteme. Pored klasične molekularne dinamike postoji i kvantna, koja ovde neće biti razmatrana.

Metodom *preskakanja*, simulirati dinamiku kretanja naelektrisanih čestica koje intereaguju Kulonovom interakcijom. Sila interakcije čestica zavisi od njihovog relativnog položaja. Kako se on menja sa vremenom, prema Njutnovom zakonu menjaće se i njihovo ubrzanje, što znači da je potrebno analizirati dinamiku sistema čestica koje se kreću neravnomerno ubrzano. Numerički verlet integracioni metod koristi se za integraciju Njutnovih jednačina kretanja, obično korišćen za traženje trajektorija intereagujućih čestica. Metod nosi naziv prema Loup Verletu, a počevši od 1960. godine koristi se za simulacije još poznatije kao molekularna dinamika. Verlet algoritmom se aproksimativno rešavaju diferencijalne jednačine drugog reda tipa $\ddot{x}(t) = a(x(t))$, pri čemu je neophodno poznavanje početnih uslova $x(t_0) = x_0$ i $\dot{x}(t_0) = v_0$. Nalaženje vrednosti $x_n = x(t_n)$ u trenutku $t = t_0 + n\Delta t$, gde je $\Delta t > 0$ korak promene, odnosno diskretizaciju po vremenu, što podrazumeva računanje promene kinematičkih veličina u uzastopnim trenucima koji se razlikuju za Δt , a potom primenu iteracije

$$x_1 = x_0 + v_0\Delta t + \frac{1}{2}a(x_0)\Delta t^2$$

$$x_{n+1} = 2x_n - x_{n-1} + a(x_n)\Delta t^2, \quad n = 1, 2, \dots,$$

gde je iskorišćen izraz za numeričko nalaženje drugog izvoda prema relaciji $a = \ddot{x} \approx \frac{x_{n+1} - 2x_n + x_{n-1}}{\Delta t^2}$.

Pored brzine, algoritam podrazumeva i računanje pozicije kao i ubrzanja u svakom narednom trenutku. Molekularna dinamika zahteva poznavanje potencijala, odnosno položaja svih čestica od čega zavisi intenzitet njihove međusobne interakcije. Potencijalna energija je u vezi sa silama interakcije čestica. Ukoliko rad sile koja deluje na telo i pomera ga od tačke A do B, ne zavisi od

pređenog puta između ovih tačaka, reč je o konzervativnoj sili. Rad primenjene sile duž neke trajektorije od tačke A do B, nezavisan od putanje, može se izraziti kao

$$A = \int_A^B \vec{F} d\vec{r} = U(\vec{r}_A) - U(\vec{r}_B),$$

gde je $U(\vec{r})$ skalarna funkcija ili potencijal. Iz prethodnog izraza vidi se da je rad konzervativnih sila, među koje spadaju gravitaciona i elektrostatička, izvršen na zatvorenoj putanji jednak nuli. Sila se u ovom slučaju može matematički formulisati kao negativan gradijent potencijalnog polja

$$\vec{F} = -\nabla U = -\left(\frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}, \frac{\partial U}{\partial z}\right). \quad (5.1)$$

Metod će biti razrađen na primeru pravolinijskog kretanja u polju konzervativne sile koja zavisi od položaja tela. Potom će rezultat biti uopšten na dinamiku višestičnog sistema u trodimenzionalnom prostoru, pri čemu će biti uračunata konzervativna interakcija između čestica sistema. Često korišćena varijanta ovog algoritma je *verlet*, u kom se pored položaja i ubrzanja čestica, shodno njihovom novom položaju, računaju i brzine. Verlet integracioni algoritam je u osnovi rešenje kinematičkih jednačina kretanja proizvoljnog objekta,

$$x = x_0 + v_0 t + \frac{1}{2} a t^2 + \frac{1}{6} b t^3 + \dots \quad (5.2)$$

gde je x položaj, t vreme, v brzina, a ubrzanje, a b je član koji isčezava u slučaju ravnomerno ubrzanog kretanja. Ovo je centralni izraz za rešavanje Njutnovih jednačina u delovima algoritama poznatijim kao *integratori sile*. Poznavanjem kinematskih veličina u trenutku t , postavlja se pitanje kako naći njihove vrednosti u sledećem bliskom trenutku $t + \Delta t$. Razvojem zavisnosti položaja od vremena u Tejlorov red dobije se izraz

$$x(t + \Delta t) = x(t) + \dot{x}(t)\Delta t + \frac{1}{2}\ddot{x}(t)\Delta t^2 + \frac{1}{6}\dddot{x}(t)\Delta t^3 + \mathcal{O}(\Delta t^4), \quad (5.3)$$

gde su $\dot{x}(t) = v(t)$, $\ddot{x}(t) = a(t)$, $\dddot{x}(t) = b(t)$. Da bi se našla pozicija $x(t + \Delta t)$ sa greškom reda $\mathcal{O}(\Delta t^3)$ potrebno je da budu poznate trenutne vrednosti položaja, brzine i ubrzanja označenih redom sa $x(t), v(t), a(t)$. Slično, za nalaženje prethodnog položaja, može se primeniti aproksimacija

$$x(t - \Delta t) = x(t) - v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 - \frac{1}{6}b(t)\Delta t^3 + \mathcal{O}(\Delta t^4). \quad (5.4)$$

Kombinovanjem datih jednačina (5.3) i (5.4) dobija se izraz

$$x(t + \Delta t) + x(t - \Delta t) = 2x(t) + a(t)\Delta t^2 + \mathcal{O}(\Delta t^4),$$

iz kog se $x(t + \Delta t)$ može naći do na numeričku grešku ako su poznate pozicije u trenucima t i $t - \Delta t$ kao i ubrzanje $a(t)$

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + a(\Delta t)\Delta t^2 + \mathcal{O}(\Delta t^4).$$

Ukoliko je potrebno izračunati kinetičku energiju, što zahteva poznavanje brzine, može se koristiti Stormer-Verlet metod. Rezultat razlike jednačina (5.3) i (5.4) je

$$x(t + \Delta t) - x(t - \Delta t) = 2v(t)\Delta t + \frac{1}{3}b(t)\Delta t^3,$$

odakle se trenutna brzina izražava sa

$$v(t) = \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2).$$

Iz jednačina

$$v(t + \Delta t/2) = v(t) + a(t) \frac{\Delta t}{2}, \quad (5.5)$$

$$x(t + \Delta t) = x(t) + v(t)\Delta t + a(t) \frac{\Delta t^2}{2} = x(t) + (v(t) + a(t) \frac{\Delta t}{2})\Delta t, \quad (5.6)$$

dobija se izraz koji opisuje promenu položaja položaj čestice sa vremenom

$$x(t + \Delta t) = x(t) + v(t + \Delta t/2)\Delta t.$$

Potencijalna energija čestice u novom položaju se razlikuje u odnosu na prethodni kao i sila koja deluje na nju. Prema Njutnovom zakonu ubrzanje u trenutku $t + \Delta t$ nalazi se iz

$$a(t + \Delta t) = -\frac{\nabla U(x(t + \Delta t))}{m}. \quad (5.7)$$

Brzina u trenutku $t + \Delta t$ prema gore datoj relaciji biće

$$v(t + \Delta t) = v(t + \Delta t/2) + a(t + \Delta t)\Delta t/2,$$

a zamenom $v(t + \Delta t/2)$ iz (5.5) dobije se izraz za brzinu u sledećem trenutku

$$v(t + \Delta t) = v(t) + \frac{a(t) + a(t + \Delta t)}{2}\Delta t, \quad (5.8)$$

iz kog se vidi da je za brzinu u sledećem trenutku potrebno naći srednje ubrzanje $\frac{a(t) + a(t + \Delta t)}{2}$ na intervalu $[t, t + \Delta t]$. Iz prethodne analize proizilazi algoritam:

- 1) $t_0 = 0$: x_0, v_0, a_0 zadati početni uslovi;
- 2) t : $x(t), v(t), a(t)$ iterativno dobijene kinematičke veličine u trenutku t ;
- 3) (a) $x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2$
 (b) $a(t + \Delta t) = -\frac{\nabla U(t + \Delta t)}{m}$
 (c) $v(t + \Delta t) = v(t) + \frac{a(t) + a(t + \Delta t)}{2}\Delta t.$

Generalizovano za sistem od n čestica, koje uzajamno intereaguju konzervativnim silama važiće sličan algoritam. Položaji ovih tela u trenutku t su $\vec{r}_k(t) = (x_k(t), y_k(t), z_k(t))$, $k = 1, 2, \dots, n$. Neka je kolona

$X(t) = (x_1(t), y_1(t), z_1(t), x_2(t), y_2(t), z_2(t), \dots, x_n(t), y_n(t), z_n(t))^T$, a

$\mathcal{M} = \text{diag}(m_1, m_2, \dots, m_n)$ dijagonalna matrica čiji su blokovi mase čestica m_k , $k = 1, 2, \dots, n$. Njutnove jednadžine kretanja za svaku pojedinačnu česticu

$$m_k \ddot{\vec{r}}_k(t) = \vec{F}_k(X(t)) = -\nabla_{\mathbf{r}_k} U(X(t)), \quad k = 1, 2, \dots, n$$

ili, generalisano za ceo sistem čestica važi

$$\mathcal{M} \ddot{X}(t) = -\nabla U(X(t)).$$

Mase čestica (tela) kao i njima karakteristični potencijali interakcije variraju u zavisnosti od izbora sistema, što predloženi algoritam čini pogodnim za opis njihove evolucije, od interakcije molekula do orbita planeta.

5.1.94 Simulacija kretanja dve čestice atraktivne interakcije.

Predstoji primer simulacije dinamike sistema dve naelektrisane čestice, koje intereaguju Kulonovom interakcijom, gde je potencijal dat kao

$$\varphi = k \frac{q_1 q_2}{r}.$$

Najpre su zadate karakteristike čestica, njihova naelektrisanja, mase i konstanta k sve izraženo u arbitrarnim jedinicama. Potom je definisana funkcija skalarnog potencijala $\varphi[x_-, y_-, z_-, q_-]$, koja zavisi od količine naelektrisanja q i koordinata relativnog položaja čestica $\vec{r}_1 - \vec{r}_2 = (x, y, z)^T$, gde su \vec{r}_1 i \vec{r}_2 vektori položaja intereagujućih čestica. Osim početnog položaja zadate su i početne brzine v_i i inicijalna ubrzanja dobijena prema Njutnovom zakonu

$$\vec{a}_i = \frac{\vec{F}_i}{m_i},$$

pri čemu je sila \vec{F}_i koja deluje na česticu i mase m_i i naelektrisanja q_i u električnom polju \vec{E}_j čestice j , računata prema obrazcu

$$\vec{F}_i = q_i \vec{E}_j, \quad \vec{E}_j = -\nabla \varphi_j(r_{ji}).$$

Sve ove relacije implementirane u programu, definišu početne veličine, neophodne za rešavanje dinamike čestica

$$E_i = -\text{Grad}[\varphi[x, y, z, q_i], \{x, y, z\}];$$

$$a_i = (E_j * q_i / m_i) / \text{Thread}[\{x, y, z\} -> r_{ij}]; \quad i, j = 1, 2$$

pri čemu se najpre računa električno polje u položaju svake čestice a potom njeno ubrzanje $\vec{a}_i = \vec{E}_j q_i / m_i$. Argumenti x, y, z vektorske funkcije električnog polja zamenjene su komponentama vektora relativnog položaja $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$. Komponente vektora početne brzine i položaja, u vidu početnih uslova, kao i karakteristike čestica proizvoljne su i mogu se menjati u zavisnosti od toga šta se ispituje. Ovim je pripremljena prva etapa predloženog algoritma.

```
(*karakteristike cestica*)
q1=1;(*naelektrisanje od 1.*)
m1=10;(*masa od 1.*)
q2=-1;(*naelektrisanje od 2.*)
m2=1;(*masa od 1.*)
k=1;(*konstanta u arb. jedinicama*)
(*funkcija elektrostatickog potencijala*)
\[CurlyPhi][x_,y_,z_,q_]:=k*q/Sqrt[x^2+y^2+z^2];

(*pocetni uslovi:u t=0: x0,v0,a0*)
(*pocetni položaji*)
r1=0{1,1,1};
r2={-2,-2,-2};
r12=r1-r2;
r21=r2-r1;
(*pocetne brzine*)
v1=0.05{1,1,-1};
v2=0.5{1,0,0.3};
(*elektricno polje*)
E1=-Grad[\[CurlyPhi][x,y,z,q1],{x,y,z}];
E2=-Grad[\[CurlyPhi][x,y,z,q2],{x,y,z}];
(*ubrzanja u t=0*)
a1=(E2*q1/m1)/.Thread[{x,y,z}->r12];
a2=(E1*q2/m2)/.Thread[{x,y,z}->r21];
```

Svakom sledećom iteracijom, iznova se iz veličina $\vec{r}_i(t), \vec{v}_i(t), \vec{a}_i(t)$ definisanih u trenutku t , računaju kinematičke veličine u predstojećem trenutku $t + \Delta t$. Vremenski interval između dva trenutka Δt u kojima se računaju relevantne veličine bira se tako da ima što manju vrednost. Od broja iteracija N_{iter} zavisi do kog trenutka se simulira evolucija sistema $t_N = t_0 + N_{iter} \Delta t$. Pre početka *Do* petlje (iteracija) otvoren je prazan skup *rez* u koji se sukcesivno sipaju rezultati

$\{\vec{r}_1, \vec{v}_1, \vec{a}_1\}, \{\vec{r}_2, \vec{v}_2, \vec{a}_2\}$ dobijeni prethodnim izvršenjem algoritma. Odmah po otvaranju *Do* petlje razrađena je promena položaja čestica a zatim i traženje relativnog položaja neophodnog za nalaženje intenziteta međučestične interakcije pa i njihovog ubrzanja u predstojećem trenutku $t + \Delta t$

$$r_1 = r_1 + v_1 * dt + 0.5 * a_1 * dt^2;$$

$$r_2 = r_2 + v_2 * dt + 0.5 * a_2 * dt^2;$$

$$r_{12} = r_1 - r_2;$$

$$r_{21} = r_2 - r_1;$$

Nakon ažuriranja položaja, na red dolazi i rekalkulacija sila odnosno ubrzanja u sledećem trenutku, neophodnih za dobijanje srednjeg ubrzanja na intervalu $[t, t + \Delta t]$. Nakon toga se nalaze brzine predstojećeg trenutka

$$a_1 t dt = (E_2 * q_1 / m_1) / .Thread[\{x, y, z\} \rightarrow r_{12}];$$

$$a_2 t dt = (E_1 * q_2 / m_2) / .Thread[\{x, y, z\} \rightarrow r_{21}];$$

$$v_1 = v_1 + 0.5 * (a_1 + a_1 t dt) * dt;$$

$v_2 = v_2 + 0.5 * (a_2 + a_2 t dt) * dt$. Ubrzanje u $t + \Delta t$ dodeljuje se čestici u tekućem trenutku t , iz kog se ponovo ažurira novi položaj. Tako, iz jedne u drugu iteraciju N_{iter} puta, pri čemu se svaki put ažuriraju podaci kinematičkih veličina i upisuju u otvoren skup

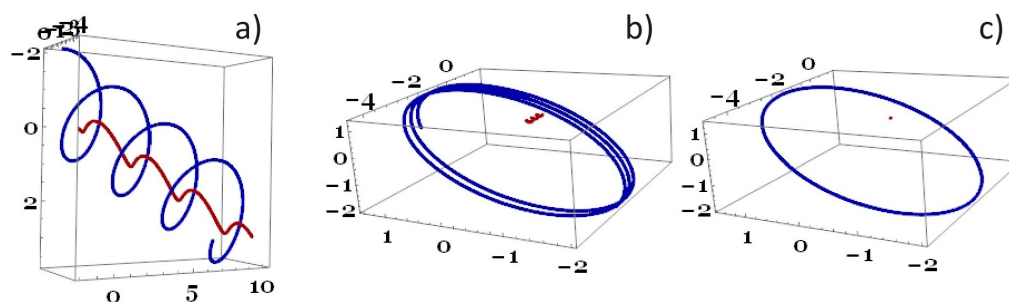
`AppendTo[rez, {{r1, v1, a1}, {r2, v2, a2}}]`.

```
(*broj iteracija*)
Niter=40000;
(*delta t*)
dt=0.002;
(*skup za deponovanje rezultata*)
rez={};
Do[
(*polozaj u t+dt*)
r1=r1+v1*dt+0.5*a1*dt^2;
r2=r2+v2*dt+0.5*a2*dt^2;
r12=r1-r2;
r21=r2-r1;
(*ubrzanje u t+dt*)
a1tdt=(E2*q1/m1)/.Thread[{x,y,z}->r12];
a2tdt=(E1*q2/m2)/.Thread[{x,y,z}->r21];
v1=v1+0.5*(a1+a1tdt)*dt;
v2=v2+0.5*(a2+a2tdt)*dt;
(*nova ubrzanja*)
a1=a1tdt;
a2=a2tdt;
AppendTo[rez,{{r1,v1,a1},{r2,v2,a2}},{Niter}];
```

Formirana lista *rez* na mestima $rez[[i,1,1]]$ i $rez[[i,2,1]]$, $i = 1, 2, \dots, N_{iter}$ sadrži prostorne koordinate čestica u trenutku $t_i = t_0 + i\Delta t$. Ekstrakcijom svih koordinata svake čestice ponaosob $rez[[All,1/2,1]]$, dobije lista uzastopnih položaja čestica tokom intervala $t \in [0, N_{iter}\Delta t]$, koje će poslužiti da se na njemu ilustruju njihove trajektorije. Radi lakšeg vizuelnog praćenja putanje su obojene različitim bojama.

```
Show[
ListPointPlot3D[rez[[All,1,1]]],
ListPointPlot3D[rez[[All,2,1]]],
PlotStyle->Darker[Blue],PlotRange->All]
```

Iz numeričkih podataka dodeljenih listi *rez*, koji sadrže informacije o položaju, brzini i u ubrzanju pojedinih čestica u različitim trenucima, moguće je napraviti animaciju njihovog kretanja uz pomoću naredbe `Manipulate[expres, {i, 1, Niter, 1}]`. Pomoću brojača zadatog kao argument ove



Slika 5.1: Crvenim linijama prikazana trajektorija pozitivnih a plavom negativnih naelektrisanja dobijenih pod različitim početnim uslovima i karakteristikama čestica: a) $q_1 = 1; m_1 = 5; q_2 = -1; m_2 = 1; \vec{r}_1 = (0, 0, 0)^T; \vec{r}_2 = (-2, -2, -2)^T; \vec{v}_1 = (0.05, 0.05, -0.05)^T; \vec{v}_2 = (0.5, 0, 0.02)^T$; b) $q_1 = 1; m_1 = 50; q_2 = -1; m_2 = 1; \vec{r}_1 = (0, 0, 0)^T; \vec{r}_2 = (-2, -2, -2)^T; \vec{v}_1 = (0, 0, 0)^T; \vec{v}_2 = (0.5, 0, 0.02)^T$; c) $q_1 = 1; m_1 = 5000; q_2 = -1; m_2 = 1; \vec{r}_1 = (0, 0, 0)^T; \vec{r}_2 = (-2, -2, -2)^T; \vec{v}_1 = (0, 0, 0)^T; \vec{v}_2 = (0.5, 0, 0.02)^T$.

funkcije biće pozivane redom koordinate odnosno svaki položaj plave i crvene izračunat na zadatom vremenskom intervalu

```
Graphics3D[{Darker[Red], PointSize[0.04], Point[rez[[i, 1, 1]]]}];
Graphics3D[{Blue, PointSize[0.03], Point[rez[[i, 2, 1]]]}],
```

za koju su karakteristike čestica date u opisu slike 5.1 pod c). U ovom primeru pozitivno naelektrisanje (crvena čestica) ima 5000 puta veću masu od plave, zbog čega je njeno kretanje skoro neprimetno u odnosu na plavu što se vidi na slici. Tokom animacije kretanja, sve vreme je crnom linijom crtana cela putanja lake čestice pridružena promenljivoj

```
gpath = Show[ListPointPlot3D[rez[[All, 1, 1]]],
ListPointPlot3D[rez[[All, 2, 1]], PlotStyle -> Black],
LabelStyle -> Directive[Black, 16, Bold], PlotRange -> All].
```

Kretanje čestice sve vreme simulacije prati vektor brzine, radi bolje ilustracije skaliran faktorom 2 i obojen zelenom linijom

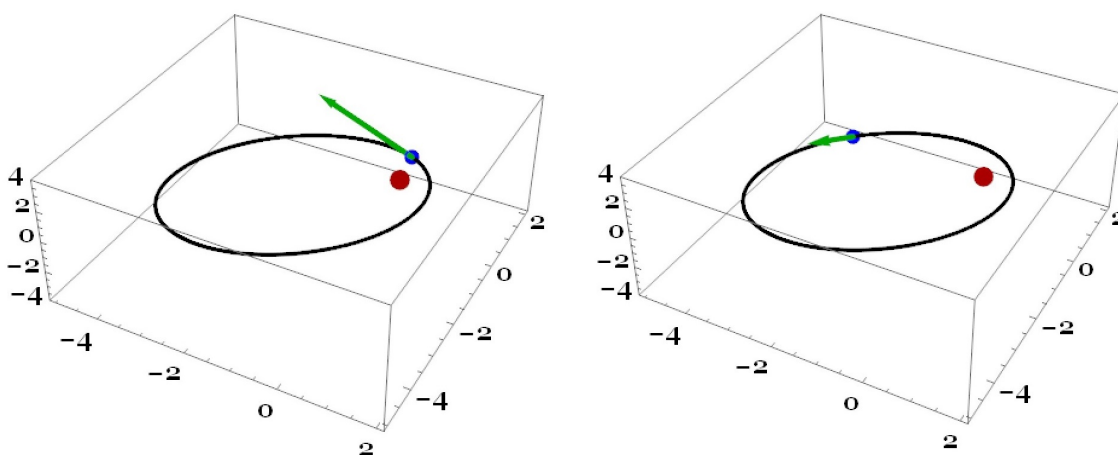
```
Graphics3D[{Darker[Green], Thickness[0.011],
Arrow[{rez[[i, 2, 1]], rez[[i, 2, 1]] + 2rez[[i, 2, 2]]}]}.}
```

Dužina vektora, odnosno intenziteta brzine se menja tokom ciklusa po elipsi. Tako je u tački najkraćeg rastojanja između čestica (analogna perihelu na putanji planete) vektor brzine najduži, a smanjuje se i najkraći je u tački na trajektoriji najudaljenijoj od crvene čestice (što odgovara afelu na eliptičnoj putanji planete).

```
(*crtanje putanje*)
gpath=Show[ListPointPlot3D[rez[[All,1,1]]],
ListPointPlot3D[rez[[All,2,1]],PlotStyle->Black],
LabelStyle->Directive[Black,16,Bold],PlotRange->All];

Manipulate[
Show[
(*putanja*)
gpath,
(*polozaj 1. crvene cestice*)
Graphics3D[{Darker[Red],PointSize[0.04],Point[rez[[i,1,1]]]}],
(*polozaj 2. plave cestice*)
Graphics3D[{Blue,PointSize[0.03],Point[rez[[i,2,1]]]}],
(*vektor brzine 2. plave cestice*)
Graphics3D[{Darker[Green],Thickness[0.011],
Arrow[rez[[i,2,1]],rez[[i,2,1]]+2rez[[i,2,2]]}],
PlotRange->{{-5,2},{-5,2},{-4,4}},{i,1,Niter,1}]

```



Slika 5.2: Kretanje plave čestice, čija je masa 5000 puta manje mase crvene, između kojih vlada Kulonova privlačna interakcija, duž eliptične (crne) trajektorije. Udaljavanjem plave čestice od crvene intenzitet vektora brzine se smanjuje i najmanji je u najudaljenijoj tački.

5.2 Numeričko rešavanje diferencijalnih jednačina drugog reda- Numerov

Osnov algoritma Numerov zasnovan je na tome da pitanje granične tretira kao problem početne vrednosti tokom rešavanja diferencijalne jednačine drugog reda. Ideja je nastala na pokušaju da se prilikom rešavanja jednačine povežu granična i početna vrednost. Metod spada u iterativne, pri čemu traženje vrednosti funkcije počinje od neke određene, dodeljene na jednoj od dve granice u predstojećim rešenim primerima. Početni uslovi se iterativno modifikuju na takav način da se na kraju ispune granični uslovi. Ovaj postupak pokazao se korisnim pri numeričkom traženju svojstvenih vrednosti, što se lako primenjuje na rešavanje jednodimenzionalne stacionarne Šredingerove jednačine. Diferencijalne jednačine drugog reda imaju važno mesto u fizici. Bilo da je reč o klasičnom sistemu opisanom Njutnovim zakonima ili kvantnom

čija se evolucija dobija rešavanjem Šredingerove jednačine, neophodno je rešiti diferencijalnu jednačinu drugog reda.

Ruski astronom Boris Numerov razvio je, kasnije po njemu nazvan algoritam, za rešavanje diferencijalnih jednačina oblika

$$\psi''(x) = -g(x)\psi(x) + s(x) \quad (5.9)$$

gde je $\psi(x)$ tražena funkcija, a $g(x)$ i $s(x)$ proizvoljne date funkcije promenljive x . Ovaj metod efikasan je u slučaju kada u jednačini ne figuriše prvi izvod $\psi'(x)$ funkcije $\psi(x)$. Za nalaženje rešenja diferencijalne jednačine neophodni su početni uslovi, to jeste poznate vrednosti funkcije $\psi(x_0)$ i njenog prvog izvoda $\psi'(x_0)$ u nekoj tački x_0 . Numerički metod Numerova pogodan je za traženje rešenja jednodimenzionalne Šredingerove jednačine potencijala proizvoljnog oblika. Metod spada među linearne i sastoji se od više koraka (*multistep*), što znači da se za nalaženje vrednosti funkcije u nekoj tački koriste linearne kombinacije vrednosti funkcije u prethodno određenim tačkama. Izvođenje algoritma svodi se na određivanje vrednosti tražene funkcije, njenim razvojem oko sledećeg i prethodnog malog koraka u okolini određene tačke u Tejlorovom redu do četvrtog stepena. Ovim razvojem dobije se sistem linearnih algebarskih jednačina iz kog se dalje izrazi vrednost funkcije sledećeg, a znajući ranije određene vrednosti prethodna dva koraka. Da bi se iteracija izvršavala, neophodno je znati vrednosti funkcije u određenim tačkama. Za granične vrednosti obično se uzimaju fizički uslovi, kao na primer da je vrednost funkcije veoma mala (skoro nula) na velikoj udaljenosti od određene oblasti prostora.

Realizacija algoritma kreće od podele prostora parametara funkcije na homogeni grid. Primera radi, ukoliko je problem jednodimenzionalan i pri tome iz određenih (fizičkih) uslova nalazimo da je deo prostora na intervalu $x \in [x_{min}, x_{max}]$ relevantan (recimo tražena funkcija konačna), interval se deli na N homogeno raspoređenih tačka

$$x_i = i \cdot h, i = 0, 1, 2, \dots, \quad \psi(x_i) = \psi_i.$$

Da bi se postigla što veća tačnost uzima se mala vrednost razlike h dve uzastopne tačke grida $h = (x_{max} - x_{min})/N$. Tejlorov razvoj funkcije ψ u tačkama $x_i + h$ i $x_i - h$ iz zadatog intervala mogu se izraziti kao

$$\begin{aligned} \psi(x_i + h) &= \psi(x_i) + h\psi'(x_i) + \frac{h^2}{2}\psi''(x_i) + \frac{h^3}{6}\psi'''(x_i) + \frac{h^4}{24}\psi''''(x_i) + \dots \\ \psi(x_i - h) &= \psi(x_i) - h\psi'(x_i) + \frac{h^2}{2}\psi''(x_i) - \frac{h^3}{6}\psi'''(x_i) + \frac{h^4}{24}\psi''''(x_i) + \dots \end{aligned}$$

Radi skraćenja pisanja, za vrednosti funkcije u $x_i + h$ uvodi se kondenzovana oznaka $\psi(x_i) \equiv \psi_i$, iz koje se nalazi $\psi(x_i \pm h) \equiv \psi_{i \pm 1}$. Oduzimanjem druge jednakosti od prve dobije se

$$\psi_{i+1} - \psi_{i-1} = 2h\psi'_i + \frac{h^3}{3}\psi'''_i + \dots,$$

odakle, ukoliko se zanemare članovi u razvoju od trećeg stepena (h^3), što je opravdano iz uslova da je h malo, dobija jedna od aproksimacija za nalaženje prvog izvoda poznata kao formula *dve-tačke centralne-razlike*

$$\psi'_i \approx \frac{\psi_{i+1} - \psi_{i-1}}{2h}. \quad (5.10)$$

Slično prethodnom, sabiranjem datih jednačina razvoja oko x_i dobije se

$$\psi_{i+1} + \psi_{i-1} = 2\psi_i + h^2\psi''_i + \frac{h^4}{12}\psi''''_i + \dots \quad (5.11)$$

Zanemarivši vrednosti članova u razvoju od četvrtog stepena (h^4) pa naviše male veličine h , iz prethodne jednakosti nalazi se izraz za procenu drugog izvoda funkcije

$$\psi''_i \approx \frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{h^2}. \quad (5.12)$$

Ovo je formula poznata pod nazivom *tri-tačke centralne-razlike*, kojom se numerički računa izvod u tački x_i , poznavajući vrednosti funkcije $\psi(x_i)$ i u njoj susednim tačkama definisanog grida $\psi(x_i \pm h)$. Dok se četvrti izvod od ψ može izraziti preko drugog izvoda, odnosno kao $\psi_i'''' = \frac{d^2\psi_i''}{dx^2}$. Kombinujući $\psi_i'' = -g_i\psi_i$ i gore datu formulu numeričkog drugog izvoda, dobije se

$$\psi_i'''' = -(\psi_i'')'' = -\frac{g_{i+1}\psi_{i+1} - 2g_i\psi_i + g_{i-1}\psi_{i-1}}{h^2}.$$

Zamenom dobijenog izraza za četvrti izvod iz (5.11), potom daljim sređivanjem dobija se izraz prema kom se računa približna vrednost funkcije ψ_{i+1} u x_{i+1} iz već poznatih, ranije određenih vrednosti ψ_i i ψ_{i-1}

$$\psi_{i+1} \approx \frac{2\psi_i(1 - \frac{5}{12}h^2g_i) - \psi_{i-1}(1 + \frac{h^2}{12}g_{i-1})}{1 + \frac{h^2}{12}g_{i+1}}. \quad (5.13)$$

5.2.1 Primena numeričkog metoda Numerov u kvantnoj mehanici

Dinamika jednočestičnog kvantnog sistema, njegova evolucija tokom vremena, nalazi se rešavanjem opšteg oblika Šredingerove jednačine

$$-i\hbar \frac{\partial |\psi(\vec{r}, t)\rangle}{\partial t} = \hat{H} |\psi(\vec{r}, t)\rangle = \left(-\frac{\hbar^2}{2m} \Delta + V(\vec{r})\right) |\psi(\vec{r}, t)\rangle. \quad (5.14)$$

Energijski spektar stacionarnog jednočestičnog i jednodimenzionalnog kvantnog sistema kao i talasne funkcije koje ga opisuju, dobiju se rešavanjem vremenski nezavisne Šredingerove jednačine,

$$\hat{H} |\psi(x)\rangle = E |\psi(x)\rangle, \quad \hat{H} = \frac{\hbar^2}{2m} \Delta + V(\vec{r}) \quad (5.15)$$

odnosno rešavanjem svojstvenog problema Hamiltonovog operatora. Za sistem zadat u koordinatnoj reprezentaciji, ovaj problem se svodi na rešavanje diferencijalne jednačine drugog reda (5.9) oblika

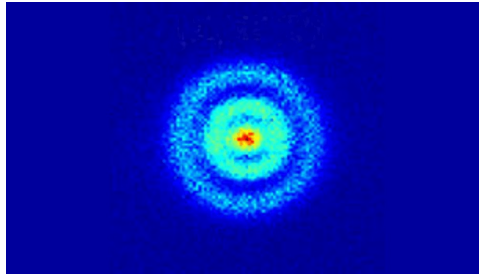
$$\left(-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x)\right) \psi(x) = E\psi(x) \quad (5.16)$$

U nastavku će biti demonstrirano rešavanje Šredingerove jednačine za nekoliko osnovnih jednodimenzionalnih i jednočestičnih modela kvantne mehanike, linearnog harmonijskog oscilatora i atoma vodonika. Ovi modeli se proučavaju na osnovnom kursu kvantne mehanike tokom kojih se nalaze njihova analitička rešenja, zbog čega su pogodna za proveru razvijenog numeričkog algoritma.

5.2.95 Šredingerova jednačina za centralni potencijal.

Primenom numeričkog metoda Numerova biće izračunati radijalni delovi talasnih funkcija za $1s, 2s, 2p, 3s, 3p$ i $3d$ orbitale atoma vodonika (ili vodoniku sličnog jona), pri čemu se njegov energijski spektar smatra poznatim. Dobijeni rezultati biće upoređeni sa egzaktnim analitičkim rešenjima. U gore navedenim stanjima, koristeći numeričke rezultate orbitala, biće izračunate očekivane vrednosti od r i $1/r$.

Vodoniku slični joni rednog broja Z su pozitivno naelektrisani sa $Z > 1$ protona u jezgru i 1 elektronom u omotaču. Usamljeni elektron vodoniku sličnog jona oseća sferno (centralno) simetričan Kulonov potencijal $V(r) = -\frac{Ze^2}{4\pi\epsilon_0 r}$. Zbog toga se za opis dinamike vodoniku sličnog



Slika 5.3: Fotografija atoma vodonika.

jona koristi sferni koordinatni sistem u kom su koordinate (r, θ, φ) . Laplasov diferencijalni operator izražen u sfernim koordinatama ima oblik

$$\Delta = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \varphi^2}. \quad (5.17)$$

Operator kvadrata angularnog momenta $\hat{L}^2 = \hat{L}_x^2 + \hat{L}_y^2 + \hat{L}_z^2$ u sfernim koordinatama reprezentovan je sa

$$\hat{L}^2 = -\hbar^2 \left[\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \varphi^2} \right], \quad (5.18)$$

a komponenta angularnog momenta \hat{L}_z sa $\hat{L}_z = -i\hbar \frac{\partial}{\partial \varphi}$.

Uvrštavanjem izraza za Laplasov operator (5.17) u (5.15), dobija se Hamiltonov operator za vodoniku slične atome

$$\hat{H} = -\frac{\hbar^2}{2m} \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{\hat{L}^2}{2mr^2} + \hat{V}(r). \quad (5.19)$$

Iz algebre operatora angularnog momenta poznato je da L^2 i L_z komutiraju ali i sa Hamiltonijanom, što znači da imaju zajednički skup svojstvenih talasnih funkcija preko kojih se klasifikuju stanja. Ukupna talasna funkcija $\psi_{nlm}(r, \theta, \varphi)$ kojom je opisan elektron u atomu vodonika nakon separacije varijabli može se napisati kao proizvod radijalnog $R_{nl}(r)$ i angularnog dela

$$\psi_{nlm}(r, \theta, \varphi) = R_{nl}(r) Y_l^m(\theta, \varphi), \quad (5.20)$$

gde je $Y_l^m(\theta, \varphi)$ sferni harmonik, istovremeno svojstvena funkcija za L_z i L^2

$$\hat{L}^2 Y_l^m = l(l+1)\hbar^2 Y_l^m, \quad \hat{L}_z Y_l^m = m\hbar Y_l^m,$$

n glavni kvantni broj, a $l = 1, 2, \dots, n-1$ i $m = -l, \dots, l$ redom angularni i magnetni kvantni brojevi.

Rešavanje stacionarne Šredingerove jednačine (5.16), svodi se na rešavanje svojstvenog problema Hamiltonijana atoma vodonika (vodoniku sličnog jona). Rešenje diferencijalne jednačine je talasna funkcija (5.20) čiji je radijalni deo rešenje jednačine

$$-\frac{\hbar^2}{2m} \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial R_{nl}(r)}{\partial r} \right) + \left(\frac{l(l+1)\hbar^2}{2mr^2} + \hat{V}(r) \right) R_{nl}(r) = E_n R_{nl}(r). \quad (5.21)$$

Kvadrat modula talasne funkcije je gustina verovatnoće nalaženja čestice u datoj tački prostora. Prema postulatu kvantne mehanike talasna funkcija je normirana na 1, što je integral kvadrata

njenog modula po celom prostoru. Odatle, verovatnoća nalaženja elektrona u zapremini između sfera radijusa r i $r + dr$ iznosi

$$\int |\psi_{nlm}(r, \theta, \varphi)|^2 r^2 \sin \theta dr d\theta d\varphi = |R_{nl}|^2 r^2 dr = |\chi_{nl}|^2 dr, \quad (5.22)$$

pri čemu je gornji integral dobijen uz uslov normiranosti sfernog harmonika

$$\int |Y_l^m(\theta, \varphi)|^2 \sin \theta d\theta d\varphi = 1 \quad (5.23)$$

dok je

$$\chi(r) = rR(r)$$

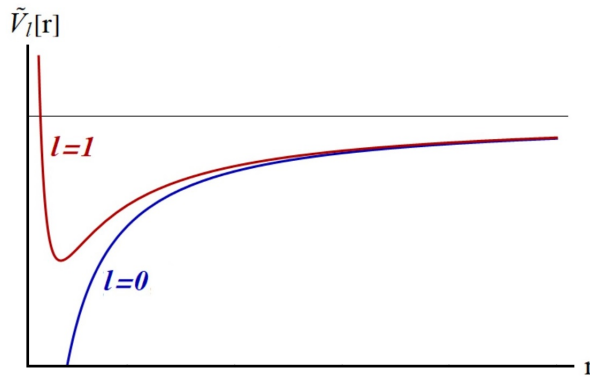
definicija pomoćne funkcije, takođe normirane na 1

$$\int_0^\infty |\chi_{nl}(r)|^2 dr = 1.$$

Sređivanjem (5.21) dobije se diferencijalna jednačina

$$-\frac{\hbar^2}{2m} \frac{d^2 \chi_{nl}(r)}{dr^2} + \left[\frac{\hbar^2 l(l+1)}{2mr^2} + V(r) - E \right] \chi_{nl}(r) = 0 \quad (5.24)$$

analogna jednočestičnoj Šredingerovoj jednačini. Rešenja ove jednačine za elektron energije E_n angularnog kvantnog broja l , je pomoćna funkcija χ_{nl} , koja će biti rešena numerički uz pomoć Numerov metoda. Radi lakšeg računanja uvodi se efektivni potencijal $\tilde{V}_l(r) = V(r) + \frac{\hbar^2 l(l+1)}{2mr^2}$, čiji je drugi sabirak zavisn od angularnog kvantnog broja l i naziva se centrifugalni potencijal.



Slika 5.4: Radijalna zavisnost efektivnog potencijala za različite angularne kvantne brojeve l .

Vrednost energije stacionarnog stanja vodoniku sličnog atoma rednog broja Z zavisi od glavnog kvantnog broja n i data je izrazom

$$E_n = -\frac{m_e e^4 Z^2}{(4\pi\epsilon_0)^2 2\hbar^2} \frac{1}{n^2}.$$

Centrifugalni član ukupnog potencijala postaje dominantan za male vrednosti radijusa r . Tako u oblasti prostora gde $r \rightarrow 0$ jednačina (5.24) se svodi na

$$\frac{d^2 \chi}{dr^2} \simeq \frac{l(l+1)}{r^2} \chi, \quad (5.25)$$

čija su rešenja $\chi(r) \sim r^{l+1}$ i $\chi \sim r^{-l}$. Naime, za proizvoljan dozvoljen angularni kvantni broj, ako bi se uzelo drugo rešenje jednačine (5.25) radijalna funkcija bila bi oblika $R_{nl} \sim 1/r^{l+1}$ i divergirala bi u $r = 0$, što nije prihvatljivo fizički. Zbog toga u blizini jezgra, radijalni deo talasne funkcije je oblika $R_{nl} \sim r^l$. Sa druge strane, na velikim rastojanjima od jezgra, ukoliko $r \rightarrow \infty$, jednačina (5.24) postaje

$$\frac{d^2\chi}{dr^2} + \frac{2m_e}{\hbar^2} E\chi = 0,$$

čije je rešenje za vezana stanja elektrona $E < 0$ oblika $\chi(r) \sim e^{\pm kr}$, gde je $k = \sqrt{-\frac{2m_e E}{\hbar^2}}$. Ponovo na osnovu fizički opravdanog zahteva da je gustina verovatnoće nalaženja čestice u beskonačnosti mala, jedino rešenje koje uz ove uslove preostaje je $\chi(r) \sim e^{-kr}$. Analitička funkcija koja zadovoljava diferencijalnu jednačinu (5.24) i pri tome ispunjava oba pomenuta uslova je oblika

$$\chi(r) = r^{l+1} e^{-kr} \sum_{n=0}^{\infty} A_n r^n. \quad (5.26)$$

Iz praktičnih razloga račun se sprovodi u atomskim jedinicama, gde je dužina izražena u odnosu na Borov radijus a_0

$$a_0 = \frac{\hbar^2 (4\pi\epsilon_0)^2}{m_e e^2} = 0.529177 \text{ \AA},$$

dok je energija data u jedinicama Ry (*Rydberg*), gde je $1\text{Ry} = \frac{m_e e^4}{2\hbar^2 (4\pi\epsilon_0)^4} = 13.6058 \text{ eV}$, ($m_e = 9.11 \times 10^{-31} \text{ kg}$, $e = 1.60217 \times 10^{-19} \text{ C}$.) U atomskom sistemu jedinica uzima se $\hbar = 1$, $m_e = 1/2$, $q_e^2 = e^2 / (4\pi\epsilon_0) = 2$. Program u kom se rešava radijalna Šredingerova jednačina radi jednostavnosti pisan je uvažavajući atomske jedinice.

Za zadate vrednosti energija elektrona izražene u jedinicama Ry u jonu sličnom vodoniku i određenih kvantnih brojeva n i l , rešava se radijalna jednačina (5.24), odakle se nalazi rešenje radijalnog dela talasne funkcije

$$\chi_{nl}(\zeta) = \sqrt{\frac{(n-l-1)! Z}{n^2 (n+l)! a_0^3}} \zeta^{l+1} e^{-\zeta/2} L_{n-l-1}^{2l+1}(\zeta), \quad (5.27)$$

gde je $\zeta = \frac{2Z}{n} \frac{r}{a_0} = 2\sqrt{-\frac{2m_e E_n}{\hbar^2}} r$ a $L_{n+l}^{2l+1}(x)$ Lagerov polinom. Za radijalni deo talasne funkcije važe relacije ortogonalnosti

$$\int_0^{\infty} \chi_{nl}(r) \chi_{n'l}(r) dr = \delta_{nn'}. \quad (5.28)$$

Klasičan pristup numeričkom rešavanju jednačine (5.24) nailazi na problem zbog singulariteta u tački $r = 0$. Jedan od načina da se on prevaziđe je da se umesto homogene pribegne prodeli radijusa promenljivog koraka. Ovaj problem je rešen smenom, tako umesto r se uvodi promenljiva x

$$x = x(r),$$

takva da je

$$\Delta x = x'(r) \Delta r.$$

Izborom

$$x(r) = \log \frac{Zr}{a_0},$$

nalazi se

$$\Delta x = \frac{\Delta r}{r},$$

čime je dobijen logaritamski grid. Da bi Numerov algoritam bio primenljiv prilikom rešavanja diferencijalne jednačine (5.24) uvedena je smena

$$y(x) = \frac{1}{\sqrt{r}}\chi(r(x)).$$

Osim predloženog, rešenje se može potražiti integracijom unazad, odnosno počevši od graničnog uslova $\chi(\infty) = 0$. Naravno, nije moguće uzeti u obzir $r = \infty$, zbog čega se preptostavlja da je za dovoljno veliko r_c ispunjeno $\chi(r_c) = 0$ (r_c cutoff radijus iza kog se smatra da je talasna funkcija jednaka nuli). Uvodeći smene $y = \frac{r}{a_0}$, $a_0 = \frac{4\pi\epsilon_0\hbar^2}{me^2} \approx 0.529\text{Å}$, $\epsilon = E/E_0$, $E_0 = \frac{\hbar^2}{2ma_0^2} = 13.6\text{eV} \approx 13.6\text{eV}$, u skladu sa datim Rydberg-ovim jedinicama, jednačina (5.24) čija su rešenja $\chi_{nl}(r)$ postaje

$$\chi''(y) - \left(\frac{l(l+1)}{y^2} - \frac{2Z}{y} + \frac{Z^2}{n^2} \right) \chi(y) = 0 \quad (5.29)$$

pri čemu se ona rešava za određene energije stacionarnih stanja $E_n = -\frac{Z^2}{n^2}$, $n = 1, 2, 3, \dots$. Od r_c pa unazad $x_i = r_c - i * h$, $h = r_c/nnum$ primenjuje se jednačina (5.13)

$$g(y, Z, n, l) = - \left(\frac{l(l+1)}{y^2} - \frac{2Z}{y} + \frac{Z^2}{n^2} \right), \quad (5.30)$$

gde je $nnum$ broj tačaka homogenog grida kojim je podeljen interval $(0, r_c]$. Zamenom $f_{1,i} = 1 + \frac{h^2}{12}g_i$, $f_{2,i} = 1 - \frac{5h^2}{12}g_i$, $g_i = g(x_i)$ jednačina (5.13) postaje

$$\chi_{i+1} = \frac{2\chi_i f_{2,i} - \chi_{i-1} f_{1,i-1}}{f_{1,i+1}}. \quad (5.31)$$

Gore razrađen algoritam implementiran je sledećim programom, čija su rešenja grafički prikazana i upoređena sa analitičkim u skladu sa zadatkom.

Najpre je isprogramirana analitička funkcija $\chi(y, Z, n, l)$, koja će biti iskorišćena za proveru dobijenih numeričkih vrednosti. Potom su zadati parametri neophodni za definisanje uslova numeričkog algoritma r_c cutoff radijusa, $nnum$ broj tačaka u gridu, razlika između tačaka h i vrednost numerički tražene funkcije $\Delta = 10^{-4}$ u prvoj tački grida. Potom su zadati kvantni brojevi, glavni n i angularni l , kojima je određen radijalni deo talasne funkcije orbitala. Radi što efikasnijeg izvršavanja algoritma biće isprogramirane a potom implementirane funkcije $f_{1,i}$, $f_{2,i}$ i g_i koje ulaze u izraz (5.13) adaptiran za traženje radijalnih funkcija orbitala vodonikovog atoma i njemu sličnih jona dat u (5.31).

```
(*definicija pomocne radijalne funkcije, dobijene analiticki,
u zavisnosti kvantnih brojeva, atomskog broja i radijusa*)
\[Chi][y_, Z_, n_, l_] :=
  Sqrt[Factorial[(n - 1 - 1)]/(n^2 Factorial[n + 1])] ((2*y/n)^(1 +
  1)) Exp[-y/n] LaguerreL[n - 1 - 1, 2 1 + 1, 2 y/n];
rc = 115;(*cutoff radijus u jedinicama a0/r*)
Z = 1;(*atomski broj vodoniku slicnog atoma*)
nnum = 100000;(*broj delova u gridu, homogenoj podeli intervala*)
h = 1. rc/nnum;(*korak podele intervala izmedju y=0 i y=r_c *)
(*vrednost trazene funkcije u prvom koraku, mala vrednost*)
\[Delta] = 10^(-4);
n = 1;
l = 0;
g[y_, Z_, n_, l_] := -(1 (1 + 1)/y^2 - 2 Z/y + Z^2/n^2);
f1[y_, Z_, n_, l_] := 1 + g[y, Z, n, l]*h^2/12;
f2[y_, Z_, n_, l_] := 1 - g[y, Z, n, l]*5 h^2/12;
```

Formirane su promenljive $y[i]$, koje odgovaraju tačkama grida na zadatoj podeli radijusa. Potom je definisana vrednosti numerički tražene funkcije u nultoj $y[0] = 0$ i prvoj tački $y[1] = \Delta$. Dalje, petljom prolazi se kroz sve ostale tačke do poslednje, računajući vrednosti tražene funkcije u skladu sa (5.31). Prema relaciji (5.28) dobijene vrednosti funkcije Ψ potrebno je normirati u odnosu na skalarni proizvod (5.22), pri čemu je za potrebe numeričke integracije upotrebljeno trapezoidno pravilo. Ova numerička metoda koristi se za nalaženje aproksimativne vrednosti integrala

$$I = \int_a^b f(x)dx,$$

neprekidne funkcije $f : \mathbb{R} \rightarrow \mathbb{R}$. Za procenu vrednosti zadatog integrala, interval $[a, b]$ se podeli u što više jednakih segmenata (grid) $x_2 - x_1 = x_3 - x_2 = \dots = x_n - x_{n-1} = h$, gde je $x_0 = a$, $x_n = b$. Interval se podeli na n segmenata, svaki širine $h = (b - a)/n$. Vrednosti funkcije u tačkama grida date su sa $f(x_i) = y_i$ za $i = 1, 2, \dots, n$. Ukoliko postoji veliki broj segmenata, tada funkciju f između svake dve, sada veoma bliske tačke $[x_i, x_{i+1}]$, možemo aproksimirati linearnom funkcijom. Pod tim uslovom, vrednost funkcije f u izdvojenim tačkama nad segmentom gradi pravougaoni trapez. Površina izdvojenog trapeza na segmentu širine h , prema Njutn-Lajbnicovom pravilu jednaka je integralu funkcije

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx h \frac{(y_{i+1} + y_i)}{2}.$$

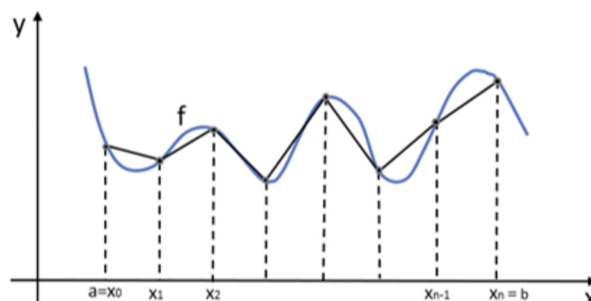
Površina ispod krive na celom intervalu $[a, b]$ približno je jednaka sumi površina svih trapeza iznad segmenata nad kojim je funkcija aproksimirana linearnom zavisnošću

$$I \approx \frac{h}{2}(y_0 + 2y_1 + \dots + 2y_{n-1} + y_n).$$

Gornji izraz se u vidu sume, pogodne za programiranje, može zadati sledećim izrazom

$$\int_a^b f(x)dx \approx h \sum_{i=0}^n f(a + ib) - \frac{h}{2}(f(a) + f(b)), \quad (5.32)$$

još poznatim kao numerička integracija trapezoidnim pravilom.



Slika 5.5: Ilustracija aproksimacije funkcije linearnom nad segmentima.

```
(*podela grida*)
Do[y[i] =i*h, {i, 0, nnum}];
(*=====*)
(*vrednost \[Chi] na velikoj udaljenosti od jezgra atoma u rc *)
Psi[0] = 0;
(*vrednost \[Chi] u prvom koraku, od rc*)
Psi[1] =\[Delta];
Do[
(*trazenje funkcionalnog oblika od \[Chi]*)
Psi[i+1] =
(2Psi[i]*f2[rc-y[i],Z,n,l]-Psi[i-1]*f1[rc-y[i-1],Z,n,l])/
f1[rc-y[i-1],Z,n,l],{i,1,nnum-1}];
(*trazenje konstante normiranja integracijom kvadrata modula \[Chi]*)
NormPsi=Sqrt[h*Sum[Abs[Psi[i]]^2, {i,1,nnum-1}]-
h*(Abs[Psi[0]]^2+Abs[Psi[nnum]]^2)/2];
(*formiranje parova polozaj-vrednost funkcije {x_i,\[Chi](x_i)},
od normiranih vrednosti \[Chi]_i*)
Chinorm=Table[{rc-y[i],Psi[i]/NormPsi},{i,0,nnum-1}];
```

Najpre je naredbom *ListPlot* nacrtan grafik numerički dobijenih rezultata *gnum*, a potom je sa *Plot* prikazan i analitički rezultat *ganlt* na datom intervalu. Poređenja radi, ovi grafici su prikazani na istom grafiku, pri tome funkcije mogu da budu različite do na fazni faktor, odnosno mogu se razlikovati do na znak ± 1 .

```
(*grafik funkcije \[Chi] dobijen numericki*)
gnum =ListPlot[Chinorm, PlotRange -> All,
PlotStyle->PointSize[0.017]];
(*grafik funkcije \[Chi] dobijen analiticki*)
ganlt=Plot[\[Chi][y, Z, n, l],{y, 0.01, rc}, PlotRange -> All,
PlotStyle -> {Thick, Red}];
Show[gnum, ganlt, PlotRange->{{0, 40}, All},
PlotLabel->Style["("<"n="<ToString[n] <>","<"l="<ToString[l] <>")"
, 24, Black, Bold],
AxesLabel->{Style["r[a0]", Large], Style["\[Chi][r/a0]", Large]},
AxesStyle -> Directive[Black, Thick, 24],
LabelStyle -> Directive[Black, 24, Bold]
```

Numerički dobijeni rezultati prikazani su na graficima, slika 5.6 plavim tačkama, zajedno sa funkcijom χ_{nl} bojenom crvenom, koja predstavlja analitičko rešenje (5.27) radijalnog dela Šredingerove jednačine (5.24).

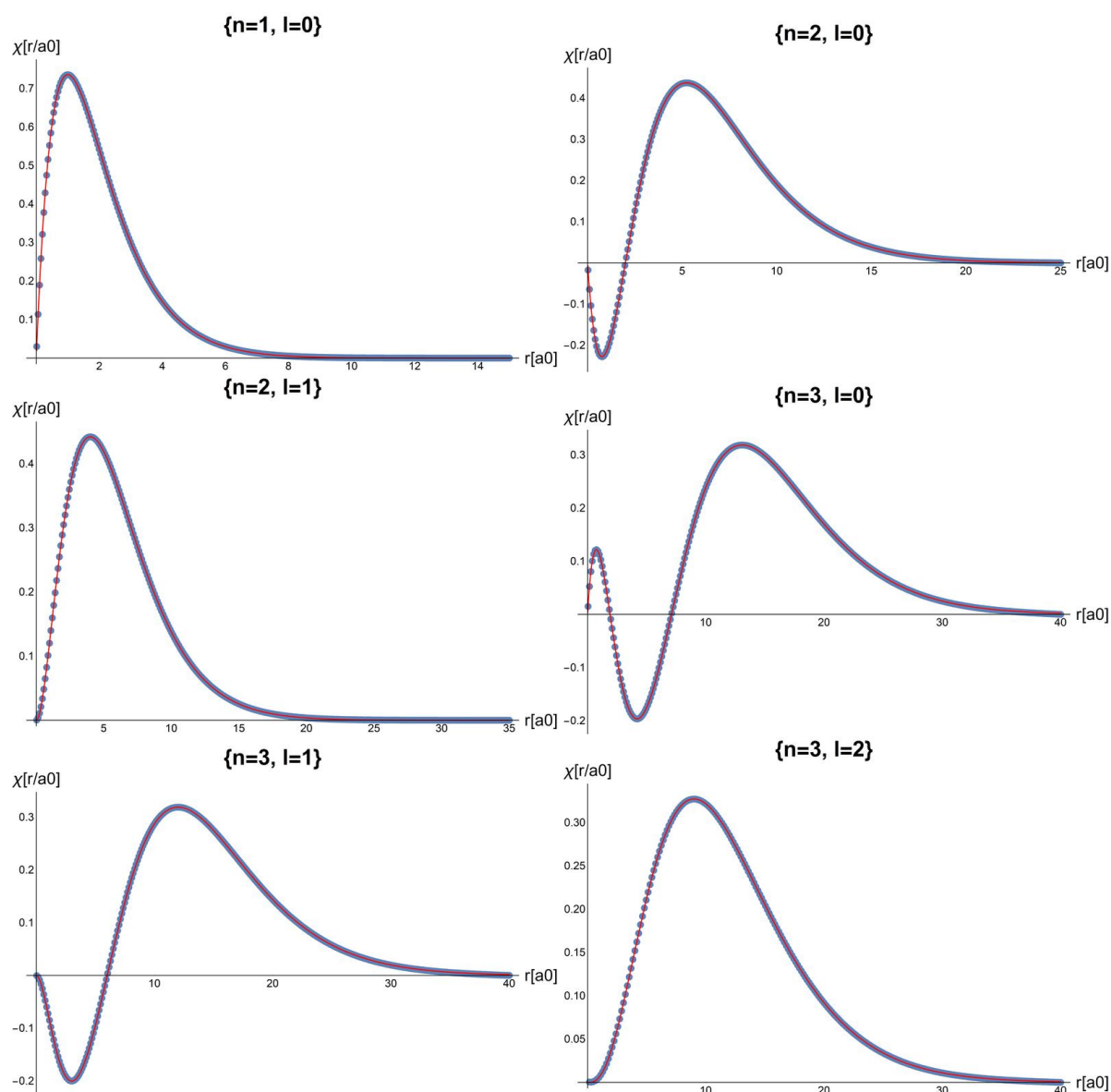
Očekivana vrednost $\langle r^s \rangle$, gde je $s \in \{-1, 1\}$ analitički se nalazi iz

$$\langle r^s \rangle = \int \psi_{nlm}^*(\vec{r}) r^s \psi_{nlm}(\vec{r}) d\vec{r} = \int_0^{2\pi} d\varphi \int_0^\pi |Y_l^m(\vartheta, \varphi)|^2 \sin \vartheta d\vartheta \int_0^\infty R_{nl}^*(r) r^s R_{nl}(r) r^2 dr.$$

Koristeći relaciju normiranosti (5.23) i definiciju pomoćne funkcije $\chi_{nl}(r)$ dobija se relacija

$$\langle r^s \rangle = \int_0^\infty |\chi_{nl}(r)|^2 r^s dr. \quad (5.33)$$

Kao rezultat izvršenog nmeričkog računa dobijena je lista parova $\{x_i, \chi_{nl}(x_i)\}$ vrednosti pomoćne funkcije na datom radijusu *Chinorm* u tačkama grida, u nastavku neophodnih za računanje srednjih vrednosti prema (5.33). Primenom trapezoidnog pravila numeričke integracije na listu pomoćne funkcije *Chinorm* nalazi se očekivana vrednost zadatog stepena radijusa. U tačkama homogenog grida, dobijenog podelom intervala od $r = 0$ do $r = r_c$, za $r_c = 115$ na $ncc = 100000$



Slika 5.6: Pomoćne funkcije χ_{nl} stanja kvantnih brojeva $n = 1, 2, 3$ vodonikovog atoma. Plave tačke dobijene su numerički, generisane programom napisanim prema izvedenom algoritmu. Funkcije analitičkih rešenja prikazane crvenom linijom se na datom intervalu poklapaju numeričkim.

delova, dobijeni su najpre radijalni delovi talasne funkcije. Potom, na osnovu ovih rezultata tražene su zadate očekivane vrednosti. Izlaz iz programa je lista četiri elementa $\{n, l, \langle r \rangle, \langle 1/r \rangle\}$, koja sadrži zadate kvantne brojeve n i l , potom očekivana vrednost radijusa $\langle r \rangle$ i njegove recipročne vrednosti $\langle 1/r \rangle$ redom. Računanje očekivanih vrednosti realizovano je sledećim programom.

```
{n, l, h
Sum[Chinorm[[i, 1]] Chinorm[[i, 2]]^2, {i, Length[Chinorm]}}-
h*(Chinorm[[1, 1]] Chinorm[[1, 2]]^2+
Chinorm[[nnum, 1]] Chinorm[[nnum, 2]]^2)/2,
h*Sum[1/Chinorm[[i, 1]] Chinorm[[i, 2]]^2,{i, Length[Chinorm]}}-
h*(1/Chinorm[[1, 1]] Chinorm[[1, 2]]^2+
1/Chinorm[[nnum, 1]] Chinorm[[nnum, 2]]^2)/2}
```

Pregled numerički dobijenih očekivanih vrednosti radijusa i njegovih recipročnih vrednosti u zadatim stanjima (kvantnim brojevima), izraženih u jedinicama a_0 , prikazani su u tabeli 5.1.

Tabela 5.1: Srednje vrednosti radijusa i njegove recipročne vrednosti za elektron u atomu vodonika u stanjima sa datim kvantnim brojevima.

n	l	$\langle r \rangle$	$\langle 1/r \rangle$
1	0	1.4999	1.0002
2	0	5.9999	0.25
2	1	4.9999	0.25
3	0	13.4999	0.1111
3	1	12.4999	0.1111
3	2	10.4999	0.1112

Opšti analitički izraz za vrednost očekivanog radijusa i njegove recipročne vrednosti u zavisnosti od kvantnih brojeva n i l , izraženih u jedinicama a_0 su redom

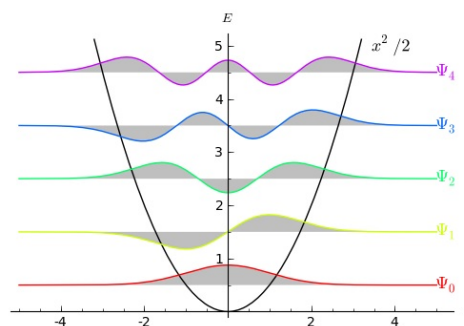
$$\langle r \rangle = \frac{1}{2}[3n^2 - l(l+1)] \text{ i } \langle \frac{1}{r} \rangle = \frac{1}{n^2}.$$

5.2.96 Kvantni linearni harmonijski oscilator (LHO).

Primenom algoritma Numerov, numerički će biti izračunate i ilustrovane talasne funkcije jednodimenzionalnog kvantnog harmonijskog oscilatora (LHO-a) uz uslov da je poznat njegov spektar

$$E_n = \hbar\omega(n + \frac{1}{2}).$$

Potom će biti proverena relacija ortogonalnosti dobijenih talasnih funkcija.



Slika 5.7: Potencijal LHO-a sa ilustracijom talasnih funkcija odgovarajućih energijskih nivoa.

Hamiltonijan jednodimenzionalnog kvantnog harmonijskog oscilatora je

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2,$$

gde su ω kružna frekvencija, m masa tela (čestice) a \hat{p} i \hat{x} redom operatori impulsa i koordinate. Prelaskom u koordinatnu reprezentaciju, gde je $\hat{p} = -i\hbar \frac{d}{dx}$, \hat{H} postaje

$$\hat{H} = \frac{-\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{1}{2}m\omega^2 x^2. \quad (5.34)$$

Da bi se uprostio račun, jednačina (5.34) je izražena u jedinicama u gde su $\hbar^2/m = 1$ i $m\omega^2 = 1$, nakon čega (5.34) postaje

$$\left(-\frac{1}{2} \frac{d^2}{dx^2} + \frac{1}{2}x^2\right)\psi(x) = E\psi(x) \quad (5.35)$$

čija će rešenja, odnosno svojstveni vektori (funkcije) od \hat{H} , biti numerički tražena za 6 energijski najnižih nivoa. Spektar od H izražen u prethodno definisanim jedinicama je oblika

$$E = n + 1/2, \quad n = 0, 1, 2, \dots$$

Shodno ideji algoritma Numerov, da je za numeričko rešavanje diferencijalne jednačine neophodno poznavati vrednost tražene funkcije u nekoj tački, zbog oblika potencijala selektovane su tačke x_{min} i njoj simetrična x_{max} , udaljenih od $x = 0$. Očekivane su male vrednosti gustina verovatnoće u izabranim tačkama, prema fizičkim uslovima da se verovatnoća nalaženje čestice smanjuje udaljavanjem od položaja minimuma potencijala. Zbog toga je uzeto da se funkcije numerički traže na intervalu $x_{min} = -6$ do $x_{max} = 6$. Izabrani interval na kom se traži funkcija je relativno blizu $x = 0$ kako bi primenjeni algoritam bio stabilan, odnosno davao rešenja malog odstupanja od egzaktno dobijenih. Iz jednačine (5.35) dobije se

$$\frac{d^2\psi_n(x)}{dx^2} + 2(E_n - x^2/2)\psi_n(x) = 0, \quad (5.36)$$

a zamenom $g_n(x) = 2(E_n - x^2/2)$ svodi se na jednačinu oblika (5.9)

$$\frac{d^2\psi_n(x)}{dx^2} + g_n(x)\psi_n(x) = 0,$$

čija se rešenja numerički nalaze implementacijom Numerov algoritma.

Za talasne funkcije $\psi_n(x)$ važi da su normirane na jedinicu i međusobno ortogonalne, odakle važi

$$\int_{-\infty}^{\infty} \psi_n^*(x)\psi_m(x)dx = \delta_{nm}. \quad (5.37)$$

Prvi deo programa sadrži granične vrednosti u kojima tražimo talasne funkcije energijski najnižih stanja x_{min} i x_{max} . Ovaj interval je podeljen na veliki broj $nnum$ malih segmenata dužine h . Nakon zadavanja numeričkih parametara, definisan je potencijal $v(x)$, potom funkcije $g(x, E_n)$, $f_1(x, E_n)$, $f_2(x, E_n)$ iz relacije (5.13). Vrednosti talasnih funkcija na početku zadatog intervala postavljene su na $\psi_1 = 0$ a $\psi_2 = \delta$, gde je δ neka mala vrednost (recimo 10^{-3}).

```

(*tacke gde se ocekuju gustine verovatnoce 0*)
xmin = -6;
xmax = 6;
(*broj tacaka podele grida*)
nnum = 10000;
(*vrednosti tacaka grida, nezavisno promenljive*)
Do[x[i] = 1. i*h, {i, 0, nnum}];
h = 1. (xmax - xmin)/nnum;
(*potencijal u definisanim jedinicama*)
v[x_] := 0.5 x^2;
(*definicija funkcije g[x]*)
g[x_,en_] :=2(en-v[x]);
(*definicije funkcija f1 i f2*)
f1[x_,en_] :=1+h^2*g[x,en]/12;
f2[x_,en_] :=1-5h^2*g[x,en]/12;
(*mala vrednost funkcije u x_1, \psi[x_1]=delta*)
\[Delta] = 10^-4;
spctrm = Table[i+1/2, {i, 0, 5}];
(*indeksi n talasnih funkcija*)
nn = Table[n, {n, 0, Length[spctrm]}];
Plot[v[x], {x, xmin, xmax}, GridLines -> {{}, spctrm}]

```

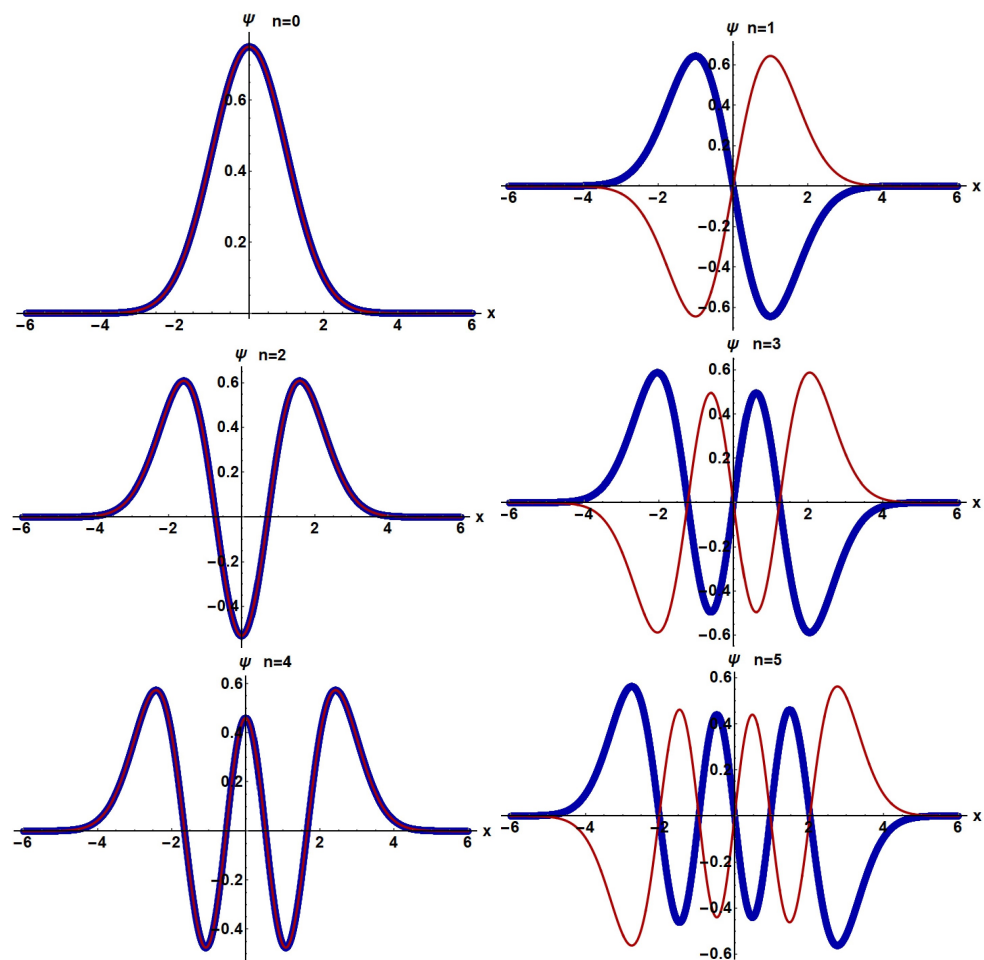
Lista *spctrm* sadži 6 najnižih energija iz spektra Hamiltonijana jednodimenzionalnog kvantnog harmonijskog oscilatora, sortiranih od najniže ka većim. Za svaku od njih Numerov algoritmom dva puta je računata odgovarajuća talasna funkcija. Jednom zadajući (*ispaljujući*) početne uslove sleva a drugi put zdesna, odnosno iz različitih krajeva intervala $[x_{min}, x_{max}]$. Najpre se zadaje nulta $\psi(x_{min/max}) = 0$ a potom prva vrednost $\psi(x_{min/max} \pm h) = \delta$. Ukoliko je zadata energija E , takva da joj se vrednost razlikuje od energije spektra od \hat{H} , dobijene funkcije sleva i zdesna nisu svojstvene i razlikuju se (ne samo po fazi). Ovo može poslužiti za razvijanje algoritma kojim bi se tražio spektar Hamiltonijana, čijim izvršavanjem bi pri svakoj promeni energije, za neku malu vrednost, bile računane talasne funkcije *ispaljene* sa različitih krajeva intervala, a potom bile upoređene. Preklapanje gustina verovatnoća na zadatom intervalu je veće što su energije bliže svojstvenim, a postaju jednake $|\psi_R(x_i)|^2 = |\psi_L(x_i)|^2, \forall x_i \in [x_{min}, x_{max}]$ za $E = E_n$, što se može koristiti kao test nalaženja svojstvenih vrednosti. Prema pretpostavci energije stanja su zadate te preostaje samo da se izračunaju talasne funkcije. Njihove vrednosti u prve dve tačke grida su zadate i prema fizičkim uslovima imaju malu vrednost na krajevima intervala, dok se preostale vrednosti $\Psi_R(L)$ u tačkama podele zadatog intervala nalaze prema (5.13). Za svaku zadatu svojstvenu energiju E_n uz pomoć *Do* petlje prebrišu se tačke zadatog grida i iterativno, iz prethodne dve, već određene ψ_{i-1} i ψ_i , nalazi vrednost funkcije u sledećoj tački ψ_{i+1} .

```

grphcs = {};
wvfncf = {};
Do[
  en = spctrm[[ie]];
  (*ispaljivanje talasne funkcije sa leve strane*)
  (*pridruzivanje prve dve vrednosti*)
  PsiL[0]=0;
  PsiL[1]=\[Delta];
  Do[PsiL[i +1] = (2 PsiL[i]*f2[xmin + x[i], en] -
    PsiL[i -1] f1[xmin+x[i-1], en])/
    f1[xmin+x[i+1],en],{i,1,nnum-1}];
  normL=Sqrt[
    h*Total[Table[Abs[PsiL[i]]^2, {i, 0, nnum}]] -
    h*(Abs[PsiL[0]]^2 + Abs[PsiL[nnum]]^2)/2];
  PsiFL=Table[{xmin+x[i],PsiL[i]/normL}, {i,0,nnum}];
  (*ispaljivanje talasne funkcije sa desne strane*)
  (*pridruzivanje prve dve vrednosti*)
  PsiR[0] = 0;
  PsiR[1]=\[Delta];
  (*racunanje ostalih vrednosti*)
  Do[PsiR[i+1]=(2PsiR[i]*f2[xmax-x[i],en] -
    PsiR[i-1] f1[xmax-x[i-1],en])/
    f1[xmax-x[i+1], en], {i,1,nnum-1}];
  normR=Sqrt[
    h*Total[Table[Abs[PsiR[i]]^2, {i, 0, nnum}]] -
    h (Abs[PsiR[0]]^2 + Abs[PsiR[nnum]]^2)/2];
  PsiFR=Table[{xmax-x[i],PsiR[i]/normR}, {i,0,nnum}];
  AppendTo[wvfncf,{PsiFL,PsiFR}];
  (*crtanje grafika*)
  gr=Show[
    ListPlot[PsiFL, PlotStyle -> {PointSize[0.014], Darker[Blue]}],
    ListLinePlot[PsiFR, PlotStyle -> {Thickness[0.005], Darker[Red]}],
    PlotRange -> All,
    PlotLabel -> Style["n=" <> ToString[nn[[ie]]], 24, Black, Bold],
    AxesLabel -> {Style["x", 24], Style["\[Psi]", 24]},
    LabelStyle -> Directive[Black, 24, Bold],
    AxesStyle -> Directive[Black, Thick, 24]];
  (*deponovanje grafika*)
  AppendTo[grphcs, gr], {ie, Length[spctrm]}]

```

Lista *wvfncf* sadrži vrednosti talasnih funkcije stanja u vidu $\{\{x_1, \psi_n^L(R)(x_1)\}, \{x_2, \psi_n^L(R)(x_2)\}, \dots\}$, dobijenih iterativno sleva i zdesna redom, obojenih plavom i tankom crvenom linijom na slici 5.8 za prvih 6 stanja najnižih energija. Iste funkcije, dobijene sa različitih strana intervala, mogu da se razlikuju do na predznak, odnosno u skladu sa postulatom kvantne mehanike do na fazni faktor. Prema relaciji (5.37) funkcije čine ortonormiran skup vektora u skladu sa skalarnim proizvodom $\int_R \psi_n^*(x) \psi_m(x) dx$ što je u nastavku provereno. Provere radi, dobijene funkcije skalarno su pomnožene svaka sa svakom, integraleći ih numerički primenom trapezoidnog pravila na opsegu gde su funkcije tražene. Naredbom *Table* formirana je matrica čiji su vandijagonalni elementi skalarni proizvod različitih $(\psi_n(x), \psi_m(x))$, $m \neq n$, a dijagonalni istih svojstvenih funkcija $(\psi_n(x), \psi_n(x))$. Rezultat je matrica čiji su dijagonalni elementi jednaki 1, a vandijagonalni po apsolutnoj vrednosti zanemarljivo mali u odnosu na dijagonalne i pri zadatim ulaznim parametrima programa kreću se između 10^{-12} i 10^{-10} .



Slika 5.8: Talasne funkcije $\psi_n(x)$ od 1D LHO za $n = 0, 1, 2, 3, 4, 5$, dobijene *ispaljivanjem* inicijalnih vrednosti sleva i zdesna bojene redom plavom i crvenom linijom.

```
MatrixForm[
Table[h*wvfncnt[[i,1,A11,2]].wvfncnt[[j,1,A11,2]]-
Total[wvfncnt[[i,1,{1,-1}]][[A11,2]]*
wvfncnt[[j,1,{1,-1}]][[A11,2]]*h/2,
{i,Length[wvfncnt]}, {j,Length[wvfncnt]}]]
```

5.3 Metod konačnih razlika

Cilj tehnike konačnih razlika je ovladavanje jednim od bazičnih algoritama rešavanja diferencijalnih jednačina drugog reda sa zadatim graničnim uslovima. Ilustracije radi, ovim metodom moguće je rešiti jednačine sa graničnim uslovima tipa

$$a(x)y''(x) + b(x)y'(x) + c(x)y(x) = f(x), \quad x \in [x_{min}, x_{max}] \quad (5.38)$$

$$\alpha_0 y(x_{min}) + \alpha_1 y'(x_{min}) = \lambda_1, \quad |\alpha_0| + |\alpha_1| \neq 0 \quad (5.39)$$

$$\beta_0 y(x_{max}) + \beta_1 y'(x_{max}) = \lambda_2, \quad |\beta_0| + |\beta_1| \neq 0. \quad (5.40)$$

Granični uslovi diferencijalnih jednačina drugog reda oblika

$$y(x_{min}) = \alpha, \quad y(x_{max}) = \beta$$

nazivaju se Dirihleovi ili granični uslovi prve vrste. Ukoliko su granični uslovi tipa

$$y'(x_{min}) = \alpha, \quad y'(x_{max}) = \beta,$$

reč je o Nojmanovim ili graničnim uslovima drugog reda. Kontinualni interval $[x_{min}, x_{max}]$ na kom se traže rešenja zadate diferencijalne jednačine se diskretizuje

$$x_k = x_{min} + (k - 1)h,$$

gde je korak grida h određen brojem podele intervala n na sledeći način

$$h = \frac{x_{max} - x_{min}}{n - 1},$$

pri čemu važi da je $x_1 = x_{min}$ i $x_n = x_{max}$. U nastavku važi notacija $y_k \equiv y(x_k)$ za svako $k = 1, 2, \dots, n$, što važi i za sve ostale funkcije u jednačini (5.38). Za izvođenje opšteg rešenja biće upotrebljeni izrazi (5.10) za prvi

$$y'_k \equiv y'(x_k) \approx \frac{y_{k+1} - y_{k-1}}{2h}$$

i (5.12) za drugi

$$y''_k \equiv y''(x_k) \approx \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2}$$

numerički izvod gde je $k = 2, \dots, n - 1$. Zamenom ovih aproksimacija izvoda u (5.38) nastaje diferencna jednačina

$$a_k \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + b_k \frac{y_{k+1} - y_{k-1}}{2h} + c_k y_k = f_k, \quad (5.41)$$

koja nakon sređivanja, grupisanja članova po y_k postaje

$$\left(\frac{a_k}{h^2} - \frac{b_k}{2h}\right) y_{k-1} + \left(c_k - \frac{2a_k}{h^2}\right) y_k + \left(\frac{a_k}{h^2} + \frac{b_k}{2h}\right) y_{k+1} = f_k$$

diferencna jednačina koja važi za $k = 2, 3, \dots, n - 1$ tačaka grida od n tačaka na intervalu $[x_{min}, x_{max}]$. Primenom ovog numeričkog algoritma za rešavanje diferencijalne jednačine formiraju se algebarske, kojima se u istom obliku pridružuju diskretizovani granični uslovi. Nakon toga ceo problem se pojednostavljuje i svodi na rešavanje sistema linearnih algebarskih jednačina. Diskretizacija graničnih uslova sledi iz relacija

$$y'_1 \equiv y'(a) \approx \frac{y_2 - y_0}{2h},$$

gde je vrednost $y_1 = y(x_{min})$. Iz prethodnog, jednačina graničnih uslova postaje

$$\alpha_0 y_1 + \alpha_1 \frac{y_2 - y_0}{2h} = \lambda_1,$$

gde je $y_0 = y(x_0)$ nepoznata vrednost funkcije izvan zadatog intervala u $x_0 = x_{min} - h$. Sređivanjem prethodnog izraza dobija se

$$y_0 = y_2 - \frac{2h}{\alpha_1} (\lambda_1 - \alpha_0 y_1), \quad \alpha_1 \neq 0.$$

Jednačina (5.41) za $k = 1$ postaje

$$\left(\frac{a_1}{h^2} - \frac{b_1}{2h}\right)y_0 + \left(c_1 - \frac{2a_1}{h^2}\right)y_1 + \left(\frac{a_1}{h^2} + \frac{b_1}{2h}\right)y_2 = f_1,$$

iz koje se nakon zemene izraza dobijenog za y_0 nalazi

$$\left[c_1 - \frac{2a_1}{h^2} + \frac{\alpha_0}{\alpha_1} \left(\frac{2a_1}{h} - b_1\right)\right]y_1 + \frac{2a_1}{h^2}y_2 = f_1 - \frac{\lambda_1}{\alpha_1} \left(b_1 - \frac{2a_1}{h}\right). \quad (5.42)$$

U specifičnom slučaju za $\alpha_1 = 0$, dobija se

$$y_1 = \frac{\lambda_1}{\alpha_0}.$$

Ista strategija primenjuje se na desni granični uslov, diskretizacijom jednačine koja ga opisuje i nalaženjem vrednosti y_{n+1} u tački grida $x_{n+1} = x_{max} + h$, koja leži izvan intervala $[x_{min}, x_{max}]$ iz

$$\beta_0 y_n + \beta_1 \frac{y_{n+1} - y_{n-1}}{2h} = \lambda_2.$$

Ova jednačina rešiva je po y_{n+1} uz uzlov da je $\beta_1 \neq 0$, pri čemu se dobija

$$y_{n+1} = y_{n-1} + \frac{2h}{\beta_1}(\lambda_2 - \beta_0 y_n).$$

Zamenom izraza za y_{n+1} u diferencnoj jednačini (5.41) za $k = n$ dobije se

$$\frac{2a_n}{h^2} y_{n-1} + \left[c_n - \frac{2a_n}{h^2} - \frac{\beta_0}{\beta_1} \left(b_n + \frac{2a_n}{h}\right)\right] y_n = f_n - \frac{\lambda_2}{\beta_1} \left(b_n + \frac{2a_n}{h}\right),$$

a u specifičnom slučaju kada je $\beta_1 = 0$ vrednost y_n je fiksna i iz graničnih uslova postaje

$$y_n = \frac{\lambda_2}{\beta_0}.$$

Iz prethodno izloženog formira se nehomogen sistem linearnih jednačina koji se može napisati u vidu

$$Ay = F, \quad (5.43)$$

gde je y vektor $y = (y_1, y_2, \dots, y_n)^T$, a vektor F je kolona

$$F = \begin{pmatrix} f_1 - \frac{\lambda_1}{\alpha_1} \left(b_1 - \frac{2a_1}{h}\right) \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \\ f_n - \frac{\lambda_2}{\beta_1} \left(b_n + \frac{2a_n}{h}\right) \end{pmatrix}, \quad (5.44)$$

dok je A matrica oblika

$$A = \begin{pmatrix} B_1 & C_1 & 0 & \dots & 0 \\ A_2 & B_2 & C_2 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & A_n & B_n \end{pmatrix}. \quad (5.45)$$

Elementi matrice (5.45) definisani su na sledeći način

$$A_k = \begin{cases} \left(\frac{a_k}{h^2} - \frac{b_k}{2h} \right), & k = 2, \dots, n-1 \\ \frac{2a_n}{h^2}, & k = n \end{cases} \quad (5.46)$$

$$B_k = \begin{cases} \left[c_1 - \frac{2a_1}{h^2} + \frac{\alpha_0}{\alpha_1} \left(\frac{2a_1}{h} - b_1 \right) \right], & k = 1 \\ \left(c_k - \frac{2a_k}{h^2} \right), & k = 2, \dots, n-1 \\ \left[c_n - \frac{2a_n}{h^2} - \frac{\beta_0}{\beta_1} \left(b_n + \frac{2a_n}{h} \right) \right], & k = n \end{cases} \quad (5.47)$$

i elementi

$$C_k = \begin{cases} \frac{2a_1}{h^2}, & k = 1 \\ \left(\frac{a_k}{h^2} + \frac{b_k}{2h} \right), & k = 2, \dots, n-1. \end{cases} \quad (5.48)$$

Nakon što se definišu elementi vektora F i matrice A sledi rešavanje sistema jednačina (5.43). Kada važi da je $\alpha_1 = \beta_1 = 0$ vrednosti funkcije u tačkama x_1 i x_n su fiksirani i iznose y_1 i y_n , a prikazani sistem linearnih jednačina se redukuje na ukupno $n - 2$ jednačine.

Za periodični granični uslov $y(x_{min}) = y(x_{max}) \neq 0$, koji diskretizacijom postaje

$$y_1 = y_n,$$

odakle se nalaze

$$a_1 = a_n, \quad b_1 = b_n, \quad c_1 = c_n.$$

Primenjujući aproksimacije na kojima se zasniva metod konačnih razlika na diferencnu jednačinu (5.41) za $k = 2, \dots, n-1$ dobija se sistem jednačina. Za $k = 2$ jednačina (5.41) postaje

$$\left(\frac{a_2}{h^2} - \frac{b_2}{2h} \right) y_1 + \left(c_2 - \frac{2a_2}{h^2} \right) y_2 + \left(\frac{a_2}{h^2} + \frac{b_2}{2h} \right) y_3 = f_2,$$

dok za $k = n-1$

$$\left(\frac{a_{n-1}}{h^2} - \frac{b_{n-1}}{2h} \right) y_{n-2} + \left(c_{n-1} - \frac{2a_{n-1}}{h^2} \right) y_{n-1} + \left(\frac{a_{n-1}}{h^2} + \frac{b_{n-1}}{2h} \right) y_n = f_{n-1}.$$

Iz uslova da je $y_1 = y_n$ gornja jednačina postaje

$$\left(\frac{a_{n-1}}{h^2} - \frac{b_{n-1}}{2h} \right) y_{n-2} + \left(c_{n-1} - \frac{2a_{n-1}}{h^2} \right) y_{n-1} + \left(\frac{a_{n-1}}{h^2} + \frac{b_{n-1}}{2h} \right) y_1 = f_{n-1},$$

dok je opšta diferencna jednačina za $k = 1$

$$\left(\frac{a_1}{h^2} - \frac{b_1}{2h} \right) y_{n-1} + \left(c_1 - \frac{2a_1}{h^2} \right) y_1 + \left(\frac{a_1}{h^2} + \frac{b_1}{2h} \right) y_2 = f_1,$$

gde su $y_0 = y(x_1 - h) \equiv y(x_n - h) = y_{n-1}$. Iz prethodne analize, rešenje diferencijalne jednačine sa periodičnim graničnim uslovima, numeričkim metodom se dobije rešavanjem sistema od $n - 1$ linearnih algebarskih jednačina, matrično zapisanih u obliku

$$Ay = F,$$

gde je $y = (y_1, \dots, y_{n-1})^T$, vektor F

$$F = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \end{pmatrix},$$

dok je matrica formata $(n-1) \times (n-1)$ data kao

$$A = \begin{pmatrix} B_1 & C_1 & 0 & \dots & 0 & A_1 \\ A_2 & B_2 & C_2 & & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & & & A_{n-2} & B_{n-2} & C_{n-2} \\ C_{n-1} & 0 & \dots & 0 & A_{n-1} & B_{n-1} \end{pmatrix}.$$

Matrični elementi matrice A u sistemu sa periodičnim graničnim uslovima su

$$A_k = \left(\frac{a_k}{h^2} - \frac{b_k}{2h} \right), \quad B_k = \left(c_k - \frac{2a_k}{h^2} \right), \quad C_k = \left(\frac{a_k}{h^2} + \frac{b_k}{2h} \right), \quad (5.49)$$

za vrednosti $k = 1, \dots, n-1$.

Elementarni primer primene metoda konačnih razlika u fizici je rešavanje stacionarne jednačine provođenja toplote, gde se posmatra šipka dužine L , čiji su krajevi u kontaktu sa toplotnim rezervoarima na temperaturama T_0 i T_n , slika 5.9.



Slika 5.9: Šipka čiji su krajevi u kontaktu sa termostatima na konstantnim temperaturama T_0 i T_n

Homogena jednačina provođenja toplote je linarni deo diferencijalne jednačine oblika

$$\frac{\partial T(x,t)}{\partial t} = \kappa \Delta T(x,t) \quad (5.50)$$

gde je $T(x,t)$ prostorno i vremenski zavisna funkcija temperature. Uz pretpostavku da je stanje provodnosti stacionarno (ne zavisi od vremena) pri zadatim graničnim uslovima, problem se svodi na rešavanje

$$\begin{cases} \frac{d^2 T(x)}{dx^2} = 0, & x \in [0, L] \\ T(0) = T_0, \\ T(L) = T_n, \end{cases} \quad (5.51)$$

Rešenje jednačine (5.51) se prosto nalazi analitički, pri čemu se dobija rešenje

$$T(x) = T_0 + (T_n - T_0) \frac{x}{L}.$$

Predstoji primena metoda konačnih razlika na složeniji 2D problem, gde će se tražiti raspored polja temperature po ravnoj ploči uz zadate granične uslove.

5.3.97 Distribucija skalarnog polja pri zadatim graničnim uslovima.

Laplasova jednačina je parcijalna diferencijalna jednačina drugog reda čije se rešenje traži u mnogim naučnim oblastima. Rešenje ove jednačine na domenu, zahteva specifikaciju određenih uslova koje tražena funkcija mora da zadovoljava na granicama domena. Kada su vrednosti funkcije zadati na delovima granice dati su Dirihleovi uslovi, a kada je specificiran izvod funkcije na granici, definisan je Nojmanov uslov. Ideja primene metoda konačnih razlika na rešavanje parcijalnih diferencijalnih jednačina svodi se na diskretizaciju jednačine, zamenom parcijalnih izvoda njihovom aproksimacijom, koja se zapravo definiše konačnom razlikom.

U nastavku biće razrađen algoritam za računanje, a potom prikazano rešenje distribucije temperature po ravnoj pravougaonoj površini pri zadatim Dirihleovim graničnim uslovima u vidu temperature oboda površine.

Metod konačnih razlika, u opštem slučaju namenjen za rašavanje diferencijalnih jednačina drugog reda, u ovom primeru biće upotrebljen za numeričko rešavanje Laplasove jednačine, primenjive u različitim oblastima fizike kao na primer elektrostatičke, stacionarnog toka fluida ili toplote. Rešavanje ove parcijalne diferencijalne jednačine drugog reda na određenom domenu, uslovljeno je poznavanjem vrednosti tražene funkcije na njegovim granicama, što je poznato kao Dirihleov uslov. Prosto opisano, metod konačnih razlika se zasniva na diskretizaciji oblasti kontinualnih promenljivih u kojoj se funkcija traži, vršeći zamenu parcijalnih izvoda njihovom numeričkom aproksimacijom, zapravo izraženih preko konačnih razlika. U narednom primeru traži se distribucija temperature po ravnoj pravougaonoj površini čiji su obodi na zadatoj temperaturi. Dvodimenzionalna oblast, ravna površ podeljena je na manje sa prirastom Δx i Δy kao na slici 5.10. Iz uslova da je proces stacionaran važi jednačina (5.51), odnosno

$$\Delta T(x, y) = 0.$$

Laplasov operator zadat u Dekartovim koordinatama je

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2},$$

primenjen na dati problemu dobija se Laplasova jednačina

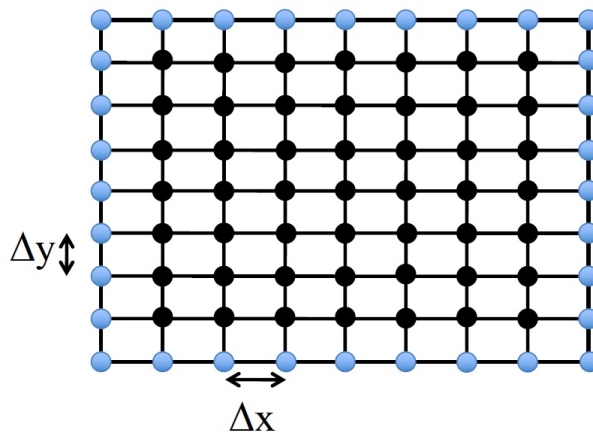
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0. \quad (5.52)$$

Svakoju tački (*nodu*) slika 5.10 dodeljuje se par indeksa i i j , koji ih prebrojavaju po vrsti i koloni, te redom odgovaraju njihovim koordinatama x i y . Pretpostavlja se da je temperatura svake tačke površine (x_i, y_j) jednaka srednjoj vrednosti temperatura njoj susednih oblasti prostora. Nakon diskretizacije Laplasova jednačina (5.52) u skladu sa nodalnim koordinatama (x_i, y_j) postaje

$$\left. \frac{\partial^2 T}{\partial x^2} \right|_{i,j} + \left. \frac{\partial^2 T}{\partial y^2} \right|_{i,j} = 0, \quad (5.53)$$

u kojoj se dati parcijalni izvodi aproksimiraju numeričkim drugim izvodnima u skladu sa relacijom (5.12), računatim centralno u 3 tačke

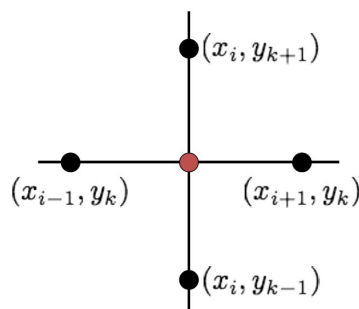
$$\left. \frac{\partial^2 T}{\partial x^2} \right|_{i,j} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2}, \quad \left. \frac{\partial^2 T}{\partial y^2} \right|_{i,j} \approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2}.$$



Slika 5.10: Diskretizacija pravougaone oblasti sa izdvojenim graničnim tačkama u kojima se definišu granične vrednosti temperaturnog polja.

Uvodeći date aproksimacije jednačina (5.53) postaje

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2} = 0. \quad (5.54)$$

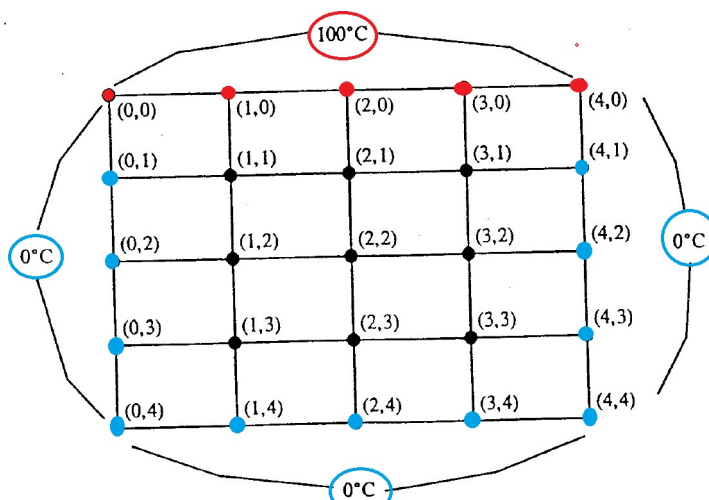


Slika 5.11: Laplasova jednačina (5.54) raspisana je za crvenu tačku.

Uz pretpostavku da se raspored nodalnih tačaka ponavlja sa istim periodom u oba pravca $\Delta x = \Delta y$, jednačina (5.54) postaje

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = 0. \quad (5.55)$$

Površina na slici 5.12 podeljena je na 25 čvorova, od kojih 16 graničnih i 9 preostalih unutrašnjih, čije se temperature traže. Koordinate ovih čvorova su (x_i, y_j) , gde su $x_i = i\Delta x$ i $y_j = j\Delta y$, uz uslov da je $\Delta x = \Delta y$. Svakom čvoru se dodeljuje temperatura $T(x_i, y_j) = T_{i,j}$. Sistem ima simetriju, što se vidi u odnosu na liniju koja sadrži tačke $T_{2,0}, T_{2,1}, T_{2,2}, T_{2,3}$ i $T_{2,4}$ po tome što je distribucija tačaka sa istom temperaturom odnosno graničnim uslovima levo i desno ista. Primenjujući simetriju uočava se da važi sledeće $T_{3,3} = T_{1,3}$, $T_{3,2} = T_{1,2}$ i $T_{3,1} = T_{1,1}$, što umesto 9 redukuje broj linearnih jednačina čije se rešenje traži na 6. Opšta relacija (5.55) primenjena na nodove sa nepoznatom temperaturom daje sledeći sistem jednačina



Slika 5.12: Nodalne tačke, obojene one u kojima su određeni granični uslovi.

$$\begin{aligned}
 4T_{1,1} - 0 - T_{1,2} - T_{2,1} - 100 &= 0 \\
 4T_{2,1} - T_{1,1} - T_{2,2} - T_{3,1} - 100 &= 0 \\
 4T_{1,2} - 0 - T_{1,1} - T_{1,3} - T_{2,2} &= 0 \\
 4T_{2,2} - T_{1,2} - T_{2,3} - T_{3,2} - T_{2,1} &= 0 \\
 4T_{1,3} - 0 - 0 - T_{2,3} - T_{1,2} &= 0 \\
 4T_{2,3} - T_{1,3} - 0 - T_{3,3} - T_{2,2} &= 0.
 \end{aligned}$$

Iz gore datih simetrijskih relacija dobija se sistem

$$\begin{aligned}
 4T_{1,1} - 0 - T_{1,2} - T_{2,1} - 100 &= 0 \\
 4T_{2,1} - 2T_{1,1} - T_{2,2} - 100 &= 0 \\
 4T_{1,2} - 0 - T_{1,1} - T_{1,3} - T_{2,2} &= 0 \\
 4T_{2,2} - 2T_{1,2} - T_{2,3} - T_{2,1} &= 0 \\
 4T_{1,3} - 0 - 0 - T_{2,3} - T_{1,2} &= 0 \\
 4T_{2,3} - 2T_{1,3} - 0 - T_{2,2} &= 0.
 \end{aligned}$$

Nakon sređivanja gornji sistem jednačina može se zapisati matrično u sledećoj formi

$$\begin{pmatrix}
 4 & -1 & -1 & 0 & 0 & 0 \\
 -2 & 4 & 0 & -1 & 0 & 0 \\
 -1 & 0 & 4 & -1 & -1 & 0 \\
 0 & -1 & -2 & 4 & 0 & -1 \\
 0 & 0 & -1 & 0 & 4 & -1 \\
 0 & 0 & 0 & -1 & -2 & 4
 \end{pmatrix}
 \begin{pmatrix}
 T_{1,1} \\
 T_{2,1} \\
 T_{1,2} \\
 T_{2,2} \\
 T_{1,3} \\
 T_{2,3}
 \end{pmatrix}
 =
 \begin{pmatrix}
 100 \\
 100 \\
 0 \\
 0 \\
 0 \\
 0
 \end{pmatrix}, \quad (5.56)$$

odnosno u obliku

$$SV = V_0. \quad (5.57)$$

Dobijena retka matrica S je nesingularna, te je moguće naći njoj inverznu S^{-1} . Granične vrednosti ulaze kao komponente vektora V_0 . Rešenje postavljenog sistema linearnih jednačina su

komponente vektora V , koje predstavljaju tražene temperature u zadatim nodalnim tačkama

$$V = S^{-1}V_0.$$

Većina matričnih elemenata matrice S su nule, što je čini retkom ili *sparse* matricom. Ovo je veoma bitno za rešavanje, budući da broj tačaka (nodova) može biti veliki, a sa time i format matrice, što komplikuje algoritam traženja njoj inverzne i zahteva primenu unapređenih i poboljšanih numeričkih tehnika.

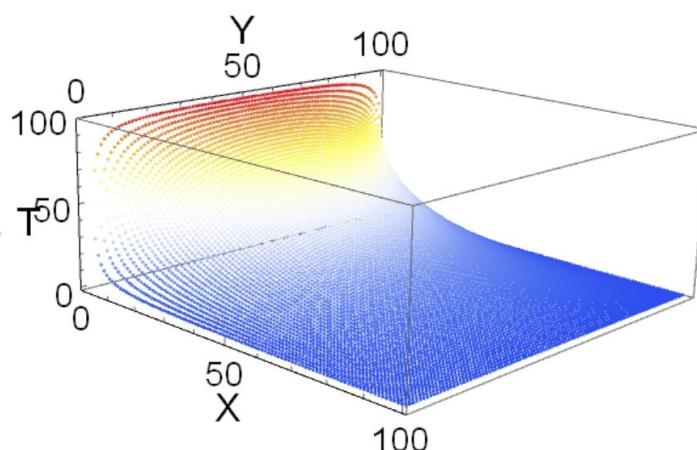
Traži se distribucija temperature ravne pravougaone površine ako su njene ivice dimenzija $0 \leq x \leq m$, $0 \leq y \leq n$, u kontaktu sa termostatima nepromenljivih temperatura $T(0, y) = 0$, $T(1, y) = 0$, $T(x, 0) = 0$ i $T(x, 1) = 100^\circ\text{C}$. Sledećim programskim kodom, u kom je implementiran metod konačnih razlika računata je distribucija temperature na kvadratnoj površini podeljenoj na 100×100 nodova, pri gore opisanim graničnim uslovima. Predloženo programsko rešenje u nastavku, kojim se numerički nalazi distribucija temperature po kvadratnoj površini čije su ivice u kontaktu sa termostatima poznatih temperatura, počinje definisanjem broja kolona i vrsta opšte matrice $tmat$, čiji matrični elementi $t[i, j]$ predstavljaju temperature čvorova na koje je površina podeljena. Prebrojavanjem matričnim elementima na ivici pravougaonika $t[i, 1]$, $t[i, n]$, $t[m, j]$ kroz petlju *Do* dodeljena je temperatura 0, a čvorovima na ivici $t[1, j]$ dodeljena je temperatura 100. Formirana je lista $edge = Union[Flatten[edge]]$ koja sadrži deponovana sva definisana pravila zamene, unijom svih skupova pravila zadatih temperatura na ivicama. Analogno se može definisati i rešavati problem distribucije skalarnog električnog potencijala.

```
(*definisanje granicnih uslova*)
(*broj kolona*)
m=100;
(*broj vrsta*)
n=100;
(*formiranje matrice sa opstim elementima t[i,j]*)
tmat=Array[t,{m,n}];
(*prazan skup u koji se sipaju granicne vrednosti*)
edge={};
Do[AppendTo[edge,{t[i,1]->0,t[i,n]->0}],{i,m}];
(*pridruzivanje temperatura nodovima T_{i,1}=0 i T_{i,n}=0*)
Do[AppendTo[edge,{t[1,j]->100,t[m,j]->0}],{j,n}];
(*pridruzivanje temperatura nodovima T_{1,j}=100 i T_{m,j}=0*)
edge=Union[Flatten[edge]];
(*=====*)
```

Nakon definisanja graničnih uslova sledi formiranje S matrice iz relacije (5.57). Najpre se definiše složena lista $Smat = Table[\{\{i, j\}, t[i, j]\}, \{i, 2, m - 1\}, \{j, 2, n - 1\}]$, koja se izravna do prvog nivoa, $Flatten[\dots, 1]$, da bi se dobio skup elemenata $\{\dots, \{i, j\}, t[i, j]\}, \dots$, $i = 2, \dots, m - 1$, $j = 2, \dots, n - 1$. Prvo mesto elementa u listi su koordinate $\{i, j\}$, a zatim njima odgovarajući matrični element $t[i, j]$ (temperatura). Elementi od vec su razdvojeni u dve liste, koordinata $vecmat = vec[[All, 1]]$ i njima odgovarajućih temperatura u istom poretku $vec = vec[[All, 2]]$. Matrici $Smat$ dodeljena je poravnata lista čiji su elementi suma (5.55) realizovani sa $Total[\{t[i, j], t[i - 1, j], t[i + 1, j], t[i, j - 1], t[i, j + 1]\}, \{4, -1, -1, -1, -1\}]$. Iz ove liste najpre je izdvojen vektor graničnih vrednosti v_0 , tako što je u listi $Smat$ izvršena zamena vrednosti prema pravilima deponovanim u listi $edge$, a potom svi ostali elementi $t[i, j]$ su u produžetku zamenjeni nulom $v_0 = -(Smat /. dege) /. Thread[vec \rightarrow Table[0, \{Length[vec]\}]]$. Čuvajući poredak jednačina kojim je nastao v_0 , u nastavku se formira matrica S , tako što se redom diferenciraju sume $Smat[[i]]$ po matričnim elementima $t[i, j]$ iz skupa vec na sledeći način $Mat = Table[D[Smat[[i]], \{vec\}], \{i, Length[Smat]\}]$. Nepoznate temperature čvorova dobijene su množenjem invertovane S matrice i kolone temperatura u graničnim čvorovima $Inverse[1.Mat].v_0$. Matrica $Mmat$ je pre invertovanja pomnožena realnim brojem 1. da bi se inverzija izvršila numeričkim algoritmima jezika Wolfram. Transponovanim parovima koordinata, dobijene su dve

liste, u jednoj i a u drugoj j koordinate čvorova, kojima je dodata dobijena lista *rez* temperatura u odgovarajućem poretku. Potom je ponovo izvršena transpozicija da bi se dobila struktura $\{\dots, \{i, j, t[i, j]\}, \dots\}$ koja odgovara $\{\dots, \{x_i, y_j, T_{ij}\}, \dots\}$, kako bi se mogla nacrtati distribucija skalarnog polja (temperature) po površini kvadrata.

```
(*formiranje S matrice, v i v0 vektora*)
vec=Flatten[Table[{{i,j},t[i,j]},{i,2,m-1},{j,2,n-1}],1];
(*konstrukcija vektora v *)
vecmap=vec[[All,1]];
vec=vec[[All,2]];
Smat=Flatten[
Table[
Total[{t[i,j],t[i-1,j],t[i+1,j],t[i,j-1],t[i,j+1]}{4,-1,-1,-1,-1}],
{i,2,m-1},{j,2,n-1}],1];
(*S- matrica*)
(*=====*)
(*zamena vrednosti*)
v0=- (Smat/. edge)/. Thread[vec->Table[0,{Length[vec]}]];
Mat=Table[D[Smat[[i]], {vec}], {i,Length[Smat]}];
rez=Inverse[1.Mat].v0;
rez=Transpose[Append[Transpose[vecmap], rez]];
g2=ListPointPlot3D[rez,ColorFunction->"Temperature"]
```



Slika 5.13: Distribucija temperature po ravnoj kvadratnoj površini podeljenoj na 100×100 nodova, čije su tri ivice na 0°C a četvrta na 100°C .

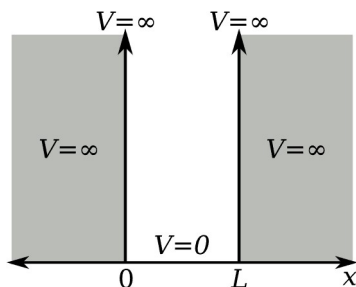
5.3.98 Čestica u beskonačno-dubokoj 1D potencijalnoj jami.

Jedan od standardnih, analitički rešivih zadataka iz kvantne mehanike je jednostavan, jednodimenzionalan model čestice u beskonačno dubokoj potencijalnoj jami širine L postavljenoj duž x -ose. Čestica se od nepropusne zidove jame

$$U(x) = \begin{cases} 0 & x \in [0, L] \\ \infty & x \notin (0, L) \end{cases} \quad (5.58)$$

reflektuje apsolutno elastično, ne postoji penetracija u prostor iza njih. Kao posledica navedenih uslova gustina verovatnoće nalaženja čestice izvan jame jednaka nuli.

Metodom konačnih razlika biće nađeno nekoliko najnižih energija spektra i njima odgovarajućih talasnih funkcija koje opisuju stanje čestice u jami sa beskonačno visokim nepropusnim zi-



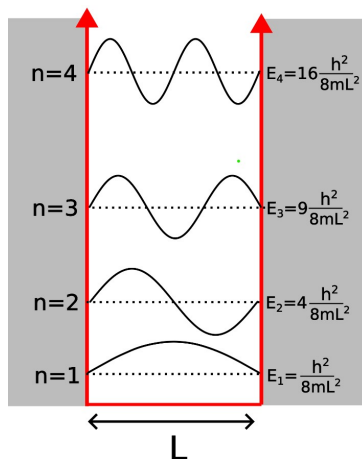
Slika 5.14: Beskonačno duboka potencijalna jama.

dovima. Rešavanjem vremenski nezavisne jednodimenzionalne Šredingerove jednačine

$$-\frac{\hbar^2}{2m} \frac{d^2\psi(x)}{dx^2} + U(x)\psi(x) = E\psi(x) \quad (5.59)$$

egzaktno se dobija diskretni energijski spektar $E_n = \frac{n^2\hbar^2\pi^2}{2mL^2}$ i odgovarajuće talasne funkcije $\psi_n(x) = \sqrt{\frac{2}{L}} \sin \frac{n\pi}{L}x$.

Pri tome, u oblast gde je $U(x) = 0$ na intervalu $x \in (0, L)$, rešenja Šredingerove jednačine ista su kao za slobodnu česticu.



Slika 5.15: Spektar i talasne funkcije beskonačno duboke potencijalne jame.

Numerički pristup rešavanja jednačine (5.59), metodom konačnih razlika, počinje transformacijom diferencijalne jednačine u matricnu. Da bi ovo bilo moguće, vrednost funkcije u različitim koordinatama potrebno je prikazati u vidu brojne kolone (vektora). Koordinata x čije su vrednosti kontinualne se diskretizuje u oblasti gde je očekivana lokalizacija čestice na intervalu $x \in (0, L)$. Interval se podeli na veliki broj delova. U svakoj tački ovog intervala $x_i = ih, h = L/n$ nađu se vrednosti kontinualne funkcije $\psi(x_i)$, koja se takođe na taj način diskretizuje. Broj koordinata nastale kolone jednak je broju delova na koje je homogeno podeljen interval u kom se očekuje nalaženje čestice. Broj na koji delimo kontinualni parametar prostora jednak je broju jednačina, koji bi trebalo da bude dovoljno veliki kako bi se postigla određena tačnost rešenja, ali s druge strane da je rešavanje programski izvodljivo. U cilju verifikacije, metod će biti primenjen na analitički rešiv model iz kvantne mehanike, nalaženje stanja čestice u beskonačno dubokoj potencijalnoj jami a dobijeni rešenja će se uporediti analitičkim.

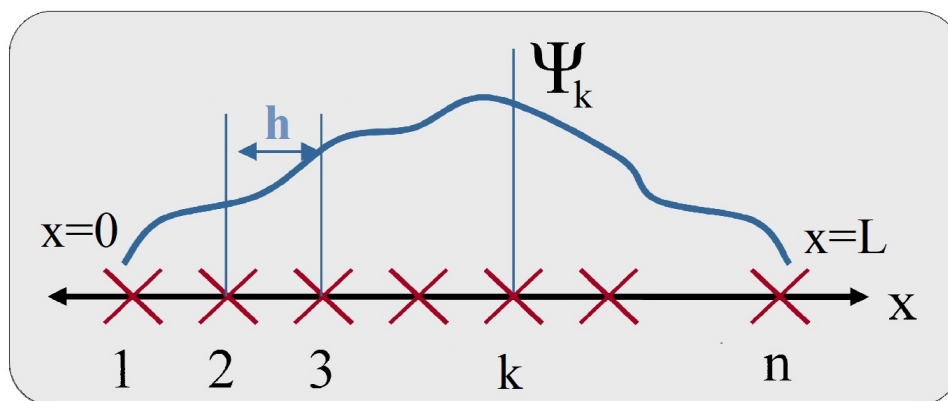
Talasna funkcija, kojom se opisuje stanje čestice datog sistema $\psi(x)$, može se napisati kao kolona, dužine jednake broju tačaka grida

$$\psi(x) \rightarrow \begin{pmatrix} \psi(x_1) \\ \psi(x_2) \\ \psi(x_3) \\ \vdots \\ \psi(x_n) \end{pmatrix} \quad (5.60)$$

čije su komponente njene vrednosti u tim tačkama. U nastavku se koriste oznake $\psi(x_i) \equiv \psi_i$. Transformacija Šredingerove jednačine diskretizacijom u matričnu odvija se prema sledećoj shemi

$$-\frac{\hbar^2}{2m} \frac{d^2\psi(x)}{dx^2} + U(x)\psi(x) = E\psi(x) \Rightarrow \begin{pmatrix} H \\ n \times n \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} = E \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}, \quad (5.61)$$

gde su ψ_i , $i = 1, 2, \dots$ vrednosti talasne funkcije u tačkama grida kao na slici 5.16.



Slika 5.16: Podela prostora u kom se nalazi čestica u beskonačno dubokoj potencijalnoj jami, sa ilustranom talasnom funkcijom iznad grida duž prostorne koordinate x .

Rešavanje matrične jednačine (5.61) svodi se na rešavanje svojstvenog problema matrice H . Na osnovu formata matrice hamiltonijana ($n \times n$), očekivan je n svojstvenih vrednosti i svojstvenih vektora. Hamiltonov operator $H = T + V$ sadrži operatore kinetičke i potencijalne energije. Operator kinetičke energije je laplasijan, a u ovom jednodimenzionalnom primeru to je drugi izvod po koordinati x . U tački grida x_i njegovo dejstvo na talasnu funkciju postaje

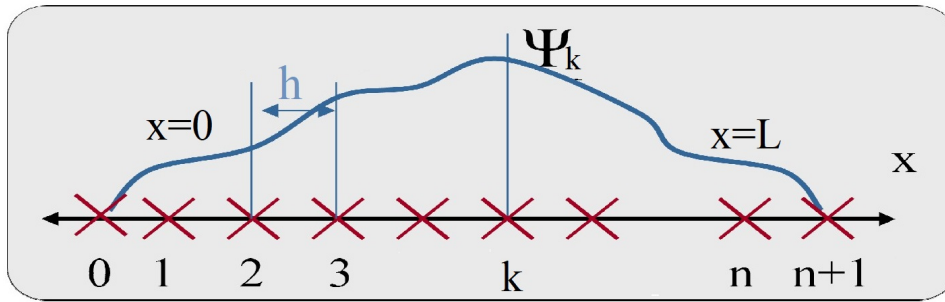
$$T\psi(x_i) = T\psi_i = -\frac{\hbar^2}{2m} \left[\frac{d^2\psi}{dx^2} \right]_i.$$

Drugi izvod od $\psi(x)$ u tački x_i numerički uz pomoć centralne formule tri tačke nalazi se

$$\left[\frac{d^2\psi}{dx^2} \right]_i = \frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{h^2}, \quad (5.62)$$

nakon čega dejstvo operatora kinetičke energije na talasnu funkciju u izabranoj tački postaje

$$T\psi_i = -t_0(\psi_{i+1} - 2\psi_i + \psi_{i-1}),$$



Slika 5.17: Ilustracija graničnih uslova, talasna funkcija na granici, odnosno na zidu beskonačne visine, jednaka je o.

gde je $t_0 = \frac{\hbar^2}{2m}$ poznata konstanta. Matrično predstavljeno dejstvo operatora kinetičke energije vidi se kao dejstvo tridijagonalne matrice na kolonu komponenti talasne funkcije

$$T\psi = \frac{t_0}{h^2} \begin{pmatrix} 2 & -1 & \dots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 \dots & 0 \\ 0 & 0 & \ddots & & & 0 \\ 0 & 0 & 0 & \ddots & 0 & -1 \\ 0 & 0 & \dots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \quad (5.63)$$

Dejstvo operatora potencijalne energije u izabranoj tački x_i na talasnu funkciju je

$$[U(x)\psi(x)]_i = U(x_i)\psi_i,$$

realizuje se kao dejstvo dijagonalne matrice na kolonu.

Šredingerova jednačina (5.59) nakon transformacije u matrični oblik postaje

$$\begin{pmatrix} 2t_0 + U(x_1) & -t_0 & \dots & 0 & 0 & 0 \\ -t_0 & 2t_0 + U(x_2) & -t_0 & 0 & \dots & 0 \\ 0 & -t_0 & 2t_0 + U(x_3) & -t_0 & 0 \dots & 0 \\ 0 & 0 & \ddots & & & 0 \\ 0 & 0 & 0 & \ddots & 0 & -t_0 \\ 0 & 0 & \dots & 0 & -t_0 & 2t_0 + U(x_n) \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} = E \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}. \quad (5.64)$$

Jednačine koje opisuju stanje na granicama intervala $(0, L)$ ilustrovano na slici 5.17 su

$$-t_0\psi_0 + (2t_0 + U_1)\psi_1 - t_0\psi_2 = E\psi_1 \quad (5.65)$$

$$-t_0\psi_{n-1} + (2t_0 + U_n)\psi_n - t_0\psi_{n+1} = E\psi_n. \quad (5.66)$$

Česticu nije moguće naći izvan jame gde je gustina verovatnoće njenog nalaženja nula, matematički je zapisano sledećim jednakostima

$$\psi_0 = 0, \quad \psi_{n+1} = 0. \quad (5.67)$$

Nalaženje spektra hamiltonijana zadatog sistema se numerički prema (5.64) svodi na traženje svojstvenih vrednosti, gde će biti iskorišćene efikasne procedure programskog jezika Wolfram .

Tokom računanja energijskog spektra i talasnih funkcija čestice u beskonačno dubokoj potencijalnoj metodi konačnih razlika funkcija potencijalne energije jame je uzeta prema (5.58), odakle su nakon diskretizacije prostora parametara vrednost potencijala u tačkama podele $U(x_i) = 0, i = 1, 2, \dots, n$.

Programsko rešenje počinje definicijom fizičkih i numeričkih parametara.

```
n=1000;(*broj tacaka homogene podele sirine jame*)
L=1.;(*sirina jame*)
h=L/n;(*rastojanje izmedju 2 tacke u gridu*)
```

Zatim je definisana matrica operatora hamiltonijana. Njen format zavisi od od broja tačaka podele prostora u kom se nalazi čestica. Promenljivoj *Ham* najpre je dodeljena jedinična matrica formata predviđenog za hamiltonijan, koja je pomnožena nulom. Potom je iz ove matrice rekonstruisan opšti oblik hamiltonijana prema izrazu (5.64). Nakon toga se množenjem hamiltonijana sa $1/h^2$ i zamenom odgovarajućih parametara $t_0 \rightarrow 1$, dobija matrica čiji su elementi konkretni realni brojevi, na koje je moguće primeniti implementirane programske pakete linearne algebre.

```
(*u pocetku hamiltonijanu se pridružuje matrica ciji su svi
elementi 0*)
Ham=0IdentityMatrix[n];
(*Konstrukcija hamiltonijana*)
Do[If[i===j,Ham[[i,j]]=2,
If[Abs[i-j]===1,Ham[[i,j]]=-1,0]],{i,n},{j,n}];
Ham=t0*Ham/(h^2);
Ham=Ham/. {t0->1.};
```

Nalaženje energijskih nivoa i odgovarajućih talasnih funkcija realizovano je rešavanjem svojstvenog problema sa *Eigensystem* koja je funkcija programskog jezika Wolfram. Radi lakšeg manipulisanja dobijenim podacima, skup *rez* sačinjen od liste svojstvenih energija i vektora je transponovan te je dobijena lista elemenata $\{\dots, \{e_i, v_i\}, \dots\}$, gde su e_i i v_i svojstvena vrednost i njoj odgovarajući svojstveni vektor.

```
(*resavanje svojstvenog problema matrice hamiltonijana Ham*)
rez=Eigensystem[Ham];(*{e1,...,en},{v1,...,vn}*)
(*grupisanje svojstvenih vrednosti i odgovarajucih vektora*)
rez = Transpose[rez];
(*{...,{ei,vi},...}*)
(*sortiranje formirane liste rez po rastucim vrednostima energije*)
(*e1<e2<e3,...*)
rez=Sort[rez];
```

Radi pojednostavljenja numeričkog rešavanja konstanti t_0 se dodeljuje vrednost 1. Poređenja radi, mogu se zameniti odgovarajuće veličine u analitičkom izrazu za spektar čestice u beskonačno dubokoj potencijalnoj jami

$$E_n = \frac{n^2 \pi^2 \hbar^2}{2mL^2}, \quad (5.68)$$

tako unosenjem t_0 u gore datu relaciju za energiju ona postaje

$$E_n = \frac{t_0 n^2 \pi^2}{L^2}. \quad (5.69)$$

Provere radi, preko relacije (5.69) izračunati su kvadrati kvantnih brojeva n^2

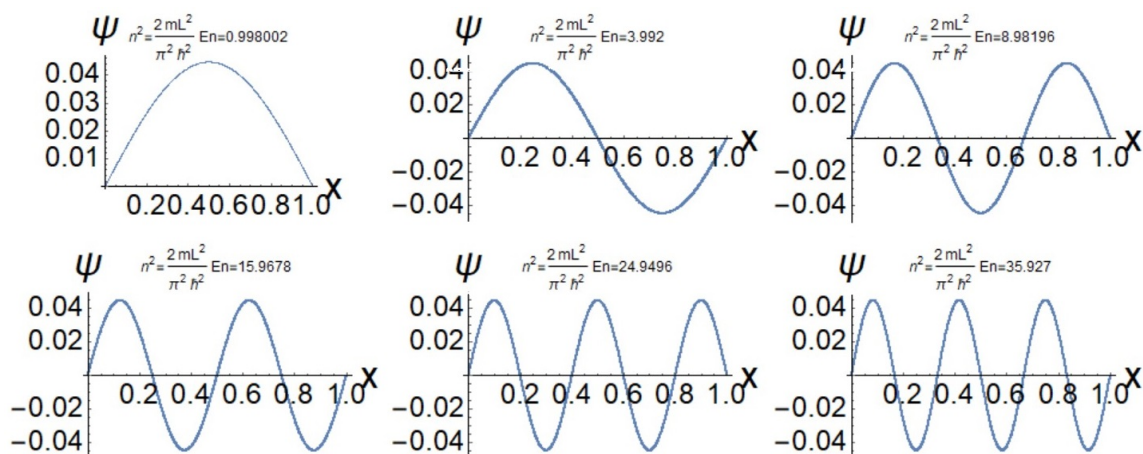
$$n^2 = \frac{L^2 E_n}{t_0 \pi^2}$$

iz numerički dobijenih vrednosti energije metodom konačnih razlika. Ove vrednosti prikazane su zajedno sa odgovarajućim talasnim funkcijama slika 5.18 za 6 energijski najnižih nivoa. Sa povećanjem n raste i odstupanje od egzaktnih rešenja, osim toga tačnost zavisi i od izbora numeričkih parametara.

```

skup={};
Do[
  coo=Table[i*h,{i,n}];
  psi=Transpose[{coo,rez[[i,2]]}];
  gr=ListPlot[psi,
    AxesLabel->{Style["x", 28], Style["\Psi", 28]},
    LabelStyle->{Thickness[0.02], Directive[Black]},
    AxesStyle->Directive[Black,24],ImageSize->Medium,
    PlotLabel->
      "\!\(\*\SuperscriptBox[\(n\), \(\2\)]\)=\!\(\*\FractionBox[\(2 \
\*\SuperscriptBox[\(mL\), \(\2\)]\), \(\*\SuperscriptBox[\(\[Pi]\), \
\(\2\)] \*\SuperscriptBox[\(\[HBar]\), \(\2\)]\)]\) En=" <>
      ToString[rez[[i, 1]]/(Pi^2)];
  AppendTo[skup, gr],{i,6}

```



Slika 5.18: Talasne funkcije čestice u beskonačno dubokoj potencijalnoj jami dobijene numerički metodom konačnih razlika za $n = 1, 2, \dots, 6$.

Deo II

Cinderella

6 Osnovni pojmovi i tehničke napomene

Cinderella je računarski program namenjen za *dinamičku* ili *interaktivnu geometriju*. Ova disciplina je nastala početkom osamdesetih godina, u vreme kada su postali dostupni računarski miš i displej dovoljno dobre rezolucije. Tada je postalo moguće zameniti tradicionalne instrumente za geometrijske konstrukcije, lenjir i šestar, računarom. Od samog početka su postale očigledne dve prednosti konstrukcija izvedenim na računaru u odnosu na klasične konstrukcije, nepromenjene od vremena Antičke Grčke: povećana preciznost i interaktivnost. Velika preciznost računara je omogućila osobama koje nisu vične crtanju, da sa lakoćom nacrtaju i odštampaju precizne i detaljne geometrijske crteže. Međutim, interaktivnost je otvorila potpuno novu perspektivu; omogućila je sagledavanje osnovnih relacija među elementima geometrijske konfiguracije, a to je zapravo, sadržaj geometrijskih teorema. Naime, nakon same konstrukcije, program nam omogućava da promenom parametara dobijemo izmenjenu konstrukciju. Ako promenimo položaj neke tačke, dužinu duži, veličinu ugla itd., računar u realnom vremenu preračunava pozicije svih ostalih elemenata konstrukcije i tako dobijamo novu konfiguraciju polazeći od stare. U takvom okruženju je daleko lakše uočiti odnose i relacije koje se ne menjaju pri promeni određenih konfiguracionih parametara. Kratko rečeno, dinamička geometrija je "geometrija u pokretu".

Prvi programi za dinamičku geometriju su se pojavili krajem osamdesetih godina: Cabri Geometre (1988.) i Geometer's Scratchpad (1989.) Do danas je razvijeno više desetina, možda i stotina različitih programa ove namene. Svi oni rade na sličan način: prilikom same konstrukcije parametrizuju geometrijske elemente i pretvaraju ih u algebarske izraze. Tipovi algebarskih izraza zavise od geometrijskog sistema primenjenog u konkretnom slučaju i oni se obično svode na formalizam kompleksne projektivne geometrije. Prema saznanjima autora ove knjige, najmoćniji sistem je tzv. Kejli-Klajnova kompleksna geometrija i ona je veoma uspešno primenjena u programskom paketu Cinderella. On omogućava da se eliminišu potencijalne nejednoznačnosti koje se mogu pojaviti u samoj konstrukciji prilikom neprekidne promene parametara konstrukcije. Istovremeno, ona omogućava generalizaciju dinamičke geometrije za primenu u fizičkim problemima. Već smo nazvali dinamičku geometriju, geometrijom u pokretu, a to nije ništa drugo do kinematika! Prema tome, o istom trošku smo dobili i program za kinematiku. Pošto se i geometrijska optika svodi na geometriju, program je primenjiv i za konstrukciju i analizu optičkih sistema. Za prelazak sa kinematike na statiku i dinamiku, potrebno je dodatno ugraditi numerički rešavač diferencijalnih jednačina, što je u programu Cinderella i urađeno. Prema tome, radi se o računarskom paketu pogodnom za interaktivne primene u geometriji, kinematici, optici, statici i dinamici.

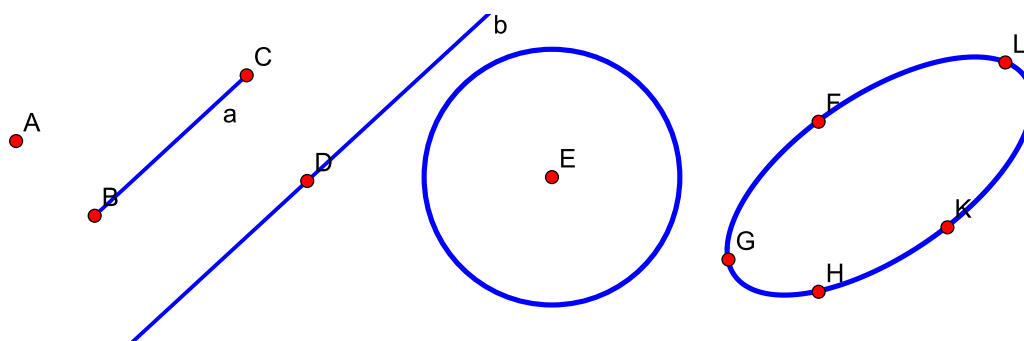
Autori programa Cinderella su Ulrich Kortenkamp i Jurgen Richter-Gebert. Program je napisan u jeziku Java i ona se izvršava posredstvom Java Virtuelne Mašine (JVM), koji je jedna vrsta interpretera koja java binarni fajl prevodi na naredbe za konkretan operativan sistem. Instalacija programa je besplatna i dostupna je na web-sajtu <https://www.cinderella.de>. Dostupne su instalacije za standardne operativne sisteme Windows, Mac OS i Unix.

Ako pažljivo pogledamo logotip programa Cinderella, zapazićemo da se on sastoji iz tri međusobno povezana kruga. Tri kruga simbolizuju tri osnovna dela iz kojih se ovaj program sastoji: Cinderella, CindyLab i CindyScript. Strelice koje ih povezuju ukazuju na to da delovi nisu nezavisni, već da međusobno interaguju na kompleksan način. Cinderella je jezgro ovog programa i zadužena je za dinamičku geometriju. To je modul koji proračunava vrednosti (vezanih) parametara, korišćenjem tehnika kompleksne Kejli-Klajnovne geometrije. CindyLab je okruženje za fizičke simulacije. Opremljeno je numeričkim rešavačem diferencijalnih jednačina (Dormand-Prince-45 integrator). CindyScript je programski jezik pogodan za pisanje kratkih skriptova.

Sve objekte u programu Cinderella možemo podeliti u tri klase:

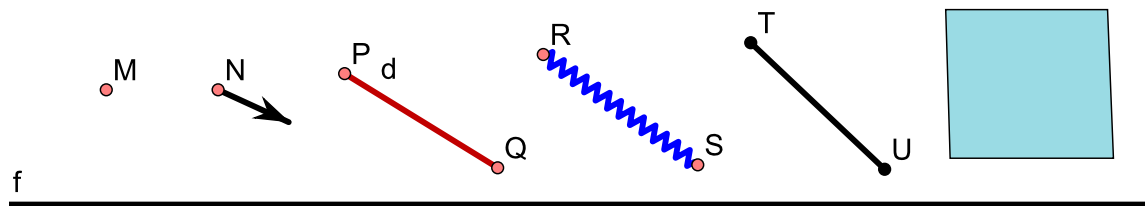
- geometrijski,
- fizički (materijalni) i
- grafički.

Geometrijski objekti su tačka, duž, prava, kružnica, konusni preseki, mnogouglovi i sl. Svaki od njih



Slika 6.1: Osnovni geometrijski objekti

je definisan skupom parametara ili koordinata. Pri tome, duž je definisana koordinatama početne i krajnje tačke, prava parametrizirana koje definišu njenu jednačinu ili koordinatama tačaka kroz koje prolazi, kružnica koordinatama centra i poluprečnikom itd. Brojne su relacije koje možemo definisati među geometrijskim objektima. Primera radi, pravu možemo definisati tako da bude paralelna, ortogonalna ili pod zadatim uglom u odnosu na drugu pravu. Zbog toga razlikujemo dva tipa parametara: slobodne i vezane. Vezani su definisani funkcionalnim odnosima sa drugim objektima, a slobodni su svi ostali.



Slika 6.2: Osnovni fizički objekti

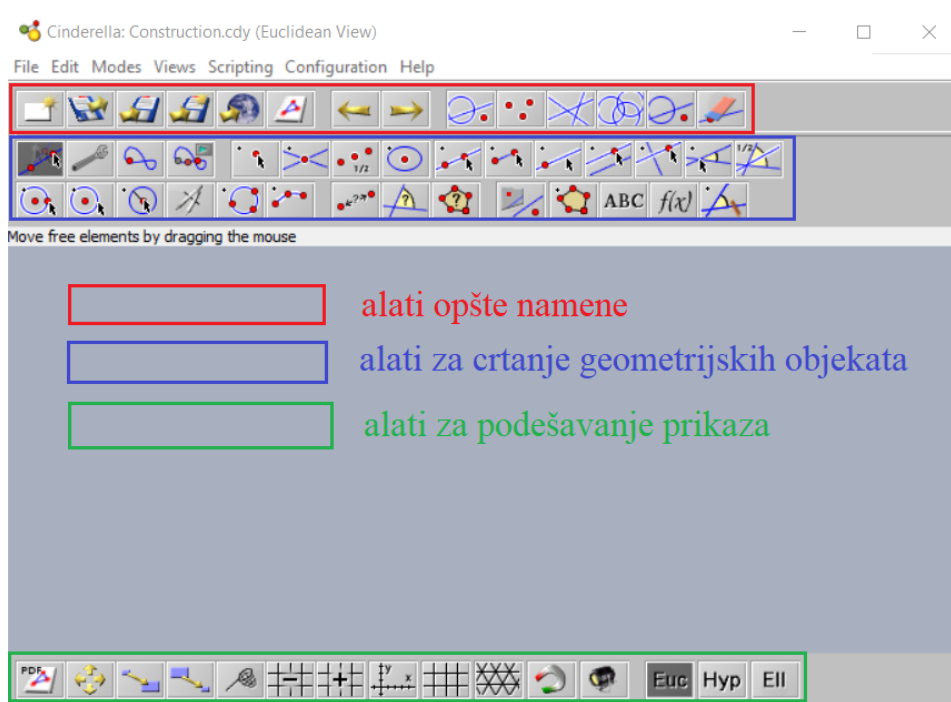
Fizički objekti su materijalna ili masena tačka sa ili bez pridruženog vektora brzine, opruga nulte i nenulte ravnotežne dužine, horizontalni i kosi reflektori (elastične površine), gravitaciono polje, magnetno

polje i sl. Opruga nulte ravnotežne dužine služi za modeliranje proizvoljne interakcije među materijalnim tačkama. Reflektori modeliraju površine od kojih se materijalne tačke elastično ili neelastično odbijaju.

Pod grafičkim objektima podrazumevamo slike tačaka, linija i sl., koje su generisane komandama iz CindyScript-a i koje nije moguće kontrolisati neposredno kursom na radnoj površini. Prikaz grafičkih objekata se na ekranu ne razlikuje od geometrijskih i fizičkih objekata. Grafičke objekte nije moguće selektovati kursom, pa je to najjednostavniji način brze provere tipa objekta.

Izgled radne površine je prikazan na slici 6.3. Pored standardnog padajućeg menija uočavamo tri vrste alata (koje je po potrebi moguće izmeniti) kao i samu površinu namenjenu za crtanje, ovde obojenu u sivo (boju površine za crtanje je takođe moguće promeniti, ali je zbog razlikovanja selektovanih od neselektovanih objekata, siva boja najbolji izbor).

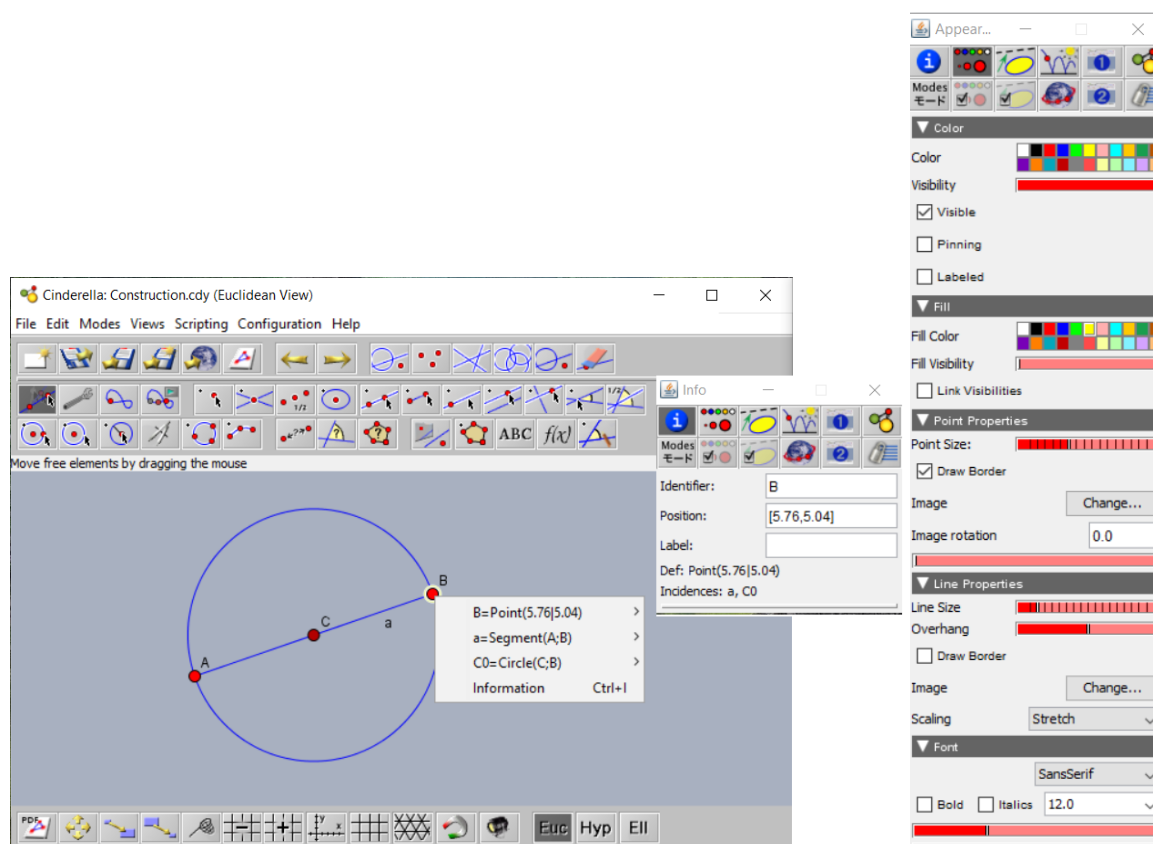
Alati opšte namene služe za otvaranje i snimanje projekata, (de)selektovanje objekata i njihovo brisanje, kao i vraćanje proizvoljan broj koraka unazad (i ponovo unapred). Na dnu su alati za podešavanje prikaza: pomeranje površine levo-desno-gore-dole, uvećavanje i umanjenje prikaza, koordinatni sistem, rešetka-grid (kvadratna i trougaona), "magnet" koji omogućava precizno crtanje tačaka: u čvorovima rešetke. U sredini su alati za crtanje osnovnih geometrijskih objekata: tačke, duži, prave (kroz dve tačke, paralele i normale), kružnice (sa zadate dve tačke, centrom i poluprečnikom, kao i sa slobodnom vrednošću poluprečnika), konusnih preseka, mnogouglova, kao i za merenje rastojanja i uglova. Važno je napomenuti da ovi alati nisu jedini; postoje i dodatni alati unutar liste modes u padajućem meniju. Zapravo broj alata je praktično neograničen, ako se uzme u obzir da je dodatnim skriptom moguće isprogramirati alate koji nam nedostaju.



Slika 6.3: Radna površina u programu Cinderella

Postojeći alati su veoma intuitivno (user friendly) prezentovani, tako da nije potrebno posebno objašnjavati kako se oni koriste. Ukoliko iz dizajna same ikonice nije jasno čemu ona služi, dovoljno je preći kursom preko same ikonice pa da se na vrhu radne površine pojavi tekstualno objašnjenje o upotrebi

datog alata. Informacije o geometrijskim objektima možemo dobiti selektovanjem odgovarajućih objekata i desnim klikom miša. Primer takve situacije je prikazan na slici 6.4. Tu su, konkretno, prikazane informacije o selektovanoj tački B : njene koordinate, kao i podaci o objektima koji su joj susedni, duž a definisana sa početnom i krajnjom tačkom, A i B , kao i kružnicom $C0$ definisana centrom (tačka) C i tačkom na periferiji A . Klikom na neki od ponuđenih oznaka, otvara se informacioni prozor prikazan na desnom delu slike. Na njemu ponovo vidimo osnovne podatke o tački B , ali na njemu postoji više dodatnih tabova-ikonika. Najčešće koristimo tab prikazan na slici desno. On se koristi za podešavanje izgleda - veličine, boje, oznake, vidljivosti selektovanog geometrijskog objekta.



Slika 6.4: Podaci o selektovanim geometrijskim objektima i tipičan izgled informacionog prozora (desno)

Zbog postojanja velikog broja ugrađenih alata intuitivno dobro prezentovanih, veoma je lako koristiti ovaj program na najjednostavnijem (basic) nivou. Može se početi odmah nakon instalacije! Svi dodatni detalji se mogu naći u veoma detaljno napisanom help-u. Zadatak ovog udžbenika se sastoji prvenstveno u izboru pogodnih primera iz odgovarajuće oblasti, na kojima je relativno lako ilustrovati funkcionalnost programa Cinderella. Dodatni zadatak je, da neke poznate probleme i primere iz oblasti fizike i matematike koji su zastupljeni u nastavi fizike, od osnovne škole do fakulteta interaktivno obradi i ilustruje iz novog ugla, ugla dinamičke geometrije.

6.1 Prvi susret sa programom

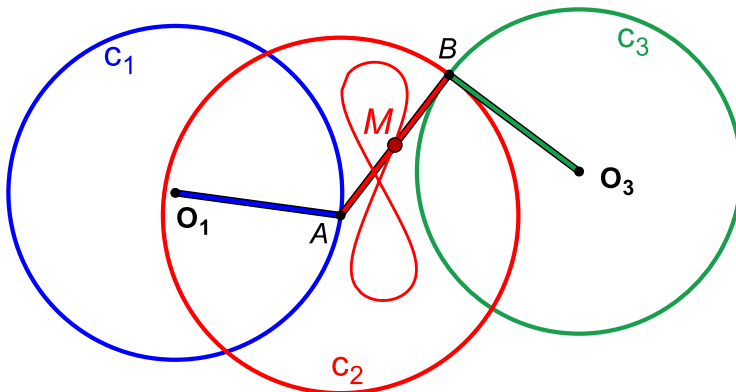
U uvodnoj glavi, u nekoliko jednostavnih primera koji slede, ilustrovaćemo upotrebu osnovnih tipova objekata kao i konstrukcije složenijih objekata od elementarnih. Cilj je samo da se stekne osnovni utisak o mogućnostima koje ovaj program nudi. U glavama koje slede, trudićemo se da, kroz pažljivo odabrane primere - zadatke, to uradimo sveobuhvatno i sistematski.

Primer 1: Vatov mehanizam

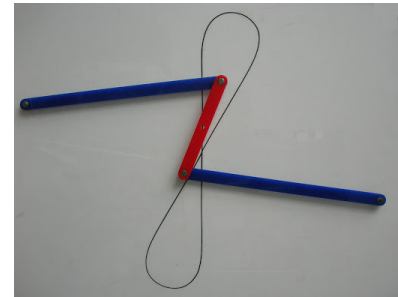
Konstruisati interaktivni model Vatovog (J. Watt) zglobnog mehanizma sa tri kruta štapa i dva zgloba.

Zglobni mehanizmi se sastoji od krutih štapova povezanih "zglobovima" oko kojih mogu slobodno da rotiraju. Najpre je potrebno modelirati kruti štapić: svaka duž čiji je jedan kraj u centru kružnice fiksnog radijusa, a drugi na njoj periferiji, ima fiksiranu dužinu, određenu poluprečnikom kružnice. Na slici je prikazana konstrukcija.

Najpre je konstruisana kružnica fiksnog radijusa C_1 sa centrom



u O_1 i duž O_1A , koja igra ulogu krutog štapa. Zatim je konstruisana kružnica C_3 , sa (slobodnim) centrom u O_3 , pa kružnica C_2 , i tačka B u preseku C_2 i C_3 . Kružnice C_2 i C_3 su fiksnog radijusa, $R(C_3) = R(C_1) \neq R(C_2)$. Na kraju konstruišemo duži O_3B i AB koje igraju ulogu preostala dva kruta štapa. Tačka M je konstruisana kao bisektrisa duži AB . Na kraju je, upotrebom alata za generisanje putanje (locus-a) tačke M kao posledica kretanja tačke A duž kružnice C_1 , generisana putanja, koja na slici podseća na osmicu. Na konačnoj konstrukciji kružnice, koje ovde imaju pomoćnu ulogu, ne treba da



Slika 6.5: Fotografija Vatovog zglobnog mehanizma zajedno sa generisanom putanjom

Slika 6.6: Prikaz interaktivne konstrukcije Watt-ovog mehanizma

Pod bisektrisom duži podrazumevamo tačku koja deli duž na dva jednaka dela. Pravu koja seše duž po sredini pod pravim uglom nazivamo simetralom duži

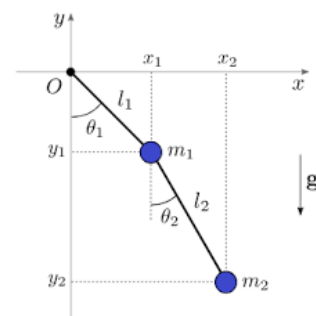
budu vidljive, a to je moguće postići podešavanjem debljine linije na nulu. Interaktivnost je omogućena slobodom u izboru položaja tačaka O_1 i O_3 . Pomeranjem jedne od njih moguće je uticati na oblik putanje tačke M .

Drugi primer ilustruje istu ideju primenjenu na fizičke objekte, tj. u fizičkom okruženju.

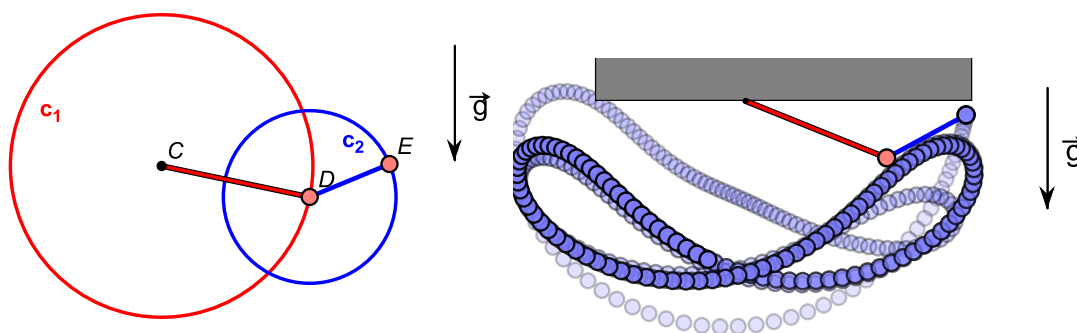
Primer 2: Dvojno matematičko klatno

Konstruisati dvojno matematičko klatno i generisati jednu putanju klatna izvedenog iz ravnoteže u gravitacionom polju.

Ovaj primer je dinamičke prirode i zato pored geometrijskih, koristimo i CindyLab alate. Dvojno matematičko klatno se sastoji od dva nadovezana obična matematička klatna. U tu svrhu je potrebno masenu (materijalnu) tačku konstruisati na kraju neistegljive niti i uključiti gravitaciono polje u okruženju. Neistegljivu nit modeliramo kao kruti štap iz prethodnog zadatka, samo je na kraju materijalna tačka umesto geometrijske. Gravitaciono okruženje se modelira prostim crtanjem vektora \vec{g} sa uključenim alatom za (homogeno) gravitaciono polje. Dakle, konstruišemo sferu c_1 sa slobodnim centrom u C , materijalnu tačku D na njenoj periferiji, zatim sferu c_2 sa centrom u D i materijalnu tačku E na njenoj periferiji. Putanja materijalne tačke se uključuje u podešavanjima (informacioni prozor) za selektovanu materijalnu tačku E . U primeru sa Arhimedovom spiralom biće pokazan i drugi način za crtanje putanje pokretne tačke, pomoću CindyScript-a. Konstrukcija je prikazana na slici 6.8 levo, a putanja na istoj slici desno.



Slika 6.7: Geometrija dvojnog klatna



Slika 6.8: Konstrukcija dvojnog matematičkog klatna (levo) i jedna putanja (desno)

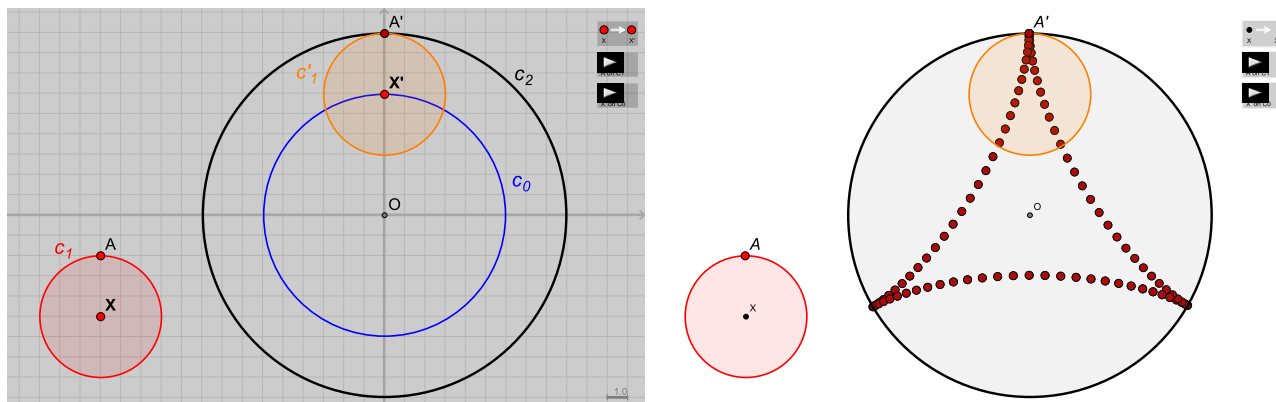
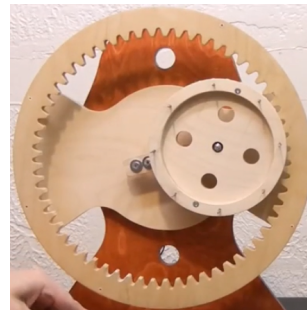
Sledeći primer ilustruje primenu simetrijskih transformacija (izometrija) u programu Cinderella.

Primer 3: Astroida

Nacrtati putanju tačke na periferiji kružnice dijametra r , koja se kotrlja bez proklizavanja unutar kružnice većeg dijametra R , u slučaju kada je: a) $R/r = 3$ i b) $R/r = 4$.

Ovde je u pitanju složeno kretanje tačke u ravni: tačka na periferiji manje kružnice rotira oko nje, dok centar manje kružnice istovremeno rotira oko (zamišljene) kružnice sa poluprečnikom $R - r$. Rezultat je kriva koja se obično naziva astroida. Program Cinderella ima ugrađeni alat za animaciju kretanja tačke duž prave ili kružne linije. Mi ćemo zadatak svesti na dve simulane animacije kretanja tačke duž kružnice.

Konstrukcija je prikazana na slici 6.9 na levoj strani, a putanja u slučaju pod (a), na desnoj. Kotrljanje je simulirano sa dve nezavisne rotacije čije su ugaone brzine naknadno usklađene. Redosled koraka u konstrukciji je sledeći. Najpre konstruišemo kružnice c_0 (zamišljena) i c_1 tako da im je odnos poluprečnika $1 : 2$. Zatim konstruišemo kružnicu c_2 sa poluprečnikom jednakim zbiru poluprečnika od c_1 i c_0 . Zatim sledi ključni korak - definisanje translacije sa vektorom $\overrightarrow{XX'}$. Kružnicu c'_1 kreiramo kao translacionu sliku kružnice c_0 , a tačku A' kao sliku od tačke A . Na kraju definišemo animaciju: tačke A duž kružnice c_1 rotira sa ugaonom brzinom $\omega = 2$ i tačka X' duž kružnice c_0 sa ugaonom brzinom $\Omega = 1$. Konačan rezultat, astroidna putanja tačke A' je prikazana na slici desno.

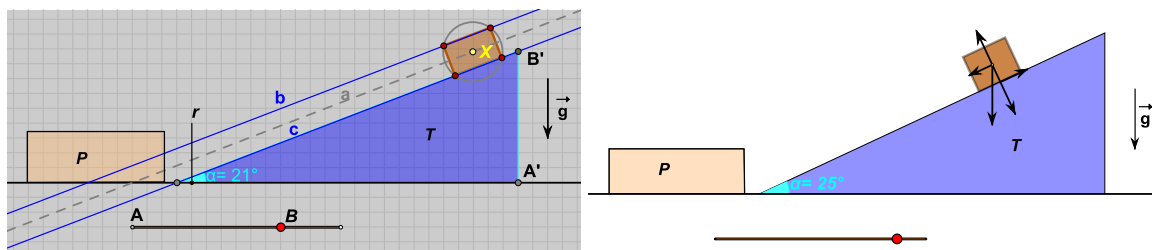


Slika 6.9: Konstrukcija kotrljajuće kružnice sa vidljivim pomoćnim elementima (levo) i konačan rezultat (desno)

U programu CindyLab je definisan alat za proračun kretanja masene tačke u homogenom gravitacionom polju. Prirodno pitanje je kako napraviti jednostavnu simulaciju često potrebnu u školi, simulaciju kretanja predmeta na strmoj ravni u vertikalnom gravitacionom polju. Sledeći primer pokazuje moguće rešenje koristeći materijalnu tačku u gravitacionom polju.

Primer 4: Strma ravan

Konstruisati animaciju kretanja tela duž strme ravni sa promenljivim nagibnim uglom strme ravni i mogućnošću prikaza sila koje deluju na telo.



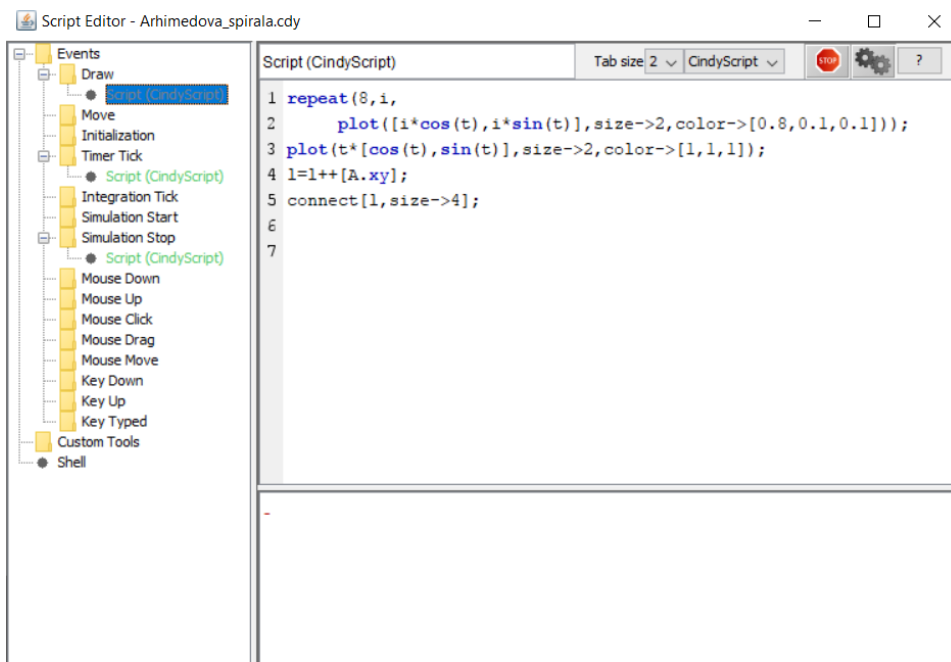
Slika 6.10: Konstrukcija strme ravni sa vidljivim pomoćnim elementima (levo) i konačan rezultat (desno)

Osnovna ideja je sledeća: ako materijalnu tačku konstruišemo tako da leži na kosoj pravoj liniji, i ako se ona nalazi u okruženju sa homogenim gravitacionim poljem, program automatski izračunava komponentu gravitacione sile duž date prave i izračunava trenutnu brzinu tačke. Konstrukcija je prikazana na slici 6.10 na levoj strani. Njena osnova je materijalna tačka X koja je fiksirana za pravu a i može da se kreće samo duž nje. Prave b i c su paralelne sa a i na jednakom rastojanju od nje. Nagib pravice a se podešava sa položajem tačke B na duži - klizaču ispod strme ravni. Dodat je i reflektor r koji ne dozvoljava da tačka X "pobegne" sa radne površine. Na samoj slici izgleda da je reflektor kutija P , ali to je zapravo samo pravougaonik koji ga simulira. Kada se učine nevidljivim svi elementi konstrukcije koji ne spadaju u simulacije tela na strmoj ravni, dobijamo sliku na desnoj strani. Samo telo je reprezentvano pravougaonikom koji je nacrtan oko tačke X : najpre je konstruisana kružnica slobodnog dijametra sa centrom u X , a temena pravougaonika su dobijena u njenom preseku sa pravama b i c . Vektori sile sa normalnom i paralelnom projekcijom su naknadno doctrani.

pomoću alata šestar je rastojanje AB preneto na $A'B'$

Sav potencijal, odnosno prava magija programa Cinderella se postiže tek primenom CindyScripta. Čitav niz novih efekata i funkcija moguće je definisati i pokrenuti sa samo par linija koda. Suština je u sledećem: već konstruisanim geometrijskim ili fizičkim objektima moguće je promeniti osobine ili dodati neku novu funkciju sa komandama CindyScript jezika, a vezu uspostavlja "dot" operator, definisan po analogiji sa operatorom koji u objektno orijentisanim jezicima služi za pristup pojedinim članovima klasa. Na primer,

ako je geometrijska tačka označena sa A , tada primenom dot operatora $A.xy$ dobijamo njene koordinate, a ako je u pitanju prava d , sa $d.slope$, njen nagib (= tangens ugla u odnosu na horizontalu). Sve komande programa CindyScript su izlistane u helpu i to po abecednom redu, ali i grupisane po tipu funkcija. Kod se unosi u posebne slotove prikazane u scrip prozoru, kao što je prikazano na slici 6.11. Njemu se pristupa preko padajućeg menije ili skraćenice "ctrl"+"9". Odabrani slot određuje u kojem se trenutku dati skript primenjuje. Npr. u slotu "Initialization" skript se izvršava samo prolikom njegove inicijalizacije, koju je uvek potrebno izvršiti da bi komande bile učitane u radnu memoriju, a postiže se pritiskom tastera "shift"+"enter". Komande upisane u "Draw" slotu se izvršavaju sve vreme pri renderovanju slike na displeju. Postoji i čitav niz slotova u kojima se komande izvršavaju pri specifičnim aktivnostima miša ili tastature.



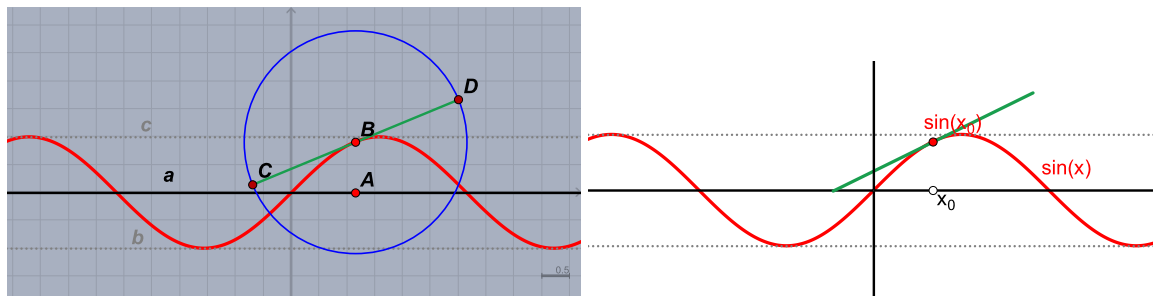
Slika 6.11: Izgled script prozora u Cinderella programu

Nekoliko jednostavnih situacija sa primenom skipta je obrađeno u primerima koji slede.

Primer 5: Sinus tangent plot

Prikazati grafik sinusne funkcije zajedno sa tangentom u proizvoljnoj tački.

Konstrukcija i konačni izgled grafika su prikazani na slici 6.12. Potrebni geometrijski elementi su: tačka A koja leži na horizontalnoj pravoj, tačka B čije su koordinate definisane u skriptu, prava (zelena) kroz tačku B čiji je nagib definisan u skriptu. Postoji i kružnica fiksnog dijametra sa centrom u B čija je uloga da definiše dve tačke



Slika 6.12: Grafik sinusne funkcije sa tangentom u proizvoljnoj tački. Konstrukcija (levo) i finalna animacija (desno)

(C , D) na jednakoj udaljenosti od B , tj. da od prave napravi duž sa centrom u tački B . Prateći skript za datu konstrukciju je:

```
//citav kod upisan u Draw slotu
//crtanje (plot) sinusoide
plot(sin(#), start->-2*pi, stop->2*pi, size->4, color->(1,0,0));
//koordinate tacke B u zavisnosti od koordinata tacke A
B.xy=[A.x, sin(A.x)];
//podesavanje nagiba prave d: nagib = prvi izvod sinusne fukcije u x=A.x
d.slope=d(sin(#), A.x);
```

Komanda // označava komentar u CindyScript-u: tekst desno od komande je komentar. Ovde je iznad svake linije koda naveden komentar koji objašnjava koja je uloga date linije koda.

U finalnom prikazu, na desnoj strani slike, svi pomoćni geometrijski elementi su prikazani sa nultom debljinom linije. Pomeranjem položaja tačke A , odnosno x_0 u finalnom prikazu, pomera se i tačka dodira tangente i sinusoide.

Pomoću skripta je lako animirati proizvoljan tip kretanja tačke.

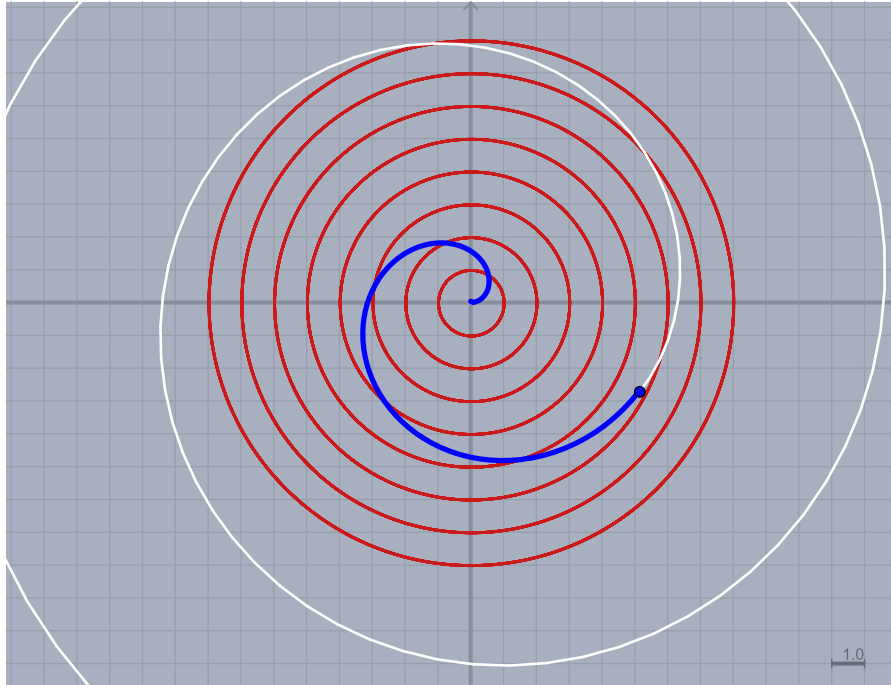
Primer 6: Arhimedova spirala

Animirati kretanje tačke duž Arhimedove spirale. Uzgred, prikazati tačke preseka spirale sa koncentričnim ekvidistantnim kružnicama.

Rešenje je prikazano na slici 6.13.

Jedini geometrijski element je tačka A locirana u koordinatnom početku. Prateći skript je:

```
//u Draw slotu
//crtanje familije kruznica, parametarske jednacine r(t)=[i*cos(t),i*sin(t)]
repeat(8,i,
  plot([i*cos(t),i*sin(t)],size->2,color->[0.8,0.1,0.1])
);
// crtanje spirale
plot(t*[cos(t),sin(t)],size->2,color->[1,1,1]);
//formira skup koordinata kroz koje je tacka prosla = putanje
```



Slika 6.13: Parametarski plot Arhimodove spirale, familije (od 6) koncentričnih kružnica i animirana plava tačka koja se kreće duž spirale

```

l=l++[A.xy];
//crtanje putanje = spajanje tacaka
connect[l,size->4];

//u Timer Tick slotu
//definise vremensku promenljivu simulacije ("brzinu tacke")
t=5*simulationtime();
//definise trenutni položaj tacke A
A.xy=t*[cos(t),sin(t)];

//u Simulation Stop slotu
// kada se zaustavi simulacija (pritisne stop tab) brise skup kordinata putanje u memoriji
l=[];

```

6.2 Dinamička geometrija

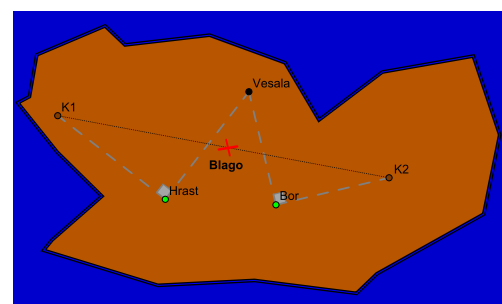
Geometrija je veština ispravnog zaključivanja na pogrešnim slikama.

Rene Dekart

Ovo poglavlje služi da ilustruje pristup geometriji koji je doveo do nastanka dinamičke geometrije. Sledeći primer služi da dočara geometrijski *eksperiment*, tj. osnovni način na koji dolazimo do geometrijskih tvrdjenja - teorema. Kada se uverimo da neko tvrdjenje treba da bude tačno, tada pristupamo drugoj fazi, dokazu, nakon kojeg tvrdjenje postaje teorema. Prvi deo procesa je daleko kreativniji i važniji, ali je obično izostavljen iz standardnih udžbenika geometrije. Dakle, da vidimo kako nastaju geometrijske teoreme, ali primenom računara, što čitav proces znatno ubrzava.

Primer 1: Ostrvo sa blagom

Na staroj piratskoj mapi zakopanog blaga piše sledeće uputstvo. "Na ostrvu Lobanje, nalaze se stara vešala i dva drveta - bor i hrast. Brojati korake od vešala do hrasta, a zatim se kod hrasta okrenuti na desno za 90° i u tom pravcu odbojati isti broj koraka i na tom mesto postaviti prvi kamen. Zatim brojati korake od vešala do bora, kod bora se okrenuti na levo za 90° , odbrojati isti broj koraka i tu postaviti drugi kamen. Blago se nalazi na polovini puta između prvog i drugog kamena." Međutim, kada se nalazač mape našao na pomenutom ostrvu, starih vešala više nije bilo; ostali su samo bor i hrast. Da li je moguće odrediti položaj blaga bez vešala?

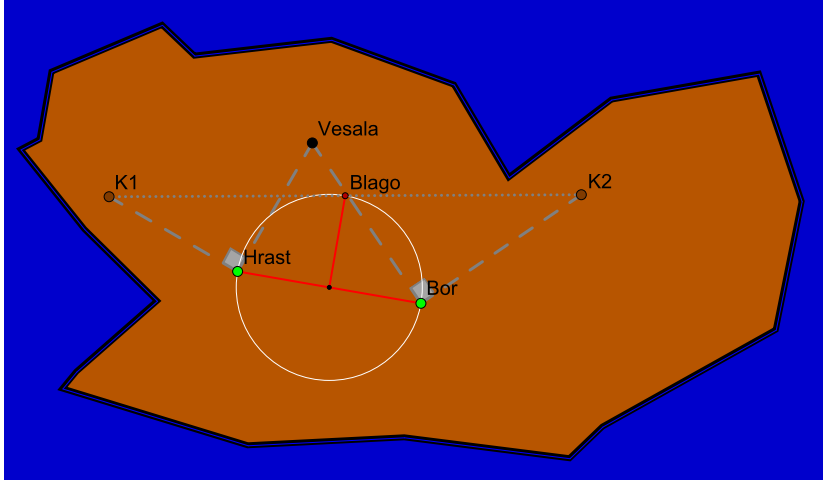


Slika 6.14: Uputstvo: zadatak rešiti posmatranjem promene položaja blaga u zavisnosti od mogućih pozicija vešala, bora i hrasta. Izneti pretpostavku o tačnoj poziciji blaga u odnosu na bor i hrast, a zatim dokazati tačnost pretpostavke.

Geometrijsku konstrukciju je moguće izvesti upotrebom samo osnovnih alata: crtanje kružnice, duži, bisektrise i normale. Konstrukcija je prikazana na slici 6.15. Pomeranjem položaja vešala se vidi da on ne utice na položaj blaga. Pomeranjem položaja bora ili hrasta, vidi se da se blago nalazi u preseku kružnice čiji je dijametar duž koja spaja bor i hrast, i simetrale te iste duži. Dodatni elementi, kružnica i normala potvrđuju hipotezu.

Hipotezu dokazujemo upotrebom kompleksnih brojeva. Neka je karta prikazana u kompleksnoj ravni i neka su objektima (vešala, hrast, bor, kamen 1 i 2, blago) pridruženi kompleksni brojevi $(z_v, z_h, z_b, z_{k1}, z_{k2}, z_B)$. Koordinatni sistem izaberemo tako da važi $z_h = -1, z_b = 1$. U opisanoj konstrukciji potrebno je vektore \vec{VH} i \vec{VB} zarotirati oko H i B za $\pm\pi/2$. Rotacija tačke z_2 oko z_1 za ugao ϕ u kompleksnoj ravni je data izrazom $e^{i\phi}(z_2 - z_1) + z_1$. Prema tome, položaj z_{k1} se dobija rotacijom za $\pi/2$ oko z_h , (množenje sa i) pa je:

$$z_{k1} = i(z_v - (-1)) - 1 = iz_v + i - 1.$$



Slika 6.15: Mapa zakopanog blaga sa svim geometrijskim elementima i idejom rešenja zadatka.

Položaj z_{k2} se dobija rotacijom za $-\pi/2$ oko z_b , (množenje sa $-i$) pa je:

$$z_{k2} = -i(z_b - 1) + 1 = -iz_b + i + 1.$$

Sredina duži koja spaja z_{k1} sa z_{k2} je aritmetička sredina. Dakle,

$$z_B = \frac{z_{k1} + z_{k2}}{2} = i,$$

tj. na mestu koje je predviđeno hipotezom.

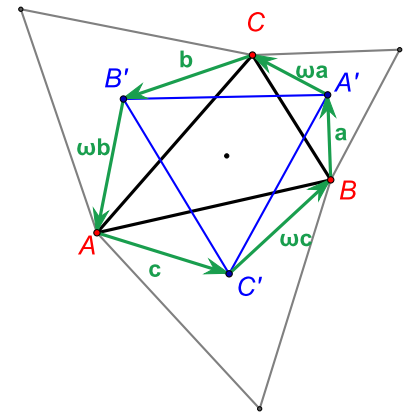
Primer 2: Napoleonova teorema

Ako nad stranicama proizvoljnog trougla ΔABC konstruišemo jednakostranične trouglove, tada njihova težišta (centroide) obrazuju takođe jednakostranični trougao $\Delta A'B'C'$ čije se težište poklapa sa težištem trougla ΔABC .

Ovo tvrđenje je lako proveriti u programu Cinderella. Konstrukcija počiva na tri slobodne tačke A, B, C koje definišu stranice trougla ΔABC . Nad njima konstruišemo, pomoću dve identične kružnice, jednakostranične trouglove. U preseku njihovih težišta (= visina) su tačke A', B', C' , koje definišu stranice trougla $\Delta A'B'C'$. Na kraju konstruišemo težišta od ΔABC i $\Delta A'B'C'$, i konstatujemo da se podudaraju.

Dokažimo njihovo podudaranje koristeći kompleksne brojeve. Neka su trouglovi dati u kompleksnoj ravni zajedno sa vektorima (kompleksni brojevi) $\mathbf{a}, \mathbf{b}, \mathbf{c}$. Označimo sa $\omega = e^{i\pi/3}$, pa je tada ispunjeno i $\omega^3 = -1$, kao i $\omega^2 + 1 = \omega$. Sa slike vidimo da je

$$\vec{AB} = (1 + \omega)\mathbf{c}, \vec{BC} = (1 + \omega)\mathbf{a}, \vec{CA} = (1 + \omega)\mathbf{b},$$



The shortest path between two truths in the real domain passes through the complex domain. J. Hadamard

kao i

$$\overrightarrow{A'B'} = \omega\mathbf{a} + \mathbf{b}, \overrightarrow{B'C'} = \omega\mathbf{b} + \mathbf{c}, \overrightarrow{C'A'} = \omega\mathbf{c} + \mathbf{a}.$$

Sobzirom da je $\overrightarrow{AB} + \overrightarrow{BC} + \overrightarrow{CA} = (1 + \omega)(\mathbf{a} + \mathbf{b} + \mathbf{c}) = 0$, sledi da je i $\mathbf{a} + \mathbf{b} + \mathbf{c} = 0$. Trougao $\Delta A'B'C'$ je pravougli akko je $\omega\overrightarrow{A'B'} = \overrightarrow{A'C'}$. Međutim,

$$\begin{aligned} \omega\overrightarrow{A'B'} - \overrightarrow{A'C'} &= \omega(\omega\mathbf{a} + \mathbf{b}) + (\omega\mathbf{c} + \mathbf{a}) \\ &= (\omega^2 + 1)\mathbf{a} + \omega\mathbf{b} + \omega\mathbf{c} \\ &= \omega(\mathbf{a} + \mathbf{b} + \mathbf{c}) = 0, \end{aligned}$$

pa je teorema dokazana.

Sledeći primer ukazuje na cenu koju treba platiti zbog rezonovanja na pogrešnim slikama.

Primer 3: Pronaći grešku u dokazu

Razmotrimo sledeću konstrukciju: trougao ABC sa simetralom l ugla A , simetralom m stranice BC i tačkom D koja je u preseku l i m . Normale iz tačke D na stranice AB i AC su takođe prikazane, kao i duži DB i DC .

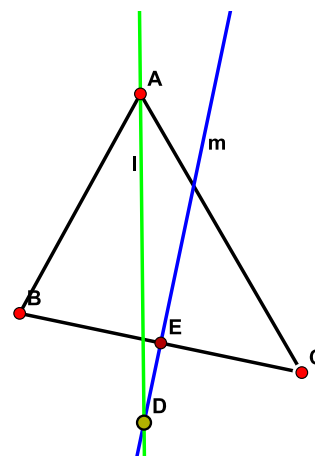
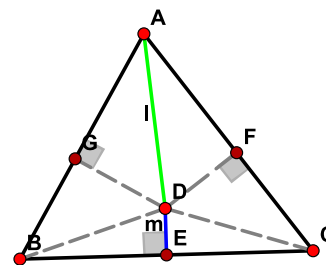
Iz podudarnosti $ADG \cong ADF$, sledi da je AG jednake dužine kao i AF , a DF jednako DG . Iz podudarnosti $BDE \cong CDE$, sledi da je BD jednako sa CD . Odatle sledi da je trougao BDG podudaran sa CDF , pa je i GB jednake dužine kao i FC . Konačno, $AB = AG + GB = AF + FC = AC$, pa je trougao ABC jednakokraki, tj. svaki trougao je jednakokraki!?

Problem nije u zaključivanju već u pogrešnoj slici. Slika je nekorektna jer se tačka D , presek simetrale stranice trougla sa simetralom naspramnog ugla nikad ne nalazi unutar trougla ABC . To možemo jednostavno proveriti pravilnom konstrukcijom u programu Cinderella i variranjem položaja temena trougla.

Dakle, pored mogućnosti variranja parametara koja omogućava eksperimentisanje, slike na osnovu kojih donosimo zaključke više nisu pogrešne. Na ovakvo tvrđenje nikada ne bi došli da smo od početka koristili program za dinamičku geometriju!

Mogućnost crtanja preciznih slika i upotreba boja često omogućava lakše zapažanje bitnih geometrijskih relacija kao i dokazivanje teorema. Razmotrimo sledeći važan primer, teoremu o vezi centralnog i perifernog ugla.

Ova teorema se često pripisuje samom Napolenu Bonaparti. U poznatom udžbeniku geometrije, Coxeter i Greitzer navode: "the possibility of [Napoleon] knowing enough geometry for this feat is as questionable as the possibility of his knowing enough English to compose the famous palindrome, ABLE WAS I ERE I SAW ELBA."



Primer 4: Veza centralnog i periferijskog ugla

Dokazati sledeću važnu teoremu: Centralni ugao (β) nad proizvoljnom tetivom (AB) kružnice uvek je dvostruko veći od periferijskog ugla (α) nad istom tetivom.

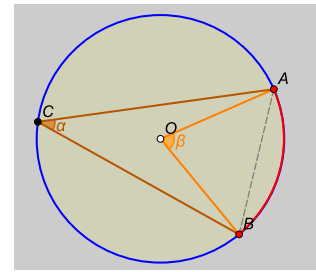
Nad jednom tetivom možemo nacrtati samo jedan centralni i beskonačno mnogo periferijskih ulova. Sve slučajeve možemo klasifikovati u tri tipa. Najjednostavniji tip (a), kojem odgovaraju samo dva slučaja je kada centralni i periferijski ugao imaju zajednički krak koji leži duž dijametra kružnice.

U ovom slučaju dokaz je očigledan. Na osnovu stava o zbiru uglova u trouglu, sledi da je $\beta = \alpha + \alpha = 2\alpha$.

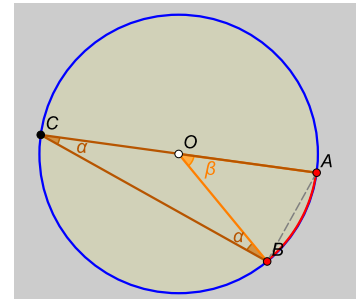
Drugi tip (b) obuhvata slučajeve kada centar kružnice O leži u unutrašnjosti periferijskog ugla. Povlačenjem pomoćne prave koja spaja teme periferijskog ugla C sa centrom kružnice O , svaki od ovih slučajeva se svodi na prethodni slučaj. Sledi da je $\beta_1 = \alpha_1$ i $\beta_2 = \alpha_2$, pa je i $\beta = \beta_1 + \beta_2 = 2(\alpha_1 + \alpha_2) = 2\alpha$.

Najsloženiji slučajevi pripadaju trćem tipu (c). To su slučajevi kada centar kružnice leži izvan periferijskog ugla. I ovaj tip je moguće svesti na tip (a), samo je konstrukcija malo složenija.

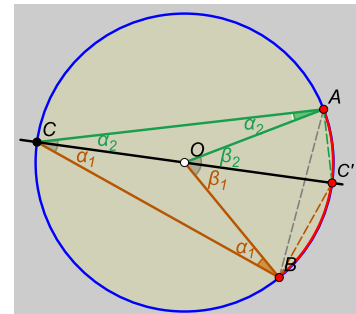
Potrebno je ponovo povući pravu koja spaja teme C sa centrom kružnice O . Ona preseca kružnicu u dijametralno suprotnom temenu C' , koje pak, treba spojiti sa temenima A i B . Na slici je sada jasno uočljivo da imamo dve nove tetive: Φ između C' i B i Γ između C' i A . Obe tetive zajedno sa pripadajućim uglovima pripadaju tipu (a). Primenom stava koji smo dokazali na tetivu Γ , sledi da je $\beta + \phi = 2(\alpha + \phi/2)$, tj. da je $\beta = 2\alpha$. Teorema je dokazana.



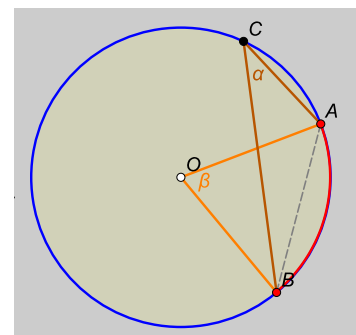
Slika 6.16: tip (a)



Slika 6.17: tip (b)



Slika 6.18: tip (c)



Slika 6.19: Svođenje tipa (c) na tip (a)

7 Pravolinijske i kružne konfiguracije i kretanja

Ova glava je posvećena geometriji i kinematici pravolinijskih i kružnih konfiguracija i kretanja. To su, sa jedne strane, najjednostavnije krive i upotrebu progama za interaktivnu geometriju je najbolje početi sa njima. Sa druge strane one su univerzalne, i sa njima se nužno susrećemo veoma često u realnim situacijama. Takođe, kao što ćemo se uveriti, one su osnova za kreiranje kompleksnijih konfiguracija i složenih kretanja. Primeri su uređeni od geometrijskih ka kinematičkim. Veza sa dinamikom će biti uspostavljena u drugoj polovini ovog dela knjige.

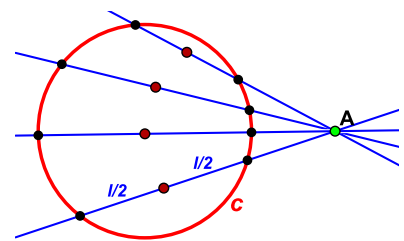
U sledećih osam zadataka je demonstrirana upotreba alata "Create a Locus". Naime, veliki broj zadataka iz geometrije i kinematike zahteva da se odredi skup tačaka ili trag jedne tačke čiji je polazaj fiksiran konstrukcijom, u zavisnosti od vrednosti slobodnih parametara konstrukcije (pozicije slobodnih tačaka, ugla slobodnih pravih itd.). Nakon crtanja tražene putanje ili skupa tačaka, program omogućava i da se pokretanjem animacije dodatno uverimo u verodostojnost rezultata. Naravno, rešenja ćemo obrazložiti geometrijskim zaključivanjem.

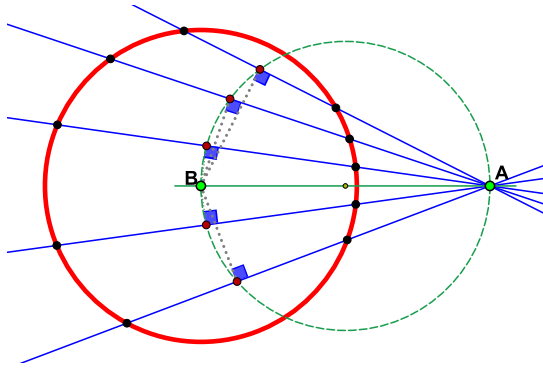
Primer 5

Data je kružnica c i tačka A koja ne pripada kružnici. Odrediti koju figuru formira skup sredina tetiva (crvene tačke na slici) kružnice c koje sadrže tačku A .

Odgovor je - kružnica. Simetrala tetive prolazi kroz centar kružnice. Na taj način, sve dobijene tačke su temena kod pravog ugla nad zajedničkom hipotenuzom BA , pa su ekvivalentne skupu tačaka koji se vidi nad datom duži pod pravim uglom, tj. kružnici.

U Cindrella-i je to moguće jako jednostavno uraditi. Potrebno je konstruisati samo jednu sečicu iz tačke A na kružnicu c . Ona je definisana koordinatama tačke A i jednim parametrom: uglom φ u odnosu na horizont AB . Vrednost parametra φ definiše koordinate sredine tetive. Primenom alata "Create a Locus" varira se parametar



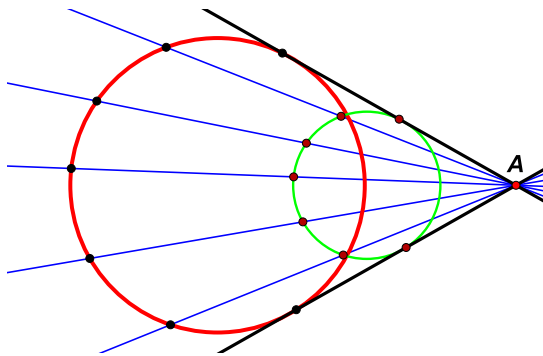
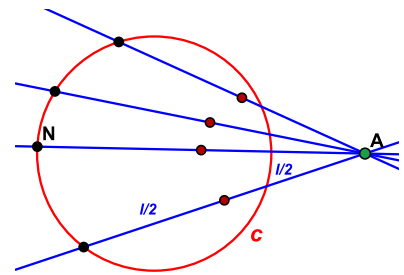


φ i izračunavaju odgovarajuće vrednosti položaja sredina tetiva koje leže na kružnici.

Primer 6

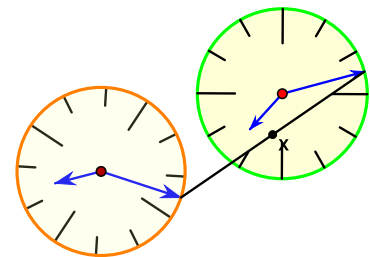
Data je kružnica c i tačka A koja ne pripada kružnici. Odrediti koju figuru formira skup sredina duži AN , (crvene tačke na slici) gde je N proizvoljna, udaljenija tačka preseka kružnice c i prave koja prolazi kroz tačku A .

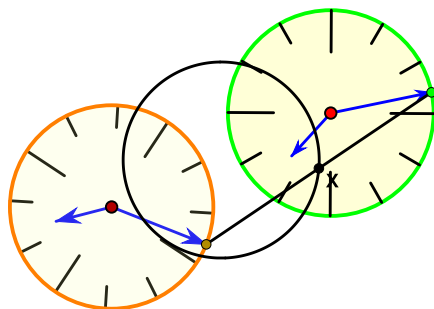
Ovo je mala varijacija prethodnog zadatka i primena alata "Create a Locus" direktno daje odgovor. Odgovor je - kružnica sa centrom u sredini duži koja spaja tačku A i centar kružnice c . Lako je uveriti se da je ona rezultat homotetije sa centrom u tački A i koeficijentom $1/2$ primenjenoj na kružnicu c .



Primer 7

Dva ravna, okrugla sata leže na stolu. Satovi su sinhronizovani (tj. pokazuju isto vreme). Odrediti oblik putanje sredine duži A_1A_2 (tj. putanju tačke x) između vrhova minutnih kazaljki.





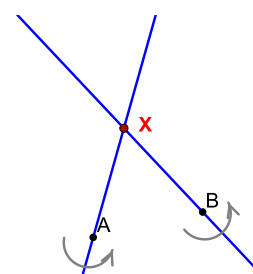
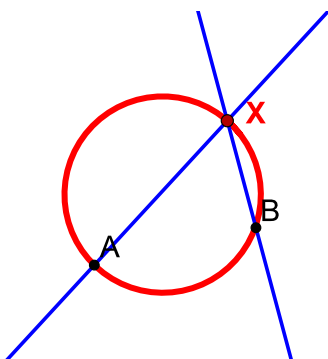
Slika 7.1: Prikaz konačne konstrukcije. Naravno sam izgled sata nije od suštinske važnosti, ali se sugeriše čitaocu da se lično uveri da je i verne prikaze moguće dosta lako nacrtati u Cinderella-i

Odgovor je - kružnica. Translirajmo jedan od satova tako da im se centri poklope. Tada je jasno da je putanja sredine duži kružnica (to je putanja simetrale ugla između minutnih kazaljki). Translirajmo sat na početnu poziciju. Translacijama kraja duži, njegova sredina se translira za polovinu dužine, pa opet dobijamo kružnicu. Naravno, primenom "Create a Locus" alata, dobijamo direktnu potvrdu.

Primer 8: "Rotirajuće prave"

Dve prave, l_A i l_B rotiraju oko tačaka A i B jednakom ugaonom brzinom ω . Odrediti putanju po kojoj se kreće tačka X preseka pravih l_A i l_B .

S'obzirom da prave rotiraju jednakom ugaonom brzinom, ugao između njih je sve vreme jednak. To znači da je ugao pod kojim se "vidi" duž AB sve vreme jednak, a kriva za koju važi to svojstvo je kružnica.

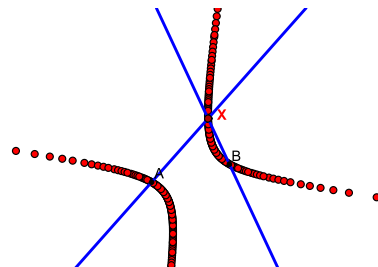


Slika 7.2: Crvena kružnica je rezultat primene alata "Create a Locus". Pokretanjem animacije dobijamo potpuno vernu predstavu o samom ogledu koji je moguće realizovati sa dva štapa.

Primer 9: "Rotirajuće prave: verzija 2"

Dve prave, l_A i l_B rotiraju oko tačaka A i B jednakom ugaonom brzinom ω , ali u suprotnim smerovima. Odrediti putanju po kojoj se kreće tačka X preseka pravih l_A i l_B .

Rešenje je najlakše dobiti analitički, tako što sredinu duži AB uzmemo za koordinatni početak, a za x -osu pravac sa kojim su u nekom trenutku obe prave paralelne. Kako koeficijenti pravaca pravih zavise linearno od vremena, to jednačina tačke njihovog preseka, kao rešenje sistema linearnih jednačina, mora biti kvadratna kriva. U trenutku kada su prave paralelne, tačka njihovog preseka je "tačka u beskonačnosti," pa zaključujemo da je reč o hiperboli.

**Primer 10: "Rotirajuće prave: verzija 3"**

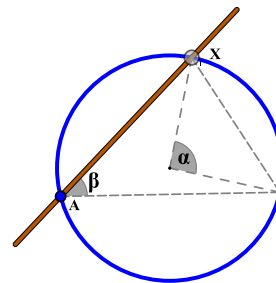
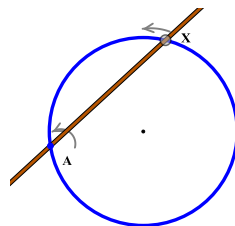
Dve prave, l_A i l_B rotiraju oko tačke A i B ugaonim brzinama ω i 2ω . Odrediti putanju po kojoj se kreće tačka X preseka pravih l_A i l_B , ako su u početnom trenutku, tačke A , B i X kolinearne.

Ovaj zadatak je ekvivalentan narednom.

Primer 11: "Rotirajući prsten"

Prsten M je navučen na kružni obruč C . Kroz prsten je provučen i štap l . Ako štap l rotira ugaonom brzinom ω oko tačke A na periferiji obruča, kojom se ugaonom brzinom rotira prsten na obruču?

Rešenje oba prethodna problema je poznati stav koji tvrdi da je periferijski ugao (β) nad proizvoljnom tetivom jednak polovini centralnog ugla (α) nad istom tetivom. Dakle, prsten se rotira dvosruko većom ugaonom brzinom od štapa. Ako kroz centar kružnice i prsten postavimo drugi štap, koji se rotira dvostruko brže od prvog, prsten će se i dalje kretati na isti način, pa je to rešenja prethodnog zadatka.

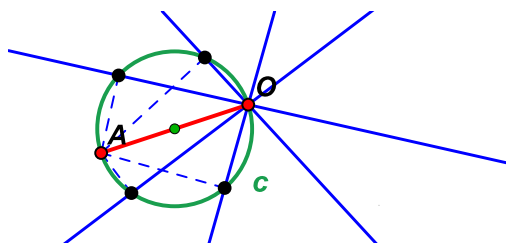


Sledeći zadatak je osnova za čitavu seriju zadataka koji slede. Osnovna ideja je uspostavljanje veze između pravolinijskog i kružnog kretanja, koja je jednostavna, ali nije potpuno očigledna, pa je u osnovi nekih od poznatih optičkih iluzija.

Primer 12

Date su dve tačke u ravni A i O . Odrediti geometrijsko mesto podnožja normala (crne tačke na slici) iz tačke A na sve prave (pramen pravih) koje prolaze kroz tačku O .

Odgovor je - kružnica (na slici označena sa c) jer se iz svake tačke duž AO vidi pod pravim uglom. Duž AO je radijus kružnice,



pa pomeranje tačke A duž pravca kojem pripada AO dovodi do promene veličine kružnice.

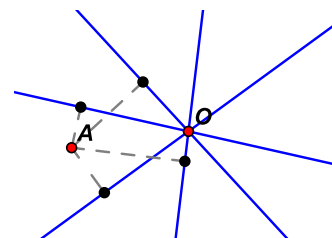
Sledeći zadatak je kinematička verzija prethodnog.

Primer 13

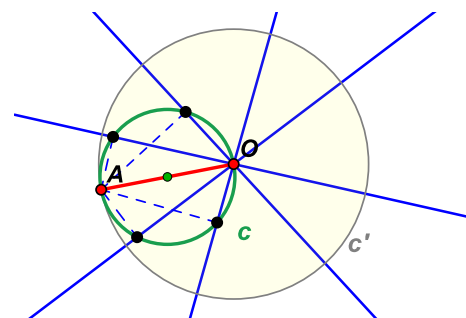
Neka je, na slici iz prethodnog zadatka, konstruisana pomoćna kružnica c' sa centrom u O i poluprečnikom OA . Neka je konstruisana i kružnica c sa centrom u bisektrisi od OA , i poluprečnikom koji je upola manji od dužine duži OA . Zamislamo da se tačka A kreće duž c' , kako se kreće kružnica c u odnosu na c' ?

Pri ravnomernoj rotaciji tačke A duž kružnice c' , njena projekcija na proizvoljan dijametral od c' se kreće harmonijski (tj. jednaka je kosinusu ugla koji prečnik OA zaklapa sa datim dijametrom). Međutim, sve projekcije se sinhrono pomeraju, a iz prethodnog zadatka znamo da formiraju kružicu (c). U svakom sledećem trenutku, kružnica c je pomerena, ali tako da je trenutna pozicija tačke A u dodiru kružnica c' i c . Pokretanjem animacije, stvara se iluzija ktrljanja kružnice c' unutar kružnice dvostruko većeg dijametra c .

Ovaj rezultat možemo iskoristiti pri rešavanju sledećeg zadatka.



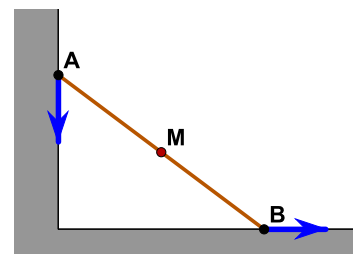
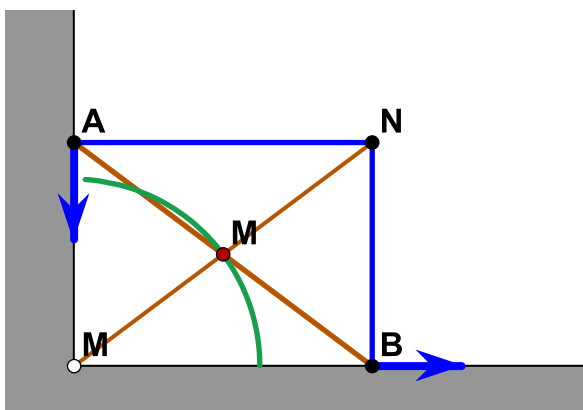
Slika 7.3: Konstrukcija kreirana uz pomoć alata "Create a Locus"



Primer 14: "Mačka na merdevinama"

Merdevine su naslonjene na zid i dodiruju ga u tački A . Tačka B označava tačku u kojoj merdevine dodiruju pod(logu). Mačka (M) stoji na sredini merdevina. Ako merdevine proklizavaju po zidu (i podu), tj. ako se tačka A kreće vertikalno naniže po zidu, a tačka B horizontalno nadesno, po kojoj putanji će se kretati mačka? Ako se mačka ne nalazi na sredini merdevina, po kojoj će se tada putanji kretati?

Rešenje je prikazano na slici 7.4. Merdevine, tj. duž AB , je jedna od dijagonala pravougaonika $ABMN$, pa pri proklizavanju duži AB , duž MN rotira oko tačke dodira zida i poda. Polovina duži MN , a to je tačka M takođe rotira, pa se, dakle, kreće po kružnici poluprečnika $\overline{MN}/2 = \overline{AB}/2$. Ukoliko se mačka ne nalazi na sredini merdevina, njena putanja je eliptična. To je najjednostavnije uočiti ako se razmatra jednačina putanje koja sledi iz konstantnosti dužine merdevina $\overline{AB} : \overline{AB}^2 = (x/a)^2 + (y/b)^2$, gde su $2x/a$ i $2y/b$ koordinate tačke M , a x i y koordinate tačke B i A .



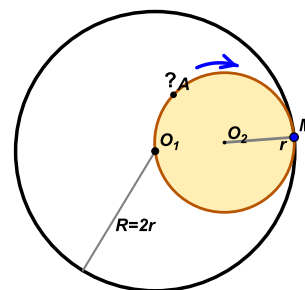
Slika 7.4: Finalna konstrukcija

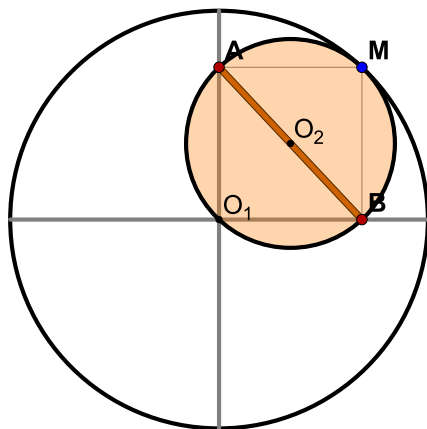
Efekat iluzije kotrljanja je istorijski dobro poznat i nalazimo ga u Kopernikovim spisima o geocentričnom planetarnom sistemu.

Primer 15: "Kopernikov zadatak"

Odrediti putanju proizvoljne tačke na kružnici (obruč) poluprečnika r koji se kotrlja unutar kružnice (obruč) poluprečnika $R = 2r$.

Rešenje je, zapravo već dato u rešenju prethodnog problema. Ako se, naime, sa centrom u tački dodira zida i poda opiše kružnica poluprečnika jednakog dužini merdevina (\overline{AB}), tada su tačke A i B projekcije na y i x osu, iste tačke, tačke M na periferiji kružnice.





Ovo je jedan od načina na koji je moguće pretvoriti kružno kretanje u pravolinijsko. Za drugi poznati način, videti zadatak o Peaucellier-ovom zglobnom mehanizmu.

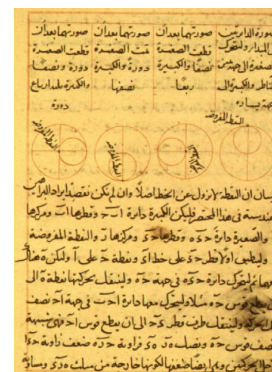
Kretanje tačke M duž kružnice, generiše kretanje tačaka A i B duž vertikale (zida) i horizontale (poda). Ako, međutim, nad duži AB kao prečnikom, opišemo drugu, manju kružnicu, tada je tačka M tačka dodira manje i veće kružnice. Rotacija tačke M je tada ekvivalentna kotrljanju manje unutar veće kružnice. Tačke na periferiji manje kružnice, A i B (sve, zbog simetrije, moraju da opisuju putanju istog oblika) se kreću duž prave linije.

Ovu vezu pravolinijskog i kružnog kretanja prvi je otkrio Persijski matematičar iz 13. veka Nasir al-Din Tusi. Na slici 7.5 je prikaz strane Tusi-jeve knjige u kojoj je princip objašnjen.

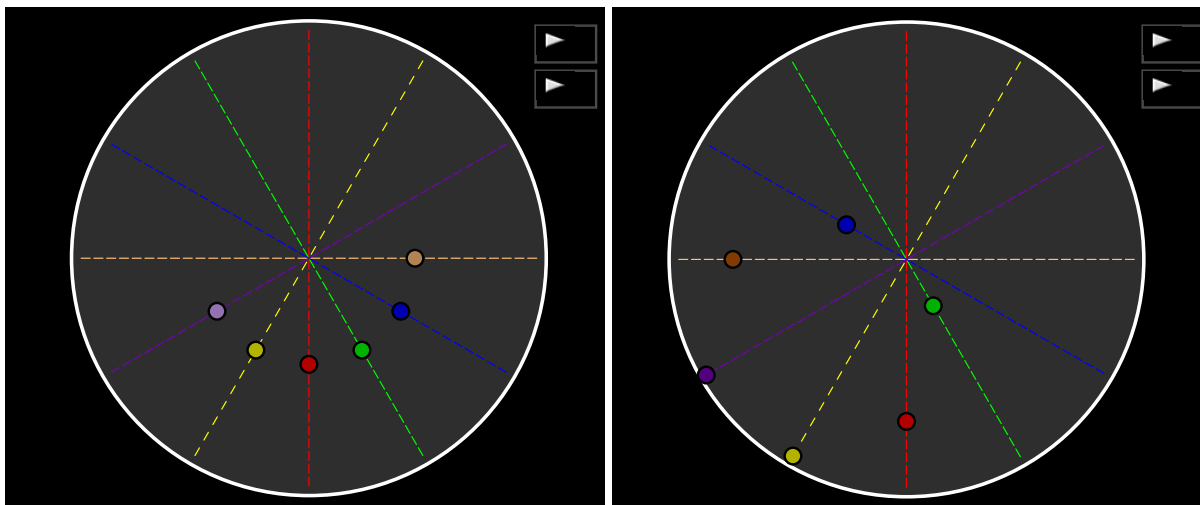
Naime, pod harmonijskim kretanjem tačaka duž pravilno raspoređenih dijametara kruga, obično se podrazumeva simultano kretanje tačaka ka i od centra. To je, u slučaju 6 tačaka koje se kreću duž velikih dijagonala dvanaestougla, prikazano na slici 7.6 na levoj strani. Međutim, pod Tusijevim kretanjem se podrazumeva fazno pomerenom kretanje duž dijagonala u skladu sa situacijom iz zadatka 7. Ako su tačke ravnomerno raspoređene po obodu kružnice dvostruko manjeg od spoljašnjeg kruga, tada harmonijsko kretanje stvara iluziju kotrljanja tačaka oko kružnice. Situacija je prikazana na slici 7.6 zdesna.

Dualni tip kretanja Tusijevom je ravnomerno kretanje sistema tačaka duž kružnica dvostruko manjeg dijametra od osnovne kružnice. I u ovom slučaju je podešavanjem faze moguće postići utisak kolektivnog kružnog i pravolinijskog kretanja. Situacija prikazana na slici 7.7.

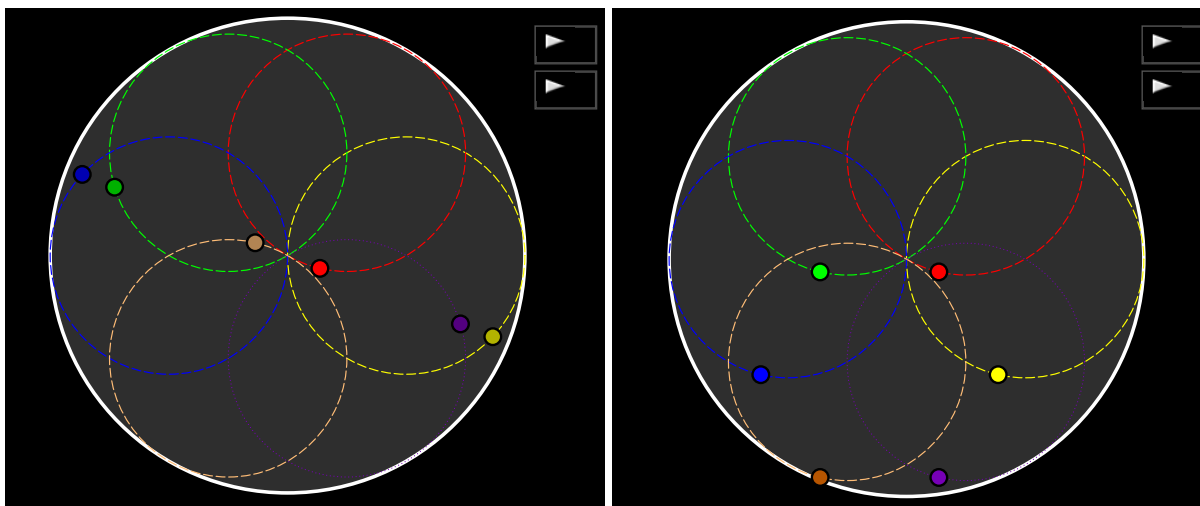
Kombinovanjem Tusijevog i dualnog Tusijevog kretanja je moguće postići efektanu vizuelnu iluziju, kao što su i učinili Arthur Shapiro and Alex Rose-Henig sa American University SAD. Oni su 2013. godine na takmičenju Best Illusion of The Year osvojili drugu nagradu, sa iluzijom Tusi or not Tusi. Originalnu iluziju je moguće naći na adresi <http://illusionoftheyear.com/2013/05/tusi-or-not-tusi/>. Iluzija realizovana



Slika 7.5: Stranica Tusijevе knjige

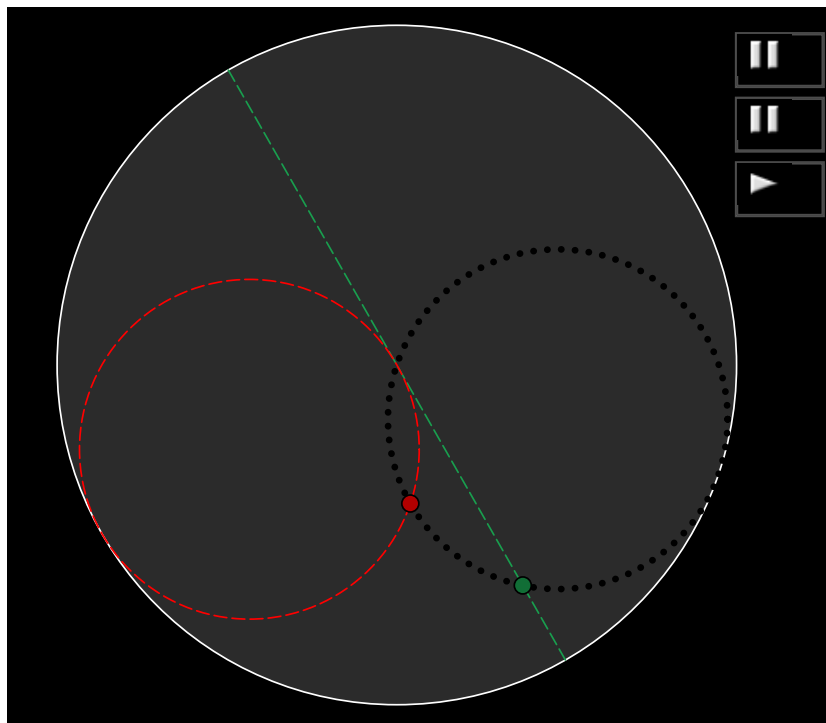


Slika 7.6: Tusijevo kretanje: simultano (levo) i fazno pomereno (desno)



Slika 7.7: Dualno Tusijevo kretanje: simultano (levo) i fazno pomereno (desno)

u Cinderella-i je prikazana na slici 7.8.



Slika 7.8: Cinderella realizacija iluzije Shapiro-a i Rose-Henig-a.

Iluzija se sastoji u simultanom kretanju dve tačke: jedne po dijametru, a druge po maloj kružnici. U svakom trenutku postoji tačno jedna mala kružnica kojoj pripadaju obe tačke. Ako se zatim putanje tačaka učine nevidljivim, a zajednička mala kružnica vidljivom, tada u zavisnosti od toga da li se pažnja usmeri na jednu ili drugu tačku, kružnica se kreće na različite načine: u jednom slučaju se kotrlja, a u drugom samo rotira oko centra velike kružnice.

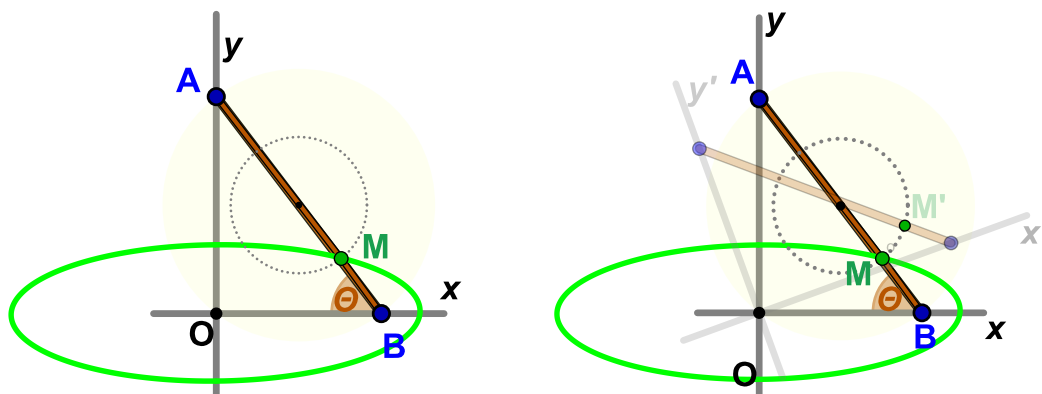
Primer 16: "Arhinedov elipsograf - trammel"

U zadatku 7, pretpostavljeno je da se mačka nalazi na sredini merdevina. Koju će putanju ona opisati ako se nalazi na nekoj drugoj tački na merdevinama? Koju će putanju ona opisati ako "lebdi" na fiksnoj udaljenosti od krajeva merdevina?

Razmotrimo situaciju na slici 7.9 (levo). Neka su sa (x, y) označene koordinate tačke M , i neka je $p = \overline{AM}$, $q = \overline{MB}$. Tada su koordinate tačke M $x = p \cos \theta$, $y = q \sin \theta$. Očigledno je da tada važi:

$$\left(\frac{x}{p}\right)^2 + \left(\frac{y}{q}\right)^2 = 1,$$

a to je jednačina elipse. Na desnom delu slike vidimo da je

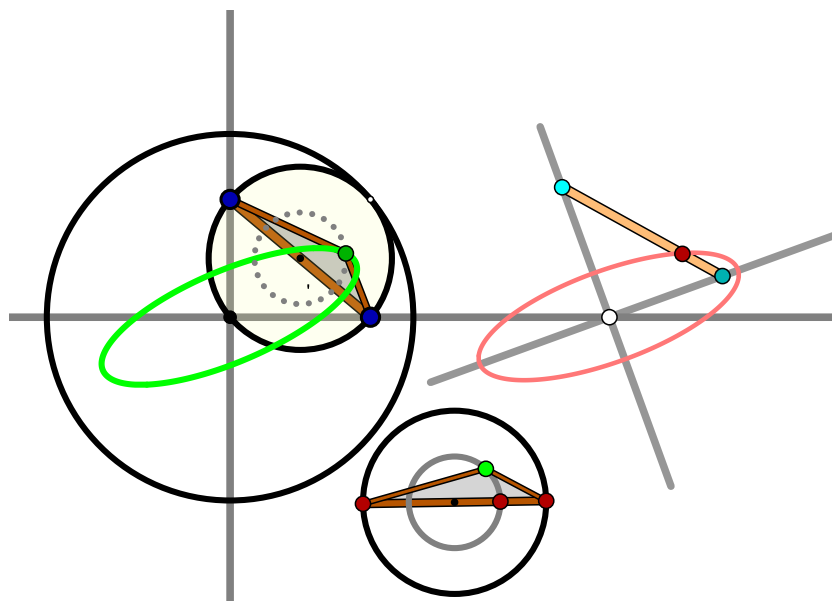


Slika 7.9: Modifikacija situacije "mačke na merdevinama"

izmeštanje položaja mačke u odnosu na krajeve merdevina duž isprekidane kružnice ima isti efekat kao da posmatramo pomoćnu konstrukciju zarotiranu u odnosu na početnu. Dakle, rezultat je opet elipsa samo zarotirana u odnosu na koordinatni početak.

Navedena konstrukcija se koristi u praksi za konstruisanje elipsi i bila je poznata još i Arhimedu, pa je, u njegovu čast, nazvana Arhimedov elipsograf (tramell of Archimedes).

Na slici 7.10 je prikazana realizacija Arhimedovog elipsografa u Cinderella-i, kao i pomoćna konstrukcija koja ilustruje rezultat prethodnog zadatka. Veličina i ugao nagiba elipse se podešavaju položajem zelene i crvene tačke na pomoćnoj kružnici.



Slika 7.10: Arhimedov trammel - elipsograf, finalna konstrukcija

Geometrijski elementi su pomoćni trougao, glavni trougao sa stranama šestarom prekopiranim sa pomoćnog i dve međusobno ortogonalne prave. Alatom "Create a Locus" generisana (zelena) putanja - elipsa.

Teorema o dve kružnice.

Tangente na trajektoriju tačke M , na periferiji kružnice radijusa r , koja se bez proklizavanja kotrlja po krivoj γ , ima pravac definisan prečnikom KL (koji prolazi kroz tačku M) kružnice dvostruko većeg dijametara koji dodiruje podlogu u istoj tački (T), kao i manja kružnica. Drugim rečima, trajektorija tačke M je obvojnica dijametara KL .

Dokaz teoreme.

U svakom trenutku kretanja, brzina tačke M je ortogonalna na duž MT , jer je tačka T trenutni centar rotacije. To znači da je brzina tačke M usmerena duž dijametara veće kružnice (jer je ugao $\angle TML$ prav, pa mora da prolazi kroz dijametralno suprotnu tačku od T , a to je centar veće kružnice).

Iskoristimo ovu teoremu pri rešavanju sledećeg zadatka.

Primer 17: "Astroida"

Odrediti obvojnici merdevina prilikom klizanja po zidu (i po podu).
Ili u drugoj verziji, odrediti obvojnici prečnika kružnice poluprečnika r koji se kotrlja unutar kružnice poluprečnika $R = 2r$.

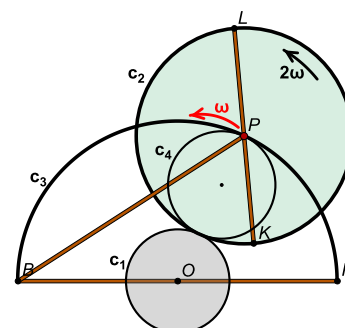
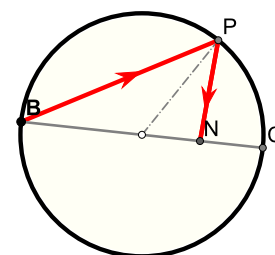
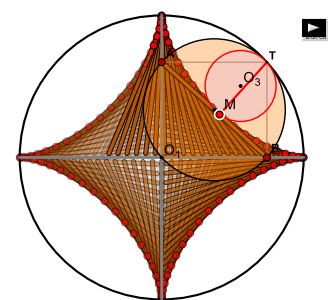
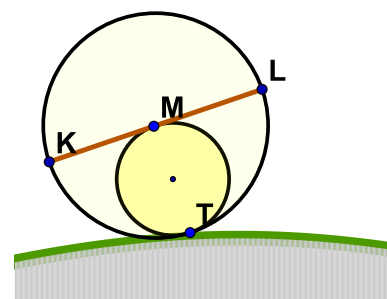
Rešenje je kriva koja se zbog svog oblika, obično naziva astroida. Astroida je definisana kao kriva koja se dobija kao trajektorija tačke na periferiji prilikom rotacije kružnice unutar kružnice dvostruko većeg dijametara. Veza dobijene krive sa definicijom astroide je prikazana na slici dole.

Prethodna teorema je primenljiva na veliki broj zadataka iz optike u kojima je potrebno odrediti obvojnici (kaustiku) sistema zraka.

Primer 18: "Zrak u kružnici"

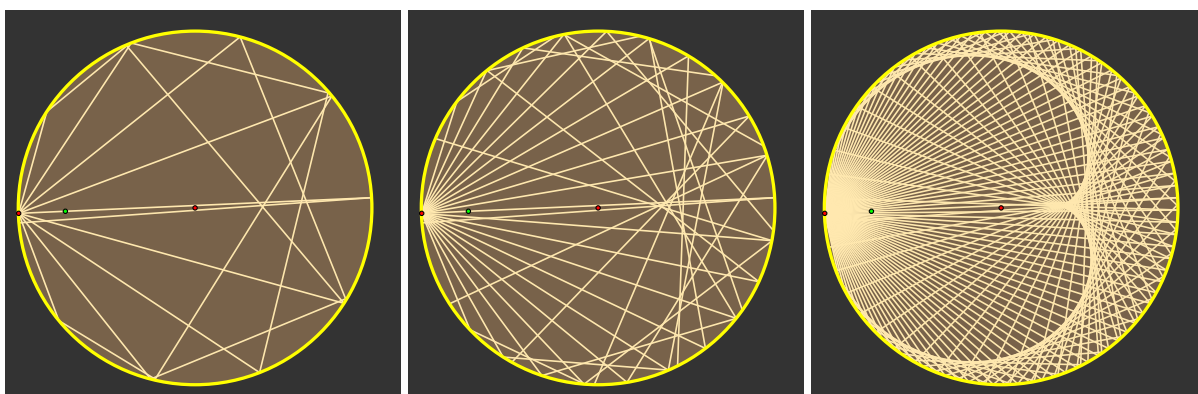
Iz tačke B na periferiji kružnice, emituje se svetlosni zrak. On se reflektuje od zidova bez atenuacije. Odrediti obvojnici reflektovanih zraka u kružnici (*kaustična kriva*).

Razmotrimo kretanje reflektovanog zraka pri rotaciji upadnog zraka oko tačke B . Sa slike je jasno da važi $\angle PNC = \angle BPN + \angle PBN = 3\angle PBN$. To znači, da ako zrak BP rotira sa ugaonom brzinom ω , tada reflektovani zrak rotira sa ugaonom brzinom 3ω , a pri tom, tačka reflektovanja P rotira sa ugaonom brzinom 2ω oko centra O (vidi zadatak 7).



Na osnovu toga, možemo generisati familiju reflektovanih zraka (PN) na sledeći način. Kotrljajmo kružnicu c_2 radijusa $2r$, na slici sa centrom u P i dijametrom LK , oko fiksne kružnice c_1 radijusa $r = |OB|/3$ sa centrom u O . Situacija je ekvivalentna rotaciji centra kružnice c_2 (tj. tačke P) po obimu kružnice c_3 dijametra $3r$ sa ugaonom brzinom 2ω , pri čemu sama kružnica c_2 rotira oko P sa ugaonom brzinom ω . U tom slučaju dijametar LK rotira ugaonom brzinom 3ω , kao i zrak. Na osnovu teoreme o dve kružnice, obvojnica dijametra kružnice c_2 je jednaka trajektoriji tačke M na periferiji kružnice c_4 , radijusa r . Dakle, obvojnica zraka je jednaka trajektoriji tačke na kružnici radijusa r koja se kotrlja oko kružnice jednakog radijusa, sroljika kriva poznata kao kardioida. Sa povećanjem broja zraka obvojnica postaje sve uočljivija.

Naziv kardioida potiče od Starogračkog $\kappa\alpha\rho\delta\iota\alpha$ - srce



Slika 7.11: Formiranje obvojnice pri povećanju broja zraka

Pokazati da jednačina kardioida u polarnim koordinatama glasi

$$r = 1 - \cos \varphi.$$



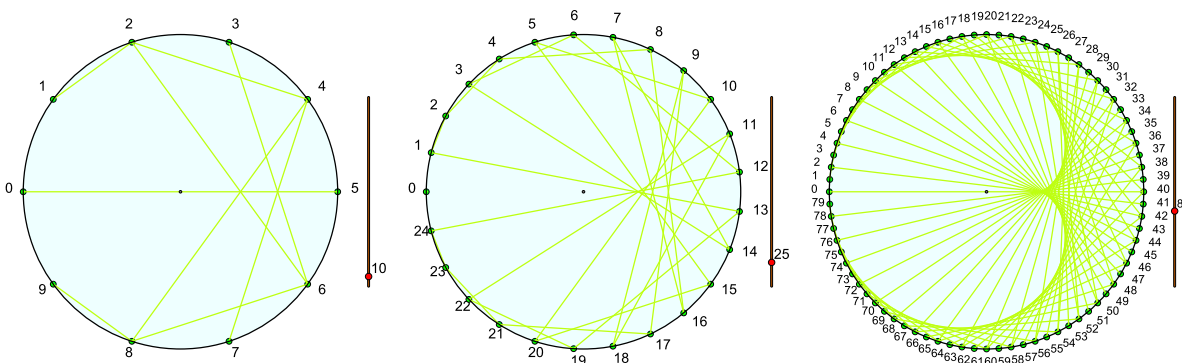
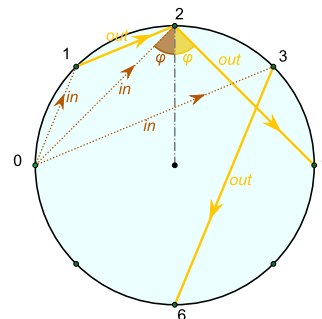
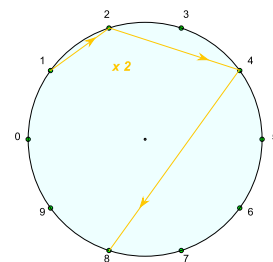
Slika 7.12: Reflekcija u šolji kafe. Levi kraj šolje je postavljen izpod izvora - sijalice.

Primer 19: "Tablica množenja na kružnici"

Na kružnici je nacrtano n ekvidistantnih tačaka numerisanih od 0 do $n - 1$. Dužima povezati parove tačaka numerisanih sa m i $2m \bmod(n)$. Kako izgleda obvojnica skupa duži pri $n \rightarrow \infty$?

Veza ovog zadatka sa prethodnim je objašnjena na slici sa strane. Duži (zraci) dobijeni množenjem rednog broja tačke sa 2 su prikazani žutom bojom i označeni sa *out*. Docrtajmo pomoćne zrake označene sa *in* koji povezuju početnu tačku, označenu sa 0 sa tekućom tačkom m . Očigledno je da su uglovi koje odgovarajuće duži obrazuju sa poluprečnikom nad tačkom m jednaki (označeni su sa φ). Prema tome, duži *in* možemo interpretirati kao upadne zrake emitovane iz tačkastog izvora u 0 i reflektovanih od kružnice. Obvojnica reflektovanih zraka je kardioda i ona je rešenje i ovog zadatka.

Konačni rezultat je prikazan na slici 7.13.



Slika 7.13: Formiranje obvojnice pri $n = 10, 25, 80$

Geometrijski elementi neophodni za konstrukciju su kružnica $C0$ sa centrom u slobodnoj tački i dva klizača: duži sa pokretnom tačkom na njoj. U konkretnom primeru, prvi klizač, koji služi za podešavanje parametra n je duž BC sa pokretnom tačkom D , a drugi, kojim podešavamo parametar m je FE sa pokretnom tačkom G .

Skript za konstrukciju je:

```
n=round(200*|B,D|/|B,C|);
r=C0.radius;A=C0.center;
textsize(18);
drawtext(D+[0.1,0.1],text(n));
m=20*|F,G|/|F,E|;
m=round(m);
```

```

drawtext(G+[0.1,0.1],"$\cdot$ "+text(m));
//formiranje tacaka na obodu kruga
num=0..n-1;
pts=1..n;
forall(pts,i,pts_i=A+r*[cos(-pi-2*pi*(i-1)/n),sin(-pi-2*pi*(i-1)/n)]);
repeat(n,i,drawtext([-0.05,-0.05]+pts_i+0.1*(pts_i-A),text(num_i)));
drawall(pts,size->4.5);
//spajanje odgovarajucih tacaka
prs=1..n;
forall(prs,i,prs_i=[pts_i,pts_(1+mod(m*num_i,n))]);
drawall(prs,size->1,color->hue(m/10),size->2);

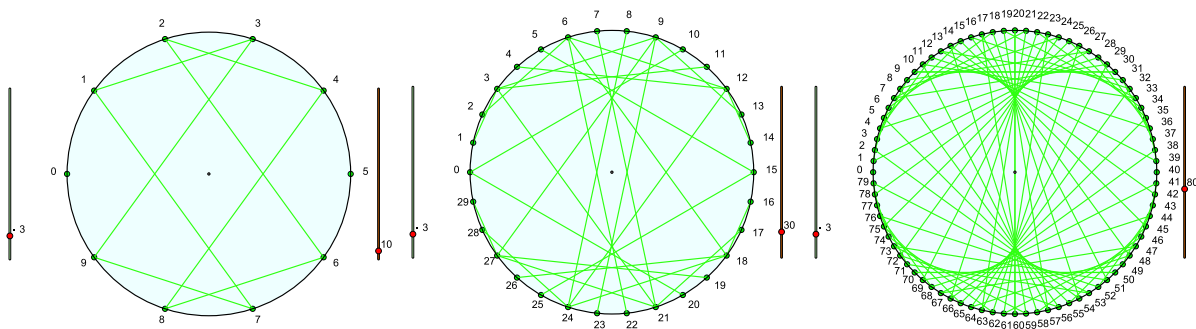
```

Primer 20

Na kružnici je nacrtano n ekvidistantnih tačaka numerisanih od 0 do $n - 1$. Dužima povezati parove tačaka numerisanih sa m i $3m \bmod(n)$. Kako izgleda obvojnica skupa duži pri $n \rightarrow \infty$?

Na osnovu analize sprovedene u prethodna dva zadatka, jasno je da je obvojnica kriva generisana rotacijom kruga radijusa r oko kruga radijusa $2r$ poznata kao "nefroida".

Nefroida od Starogrčke reči za bubreg - νεφρός.



Slika 7.14: Formiranje obvojnice pri $n = 10, 30, 80$.

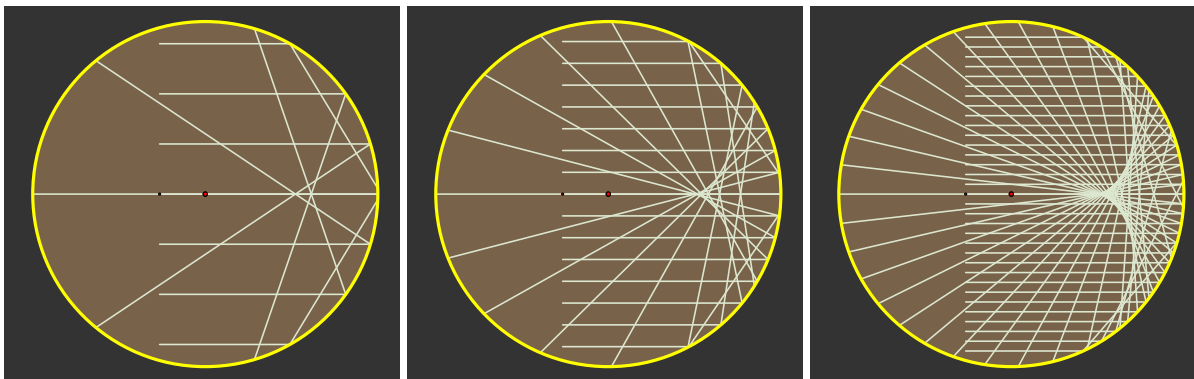
Primer 21: "Kružno ogledalo"

Odrediti obvojnici reflektovanih zraka koji paralelno padaju na polukružno ogledalo (tj. ako je izvor u beskonačnosti).

Odgovor je - kardioida. Rešenje nacrtano u Cinderella-i je prikazano na slici 7.16.



Slika 7.15: Reflekcija u šolji kafe. Izvor je sunčeva svetlost.



Slika 7.16: Formiranje obvojnice pri povećanju broja zraka

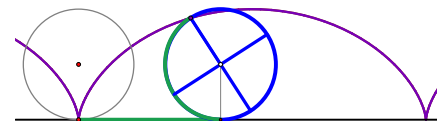
8 Složena kretanja: cikloida i epiciklične krive

Cikloida je kriva koju opisuje tačka na kružnici koja se kotrlja duž prave linije bez proklizavanja. S'obzirom da je točak pronađen bar pre pet hiljada godina, očekivali bi da su za nju, u najmanju ruku znali i stari Grci. Ali nije tako! Prvi put se pojavljuje 1501. godine u opisu Šarla de Bovela (Charles de Bouvelles) kao mehaničko rešenje problema kvadrature kruga. Svoje ime, koje u prevodu sa starogračkog znači "sličan kružnici", duguje Galileju. U svom pismu Kavalieriju iz 1640. on navodi kako razmišlja o ovoj krivoj više od 40 godina. Istoričari matematike su je nazvali "Helen of Geometry". To ime, koje aludira na Jelenu Trojansku, nije izabrano samo zbog njenih brojnih lepih osobina, nego i stoga što je izazvala brojne svađe i nesporazume među najvećim matematičarima XVII veka. Ona je specijalni primer tzv. ruletnih (roulette) krivih, koje nastaju kotrljanjem jedne krive duž druge bez proklizavanja. Cikloide i njene generalizacije su nužan pratilac pravolinijskog i kružnog kretanja. Zajedno sa konusnim preseccima, koji su prirodno uopštenje prave i kružnice.

Matematička analiza je nastala u XVII veku motivisana problemima geometrije i kinematike krivih, među kojima cikloida ima počasno mesto. Njene osnovne osobine vezane za dužinu luka, površinu, probleme tautohrone i brahistohrone, biće opisane kroz zadatke u ovoj zbirci. Pored zadataka, biće ukratko ispričana i njena istorija.

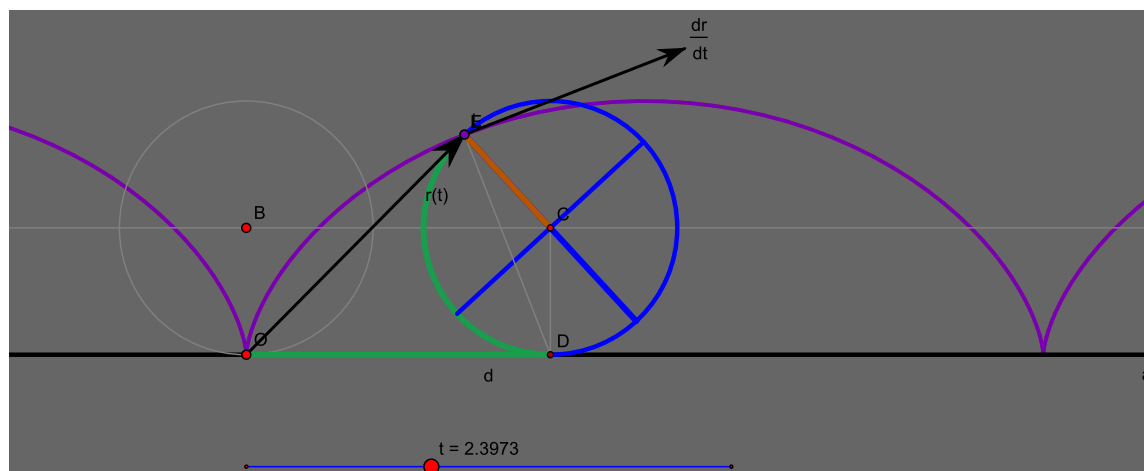
Primer 22: Cikloida 1

Generisati cikloidu kao trajektoriju tačke na obodu obruča koji se kotrlja duž prave linije.



Cinderella nema alat koji može direktno da bude iskorišćen za crtanje cikloide, ali postoji niz načina na koji je to moguće uraditi sa postojećim alatima. Prvi i najopštiji način je pomoću skripta, i to je način na koji je moguće nacrtati bilo koju krivu u bilo kojem programu. Ono što je specifično za ovaj program je interakcija grafičkih i geometrijskih objekata i koja je, zapravo, suština programa za dinamičku geometriju. Prva verzija će ujedno biti iskorišćena za prikaz kontrole geometrijskih objekata pomoću programa CindyScript.

Geometrijski elementi su horizontalna prava a , dve kružnice sa centrom u tački B i C , duž koja spaja koordinatni početak O i tačku E na periferiji druge kružnice, kao i slobodnu pravu kroz E . Prateći



Slika 8.1: Izgled finalne konstrukcije kojom se generiše cikloida

skript za datu konstrukciju je:

```
//podesavanje parametara
B.x=0;
t=(0.x-M.x)/|M,N|*2*pi;
drawtext(0+(.2,.3),"t = "+t);
r=C0.radius;

// parametrizacija
x(t):=r*(t-sin(t),1-cos(t));
C.x=r*t;
E.xy=x(t);

//crtanje krivih
linesize(2.5);
plot(x(t),t,start->-2*pi,stop->4*pi,step->300,
      color->(0.6,0.2,0.1));

//Tangenta
xt(t):=[d((x(#)_1),t),d((x(#)_2),t)];
S.xy=E.xy+xt(t);
```

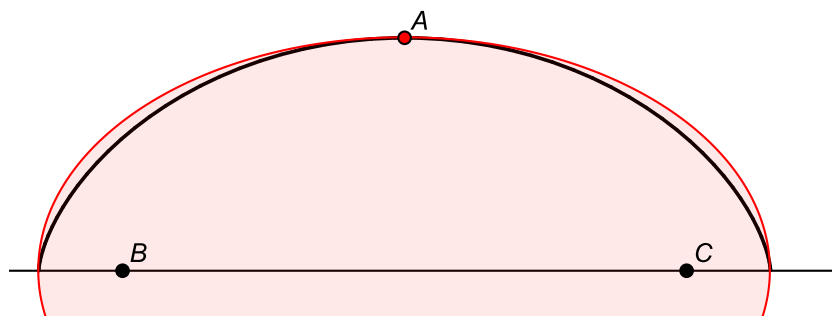
Položaji tačaka C - centar rotirajućeg kruga, i E - tačka na njegovoj periferiji, su definisani funkcijama $r * t$ i $x(t)$ od parametra t . Parametar t je, pak definisan položajem tačke na duži ispod podloge kretanja (horizontalne prave) i skalirana sa 2π . Sama putanja je parametarski plot funkcije $x(t)$. Tangentna linija je konstruisana kao normala na duž \overline{DE} koja je poluprečnik rotacije tačke E oko trenutnog centra rotacije D . Dužina vektora tangente (tj. rastojanje $|E, S|$ od tačke E do tačke S , nevidljive na slici) je određena po definiciji, kao izvod funkcije $x(t)$ za datu vrednost parametra.

Siva kružnica C_0 nad koordinatnim početkom služi za kontrolu radijusa pokretne kružnice - parametar r je definisan kao radijus od C_0 . Pri tom je x -koordinata njenog centra fiksirana na 0.

Pod uticajem Galileja, za cikloиду se zainteresovao francuski matematičar Mersen. On je, oko 1615. godine pretpostavio da je cikloida zapravo polu-elipsa. U sledećem zadatku ćemo se uveriti da je Mersenova pretpostavka bila opravdana.

Primer 23: Mersenova pretpostavka

Nacrtati cikloidu i elipsu čija je velika osa jednaka širini luka cikloide, a mala poluosa jednaka njenoj visini.



Rešenje je prikazano na slici. Luk cikloide je prikazan crnom bojom i nacrtan je na osnovu parametarske formule iz prethodnog zadatka. Elipsa, prikazana crvenom bojom, je konstruisana pomoću žiža B i C i tačke A koja je na najvišoj tački elipse. x koordinata tačke C je definisana sa: $C.x = 2 * A.x - B.x$. Konačno, tačku A pomeramo sve dok velika osa elipse ne postane jednaka širini luka cikloide.

Godine 1628. Mersen je uspeo da nagovori Roberval da proučava cikloidu. Nakon šest godina, on je uspeo da reši tzv. kvadrturni problem cikloide, tj. da odredi površinu ispod luka cikloide. Narednih nekoliko godina uspeo je da reši još dva važna problema: da opiše konstrukciju tangente na proizvoljnu tačku cikloide, kao i da odredi zapreminu dobijenu rotacijom luka cikloide oko horizontale. Međutim, nijedno od ovih otkrića on nije publikovao. Razlog za to je veoma praktičan. Naime, u to vreme, Roberval je predavao na matematičkoj katedri prestižnog Kolež Rojala (Collège Royal), preteče današnjeg Kolež de Fransa (Collège de France). Na svake tri godine predavač se iznova birao na otvorenom takmičenju na kojem bi zadatke postavljao dotadašnji predavač, pa je Roberval čuvajući rešenja u tajnosti mogao da bude u prednosti u odnosu na potencijalne izazivače.

Osnovni problemi analize su u ovo vreme dobro definisani: izračunavanje površina ispod luka krive, kao i zapremine obrtnih tela oko date krive, konstrukcijatangente na proizvoljnu tačku krive, izračunavanje dužine luka krive. Nije postojao univerzalni metod za njihovo rešavanje, već su matematičari svakom problemu iznova pristupali.

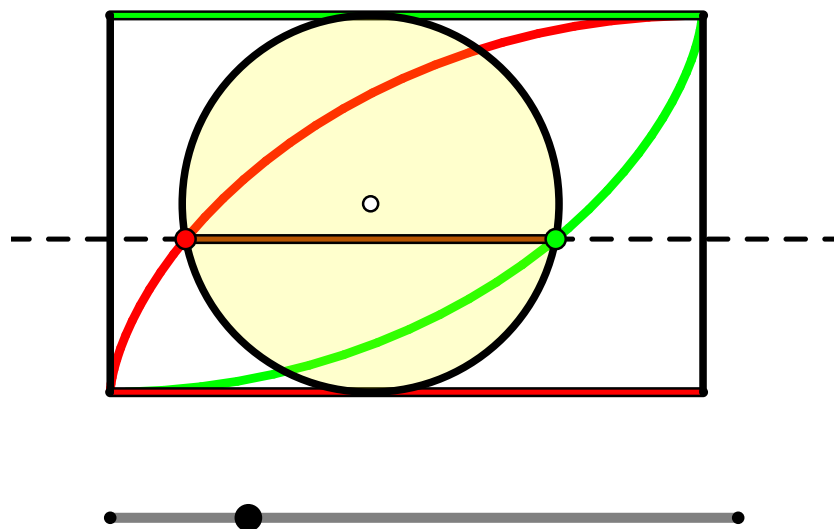
Površinu cikloide Roberval je odredio primenom Toričelijevo principa.

Primer 24: Robervalovo rešenje

Konstruisati interaktivno Robervalovo rešenje.

Rešenje je prikazano na slici. Crveni poluluk cikloide je putanja crvene tačke dok kružnica rotira duž crvene stranice pravougaonika, a zelena - duž zelene. Ako površinu cikloide označimo sa $P(r)$, gde je r poluprečnik kružnice koje je generiše, tada je $P(r)/2 = (2r)(r\pi) - T = 2r^2\pi - T$, gde je sa T označena površina isečka između pravougaonika i cikloide. Primenom Toričelijevo principa zaključujemo da je površina kruga jednaka površini sektora S ograničenog crvenom i zelenom cikloidom. Dakle,

$$(2r)(r\pi) = S + 2T = r^2\pi + 2T \Rightarrow T = r^2\pi/2 \Rightarrow A(r) = 3r^2\pi.$$



Slika 8.2: Prikaz konstrukcije Robervalovog rešenja. Zeleni (crveni) deo luka cikloide je formiran "kotrljanjem" kružnice duž zelene (crvene) stranice pravougaonika

Geometrijski elementi su pravougaonik, kružnica $C0$ upisana u pravougaonik i slobodne tačke M i G na periferiji od $C0$. Prateći skript je:

```
// crtanje lukova cikloide i klizaca
r=C0.radius;
t=|E,K|/|E,F|*2*pi;//polozaj klizaca
```

```

x(t):=r*(t-sin(t),1-cos(t)); // crveni luk
y(t):=((r*(-t-sin(-t))+r*pi),(-r*(1-cos(-t))+2*r)); //zeleni luk

//trenutni položaj: A - centar kruznice, M - vodeca tacka kruznice u zavisnosti od t
if(t<=pi,A.x=r*t;M.xy=x(t),A.x=r*(-t+2*pi);M.xy=y(t-pi));

//trenutni položaj pratece tacke preseka sa cikloidom - zelena tacka
G.xy=[M.x+2*(A.x-M.x),M.y];

// plotovanje cikloidnih lukova
linesize(2.5);
plot(x(t),t,start->0*pi,stop->1*pi,step->400,
      color->(1,0,0));
plot(y(t),t,start->0*pi,stop->1*pi,step->400,
      color->(0,1,0));

```

Robervalovo rešenje kvadrarnog problema Mersen je prosledio Dekartu i Fermau na verifikaciju što su oni i uradili. Dekart je čak zlobno dodao da je *"Roberval previše radio da bi dobio rezultat od tako malog značaja"*. Tokom 1638. godine, sva trojica su nezavisno jedan od drugog, poslali Mersenu rešenje problema konstrukcije tangente na cikloidu. To je bio povod za svađu. Fermaovo rešenje, koje je poslužilo za modernu definiciju tangente kao graničnog položaja sečice, Dekart je nazvao *"najsmešnijom glupošću koju je u životu video"*.

Dekartovo rešenje se zasniva na pojmu trenutne ose rotacije, poznate iz prve glave knjige.

Primer 25: Dekartovo rešenje

Ilustrovati Dekartov metod konstrukcije tangente na cikloidu.

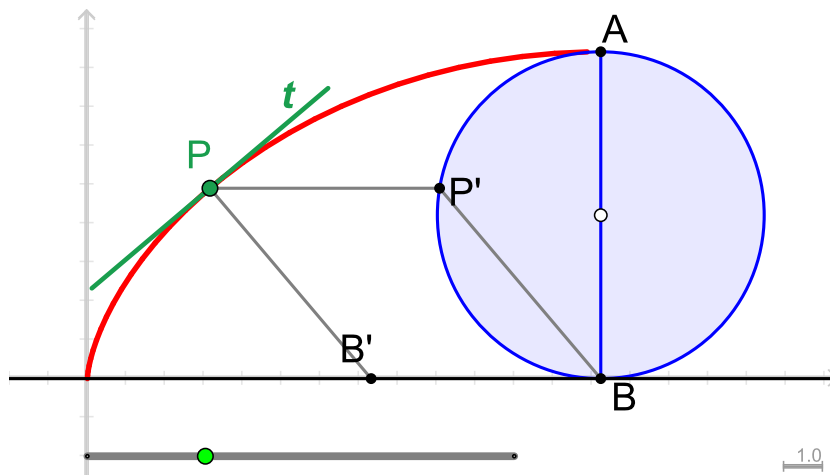
Duž PP' je konstruisana tako da bude paralelna horizontali po kojoj se kružnica kotrlja; Duž PB' je paralelna duži $P'B$, a sama tangenta predstavlja normalu kroz tačku P na duž PB' . Prateći skript je:

```

\\ u Draw slotu
r=C0.radius;
t=|E,C|/|C,D|*2*pi;
x(t):=r*(t-sin(t),1-cos(t));
A.xy=[x(pi)-1,0.5*x(pi)-2];
B.xy=x(t);
linesize(3.5);
plot(x(t),t,start->0*pi,stop->pi,step->400,
      color->(1,0,0));

```

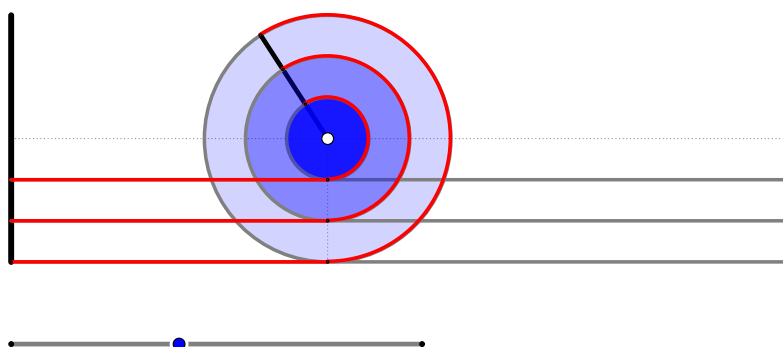
Kao što smo već napomenuli, stari Grci nisu znali za cikloidu. Ali je zato kotrljanje kruga po ravnoj podlozi i kod njih izazivala



nedoumice i dovodila do paradoksalnih zaključaka. Najpoznatiji je svakako onaj koji se pojavljuje u Aristotelovoj Mehanici.

Primer 26: Aristotelov paradoks točka

Razmotrimo točak sa tri cilindra centralno zalepljena osnovu za osnovu. Ako takav točak kotrljamo duž šine sa tri različite visine prilagođene navedenom točku, "zaključujemo" da kružnice različitog dijametra opisuju jednake putanje za jedan pun obrtaj. Do takvog zaključka su došli razmatranjem bijektivnog preslikavanja tačaka na kružnicama i pravolinijskim putanjama. Naravno, kao što danas znamo, ti skupovi su ekvipotentni. Šta je fizičko objašnjenje ovog paradoksa?



Slika 8.3: Aristotelov paradoks

Naravno, izostala je precizna definicija kotrljanja. Ono može da bude sa i/ili bez proklizavanja. Uslov kotrljanja bez proklizavanja je jednakost brzina dodirnih tačaka točka i podloge. S'obzirom da

podloga miruje, a tačke na točku se kreću istovremeno - rotaciono oko ose točka, sa ugaonom brzinom ω i translaciono zajedno sa osom točka, brzinom v_t . Kod tačke koja u datom trenutku dodiruje podlogu, tj. trenutne ose rotacije, ove dve brzine se anuliraju, $v_t = \omega r$. Dakle, samo jedan od prikazanih diskova se kotrlja bez proklizavanja.

Geometrijski elementi su tri koncentrične kružnice i tri paralelne prave. Tačka dodira je definisana normalom (n) iz centra kružnica (A) na paralelne prave. Nacrtna je i dodatna duž - zajednički poluprečnik, koja spaja centar A sa tačkom M na periferiji najveće kružnice. Konstruisan je i klizač (EF) koji definiše trenutni položaj centra A i tačke M pomoću skripta:

```

\\ u Draw slotu
r=C0.radius;
t=|E,K|/|E,F|*2*pi;
A.x=r*t;\\ trenutni položaj tačke A
x(t):=r*(t-sin(t),1-cos(t));
M.xy=x(t);\\ trenutni položaj tačke M

```

Crvena boja putanja je dobijena doctavanjem crveno obojenih duži od levog kraja puta do preseka sa normalom n , i lukova od preseka sa n , do radijusa (AM).

Sve cikloide su slične, jer se mogu dobiti iz jedne jedine homotetijom. Međutim, kotrljanjem jedne kružnice duž druge, moguće je dobiti beskonačno (kontinuum) mnogo različitih krivih koje se jednim imenom nazivaju centrirane trohoidne. Ako je kriva nastala kotrljanjem manje oko spoljašnje strane veće kružnice, onda se ona naziva epicikloida, a ako je sa unutrašnje strane - hipocikloida. U mnogim primerima uslov kotrljanja nije ispunjen ili nije ni potreban. Tada je dovoljno opisati familiju krivih nastalih jednostavnom superpozicijom dva ili više kružnih kretanja. Takve krive, kod kojih se tačka rotira oko kružnice (epicikla) čiji se centar rotira oko druge kružnice itd. se nazivaju epicikličnim krivama. Sve centrirane trohoidne krive, generisane kotrljanjem jedne kružnice oko ili unutar druge su specijalni slučaj epicikličnih krivih.

Definicija 1: Epiciklična kriva

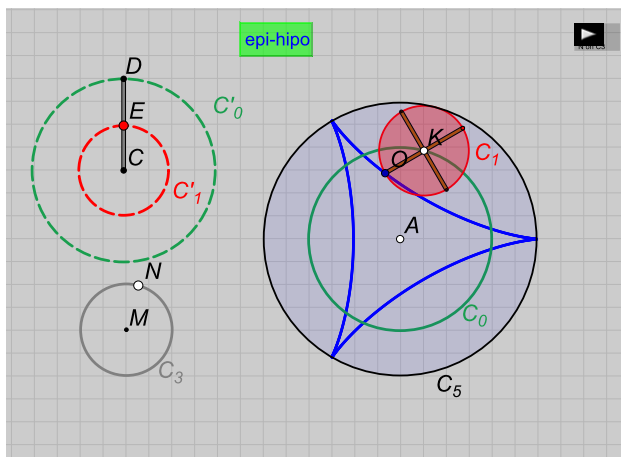
Neka je dat sistem od n kružnica c_1, c_2, \dots, c_n , tako da se centar i -te nalazi na periferiji $i - 1$ -ve (centar prve kružnice je proizvoljan, npr. u koordinatnom početku). Neka svaka od kružnica istovremeno rotira ugaonim brzinama $\omega_1, \omega_2, \dots, \omega_n$. Tada kriva koju opisuje proizvoljna tačka na periferiji c_n nazivamo *epicikličnom krivom*.

U praksi je epiciklične krive obično najjednostavnije generisati pomoću zglobnog mehanizma sa n segmenta, tako da je svaka kružnica c_i iz definicije epiciklične krive, generisana rotiranjem odgovarajućeg segmenta mehanizma. Drugi standardni uređaj koji se koristi za crtanje centriranih trohoida i koji je danas poznatiji kao igračka, se naziva *spirograf*.

Primer 27: Spirograf

Generisati nekoliko primera (hipo)epicikloida pomoću sistema kružnica (epiciklova), drugim rečima, napraviti animaciju spirografa.

Najjednostavniji način generisanja epi i hipocikloida je kao specijalni slučaj epicikličke krive, sa odnosom radijusa i ugaonih brzina takvih da ispunjavaju uslov odsustva proklizavanja između epicikla i jedne od fiktivnih kružnica radijusa $r_1 = R + r$ ili $r_2 = R - r$.

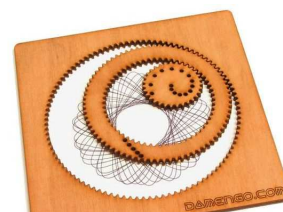


```
//podesavanje radijusa
R=C0.radius;r=|E,C|; //za C0 i C1 kruznicu
C4.radius=R-r;
C5.radius=R+r;
//podesavanje ugaone brzine manje kruznice - epicikla
// da bi bio ispunjen uslov proklizavanja
om=round(R/r);

//podesavanje uslova sta ce biti prikazano epi ili hipo cikloida
if(Text0.pressed,C4.visible=true;C5.visible=false,
C4.visible=false;C5.visible=true);

//jednacina krive
x(t):=(r*cos(om*t)+R*cos(t),r*sin(om*t)+R*sin(t));

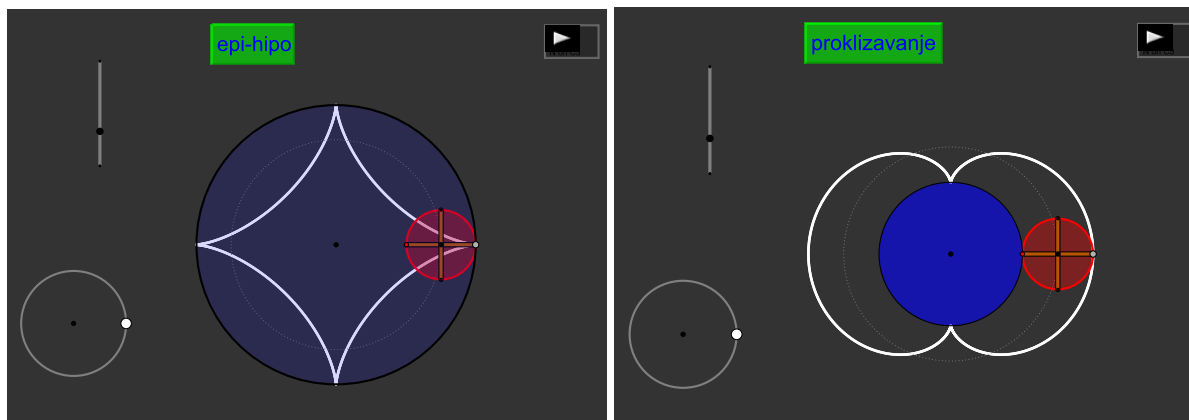
//polozaj tacaka K i L diktira polozaj tacke N
t=N.angle;
K.angle=t; //ili K.xy=R*(cos(t),sin(t));
O.xy=A.xy+x(t+pi);
```



Slika 8.4: Spirograf

Osnova konstrukcije su kružnice C_0 -*deferent* i C_1 -*epicikl*. Radijus deferenta označen sa R je fiksiran, a radijus epicikla r je jednak dužini duži CE i kontroliše se položajem tačke E . Pomoćna kružnica je C_5 sa radiusom $r_1 = R + r$. Tražena kriva je putanja tačke O . Položaj tačke O je definisan (vidi skript) trenutnim položajem kontrolne tačke N , a da bi iluzija kotrljanja bila potpuna u položaj tačke K , centra epicikla je takođe definisan položajem kontrolne tačke. Ostali detalji su čitljivi iz skripta.

```
//crtanje epi-hipocikloide
linesize(3);
plot(x(t)+A.xy,t,start->0*pi,stop->2*pi,steps->1000,color->(0,0,1));
```



Slika 8.5: Konačni izgled animacije: hipo i epicikloida kao epiciklične krive.

Generalizacija prethodne konstrukcije je epiciklični mehanizam. Epiciklični mehanizam je sistem kružnica koji generiše epicikličnu krivu i jednoznačno je definisan epicikličnim sistemom. Epiciklični sistem je funkcija $g : \mathbb{R} \mapsto \mathbb{R}^2$ koja svakoj ugaonoj brzini ω pridružuje uređeni par (r, ϕ) - radijus i inicijalni ugao kružnice mehanizma, koja rotira ugaonom brzinom ω . Jednoznačnost pridruživanja mehanizam-sistem je posledica sledeće dve teoreme. Dakle, krivu definiše skup ugaonih brzina zajedno sa njima odgovarajućim poluprečnicima i početnim položajima (ϕ). Međutim, nije naveden redosled - koja kružnica na kojoj rotira. Opravdanost definicije postaje jasna nakon sledećeg zadatka.

Primer 28: Osnovna teorema epicikličnog kretanja 1

Dokazati i demonstrirati da generisana epiciklična kriva zavisi samo od relativnog odnosa dužina i ugaonih brzina segmenata koji je generišu, ali ne i od njihovog redosleda.

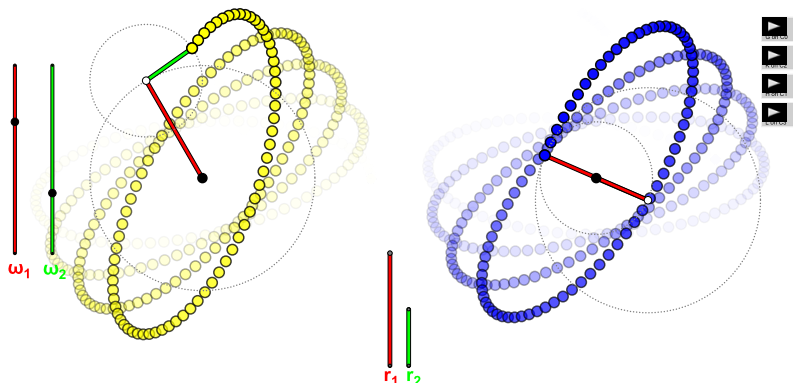
Dokaz sledi iz komutativnosti sabiranja kompleksnih brojeva.

Primer 29: Osnovna teorema epicikličnog kretanja 2

Dokazati i demonstrirati da ako dve kružnice mehanizma rotiraju istom brzinom tada ih je moguće zameniti jednom kružnicom sa različitim radijusom i uglom.

Stav je očigledan iz eksponencijalne forme kompleksnih brojeva:

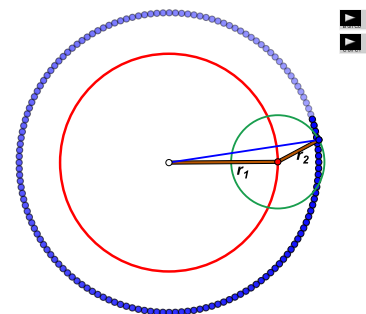
$$r_1 e^{i(\omega t + \phi_1)} + r_2 e^{i(\omega t + \phi_2)} = e^{i\omega t} (r_1 e^{i\phi_1} + r_2 e^{i\phi_2}) = e^{i\omega t} r e^{i\phi}.$$



Slika 8.6: Nezavisnost generisane epiklične krive od redosleda segmenata (epiciklova).

Sve ovo možemo ilustrovati interaktivnom simulacijom - animiranim slobodnim tačkama koje rotiraju duž periferije kružnice. Na slici vidimo da ako poluprečnik r_1 i r_2 rotiraju istom brzinom, tada je i ugao između njih jednak, pa trougao koji oni obrazuju rotira kao celina. Novu kružnicu opisuje treća stranica trougla.

Pitanje: koju putanju će opisivati zelena tačka ako isključimo rotaciju manje kružnice? Kružnicu podudarnu sa većom samo transliranu za vektor r_2 .



Slika 8.7: Ilustracija druge osnovne teoreme epikličnog kretanja

Epiklične krive su prirodan uvod u Furijeovu analizu.

Definicija 2: Furijeov transform

Ravanska kriva je funkcija $f : \mathbb{R} \mapsto \mathbb{R}^2$. Operacija koja krivoj f pridružuje epiklični sistem g , i na taj način je predstavlja kao epikličnu krivu, naziva se Furijeov transform.

Osnovni stav Furijeove analize tvrdi da je svaku neprekidnu krivu moguće predstaviti kao kao epikličnu krivu. Na primer, konturu lica Homera Simpsona je moguće rekonstruisati pomoću dovoljnog broja Furijeovih sabiraka. To je i učinjeno, a animaciju je moguće pogledati na sajtu <https://www.youtube.com/watch?v=QVuU2YCwHjw>. Desno je prikazana jedna sekvenca iz animacije.

Kada je kriva periodična, tada je moguće predstaviti je epikličnim sistemom sa kružnicama koje rotiraju celobrojnim brzinama (Furijeov red). Furijeov red generišemo na sledeći način. Sabirci

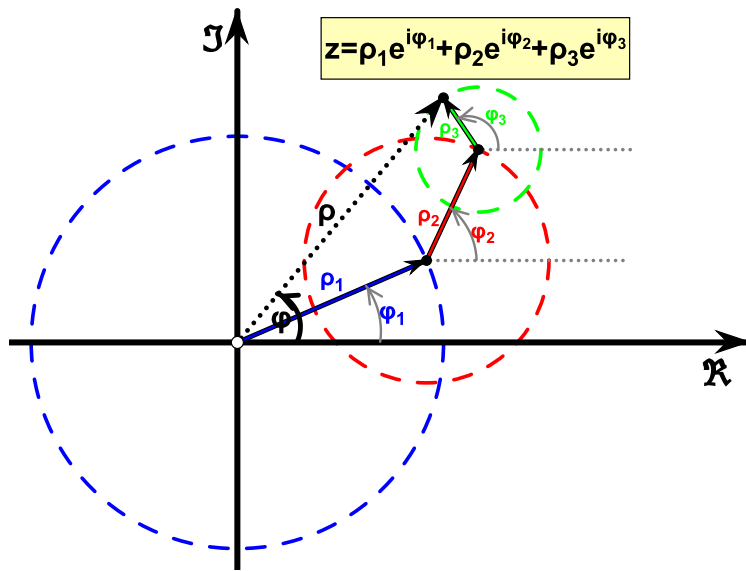
$$\sum_{n=1}^N a_n e^{int}$$

su dobijaju nadovezivanjem kružnica u kompleksnoj ravni.

Ako kompleksne brojeve predstavimo kao vektore u ravni, tada je zbir $z = z_1 + z_2 + z_3$ rezultat nadovezivanja vektora. Pored toga, kompleksne brojeve je moguće predstaviti u eksponencijalnom obliku, pomoću polarnih koordinata, pa je tada zbir $z = \rho_1 e^{i\phi_1} + \rho_2 e^{i\phi_2} + \rho_3 e^{i\phi_3}$. Parametri ρ_i su tada poluprečnici kružnica oko kojih rotira vektor z_i , $|z_i| = \rho_i$, a uglovi ϕ_i određuju položaj na kružnici. Konstrukcija je prikazana na



slici dole.



Slika 8.8: Zbir kompleksnih brojeva konstruisan pomoću epiciklova.

Ako još dozvolimo pojedinim vektorima da rotiraju sa određenim ugaonim brzinama ω_i , tada je zbir u nekom trenutku t jednak

$$z(t) = \rho_1 e^{i\omega_1 t} + \rho_2 e^{i\omega_2 t} + \rho_3 e^{i\omega_3 t}.$$

Projekcija na \Re -osu daje realne delove, tj. kosinusni Furijeov red, a na \Im -osu, imaginarne delove, tj. sinusni Furijeov red.

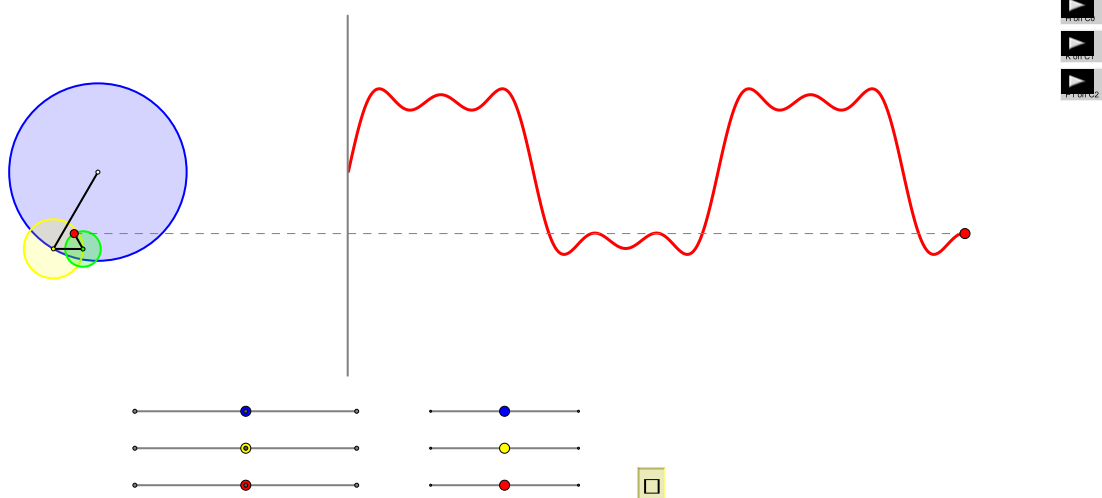
Primer 30: Furijeov red pravougaonog impulsa

Konstruisati epiciklični model koji generiše parnu pravougaono - testerastu funkciju (talasni niz). Prikazati samo prva tri člana reda.

Rešenje: $f(t) = \sum_{n=1}^3 \frac{\cos(2n-1)}{(2n-1)^2}$. Konstrukcija je prikazana na slici 8.9.

Radijusi kružnica su $\rho_n = 1/(2n-1)^2$, a odgovarajuće ugaone brzine, tj. brzine animacije su $\omega_n = 2n-1$. Prateći skript za ovaj i naredni primer je:

```
//u Draw slotu
C0.radius=6*1*|E,U|/|E,P2|;
C1.radius=6*if(Text1.pressed,(1/3),(1/9))*|A,V|/|A,P3|;
C2.radius=6*if(Text1.pressed,(1/5),(1/25))*|C,W|/|C,P4|;
Anim0.speed=1*|L,X|/|L,R|;
Anim1.speed=3*|N,Y|/|N,S|;
Anim2.speed=5*|P,Z|/|P,T|;
```



Slika 8.9: Zbir prva tri člana Furijeovog reda pravougaonog talasnog niza.

```

moveto(P0, [P5.x+25*simulationtime(), P5.y]);
//u TimerTick slotu
l=l+[[P5.x+25*simulationtime(), P5.y]];
connect[l, size->3, color->[1,0,0]];
// u SimulationStart slotu
l=[];

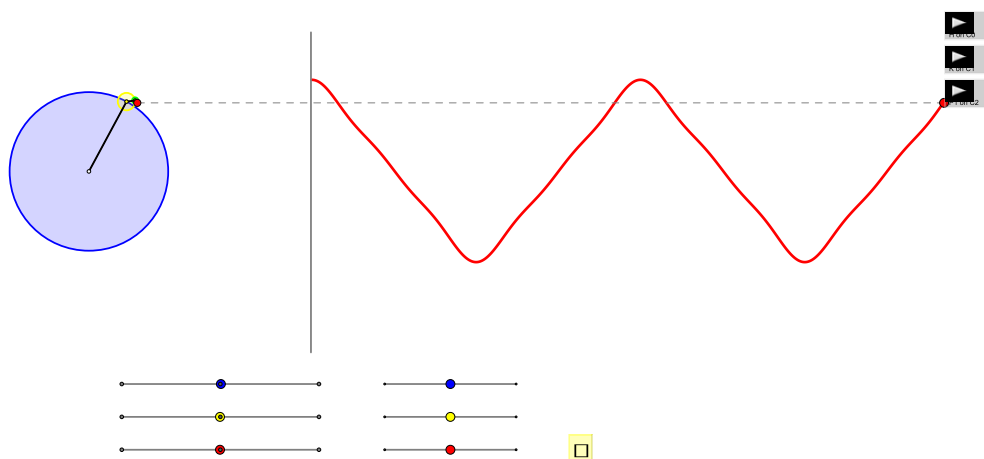
```

Geometrijski elementi su 6 klizača, tri kružnice sa tri slobodne periferijske tačke. Vodeća tačka ($P5$) se nalazi na periferiji najmanje kružnice, a tačka $P0$ je slobodna i njen zadatak je da iscrtava konačni grafik trougaonog talasnog niza. Naravno, nacrtane su i koordinatne ose.

Primer 31: Furijeov red trougaonog impulsa

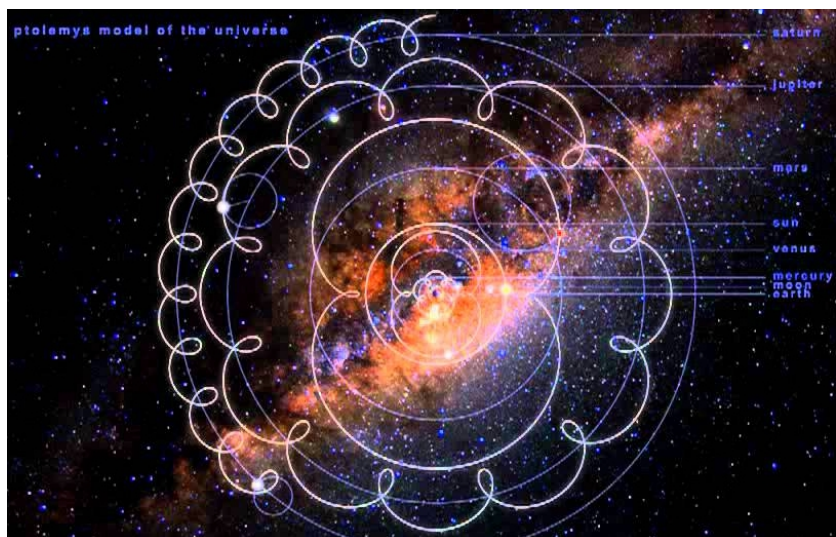
Konstruisati epiciklični model koji generiše neparnu trougaonu funkciju. Prikazati samo prva tričlana reda.

Rešenje: $f(t) = \sum_{n=1}^3 \frac{\sin(2n-1)}{(2n-1)}$. Konstrukcija je prikazana na slici 8.10. U čemu se sastoji razlika između ove i prethode animacije?

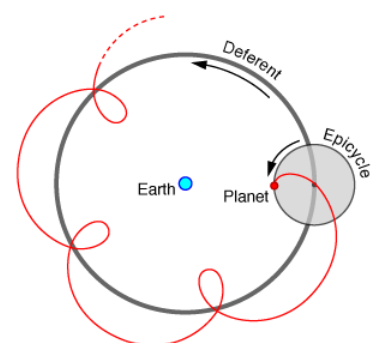


Slika 8.10: Zbir prva tri člana Furijeovog reda pravougaonog talasnog niza.

Epiciklične krive su koristili starogrački astronomi za opis putanja planeta. Na taj način su uspjeli da objasne retrogradno kretanje pojedinih planeta (tzv. spoljašnjih planeta, tj. onih daljih od Zemlje). Sistem planetarne astronomije starog veka - *geocentrični model*, je najpotpunije izložen u delu Klaudija Ptolomeja, *Almagest*.



Slika 8.11: Ptolomejev planetarni sistem.



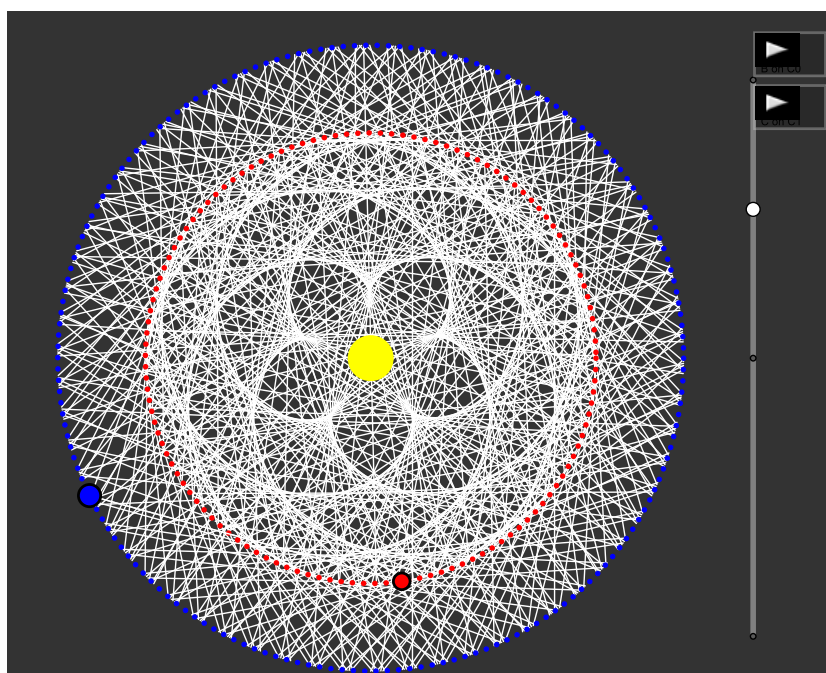
Slika 8.12: Definicija epicikla i deferenta.

U Ptolomejevom sistemu astronomije, model epicikla (u prevodu sa Srarogrčkog "na kružnici") je korišćen da bi objasnio varijaciju u brzini i pravcu prividnog kretanja planeta. Pre svega, sistem je uspeo da objasni prividno retrogradno kretanje pet planeta poznatih u to vreme. Takođe, uspeo je i da objasni promene u prividnim rastojanjima planeta od Zemlje.

U odnosu na savremeni, heliocentrični model, Ptolomejev deluje znatno komplikovanije i "neočigledno." Međutim, na primeru kretanja Venere, odnosno sistema Zemlja-Venera možemo očigledno predstaviti promenu međusobnog položaja Zemlje i Venere i na osnovu njega se uveriti, da u ovom slučaju, oba modela daju ekvivalentne rezultate.

Primer 32: Zemlja - Venera sistem

Animirati kretanje Zemlje i Venere, pod pretpostavkom da se kreću po kružnim putanjama, da je radijus Venerine orbite $r_V = 0.72$ a.j. Osam Zemaljskih godina je približno jednako trinaest Venerinih godina. Međusobni položaj prikazati pomoću duži koja spaja planete.



Slika 8.13: Obvojnica Venerinog pentagrama (Zemlja - plavo, Venera - crveno, Sunce - Žuto).

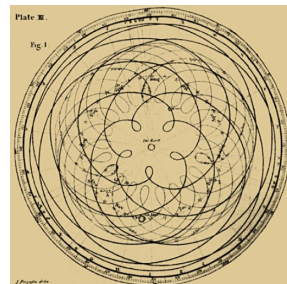
Sistem i obvojnica generisana kretanjem duži relativnog položaja je prikazana na slici 8.13. Geometrijski elementi su plava (C0) kružnica sa slobodnom tačkom na periferiji B, koja predstavlja Zemlju; manja, crvena kružnica (C1) sa slobodnom periferijskom tačkom - Venera; zajednički centar (A) - Sunce i belo obojena duž (AB). Kretanje obe tačke je animirano. Prateći skript za konstrukciju je:

```
//u Draw slotu
C1.radius=0.72*C0.radius;
Anim0.speed=-1;
Anim1.speed=(13/8)*Anim0.speed;
```

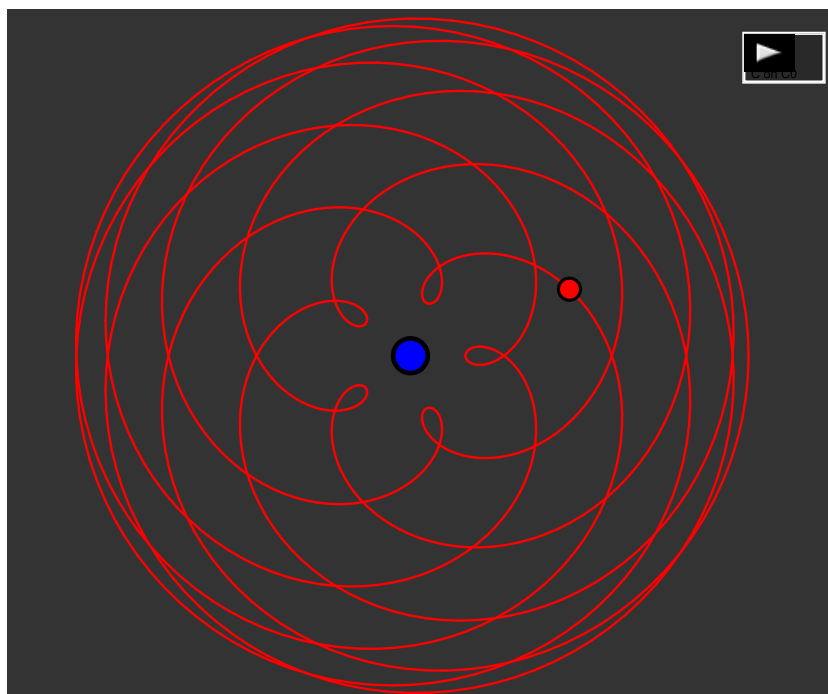
```
//u TimerTick slotu
l=l+[[B.xy,C.xy]];
drawall(l,size->1,color->(1,1,1));
// u SimulationStart slotu
l=[];
// u SimulationStop slotu
C.x=B.x;
```

Obratiti pažnju na obrazac sa 5-tostrukom simetrijom, tzv. "Venerin pentagram" (The Pentagram of Venus). On je bio poznat još u starom veku, a na slici desno je prikazana reprodukcija iz knjiga Džejmisa Fergusona (James Ferguson) *Astronomy Explained Upon Sir Isaac Newton's Principles* iz 1799. godina.

Evo kako zapravo izgleda putanja Venere viđena sa Zemlje za period od osam Zemaljskih godina. Za simulaciju su potrebne



Slika 8.14: Venerin pentagram.



dve slobodne tačke (A , D) i jedna jedinična kružnica sa slobodnom tačkom na periferiji (C). Prateći skript je:

```
//draw slot
plot(gauss(complex(A.xy)+5*(1*(exp(i*t*8))+0.72*(exp(i*t*13)))),
start->0,stop->2*pi,steps->1000,color->(1,0,0),size->1.5);
tt=C.angle;
D.xy=gauss(complex(A.xy)+5*(1*(exp(-i*tt*8))+0.72*(exp(-i*tt*13))));
```

Za 8 godina Venera pride Zemlji na minimalno rastojanje $13 - 8 = 5$ puta. Svaki lobus pentagrama je za $8 - 5 = 3$ koraka ispred prethodnog. Tačnije vrednosti za Zemaljsku i Venerinu godinu su, 365.256 za Zemlju i 224.701 za Veneru. Odnos je $224.701/365.256 \approx 0.615187$, dok je $8/13 \approx 0.615387$, što je zaista veoma blizu! Dobijeni količnik je dobra aproksimacija za zlatni presek $\phi = (\sqrt{5} - 1)/2$, što je očekivano, jer su 8 i 13 uzastopni elementi Fibonačijevog niza, a on je u limesu jednak tačno ϕ .

Inače, Venerin pentagram se pominje i u popularnom romanu *Da Vinčijev kod*. Ali čak ni tamo nije napomenuta dodatna numerološka koincidencija da je latinski naziv za Veneru bio Lucifer, u doslovnom prevodu: donosilac zore (nosilac svetla) tj. jutarnja zvezda - Venera.

Čak su napravljani i mehanizmi koji su omogućavali da se predvidi položaj planeta u budućnosti. Verovatno je najpoznatiji Antiki-tera mehanizam koji datira iz 80 godina PNE.

Slični mehanizmi se koriste i danas u tehnicima kao epiciklični diferencijal, ali je princip rada bilo kojeg diferencijala isti. Isti princip je iskorišćen i kod kugličnih ležajeva.

Primer 33: Epiciklični diferencijal

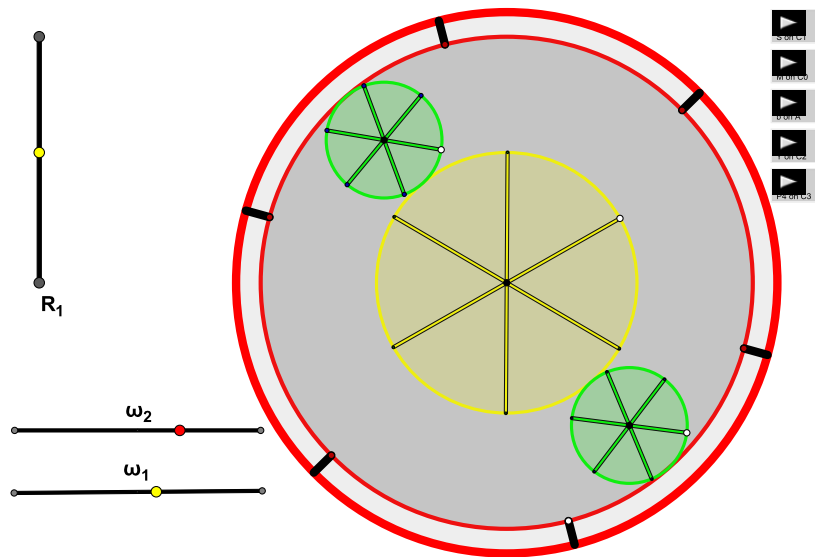
Animirati rad epicikličnog diferencijala sa dva planetarna zupčanika.

Na slici je prikazan mehanizam epicikličnog diferencijala. Žutom bojom je označen tzv. sunčani zupčanik, zelenom, planetarni, a crvenom, prstenasti zupčanik. Obično se koristi za asimetričnu raspodelu angularnog momenta. To se postiže tako što se direktno pokreću planetarni zupčanici, a oni asimetrično prenose moment na prstenasti (veći moment) i sunčani (manji moment). Takođe, u zavisnosti od opterećenja prstenastog ili sunčanog zupčanika, kretanje se prenosi većinom na manje opterećen. Kuglični ležajevi koriste isti princip samo u inverznom pravcu: tamo prstenasti i sunčani cilindar generišu kretanje, a planetarni svojim kretanjem obezbeđuju da nema trenja klizanja, već samo trenja kotrljanja, koje je znatno manje. Prateći skript:

```
//draw slot
r1=|C,D|; r2=C0.radius; r3=r2-r1;C9.radius=9;
om1=Anim0.speed=(-1+|P14,P17|/|P14,P16|)*4;
om2=Anim1.speed=(-1+|P10,P13|/|P10,P12|)*4;
omx=Anim2.speed=(om1*r1+om2*r2)/(r1+r2);
omy=Anim3.speed=(om2*r2-om1*r1)/(r2-r1);
Anim4.speed=(om2*r2-om1*r1)/(r2-r1);
```

Na osnovu KAM teorije, stabilne orbite planeta bi trebalo da budu u odnosima koji nisu jednaki količnicima malih celih brojeva.

u upotrebi na Tojoti Prijus



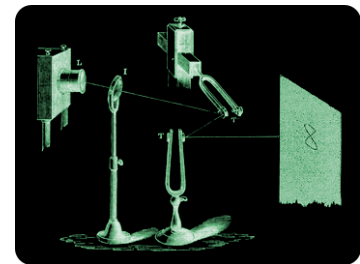
Slika 8.15: Epiciklični diferencijal.

Uslov neproklizavanja u tačkama dodira planetarnih i sunčanog (i prstenastog) točka nameće da je (linijska) brzina centra planetarnog točka jednaka aritmetičkoj sredini brzina tačaka na periferiji sunčanog i prstenastog, dok je ugaona brzina planetarnog točka kompenzuje razliku brzina centra i periferije.

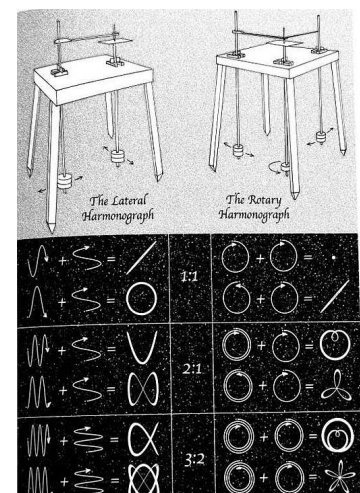
Zanimljiva je veza epicikličnih krivih sa Lisažuovim figurama. I jedne i druge su dobijene sabiranjem dva harmonijska kretanja. U slučaju epiciklova, sabiraju se direktno dva (ili više) kružna kretanja, a u slučaju Lisažuovih figura, harmonijska kretanja (tj. komponente kružnih kretanja) duž dve ortogonalne ose. Drugim rečima, postoji dualizam između ove dve klase krivih.

Lisažu je svoje krive dobio pokušavajući da vizuelizuje zvučnu harmoniju, tj. sazvučje dve zvučne viljuške.

Kasnije je konstruisan mehanički uređaj - harmonograf, koji iscrtava Lisažuove krive jednostavnim sabiranjem dva harmonijska, međusobno ortogonalna oscilovanja. Ali pored tzv. lateralne verzije, postoji i rotaciona verzija, koja umesto oscilacija u ravni, sabira kružne oscilacije, tj. rotacije. Inače, savremeni uređaj za crtanje Lisažuovih figura je osciloskop.



Slika 8.16: Lisažuov ogled.



Slika 8.17: Harmonografi

Primer 34: Lisažu vs. epicikli

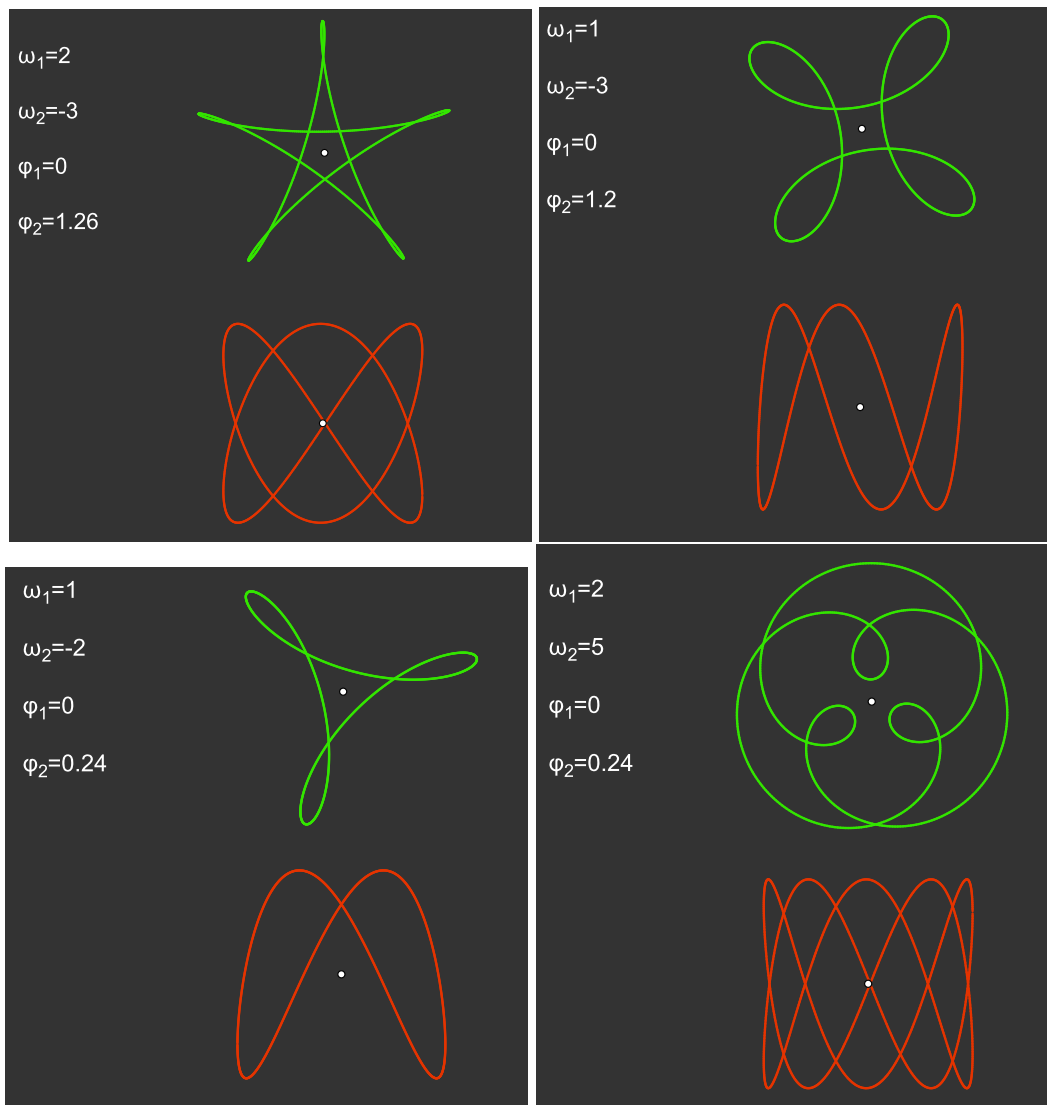
Prikazati Lisažuove verzije nakoliko epicikličnih krivih.

Epiciklične predstaviti kao kompleksni polarni plot

$$t \rightarrow r_1 * (\exp(i * (t * n_1 + \phi_1))) + r_2 * (\exp(i * (t * n_2)) + \phi_2),$$

a Lisažuove kao realni polarni plot od

$$t \rightarrow (r_1 * \cos(n_1 * t + \phi_1), r_2 * \sin(n_2 * t + \phi_2)).$$



Slika 8.18: Veza Lisažuovih sa epicikličnim krivama

Prateći skript:

```
//u Draw slotu
n=10; n1=round(n*|E,S|)-2*n; n2=round(n*|G,T|)-2*n;
d1=pi*|K,U|/|K,L|; d2=pi*|Y,P0|/|Y,Z|; r=|B,D|/2;
r1=round(|M,V|*2)/2; r2=round(|O,W|*2)/2;

drawtext((-7,6), "$\omega_1$="+n1,size->20,color->[1,1,1]);
drawtext((-7,5), "$\omega_2$="+n2,size->20,color->[1,1,1]);
drawtext((-7,4), "$\phi_1$="+format(d1,2),size->20,color->[1,1,1]);
drawtext((-7,3), "$\phi_2$="+format(d2,2),size->20,color->[1,1,1]);
drawtext((9,6), "$r_1$="+format(r1,2),size->20,color->[1,1,1]);
drawtext((9,5), "$r_2$="+format(r2,2),size->20,color->[1,1,1]);
drawtext((9,2), "$r$="+format(r,2),size->20,color->[1,1,1]);

lis(t)=1.3*r*(r1*cos(n1*t-d1*pi), r2*sin(n2*t-d2*pi));

plot(gauss(complex(A.xy)+r*r1*(exp(i*(t*n1+d1*pi)))+r2*(exp(i*(t*n2+d2*pi)))),
      t,start->0,stop->2*pi,steps->1000,color->[0.2,0.9,0],size->2);

plot(Q.xy+lis(t), t,start->-pi,stop->pi,steps->1000,color->[0.9,0.2,0],size->2);
```

Postoji još jedna, ekvivalentna familija krivih strukturalno ekvivalentna epicikličnim i Lisažuovim krivama, a to su putanje tačke u tzv. kvadratnim bilijarima. O tome videti u poglavlju knjige posvećenom optici i matematičkim bilijarima.

Klasa ovih krivih se u matematici nazivaju torusnim omotačima

9 Složena kretanja: traktrise i neholonomni sistemi

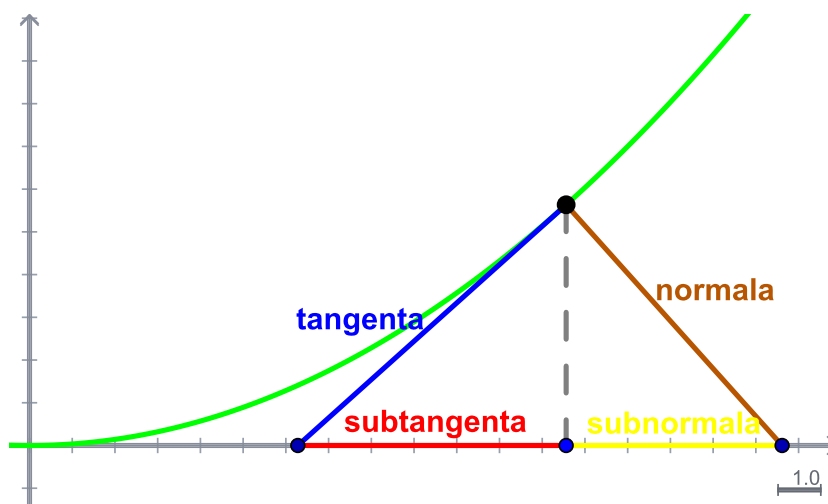
U ovoj glavi se i dalje bavimo analizom krivih, ali iz perspektive teoreme o brišućoj tangenti. Ova elegantna teorema veoma podseća na matematiku XVII veka (vidi prethodnu glavu i delove posvećene proučavanju cikloide), ali su zapravo tekovina XX veka. Teorema potiče iz 50-ih godina i njen autor je Jermski matematičar i seizmolog Mamikon Mnatsakanian. Teorema je direktno primenljiva na čitavu klasu krivih koje se obično nazivaju traktrisama, čiji je najpoznatiji predstavnik trag generisan točkom bicikla. Zapravo, ogromna klasa neholonomnih sistema generiše traktrise i o tome će biti reči u ovoj glavi.

Pre same formulacije teoreme, definišmo i prikazimo na jednoj slici potrebne pojmove iz analize krivih. Svi oni su bili poznati početkom XVII veka i sledeća praktična konstrukcija se pripisuje Paskalu.

Paskalov infinitezimalni trougao

Nekoliko osnovnih geometrijskih pojmova na kojima je utemeljen diferencijalni račun je moguće elegantno prikazati i lako zapamtiti na tzv. Paskalovom infinitezimalnom trouglu koji je ilustrovan na slici.

Ovo je manje poznati Paskalov trougao. Poznatiji se odnosi na binomne koeficijente.



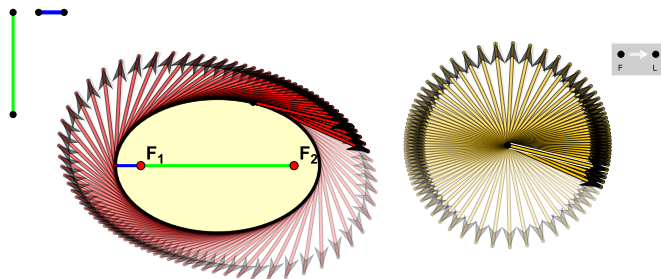
Slika 9.1: Paskalov infinitezimalni trougao. Standardne skraćenice za dužine odgovarajućih duži su: tangenta - $|t|$, normala - $|n|$, subtangenta - $|st|$, subnormala - $|sn|$

Definišimo sada pojmove neophodne za samu teoremu o brišući tangenti.

Primer 35: Osnovna teorema

Teorema o brišući tangenti glasi: *tangentni prebrisak (sweep) ima jednaku površinu tangentnom klasteru (cluster)*. Tangentni prebrisak je površina koju prebriše tangenta čija se početna tačka kreće duž krive, a tangentni klaster prebriše slika tangente kojoj je početak fiksiran. Napraviti interaktivnu konstrukciju tangentnog prebriska i klastera na primeru elipse.

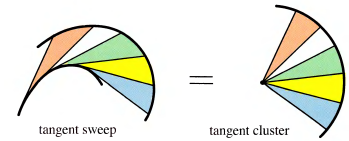
Konstrukcija je generisana prostom translacijom tangente sa date tačke na periferiji elipse na fiksiranu tačku izvan elipse. Tangenta je konstruisana koristeći optičko svojstvo elipse (videti primer u glavi posvećenoj statičkim konfiguracijama). Rastojanje između fokusa elipse i između fokusa i perihela je kontrolisano klizačima. Animacija je generisana kretanjem slobodne tačke duž periferije elipse, a podešeno je da tangentna duž i njena translirana slika, iscrtavaju trag.



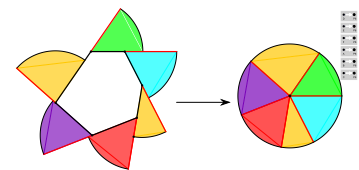
Primer 36

Dokazati teoremu o brišući tangenti.

Teorema je posledica činjenice da se kretanje tangente pri pomeranju tekuće tačke na krivoj, može razložiti na kompoziciju (naizmeničnih infinitezimalnih) rotacija i translacija. Prebrisana površina prilikom translacije je nulta, pa je ukupna prebrisana površina generisana rotacijom. Ovaj zaključak je očigledan ako se najpre razmotri analogna konstrukcija na mnogouglu i primeti da on važi nezavisno od broja stranica n , pa i pri $n \rightarrow \infty$.



Slika 9.2: Ilustracija teoreme o brišući tangenti.



Slika 9.3: Ilustracija teoreme o brišući tangenti na mnogouglu. Pomeranjem temena mnogougla se ne menja konačna kružnica, već se samo preraspodeljuje površina po sektorima.

Primer 37

Primenom teorema o brišućoj tangenti odrediti površinu ispod eksponencijalne krive, tj. izračunati integral $\int_{-\infty}^x e^{t/b} = be^{t/b}$.

Ilustrovati primenu metoda tako da se pomeranjem tačke x automatski pojavljuje njoj odgovarajuća tangenta kao i njena translirana slika (kao na slici desno).

Osnovna osobina eksponencijalne funkcije (i odg. krive) je da ima konstantnu subtangentu, za funkciju oblika $e^{x/b}$ njena dužina je $|st(x)| = b$ (jer je $f(x)/|st(x)| = (f(x))' = f(x)/b$). Sa slike se vidi da je površina prebrisana tangentom u intervalu $(-\infty, x]$ jednaka površini trougla iznad subtangente u početnom trenutku, pa je ukupna površina ispod eksponencijalne krive jednaka dvostrukoj površini trougla, tj. $be^{x/b}$.

Konstrukcija je generisana translacijom tangente, a sama tangenta i grafik pomoću sledećeg skripta:

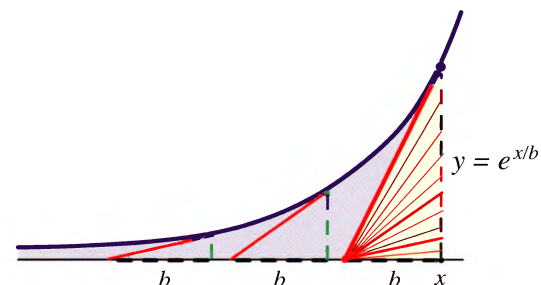
```
//u Draw slotu
//def. granicne tacke (gornje granica)
If(C.x>=F.x,C.x=F.x);

//def. funkcije i prateceg teksta;
b=|A,B|*2;
f(x):=exp(x/b);
drawtext([-1.5,10.5], "$b=$"+|A,B|, color->[0,1,0],
size->18);
drawtext([2,12], "$f(x)=e^{\left(\frac{x}{b}\right)}$",
color->[0,1,0], size->22);

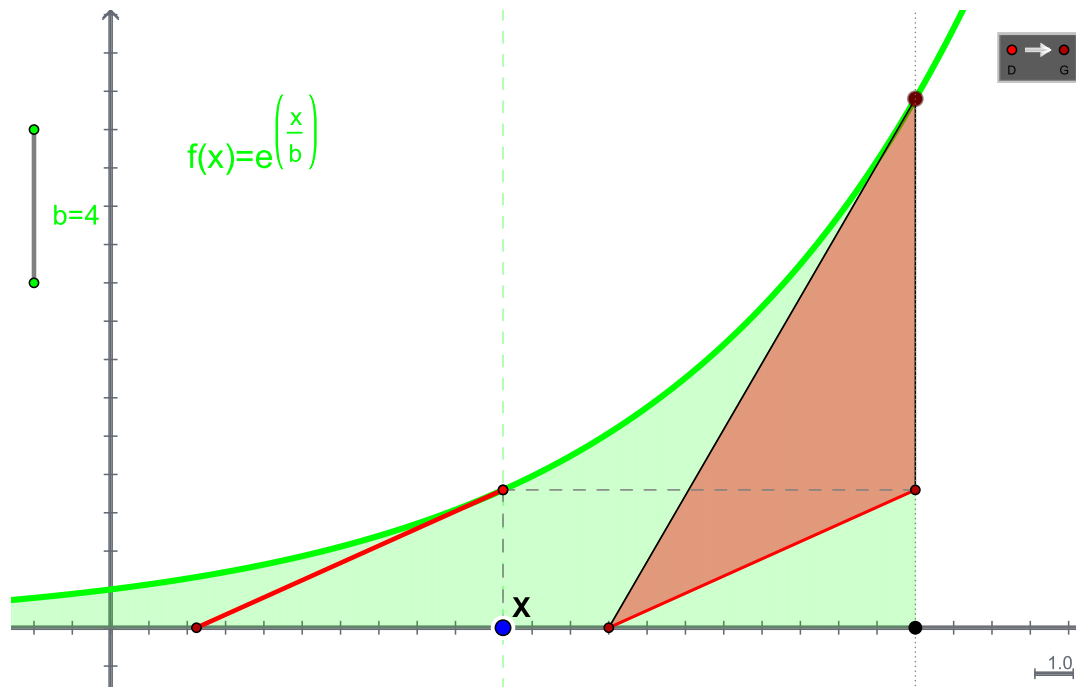
//crtanje pomocne duzi koja ogranicava prebrisani trougao;
a1=[F.x, f(F.x)]; b1=H.xy;
createpoint("Y", a1);
draw([a1, b1], color->[0.2,0.2,0.2]);

// crtanje grafika funkcije i tacaka na grafiku;
plot(f(x), size->4, color->[0,1,0]);
fillplot(f(x), start->-10, stop->F.x,
size->1, color->[0,1,0], alpha->0.2);
draw([F.x, f(F.x)], size->4, color->[0,0,0]);
draw([C.x, f(C.x)], size->3, color->[0,0,0]);

//izracunavanje tangente;
t=tangent(f(#), C.x);
```



```
//pomeranje prave d kroz ta\ v cku D tako da prolazi kroz teku\ ' cu
ta\ v cku na grafiku i ima nagib tangente;
moveto(D,[C.x,f(C.x)]);
d.slope=d(f(#),C.x);
```



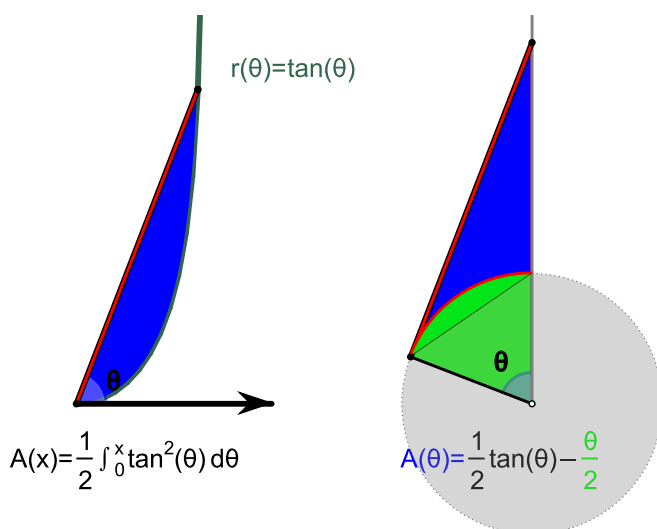
Geometrijski elementi konstrukcije su horizontalna (c , u ulozi x -ose), i vertikalna ($b \perp c$) prava koje se seku u tački F . Pokretna tačka C na osi c koja je u konačnoj konstrukciji označena sa X ; tangenta - duž $d = ED$. Na kraju, definisana je translacija sa $D \rightarrow G$, gde je tačka G dobijena u preseku prave paralelne sa x -osom i verikale b . Koeficijent eksponente je regulisan dužinom klizača $a = AB$.

Primer 38

Primenom teorema o brišućoj tangenti odrediti vrednost integrala

$$A(x) = \frac{1}{2} \int_0^x \operatorname{tg}^2(\theta) d\theta.$$

Pokazaćemo da je $A(x) = \frac{1}{2} \operatorname{tg}(x) - \frac{x}{2}$. Na slici 9.4 je na levoj strani prikazan tangentni klaster obojen plavom bojom, a na desnoj, odgovarajući tangentni prebrisak. Tangentni klaster odgovara šrafiranoj površini između polarnog grafika funkcije $r(\theta) = \operatorname{tg}(\theta)$ i finalnog radijus vektora $r(x)$. Tangentni prebrisak je razlika površine pravouglog trougla sa stranicama 1 i $\operatorname{tg}(x)$ ($P = 1/2 \operatorname{tg}(x)$) i površine kružnog sektora (zeleno šrafiran) sa uglom x ($P = 1/2x$).



Slika 9.4: Finalna konstrukcija brišuce tangente. Na slici i u skriptu je korišćena TeX-ovska oznaka $\tan \equiv \operatorname{tg}$.

Skript uz konstrukciju je sledeći:

```
//u Draw slotu
f(t):=[cos(t),sin(t)]*tan(t);
plot(f(t),t,start->0,stop->pi/2,color->[0.2,0.4,0.3],size->4);

drawtext([-0.5,-0.5],"$A(x)=\frac{1}{2}\int_0^x\,\tan^2(\theta)\,d\theta$",size->18);
drawtext([1.2,2.5],"$r(\theta)=\tan(\theta)$",size->18,color->[0.2,0.4,0.3]);
drawtext([2.5,-0.5],"$A(\theta)=$",size->18,color->[0,0,1]);
drawtext([3,-0.5],"$\frac{1}{2}\tan(\theta)-$",size->18);
drawtext([3.87,-0.5],"$\frac{\theta}{2}$",size->18,color->[0,1,0]);

//draw(f(d.angle),size->2,color->[1,1,1]);
moveto(B,f(d.angle));
```

```

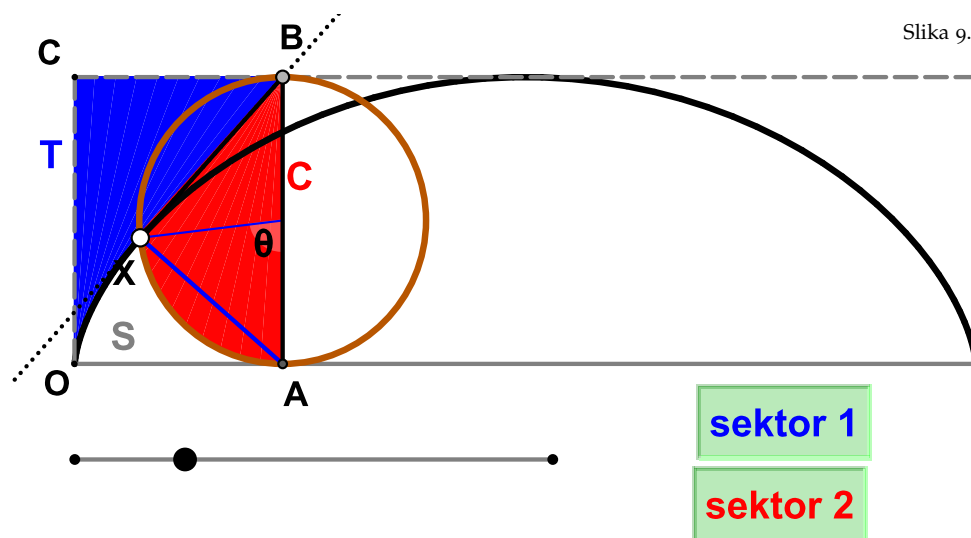
aa=0..50;
aa=(d.angle)*aa/50;
ab=[];
repeat(51,i,ab=append(ab,f(aa_i)));
fillpoly(ab);

```

Geometrijski elementi su kružnica sa slobodnom tačkom na periferiji (D), vertikalna prava kroz njen centar (b), tangenta duž na kružnicu $d = DE$, gde je E tačka preseka tangente sa pravom b . Konačno, definisana je duž $a = AB$, kao paralela kroz slobodnu tačku A tangente d .

Primer 39

Odrediti površinu cikloide primenom teoreme o brišućoj tangenti. Dokazati da je površina sektora cikloide ($OAXO$) u svakom trenutku jednaka trostrukoj vrednosti površine odsečka kružnice nad sečicom AX .



Slika 9.5: Finalna animacija

Prvi deo zadatka je direktna posledica drugog dela. Najpre primećimo (na slici 9.5) da je površina pravougaonika $OABC$ jednaka zbiru površina nadsektora cikloide (prikazan plavom bojom i označen sa T), sektora kružnice (crvenom bojom, oznaka C) i (nebojenog) "klina" označenog sa S . Dakle,

$$P_{OABC} = P_T + P_C + P_S; P_{OABC} = 2r \cdot r\theta = 2r^2\theta.$$

Na osnovu teoreme o brišućoj tangenti, $P_T = P_C$. Takođe, P_C je jednako zbiru površine trougla ABX (označimo ga sa P_Δ) i otsečka nad sečicom AX (označimo ga sa P_{segm}). Dakle, $P_S = P_{OABC} - 2(P_\Delta + P_{segm})$. Međutim, površina centralnog kružnog sektora sa uglom θ pri vrhu je jednaka $1/2r^2\theta = P_{segm} + 1/2P_\Delta$, a istovremeno i $1/2r^2\theta = 1/4P_{OABC}$, pa je $P_{OABC} = 4P_{segm} + 2P_\Delta$. Sledi da je $P_S = 2P_{segm}$, a odavde i $P_{OAXO} = 3P_{segm}$, što je i trebalo dokazati.

Geometrijski elementi su pravougaonik $CONB$, prečnik PG (na slici označen sa AB), kružnica C_0 sa centrom A i tačkom na periferiji koja "generiše" cikloidu M (na slici označena sa X) i klizač EF kojim se generiše parametar t . Prateći skript je:

```
// crtanje cikloide
r=C0.radius;
t=|E,K|/|E,F|*2*pi;
x(t):=r*(t-sin(t),1-cos(t));
y(tt,t):=(r*cos(tt)+r*t,r*sin(tt)+r);

//definisanje pomocnih tacaka za crtanje
A.x=r*t;M.xy=x(t);
GG(t):=[r*t,2*r];
MM=x(t);
ss=0..10;
s=t*ss*pi/31;
xx=C;
repeat(11,i,xx=append(xx,[x(s_(i)),x(s_(i+1)),GG(s_(i+1)),GG(s_(i))]));
yy=apply(s,y(-(pi/2+#),t));

// crtanje
repeat(11,i,fillpoly(xx_i,color->if(Text0.pressed,[0,0,1],hue((i-1)/13))););
repeat(11,i,fillpoly([G.xy,yy_(i),yy_(i+1)],color->if(Text1.pressed,[1,0,0],hue((i)/13))););
linesize(3);
plot(x(t),t,start->0*pi,stop->2*pi,step->400,color->(0,0,0));
linesize(8);
```

Primer 40

Odrediti površinu kardioide primenom teoreme o brišućoj tangenti.

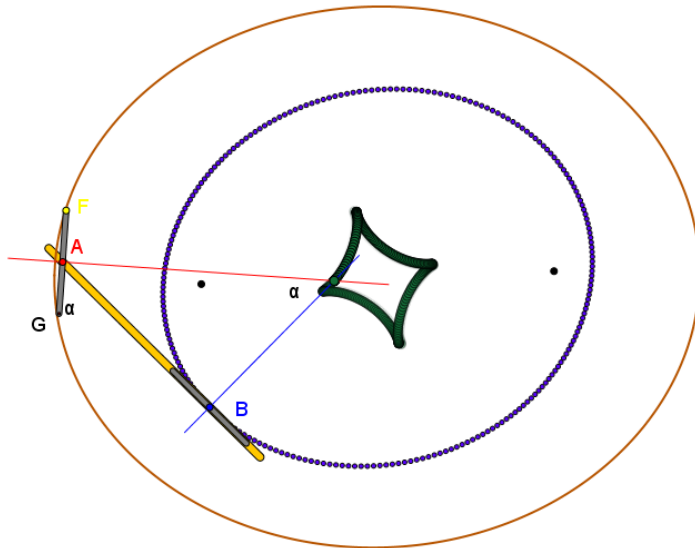
- Iskoristiti osobinu da je kardioida (u opštijem slučaju limakon) pedalna kriva kružnice.
- Iskoristiti osobinu da je kardioida (opštije epicikloida) ruletna kriva nad kružnicom.

Dok su epiciklične krive definisane kompozicijom rotacija, traktrise su specifične krive generisane kompozovanjem translacije i rotacije ali duž različitih pravaca. Matematički model bicikla je: duž fiksne dužine RF , kod koje se prednja tačka F pomera slobodno, duž proizvoljne krive, dok je zadnja tačka R prinuđena da se kreće tako da joj vektor brzine ima pravac duži RF . Pod *putanjom* bicikla podrazumevamo trag zadnje tačke. Kriva duž koje se kreće tačka F naživamo *vođicom*.

Primer 1: Krivina putanje

Pokazati da se centar C krivine putanje bicikla nalazi u preseku normala na putanju i vođicu (tj. u preseku normala na zadnji i prednji točak).

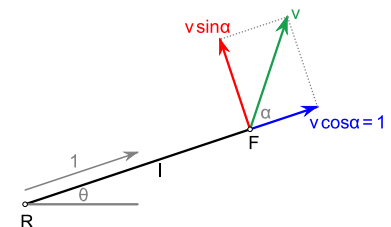
Situacija je prikazana na slici 9.6 u slučaju eliptične vođice.



Slika 9.6: Putanja bicikla sa eliptičnom vođicom. Bicikl je konstruisan kao duž AB . Prednji i zadnji točak su takođe duži - sivo obojene. Vodeća tačka F je slobodna i prednji točak je konstruisan kao udvojena duž FA : druga tačka prednjeg točka G , je dobijena u preseku kružnice sa centrom u A i periferijskom tačkom F . Zadnji točak je dobijen presekom kružnice sa slobodnim poluprečnikom i centrom u B . Za putanju je fiksirana samo tačka F .

Najpre dokažimo da krivina (κ) putanje u datoj tački zavisi samo od ugla α (među točkovima), ali ne i od njegove promene $d\alpha/ds$. Neka se zadnji točak kreće jediničnom brzinom. Tada važi:

$$\kappa = \frac{d\theta}{ds} = \frac{v \sin \alpha}{l} = \frac{\operatorname{tg} \alpha}{l}.$$



Slika 9.7: Vektori brzine prednjeg i zadnjeg točka.

Inače, model bicikla je generisan sa nekoliko linija koda:

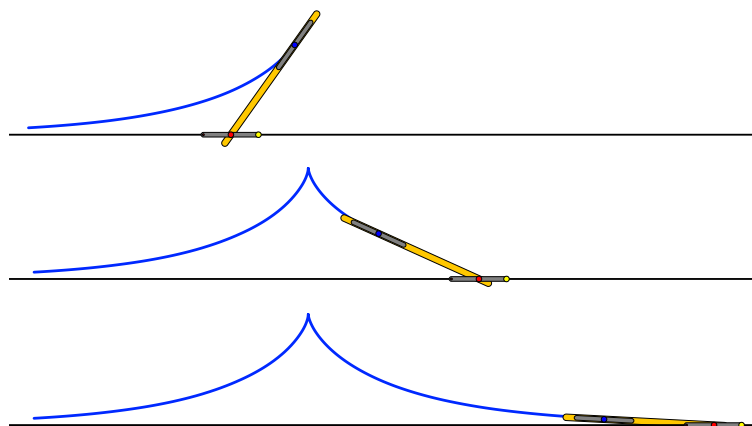
```
//inicialisation slot
l=[];//setovanje liste za trag

//draw slot
r=|C,E|;
ep=A.xy-B.xy;
B.xy=A.xy-r*ep/|ep|;
l=l+[B.xy]);//generise trag zadnjeg tocka
drawall(connect(l),size->3,colour->[0,0.7,0.3]);//crta trag zadnjeg tocka
```

Primer 2: Traktrisa

Putanja bicikla sa pravolinijskom vođicom se naziva *traktrisa*. Pomoću modela bicikla iz prethodnog primera, generisati traktrisu i odrediti njenu jednačinu.

Animacija za crtanje traktrise je prikazana na slici 9.8.



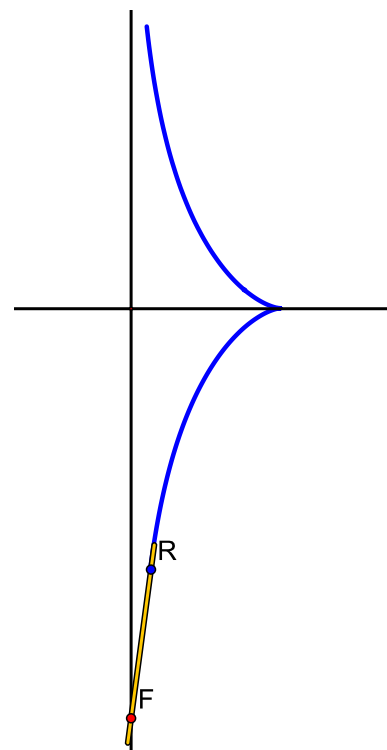
Slika 9.8: Traktrisa: putanja bicikla sa pravolinijskom vođicom.

Odredimo jednačinu traktrise na pojednostavljenom modelu prikazanom na slici desno. Neka je zadnji točak bicikla (R tačka duži) u početku locirana u tački $(a, 0)$, $|F, R| = a$. Ako se prednji točak bicikla kreće duž y - ose, u svakom trenutku je FR tangenta na krivu $y = y(x)$. Nagib tangente:

$$\frac{dy}{dx} = -\frac{\sqrt{a^2 - x^2}}{x}, \quad y(a) = 0.$$

Integracijom dobijamo

$$y = \pm \left(a \ln \frac{a + \sqrt{a^2 - x^2}}{x} - \sqrt{a^2 - x^2} \right) = \pm \left(a \operatorname{arcsech} \frac{x}{a} - \sqrt{a^2 - x^2} \right).$$



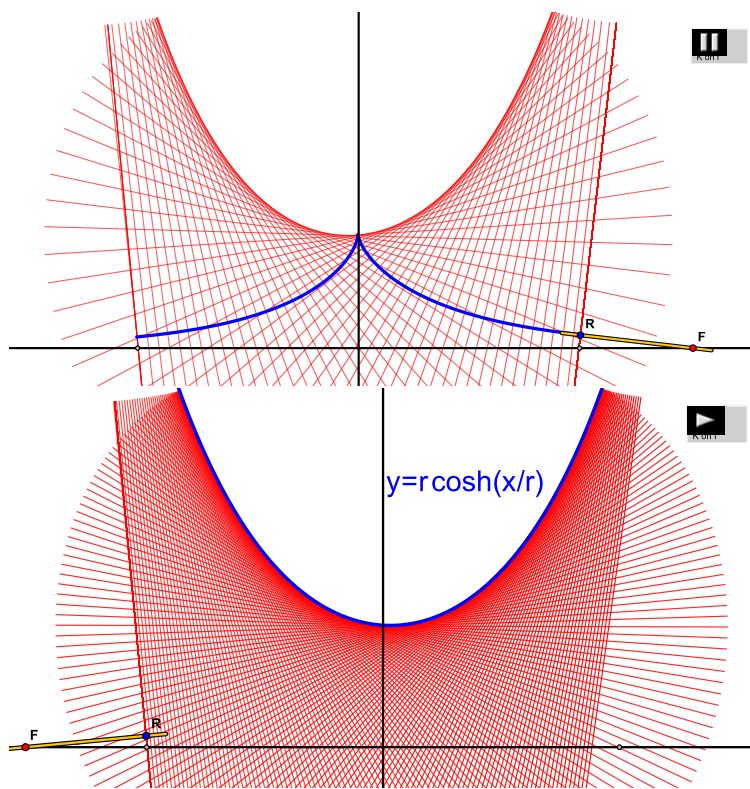
Primer 3: Involuta lančanice

Demonstrirati pomoću modela bicikla da je traktrisa involuta lančanice, tj. da je lančanica evoluta traktrise (skup centara kružnice dodira).

Na slici 9.9 je prikazan trag normale bez i sa obvojnicom. Obvojnica normala je evoluta, a njena eksplisitna jednačina je

$$y = r \cosh \frac{x}{r},$$

gde je $|FR| = r$.

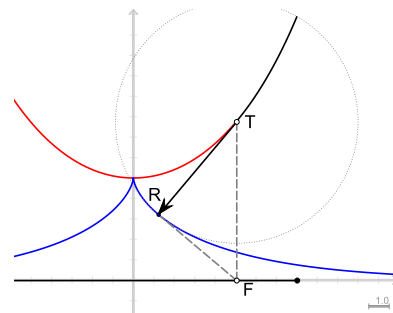


Posmatrajmo pravougli trougao TRF sa slike desno. Dužina duži FT je $|FT| = y = \cosh x$. Ako ugao kod temena T označimo sa θ , tada je $y = 1 / \sin \theta$. Istovremeno, θ je ugao između prednjeg i zadnjeg točka. Prednji se kreće duž horizontalne (x) ose, a zadnji je uvek tangenta na traktrisu. Promena ugla θ u zavisnosti od x jednaka je ugaonoj brzini prednjeg točka u sistemu reference zadnjeg točka: normalna komponenta je $\sin \theta$, pa ako uzmemo da je $|RF| = 1$, dobijamo da je

$$\frac{d\theta}{dx} = -\sin \theta.$$

Evoluta date krive γ je kriva $\bar{\gamma}$ koja je formirana od centara krivine krive γ . Obratno, za krivu γ se kaže da je involuta od $\bar{\gamma}$.

Slika 9.9: Traktrisa i njena evoluta.



Diferenciranjem y po x dobijamo,

$$y' = \operatorname{ctg}\theta, y'' = -\frac{1}{\sin^2\theta}(-\sin\theta) = \frac{1}{\sin\theta} = y.$$

To znači da je y linearna kombinacija funkcija $\sinh x$ i $\cosh x$. Početni uslovi su:

$$y(0) = 1, y'(0) = 0,$$

pa je konačno rešenje:

$$y = \cosh x.$$

Za konstrukciju je iskorišćen pojednostavljeni model bicikla, model "neholonomnog štapa": model krutog segmenta čiji zadnji kraj (prateća tačka) prati kretanje prednjeg (vodeća tačka). Za ovaj model dovoljno je nacrtati duž sa slobodnim krajevima AB . Potreban skript je

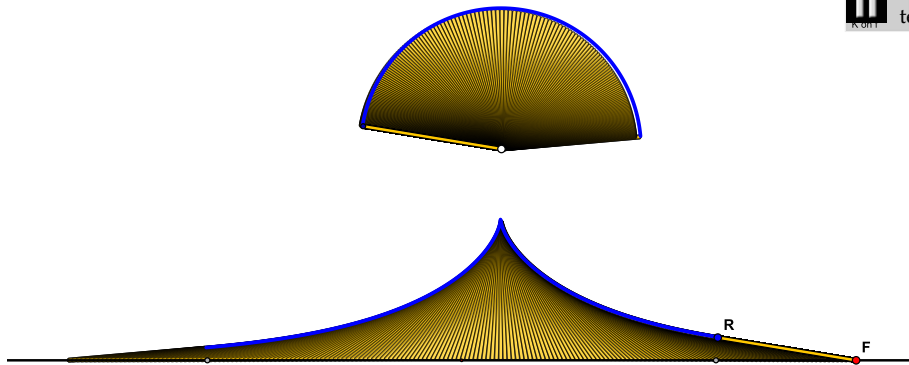
```
//Initialization slot
ap=A.xy;
l=[];
//Draw slot
d=3;// def. duzinu stapa
del=A.xy-ap;
ep=A.xy-B.xy;
B.xy=A.xy-d*ep/|ep|;//definise na koji nacin tacka B prati tacku A
ap=A.xy;
l=l+[B.xy];//lista predjenih tacaka za crtanje traga
drawall(connect(l),size->3,colour->[0,0.7,0.3]);//crtanje traga
```

Primer 4: Površina traktrise

Odrediti površinu ispod traktrise metodom brišuće tangente. Rešenje: $A = \pi r^2/2$.

Rešenje je prikazano na slici 9.10.

Kao osnovno sredstvo za proučavanje traktrise koristili smo model "neholonomnog štapa". Na ovom mestu možemo napraviti digresiju primedbom da je model segmenta možemo iskoristiti za konstrukciju modela lanca - niza povezanih krutih segmenata.

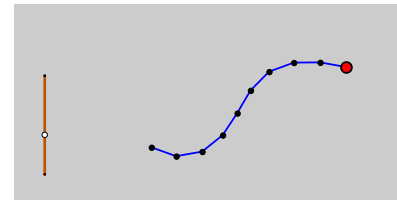


Primer 5: Model asimetričnog lanca

Konstruisati model asimetričnog lanca, tj. lanca koji je moguće povlačiti samo sa jednog kraja.

Konstrukcija i prateći skript. Jedini geometrijski elementi su slobodne tačke. Vodeća tačka lanca je prikazana crvenom bojom.

```
r=1*|C,M|/|C,B|;
if(Text0.pressed==false,l=[]);
ep(p):=((p_1).xy-(p_2).xy)/|p_1,p_2|;//funkcija jedinичnog vektora od 2. ka 1. tački
pts=allpoints--[C,B,M]);// setuje tačke koje formiraju lanac
a=consecutive(pts);
repeat(length(pts)-1,i,(pts_(i+1)).xy=(pts_i).xy-r*ep(a_(i)));// def. kretanje "bicikla"
connect(pts,size->2);// crta segmente lanca
l=l++[0.xy];connect(l,size->10,color->red(0.6));//crta trag ako je dugme uključeno
```

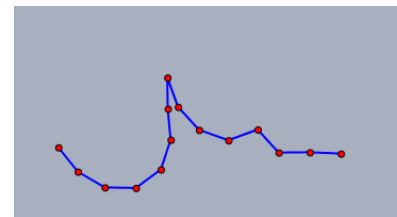


Primer 6: Model realnog lanca

Konstruisati model lanca koji je moguće povlačiti polazeći od proizvoljnog segmenta.

U ovom slučaju potrebno je napraviti malu modifikaciju skripta: nakon levog klika miša na neku tačku na lancu, skript deli sve tačke u dva skupa i primenjuje skript iz prethodnog zadatka na oba skupa nezavisno.

```
d=0.4;
if(Text0.pressed==false,l=[]);
ep(p):=((p_1).xy-(p_2).xy)/|p_1,p_2|;
```



```

pts=allpoints();
m=mover();
l=[];j=1;
while(pts_j!=m,l=prepend(pts_j,l);j=j+1);
r=pts--l;l=prepend(pts_j,l);

a=consecutive(l);b=consecutive(r);
repeat(length(l)-1,i,(l_(i+1)).xy=(l_i).xy-d*ep(a_(i)));
repeat(length(r)-1,i,(r_(i+1)).xy=(r_i).xy-d*ep(b_(i)));
connect(pts,size->2);

```

Primer 7: Igračka na lancu

Konstruisati model lanca sa figuricom na kraju.

Za proizvoljnu tačku lanca je lako zakačiti sliku. Problem je samo konstantno podešavati ugao slike prema uglu, koji u odnosu na horizontalu formira neki od segmenata lanca. Takođe, slika mora biti bez background-a. Background je lako ukinuti u Mathematica programu komandom:

```

a = Import["d:\\car.jpg"]; a = RemoveBackground[a];
(*c=ImageReflect[b,Left]; - ako je potrebno*)
Export["d:\\car1.png", a];

```

Dodatne komande za podešavanje ugla i crtanje slike vezane za tačku A.

```

ang=im(log(complex(ep(a_(-1))))); //ugao slike
drawimage(A,"car",scale->0.2,angle->ang);

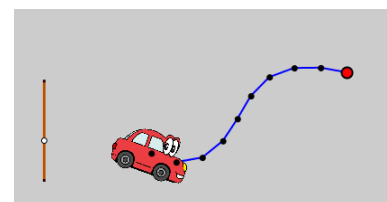
```

Primer 8: Pricov (Prytz) planimetar

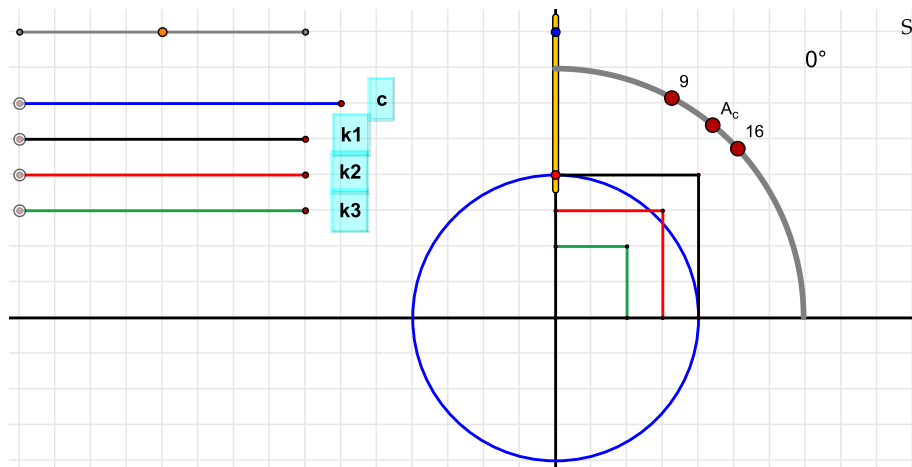
Demonstrirati merenje površine četvrtine kruga sa Pricovim planimetrom.

Pricov planimetar nije ništa drugo do naš model bicikla.

Na slici 9.11 je demonstrirano približno izražunavanje površine četvrtine kruga, nakon kalibracija planimetra pomoću kvadrata poznate površine.



Instruktivno je demonstrirati merenje površine kruga, tj. integrala $\int_0^1 \sqrt{1-x^2} dx$ pravim biciklom. Umesto bicikla, moguće je iskoristiti i kašiku!



Slika 9.11: Pricov planimeter "u akciji".

Primer 9: Teorija Pricovog planimetra

Odrediti povezanost krajnjeg položaja planimetra i merene površine. Na osnovu toga odrediti od čega zavisi greška merenja i odrediti metod merenja pri kojem je minimalna.

10 Polinomijalne aproksimacije proizvoljnih krivih

Bezijeove (P. Bezier) krive su sveprisutne u kompjuterskoj grafici. Definisane su još u 19. veku u radovima matematičara Ermita (C. Hermite) i Bernštajna (S. Bernstein), ali su tek u 20. veku, zahvaljujući Bezijeu, inženjeru Renoa i de Kasteljau (P. de Casteljau), inženjeru Citroena, postale značajne u računarskoj grafici. Sa druge strane, kao i sve duboke matematičke ideje, i ova je značajna kako u teoriji (analiza) tako i u praktičnim primenama.

Osnovni zadatak računarske grafike je sledeći: za dati skup tačaka, npr. u ravni, konstruisati dovoljno glatku krivu koja ih aproksimira. Pritom, kriva ne mora nužno da prolazi kroz date tačke. Ukoliko prolazi, radi se o interpolaciji. Uslov je da metod mora biti efikasan, tj. da sadrži relativno mali broj računskih operacija.

10.1 Bezierove krive i Bernštajnovi polinomi

Bernštajnov polinom stepena n , asociran nizu brojeva x je definisan na sledeći način:

$$B_{n,x}(t) = \sum_{k=0}^n \binom{n}{k} (1-t)^{n-k} t^k x_k, \quad x = (x_0, \dots, x_n).$$

Ukoliko je niz x formiran smplovanjem funkcije f u n ekvidistantnih tačaka, $x = (f(0), \dots, f(k/n), \dots, f(1))$, tada je Bernštajnova aproksimacija funkcije (krive) f definisana sa:

$$B_{n,f}(t) = f(0)(1-t)^n + f\left(\frac{1}{n}\right) n(1-t)^{n-1}t + \dots + f\left(\frac{1}{n}\right) \binom{n}{k} (1-t)^{n-k}t^k + \dots + f(1)t^n.$$

Odgovarajuća kriva, dobijena plotovanjem Bernštajnovog polinoma se naziva Bezijeovom krivom.

Primer 1: Bezijeove krive: algebarski pristup

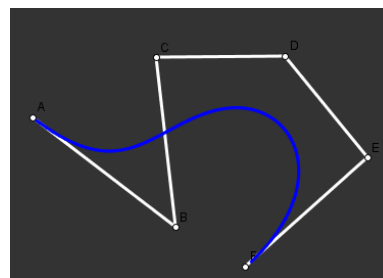
Kreirati program koji za skup od pet nadovezanih duži (definisanih sa šest tačaka) crta Bezijeovu krivu. Bezijeova kriva je glatka aproksimacija izlomljene kkrive od nadovezanih duži.

Rešenje je prikazano na slici na desnoj strani. Neka je skup semplovanih tačaka označen sa A, \dots, F . Radi preglednosti, tačke su spojene dužima, pa je dobijeno rešenje ujedno i neprekidna aproksimacija izlomljene linije $A - F$. Berštajnov polinom i njegov plot su dati u pratećem skriptu.

```
//u Draw slotu
pts=allpoints--[G,H,K];
pp=pts;
connect(pts,size->3,color->[1,1,1]);
n=length(pts);
t=|K,H|/|H,G|;
repeat(n-1,k,
pts1=[];
repeat(n-k,i,
pts1=append(pts1,(1-t)*pts_i+t*pts_(i+1)));
color(hue((k-1)/(n)));
connect(pts1,size->k);
drawall(pts1,size->k);
pts=pts1);

//def. faktorijela i binomnih koeficijenata
fak(n):=if(n<=0,1, n*fak(n-1));
bp(n,i,t):=fak(n)/fak(n-i)/fak(i)*pow(1-t,n-i)*pow(t,i);

//def. Bernstajnovog polinoma i plot
f(t):=sum(1..n,i,pp_i*bp(n-1,i-1,t));
plot(f(t),t,start->0,stop->1,steps->100, size->3);
```



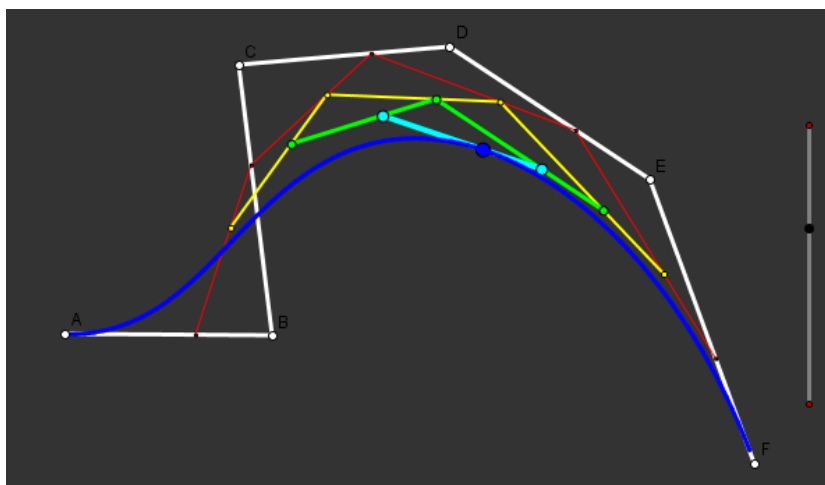
Slika 10.1: Bezijeova aproksimacija izlomljene linije

Primer 2: Bezirove krive: geometrijski pristup

Kreirati geometrijsku konstrukciju koja za datu trojku tačaka crta Bezirovu krivu pomoću de Casteljaovog algoritma.

Sam algoritam se temelji na sledećoj rekurziji. Pretpostavimo da je data kvadratna Bezierova kriva sa kontrolnim tačkama P_i . Podelimo je na dve polovine, sa $P_{012} = B(1/2)$. Neka e P_{01} tačka između P_0 i P_1 , a P_{12} između P_1 i P_2 . U tom slučaju: (1) tačka P_{012} je na sredini između P_{01} i P_{12} ; (2) polovina krive između P_0 i srednje tačke P_{012} je takođe kvadratna Bezierova kriva sa kontrolnim tačkama $P_0, P_{01},$ i P_{012} . Isto važi i za drugu polovinu.

Rešenje je bazirano na pravljenju transferu konveksnih kombinacija pomoću sličnosti odgovarajućih trouglova. Parametar t je definisan položajem tačke st , a na tačke E i F je prenet paralelogramom. Na finalnu tačku K parametar se prenosi na osnovu sličnosti trouglova $\triangle EGK$ i $\triangle EHF$.

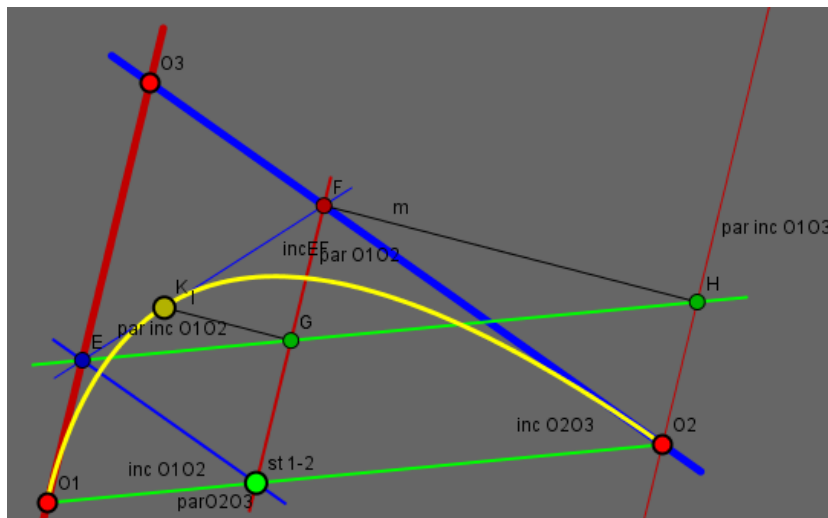


Slika 10.2: Bezierova kriva sa 6 kontrolnih tačaka prikazana sa de Casteljavom konstrukcijom.

Primer 3: Osobine Bezierove krive

Proveriti osnovne osobine Bezierove krive: pozicioniranje u konveksnom omotaču, tangენტnost na početnu i krajnju tačku.

Iz definicije Bernštajnovih polinoma sledi da je za svako $t \in [0, 1]$ vrednost polinoma jednaka otežinjenoj konveksnoj kombinaciji kontrolnih tačaka pa mora ležati i u njihovom konveksnom omotaču. Takođe, lako je videti da je prvi izvod Bernštajnovog polinoma za $t = 0$ umnožak razlike prve dve tačke, a za $t = 1$ umnožak razlike poslednje dve.



Slika 10.3: Konstrukcija na kojoj je paralelizam naglašen istom bojom.

Primer 4: Ermitova interpolacija

Pokazati da za datu četvorku realnih brojeva y_0, v_0, y_1, v_1 , postoji jedinstveni polinom trećeg stepena tako da važi:

$$P(0) = y_0, P'(0) = v_0, P(1) = y_1, P'(1) = v_1.$$

Realizovati datu konstrukciju.

Proizvoljan Bernštajnov polinom 3 stepena može biti izražen kao

$$B_{3,P_i}(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3.$$

Njegov prvi izvod glasi:

$$B'(t) = 3[(1-t)^2(P_1 - P_0) + 2t(1-t)(P_2 - P_1) + t^2(P_3 - P_2)],$$

tj. važe sledeće jednakosti:

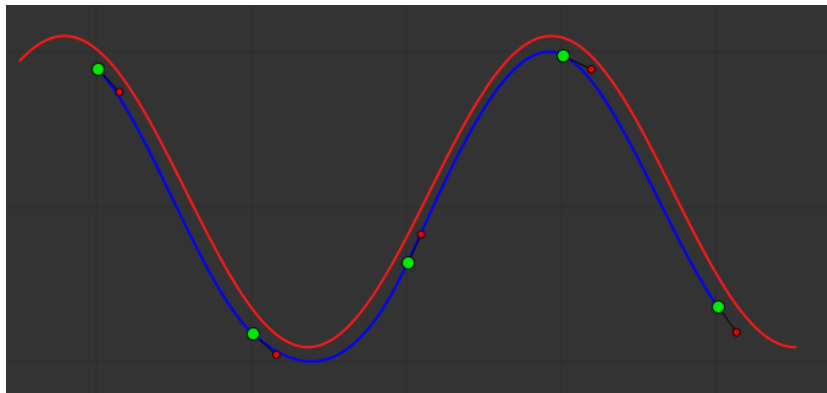
$$B(0) = P_0, B'(0) = 3(P_1 - P_0), B'(1) = 3(P_3 - P_2), B(1) = P_3.$$

Dakle, u oznakama samog zadatka,

$$P(t) = (1-t)^3 y_0 + 3(1-t)^2 t [y_0 + (1/3)v_0] + 3(1-t)t^2 [y_1 - (1/3)v_1] + t^3 y_1.$$

Ovo je specijalni slučaj tzv. Ermitove interpolacije. Naime, na osnovu Ermitove teoreme, uvek je moguće odrediti jedinstveni polinom stepena $n - 1$ koji zadovoljava ukupno n uslova na vrednosti izvoda nižag reda, ne nužno u istoj tački. Navedeni slučaj za polinom 3 stepena se naziva Ermitov kubični splajn. On omogućava efikasnu

Splajn je funkcija definisana deo po deo polinomom.



Slika 10.4: Rukom podešena aproksimacija f -je $\sin(2x)$.

aproksimaciju klase C_1 krivih u n tačaka sa zadatim vrednostima i nagibima u datim tačkama. Realizacija u 5 celobrojnih tačaka je data na slici ispod. Odgovarajući skript je:

```
//u Draw slotu
pts=[A,B,C,D,E];
seg=[a,b,c,d,e];
n=length(pts);
forall(1..n,(pts_#).x=#);
p1(x):=(1-x)^2*(1+2*x);
p2(x):=(3-2*x)*x^2;
q1(x):=x*(1-x)^2;
q2(x):=-x^2*(1-x);

f(y1,y2,h1,h2,x):=y1*p1(x)+y2*p2(x)+h1*q1(x)+h2*q2(x);
repeat(n-1,i,
plot(f((pts_i).y,(pts_(i+1)).y),(seg_i).slope,(seg_(i+1)).slope,x-i),start->i,stop->i+1,size->2);
);
plot(0.1+sin(2*#),start->0.5,stop->5.5,color->[1,0.1,0.1],size->2);
```

Primer 5: Bernštajnovi polinomi

Pokazati da za Bernštajnovne polinome važi generalizacija de Kasteljaovog algoritma: Restrikcija $B_{n,x}(t)$ na svaku poluoblast $[0, 1/2]$ i $[1/2, 1]$ je takođe Bernštajnov polinom, u smislu da svaki od $B_{n,x}(2t)$ i $B_{n,x}(1-2t)$ može biti izražen sa $B_{n,y}(t)$ za neki niz y koji je jednostavno izračunati.

Primer 6: Bernštajnova aproksimacija

Kreirati geometrijsku konstrukciju koja datu krivu aproksimira Bernštajnovim polinomom.

Neka je f neprekidna funkcija na $[0, 1]$. Neka je za svako n , f_n niz od $n + 1$ vrednosti $f(k/n)$ za $k = 0, \dots, n$. Konkretno, neka je

$$B_{n,f}(t) = B_{n,f_n}(t) = f(0)(1-t)^n + f(1/n)n(1-t)^{n-1}t + \dots + f(k/n)C_{n,k}(1-x)^{n-k}t^k + \dots + f(1)t^n.$$

Tada važi sledeća teorema: Polinomi $B_{n,f}(x)$ konvergiraju uniformno na $[0, 1]$ ka $f(x)$, pri $n \rightarrow \infty$.

11 Statičke konfiguracije

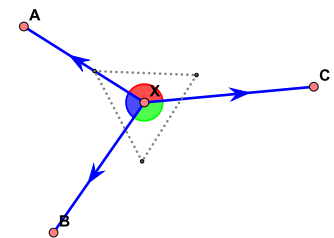
Važnost proračuna ravnoteže i stabilnosti statičkih konfiguracija je teško preceniti, posebno ako vas put do škole ili posla svakodnevno vodi preko mosta. Zbog toga je i nastala nauka koja se bavi njihovim proučavanjem: statika. Statika je istovremeno i primenjena i teorijska nauka. Fizičari je obično ne razdvajaju od dinamike; izdvojena je samo na osnovu tipa zadataka koje rešava, a to su uslovi ravnoteže fizičkog sistema sa interakcijama. Matematički, zadatak se svodi na minimizaciju odgovarajuće funkcije (potencijala, energije, slobodne energije...). Stoga je često u matematičkim problemima sa minimizacijama korisno kreirati odgovarajuću fizičku analogiju, na osnovu koje je moguće dati intuitivan odgovor bez detaljnih izračunavanja.

Primer 7: Ferma-Toričelijev problem

Pjer Ferma je u pismu Toričeliju postavio sledeći problem. Date su tri tačke u ravni, A , B i C . Odrediti položaj tačke X tako da zbir udaljenosti $|XA| + |XB| + |XC|$ bude minimalan. Zadatak rešiti mehaničkom analogijom i geometrijski.

Mehanička analogija se svodi na određivanje statičke konfiguracije, a ona zadovoljava uslov minimalne potencijalne energije. Potrebno je samo izabrati odgovarajući tip interakcije kod koje energija zavisi linearno od rastojanja. Standardni primer je gravitaciona potencijalna energija u homogenom polju (Zemlje). (Ako bi okačili tegove jednake mase i niti provukli ispod površine na kojoj su izbušene rupe na mestima gde su zadate tačke). Druga mogućnost je "opruga fiksnog napona", tj. ona kod koje je sila konstantna, nezavisna od istežanja (za razliku od standardne, "Hukovske opruge"). Pošto su sile jednake, a telo X miruje, znači da je rezultanta nulta, pa je ugao među oprugama 120° . Takva tačka se naziva Toričelijevom tačkom. Opruga fiksnog napona je kreirana sledećim skriptom:

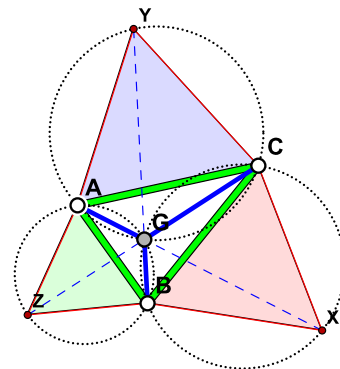
```
//u Integration Tick slotu
av=(A-X).xy;a=|av|;
bv=(B-X).xy;b=|bv|;
cv=(C-X).xy;c=|cv|;
```



```
setforce(X, (1/a)*av+(1/b)*bv+(1/c)*cv); //definicija rez. sile na masu X
```

Fizički elementi su masene tačke A, B, C i X .

Geometrijsko rešenje je prikazano na slici desno. Nad stranicama trougla ABC konstruisana su tri jednakostranična trougla (sa dodatnim temenima X, Y, Z). Dodatna temena jednakostraničnih trouglova su spojena sa naspramnim temenima trougla ABC : X sa A , Y sa B i Z sa C . U preseku te tri duži se nalazi Toričelijeva tačka. Dokaz: Opišimo kružnicu oko trougla BXC i oko trougla ZBA . Te dve kružnice se seku u tački B i dodatnoj tački koju ćemo označiti sa G . Tada duži ZG i GC pripadaju istoj pravoj, jer je $\angle AGB = 120^\circ$ (nad duži BA sa suprotne strane) i $\angle BGZ = 60^\circ$ (nad duži ZB sa iste strane). Na sličan način se dokazuje da duži DG i GY pripadaju istoj pravoj. S'obzirom da su uglovi $\angle CGB = \angle BGA = 120^\circ$ tada je i $\angle AGC = 120^\circ$, pa sledi da tačka G pripada i kružnici opisanoj oko trougla ACY . Naravno, duži YG i GB pripadaju istoj pravoj.



Slika 11.1: Konstrukcija i dokaz Ferma-Toričelijevog problema.

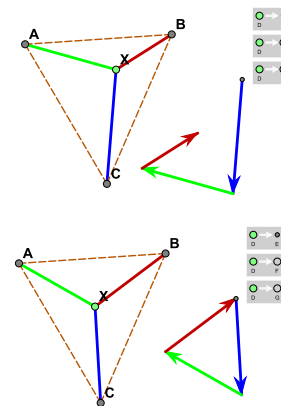
Primer 8: Ferma-Toričelijev problem: verzija d_2

Date su tri tačke u ravni, A, B i C . Odrediti položaj tačke X tako da zbir kvadrata udaljenosti $|XA|^2 + |XB|^2 + |XC|^2$ bude minimalan. Zadatak rešiti mehaničkom analogijom i geometrijski.

Sumu kvadrata je moguće interpretirati kao potencijalnu energiju standardnih (Hukovskih) opruga sa nultom ravnotežnom dužinom, koje spajaju masenu tačku X sa masenim tačkama A, B i C . Energija sistema opruga je $\sum_{i=1}^3 \frac{1}{2} k_i L_i^2$ pa za $k_i = 2$, dobijamo traženi zbir kvadrata. U ravnoteži je zbir sila koje deluju na X nulti, pa je, zbog

$F_i = k_i L_i, \sum_{i=1}^3 \vec{F}_i = 0$. To znači da je tačka X centar mase trougla.

(Jer je $\vec{r}_{cm} = (m\vec{X}A + m\vec{X}B + m\vec{X}C)/(3m) = \vec{X}M$, a to je jednako nuli ako je tačka X u centru mase, tj. kada je $X = M$.) Slika zbira vektora je dobijena translacijama opruga i njihovim sukcesivnim nadovezivanjem: kraj jedne za početak druge.



Slika 11.2: Situacija pre i nakon pokretanja opruga

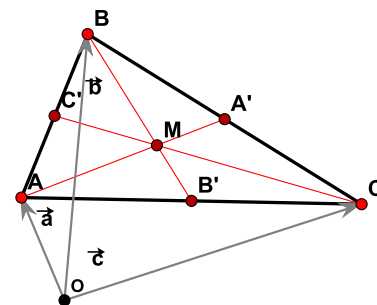
Centar mase materijalnih tačaka A , B i C sa jednakim masama, se podudara sa težištem trougla ABC .

Dokaz:

Označimo radijus vektore tačaka A , B i C sa \vec{a} , \vec{b} i \vec{c} . Tada je radijus vektor tačke A' sredine stranice BC jednak $1/2(\vec{b} + \vec{c})$ itd. Tačkama na duži (težišna duž) AA' odgovaraju radijus vektori

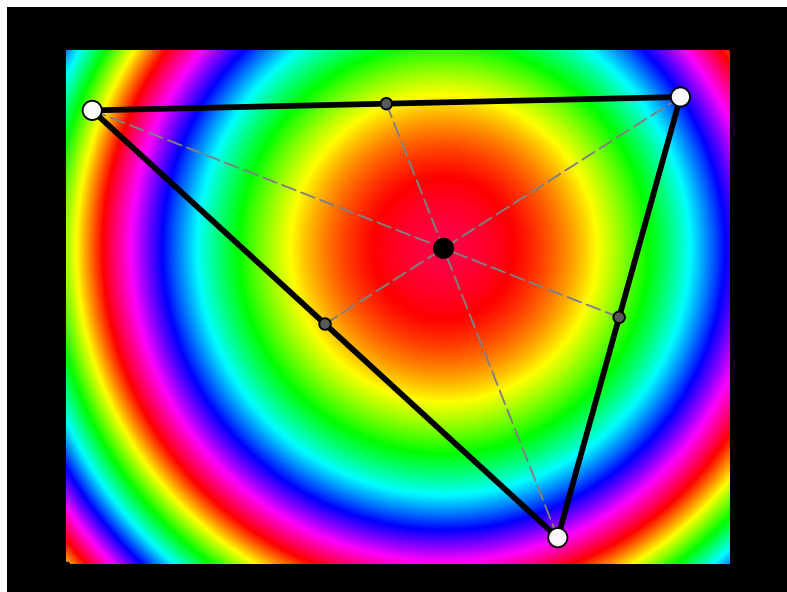
$$(1 - \lambda)\vec{a} + \lambda \frac{1}{2}(\vec{b} + \vec{c}).$$

Za $\lambda = 2/3$ dobijamo vektor $\frac{1}{3}(\vec{a} + \vec{b} + \vec{c})$, on pripada duži AA' , ali zbog simetrije na zamenu temena, pripada istovremeno i dužima BB' i CC' , pa zaključujemo da se one seku u jednoj tački (konkurentne su), a ta tačka je upravo težište. Takođe, težište deli težišne duži u odnosu 2 : 1.



Primer 9: Težište - colorplot

Date su tri tačke u ravni, A , B i C . Prikazati pomoću funkcije `colorplot` (hue funkcijom) funkciju zbira kvadrata udaljenosti $|XA|^2 + |XB|^2 + |XC|^2$, od tačaka A , B i C do proizvoljne tačke u ravni (X). Prikazati na slici i položaj težišta trougla ABC .



Slika 11.3: Funkcija zbira kvadrata rastojanja do tačaka A , B i C . Ekviskalarne linije su kružnice. Prikazano je i težište trougla ABC tako da se jasno vidi da se položaj težišta nalazi u minimumu funkcije (centar ekviskalarnih kružnica).

Skript:

```
colorplot(hue(0.006*((dist(C,#))^2+(dist(D,#))^2+(dist(E,#))^2))
,A,B,pxlres->0.2,startres->15);
```

Važi opštiji stav od tvrđenja prethodnog zadatka.

Neka je dato n tačaka u ravni. Skup tačaka za koje je zbir kvadrata rastojanja od datih tačaka konstantno je kružnica čiji je centar centar mase (centroida) datih tačaka.

Označimo sa O centroidu datih tačaka i postavimo koordinatni početak kompleksne ravni. Date tačke su tada reprezentovane kompleksnim brojevima z_1, z_2, \dots, z_n , za koje važi

$$\sum_{k=1}^n z_k = 0.$$

Neka je z proizvoljna tačka kompleksne ravni, Izračunajmo zbir kvadrata modula rastojanja do datih tačaka:

$$\sum_{k=1}^n |z - z_k|^2.$$

Tada je k -ti član sume

$$(z - z_k)(z^* - z_k^*) = |z|^2 + |z_k|^2 - z z_k^* - z^* z_k.$$

Sumiranjem po k i korišćenjem uslova za centroidu,

$$\sum_{k=1}^n |z - z_k|^2 = n|z|^2 + \sum_{k=1}^n |z_k|^2 = n|z|^2 + D_n^2,$$

gde je $D_n^2 = 1/n \sum_{k=1}^n |z_k|^2$ srednji kvadrat rastojanja od centroide do datih tačaka. Prema tome, $\sum_{k=1}^n |z - z_k|^2$ je konstantno samo ako je $n|z|^2 + D_n^2$ konstantno, a to važi samo za tačke na kružnicama centriranim u centroidi O (tj. za njih je $|z|^2$ konstantno).

Posledica: ako date tačke z_1, z_2, \dots, z_n , leže na kružnici radijusa R , tada važi

$$\sum_{k=1}^n |z - z_k|^2 = n(|z|^2 + R^2).$$

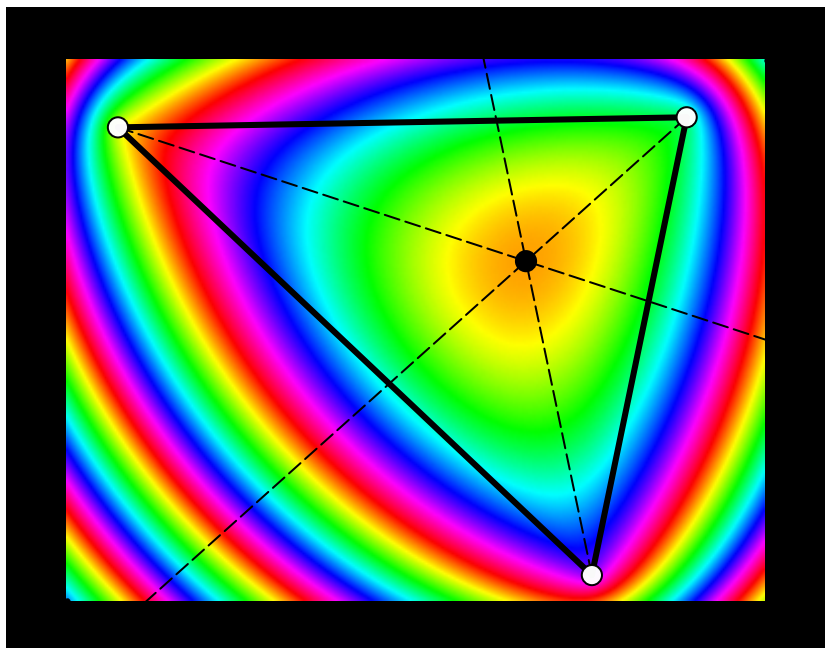
Ako su tačke z_1, z_2, \dots, z_n , temena centralno simetričnog mnogougla, i ako tačka z takođe leži na istoj (opisanoj) kružnici, tada važi

$$\sum_{k=1}^n |z - z_k|^2 = 2nR^2,$$

što je svojevrsna generalizacija Pitagorine teoreme.

Primer 10: Toričeli - colorplot

Date su tri tačke u ravni, A , B i C . Prikazati pomoću funkcije colorplot funkciju zbira udaljenosti $|XA| + |XB| + |XC|$, od tačaka A , B i C do proizvoljne tačke u ravni (X). Prikazati na slici i položaj Toričelijeve tačke trougla ABC .



Slika 11.4: Funkcija zbira rastojanja do tačaka A , B i C . Prikazano je i Toričelijeva tačka trougla ABC tako da se jasno vidi da se njen položaj nalazi u minimumu funkcije.

Prethodna dva problema je moguće generalizovati na sledeći način. Za date tri tačke u ravni, A , B , C odrediti tačku M za koju je $d_n \equiv |||_n$ rastojanje minimalno. Rastojanje definisano metrikom $|||_n$ je

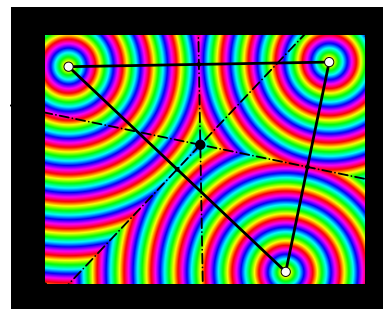
$$||M, A||_n = (|M_x - A_x|^n + |M_y - A_y|^n)^{1/n}.$$

U slučaju kada je $n = \infty$, tada je

$$||M, A||_\infty = \max(|M_x - A_x| + |M_y - A_y|).$$

Primer 11: $\|\cdot\|_\infty$ colorplot

Date su tri tačke u ravni, A , B i C . Prikazati colorplot-om (hue funkcijom) funkciju zbira udaljenosti od tačaka A , B i C do proizvoljne tačke u ravni X u odnosu na metriku $\|\cdot\|_\infty$. Prikazati na slici i položaj tačke sa minimalnim rastojanjem.



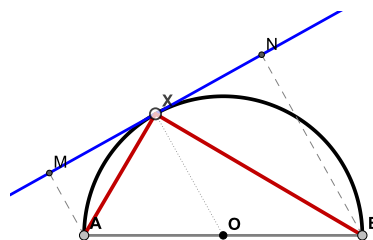
Skript:

```
colorplot(hue
(0.8*min([dist(C,#),dist(D,#),dist(E,#)]))
,A,B,pxlres->0.2,startres->15);
```

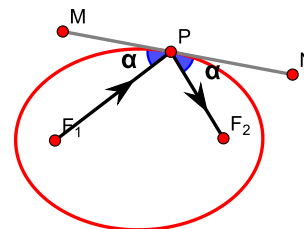
Primer 12: Prsten na polukružnici

Mali prsten X klizi bez trenja duž krute polukružnice. Dve identične opruge (Hukovske, nulte ravnotežne dužine) XA i XB spajaju prsten sa krajevima prečnika. Demonstrirati i dokazati da je prsten u ravnoteži u proizvoljnoj poziciji na polukružnici, a zatim pokazati da iz te činjenice sledi Pitagorina teorema.

Prsten na krugu spojen oprugama nulte dužine je zaista u ravnoteži. Kako je tangenta normalna na popuprečnik OX , to su i projekcije poluprečnika OA i OB na tangentu jednake ($|XM| = |XN|$). Međutim, te projekcije su ujedno i tangencijalne projekcije vektora elastičnih sila kojima opruge deluju na prsten.

**Primer 13: Optičko svojstvo elipse**

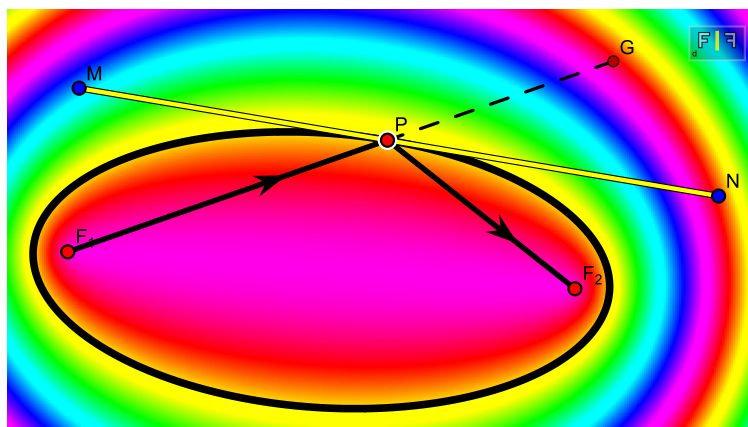
Dokazati tzv. optičko svojstvo elipse: svetlosni zrak emitovan iz jedne žiže elipse će nakon refleksije od elipse, proći kroz drugu žižu. Geometrijska formulacija: Neka je P tačka na elipsi sa žižama F_1 i F_2 i neka je MN tangenta na elipsu u tački P , tada je $\angle F_1PM = \angle F_2PN$. Dati geometrijski i mehanički dokaz.



Geometrijsko rešenje se zasniva na Fermatovom principu, na osnovu kojeg je vreme potrebno svetlosnom zraku da pređe put između dve tačke, ekstremalno (u ovom slučaju minimalno). Vreme kretanja svetlosti u homogenoj sredini je proporcionalno rastojanju, a rastojanje između žiža je $|F_1, F_2| = |F_1, G|$ jer je tačka G slika tačke F_2 osnom simetrijom u odnosu na tangentu MN . Najkraća putanja je pravolinijska, pa su upadni i izlazni ugao zraka jednaki. Zrak reflektovan u odnosu na bilo koju drugu tačku na tangenti MN bi imao duži put, jer su, na osnovu definicije elipse sve tačke izvan elipse imaju veći zbir rastojanja u odnosu na tačke na samoj elipsi. Bojom na grafiku je

prikazana funkcija zbira rastojanja od proizvoljne tačke ravni do žiža elipse. Odgovarajući skript je

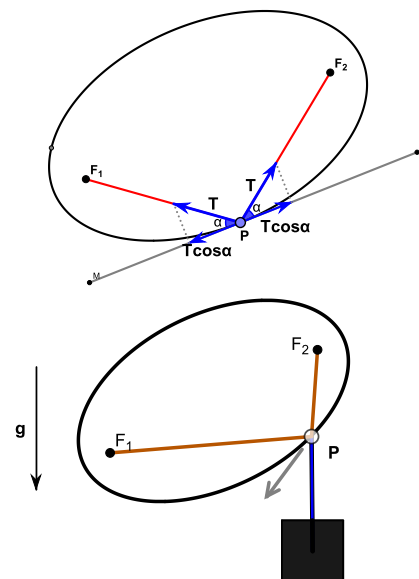
```
//u Draw slotu
f(x,y):=sqrt((x-A.x)^2+(y-A.y)^2)+
sqrt((x-B.x)^2+(y-B.y)^2);
colorplot(hue(0.1*f(x,y)),H,K);
```



Ukoliko prsten koji klizi bez trenja duž elipse spojimo sa žižama oprugom fiksnog napona, tada je prsten u ravnoteži u proizvoljnoj tački na elipsi. To znači da su tangencijalne komponente napona na oprugama jednake, pa iz $T \cos \alpha = T \cos \beta$ sledi da je $\alpha = \beta$. Prateći skript je

```
//u Integration Tickc slotu
av~=(D-A).xy;aa~=|av|;
bv~=(D-B).xy;bb~=|bv|;
setforce(D, -(1/aa)*av-(1/bb)*bv);
```

Druga mogućnost je da vežemo krajeve konca za žiže elipse, montiramo koturaču (bez trenja) na konac, opteretimo je tegom i čitav mehanizam postavimo vertikalno. Promenom položaja (visine) žiža u ravni, centar koturače opisuje elipsu, i u svakom trenutku je centar koturače najniža tačka elipse. Tada je horizontala na elipsu (kroz centar koturače) tangenta elipse, a uglovi koje konac zaklapa sa horizontalom su jednaki jer je sila zatezanja konca jednaka kao i njihova horizontalna projekcija.

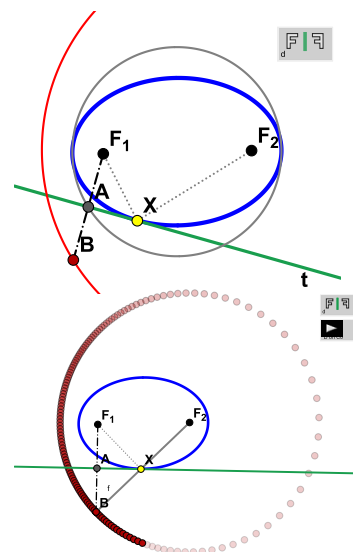


Razmotrimo nekoliko posledica optičkog svojstva elipse.

Primer 14

Data je elipsa sa žižama F_1 i F_2 . a) Dokazati da je skup tačaka (B) osno simetričnih sa F_1 u odnosu na tangente elipse kružnica. b) Dokazati da je skup podnožja normala iz žiže F_1 na tangente elipse (A) kružnica.

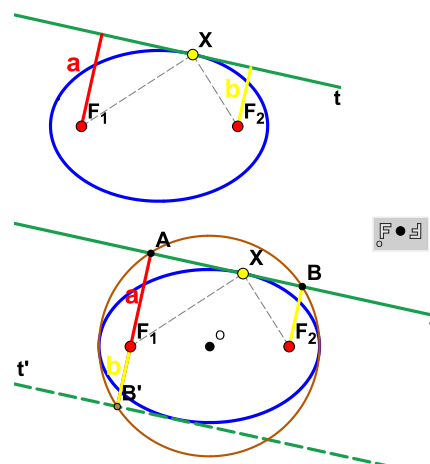
- a) Rešenje sledi direktno iz geometrijskog rešenja prethodnog zadatka. Tačka B se nalazi na pravoj F_2X , pa je F_2B duž čija dužina ne zavisi od položaja tačke X na elipsi, jer je $|F_2, B| = |F_2, X| + |X, B| = |F_2, X| + |X, F_1|$. Prema tome, traženi skup tačaka je kružnica.
- b) Zapazimo da je $|F_1, A| = 1/2|F_1, B|$, i da tačka B leži na kružnici. Zadatak se svodi na primer 7 iz prve glave..



Primer 15

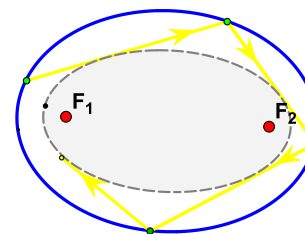
Dokazati da je proizvod rastojanja od žiže elipse do proizvoljne tangente konstantan, tj. nezavisan od tangente.

Definišimo refleksiju u odnosu na centar elipse O . Refleksijom preslikamo tangentu t u f' i duž F_2B u F_1B' . Pošto su F_2B i F_1A paralelne, F_1A i F_2B' leže na istoj pravoj. U prethodnom zadatku je dokazano da podnožja normala iz žiže na tangente elipse pripadaju istoj kružnici, proizvod $a \cdot b$ je jednak potenciji žiže u odnosu na tu kružnicu.



Primer 16

Razmotrimo prostiranje svetlosnog zraka unutar elipse. Pretpostavimo da zrak ne prolazi kroz žiže (ogledalne) elipse. Dokazati da: a) ako početni segment (deo zraka između dve refleksije) ne seče fokalnu duž F_1F_2 , tada nijedan od ostalih segmenata zraka ne seku fokalnu duž i istovremeno su tangentni na jedinstvenu elipsu konfokalnu sa ogledalnom elipsom; b) ako početni segment seče fokalnu duž, tada je seku i ostali segmenti i istovremeno su tangentni na jedinstvenu hiperbolu konfokalnu sa ogledalnom elipsom.



a) Dokažimo da ako (manja) elipsa tangira početni segment zraka P_0P_1 tada ona tangira i sledeći segment P_1P_2 . Konstruišimo tačku A' , osno simetričnu žiži A u odnosu na pravac P_0P_1 i tačku B' osno simetričnu žiži B u odnosu na pravac P_1P_2 . Trouglovi $A'P_1B$ i AP_1B' su podudarni, zbog reflektivnog svojstva veće elipse. Rastojanja $|AB'|$ i $A'B$ su jednaka. Tačka B' pripada većoj kružnici, a tačka M manjoj kružnici iz zadatka 11.

Sledeća tri zadatka se bave osnovnim osobinama cikloide.

Primer 17: Cikloida=tautohrona=izohrona

Pokazati da je cikloida tautohrona (izohrona) kriva. Tautohrona izohrona kriva je, po definiciji, ona kriva duž koje je vreme klizanja (bez trenja) tačke, u homogenom gravitacionom polju, do tačke minimalne visine jednako, nezavisno od početnog položaja.

Zadatak je odrediti oblik krive niz koju će masivna perla bez trenja u homogenom gravitacionom polju preći za najkraće vreme. Pokazaćemo da je ta kriva cikloida. Tačka P na obodu kružnice koja se kotrlja duž ravne podloge se kreće duž cikloide. Neka je PC' tangenta na cikloidu u tački P , dok C' leži na obodu kruga. Primitimo da je $PC' \perp CP$: vektor brzine tačke krutog tela je ortogonalna na radijus vektor u odnosu na trenutni centar rotacije. CC' je dijametar kruga pa je

$$y = CP \sin \theta = D \sin^2 \theta.$$

Situacija je prikazana na slici ispod. Ako razmotrimo perlu koja klizi niz krivu γ je $\int ds/v$. Kako je $v = \sqrt{2gy}$ (sledi iz zakona održanja energija i pretp. $v_A = 0$), sledi da je

$$k \int_{\gamma} \frac{ds}{\sqrt{y}}.$$

Minimizacija datog funkcionala daje

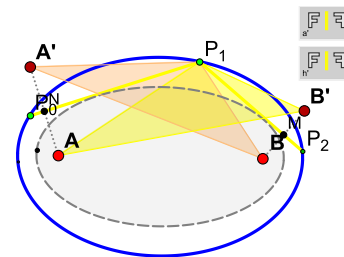
$$\frac{\sin \theta}{\sqrt{y}} = \text{const},$$

a to je upravo jednačina koju smo izveli za cikloidu.

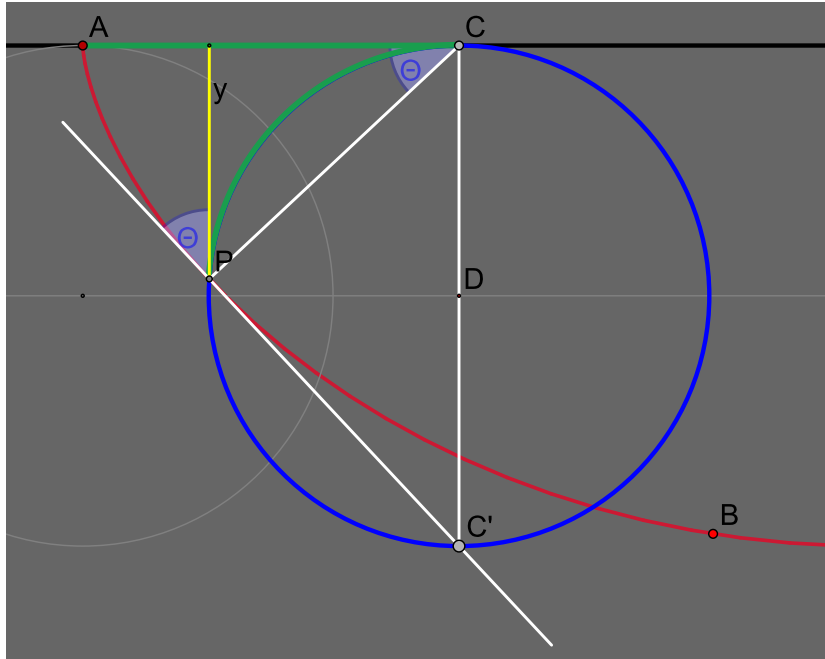
Naime, opšta klasa funkcionala, na koji se svode klasični problemi varijacionog računa, glasi

$$\int_a^b F(y) \sqrt{1 + y'^2} dx = \int_a^b F(y) ds.$$

Za njega važi sledeća teorema koja glasi:



Starogrčki: tauto, iso = isto



Slika 11.5: Cikloida kao tautohrona

funkcija $y = y(x)$ minimizuje prethodni funkcional uz $y(a) = A$, $y(b) = B$ ako važi

$$\frac{F(y)}{\sqrt{1+y'^2}} = \text{const.} \Leftrightarrow F(y) \sin \theta = \text{const.}$$

Formalni dokaz se bazira na primeni Neter teoreme, međutim, moguće ga je očigledno rešiti primenom mehaničke analogije ili optičke analogije (originalno Bernulijevo rešenje problema brahistohrone).

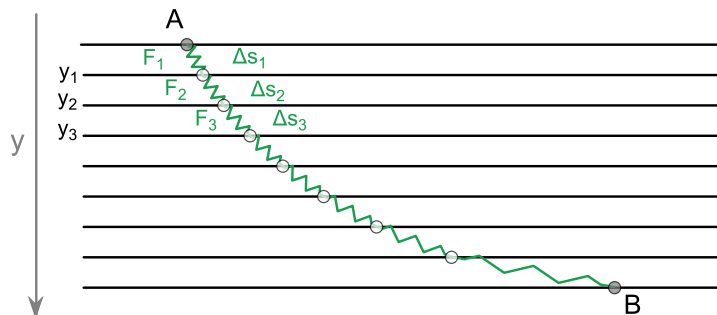
Primer 18: Varijacije - mehanička analogija

Pokazati da je cikloida tautohrona kriva pomoću mehaničke analogije.

Funkcional $\int_A^B F(y) ds$ diskretizujemo tako što diskretizujemo y -osu:

$$P_N = \sum F(y_k) \Delta s_k \approx \int_A^B F(y) ds.$$

Međutim, diskretizovani izraz predstavlja potencijalnu energiju sistema perli na horizontalnim nitima ($y = y_k$) spregnutih oprugama sa konstantnim naponom (slika 11.6). U ravnotežnoj konfiguraciji, potencijalna energija je minimalna, a dinamički uslov za to (jednakost



Slika 11.6: Mehanička varijaciona analogija.

horizontalnih komponentata sile) je

$$F(y_k) \sin \theta_k = F(y_{k+1}) \sin \theta_{k+1}, \quad k = 1 \dots N,$$

odnosno

$$F(y) \sin \theta = \text{const.}$$

Na slici 11.6 minimizovan je funkcional brahistohrone $F(y) = 1/\sqrt{y}$, pa je stabilna konfiguracija cikloidna.

Primer 19: Cikloida=izohrona 2

Pokazati da je cikloida izohrona kriva.

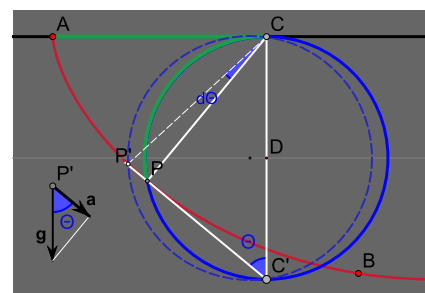
Parametar oscilovanja je dužina luka merena od dna cikloide s . Ideja je pokazati da se on ponaša kao LHO, tj. da važi

$$\ddot{s} \equiv \ddot{a} = -ks,$$

gde je k konstanta. Sobzirom da je $a = 0$ za $s = 0$, treba samo pokazati da je

$$da = -kds.$$

Sa skice na levoj strani slike vidimo da je $da = d(g \cos \theta) = -g \sin \theta d\theta$. Na desnoj strani slike je izveden izraz $ds = D \sin \theta d\theta$. Upoređivanjem, zaključujemo da je $k = g/D$, tj. cikloida je izohrona.



Slika 11.7: Cikloida kao izohrona

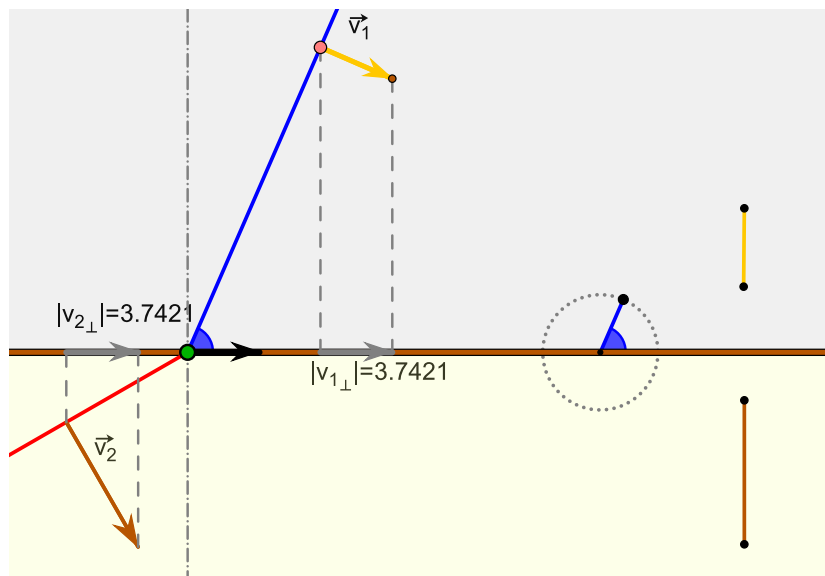
Primer 20: Snelijus-Dekartov zakon (zakon prelamanja)

Zakon prelamanja glasi: "Za svetlosni zrak koji prelazi granicu dve sredine, uglovi α_1 i α_2 , između zraka i normale na površ koja razdvaja sredine, je proporcionalna brzinama v_1 i v_2 svetlosti u datim sredinama:

$$\frac{\sin \alpha_1}{v_1} = \frac{\sin \alpha_2}{v_2}.$$

Drugim rečima, veličina $\frac{\sin \alpha}{v}$ je konstantna pri prelasku iz jedne sredine u drugu. Šta je fizički smisao veličine $\frac{\sin \alpha}{v}$?

Veličina $\frac{\sin \alpha}{v}$ je brzina tačke dodira talasnog fronta i granice među sredinama. Ova interpretacija je ilustrovana na slici ispod.



Slika 11.8: Zakon prelamanja i veličina $\frac{\sin \alpha}{v}$.

Zakon prelamanja je posledica Fermaovog principa na osnovu kojeg je putanja svetlosnog zraka između dve tačke, definisana uslovom minimalnog vremena.

Primer 21: Fermaov princip - kinematički

Spasilac se nalazi u tački A u trenutku kada je plivač u tački B potrebna pomoć. Ka kojoj tački na obali (X) treba da dotrči spasilac, da bi stigao do plivača za najkraće vreme.

Zadatak je moguće rešiti koristeći optičku analogiju i zakon prelamanja. Ovde je dato rešenje koje se zasniva na mehaničkoj analogiji.

Na slici je prikazan prsten X koji može da klizi bez trenja po horizontalnoj žici. Prsten je spojen sa tačkom A oprugom konstantnog napona $F_1 = 1/v_1$ i sa tačkom B sličnom oprugom napona $F_2 = 1/v_2$. Potencijalna energija sistema je

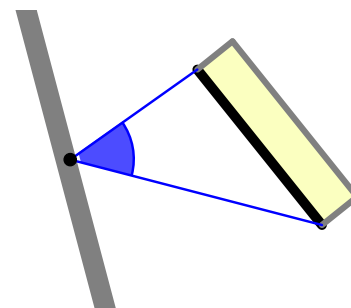
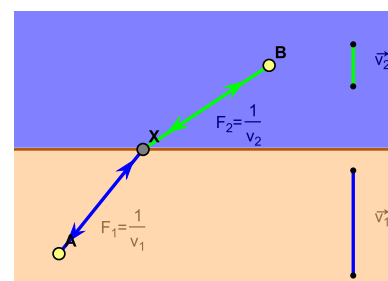
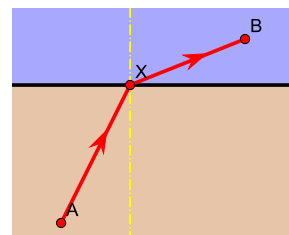
$$\frac{|A, X|}{v_1} + \frac{|B, X|}{v_2},$$

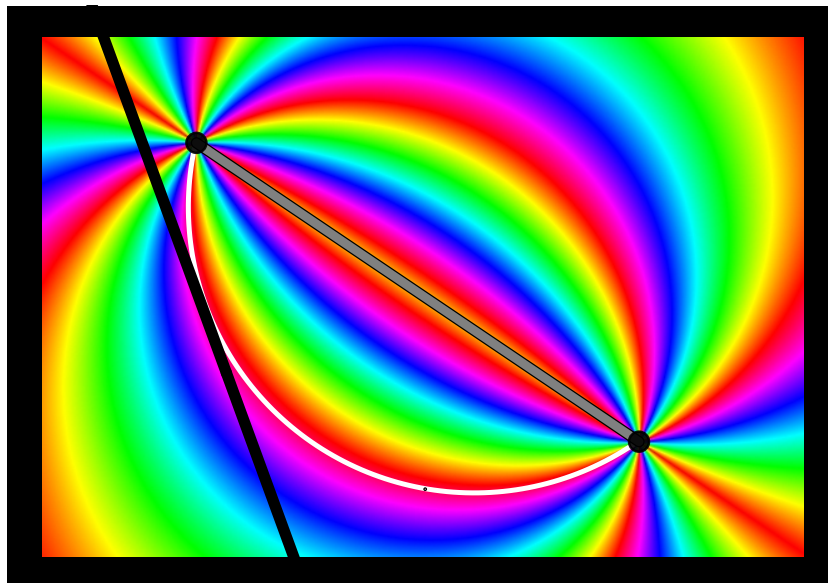
a to je izraz identičan izrazu za vreme puta od A do B . Skript:

```
// Draw slot
v1=2*|F,G|;
v2=2*|D,E|;
a=(A-C).xy;a1=|a|;
b=(B-C).xy;b1=|b|;
drawtext([2.5,1], "$F_2=\frac{1}{v_2}$", size->20);
drawtext([-3,-4], "$F_1=\frac{1}{v_1}$", size->20);
drawtext((D+E)/2+[1,0], "$\vec{v_2}$", size->20);
drawtext((F+G)/2+[1,0], "$\vec{v_1}$", size->20);
//Integration Tick slot
setforce(C, (1/(v1*a1))*a+(1/(v2*b1))*b);
```

Primer 22: Problem turističkog vodiča

Prednja fasada arhitektonski zanimljive zgrade se nalazi na pravoj koja je pod nekim uglom α u odnosu na pravolinijski put koji prolazi pored zgrade. Turistički vodič treba da odluči na kojem mestu na putu da zaustavi autobus tako da se prednja fasada iz te tačke najbolje vidi, tj. pod najvećim uglom. Rešiti zadatak geometrijski i mehaničkom analogijom.

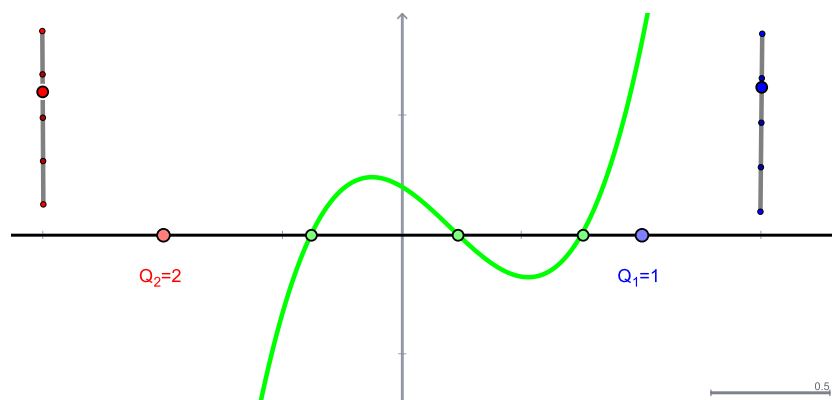




Slika 11.9: Geometrijsko rešenje problema turističkog vodiča. Prikazan je colorplot funkcije ugla $\angle AXB$ pod kojim se vidi duž AB iz proizvoljne tačke X . Ekviskalarne linije su kružni lukovi.

Primer 23: Jakobijevi polinomi i elektrostatika

Krajem XIX veka, Stiltjes je otkrio zanimljivu fizičku analogiju za nule Jakobijevih polinoma $P_n^{a,b}(x)$: ako su dve čestice, naelektrisanja q_1 i q_2 , fiksirane u tačkama sa koordinatama $(1,0)$ i $(-1,0)$ i, na pravolinijskoj žici između njih, n čestica jediničnog naelektrisanja. Ako je $a = 2q_2 - 1$ i $b = 2q_1 - 1$, tada se, u ravnotežnoj konfiguraciji, čestice nalaze u tačkama $(x_i, 0)$ gde su sa x_i označene nule Jakobijevog polinoma $P_n^{a,b}(x)$.



Odgovarajući skript je:

```
// Draw slot
```

```
j3(a,b,x):=(1/6)*(1+a)*(2+a)*(3+a)+(1/4)*(2+a)*(3+a)*(4+a+b)*(-1+x)+(1/8)*(3+a)*(4+a+b)*
```

```

(5+a+b)*(-1+x)^2+(1/48)*(4+a+b)*(5+a+b)*(6+a+b)*(-1+x)^3;

j4(a,b,x):=(1/24)*(1+a)*(2+a)*(3+a)*(4+a)+(1/12)*(2+a)*(3+a)*(4+a)*(5+a+b)*(-1+x)+
(1/16)*(3+a)*(4+a)*(5+a+b)*(6+a+b)*(-1+x)^2+(1/48)*(4+a)*(5+a+b)*(6+a+b)*(7+a+b)*(-1+x)^3+
(1/384)*(5+a+b)*(6+a+b)*(7+a+b)*(8+a+b)*(-1+x)^4;

//p=B.charge;q=A.charge;
q=round(|G,F|/|G,R|);
p=round(|K,H|/|K,S|);
ac=2*p-1;bc=2*q-1;
drawtext(A+[-0.1,-0.2],"$Q_2$="+q,size->20,color->[1,0,0]);
drawtext(B+[-0.1,-0.2],"$Q_1$="+p,size->20,color->[0,0,1]);

plot(0.2*j3(ac,bc,x),start->-1.2,stop->1.2,size->5,color->[0,1,0]);

//Integration Tick slot
// definicija sile medju cesticama
ac=abs(A.x-C.x);
cd=abs(C.x-D.x);
de=abs(D.x-E.x);
eb=abs(E.x-B.x);

setforce(C,0.1*(q*(1/ac)^2*(C-A)+(1/cd)^2*(C-D)+(1/(cd+de))^2*(C-E)+p*(1/(cd+de+eb))^2*(C-B)));
setforce(D,0.1*(q*(1/(ac+cd))^2*(D-A)+(1/cd)^2*(D-C)+(1/de)^2*(D-E)+p*(1/(de+eb))^2*(D-B)));
setforce(E,0.1*(q*(1/(ac+cd+de))^2*(E-A)+(1/(cd+de))^2*(E-C)+(1/de)^2*(E-D)+p*(1/eb)^2*(E-B)));

```

12 Dinamika jednodimenzionalnih sistema

Najjednostavniji mehanički sistemi su oni sa jednim stepenom slobode (ili kraće, 1D sistemi) poput materijalne tačke na pravoj liniji ili na krivoj. Jedini stepen slobode je trenutni položaj tačke na krivoj opisan koordinatom x u prvom slučaju, ili parametrom s , dužinom luka, u drugom. Njihov veliki značaj se temelji na činjenici da se svi *kompletno integrabilni sistemi* sa većim brojem stepeni slobode, svode na skup raspregnutih (dekuplovanih) sistema sa jednim stepenom slobode. Većinu fundamentalnih pojmova i ideja klasične mehanike je moguće ilustrovati na primeru 1D sistema.

Precizna definicija nalazhe dodatnu karakterizaciju takvih sistema kao *holonomnih* i *konzervativnih*. Pod holonomnim podrazumevamo da su jedina definisana ograničenja geometrijskog tipa, tj. da je položaj tačke u svakom trenutku na pravoj (ili krivoj), a da sva ograničenja na brzinu slede iz prethodnih, tj. da je brzina uvek tangenta na krivu (npr. uslov neproklizavanja kod kotrljanja spada u neholonomna ograničenja). Sve sisteme u ovom poglavlju smatramo konzervativnim. Konzervativnost podrazumeva važenje zakona održanja ukupne mehaničke, $E_u = T(v) + U(x)$ energije; drugim rečima, nema trenja između materijalne tačke i krive po kojoj se kreće. Kod konzervativnih sistema dinamika je potpuno definisana sa potencijalnom energijom $U(x)$. Sila koja deluje na materijalnu tačku u položaju x je tada data izrazom $F(x) = -dU(x)/dx$.

Primer 24: Osnovni modeli na x -osi

Kreirati Cinderella animaciju materijalne tačke na pravoj koja se kreće pod dejstvom a) Hukove opruge, sa potencijalnom $U(x) = kx^2/2$, b) "nelinearne opruge", sa potencijalnom $U(x) = p(x)$, gde je sa $p(x)$ označen polinom po promenljivoj x .

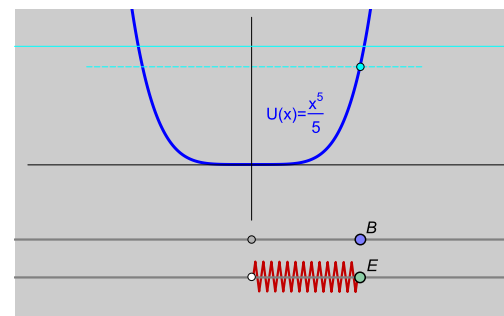
Primetimo da u CindyLab okruženju postoje već isprogramirani alati za Hukovu oprugu nulte i nenulte dužine, kao i modifikacije za Njutnovu i Kulonovu interakciju. Naime, možemo smatrati da između svake dve materijalne tačke koje međusobno interagiju, postoji virtualna opruga koja tu interakciju omogućava. Svaka dvočestična interakcija definisana trenutnim međučestičnim rastojanjem x , i u slučaju Hukove opruge, ona je jednaka kx , gde k označava krutost opruge. Ako definišemo krutost opruge kao funkciju $k = k(x)$, tada je moguće opisati proizvoljnu interakciju.

Hukova opruga nulte dužine je ona kod koje je sila zatezanja srazmerna dužini opruge, $F = kx$. Kod standardne opruge, nenulte dužine, sila je $F = k(x - x_0)$, i proporcionalna je veličini istezanja odnosno sabijanja opruge.

Drugi način, numerički neprecizniji od prethodnog, je upotrebom komande `setforce`. Oba rešenja su prikazana na slici. Tačka B se kreće pod dejstvom komande, a tačka E kao modifikovana Hukova opruga nulte ravnotežne dužine. Plave linije prikazuju referentne nivoe početne i trenutne potencijalne energije.

Odgovarajući skript je:

```
//Draw slot
n=5;
pot(x):=1/n*x^n;
b.strength=(abs(B.x))^(n-2);
f(x):=-d(pot(#),x);
plot(1+abs(pot(x)),start->-4,stop->4,size->4);
B.y=0;
D.xy=[B.x,1+abs(pot(B.x))];
F.xy=[-B.x,1+abs(pot(B.x))];
drawtext([0.2,1.6], "U(x)=$\frac{x^n}{n}$", size->22, color->[0,0,1]);
//Integration Tick slot
ab=abs(A.x-B.x);
vecn=(B-A)/ab;
setforce(B,1*vecn*f(ab));
//Simulation Strat slot
x=B.x;
```

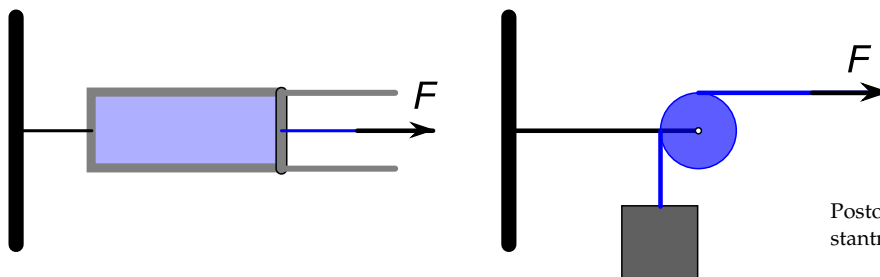


Slika 12.1:

Primer 25: Opruga konstantne sile zatezanja

Oprugu konstantne sile zatezanja je lako modelirati - odgovarajući potencijal je linearan. Koji sistemi poznati u fizici se ponašaju kao opruga konstantne sile.

Jedna mogućnost je gasni klip na konstantnoj temperaturi, a drugi teg na koturači. Oba su ilustrovana na slici 12.2.



Slika 12.2:

Postoji i 2D analogija opruge konstantne sile - to je film od sapunice!

Dinamika 1D sistema se često modelira perlom koja klizi bez trenja duž žice u obliku grafika funkcije $f(x)$ u homogenom gravitacionom polju. Treba napomenuti da funkcija $f(x)$ nije ekvivalentna potencijalnoj energiji $U(x)$ - ona važi samo u oblastima u kojima je nagib $U'(x)$ odnosno $f'(x)$ dovoljno mali.

Primer 26: Osnovni modeli na krivoj

Kreirati Cinderella animaciju materijalne tačke koja se kreće duž krive pod dejstvom vertikalnog homogenog gravitacionog polja u slučaju kada je kriva a) kružnica (matematičko klatno), b) cikloida (Hajgensovo klatno).

Najpre odredimo jednačinu kretanja perle duž žice proizvoljnog oblika. Relevantne sile su prikazane na slici 12.3.

Iz Njutnovog dinamičkog (II) zakona $ma = F_r$, činjenice da je rezultujuća sila zapravo tangencijalna F_t i da je $a = \ddot{s}$, gde je s sa s je označena dužina luka krive, sledi jednačina

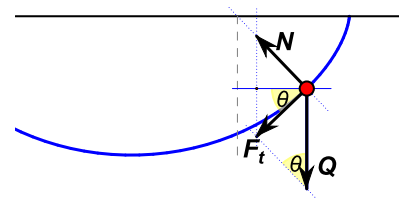
$$\ddot{s} = -g \sin(\theta(s)).$$

Ako analiziramo kretanje oko konkretnog lokalnog minimuma krive $y = f(x)$, jednačinu je moguće formulisati i tako da funkcija položaja bude ugao $\theta = \theta(s)$.

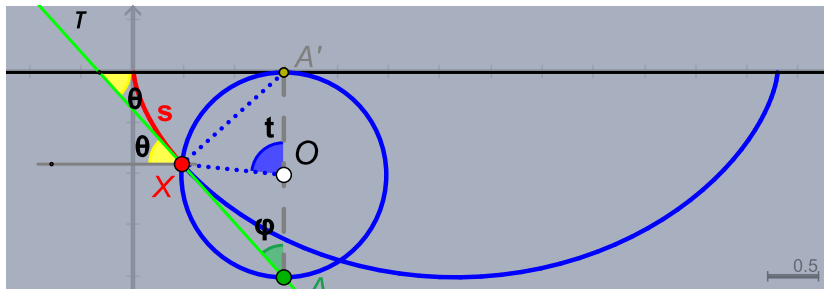
(a) U slučaju kružnice, položaj tačke na njoj je jednoznačno definisan uglom θ i važi $s = R\theta$, gde je R poluprečnik kružnice. Jednačina kretanja je dobro poznata:

$$\ddot{\theta} = -\frac{g}{R} \sin(\theta).$$

(b) Kada je u pitanju cikloida, najpre je potrebno dužinu luka povezati sa parametrom cikloide i uglom θ . Parametri cikloide t, ϕ, θ su prikazani na slici. Obično se jednačina cikloide zadaje preko pa-



Slika 12.3:



Slika 12.4: Parametri cikloide

rametra t koji opisuje ugao za koji se zarotirala kružnica (generatrisa cikloide) i glasi:

$$(x(t), y(t)) = r(t - \sin(t), \cos(t) - 1).$$

Uglovi ϕ i θ su ulovi koje zaklapa tangenta τ sa vertikalom i horizontalom. Treba primetiti da tangenta u tački X na cikloidu uvek prolazi i kroz tačku A na generatrisi, koja je dijametralno suprotna od tačke A' , trenutnog centra rotacije generatriše. Sa slike su očigledne relacije $t = 2\phi$ i $\theta = \pi/2 - \phi$. Prema tome parametar θ koji figuriše u izvedenoj jednačini kretanja je uvek moguće zameniti i sa t i ϕ .

Potrebno je još rešiti važan problem veze između dužine luka cikloide i ugaonih parametara. Izračunajmo najpre vezu između dužine luka s i parametra t cikloide.

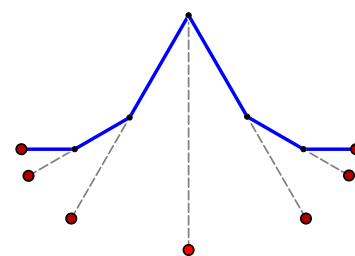
$$\begin{aligned} s &= \int_0^t \sqrt{\left(\frac{dx}{d\tau}\right)^2 + \left(\frac{dy}{d\tau}\right)^2} d\tau \\ &= \int_0^t r\sqrt{2 - 2\cos\tau} d\tau \\ &= 2r \int_0^t \sin\frac{\tau}{2} d\tau \\ &= 4r \left(1 - \cos\frac{t}{2}\right). \end{aligned}$$

Naravno, sada je moguće direktnom zamenom napisati i rešiti jednačinu kretanja za perlu na cikloidi. Ali, umesto direktne zamene, u sledećem zadatku primenićemo drugi metod..

Godine 1657., holandski matematičar Kristian Hajgens (C. Huygens) pokušavao je da napravi dovoljno precizan sat. Mersen (M. Mersenne) mu je sugerisao da koristi matematičko klatno kao kontrolni uređaj. Hajgens je znao da to nije dobro rešenje, jer period matematičkog klatna zavisi od amplitude: klatna sa većom amplitudom su sporija. Da bi sinhronizovao dva klatna sa različitim amplitudama, Hajgens je pomoću niza eksera primorao klatna da odstupe od kružne putanje. Njihova putanja je bila deo-po-deo kružna ali sa sve manjom dužinom niti. Ova, relativno gruba slika se pojavljuje u njegovoj prvoj knjizi o satovima *Horologium*, objavljenoj 1658. godine. Da bi postigao što bolju sinhronizaciju, on je eksere koji ograničavaju putanju klatna sve gušće ukucavao sve dok ekseri nisu počeli da dobijaju formu neprekidne krive. Pretpostavio je i na kraju uspeo da dokaže da je tražena kriva upravo cikloida. Na taj način je dokazao tautohrono svojstvo cikloide. Rezultat je objavio u svojoj drugoj knjizi *Horologium Oscillatorium* iz 1673. godine.

Digresija o istoriji osobina cikloide koja se nastavlja na priču započetu u drugoj glavi. Sve do 50-tih godina 17. veka problem rektifikacije, tj. određivanja duži čija je dužina jednaka dužini luka zadate krive, je smatran nerešivim. Dekart je u svojoj drugoj knjizi o geometriji, napisao da veza dužina pravih i krivih linija nije, niti će ikada biti poznata! Sve do 1658. kada je ser Kristofer Ren (Christopher Wren), čuveni Engleski arhitekta i Njutnov prijatelj, poslao Paskalu pismo sa tvrdjenjem da je luk cikloide tačno 8 puta duži od radijusa kružnice koja je generiše. Njegov dokaz je kasnije publikovao Volis u svom traktatu o cikloidi. Dokaz je izveden primenom beskonačnih redova.

Za izračunavanje dužine luka cikloide bez primene integrala pogledati treću glavu knjige, posvećenu teoremi o brišućoj tangenti



Slika 12.5: Hajgensovo "grubo" cikloidno klatno

Primer 27: Izohrono svojstvo cikloide

Pokazati da je cikloida izohrona kriva.

Izohrona kriva je ona duž koje je period oscilovanje perle sa proizvoljnim početnim položajem - amplitudom, jednak. Znamo da je to odlika linearnog harmonijskog oscilatora (LHO), tj. to je posledica linearnosti odgovarajuće jednačine. Ideja je pokazati da se perla na cikloidi ponaša kao LHO, odnosno da važi

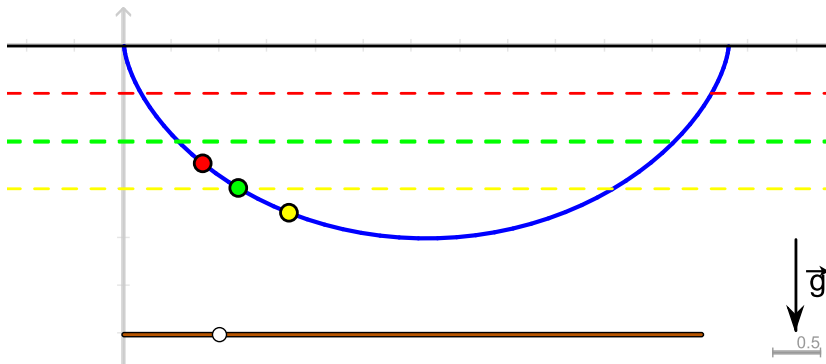
$$\ddot{s} \equiv a = -ks,$$

gde je k konstanta. S obzirom da je $a = 0$ za $s = 0$, treba samo pokazati da je

$$da = -kds.$$

Sa skice na levoj strani slike vidimo da je $da = d(g \cos \theta) = -g \sin \theta d\theta$. Na desnoj strani slike je izveden izraz $ds = D \sin \theta d\theta$. Upoređivanjem, zaključujemo da je $k = g/D$, ili drugim rečima, cikloida je izohrona.

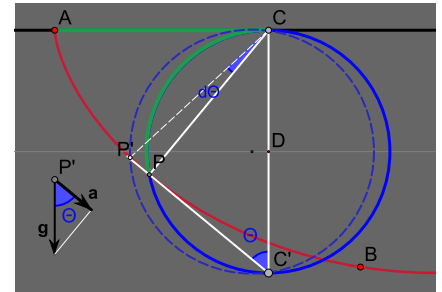
Sada imamo sve elemente da završimo zadatak sa animacijom cikloidnog klatna. Animacija je prikazana na slici, zajedno sa pratećim skriptom.



```

\\Draw slot
r(t):=[t-sin(t),cos(t)-1];
s(t):=4*(1-cos(t/2));
//definicija parametra na klizacu
p=2*pi*|F,D|/|E,D|;
//parametrizacija pocetnog nivoa
A.xy=r(p);
Q.xy=r(1.5*p);
R.xy=r(2*p);

```



Slika 12.6: Cikloida kao izohrona

Slika 12.7: Demonstracija tautohronog svojstva cikloide

```

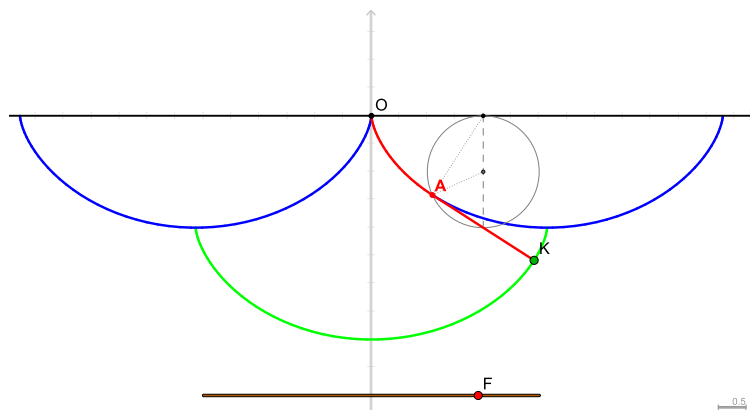
C.x=p;
//parametrizacija trenutnog položaja tacaka
K.xy=r(2*arccos(cos(p/2)*cos(tm)));
L.xy=r(2*arccos(cos(1.5*p/2)*cos(tm)));
M.xy=r(2*arccos(cos(2*p/2)*cos(tm)));
plot(r(t),start->0,stop->2*pi,color->[0,0,1],size->3);
//Timer Tick slot
tm=5*simulationtime();
// Simulation Stop slot
//podesavanje pocetnih parametara
tm=0;
K.xy=r(p);
L.xy=r(1.5*p);
M.xy=r(2*p);

```

Primer 28: Involuta cikloide

Demonstrirati auto-involutivno svojstvo cikloide.

Rešenje je prikazano na slici 12.8, sa pratećim skriptom ispod.



Prilikom dokazivanja da je tražena tautohrona kriva zapravo cikloida, Hajgens je 1659. godine dokazao još jedno važno svojstvo cikloide: cikloida je sama sebi involuta. Involuta krive (generatriše) je kriva koju iscrtava kraj neistegljive niti, čiji je početak fiksiran u nekoj tački generatriše, koja se namotava na istu tu generatrisu.

Slika 12.8: Involuta cikloide

```

r(t):=[t-sin(t),cos(t)-1]; s(t):=4*(1-cos(t/2));
p=2*pi*F.x/6; A.xy=r(p); C1.radius=4-s(p);
plot(r1(t),start->-2*pi,stop->0,color->[0,1,0],size->3);
if(p<=0,plot(r(t),start->-2*pi,stop->p+0.01,color->[0,0,1],size->3);
  plot(r(t),start->p-0.01,stop->0,color->[1,0,0],size->3);
  plot(r(t),start->0,stop->2*pi,color->[0,0,1],size->3),
  plot(r(t),start->-2*pi,stop->0,color->[0,0,1],size->3);
  plot(r(t),start->0,stop->p+0.01,color->[1,0,0],size->3);
  plot(r(t),start->p-0.01,stop->2*pi,color->[0,0,1],size->3));

```

Konzervativne jednodimenzionalne sisteme možemo efikasno geometrijski prikazati primenom faznog prostora i vektorskih polja u njima. Takav pristup rešavanju diferencijalnih jednačina, i to ne samo mahaničkih sistema, spada u *teoriju dinamičkih sistema*.

Primer 1: LHO

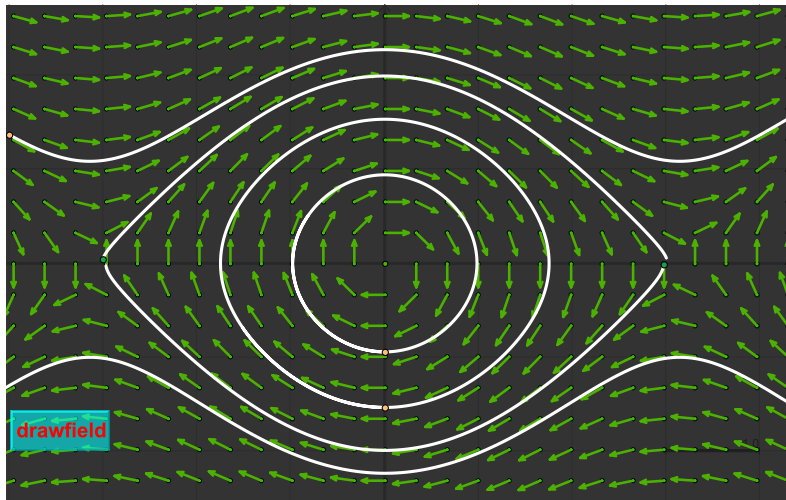
Kreirati fazni portret sa odgovarajućom animacijom za linearni harmonijski oscilator.

Videti rešenje prvog primera u poglavlju o diferencijalnim jednačinama drugog reda.

Primer 2: Fazni portret za matematičko klatno

Kreirati fazni portret sa odgovarajućom animacijom za matematičko klatno.

Potencijalna energija je $V(x) = -\cos(x)$ i periodična je sa periodom 2π . U tački sa minimumom potencijalne energije imamo stabilnu fiksnu tačku (centar), u tački sa maksimumom, nestabilnu (hiperbolička fiksna tačka - sedlo). U okolini stabilne fiksne tačke faze trajektorije su dobro aproksimirane elipsama, a one se udaljavanjem od stabilne i približavanjem nestabilnoj fiksnoj tački sve više deformišu i pretvaraju u neanalitičku krivu. Prisutne su i periodične trajektorije drugog tipa (brzina im ni u jednoj tački nije nulta) koje se dobijaju za početne brzine dovoljne da u zbiru sa potencijalnom prevaziđu maksimum potencijalne energije.



Slika 12.9: Fazni portret matematičkog klatna

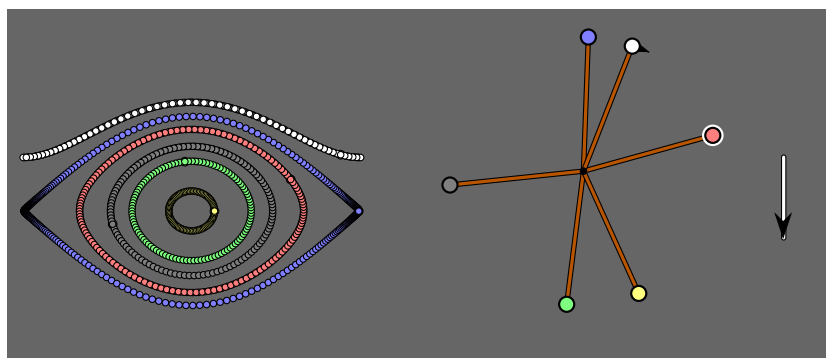
Skript:

```
// Draw slot
//crtanje resetke
rot(f):=[[cos(f),-sin(f)],[sin(f),cos(f)]];
rots(f,pt1,pt2):=[pt1,pt1+rot(f)*(pt2-pt1)];
lat=0.33*directproduct(-12..18,-9..9);
if(Text0.pressed==false,
drawall(lat,size->2));
f(v):=[v_2,-sin(v_1)];

//crtanje vektorskog polja na resetci
matf(f,pt):=if(pt==[0,0],[0,0],[pt,pt+f(pt)/|f(pt)|*0.25]);
str(f,vec):=if(vec==[[0,0],[0,0]],[[0,0],[0,0]],
[vec,rots(f,vec_2,0.25*vec_1+0.75*vec_2),rots(-f,vec_2,0.25*vec_1+0.75*vec_2)]);
if(Text0.pressed==false,
repeat(length(lat),i,drawall(str(pi/8,matf(f,lat_i)),size->2.5,color->(0.3,0.7,0))));

//num. resavanje jednacine i crtanje int. krivih
nn=750;h=0.01;
y0=[C.xy,D.xy,E.xy,F.xy];
yc=y0;y=y0;
repeat(length(y0),i,
  repeat(1.2*nn,
    c1=h*f(y_i);
    c2=h*f(y_i+c1/2);
    c3=h*f(y_i+c2/2);
    c4=h*f(y_i+c3);
    y_i=y_i+(c1+2*c2+2*c3+c4)/6;
  yc_i=append(yc_i,y_i);
  connect(yc_i,color->(1,1,1),size->3));
```

Drugo rešenje, fazni portret prikazan na osnovu animacije u CindyLab-u je prikazan na slici desno.



Odgovarajući skript:

```
//Draw slot
pts=[B,F,G,E,K,H];\\ oznake materijalnih tacaka koje osciluju
repeat(length(pts),i,
  draw(2*[if(
    mod((pts_i).angle-3*pi/2,2*pi)>pi,
    -2*pi+mod((pts_i).angle-3*pi/2,2*pi),
    mod((pts_i).angle-3*pi/2,2*pi)),
    8*(pts_i).vx/((pts_i)-A).y],
    size->4,color->(pts_i).color)
  );
repeat(length(pts),i,
  l_i=l_i++[2*[if(mod((pts_i).angle-3*pi/2,2*pi)>pi,
    -2*pi+mod((pts_i).angle-3*pi/2,2*pi),
    mod((pts_i).angle-3*pi/2,2*pi)),8*(pts_i).vx/((pts_i)-A).y]];
  if(length(l_i)>(if(i==1,99,55))*i,l_i=l_i--[l_i-1]);
  drawall(l_i,color->(pts_i).color,size->3)
  );
// SimulationStart slot
l=[[[]],[[]],[[]],[[]],[[]],[[]]];
```

Primer 3: Fazni portret konzervativnog polinomijalnog sistema

Kreirati fazni portret sa animacijom za proizvoljan konzervativni sistem sa potencijalnom energijom opisanom polinomom a) trećeg, b) četvrtog stepena.

Dinamičko ponašanje linearnog harmonijskog oscilatora je karakteristično za dinamiku čestice sa jednim stepenom slobode u proizvoljnom konzervativnom polju. Posmatrajmo pravolinijsko kretanje čestice mase m , duž x -ose pod dejstvom nelinearne sile $F(x)$. Diferencijalna jednačina kretanja, prema II Njutnovom zakonu, je

$$m\ddot{x} = F(x).$$

Ako je sila F konzervativna, tada postoji potencijalna energija V takav da je $F(x) = -dV/dx$, pa jednačina kretanja dobija oblik:

$$m\ddot{x} + \frac{dV}{dx} = 0,$$

i u opštem slučaju je nelinearna i najčešće nije analitički rešiva. Posle množenja obe strane jednačine sa \dot{x} i daljeg sređivanja, dobija se izraz:

$$\frac{d}{dt} \left[\frac{m\dot{x}^2}{2} + V(x) \right] = 0.$$

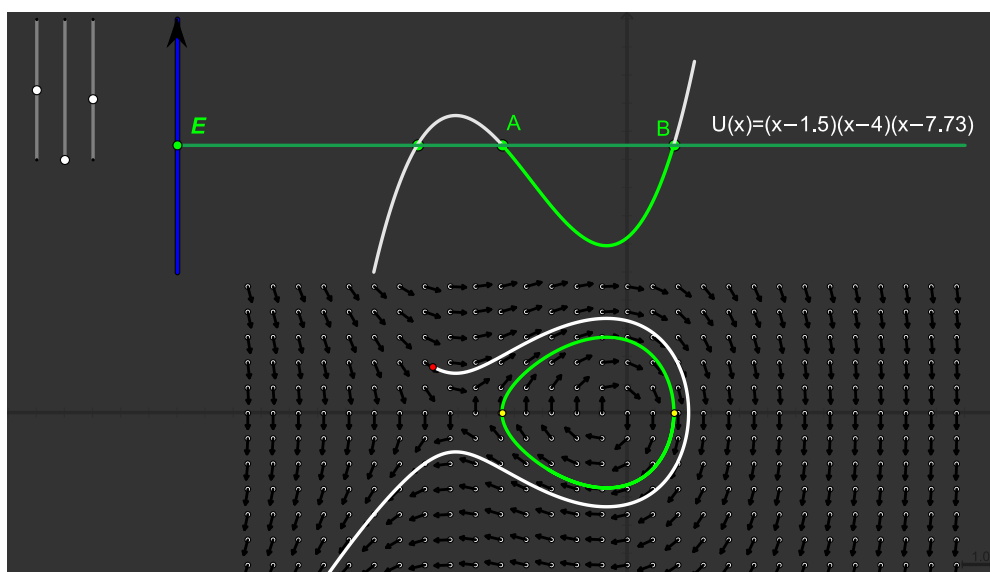
Član $m\dot{x}^2/2$ predstavlja kinetičku energiju E_k , a V potencijalnu. Iz čega sledi izraz:

$$E_k + V = E = \text{const},$$

a to je karakteristika konzervativnih sistema. Drugim rečima, konzervativni sistemi su oni kod kojih se energija ne menja sa vremenom - važi zakon održanja mehaničke energije. Na ovaj način praktično smo integralili polaznu jednačinu i dobili diferencijalnu jednačinu prvog reda (nelinearnu). Za dato rešenje $x(t)$, ukupna energija sistema

$$E = \frac{m\dot{x}^2}{2} + V(x)$$

je funkcija samo položaja čestice, a ne i vremena.



Na slici je prikazan fazni portret konzervativnog sistema sa potencijalnom energijom opisanom polinomom trećeg stepena. Zapažamo da se u okolini minimuma potencijalne energije skoro ne razlikuje od faznog portreta linearnog harmonijskog oscilatora. Tu oblast nazivamo oblašću malih oscilacija. U samom minimumu se nalazi fiksna tačka (tipa centar). Međutim, udaljavanjem od lokalnog minimuma, fazne trajektorije sve više odstupaju od eliptičnog oblika karakterističnog za linearne sisteme, pa se uticaj nelinearnosti ogleda u postepenoj deformaciji elipse u ovaloidni oblik. Druga razlika je posledica postojanja lokalnog maksimuma potencijalne energije $V(x)$. U samoj tački maksimuma ponovo imamo fiksnu tačku, ali sada drugog tipa, hiperboličkog. U okolini hiperboličke tačke orbite nisu periodične. Odgovarajući skript:

```

//Draw slot
rot(f):=[[cos(f),-sin(f)],[sin(f),cos(f)]];
rots(f,pt1,pt2):=[pt1,pt1+rot(f)*(pt2-pt1)];
matf(g,pt):=[pt,pt+g(pt)/|g(pt)|*0.55];
str(g,vec):=if(vec==[[0,0],[0,0]],[[0,0],[0,0]],
[vec,rots(g,vec_2,0.25*vec_1+0.75*vec_2),rots(-g,vec_2,0.25*vec_1+0.75*vec_2)]);

lat=0.9*directproduct(-15..13,-8..5);
drawall(lat,size->2.5,color->(1,1,1));

En=R.y;
k1=1+|D,N|/|D,E|;
k2=-3+|F,0|/|F,G|;
k3=-5+2*|H,Q|/|H,K|;

u(x):=9+0.06*(x-k1)*(x-2*k2)*(x-2*k3)-En;

// odredjivanje nula polinoma za dati nivo Ep
rt=roots([9-En-0.24*k1*k2*k3,0.12*k1*k2+0.12*k1*k3+0.24*k2*k3,-0.06*k1-0.12*k2-0.12*k3,0.06]);
repeat(length(rt),i,draw([rt_i,En],size->5));
C.xy=[rt_3,0];
L.xy=[rt_2,0];
drawtext([rt_2,En]+[0.2,0.5],"A",size->19,color->[0,1,0]);
drawtext([rt_3,En]+[-0.6,0.3],"B",size->19,color->[0,1,0]);

// crtanje grafika pot. energije
drawtext([3,10],"$U(x)=$"+$(x$"+(-k1)+"$)(x$"+2*k2+"$)(x$"+2*k3+"$)$",size->19,color->[1,1,1]);
plot(u(x)+En,start->-9,stop->rt_2,size->3,color->(0.9,0.9,0.9));
plot(u(x)+En,start->rt_2,stop->rt_3,size->3,color->(0,1,0));
plot(u(x)+En,start->rt_3,stop->2.4,size->3,color->(0.9,0.9,0.9));
repeat(length(lat),i,drawall(str(pi/6,matf(g,lat_i)),size->2.5,color->(0,0,0)));

// num. resenje
x0=C.xy;
nn=1000;h=0.01;xc=x0;y=x0;
repeat(nn,
  c1=h*g(y);
  c2=h*g(y+c1/2);
  c3=h*g(y+c2/2);
  c4=h*g(y+c3);
  y=y+(c1+2*c2+2*c3+c4)/6;
xc=append(xc,y));
connect(xc,color->(0,1,0),size->3);
x00=M.xy;

```

```
xcc=x00;y=x00;  
repeat(nn,  
  c1=h*g(y);  
  c2=h*g(y+c1/2);  
  c3=h*g(y+c2/2);  
  c4=h*g(y+c3);  
  y=y+(c1+2*c2+2*c3+c4)/6;  
xcc=if(abs(y_2)<=8,append(xcc,y),xcc));  
connect(xcc,color->(1,1,1),size->3);
```

Primer 4: Fazni portret za Hajgensovo klatno

Kreirati fazni potret sa odgovarajućom animacijom za Hajgensovo klatno.

13 Diferencijalne jednačine i dinamički sistemi

Diferencijalne jednačine su osnovni matematički alat za izučavanje prirodnih zakona. Njihov značaj je prvi shvatio Isak Njutn (I. Newton). Smatrao ih je toliko bitnim, da je u pismu sekretaru Kraljevskog društva Oldenburgu svoje najvažnije otkriće zašifrovao u vidu anagrama, koji u savremenom, slobodnom prevodu glasi: *Zakoni prirode izraženi su diferencijalnim jednačinama*. Svoje drugo veliko otkriće je takođe zašifrovao, a ono se odnosi na univerzalnu tehniku rešavanja diferencijalnih jednačina razvojem nepoznate funkcije u Tejlorov red.

Prvi Njutnov anagram aktuelan je i u naše vreme, dok je drugi umnogome prevaziđen. Tehnike rešavanja i analize diferencijalnih jednačina znatno su napredovale i svrstavaju se u tri velike klase: analitičke (egzaktne), numericke (približne) i kvalitativne. Kvalitativna teorija diferencijalnih jednačina ili kako se poslednjih decenija obično naziva, teorija dinamičkih sistema, nastala je na prelasku iz devetnaestog u dvadeseti vek u redovima Poenkarea. On je uvideo da nam u primenama egzaktno rešenje obično nije ni potrebno. Štaviše, ako je prezentovano preko složenih analitičkih izraza, često otežava analizu. Cilj analize je obično asimptotsko ili stacionarno ponašanje rešenja, a njega je znatno lakše razumeti ukoliko rešenje interpretiramo geometrijski i to je bila osnovna Poenkareova ideja.

Savremena praksa rada sa diferencijalnim jednačinama je, kao uostalom i čitava matematička praksa, obeležena primenom lako dostupnih računara i odgovarajućih programa. Stoga se danas pod rešavanjem jednačina obično podrazumeva primena neke od programski implementiranih numeričkih tehnika. Programi poput Mathematica ili Maple velikom brzinom rešavaju većinu praktično potrebnih jednačina. Ali takav način rešavanja ne omogućava i razumevanje rešenja, a samim tim ni pojave opisane jednačinom. Stoga je, u slučajevima kada je osnovni cilj razumevanje, a ne samo rešenje, najbolje koristiti kvalitativni pristup. Međutim, i kvalitativni metod je moguće automatizovati primenom računara, posredstvom programa za dinamičku geometriju.

13.1 Diferencijalne jednačine prvog reda

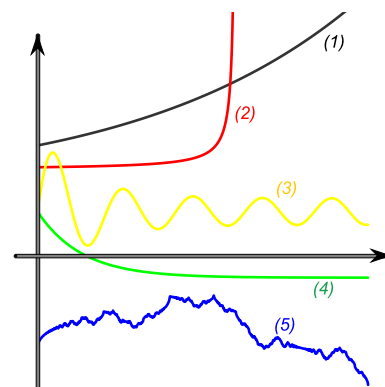
Najjednostavnije diferencijalne jednačine su jednačine prvog reda. Sve jednačine prvog reda možemo zapisati prostom formulom

$$\dot{x}(t) = f(t, x),$$

gde je f poznata funkcija nezavisne promenljive t i nepoznate funkcije $x(t)$. Pod rešenjem se podrazumeva funkcija $x(t)$ koja zamenom u prethodni izraz daje tačnu jednakost.

Geometrizacija diferencijalnih jednačina olakšava kvalitativnu analizu rešenja, a numeričko rešenje postaje očigledno. Cilj kvalitativne analize je da odgovori na pitanje o ponašanju rešenja kad $t \rightarrow \infty$. Mogući slučajevi su:

1. rešenje x teži beskonačnosti za beskonačno vreme,
2. rešenje x teži beskonačnosti za konačno vreme,
3. rešenje x nakon nekog vremena postaje periodična funkcija,
4. rešenje x teži ka konačnoj stacionarnoj vrednosti,
5. rešenje x je potpuno iregularno.



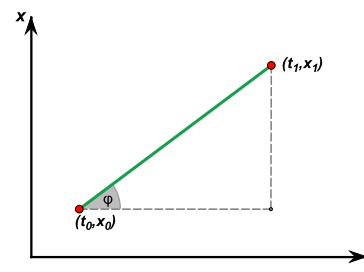
Jednačine prvog reda imaju jednostavnu geometrijsku interpretaciju preko *polja pravaca*. Obično je dovoljno pogledati polje pravaca pa da se zaključi o tipu ponašanja rešenja. Sa druge strane, najjednostavniji metod numeričkog rešavanja jednačina se praktično svodi na na povezivanje pojedinih tačaka u (t, x) ravni duž polja pravaca.

13.1.1 Polje pravaca

Preslikavanje tipa $f : \mathbb{R}^2 \mapsto \mathbb{R}$, koje potpuno definiše jednačinu prvog reda, se u fizici obično naziva *skalarno polje* i prikazuje trodimenzionalnim ili konturnim ("ekvipotencijalne linije") plotom.

Polje pravaca je nova interpretacija skalarnog polja. Naime, svakoj tački u ravni (t, x) pridružen je jedan broj kojim je definisan nagib (tangens ugla u odnosu na x osu) prave, tj. pravac. Pravac obično prikazujemo kao neorjentisanu duž. Dakle, polje pravaca je pravilo kojim je svakoj tački ravni pridružena neorjentisana duž. U tački (t_0, x_0) polje pravaca ima vrednost $f(t_0, x_0)$, pa je odgovarajuća duž pod uglom $\varphi = f(t_0, x_0)$ u odnosu na osu x .

Rešenje jednačine je sada lako interpretirati: to je svaka kriva koja je u svakoj tački (t, x) tangentna na odgovarajući pravac polja (tj. duž kroz (t, x)). Takve krive nazivamo *integralne krive*. Kroz proizvoljnu tačku prolazi tačno po jedna integralna kriva i to je posledica teoreme o jedinstvenosti rešenja.



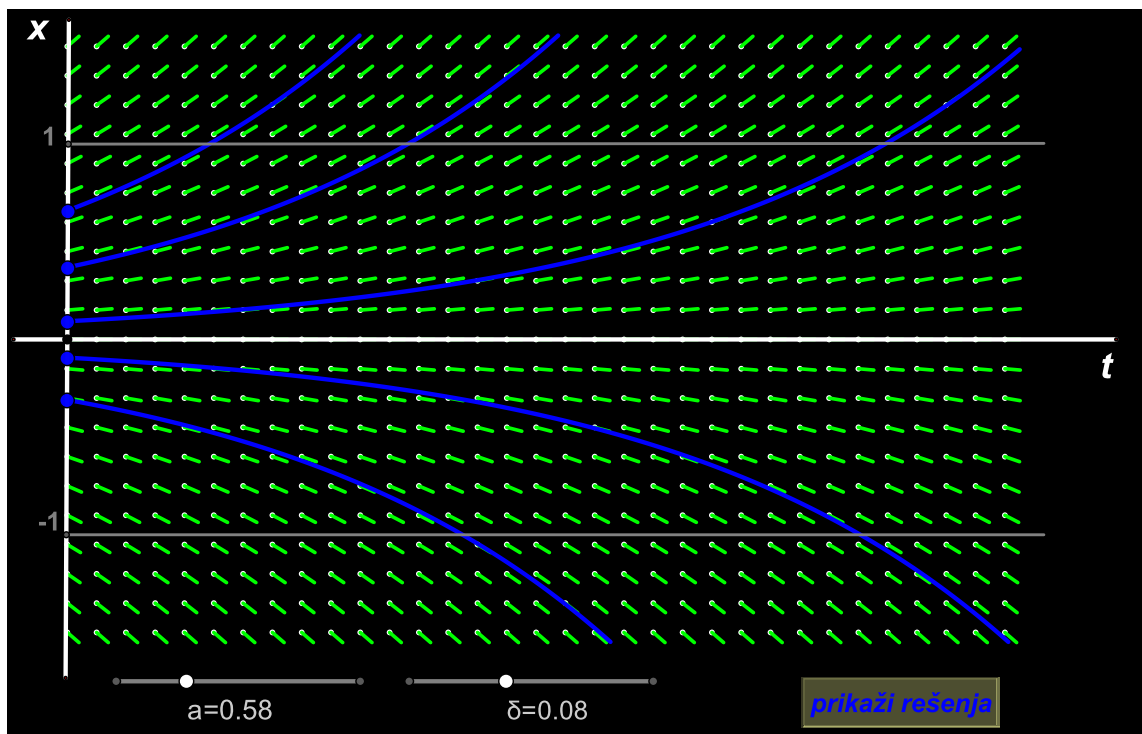
Teorema o jedinstvenosti rešenja

Za diferencijalnu jednačinu oblika $\dot{x}(t) = f(t, x)$, gde je funkcija $f(t, x)$ glatka (neprekidno diferencijabilna) po t i x važi: kroz svaku tačku (t, x) prolazi jedinstvena integralna kriva; ako kroz tačku (t, x) prolaze dve integralne krive, onda se one podudaraju.

Primer 1: Polje pravaca

Prikazati polja pravaca zajedno sa nekoliko integralnih krivih, za jednačinu $\dot{x}(t) = a x(t)$.

Navedena jednačina razdvaja promenljive i njeno rešenje se dobija direktnom integracijom: $x(t) = x_0 e^{ax}$, pri čemu je vrednost konstante x_0 određena početnim uslovima ($x_0 = x(0)$). Polje pravaca sa integralnim krivama je prikazano na slici. Sa slike je očigledno da



rešenja ove jednačine pripadaju tipu (1) : teže beskonačnosti za beskonačno vreme, izuzev rešenja koje prolazi kroz koordinatni početak, koje je konstantno, pa pripada tipu (4). Štaviše, možemo odmah konstatovati da rešenje autonomne jednačine, tj. one koja ne zavisi eksplicitno od vremena $f(t, x) = f(x)$, ne može pripadati tipu (4) i

(5). Za to je potrebno da se polje pravaca na pravim linijama (t, x_0) menja duž prave, a to je nemoguće jer ne zavisi od t .

Odgovarajući skript je

```
//osnovne definicije
f(t,x):=a*x;//polje pravaca
a=2*|F,B|/|B,C|;
del=0.2*|G,D|/|E,D|;//parametar koji def. duzinu duzi
x00=[H.y,S.y,T.y,U.y,V.y]);//pocetni uslov

//ispis trenutnih vrednosti parametara a i delta
drawtext(F.xy+[0.01,-0.2],"$a=$"+text(format(a,2)),size->19,color->[0.8,0.8,0.8]);
drawtext(G.xy+[0.01,-0.2],"$\delta=$"+text(format(del,2)),size->18,color->[0.8,0.8,0.8]);

//crtanje polja pravaca
l=0.15*directproduct(0..32,(-10..10));//l je skup pocetnih tacaka
lp=l;//lp je skup duzi - pravaca
drawall(l,color->(1,1,1),size->2.5);//crta pocetne tacke
repeat(length(l),i,
  xx=[l_i_1+del,del*f(l_i_1,l_i_2)+l_i_2]);//u racuna poloazaje krajnjih tacaka
  xx=l_i+del*((xx-l_i)/|xx-l_i|);// normira duzine duzi na del
  lp_i=[l_i,xx]);//formira duz sa pocetnom i krajnjom tackom
drawall(lp,size->2.5,color->(0,1,0));//crta duzi

//plotovanje resenja
sol1(i,x):=(x00_i)*exp(a*x);
if(Text1.pressed,repeat(length(x00),i,plot(sol1(i,x),
start->0,stop->min(0.75*del+(log(1.5/(x00_i))/a),4.8+del),size->3)));
```

Primer 2: Polje pravaca 2

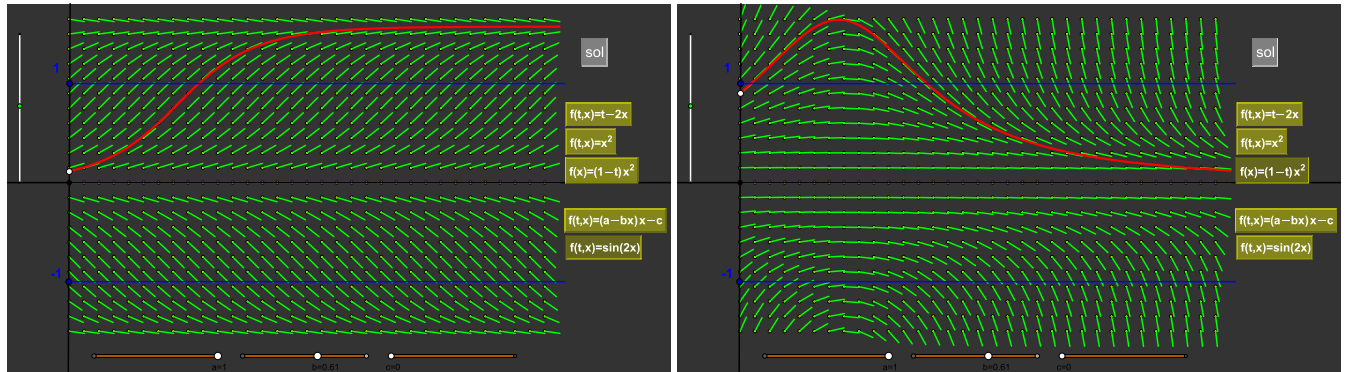
Navesti primer i nacrtati odgovarajuće polje pravaca za a) autonomnu nelinearnu jednačinu sa rešenjem tipa (3), b) neautonomnu jednačinu sa rešenjem tipa (3).

Primeri jednačina su, recimo,

a) $\dot{x}(t) = \sin(x(t)), x(0) = x_0$, i

b) $\dot{x}(t) = (1-t)x^2, x(0) = x_0$.

Rešenja su prikazana na slici, a odgovarajući skript je dat ispod slike. Funkcionalizacijom izabranog teksta kao dugmeta ("use as button"), moguće je prikazati u jednoj animaciji polja pravaca i izbrane integralne krive za čitav niz unapred zadatih jednačina, što je i urađeno u datom primeru.



```
a=|F,B|/|B,C|;b=|G,D|/|E,D|;del=0.2*Q.y;
```

```
c=|T,R|/|S,R|;0=H.y;
```

```
yn=if(Text2.pressed,2,
```

```
  if(Text3.pressed,3,
```

```
    if(Text4.pressed,4,
```

```
      if(Text5.pressed,5,
```

```
        if(Text6.pressed,6)))));
```

```
g2(t,x):=sin(2*x);g3(t,x):=t-2*x;g4(t,x):=x^2;g5(t,x):=(1-t)*x^2;
```

```
g6(t,x):=(a-b*x)*x-c;
```

```
f(t,x):=[g2(t,x),g3(t,x),g4(t,x),g5(t,x),g6(t,x)]_(yn-1);
```

```
drawtext([20,7],"$a=$"+text(a),size->19);
```

```
drawtext([20,6],"$b=$"+text(b),size->19);
```

```
drawtext([-2.5,0],"$x_0=$"+text(x0),size->15);
```

```
l=0.15*directproduct(0..32,(-10..11));lp=l;
```

```
drawall(l,color->(1,1,0),size->2.2);
```

```
repeat(length(l),i,
```

```
  xx=[l_i_1+del,del*f(l_i_1,l_i_2)+l_i_2];
```

```
  xx=l_i+del*((xx-l_i)/|xx-l_i|);
```

```
  lp_i=[l_i,xx]);
```

```
drawall(lp,size->2,color->(0,1,0));
```

```
drawtext(F.xy-[0.05,0.15],"a="+a);
```

```
drawtext(G.xy-[0.05,0.15],"b="+b);
```

```
drawtext(T.xy-[0.05,0.15],"c="+c);
```

```
sol2(t):=arctan(1/(exp(-2*t)*(1/tan(x0))));sol3(t):=t/2-1/4+(x0+1/4)*exp(-2*t);
```

```
sol4(t):=1/(1/x0-t);sol5(t):=2*x0/(2-2*t*x0+x0*t^2);
```

```
sol6(t):=(a+sqrt(-a^2+4*b*c))*tan(0.5*(-sqrt(-a^2+4*b*c))*t-
```

```
  2*arctan((-a*sqrt(-a^2+4*b*c)+2*b*sqrt(-a^2+4*b*c)*x0)/(a^2-4*b*c)))/(2*b);
```

```
sol(x):=[sol2(x),sol3(x),sol4(x),sol5(x),sol6(x)]_(yn-1);
```

```
if(Text7.pressed,plot(sol(x),start->0,stop->4.8+del,size->3,color->[1,0,0]));
```

Neautonomne jednačine oblika $\dot{x}(t) = g(x)h(t)$, nazivamo *separabilnim*.
One razdvajaju promenljive i lako se rešavaju direktnom integracijom.

13.1.2 Populaciona jednačina

Jednačine prvog reda se često sreću u praksi, gde na analogan način opisuju potpuno različite veličine. Najtipičnija jednačina prvog reda je tzv. *populaciona jednačina*, oblika $\dot{x}(t) = ax$, $a > 0$. Ona opisuje idealizovani rast populacije (bakterija, glodara, riba, itd.) u ograničenoj oblasti. Ukoliko prostora i resursa (hrana, voda, svetlost,...) ima dovoljno, tada je priraštaj broja jedinki ($\dot{x}(t)$) proporcionalan njihovom trenutnom broju-populaciji ($x(t)$). Ovu jednačinu je lako rešiti jer ona razdvaja promenljive, a njeno rešenje je $x(t) = x_0 e^{at}$, $x_0 = x(0)$. Dakle, rešenje je tipa (1) : populacija raste eksponencijalno sa vremenom (postaje beskonačna za beskonačno vreme). Eksponencijalni rast se često naziva i rastom po geometrijskoj progresiji: za fiksni interval vremena ($\ln 2/a$) populacija se udvostruči! Fizičari takve procese slikovito nazivaju lavinom i to nije slučajno. Navedimo dobro poznate *jonske lavine*, na čijem principu rade jonizacione komore za detekciju x-zračenja. Ona se sastoji od komore ispunjene gasom sa dve metalne ploče pod naponom. Pod dejstvom x-zraka dolazi do jonizacije atoma, elektron se zatim ubrzava ka anodi i pritom se sudara sa drugim atomima i na rastojanju x dovodi do jonizacije, a na taj način oslobađa dodatnih n elektrona itd. Tako nastaje lavina. Priraštaj broja elektrona po jedinici dužine je $dN/dx = nN$, pa je $N = N_0 e^{nx}$. Odavde je jasno da do lavine dolazi u slučaju kada je koeficijent $n > 0$, pa je uslov da jonizacija jednog atoma mora dovesti, u srednjem, do jonizacije više od jednog atoma (npr. 1.05 je dovoljno).

Jasno je da ovaj model rasta populacije ima ograničen opseg primena - važi u relativno kratkom intervalu jer eksponencijalni rast brzo dovodi do pomanjkanja resursa i konkurencije za prostor, hranu, ... U tom slučaju rast se usporava pa je potrebno primeniti opštiji model rasta - takozvani *logistički model*. Zanimljivo je da je krajem 18. veka, na osnovu ovog prostog modela, Britanski ekonomista Tomas Robert Maltus razvio teoriju po kojoj su katastrofe poput ratova, gladi i zaraza nezaobilazne. Usled eksponencijalnog rasta populacije resursi se brzo iscrpljuju, pa dolazi do ratova koji imaju za cilj njihovu preraspodelu ili do gladi i bolesti, koje smanjuju populaciju, a time i potrebu za resursima. Logistički model se često naziva i *epidemijskim* jer opisuje širenje epidemije u populaciji. Ako pretpostavimo da je deo populacije x već zaražen, a $1 - x$ još uvek nije. U najjednostavnijem modelu pretpostavljamo da se pripadnici populacije slobodno susreću i da je infektivnost (stepen prenosa, "rate") jednak r , tada vži:

$$\frac{dx}{dt} = rx(1 - x).$$

Jednačina razdvaja promenljive i njeno rešenje je

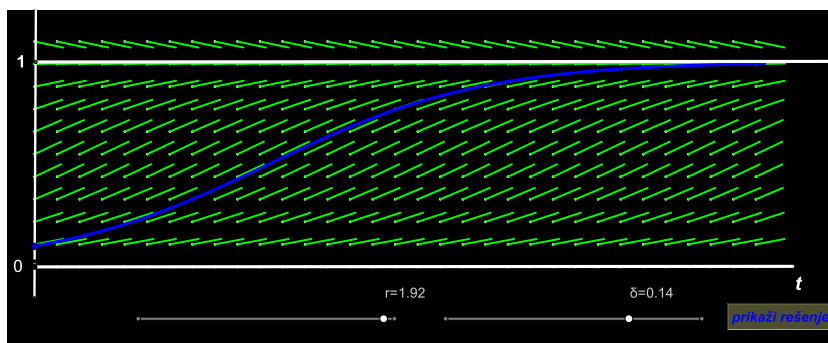
$$x = \frac{1}{1 + Be^{-rt}} = \frac{1}{1 + \left(1 - \frac{1}{x_0}\right) e^{-rt}}, \quad x_0 = x(0).$$

Tipične integralne krive imaju "S" oblik - nakon početnog intervala eksponencijalnog rasta u kojem dominira član x , postepeno dolazi do zasićenja i zaraza ima sve manje mogućnosti da pređe, pa dominira član $1 - x$.

Primer 3: Epidemijska jednačina

Prikazati polje pravaca i izabranu integralnu krivu za epidemijsku jednačinu sa podesivim parametrom r .

Epidemijska jednačina ima 2 stacionarna rešenja (kritične tačke), $x_0 = 0$ i $x_0 = 1$. Prvo je nestabilno, a drugo stabilno. To znači da mala promena početnog uslova u nestabilnom slučaju, brzo počinje da se udaljava od stacionarne vrednosti, dok se u drugom, stabilnom slučaju, ponovo približava stacionarnoj vrednosti. Stacionarna rešenja se dobijaju u slučaju kada je $\frac{dx}{dt} = 0$, a njegova stabilnost zavisi od vrednosti drugog izvoda: ako je $f'(x) < 0$, rešenje je stabilno, a ako je $f'(x) \geq 0$ nestabilno. Prema tome, za određivanje stacionarnih rešenja i proveru njihove stabilnosti, u slučaju jednačine $\frac{dx}{dt} = f(x)$, dovoljno je nacrtati grafik $y = f(x)$, odrediti mu nule i nagibe tangenti u nulama.



Slika 13.1: Polje pravaca i tipična integralna kriva za epidemijsku jednačinu.

Pored epidemija, ovo je standardni model kojim se opisuje populacija riba, npr. šarana u nekoj reci. Član $1 - x$ u logističkoj jednačini opisuje uticaj konkurencije zbog koje je raspoloživa hrane ograničena. Međutim, pored konkurencije, na ukupnu populaciju šarana utiče i ribolov.

Primer 4: Ribarske kvote

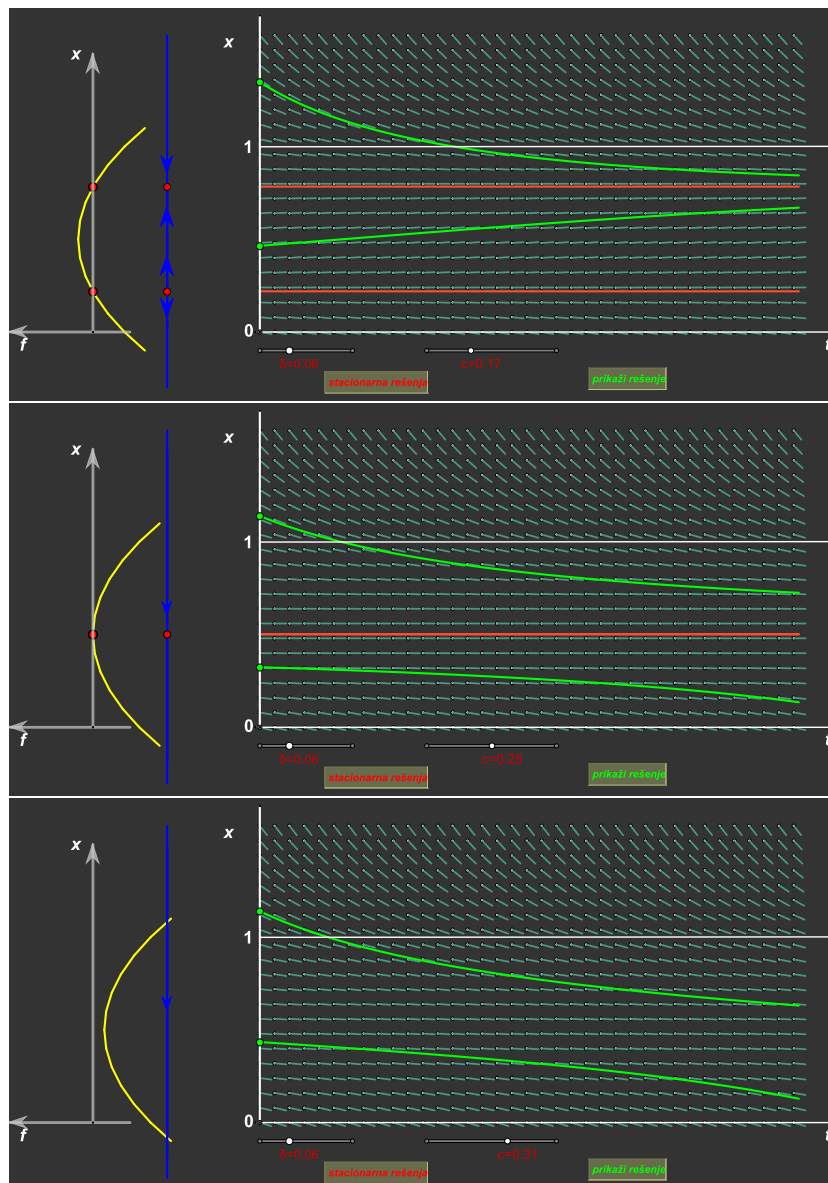
Modelirati uticaj ribolova na populaciju šarana u reci.

Ako brzinu izlova ribe, tzv. ribarsku kvotu, označimo sa c , tada je tražena jednačina, u propisno skaliranim jedinicama,

$$\frac{dx}{dt} = x(1 - x) - c.$$

Jednačina zavisi od parametra c i njegova promena može dovesti do kvalitativne promene polja pravaca, a time i mogućih tipova

rešenja. Takva kvalitativna promena se naziva *bifurkacija*. Na donje tri slika je dat tzv. *bifurkacioni dijagram*. Na njemu su prikazani grafik $y = x(1 - x) - c$, fazni dijagram i polje pravaca sa karakterističnim integralnim krivama. Ukoliko je kvota $0 < c < 1/4$ postoje



Slika 13.2: Prvi tip rešenja ($0 < c < 1/4$) sa 2 stacionarna rešenja, stabilnim i nestabilnim, drugi tip ($c = 1/4$) sa jednim nestabilnim i treći tip ($1/4 < c < 1$) bez stacionarnih rešenja.

dva stacionarna rešenja, pri čemu je niže nestabilno. To znači, da ukoliko populacija šarana, zbog bolesti ili zagađenja padne malo ispod donjeg rešenja, čitava populacija će isčeznuti za konačno vreme. Gornje rešenje opisuje stacionarno stanje pri normalnim uslovima izlova. Ako kvota postane veća od $1/4$, izlov neminovno dovodi do

istrebljenja populacije za konačno vreme. Vrednost kvote $c = 1/4$, sa jednim nestabilnim stacionarnim rešenjem, koja može na prvi pogled da izgleda optimalnom, zapravo je put u katastrofu. Naime, striktno matematički, izlov u takvim uslovima može da traje beskonačno dugo. Međutim, proizvoljno malom perturbacijom (koja se ne može sprečiti u dužem vremenskom periodu) populacija prelazi ispod stacionarne vrednosti i tada je neminovno osuđena na istrebljenje!

Iz svega ovoga zaključimo da fiksna kvota $c = const$ nije najbolje rešenje. Pametnije rešenje je kvota definisana tzv. *povratnom spregom*. To znači da se kvota formira u zavisnosti od trenutne veličine populacije. Jednačina modela sa povratnom spregom glasi:

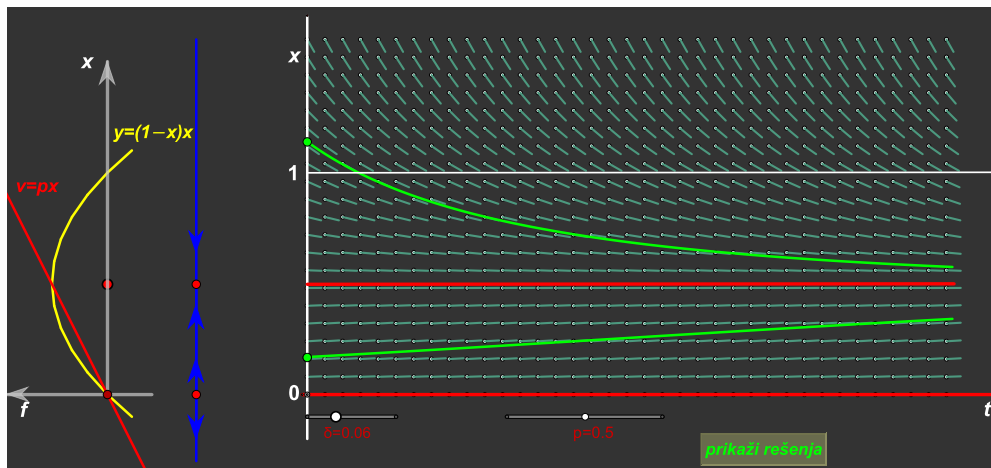
$$\frac{dx}{dt} = x(1-x) - px.$$

Istorijski primer takvog istrebljenja je Steller-ova morska krava. Istrebljena je 1768. godine, za svega 27 godina intenzivnog izlova.

Primer 5: Ribarske kvote - povratna sprega

Analizirati model ribolova sa kvotom baziranom na povratnoj sprezi.

U ovom slučaju stacionarna rešenja se dobijaju u preseku grafika $y = (1-x)x$ i $v = px$, u $x = \{0, 1-p\}$. Prvo rešenje je nestabilno, a drugo stabilno za svako $p \in [0, 1]$! Veličina $v = px$ predstavlja ribolovnu kvotu i maksimalna je u piku parabole, tj. za $p = 1/2$, gde je $v = 1/4$. Dakle, ponovo imamo isti maksimum kao i u prethodnom zadatku, samo je sada optimalno rešenje stabilno.



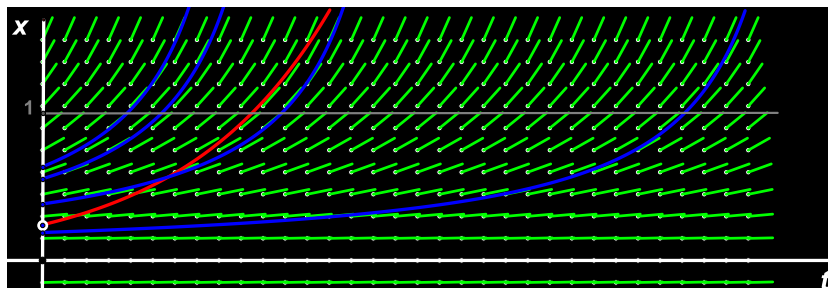
Primer 6: Eksploziona jednačina

Da li postoji modifikacija populacione jednačine kod koje rešenje postaje beskonačno u konačnom vremenskom intervalu? Odrediti primer i prikazati polje pravaca i nekoliko tipičnih integralnih krivih.

Odgovor je postoji. Kod standardne populacione jednačine populacioni priraštaj je srazmeran broju jedinki. Ako pretpostavimo da je priraštaj proporcionalan broju parova jedinki, dobijamo jednačinu $\dot{x}(t) = ax^2$. Za male vrednosti x brzina rasta je manja nego kod populacione jednačine, ali rast brzo postaje "eksplozivan" i teži beskonačnosti. Pretpostavimo da je, radi jednostavnosti, $a = 1$. Jednačina razdvaja promenljive i rešavanjem po t (direktnom integracijom) dobijamo

$$t = \int \frac{dx}{x^2} + C \Rightarrow x = -\frac{1}{t-C}, t < C.$$

Integralne krive su hiperbole sa vertikalnim asimptotama. Ova jednačina se često sreće u hemiji i opisuje dinamiku hemijskih reakcija, gde je brzina reakcije srazmerna koncentracija dva reagensa. Beskonačnost rešenja za konačno vreme u praksi opisuje eksploziju, pa se ova jednačina stoga i naziva *eksplozionom*.



Slika 13.3: Polje pravaca i integralne krive za eksplozionu jednačinu $\dot{x}(t) = x^2$. Radi poredjenja, crvenom bojom je prikazana integralna kriva jednačine $\dot{x}(t) = x$.

13.1.3 Eksponencijalna asimptotika

Populaciona jednačina opisuje situacije u kojima dolazi do eksponencijalnog rasta. U praksi su još brojniji sistemi kod kojih je dinamika opisana eksponencijalnim opadanjem ili eksponencijalnim približavanjem nekoj konstantnoj-stacionarnoj vrednosti. Najpoznatiji i najjednostavniji primer je radioaktivni raspad. Poznato je, na primer, da je verovatnoća da se pojedinačno atomsko jezgro radijuma raspadne u toku naredne sekunde jednaka $p = 1.37 \cdot 10^{-11}$. iz toga zaključujemo da je broj raspadnutih jezgara u jedinici vremena jednak proizvodu verovatnoće i ukupnog broja jezgara u datom trenutku, tj. $dN/dt = -pN$. Ova jednačina ima isti oblik kao i populaciona samo je koeficijent sa desne strane negativan ($-p < 0$). Njeno rešenje je zakon radioaktivnog raspada

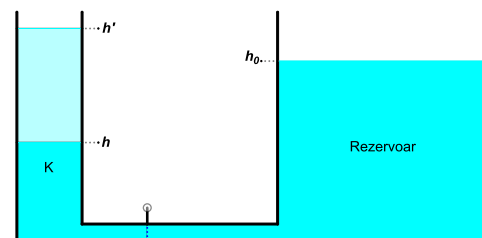
$$N = N_0 e^{-pt},$$

i on tvrdi da se broj preostalih jezgara radioaktivnog materijala sa vremenom eksponencijalno asimptotski približava 0. Osnovna karakteristika eksponencijalne asimptotike je da postoji vremenski interval nakon kojeg se relevantna veličina prepola i taj interval se naziva *vreme poluživota*. U slučaju raspada je jednak $t_{1/2} = \ln 2/p$.

Međutim, postoje brojni analogni problemi u fizici u kojima (terminalna) stacionarna vrednost nije jednaka nuli. Razmotrimo sledeći opštiji problem.

Primer 7

Neka se manji rezervoar (K) napaja iz velikog rezervoara čija je zapremina znatno veća (više redova veličine). U situaciji kada je nivo vode u K (h') viši od nivoa u rezervoaru (h_0), dolazi do pražnjenja, a kada je niži (h) od nivoa h_0 , do punjenja malog rezervoara. Pitanje je kako se menja nivo vode u malom rezervoaru u prvom, a kako u drugom slučaju? Naravno, zbog velike razlike u zapreminama, razumno je pretpostaviti da se nivo vode u velikom rezervoaru ne menja.



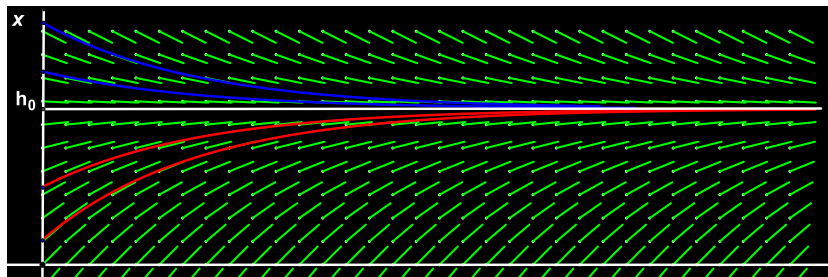
S'obzirom da je protok (dV/dt) kroz spojnicu rezervoara proporcionalan razlici pritiska ($dV/dt = -C(p_2 - p_1)$), a pritisci u rezervoarima su hidrostatički, dolazimo do jednačine

$$\frac{dh}{dt} = \frac{-C\rho g}{A}(h - h_0),$$

koja je specijalni slučaj opšte jednačine $\dot{x}(t) = -a(x - c)$, $a > 0$. Jednačina razdvaja promenljive, pa direktnom integracijom dobijamo

$$h = h_0(1 - e^{\frac{C\rho g}{A}t}).$$

Sa slike vidimo da u jednom slučaju $h < h_0$ imamo rastuću, a u drugom $h > h_0$, opadajuću funkciju sa vremenom. Međutim, u oba slučaja dolazi do eksponencijalnog asimptotskog približavanja stacionarnoj vrednosti $h = h_0$. Teorijski, da bi nivo dostigao stacionarnu



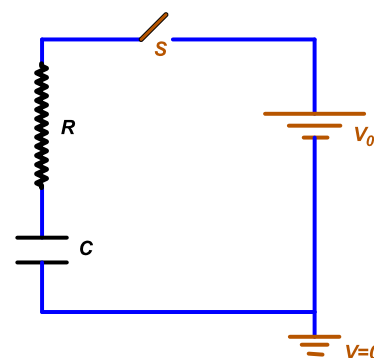
Slika 13.4: Rešenja (integralne krive)
 $h = h_0(1 - e^{-\frac{C\rho g}{A}t})$ za $h < h_0$ -plave i
 $h > h_0$ -crvene.

vrednost potrebna je čitava večnost, ali praktično, nakon vremena jednakog npr. $10A/C\rho g$, praktično je nemoguće izmeriti bilo kakvu promenu veličine h sa vremenom.

Ista jednačina opisuje mnoštvo različitih fizičkih procesa. Punjenje i pražnjenje kondenzatora je potpuno analogno prethodnom problemu, samo što ulogu nivoa h preuzima napon V , ulogu zapremine vode preuzima količina naelektrisanja, ulogu koeficijenta C igra otpor R , a umesto $A/\rho g$ je kapacitet C . Sama jednačina punjenja/pražnjenja kondenzatora je

$$V = V_0(1 - e^{-\frac{t}{RC}}).$$

Druga mehanička analogija je postizanje terminalne brzine prilikom ubrzanog kretanja u sredini sa otporom (primer je slobodan pad kroz vazduh). Tu je relevantna jednačina $-\dot{v} = -g + \frac{w}{m}v$, gde je sa w označen koeficijent otpora sredine.



Prilikom pristajanja brodova uz obalu ili kosmičkih letilica na planetu, upravljanje se vrši na principu (negativne) povratne sprege. Upravljanje po principu povratne sprege opisano je zavisnošću brzine od rastojanja od cilja $\dot{x}(t) = f(x)$, pri čemu je najjednostavniji i najčešći izbor $f(x) = ax$, $a < 0$. Integralna kriva za početni uslov $x_0 = 0$ je konstantna funkcija $x(t) = x_0 = 0$, pa na osnovu teoreme o jedinstvenosti rešenja nijedna druga integralna kriva je ne može seći nu za jedan konačni trenutak t . To znači da je (teorijski) potrebno beskonačno vreme za dolazak do cilja, a pri dostizanju cilja brzina je nulta. U praksi je situacija različita. Da bi vreme prilaska bilo realno prihvatljivo, neophodno je u trenutku kada je brod (ili letilica) blizu cilja promeniti strategiju i npr. u slučaju broda, ručno privući brod obali. U tom slučaju brod konačnom brzinom udara u obalu (ili na planetu), pa je to razlog zbog kojeg su brodovi bočno obloženi automobilskim gumama, a letilice opremljene odgovarajućim amortizerima.

13.1.4 Numerički metodi

It's better to be approximately right than exactly wrong.

Caroeth Read

Veoma često je nemoguće rešiti diferencijalnu jednačinu, posebno kada je u pitanju nelinearna jednačina. Čak i kada je moguće dobiti rešenje analitički, ono je obično izraženo preko komplikovanih integrala ili redova, pa je praktično neupotrebljivo. Zbog toga su razvijeni *približni i numerički* metodi rešavanja jednačina. Najjednostavniji metod potiče od Ojlera (Euler, 1768.) i posebno je efektivan ako se implementira na računaru.

Geometrijska interpretacija preko polja pravaca omogućava intuitivno razumevanje Ojlerovog metoda, štaviše, čini ga očiglednim! Pođimo od najopštije jednačine prvog reda,

$$\dot{x}(t) = f(x, t), x(t = 0) = x_0.$$

Vrednost funkcije x u početnom trenutku je poznata $x(0) = x_0$, pa početna tačka integralne krive ima koordinate $(x_0, 0)$. Nakon kratkog vremenskog intervala u trajanju h , približna vrednost funkcije $x(h) \approx x_1$ je

$$x_1 = x_0 + hf(x_0, 0).$$

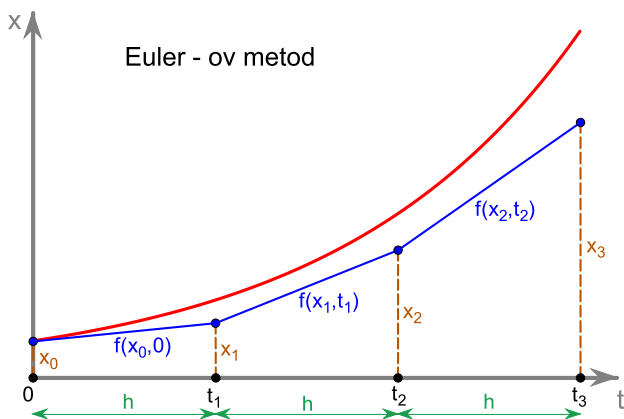
U sledećem trenutku vrednost je

$$x_2 = x_1 + hf(x_1, t_1), t_1 = 0 + h,$$

i procedura se nastavlja u svakom sledećem koraku. Dakle, ako je poznato rešenje u trenutku t_n , rešenje u sledećem trenutku se dobija primenom pravila

$$x_{n+1} = x_n + hf(x_n, t_n), t_{n+1} = t_n + h.$$

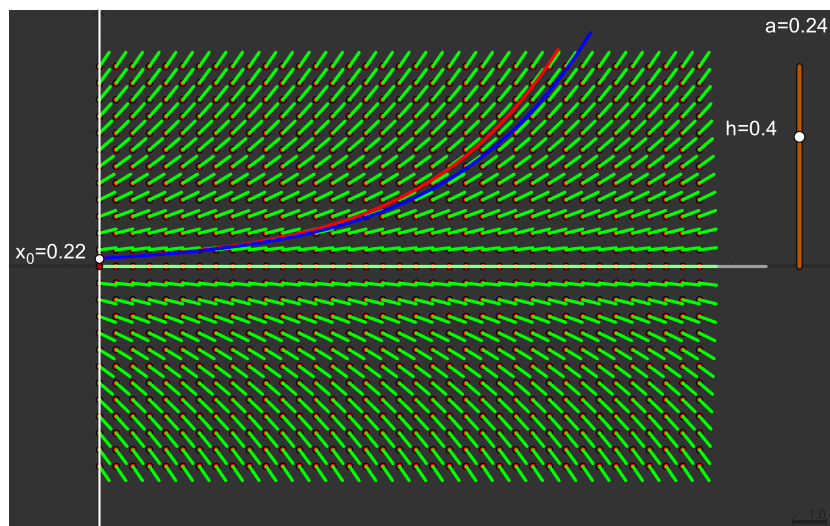
Geometrijski, ova procedura odgovara prostom nadovezivanju duži koje obrazuju polje pravaca. Ili, drugim rečima, numerička procedura se sastoji u diskretizaciji vremena: rešenje se izračunava u diskretnom skupu tačaka, a unutar intervala diskretizacije se aproksimira pravom linijom.



Slika 13.5: Ojlerov metod numeričkog rešavanja diferencijalnih jednačina. Crvenom bojom je označeno egzaktno rešenje, a plavom približno. Sa $f(x_i, t_i)$ je označen nagib odgovarajuće duži u odnosu na x -osu.

Primer 8

Ilustrovati Ojlerov (L. Euler) metod u programu Cinderella na primeru populacione jednačine: $\dot{x}(t) = ax$, $x(0) = x_0$.



Slika 13.6: Približno rešenje je prikazano plavom bojom i dobijeno je za vrednost $h = 0.5$ koje na slici odgovara dužini pojedinačne duži polja pravaca.

Rešenje je prikazano na slici ispod. Znamo da je egzaktno rešenje $x = x_0 e^{at}$. Sa druge strane, primenom Ojlerovog metoda dobijamo,

$$x_{n+1} = x_n + hax_n = (1 + ha)x_n.$$

Rešenje u trenutku $t = (n + 1)h$ dobijamo iz rešenja u prethodnom trenutku $t = nh$ množenjem sa $1 + ha$, pa je $x_n = x_0(1 + ha)^n$, a kako je $h = t/n$, sledi da je

$$x_n = x_0 \left(1 + \frac{at}{n}\right)^n.$$

U limesu $h \rightarrow 0$, pri fiksiranom t :

$$\lim_{h \rightarrow 0} x_n = x_0 \lim_{n \rightarrow \infty} \left(1 + \frac{at}{n}\right)^n = x_0 e^{at}.$$

Očigledno, tačnost aproksimacije zavisi od veličine koraka h , tj. greška je proporcionalna sa h . Pri $h \rightarrow 0$ rešenje teži egzaktnom rešenju. Prateći skript glasi:

```
//Draw slot
f(t,x):=a*x;
sol(x):=x0*exp(a*x);
```

```
//podesavanje parametara klizaca
del=|F,D|/|E,D|;
a=0.2*K.y;
h=0.1*N.y+0.01;
x0=B.y;
tt=36;
```

```

//ispis teksta
drawtext([20,7], "$a=$"+text(a), size->19, color->[1,1,1]);
drawtext(N.xy+[-2.2,0], "$h=$"+text(h), size->19, color->[1,1,1]);
drawtext(B.xy+[-2.5,0], "$x_0=$"+text(x0), size->19, color->[1,1,1]);
//crtanje polja pravaca
l=0.5*directproduct(0..tt, (-12..12));
lp=l;
drawall(l, color->(1,0,0));
repeat(length(l), i,
  xx=[l_i_1+del, del*f(l_i_1, l_i_2)+l_i_2];
  xx=l_i+del*((xx-l_i)/|xx-l_i|);
  lp_i=[l_i, xx]);
drawall(lp, size->3, color->(0,1,0));

//crtanje egzaktnog resenja (int. krive)
plot(sol(x), start->0, stop->min([log(6.5/x0)/a, 18]), size->3, color->(1,0,0));
//priblizno resenje - Euler ov metod
xc=[[0,x0]];
xx=x0; ti=0;
repeat(1+round(tt/(2*h)),
  xx=xx+h*f(0, xx);
  ti=ti+h;
  if(abs(xx)<=7, xc=append(xc, [ti, xx])));
connect(xc, size->3, color->(0,0,1));

```

Kao što je napomenuto, greška Ojlerovog metoda je proporcionalna sa h , pa je za postizanje velike tačnosti potrebno da h bude jako malo, odnosno, da procedura uključi veliki broj koraka, a to može uticati na dugo vreme računa. Međutim, postoje i tačniji numerički metodi.

Ako integralimo levu i desnu stranu polazne jednačine

$$\dot{x}(t) = f(x, t), \quad x(t = 0) = x_0,$$

dobijamo

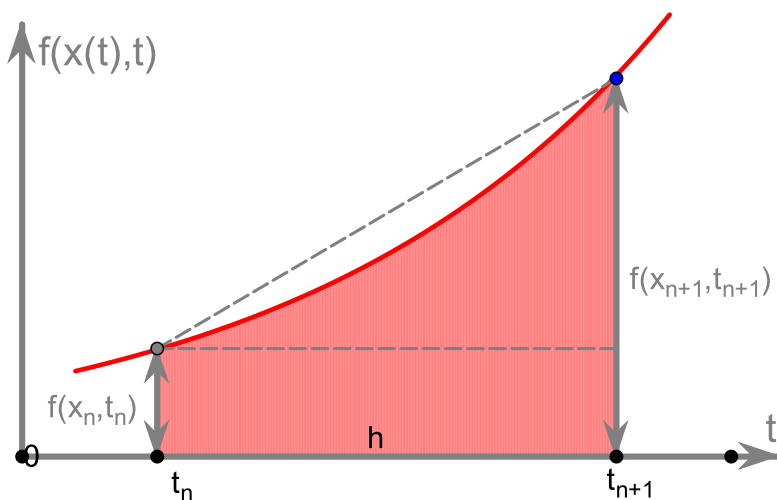
$$x(t_{n+1}) - x(t_n) = \int_{t_n}^{t_{n+1}} f(x(t), t) dt.$$

Dakle, ako znamo vrednost integrala $\int_{t_n}^{t_{n+1}} f(x(t), t) dt$, onda možemo da iz rešenja u trenutku t_n dobijemo rešenje u trenutku t_{n+1} , pa je približno rešavanje jednačine prvog reda svedeno na približno izračunavanje integrala. Najgrublja aproksimacija funkcije pri integraciji je konstantna funkcija, pri čemu se vrednost integrala svodi na površinu pravougaonika.

U tom slučaju je $\int_{t_n}^{t_{n+1}} f(x(t), t) dt \approx hf(x_n, t_n)$, tj. dolazimo do Ojlerovog metoda. Tačniji rezultat dobijamo ako umesto površine pravougaonika računamo površinu trapeza, odnosno, ako napišemo

$$\int_{t_n}^{t_{n+1}} f(x(t), t) dt \approx \frac{h}{2}(f(x_n, t_n) + f(x_{n+1}, t_{n+1})).$$

Dobijeni metod nazivamo *poboljšani Ojlerov metod* i može se pokazati da je njegova greška proporcionalna sa h^2 .



Slika 13.7: Ojlerov metod i poboljšani Ojlerov metod se svode na približno izračunavanje integrala. Kod Ojlerovog metoda, izračunava se površina pravougaonika, a u kod poboljšanog Ojlerovog metoda, površina trapeza.

Sa stanovišta kompjuterske implementacije, potrebno je realizovati integracioni korak kroz petlju. U najjednostavnijem slučaju (Ojlerov metod), imamo sledeću petlju:

```
c=x0; ti=0;
xc=[[ti, c]];
repeat(n,
  c=h*f(ti, c);
  ti=ti+h;
  xc=append(xc, [ti, c]));
```

Kod poboljšanog Ojlerovog metoda imamo jedan dodatni korak u petlji:

```
x=x0; ti=0;
xc=[[ti, x]];
repeat(n,
  c1=h*f(ti, x);
  c2=h*f(ti+h, x+c1);
  x=x+(c1+c2)/2;
  ti=ti+h;
  xc=append(xc, [ti, x]));
```

Kod rešavanja složenih diferencijalnih jednačina, obično se primenjuje još složeniji metod, tzv. Runge-Kuta (Runge-Kutta) četvrtog reda (RK4). Greška ovog metoda je proporcionalna sa h^4 . Poput prethodna dva metoda i ovaj se svodi na približnu integraciju, ali u ovom slučaju je kriva aproksimirana parabolom. Ukoliko je diferencijalna jednačina $\frac{dx}{dt} = f(t, x)$ takva da f ne zavisi od x , tada se rešenje

direktno svodi na integral i tada je metod RK_4 ekvivalentan (*Simpsonovom metodu*). Odgovarajući skript koji realizuje metod Runge-Kuta je:

O tome videti više u primeru 20. dodatka *Matematička analiza u slikama*.

```

x=x0; ti=0;
xc=[[ti,x]];
repeat(n,
  c1=h*f(ti,x);
  c2=h*f(ti+h/2,x+c1/2);
  c3=h*f(ti+h/2,x+c2/2);
  c4=h*f(ti+h,x+c3);
  x=x+(c1+2*c2+2*c3+c4)/6;
  ti=ti+h;
  xc=append(xc,[ti,x]));

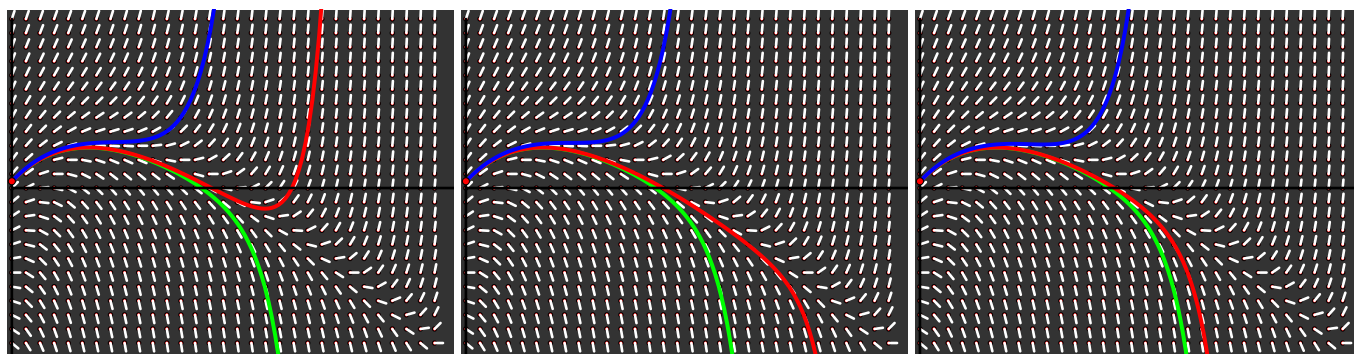
```

Primer 9

Uporediti prethodna tri numerička metoda na primeru neautonomne jednačine

$$\dot{x}(t) = (1+t)x + 1 - 3t + t^2.$$

Rešenje:

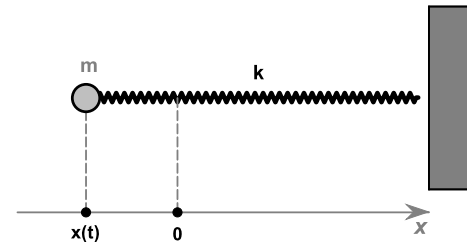


Na slici su prikazane integralne krive: plava dobijena Ojlerovim, crvena poboljšanim Ojlerovim i zelena metodom Runge-Kuta. Početne koordinate za sve tri integralne krive su $(0, 0.0655)$, a vrednosti parametra h su: levo 0.04, u sredini 0.036 i desno 0.03.

13.2 Diferencijalne jednačine drugog reda

Starogrčki filozofi su smatrali da je stanje nekog tela u potpunosti određeno njegovim položajem. Otuda i poznati paradoks Zenonove strele: "Ako uporedimo mirujuću strelu, koja u nekom trenutku vremena zauzima isti položaj kao i strela koja se kreće, a nikakav spoljašnji uzrok kretanja ne deluje ni na jednu od njih, tada ih je nemoguće razlikovati, pa je kretanje strele samo iluzija"! Međutim, od vremena Galileja je poznato da je stanje sistema opisano položajem i brzinom, što je ekvivalentno tvrđenju da je dinamika sistema opisana diferencijalnim jednačinama drugog reda.

Ono je posebno očigledno ako posmatramo teg na opruzi koji osciluje oko ravnotežnog položaja. Samo na osnovu slike oscilatora ne možemo reći gde će teg biti u sledećem trenutku, tj. da li će se pomeriti na levo ili na desno. Takođe, u prethodnoj glavi je već pokazano da autonomna jednačina prvog reda ne može da opiše nikakvo oscilatorno kretanje. Zbog toga je linearan harmonijski oscilator tipičan sistem drugog reda i jednačina koja ga opisuje, među jednačinama drugog reda ima ulogu sličnu onoj koju ima populaciona jednačina među svim jednačinama prvog reda.



Obična diferencijalna jednačina drugog reda formalno je ekvivalentna sistemu od dve jednačine prvog reda. Štaviše, jednačina n -tog reda je ekvivalentna sistemu od n jednačina prvog reda. Opšti oblik jednačine n -tog reda glasi:

$$x^{(n)}(t) = f(t, x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}),$$

gde je sa t označena nezavisna promenljiva koju obično interpretiramo kao vreme, sa x nepoznata funkcija, koju tada interpretiramo kao veličinu koja se menja sa vremenom, a sa tačkom je označen izvod po promenljivoj t . U većini primera funkcija f je neprekidno diferencijabilna po svakom argumentu. Ako se u običnoj diferencijalnoj jednačini pojavljuje više nepoznatih funkcija, tako da izvod jedne zavisi od ostalih, onda imamo sistem običnih diferencijalnih jednačina prvog reda:

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, x_2, \dots, x_n) \\ \dot{x}_2 &= f_2(t, x_1, x_2, \dots, x_n) \\ &\vdots \\ \dot{x}_n &= f_n(t, x_1, x_2, \dots, x_n). \end{aligned}$$

Transformacija jednačine n -tog reda u sistem od n jednačina prvog reda se postiže smenom

$$x = x_1, \dot{x} = x_2, \dots, x^{(n-1)} = x_n.$$

Dobijeni sistem glasi

$$\begin{aligned} \dot{x}_n &= f_n(t, x_1, x_2, \dots, x_n) \\ \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{n-1} &= x_n, \end{aligned}$$

što se može napisati kompaktnije u vektorskom obliku,

$$\dot{\vec{x}}(t) = \vec{f}(t, \vec{x}), \quad \vec{x}(t) = (x_1, x_2, \dots, x_n).$$

To je takozvana *normalna forma* i predstavlja polaznu tačku za geometrijsku interpretaciju. Naime, jednačina je potpuno definisana vektorskim poljem $\vec{f}(t, \vec{x})$ koje se menja u vremenu. Dalji opis nastavljamo na konkretnom primeru LHO.

Primer 1

Prikazati i animirati *fazni portret* linearnog harmonijskog oscilatora.

Kretanje tega mase m zakačenog za linearnu (Hukovu) oprugu krutosti k je opisano linearnom diferencijalnom jednačinom

$$m\ddot{x} + kx = 0.$$

Jednačina je drugog reda pa je stanje sistema u svakom trenutku t opisano sa dva realna broja: trenutnim položajem x i brzinom v . Ova dva broja možemo da interpretiramo kao tačke u *faznoj ravni*. Fazne koordinate se menjaju sa vremenom i njihova evolucija je opisana jednačinama

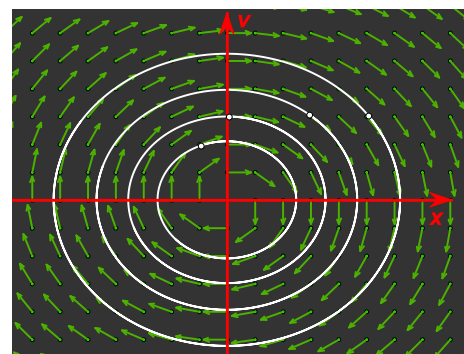
$$\begin{aligned} \dot{x} &= v, \\ \dot{v} &= -\frac{k}{m}x \equiv -\omega^2 x, \end{aligned}$$

koje u normalnoj formi glase

$$(\dot{x}, \dot{v}) = (v, -\omega^2 x).$$

Na taj način je definisano vektorsko polje u faznoj ravni. Intenzitet polja varira od tačke do tačke, pa se radi boljeg pregleda, obično crta normirano vektorsko polje. Na takvim slikama ne dolazi do presecanja strelica, a slika i dalje nosi svu relevantnu informaciju o sistemu. Rešenje ove jednačine je svaka kriva, koju nazivamo integralna kriva sistema, koja je tangentna na tako definisano vektorsko polje. Integralne krive se pak interpretiraju kao putanje delića zamišljenog faznog fluida pa se, stoga, ponekad nazivaju i orbitama ili trajektorijama sistema.

Fazni fluid se na slici obrće oko koordinatnog početka. Koordinatni početak je izuzetak - tačka u faznoj ravni, koja se nalazi u koordinatnom početku ostaje u stanju mirovanja, jer je $(\dot{x}, \dot{v}) = (0, 0)$ za $(x, v) = (0, 0)$. Drugim rečima, koordinatni početak je fiksna ili stacionarna tačka. Međutim, tačka u faznoj ravni, koja polazi iz bilo kojeg drugog početnog položaja, kružiće oko koordinatnog početka i



konačno će se vratiti u svoj početni položaj. Takve trajektorije formiraju zatvorene orbite, kao što je prikazano na slici, i zajedno formiraju fazni portret sistema.

Prateći skript:

```
// Draw slot
rot(f):=[cos(f), -sin(f)], [sin(f), cos(f)];
rots(f,pt1,pt2):=[pt1,pt1+rot(f)*(pt2-pt1)];
f(v):=[1.4*v_2, -sin(v_1)];

dens=1.5*|Q,N|/|M,N|;//kontrolni parametri
norm=0.1*|R,P|/|O,P|;//(klizac se ne vidi na slici)

//crtanje resetke
lat=0.04*directproduct(dens*(-12..18),dens*(-9..9));
drawall(lat,size->2);

//crtanje vektorskog polja na resetci
matf(f,pt):=if(pt==[0,0],[0,0],[pt,pt+f(pt)/|f(pt)|*norm]);
str(f,vec):=if(vec==[[0,0],[0,0]],[[0,0],[0,0]],
[vec,rots(f,vec_2,0.2*vec_1+0.8*vec_2),rots(-f,vec_2,0.25*vec_1+0.75*vec_2)]);
repeat(length(lat),i,drawall(str(pi/8,matf(f,lat_i)),size->2,color->(0.3,0.7,0)));

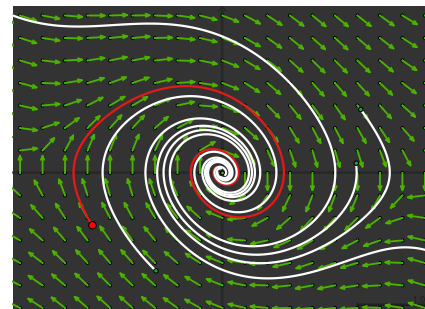
//num. resavanje jednacine i crtanje int. krivih
nn=750;h=0.01;
y0=[C.xy,D.xy,E.xy,F.xy];
yc=y0;y=y0;
repeat(length(y0),i,
  repeat(1.2*nn,
    c1=h*f(y_i);
    c2=h*f(y_i+c1/2);
    c3=h*f(y_i+c2/2);
    c4=h*f(y_i+c3);
    y_i=y_i+(c1+2*c2+2*c3+c4)/6;
  yc_i=append(yc_i,y_i));
connect(yc_i,color->(1,1,1),size->2));
```

Za više primera faznog prostora konzervativnih, 1D sistema videti u poglavlju posvećenom dinamici jednodimenzionalnih sistema.

13.2.1 Autooscilacije

Konzervativni sistemi iz prethodnog poglavlja su očigledno idealizacija. Jednom pokrenut, linearni harmonijski oscilator će većito oscilovati. Nerealna dinamika je posledica zanemarivanja energetskih gubitaka - različitih tipova trenja. Uključivanjem trenja dobijamo prigušeno oscilovanje. Fazni portret prigušenog oscilovanja je prikazan na slici ispod. Stabilna fiksna tačka je i dalje u koordinatnom početku ali je sada tipa fokus; periodične trajektorije oko nje više nisu periodične, već se asimptotski približavaju fiksnoj tački. Sedlaste fiksne tačke, kao i separatriše, tj. orbite koje su u idealnom slučaju razdvajale dva tipa periodičnih orbita više ne postoje. Bez obzira na početne uslove, nakon dovoljno vremena sistem je faktički u stanju mirovanja, odnosno u neposrednoj okolini fokusa.

Nameće se pitanje kako zapravo opisati oscilatornu dinamiku realnih fizičkih sistema koji mogu da osciluju proizvoljno dugo. Prvi odgovor je sistem koji prinudno osciluje, ali tada je prisutan dodatni sistem (prinuda) koji osciluje, pa se problem samo prenosi na dinamički opis prinude. Međutim, očigledno je da je sistemu koji osciluje neophodno dovoditi energiju spolja. Ukoliko je dovedena energija u jedinici vremena manja od gubitaka, dinamika sistema se neće kvalitativno promeniti. Ako je, pak, veća, tada sistem osciluje sa sve većom brzinom i/ili elongacijom, tako da, pre ili kasnije, mora doći do lomljenja sistema ili eksplozije. Jedina mogućnost za uspostavljanje stabilnih oscilacija je nelinearnost sistema.

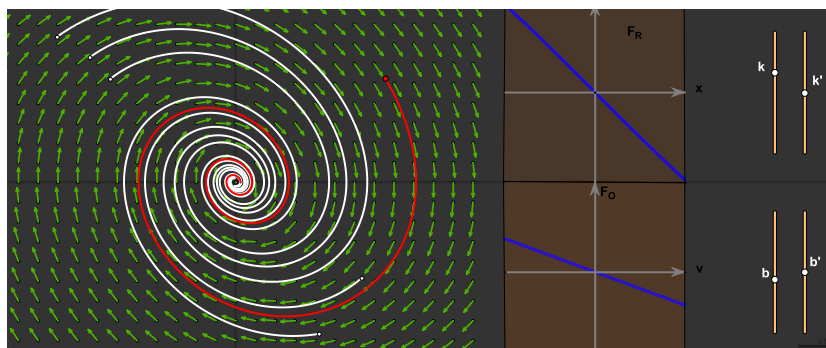


Ukoliko trenja nelinearno zavise od faznih varijabli (elongacija, brzina), tada je moguća situacija da je priliv energije kod stanja sa malim vrednostima parametara, veći od gubitaka, a da pri velikim vrednostima, postaje manji. Takvi sistemi su brojni i prvi teorijski opis nelinearnih oscilacija je dao Lord Rejli (Rayleigh) 1877. godine u svojoj poznatoj raspravi o akustici *Teorija zvuka*. Ovakav tip oscilacija je kasnije ruski fizičar Andronov nazvao *autooscilacijama ili samooscilacijama*. Tipični primeri samooscilacija su nastajanje zvuka kod duvačkih i gudačkih instrumenata, ali i škripa vrata, rad srca, časovnika, radiotalasnog predajnika itd.

U programu Cinderella je lako sprovesti analizu koja je Rejlja dovela do njegove poznate jednačine koja opisuje samooscilacije kod duvačkih instrumenata. Kod duvačkih instrumenata, recimo klarineta, zvuk nastaje kada mlaz vazduha proizveden duvanjem nailazi na trščani jezičak (pisak) koji počinje da osciluje. Da bi opisao njegove oscilacije, Rejli je modelirao ponašanje sila koje dovode do oscilacija: elastične (restitucione) i sile otpora. Elastična sila kod izvijanja piska približno linearno zavisi od elongacije, dok se sila otpora kvalitativno menja u zavisnosti od brzine mlaza tj. jačine duvanja.

Primer 2

Modelirati elastičnu i silu otpora i prikazati odgovarajući fazni portret u slučaju kada nema vazdušnog mlaza, tj. bez duvanja.

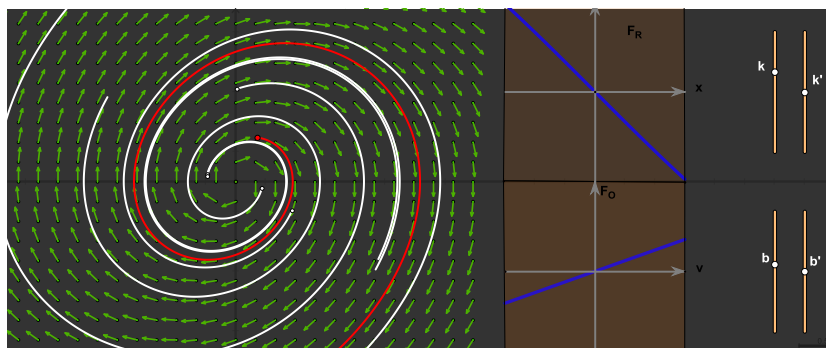


Slika 13.8: Dinamika jezička bez vazdušnog mlaza

Situacija kada nema vazdušnog mlaza je prikazana na slici. Obe sile su linearne i suprotnog znaka od znaka odgovarajuće fazne varijable. Na desnoj strani su grafici zavisnosti elastične sile od elongacije i sile otpora od brzine. Na levoj strani je prikazan odgovarajući fazni portret. Naravno, dobijeni fazni portret opisuje linearne prigušene oscilacije.

Primer 3

Modelirati elastičnu i silu otpora i prikazati odgovarajući fazni portret u slučaju kada je prisutan vazdušni mlaz, a brzina oscilovanja jezička mala.



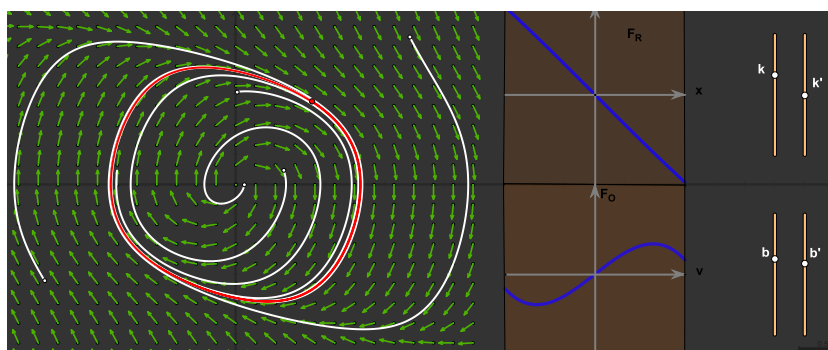
Slika 13.9: Dinamika jezička sa vazdušnim mlazom i malom amplitudom

Slučaj kada je prisutan vazdušni mlaz i kada je brzina oscilovanja jezička mala, jednostavno se modelira promenom znaka sile otpora.

Ona je sada proporcionalna i u istom pravcu sa brzinom. Situacija je prikazana na slici ispod. Na faznom dijagramu dolazi do promene fiksne tačke portreta: privlačni fokus se transformiše u odbojni. Međutim, ovaj modelni režim je neodrživ pri većim, odnosno bržim oscilacijama piska. Sila otpora je nelinearna, pa pri većim brzinama oscilovanja sila otpora ipak menja znak.

Primer 4

Razviti konačan Rejljev model sa nelinearnom silom otpora koja je usklađena sa graničnim slučajevima iz prethodna dva zadatka i detektovati formiranje *graničnog cikla* na faznom portretu sistema.



Slika 13.10: Kompletan opis dinamike jezička

Konačni model i njegov fazni prostor su prikazani na slici 13.10. Tu se pri malim oscilacijama sila otpora ponaša kao na slici 13.8, a pri velikim, kao na slici 13.9. Logično je da i fazni portret na slici 13.10 bude kombinacija prethodna dva. Pri malim oscilacijama, fazne trajektorije se udaljavaju od fiksne tačke, a za velike se približavaju. Kompromisno rešenje je da se i jedne i druge namotavaju na novu vrstu orbite *izolovani granični cikl*. Granični cikl ne treba mešati sa zatvorenim orbitama (obični cikl) kod linearnog harmonijskog osciatora. U svakoj okolini običnog cikla se nalazi bar još jedan običan cikl, dok je granični cikl uvek *izolovan* od ostalih.

Prateći skript:

```
//Draw slot
rot(f):=[cos(f), -sin(f)], [sin(f), cos(f)];
rots(f,pt1,pt2):=[pt1,pt1+rot(f)*(pt2-pt1)];
lat=0.33*directproduct(-12..12,-9..9);
drawall(lat,size->2);
mat=[[0, -1], [1, 0]];

k=|K,G|/|G,H|;
```

```

b=-1+2*|P6,P1|/|P2,P1|;
k1=0.5-|P5,Z|/|P0,Z|;
b1=0.5-|P7,P3|/|P4,P3|;
g(x):=3*(-k*x-k1*x^3);
h(x):=3*(-b*x-b1*x^3);
f(v):=[v_2,g(v_1)+h(v_2)];

plot(1.5+g(x-6),start->4.5,stop->7.5,size->5);
plot(-1.5+h(x-6),start->4.5,stop->7.5,size->5);

matf(f,pt):=if(pt==[0,0],[0,0],[pt,pt+f(pt)/|f(pt)|*0.24]);
str(f,vec):=if(vec==[[0,0],[0,0]],[[0,0],[0,0]],[vec,rots(f,vec_2,0.25*vec_1+0.75*vec_2),rots(-f,vec_2,0.25*vec_1+0.75*vec_2)]);
repeat(length(lat),i,drawall(str(pi/8,matf(f,lat_i)),size->2.5,color->(0.3,0.7,0)));

nn=1000;h=0.01;

y0=[C.xy,D.xy,E.xy,F.xy];
yc=y0;y=y0;
repeat(length(y0),i,
  repeat(nn,
    c1=h*f(y_i);
    c2=h*f(y_i+c1/2);
    c3=h*f(y_i+c2/2);
    c4=h*f(y_i+c3);
    y_i=y_i+(c1+2*c2+2*c3+c4)/6;
  yc_i=append(yc_i,y_i));
connect(yc_i,color->(1,1,1),size->3));

```

13.2.2 Tipovi faznih portreta u ravni

Pod faznim portretom podrazumevamo razbijanje faznog prostora na fazne trajektorije. Postojanje takvog razbijanja sledi direktno iz teoreme o egzistenciji i jedinstvenosti rešenja: ukoliko je vektorsko polje f definisano diferencijalnom jednačinom

$$\dot{\vec{x}}(t) = \vec{f}(t, \vec{x})$$

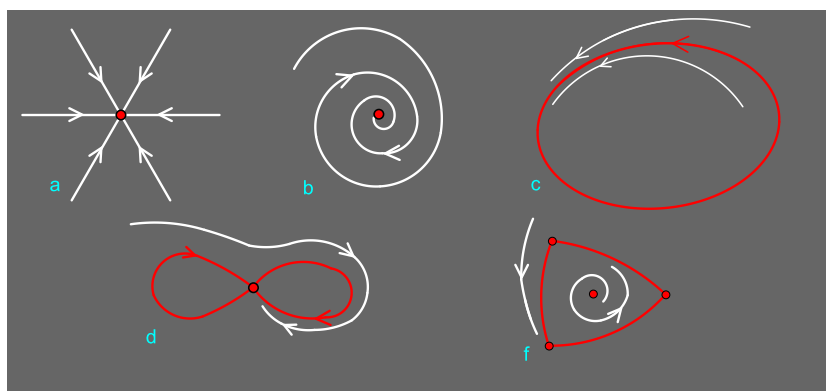
neprekidno diferencijabilno, tada kroz svaku tačku faznog prostora prolazi tačno jedna trajektorija, pri čemu je zavisnost trajektorije od koordinata fazne tačke diferencijabilna. U kvalitativnoj teoriji diferencijalnih jednačina osnovni zadatak je odrediti tipove faznih portreta sa tačnošću do difeomorfizma (glatke deformacije) faznog prostora, u našem slučaju, fazne ravni.

Prvo pitanje kvalitativne teorije je određivanje osnovnih topoloških tipova faznih trajektorija. Odgovor je jednostavan. Postoje samo tri topološki različita tipa: fiksna tačka, otvorena i zatvorena trajektorija. Sva tri tipa su prisutna u primeru sa oscilacijama piska kod duvačkih instrumenata. Fiksna tačka je u koordinatnom početku fazne ravni, zatvorena trajektorija je granični cikl, a sve ostale tačke pripadaju otvorenim

trajektorijama koje se asimptotski namotavaju na granični cikl. Znatno je teže pitanje klasifikacije topološki različitih faznih portreta. Pod tipom faznog portreta podrazumevamo uzajamni raspored prisutnih faznih trajektorija. Zapravo, dovoljno je odrediti ponašanje svih otvorenih trajektorija u limesu $t \rightarrow \infty$, tj. skup svih graničnih tačaka ka kojima otvorene trajektorije teže (granični skup). U slučaju dvodimenzionalnog faznog prostora, odgovor daje Poenkare-Bendiksonova teorema. Teorema se odnosi na disipativne sisteme u ravni i na sferi. Kod takvih sistema u faznom prostoru postoji kružnica dovoljno velikog radijusa takva da sve fazne trajektorije u njenoj okolini seku datu kružnicu i to u smeru od spoljašnjosti ka unutrašnjosti.

Poenkare-Bendiksonova (Poenkare - Bendikson) teorema

Kod disipativnih sistema u ravni ili na sferi granični skup proizvoljne trajektorije je fiksna tačka ili granični cikl ili unija fiksnih tačaka zajedno za separatriisama koje ih spajaju (*separatrisni mnogougao*).



Slika 13.11: Primeri faznih trajektorija (belo) zajedno sa graničnim skupom (crveno)

Primeri trajektorija i njima odgovarajućih graničnih skupova prikazani su na slici 13.11. Fiksne tačke su označene slovima a i b , granični cikl sa c , a separatriisni mnogouglovi sa d i f . Poenkare-Bendiksonova teorema je posledica Žordanove teoreme o prostoj krivoj, teoreme o jedinstvenosti, koja zabranjuje presecanje faznih trajektorija i uslova disipativnosti. Važno je još napomenuti da, premda trajektorije prikazane crvenom bojom na slici 13.11 pod d izgledaju kao da se seku, zapravo se radi o tri nepresecajuće trajektorije: jedna je fiksna tačka, a druge dve se asimptotski udaljavaju i približavaju ka njoj.

Iz svih prethodnih primera je jasno da postoje različiti tipovi fiksnih tačaka: otvorene trajektorije u njenoj okolini prilaze ili se udaljavaju na različite načine. Nekat pravolinijski, a nekad spiralno, nekad se sve udaljavaju ili prilaze, a nekad u jednom pravcu prilaze, a u drugom se udaljavaju itd. Zbog toga je sledeći zadatak da se eksplicitno pobroje mogući tipovi fiksnih tačaka. Rešenje se dobija primenom linearne algebre, odnosno preciznije, linearizacijom vektorskog polja oko fiksne tačke. Interesantno je napomenuti da je sam nastanak linearne algebre blisko povezan sa problemom klasifikacije fiksnih tačaka. Sa druge strane, linearna vektorska polja su verovatno najbolji način da se dejstvo linearnog operatora u vektorskom prostoru vizuelno prikaže.

13.2.3 Linearna vektorska polja

Fiksnom tačkom nazivamo tačku \vec{a} u kojoj je vektorsko polje nula, tj. $\vec{f}(\vec{a}) = 0$. Ukoliko se sistem u početnom trenutku nalazi u stanju \vec{a} , onda će u tom stanju i ostati, jer je u tom slučaju $\dot{\vec{x}}(t) = 0$. Ako sistem od dve jednačine napišemo po komponentama, tada dobijamo:

$$\dot{x}(t) = f(x, y),$$

$$\dot{y}(t) = g(x, y).$$

Ako uvedemo nove koordinate $u = x - a$, $v = y - b$ i funkcije f i g razvijemo u red oko fiksne tačke $\vec{a} = (a, b)$, dobijamo linearizovane izraze

$$\dot{u} = u \frac{\partial f}{\partial x} + v \frac{\partial f}{\partial y} + O(u^2, v^2, uv),$$

$$\dot{v} = u \frac{\partial g}{\partial x} + v \frac{\partial g}{\partial y} + O(u^2, v^2, uv).$$

Isti sistem, zapisan u matricnoj formi glasi:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + O(u^2, v^2, uv).$$

Matricu

$$A = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix}_{(a,b)}$$

nazivamo Jakobijevom matricom u fiksnoj tački \vec{a} . Postupak linearizacije se sastoji u zanemarivanju člana $O(u^2, v^2, uv)$, pa se linearizovani dvodimenzionalni sistem diferencijalnih jednačina u matricnom obliku može zapisati kao:

$$\dot{\vec{x}}(t) = A\vec{x}, \quad A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \vec{x} = \begin{pmatrix} u(t) \\ v(t) \end{pmatrix}.$$

Takav sistem je linearan u smislu da ako su x_1 i x_2 rešenja, tada je rešenje i bilo koja njihova linearna kombinacija $c_1x_1 + c_2x_2$. Rešenja linearnog sistema se dobijaju jednostavnim eksponenciranjem matrice sistema: $\vec{x}(t) = e^{At}\vec{x}(0)$. Eksplicitni izraz za matricni eksponent zavisi od tipa matrice (vidi sledeći primer). Prema tome, klasifikacija fiksnih tačaka se svodi na spektralnu klasifikaciju realnih 2×2 matrica. U slučaju kada je matrica dijagonalizabilna, svojstveni bazis definiše svojstvene pravce, a oni se u faznoj ravni lako prepoznaju po tome što fazna tačka koja se u početnom trenutku nalazi na faznom pravcu, tokom vremena ostaje na njemu.

Primer 5

Eksplicitno navesti moguća rešenja sistema od dve linearne diferencijalne jednačine sa konstantnim koeficijentima.

Rešenja se dobijaju matricnim eksponenciranjem, a rezultat eksponenciranja zavisi od tipa matrice. U linearnoj algebri se dokazuje sledeća teorema:

Neka je A realna 2×2 matrica. Tada postoji realna, invertibilna 2×2 matrica P tako da važi,

$$P^{-1}AP = A_i,$$

pri čemu je A_i pripada jednom od sledeć tri tipa:

$$A_1 = \begin{bmatrix} l_1 & 0 \\ 0 & l_2 \end{bmatrix}, A_2 = \begin{bmatrix} l & 1 \\ 0 & l \end{bmatrix}, A_3 = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}.$$

Matricu nazivamo dijagonalizabilnom ako je slična dijagonalnoj matrici, slučaj A_1 . Realna 2×2 matrica je nedijagonalizabilna ako a) ima konjugovano-kompleksne svojstvene vrednosti, slučaj A_3 , b) ima jednu degenerisanu svojstvenu vrednost i samo jedan svojstveni vektor, slučaj A_2 .

Razmotimo eksponencijalnu funkciju i odgovarajuće rešenje jednačine za svaki pojedinačni slučaj.

1. Rešenje je $\vec{x}(t) = e^{At}\vec{x}(0)$, koje u svojstvenom bazisu matrice A glasi:

$$\dot{\vec{x}}(t) = c_1 e^{l_1 t} \vec{e}_1 + c_2 e^{l_2 t} \vec{e}_2,$$

gde su c_1 i c_2 koeficijenti u razvoju vektora $\vec{x}(0)$ u svojstvenom bazisu. Dobijeno rešenje je sedlo ako su svojstvene vrednosti različitog znaka, a čvor ako su istog.

2. U ovom slučaju je $e^{At} = e^{lt} \begin{bmatrix} l & 1 \\ 0 & l \end{bmatrix}$, pa je rešenje

$$\vec{x}(t) = c_1 e^{lt} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_2 e^{lt} \begin{pmatrix} t \\ 1 \end{pmatrix},$$

pri čemu, koeficijente c_1 i c_2 treba odrediti tako da pri $t = 0$ bude ispunjen početni uslov. Dobijeno rešenje je degenerisani čvor.

3. U poslednjem slučaju je $e^{At} = e^{at} \begin{bmatrix} \cos(bt) & \sin(bt) \\ -\sin(bt) & \cos(bt) \end{bmatrix}$, a rešenje je

$$\vec{x}(t) = c_1 e^{at} \begin{pmatrix} \cos(bt) \\ -\sin(bt) \end{pmatrix} + c_2 e^{at} \begin{pmatrix} \sin(bt) \\ \cos(bt) \end{pmatrix}.$$

Naravno, i ovde je potrebno odrediti koeficijente tako da početni uslovi budu zadovoljeni. Rešenje predstavlja fokus za $a \neq 0$ i centar za $a = 0$.

Primer 6

Napraviti animaciju koja prizvoljnoj realnoj 2×2 matrici pridružuje odgovarajuće vektorsko polje, separatriše (definisane svojstvenim vektorima) i svojstvene vrednosti (na posebnom panelu).

Za proizvoljnu matricu $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ odgovarajuća svojstvena jednačina glasi:

$$\lambda^2 - \operatorname{tr}(A)\lambda + \det(A) = 0,$$

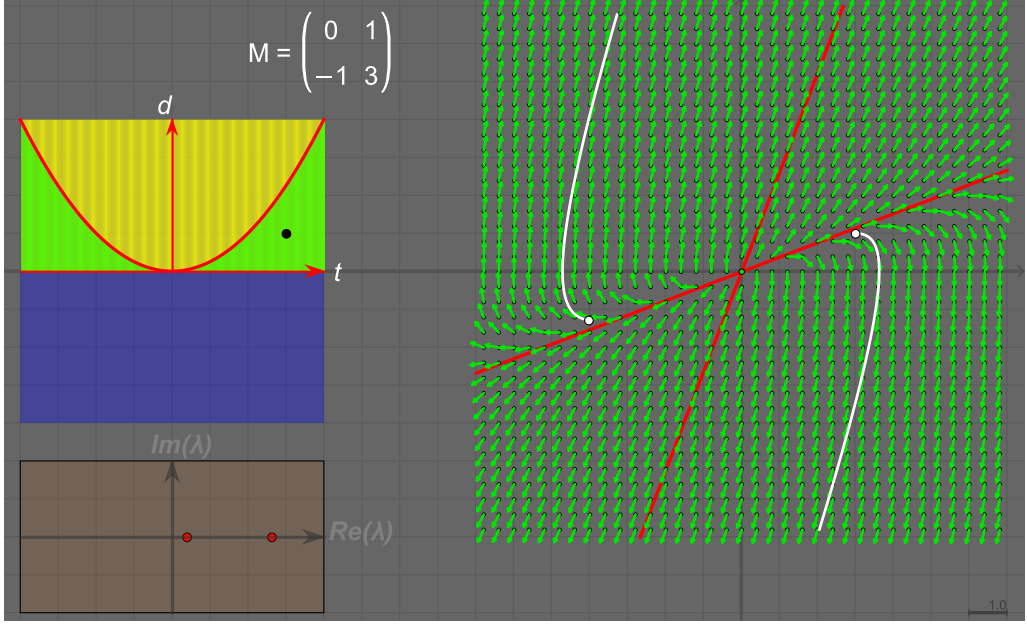
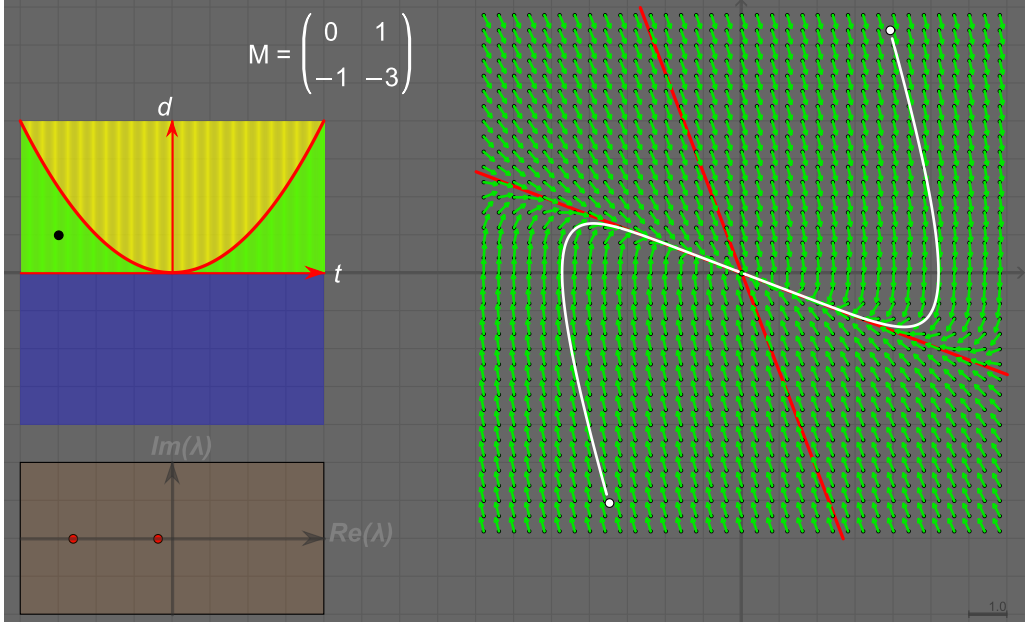
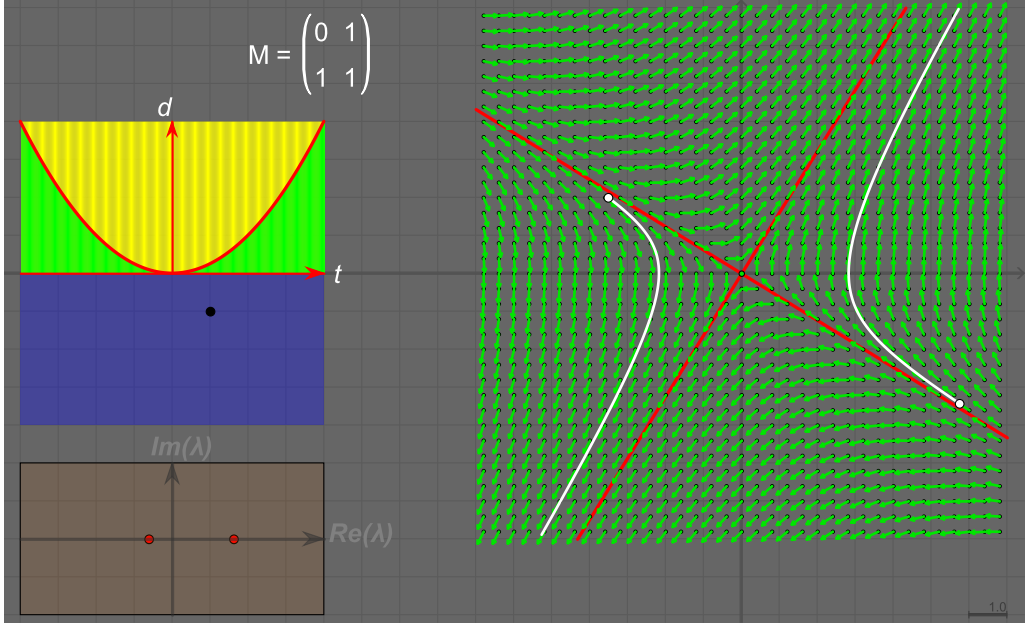
a svojstvene vrednosti imaju sledeći oblik

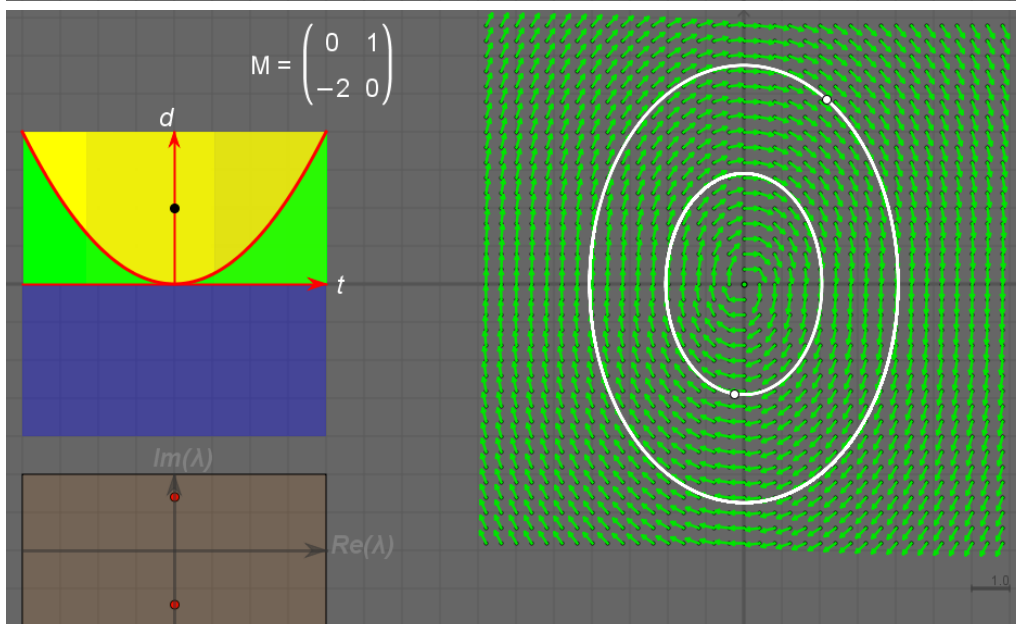
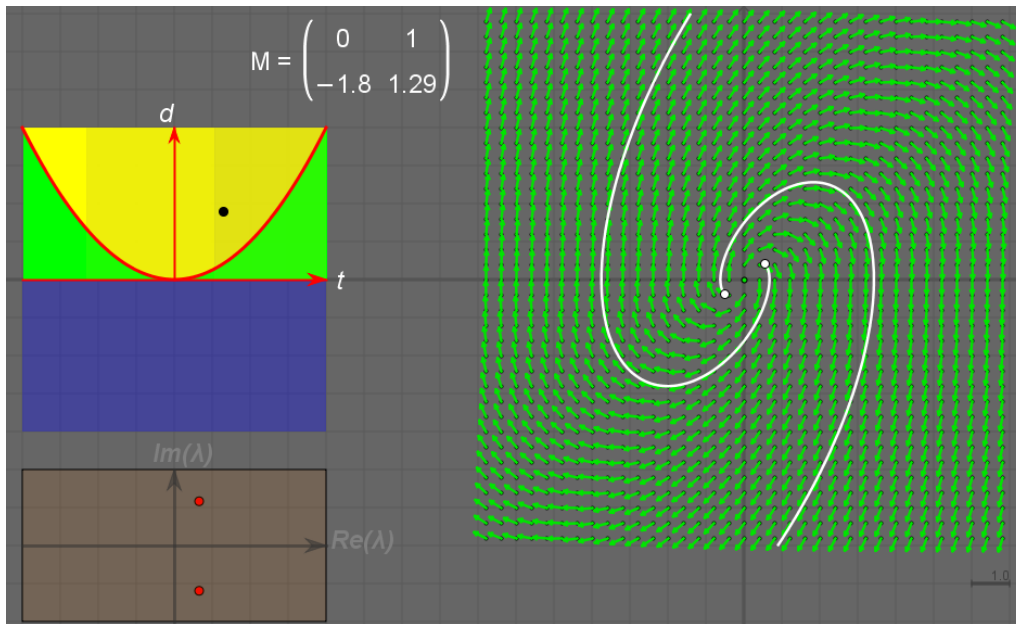
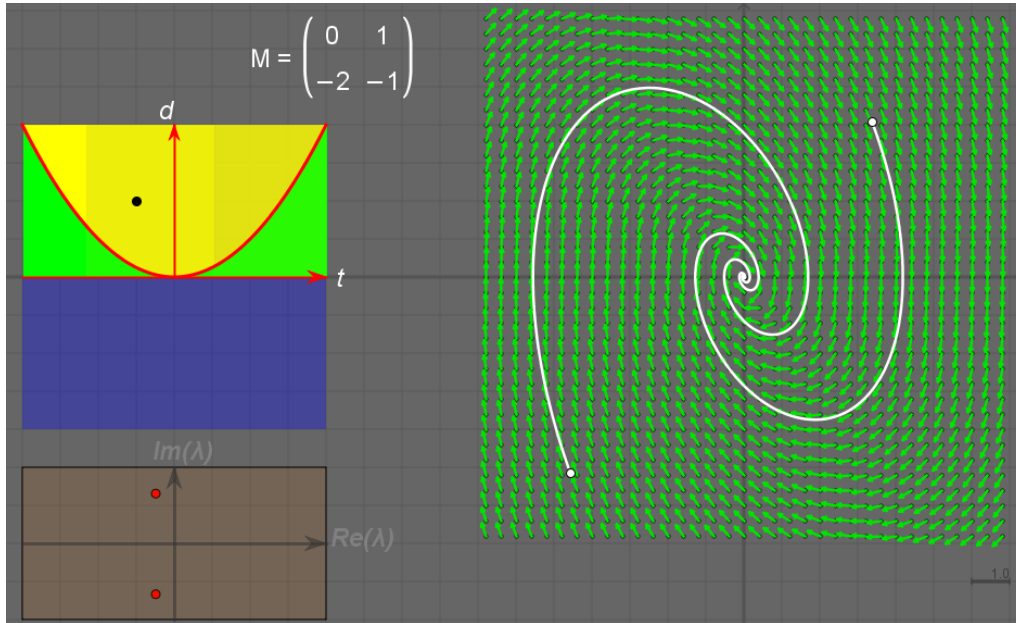
$$\lambda_{1,2} = \frac{1}{2} \left(\operatorname{tr}(A) \pm \sqrt{\operatorname{tr}^2(A) - 4 \det(A)} \right).$$

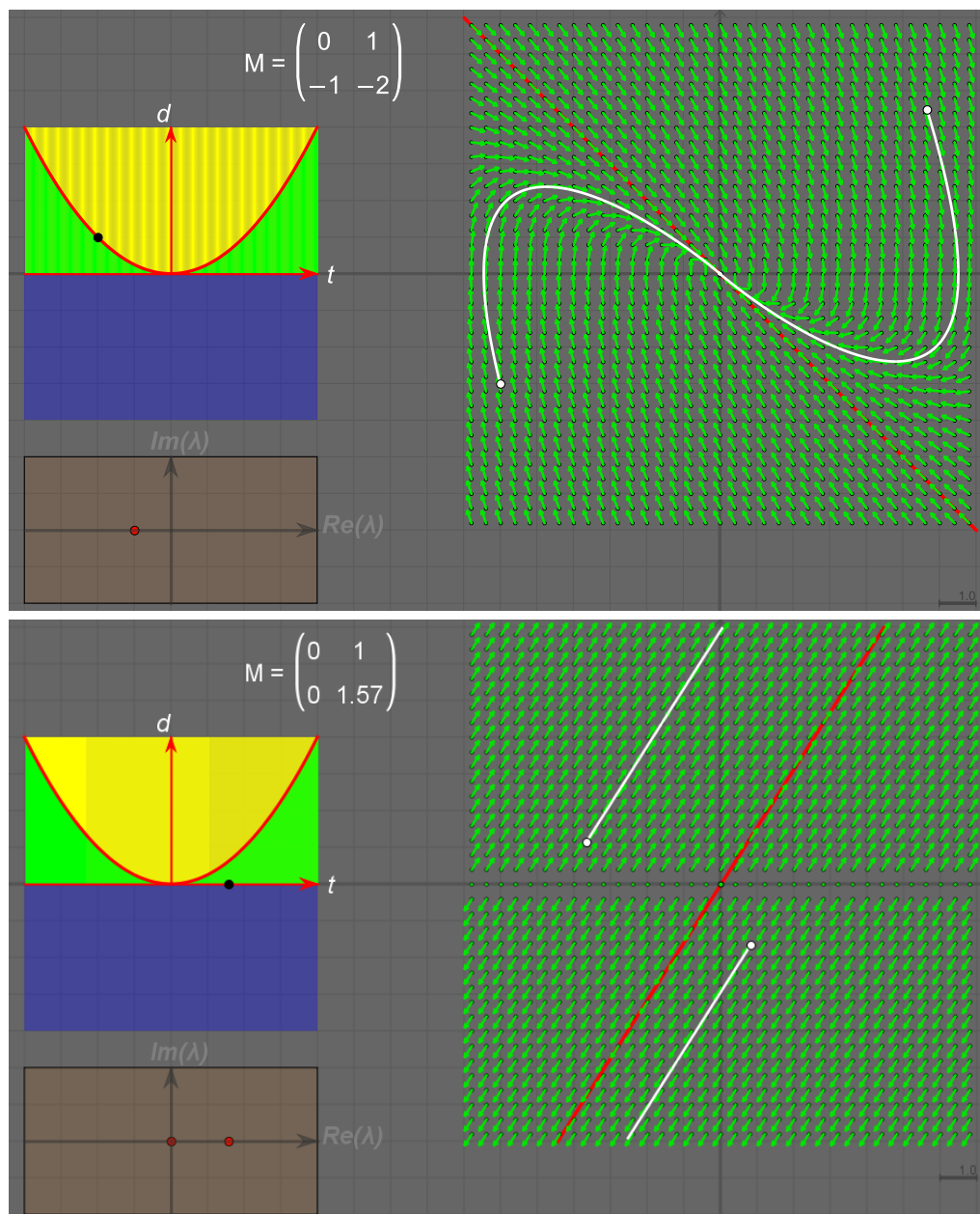
Dakle, bez obzira na činjenicu da je matrica definisana sa četiri realna parametra, svojstveni polinom zavisi samo od dva, traga ($t = \operatorname{tr}(A)$) i determinante ($d = \det(A)$) matrice A . To znači da postoje čitave klase matrica sa istim svojstvenim polinomom. Nama je za potpunu klasifikaciju dovoljno da razmotrimo po jednu matricu iz svake klase. To je moguće uraditi na puno načina, npr. da normalna forma matrice bude $M = \begin{bmatrix} 0 & 1 \\ -d & t \end{bmatrix}$. Prostor parametara normalnih formi je dvodimenzionalan, tzv. $t - d$ ravan. Matrica M , predstavnik klase, je definisana tačkom u $t - d$ ravni, a samim tim i linearno vektorsko polje.

Svojstveni polinom je kvadratni i njegove nule zavise od vrednosti diskriminante $\Delta = \operatorname{tr}^2(M) - 4 \det(M) = t^2 - 4d$. Diskriminanta u $t - d$ ravni definiše parabolu. Unutar parabole, svojstvene vrednosti su konjugovano kompleksne i ne postoje realni svojstveni vektori. Izvan parabole imamo realne svojstvene vrednosti. Ako je, pritom, $d < 0$, rešenja su različitog znaka. Situacije kada je: a) $d = 0$, b) $t = 0, d \geq 0$, c) $\Delta = 0$, nazivamo *degenerisanim*.

Ideja animacije realizovane u Cindereli je sledeća. Animacija sadrži tri panela: jedan predstavlja $t - d$ ravan, drugi je deo kompleksne ravni oko nule na kome su prikazane svojstvene vrednosti matrice M i treći, glavni, je deo fazne ravni sa vektorskim poljem, svojstvenim pravcima, ako postoje i dve tipične fazne trajektorije. Na sledećih osam slika su prikazani svi nedegenerisani tipovi fiksnih tačaka. Ispod svake je naveden njen standardni naziv. Zajednički pridev







Slika 13.12: Tipovi linearnih vektorskih polja sa odgovarajućim fiksnim tačkama: 1. sedlo, 2. stabilni čvor, 3. nestabilni čvor, 4. stabilni fokus, 5. nestabilni fokus, 6. centar, 7. degenerisani čvor 8. neizolovana fiksna tačka

za nedegenerisane fiksne tačke, odomaćen u literaturi posvećenoj dinamičkim sistemima je *hiperbolički*.

Poslednja tri primera nemaju hiperboličko svojstvo. Poslednji primer čak nije ni izolovana fiksna tačka već se čitava apscisa sastoji od fiksnih tačaka. Osnovna odlika hiperboličkih fiksnih tačaka je ta što opisana procedura linearizacija kvalitativno ne menja geometriju vektorskog polja u okolini fiksne tačke. Dodavanje nelinearnih članova (perturbacija) može samo dovesti do male deformacije polja, ali ne i do promene tipa; npr. nestabilni čvor će to ostati i nakon perturbacije. To je stav poznat kao Hartman-Grobmanova teorema. Kod nehiperboličkih fiksnih tačaka to nije slučaj, tako da i najmanja perturbacija dovodi do promene tipa, pa fiksna tačka postaje hiperboličkog tipa. Tipičan primer je centar, koji se javlja kod linearnog harmonijskog oscilatora. I najmanje trenje ili prinudna sila ga transformišu u fokus, privlačni ili odbojni.

13.2.4 Strukturna stabilnost

Pojam stabilnosti se često javlja u kontekstu diferencijalnih jednačina, ali ima dva potpuno različita smisla: kao stabilnost rešenja i strukturalna stabilnost. To je posledica činjenice da rešenje diferencijalne jednačine zavisi od dva podatka, od početnih uslova i oblika same jednačine. Stabilnost rešenja govori o tome da li mala promena početnih uslova utiče na asimptotsko ponašanje rešenja. Ako ne utiče, tada rešenje nazivamo strukturalno stabilnim. Primer stabilnih rešenja su ona u okolini stabilnog fokusa ili čvora. Sedlo je specifično jer izdvaja dva pravca u faznom prostoru: u okolini jednog rešenja su stabilna, a u okolini drugog, nestabilna. Strukturna stabilnost je suptilniji, ali i znatno važniji pojam. S obzirom da su diferencijalne jednačine osnovno sredstvo za opis prirodnih fenomena i procesa, parametre diferencijalne jednačine dobijamo iz eksperimenta, pa usled ograničene tačnosti fizičkih merenja, možemo ih znati samo približno. To znači da najopštiji planarni sistem u primenama ima oblik:

$$\begin{aligned}\dot{x}(t) &= f(x, y) + \varepsilon_1 f_1(x, y), \\ \dot{y}(t) &= g(x, y) + \varepsilon_2 g_1(x, y).\end{aligned}$$

Sa ε_1 i ε_2 su označeni mali realni brojevi, a f_1 i g_1 su nepoznate funkcije. Svaki realni model fizičke pojave mora biti takav da njegovo rešenje kvalitativno ne zavisi od nepoznatih funkcija. Drugim rečima, model mora biti *strukturno stabilan*. Primer strukturalno stabilnih fiksnih tačaka su hiperboličke fiksne tačke. Pojam strukturalne stabilnosti je prvi definisao ruski fizičar Andronov 30-tih godina dvadesetog veka. Koliko je ova ideja uticala na čitavu oblast, rečito govori sledeća šaljiva skica istorije teorije diferencijalnih jednačina.

U istoriji diferencijalnih jednačina postoje tri glavna perioda:

- Njutn: "Data je jednačina. Rešiti je."
- Poenkare: "Data je jednačina. Opisati svojstva njenog rešenja bez njenog rešavanja."
- Andronov: "Nije data jednačina. Opisati svojstva njenog rešenja."

Najvažnija posledica zahteva za strukturalnom stabilnošću je ta što sužava broj tipova faznih portreta koji se kod realnih sistema mogu javiti. Kod planarnih sistema zaključak je sadržan u teoremi Andronova i Pontrjagina (1937).

Teorema Andronova i Pontrjagina

Kod strukturalno stabilnog dinamičkog sistema u ravni, fazne krive mogu težiti samo hiperboličkim fiksnim tačkama, graničnim ciklovima ili beskonačnosti.

Ova teorema je posledica Poenkare-Bendiksonove teoreme iz koje su isključene degenerisane fiksne tačke i separatriski mnogouglovi jer nisu strukturalno stabilni.

Dakle, osnovna lekcija kvalitativne teorije diferencijalnih jednačina u dve dimenzije je sledeća: bez obzira na beskonačan broj i veliku složenost svih diferencijalnih jednačina, jednačina sa kvalitativno različitim rešenjima ima znatno manje, a samo nekoliko njih su strukturalno stabilna i na njih treba obratiti pažnju. Međutim, u prirodi postoje sistemi sa tri i više dimenzija. Do početka 60-tih godina prošlog veka nije se znalo da li i za njih postoji slična teorema koja bi drastično ograničila broj mogućih tipova rešenja. Ali, zahvaljujući radovima matematičara S. Smejla i meteorologa E. Lorenca, znamo da nije tako. U trodimenzionalnom faznom prostoru postoje strukturalno stabilni sistemi sa beskonačnim brojem nestabilnih (u odnosu na početne uslove) orbita. Takve sisteme nazivamo *haotičnim*. Otkriće haosa u teoriji dinamičkih sistema predstavlja svojevrsnu naučnu revoluciju. Ispostavilo se da haotični sistemi nisu izuzetak već pravilo u višedimenzionalnim faznim prostorima. Ali to ne znači da u tom slučaju nije moguća nikakva analiza rešenja i predikcija, već samo da je ona drugog tipa, da se zasniva na metodama teorije verovatnoće.

14 Iteracije i diskretni dinamički sistemi

U ovoj glavi biće reči o geometrijskoj reprezentaciji jednostavnih iterativnih procedura koje se prirodno pojavljuju prilikom aproksimativnog rešavanja jednačina ili izračunavanja vrednosti funkcija. Ideja je da aproksimativni pristup postane jedan od standardnih metoda rešavanja problema na srednjoškolskom nivou školovanja, dok geometrijski pristup služi da (ne)tačnost konkretne aproksimacije učini očiglednom.

Primer 7: Zenonov paradoks

Prvi primer uzastopnih aproksimacija smislio je Zenon iz Eleje i formulisao ga u vidu paradoksa. Paradoks glasi ovako: Početno rastojanje između Ahila i kornjače je 1000 koraka. Iako Ahil trči 10 puta brže od kornjače, on je ne može stići, jer dok Ahil pređe 1000 koraka, kornjača napravi 100 koraka, dok Ahil pređe tih novih 100 koraka, kornjača uspostavi novu razliku od 10 koraka itd. Na Zenonovom primeru ilustrovati metod iterativnog deljenja.

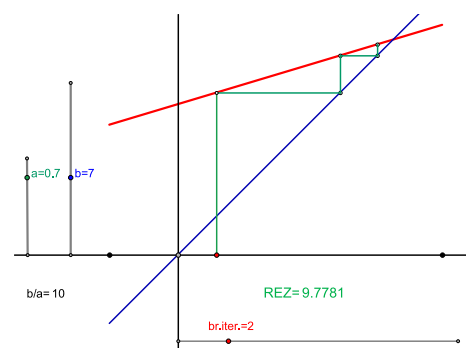
Ovaj jednostavni kinematički zadatak je lako rešiti:

$$10t - t = 1000 \Rightarrow t = 1000 / (10 - 1),$$

tj. rešenje se svodi na deljenje brojeva. Međutim, početnu jednačinu je moguće napisati i na sledeći način: $t = 100 + 10/t$ i rešiti je Zenonovski - uzastopnim aproksimacijama. To je moguće primeniti kod proizvoljnog deljenja $x = b/a$:

$$ax = b \Rightarrow x = (1 - a)x + b.$$

Iterativna procedura je $x_{n+1} = (1 - a)x_n + b$. To je obično algoritam za deljenje primenjen kod digitalnih računara. Procedura je ilustrovana na slici.



Slika 14.1: Iterativna procedura deljenja.

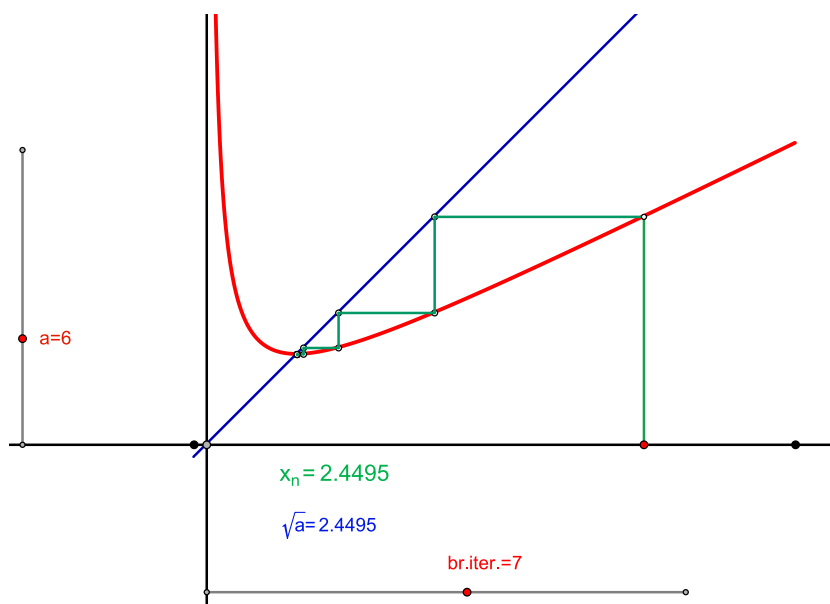
Primer 8: Iterativno korenovanje

Ilustrovati iterativnu proceduru korenovanja.

Zadatak je odrediti x tako da važi $\sqrt{a} = x$. Ako je prva aproksimacija rešenja x_0 , tada je $x - x_0$ mali broj, pa važi

$$a^2 = (x - x_0)^2 \approx x^2 - 2xx_0 \Rightarrow x \approx \frac{1}{2} \left(x + \frac{a}{x} \right).$$

Iterativna procedura je $x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right)$ i ona je ilustrovana na slici.



Treba primetiti da ako je egzaktna vrednost korena data izrazom za geometrijsku sredinu $\sqrt{x \frac{a}{x}}$, aproksimativna vrednost je zamenjena izrazom za aritmetičku sredinu $\frac{1}{2} \left(x + \frac{a}{x} \right)$.

Slika 14.2: Iterativna procedura korenovanja.

Za animaciju su potrebni sledeći geometrijski elementi: horizontalna i vertikalna prava (koordinatne ose), tri tačke na horizontali, jedna (K) koja definiše početnu vrednost funkcije i dve (P, Q) koje služe da definišu levu i desnu granicu nacrtanih grafika. Postoji još jedna slobodna tačka R , tako da normala KR označava početni korak procedure. Postoje i dva klizača, DH i MO čija je uloga objašnjena u samom skriptu. Odovarajući skript je:

```
c=round(4*|D,E|/|D,H|); //klizac koji definise parametar funkcije f
jj=round(20*|M,0|); //klizac koji definise broj iterativnih koraka
f(x):=0.5*(x+c/x); // funkcija ciju fiksnu tacku f(x)=x iterativno odredjujemo
plot(f(x),color->[1,0,0],size->3,start->P.x,stop->Q.x);
plot(x,color->[0,0,0.7],size->2,start->P.x,stop->Q.x);
z=K.x;R.y=f(z);
```

```

repeat(jj,i,AA=[z,f(z)];BB=[f(z),f(z)];CC=[f(z),f(f(z))]);
drawall([BB,CC],color->[1,1,1],size->2.3);
drawall([AA,BB],[BB,CC],color->[0,0.6,0.4],size->2);z=f(z);i++;
//drawtext([CC_2,f(CC_2)-0.04],"$f^n(x)\,=\,$"+CC_2,size->18,color->[0,0.7,0.3]);
drawtext(K+[-0.02,-0.05],"$x_0=\,$"+K.x,size->15,color->[1,0,0]);
drawtext(0+[-0.03,0.03],"$br.iter.="+jj,size->15,color->[0.3,0.3,0.3]);
drawtext(H+[0.02,-0.01],"c="+c,size->15,color->[0.3,0.3,0.3]);
drawtext(G-[1,-0.2],"k="+k,size->15,color->[0,0,0]);

```

Ovaj skript uz odgovarajuću definiciju funkcije f je iskorišćen za rešavanje prethodnog, kao i narednog zadatka.

Primer 9: Iterativno n-korenovanje

Ilustrovati iterativnu proceduru izračunavanja n-tog korena.

Iz Njutnove binomne formule,

$$(x+b)^k = x^k + kx^{k-1}b + \dots,$$

približno rešenje jednačine $\sqrt[k]{a} = x = x_0 + x'$ se svodi na

$$x_0^k + kx_0^{k-1}x' + o(x') = a \Rightarrow x' \approx \frac{a - x_0^k}{kx_0^{k-1}},$$

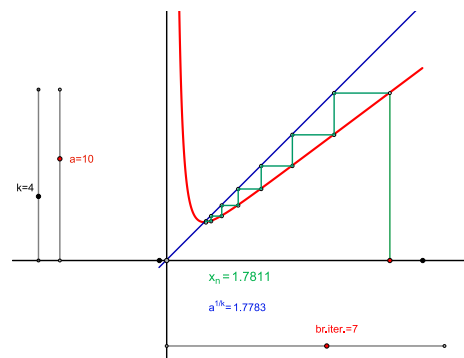
pa je sledeća aproksimacija

$$x_1 = x_0 + \frac{a - x_0^k}{kx_0^{k-1}}$$

itd. Dakle, iterativna procedura glasi

$$x_{n+1} = \frac{a + (k-1)x_n^k}{kx_n^{k-1}},$$

i ona je ilustrovana na slici 14.3.



Slika 14.3: Iterativna procedura za računanje n-tog korena.

Na sličan način je moguće rešiti i transcendentne jednačine, za koje ne postoji analitičko rešenje. Na primer:

Primer 10

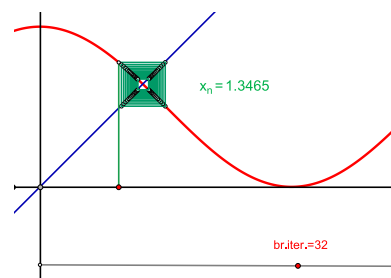
Metodom uzastopnih aproksimacija rešiti jednačinu:

$$x - 1 - \cos x = 0.$$

Iterativni algoritam je

$$x_{n+1} = 1 + \cos x_n.$$

Rešenje je prikazano na slici desno.



Primer 11

Aproksimirati graničnu vrednost izraza

$$\sqrt{2 + \sqrt{2 + \sqrt{2 + \dots}}}$$

Rešenje: procedura $x_{n+1} = x_n^2 - 2$ ili $x_{n+1} = 1 + \frac{2}{x_n}$.

Primer 12

Odrediti graničnu vrednost izraza

$$2 + \frac{1}{2 + \frac{1}{2 + \dots}}$$

Rešenje: procedura $x_{n+1} = \frac{1}{x_n - 2}$.

Primer 13: Stepena kula

Odrediti graničnu vrednost izraza

$$\sqrt{2}^{\sqrt{2}^{\sqrt{2}^{\dots}}}$$

Rešenje: procedura je $x = \sqrt{x}^x$, logaritmovanjem $\ln x = x \ln x / 2$.

U vezi sa procedurom uzastopnih aproksimacija nameću se tri pitanja:

- Da li uvek niz aproksimacija x_0, x_1, x_2, \dots konvergira ka nekom broju ζ ?
- Ako važi $\lim_{n \rightarrow \infty} x_n = \zeta$, da li je ζ istovremeno i rešenje jednačine $x = f(x)$?
- Koliko brzo (efikasno) niz x_0, x_1, x_2, \dots konvergira ka rešenju od $x = f(x)$?

Odgovor na drugo pitanje je trivijalan: da ako je funkcija f neprekidna. Odgovor na prvo pitanje je negativan: kontraprimer je

$$x = 10^x - 2,$$

gde za $x_0 = 1$ dobijamo niz $x_1 = 8, x_2 = 10^8 - 2, \dots$ koji je divergentan. Ali ako prethodnu jednačinu prepíšemo na sledeći način

$$x = \log(x + 2),$$

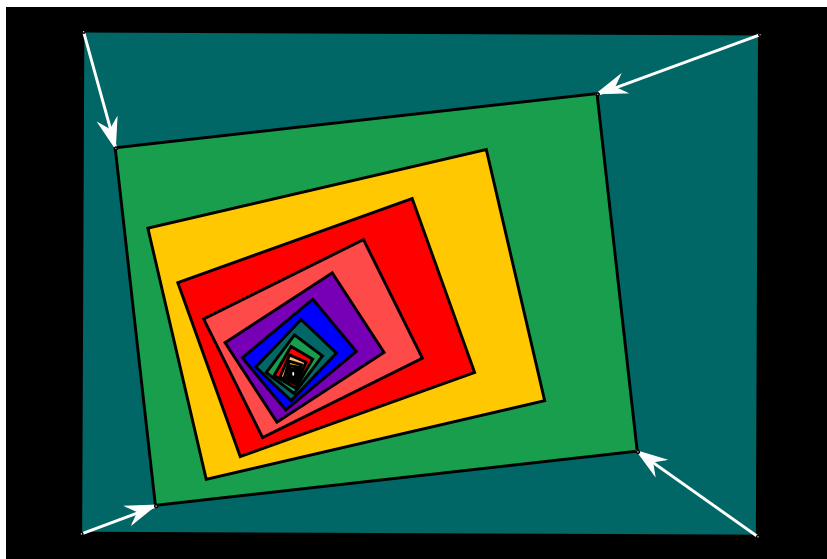
procedura konvergira i dobija se rešenje $x = 2.38$. Dovoljno je razmotriti rekurzivnu proceduru na grafiku. Ona se sastoji od naizmeničnih kretanja duž vertikale i horizontale između grafika $y = x$ i $y = f(x)$. Ako je grafik oko $x = \zeta$ pozitivnog nagiba, tada je putanja stepeničasta, a ako je negativnog, putanja je spiralnog oblika. Postavlja se samo pitanje da li se naredne tačke niza x_i približavaju ili udaljavaju od tačke preseka grafika $y = x$ i $y = f(x)$. Ako posmatramo preslikavanje f kao preslikavanje realne se na samu sebe $f : \mathbb{R} \mapsto \mathbb{R}$, i ako je f neprekidno, tada se interval uvek preslikava u interval.

Ukoliko za svaki par $x_1, x_2 \in [a, b]$ postoji broj $0 < q < 1$ tako da važi

$$|f(x_1) - f(x_2)| < q|x_1 - x_2| < 1,$$

tada kažemo da je preslikavanje f *kontraktibilno* na $[a, b]$. Tada za svako $x_0 \in [a, b]$ niz x_0, x_1, x_2, \dots , gde je $x_{n+1} = f(x_n)$ konvergira ka fiksnoj tački od f , tj. tački za koju važi $x = f(x)$.

Osobinu kontraktibilnosti koja definiše atraktore na delovima domena koji konvergiraju ka fiksnim (stacionarnim) tačkama.



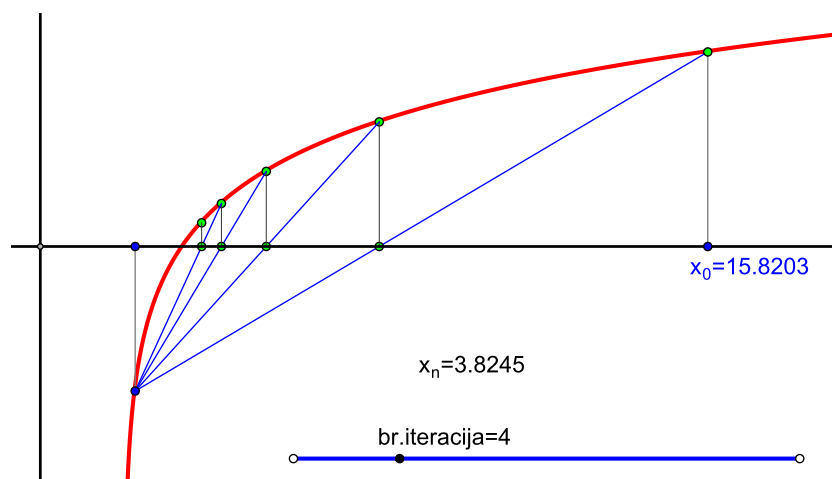
Slika 14.4: Fiksna tačka Euklidske sličnosti, sa koeficijentoma manjim od 1.

Različite verzije iterativnih algoritama se koriste za određivanje nule funkcije.

Primer 14: Metoda sečice

Ilustrovati primenu metoda sečice, za približno određivanje nula funkcije.

Procedura je ilustrovana na primeru funkcije $f(x) = 2 \ln(2x - 4) - 2$.
Iterativna procedura je



Slika 14.5: Ilustracija iterativnog metoda sečice.

$$b_{n+1} = b_n - f(b_n) \frac{b_n - b}{f(b_n) - f(b)},$$

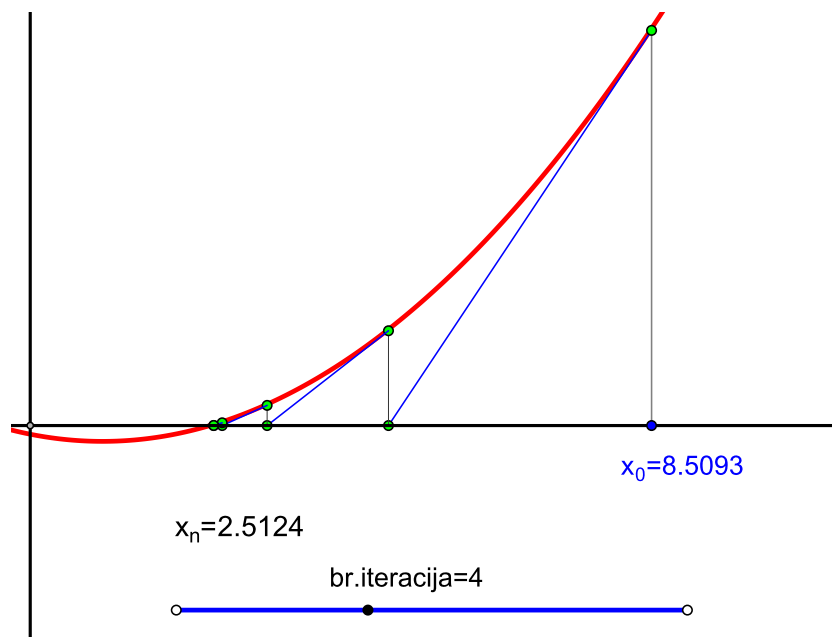
gde je $b = x_0$.

Primer 15: (Njutnov) Metod tangente

Ilustrovati primenu metoda tangente za približno određivanje nula funkcije.

Metod je ilustrovan na primeru funkcije $f(x) = 0.1(x - 2)^2 + 0.2x - 0.5$.
Iterativna procedura je

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$



Slika 14.6: Ilustracija iterativnog metoda tangente.

Iteracije su osnovni primer dinamičkih sistema - diskretni dinamički sistemi. Numeričke procedure, poput izračunavanja kvadratnog korena, kod koje je funkcija koja se iterira $f(x) = 1/2(x + a/x)$ ili Njutnova za izračunavanje nule funkcije f , kod koje je iteraciona funkcija $g(x) = x - f(x)/f'(x)$, su standardni primeri funkcione iteracije.

Primetiti da je procedura za izračunavanje kvadratnog korena specijalan slučaj Njutnove za $f(x) = x^2 - a$.

Primer 16: Iteracija 1

Ako je $f(x) = x + 2\sqrt{x} + 1$, $x \geq 0$, odrediti $f^n(x)$.

Odredimo n -tu iteraciju funkcije metodom konjugacije. Neka je $f = h^{-1} \circ g \circ h$. Tada je i $f^n = h^{-1} \circ g^n \circ h$. Najpre uklonimo koren iz izraza smenom $y = h(x) = \sqrt{x}$, a odatle je i $x = h^{-1}(y) = y^2$. Funkcija sada glasi:

$$f \circ h^{-1}(x) = f(h^{-1}(x)) = y^2 + 2y + 1 = (y + 1)^2,$$

pa je

$$h \circ f \circ h^{-1}(x) = \sqrt{(y + 1)^2} = y + 1 = g(y).$$

Sada je jednostavno odrediti proizvoljni stepen: $g^n(y) = y + n$, a odatle i

$$f^n(x) = h^{-1} \circ g^n \circ h(x) = (\sqrt{x} + n)^2.$$

Primer 17: Iteracija 2

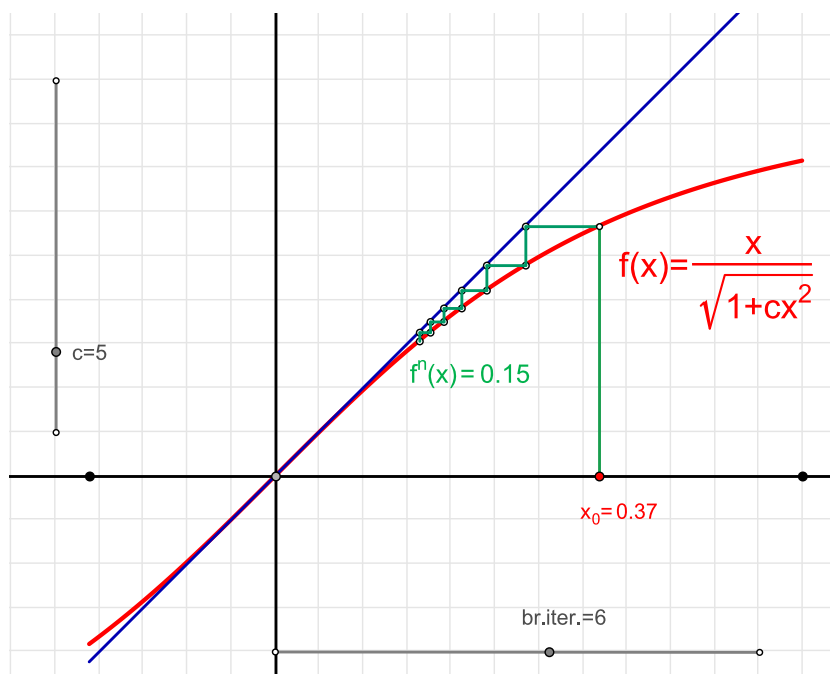
Konstruisati funkciju f koja preslikava $x = 1/\sqrt{y}$ u $f(x(y)) = 1/\sqrt{y+c}$. Tada je jasno da će ona nakon n iteracija preslikavati x u $f^n(x) = 1/\sqrt{y+nc}$.

Iz same postavke, $y(x) = 1/x^2$, dok je $g(y) = y + c$. Tražena funkcija je

$$f(y) = \frac{1}{\sqrt{y+c}} = \frac{1}{\sqrt{c+1/x^2}} = \frac{x}{\sqrt{1+cx^2}}.$$

Prema tome,

$$f^n(x) = \frac{x}{\sqrt{1+ncx^2}}.$$

**Primer 18: Iteracija 2'**

Metodom konjugacije odrediti n -tu iteraciju funkcije

$$f(x) = \frac{x}{\sqrt{1+cx^2}}.$$

Primer 19: Iteracija 3

Metodom konjugacije odrediti n -tu iteraciju funkcije

$$f(x) = \frac{2x}{1-x^2}.$$

Za razliku od prethodnih primera, ne postoji domen koji ova funkcija preslikava na samog sebe, jer npr. vrednosti $f(1)$ i $f(-1)$ nisu definisane. Sam izraz podseća na tangens dvostrukog ugla, pa je očigledan izbor funkcije h , tj., iz

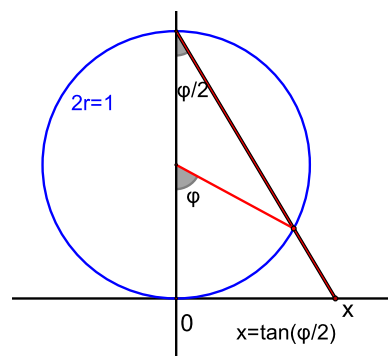
$$\tan 2y = \frac{2\operatorname{tgy}}{1-\operatorname{tg}^2y},$$

sledi $y = h(x) = \operatorname{arctg}x$. Funkcija g je tada $g^n(y) = 2^n y$. Konačno je

$$f^n(x) = \operatorname{tg}(2^n \operatorname{arctg}x).$$

Funkcija ima i očiglednu geometrijsku interpretaciju.

Domen je bio $[0, \infty]$



Slika 14.7: interpretacija funkcije $f(x) = \operatorname{tg}(\varphi/2)$. Oznaka $\tan \equiv \operatorname{tg}$.

14.1 Orbite 1-dim diskretnih sistema

Pored toga što su dobra aproksimacija kontinualnih dinamičkih sistema, diskretni dinamički sistemi (funkcionalne iteracije) su važni jer omogućavaju razumevanje univerzalnih dinamičkih fenomena. Nekoliko sledećih zadataka pomoću grafičke analize ilustruje različite tipove orbita na primeru *logističkog preslikavanja* (R. May 1976.):

$$f(x) = rx(1 - x).$$

Postepenim povećanjem parametra r moguće je analizirati prelazak sa stabilne na nestabilne fiksne tačke, udvajanje perioda ciklusa (tangenta bifurkacija) sve do haotičnih orbita.

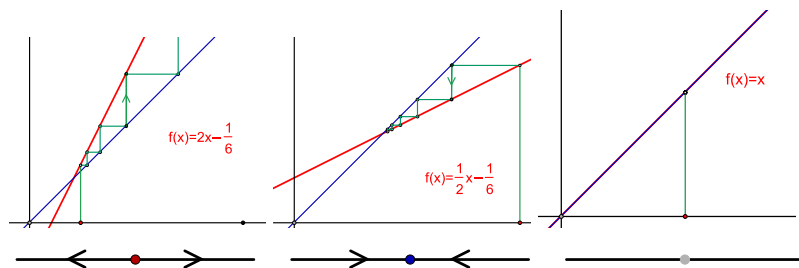
Počnimo od najvažnijeg tipa dinamičkog ponašanja: fiksne tačke. Jednostavna topologija 1D segmenta implicira osnovni stav da *ako je preslikavanje $F : [a, b] \mapsto [a, b]$ neprekidno, tada postoji fiksna tačka za F u $[a, b]$* . Sledeće pitanje je kako se funkcija ponaša u okolini fiksne tačke, tj. u slučaju male perturbacije fiksne tačke.

Primer 1: Hiperboličko svojstvo

Grafički demonstrirati važenje teoreme.

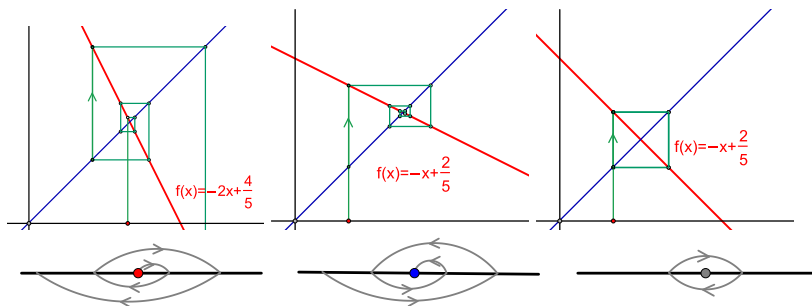
Neka je x_0 fiksna tačka preslikavanja F . Tada je x_0 privlačna fiksna tačka ako je $|F'(x_0)| < 1$, a odbojna fiksna tačka za $|F'(x_0)| > 1$. (Odbojne i privlačne fiksne tačke nazivamo zajedničkim nazivom "hiperboličke"). Za $|F'(x_0)| = 1$, dobijamo neutralnu fiksnu tačku. Skicirati dinamiku u faznom prostoru - fazna prava.

U okolini tačke x_0 funkciju možemo aproksimirati njenim prvim izvodom. U tom slučaju treba samo izvršiti grafičku analizu dinamike između dve prave $y = x - x_0$ i $y = F'(x_0)(x - x_0)$ oko tačke $x = x_0$. U tri slučaja koja su prikazana na slici ispod, prvi izvod je pozitivan $F'(x_0) > 0$. U slučaju da je prvi izvod manji od nule, tip fiksne



Slika 14.8: Grafička analiza dinamike oko (sa leva na desno) odbojne, privlačne i neutralne fiksne tačke. Dinamika na faznoj pravoj je tipa čvorni izvor/uvir. U slučaju $F'(x_0) = 1$ sve tačke su fiksne.

tačke i dalje zavisi od apsolutne vrednosti prvog izvoda, ali je dinamičko ponašanje na faznoj pravoj spiralnog tipa (spiralni izvor/uvir).

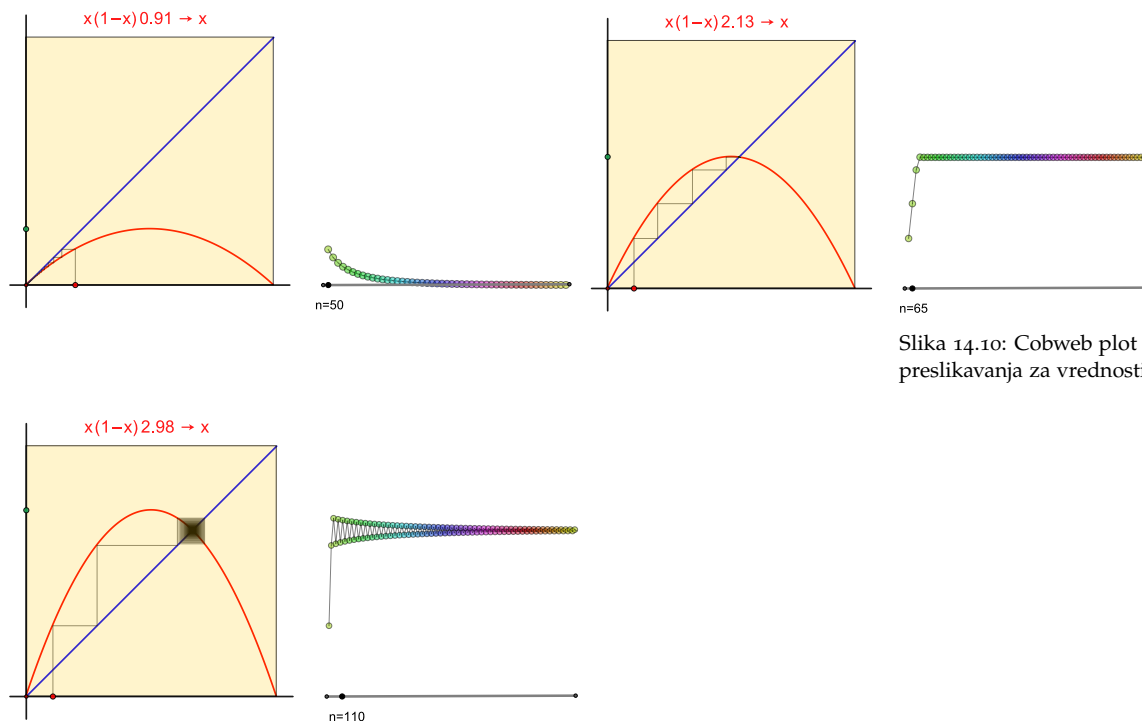


Slika 14.9: Grafička analiza dinamike oko (sa leva na desno) odbojne, privlačne i neutralne fiksne tačke, u slučaju negativne vrednosti prvog izvoda. U ovom slučaju, za $F'(x_0) = -1$, samo jedna tačka je fiksna.

Pod grafičkom analizom se obično podrazumeva cobweb ("paukova mreža") plot, ili Verhulstov (Verhulst) dijagram. U sledećem zadatku je ilustrovana primena cobweb plot na analizu fiksni tačaka logističkog preslikavanja.

Primer 2: Fiksne tačke logističkog preslikavanja

Ispitati postojanje jedinstvene fiksne tačke logističkog preslikavanja na intervalu $[0, 1]$ za vrednost parametra r u oblasti $[0, 4]$.



Slika 14.10: Cobweb plot logističkog preslikavanja za vrednosti $0 < r < 3$.

Na slikama su priložene cobweb plotovi za tri različite vrednosti $0 < r < 3$. Pažljiva analiza pokazuje da za $0 \leq r \leq 1$, postoji jedna

fiksna tačka $r = 0$. Za $1 < r < 3$ postoji takođe jedna fiksna tačka: $(r - 1)/r$.

Ako dinamičko ponašanje u fiksnoj tački nazovemo *stacionarnim*, sledeći važan tip ponašanja je *periodični*. On se javlja u slučaju kada postoji *ciklus* dužine k , koji se sastoji od tačaka $\{x_0, x_1 = F(x_0), x_2 = F(x_1), x_3 = F(x_2), \dots, x_k = F(x_{k-1}, x_0, x_1, \dots)\}$. Minimalno k za koje se ciklus ponavlja se naziva *osnovnim periodom*. Očigledno je da tačka x pripada ciklusu dužine n ukoliko je fiksna tačka od F^n , tj. ukoliko važi $F^n(x) = x$. Naravno, to važi za sve tačke iz istog ciklusa (one su dinamički ekvivalentne).

Za animaciju su potrebni sledeći geometrijski elementi: kvadrat (4 duži) sa slobodnom tačkom C na horizontalnoj stranici, tačkom B na vertikalnoj stranici i klizač GH sa pokretnom tačkom H . Odgovarajuću skript je:

```
x=0;
linecolor([1,0,0]);
if(Text2.pressed,lbd=1+sqrt(6),
  if(Text3.pressed,lbd=3.54409,
    if(Text4.pressed,lbd=3.56995,
      if(Text5.pressed,lbd=3.82843,
        if(Text6.pressed,lbd=4,lbd=4*B.y);//definisana su pomocna dugmad za specijalne vrednosti od r
      )));
f(x):=x*(1-x)*lbd;
plot(f(x),start->A.x,stop->F.x,size->2);
draw([0,0],[1,1],size->2,color->[0,0,1]);

if(K.x>=H.x-0.01,K.x=H.x-0.01);
itr=20+15*round(100*(|G,K|)); //klizac GK definise broj iterativnih koraka
drawtext(K.xy+[-0.05,-0.1],"n="+itr,size->15);
x=C.x;// C.x definise pocetnu vrednost iteracije
y=0;
xx=1.2;
linecolor([0,0,0]);
ii=1;
repeat(itr,
  y1=x*(1-x)*lbd;
  alpha(0.5);
  draw([x,y],[x,y1],size->1);
  draw([x,y1],[y1,y1],size->1);
  pointsize(log(3000/itr));
  pointcolor(hue(xx)*0.8);
  if(ii>1,draw([xx,y]);if(Text1.pressed,draw([[xx,y],[xx+1/itr,y1]])));
  x=y1;
  y=y1;
```

```

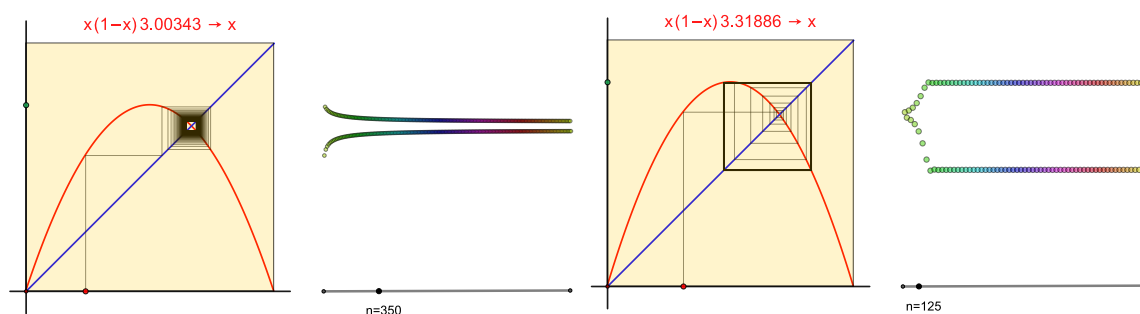
alpha(0.8);
xx=xx+1/itr;ii=ii+1;
);

```

Primer 3: 2-ciklusi logističkog preslikavanja

Ispitati postojanje 2-ciklusa logističkog preslikavanja na intervalu $[0, 1]$ za vrednost parametra r u oblasti $[3, 4]$.

Logističko preslikavanje ima dve fiksne tačke za sve vrednosti $r \in [3, 1 + \sqrt{6} \approx 3.44949]$. Kaže se da u tački $r = 1 + \sqrt{6}$ dolazi do prve bifurkacije. Zapažimo da je tačke koje pripadaju ciklusima dužine



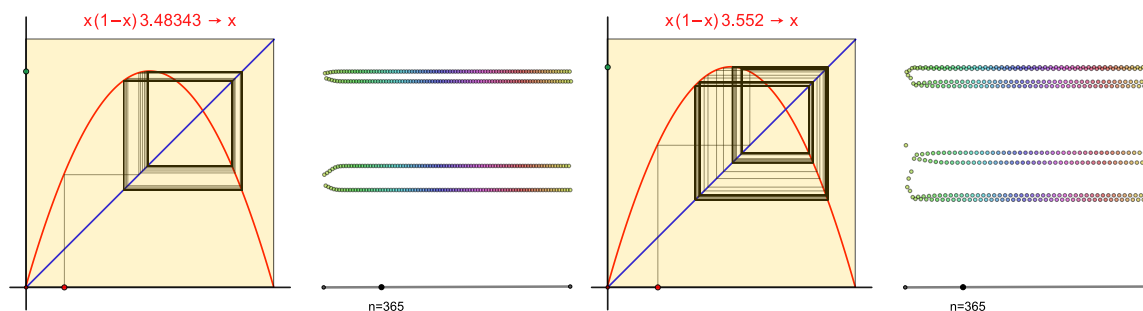
2 jednostavno prepoznati - to su tačke na grafiku funkcije F koje se uzajamno simetrične u odnosu na pravu $y = x$. Da li je na sličan način moguće otkriti tačke ciklusa dužine 4?

Primer 4: 4-ciklusi logističkog preslikavanja

Odrediti približnu vrednost parametra r za koju dolazi do povećanja dužine ciklusa sa 4 na 8, tj. dolazi do druge bifurkacije.

Približna vrednost je $r \approx 3.54409$. Može se pokazati da je navedeni broj koren polinoma stepena 12, i njegove decimale su u Online enciklopediji celobrojnih nizova (OEIS) zaveden pod šifrom A086181.

Slika 14.11: Cobweb plot logističkog preslikavanja za vrednosti $3 < r < 1 + \sqrt{6}$.



Slika 14.12: Primer 4-ciklusa i 8-ciklusa nakon druge bifurkacije.

Primer 5

Odrediti količnik vrednosti dužina parametarskog intervala $[r_i, r_{i+1}]$ za nultu i prvu bifurkaciju, tj. $\delta_i = \frac{r_{i-1} - r_{i-2}}{r_i - r_{i-1}}$. Ako pretpostavimo da je $\delta_3 \approx \delta_4$, proceniti vrednost r_4 .

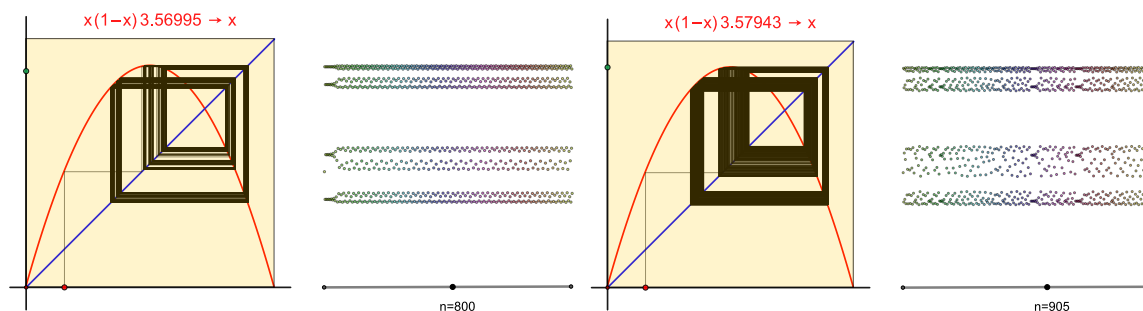
Prvi količnik je

$$\delta_3 \approx \frac{3.44949 - 3}{3.54409 - 3.44949} \approx 4.75148.$$

Vrednost za r_4 je 3.5644073, a za $r_5 \approx 3.5687594$. Feigenbaum je 1978. pokazao da ovaj niz konvergira ka $\delta \approx 4.66920$ koja je nazvana njegovim imenom.

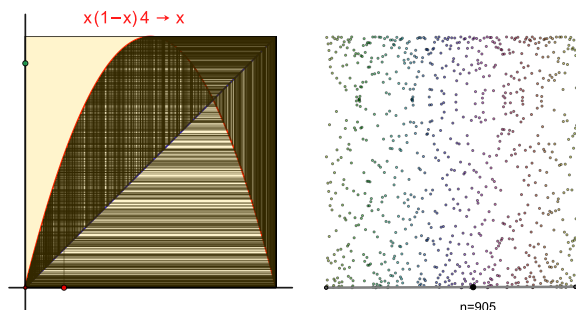
Primer 6: Haos

Ispitati dinamiku logističkog preslikavanja za parametre $r \geq 3.56995$.



Slika 14.13: Haotično ponašanje logističkog preslikavanja.

U slučaju $r = 4$ je moguće i egzaktno rešiti dinamički problem: tada je $x_n = \sin^2(2^n \theta \pi)$, $\theta = 1/\pi \arcsin(\sqrt{x_0})$.

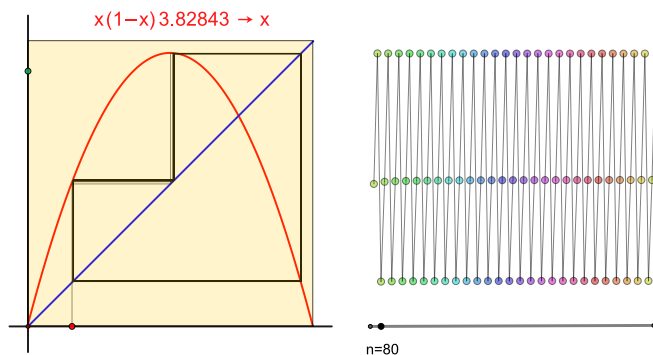


Kompleksnost haotičnog ponašanja se ogleda u finoj zavisnosti od početne vrednosti x_0 . Za svaku vrednost $x_0 \in [0, 1]$ orbite su ekstremno kompleksne ali i međusobno potpuno različite. Sa druge strane, unutar parametarske oblasti $r > 3.56995$, postoje oblasti u kojima je ponašanje ciklično, npr. $3n$ -ciklično.

Primer 7: Prozor u haosu

Odrediti vrednosti parametra r unutar haotične oblasti u kojima se nalaze prozori 3 -ciklusa, opštije $3n$ -ciklusa.

Za $r = 1 + \sqrt{8} \approx 3.82843$ imamo prozor 3 -ciklusa.

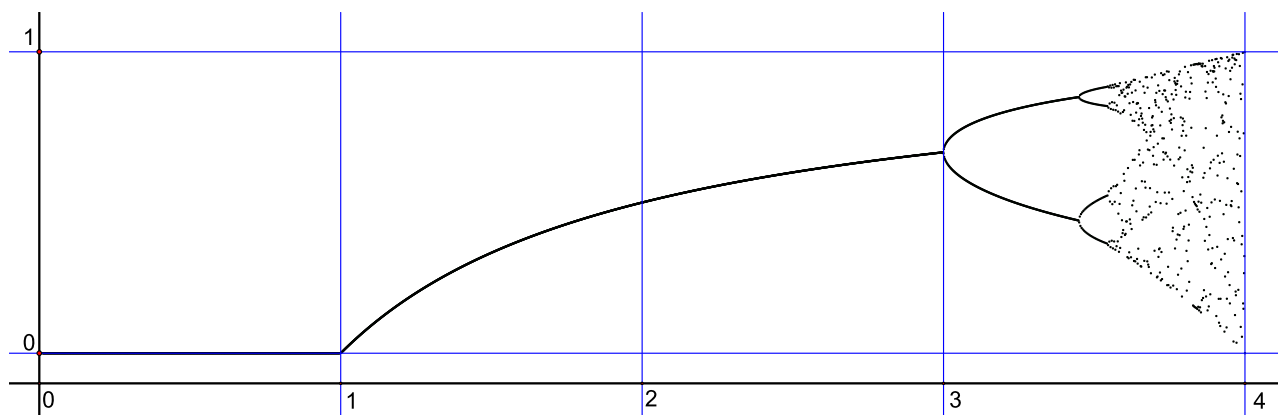


Slika 14.14: 3-ciklični prozor u oblasti haotičnog ponašanja.

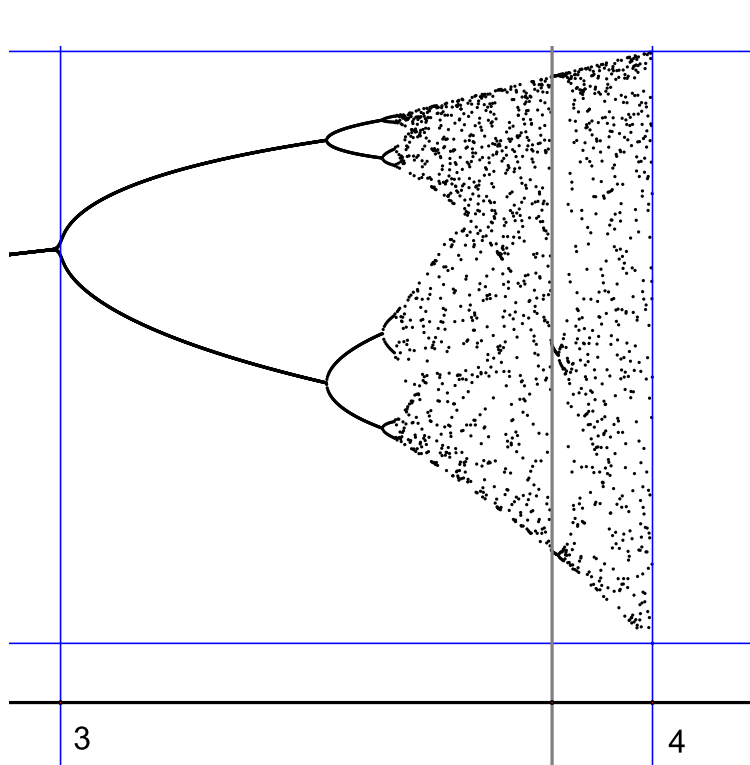
Pregled globalnog dinamičkog ponašanja najlakše je steći posredstvom *bifurkacionog dijagrama*. Na horizontalnoj osi je parametar r , a na vertikalnoj vrednosti fiksnih i cikličnih tačaka.

Primer 8: Bifurkacioni dijagram

Prikazati bifurkacioni dijagram logističkog preslikavanja.



Slika 14.15: Celokupni bifurkacioni dijagram za $r \in [0, 4]$.



Slika 14.16: Detalji bifurkacionog dijagrama sa vertikalom koja ukazuje na prozor 3-ciklusa.

14.2 Topološka konjugacija

Kao što je već napomenuto u prvom paragrafu ovog poglavlja, metod konjugacije je koristan pri određivanju eksplicitnih izraza funkcionalnih iteracija. Međutim, koristan je i pri kvalitativnoj analizi dinamičkog ponašanja.

Za preslikavanja $f : X \mapsto X$ i $g : Y \mapsto Y$ kažemo da su topološki konjugovana, ako postoji homeomorfizam h iz X u Y i važi $g = h^{-1} \circ f \circ h$.

Primer 1: Svojstva topološke konjugacije

Dokazati da važenje sledećih stavova:

- Ako su f i g konjugovane, tada su i f^n i g^n , $n \in \mathbb{N}$ konjugovane;
- Ako je x_0 fiksna tačka za f , tada je i $y_0 = h(x_0)$ fiksna za g i pritom je $f'(x_0) = g'(y_0)$.

Primer 2: Konjugacija 1

Neka je $r > 0$. Pokazati da je $f(x) = rx(1-x)$ konjugovana sa $g(x) = x^2 + c$ za pogodno odabranu vrednost c .

Primer 3: Konjugacija 2

Pokazati da je $f_4(x) = 4x(1-x)$ konjugovana sa "šator" preslikavanjem definisanim sa

$$t(x) = \begin{cases} 2x, & 0 \leq x \leq 0.5, \\ 2(1-x), & 0.5 \leq x \leq 1, \end{cases}$$

posredstvom konjugacione funkcije $h(x) = \sin^2(\pi x/2)$.

Šator funkcija se pri iteraciji ponaša poput jednostavnije funkcije - udvajanja po modulu 1 ili frakcionom udvajanju. Definisana je izrazom

$$d(x) = 2x \bmod(1) \equiv \{2x\},$$

gde je sa $\{x\}$ označen frakcioni deo realnog broja x .

Primer 4: Binarni šift

Odrediti dejstvo frakcionog udvajanja na realni broj zapisan u binarnom sistemu. Zašto se ona često naziva "binarni šift"?

Pojava prozora sa orbitama perioda 3 kod logističkog preslikavanja nije slučajna. Naime teorema Li-Yorke-a tvrdi da: *Ako je $f : I \mapsto I$ neprekidna i ukoliko postoji ciklus dužine 3, tada postoji i ciklusi svih ostalih dužina.* Tačke ciklusa dužine 3 nije moguće jednostavno uočiti na grafiku funkcije.

Funkciju koja ima period 3 je lako konstruisati polazeći od troelementne ciklične permutacije (permutacija je bijekcija), koja se zatim deo-po-deo linearno proširi na čitav interval.

Primer 5: Diskretna f-ja sa periodom 3

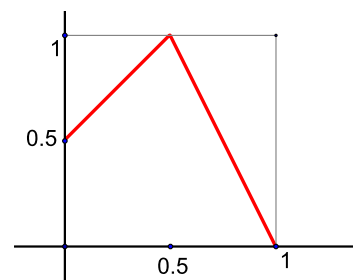
Polazeći od permutacije (012) skaliranjem sa 1/2 i linearnim proširivanjem konstruisati funkciju sa periodom 3.

Funkcija je definisana sa

$$f(x) = \begin{cases} x + \frac{1}{2}, & x \in [0, \frac{1}{2}], \\ 2(1-x), & x \in [\frac{1}{2}, 1]. \end{cases}$$

Primer 6

Analizirati dinamiku funkcije definisane u prethodnom primeru.



15 Iteracije, fraktali i diskretna dinamika

Računarski algoritam je potpuno precizna verzija kuvarskog recepta: sastoji se iz niza jasnih instrukcija - elementarnih upustava. Međutim, upotreba računara postaje efektivna tek kad se zahteva ponavljanje jedne ili više instrukcija sve dok neki uslov ne bude zadovoljen. Takva procedura se naziva *iteracija*, a deo koda koji se više puta izvršava *petlja* (*loop*).

Na iterativnim procedurama su obično zasnovani i kombinatorni i numerički algoritmi, tako da bezbedno možemo tvrditi da su iterativne procedure osnova svih algoritama. Često se nameće pitanje vizualizacije iterativnih procedura. Naravno, uvek je datu proceduru moguće na neki način prikazati skupovim tačaka ili drugih geometrijskih objekata. Pitanje je suptilnije: da li postoji nešto univerzalno, zajedničko svim iterativnim algoritmima, a da je to moguće odmah zapaziti na slici. Odgovor je DA i dobijen je tek na prelasku iz 70-tih u 80-te godine XX veka, u vreme kada je kompjuterska grafika dostigla potreban stepen razvoja. Sadržan je u dva centralna pojma, *fraktal* i *haos*, koji u to vreme postaju enormno popularni i sveprisutni kako u stručnoj, tako i u popularnoj naučnoj literaturi. Potrebno je istaći da je njihovo poreklo u problemima dinamike, koji su na računaru implementirani kroz odgovarajuće iterativne procedure.

Započnimo analizu od jednostavnog, ali i veoma poznatog primera.

Primer 7: Haotična igra

Pera je neodlučan da li da ide u biblioteku (A), u bioskop (B) ili na bazen (C). Rešio je da baca kockicu, pa ako padne 1 ili 2 ide u biblioteku, 3 ili 4 u bioskop, 5 ili 6 na bazen. Međutim, kada dođe na pola puta ka svojoj destinaciji, zabeleži svoj položaj i ponovo baca kockicu. Odrediti skup mogućih zabeleženih pozicija ako ovakvo kretanje sadrži a) 100, b) 1000, c) 8000 bacanja kockice.

Jasno je da će se nakon prvih nekoliko poteza obeležena tačka naći unutar trougla ABC . Geometrijski elementi su tačke A, B, C (temena trougla), D (početna pozicija). U ulozi klizača je tačka G , fiksirana

Uslov je veoma bitan jer se u protivnom dobija tzv. *većita petlja*. U programerskom slengu se algoritam koji sadrži većitu petlju naziva "shampoo algoritam."

Shampoo algoritam:

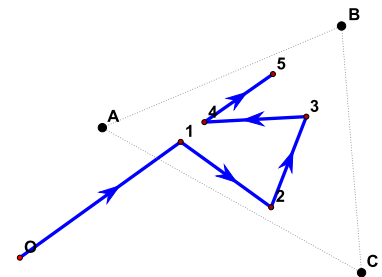
Q: Why did the computer scientist die in the shower?

A: He followed the instructions on the shampoo bottle (lather, rinse, repeat).

Nužnost postojanja uslova su shvatili i neki proizvođači šampona. U to se možemo uveriti proverom poledine nekih šampona, npr. "Head and shoulders":

Directions

- for best results use at least twice a week or as directed by a doctor.
- for maximum dandruff control, use every time you shampoo.
- shake before use.
- wet hair, massage onto scalp, rinse, repeat if desired.



Slika 15.1: Perina dilema: zabeležene pozicije su označene crvenim tačkama

za y – osu, koja nije prikazana na slici desno. Prateći skript za datu konstrukciju, koji omogućava rekurzivnu konstrukciju skupa tačaka sa variranjem broja rekurzivnih koraka ("bacanja kockice") je sledeći.

```
t=500*round(G.y);\y-koord. tacke G definise broj iteracija
drawtext([1,12],"broj iteracija =" +t,
color->[1,1,1],size->15);
```

```
pointsize(1);i=1;
repeat(t,n=randomint(3)+1;D=(D+[A,B,C]_n)/2;F
if(i>10,draw(D,color->hue(n/3),border->>false));i=i+1);
```

Dobijena figura se naziva *trougao Serpinjskog* (Serpinski gasket) i jedan je od najpoznatijih primera fraktala. Nastaje kao fiksni skup *iterativnog funkcionalnog sistema*. Razmotrimo sam proces kreiranja figure kao rezultat beskonačne kompozicije tri preslikavanja: f_A preslikava sve tačke trougla ABC u levi polutrougao AB_1C_1 (to je trougao sa temenima AB_1C_1 , gde je B_1 sredina stranice AB , a C_1 sredina stranice AC), f_B preslikava sve tačke trougla ABC u gornji polutrougao A_1BC_1 , i f_C preslikava sve tačke trougla ABC u desni polutrougao A_1B_1C , koja deluje na početnu tačku O . Trougao Serpinjskog se sastoji od tačaka dobijenim dejstvom beskonačne kompozicije $f_{X_1} \circ f_{X_2} \circ f_{X_3} \cdots$, $X_i \in \{A, B, C\}$ na početnu tačku (začetak-seed). Prema tome, svakoj tački trougla Serpinjskog je moguće jednoznčno pridružiti (beskonačnu) reč $X_1X_2X_3 \cdots$. Računarska slika samo nagoveštava izgled fraktala tako što postepeno izdvaja deo prostora koji ne pripada fraktalu.

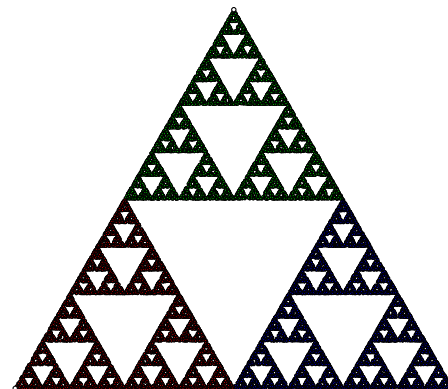
Fraktalni objekti su vredni proučavanja zbog njihove *univerzalnosti*. Naime, na jedne te iste fraktalne objekte se nailazi u, na prvi pogled potpuno različitim problemima.

Primer 8: Binarni Paskalov trougao

Generisati Paskalov trougao po modulu 2, tj. na mestu parnih brojeva nacrtati beli, a na mestu neparnih crni kvadratić, za $n = 2^p - 1$, a) $p = 4$, b) $p = 5$, c) $p = 10$;

Za zadatke u kojima se manipuliše brojevima ili algebarskim izrazima, najbolje je koristiti kompjuterske algebarske sisteme kakav je npr. Mathematica. Paskalov trougao ili trougao binomnih koeficijenata se takođe dobija dobro poznatom rekurzivnom procedurom koja se zasniva na identitetu

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.$$



Slika 15.2: Trougao Serpinjskog

Rezultat je prikazan u odgovarajućoj trougaonoj formi i generisan sa jednom linijom koda:

```
p = 5; Column[Table[If[Mod[Binomial[n,k],2]==1,Black,White],
  {n,0,2^p-1},{k,0,n}],Center]
```

i prikazan je na slici 15.3.

Drugi način da se u programu Mathematica prikaže binarni Paskalov trougao je da se iskoristi definicija 1D celularnih automata sa dve boje, kojih ima ukupno $2^8 = 256$. Paskalov trougao je kodiran brojem 90 i poznat je kao "rule 90." Komanda je:

```
ArrayPlot[CellularAutomaton[90, {{1}, 0}, 2^p - 1]]
```

a rezultat je prikazan na slici 15.4.

Zanimljivo je da je tvorca programa Mathematica, Steven Wolfram (Stephen Wolfram) inicijalno razvio sam program za potrebe analize celularnih automata. Rezultat zaista impresivnog proučavanja je publikovao u obimnoj knjizi *A New Kind of Science* objavljenoj 2002. godine.

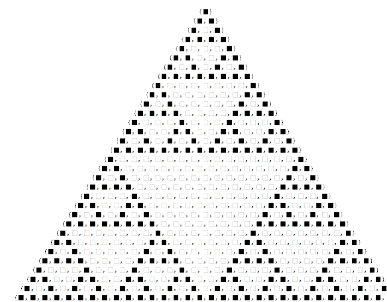
Primer 9: Haotična igra - šestougao

Generalisati prethodnu konstrukciju na sledeći način. Data su temena pravilnog šestougla. Polazeći od proizvoljne početne tačke, bacanjem kockice odrediti ka kojem se temenu prilazi, i nakon toga se u pravcu ka datom temenu pređe $2/3$ puta; zatim se ponovo baca kockica. Zašto je za konačni rezultat bitno da se prelazi tačno $2/3$ puta?

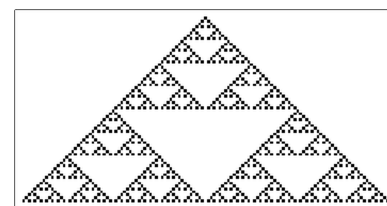
Dobijena figura je tzv. šestougao Serpinjskog. Sam kod je minorna varijacija prethodnog u kojoj se u petlji ponavljaju operacije

```
n=randomint(6)+1;X=(X+2*[A,B,C,D,E,F]-n)/3;
```

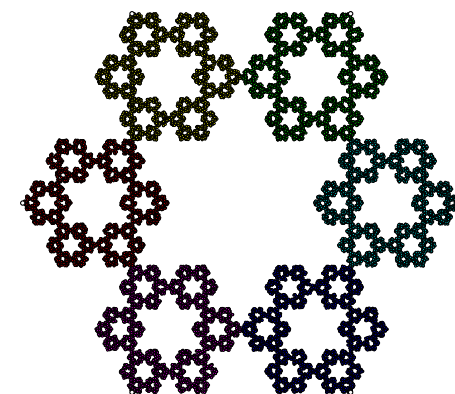
Sa otežinjavanjem 1 : 2 je postignuto da se pojedini šestouglovi u koje se slika početni šestougao međusobno ne preklapaju već da se samo dodiruju.



Slika 15.3: Binarni Paskalov trougao.



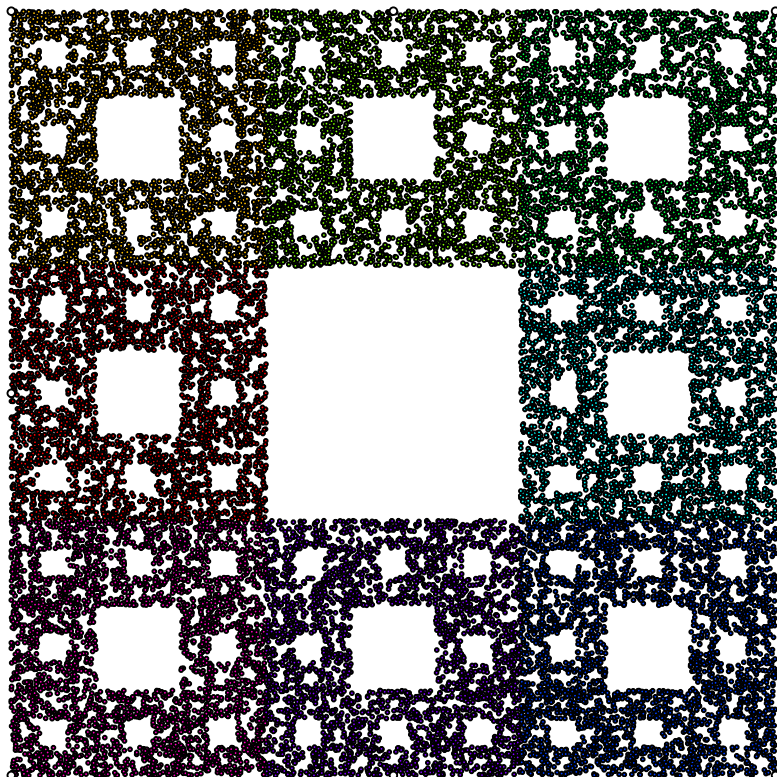
Slika 15.4: Celularni automat - rule 90.



Slika 15.5: Šestougao Serpinjskog.

Primer 10: "Haotična igra - tepih"

Generalizacija br.2: data su temena kvadrata zajedno sa sredinama stranica - ukupno 8 tačaka. Polazeći od proizvoljne početne tačke, generisanjem slučajnog celog broja iz $[1,8]$ odrediti ka kojem se temenu prilazi, i nakon toga se u pravcu ka datom temenu pređe $2/3$ puta; procedura se ponavlja itd.

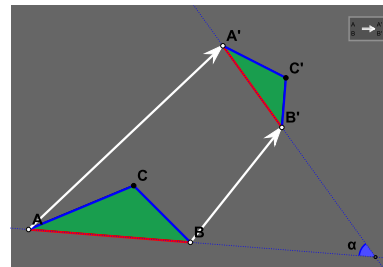


Slika 15.6: Tepih Serpinjskog.

Dobijena figura je tepih Serpinjskog.

Program Cinderella je program koji, pomoću transformacije sličnosti, transformacione grupe i iteriranih funkcionalnih sistema omogućava jednostavno kreiranje i analizu fraktalnih objekata. Fraktal kao "samosličan" objekat, u svojoj osnovi ima nekoliko transformacija sličnosti koje generišu (slobodnu) grupu. Dejstvom na pogodno odabrane početne objekte, pomoću alatke transformaciona grupa ("Modes/Special/Transformation Group"), lako se generiše početni niz objekata koji odgovara elementima grupe do unapred izabranog nivoa iteracije. Terminalne tačke fraktala je lako generisati pomoću alatke iterirani funkcioni sistemi ("Modes/Special/IFS").

Sličnost je definisana kao transformacija koja preslikava geometrijsku figuru u njom sličnu, npr. trougao u njemu sličan. Može se predstaviti kao kompozicija rotacije i homotetije. Analitički, u kompleksnoj ravni se predstavlja kao $z' = w \cdot z$, gde je $w \in \mathbb{C}$. Koeficijent homotetije je jednak $|w|$, ugao rotacije $\arg(w)$. Definicija sličnosti u programu Cinderella je jednostavna: dva para tačaka (jedan par je original, a drugi slika) jednoznačno definiše sličnost. Primer je prikazan na slici desno.



Slika 15.7: Definicija sličnosti u programu Cinderella: potrebno je odrediti samo preslikavanje jedne duži na drugu, odnosno njihovih graničnih tačaka, $A \mapsto A'$, $B \mapsto B'$.

Afina transformacija je definisana kao najopštija linearna transformacija u vektorskom prostoru i može se predstaviti kao kompozicija opšte linearne transformacije (reprezentovane proizvoljnom matricom) i translacijom. Definicija afinog preslikavanja je jednostavna: dve trojke tačaka jednoznačno je definišu. Za razliku od sličnosti, affine transformacije ne održavaju uglove, ali održavaju paralelnost pravih.

Ako definišemo nekoliko transformacija, npr. sličnosti, tada je moguće i generisati transformacionu grupu i njenu orbitu (sliku njenog dejstva na selektovani element). Orbita transformacione grupe podrazumeva kreiranje slike dejstva svih transformacija na selektovani element, zatim dejstva svih transformacija na dobijene slike elemenata itd. Kod fraktalnih objekata cilj je formirati *slobodnu transformacionu grupu*. Fraktal je objekat koji se sastoji od graničnih tačaka orbita slobodne grupe. Tipične geometrijske transformacije koje obrazuju slobodnu grupu sadrže skaliranje, tj. obrazovane su od sličnosti i afinih transformacija. Alat "IFS" tj. Iterirani Funkcionalni Sistem primenjuje nasumično sve generatore sa podjednakom verovatnoćom (ukoliko se to u skriptu ne promeni) i iscrtava orbitu tačke.

Trougao Serpinjskog je orbita slobodne transformacione grupe generisane sa tri affine transformacije.

Primer 11: Trougao Serpinjskog - po drugi put

Generisati trougao Serpinjskog pomoću transformacione grupe tj. IFS.

Rešenje je prikazano na slici desno. U gornjem desnom uglu slike su definisane tri affine transformacije kao i transformaciona grupa. Alat "transformaciona grupa" slučajnim izborom bira redosled i primenjuje generatore. Nije nužno da sve tri transformacije budu zastupljene sa jednakom verovatnoćom. To je moguće kontrolisati komandom

```
IFS0.prob0=cc;
```

```
IFS0.prob1=bb;
```

```
IFS0.prob2=aa;
```

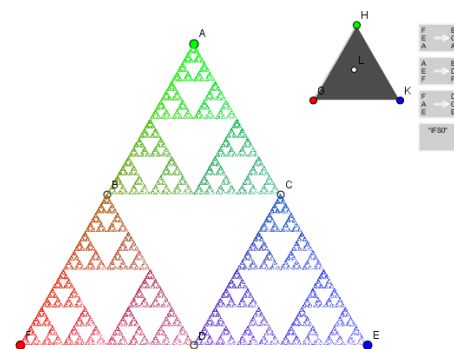
gde su sa aa, bb, cc označene baricentrične koordinate tačke L u ošćenom trouglu. Ako uvedemo oznaku $\omega = e^{\pi i/3}$, navedena preslikavanja možemo u kompleksnoj ravni definisati sa

$$f_1(z) = \frac{z}{2}, f_2(z) = \frac{z + \omega}{2}, f_3(z) = \frac{z + 1}{2}.$$

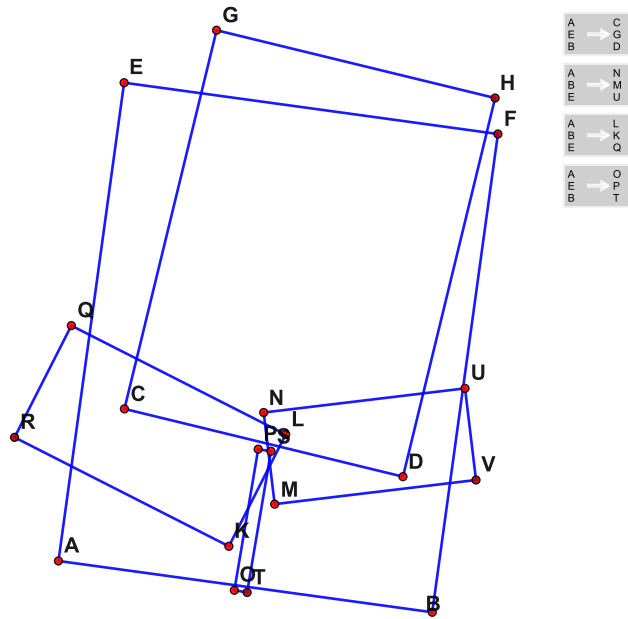
Primer 12: Bernslijeve paprat (Barnsley fern)

Veštom primenom IFS moguće je dobiti veoma realistične slike npr. paprati. Najpoznatiji primer je kreirao M. Barnsli. Sledi opis konstrukcije.

Grupa formirana pomoću svega 4 affine transformacije, koje preslikavaju osnovni pravougaonik $ABFE$ u 4 pomoćna: $CDHG, NMVU, LKRQ, PTSP$ kao što je prikazano na slici 15.9. Ako uklonimo oznake tačaka, obojimo tačke i duži u crno i obojimo i pozadinu u crno, a boje za slike pojedinih transformacija obojimo u zeleno ili oker, dobijamo konačni efekat prikazan na slici 15.10.



Slika 15.8: Tri afina preslikavanja preslikavaju veliki trougao u tri manja nastala povlačenjem srednje linije trougla.



Slika 15.9: Postavka IFS za Barnsley-ovu paprat.



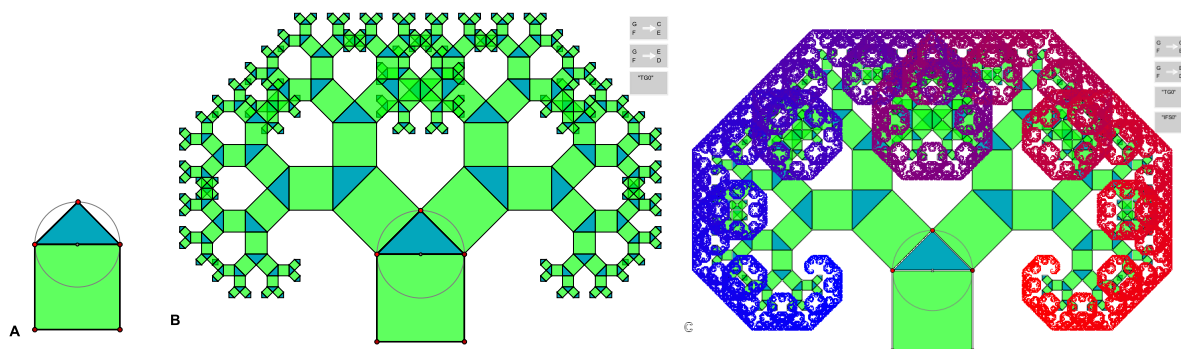
Slika 15.10: Barnsley-eva paprat.

Primenu transformacione grupe koja se sastoji od sličnosti ilustriramo na primeru tzv. Pitagorinog stabla.

Primer 13: Pitagorino stablo

Konstruisati tzv. Pitagorino stablo dejstvom transformacione grupe sa dva generatora: prvi preslikava stranicu kvadrata nad hipotenuzom u jednu, a drugi u drugu katetu.

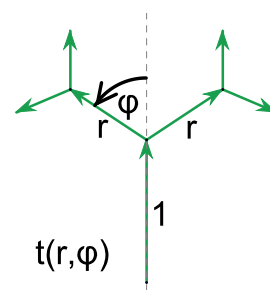
Najpre konstruišemo pravougli trougao i kvadrat nad njegovom hipotenuzom (sl. A). Zatim kreiramo transformacije sličnosti koje preslikavaju stranicu kvadrata naspram hipotenuze, u katete trougla. Pomoću navedene dve sličnosti generišemo transformacionu grupu. Dejstvo transformacione grupe na kvadrat i trougao je prikazano na slici B. Na njoj je već uočljiva fraktalna struktura slike. Fraktal koji odgovara Pitagorinom drvetu se sastoji od fiksnih tačaka transformacione grupe. Njih je, takođe jednostavno generisati pomoću iteriranih funkcionijskih sistema. Odgovarajuća slika je prikazana pod C. Kako se menja slika promenom položaja pravog na kružnici?



Slika 15.11: Faze generisanja Pitagorinog drveta - fraktala.

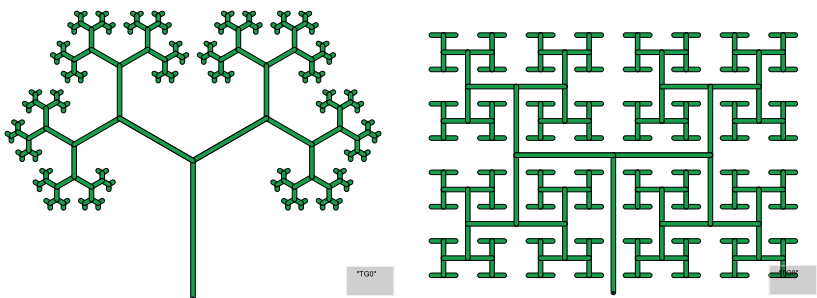
Primer 14: Simetrično binarno stablo

Simetrično binarno stablo, sa oznakom $t(r, \phi)$, je definisano na sledeći način. Neka je dato inicijalno stablo - vertikalna duž jedinične dužine i dve transformacije sličnosti (zadate u kompleksnoj ravni), $L(z) = re^{i\phi} + 1$ i $R(z) = re^{-i\phi} + 1$, pri čemu je $r \in \mathbb{R}$, $r < 1$. Stablo se dobija dejstvom svih mogućih kompozicija operacija R i L na inicijalno stablo. Prikazati početne nizove za sledeće primere: $t((1 + \sqrt{5})/2 - 1, \pi/3)$, $t(1/\sqrt{2}, \pi/2)$, $t(1/\sqrt{2}, 3\pi/4)$, $t((1 + \sqrt{5})/2 - 1, 4\pi/5)$.

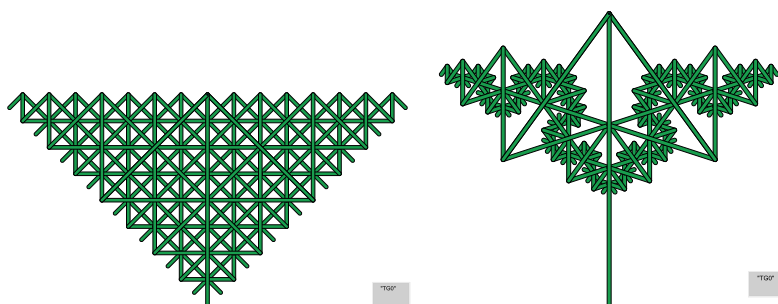


Slika 15.12: Definicija simetričnog binarnog stabla.

Rešenje je prikazano na slikama 15.13 i 15.14.



Slika 15.13: Binarno simetrično stablo: $t((1 + \sqrt{5})/2 - 1, \pi/3)$ levo i $t(1/\sqrt{2}, \pi/2)$ desno.



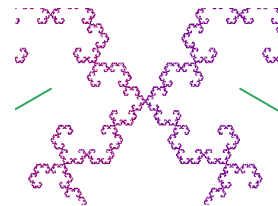
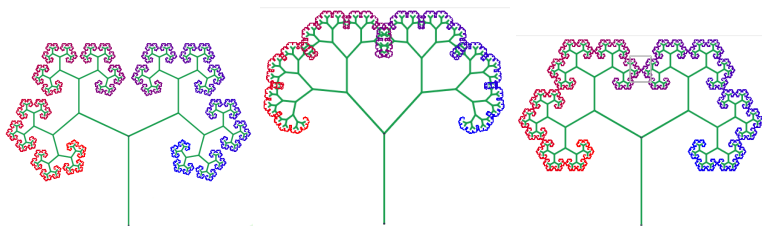
Slika 15.14: Binarno simetrično stablo: $t(1/\sqrt{2}, 3\pi/4)$ levo i $t((1 + \sqrt{5})/2 - 1, 4\pi/5)$ desno.

Ako posmatramo faze u konstrukciji stabla, zapažamo da za neke vrednosti parametara, nakon nekoliko (konačno mnogo!) iteracionih koraka, dolazi do preklapanja nekih grana i takav tip drveta nazivamo "samopreklapajući" (self-overlapping). Ona kod kojih ne dolazi do preklapanja ni samog fraktalnog skupa (nakon beskonačno iteracija) nazivamo "samoizbegavajućim" (self-avoiding). Granični slučaj predstavljaju parametri za koje se grane drveta nakon konačnog broja iteracija ne dodiruju, a fraktalni skupovi se dodiruju i njih nazivamo "samokontrahujućim" (self-contracting). Za svaku vrednost parametra ϕ postoji vrednost parametra $r = r_{sc}$ za koju fraktalni skup postaje samokontrahujuć. Skup tačaka (r_{sc}, ϕ) nazivamo Mandelbrotovim skupom za binarna simetrična drveta.

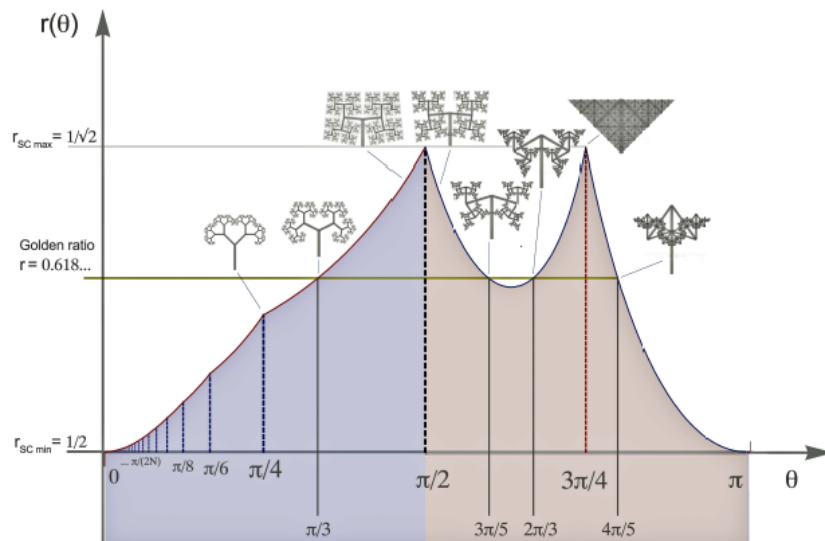
Primer 15: Simetrično binarno stablo - fraktalni skup

Upotrebom alata IFC i variranjem vrednosti parametra r pokazati da su sva četiri stabla iz prethodnog primera samokontrahujuća.

Varijacija parametra r na prvom primeru je prikazana na slici ispod. Prikazano je stablo $t(r, \pi/3)$ za $r < (1 + \sqrt{5})/2 - 1$ (levo) $r > (1 + \sqrt{5})/2 - 1$ (sredina) i $r = (1 + \sqrt{5})/2 - 1$ (desno). Takođe je prikazan i Mandelbrotov skup preuzet iz rada *Symmetry: Culture and Science*, Vol. 21, Nos. 1-4, 333.



Slika 15.15: Detalj stabla sa prethodne slike za $r = (1 + \sqrt{5})/2 - 1$ u tački dodira na y -osi.



Slika 15.16: Mandelbrotov skup.

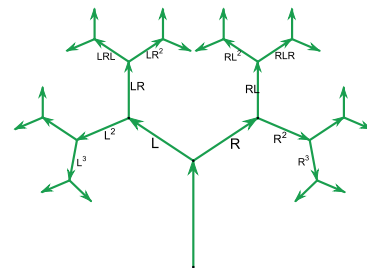
Svakom povezanom nizu grana drveta možemo pridružiti niz slova, tj. reč npr. $LRRLRLLLRLR \dots$, ili u sažetijoj formi, $LR^2LR^2L^3RLR$, koja odgovara nizu transformacija primenjenom na inicijalno stablo. Izborom reči lako je je mapirati odgovarajući put prostim delovanjem odgovarajućim elementom.

Primer 16

Za stablo $t((1 + \sqrt{5})/2 - 1, \pi/3)$ i $t(1/\sqrt{2}, \pi/2)$ skicirati put koji odgovara reči $(LR)^n$ i $(RL)^n$, gde je n veliki broj.

Primer 17

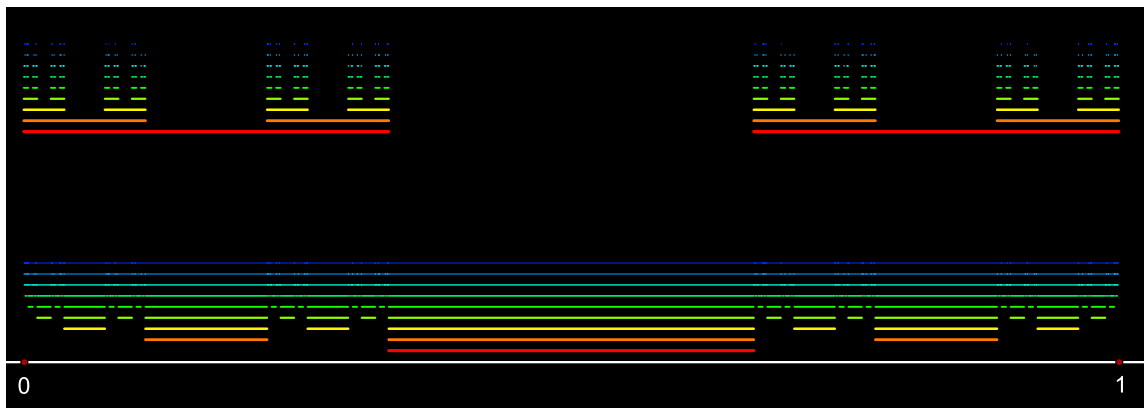
Za stablo $t((1 + \sqrt{5})/2 - 1, \pi/3)$ analitički odrediti kodove elementa koji se dodiruju na y -osi.



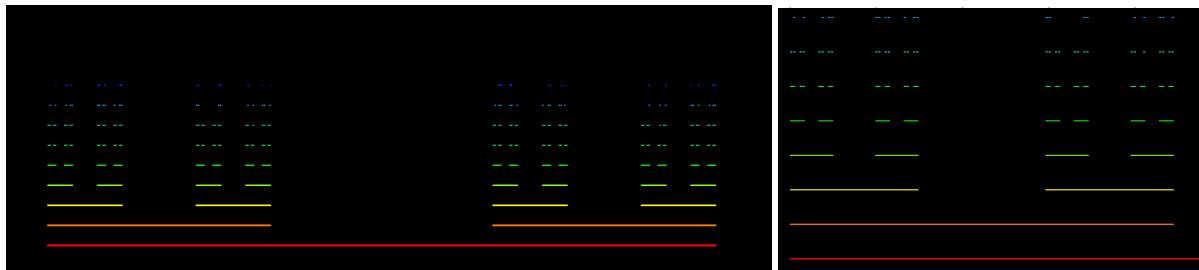
Verovatno najstariji primer fraktalnog objekta je Kantorov skup. On se dobija rekurzivnom primenom komplementa karakteristične funkcije srednje trećine intervala. Lako je simbolički reprezentovati: ako tačke na intervalu $[0, 1]$ predstavimo u sistemu sa osnovom 2, tada u n -tom iteracionom koraku funkcija obriše one tačke koje na n -tom decimalnom mestu imaju 1. Granični skup sadrži samo tačke koje u decimalnom zapisu nemaju nijednu jedinicu. Kantorov skup zajedno sa njegovim komplementom i nekoliko zumiranih slika je prikazan ispod.

Primer 18

Da li je Kantorov skup ekvipotentatan kontinuumu? Da li on sadrži dve povezane tačke?



Slika 15.17: Kantorov skup i njegov komplement (dole), nakon 10 iteracija



Slika 15.18: Zumirana prva trećina Kantorovog skupa

Odgovarajući skript je:

```
g(x):=mod(ceil(x),2);
repeat(9,n,
plot(0.01*(n)+0.2*product(1..n,i,g(3^i*x)),start->-0.000000001,
stop->1.000000001,size->(3-n*0.25),color->hue(n*0.08-0.08));
);
```

16 Geometrijska optika i bilijari

Cilj ove glave da ukaže, nakon prethodnih, u kojima se čitalac upoznao sa osnovnim mogućnostima programa Cinderella, na samu bit dinamičke geometrije i njenu vezu sa fizikom. Ona daje centralnu sponu između mehanike i optike, sa jedne strane, geometrije i analize, sa druge i objektno orjentisanog programiranja sa treće. Njihova isprepletanost ukazuje na zajedničko poreklo ideja koje stoje iza ovih, iz istorijskih razloga, podeljenih nauka. Matematički gledano, optika i mehanika su ista nauka, dok je analiza, geometrijski interpretirana, njihov matematički opis. Sa stanovišta algebre, organizovani skupovi izraza pridruženi konkretnim objektima, daju vizuelni primer objektno orjentisanog programiranja. Programiranje omogućava konstrukciju alata, koje zatim koristimo u proučavanju specifičnih dinamičko-optičkih problema.

16.1 Programiranje

Prvi skup zadataka spada u programiranje. Najpre treba definisati zrak, osnovne optičke objekte i interakciju zraka sa tim objektima.

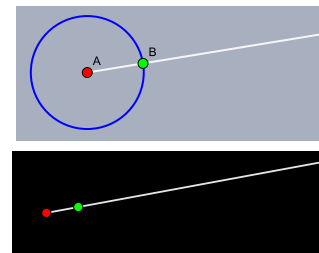
Primer 1: Zrak

Konstruisati zrak u programu Cinderella.

Poluprava na postoji kao definisani objekat u programu Cinderella, a razlog je jednostavan - ona nije dobro definisan pojam projektivne geometrije. Ono što možemo da definišemo je aproksimacija poluprave konačne dužine. To je jednostavno uraditi: nacrtamo kružnicu slobodnog poluprečnika oko tačke A i na njoj tačku B . Aproksimativna poluprava, je definisana početnom tačkom A , vektorom pravca $\vec{s} = \frac{\vec{AB}}{|\vec{AB}|}$ i dužinom q . Sam zrak, kao modelni objekat je definisan sa A i \vec{s} , sa dodatnom mogućnošću da se uvede i treći parametar koji definiše providnost (opacitet).

Skript konstrukcije je

```
q=20;aa=0.9;
p=A.xy;
s=B-A;
s=s/|s|;
draw(p,p+q*s,color->(1,1,1),size->2,alpha->aa);
```



Slika 16.1: Gore - analiza, dole - finalna realizacija objekta "zrak"

U slučaju da se koristi default jedinice (dužine) programa, realistična vrednost za q je reda 100.

Dinamika je opisana oblikom (putanjom) zraka pri zadatoj konfiguraciji objekata. Da bi odredili oblik zraka moramo najpre procesirati njegovu putanju pomoću procedure (operatora) `pBeam`. To podrazumeva da

- polazeći iz tačke (x_0) sa inicijalnim zrakom (z_0) , odredimo prvi presek zraka sa skupom zadatih objekata (x_1) ;
- iz tačke preseka (x_1) konstruišemo reflektovani zrak (z_1) ;
- odredimo presek (z_1) sa skupom objekata, tj. generišemo (x_2) , itd.

Dakle, procedura `pBeam` se mora pozivati na 2 subrutine, `intersect`, koja određuje presek x_n zraka z_{n-1} sa skupom objekata, i `reflect`, koja generiše zrak z_n iz tačke preseka x_n . Čitava procedura se sprovodi sa brojem koraka koji je unapred definisan, npr. 500. Delovanje subrutina je posebno definisano za različite tipove objekata.

Primer 2: Kružno ogledalo

Realizovati proceduru procesiranja zraka na sistemu kružnih ogledala.

Najpre programiramo subrutinu `intersect`, kojom određujemo tačke preseka zraka i objekta: ovde je objekat kružnica. Objekat - kružno ogledalo definišemo kao trojku ("circ", (x_0, y_0) , r): ime (string "circ"), koordinate centra (x_0, y_0) i poluprečnik r . Primenimo stav o potenciji tačke (A) u odnosu na kružnicu. U oznakama sa slike 16.2, potencija tačke A je jednaka

$$|A, K_1| \cdot |A, K_2| = (|\vec{q}| - r)(|\vec{q}| + r).$$

Za rastojanja $|A, K_1|$, $|A, K_2|$ važi:

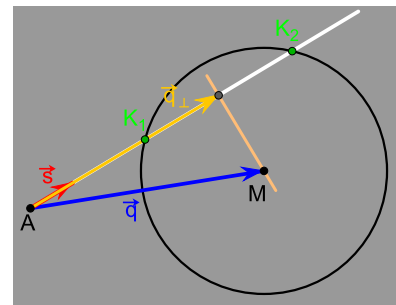
$$|A, K_1| = |\vec{q}_\perp| - x, |A, K_2| = |\vec{q}_\perp| + x, |\vec{q}_\perp| = \vec{s} \cdot \vec{q},$$

pa ako $\overrightarrow{AK_1}$ izrazimo kao $\overrightarrow{AK_1} = a \vec{s}$ i $\overrightarrow{AK_2} = b \vec{s}$, tada su a i b rešenja kvadratne jednačine

$$a^2 - 2a\vec{s} \cdot \vec{q} + |\vec{q}|^2 - r^2 = 0.$$

Odgovarajući skript subrutine je

```
intersect(p, s, o) :=(
  ll=[];
  if(o_1=="circ", (
    m=o_2;
    r=o_3;
    q=p-m;
```



Slika 16.2: Primena potencije tačke u odnosu na kružnicu.

```

sq=s*q;
qr=q*q-r^2;

a=-sq+sqrt(sq^2-qr);
b=-sq-sqrt(sq^2-qr);

if(|im(a)|<0.00000001 & re(a)>0.00000001,ll=ll+[[o,re(a)]];
if(|im(b)|<0.00000001 & re(b)>0.00000001,ll=ll+[[o,re(b)]]);
);
ll;
);
);

```

Ulaz su parametri upadnog zraka, koordinate izvora p i vektor pravca s , objekat o , okarakterisan nazivom, i parametrima (kod kružnog ogledala to su koordinata centra i poluprečnik). Izlaz je objekat o i parametri a_i , tako da su tačke preseka zraka i objekta dati izrazom $\vec{x}_i = \vec{p} + a_i \vec{s}$.

Druga subrutina je `reflect` i ona deluje tako što u tački preseka x zraka i ogledala generiše novi zrak, sa početkom u x i vektorom pravca reflektovanim u odnosu tangentu u tački x . Refleksiju u tački x je moguće interpretirati i kao refleksiju u odnosu na pravu koja spaja centar kružnice sa tačkom x praćena obrtanjem smeru vektora. Skript subrutine je

```

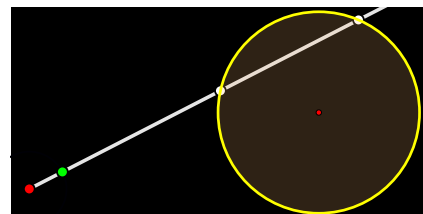
reflect(p,s,o,a):=(
  x=p+a*s;
  ss=s;
  if(o_1=="circ", (
    sx=x-o_2;
    sx=sx/|sx|;
    ss=s-2*(s*sx)*sx;
  ));
  [x,ss];
);

```

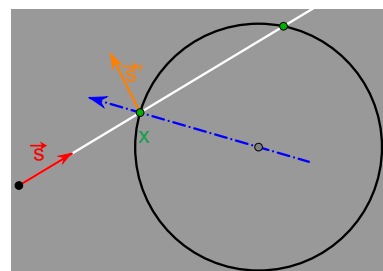
Ulaz su parametri upadnog zraka, objekat o , i parametar a , dobijen kao izlaz i subrutine `intersect`, dok je izlaz novi zrak definisan sopstvenim parametrima p_1 i s_1 .

Na kraju treba formirati putanju zraka pomoću `while` petlje. Na ulazu u petlju imamo parametre upadnog zraka, zatim, u svakom koraku se određuju tačke preseka aktuelnog zraka sa sistemom, nalazi tačka preseka najbliža izvoru, nju proglašava za novi izvor i reflektovanjem određuje novi pravac zraka itd. Skript glasi:

```
stop=false;
```



Slika 16.3: Primena subrutine za određivanje preseka zraka i kružnice.



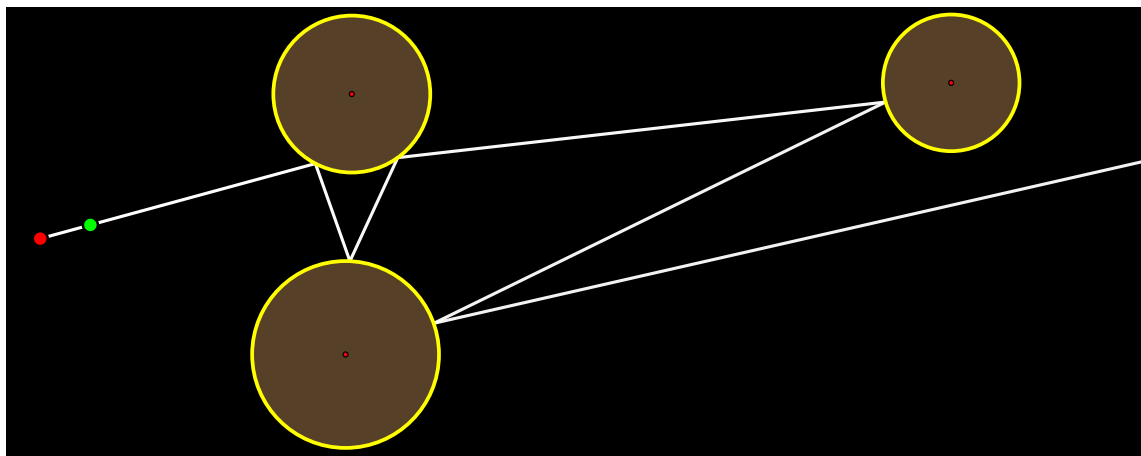
Slika 16.4: Reflektovanje zraka u odnosu na kružno ogledalo.

```

count=0;
n=2; // broj refleksija koje racuna
alpha=1;
dim=0.99; //stepen zatamnjenja pri refleksiji
while(count<n & !stop,
    count=count+1;
    l=intersect(p,s,o1)++intersect(p,s,o2)++intersect(p,s,o3); //formira skup tacaka preseka
    l=sort(l,#_2); //odredjuje prvu (najblizu)
    if(length(l)>0, //proverava ima li uopste preseka sa objektima
        (
            hit=l_1;
            erg=reflect(p,s,hit_1,hit_2); //reflektuje zrak
            draw(p,erg_1,color->(1,1,1),alpha->alpha); //crta zrak od izvora to tacke refleksije
            p=re(erg_1); // definise
            s=re(erg_2); //      novi      zrak
        ),
        (draw(p,p+100*s,color->(1,1,1),alpha->alpha); //crta zrak nakon poslednje refleksije
        stop=true;
        );
    );
    alpha=alpha*dim; // daje efekat slabljenja zraka nakon svake refleksije
);

```

Krajnji efekat je prikazan na slici 16.5.



Slika 16.5: Višestruka refleksija od tri kružna ogledala.

Primer 3: Pravolinijsko ogledalo

Realizovati proceduru procesiranja zraka na sistemu ravnih ogledala.

Pravolinijsko ogledalo je geometrijska duž i definisana je sa dve tačke: početnom i krajnjom. Prema tome, definišemo objekat kao trojku ("mirror", (x1, y1), (x2, y2)): ime (string "mirror"), koordinate jednog kraja (x1, y1) i drugog kraja (x2, y2).

Razradimo subrutinu intersect, kojom određujemo tačke preseka zraka i objekta: za objekat ravno ogledalo. Primenimo stav o kolinearnosti tri tačke iz projektivne geometrije: tačke A, B i C su kolinearne, akko za njihove homogene koordinate važi:

$$\det(a, b, c) \equiv \begin{vmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \\ c_1 & c_2 & 1 \end{vmatrix} = 0.$$

Primenimo teoremu na tačke a, b i x sa slike 16.6.

Kako je $x = p + \lambda s$, iz teoreme sledi

$$\begin{vmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \\ x_1 & x_2 & 1 \end{vmatrix} = \begin{vmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \\ p_1 + \lambda s_1 & p_2 + \lambda s_2 & 1 \end{vmatrix} = \begin{vmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \\ p_1 & p_2 & 1 \end{vmatrix} + \lambda \begin{vmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \\ s_1 & s_2 & 0 \end{vmatrix} = 0,$$

tj. parametar λ je jednak

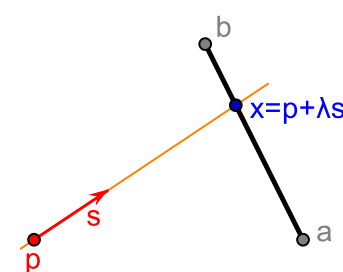
$$\lambda = \frac{\begin{vmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \\ p_1 & p_2 & 1 \end{vmatrix}}{\begin{vmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \\ s_1 & s_2 & 0 \end{vmatrix}}.$$

Međutim, na ovaj način smo odredili presek pravca zraka sa pravcem kojem ogledalo pripada. Potrebno je proveriti da li je tačka x između tačaka a i b. To je moguće uraditi proverom znaka površina prikazanih na slici 16.7.

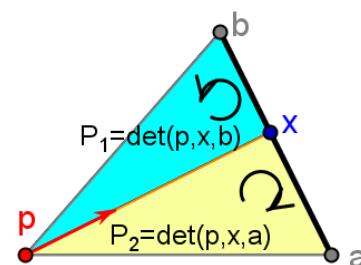
Konačno, skript glasi:

```
intersect(p, s, o) := (
  ll=[];
  if(o_1=="mirror", (
    aa=o_2++[1];
    bb=o_3++[1];
    pp=p++[1];
    ss=s++[0];
```

Zadatak: pokazati da je teorema ekvivalentna uslovu da je površina trougla $\triangle ABC$ jednaka nuli.



Slika 16.6: Primena uslova kolinearnosti



Slika 16.7: Provera položaja tačke x u odnosu na a i b.

```

lam=-det(aa,bb,pp)/det(aa,bb,ss);

x=p+lam*s;
dets=area(pp,x++[1],aa)*area(pp,x++[1],bb);

if(dets<0.0000000&a>0.00000001,ll=ll+[[o,re(lam)]]);
);
);
ll;
);

```

Kao i u prethodnom slučaju - zadatku, ulaz u intersect su parametri zraka i objekta, a izlaz su parametri objekta i parametar λ , tako da je tačka preseka $x = p + \lambda s$.

Refleksiju realizujemo tako što odredimo normalu na pravac ogledala i zatim invertovanu refleksiju zraka u odnosu na nju. To je urađeno u sledećem skriptu:

```

reflect(p,s,o,a):=(
  pp=p+a*s;
  ss=s;
  if(o_1=="mirror", (
    sp=perp(o_3-o_2);
    sp=sp/|sp|;
    ss=s-2*(s*sp)*sp;
  ));
);

```

Trivijalnom modifikacijom prethodne subrutine možemo realizovati još jedan važan optički element: pravolinijski apsorber. To je objekat koji potpuno apsorbuje zrake koji padaju na njega, tako da od njega nema refleksije. To se jednostavno uradi tako što se iza tačke preseka sa apsorberom više ne izscrtava putanja zraka, tj. promenljivoj stop, koja kontroliše iscrtavanje zraka u petlji dodeli vrednost true. Sam skript glasi:

```

if(o_1=="absorber", (
  stop=true;
));
);
[pp,ss];
);

```

U ovoj fazi možemo omogućiti postojanje proizvoljnog broja objekata, ali i proizvoljnog broja zraka. Zbog toga je na samom početku neophodno definisati i zrake kao

```

p1=A1.xy;
s1=B1-A1;
s1=s1/|s1|;

beam1=[p1,s1,[1,1,0]]; // na trecem mestu definisemo boju zraka
//itd

```

Finalni program za procesiranje pojedinih zraka pBeam glasi:

```

pBeam(beam) :=(
  p=beam_1;
  s=beam_2;
  color=beam_3;
  stop=false;
  count=0;
  n=100; // broj refleksija
  alpha=1;
  dim=0.99;
  while(count<n & !stop,
    count=count+1;
    l=[];
    forall(objects,o,l=l++intersect(p,s,o));

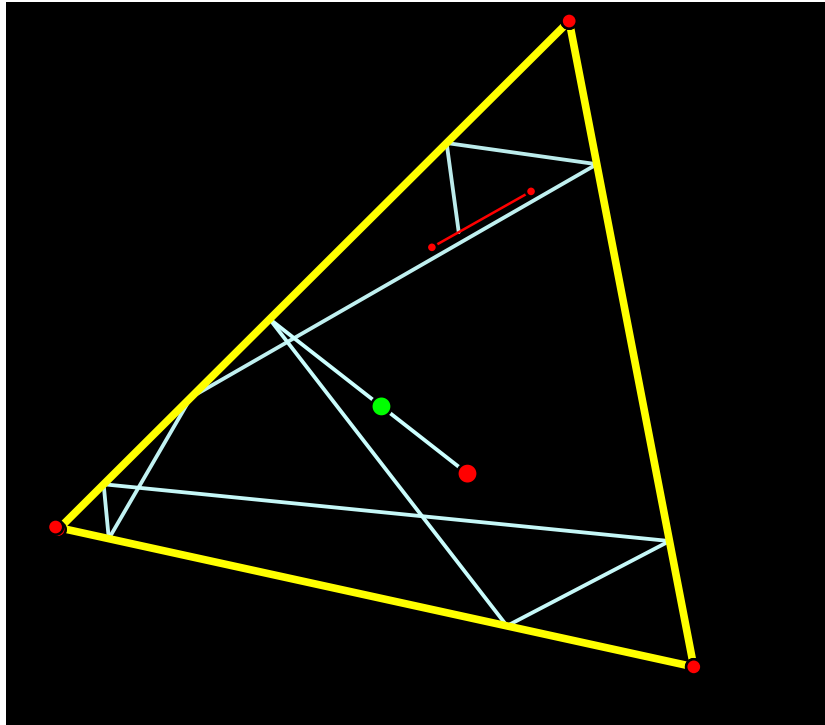
    l=sort(l,#_2);

    if(length(l)>0,(
      hit=l_1;
      erg=reflect(p,s,hit_1,hit_2);
      draw(p,erg_1,color->color,alpha->alpha);
      // draw(erg_1,erg_1+erg_2*.1,color->(1,0,0),size->2);
      p=re(erg_1);
      s=re(erg_2);
    ),(
      draw(p,p+100*s,color->color,alpha->alpha);

      stop=true;
    )
  );
  alpha=alpha*dim;
);

```

Primer sa tri ogledala, jednim zrakom i jednim apsorberom je prikazan na slici 16.8.



Slika 16.8: Višestruka refleksija od tri kružna ogledala.

Primer 4: Eliptično ogledalo

Realizovati proceduru procesiranja zraka na sistemu ogledala u obliku konusnih preseka.

U slučaju refleksije na konusnim presecima je najopštiji, jer su kružnice i prave linije specijalni slučajevi konusnih preseka. Po prirodi stvari, konusni preseki spadaju u projektivnu geometriju jer su definisani kao jedinstveni objekat - *konika*. Konike su opisane u homogenim koordinatama sa matricom konike - realnom, simetričnom, 3×3 matricom. Prvi problem je određivanje preseka zraka i konike. Zrak sa početkom u tački p i kroz tačku b je opisan izrazom (u homogenim koordinatama) $p + \lambda b$, $\lambda \geq 0$. Sa druge strane, presek prave l i konike sa matricom A je definisan jednakošću $l^T A l = 0$. Dakle, presek zraka sa konikom je dat kao nenegativno (po λ) rešenje jednačine

$$(p + \lambda b)^T A (p + \lambda b) = 0.$$

To je regulisano skriptom, koji na primeru elipse, glasi

```
m(p1,p2,p3,s1,s2,s3,a,b,c,d,e,f,sg) :=
(-2*a*p1*s1-2*b*p2*s1-2*d*p3*s1-2*b*p1*s2-2*c*p2*s2-2*e*p3*s2-2*d*p1*s3-2*e*p2*s3-2*f*p3*s3+sg*
sqrt((2*a*p1*s1+2*b*p2*s1+2*d*p3*s1+2*b*p1*s2+2*c*p2*s2+2*e*p3*s2+2*d*p1*s3+2*e*p2*s3+2*f*p3*s3)^2-
```

```

4*(a*p1^2+2*b*p1*p2+c*p2^2+2*d*p1*p3
+2*e*p2*p3+f*p3^2)*(a*s1^2+2*b*s1*s2+c*s2^2+2*d*s1*s3+2*e*s2*s3+f*s3^2))
/(2*(a*s1^2+2*b*s1*s2+c*s2^2+2*d*s1*s3+2*e*s2*s3+f*s3^2));
//-----
p=D.xy;
po=E.xy;
s=(po-p)/|po-p|;
mat=C0.matrix;
f1=A.xy;f2=B.xy;
//-----
o1=["conicE",[f1,f2]];
beam1=[p,s,(.8,1,1)];

intersect(p,s,o):=(
  ll=[];
  if(o_1=="conicE", (
    a=mat_1_1;b=mat_1_2;d=mat_1_3;
    c=mat_2_2;e=mat_2_3;f=mat_3_3;

    //homogenizacija
    p1=(p.homog)_1;p2=(p.homog)_2;p3=(p.homog)_3;
    s1=((p+s).homog)_1;s2=((p+s).homog)_2;s3=((p+s).homog)_3;

    mm1=m(p1,p2,p3,s1,s2,s3,a,b,c,d,e,f,1);
    mm2=m(p1,p2,p3,s1,s2,s3,a,b,c,d,e,f,-1);

    //tacka preseka i normalizacija
    pt1=(p.homog)+mm1*((p+s).homog);pt1=pt1/(pt1_3);
    pt2=(p.homog)+mm2*((p+s).homog);pt2=pt2/(pt2_3);

    //test pozitivnosti
    if((|im(mm1)|<0.0000001)&(( [pt1_1,pt1_2]-p)*s>0.00000001),ll=ll+[[o,s*([pt1_1,pt1_2]-p)]]);
    if((|im(mm2)|<0.0000001)&(( [pt2_1,pt2_2]-p)*s>0.00000001),ll=ll+[[o,s*([pt2_1,pt2_2]-p)]]);
  );
);
ll;
);

```

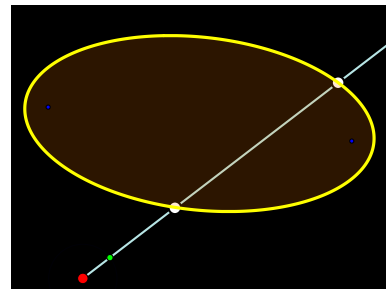
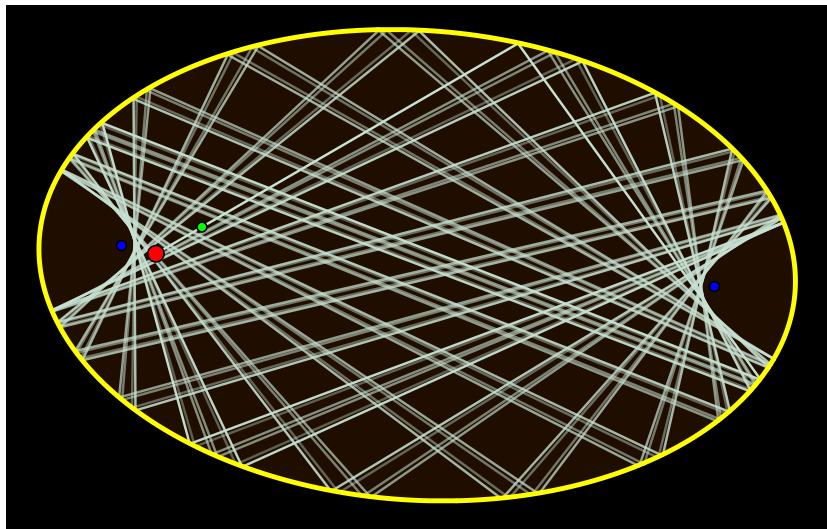
Rešenje jednačine uslova za presek je dato funkcijom m . Objekat je definisan matricom i , u slučaju elipse i hiperbole, žižama, a u slučaju parabole, žižom i direktrinom. Ulaz u proceduru su zraci i objekat, a kako su zraci definisani u Euklidskim koordinatama, u prvom delu se određuju njihove homogene koordinate (homogenizacija). Zatim se pomoću funkcije m određuju tačke preseka (u homogenim koordinatama), pa se njihova treća koordinata normalizuje, što je

priprema za dehomogenizaciju. Na kraju se proverava pozitivnost parametra. Rezultat je prikazan na slici 16.9.

Problem refleksije je jednostavno rešiti ako znamo osnovno optičko svojstvo konusnih preseka - zrak iz jedne žiže se se reflektuje od konike ka drugoj žiži. To znači da normala na koniku zaklapa jednake uglove u odnosu na duži koje spajaju tekuću tačku sa žižama. Zrak treba samo reflektovati (sa promenom smera) u odnosu na datu normalu. Evo skripta za određivanje refleksije od konike:

```
reflect(p,s,o,a):=(
  pp=p+a*s;
  ss=s;
  if(o_1=="conicE", (
    fv1=pp-o_2_1;fv1=fv1/|fv1|;
    fv2=pp-o_2_2;fv2=fv2/|fv2|;
    sp=fv1+fv2; sp=sp/|sp|;
    ss=s-2*(s*sp)*sp;
  ));
  [pp,ss];
);
```

Konačni efekat refleksije zraka unutar eliptičnog ogledala je prikazan na slici 16.10.



Slika 16.9: Presek zraka sa elipsom.

Slika 16.10: Višestruka refleksija unutar eliptičnog ogledala. Plave tačke su žiže elipse.

Primer 5: Tipovi zraka

Generalisati objekat zrak, sa dva nova tipa: paralelni i divergentni snop zraka.

Paralelni snop konstruišemo transliranjem, a divergentni rotiranjem inicijalnog zraka. U prvom slučaju transliramo tačku izvora ortogonalno na pravac zraka, a u drugom rotiramo pravac zraka oko tačke izvora. Preciznije,

```
beamShift(beam, shift) := [beam_1 + (perp(beam_2)) * shift, beam_2, beam_3];
beamRot(beam, w) := [beam_1, gauss(complex(beam_2) * exp(i * w * r)), beam_3];
```

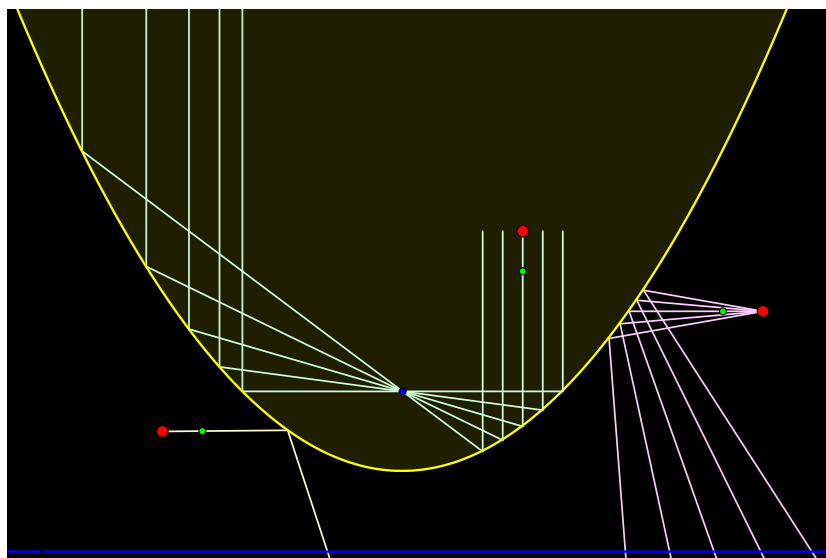
Nakon definicije funkcija, formiramo objekat zraci:

```
beams = apply(-2..2, beamShift(beam1, #*.5)) ++ apply(-2..2, beamRot(beam2, #*5)) +++ [beam3];
```

tj. od prvog formiramo divergentni, od drugog paralelni, a treći zrak ostaje pojedinačni. Naravno, na kraju treba procesirati sve zrake čije putanje želimo da iscrtamo.

```
forall(beams, b, processBeam(b));
```

Refleksije različitih tipova zraka na paraboličnom ogledalu su prikazane na slici 16.11.

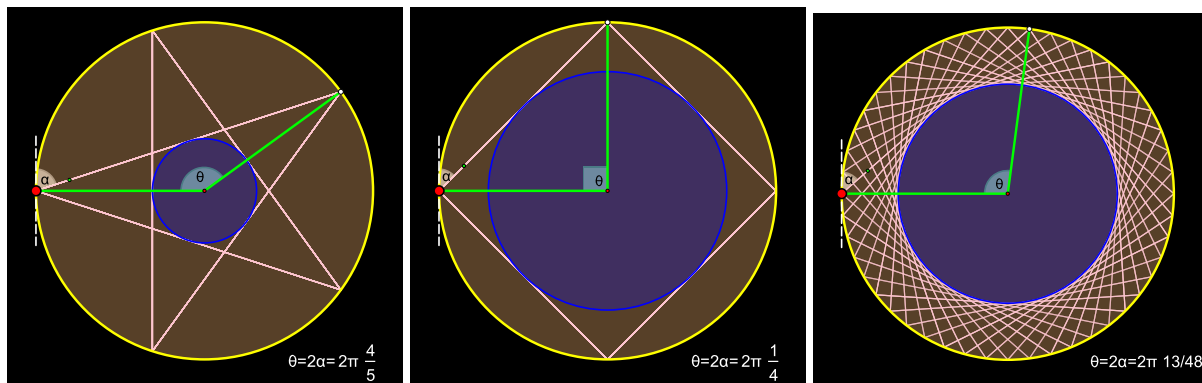


Slika 16.11: Višestruke refleksije na paraboličnom ogledalu.

Nakon konstrukcije osnovnih alata, prelazimo na izučavanje dinamike, konkretno, bilijara na krugu i kvadratu.

16.2 Dinamika kružnog i kvadratnog bilijara

Bilijar na krugu je u potpunosti definisan početnim upadnim uglom α , koji ostaje isti nakon svake refleksije. Pritom se nova tačka sudara dobija rotacijom kruga za $\theta = 2\alpha$. Ukoliko je $\theta = 2\pi p/q$, orbita je q -periodična i nakon q sudara zrak napravi p okreta oko kruga.



Slika 16.12: Primeri racionalnih trajektorija bilijara na krugu.

Primer 1: Gustina π iracionalne orbite (Jakobi)

Pokazati da ako je θ π -iracionalno, tada je T_θ orbita svake tačke gusta.

Primer 2: Orbite segmenata

Segmenti bilijarne trajektorije koji su pod uglom α u odnosu na kružnicu su tangentni u odnosu na koncentrični krug radijusa $\cos \alpha$. Pokazati da u slučaju π -iracionalnog α , segmenti bilijarne trajektorije gusto ispunjavaju prsten između krugova.

Postoji jači iskaz od gustine π -iracionalnih orbite - one poseduju dodatnu pravilnost koja je generalizacija ponašanja orbite pri $\theta = 2\pi p/q$. Osobina se naziva *ekvidistribucionost*.

Definicija 3: Ekvidistributivnost

Neka je dat luk I i neka je $k(n)$ broj članova niza x_0, x_1, \dots, x_{n-1} koji se nalaze u I . Tada niz nazivamo ekvidistribuiranim na krugu $\mathbb{R}/2\pi\mathbb{Z}$ ako važi

$$\lim_{n \rightarrow \infty} \frac{k(n)}{n} = \frac{|I|}{2\pi}.$$

Primer 3: Vejlova ekvidistribuciona teorema

Pokazati da važi sledeći stav: ako je θ π -iracionalan, tada je niz $x_n = x + n\theta \bmod 2\pi$ ekvidistribuiran na krugu.

Iako na prvi pogled potpuno različit od kruga, bilijar na kvadratu su blisko povezani sa bilijarom na krugu.

Primer 4: Razmotavanje orbite

Pokazati da je proizvoljna orbita bilijara na jediničnom kvadratu ekvivalentan pravolinijskoj putanji na "razmotanoj" rešetci $2\mathbb{Z} + 2\mathbb{Z}$. (Susedni kvadrati imaju suprotnu orijentaciju). Ili, bilijar na kvadratu je ekvivalentan geodezijskoj putanji na ravnom torusu.

Primer 5: Ekvivalencija sa kružnim bilijarom

Razmotrimo trajektoriju sa uglom α i početnom tačkom x na donjoj stranici 2×2 kvadrata. Pokazati da ona preseca gornju stranicu u tački $x + 2 \cot \alpha \bmod 2$. Reskaliranjem sa $1/2$ dobijamo rotacija $x \mapsto x + \cot \alpha \bmod 1$ na krugu $S^1 = \mathbb{R}/\mathbb{Z}$. Na sličan način je moguće svesti bilijar na jediničnoj kocki \mathbb{R}^n na rotaciju na torusu T^{n-1} .

Trajektoriju bilijara na kvadratu je moguće binarno kodirati: simbol 0 odgovara refleksiji od horizontalne i 1 , od vertikalne stranice. Odgovarajući niz nazivamo *presecajući niz* trajektorije.

Definicija 4: Kvaziperiodičnost

Niz nazivamo kvaziperiodičnim ako se svaki početni segment pojavljuje beskonačno puta.

Primer 6: Kvaziperiodične trajektorije

Pokazati da je trajektorije sa iracionalnim uglom kvaziperiodična.

Primer 7: Fibonačijev niz

Posmatrajmo trajektoriju sa nagibom "zlatnog količnika" $\varphi = (1 + \sqrt{5})/2$ i početnom tačkom u koordinatnom početku. Odgovarajući presecajući niz je

$$\omega = \dots 0100101001001 \dots$$

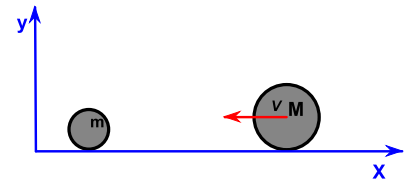
i nazivamo ga Fibonačijev niz. a) Pokazati da je Fibonačijev niz invarijantan na supstituciju

$$\sigma : 0 \mapsto 01, 1 \mapsto 0.$$

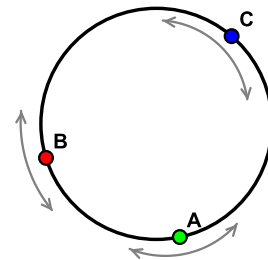
b) Neka je $\omega_n = \sigma^n(0)$. Pokazati da je dužina reči ω_n jednaka n -tom članu Fibonačijevog niza.

Primer 8: Model desne poluravni

Razmotrimo sistem od dve tačkaste mase (kuglice) mase m_1 i m_2 koje se kreću duž x -ose u oblasti $x \geq 0$ (u $x = 0$ se nalazi zid). Sudari među kuglicama i između zida i kuglica su apsolutno elastični. Pokazati da je ovaj sistem ekvivalentan bilijaru.

**Primer 9: Sudar perli**

Razmotrimo sistem od tri tačkaste mase (kuglice) mase m_1 , m_2 i m_3 koje se kreću na kružnici. Sudari među kuglicama su apsolutno elastični. Pokazati da je ovaj sistem ekvivalentan bilijaru unutar oštro-uglog trougla.

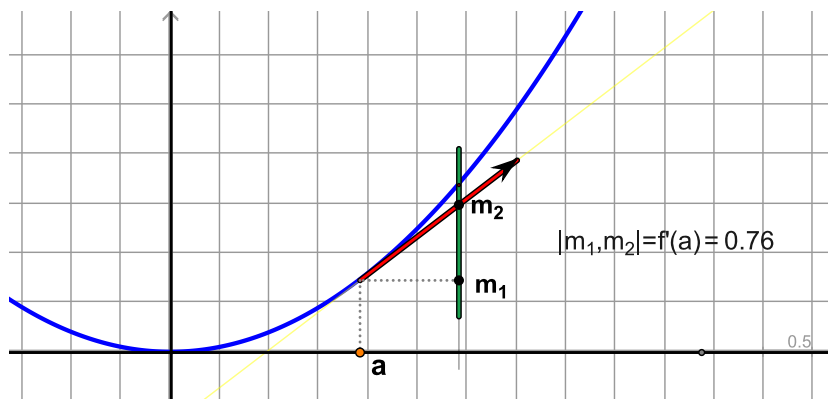


1 Dodatak: Matematička analiza u slicama

U ovom dodatku su date ilustracije nekoliko stavova iz matematičke analize koji, po mišljenju autora, obično nedostaju u udžbenicima, a mogu da budu od velike koristi studentima. Program Cinderella je veoma zahvalan za kreiranje takvih slika, a takođe i dobar alat za proveru za svakog ko se po prvi put susreće sa teoremom. Dobra pedagoška praksa, a to je i glavni cilj kursa kojem je ova knjiga posvećena, da prilikom praćenja matematičkih i fizičkih kurseva treba aktivno učestvovati kroz korisne alate pristupačne u programima Mathematica i Cinderella.

Primer 10: Osa nagiba

Pomoću "ose nagiba," vertikalne prave na rastojanju 1 od tekuće tačke P na krivoj γ prikazati vrednost prvog izvoda funkcije f u tački $P = (a, f(a))$.

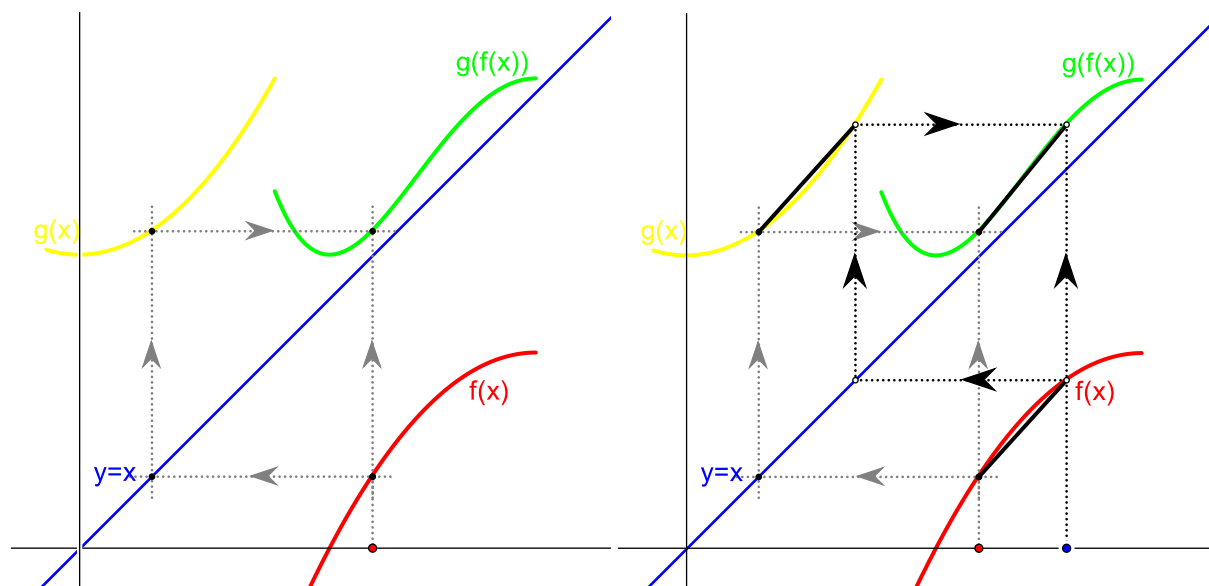


Slika 1.1: Prvi izvod funkcije na osi nagiba.

Osa nagiba je vertikalna prava $x = a + 1$, na rastojanju 1 od tačke a u kojoj se računa izvod. Kako je nalegla kateta jedinična, dužina naspramne katete je jednaka tangensu ugla tangente na krivu, u odnosu na horizontalu, tj. prvom izvodu $f'(a)$.

Primer 11: Izvod kompozicije

Prikazati metod konstrukcije grafike kompozicije funkcija $g \circ f$, a zatim, na osnovu te konstrukcije, izvesti pravilo za izvod kompozicije funkcija.



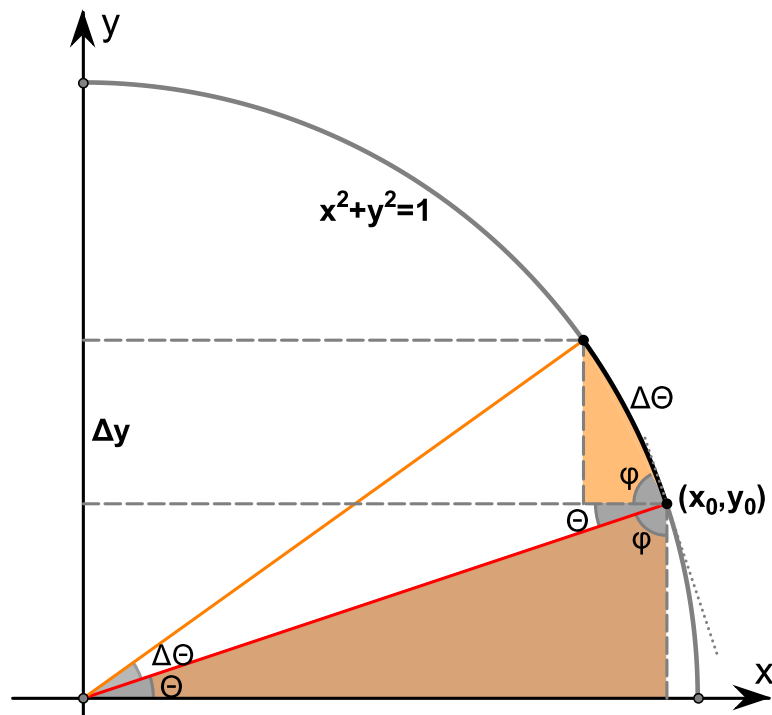
Pravilo konstrukcije grafika kompozicije funkcija je prikazano na slici gore levo. Pravilo izvoda kompozicije dobijamo na osnovu slike na desnoj strani. Nagibi sečica su: od f , $k_f = \frac{f(t) - f(x)}{t - x}$, od g , $k_g = \frac{g(f(t)) - g(f(x))}{f(t) - f(x)}$, i od $g \circ f$, $k_{g \circ f} = \frac{g(f(t)) - g(f(x))}{t - x}$. Očigledno važi $k_g \cdot k_f = k_{g \circ f}$, pa uzimajući u obzir da nagibi sečica pri $t \rightarrow x$ teže prvom izvodu odgovarajućih funkcija, direktno dobijamo pravilo diferenciranja kompozicije funkcija.

Primer 12: Izvod funkcije sin

Izvesti sa slike izvod sinusne funkcije.

Na slici su prikazane dve osenčene figure: tamnija je trougao, a svetlija je ograničena sa dve duži i kružnim lukom. Za jako mali ugao, kružni luk je je dobro aproksimiran sa duži, pa je i svetlija oblast približno trougao. Iz sličnosti dva osenčena "trougla" sledi

$$\frac{dy}{d\Theta} \approx \frac{\Delta y}{\Delta \Theta} = \frac{x_0}{1} = \sin \phi = \cos \Theta.$$



Slika 1.2: Izvod sinusne funkcije.

Primer 13: Izvod funkcije arctan

Izvesti sa slike izvod arkustangensne funkcije.

Luk AB pripada kružnici radijusa $\sqrt{1+t^2}$, a luk CD kružnici radijusa $\sqrt{1+(t+\Delta t)^2}$, pa je $\Theta = \text{arctg}(t+\Delta t) - \text{arctg}t$. Površina sektora AOB je manja od površine trougla OAD , a koja je jednaka $\Delta t/2$, pa sledi nejednakost:

$$\frac{1}{2}(1+t^2)(\text{arctg}(t+\Delta t) - \text{arctg}t) < \frac{\Delta t}{2},$$

tj.

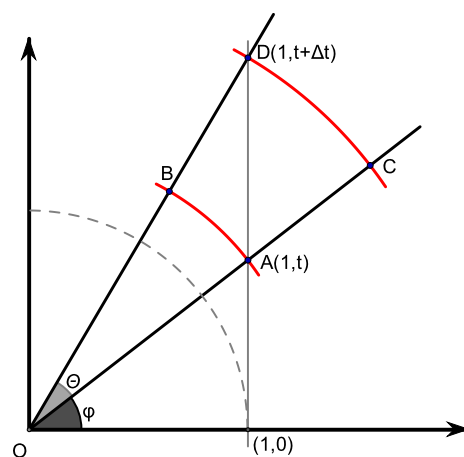
$$\frac{\text{arctg}(t+\Delta t) - \text{arctg}t}{\Delta t} < \frac{1}{1+t^2}.$$

Sa druge strane, površina trougla OAD je manja od površine sektora COD , pa važi i

$$\frac{\Delta t}{2} < (1+(t+\Delta t)^2)(\text{arctg}(t+\Delta t) - \text{arctg}t),$$

tj.

$$\frac{1}{1+(t+\Delta t)^2} < \frac{\text{arctg}(t+\Delta t) - \text{arctg}t}{\Delta t}.$$



Slika 1.3: Izvod funkcije arctan.

Na osnovu stava o dva policajca, iz prethodne dve jednakosti pri $\Delta t \rightarrow 0$ sledi

$$\frac{d}{dt} \operatorname{arctg} t = \frac{1}{1+t^2}.$$

U prethodnom zadatku koordinate tačka su parametrizovane sa $t = \operatorname{tg} \varphi$, $\Delta t = \operatorname{tg} \Theta$.

Primer 14: Izvod tangensa

Potpuno analognim rasuđivanjem izvesti izraz za $\operatorname{tg} x$ tako što koordinate istih tačka parametrizujemo preko uglova

$$\operatorname{tg} \varphi, \operatorname{tg} \Theta \equiv \operatorname{tg}(\varphi + \Delta \varphi).$$

Primer 15: Izvod funkcije arcsin

Izvesti sa slike izvod arkussinusne funkcije.

Površinu osenčenog sektora možemo izračunati na dva načina:

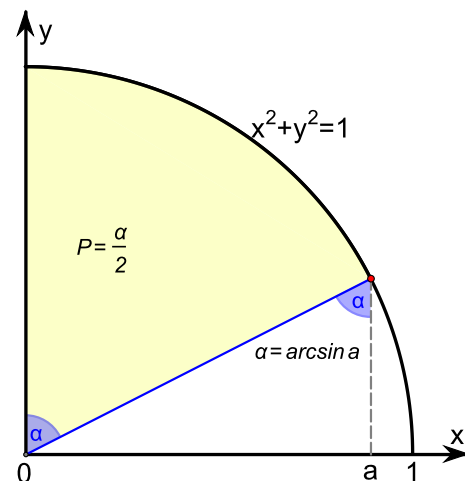
$$\frac{1}{2} \alpha = \int_0^a \sqrt{1-x^2} dx - \frac{1}{2} a \sqrt{1-a^2}.$$

Kako je $\alpha = \arcsin a$, diferenciranjem obe strane dobijamo

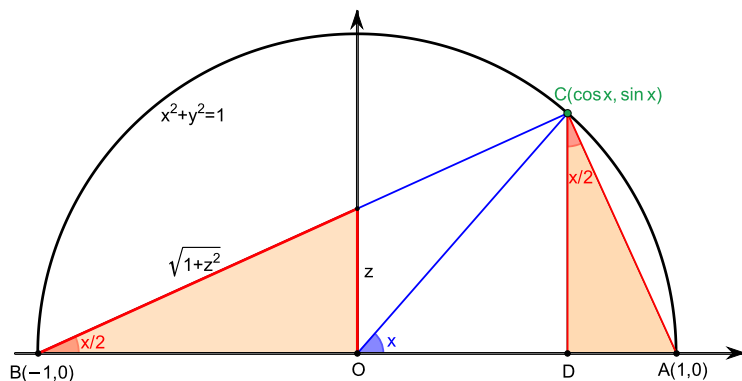
$$\frac{d}{da} \arcsin a = 2\sqrt{1-a^2} - \left(\sqrt{1-a^2} - \frac{a^2}{\sqrt{1-a^2}} \right) = \frac{1}{\sqrt{1-a^2}}.$$

Primer 16: Vajerštrasove smene

Izvesti sa slike vezu trigonometrijskih funkcija i diferencijala od tangensa poluugla.



Slika 1.4: Izvod funkcije arcsin.



Slika 1.5: Vajerštrasove smene.

Vajerštrasove poznate smene daju zapravo vezu između dve standardne parametrizacije kruga: preko sinusa i kosinusa i preko tangensa polovine ugla (važne u algebarskoj geometriji). Na slici je uvedena oznaka $z = \tan(x/2)$, dok je stranica $|CA| = 2 \sin \frac{x}{2} = \frac{2z}{\sqrt{1+z^2}}$.

Iz sličnosti osenčenih trouglova sledi:

$$\frac{|CD|}{|CA|} = \frac{1}{\sqrt{1+z^2}} \Rightarrow \sin x = \frac{2z}{1+z^2},$$

a takođe i

$$\frac{|DA|}{|CA|} = \frac{z}{\sqrt{1+z^2}} \Rightarrow \cos x = \frac{1-z^2}{1+z^2}.$$

Oдавде se lako dobija i $dx = \frac{2dz}{1+z^2}$.

Primer 17: Funkcije dvostrukog ugla

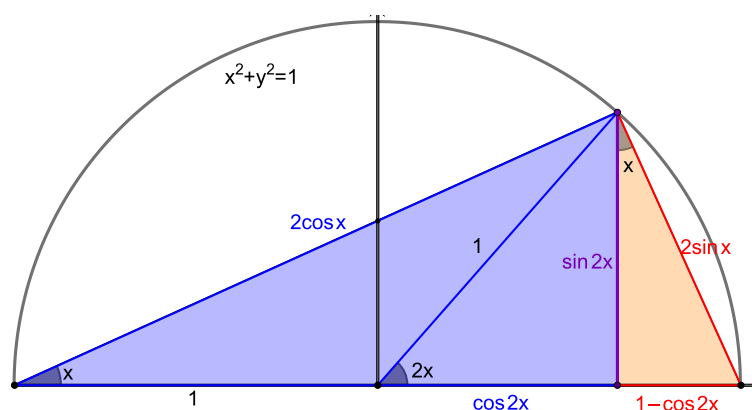
Koristeći sliku iz prethodnog zadatka, odrediti izraze za $\sin 2x$ i $\cos 2x$.

Rešenje je prikazano na slici ispod. Hipotenuza velikog trougla je dužine 2, pa su joj odgovarajuće katete $2 \cos x$, i $2 \sin x$. Iz plavog trougla dobijamo

$$\sin x = \frac{\sin 2x}{2 \cos x} \Rightarrow \sin 2x = 2 \sin x \cos x.$$

Iz crvenog trougla dobijamo

$$\sin x = \frac{1 - \cos 2x}{2 \sin x} \Rightarrow \cos 2x = 1 - 2 \sin^2 x = \cos^2 x - 1 = \cos^2 x - \sin^2 x.$$



Slika 1.6: Trigonometrijske funkcije dvostrukog ugla.

Primer 18: Trigonometrijski identiteti

Na jednoj slici prikazati (sve) osnovne trigonometrijske identitete.

Iz pravouglanih trouglova na slici lako je uočiti da važi

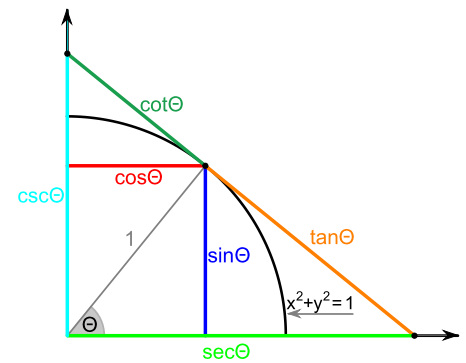
$$\sin^2 \Theta + \cos^2 \Theta = 1$$

$$\operatorname{tg}^2 \Theta + 1 = \sec^2 \Theta,$$

$$\operatorname{cot}^2 \Theta + 1 = \operatorname{csc}^2 \Theta,$$

$$(\operatorname{tg} \Theta + \operatorname{ctg} \Theta)^2 = \sec^2 \Theta + \operatorname{csc}^2 \Theta,$$

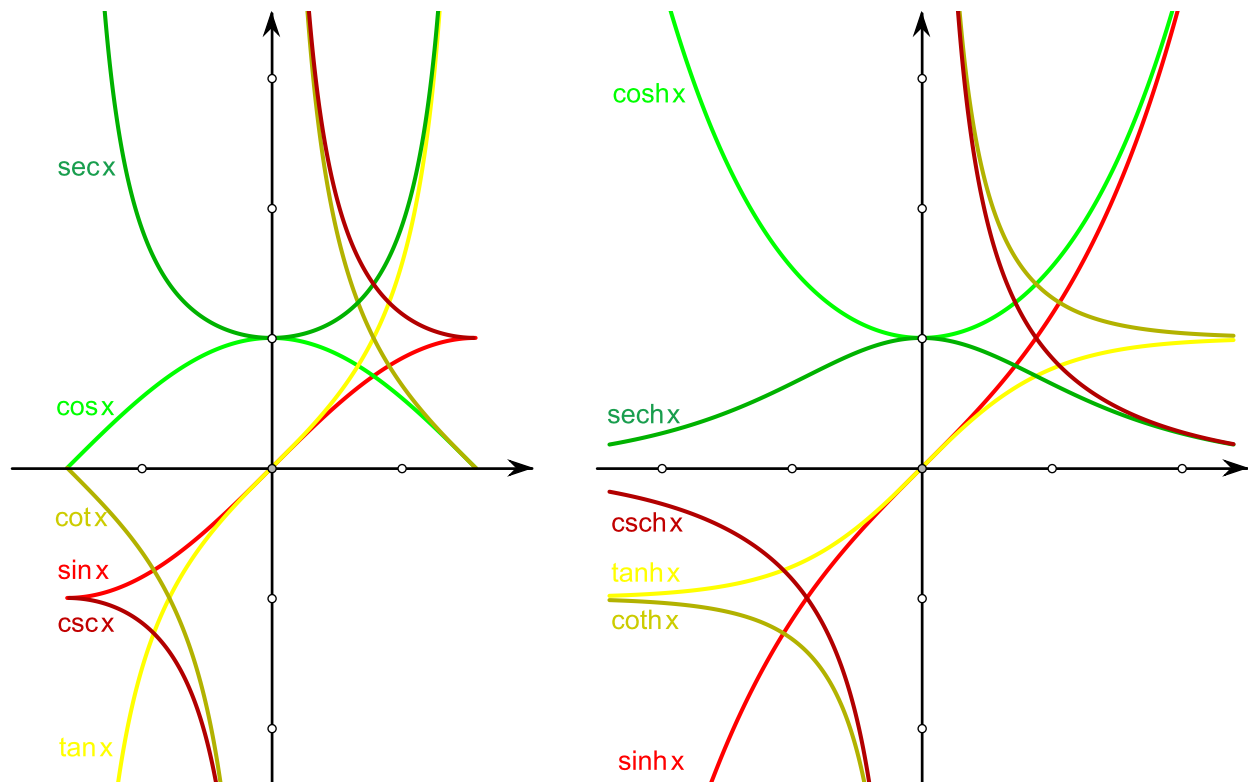
$$\frac{\sec \Theta - \cos \Theta}{\cos \Theta} = \frac{\sin \Theta}{\operatorname{csc} \Theta - \sin \Theta}.$$



Slika 1.7: Trigonometrijski identiteti.

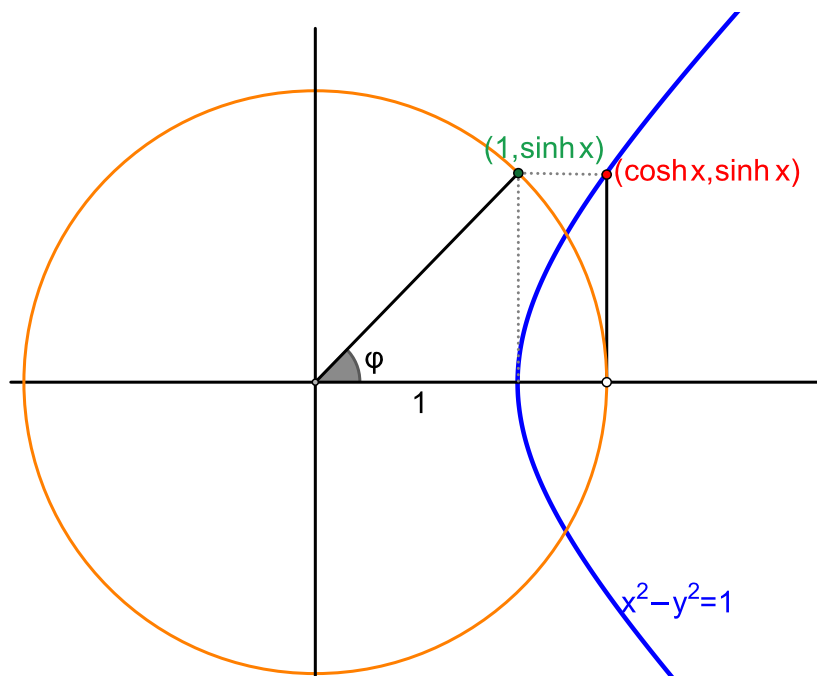
Primer 19: Hiperboličko-trigonometrijski identiteti

Na istoj slici uporedno prikazati hiperboličke i trigonometrijske funkcije. Da li postoji sličnost?



Slika 1.8: Grafici trigonometrijske i hiperboličke funkcija.

Iz sličnosti zaključujemo da pored jednostavne veze preko kompleksnih brojeva postoji i realna veza. Vezu je moguće ilustrovati slikom.



Slika 1.9: Veza hiperboličnih i trigonometrijskih funkcija.

Sa slike očitavamo

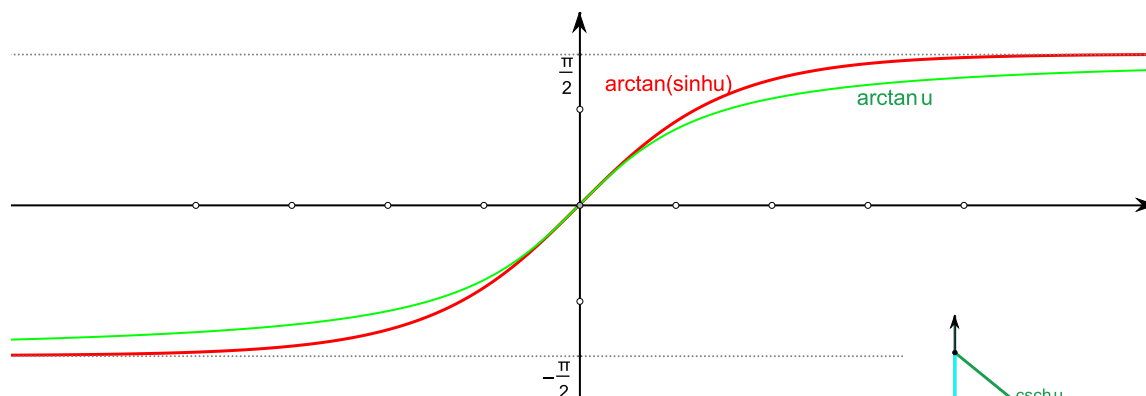
$$\begin{aligned} \operatorname{tg} \varphi &= \sinh u, \operatorname{sec} \varphi = \cosh u, \\ \sin \varphi &= \operatorname{tgh} u, \cos \varphi = \operatorname{sech} u. \end{aligned}$$

Veza između trigonometrijskih i hiperboličkih funkcija je definisana vezom ugla φ i hiperboličkog parametra u ,

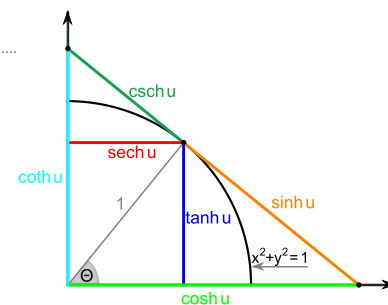
$$\varphi \equiv \operatorname{gn}(u) = \operatorname{arctg}(\sinh u) = \operatorname{arcsin}(\operatorname{tgh} u)$$

koju obično nazivamo "Gudermanianom". Ako Pitagorejske

Po K. Gudermanu (C. Gudermann).



identitete iz prethodnog zadatka primenimo na Gudermaniane, dobijamo odgovarajuće identitete hiperboličke trigonometrije.

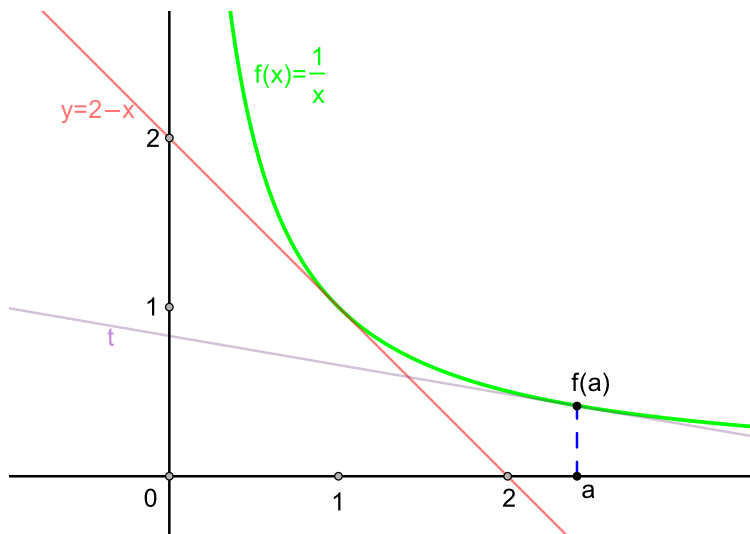


Slika 1.11: Identitete hiperboličke trigonometrije.

Primer 20: $x + \frac{1}{x}$

Da li je $\pi/e + e/\pi > 2$?

Ovaj, kao i brojni drugi zadaci, se jednostavnije rešava zamenom konkretnog problema opštijim. Naime, nejednakost $\pi/e + e/\pi > 2$ je specijalni slučaj $x + \frac{1}{x} \geq 2$, $x > 0$. Međutim, i drugu jednakost je moguće zameniti veoma generalnim stavom da *grafik funkcije konkavne nagore leži iznad svoje tangente*. Tangenta na hiperbolu $y = 1/x$ u



tački (1, 1) je prava $y = 2 - x$, pa je, na osnovu prethodnog principa,

$$\frac{1}{x} \geq 2 - x \Leftrightarrow x + \frac{1}{x} \geq 2.$$

Ovaj princip je moguće primeniti kod većine nejednakosti.

Primer 21: AM vs. GM

Slikom pokazati da važi a) standardna i b) generalizovana nejednakost aritmetičke i geometrijske sredine.

Najpre zapišimo nejednakost $\sqrt{ab} \leq \frac{a+b}{2}$ tako da bude bezdimenzi-ona, tj. po jednoj promenljivoj, npr. deljenjem obe strane sa a :

$$\sqrt{b/a} \leq \frac{1+b/a}{2} \Leftrightarrow \sqrt{x} \leq \frac{1+x}{2}, x = b/a.$$

Generalizovana nejednakost trougla glasi:

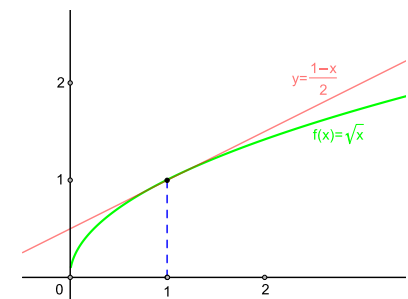
$$G \equiv \sqrt[n]{a_1 a_2 \cdots a_n} \leq \frac{a_1 + a_2 + \cdots + a_n}{n} \equiv A,$$

Strategija 1: zamena brojeva parametrima.

Strategija 2: prepoznavanje funkcionalnih veza među parametrima.

Slika 1.12: Odnos tangente i grafika konkavne funkcije.

tng. na $y = f(x)$ u (x_0, y_0) je $y - y_0 = f'(x_0)(x - x_0)$



Slika 1.13: Odnos tangente i grafika konkavne nadole funkcije.

uz jednakost kada su svi a_i jednaki. Nejednakost sabiraka i proizvoda ukazuje na nejednakost između linearne i eksponencijalne funkcije, npr. $e^x \geq ex$.

Ako označimo $x = a_i/G$, tada za svako $i \in \{1, 2, \dots, n\}$ važi $e^{\frac{a_i}{G}} \leq e^{a_i/G}$, pa množenjem dobijamo

$$e^n \cdot \frac{a_1}{G} \dots \frac{a_n}{G} \leq e^{(a_1 + \dots + a_n)/G} \Leftrightarrow e^n \leq e^{nA/G} \Rightarrow G \leq A.$$

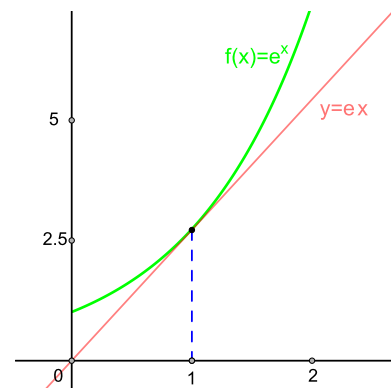
Nekoliko sledećih primera je rešeno "bez reči."

Primer 22: Bernuli

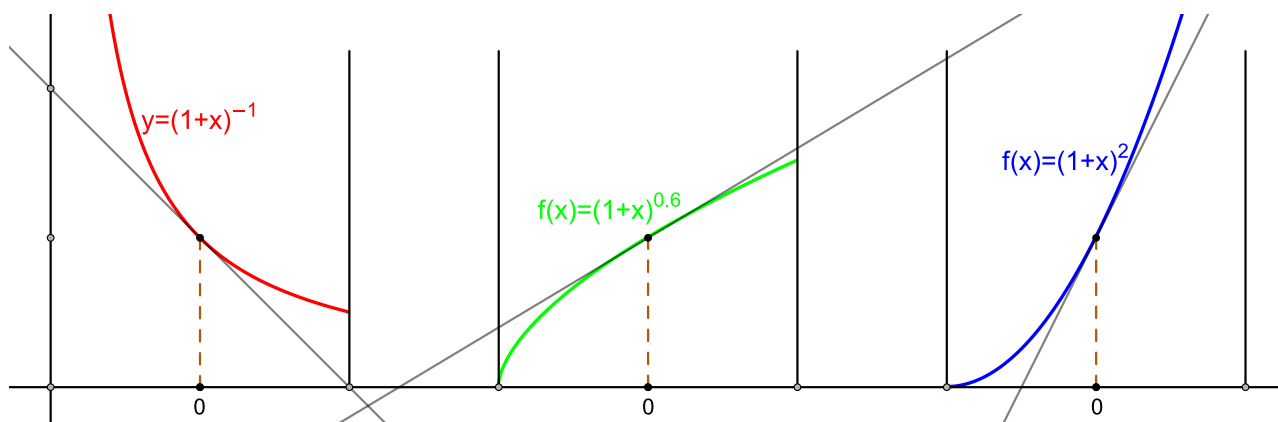
Ilustrovati važenje Bernulijevе nejednakosti:

$$(1+x)^r \geq 1+rx, \quad r > 0 \wedge r < 0,$$

$$(1+x)^r \leq 1+rx, \quad 0 < r < 1.$$



Slika 1.14: Odnos tangente i grafika eksponencijalne funkcije.

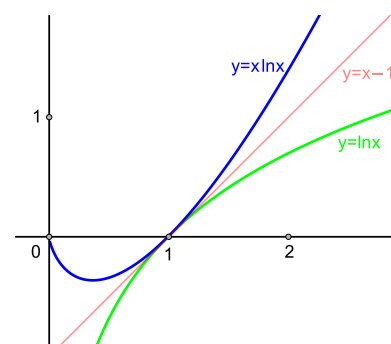


Slika 1.15: Bernulijevе nejednakosti.

Primer 23

Ilustrovati važenje nejednakosti:

$$\ln x \leq x - 1 \leq x \ln x, \quad x > 0.$$



Primer 24

Ilustrovati važenje nejednakosti:

$$\sin x < x < \operatorname{tg} x \Leftrightarrow x \cos x < \sin x < x, \quad x \in (0, \pi/2).$$

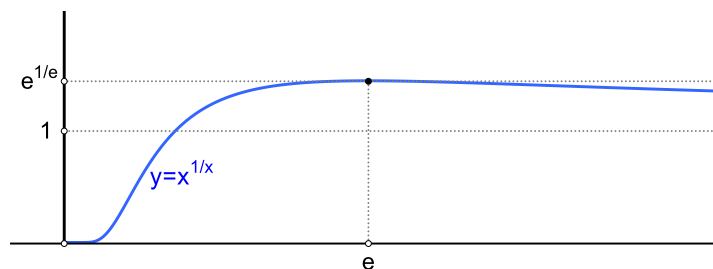
Primer 25: e^π ili π^e

Šta je veće e^π ili π^e ? Ilustrovati odgovor. (Jakob Steiner, Crelle's Journal, 1849.)

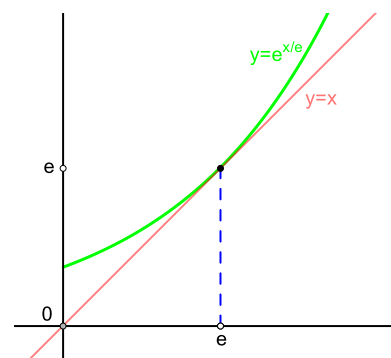
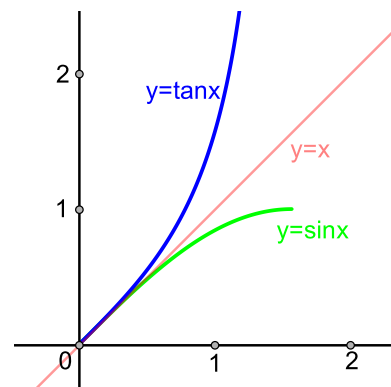
Sa grafika je očigledno da važi $e^{x/e} \geq x$, odnosno uzimanjem x -tog korena obe strane, $e^{1/e} \geq x^{1/x}$, pa i

$$\underbrace{e^{1/e}}_{\approx 23.14} \geq \underbrace{\pi^{1/\pi}}_{\approx 22.46}.$$

Prethodno tvrđenje je ekvivalentno stavu da je $e^{1/e}$ maksimalna vrednost funkcije $x^{1/x}$. Sa ovom funkcijom blisko je povezan i



naredni zadatak (za samostalni rad).



Slika 1.16: Grafik funkcije $f(x) = e^{x/e}$ i tangenta u tački $x = e$.

Slika 1.17: Grafik funkcije $f(x) = x^{1/x}$.

Primer 26

Razmotrimo jednačinu:

$$x^{x^{x^{\dots}}} = 2.$$

Kako je sam eksponent jednak čitavom izrazu sa leve strane, to je jednačina ekvivalentna sa $x^2 = 2$, pa zaključujemo da je $x = \sqrt{2}$. Sa druge strane, da je jednačina glasila

$$x^{x^{x^{\dots}}} = 4,$$

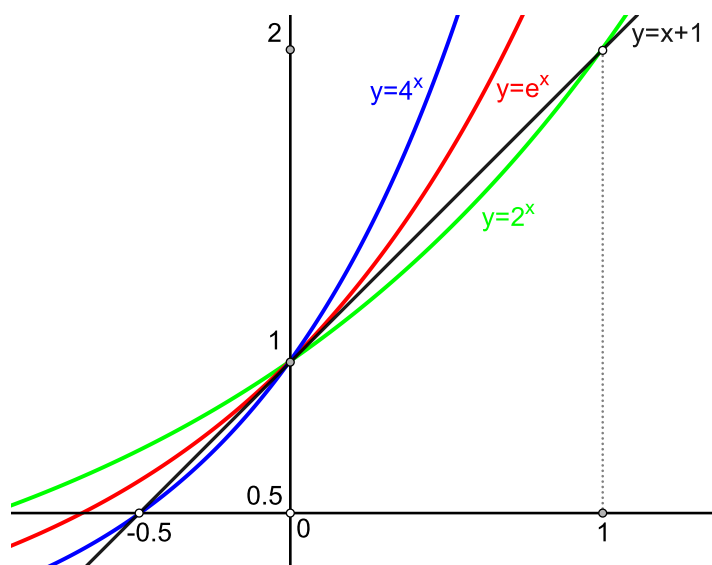
istim metodom bi došli do rešenja $x = \sqrt[4]{4} = \sqrt{2}$. Čemu je onda jednak izraz

$$\sqrt{2}^{\sqrt{2}^{\sqrt{2}^{\dots}}} ?$$

Primer 27: e - limes

Ilustrovati definiciju od e :

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e.$$



Slika 1.18: Grafici funkcija 2^x , e^x , 4^x .

Na slici su prikazana tri grafika funkcije $f(x) = b^x$, za $b = 2, e$ i 4 , kao i tangenta na e^x u $x = 0$ ($y = x + 1$). Ona je ujedno i sečica za sve grafike funkcija b^x , $b \neq e$. Ta znači da pored tačke $(0, 1)$ postoji još

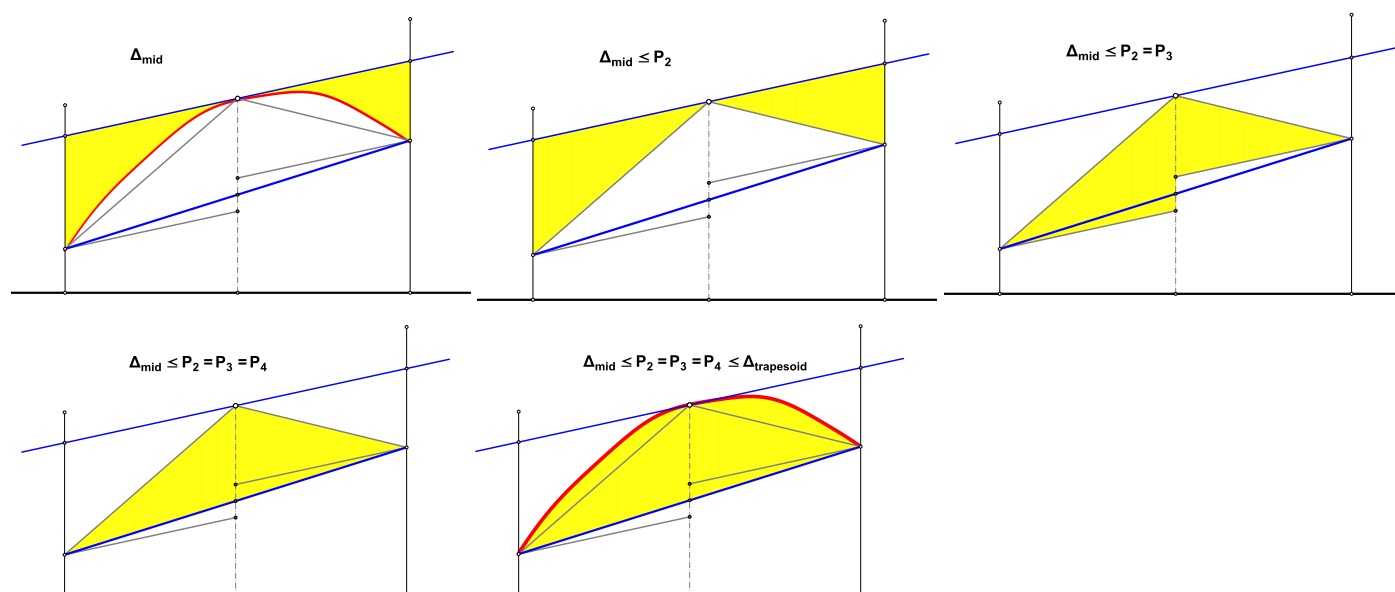
jedna zajednička tačka sa pravom $y = x + 1$, tj. postoji $t \neq 0$, $b^t = 1 + t$. Za $t \neq 0$, rešenje je jednostavno: $b = (1 + t)^{1/t}$. Dakle, parametar t zavisi od b , pa se variranjem b tačke t pomera od ili ka 0. Sečica prelazi u tangentu samo u slučaju kad je $b = e$, pa sledi da je

$$\lim_{t \rightarrow 0} (1 + t)^{1/t} = e.$$

Primer 28: Pravilo srednje tačke vs. trapezoidno

Vizuelno pokazati da je za konkavne nagore, pozitivne, neprekidne funkcije aproksimacija srednje tačke bolja od trapezoidne. Da li je moguće poboljšati pravilo srednje tačke?

Pratiti sledeći niz slika. Pravilo srednje tačke nije moguće po-



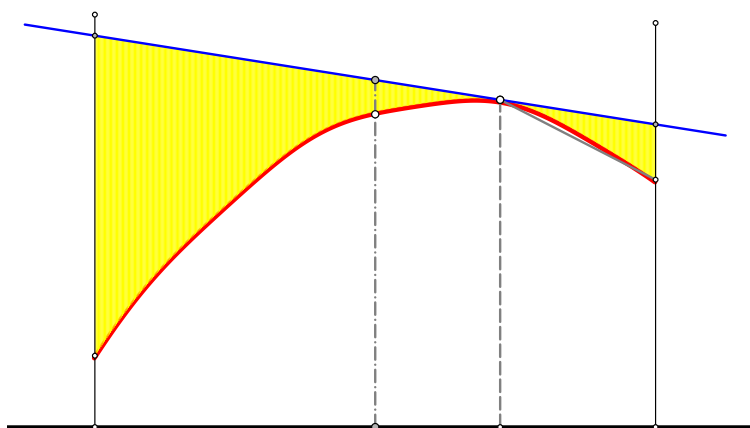
Slika 1.19: Nejednakost za konkavne funkcije.

boljšati uzimanjem druge tačke iz intervala. To se vidi na slici ispod.

Primer 29: Simpson za kubike

Vizuelno pokazati da je Simpsonova aproksimacija egzaktna i za kubike (a ne samo za kvadrike).

Opšta tehnika aproksimacije integrala se zasniva na podeli intervala integracije na male podintervale na kojima se integrand aproksimira jednostavnijom funkcijom, podesnom za integraciju. U aproksimaciji



srednje tačke, integrand se zamenjuje konstantnom funkcijom jednakom intgrandu na sredini podintevala, pa je aproksimirana vrednost na podintervalu $[a, b]$

$$(b - a)f\left(\frac{a + b}{2}\right).$$

U slučaju trapezoidnog pravila, aproksimaciona funkcija je linearna i važi:

$$\frac{1}{2}(b - a)(f(a) + f(b)).$$

Simpsonovo pravilo zamenjuje integrand sa parabolom koja ima iste vrednosti u graničnim $(a$ i $b)$ i srednjoj $(m = (a + b)/2)$ tački podintevala. Parabolu dobijamo Lagranževom interpolacijom,

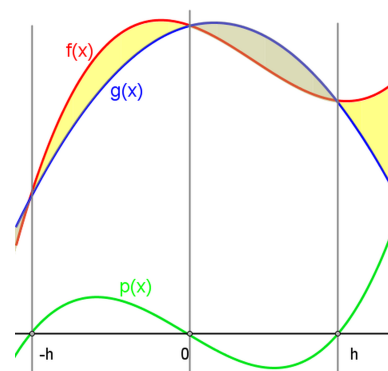
$$P(x) = f(a)\frac{(x - m)(x - b)}{(a - m)(a - b)} + f(m)\frac{(x - a)(x - b)}{(m - a)(m - b)} + f(b)\frac{(x - a)(x - m)}{(b - a)(b - m)}.$$

Integracijom dobijamo

$$\int_a^b P(x) dx = \frac{b - a}{6} \left(f(a) + 4f\left(\frac{a + b}{2}\right) + f(b) \right).$$

Jasno je da se Simpsonovim metodom dobija egzaktna vrednost integrala kvadratnog polinoma. Manje je poznato da isto važi i za kubni!

Na slici je prikazan grafik (crvena) kubnog polinoma $f(x)$ na intervalu $[-h, h]$ zajedno sa odgovarajućom Simpsonovom parabolom (plavo) $g(x)$ i njihovom razlikom (zeleno) $p(x) = f(x) - g(x)$. Kako je $p(-h) = p(0) = p(h) = 0$, to je $p(x) = ax(x - h)(x + h) = ax^3 - ah^2x = 0$. Dakle, funkcija razlike p je neparna pa je $\int_{-h}^h p(x) dx = 0$.



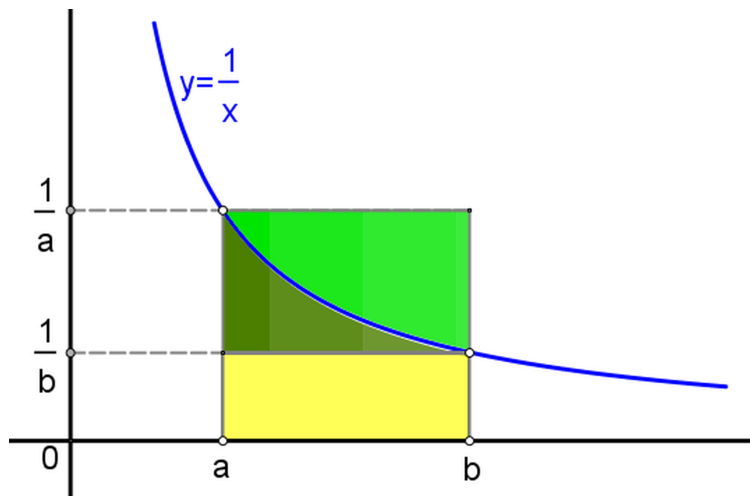
Slika 1.20: Simpsonov metod na kubnoj krivoj.

Primer 30: Neperova nejednakost

Vizuelno pokazati važenje Neperove nejednakosti,

$$0 < a < b, \text{ tada } \frac{1}{b} < \frac{\ln b - \ln a}{b - a} < \frac{1}{a}$$

i na osnovu nje, dokazati definicioni limes za e .



Slika 1.21: Neperova nejednakost

Sa slike je očigledno da je površina ispod hiprbole ograničena sa upisanim i opisanim pravougaonikom pa je

$$\frac{(b-a)}{b} < \int_a^b \frac{1}{x} dx < \frac{(b-a)}{a},$$

odakle integracijom dobijamo traženu nejednakost. Ako izaberemo specifične vrednosti: $a = n$ i $b = n + 1$, iz Neperove nejednakosti sledi

$$\frac{1}{n+1} < \ln \left(1 + \frac{1}{n} \right) < \frac{1}{n},$$

pa eksponenciranjem dolazimo do

$$e^{n/(n+1)} < \left(1 + \frac{1}{n} \right)^n < e.$$

Iz poslednje nejednakosti, na osnovu stava "o dva policajca" sledi limes:

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n = e.$$

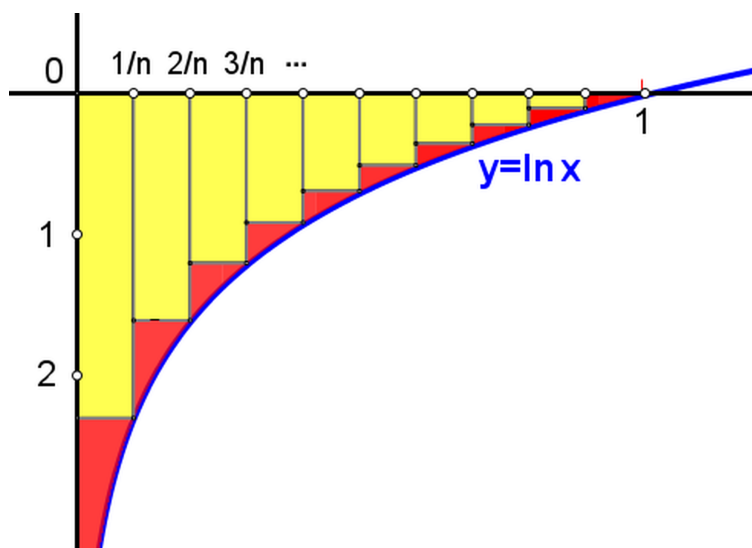
Primer 31: e - limes 2

Vizuelizovati asimptotiku od $\sqrt[n]{n!}$, tj. važenje limesa $\lim_{n \rightarrow \infty} \frac{n}{\sqrt[n]{n!}} = e$.

Najpre primetimo da važi:

$$\ln \left(\frac{\sqrt[n]{n!}}{n} \right) = \ln \left(\frac{n!}{n^n} \right) = \ln \left(\frac{n!}{n^n} \right) \cdot \frac{1}{n} = \ln \left(\prod_{k=1}^n \frac{k}{n} \right) \cdot \frac{1}{n} = \sum_{k=1}^n \ln \left(\frac{k}{n} \right) \cdot \frac{1}{n}.$$

Poređenje poslednje sume sa integralom funkcije $\ln x$ u intervalu $[0, 1]$ je prikazano na slici ispod. Kako je



Slika 1.22: Rimanova suma za funkciju $\ln x$.

$$\int_0^1 \ln x \, dx = [-x + x \ln x]_0^1 = -1,$$

zaključujemo da je i

$$\lim_{n \rightarrow \infty} \left(\frac{\sqrt[n]{n!}}{n} \right) = \frac{1}{e},$$

odnosno,

$$\lim_{n \rightarrow \infty} \left(\frac{n}{\sqrt[n]{n!}} \right) = e.$$

Primer 32: Ermit - Adamar

Vizuelno pokazati važenje Ermit-Adamarove nejednakosti i primeniti je na funkcije $\cos(x)$ i e^x .

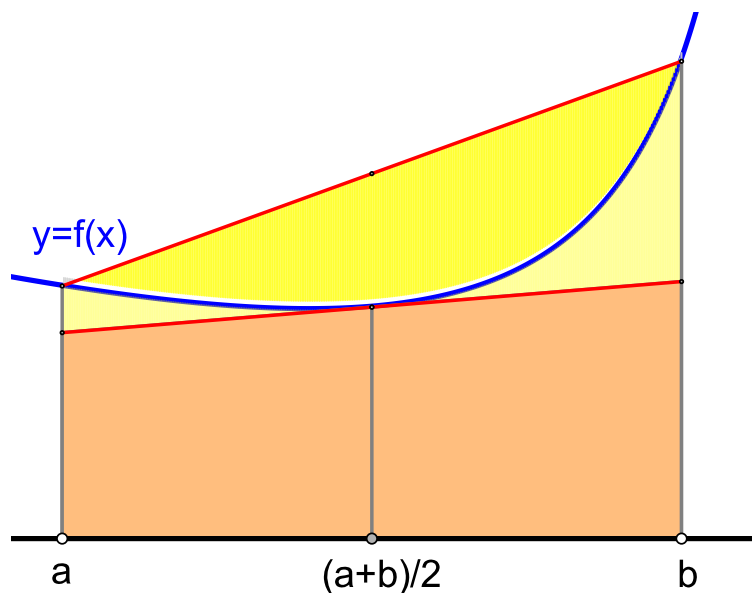
Ermit-Adamarove nejednakost glasi:

Ako je f neprekidna i konkavna nagore na $[a, b]$, tada

$$f\left(\frac{a+b}{2}\right) \leq \frac{1}{b-a} \int_a^b f(x) dx \leq \frac{f(a)+f(b)}{2}$$

i sa obrnutim nejednakostima za konkavnu nadole.

Dokaz je očigledan sa slike. Za $f(x) = \cos(x)$, (konkavna nadole



Slika 1.23: Ermit-Adamarove nejednakost.

na $[-\pi/2, \pi/2]$) dobijamo nejednakost:

$$\frac{\cos a + \cos b}{2} \leq \frac{\sin b - \sin a}{b-a} \leq \cos \frac{a+b}{2}.$$

Za $a = 0$ i $b = t$, nejednakost se svodi na

$$\frac{1 + \cos t}{2} \leq \frac{\sin t}{t} \leq \cos \frac{t}{2}.$$

Ovo je popravka poznate nejednakosti $\cos t \leq (\sin t)/t \leq 1$.

Primenom na funkciju $f(x) = e^x$, na intervalu $[\ln a, \ln b]$, $0 < a < b$, dobijamo

$$e^{(\ln a + \ln b)/2} < \frac{e^{\ln b} - e^{\ln a}}{\ln b - \ln a} < \frac{e^{\ln a} + e^{\ln b}}{2},$$

tj.

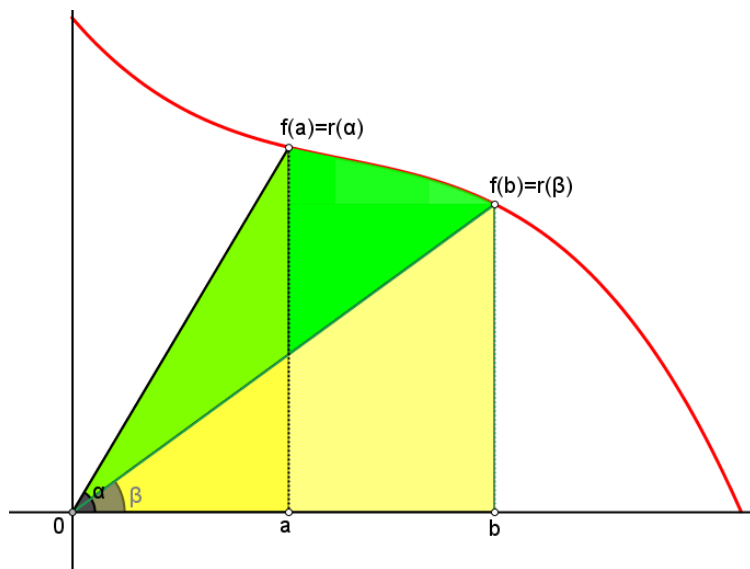
$$\sqrt{ab} < \frac{b-a}{\ln b - \ln a} < \frac{a+b}{2}.$$

Generalizacija G-A nejednakosti.

Ako stavimo $a = n$, $b = n+1$, dobijamo definiciju broja e (stav o dva policajca).

Primer 33: Polarna vs. Dekartova površina

Vizuelno prikazati odnos među izrazima za površinu u polarnim i Dekartovima koordinatama.



Slika 1.24: Grafik funkcije $f(x) = r(\phi)$.

Površina (zelenog trougla) u Dekartovim koordinatama glasi

$$\begin{aligned} A_D &= \int_a^b f(x) dx + \frac{1}{2}af(a) - \frac{1}{2}bf(b) \\ &= \frac{1}{2} \left(xf(x) \Big|_a^b - 2 \int_a^b f(x) dx \right). \end{aligned}$$

Sa druge strane, u polarnim kordinatama ista površina glasi

$$\begin{aligned} A_p &= \frac{1}{2} \int_\alpha^\beta (r(\phi))^2 d\phi \\ &= \frac{1}{2} \int_\alpha^\beta (r(\phi) \cos \phi)^2 d(\tan \phi) \\ &= \frac{1}{2} \int_a^b x^2 d\left(\frac{f(x)}{x}\right) \\ &= \frac{1}{2} \left(x^2 \frac{f(x)}{x} \Big|_a^b - \int_a^b \frac{f(x)}{x} 2x d(x) \right). \end{aligned}$$

Smena: $x = r(\phi) \cos \phi$, $f(x) \equiv y = r(\phi) \sin \phi$.

Prema tome, površine su jednake.

Primer 34: Stirlingova formula

Demonstrirati važenje Stirlingove formule kao aproksimacije od $\ln n!$.

Iz $\ln n! = \sum_1^n \ln k$, sledi da je i

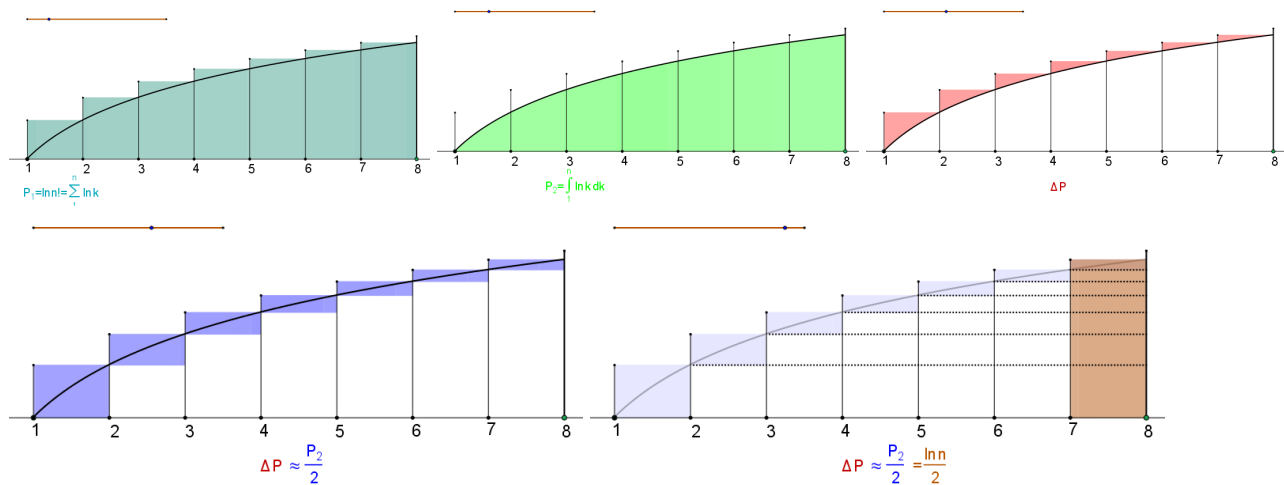
$$\ln n! \approx \int_1^n \ln k dk = n \ln n - n + 1.$$

Eksponciranjem dobijamo

$$n! \approx n^n \cdot e^{-n} \cdot e$$

$$\text{Stirling: } n! \approx n^n \cdot e^{-n} \cdot \sqrt{n} \cdot \sqrt{2\pi}.$$

Popravka $\sim \sqrt{n}$ se dobija aproksimacijom razlike diskretne i kontinualne krive sa nizom trouglova $P \approx (\ln n)/2$.

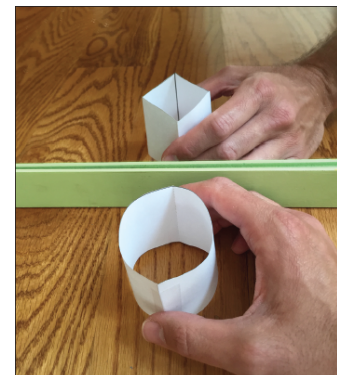


Slika 1.25: Stirlingova aproksimacija.

Primer 35: Sugiharin nemogući cilindar

Kokichi Sugihara, japanski profesor (inženjerstva) je posvetio decenije građenju objekata koji izgledaju i kreću se na prividno paradoksalan način. Na takmičenju koje organizuje Neural Correlate Society za 2016. godinu je osvojio drugu nagradu sa videom "Ambiguous Optical Illusion" (<http://bit.ly/ambcylin>).

Odrediti prostornu krivu koja se iz pravca $[0, 1, 1]$ vidi kao kvadrat (dijamant orijentacija), a iz pravca $[0, -1, 1]$ kao kružnica.



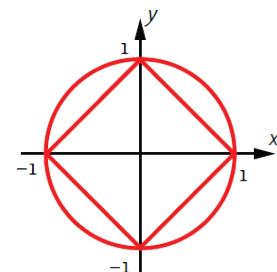
Polazna tačka su nam kružnica i kvadrat u xOy ravni (slika desno).

Jednačine gornjih (pozitivnih) polovina kvadrata i kružnice su:

$$f(x) = 1 - |x|,$$

$$g(x) = \sqrt{1 - x^2}.$$

Zamislamo sada dva posmatrača koji iz zadatih pravaca posmatraju traženu krivu $\mathbf{r}(t)$. Jedan će misliti da vidi $(x, f(x), 0)$, (kvadrat), a



drugi $(x, g(x), 0)$ (kružnica). Pozicija prvog je definisana vektorom $(0, a, a)$, a drugog $(0, -a, a)$, za dovoljno veliko a .

Oredimo sada prostornu krivu $\mathbf{r}(t)$. Fiksirajmo vrednost parametra t . Tada prvi posmatrač vidi tačku $A = (t, f(t), 0)$, a drugi $B = (t, g(t), 0)$. To znači da istovremeno, "zrak" (poluprava) iz A u pravcu $(0, 1, 1)$ i zrak iz B u pravcu $(0, -1, 1)$ prolaze kroz $\mathbf{r}(t)$. Situacija je prikazana na slici desno.

Prvi zrak je parametrizovan sa $\mathbf{r}_A(t) = (t, s + f(t), s)$, a drugi sa $\mathbf{r}_B(t) = (t, -s + g(t), s)$. Njihova tačka preseka je

$$(t, (g(t) + f(t))/2, (g(t) - f(t))/2).$$

Dakle, parametarska jednačina prostorne krive je

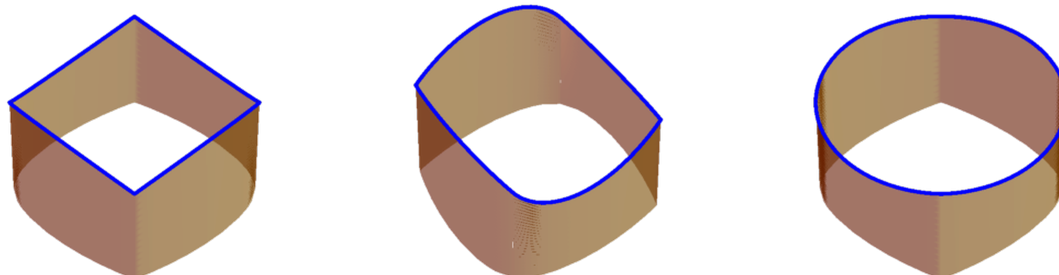
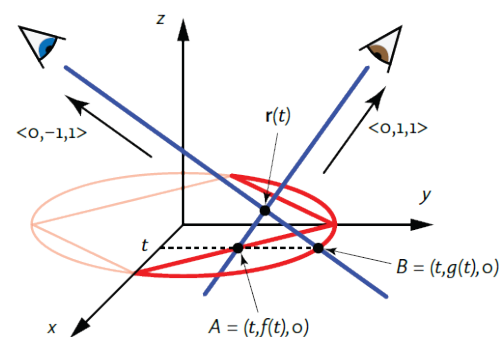
$$\mathbf{r}(t) = (t, (g(t) + f(t))/2, (g(t) - f(t))/2).$$

Konkretno, rešenje našeg zadatka je unija (gornje i donje polovine kružnice i kvadrata):

$$\mathbf{r}_1(t) = (t, (1 - |t| + \sqrt{1 - t^2})/2, (\sqrt{1 - t^2} - 1 + |t|)/2),$$

$$\mathbf{r}_2(t) = (t, (|t| - 1 - \sqrt{1 - t^2})/2, (1 - \sqrt{1 - t^2} - |t|)/2).$$

Rešenje je prikazano na slici ispod.



Slika 1.26: Sugihara cilindar - tri različita ugla gledanja.

Zumiranje je osnovna tehnika analize, ali važi i obratno analiza je osnovna tehnika zumiranja.

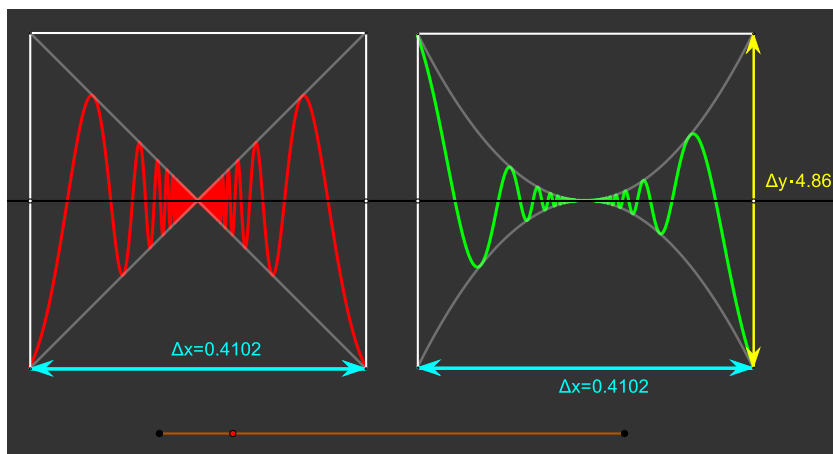
Primer 36: Zumiranje

Razmotrimo sledeće dve funkcije:

$$f(x) = \begin{cases} x \sin\left(\frac{1}{x}\right), & x \neq 0, \\ 0, & x = 0, \end{cases}$$

$$g(x) = \begin{cases} x^2 \sin\left(\frac{1}{x}\right), & x \neq 0, \\ 0, & x = 0. \end{cases}$$

Njihovi grafici izgledaju slično u okolini $x = 0$, i obe su neprekidne, ali samo je f diferencijabilna u nuli? Zašto je to tako?



Slika 1.27: Grafici funkcija $f(x)$ i $g(x)$ prikazani na način na koji se obično zamišljaju. Prisetiti da je grafik od $g(x)$ skaliran duž y -ose da bi izgledao sličnije grafiku od $f(x)$.

Iz

$$-|x| \leq x \sin(1/x) \leq |x|, \quad -x^2 \leq x^2 \sin(1/x) \leq x^2$$

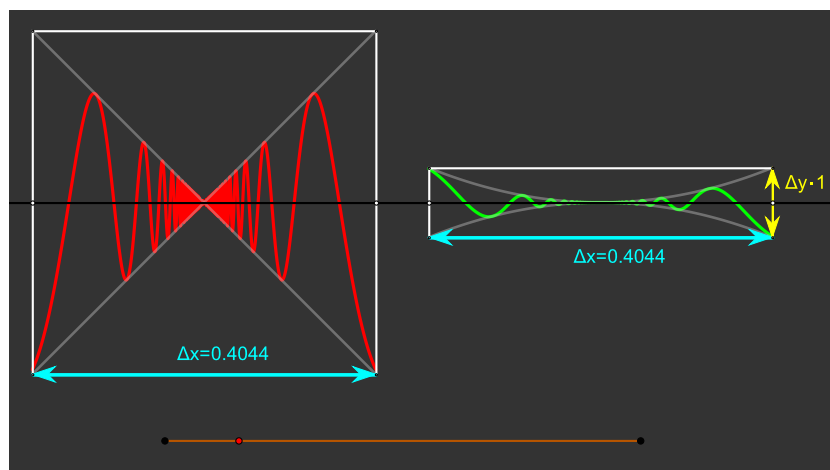
sledi da obe funkcije konvergiraju ka 0 u $x = 0$. Sa druge strane, iz same definicije prvog izvoda sledi da je

$$\lim_{x \rightarrow 0} \frac{g(x) - g(0)}{x - 0} = \lim_{x \rightarrow 0} \frac{x^2 \sin(1/x)}{x} = 0,$$

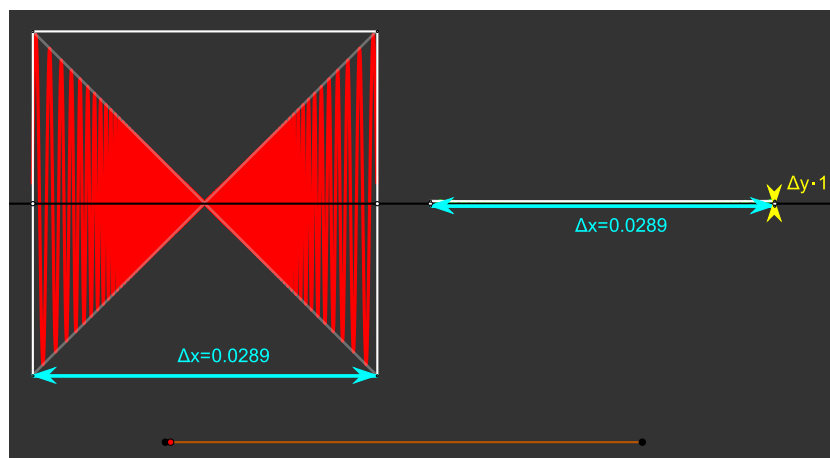
dok

$$\lim_{x \rightarrow 0} \frac{f(x) - f(0)}{x - 0} = \lim_{x \rightarrow 0} \frac{x \sin(1/x)}{x} = \lim_{x \rightarrow 0} \sin(1/x),$$

ne postoji.



Slika 1.28: Grafici funkcija $f(x)$ i $g(x)$ bez skaliranja grafika od $g(x)$ duž y -ose.



Slika 1.29: Zumirani grafici na kojima je očigledno da u slučaju funkcije f nagib tangente (k) može biti bilo koji broj $-1 \geq k \geq 1$, dok je u slučaju funkcije g nagib očigledan $k = 0$.

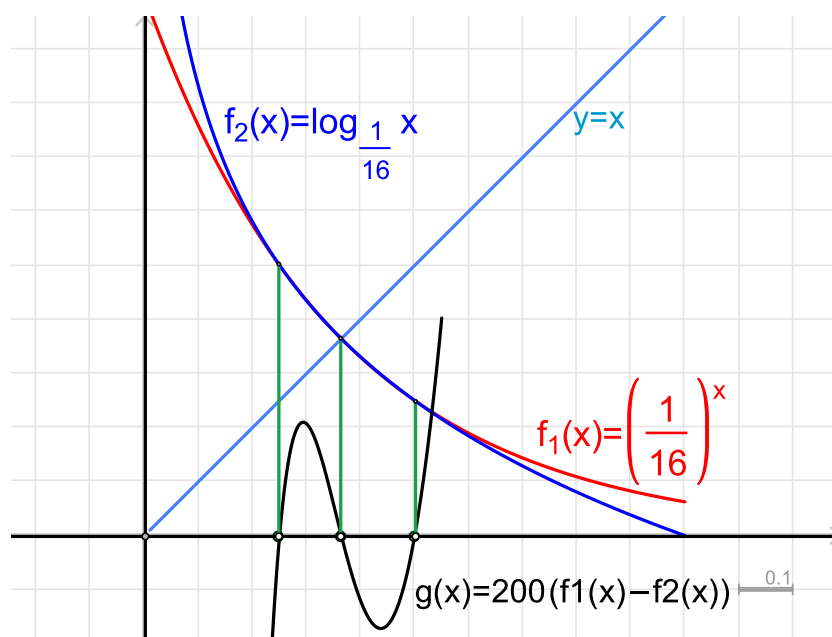
Primer 37: Log jednačina

Grafički rešiti jednačinu

$$\log_{1/16} x = \left(\frac{1}{16}\right)^x.$$

Da li jednačina ima više od jednog rešenja? Proveriti za $x = 1/2$ i $x = 1/4$.

Rešenje je prikazano na slici. Ideja je uvećati razliku među funkcijama, a to je postignuto plotovanjem funkcije $g(x) = 200(f_1(x) - f_2(x))$.



Slika 1.30: Grafičko rešenje jednačine

$$\log_{1/16} x = \left(\frac{1}{16}\right)^x.$$

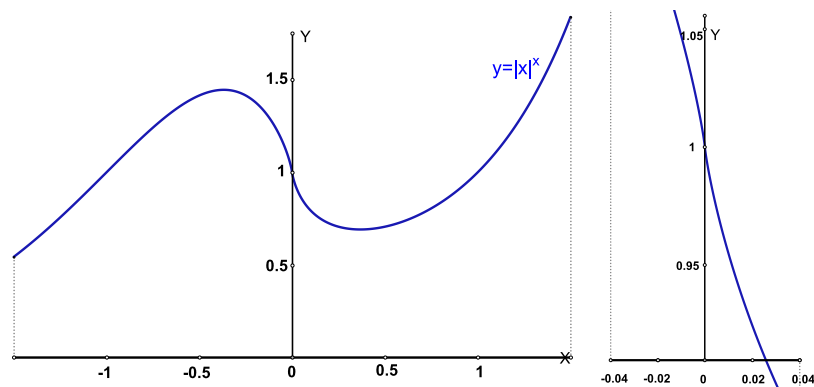
Primer 38: Computer delusion

Pomoću računara nacrtati grafik funkcije

$$f(x) = \begin{cases} |x|^x, & x \neq 0, \\ 1, & x = 0, \end{cases}$$

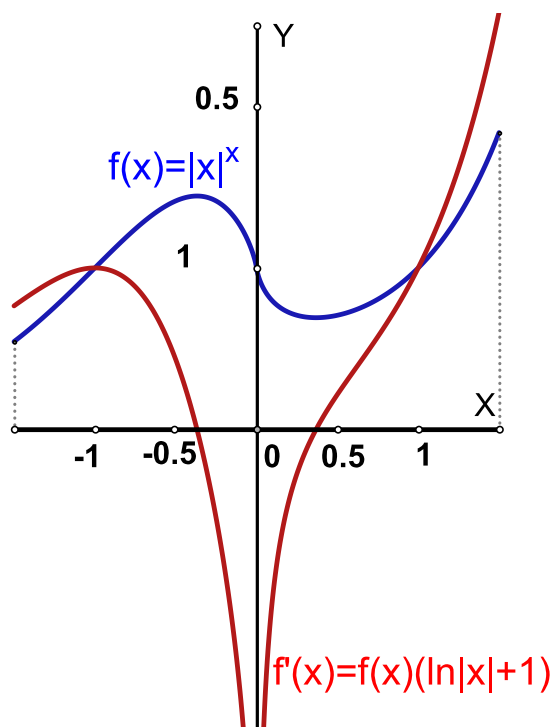
a zatim odrediti nagib tangente u tački $x = 0$.

Plot funkcije $f(x)$ i odgovarajući zum oko tačke $(0, 1)$ su prikazani na slici ispod. Sa grafika možemo proceniti vrednost nagiba, ali se ispostavlja da, što veći zum koristimo, nagib tangente izgleda sve veći

Slika 1.31: Grafik funkcije $y = f(x)$.

i veći. Prema tome, neophodno je izračunati prvi izvod funkcije $f(x)$ u $x = 0$ i on glasi $f'(x) = f(x)(\ln|x| + 1)$, pa nagib funkcije u datoj tački teži beskonačnosti. Međutim, zbog sporog rasta logaritamske funkcije, jako je teško odrediti da li divergira ili ne.

Poput praktičnog testa sume harmonijskog reda

Slika 1.32: Grafik funkcije $f(x)$ i $f'(x)$.

2 Dodatak: Zglobni i ostali važni mehanizmi

Jedan od osnovnih zadataka u oblasti mašinstva je dizajniranje i izrada mehanizama čiji je zadatak da obezbede da se pojedini elementi kreću na unapred zadat način, duž željene putanje. Ukoliko su dimenzije uređaja dovoljno male, taj zadatak je moguće rešiti jako precizno upotrebom mikrokontrolera. Međutim, ako se radi o većim uređajima, neophodno je koristiti klasične mašinske elemente, poluge, zupčanike, remenove, ležajeve itd. Zglobni mehanizmi (linkages) se sastoje od niza krutih štapova međusobno povezanih pokretnim zglobovima, od kojih neki mogu biti fiksirani. Jedan od najstarijih primera je Vatov (J. Watt 1784.) mehanizam, koji služi da transformiše pravolinijsko kretanje klipa parne mašine u kružno kretanje točkova. Premda je praktično upotrebljiv, on je daleko od teorijski savršenog pretvarača pravolinijskog u kružno kretanje.

Nakon Vata, mnogobrojni inženjeri, ali i matematičari su se posvetili rešavanju Vatovog i srodnih problema transformacije jednog tipa kretanja u drugi. Poznati ruski matematičar Čebišev je značajan deo svoje karijere posvetio zglobnim mehanizmima. Među brojnim mehanizmima koje je konstruisao, najpoznatiji su tzv. "lambda" mehanizam i na njemu zasnovana "plantigrade" mašina, koja realizuje mehanički hod i prikazana je na svetskoj izložbi u Parizu 1874. godine. Pravnik i matematičar amater Kempe (A. B. Kempe) se takođe intenzivno bavio zglobnim mehanizmima i 1877. god. objavio knjigu "How to Draw a Straight Line" posvećenu njima. On je publikovao poznatu teoremu koja tvrdi da je bilo koji ograničeni deo algebarske krive moguće generisati odgovarajućim zglobnim mehanizmom, koja je kasnije strogo dokazana. Veliki matematičar V. Thurston (W. Thurston) je parafrazirao teoremu rečima *One can design a linkage which will sign your name!* Naime, primenom Vajerštrasove teoreme o aproksimaciji, proizvoljnu ograničenu neprekidnu krivu možemo aproksimirati interpolacionim polinomom, a polinom zatim realizovati pomoću zglobnog mehanizma.

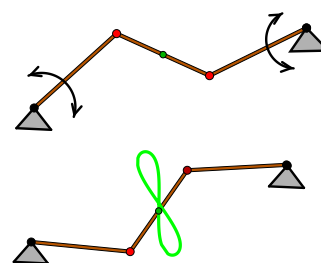
Zadatak o pretvaranju pravolinijskog u kružno kretanje je konačno rešio francuski inženjer Poselije (Peaucellier) 1867. god. i nezavisno od njega ruski matematičar Lipkin, pomoću mehanizma koji se danas naziva Poselije-Lipkinov ili "inverzor," zbog transformacije inverzije koju on zapravo realizuje. Pored inverzije, moguće je jednostavnim mehanizmima realizovati i osnu simetriju, translaciju, rotaciju (sabiranje i množenje uglova), sličnost ("pantograf") itd.

Interes savremenih inženjera za zglobne mehanizme je danas ponovo oživeo, prvenstveno zbog problema koji se javljaju u robotici.

Primer 39: Vatov zglobni mehanizam

Odrediti putanju centra srednjeg segmenta u jednostavnom Vatovom trosegmentnom mehanizmu sa fiksiranim krajevima.

Rešenje je prikazano na slici. Kruti štapovi su konstruisani kao radijusi kružnice sa fiksnim radijusom. Kriva se naziva Vatovom lemniskatom, i poznata, Bernulijeva lemniskata je specijalan slučaj Vatove. Kriva je algebarska i opisana je polinomom 6 stepena.

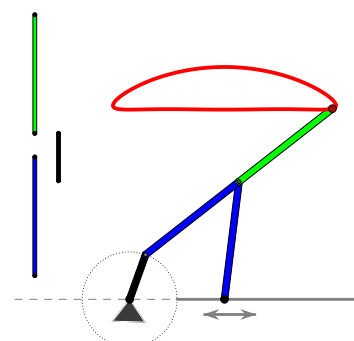
**Primer 40: Čebišovljev λ -mehanizam**

Na slici desno je prikazan λ mehanizam. Odrediti eksperimentalno odnose dužina segmenata (štapova) mehanizma tako da putanja tačke M ima oblik putanje stopala pri hodu.

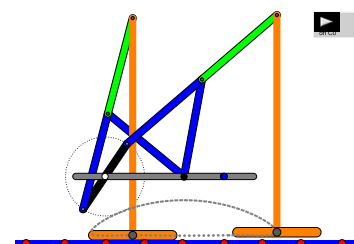
Rešenje je sledeće: ako dužinu crnog segmenta proglasimo za 1, tada je dužina plavih i zelenog segmenta 2.5, a rastojenje između fiksiranih tačaka je 2.

Primer 41: Čebišovljev (plantigrade) mehanizam

Spregom dva λ -mehanizma dizajnirati Čebišovljev hodajući (plantigrade) mehanizam.

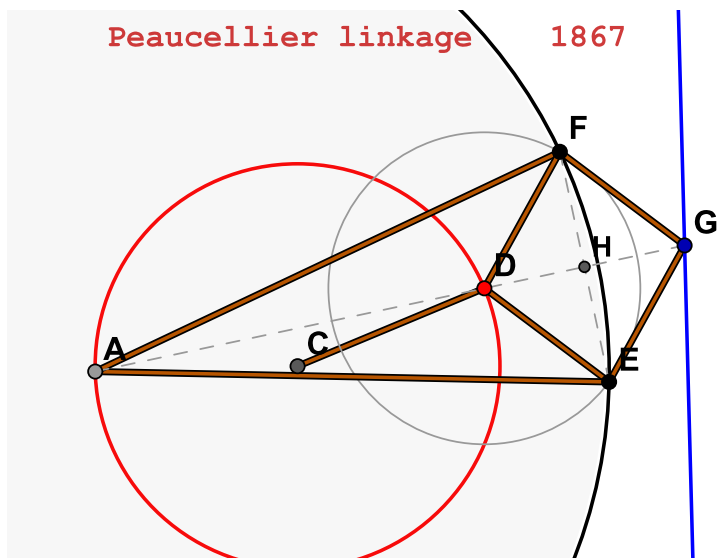


Slika 2.1: Naziv λ mehanizam je jasan sa slike.



Primer 42: Poselijeov mehanizam (inverzor)

Dizajnirati Poselijeov mehanizam i proveriti da li on zaista ima funkciju inverzora.



Poselijeova konstrukcija se sastoji od 7 segmenata, sa dužinama

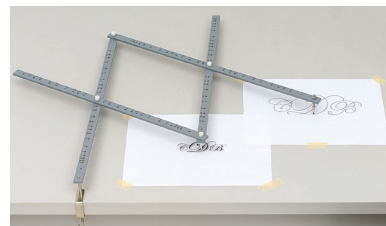
$|D, E| = |E, G| = |G, F| = |F, D| = a$, $|A, E| = |A, F| = b$ i $|C, D| = |A, C| = c$. Sa H je označen presek dijagonala romba $FDEG$. Za rastojanja $|A, D|$ i $|A, G|$ važi:

$$\begin{aligned} |A, D| \cdot |A, G| &= (|A, H| - |D, H|) \cdot (|A, H| + |D, H|) \\ &= (|A, H|^2 - |D, H|^2) \\ &= (b^2 - |H, E|^2 - |D, H|^2) \\ &= (b^2 - a^2) \\ &= \text{const} \end{aligned}$$

Prema tome, tačke D i G su povezane inverzijom u odnosu na tačku A (centar crne kružnice).

Primer 43: Pantograf

Dizajnirati pantograf i pokazati da on vrši transformaciju sličnosti.



Primer 44: Zglobni paralelogram

Teme O zglobnog paralelograma $OPMQ$ je fiksiran, dok strane OP i OQ rotiraju sa jednakim ugaonim brzinama u suprotnim smerovima. Odrediti putanju duž koje se kreće teme M .

Tačka M se kreće duž elipse. Neka je $|OP| = p$ i $|OQ| = q$. U nekom trenutku mora doći do poklapanja štapova OP i OQ ; taj pravac proglasimo za x -osu, a odgovarajući trenutak sa $t = 0$. Tačke P i Q se kreću duž kružnica

$$C_1 : (p \cos \omega t, p \sin \omega t); C_2 : (q \cos \omega t, -q \sin \omega t).$$

Trenutni položaj tačke $M(x, y)$ je

$$x = (p + q) \cos \omega t, y = (p - q) \sin \omega t.$$

Prema tome, koordinate tačke M zadovoljavaju jednačinu elipse

$$\frac{x^2}{(p + q)^2} = 1.$$

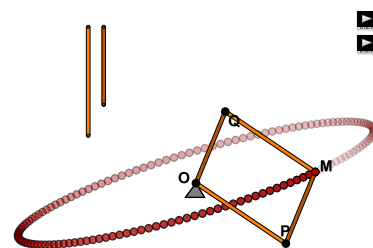
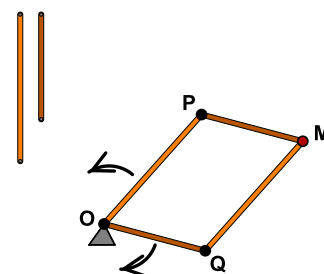
Primer 45: Zglobni paralelogram 2

Teme O zglobnog paralelograma $OPMQ$ je fiksiran, dok strane OP i OQ , za koje važi $|OP| = 2|OQ|$, rotiraju sa jednakim ugaonim brzinama ω i 2ω . Odrediti putanju duž koje se kreće teme M .

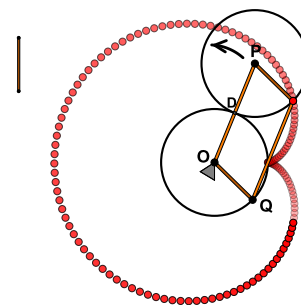
Odgovor je kardioida i konstrukcija je prikazana na slici desno. Na dodatnoj slici (dole), je prikazana pomoćna konstrukcija koja objašnjava zašto je rezultat kardioida. Naime, kardioida je kriva koju opisuje tačka na periferiji kružnice koja se kotrlja oko druge, fiksirane kružnice istog radijusa. Konstrukcija zahteva da se pomoću skripta fiksira odnos uglova duži OP i PM u odnosu na horizontalu: $M.\text{angle} = 180^\circ + 2 * b.\text{angle}$, , gde je sa b označena prava kojoj pripada duž OP .

Primer 46

Odrediti jednačinu kardioida kinematičkim razmatranjem prethodnog primera.



Slika 2.2: Klizačima u levom gornjem uglu podešavamo dužine stranica paralelograma.



Slika 2.3: Kotrljanje bez proklizavanja uslovljava da se tačka M kreće dvostruko većom ugaonom brzinom od tačke P .

2.1 Dokaz Kempeove teoreme

Nakon uspeha Poselijeja i Lipkina (1864) i Harta (1875) da pomoću mehaničkog uređaja konvertuju kružno u pravolinijsko kretanje, već 1876. godine, Kempe prezentuje opšti metod za dizajniranje zglobnog mehanizma koji može da iscrta proizvoljnu algebarsku krivu stepena n .

Parafrazirajući W. Thurstona, teorema tvrdi da Kempeov mehanizam može da iscrta nečiji popis, naravno, ako je modeliran neprekidnom krivom $f(x, y) = 0$. Krajnja konstrukcija je veoma kompleksna ("the complexity of the linkage is a necessary consequence of the perfect generality of demonstration." Kempe), pa je više od teorijske nego praktične važnosti, ali su osnovni elementi konstrukcije veoma jednostavni: paralelogram i anti-paralelogram. Konkretno, osnovni elementi Kempeovog mehanizma su sabirač uglova, multiplikator uglova i translator.

Primer 1: Kempeov translator

Zadatak translatora je da dati segment OB translira na datu lokaciju npr. u link TS . Konstruisati translator upotrebom paralelograma.

Jednostavna konstrukcija sa dva paralelograma je prikazana na slici. Više nadovezanih paralelograma povećavaju "fleksibilnost" mehanizma.

Primer 2: Kempeov reverzor

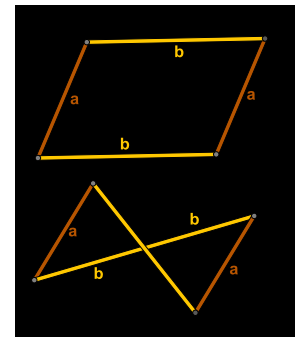
Zadatak udvajanje datog ugla, tj. za dati ugao generiše njemu podudaran sa zajedničkim temenom i krakom, ali suprotno orjentisan. Konstruisati reverzor upotrebom anti-paralelograma.

Konstrukcija sa dva anti-paralelograma je prikazana na slici. Položaj tačke D na duži BC je određen tako da anti-paralelogrami $OEDB$ i $OBCA$ budu slični, tj. $|BD| = |A, C|^2 / |B, C|$.

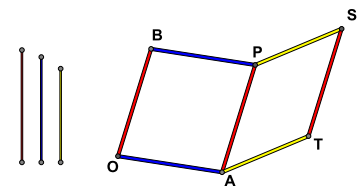
Primer 3: Kempeov multiplikator uglova

Zadatak multiplikatora je na osnovu datog ugla θ konstruiše ugao β sa zajedničkim temenom i jednim krakom, tako da je $\beta = n\theta$, $n \in \mathbb{N}$.

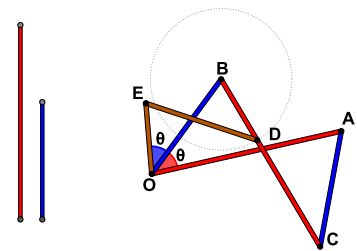
Rešenje je generalizacija reverzora. Prostim nadovezivanjem anti-paralelograma povećava se broj nadovezanih uglova θ , i na taj način ostvaruje multiplikacija. Rešenje za slučaj $n = 3$, je prikazano na



Slika 2.4: Paralelogram i anti-paralelogram.



Slika 2.5: Kempeov translator.



Slika 2.6: Kempeov reverzor.

slici desno. Interesantno je napomenuti, da se isti uređaj može koristiti i kao trisektor. Postoji nekoliko poznatih trisektora realizovanih pomoću zglobnih mehanizama. Jedan od najpoznatijih je Laisanov (Laisant 1875). On se sastoji od dva, duž dijagonala spregnuta paralelograma.

Kako je $\angle A'OB$ istovremeno polovina ugla $\angle AOB$ i $\angle A'OB'$, jasno je da je ujedno i trećina ugla $\angle AOB'$.

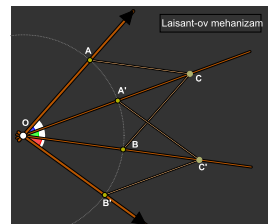
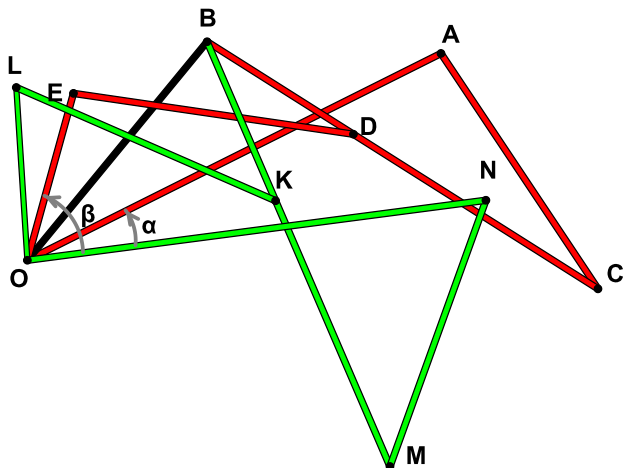
Druga verzija, slična Laisanovoj, je Silvesterov trisektor poznat kao "lepeza". Naizmenični nizovi jednakokrakih trouglova obezbeđuju da parovi uglova budu jednaki, tako da su tri para uglova međusobno jednaka. Obično je ugrađivan u optičke instrumente kod kojih je bilo nužno da se obezbedi pomeranje prizmi pod jednakim međusobnim uglovima.

Primer 4: Kempeov sumator uglova

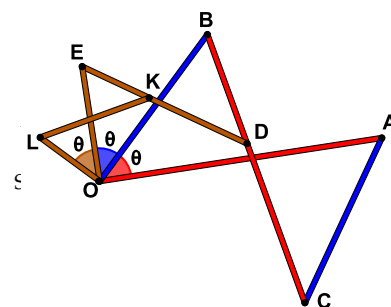
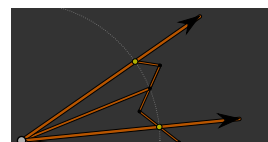
Zadatak sumatora je da za dva data ugla sa zajedničkim temenom i jednim krakom, konstruiše treći ugao, sa zajedničkim temenom i krakom, jednakim njihovom zbiru. U konstrukciji upotrebiti dva reverzora.

Sabirač je konstruisan kombinovanjem dva reverzora. Princip rada je prikazan na slici desno.

Pomoću jednog reverzora je obezbeđena jednakost uglova $\angle XOM$ i $\angle MOX'$, a pomoću drugog, $\angle YOM$ i $\angle MOY'$. Sama konstrukcija, nastala sprežanjem dva reverzora, je prikazano na slici 2.11 ispod.

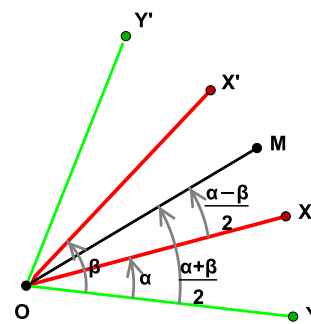


Slika 2.7: Laisanov trisektor.



Slika 2.9: Kempeov multiplikator.

Slika 2.11: Kempeov sumator



Slika 2.10: Princip rada sumatora

Dokaz Kempeove teoreme.

Teorema tvrdi da za polinom $P(x, y)$ stepena n , u konačnom delu ravni (tj. u preseku sa diskom nekog radijusa), postoji planarni

zglobni mehanizam koji opisuje krivu $P(x, y) = 0$. Polazna tačka je početni paralelogram koji definiše tačku Q . Neka je tačka Q nula polinoma $P(x, y)$. Koordinate tačke Q izražene preko parametara početnog paralelograma su:

$$x = l_1 \cos \phi + l_2 \cos \theta; y = l_1 \sin \phi + l_2 \sin \theta.$$

Polinom P je definisan kao zbir monoma

$$P(x, y) = \sum_{0 \leq i+j \leq n} a_{i,j} x^i y^j.$$

Smenom promenljivih polinoma $(x, y) \mapsto (\phi, \theta)$ dobijamo polinom koji zavisi od trigonometrijskih funkcija od uglova (ϕ, θ) . Upotrebom trigonometrijskih identiteta

$$\sin \alpha = \cos \alpha - \pi/2,$$

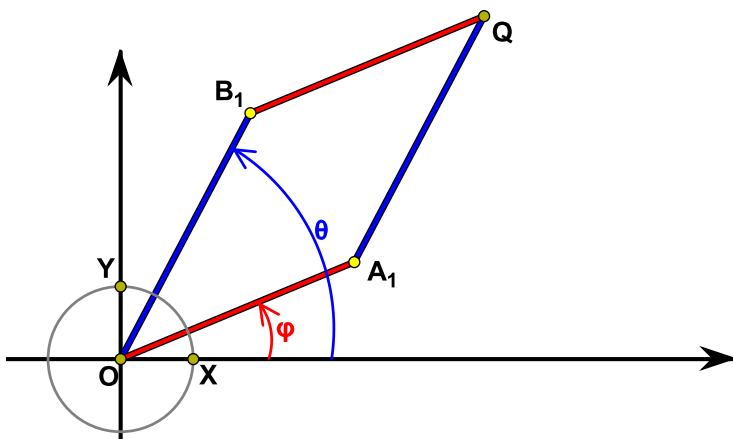
$$\cos^n \alpha = 1/2^n \sum_{k=0}^n \binom{n}{k} \cos((n-2k)\alpha),$$

$$\cos \alpha \cos \beta = 1/2(\cos(\alpha + \beta) + \cos(\alpha - \beta)),$$

može se pokazati da važi

$$P(x, y) = c + \sum_{\substack{0 \leq s \leq n, \\ -n \leq t \leq n, \\ (s,t) \neq (0,0)}} (a_{s,t} \cos(s\phi + t\theta) + b_{s,t} \cos(s\phi + t\theta - \pi/2)),$$

gde su $a_{s,t}, b_{s,t}, c \in \mathbb{R}$. Funkcija čije nule treba iscrtati je linearna kombinacija kosinusa od linearne kombinacije uglova. Koeficijenti linearne kombinacije zavise samo od parametara l_1, l_2 . Zadatak je konstruisati zglobni mehanizam tako da, pomeranjem slobodnog terminalnog zgloba npr. S , duž prave, zglob Q opisuje nul-skup datog polinoma.



Slika 2.12: Inicijalni Kempeov paralelogram.

Konstrukcija ce izvodi u koracima.

- Neka je definisan koordinatni početak O , jedinična tačka duž x -ose X , i neka je dat inicijalni paralelogram OA_1QB_1 , tako da su $\angle XOA_1 = \phi$ i $\angle XOB_1 = \theta$.
- Pomoću jednostavnog (npr. trougaonog) mehanizma, konstruišimo statički ram tako da dobijemo segment OY koji definiše jediničnu tačku na y -osi Y .
- Za sve cele brojeve s i $t \leq n$ konstruišemo multiplikator uglova na segmente OA_1 i OB_1 , i tako dobijemo segmente OA_s , $s = 2, \dots, n$ i OB_t , $t = -n, \dots, -1, 2, \dots, n$, tako da zadovoljavaju uslove $\angle XOA_s = s\phi$ i $\angle XOB_t = t\theta$.
- Pomoću sumatora uglova, konstruišemo nad A_s i B_t segmente $OC_{s,t}$, za koje važi $\angle XOC_{s,t} = \angle XOA_s + \angle XOB_t = s\phi + t\theta$.
- Na isti način, upotrebom sumatora nad segmentima OY i $OC_{s,t}$, konstruišemo segmente $OD_{s,t}$ tako da važi $\angle XOD_{s,t} = \angle XOC_{s,t} + \angle XOY = s\phi + t\theta - \pi/2$.
- Za svaki koeficijent $a_{s,t}$ i $b_{s,t}$ zakačimo segment $OE_{s,t}$ ili $OF_{s,t}$, koji odgovaraju skaliranim segmentima $OC_{s,t}$ i $OD_{s,t}$, tako da važi $|OE_{s,t}| = a_{s,t}$, $\angle XOE_{s,t} = s\phi + t\theta$ i $|OF_{s,t}| = b_{s,t}$, $\angle XOF_{s,t} = s\phi + t\theta - \pi/2$.
- Pomoću sumatora uglova konstruišemo niz suma. Transliramo $OE_{0,-n}$ u $E_{0,-(n-1)}$ tako da dobijemo $K_{0,-(n-1)}^E$ kao krajnju tačku, zatim $OK_{0,-(n-1)}^E$ u $OK_{0,-(n-2)}^E$ da dobijemo $K_{0,-(n-2)}^E$ itd. sve do $K_{n,n}^E$. Na isti način sabiramo segmente $F_{s,t}$ da dobijemo niz tačaka $K_{s,t}^F$ sve do $K_{n,n}^F \equiv S$ koju proglasimo za slobodan terminalni zglob S .

Pomoću ove konstrukcije dobijamo tačke $E_{s,t}$ i $F_{s,t}$ čije su x -koordinate jednake $a_{s,t} \cos(s\phi + t\theta)$ i $b_{s,t} \cos(s\phi + t\theta - \pi/2)$. Prema tome, x -koordinata tačke S je

$$x = \sum_{\substack{0 \leq s \leq n, \\ -n \leq t \leq n, \\ (s,t) \neq (0,0)}} (a_{s,t} \cos(s\phi + t\theta) + b_{s,t} \cos(s\phi + t\theta - \pi/2)) = f(x, y) - c$$

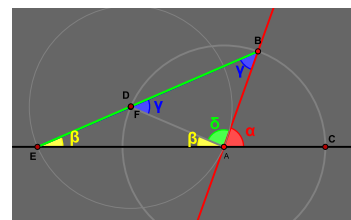
Kada se tačka (zglob) Q kreće duž nul-skupa $f(Q) = f(x, y) = 0$, tada se tačka S kreće duž krive $x = f(x, y) - c = -c$, tj. kreće se duž prave $x + c = 0$. Kretanje zgloba S duž prave $x + c = 0$ se lako postiže pomoću Poselijeovog mehanizma. Treba još napomenuti da su dimenzije diska unutar kojeg se vrši iscrtavanje, određene dužinama stranica l_1 i l_2 , početnog paralelograma.

Lenjir i šestar takođe spadaju u mehanizme. Njih su regularno inženjeri starog veka u Mesopotamiji, Egiptu itd. U matematici su postali standardni uređaji u Staroj Grčkoj, posredstvom zadataka posvećenim konstrukcijama. Iz nekog razloga, oni su za dozvoljavali upotrebu samo lenjira i šestara, premda je značajan broj mehanizama pored navedenih, bio poznat u starom veku. Neke od konstrukcija nisu uspeli da izvedu samo lenjirom i šestarom, od kojih su najpoznatiji trisekcija ugla, kvadratura kruga i udvajanje kocke. U 19. veku je dokazano da su one zapravo nemoguće. To nije ograničavalo pojedince da kostrukcije obave upotrebom dodatnih sredstava.

Poznata je Arhimedova trisekcija ugla, za koju su potrebni samo lenjir i šestar, ali je neophodno da lenjir ima (bar dve) oznake na sebi, što nije dozvoljeno. Njegova konstrukcija je specijalan slučaj *neuzis* konstrukcija, u kojima se koristi lenjir sa dve oznake koji je moguće rotirati oko neke svoje tačke (npr. tačke fiksirane iglom za podlogu).

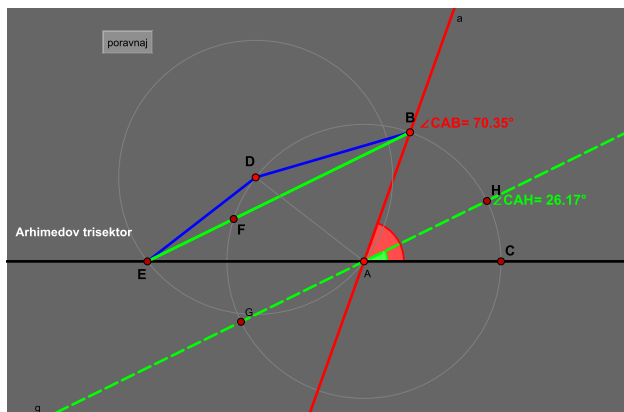
Arhimedov trisektor.

Nad horizontalnom linijom EC , dat je krak AB , tako da obrazuje ugao $\angle CAB$ koji treba podeliti na tri dela. Tada se jedna oznaka lenjira centrira na A , lenjir orjentiše duž kraka AB , tako da druga se druga oznaka poklopi sa B (to je zapravo definicija tačke B). Oko tačke A se opiše kružnica sa radijusom AB . Zatim se lenjir pomeri tako da jedna oznaka bude na horizontalnoj liniji (tačka E), a druga na kružnici (tačka D). Lenjir se dalje pomera, tako da su tačke E na horizontali, D na kružnici, sve dok tačka B ne bude na pravcu ED (tj. na lenjiru). Tada je ugao $\angle AED = 1/3\angle CAB$.



Slika 2.13: Dokaz valjanosti procedure trisekcije

Konstrukcija je prikazana na slici ispod. Potrebno je pomerati tačku D duž kružnice sve dok se ne poklopi sa tačkom F . Radi očiglednosti, nacrtana je i prava GH kao paralela prave EB koja prolazi kroz tačku A .



Slika 2.14: Arhimedov trisektor.

2.2 Mehanizmi i krive konstantnog dijametra

Primer 1: Krive konstantnog dijametra

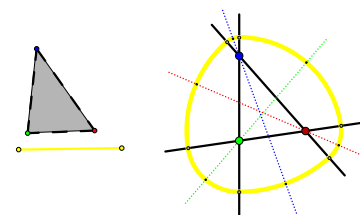
Dijametar zatvorene krive ("oval") je rastojanje između paralelnih pravih koje dodiruju (ili tangiraju ako je tangenta definisana) oval. Kružnica je primer ovala koji ima konstantan dijametar (nezavisan od pravca paralelnih pravih). Da li postoje ovali različiti od kružnice koji imaju konstantan dijametar? Konstruisati jednu takvu familiju krivih baziranih na trouglu.

Konstrukcija zasnovana na trouglu je prikazana na slici. Lako je proveriti da su dijometri jednaki jer su jednaki zbiru poluprečnika dve kružnice sa centrom u istom temenu.

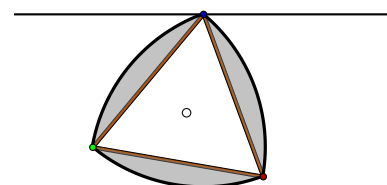
Krive konstantnog dijametra imaju zanimljivo kinematičko svojstvo: mogu se koristiti umesto točka pri kotrljanju između dve paralelne (horizontalne) prave. Najpoznatiji primer je tzv. Reloov (Reuleaux) trougao: jednakostranični trougao čije su stranice delovi kružnog luka.

Primer 2: Reloovo kotrljanje

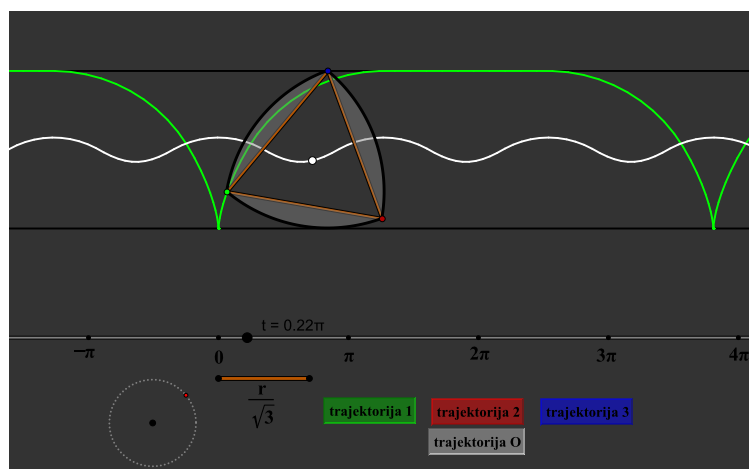
Konstruisati animaciju Reloovog trougla koji se kotrlja između dve paralelne prave. Omogućiti crtanje trajektorija temena i centra trougla pri rotaciji.



Slika 2.15: Kriva konstantnog dijametra bazirana na trouglu

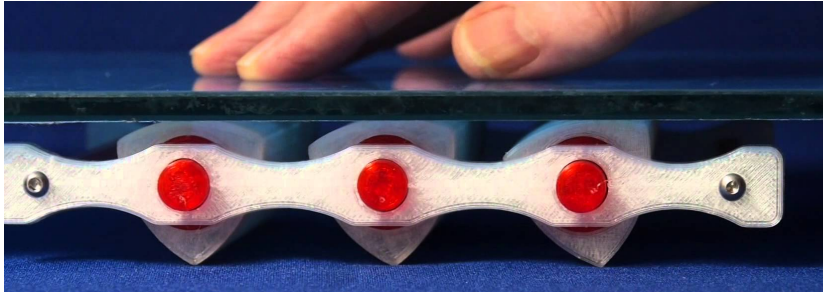


Slika 2.16: Reloov trougao.



Slika 2.17: Reloov trougao - animacija.

Ideja konstrukcije je da se pomoću scripta slobodne tačke jednakostraničnog trougla (jedno teme i centar) nateraju na kretanje po



Slika 2.18: Prikaz vozila sa Reloovim točkovima.

unapred (parametarski) zadatim krivama (trajektorije koje se sastoje iz lukova cikloide, kružnice i prave). Detalje je lako pročitati iz scripta. Skriptu za konstrukciju je sledeći:

```

r1=|A,B|;
r=r1*sqrt(3);
tt=N.angle;
G.x=6*tt/pi;
//tt=(pi*G.x)/6;
drawtext(G+(.7,.3),"t = "+tt/pi+"\pi$",size->16);
l=r1/r;

//tacke
x0=[0,0];
x1=[r*pi/3-r*sqrt(3)/2,r/2];
x2=[r*pi/3,r];
x3=[2*r*pi/3,r];
x4=[2*r*pi/3+r*sqrt(3)/2,r/2];
x5=[r*pi,0];

a0=[r*sqrt(3)/2,r/2];
a1=[r*pi/3,0];
a2=[r*pi/3,0];
a3=[2*r*pi/3-r*sqrt(3)/2,r/2];
a4=[2*r*pi/3,r];
a5=[r*pi,r];

b0=[0,r];
b1=[r*pi/3,r];
b2=[r*pi/3+r*sqrt(3)/2,r/2];
b3=[2*r*pi/3,0];
b4=[2*r*pi/3,0];
b5=[r*pi-r*sqrt(3)/2,r/2];

```

```

//drawall([x0,x1,x2,x3,x4,x5]);
//drawall([a0,a1,a2,a3,a4,a5],color->[1,0,0]);
//drawall([b0,b1,b2,b3,b4,b5],color->[0,0,1]);
draw(join(x0,x5),color->[0,0,0],size->2);
draw(join(x2,x3),color->[0,0,0],size->2);

// parametrizacija
n(t):=floor(3*t/pi);
md(t):=floor(t/(2*pi));
rez(t):=t-2*pi*md(t);
//////////
g(t):=r*(t-sin(t),1-cos(t));
f(t):=r*(cos(t),-sin(t));
//////////
//izdeli fazu na komade od po 2*pi/3; zatim podesi koord. pocetak
// i fazu za svaki korak. Faza je odredjena trenutnom fazom (korak)
// i polozajem tacke koja se krece
x(t):=if(mod(n(t),6)==0,[md(t)*r*pi,0]+g(rez(t)),
        if(mod(n(t),6)==1,[md(t)*r*pi,0]+a1+f(rez(t)-7*pi/6),
        if(mod(n(t),6)==2,[md(t)*r*pi,0]+[x2_1+r*(rez(t)-2*pi/3),x2_2],
        if(mod(n(t),6)==3,[md(t)*r*pi,0]+b3+f(rez(t)-3*pi/2),
        if(mod(n(t),6)==4,[md(t)*r*pi,0]+x5+g(rez(t)-5*pi/3),
        if(mod(n(t),6)==5,[md(t)*r*pi,0]+x5))))));

a(t):=if(mod(n(t),6)==0,[md(t)*r*pi,0]+a1+g(rez(t)+5*pi/3)-(2*r*pi,0),
        if(mod(n(t),6)==1,[md(t)*r*pi,0]+a1,
        if(mod(n(t),6)==2,[md(t)*r*pi,0]+a2+g(rez(t)-2*pi/3),
        if(mod(n(t),6)==3,[md(t)*r*pi,0]+b3+f(rez(t)-11*pi/6),
        if(mod(n(t),6)==4,[md(t)*r*pi,0]+[a4_1+r*(rez(t)-4*pi/3),a4_2],
        if(mod(n(t),6)==5,[md(t)*r*pi,0]+x5+f(rez(t)-pi/6))))));

b(t):=if(mod(n(t),6)==0,[md(t)*r*pi,0]+b0+[r*rez(t),0],
        if(mod(n(t),6)==1,[md(t)*r*pi,0]+a1+f(rez(t)+3*pi/2-pi/3),
        if(mod(n(t),6)==2,[md(t)*r*pi,0]+b3+g(rez(t)-pi),
        if(mod(n(t),6)==3,[md(t)*r*pi,0]+b3,
        if(mod(n(t),6)==4,[md(t)*r*pi,0]+b3+g(rez(t)-4*pi/3),
        if(mod(n(t),6)==5,[md(t)*r*pi,0]+x5+f(rez(t)-3*pi/6))))));

c(t):=(a(t)+b(t)+x(t))/3;
C.xy=c(tt);
D.xy=x(tt);

//plotovanje krivih
linesize(2);
If(Text9.pressed,plot(x(t),t,start->-8*pi,stop->8*pi,step->400,color->(0,1,0)));

```

```
If(Text10.pressed,plot(a(t),t,start->-8*pi,stop->8*pi,step->400,color->(1,0,0)));
If(Text11.pressed,plot(b(t),t,start->-8*pi,stop->8*pi,step->400,color->(0,0,1)));
If(Text12.pressed,plot(c(t),t,start->-8*pi,stop->8*pi,step->400,color->(1,1,1)));
```

Oko krive sa konstantnim dijametrom je moguće opisati kvadrat koji je moguće slobodno rotirati oko krive tj. krivu je moguće slobodno rotirati u kvadratu opisanom oko nje.

Primer 3: Reloov rotator

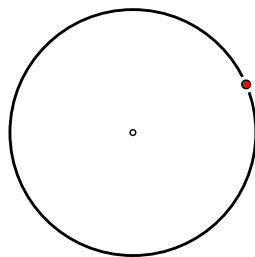
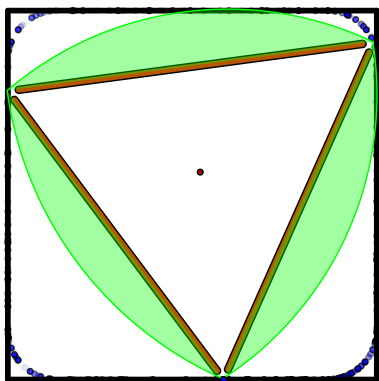
Konstruisati animaciju Reloovog trougla koji se kotrlja u opisanom kvadratu. Omogućiti crtanje trajektorija temena i centra trougla pri rotaciji.

Ideja konstrukcije je da se za odgovarajuće vrednosti parametra po dve od tri slobodne tačke pomeraju duž stranica kvadrata (stranice a), a treća je definisana u preseku kružnica poluprečnika a sa centrom u njima. Skript za konstrukciju:

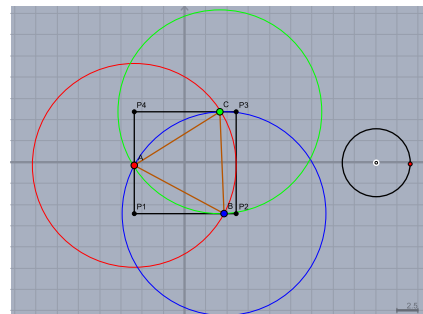
```
f(t,U,V):=(1-t)*U.xy+t*V.xy;
s=(sqrt(3)-1)/2;v=1-(sqrt(3)/2);
t=(E.angle)*6*s/pi;
if((t>=0)&(t<=s),A.xy=f(1/2+t,P1,P4);
B.xy=f(v+t,P2,P1);C.xy=Xab1.xy,
  if((t>=s)&(t<=2*s),B.xy=f(1/2+(t-s),P2,P1);
C.xy=f(v+(t-s),P3,P2);A.xy=Xbc.xy,
  if((t>2*s)&(t<=3*s),C.xy=f(1/2+(t-2*s),P3,P2);
A.xy=f(v+(t-2*s),P4,P3);B.xy=Xac.xy,
  if((t>3*s)&(t<=4*s),A.xy=f(1/2+(t-3*s),P4,P3);
B.xy=f(v+(t-3*s),P1,P4);C.xy=Xab1.xy,
  if((t>4*s)&(t<=5*s),B.xy=f(1/2+(t-4*s),P1,P4);
C.xy=f(v+(t-4*s),P2,P1);A.xy=Xbc.xy,
  if((t>5*s)&(t<=6*s),C.xy=f(1/2+(t-5*s),P2,P1);
A.xy=f(v+(t-5*s),P3,P2);B.xy=Xac.xy,
  if((t>6*s)&(t<=7*s),A.xy=f(1/2+(t-6*s),P3,P2);
B.xy=f(v+(t-6*s),P4,P3);C.xy=Xab1.xy,
  if((t>7*s)&(t<=8*s),B.xy=f(1/2+(t-7*s),P4,P3);
C.xy=f(v+(t-7*s),P1,P4);A.xy=Xbc.xy,
  if((t>8*s)&(t<=9*s),C.xy=f(1/2+(t-8*s),P1,P4);
A.xy=f(v+(t-8*s),P2,P1);B.xy=Xac.xy,
  if((t>9*s)&(t<=10*s),A.xy=f(1/2+(t-9*s),P2,P1);
B.xy=f(v+(t-9*s),P3,P2);C.xy=Xab1.xy,
  if((t>10*s)&(t<=11*s),B.xy=f(1/2+(t-10*s),P3,P2);
C.xy=f(v+(t-10*s),P4,P3);A.xy=Xbc.xy,
  if((t>11*s)&(t<=12*s),C.xy=f(1/2+(t-11*s),P4,P3);
A.xy=f(v+(t-11*s),P1,P4);B.xy=Xac.xy))))))));
```



Slika 2.19: Reloov trougao u kvadratu.



Slika 2.20: Reloov trougao u kvadratu.



Slika 2.21: Reloov trougao u kvadratu - prilikom konstrukcije.

Tu osobinu je moguće iskoristiti za bušenje (približno)kvadratnih otvora.

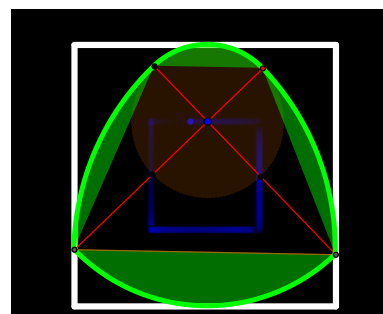
Međutim, postoji ingeniozni način za bušenje idealno kvadratnog otvora, koji takođe koristi krivu konstantnog dijametra. Konstrukcija je anonimno objavljena 1939. godine u časopisu Mechanical World i predstavlja varijaciju na standardni Reloov trougao.

Konstrukcija je prikazana na slici, a odgovarajući skript je:

```
//draw slot
N1=[-1,1-sqrt(2)];N2=[1,1-sqrt(2)];
N3=[1,3-sqrt(2)];N4=[-1,3-sqrt(2)];
drawall([[N1,N2],[N2,N3],[N3,N4],[N4,N1]],color->[1,1,1],size->5);
trans(t):=if((t>=0)&(t<=pi/4),[1-cos(t)-sin(t),0],
            if((t>=pi/4)&(t<=pi/2),[1-sqrt(2),cos(t)+sin(t)-sqrt(2)],
            if((t>=pi/2)&(t<=3*pi/4),[1-sqrt(2),cos(t)-sin(t)+2-sqrt(2)],
            if((t>=3*pi/4)&(t<=pi),[cos(t)-sin(t)+1,2-2*sqrt(2)],
            if((t>=pi)&(t<=5*pi/4),[-cos(t)-sin(t)-1,2-2*sqrt(2)],
            if((t>=5*pi/4)&(t<=3*pi/2),[sqrt(2)-1,cos(t)+sin(t)+2-sqrt(2)],
            if((t>=3*pi/2)&(t<=7*pi/4),[sqrt(2)-1,cos(t)-sin(t)-sqrt(2)],
            if((t>=7*pi/4)&(t<=2*pi),[cos(t)-sin(t)-1,0]))))));

t=X.angle;
if(Text0.pressed,B.xy=trans(t)+[0,1]+([[cos(t),-sin(t)],[sin(t),cos(t)]]*([0,-1]+b0)),B.xy=b0);
if(Text0.pressed,A.xy=a0+trans(t),A.xy=a0);

//initialization slot
a0=[0,1];b0=[sqrt(2)-1,sqrt(2)];X.y=V.y;
```



Slika 2.22: Kriva konstantnog dijametra za bušenje idealnog kvadrata.

Krive koje poseduju svojstvo analogno krivama konstantnog dijametra, da mogu da rotiraju unutar trougla se nazivaju Δ krivama. Najjednostavniji primer je Δ -dvougao.

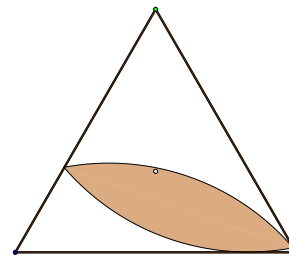
Konstrukcija Δ dvougla je jednostavna. Na pravou d koja je paralelna stranici CD jednakostraničnog trougla BCD , izaberemo tačku E . Konstruišemo kružnicu c_1 sa centrom u E i poluprečnikom jednakim visini trougla. On seče stranice BC i BD u tačkama G i H . Druga kružnica c_2 se dobija osnom refleksijom prve (c_1) u odnosu na pravu GH . Pomeranjem tačke E duž prave d , dvougao se "kotrlja" duž stranice CD .

Animacija same rotacije je dobijena crtanjem po jednog dvougla za svaku stranicu trougla. Kretanja vodećih tačaka (E i njoj analogne) je definisano parametrom. Na kraju je kontrolom vidljivosti pojedinih elemenata u zavisnosti od parametra postignut konačni efekat. Skript:

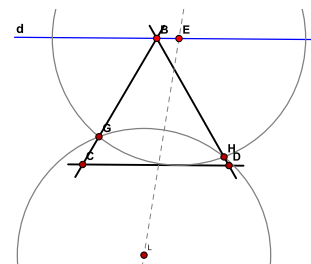
```
alpha=(B.angle/(2*pi/3));
alpha1=(B.angle-(2*pi/3))/(4*pi/3);
L.xy=if((B.angle>=0)&(B.angle<=2*pi/3),alpha*T.xy+(1-alpha)*S.xy,
        alpha1*S.xy+(1-alpha1)*T.xy);
if((B.angle>=0)&(B.angle<=2*pi/3),C5.visible=true;C6.visible=true,
    C5.visible=false;C6.visible=false);

beta=((B.angle-(2*pi/3))/(2*pi/3));
beta1=(B.angle+(2*pi/3))/(4*pi/3);
W.xy=if((B.angle>2*pi/3)&(B.angle<=4*pi/3),beta*U.xy+(1-beta)*V.xy,
        beta1*V.xy+(1-beta1)*U.xy);
if((B.angle>2*pi/3)&(B.angle<=4*pi/3),C10.visible=true;C11.visible=true,
    C10.visible=false;C11.visible=false);

gamma=(B.angle-(4*pi/3))/(2*pi/3);
gamma1=(B.angle)/(4*pi/3);
P5.xy=if((B.angle>4*pi/3)&(B.angle<=2*pi),gamma*P3.xy+(1-gamma)*P4.xy,
        gamma1*P4.xy+(1-gamma1)*P3.xy);
if((B.angle>4*pi/3)&(B.angle<2*pi),C15.visible=true;C16.visible=true,
    C15.visible=false;C16.visible=false);
```



Slika 2.23: Delta dvougao.

Slika 2.24: Konstrukcija Δ dvougla.

3 Bibliografija

Deo I

- [1] Steven E. Koonin, Dawn C. Meredith, *Computational Physics Fortran Version*, Westview Press Published, 1990.
- [2] Erich W. Schmid, Gerghard Spitz, Wolfgang Losch, *Theoretical Physics on the Personal Computer*, Springer-Verlag, 1988.
- [3] Robert L. Zimmerman, Frederick I. Olness, *MATHEMATICA FOR PHYSICS, 2nd EDITION*, Addison-Wesley Publishing Company, Inc, 2002.
- [4] Paolo Giannozzi, Furio Ercolessi, Stefano de Gironcoli, *Numerical Methods in Quantum Mechanics*, Lecture notes, University of Udine, 2014.
- [5] Ivanka Milošević, *Vektorski prostori i elementi vektorske analize*, Fizički fakultet Univerzitet u Beogradu, 1997.
- [6] Tatjana Vuković, Saša Dmitrović, *OSNOVI MATEMATIČKE FIZIKE*, Fizički fakultet Univerzitet u Beogradu, 2016.
- [7] Edib Dobardžić, *KVANTNA FIZIKA*, Fizički fakultet Univerzitet u Beogradu, 2011.
- [8] Patric T. Tam, *A Physicist's Guide to Mathematica*, 2nd EDITION, Department of Physics and Astronomy, Humbolt State University Arcata, California, Academic Press is an imprint of Elsevier, 2008.
- [9] PD Roman Schmied, *Using Mathematica for Quantum Mechanics A Student's Manual*, Univesity of Basel, Switzerland, 2019.
- [10] Dilip Datta, *LATEX in 24 Hours A Practical Guide for Scientific Writing*, Springer Nature, 2017.
- [11] Benjamin A. Stickler, Ewald Schachinger, *Basic Concepts in Computational Physics, Second Edition*, Springer International Publishing AG Switzerland, 2016.

- [12] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes in C The art of Scientific Computing, Second Edition*, Cambridge University Press, 1997.

Deo II

- [13] Jurgen Richter-Gebert, Ulrich H. Kortenkamp, *The Cinderella.2 Manual*, Springer-Verlag Berlin Heidelberg, 2012.
- [14] Victor Gutenmacher, N.B. Vasilyev, *Lines and Curves: A Practical Geometry Handbook, 3rd Edition*, Birkhauser Boston, Inc., 2004.
- [15] Mark Levi, *The Mathematical Mechanic*, Princeton University Press, 2009.
- [16] Mark Levi, *Classical Mechanics with Calculus of Variations and Optimal Control*, American Mathematical Society, 2014.
- [17] Tom M. Apostol, Mamikon A. Mnatsakanian, *New Horizons in Geometry*, American Mathematical Society, 2013.
- [18] Ralph H. Abraham, Christopher D. Shaw, *Dynamics: The Geometry of Behavior, 2th Edition*, Addison-Wesley Publishing Company, 1992.
- [19] Gengzhe Chang, Thomas W. Sederberg, *Over and Over Again*, The Mathematical Association of America, 1997.
- [20] David Acheson, *From Calculus to Chaos: An Introduction to Dynamics*, Oxford University Press, 1997.
- [21] N. Ya. Vilenkin, *Method of Successive Approximations*, Mir Publishers, 1979.
- [22] Harold Abelson, Andrea A. diSessa, *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*, The MIT Press, 1981.
- [23] Denny Gulick, Jon Scott, eds. *The Beauty of Fractals: Six Different Views*, The Mathematical Association of America, 2010.