MANUALIA POLYTECHNICI STUDIORUM ZAGRABIENSIS

BOJAN KOVAČIĆ

MATEMATIČKI ALATI U ELEKTROTEHNICI

TVZ

TEHNIČKO VELEUČILIŠTE U ZAGREBU
POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

Nakladnik

Tehničko veleučilište u Zagrebu

Elektrotehnički odjel

Autor

mr.sc. Bojan Kovačić

Recenzenti

prof.dr.sc. Juraj Šiftar
doc.dr.sc. Zoran Tomljanović

Objavljivanje je odobrilo Stručno vijeće Tehničkog veleučilišta u Zagrebu,
odlukom broj: 1447-8/13 od 25. svibnja 2013. godine.
Udžbenik

CIP zapis dostupan u računalnom katalogu
Nacionalne i sveučilišne knjižnice u Zagrebu pod brojem 846427

ISBN 978-953-7048-28-0

TEHNIČKO VELEUČILIŠTE U ZAGREBU
POLYTECHNICUM ZAGRABIENSE
ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

Bojan Kovačić

MATEMATIČKI ALATI U ELEKTROTEHNICI

udžbenik

Sadržaj

PREDGOVOR

Suvremena nastava visokoškolske matematike u moderno je vrijeme nezamisliva bez aktivne uporabe računala. Pritom se sve više napušta klasičan i zastarjeli pristup prema kojemu je računalo samo pomoćno sredstvo (poput džepnoga računara) u provjeri ispravnosti rješenja nekoga matematičkoga problema. Slobodno se može reći da se danas i matematika uči "na računalima" jer kvalitetni programski paketi poput *Mathematice*, MATLAB-a i drugi omogućavaju ne samo rješavanje postavljenih zadataka, nego i analizu rješenja, pa čak i stvaranje i rješavanje posve novih zadataka.

Udžbenik pred vama namijenjen je svim studentima, ali i ostalim korisnicima koji se u svojemu radu susreću s potrebom rješavanja određenih matematičkih problema u MATLAB-u. Prva četiri poglavlja sadrže osnove rada s MATLAB-om i osnove programiranja u njemu. U poglavljima 5. – 7. obrađena je primjena MATLAB-a u diferencijalnom i integralnom računu. U poglavljima 8. – 10. izlaže se primjena MATLAB-a na rješavanje problema vjerojatnosti i statistike, odnosno numeričke matematike. Na kraju svakoga poglavlja nalaze se zadaci za vježbu namijenjeni samostalnomu radu. Preporuča se da svaki čitatelj sam riješi što više tih zadataka. Njihova težina je najviše jednaka težini izloženih i potpuno riješenih primjera, pa student koji je uspješno svladao izloženo gradivo ne bi trebao imati nikakvih poteškoća riješiti ih.

Napominjem da je udžbenik pisan uz pretpostavku rada u MATLAB-u verzije 7.0. Vjerujem da će se čitatelji lako snaći primjenjujući ga i na neku drugu (raniju ili noviju) verziju toga programa.

Za nesmetano i kvalitetno praćenje tijeka izlaganja nužno je poznavanje osnovnih pojmova matematičke analize i linearne algebre, a poželjno je (ali nije nužno) poznavanje i osnovnih načela programiranja. Radi izbjegavanja povećanja opsežnosti osnovnoga teksta, u ovom udžbeniku se detaljno ne razmatraju matematički pojmovi nužni za praćenje teksta, već se spominju samo okvirno. Međutim, u tom smislu ovaj udžbenik potpuno slijedi nastavne planove predmeta *Matematika* 1, *Matematika* 2, *Vjerojatnost i statistika* i *Numerička matematika* koji se predaju na prvoj i drugoj godini stručnoga studija elektrotehnike Tehničkoga veleučilišta u Zagrebu. Stoga se svo matematičko gradivo potrebno za praćenje tijeka izlaganja može pronaći u skriptama [6] i [9], odnosno udžbenicima [7], [8] i [10] navedenima u popisu korištene literature. Za produbljivanje znanja toga gradiva preporučujemo udžbenike [1] i [2].

Izvorni rukopis udžbenika vrlo su pomno i savjesno pročitali recenzenti prof.dr.sc. Juraj Šiftar s PMF-Matematičkoga odsjeka Sveučilišta u Zagrebu i doc.dr.sc. Zoran Tomljanović s Odjela za matematiku Sveučilišta u Osijeku, te iznijeli brojne konstruktivne primjedbe, kritike i prijedloge u svrhu poboljšanja kvalitete teksta. Nastavnici Tehničkoga veleučilišta u Zagrebu Luka Marohnić, Renata Opačić i Tihana Strmečki koristili su internu verziju udžbenika u

nastavnoj praksi, te iznijeli svoje vrlo korisne primjedbe i prijedloge na temelju vlastitih iskustava iz nastavne prakse. Bivši demonstrator iz matematičkih predmeta na stručnom studiju računarstva Veleučilišta u Splitu[1] Dražen Blanuša (danas viši savjetnik za razvoj računalnih programa u tvrtki NTH Group) sa zavidnim je poletom i entuzijazmom sudjelovao u nastanku prve verzije ovoga udžbenika (kao autoriziranih predavanja), te svojim primjedbama, prijedlozima i konkretnim zadatcima utjecao na koncept i sadržaj spomenute verzije. Studenti-slušači predmeta *Matematički alati u elektrotehnici* na Tehničkom veleučilištu u Zagrebu u ak.god. 2011/2012., te Andrijana Petrović i Martina Vučićević, demonstratorice iz matematičkih predmeta na stručnom studiju elektrotehnike istoga veleučilišta, svojim su pitanjima i komentarima izravno doprinijeli da pojedini dijelovi teksta budu jasniji i razumljiviji.

Dekanica Tehničkoga veleučilišta u Zagrebu prof.dr.sc. Slavica Ćosović-Bajić i pročelnik Elektrotehničkoga odjela Tehničkoga veleučilišta u Zagrebu prof.dr.sc. Krešimir Meštrović svojim su poticajima izravno utjecali na finaliziranje pisanja udžbenika i njegovu pojavu u javnosti.

Vrlo ugodna mi je dužnost iskreno zahvaliti svim spomenutim ljudima na pomoći i suradnji.

Svjestan sam da je, unatoč pozornosti prigodom višestrukih korektura teksta, izvjestan broj pogrješaka ipak „preživio“. Krivicu za te pogrješke potpuno preuzimam na sebe. Unaprijed iskreno zahvaljujem svima koji uoče propust bilo koje vrste, te iznesu dodatne konstruktivne kritike i prijedloge za poboljšanje kvalitete izloženoga teksta.

U Vukovaru, studenoga 2005. i Zagrebu, srpnja 2012.

Autor

[1] Veći dio udžbenika korišten je i u nastavi iz kolegija *Primijenjena i numerička matematika* na stručnim studijima elektrotehnike i računarstva Veleučilišta u Splitu (studiji u Vukovaru).

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

§1. UVOD U MATLAB

1.1.Što je MATLAB?

MATLAB (naziv je skraćenica engleskoga naziva ***MAT**rix **LAB**oratory*) istodobno je i programski paket namijenjen numeričkomu računanju i modeliranju, ali i viši programski jezik namijenjen raznim znanstvenim i tehničkim primjenama.

1.2. Aritmetika digitalnoga elektroničkoga računala

Jedan od glavnih problema koji se javlja prigodom zapisivanja realnih brojeva u računalu jest problem unosa "velikih" realnih brojeva ili realnih brojeva čiji se decimalni zapis sastoji od "velikoga" broja (možda i beskonačno mnogo) decimala. Naime, sklopovi naših računala su fizički ograničeni i u njih brojevi s tako velikim brojem decimala jednostavno ne mogu "stati". Stoga uvijek nastojimo što bolje iskoristiti raspoloživi prostor u računalu kako bi u nj mogli "stati" i brojevi s većim brojem znamenaka.

1.2.1. Znanstveni oblik realnoga broja

Ovaj se oblik često naziva i *eksponencijalni*, a u stručnoj literaturi rabi se i naziv *format pomične točke* (engl. *floating point format*). Osnova toga oblika jest sljedeći:

Poučak 1. Neka je $a \in \mathbf{R}$ broj s konačnim decimalnim zapisom[2]. Tada se a može zapisati u obliku:

$$a = \pm x.y_1 y_2 ... y_m \cdot 10^{\pm z_1 z_2 ... z_n}$$

pri čemu vrijedi:

1.) $x \in \{1, 2, 3, \ldots, 9\}$;
2.) $m, n \in \mathbf{N}$;
2.) $y_1, \ldots, y_m, z_1, \ldots, z_n \in \{0, 1, 2, \ldots, 9\}$. ■

Važna napomena: Zapisi $y_1 y_2 \ldots y_m$ i $z_1 z_2 \ldots z_n$ nisu skraćeni zapisi umnožaka $y_1 \bullet y_2 \bullet \ldots \bullet y_m$, odnosno $z_1 \bullet z_2 \bullet \ldots \bullet z_n$, nego označavaju nizove znamenki.

[2] *Konačan decimalni zapis* zapravo znači da se dotični zapis sastoji od konačno mnogo (dekadskih) znamenaka. Može se pokazati da je ova pretpostavka ekvivalentna pretpostavci da postoje cijeli brojevi b, k i l takvi da vrijedi $a = \frac{b}{2^k \cdot 5^l}$, pri čemu je navedeni razlomak potpuno skraćen.

Oblik iz Poučka 1. naziva se *znanstveni oblik realnoga broja*. On se sastoji od dva dijela. Broj $\pm x.y_1y_2\ldots y_m$ naziva se *mantisa*, a broj $\pm z_1z_2\ldots z_n$ *eksponent*. Uočimo odmah da je mantisa uvijek *realan* broj čija je apsolutna vrijednost (modul) barem jednak 1 (jer joj je prva znamenka barem jednaka 1), dok je eksponent uvijek *cijeli* broj. Veličina zapisa mantise i eksponenta unutar računala ovisi o realizaciji zapisa unutar računala.

Iole bolji džepni računar posjeduje zaslon na kojemu je (barem) šest mjesta predviđeno za zapis znamenaka mantise, a (točno) dva mjesta predviđena su za zapis znamenaka eksponenta. Stoga se na takvom džepnom računaru može zapisati i prikazati ukupno 180 milijuna brojeva iz segmenta $[-9.99999 \cdot 10^{99}, 9.99999 \cdot 10^{99}]$. Pritom je najmanji prikazivi pozitivan broj $1.00000 \cdot 10^{-99}$.

Postavlja se pitanje: kako bilo koji realan broj a zapisati u znanstvenomu obliku? Jedan od mogućih algoritama jest sljedeći:

Korak 1. Utvrditi je li prva znamenka x broja a barem jednaka 1. Ako jest, ići na Korak 2. Ako nije, ići na Korak 6.

Korak 2. Utvrditi je li iza prve znamenke decimalna točka. Ako jest, ići na Korak 3. Ako nije, ići na Korak 4.

Korak 3. Mantisa realnoga broja a jednaka je samome broju, a eksponent je jednak 0. Kraj postupka.

Korak 4. Postaviti početnu vrijednost eksponenta na 0. Ići na Korak 5.

Korak 5. Prebrojati koliko se ukupno znamenaka nalazi *ispred* decimalne točke. (Označimo taj broj sa b.) Tada je matrisa broj dobiven postavljanjem decimalne točke neposredno iza prve znamenke, a eksponent je jednak $b - 1$. Kraj postupka.

Korak 6. Prebrojati koliko se ukupno nula nalazi ispred prve znamenke broja a različite od nule. (Označimo taj broj sa b.) Tada je matrisa broj dobiven postavljanjem decimalne točke neposredno iza prve znamenke različite od nule, a eksponent je jednak $(-b)$. Kraj postupka.

Primjer 1. Zapišimo u znanstvenomu obliku realne brojeve 3.14159, –100 000 i 0.00025 pri čemu ćemo uzeti da mantisa ima ukupno točno 6 znamenaka (jednu ispred i pet iza decimalne točke), te odredimo mantisu i eksponent za svaki od dobivenih zapisa.

Gore navedeni algoritam provodimo za svaki broj zasebno:

3.14159 → prva znamenka je jednaka 3, pa prelazimo na Korak 2. → iza prve znamenke jest decimalna točka pa prelazimo na Korak 3. → mantisa je jednaka 3.14159, a eksponent 0 → $3.14159 = 3.14159 \cdot 10^0$. Kraj postupka.

–100 000 → prva znamenka je jednaka 1, pa prelazimo na Korak 2. → iza prve znamenke nije decimalna točka pa prelazimo na Korak 4. → ispred decimalne točke ima ukupno 6 znamenki pa je b = 6. Mantisa je jednaka –1.00000, a eksponent 6 – 1 = 5. Stoga je –100 000 = $-1.00000 \cdot 10^5$. Kraj postupka.

0.00025 → prva znamenka je 0, nije barem jednaka 1 pa prelazimo na Korak 6. → prva znamenka različita od nule jednaka je 2, a ispred nje se nalaze ukupno 4 nule, te je b = 4. Mantisa je jednaka 2.50000, a eksponent –4. Stoga je $0.00025 = 2.50000 \cdot 10^{-4}$.

Zbog obveze pisanja eksponenta u *superscriptu*, pri implementaciji u MATLAB-u nametnula se potreba za modifikacijom znanstvenoga prikaza kako bi i eksponent bio ispisan "običnim" tekstom. U tu je svrhu uporabljena sljedeća doskočica:

Pretpostavimo da je $a = \pm x.y_1 y_2 ... y_m \cdot 10^{\pm z_1 z_2 ... z_n}$ znanstveni oblik realnoga broja a. Tada je MATLAB-ov znanstveni oblik toga broja

$$a = \pm x.y_1 y_2 ... y_m e \pm z_1 z_2 ... z_n .$$

Dakle, iza mantise "dopišemo slovo" e (koje nema nikakve veze s oznakom baze prirodnoga logaritma e = 2.7182818…!), a eksponent "spustimo" iz *superscripta* u "obični" tekst. Pritom treba napomenuti da zapis eksponenta prema osnovnom dogovoru (*default*-u) sadrži točno tri znamenke (ako eksponent ima samo dvije znamenke (n = 2), zapisuje se u obliku $0z_1z_2$).

<u>Primjer 2.</u> MATLAB-ovi (znanstveni) oblici realnih brojeva iz Primjera 1. su redom:

3.14159e + 000,
-1.00000e + 005,
2.50000e – 004.

<u>*1.2.2 Zaokruživanje realnih brojeva i "rezanje" suviška decimala*</u>

Beskonačne periodične decimalne brojeve (kakvi su npr. decimalni zapisi razlomaka $\frac{1}{3}$, $\frac{5}{6}$, itd.) ne možemo "čitave" pohraniti u računalo u formatu pomične točke zbog fizičke ograničenosti njegovih sklopova. Zbog toga se dio tih brojeva pohranjuje, a dio "zanemaruje", odnosno izostavlja. Uzmimo npr. racionalan broj $\frac{11}{12}$ = 0.916666…. . Taj je broj beskonačan periodičan decimalni broj i ne možemo ga "čitavoga" pohraniti u računalo. Budući da MATLAB dozvoljava ukupno 6 znamenki u mantisi s dvostrukom preciznošću[3] (tj. s ukupno 15 značajnih znamenaka), zaokruživanje se obično radi u ovisnosti o parnosti posljednje znamenke. Alternativno, moguće je postupiti i ovako:

[3] O IEEE standardu i oblicima zapisa realnoga broja vidjeti u točki 1.2.3., str. 9.

1.) Prvih 5 znamenki mantise prepisati, a šestu ili prepisati (ako je manja od 5) ili uvećati za 1 (ako je barem jednaka 5).

2.) Prepisati svih 6 znamenaka mantise zanemarujući sve ostale znamenke u standardnomu prikazu.

Prvi način nazivamo *zaokruživanje* (*s točnošću na* 5 *decimala*), a drugi "*rezanje*" *suviška decimala*. Nameće se pitanje koji je od tih dvaju načina bolji. Ovo je jedan od osnovnih problema *numeričke matematike*. Ovdje ćemo njegovo rješenje dati rabeći promatrani primjer.

Označimo sa x broj dobiven zaokruživanjem broja $\frac{11}{12}$ na 5 decimala, a sa y broj dobiven "rezanjem" suviška decimala broja $\frac{11}{12}$. Mantisa broja $\frac{11}{12}$ jednaka je 9.166666666. Njezina je šesta znamenka upravo jednaka 6. Stoga je

$$x = 0.916667, \quad y = 0.916666.$$

Izračunajmo sada apsolutne vrijednosti razlika $\frac{11}{12} - x$ i $\frac{11}{12} - y$ kako bismo vidjeli koliko smo pogriješili u prvom, a koliko u drugom slučaju:

$$\left|\frac{11}{12} - x\right| = |-0.0000003333\ldots| = 3.33333 \cdot 10^{-7},$$

$$\left|\frac{11}{12} - y\right| = |0.0000006666\ldots| = 6.66666 \cdot 10^{-7}.$$

Prva je razlika manja od druge, što znači da smo u prvom slučaju napravili manju pogrješku. Stoga možemo zaključiti da je zaokruživanje bolja metoda od "rezanja" suviška decimala. Zbog toga se sva računala i iole bolji računari uvijek služe zaokruživanjem realnih brojeva.

1.2.3. IEEE *standard*

Vjerojatno vam je još iz srednje škole poznato da je osnova za prikazivanje svih realnih brojeva u računalu tzv. *binarni sustav*. Baza toga sustava jest prirodan broj 2, a znamenke koje ga tvore su 0 i 1. Može se pokazati da u tom slučaju vrijedi sljedeća varijanta Poučka 1.:

Poučak 2. Neka je $a \in \mathbf{R}$ broj s konačnim zapisom u binarnom brojevnom sustavu. Tada se a može zapisati u obliku:

$$a = \pm x.y_1 y_2 \ldots y_m \cdot 2^{\pm z_1 z_2 \ldots z_n}$$

pri čemu vrijedi:

1.) $x = 1$;
2.) $m, n \in \mathbf{N}$;
2.) $y_1, \ldots, y_m, z_1, \ldots, z_n \in \{0, 1\}$. ■

Stoga sva razmatranja koja smo provodili u podtočkama 1.2.1. i 1.2.2. vrijede i za brojeve zapisane u binarnomu sustavu. Budući da je ovaj sustav od iznimne važnosti u računarstvu, brojni znanstveni instituti predlagali su razne standarde za organizaciju podataka u digitalnome računalu. Jedan od opće prihvaćenih je tzv. IEEE *standard* kojega je predložio ugledni američki znanstveni institut **I***nstitute of* **E***lectrical and* **E***lectronic* ***E****ngineers*, a rabi se na gotovo svim osobnim računalima, u programskim paketima kao što su MATLAB, Mathematica itd. Prema tome standardu, realni se brojevi najčešće zapisuju u jednom od sljedeća tri oblika (ili, kako to inženjeri računarstva vole reći, *formata*):

1.) format jednostruke preciznosti (engl. *single precision format*);
2.) format dvostruke preciznosti (engl. *double precision format*);
3.) format četvrtostruke preciznosti (engl. *quadriple precision format*).

MATLAB koristi format dvostruke preciznosti. U njemu su za zapis realnoga broja predviđena točno 64 bita, i to:

1 bit za zapis predznaka mantise;
11 bitova za zapis predznaka eksponenta i samoga eksponenta;
52 bita za zapis mantise.

Budući da je prema Poučku 2. prva znamenka mantise uvijek jednaka 1, radi uštede memorijskoga prostora ta se znamenka ne pohranjuje. Zbog toga se u formatu dvostruke preciznosti mogu zapisati brojevi od reda veličine 10^{-307} do reda veličine 10^{307} čiju mantisu tvori ukupno šesnaest znamenaka.

U ovakvom sustavu definiraju se brojevi *Inf* (skraćenica od engl. *infinity* = beskonačno), *NaN* (skraćenica od engl. *Not a Number* = "nije broj") i *eps* ("strojni epsilon") s:

$$10^{400} \to Inf$$
$$\frac{0}{0} \to NaN$$
$$eps = 2^{-52} \approx 2.2204 \cdot 10^{-16}.$$

Od tih triju brojeva mi ćemo najčešće rabiti konstantu *eps*. Ona je, zapravo, najmanji pozitivan broj x takav da u aritmetici računala vrijedi $1 + x > 1$, a može se interpretirati i kao ocjena pogrješke pri zaokruživanju u binarnoj *floating point* aritmetici dvostruke preciznosti. U detalje ovdje nećemo ulaziti.

1.3. Zapis osnovnih matematičkih i logičkih operatora u MATLAB-u

Radi preglednosti, te zapise navodimo u sljedećoj tablici.

Naziv operatora	*Zapis*
zbrajanje	+
oduzimanje	–
množenje	*
(matrično) lijevo dijeljenje	\
dijeljenje	/
potenciranje	^
"…je manje od…"	<
" … je manje ili jednako…"	<=
"… je veće od …"	>
"… je veće ili jednako …"	>=
"… je jednako …"	==
"… je različito od …"	~=
"…i…"	&
"…ili…"	|
"ne…"	~

Važna napomena: Jednostruki operator = u MATLAB-u se rabi prigodom inicijaliziranja vrijednosti neke varijable, dok se dvostruki operator == rabi ako se želi provjeriti istinitost logičke tvrdnje. To ćemo detaljnije obraditi u Primjeru 4.

Primjer 3. Izračunajmo vrijednost brojevnoga izraza

$$\left\{\left[\left(\frac{2}{3}-\frac{1}{2}\right)^{2}\cdot 18+\frac{1}{3}\right]^{-1}-0.2\right\}^{2012}-1.$$

U komandnomu prozoru (engl. *Command Window*) iza znaka >> smjestio se trepćući pokazivač. Rabeći tipkovnicu utipkavamo redom (bez razmaka!):

```
(((2/3-1/2)^2*18+1/3)^(-1)-0.2)^2012-1
```

Obratite pozornost na redoslijed izvedbe računskih operacija (^ je "najjača", potom slijede * i /, a "najslabije" su + i –). Uočite i da vam tijekom utipkavanja brojeva i operatora MATLAB daje do znanja koja otvorena i koja zatvorena zagrada tvore par, pa u svakomu trenutku možete vidjeti jesu li sve zagrade zatvorene ili nisu. Također, negativne brojeve (zbog predznaka) obvezatno pišite unutar okruglih zagrada radi preglednosti izraza.

Nakon što smo utipkali gornji niz brojeva i operatora, pritisnemo točno jednu od dviju tipki *Enter* na svojoj tipkovnici. Pojavljuje se rezultat:

```
ans =
    8.895106873296754e-013
```

i to je tražena vrijednost zadanoga brojevnoga izraza. Ako bismo istu vrijednost željeli dobiti "klasičnim" putem (pomoću olovke i praznoga lista papira), utvrdili bismo da je ona jednaka 0. Što se dogodilo i zbog čega nam MATLAB nije ispisao vrijednost 0?

Do pogrješke je došlo uslijed zaokruživanja. Naime, već smo istaknuli da MATLAB ne zna baratati s razlomcima pa ih pretvara u decimalne brojeve (rabeći aritmetiku dvostruke preciznosti) i računa s njima. Pri toj pretvorbi nužno dolazi do grešaka – u ovome slučaju one su reda veličine 10^{-16}. Provedete li i u MATLAB-u računanje "korak po korak", vidjet ćete da se rezultati podudaraju s onima "ručno" izračunatima sve do potenciranja s 2012. 2012 je relativno velika potencija pa "malu" pogrješku pretvara u znatno veću. Drugim riječima, naša pogrješka ne bi bila tako izražena da je umjesto 2012 upisan neki bitno manji eksponent (npr. 2).

Zapamtimo sljedeće važno:

<u>*PRAVILO*: Kada završimo unos naredbe u tekućemu redu komandnoga prozora, za njezino izvršenje i prelazak u novi red pritisnemo jednu od tipki *Enter* na svojoj tipkovnici.</u>

<u>Primjer</u> **<u>4.</u>** Utipkajmo u novomu retku komandnoga prozora

$$x\ =\ 5$$

MATLAB će ispisati:

```
x =
     5
```

Ovime smo istodobno i deklarirali varijablu x i definirali da je vrijednost te varijable jednaka 5. Utipkajmo sada u novomu retku komandnoga prozora (opet bez razmaka):

```
x==100
```

MATLAB će ispisati:

```
ans =
     0
```

Otkuda ovaj rezultat? U drugome smo slučaju zahtijevali provjeru istinitosti jednakosti

$$x = 100.$$

Budući da smo ranije definirali da je vrijednost varijable x jednaka 5, jednakost $x = 100$ očito nije točna, pa je njezina istinitost jednaka 0 (*false*). Otuda slijedi navedeni rezultat.

1.4. Posebne varijable u MATLAB-u

Pored već spomenutih *Inf*, *NaN* i *eps*, u MATLAB-u postoje još tri posebne, unaprijed deklarirane varijable:

ans (skraćenica od engl. *answer* = odgovor) – automatski poprima vrijednost nekoga izraza ako izraz nije pridružen toj varijabli;
i – imaginarna jedinica (za zaboravne: $i := \sqrt{-1}$);
pi – Ludolfov broj (3.14159265358979…).

Napomena: MATLAB dozvoljava da se ove posebne varijable „pregaze", tj. da im se dodijele posve nove vrijednosti. Npr. utipkamo li

```
i=5
```

i pritisnemo *Enter*, varijabla *i* više neće biti imaginarna jedinica, nego varijabla čija je vrijednost jednaka 5.

Primjer 5. Utipkajmo u novomu retku komandnoga prozora

```
pi
```

pa ćemo (nakon što pritisnemo *Enter* i podesimo odgovarajući format) dobiti:

```
ans =
   3.14159265358979
```

Primjer 6. Utipkajmo u novomu retku komandnoga prozora:

```
i
```

Ponovno pritisnimo tipku *Enter* i dobit ćemo:

```
ans =
              0 + 1.00000000000000i
```

Primjer 7. U MATLAB-u nisu implementirane sve nama poznate realne konstante. Utipkajmo u novomu retku komandnoga prozora

```
e
```

i ponovno pritisnimo tipku *Enter*. Na naše iznenađenje, umjesto broja 2.7182818… MATLAB ispisuje:

```
??? Undefined function or variable 'e'.
```

što znači: *Nedefinirana funkcija ili varijabla* '*e*'. Nasreću, u MATLAB-u imamo ugrađenu funkciju **exp** (vidjeti točku 1.5.) pomoću koje računamo bilo koju potenciju broja *e*, pa nam definiranje broja *e* kao realne konstante nije nužno.

1.5. Osnovne matematičke funkcije ugrađene u MATLAB

Radi lakšega snalaženja i preglednosti, osnovne matematičke funkcije ugrađene u MATLAB navodimo tablično zajedno sa sintaksom poziva funkcije.

Naziv funkcije	*Značenje funkcije*	*Poziv funkcije*
sin	sinus	sin(*argument*)
cos	kosinus	cos(*argument*)
tan	tangens	tan(*argument*)
cot	kotangens	cot(*argument*)
asin	arkus sinus	asin(*argument*)
acos	arkus kosinus	acos(*argument*)
atan	arkus tangens	atan(*argument*)
acot	arkus kotangens	acot(*argument*)
sinh	hiperbolni sinus	sinh(*argument*)
cosh	hiperbolni kosinus	cosh(*argument*)
tanh	hiperbolni tangens	tanh(*argument*)
coth	hiperbolni kotangens	coth(*argument*)
asinh	area sinus hiperbolni	asinh(*argument*)
acosh	area kosinus hiperbolni	acosh(*argument*)
atanh	area tangens hiperbolni	atanh(*argument*)
acoth	area kotangens hiperbolni	acoth(*argument*)
pow2	potencija broja 2	pow2(*argument*)
sqrt	kvadratni korijen	sqrt(*argument*)
exp	potencija broja e (e^x)	exp(*argument*)
log	prirodni logaritam ($\ln x$)	log(*argument*)
log2	logaritam s bazom 2 ($\log_2 x$)	log2(*argument*)
log10	dekadski logaritam ($\log x$)	log10(*argument*)
abs	apsolutna vrijednost (modul) kompleksnoga broja	abs(*argument*)
real	realni dio kompleksnoga broja	real(*argument*)
imag	imaginarni dio kompleksnoga broja	imag(*argument*)
conj	kompleksno konjugiranje	conj(*argument*)
round	zaokruživanje broja na najbliži cijeli broj	round(*argument*)
ceil	zaokruživanje broja *x* na najmanji cijeli broj jednak ili veći od *x*	ceil(*argument*)
floor	zaokruživanje broja *x* na najveći cijeli broj jednak ili manji od *x*	floor(*argument*)

***Važne napomene*:** **1.)** Za računanje trećega, četvrtoga, …, *n* – toga korijena iz realnoga broja ne postoji "gotova" matematička funkcija ugrađena u MATLAB. Zbog toga se ti korijeni računaju kao potencije pomoću megapoznate formule pretvorbe korijena u potenciju:

$$\sqrt[n]{a} = a^{\frac{1}{n}}.$$

2.) Ne zaboravite da funkcija *exp* računa isključivo potenciju broja $e = 2.7181828\ldots$, a ne bilo koju opću potenciju. Za računanje potencije oblika a^x koristi se zapis `a^x`.

Primjer 8. Izračunajmo vrijednosti funkcija *ceil*, *floor* i *round* za $x = -0.5$, pa usporedimo dobivene rezultate.

U novom retku komandnoga prozora utipkajmo:

```
[ceil(-0.5) floor(-0.5) round(-0.5)]
```

pa će MATLAB ispisati:

```
ans =
      0    -1    -1
```

Dakle, u prvom je slučaju broj zaokružen na najmanji cijeli broj jednak ili veći od –0.5, a to je 0. U drugom i trećem slučaju rezultati su isti: cijeli broj najbliži broju –0.5 jednak je najmanjem cijelom broju jednakom ili manjem od –0.5, a to je –1.

Primjer 9. Izračunajmo vrijednost sljedećega brojevnoga izraza:

$$\left\{\left\lfloor \arcsin\left(\sin\left(\frac{2011\cdot\pi}{2}\right)+\cos\left(\frac{2013\cdot\pi}{2}\right)-\mathrm{tg}\left(\frac{2011\cdot\pi}{4}\right)\cdot\mathrm{ctg}\left(\frac{2013\cdot\pi}{4}\right)\right)+\arccos\left(\sqrt[7]{2187}-2\ln\sqrt[4]{e^3}-\log\sqrt{10}\right)-\mathrm{arctg}\left(\left|\overline{\frac{1}{2}-\frac{\sqrt{3}}{2}i}\right|-\mathrm{Re}(i^{2012})+\mathrm{Im}(i^{2013})\right)\cdot\frac{4}{\pi}\right\rceil^2+3\right\}^{-\frac{1}{2}}+\left\lfloor-\frac{\pi}{4}\right\rceil+0.5.$$

Rabeći tablice iz točaka 1.4 i 1.6 utipkavamo sljedeći niz brojeva i operatora (u jednom retku, bez razmaka):

```
((asin(sin(2011*pi/2)+cos(2013*pi/2)-tan(2011*pi/4)*cot(2013*pi/4))+acos(2187^(1/7)-
2*log(exp(3/4))-log10(sqrt(10)))-atan(abs(conj(1/2-sqrt(3)/2*i))-
real(i^2012)+imag(i^2013))*4/pi)^2+3)^(-1/2)+round(-pi/4)+0.5
```

Dobijemo:

```
ans =
2.264854970235319e-014.
```

Izračunamo li pravu vrijednost ovoga izraza, dobit ćemo da je ona jednaka 0. Pogrješka (reda veličine 10^{-14}) se pojavila zbog približnih izračuna iracionalnih brojeva oblika $\frac{k\cdot\pi}{n}$, $k \in$ $\in \{2011, 2013\}$, $n \in \{2, 4\}$, no, praktično se ona može zanemariti.

TEHNIČKO VELEUČILIŠTE U ZAGREBU
POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

1.6. Zadatci za vježbu

1. Zapišite u znanstvenomu obliku sljedeće realne brojeve:

a) 2012.2012; **b)** 0.002012; **c)** –2.002012; **d)** –0.20112012.

U svakomu od tih zapisa odredite mantisu i eksponent, te pripadni zapis u MATLAB-u.

2. Zapišite u standardnom obliku sljedeće realne brojeve zapisane u MATLAB-ovu znanstvenu obliku:

a) 2.012e + 001; **b)** –2.012e + 004; **c)** 2.012e – 002; **d)** –2.012e – 003.

U svakomu od navedenih zapisa odredite mantisu i eksponent.

3. Koristeći elementarne matematičke operacije ugrađene u MATLAB izračunajte vrijednosti sljedećih brojevnih izraza:

a) $\sin\left(\frac{3\cdot\pi}{4}\right)-\cos(2\cdot\pi)+\mathrm{tg}\left(\frac{2\cdot\pi}{3}\right)$;

b) $\mathrm{ctg}\,\frac{7\cdot\pi}{3}$;

c) $\arcsin 0.2 - \arccos\frac{1}{4} + \mathrm{arctg}\sqrt{2}$;

d) $\mathrm{sh}\ 1.25 + \mathrm{ch}\frac{\sqrt{2}}{2} - \mathrm{Arsh}\frac{1}{\sqrt{5}} + \mathrm{Arth}\frac{\sqrt{\pi}}{\pi}$;

e) $e^{\sqrt{\ln\frac{1}{\pi}}}$;

f) $\log[\ln(0.5^{-0.1234})]$;

g) $|1-i| + \mathrm{Re}(2^{0.5} + \sqrt[3]{3}\cdot i) - \mathrm{Im}\sqrt[5]{\overline{2-i}}$;

h) $\left[\overline{\frac{2+i}{1+2\cdot i}}-\frac{\mathrm{Re}\left(\overline{\frac{3-5\cdot i}{\overline{1+i}}}\right)}{\mathrm{Im}\left(\frac{3+5\cdot i}{1-i}\right)}\right]^3$.

4. Zaokružite sljedeće realne brojeve na njima najbliži cijeli broj:

a) $2\cdot\sqrt[3]{2+\sqrt{3}}$; **b)** $3^{1+\sqrt{2}-\mathrm{tg}\frac{\pi}{8}}$; **c)** $\dfrac{\ln\left(\arcsin\frac{1}{\sqrt{5}}\right)}{\left|\frac{1}{5}-\frac{\sqrt{3}}{5}\cdot i\right|}$.

5. Pomoću MATLAB-a odredite vrijednosti sljedećih varijabli:

a) $x = \log(-2)$; **b)** $y = \arcsin(2012)$; **c)** $z = \sqrt{-1.5}$?

Ima li smisla svaki od ispisanih rezultata? Objasnite zašto!

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

§2. OSNOVE MATRIČNOGA RAČUNA U MATLAB-U

2.1. Zadavanje (generiranje) matrica u MATLAB-u

Matrice u MATLAB-u zadajemo prema sljedećoj sintaksi:

ime matrice = [*elementi matrice*].

Ime matrice obično je veliko tiskano slovo abecede: *A*, *B*, *C*, … . *Članovi* ili *elementi matrice* su (obično) realni brojevi. Upisujemo ih **po retcima** tako da između svaka dva člana bude po jedan razmak ili zarez. Kraj retka naznačujemo ili pritiskom tipke *Enter* ili oznakom `;` iza koje bez razmaka upisujemo prvi element sljedećega retka. Pokažimo to na primjeru.

***Primjer* 1.** Zadajmo u MATLAB-u realnu matricu *A* tipa (5, 4) definiranu s

$$A=\begin{bmatrix} -1 & \frac{1}{2} & \frac{\sqrt{3}}{2} & -0.25 \\ 0 & \frac{\cos 2}{\operatorname{tg} 5} & -\frac{\ln 3}{\log 5} & \arcsin 0.4 \\ \sqrt[3]{2} & -e^{-2} & -\sqrt{2} & 1-\frac{\sqrt[4]{3}}{2} \\ -\sin 6 & \operatorname{ctg}(21) & \frac{1}{\operatorname{arctg}(3)} & -\operatorname{Arsh}(2) \\ \frac{1}{\operatorname{Arch}(1.5)} & \operatorname{Arth}(\log 2) & \ln\left(\frac{1+e}{3-e}\right)^{-1} & \frac{\sqrt{2}}{\sqrt[3]{3}} \end{bmatrix}.$$

U tekući redak komandnoga prozora upisujemo:

```
A=[-1  1/2  sqrt(3)/2  -0.25;0  cos(2)/tan(5)  -log(3)/log10(5)  asin(0.4);2^(1/3)
-exp(-2)  -sqrt(2)  1-3^(1/4)/2;  -sin(6)  cot(21)  1/atan(3)  -asinh(2);1/acosh(1.5)
atanh(log10(2)) log((3-exp(1))/(1+exp(1))) sqrt(2)/3^(1/3)]
```

Detaljno proučite i analizirajte kako je zadan svaki pojedini član! Dobijemo:

```
A =
  -1.00000000000000   0.50000000000000   0.86602540378444  -0.25000000000000
                  0   0.12310160900874  -1.57175884780882   0.41151684606749
   1.25992104989487  -0.13533528323661  -1.41421356237310   0.34196299352375
   0.27941549819893  -0.65466511548606   0.80061117222304  -1.44363547517881
   1.03904346061751   0.31065185251892  -2.58010978698585   0.98056091781096
```

***Napomena*:** Ako završetak retka naznačavamo pritiskom tipke *Enter*, onda sve gore navedene funkcije, operatori i znamenke ne moraju biti zapisani u istom retku komandnoga prozora.

***Važno upozorenje*:** Matricu nije moguće zadati navodeći članove po stupcima!

2.2. Algebarske operacije s matricama

Osnovne algebarske operacije s matricama zapisuju se potpuno jednako kao i algebarske operacije s brojevima. No, u radu s matricama rabe se još neke algebarske operacije koje pregledno prikazujemo u sljedećoj tablici.

Operacija	*Znak*
"Desno dijeljenje" matrica	/
"Lijevo dijeljenje" matrica	\
Transponiranje matrice	'
Množenje elemenata dviju matrica član po član	.*
Potenciranje matrice	^
Potenciranje elemenata dviju matrica član po član	.^
"Desno dijeljenje" elemenata dviju matrica član po član	./
"Lijevo dijeljenje" elemenata dviju matrica član po član	.\

Operacije "desnoga" i "lijevoga dijeljenja" rabe se prigodom rješavanja matričnih jednadžbi. Prva se koristi za rješavanje jednadžbe oblika $X \cdot A = B$, a druga za rješavanje jednadžbe oblika $A \cdot X = B$. Ilustrirajmo to primjerima.

Primjer 2. Riješimo matričnu jednadžbu $X \cdot A = B$ ako je:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \ B = \begin{bmatrix} 5 & 6 \end{bmatrix}.$$

"Običnim" bismo računom dobili:

$$X = B \cdot A^{-1}.$$

No, u MATLAB-u možemo postupiti brže. Utipkamo:

```
A=[1 2;3 4]
```

Pritisnemo *Enter*, pa potom utipkamo:

```
B=[5 6]
```

i opet pritisnemo *Enter*. Time smo definirali matrice A i B. Rješenje našega sustava dobivamo "desnim dijeljenjem" matrica B i A. Utipkamo:

```
X=B/A
```

i dobijemo:

```
X =
    -1      2
```

Dakle, rješenje polazne jednadžbe jest jednoretčana matrica X = [-1 2].

***Važna napomena*:** "Desno dijeljenje" matrica moguće je ako i samo ako matrice A i B imaju *jednak broj stupaca*!

Primjer 3. Riješimo matričnu jednadžbu $A \cdot X = B$ ako je:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix},\ B = \begin{bmatrix} 5 \\ 6 \end{bmatrix}..$$

"Običnim" bismo računom dobili:

$$X = A^{-1} \cdot B.$$

No, u MATLAB-u možemo postupiti brže. Redefinirajmo matricu B u novomu retku komandnoga prozora utipkavajući

```
B=[5;6]
```

nakon čega pritisnimo *Enter*. Rješenje polaznoga sustava dobiva se "lijevim dijeljenjem" matrica A i B. Utipkajmo:

```
X=A\B
```

i dobit ćemo:

```
X =
  -4.00000000000000
   4.50000000000000
```

Dakle, rješenje polazne matrične jednadžbe jest matrica $X = \begin{bmatrix} -4 \\ 4.5 \end{bmatrix} = \begin{bmatrix} -4 \\ \frac{9}{2} \end{bmatrix}$.

***Važna napomena*:** "Lijevo dijeljenje" matrica moguće je ako i samo ako matrice A i B imaju *jednak broj redaka*!

Transponiranje matrica obavlja se bez ikakvih teškoća kako pokazuje sljedeći primjer.

Primjer 4. Odredimo matricu X^T za matricu X iz prethodnoga primjera. U novi redak komandnoga prozora utipkajmo

```
X'
```

pa dobivamo:

```
ans =
  -4.00000000000000   4.50000000000000
```

Uočimo ovdje značenje varijable *ans*. Matrici X' nismo dali ime, već smo samo "rekli" MATLAB-u što želimo napraviti s matricom *X*. Budući da operacijom transponiranja dobivamo jedan novi objekt, MATLAB mu je dogovorno dodijelio ime *ans*.

Množenje i "dijeljenje" matrica član po član su operacije koje se u linearnoj algebri obično ne spominju prigodom obrade standardnoga matričnoga računa. U tomu se računu član po član izvode isključivo operacije zbrajanja i oduzimanja. No, ove su operacije daleko jednostavnije od klasičnih operacija množenja i "dijeljenja" matrica, a u to će nas uvjeriti i sljedeća definicija.

***Definicija* 1.** Neka su $A = [a_{ij}]$ i $B = [b_{ij}]$ matrice *istoga tipa* (m, n). Definiramo matrice $C = [c_{ij}]$, $D = [d_{ij}]$ i $E = [e_{ij}]$ s:

$$c_{ij} := a_{ij} \cdot b_{ij};$$

$$d_{ij} := \frac{a_{ij}}{b_{ij}};$$

$$e_{ij} := \frac{b_{ij}}{a_{ij}},$$

za svaki $i = 1, 2, \ldots, m$ i za svaki $j = 1, 2, \ldots, n$. Kraće pišemo:

$$C := A.*B$$
$$D := A./B$$
$$E := A.\backslash B = B./A$$

Operaciju .* nazivamo *množenje matrica član po član*, operaciju ./ *"desno dijeljenje" član po član*, a operaciju .\ *"lijevo dijeljenje" član po član*.

Uočimo da je desno dijeljenje matrica *A* i *B* istovjetno lijevom dijeljenju matrica *B* i *A* i obrnuto. Pokažimo primjenu ovih operacija na primjeru.

***Primjer* 5.** U novi redak komandnoga prozora utipkajmo:

```
A = [5 6]
```

Pritisnimo *Enter*, pa utipkajmo:

```
B = [10 12]
```

Potom ponovno pritisnimo *Enter*. U sljedećemu retku utipkajmo

```
C=A.*B,D=A./B,E=A.\B
```

pa će MATLAB ispisati:

```
C =
    50    72
D =
   0.50000000000000   0.50000000000000
E =
     2     2
```

Uočite formulu prema kojoj su izračunani elementi svake pojedine matrice:

$$50 = 5 \cdot 10 \qquad 72 = 6 \cdot 12$$

$$0.5 = 5 : 10 \qquad 0.5 = 6 : 12$$

$$2 = 10 : 5 \qquad 2 = 12 : 6.$$

Potenciranje matrica član po član poseban je slučaj množenja matrica član po član, a nastupa kada vrijedi jednakost $A = B$.

Primjer 6. Za matricu A iz prethodnoga primjera odredimo A.^2. Utipkajmo:

```
F=A.^2
```

pa će MATLAB ispisati:

```
F =
    25    36
```

Napomenimo da gore definirane operacije množenja i dijeljenja član po član imaju smisla ako bilo koju matricu zamijenimo nekim realnim brojem. Tada dobivamo "klasično" množenje (odnosno, "dijeljenje") matrice nekim skalarom.

Primjer 7. Za matricu B iz prethodnoga primjera odredimo matrice F :=7.*B, G := B./10 i H := 10.\B. U novi redak komandnoga prozora utipkajmo:

```
F=7.*B, G=B./10, H=10.\B
```

pa će MATLAB ispisati:

```
F =
    70    84
G =
   1.00000000000000   1.20000000000000
```

```
H =
   1.00000000000000   1.20000000000000
```

Uočimo odmah da se rezultati "lijevoga" i "desnoga dijeljenja" član po član u ovom slučaju podudaraju jer su oba matrična "dijeljenja" član po član zamijenjena "klasičnim" dijeljenjem.

Sve osnovne matematičke funkcije navedene u točki 1.5. imaju smisla i ako je njihov argument matrica *bilo kojega* tipa jer "djeluju" na svaki član matrice zasebno. Evo primjera.

Primjer 8. Izračunajmo sin(A) za matricu A definiranu s

$$A=\begin{bmatrix} \frac{2009\cdot\pi}{3} & -\frac{2011\cdot\pi}{2} & 2013\cdot\pi & -\frac{2015\cdot\pi}{6} \end{bmatrix}.$$

U novi redak komandnoga prozora utipkajmo:

```
A=[2009*pi/3 -2011*pi/2 2013*pi -2015*pi/6]
```

Pritisnimo *Enter*, pa potom u novi redak utipkajmo:

```
sin(A)
```

MATLAB će ispisati:

```
ans =
    -0.86602540378453  -1.00000000000000   0.00000000000015  -0.49999999999999
```

Točna vrijednost izraza sin(A) jest $\begin{bmatrix} -\frac{\sqrt{3}}{2} & -1 & 0 & -\frac{1}{2} \end{bmatrix}$.

Primjer 9. Izračunajmo $e^{\sqrt{A}}$ za matricu A definiranu s

$$A=\begin{bmatrix} 0 & 1 & 4 \end{bmatrix}.$$

U nova dva retka komandnoga prozora utipakmo:

```
A=[0 1 4];
exp(sqrt(A))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
   1.00000000000000   2.71828182845905   7.38905609893065
```

Točna vrijednost izraza $e^{\sqrt{A}}$ jest $\begin{bmatrix} 1 & e & e^2 \end{bmatrix}$.

2.3. Matrične funkcije ugrađene u MATLAB

Pregledno ih navodimo u sljedećoj tablici.

Ime funkcije	*Sintaksa*	*Argument(i) funkcije*	*Namjena funkcije*
zeros	zeros(m, n)	oba argumenta su prirodni brojevi	Generira nulmatricu s m redaka i n stupaca.
eye	eye(n)	prirodan broj	Generira jediničnu matricu reda n.
diag	diag(A)	jednoretčana ili jednostupčana matrica	Generira dijagonalnu matricu čiji su elementi redom elementi matrice A.
rank	rank(A)	bilo koja matrica	Računa rang matrice A.
size	size(A, n)	prvi argument je bilo koja matrica, a drugi točno jedan element skupa {1, 2}	size(A,1) ispisuje broj redaka, a size(A,2) broj stupaca matrice A.
sum	sum(A)	bilo koja matrica	Računa zbroj članova u stupcima matrice A.
prod	prod(A)	bilo koja matrica	Računa umnožak članova u stupcima matrice A.
det	det(A)	bilo koja kvadratna matrica	Računa determinantu matrice A.
inv	inv(A)	bilo koja regularna matrica	Određuje inverz matrice A.

Primjenu svake pojedine funkcije pokazat ćemo na primjerima matrica

$$A = \begin{bmatrix} -1 & \frac{\sqrt{2}}{2} & 1 & 2\sqrt{2} \end{bmatrix},\ B = \begin{bmatrix} -1 \\ \frac{1}{3} \\ 2 \\ -\frac{4}{3} \end{bmatrix},\ C = \begin{bmatrix} 1 & -1 & 2 \\ 4 & -2 & 1 \\ 1 & -3 & 9 \end{bmatrix},\ D = \begin{bmatrix} 3 & -1 \\ 5 & -2 \end{bmatrix}.$$

Zadajmo ih u MATLAB-u upisivanjem sljedećega niza znakova u novi redak komandnoga prozora:

```
A=[-1 sqrt(2)/2 1 2*sqrt(2)], B=[-1;1/3;2;-4/3], C=[1 -1 2;4
-2 1;1 -3 9], D=[3 1;5 -2]
```

Potom prijeđimo na sljedeći primjer.

***Primjer* 10.** Generirajmo nulmatricu tipa (4, 5) i jediničnu matricu reda 6. Označimo te dvije matrice redom s E i F. U novi redak komandnoga prozora upišimo:

```
E=zeros(4,5), F=eye(6)
```

MATLAB će ispisati:

```
E =
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
F =

     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
```

***Primjer* 11.** Generirajmo dijagonalne matrice G i H čiji su elementi redom elementi matrice A, odnosno matrice B. U novi redak komandnoga prozora upišimo:

```
G=diag(A),H=diag(B)
```

i MATLAB će ispisati:

```
G =
  -1.00000000000000                  0                  0                  0
                  0   0.70710678118655                  0                  0
                  0                  0   1.00000000000000                  0
                  0                  0                  0   2.82842712474619
H =
  -1.00000000000000                  0                  0                  0
                  0   0.33333333333333                  0                  0
                  0                  0   2.00000000000000                  0
                  0                  0                  0  -1.33333333333333
```

***Primjer* 12.** Odredimo rangove matrica A, B, C i D. (Kažemo da neka matrica A ima *rang* jednak r ako maksimalan linearno nezavisan podskup skupa njezinih redaka (ili stupaca) ima točno r elemenata.) Označimo tražene rangove redom s a, b, c i d. U novi redak komandnoga prozora upišimo:

```
a=rank(A), b=rank(B), c=rank(C), d=rank(D)
```

pa će MATLAB ispisati:

```
a =
      1
b =
      1
c =
      2
d =
      2
```

Koje smo od dobivenih rezultata mogli unaprijed očekivati?

***Primjer* 13.** Označimo sa e broj redaka matrice C, a sa f broj stupaca matrice D. (Očito je $e = 3$ i $f = 2$.) Generirajmo te veličine u MATLAB-u. U novi redak komandnoga prozora upišimo:

```
e=size(C,1), f=size(D,2)
```

i dobit ćemo očekivane rezultate:

```
e =
     3
f =
     2
```

Funkcija *size* nam u prvi trenutak može izgledati posve beskorisno, no pravu njezinu korist vidjet ćemo u 4. poglavlju kada se budemo susreli s tzv. funkcijskim *m*-datotekama.

***Primjer* 14.** Odredimo zbroj i umnožak elemenata u svakomu pojedinomu stupcu matrice C. Označimo traženi zbroj sa z, a umnožak sa u. U novi redak komandnoga prozora upišimo:

```
z=sum(C),u=prod(C)
```

MATLAB će ispisati:

```
z =
      6    -6    12
u =
      4    -6    18
```

Tako je, npr., zbroj svih elemenata prvoga stupca matrice C jednak 6, a umnožak svih elemenata drugoga stupca te matrice jednak –6.

Pogledajmo kako "rade" te dvije funkcije kad su im argumenti jednoretčane ili jednostupčane matrice. U tu ćemo svrhu odrediti njihove pripadne vrijednosti za matrice A i B. Pritisnimo tipku $\uparrow$ pa će se u tekućemu retku komandnoga prozora pojaviti posljednje upisana naredba:

```
z=sum(C),u=prod(C)
```

Pomicanjem pokazivača i zamjenom slova C slovom A preuredimo tu naredbu u:

```
z=sum(A),u=prod(A)
```

Dobit ćemo:

```
z =
   3.53553390593274
u =
  -2.00000000000000
```

Uočavamo da je vrijednost varijable z jednaka zbroju svih elemenata matrice A, a vrijednost varijable u jednaka je njihovu umnošku.

Ponovno pritisnimo tipku ↑ i preuredimo posljednje upisanu naredbu ovako:

```
z=sum(B),u=prod(B)
```

Dobit ćemo:

```
z =
     0
u =
   0.88888888888889
```

Uočavamo da je vrijednost varijable z jednaka zbroju svih elemenata matrice B, a vrijednost varijable u jednaka je njihovu umnošku.

Možemo zaključiti: Ako je A jednoretčana ili jednostrupčana matrica, onda su vrijednosti funkcija $sum(A)$ i $prod(A)$ zbroj, odnosno umnožak svih elemenata matrice A. No, vrijedi i općenito:

Pravilo: Ako je A bilo koja matrica, onda su

```
z=sum(sum(A)),u=prod(prod(A))
```

redom zbroj, odnosno umnožak svih elemenata te matrice.

Primjer 15. Jedan od standardnih zadataka o matricama obično glasi: *Ispitajte ima li matrica… inverz i, ako ima, odredite ga.* Taj smo zadatak rješavali tako što smo najprije računali determinantu zadane matrice, pa ako je ona bila različita od nule, zaključili smo da zadana matrica ima inverz i njega smo potom određivali tehnički složenim postupkom. Ono što smo tada radili 15-ak minuta MATLAB će sad napraviti za približno 0.00015 sekundi.

Dakle, ispitajmo imaju li matrice C i D svoje inverze i, ako imaju, odredimo ih. Rabit ćemo funkcije *det* i *inv*. Za matricu C označimo $c := \det(C)$ i $K := C^{-1}$. U novomu retku komandnoga prozora izračunajmo vrijednosti tih veličina. Utipkajmo:

```
c=det(C),K=inv(C)
```

Dobit ćemo sljedeći ispis:

```
c =
     0

Warning: Matrix is singular to working precision.

K =
   Inf   Inf   Inf
   Inf   Inf   Inf
   Inf   Inf   Inf
```

Pogledajmo što se dogodilo. Iz ispisa vidimo da je vrijednost varijable *c* jednaka 0. Budući da ta vrijednost predstavlja vrijednost determinante matrice *C*, to znači da je determinanta matrice *C* jednaka 0. Stoga je matrica *C* singularna, a ne regularna matrica i nema svoj inverz[4]. To isto je ustanovio i MATLAB te nam ispisao poruku:

```
Upozorenje: S obzirom na postavljenu preciznost matrica je singularna.
```

Time MATLAB dozvoljava mogućnost da je determinanta matrice ipak različita od nule (npr. reda veličine 10^{-500}), ali napominje da s obzirom na postavljeni format dvostruke preciznosti on to ne može utvrditi. Kako bi varijabli *K* ipak bila dodijeljena određena vrijednost, uporabljena je unaprijed deklarirana varijabla *Inf* (vidjeti točku 1.2.).

Isti postupak sada provedimo za matricu *D*. Označimo najprije $d := \det(D)$ i $L := D^{-1}$. Pritisnimo tipku ↑ i preuredimo posljednje upisanu naredbu ovako:

```
d=det(D),L=inv(D)
```

Dobit ćemo:

```
d =
    -1
L =
   2.00000000000000  -1.00000000000000
   5.00000000000000  -3.00000000000000
```

Determinanta matrice *D* jednaka je –1, što znači da je matrica regularna i da ima inverz. Nije teško analitički provjeriti valjanost jednakosti (učinite to sami):

$$D^{-1} = \begin{bmatrix} 2 & -1 \\ 5 & -3 \end{bmatrix}.$$

---

4 Funkcija `det` ipak nije dovoljno pouzdana za provjeru singularnosti neke realne matrice . Naime, za singularne matrice ta funkcija često vraća neku malu vrijednost različitu od nule, pa je u takvim slučajevima teško razlikovati radi li se o singularnoj matrici ili regularnoj matrici .

2.4. Uporaba znakova , ; i : u MATLAB-u

U ovoj ćemo točki dati kratak pregled uporabe znakova zareza (,), točke-zareza (;) i dvotočke u MATLAB-u, i to u onoj mjeri u kojoj će nam trebati u sljedećim poglavljima.

Jednu od mogućih uporabi zareza već smo vidjeli. Njime smo odvajali članove niza naredbi zapisanoga u jednomu retku. To je osobito praktično ako su naredbe "kratke" (kao u većini naših dosadašnjih primjera). U slučaju "duljega" niza naredbi, radi preglednosti, preporuča se svaku od njih zapisati u poseban redak komandnoga prozora. Zarez se obično rabi u kombinacijama sa znakom točke-zareza radi reguliranja ispisa rezultata naredbi. Pogledajmo to na primjeru.

***Primjer* 1.** U novi redak komandnoga prozora utipkajmo:

```
a=5;b=7;
```

Posljednja dva retka u našemu komandnomu prozoru sada izgledaju ovako :

```
>> a=5;b=7;
>> _
```

U prvi mah mislimo da je MATLAB "zatajio" jer ništa nije ispisao. A ništa nije ispisao jer smo mu stavljajući znakove ; rekli da ne želimo ispis niti vrijednosti varijable a niti vrijednosti varijable b. On je "poslušno" tim varijablama dodijelio vrijednosti koje smo mu zadali, pa "čeka" naše daljnje naputke što da radi s njima.

Utipkajmo sad (u retku gdje se nalazi trepćući pokazivač):

```
>> a=5,b=7;c=6,d=8;
```

MATLAB će ispisati:

```
a =
       5
c =
       6
```

Protumačimo što se ovdje dogodilo. Stavljajući znak zareza "rekli" smo MATLAB-u: "Unos naredbe je završen. Želimo njezino izvršenje i ispis rezultata." Stavljajući znak točka-zarez "rekli" smo MATLAB-u: "Unos naredbe je završen. Želimo njezino izvršenje, *ali ne* i ispis rezultata." Nakon što smo pritisnuli *Enter*, MATLAB je "pročitao" posljednje upisani redak na sljedeći način: "Varijabli a trebam dodijeliti vrijednost 5 i ispisati rezultat. Varijabli b trebam dodijeliti vrijednost 7 bez ispisa rezultata. Varijabli c trebam dodijeliti vrijednost 6 i ispisati rezultat. Varijabli d trebam dodijeliti vrijednost 8 bez ispisa rezultata." Zbog toga se kao rezultat ovoga niza naredbi pojavio ispis vrijednosti varijabli a i c.

Ovome smo posvetili posebnu pozornost jer je česta početnička pogrješka u MATLAB-u izostavljanje točke-zareza u pisanju *m*-datoteka (koje ćemo upoznati u 4. poglavlju) zbog kojega se javlja "glomazan" ispis posve nepotrebnih podataka. Važno je zapamtiti da svaki ispis rezultata neke naredbe možemo spriječiti stavljajući točku-zarez na kraj te naredbe.

Još jedan koristan znak je i dvotočka. Ona je posebno korisna pri generiranju različitih matrica. Tako naredba

$$ime\_varijable = matrica(n,:)$$

gdje su *matrica* unaprijed zadana realna matrica i $n \in \mathbf{N}$ prirodan broj, pohranjuje n – ti *redak* matrice *matrica* u varijablu *ime_varijable*. Analogno, naredba

$$ime\_varijable = matrica(:, n)$$

gdje su *matrica* unaprijed zadana realna matrica i $n \in \mathbf{N}$ prirodan broj, pohranjuje n – ti *stupac* matrice *matrica* u varijablu *ime_varijable*. Pogledajmo to na primjeru.

Primjer 2. Zadajmo sljedeću matricu u MATLAB-u, te u varijable *TRECIREDAK* i *DRUGISTUPAC* pohranimo, redom, treći redak i drugi stupac te matrice s odgovarajućim ispisom rezultata.

$$A = \begin{bmatrix} -1 & 2 & 0 \\ \log 5 & e^{-2} & \ln 3 \\ \sin\dfrac{\pi}{12} & \operatorname{ctg} 2 & \arcsin\dfrac{1}{3} \\ \operatorname{Re}\left(\dfrac{1+i}{i}\right) & \operatorname{Im}\left(\dfrac{1-i}{2\cdot i}\right) & \left|\overline{-\dfrac{\sqrt{3}}{2}-\dfrac{1}{2}i}\right| \end{bmatrix}$$

U novi redak komandnoga prozora najprije utipkajmo:

```
A=[-1  2  0;log10(5)  exp(-2)  log(3);sin(pi/12)  1/tan(2)
asin(1/3);real((1+i)/i)  imag((1-i)/(2*i))  abs(conj(-sqrt(3)/2-
1/2*i))]
```

pa dobijemo:

```
A =
  -1.00000000000000   2.00000000000000                  0
   0.69897000433602   0.13533528323661   1.09861228866811
   0.25881904510252  -0.45765755436029   0.33983690945412
   1.00000000000000  -0.50000000000000   1.00000000000000
```

Preostaje upisati:

```
TRECIREDAK=A(3,:),DRUGISTUPAC=A(:,2)
```

Dobivamo:

```
TRECIREDAK =
   0.25881904510252  -0.45765755436029   0.33983690945412

DRUGISTUPAC =
   2.00000000000000
   0.13533528323661
  -0.45765755436029
  -0.50000000000000
```

Ako imamo *jednoretčanu* matricu čiji elementi tvore aritmetički niz (vrlo čest slučaj kod petlji!), možemo je generirati rabeći naredbu čija je sintaksa:

ime_matrice=*prvi_član_niza*:*razlika_niza*:*posljednji_član_niza*

gdje su nazivi varijabli sugestivno odabrani kako bi se znalo o čemu je riječ. Takav način generiranja matrice naziva se *generiranje prema razmaku među* (*susjednim*) *elementima.* Ako je razlika niza jednaka 1, ona se ne mora pisati. Pokažimo to na primjeru.

Primjer 3. Generirajmo matricu $A = \begin{bmatrix} 0 & \frac{\pi}{5} & \frac{2\cdot\pi}{5} & \frac{3\cdot\pi}{5} & \frac{4\cdot\pi}{5} & \pi \end{bmatrix}$. Uočimo da članovi te matrice tvore aritmetički niz čiji je prvi član 0, posljednji π, a razlika $\frac{\pi}{5}$. Matrica je jednoretčana pa možemo primijeniti gornju naredbu. U novi redak komandnoga prozora utipkamo:

```
A=0:pi/5:pi
```

pa dobivamo:

```
A =
  Columns 1 through 4
                 0   0.62831853071796   1.25663706143592   1.88495559215388
  Columns 5 through 6
   2.51327412287183   3.14159265358979
```

Ovakav se način ispisa u MATLAB-u pojavljuje kad zatražimo da MATLAB sâm generira neku matricu s više od 4 stupca. Budući da sve elemente svih stupaca jednoga retka ne može ispisati u istomu retku (zbog ograničenosti komandnoga prozora), MATLAB ispisuje elemente *po stupcima* svrstavajući stupce u grupe s (najviše) 4 elementa.

***Napomena*:** Primjer 3. moguće je riješiti i uporabom naredbe *linspace* čija je sintaksa

ime_matrice=`linspace`(*prvi_član_niza*,*posljednji_član_niza*,*ukupan_broj_članova_niza*)

Već smo utvrdili koji je prvi, a koji posljednji član niza. Još samo trebamo podatak koliko ukupno članova ima naš niz. U ovome je slučaju taj broj jednak 6, pa utipkamo:

```
A=linspace(0,pi,6)
```

i dobivamo isti rezultat kao i u Primjeru 3. Ovakav način generiranja matrice naziva se *generiranje prema broju elemenata*.

2.5. Još neke primjene matričnoga računa

Na kraju ove točke pokazat ćemo kako pomoću matričnoga računa možemo riješiti neke zadatke s radijvektorima, te zadatke s rješavanjem linearnih sustava reda n, tj. sustava n linearnih jednadžbi s n nepoznanica. Podsjetimo ukratko na osnovne poučke koje smo upoznali u *Matematici* 1.

***Poučak* 1.** Neka je $S = \{(a_{11}, a_{12}, \ldots, a_{1n}), (a_{21}, a_{22}, \ldots, a_{2n}), \ldots (a_{n1}, a_{n2}, \ldots, a_{nn})\}$ skup vektora. Taj skup je linearno nezavisan ako i samo ako je

$$\left|\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}\right| \neq 0.$$

Ekvivalentno, skup S je linearno zavisan ako je determinanta matrice koju tvore njegovi elementi jednaka nuli.

Napomenimo da u Poučku 1. *nije bitan* poredak elemenata skupa S prigodom formiranja pripadne matrice. Naime, mi jedino želimo utvrditi je li determinanta te matrice jednaka nuli ili različita od nule. Budući da zamjena dvaju redaka (stupaca) determinante mijenja jedino njezin predznak, zaključujemo da promjena poretka vektora ne utječe na zaključak o linearnoj (ne)zavisnosti polaznoga skupa vektora.

Posebno, za $n = 3$ linearnu nezavisnost tročlanoga skupa radijvektora možemo opisati i na još neke načine.

***Poučak* 2.** Neka je $S = \{(a_{11}, a_{12}, a_{13}), (a_{21}, a_{22}, a_{23}), (a_{31}, a_{32}, a_{33})\}$ skup radijvektora. Tada je skup S linearno nezavisan ako i samo ako je *mješoviti umnožak* elemenata skupa *S u bilo kojem poretku* tih elemenata različit od nule.

Poučak 2. je, dakako, izravna posljedica činjenice da je mješoviti umnožak triju radijvektora jednak upravo determinanti matrice koju tvore ti radijvektori. Geometrijska interpretacija mješovitoga umnoška radijvektora jest

***Poučak* 3.** Neka je $S = \{(a_{11}, a_{12}, a_{13}), (a_{21}, a_{22}, a_{23}), (a_{31}, a_{32}, a_{33})\} \subseteq V^3(O)$ skup radijvektora.

a) Obujam paralelepipeda kojega razapinju svi elementi skupa S jednak je apsolutnoj vrijednosti mješovitoga umnoška tih elemenata.

b) Obujam tetraedra kojega razapinju svi elementi skupa S je šest puta manji od obujma paralelepipeda iz **a)** podzadatka.

Naglasimo da mješoviti umnožak triju radijvektora općenito može biti bilo koji realan broj, pa, zbog činjenice da obujam geometrijskoga tijela ne može biti strogo negativan realan broj, moramo izračunati apsolutnu vrijednost toga umnoška.

***Poučak* 4.** Neka je zadan sustav n linearnih jednadžbi s n nepoznanica:

$$\begin{array}{ccccccc} a_{11}\cdot x_1 & + & a_{12}\cdot x_2 & + & \ldots & + & a_{1n}\cdot x_n = b_1 \\ a_{21}\cdot x_1 & + & a_{22}\cdot x_2 & + & \ldots & + & a_{2n}\cdot x_n = b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1}\cdot x_1 & + & a_{n2}\cdot x_2 & + & \ldots & + & a_{nn}\cdot x_n = b_n \end{array}$$

Neka su

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \; b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

redom *matrica sustava*, *matrica nepoznanica* i *matrica slobodnih članova* zadanoga sustava. Za svaki $k = 1, 2, \ldots, n$ označimo s A_k matricu dobivenu zamjenom k–toga stupca matrice A matricom slobodnih članova sustava. Tada:

- sustav ima jedinstveno rješenje (tj. sustav je *Cramerov*) ako je $\det(A) \neq 0$ i u tome je slučaju njegovo rješenje dano formulom $X = A^{-1} \cdot b$;
- sustav nema rješenja ako je $\det(A) = 0$ i ako postoji barem jedan $k \in \{1, \ldots, n\}$ takav da je $\det(A_k) \neq 0$;
- sustav ima beskonačno mnogo različitih rješenja ako je $\det(A) = \det(A_1) = \ldots = \det(A_n) = 0$.

Pogledajmo primjenu ovih poučaka na nekoliko primjera.

Primjer 1. Ispitajmo je li skup radijvektora S = {(–2, 3, 4), (–3, 4, –2), (1, 0, 22)} linearno nezavisan. Detaljno obrazložimo sve svoje tvrdnje.

Strategija rješavanja slijedi iz gornje napomene. Formirat ćemo matricu A čiji će retci biti vektori koji tvore skup S, pa ćemo izračunati determinantu te matrice. Bude li ta determinanta jednaka nuli, skup S bit će linearno zavisan, a u suprotnom, skup S bit će linearno nezavisan. Vrijedi istaknuti da pri formiranju matrice A redoslijed vektora koji tvore skup S može biti proizvoljan. Naime, mi jedino želimo usporediti vrijednost determinante matrice A s nulom, pa zamjena dvaju redaka determinante (prigodom koje se mijenja jedino predznak determinante) nema utjecaja na konačan zaključak.

Radi preglednosti, najprije „počistimo“ MATLAB-ov komandni prozor. U novi redak toga prozora utipkajmo:

```
clc
```

i pritisnimo *Enter*. Što se dogodilo? Sav sadržaj koji je bio ispisan u komandnom prozoru je nepovratno nestao! Bez brige, nije se dogodio smak svijeta: „očišćen“ je samo komandni prozor, ali vrijednosti varijabli iz ranijih primjera ostale su „netaknute“ (tj. pohranjene u odgovarajućim memorijskim ćelijama), pa ih, prema potrebi, možemo i nadalje koristiti.

U nova dva retka komandnoga prozora utipkajmo:

```
A=[-2 3 4;-3 4 -2;1 0 22];
det(A)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
     0
```

Dakle, determinanta pripadne matrice jednaka je nuli, pa je skup S linearno zavisan. Za vježbu, izrazite treći element toga skupa kao linearnu kombinaciju preostalih dvaju elemenata.

Primjer 2. Ispitajmo je li skup $S = \left\{ \left(\ln 2,\ \log 2,\ \log_2 3 \right), \left(e^{\pi},\ \pi^{e}, \log_{\pi} e \right), \left(\sqrt{e}, \sqrt[3]{\pi}, \sqrt[3]{e} \right) \right\}$ linearno nezavisan. Detaljno obrazložimo sve svoje tvrdnje.

Primjenjujući jednakost $\log_{\pi} e = \dfrac{1}{\ln \pi}$ utipkamo:

```
A=[log(2) log10(2) log2(3);exp(pi) pi^exp(1) 1/log(pi);sqrt(exp(1)) pi^(1/3) exp(1/3)];
det(A)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
   6.57866424431862
```

Dakle, determinanta matrice koju tvore elementi zadanoga skupa je različita od nule, pa je zadani skup linearno nezavisan.

***Primjer* 3.** Neka je $S = \{(1, 2, 3), (3, 2, 1), (6, 4, 5)\}$. Izračunajmo:
a) mješoviti umnožak elemenata skupa S (u zadanom poretku);
b) obujam paralelepipeda razapetoga svim elementima skupa S;
c) obujam tetraedra razapetoga svim elementima skupa S.

„Počistimo“ komandni prozor koristeći funkciju `clc`, pa u nova dva njegova retka utipkajmo:

```
A=[1 2 3;3 2 1;6 4 5];
M=det(A), Vp=abs(M), Vt=1/6*Vp
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
M =
   -12
Vp =
    12
Vt =
     2
```

Dakle, mješoviti umnožak elemenata skupa S u zadanom poretku jednak je –12. Obujam paralelepipeda kojega razapinju svi elementi skupa S iznosi 12 kub. jed. Obujam tetraedra kojega razapinju svi elementi skupa S iznosi 2 kub. jed.

***Primjer* 4.** Provjerimo jesu li radijvektori $a = (1, 2, 3)$, $b = (4, 5, 6)$ i $c = (7, 8, 9)$ komplanarni (tj. pripadaju li istoj ravnini). Detaljno obrazložimo sve svoje tvrdnje.

Zadani radijvektori su komplanarni ako i samo ako je njihov mješoviti umnožak jednak nuli. Stoga u nova dva retka komandnoga prostora utipkamo:

```
A=[1 2 3;4 5 6;7 8 9];
det(A)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
     0
```

Dakle, mješoviti umnožak zadanih radijvektora jednak je nuli, pa su oni komplanarni. Za vježbu, izrazite treći radijvektor kao linearnu kombinaciju prvih dvaju radijvektora.

Bilo koje linearne sustave u MATLAB-u izravno možemo rješavati koristeći funkciju `solve` i operacije s tzv. simboličkim objektima koje ćemo upoznati u 5. poglavlju. Ovdje ćemo prikazati tehnički malo složeniji način rješavanja linearnih sustava reda n, i to pomoću Cramerova pravila. Preciznije, na primjerima ćemo riješiti

Problem: Za zadani linearni sustav reda n (sustav od n jednadžbi s n nepoznanica) odredimo ukupan broj njegovih različitih rješenja. Ako je sustav Cramerov (tj. ako ima jedinstveno rješenje), odredimo to rješenje Cramerovim pravilom.

Primjer 5. Odredimo ukupan broj rješenja sustava

$$\begin{aligned} 2 \cdot x - 3 \cdot y + 4 \cdot z &= 3, \\ 3 \cdot x - 4 \cdot y + 2 \cdot z &= -3, \\ 4 \cdot x + 2 \cdot y - 3 \cdot z &= -8. \end{aligned}$$

Ako je sustav Cramerov, riješimo ga Cramerovim pravilom i provjerimo svoje rješenje koristeći jednakost $X = A^{-1} \cdot b$.

Matrica sustava A i matrica slobodnih članova b zadanoga sustava su redom

$$A = \begin{bmatrix} 2 & -3 & 4 \\ 3 & -4 & 2 \\ 4 & 2 & -3 \end{bmatrix}, \; b = \begin{bmatrix} 3 \\ -3 \\ -8 \end{bmatrix}.$$

„Počistimo" MATLAB-ov komandni prozor, pa u novom retku toga prozora utipkajmo:

```
A=[2 -3 4;3 -4 2;4 2 -3]; b=[3;-3;-8];
```

Izračunajmo determinantu matrice A kako bismo dobili podatak o tome je li zadani sustav Cramerov ili nije. Utipkamo:

```
det(A)
```

pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
     53
```

Dakle, determinanta matrice sustava je različita od nule, pa je sustav Cramerov i ima jedinstveno rješenje. Da bismo dobili to rješenje pomoću Cramerova pravila, u nova dva retka komandnoga prozora utipkamo:

```
A1=[b A(:,2) A(:,3)];A2=[A(:,1) b A(:,3)]; A3=[A(:,1) A(:,2) b];
X=1/det(A)*[det(A1) det(A2) det(A3)]
```

Uočimo kako smo efektivno dobili matrice A_1, A_2 i A_3. U svakoj od njih smo točno jedan stupac matrice A zamijenili matricom slobodnih članova, a ostala dva stupca matrice A smo

„prepisali“. Dakle, matricu možemo zadati i koristeći retke/stupce već zadanih matrica. Matricu nepoznanica X dobili smo izravno primjenjujući Cramerovo pravilo.

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
X =
    -1      1      2
```

Provjera pomoću jednakosti $X = A^{-1} \cdot b$, tj. utipkavanje

```
X=A\b
```

(ili, ekvivalentno, `X=inv(A)*b`) ponovno kao rezultat (ali zapisan u drugačijem obliku zapisa realnoga broja) daje

```
X =
  -1.00000000000000
   1.00000000000000
   2.00000000000000
```

Zaključimo: polazni sustav ima jedinstveno rješenje $(x, y, z) = (-1, 1, 2)$.

Napomena: Za provjeru valjanosti rješenja pomoću jednakosti $X = A^{-1} \cdot b$ efikasnije je koristiti matrično lijevo dijeljenje, tj. jednakost $X = A \setminus b$.

Primjer 6. Odredimo ukupan broj rješenja sustava

$$\begin{aligned} 36 \cdot x + 31 \cdot y - 26 \cdot z &= 21, \\ 21 \cdot x - 26 \cdot y + 31 \cdot z &= -36, \\ 213 \cdot x - 37 \cdot y + 77 \cdot z &= -117. \end{aligned}$$

Ako je sustav Cramerov, riješimo ga Cramerovim pravilom i provjerimo svoje rješenje koristeći jednakost $X = A^{-1} \cdot b$.

Postupimo analogno kao u prethodnom primjeru. Matrica sustava i matrica slobodnih članova zadanoga sustava su redom

$$A = \begin{bmatrix} 36 & 31 & -26 \\ 21 & -26 & 31 \\ 213 & -37 & 77 \end{bmatrix}, \; b = \begin{bmatrix} 21 \\ -36 \\ -117 \end{bmatrix}.$$

„Počistimo“ MATLAB-ov komandni prozor, pa u dva nova retka toga prozora utipkamo:

```
A=[36 31 -26;21 -26 31;213 -37 77]; b=[21;-36;-117];
det(A)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
     0
```

Odatle slijedi da polazni sustav ili nema niti jedno rješenje ili ima beskonačno mnogo rješenja. Utipkamo li kao u Primjeru 5.

```
A1=[b A(:,2) A(:,3)];A2=[A(:,1) b A(:,3)]; A3=[A(:,1) A(:,2) b];
X=1/det(A)*[det(A1) det(A2) det(A3)]
```

i pritisnemo *Enter*, MATLAB će ispisati:

```
Warning: Divide by zero.
X =
   NaN   NaN   NaN
```

Pojasnimo što se dogodilo. Najprije se pojavilo upozorenje da dijelimo s nulom, što smo, naravno, mogli i očekivati jer je determinanta matrice sustava jednaka nuli. Potom je MATLAB ispisao da je rješenje sustava (x, y, z) = (NaN, NaN, NaN), tj. tročlani vektor čije su komponente međusobno jednake konstanti NaN. Prisjetimo se da se ta „konstanta" pojavljuje pri „dijeljenju" $\frac{0}{0}$, pa zaključujemo da su determinanta sustava i sve tri pomoćne determinante međusobno jednake i jednake nuli. Stoga polazni sustav ima beskonačno mnogo različitih realnih rješenja. MATLAB nam nije ispisao niti jedno od njih, što je potpuno u skladu s Cramerovim pravilom koje – za razliku od ostalih metoda i tehnika rješavanja linearnih sustava – *ne* omogućuje ispis niti jednoga rješenja u ovakvom slučaju (iako znamo da ih ima beskonačno mnogo!).

Zaključimo: ako linearni sustav reda n ima beskonačno mnogo međusobno različitih rješenja, MATLAB će nam ispisati da je svaka pojedina komponenta *jednoga* rješenja jednaka NaN.

Pokušamo li, međutim, dobiti rješenje pomoću jednakosti $X = A^{-1} \cdot b$, tj. utipkamo li:

```
X=A\b
```

u novom retku komandnoga prostora i potom pritisnemo *Enter*, MATLAB će ispisati:

```
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 5.908903e-018.
X =
  -0.17958412098299
   0.04725897920605
  -1.00000000000000
```

U ovom nas je slučaju MATLAB upozorio da je matrica približno singularna (ili tzv. *loše skalirana*, što se odnosi na problem tzv. *skaliranja matrice* o kojemu ovdje nećemo govoriti), da rezultat može biti netočan i da je rezultat procijenjen s parametrom $5.9 \cdot 10^{-18}$ (radi ilustracije, procjenitelj je dobar što je njegova vrijednost bliža jedinici, a lošiji što je njegova vrijednost bliža konstanti `eps`). Stoga je riječ o loše procijenjenom rješenju sustava. Ako ste se poveselili da u ovakvim slučajevima MATLAB ispisuje jedno od spomenutih beskonačno mnogo postojećih realnih rješenja polaznoga sustava, to ste prerano učinili. Naime, nije teško provjeriti da $(x, y, z) = (-0.17958412098299, 0.04725897920605, -1)$ nije rješenje polaznoga sustava. Dakle, niti ova metoda ne omogućuje ispis barem jednoga od beskonačno mnogo postojećih realnih rješenja polaznoga sustava.

Primjer 7. Odredimo ukupan broj rješenja sustava

$$\begin{aligned} 4 \cdot x - 7 \cdot y + 9 \cdot z &= 11, \\ 5 \cdot x + 13 \cdot y - 7 \cdot z &= 8, \\ x + 107 \cdot y - 89 \cdot z &= -25. \end{aligned}$$

Ako je sustav Cramerov, riješimo ga Cramerovim pravilom i provjerimo svoje rješenje koristeći jednakost $X = A^{-1} \cdot b$.

Postupimo analogno kao u prethodnim primjerima. Matrica sustava i matrica slobodnih članova zadanoga sustava su redom

$$A = \begin{bmatrix} 4 & -7 & 9 \\ 5 & 13 & -7 \\ 1 & 107 & -89 \end{bmatrix}, \; b = \begin{bmatrix} 11 \\ 8 \\ -25 \end{bmatrix}.$$

„Počistimo“ MATLAB-ov komandni prozor, pa u dva nova retka toga prozora utipkamo:

```
A=[4 -7 9;5 13 -7;1 107 -89]; b=[11;8;-25];
det(A)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
     0
```

Odatle slijedi da polazni sustav ili nema niti jedno rješenje ili ima beskonačno mnogo rješenja. Utipkamo li kao u Primjeru 5.

```
A1=[b A(:,2) A(:,3)];A2=[A(:,1) b A(:,3)]; A3=[A(:,1) A(:,2) b];
X=1/det(A)*[det(A1) det(A2) det(A3)]
```

i pritisnemo *Enter*, MATLAB će ispisati:

```
Warning: Divide by zero.
X =
  -Inf   Inf   Inf
```

I opet smo dobili (očekivanu) poruku da dijelimo s nulom, ali ovoga je puta rješenje sustava drugačije: sve tri komponente po apsolutnoj su vrijednosti jednake konstanti `Inf`. Ta se konstanta dobije kad neki „konkretan" realan broj dijelimo nulom. To znači da smo pri izračunu komponenata dijelili „konkretne" realne brojeve nulom, odnosno da su determinante svih triju pomoćnih matrica različite od nule. Stoga zaključujemo da polazni sustav nema niti jedno realno rješenje.

Pokušamo li ponovno dobiti rješenje pomoću jednadžbe $X = A^{-1} \cdot b$, tj. utipkamo li:

```
X=A\b
```

u novom retku komandnoga prostora i potom pritisnemo *Enter*, MATLAB će ispisati:

```
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 1.601150e-017.
X =
  1.0e+013 *
  -2.44448255636308
   2.62422392080175
   3.12749974122953
```

Prvi dio ispisane poruke je jednak kao u Primjeru 6. Vrijednost „procjeniteljske" varijable RCOND i u ovom je slučaju vrlo blizu konstanti `eps`, pa zaključujemo da je riječ o vrlo lošoj procjeni rješenja[5]. Stoga ispisano rješenje *nije* rješenje polaznoga sustava.

Zaključimo: ako linearni sustav reda *n* nema rješenja, MATLAB će nam ispisati da je svaka pojedina komponenta *jednoga* rješenja (po apsolutnoj vrijednosti) jednaka Inf.

[5] Iz dobivene vrijednosti „procjeniteljske" varijable zapravo možemo zaključiti da su *sve* znamenke *svake* komponente dobivenoga rješenja netočne. Npr. za vrijednost *RCOND* = 0.01 mogli bismo zaključiti da su netočne samo posljednje dvije znamenke *svake* komponente dobivenoga rješenja itd.

TVŽ

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

2.6. Zadatci za vježbu

1. Generirajte sljedeće matrice u MATLAB-u:

a) $A = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix};$

b) $B = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix};$

c) $C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix};$

d) $D = \begin{bmatrix} -1 & 2 & -3 \\ 4 & -5 & 6 \\ -7 & 8 & -9 \end{bmatrix};$

e) $E = \begin{bmatrix} 1.2 & 0.5 & \frac{1}{3} & \sqrt{2} \\ \sqrt[3]{7} & \log 0.5 & 4^{1.23} & \frac{3}{\sqrt{2}} \\ -1.9 & -\frac{1}{3} & -\sqrt{2}-\sqrt[3]{3} & \ln 2 \end{bmatrix}.$

2. **a)** Za matrice A, B, C, D i E iz prethodnoga zadatka odredite postoje li umnošci $A \cdot B$, $B \cdot A$, $A \cdot C$, $C \cdot A$, $A \cdot D$, $D \cdot A$, $A \cdot E$, $E \cdot A$, $B \cdot C$, $C \cdot B$, $B \cdot D$, $D \cdot B$, $B \cdot E$, $E \cdot B$, $C \cdot D$, $D \cdot C$, $C \cdot E$, $E \cdot C$, $D \cdot E$ i $E \cdot D$ i izračunajte sve umnoške koji postoje.

b) Odredite rangove svih (postojećih) umnožaka iz **a)** podzadatka. Interpretirajte dobivene rezultate. Ima li među dobivenim matricama regularnih matrica? Obrazložite svoje odgovore.

3. Izračunajte umnožak svake od matrica iz **1**. zadatka s njoj transponiranom matricom, pa odredite rangove svih tako dobivenih matrica. Jesu li koje od njih regularne? Obrazložite svoj odgovor.

4. **a)** Za svaku od matrica iz **1**. zadatka odredite ima li determinantu. Ako ima, izračunajte je.

b) Za svaku od matrica iz **1**. zadatka odredite ima li inverz. Ako ima, izračunajte ga.

5. Za matricu A iz **1.a**) zadatka odredite matricu $X = \sqrt{A+1}$. Za svaki član matrice X napišite formulu po kojoj je izračunan. Potom izračunajte kvadrat zbroja svih elemenata matrice X. Provjerite svoja rješenja rabeći MATLAB.

6. **a)** Za matricu B iz **1.b)** zadatka odredite matricu $Y = \ln\dfrac{B+\sqrt{3}}{2}$. Za svaki član matrice Y napišite formulu po kojoj je izračunan.

b) Označimo sa r apsolutnu vrijednost (modul) razlike umnoška i zbroja svih elemenata matrice Y iz **a)** zadatka. Napišite izraz za računanje vrijednosti varijable r. Provjerite valjanost svojega rješenja rabeći MATLAB.

7. Neka su D i E matrice iz zadataka **1.d)** i **1.e)**. Označimo sa G i H jednoretčane matrice koje tvore redom drugi redak matrice D, odnosno treći stupac matrice E. Bez upisivanja članova matrica G i H izračunajte $(\det D^T) \cdot G - (E \cdot E^T) \cdot (2H)^T$.

8. Označimo sa X matricu tipa 1×6 čiji su članovi prvih 6 višekratnika broja 6. Generirajte matricu X prema:

a) broju elemenata;
b) razmaku među susjednim elementima.

10. Ispitajte jesu li sljedeći skupovi vektora linearno nezavisni i obrazložite svoje odgovore:

a) $S = \left\{\left(\sqrt{2}-\sqrt{3}, \sqrt{3}+\sqrt{2}, \sqrt{5}\right), \left(\sqrt{5}-\sqrt{3}, \sqrt{5}+\sqrt{3}, \sqrt{2}\right), \left(\sqrt{5}-\sqrt{2}, \sqrt{5}+\sqrt{2}, \sqrt{3}\right)\right\}$;

b) $S = \left\{\left(e, e^2, e^3\right), (1, \log 2, \log 3), (1, \ln 2, \ln 3)\right\}$;

c) $S = \{(-2009, 2010, -2011), (2009, -2010, 2011), (0, 1, 1)\}$;
d) $S = \{(1, 9, 7), (2, 1, 1), (-1, 8, 6)\}$;
e) $S = \{(1, -3, 4), (-2, 1, 7), (8, -9, -13)\}$.

11. Izračunajte mješoviti umnožak, obujam paralelepipeda i obujam tetraedra razapetoga svim elementima skupa S ako je:

a) $S = \{(10, 8, 9), (6, 4, 5), (3, 2, 4)\}$;
b) $S = \{(-2, 1, 1), (-12, 6, -18), (1, -1, 3)\}$;
c) $S = \{(-2, 2, 4), (-3, 6, 9), (1, -1, 1)\}$;
d) $S = \{(4, 0, -2), (3, 3, -3), (2, 3, -1)\}$;
e) $S = \{(-14, -12, -10), (15, 12, 9), (-16, -9, -4)\}$.

12. Klasificirajte sljedeće sustave linearnih jednadžbi u zavisnosti o ukupnom broju njihovih rješenja i obrazložite svoje odgovore:

a) $7 \cdot x - 11 \cdot y + 13 \cdot z = 6$, $11 \cdot x - 13 \cdot y + 7 \cdot z = -4$, $13 \cdot x + 7 \cdot y - 11 \cdot z = -24$;
b) $24 \cdot x - 31 \cdot y + 15 \cdot z = 131$, $15 \cdot x + 24 \cdot y - 31 \cdot z = -126$, $(-31) \cdot x - 15 \cdot y + 24 \cdot z = 71$;
c) $(-5) \cdot x + 4 \cdot y - 3 \cdot z = -2$, $(-2) \cdot x - 3 \cdot y + 4 \cdot z = 5$, $x - 33 \cdot y + 37 \cdot z = 40$;
d) $(-10) \cdot x - 11 \cdot y + 12 \cdot z = -13$, $(-13) \cdot x + 12 \cdot y - 11 \cdot z = -10$, $8 \cdot x - 149 \cdot y + 150 \cdot z = -32$;
e) $35 \cdot x + 71 \cdot y + 97 \cdot z = 1$, $(-71) \cdot x + 97 \cdot y + 35 \cdot z = 2$, $28 \cdot x + 1744 \cdot y + 1700 \cdot z = 29$.

13. Isključivo pomoću Cramerova pravila riješite sljedeće sustave linearnih jednadžbi:

a) $\begin{aligned} 3\cdot x+2\cdot y&=12,\\ 2\cdot x+3\cdot y&=13; \end{aligned}$

b) $\begin{aligned} x+y+z&=4,\\ x+2\cdot y+3\cdot z&=5,\\ 5\cdot x+3\cdot y+5\cdot z&=14; \end{aligned}$

c) $\begin{aligned} 3\cdot x+2\cdot y+z&=-1,\\ 7\cdot x+6\cdot y+5\cdot z&=2,\\ 5\cdot x+4\cdot y+3\cdot z&=2; \end{aligned}$

d) $\begin{aligned} 5\cdot x-2\cdot y+3\cdot z&=11,\\ -3\cdot x+7\cdot y-z&=-9,\\ 2\cdot x+5\cdot y+2\cdot z&=2; \end{aligned}$

e) $\begin{aligned} 2\cdot x-5\cdot y+4\cdot z+3\cdot w&=4,\\ 3\cdot x-4\cdot y+7\cdot z+5\cdot w&=11,\\ 4\cdot x-9\cdot y+8\cdot z+5\cdot w&=8,\\ -3\cdot x+2\cdot y-5\cdot z+3\cdot w&=-3. \end{aligned}$.

Potom riješite te sustave bez uporabe MATLAB-a i usporedite dobivena rješenja.

14. Bez uporabe MATLAB-a odredite što će se ispisati utipkavanjem sljedećih naredbi:

a) `a=3/2;b=4/5`
b) `a=3+5,b=4-5;`
c) `a=5/6;c=a+1/6,b=a-1/3`
d) `a=sqrt(9/4);b=log(a-1/2);c=log10(b+9);`

Provjerite svoje odgovore rabeći MATLAB.

15. Bez uporabe MATLAB-a odredite vrijednosti varijabli a i b nakon izvršenja sljedećega niza naredbi:

```
a=3/2;b=1/4;a=a+b,b=b-a
```

16. Bez uporabe MATLAB-a odredite vrijednosti varijabli x, y i z nakon izvršenja svakoga od sljedećega niza naredbi:

a) `x=-1/2;y=1/2;z=1;x=x-y+z,y=x-y-z,z=x+y+z;`
b) `x=-1/6;y=1/3;z=1/2;x=x+y+z,y=y+z-x,z=z+x-y;`
c) `x=i;y=pi;z=exp(1);x=x*y*z,y=x/y,z=z/(x*y).`

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

§ 3. GRAFIKA U MATLAB-U

MATLAB nam omogućuje crtanje različitih vrsta grafikona (linijskih, retčanih, stupčanih itd.) U ovom ćemo poglavlju upoznati "klasične", odnosno linijske grafikone, dok ćemo retčane i stupčane upoznati u 5. poglavlju prigodom obrađivanja grafičkih prikaza statističkih veličina.

3.1. Jednostavni linijski grafikoni

Za crtanje jednostavnih linijskih grafikona u MATLAB-u se rabi funkcija *plot* čija je jedna od sintaksi

plot(*matrica*)

gdje je *matrica* bilo koja realna matrica. Ideja "crtanja" takve matrice je sljedeća: Pretpostavimo da je $A = [a_{ij}]$ realna matrica tipa (m, n) koju želimo "nacrtati".

***Korak* 1**. Svakom elementu a_{ij} pridruži se uređeni par (i, a_{ij}). Tako se dobije ukupno $m \cdot n$ različitih točaka.

***Korak* 2**. Točke dobivene u Koraku 1. ucrtaju se u pravokutni koordinatni sustav u ravnini.

***Korak* 3**. Stavi se $j := 1$. Nacrta se ukupno m spojnica točaka (i, a_{ij}) i $(i + 1, a_{i+1, j})$, za svaki $i = 1, 2, \ldots, m - 1$. Nakon povlačenja zadnje (m – 1 –ve) spojnice stavi se $j := j + 1$.

***Korak* 4**. Ako je $j \leq n$, ponovi se Korak 3. (za novi j). Ako je $j > n$, postupak je gotov.

Na osnovi ovoga algoritma pokreće se i izvršava funkcija *plot*. Pogledajmo njezinu primjenu na primjerima.

***Primjer* 1.** "Nacrtajmo" matricu

$$A = [-1\ \ 2\ \ 0\ \ 1].$$

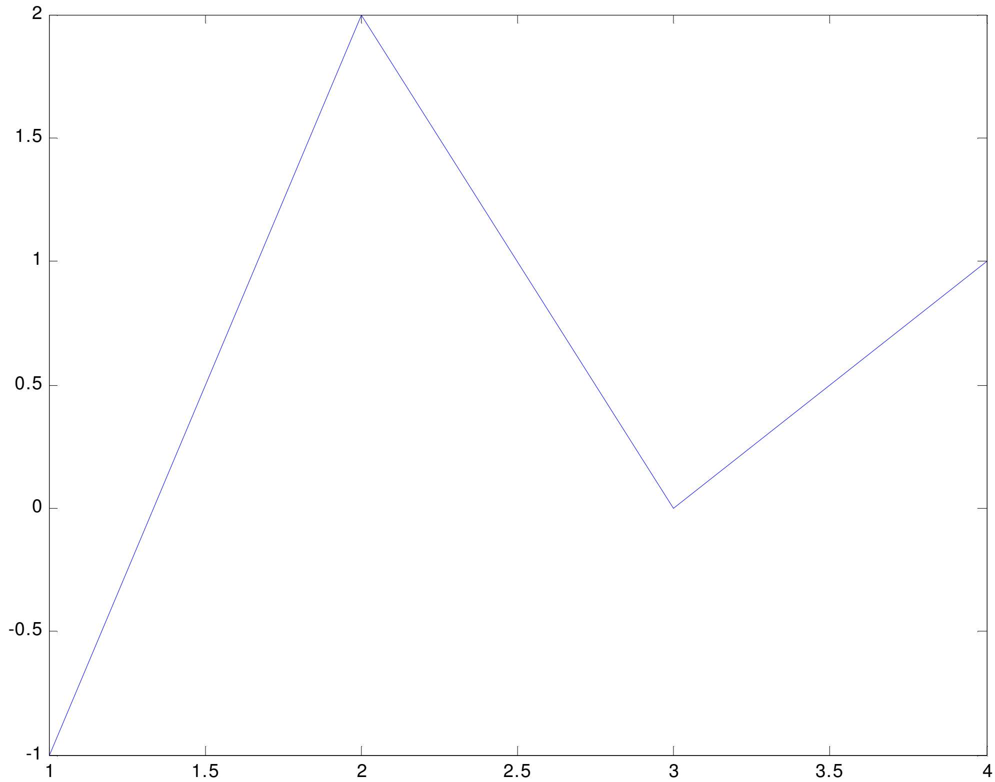
Najprije u novomu retku komandnoga prozora generiramo matricu A utipkavanjem:

```
A=[-1 2 0 1]
```

Pritisnemo *Enter*, pa u novomu retku komandnoga prozora utipkamo:

```
plot(A)
```

Dobivamo sljedeći grafikon otvoren u novomu prozoru (vidjeti Sliku 1.):



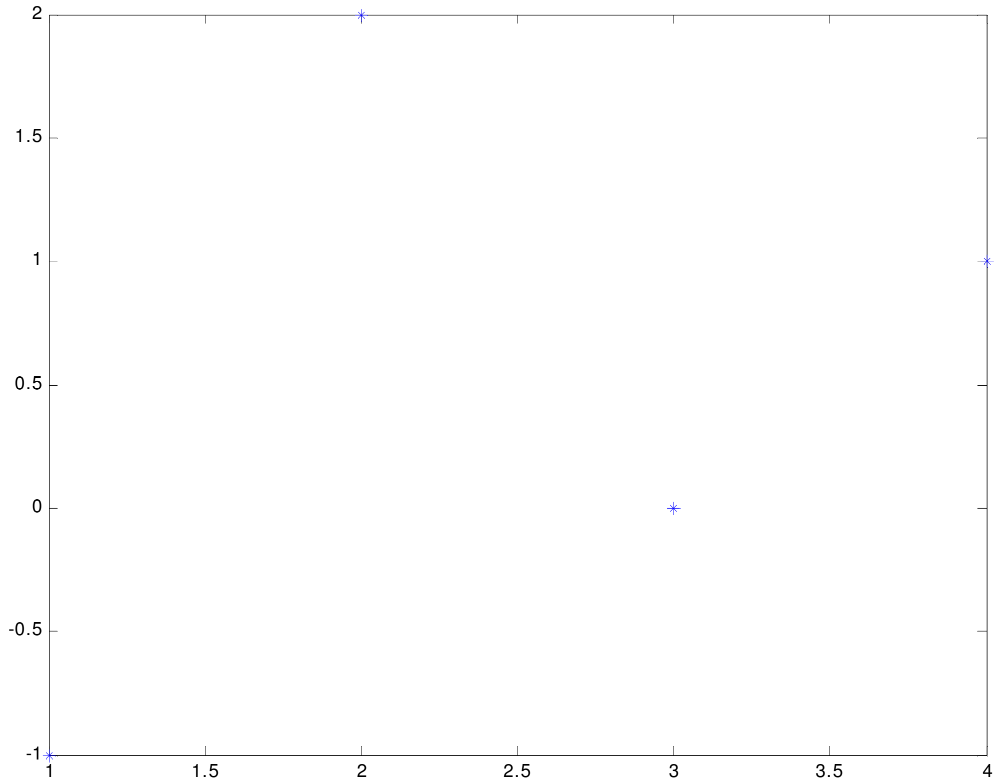
Slika 1.

Uočimo kako je dobivena ova izlomljena crta. Najprije su ucrtane točke (1,–1), (2,2), (3,0) i (4,1). Potom su crtom spojene točke (1,–1) i (2,2), (2,2) i (3,0), te (3,0) i (4,1). Novootvoreni prozor zatvaramo tako da najprije pomicanjem miša po podlozi premjestimo pokazivač u gornji desni kut na ikonicu s oznakom ×, a potom jednim klikom lijeve tipke miša na tu ikonicu zatvorimo prozor i vratimo se u "stari" komandni prozor. (Onima koji više vole tipkovnicu preporučamo istodobno pritiskanje tipki `Ctrl` i W.)

Ako ne želimo crtom spojiti susjedne točke, to moramo "reći" MATLAB-u. Utipkajmo u novi redak komandnoga prozora

```
plot(A,'*')
```

pa ćemo u novootvorenom prozoru dobiti sljedeću sliku (vidjeti Sliku 2.):



Slika 2.

Primjer 2. "Nacrtajmo" u MATLAB-u sljedeću matricu:

$$B = \begin{bmatrix} -1 & 2 & -3 \\ 4 & -5 & 6 \\ -7 & 8 & -9 \end{bmatrix}$$

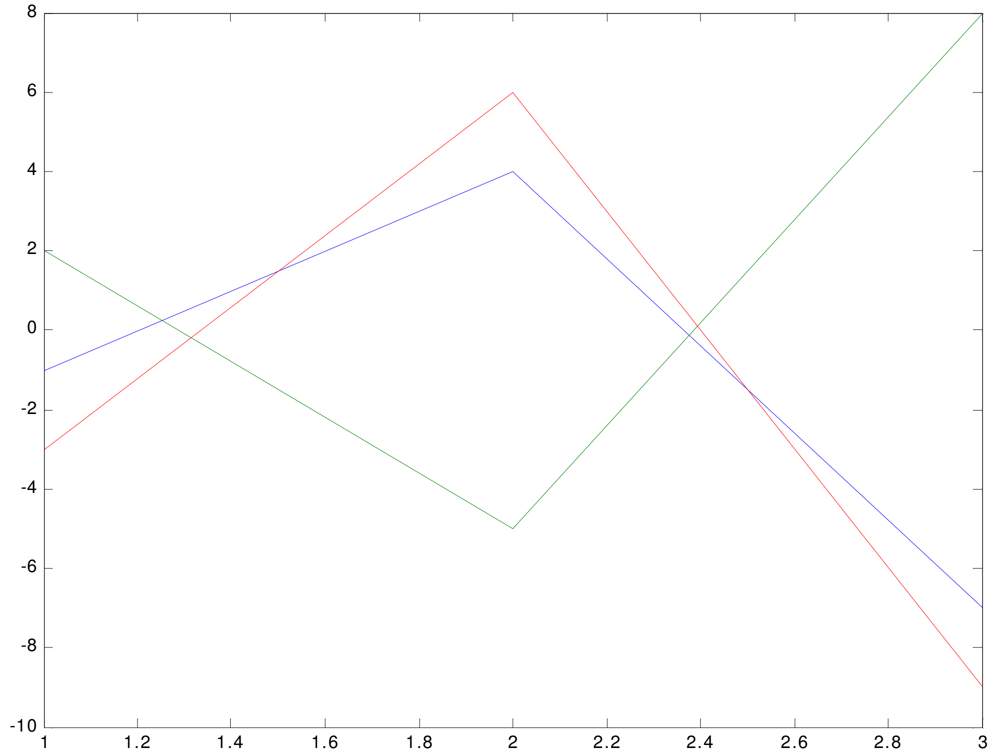
i spojimo susjedne točke izlomljenom crtom. Zadanu matricu najprije generiramo u MATLAB-u utipkavanjem:

```
B=[-1 2 -3;4 -5 6;-7 8 -9]
```

pritisnemo *Enter* i u novomu retku komandnoga prozora utipkamo jednostavno:

```
plot(B)
```

Dobijemo sljedeći skup linijskih grafikona (vidjeti Sliku 3.):



Slika 3.

Objasnimo kako je dobivena npr. crvena izlomljena crta. Ona je nastala spajanjem točaka (1,–3) i (2, 6), te (2, 6) i (3, –9). Te su točke dobivene pomoću elemenata 3. stupca matrice *B*. Npr.: element –3 stoji na presjeku toga stupca i 1. retka, što znači da je i = 1 (varijabla i označava redak u kojemu se nalazi element a_{ij}), pa se (u Koraku 1. algoritma funkcije *plot*) dobije točka (1, –3). Potpuno analogno se dobiju i ostale dvije točke promatrane crte.

3.2. Crtanje grafova funkcija na segmentu

"Prava" korist funkcije *plot* nije u grafičkom prikazivanju matrica, nego u crtanju grafova realnih funkcija. Odmah se mora napomenuti nešto što se vrlo često previdi ili zaboravi:

***Pravilo*: U MATLAB-u se grafički prikazuju isključivo realne funkcije definirane na segmentima.**[6]

[6] Navedeno se pravilo odnosi isključivo za funkciju `plot` jer postoje i druge funkcije MATLAB-a (npr. `ezplot`) pogodne za crtanje grafova funkcija više varijabli kojima domena nije segment. S druge strane, koristeći varijablu NaN realnu funkciju možemo crtati npr. i na uniji intervala (vidjeti Primjer 2.).

To se posebno odnosi na realne funkcije realne varijable kojima je područje definicije cijeli skup **R**. **Nije moguće** nacrtati graf takve funkcije na cijelom skupu **R** jer taj skup nema niti najmanji niti najveći element. Uostalom, kada "ručno" crtate grafove funkcija u svojim bilježnicama, nesvjesno se opredijelite za određeni dio ravnine (u pravilu onaj koji sadrži ishodište pravokutnoga koordinatnoga sustava u ravnini) pa crtate traženi graf uzimajući vrijednosti nezavisne varijable iz toga područja. Isto to radi i MATLAB, samo mu moramo "reći" koje područje ravnine želimo uzeti.

U crtanju funkcija glavnu "ulogu" imaju matrice. I kada "ručno" crtamo grafove, mi najprije odredimo parove točaka kroz koje prolazi traženi graf i zapišemo ih tablično. Da MATLAB-u ne bismo morali neprekidno "govoriti" što treba uzeti za x kako bi izračunao $f(x)$, koristimo *jednoretčane matrice čiji elementi tvore aritmetički niz*. (Generiranje takvih matrica naučili smo u prethodnom poglavlju (vidjeti točku 2.4.).) Prvi član toga niza je donja granica segmenta, a posljednji gornja granica segmenta. Kako bi slika bila što bolja, za razliku niza pogodno je uzeti "mali" decimalan broj (npr. 0.01, 0.005 i sl.). Ne istaknemo li drugačije, dogovorno pretpostavljamo da je spomenuta razlika jednaka 0.01.

Crtanje grafa funkcije na segmentu radi se algoritamski u tri koraka:

***Korak* 1**. Rabeći generiranje prema razmaku među susjednim elementima generira se matrica čiji je prvi član donja granica segmenta, razlika (dogovorno) jednaka 0.01, a posljednji član gornja granica segmenta.

***Korak* 2**. Generira se matrica vrijednosti zadane funkcije u svakome elementu matrice dobivene u Koraku 1.

***Korak* 3**. Svakom elementu dobivenomu u Koraku 1. pridruži se vrijednost funkcije dobivena u Koraku 2., te se dobiveni uređeni par ucrta u pravokutni koordinatni sustav u ravnini.

Pokažimo primjenu ovoga algoritma na primjerima.

***Primjer* 1.** Nacrtajmo graf realne funkcije $f : [-6, 6] \to \mathbf{R}$ definirane propisom:

$$f(x) = \frac{2 \cdot x}{x^2 + 1}.$$

Slijedimo korake gore opisanoga algoritma. Prvi je korak generiranje "matrice x-eva". Budući da MATLAB dozvoljava da ime matrice bude i malo tiskano slovo, tu ćemo matricu označiti sa x. Njezin je prvi član –6, razlika 0.01, a posljednji član 6. Dakle, u novi redak komandnoga prozora utipkavamo:

```
x=-6:0.01:6;
```

Uočimo ovdje ključnu ulogu znaka točka-zarez. Da ga nismo stavili na kraj retka, MATLAB bi ispisao svih 1200 elemenata matrice x i dobili bismo nepotrebnu „gužvu“ u MATLAB-ovu komandnomu prozoru. Zbog toga ga uvijek valja staviti na kraj retka.

U novomu retku komandnoga prozora formiramo matricu vrijednosti zadane funkcije, odnosno "matricu y-a". Nju dobivamo računski množenjem, dijeljenjem i potenciranjem članova matrice x **<u>član po član</u>**. (Ovo dobro zapamtite jer se pri crtanju grafova često griješi zamijenjujući operacije "član po član" s "klasičnim" aritmetičkim operacijama. Zato: **<u>oprez</u>**!) Utipkavamo:

```
y=2*x./(x.^2+1);
```

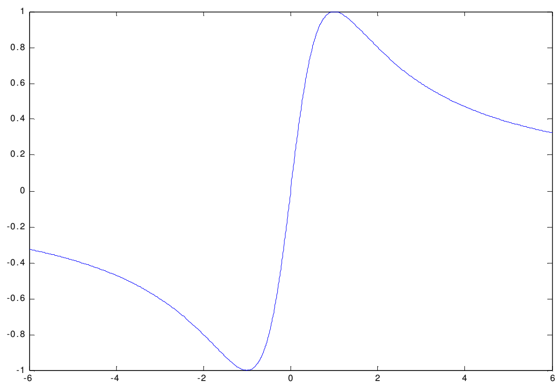
(Podsjetimo se: Množenje, odnosno zbrajanje matrice sa skalarom član po član podudara se s "običnim" množenjem, odnosno zbrajanjem sa skalarom.) I opet smo na kraju retka stavili točku-zarez kako ne bismo dobili ispis svih elemenata matrice y.

<u>Napomena</u>: U MATLAB-u izraz `x.^2+1` zapravo znači dodavanje matrice [1 1 … 1] matrici `x.^2`, pa se time izbjegava problem s dimenzijama matrica.

Preostaje nam svakome od 1200 x-eva pridružiti pripadni y, pa dobivenu točku ucrtati u pravokutni koordinatni sustav u ravnini. U novi redak komandnoga prozora utipkavamo:

```
plot(x,y)
```

pa dobivamo sljedeći graf (vidjeti Sliku 4.):



Slika 4.

<u>Primjer 2.</u> U Primjeru 1. zadatak nam je bio nacrtati grafički prikaz *neprekidne* funkcije. No, što će se dogoditi ako funkcija ima prekid (1. ili 2. vrste) na segmentu iznad kojega je crtamo? Npr. uzmimo realnu funkciju $f : [-6, 6] \to \mathbf{R}$ definiranu propisom:

$$f(x) = \frac{2 \cdot x}{x^2 - 1}.$$

Iole bolji znalci matematike odmah će uočiti da će tu biti dva problema: Kako odrediti vrijednosti funkcije za $x = -1$ i $x = 1$ kad u tim točkama funkcija uopće nije definirana? "Spas" je u varijabli *NaN*: ako postoji x za koji ne postoji pripadni $f(x)$, MATLAB će tome x-u kao $f(x)$ dodijeliti "vrijednost" *NaN*, a dobivenu "točku" neće ucrtati u koordinatni sustav.

Nakon što u svaki redak zasebno utipkamo

```
x=-6:0.01:6;
y=2*x./(x.^2-1);
```

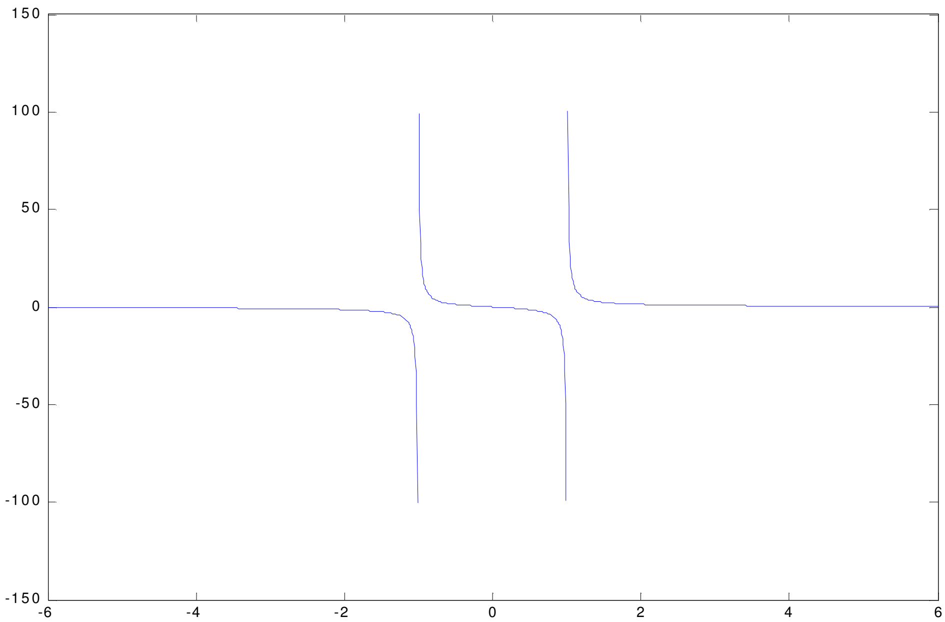
MATLAB će nas upozoriti da smo u jednom ili više slučajeva dijelili s nulom (ali neće "reći" u koliko smo slučajeva napravili taj, s matematičkoga stajališta, "smrtni grijeh"):

```
Warning: Divide by zero.
```

Nakon što u sljedećemu retku utipkamo

```
plot(x,y)
```

u novootvorenome ćemo prozoru dobiti sljedeći graf (vidjeti Sliku 5.):



Slika 5.

Uočimo da se iz grafa odmah mogu vidjeti prekidi u točkama $x = -1$ i $x = 1$, te da su ti prekidi 2. vrste (ne mogu se ukloniti jer očito ne postoje granične vrijednosti zadane funkcije u navedenim dvjema točkama).

U prethodnim smo primjerima na jednoj slici prikazivali graf samo jedne funkcije. No, prigodom grafičkoga rješavanja nelinearnih jednadžbi iznimno je važno na istoj slici prikazati grafove dviju ili više funkcija (kako bi se približno mogla očitati sjecišta tih grafova) definiranih *na istom segmentu*. (Takve ćemo probleme razmatrati i rješavati u 10. poglavlju.) Tu se ponovno javlja funkcija *plot*, ali ovoga puta s malo drugačijom sintaksom. U praksi se vrlo često zaboravi da se grafovi funkcija mogu crtati na istoj slici pomoću funkcije *plot* ako i samo ako su sve realne funkcije definirane <u>na istom segmentu</u> pa imajte to na umu prigodom rabljenja ove funkcije.

Ovu, malo složeniju primjenu funkcije *plot* pokazat ćemo na konkretnom primjeru.

<u>Primjer 3.</u> Na istoj slici nacrtajmo grafove funkcija $f(x) = 2 \cdot x + 1$, $g(x) = 1 - x$ i $h(x) = \ln x$ na segmentu $[1, 20]$. Najprije generiramo matricu x-eva:

```
x=1:0.01:20;
```

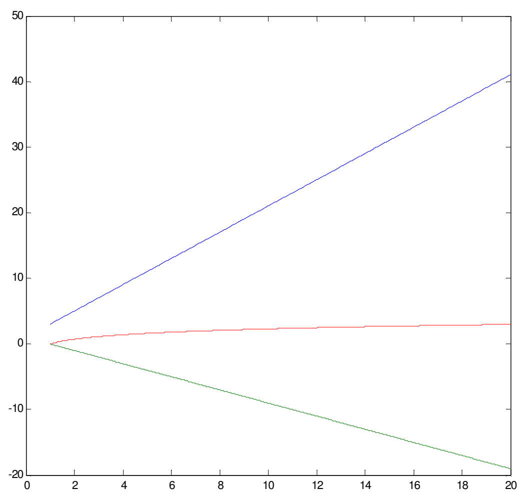
a potom i tri matrice $y1$, $y2$ i $y3$ s odgovarajućim vrijednostima funkcija:

```
y1=2*x+1;         (matrica vrijednosti funkcije f)
y2=1-x;           (matrica vrijednosti funkcije g)
y3=log(x);        (matrica vrijednosti funkcije h)
```

Sve tri krivulje prikazat ćemo na istoj slici utipkavanjem:

```
plot(x,y1,x,y2,x,y3)
```

(Uočite da su argumenti funkcije *plot* ovdje odvojeni zarezima.) U novootvorenom ćemo prozoru dobiti sljedeću sliku (vidjeti Sliku 6.):



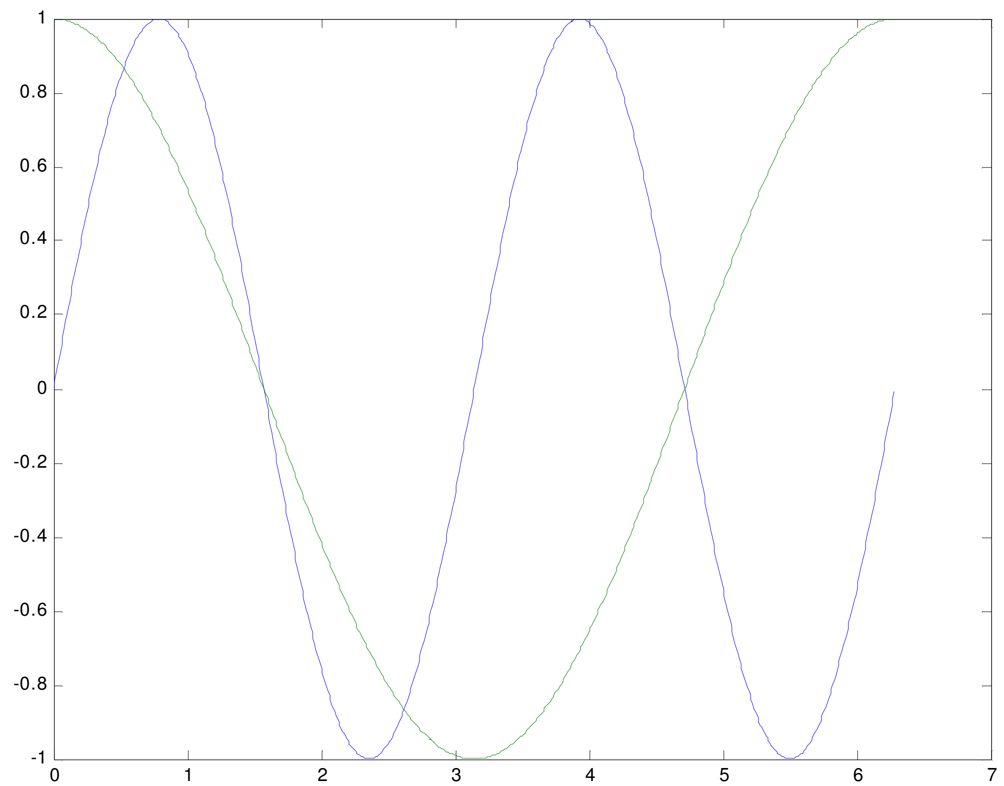
Slika 6.

Plava je krivulja graf funkcije $f(x)$, crvena graf funkcije $h(x)$, a zelena graf funkcije $g(x)$. Uočimo da inače "brza" logaritamska funkcija ovdje prividno "sporo" raste jer jedinična dužina na osi Oy ima vrlo malenu duljinu. Takva je duljina odabrana kako bi se mogle prikazati relativno velike vrijednosti funkcije $f(x)$ u točkama blizu gornjega kraja segmenta.

***Primjer* 4.** Grafički riješimo nejednadžbu $\sin(2 \cdot x) > \cos x$ na intervalu $[0, 2 \cdot \pi]$. Na istoj ćemo slici nacrtati grafove funkcija $f(x) = \sin(2 \cdot x)$ i $g(x) = \cos x$, pa ćemo očitati iznad kojih je dijelova zadanoga intervala graf funkcije $f(x)$ iznad grafa funkcije $g(x)$. Utipkavamo redom:

```
x=0:0.01:2*pi;
y1=sin(2*x);
y2=cos(x);
plot(x,y1,x,y2)
```

pa dobijemo sljedeću sliku (vidjeti Sliku 7.):



Slika 7.

Plava krivulja je graf funkcije $f(x)$, a zelena graf funkcije $g(x)$. Intervale na kojima je plava krivulja iznad zelene možemo očitati samo približno. Dva su takva intervala: $\langle 0.6,\ 1{,}6\rangle$ i $\langle 2.6\ ,\ 4.6\rangle$.

Riješimo li zadanu nejednadžbu "klasično", dobit ćemo:

$$x \in \left\langle \frac{\pi}{6}, \frac{\pi}{2} \right\rangle \cup \left\langle \frac{5}{6}\cdot\pi, \frac{3}{2}\cdot\pi \right\rangle,$$

odnosno približno

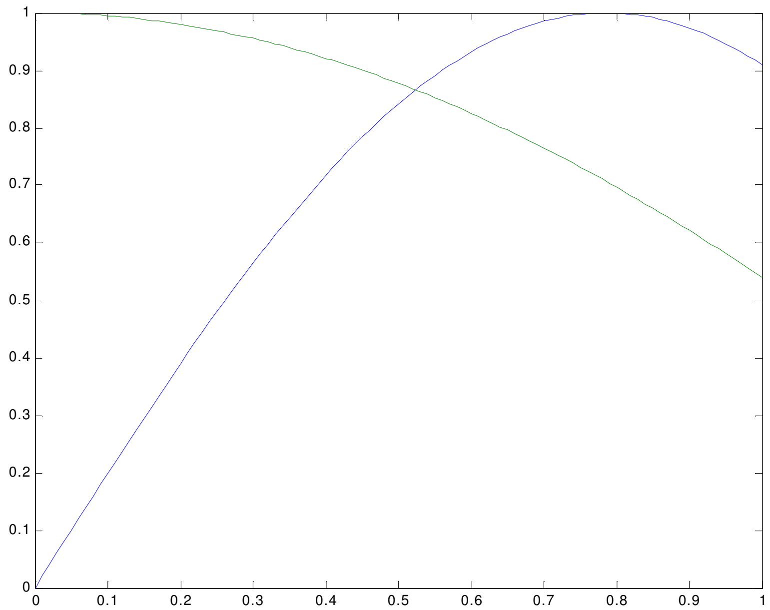
$x \in \langle 0.5235987755983,\ 1.5707963267949\rangle \cup \langle 2.61799387799149,\ 4.71238898038469\rangle$.

Naravno da u praksi nema potrebe "pogađati" granice intervala, nego se postupa npr. ovako:

Sa slike vidimo da prvi od intervala - rješenja nejednadžbe ima donju granicu u segmentu [0,1]. Stoga nacrtamo grafove promatranih funkcija u tom segmentu. Utipkamo:

```
x=0:0.01:1;
y1=sin(2*x);
y2=cos(x);
plot(x,y1,x,y2)
```

i dobit ćemo sljedeću sliku (vidjeti Sliku 8.):

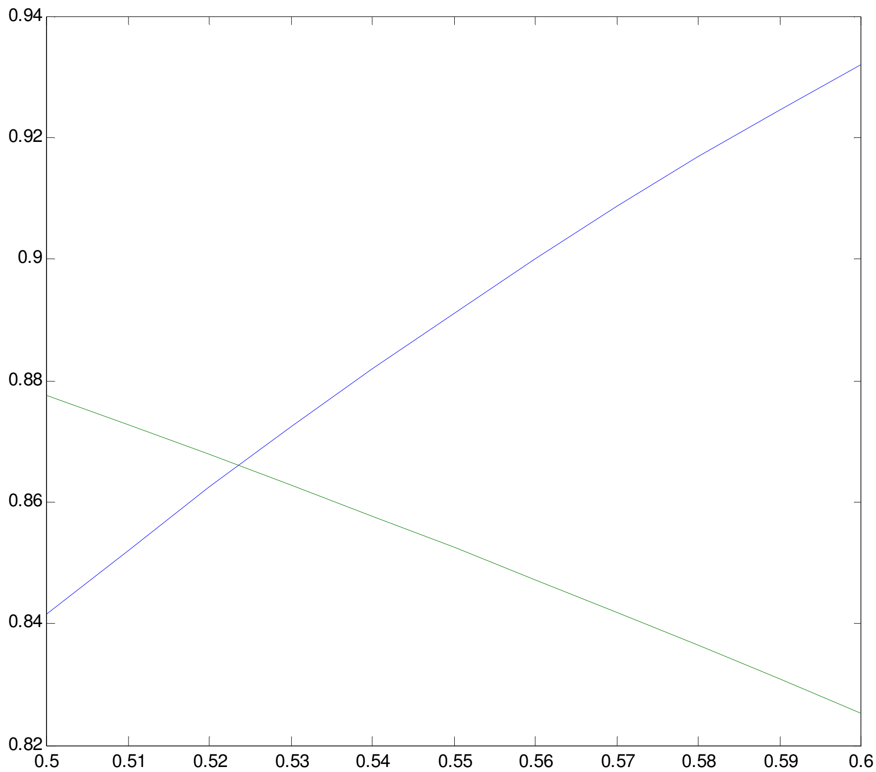


Slika 8.

S ove je slike očito da je tražena donja granica u intervalu [0.5, 0.6]. Nacrtajmo grafove promatranih funkcija i u tom segmentu. Utipkamo:

```
x=0.5:0.01:0.6;
y1=sin(2*x);
y2=cos(x);
plot(x,y1,x,y2)
```

i dobit ćemo sljedeću sliku (vidjeti Sliku 9.):



Slika 9.

Sad traženu donju granicu možemo odrediti još preciznije: ona se nalazi u segmentu [0.52, 0.53]. Ovisno o tome koliku točnost određivanja donje granice želimo, ponavljamo opisani postupak uzimajući svaki put sve "kraće" intervale. Potpuno analogno se dobivaju i ostale tri granice intervala – rješenja nejednadžbe.

Grafove realnih funkcija jedne realne varijable možemo crtati i koristeći funkciju `ezplot` čija je sintaksa ponekad i jednostavnija od sintakse funkcije `plot`. Pogledajmo nekoliko primjera.

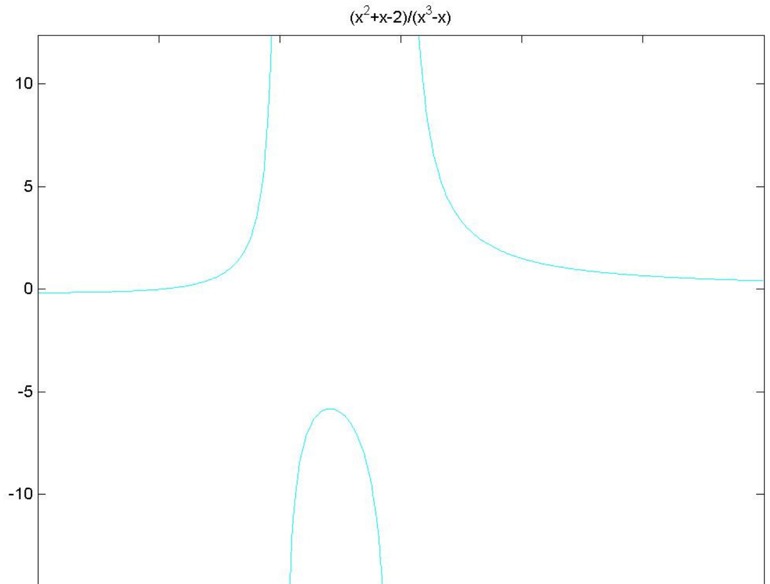
Primjer 5. Nacrtajmo graf prave racionalne funkcije $f(x) = \frac{x^2 + x - 2}{x^3 - x}$ na segmentu [–3, 3]. Klasificirajmo sve točke prekida prema uklonjivosti prekida, pa provjerimo svoje rješenje koristeći dobiveni graf.

Ovaj zadatak riješit ćemo pomoću funkcije `ezplot`. U novi redak MATLAB-ova komandnoga prostora utipkajmo:

```
ezplot('(x^2+x-2)/(x^3-x)',[-3,3])
```

Objasnimo sintaksu funkcije `ezplot`. Unutar okruglih zagrada pod jednostrukim navodnicima pišemo propis funkcije (pri čemu množenje i potenciranje pišemo uobičajenim znakovima), potom stavljamo zarez, pa iza zareza u uglatim zagradama navodimo segment iznad kojega crtamo graf funkcije.

Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku:



Slika 10.

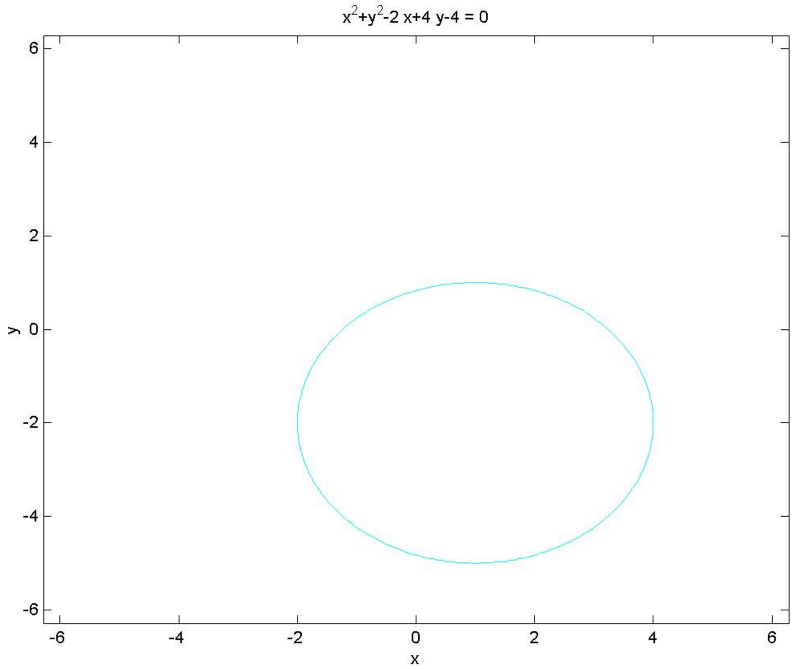
Analitički (rješavanjem jednadžbe $x^3 - x = 0$) lako nalazimo da funkcija ima prekid u točkama $x_1 = -1$, $x_2 = 0$ i $x_3 = 1$. Da u prvim dvjema točkama (x_1 i x_2) funkcija ima prekid, očito je i iz gornje slike. Međutim, iz gornje slike nije očito da funkcija ima prekid u točki $x_3 = 1$. Ta neočiglednost je posljedica činjenice da je prekid u točki $x_3 = 1$ uklonjiv jer je $x_3 = 1$ jednostruka nultočka i brojnika i nazivnika zadane funkcije. Stoga možemo zaključiti da su prekidi funkcije u prvim dvjema točkama neuklonjivi, dok je prekid u trećoj točki uklonjiv.

Primjer 6. Nacrtajmo krivulju implicitno zadanu jednadžbom $x^2 + y^2 - 2 \cdot x + 4 \cdot y = 4$. O kojoj je ravninskoj krivulji riječ?

Jedna od prednosti funkcije `ezplot` jest mogućnost crtanja algebarskih krivulja zadanih implicitnim jednadžbama. Tako u novi redak komandnoga prostora utipkamo:

```
ezplot('x^2+y^2-2*x+4*y-4')
```

Ovakva je sintaksa potrebna jer u slučaju implicitno zadanih funkcija MATLAB zahtijeva da desna strana pripadnoga analitičkoga izraza bude jednaka nuli. Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku:

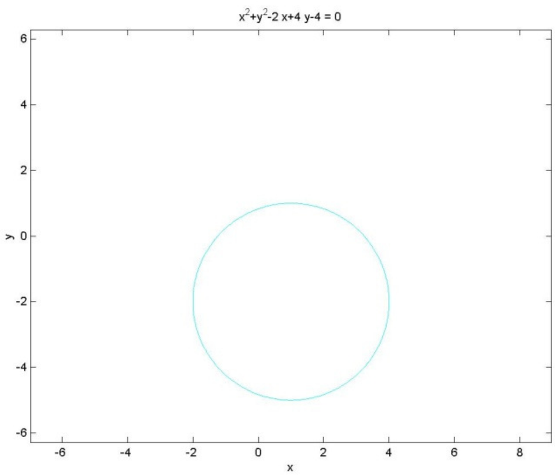


Slika 11.

Iako sa slike nije baš razvidno, dobivena krivulja je kružnica. Analitički se lako pokazuje da je riječ o kružnici sa središtem u točki $S = (1, -2)$ i polumjerom $r = 3$. (Učinite to!) Kako ipak postići da bude razvidnije da je riječ o kružnici? Odgovor je jednostavan: treba podesiti da na osima budu ista mjerila. Stoga u novom retku radnoga prostora utipkajmo:

```
ezplot('x^2+y^2-2*x+4*y-4'); axis equal
```

(Za „neengleski“ nastrojene: `axis equal` znači „jednake osi“.) Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku (iz koje je bitno razvidnije da se radi o kružnici):



Slika 12.

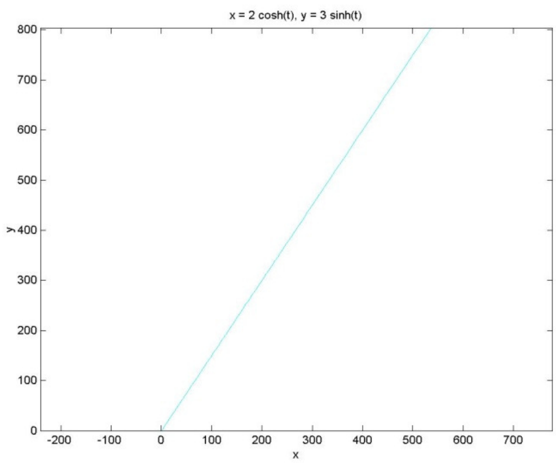
Da funkcija `ezplot` ima i svoje nedostatke, uvjerit će nas sljedeći primjer:

Primjer 7. Nacrtajmo krivulju parametarski zadanu s $\begin{cases} x = 2 \cdot \text{ch}\, t, \\ y = 3 \cdot \text{sh}\, t. \end{cases}$ O kojoj je ravninskoj krivulji riječ?

Funkcija `ezplot` može crtati i krivulje zadane parametarski. Prigodom poziva te funkcije najprije upisujemo izraz za x, potom izraz za y, a na kraju (neobavezno) segment kojemu pripada parametar t. Stoga u novi redak komandnoga prostora utipkamo:

```
ezplot('2*cosh(t)','3*sinh(t)')
```

Pritisnemo *Enter*, pa ćemo, na prilično neugodno iznenađenje, dobiti sljedeću sliku:

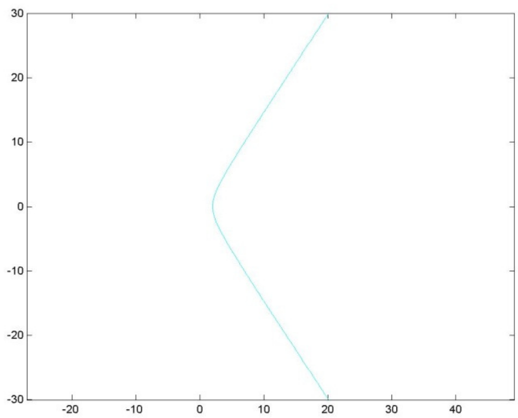


Slika 13.

Iz navedene slike mogli bismo zaključiti da je riječ o pravcu, što očito nije točno jer veza varijabli x i y nije linearna. MATLAB je ovdje prividno pogriješio isključivo zbog velikoga mjerila na objema koordinatnim osima koje ne daje dovoljno preciznu sliku. Stoga primjenom funkcije `plot` i crtanjem krivulje na segmentu [–3, 3]

```
t=-3:0.01:3;
x=2*cosh(t);
y=3*sinh(t);
plot(x,y);axis equal
```

dobivamo nešto jasniju sliku:



Slika 14.

Sad već možemo naslutiti da se radi o jednoj grani hiperbole. To je točno: riječ je o hiperboli čija je velika (tzv. *realna*) poluos a = 2, a mala (tzv. *imaginarna*) poluos b = 3. Ništa bolju sliku ne bismo dobili niti da smo koristili trigonometrijsku parametrizaciju hiperbole

$$\begin{cases} x = \dfrac{2}{\cos t}, \\ y = 3 \cdot \operatorname{tg} t. \end{cases}$$

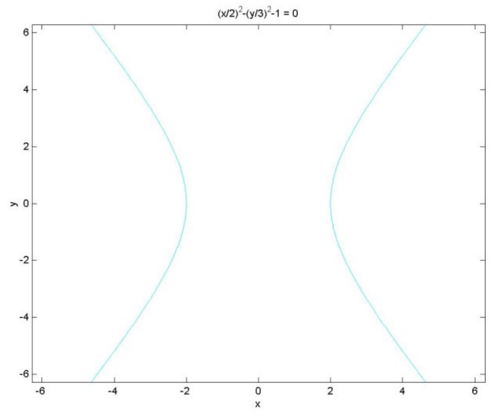
Zapišemo li njezinu jednadžbu u uobičajenom implicitnom obliku

$$\left(\frac{x}{2}\right)^2 - \left(\frac{y}{3}\right)^3 = 1$$

i nacrtamo li tako dobivenu krivulju pomoću funkcije `ezplot`

```
ezplot('(x/2)^2-(y/3)^2-1')
```

dobit ćemo krivulju koju smo i očekivali:



Slika 15.

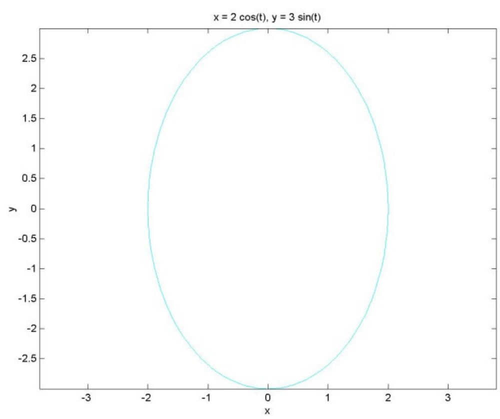
Primijetimo da iz gornje slike izravno možemo odrediti i raspone vrijednosti varijabli x i y. Očito je $x \in \mathbf{R} \setminus \langle -2, 2 \rangle$ i $y \in \mathbf{R}$. (Provjerite dobivene rezultate analitički!).

Primjer 8. Nacrtajmo krivulju parametarski zadanu s $\begin{cases} x = 2 \cdot \cos t, \\ y = 3 \cdot \sin t. \end{cases}$ O kojoj ravninskoj krivulji je riječ? Koristeći dobiveno rješenje odredimo raspon vrijednosti varijabli x i y.

Analogno kao u Primjeru 7. utipkamo:

```
ezplot('2*cos(t)','3*sin(t)')
```

Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku:



Slika 16.

Iz slike vidimo da se radi o elipsi. Duljina velike osi elipse je $2 \cdot a = 4$, a duljina male osi $2 \cdot b = 6$. Također, iz slike je očito da je $x \in [-2, 2]$ i $y \in [-3, 3]$. Provjerite dobivene rezultate i analitički.

3.3. Zadatci za vježbu

1. U MATLAB-u generirajte matrice

$$A = [1 \quad -1 \quad 2], \quad B = \begin{bmatrix} -2 \\ 1 \\ -3 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 3 \\ 6 & -4 \end{bmatrix} \text{ i } D = \begin{bmatrix} 1 & 0 & -1 \\ -2 & \frac{1}{3} & \sqrt{2} \\ 0 & -\sqrt[3]{4} & 0.3 \end{bmatrix},$$

pa odredite rezultate primjene funkcije `plot` na svaku od njih. Za svaku krivulju objasnite kako je dobivena.

2. Nacrtajte grafove funkcija:

a) $f(x) = \sin x$ na segmentu [–1.5, 1.5];
b) $f(x) = \ln x$ na segmentu [0.5, 2];
c) $f(x) = \dfrac{x}{x^2+1}$ na segmentu [–1, 1];
d) $f(x) = \dfrac{2 \cdot x}{\sqrt{x+2}}$ na segmentu [2, 7];
e) $f(x) = \dfrac{x}{e^x}$ na segmentu [–2, 3];
f) $f(x) = \dfrac{e^x}{x^2+1}$ na segmentu [–1, 2].

3. Na istoj slici nacrtajte grafove svih funkcija iz 2. zadatka na segmentu [0, 2].

4. **a)** Na istoj slici nacrtajte sinusoidu i kosinusoidu na segmentu $[0, 2 \cdot \pi]$. Pomoću dobivene slike s točnošću od 10^{-2} riješite nejednadžbu

$$\sin x > \cos x$$

Provjerite ispravnost svojega rješenja analitičkim rješavanjem zadane nejednadžbe.

b) Utvrdite možete li na intervalu iz **a)** zadatka nacrtati tangensoidu i kotangensoidu. Objasnite što se dogodilo!

5. S točnošću od 10^{-2} riješite nejednadžbu

$$1 - \sin x \leq \cos^2 x$$

na segmentu $[0, 4 \cdot \pi]$. Provjerite ispravnost svojega rješenja analitičkim rješavanjem zadane nejednadžbe.

6. S točnošću od 10^{-2} odredite sve zajedničke točke krivulja:

a) $y = x^3$ i $y = x - 1$;

b) $y = \ln x$ i $y = \dfrac{1}{x}$;

c) $y = \sin x$ i $y = e^x - 1$;

d) $y = \arcsin x$ i $y = \ln(x + 1) + \dfrac{1}{3}$;

e) $y = e^{-x^2}$ i $y = x + 1$;

7. Nacrtajte ravninske krivulje implicitno zadane sljedećim jednadžbama i, ako nije navedeno, utvrdite o kojim je krivuljama riječ:

a) $x^2 + y^2 + 8 \cdot x - 6 \cdot y = 0$;
b) $x^2 + y^2 - 6 \cdot x + 10 \cdot y = 2$;
c) $25 \cdot x^2 - 9 \cdot y^2 = 144$;
d) $25 \cdot x^2 + 9 \cdot y^2 = 144$;
e) $y^2 = 8 \cdot x$;
f) $\sqrt[3]{x^2} + \sqrt[3]{y^2} = 1$ (tzv. *astroida*) .

8. Nacrtajte ravninske krivulje parametarski zadane sljedećim jednadžbama i, ako nije navedeno, utvrdite o kojim je krivuljama riječ:

a) $\begin{cases} x = t^3, \\ y = t^2; \end{cases}$ (tzv. *semikubna parabola*);

b) $\begin{cases} x = 2 \cdot \cos t, \\ y = 2 \cdot \sin t; \end{cases}$

c) $\begin{cases} x = 3 \cdot \cos t, \\ y = 2 \cdot \sin t; \end{cases}$

d) $\begin{cases} x = 4 \cdot \cos^3 t, \\ y = 4 \cdot \sin^3 t; \end{cases}$ (tzv. *astroida*)

e) $\begin{cases} x = \dfrac{1}{\sqrt{1+t^2}}, \\ y = \dfrac{t}{\sqrt{1+t^2}}; \end{cases}$

f) $\begin{cases} x = 4 \cdot \cos t - 2 \cdot \cos(2 \cdot t), \\ y = 4 \cdot \sin t - 2 \cdot \sin(2 \cdot t). \end{cases}$ (tzv. *kardioida*)

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

§4. OSNOVE PROGRAMIRANJA U MATLAB-U

Već smo u uvodu naglasili da MATLAB nije samo programski paket namijenjen numeričkomu računanju i modeliranju, nego i viši programski jezik namijenjen raznim znanstvenim i tehničkim primjenama. U ovoj ćemo točki upoznati osnove programiranja u MATLAB-u i naučiti kako u njemu možemo sami pisati razne korisne (i manje korisne) program(čić)e.

Budući da se program najčešće sastoji od niza naredbi koji se u radu češće ponavlja, vrlo je nepraktično i nezgodno pisati taj niz naredbi u komandnomu prozoru. Zbog toga programe pišemo u posebne vrste datoteka – tzv. *m*-datoteke. Naučimo najprije kako stvoriti takve datoteke.

4.1. Kako stvoriti jednu običnu m-datoteku

Postupak stvaranja *m*-datoteke vrlo je jednostavan. Možemo ga provesti na dva načina, ovisno o tome jesmo li u prijateljskim odnosima s mišem svojega računala ili nismo:

1.) (za one koji vole miševe) **a)** Pomičući miš po podlozi postavimo pokazivač na natpis *File* blizu gornjega lijevoga kuta našega zaslona (odmah ispod natpisa `MATLAB`).
b) Jednom kliknemo lijevom tipkom miša na taj natpis pa se pojavi padajući izbornik s opcijama *New*, *Open*, *Close Command Window* itd.
c) Postavimo pokazivač na opciju *New*. Čim to učinimo, desno od natpisa *New* pojavit će se novi padajući izbornik koji nam nudi odabir četiriju opcija: *M-file*, *Figure*, *Model* i *GUI*.
d) Nas zanima opcija *M-file* pa ćemo pokazivač postaviti na taj natpis i jednom kliknuti lijevom tipkom miša. Tada će se otvoriti tekst-editor *Notepad* i u njemu ćemo pisati svoje naredbe.

2.) (za one koji ne vole miševe) Postupak opisan u **1.)** možemo napraviti i isključivo rabeći tipkovnicu. Redom kratko pritisnimo sljedeće tipke (svaku posebno):

`Alt, N, F, I`

(opcija *Caps Lock* na tipkovnici ne mora biti uključena jer MATLAB-u ovdje nije važno jesu li slova velika ili mala) i otvorit će se ranije spomenuti *Notepad*.

Uočimo da se trepćući pokazivač nalazi u retku ispred kojega piše broj 1. Broj ispred linije njezin je redni broj u datoteci, što nam vrlo često olakšava snalaženje u njoj. Upišimo u prvi redak te datoteke:

```
sin(pi/2)+cos(pi)*tan(3*pi/4)
```

Pohranimo upisani tekst pod imenom `proba.m` na sljedeći način:

***Korak* 1**. Istodobno pritisnemo tipke `Ctrl` i `S`. (Ili – ako više volimo miševe – postavimo pokazivač na natpis *File*, jednom kliknemo lijevom tipkom miša, potom postavimo pokazivač na natpis *Save* i ponovno jednom kliknemo lijevom tipkom miša. Što vam se čini brže?)

***Korak* 2**. U novootvorenome prozoru MATLAB nas najprije obavještava da će datoteku – ma kako god je nazvali – pohraniti u direktorij *C:\matlabR12\work*. Taj je direktorij dogovorni direktorij za pohranu svih tipova datoteka. Ako želimo datoteku pohraniti u neki drugi direktorij, tada pomicanjem miša po podlozi postavimo pokazivač na mali trokutić ▼ odmah pokraj natpisa *work*. Jednom kliknemo lijevom tipkom miša na taj trokutić i otvorit će nam se izbornik u kojemu klikovima na lijevu tipku miša sami formiramo stazu (engl. *path*) do direktorija u kojega želimo pohraniti svoju datoteku. Ovdje se nećemo odlučiti na tu mogućnost, nego ćemo datoteku pohraniti u direktorij *C:\matlabR12\work*.

Napomena: Ispod natpisa *Save in*: možemo vidjeti popis svih datoteka dosad pohranjenih u direktoriju *C:\natlabR12\work*.

***Korak* 3.** Ispod popisa datoteka nalazi se natpis *File name*: kraj kojega je bijeli pravokutnik s plavo obrubljenim slovima *Untitled* odmah do kojih je trepćući pokazivač. Naime, MATLAB predviđa mogućnost da postoji nemaštoviti korisnik programa bez ideje kako nazvati datoteku koju je upravo stvorio. Da takvomu korisniku cjelokupni dosadašnji trud ne bi otišao u nepovrat, MATLAB nudi mogućnost da stvorenu datoteku nazove *Untitled* (engl.: *bez naslova*). No, mi smo se već ranije odlučili da ćemo takvu datoteku nazvati `proba.m` pa pritisnimo strelicu ← koja se nalazi odmah iznad veće od dviju tipki *Enter* na našoj tipkovnici. Kad to učinimo, natpis *Untitled* će nestati i ostat će samo trepćući pokazivač.

5.) Upisujemo ime datoteke:

```
proba.m
```

***Važna napomena*:** MATLAB zahtijeva da pohranu imena datoteke zajedno s njezinom ekstenzijom. Budući da se ovdje radi o *m*-datoteci, ona ima kratku, ali jasnu ekstenziju: `.m` Zbog toga prigodom pohrane moramo upisati ime datoteke u obliku:

ime_datoteke.m

6.) Pritisnemo *Enter* ili jednom kliknemo lijevom tipkom miša na natpis *Save*. Datoteka je pohranjena, što nam potvrđuje natpis na vrhu zaslona:

```
C:\matlabR12\work\proba.m
```

Za izlaz iz upravo pohranjene datoteke možemo istodobno pritisnuti tipke `Ctrl` i `W` ili ponovo jednom kliknuti lijevom tipkom miša na natpis *File*, postaviti pokazivač na natpis *Close proba.m* i jednom kliknuti lijevom tipkom miša na taj natpis. U oba slučaja zatvaramo pohranjenu *m*-datoteku i vraćamo se u komandni prozor.

Učinak upravo provedenih radnji možemo vidjeti ako u novomu retku komandnoga prozora utipkamo:

```
proba
```

i pritisnemo *Enter*. Dobit ćemo:

```
ans =
      2
```

Što se dogodilo? MATLAB je pokrenuo datoteku s nazivom *proba.m*, izvršio sve naredbe zapisane u njoj kao da su zapisane u njegovu komandnomu prozoru i ispisao krajnji rezultat tih naredbi. Budući da vrijedi:

$$\sin\frac{\pi}{2}+\cos\pi\cdot\operatorname{tg}\frac{3\cdot\pi}{4}=1+(-1)\cdot(-1)=2,$$

rezultat jedine upisane naredbe u datoteci `proba.m` jest 2 i to je rezultat koji je ispisao MATLAB.

Ovakav je način zapisivanja i izvršavanja naredbi praktičan jer se lako mogu popravljati pogrješke *prije* nego li se program izvrši (za razliku od naredbi u komandnomu prozoru gdje se pogrješke otklanjaju tek nakon izvršavanja naredbe). Čak i ako pogriješimo, MATLAB će nas pristojno obavijestiti u kojemu je retku naše datoteke pronašao pogrješku, te nas obavijestiti o kakvoj se grešci radi, pa ćemo je moći ispraviti.

4.2. Funkcijske m–datoteke

M–datoteke vrlo su korisne i u stvaranju tzv. *funkcijskih datoteka*. To su datoteke kojima se nizom naredbi definira nova programska funkcija koja se pohranjuje u isti direktorij kao i obične *m*-datoteke (*C:\matlabR12\work*), ali se poziva drugačije nego obična *m*-datoteka. Njezina sintaksa je:

`function` [*izlazne varijable*]=*ime_funkcije*(*ulazne varijable*)
naredbe
`end`

Objasnimo ukratko o čemu se ovdje radi. Neka funkcija u MATLAB-u kao rezultat ne mora dati samo jednu, nego i više vrijednosti[7]. Zbog toga se sve te varijable moraju staviti u uglate zagrade i međusobno razdvojiti zarezima. Takva se funkcija u komandnomu prozoru poziva na sljedeći način:

[7] Postoje funkcije u MATLAB-u koje nemaju niti jednu ulaznu i/li izlaznu varijablu. Tipičan primjer takve funkcije je funkcija `clc`. Ona nema niti jednu ulaznu, odnosno izlaznu varijablu.

[*izlazne varijable*]=*ime_funkcije*(*ulazne varijable*)

***Važna napomena*:** Česta je pogrješka da se prigodom poziva funkcije uopće ne navedu, djelomično navedu ili redosljedno pogrešno navedu *ulazne* varijable. U takvim će slučajevima MATLAB odmah javiti pogrješku i neće izvršiti željenu funkciju. S izlaznim varijablama je malo drugačija priča jer ako ne navedemo ime izlazne varijable, MATLAB će joj sam dodijeliti ime *ans*. No, problem nastaje ako želimo više izlaznih varijabli: ne navedemo li im imena, MATLAB će ispisati točno jednu od njih. Zato: oprez!

Ako funkcija ima točno jednu izlaznu varijablu, ona se ne mora stavljati u uglate zagrade.

Ime funkcije **mora biti jednako** imenu funkcijske *m*-datoteke (bez ekstenzije). Dakle, ne smije se dogoditi da u datoteci imamo zapisano npr.

`function` y = kotangens(x)

a da funkcijsku datoteku nazovemo npr. *špajza.m*. Ako smo funkciju nazvali *kotangens*, onda i pripadnu funkcijsku *m*-datoteku moramo nazvati *kotangens.m* jer je u suprotnom pri njezinu pozivu MATLAB neće izvršiti.

Sama funkcija može imati jednu ili više ulaznih varijabli. Neovisno o njihovu broju, moraju se navesti unutar okruglih zagrada. Ako funkcija ima barem dvije ulazne varijable, one se odvajaju zarezima. Varijable, općenito, mogu biti brojevi i matrice. MATLAB razlikuje što je broj, a što matrica, pa se ne smije dogoditi da varijablu matričnoga tipa prigodom poziva pozovemo kao varijablu brojčanoga tipa.

Pogledajmo na primjeru kako stvoriti jednu "novu" funkciju u MATLAB-u.

***Primjer* 1.** Napišimo funkcijsku *m*-datoteku **zbrajanje.m** koja sadrži jedino funkciju *zbrajanje* čije su ulazne varijable dva realna broja, a koja kao rezultat vraća zbroj tih brojeva.

Otvorimo novu *m*-datoteku pa upišimo:

```
function z=zbrajanje(x,y)
z=x+y;
```

Pohranimo dobivenu datoteku pod nazivom *zbrajanje.m* u direktorij *C:\matlabR12\work*, pa je zatvorimo.

Uočimo što smo napravili: zbroj dvaju realnih brojeva je jedinstven realan broj, pa imamo točno jednu izlaznu varijablu koju smo označili sa *z*. Funkciji smo dali ime *zbrajanje* jer zadatak traži da se datoteka zove *zbrajanje.m*, a znamo da ime funkcije i ime datoteke moraju biti jednaki. Ulazne varijable označene su s *x* i *y*, te odvojene zarezima.

Kako pozvati ovakvu funkciju iz komandnoga prozora? Kao i svaku drugu ugrađenu funkciju, tako i funkciju pohranjenu u *m*-datoteci pozivamo navodeći izlazne varijable, ime funkcije i ulazne varijable. Za razliku od izlazne varijable kojoj će se vrijednost dodijeliti nakon što se izvrše sve naredbe u funkcijskoj *m*-datoteci, ulazne varijable moraju biti konkretne, odnosno njima se vrijednosti moraju dodijeliti prije poziva funkcije.

Želimo li zbrojiti npr. 123456789 i 987654321, te rezultat zapisati u varijablu *zbroj*, utipkat ćemo:

```
zbroj=zbrajanje(123456789,987654321)
```

i dobiti:

```
zbroj =
    1.111111110000000e+009
```

Ulazne varijable ne moraju nužno biti eksplicitno zadani (realni ili kompleksni) brojevi, nego mogu biti i vrijednosti nekih brojevnih izraza. U novi red komandnoga prozora utipkajmo:

```
zbroj=zbrajanje(conj(1-3*i),abs(1+sqrt(3)*i)/i)
```

i dobit ćemo:

```
zbroj =
  1.00000000000000 + 1.00000000000000i
```

Kako bismo uspješno mogli stvarati nove i složenije funkcije u MATLAB-u, upoznat ćemo još neke njegove naredbe.

4.3. Uvjetne naredbe (naredbe kontrole tijeka)

Zajednička karakteristika svih uvjetnih naredbi jest da izvode određeni skup naredbi sve dok je zadovoljen određeni uvjet, pri čemu se parametri toga uvjeta mijenjaju svaki put kad se izvede spomenuti skup naredbi. Tri su standardne uvjetne naredbe: *for* (engl: za), *while* (engl.: dok), *if.. else* (engl.: ako…inače) U nastavku ćemo upoznati svaku od njih.

4.3.1. Naredba for

Naredba *for* služi za ponavljanje niza naredbi unaprijed zadani broj puta. Budući da uvijek dolazi zajedno s tim nizom, u pravilu se govori o *for–petlji*. Njezina je sintaksa:

`for` *varijabla*=*izraz*1:*izraz*2:*izraz*3
niz naredbi
`end`

Prvi redak u ovoj sintaksi predstavlja zapis zahtjeva da *varijabla* poprima sve vrijednosti od *izraza*1 do *izraza*3 s korakom *izraz*2. Što to zapravo znači? Kad MATLAB izvršava ovu petlju, pri "prvom čitanju" on varijabli *varijabla* dodijeli vrijednost *izraz*1 i prelazi na izvršavanje cijeloga niza naredbi smještenoga u petlji. Kraj toga niza označava naredba *end*. Kad u "prvom čitanju" MATLAB "stigne" do naredbe *end*, vrati se ponovno na redak u kojemu piše naredba *for* i vrijednost varijable *varijabla* promijeni (uveća ili smanji) za vrijednost *izraz*2. Potom usporedi novodobivenu vrijednost varijable *varijabla* s vrijednošću *izraz*3. Ako je vrijednost varijable *varijabla* manja ili jednaka vrijednosti *izraz*3, MATLAB ponovno izvršava niz naredbi smješten (ili, kako se to stručno kaže, "ugniježđen") u petlji nakon čega se opet vraća na redak u kojemu je naredba *for*. Ako je vrijednost varijable *varijabla* veća od vrijednosti *izraz*3, petlja je završena pa se prelazi na prvu naredbu iza naredbe *end*.

Napomenimo da ako je vrijednost varijable *izraz*2 jednaka 1, ta se varijabla može izostaviti u sintaksi petlje. Također, ako iz bilo kojih razloga želimo prekinuti izvršavanje petlje, rabimo naredbu *break*. Želimo li, pak, proslijediti upravljanje petljom na sljedeći korak (odnosno, kako se to stručno kaže, „iteraciju") petlje, koristimo naredbu *continue*.

Pogledajmo primjenu ove petlje na primjerima.

***Primjer* 1.** Napišimo funkciju *zbroj* čija je jedina ulazna varijabla prirodan broj *n*, a jedina izlazna varijabla vrijednost zbroja prvih *n* prirodnih brojeva.

Označimo li izlaznu varijablu sa *z*, onda u prvomu retku naše funkcijske *m*-datoteke **zbroj.m** (ponovimo: naziv datoteke mora biti jednak nazivu funkcije!) zapisujemo:

```
function z=zbroj(n)
```

Na početku je vrijednost izlazne varijable *z* jednaka 0, pa to zapisujemo u drugomu retku:

```
z=0;
```

Traženi ćemo zbroj izračunati pomoću *for*–petlje. Za vrijednost varijable *izraz*1 uzet ćemo 1, za vrijednost varijable *izraz*3 uzet ćemo ulaznu varijablu *n*, a budući da moramo obuhvatiti sve prirodne brojeve od 1 do *n*, vrijednost varijable *izraz*2 mora biti jednaka 1, pa je izostavljamo. Sada pišemo naredbu *for*:

```
for m=1:n
```

Varijabla *varijabla* iz gornje sintakse ovdje je kraće označena slovom *m*. Ona će poprimati vrijednosti svih prirodnih brojeva od 1 do *n*, pa ćemo je iskoristiti za računanje traženoga zbroja:

```
z=z+m;
end.
```

Izvršite "ručno" ovu petlju i uvjerite se da je krajnja vrijednost izlazne varijable *z* uistinu jednaka zbroju prvih *n* prirodnih brojeva.

Pohranimo dobivenu *m*-datoteku, vratimo se u komandni prozor i provjerimo ispravnost svojega rada uzimajući $n = 100$. Pozovimo funkciju `zbroj` utipkavajući:

```
z=zbroj(100)
```

i dobit ćemo:

```
z =
        5050
```

Koristeći se formulom za računanje zbroja prvih *n* članova aritmetičkoga niza, provjerite "ručno" da je zbroj prvih 100 prirodnih brojeva uistinu jednak 5 050.

Primjer 2. Napišimo funkciju *faktorijel* čiji je jedina ulazna varijabla prirodan broj *n*, a jedina izlazna varijabla umnožak prvih *n* prirodnih brojeva[8].

Označimo tu izlaznu varijablu s *u*. Ideja za rješenje zadatka je slična kao u Primjeru 1., samo što na početku varijabli *u* dodijeljujemo vrijednost 1, a u petlji umjesto `z=z+m` stavljamo `z=z*m`. Stoga u funkcijsku *m*-datoteku **faktorijel.m** utipkavamo redom:

```
function u=faktorijel(n)
u=1;
for m=1:n
u=u*m;
end
```

Pohranimo upisane naredbe, vratimo se u komandni prozor i izračunajmo 40! utipkavajući:

```
u=faktorijel(40)
```

MATLAB će ispisati:

```
u =
    8.159152832478977e+047
```

Napomena: Za računanje vrijednosti $n!$ u MATLAB se koristi ugrađena funkcija `factorial`. Njezina jedina ulazna varijabla je nenegativan cijeli broj *n*, a jedina izlazna varijabla vrijednost $n!$. Za $n \geq 171$ funkcija kao rezultat vraća konstantu `Inf`.

Primjer 3. Napišimo funkciju *zp* čija je jedina ulazna varijabla prirodan broj *n*, a jedina izlazna varijabla zbroj prvih *n* parnih prirodnih brojeva.

[8] Taj se umnožak označava s $n!$ i čita *en faktorijela*. Prirodno područje definicije funkcije $f(n) = n!$ je skup $\mathbf{N}_0 := \{0, 1, 2, \ldots\}$. Dogovorno se uzima $0! = 1$.

Označimo traženi zbroj ponovno sa z. Prvi paran prirodan broj je 2, a kako ih mora biti ukupno n, posljednji od njih je $2 \cdot n$. Ideja rješavanja zadatka je slična kao u prethodnim primjerima, samo što se naredba *for* mora malo preraditi. U funkcijsku *m*-datoteku **zp.m** utipkavamo:

```
function z=zp(n)
z=0;
for m=2:2:(2*n)
z=z+m;
end
```

Pohranimo upisane naredbe i vratimo se u komandni prozor. Želimo li izračunati zbroj prvih 10 parnih prirodnih brojeva, u novomu retku komandnoga prozora upišimo:

```
z=zp(10)
```

i dobit ćemo:

```
z =
   110
```

Koristeći se formulom za računanje zbroja prvih n članova aritmetičkoga niza, provjerite "ručno" da je zbroj prvih 10 parnih prirodnih brojeva uistinu jednak 110.

4.3.2. Naredba while

Ova naredba koristi se prigodom ponavljanja niza naredbi sve dok je valjan određeni logički uvjet. Njezina sintaksa je

`while` *logički uvjet*
naredbe
`end`

Za "prisilni" izlaz iz petlje možemo rabiti naredbu `break`, dok za nastavak izvođenja petlje u njezinoj sljedećoj iteraciji možemo rabiti naredbu `continue`. Valja napomenuti da ako logički uvjet "u startu" nije valjan, niti jedna od naredbi unutar *while*-petlje niti jednom neće biti izvršena.

Pogledajmo uporabu ove naredbe na primjerima.

Primjer 4. Niz$(a_n)_{n\in\mathbf{N}}$ definiran je svojim općim članom

$$a_n = \frac{n+1}{n^4}.$$

Izračunajmo zbroj svih članova toga niza koji nisu strogo manji od konstante *eps*.

Primijetimo da je zadani niz strogo padajući (provjerite to!), pa članova koji nisu manji od konstante *eps* ima konačno mnogo. Stoga za rješavanje zadatka možemo rabiti petlje. Radi preglednosti, rješenje zadatka zapisat ćemo u obliku obične *m*–datoteke **niz1.m**. U tu datoteku utipkamo:

`s=0;`	inicijalizacija zbroja s
`n=1;`	inicijalizacija indeksa n
`while (n+1)/(n^4)>=eps`	uvjet da član niza mora biti barem jednak *eps*
`  s=s+(n+1)/(n^4);`	zbroj prvih n članova niza
`  n=n+1;`	povećanje indeksa za 1
`end`	kraj petlje
`s`	ispis traženoga zbroja

Pohranimo tu datoteku i vratimo se u komandni prozor. U njegovu novomu retku upišimo

```
niz1
```

pritisnimo *Enter* i dobit ćemo traženu vrijednost zbroja:

```
s =
   2.28438013685600
```

<u>Primjer 5.</u> Niz $(a_n)_{n\in\mathbf{N}}$ zadan je svojim općim članom

$$a_n = \frac{(-1)^{n+1}}{n!}$$

Izračunajmo zbroj svih članova toga niza koji po apsolutnoj vrijednosti nisu strogo manji od konstante *eps*.

U ovome ćemo zadatku vidjeti kako unutar neke petlje pozivamo zasebno definirane MATLAB-ove funkcije. U Primjeru 2. stvorili smo funkcijsku *m*–datoteku **faktorijel.m** koju ćemo iskoristiti pri rješavanju ovoga zadatka. Napomenimo da je zadatak moguće riješiti i korištenjem ranije spomenute „gotove“ MATLAB-ove funkcije `factorial`.

Radi preglednosti, rješenje zadatka zapisujemo u obliku obične *m*–datoteke **niz2.m** U tu datoteku utipkamo:

```
s=0;
n=1;
while abs((-1)^(n+1)/faktorijel(n))>=eps
  s=s+(-1)^(n+1)/faktorijel(n);
  n=n+1;
```

```
end
s
```

Zašto smo u ovom primjeru zahtijevali usporedbu apsolutnih vrijednosti članova niza s konstantom *eps*? Primijetimo da članovi niza mijenjaju predznak: prvi član je pozitivan, drugi negativan, treći opet pozitivan itd. Stoga bismo izostavljanjem uvjeta usporedbe apsolutne vrijednosti svakoga člana niza s konstantom *eps* za vrijednost *s* dobili vrijednost prvoga člana niza jer bi *while*–petlja završila već u drugom koraku.

Uočimo uporabu funkcije *faktorijel* u trećemu retku i način njezina pozivanja. Jedina njezina ulazna varijabla je prirodan broj *n*. Na početku je $n = 1$. U petomu retku petlje piše naredba da se *n* prigodom svakoga izvršavanja petlje uveća za 1, što znači da *n* poprima vrijednosti 1, 2, …, baš kako i zahtijeva funkcija *faktorijel*. Dakako, prije poziva funkcije moramo inicijalizirati sve njezine ulazne varijable, što smo i učinili inicijalizirajući varijablu *n* u drugomu redu.

Pohranimo ovu datoteku i vratimo se u komandni prozor. U njegovu novomu retku upišimo

```
niz2
```

Pritisnimo *Enter* i dobit ćemo traženu vrijednost zbroja:

```
s =
   0.63212055882856
```

Napomena: Primjere 4. i 5. treba razlikovati od zadataka tipa *Izračunajte zbroj reda … do na točnost T*. U takvim se zadatcima, naime, zahtijeva da apsolutna vrijednost razlike točne vrijednosti zbroja konvergentnoga reda i približne vrijednosti istoga zbroja bude najviše jednaka *T*. U točki 6.1. pokazat ćemo kako u MATLAB-u možemo (točno i približno) računati zbrojeve konvergentnih redova brojeva.

4.3.3. Naredba if…else

Naredba *if…else* (hrv.: ako… inače) koristi se za izvršavanje određenoga bloka naredbi ako je ispunjen određeni logički uvjet. Ako taj uvjet nije ispunjen, može se izvršiti neki drugi blok naredbi. Sintaksa ove naredbe je:

```
if logički izraz
   naredbe1
else
   naredbe2
end
```

Objasnimo ukratko značenje ove naredbe. Kad prigodom izvršavanja programa MATLAB "dođe" do naredbe *if*, "pogleda" je li logički izraz naveden odmah iza naredbe *if* istinit ili nije. Ako jest, točno jednom se izvršava skup naredbi *naredbe*1. U suprotnom, tj. ako logički izraz nije istinit, točno jednom se izvršava skup naredbi *naredbe*2 i nakon toga se prelazi na prvu naredbu iza naredbe *end*. Ako je skup naredbi *naredbe*2 prazan skup, dio

```
else
  naredbe2
```

možemo izostaviti.

Pogledajmo uporabu ove naredbe na primjerima.

Primjer 6. Kreirajmo funkcijsku *m*–datoteku **zbrojm.m** koja sadrži jedino funkciju *zbrojm* čije su ulazne varijable realne matrice x i y, a izlazna varijabla 1 ako postoji zbroj $x + y$, a 0 inače.

Podsjetimo da se matrice mogu zbrojiti ako i samo ako su istoga tipa. Stoga najprije moramo usporediti ukupne brojeve redaka, odnosno stupaca tih matrica, što ćemo učiniti rabeći naredbu *size*. To će ujedno biti i naš logički uvjet u naredbi *if*. Otvorimo novu *m*–datoteku i utipkamo:

```
function z=zbrojm(x,y)
if (size(x,1)==size(y,1)&size(x,2)==size(y,2))
    z=1;
else
    z=0;
end
```

Ovdje prvi put susrećemo uporabu logičkoga operatora *and* (hrv.: i) čiju smo oznaku već upoznali u točki 1.3.

Uočimo uporabu naredbe *if*. Logički uvjet zapisan u drugomu retku "preveden" na "običan" jezik glasi: "*Ako je broj redaka matrice x jednak broju redaka matrice y* **i** *ako je broj stupaca matrice x jednak broju stupaca matrice y*, *onda…*" Ako je taj logički uvjet istinit, prelazi se na naredbu navedenu odmah u retku ispod. U tome se retku izlaznoj varijabli z dodijeljuje vrijednost 1 jer tako zahtijeva zadatak. Ako dotični logički uvjet nije istinit, prelazi se na prvu naredbu ispod naredbe *else*. To je $z = 0$ kojom se izlaznoj varijabli z dodijeljuje vrijednost 0 ponovno zbog zahtjeva zadatka. Treće mogućnosti nema pa se time završava cijela funkcija.

Pohranimo dobivenu datoteku i vratimo se u komandni prozor. Provjerimo "ispravnost rada" naše funkcije na primjeru matrica

$$A = \begin{bmatrix} -1 & 2 & 3 \end{bmatrix},\ B = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \text{ i } C = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

U novi redak komandnoga prozora utipkajmo:

```
A=[-1 2 3];B=[3 1 2];C=[1;2;3];
```

Potom pozovimo funkciju *zbrojm* za parove matrica (A, B) i (B, C). Očekujemo da će *zbrojm*(A, B) biti jednak 1 jer su matrice A i B istoga tipa (tj. tipa (1, 3)), odnosno da će *zbrojm*(A, B) biti jednak 0 jer matrice B i C nisu istoga tipa. Utipkamo:

```
z1=zbrojm(A,B),z2=zbrojm(B,C)
```

Kako smo i očekivali, MATLAB će ispisati:

```
z1 =
     1
z2 =
     0
```

U „kombinaciji“ s naredbom `if` često se koristi i naredba `elseif`. Skup naredbi koji se pojavljuje iza te funkcije izvršava se samo ako je istinit uvjet naveden u sastavu naredbe `elseif`, a nije istinit uvjet naveden u sastavu naredbe `if`. Pogledajmo primjenu ove naredbe na sljedećm primjeru.

Primjer 7. Kreirajmo funkcijsku *m*–datoteku **ekstremi2.m** koja sadrži jedino funkciju *ekstremi*2 čiji su argumenti realni brojevi a, b i c i koja ima dvije izlazne varijable: *min* koja se ispisuje ako i samo ako polinom $p(x) = a \cdot x^2 + b \cdot x + c$ ima globalni minimum i jednaka je vrijednosti toga minimuma, te *max* koja se ispisuje ako i samo ako polinom $p(x) = a \cdot x^2 + b \cdot x + c$ ima globalni maksimum i jednaka je vrijednosti toga maksimuma. U slučaju da polinom nema globalnoga ekstrema, treba ispisati tekst *Nema globalnoga ekstrema*.

Podsjetimo se da polinom $p(x) = a \cdot x^2 + b \cdot x + c$, pri čemu je $a \neq 0$, uvijek ima globalni ekstrem $\frac{4 \cdot a \cdot c - b^2}{4 \cdot a}$. Taj ekstrem je minimum ako je $a > 0$, a maksimum ako je $a < 0$. U slučaju $a = b = 0$ polinom p ima trivijalni globalni minimum i globalni maksimum jednak c, dok za $a = 0$ i $b \neq 0$ polinom nema globalnih ekstrema.

Stoga u *m*–datoteku **ekstremi2.m** utipkamo:

```
function y=ekstremi2(a,b,c);
if (a^2+b^2==0)
    min=c
    max=c
elseif (a==0)
    'Nema globalnoga ekstrema'
else
ekstrem=(4*a*c-b^2)/(4*a);
    if a>0
        min=ekstrem
    else
        max=ekstrem
    end;
end
```

Opišimo ukratko strukturu ove funkcije.

Najprije se provjerava uvjet $a = b = 0$ koji je ekvivalentan jednakosti $a^2 + b^2 = 0$. Ako je taj uvjet ispunjen, ispisuje se da su minimum i maksimum jednaki c. Time je izvršenje funkcije završeno. Ako uvjet $a^2 + b^2 = 0$ nije ispunjen, provjerava se vrijedi li uvjet $a = 0$. Ako je taj uvjet ispunjen, ispisuje se tekst *Nema globalnoga ekstrema* i time se završava izvršenje funkcije. Naredba *else* osigurava razmatranje preostaloga slučaja $a \neq 0$. Najprije se računa globalni ekstrem (prema navedenoj formuli), a onda se postavlja još jedna *if…else* funkcija u kojoj se određuje je li izračunani ekstrem minimum ili maksimum.

***Primjer* 8.** Odredimo sva rješenja nejednadžbe $4 \cdot x^2 + 9 \cdot y^2 \leq 9$ koja se nalaze unutar skupa $[-5, 5]^2$. ($[-5, 5]^2 := [-5, 5] \times [-5, 5]$ je Kartezijev kvadrat segmenta $[-5, 5]$.)

Radi jednostavnosti, zapišimo rješenje ovoga primjera u datoteku **nejednadzba1.m**. Osnovna ideja je provjeriti zadovoljava li svaka točka skupa $[-5, 5]^2$ zadanu nejednadžbu. No, tih točaka ima neprebrojivo mnogo (tj. koliko i realnih brojeva), pa ćemo provjeru ograničiti na ukupno „samo“ 1 000 000 točaka. Utipkamo:

```
hold
x=(-5):0.01:5;
y=(-5):0.01:5;
for m=1:1000
    for n=1:1000
            if 4*x(m)^2+9*y(n)^2<=36
                plot(x(m),y(n))
            end
    end
end
```

Ovdje smo uporabili i novu naredbu *hold* koja nam omogućuje zadržavanje dobivenoga grafa na zaslonu.

Opišimo ukratko strukturu ove obične *m*–datoteke.

U drugomu i trećemu retku inicijalizirali smo jednoretčane matrice apscisa, odnosno ordinata. Svaka od njih ima točno 1000 elemenata, što znači da ćemo napraviti provjeru za ukupno 1 000 000 različitih elemenata skupa $[-5,5]^2$.

U četvrtomu i petomu retku počinju dvije *for* – petlje u kojima su ulazne varijable indeksi matrice x, odnosno matrice y.

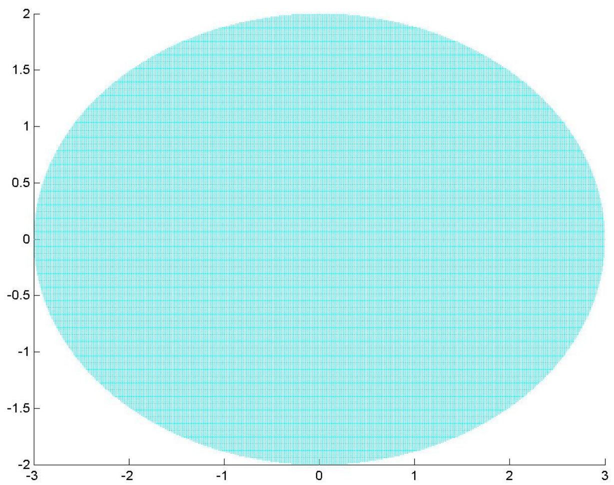
U šestomu retku postavljamo logički uvjet: to je zadana nejednadžba, pa provjeravamo zadovoljavaju li koordinate točke ($x[m]$, $y[n]$) tu nejednadžbu. Ako je logički uvjet istinit, tj. ako točke zadovoljavaju nejednadžbu, u sedmomu retku tražimo od MATLAB-a neka ih ucrta u pravokutni koordinatni sustav u ravnini. Ako logički uvjet nije istinit, naredba *if* je završena, pa MATLAB provjerava je li završena i *for* – petlja za varijablu n. Ako nije, ponovno se izvršava naredba *if*, ali ovoga puta s novom vrijednošću varijable n (za jedan većom nego u prethodnomu koraku). Ako je *for* – petlja za varijablu n završena, MATLAB provjerava je li završena i *for*–petlja za varijablu m. Ako nije, ponovno se izvršava *for*–petlja za varijablu n, ali ovoga puta s novom vrijednošću varijable m (za jedan većom nego u prethodnomu koraku). Ako jest, sve petlje su završene, a kako nema drugih naredbi izvan njih, i sam program je gotov.

Primijetite da se petlja za m izvršava točno 1000 puta, a petlja za n $1000 \cdot 1000 = 1\ 000\ 000$ puta (za svaki m po 1000 puta!). Naredba *if* također se izvršava 1 000 000 puta, ali naredba *plot* ne jer njezino izvršavanje ovisi je li istinit logički uvjet naveden iza naredbe *if*.

Pohranimo dobivenu datoteku i vratimo se u komandni prozor, te u njegovu novomu retku utpikajmo:

```
nejednadzba1
```

i dobit ćemo sljedeću sliku (vidjeti Sliku 1.):



Slika 1.

Analitički rješavajući zadanu nejednadžbu provjerite da je gornji skup točaka uistinu rješenje postavljenoga zadatka.

<u>Primjer 9.</u> Kreirajmo funkcijsku *m*–datoteku **najmanji.m** koja sadrži jedino funkciju *najmanji* čija je jedini argument realna matrica *x*. Funkcija treba ispisati najmanji element matrice *x*.

Ovo je klasičan programerski zadatak koji se susreće u svim tečajevima iz osnovâ programiranja. Osnovna ideja za rješavanje jest sljedeća:

Izlazna varijabla (nazovimo je *y*) inicijalizira se na element x_{11} zadane matrice. Potom tu vrijednost počinjemo redom uspoređivati sa sljedećim elementima matrice (ako takvih ima, tj. ako matrica *x* sadrži barem dva elementa). Ako naiđemo na element manji od vrijednosti pohranjene u varijabli *y*, taj element pohranimo u varijablu *y*. Time se ujedno „briše" posljednje pohranjena vrijednost varijable *y*. Zbog tranzitivnosti[9] relacije "biti jednak ili manji od", novopohranjeni element ne moramo uspoređivati sa svim elementima koje smo "obradili" prije njega, već uspoređivanje nastavljamo dalje sve dok ne dođemo do posljednjega elementa zadane matrice. Na taj način osiguravamo da će u *svakom* trenutku vrijednost varijable *y* biti jednaka najmanjem od svih dotad „obrađenih" elemenata polazne matrice, a samim tim i da će krajnja vrijednost te varijable biti traženi najmanji element polazne matrice.

U datoteku **najmanji.m** utipkajmo sljedeći niz naredbi:

[9] Kažemo da je relacija R tranzitivna ako (aRb) i (bRc) povlače (aRc). U ovom slučaju to znači da nejednakosti $a \le b$ i $b \le c$ povlače nejednakost $a \le c$.

```
function y=najmanji(x)
y=x(1,1);
for m=1:size(x,1)
    for n=1:size(x,2)
        if x(m,n)<=y
            y=x(m,n);
        end
    end
end
```

Pohranimo upisane naredbe i vratimo se u komandni prozor. Provjerimo ispravnost rada naše funkcije na primjerima matrica:

$$A = \begin{bmatrix} -1 & 2 & 3 \end{bmatrix},\ B = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \text{ i } C = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Označimo sa $n1$, $n2$ i $n3$ najmanje elemente tih matrica. Očito je $n1 = -1$, $n2 = n3 = 1$. Pogledajmo što će nam ispisati MATLAB kada u novomu retku komandnoga prozora utipkamo:

```
n1=najmanji(A), n2=najmanji(B), n3=najmanji(C)
```

i pritisnemo *Enter*. Dobivamo:

```
n1 =
    -1
n2 =
     1
n3 =
     1
```

Napomena: Početnici u programiranju su u ovakvim slučajevima vrlo često skloni (potpuno pogrešno!) inicijalizirati vrijednost varijable y na nulu, kao što se to radi prigodom računanja različitih zbrojeva (vidjeti Primjere 4. i 5.). Treba istaknuti da mi *unaprijed* pretpostavljamo da je x matrica koja sadrži barem jedan element (i to onaj na poziciji (1, 1), tj. na presjeku prvoga retka i prvoga stupca). Zbog toga funkcija *najmanji.m* ispravno radi i ako je x matrica reda 1.

4.4. Zadatci za vježbu

1. Zadane su jednoretčane matrice x = [0.5 1 –0.25 0 4] i y = [2 –1 4 0.75 –0.25]. Koristeći *for* – petlju izračunajte vrijednosti sljedećih izraza::

a) $\sum_{k=1}^{5} x_k \cdot y_k$;

b) $\sum_{k=1}^{5} x_k^2 \cdot y_k^3$;

c) $\sum_{k=1}^{5} |x_k| \cdot \sqrt{|y_k|}$;

d) $\sum_{k=1}^{5} \sqrt[3]{\frac{x_k}{y_k}}$.

2. Koristeći *for* – petlju izračunajte duljine radijvektora $x = \left(\sqrt{2} \quad -\sqrt{3} \quad 1 \quad 0 \quad -5\cdot\sqrt{3}\right)$ i $y = \left(\cos\frac{2011\cdot\pi}{4} \quad \sin\frac{2013\cdot\pi}{4} \quad \cos\frac{2011\cdot\pi}{4} \quad \sin\frac{2013\cdot\pi}{4}\right)$.

(Formule za računanje duljine ovih radijvektora analogne su formuli za računanje duljine radijvektora u trodimenzionalnom prostoru.)

3. Zadana je funkcija *nepoznata* s:

```
function y=nepoznata(n)
y=0;
for m=2:3:n
y=y+m^2;
end
```

<u>Bez uporabe MATLAB-</u>a odredite što će se ispisati ako utipkamo:

```
y=nepoznata(9)
```

Potom provjerite svoje rješenje rabeći MATLAB.

4. Zadana je funkcija *nepoznata*1 s:

```
function y=nepoznata1(n)
y=1;
for m=n:-1:1
y=y*m;
end
```

Bez uporabe MATLAB-a odredite što će se ispisati ako utipkamo:

```
y=nepoznata1(9)
```

Potom provjerite svoje rješenje rabeći MATLAB.

5. U nekoj funkcijskoj *m*–datoteci, između ostaloga, piše:

```
for m=1:2:101
   for n=2:2:102
      if (m+n=49)
           .
           .
      end
   end
end
```

(S · su označene naredbe u sastavu naredbe *if* nevažne za rješenje zadatka.)

a) Koliko će se puta izvršiti gornja *for*–petlja za varijablu *m*?
b) Koliko će se puta izvršiti gornja *for*–petlja za varijablu *n*?
c) Koliko će se puta izvršiti naredba *if* ?

6. U nekoj funkcijskoj *m*–datoteci između ostaloga piše:

```
i=1;
while (i<=2012)
  .
  .
  i=i+3;
end
```

(S · su označene naredbe u sastavu naredbe *if* nevažne za rješenje zadatka.)

Koliko će se puta izvršiti gornja *while*–petlja?

7. Izračunajte zbroj svih članova sljedećih nizova koji nisu strogo manji od *eps* ako je:

a) $a_n = \dfrac{1}{n^2}$;

b) $b_n = \dfrac{\sin n}{n^3 + 2 \cdot n^2}$;

c) $c_n = \dfrac{n + \operatorname{tg} n}{2 \cdot n^2 - \ln n + \sqrt{n+1}}$;

d) $d_n = \dfrac{\ln(\sqrt[4]{n} + 1)}{e^{\sqrt{n}-1}}$.

8. Grafički riješite sljedeće nejednadžbe na segmentu [–6, 6]:

a) $x^2 + y^2 \leq 9$;
b) $4 \cdot x^2 + 4 \cdot y^2 \leq 25$;
c) $2 \cdot x^2 + 3 \cdot y^2 \leq 6$;
d) $3 \cdot x^2 + 4 \cdot y^2 \leq 24$;
e) $4 \cdot x^2 - y^2 \leq 4$;
f) $9 \cdot x^2 - 4y^2 \leq 36$;
g) $2 \cdot x + 3 \cdot y < 6$;
h) $3 \cdot x - 2 \cdot y + 12 < 0$.

9. Kreirajte običnu *m*-datoteku pozivom koje će se ispisati:

a) zbroj kvadrata prvih 50 prirodnih brojeva;
b) zbroj trećih korijena prvih 1 000 000 prirodnih brojeva.

10. Kreirajte običnu *m*-datoteku pozivom koje će se prikazati:

a) graf funkcije $f(x) = x^2 + 5 \cdot x + 6$ na segmentu [-6, 6];
b) graf funkcije $f(x) = \sin x$ na segmentu $[-8 \cdot \pi, 8 \cdot \pi]$;
c) grafovi funkcija $f(x) = \sqrt{x+1}$ i $g(x) = \ln(x + 1)$ na segmentu [0, 3].

11. Kreirajte običnu *m*-datoteku **obični.m** pozivom koje će se prikazati graf funkcije $f(x) = \sqrt{x+1}$ na segmentu [–3, 3]. Što se događa nakon poziva te datoteke iz komandnoga prozora? Obrazložite!

12. Kreirajte funkcijsku *m*–datoteku **zn.m** koja će sadržavati jedino funkciju *zn* čiji je jedina ulazna varijabla prirodan broj n. Funkcija treba ispisati vrijednost zbroja prvih n neparnih prirodnih brojeva. Provjerite svoje rješenje za $n = 10$. (Funkcija ne treba provjeravati je li n prirodan broj.)

13. Kreirajte funkcijsku *m*–datoteku **zkp.m** koja će sadržavati jedino funkciju *zkp* čija je jedina ulazna varijabla prirodan broj n. Funkcija treba ispisati vrijednost zbroja kvadrata prvih n parnih prirodnih brojeva. Provjerite svoje rješenje za $n = 10$. (Funkcija ne treba provjeravati je li n prirodan broj.)

14. Kreirajte funkcijsku *m*–datoteku **kvadratna.m** koja će sadržavati jedino funkciju *kvadratna* čije su ulazne varijable realni brojevi a, b i c. Funkcija treba ispisati sva rješenja kvadratne jednadžbe $a \cdot x^2 + b \cdot x + c = 0$.

15. Trag kvadratne matrice A (reda n) jest zbroj elemenata na njezinoj glavnoj dijagonali. Kreirajte funkcijsku *m*–datoteku **trag.m** koja sadrži jedino funkciju *trag* čija je jedina ulazna varijabla realna matrica x. Funkcija treba ispisati trag matrice x ako je x kvadratna matrica, odnosno tekst "*Matrica nije kvadratna*." inače.

16. Kreirajte funkcijsku *m*-datoteku **kvadrati1.m** koja sadrži jedino funkciju *kvadrati*1 čije su ulazne varijable prirodni brojevi m i n. Funkcija treba ispisati zbroj kvadrata svih prirodnih brojeva između m i n ako je $m < n$, odnosno 0 inače.

17. Kreirajte funkcijsku *m*–datoteku **umnozakm.m** koja sadrži jedino funkciju *umnozakm* čije su ulazne varijable matrice x i y. Funkcija treba ispisati 1 ako postoji umnožak $x \times y$ ($\times$ je standardno matrično množenje), odnosno 0 inače.

18. Kreirajte funkcijsku *m*–datoteku **najveci.m** koja sadrži jedino funkciju *najveci* čija je jedina ulazna varijabla realna matrica x. Funkcija treba ispisati najveći element matrice x.

19. Kreirajte funkcijsku *m*–datoteku **zbrojmin.m** koja sadrži jedino funkciju *zbrojmin* čija je ulazna varijabla realna matrica x. Funkcija treba ispisati zbroj najmanjih elemenata u retcima matrice x.

20. Kreirajte funkcijsku *m*–datoteku **umnozakmax.m** koja sadrži jedino funkciju *umnozakmax* čija je jedina ulazna varijabla realna matrica x. Funkcija treba ispisati umnožak najvećih elemenata u stupcima matrice x.

21. Kreirajte funkcijsku *m*-datoteku **aritmeticki.m** koja sadrži jedino funkciju *aritmeticki* čije su ulazne varijable realni brojevi a i d, te prirodan broj n. Funkcija treba ispisati zbroj prvih n članova aritmetičkoga niza kojemu je prvi član a, a razlika d. Provjerite svoje rješenje za $a = 1$, $d = 2$ i $n = 20$. (Funkcija ne treba provjeravati je li n prirodan broj.)

22. Kreirajte funkcijsku *m*-datoteku **naritm.m** koja sadrži jedino funkciju *naritm* čije su ulazne varijable realni brojevi a i d, te prirodan broj n. Funkcija treba ispisati n – ti član aritmetičkoga niza kojemu je prvi član a, a razlika d. Provjerite svoje rješenje za $a = 1$, $d = 7$ i $n = 10$. (Funkcija ne treba provjeravati je li n prirodan broj.)

23. Kreirajte funkcijsku *m*-datoteku **geometrijski.m** koja sadrži jedino funkciju *geometrijski* čije su ulazne varijable realni brojevi a i q, te prirodan broj n. Funkcija treba ispisati zbroj prvih n članova geometrijskoga niza kojemu je prvi član a, a količnik q. Provjerite svoje rješenje za $a = 1$, $q = 2$ i $n = 10$. (Funkcija ne treba provjeravati je li n prirodan broj. <u>Oprez</u>: Vrijednost broja n može biti jednaka 1.)

24. Kreirajte funkcijsku *m*-datoteku **ngeom.m** koja sadrži jedino funkciju *ngeom* čije su ulazne varijable realni brojevi a i q, te prirodan broj n. Funkcija treba ispisati n–ti član geometrijskoga niza kojemu je prvi član a, a količnik q. Provjerite svoje rješenje za $a = 1$, $q = 4$ i $n = 5$. (Funkcija ne treba provjeravati je li n prirodan broj.)

25. Kreirajte funkcijsku *m*-datoteku **geomred.m** koja sadrži jedino funkciju *geomred* čije su ulazne varijable realni brojevi a i q. Funkcija treba ispisati tekst „Red je konvergentan i njegov zbroj je jednak:“ i potom zbroj geometrijskoga reda ako je geometrijski red $\sum a \cdot q^n$ konvergentan, odnosno tekst „Red nije konvergentan.“ inače.

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

§5. DIFERENCIJALNI I INTEGRALNI RAČUN U MATLAB-U

Jedan od za elektrotehničare najvažnijih matematičkih aparata jest diferencijalni i integralni račun. U *Matematici* 1 naučili smo derivirati realnu funkciju jedne realne varijable, a u *Matematici* 2 naučili smo integrirati realnu funkciju jedne realne varijable i računati određene, odnosno neprave integrale. Ovdje ćemo vidjeti kako te zadatke riješiti uz pomoć MATLAB-a.

Prije negoli navedemo konkretne primjere, istaknimo jednu vrlo važnu činjenicu. MATLAB smo dosad ponajviše koristili za računanje vrijednosti konkretnih numeričkih izraza. No, u MATLAB-u je moguće izvršavati matematičke operacije u kojima se ne zahtijeva računanje numeričkih vrijednosti. Takve operacije su upravo deriviranje i integriranje. Da bismo mogli izvršiti takve operacije, prije zadavanja propisa odgovarajuće funkcije moramo definirati tzv. *simboličke objekte*. Tu "definiciju" u MATLAB-u obavlja funkcija `syms`. Npr.

```
syms x;
```

definira x kao varijablu s kojom MATLAB dalje može nešto raditi (računati vrijednosti te varijable, ispisati neku funkciju dotične varijable itd.). Ako imamo više takvih varijabli, možemo ih konstruirati istovremeno tako da napišemo npr.

```
syms x y z
```

Uočite da se objekti ne odvajaju znakom zareza, nego isključivo prazninama. Ovdje se nećemo zadržavati na potankostima takvih konstrukcija, nego ih spominjemo tek uzgredno radi rješavanja primjera koji slijede.

5.1. Deriviranje u MATLAB–u

Deriviranje funkcija u MATLAB-u može se izvršiti izravno iz komandnoga prozora ili korištenjem funkcijskih *m*-datoteka. Za početnike sklone sintaktičkim pogrješkama primjereniji je drugi način (pomoću *m*-datoteka), dok je za iskusnije korisnike pogodan prvi način (koji ćemo i mi koristiti).

Derivacije realnih funkcija određujemo koristeći funkciju `diff`. Jedna od značajnih prednosti te funkcije jest mogućnost određivanja n-te derivacije polazne funkcije za „konkretan" prirodan broj n. Dobiveni izraz možemo donekle pojednostavniti i srediti koristeći funkciju `simple`, te ga eventualno zapisati u „ljepšem" obliku koristeći funkciju `pretty`.

Jednako je važno i izračunati vrijednost dobivena izraza za konkretne vrijednosti varijabli koje ga tvore. U tu svrhu koristit ćemo funkciju `subs` koja omogućuje uvrštavanje „konkretnih" vrijednosti pojedinih varijabli.

Pogledajmo sve navedeno na primjerima.

***Primjer* 1.** Odredimo prve tri derivacije polinoma $p(x) = x^{100} - x^{50} + 1$, pa izračunajmo njihovu vrijednost u točki $x = 1$.

Krenimo redom. Ako smo dosad u našemu komandnom prozoru imali neke tekstove, formule, funkcije i sl. „počistimo“ ih na uobičajen način koristeći funkciju `clc`. U prvi redak „praznoga“ komandnoga prozora utipkajmo:

```
syms x;
```

Pritisnimo *Enter*. MATLAB nije ispisao ništa. Negdje smo pogriješili? Nismo, sve je u redu. Tako će uvijek biti nakon deklariranja simboličkih objekata. U sljedeći redak komandnoga prozora utipkajmo:

```
p=x^100-x^50+1;
```

Što je sad ovo? Polinom iz zadatka zove se $p(x)$. Zašto smo onda napisali p= (…)? Razlog je opet jednostavan: ime ili naziv funkcije ne može biti $f(x)$, $p(x)$ i sl., nego samo f, p itd. Hoće li onda MATLAB znati što je nezavisna, a što zavisna varijabla? Neće, jer MATLAB ne zna niti hrvatski jezik, niti matematiku, ali zna da se varijabla p dobije provedbom određenih simboličkih operacija nad varijablom x. To mu je sasvim dovoljno (a i nama također).

Ovime smo zadali polinom p. Tražene derivacije toga polinoma odredimo primjenom funkcije `diff`. Te derivacije ne smijemo označavati s p', p'' i p''' jer znak ' u MATLAB-u ima posve drugo značenje (to je ispis određenoga teksta, kao što možemo vidjeti u Primjeru 7. iz točke 4.3.). Zbog toga ćemo tražene derivacije označiti s p1, p2 i p3. Odredit ćemo ih u jednom retku, a potom rezultat ispisati u matričnom obliku:

```
p1=diff(p,1);p2=diff(p,2);p3=diff(p,3);[p1 p2 p3]
```

pa će MATLAB ispisati:

```
ans =
[100*x^99-50*x^49, 9900*x^98-2450*x^48, 970200*x^97-117600*x^47]
```

Dakle, prva derivacija polinoma p je $p'(x) = 100 \cdot x^{99} - 50 \cdot x^{49}$, druga $p''(x) = 9900 \cdot x^{98} - 2450 \cdot x^{48}$, a treća $p'''(x) = 970200 \cdot x^{97} - 117600 \cdot x^{47}$. Ispravnost dobivenih rezultata lako možemo provjeriti i analitičkim deriviranjem.

Preostaje izračunati vrijednosti dobivenih derivacija u točki $x = 1$. U tu ćemo svrhu iskoristiti funkciju `subs`. Utipkamo:

```
[subs(p1,1) subs(p2,1) subs(p3,1)]
```

pa će MATLAB ispisati:

```
ans =
          50          7450        852600
```

Dakle, $p'(1) = 50$, $p''(1) = 7\,450$ i $p'''(1) = 852\,600$. Ovime je primjer u potpunosti riješen.

Primjer 2. Odredimo prve tri derivacije funkcije $f(x) = \dfrac{x^2}{e^{x+1}}$ i izračunajmo njihovu vrijednost u točki $x = -1$.

Postupamo na način analogan onom iz rješenja prethodnoga primjera. Pritom se moramo podsjetiti da potencije broja *e* računamo pomoću funkcije `exp`. Dakle, u sljedeća tri retka komandnoga prozora utipkamo:

```
f=x^2/exp(x+1);
f1=simple(diff(f,1));f2=simple(diff(f,2));f3=simple(diff(f,3));
pretty([f1 f2 f3])
```

pa će MATLAB ispisati prilično neočekivan rezultat:

```
[                               2                  2]
[  x (-2 + x)    2 - 4 x + x         6 - 6 x + x ]
[- ----------    ------------     - ------------]
[  exp(x + 1)     exp(x + 1)          exp(x + 1) ]
```

dobiven zajedničkom primjenom funkcije za pojednostavljivanje algebarskih izraza `simple` i funkcije za „ljepši“ zapis dobivena rješenja `pretty`. Korištenje ovih dviju funkcija u ovom zadatku, naravno, nije bilo nužno, ali je bilo zgodno ukazati kakve <u>zapise</u> rezultata možemo dobiti.

I ovaj primjer pokazuje da je „ljudski faktor“ u nekim situacijama nezamjenjiv. Naime, prvi ali i treći dobiveni izraz mogu se dodatno pojednostavniti (ali ne pomoću MATLAB-a!), pa dobivamo konačno rješenje:

$$f'(x) = \frac{2 \cdot x - x^2}{e^{x+1}},\ f''(x) = \frac{x^2 - 4 \cdot x + 2}{e^{x+1}},\ f'''(x) = \frac{-x^2 + 6 \cdot x - 6}{e^{x+1}}.$$

Izračunajmo još vrijednosti dobivenih derivacija u točki $x = -1$. U novi redak komandnoga prozora utipkajmo:

```
[subs(f1,-1) subs(f2,-1) subs(f3,-1)]
```

pa će MATLAB ispisati:

```
ans =
    -3      7    -13
```

Dakle, $f'(-1) = -3$, $f''(-1) = 7$ i $f'''(-1) = 13$.

***Primjer* 3.** Odredimo jednadžbu (u eksplicitnom obliku) tangente povučene na krivulju $y = x^3 - 7 \cdot x^2 + 12 \cdot x - 1$ u točki krivulje s apscisom $x = 1$.

Ovakav smo zadatak analitički rješavali u kolegiju *Matematika* 1, pa već manje-više dobro znamo ideju za njegovo rješavanje. Najprije odredimo obje koordinate dirališta D u kojemu povlačimo tangentu na zadanu krivulju. Potom odredimo prvu derivaciju funkcije čiji je graf zadana krivulja, pa izračunamo vrijednost k dobivenoga izraza za $x = 1$. Vrijednost k bit će koeficijent smjera tražene tangente. Naposljetku, pišemo jednadžbu pravca koji ima koeficijent smjera k i prolazi točkom D.

Krenimo na rješavanje. S obzirom na broj „nepotrebnih“ međurezultata, ovaj je primjer najpogodnije riješiti rabeći funkcijsku *m*-datoteku *tangenta.m*. U odgovarajući prostor utipkamo redom:

```
function z=tangenta(f,xd)
yd=subs(f,xd);
f1=diff(f,1);
k=subs(f1,xd);
sprintf('Jednadzba tangente je: t... y = %d * x + %d',k,yd-k*xd)
end
```

i pohranimo dobiveni unos kao funkcijsku *m*-datoteku *tangenta.m*.

Pojasnimo ukratko sadržaj datoteke *tangenta.m*.

U prvom smo retku definirali funkciju *tangenta* koja ima dvije ulazne varijable: funkciju f i apscisu dirališta *xd*. (Zapravo bi trebalo pisati x_d, ali u MATLAB-u takav manevar nije moguće napraviti.)
U drugome smo retku izračunali ordinatu dirališta *yd* tako što smo MATLAB-u rekli neka u izraz za funkciju f umjesto nezavisne varijable x uvrsti vrijednost *xd*. Doista, ordinata dirališta upravo se tako i računa: uvrštavanjem apscise dirališta u jednadžbu krivulje.
U trećem smo retku odredili prvu derivaciju $f1$ ulazne funkcije f.
U četvrtom smo retku izračunali vrijednost funkcije $f1$ za ulaznu vrijednost *xd*. Kako znamo, vrijednost prve derivacije f u nekoj točki c jednaka je koeficijentu smjera tangente povučene na graf funkcije f u točki c.
U petom smo retku ispisali jednadžbu tangente koristeći MATLAB-ovu funkciju `sprintf`. Sintaksu te funkcije, kao i pojedinosti o njoj, možete pogledati u izborniku *MATLAB Help*. Ovdje tek istaknimo da se prvi zapis *%d* odnosi na ispis prvoga broja navedenoga iza zareza koji neposredno slijedi iza znaka ' kojim završava tekst koji treba ispisati. Drugi se zapis *%d* odnosi na ispis drugoga broja navedenoga iza spomenutoga zareza. Uočimo: prvi broj je k, a drugi broj je $yd - k \cdot xd$.

Jedina eventualna nejasnoća jest: otkuda se pojavio ovaj „čudan“ izraz $yd - k \cdot xd$? Odgovor biste trebali znati iz *Matematike* 1. Naime, jednadžba tangente t koja ima koeficijent smjera k i prolazi točkom $D = (x_d, y_d)$ glasi:

$$y - y_d = k \cdot (x - x_d),$$

odnosno, u preuređenom obliku,

$$y = k \cdot x + (y_d - k \cdot x_d).$$

Dakle, koeficijent smjera tangente t jednak je k, a odsječak na osi Oy jednak je $y_d - k \cdot x_d$, pa otuda slijede veličine navedene u datoteci.

Pohranimo dobiveni unos, zatvorimo *m*-datoteku *tangenta.m* i vratimo se u komandni prozor. U dvama novim retcima komandnoga prozora utipkajmo:

```
f=x^3-7*x^ 2+12*x-1;
tangenta(f,1)
```

pa će MATLAB ispisati traženu jednadžbu:

```
Jednadzba tangente je: t... y = 1 * x + 4.
```

***Primjer* 4.** Koristeći datoteku *tangenta.m* odredimo jednadžbu tangente povučene na krivulju $y = \ln x$ u točki s apscisom $x = 0$.

U ovom zadatku odmah uočavamo opaku zamku koju postavljamo MATLAB-u: niti jedna realna logaritamska funkcija oblika $f(x) = \log_a x$, pri čemu je $a > 0$ i $a \neq 1$, nije definirana u točki $x = 0$. Stoga unaprijed znamo da zadatak nema rješenja. Što li će onda MATLAB napraviti u tom slučaju? Pogledajmo sami. U nova dva retka komandnoga prozora utipkajmo:

```
f=log(x);
tangenta(f,0)
```

i promatrajmo učinak posljednje upisane naredbe. MATLAB će ispisati:

```
Warning: Log of zero.
> In d:\matlabr12\toolbox\symbolic\@sym\subs.m at line 118
  In D:\matlabR12\work\tangenta.m at line 2
Warning: Divide by zero.
> In d:\matlabr12\toolbox\symbolic\@sym\subs.m at line 118
  In D:\matlabR12\work\tangenta.m at line 4
ans =
Jednadzba tangente je: t... y = Inf * x + NaN
```

Dakle, prijevara nam nije uspjela. MATLAB je, naravno, otkrio i upozorio nas da zahtijevamo računanje prirodnoga logaritma od nule, te da pri računanju prve derivacije logaritamske funkcije u točki $x = 0$ dijelimo s nulom (što je, naravno, točno jer, kako dobro znamo, vrijedi jednakost $(\ln x)' = \frac{1}{x}$.) Stoga je u rezultatu ispisao unaprijed deklarirane konstante *Inf* i *NaN*. (vidjeti podtočku 1.2.3, stranica 9.) Ispisani rezultat je praktički besmislen, ali izvrsno pokazuje kako računalni program poput MATLAB-a djeluje u takvim „ekstremnim“ situacijama.

Primjer 5. Koristeći datoteku *tangenta.m* odredimo jednadžbu tangente povučene na središnju jediničnu kružnicu u točki s apscisom $x = 1$.

Iz *Matematike* 1 znamo: središnja jedinična kružnica nije graf niti jedne realne funkcije jedne realne varijable jer postoji beskonačno mnogo pravaca usporednih s osi ordinata koji sijeku kružnicu u točno dvije točke (umjesto u najviše jednoj točki, kako zahtijeva jedno od osnovnih svojstava grafa funkcije). Međutim, „razdvojimo“ li središnju jediničnu kružnicu na dvije polukružnice (koje imaju središte u ishodištu i jedan promjer koji pripada osi apscisa), onda takve polukružnice predstavljaju grafove realnih funkcija. Budući da središnja jedinična kružnica ima jednadžbu $x^2 + y^2 = 1$, spomenute dvije polukružnice određene su jednadžbama $y = \sqrt{1-x^2}$ i $y = -\sqrt{1-x^2}$. Odaberimo npr. prvu polukružnicu, pa u dvama novim retcima komandnoga prozora utipkajmo:

```
f=sqrt(1-x^2);
tangenta(f,1)
```

pa će MATLAB ispisati:

```
Warning: Divide by zero.
> In d:\matlabr12\toolbox\symbolic\@sym\subs.m at line 118
  In D:\matlabR12\work\tangenta.m at line 4
ans =
Jednadzba tangente je: t... y = -Inf * x + Inf
```

Dakle, analogno kao i u prethodnom primjeru, zaključujemo da tražena tangenta ne postoji. (Istu poruku bismo dobili i da smo odabrali drugu polukružnicu). No, takav zaključak očito nije istinit jer tražena tangenta postoji i ima jednadžbu $x = 1$ (pravac usporedan s osi ordinata koji prolazi diralištem $D = (1, 0)$). Gdje je MATLAB pogriješio? Ispravan odgovor je: nigdje. U *Matematici* 1 smo istaknuli da tangente usporedne s osi ordinata ne možemo izravno „otkriti“ primjenom tehnika diferencijalnoga računa na krivulju $y = f(x)$, već posredno (najbolje grafički)[10].

U nastavku posebnu ćemo pozornost posvetiti određivanju lokalnih ekstrema realne funkcije jedne realne varijable uz korištenje MATLAB-a. U tu svrhu najprije razmotrimo sljedeći

Problem: Je li moguće napisati funkcijsku *m*–datoteku čiji će jedini ulazni podatak biti realna funkcija *f* i koji će, kao izlaz, vratiti sve lokalne ekstreme zadane funkcije? Obrazložite svoj odgovor.

[10] Navedeni problem moguće je riješiti zamjenom „uloga“ varijabli x i y (tj. promatranjem varijable x kao funkcije varijable y), a potom primjenom spomenutih tehnika diferencijalnoga računa. Za vježbu, preuredite datoteku *tangenta.m* tako da se pomoću funkcije *tangenta* mogu otkriti i tangente usporedne s osi x.

***Odgovor*:** Nije moguće. Naime, kad pišemo funkcijsku *m*–datoteku mi unaprijed moramo deklarirati ukupan broj ulaznih i izlaznih varijabli. Za ukupan broj ulaznih varijabli to je jednostavno učiniti: taj je broj jednak 1. Međutim, ukupan broj izlaznih varijabli nije moguće unaprijed deklarirati. Problem „zapinje" već kod polinoma gdje se kao dodatna ulazna varijabla mora zadati stupanj polinoma n (jer znamo da svaki polinom stupnja n ima najviše $n - 1$ lokalnih ekstrema), a kod racionalnih i transcendentnih funkcija postaje elementarno (programerski) nerješiv[11].

Kojim postupkom određujemo i klasificiramo lokalne ekstreme? U *Matematici* 1 naučili smo dva načina: $f\,'$ – *test* (ispitivanje određenih svojstava prve derivacije zadane funkcije na otvorenom intervalu oko svake stacionarne točke) i $f\,''$ – test (ispitivanje određenih svojstava vrijednosti druge derivacije zadane funkcije izračunane u svakoj stacionarnoj točki). Ukratko ćemo opisati implementaciju svakoga od tih dvaju testova. Pritom nećemo provjeravati svojstva poput neprekidne derivabilnosti funkcije na nekom intervalu oko stacionarne točke itd., već ćemo prešutno pretpostaviti da su ta svojstva ispunjena.

Podsjetimo da se $f\,'$ – test algoritamski može zapisati ovako:

***Ulazni podatak*:** realna funkcija f
***Korak* 1.** Odrediti prvu derivaciju $f\,'$.
***Korak* 2.** Odrediti nultočke funkcije $f\,'$ (tj. stacionarne točke funkcije f) koje pripadaju prirodnu području definicije funkcije f. Označimo te stacionarne točke s $x_1, x_2, \ldots, x_n$.
***Korak* 3.** Za svaki $i = 1, \ldots, n$ odabrati $e_i > 0$ takav da je funkcija f neprekidno derivabilna na intervalu $\langle x_i - e_i, x_i + e_i\rangle$ i da navedeni interval ne sadrži niti jednu drugu stacionarnu točku osim x_i.
***Korak* 4.** Odrediti *predznak* brojeva $a_i = f\,'\left(x_i - \frac{e_i}{2}\right)$ i $b_i = f\,'\left(x_i + \frac{e_i}{2}\right)$.[12]
***Korak* 5.** Ako je istodobno $a_i < 0$ i $b_i > 0$, onda za $x = x_i$ f ima lokalni minimum $f(x_i)$. Ako je istodobno $a_i > 0$ i $b_i < 0$, onda za $x = x_i$ f ima lokalni maksimum $f(x_i)$. Inače, za $x = x_i$ f nema lokalni ekstrem.

Pogledajmo kako implementirati ovaj algoritam u MATLAB-u. Već smo vidjeli kako odrediti prvu derivaciju neke realne funkcije jedne realne varijable (koristeći funkciju `diff`) i kako izračunati vrijednosti bilo koje realne funkcije u proizvoljnoj točki njezina prirodna područja definicije (koristeći funkciju `subs`). Stoga je jedini preostali problem implementacija Koraka 2., tj. određivanje nultočaka prve derivacije, odnosno, općenito, određivanje nultočaka bilo koje realne funkcije jedne realne varijable.

[11] O rješivosti algebarskih jednadžbi vidjeti npr. u [7], str. 161.
[12] Kao nazivnik razlomka u zagradama može se uzeti bilo koji realan broj strogo veći od 1. Također, odabrani otvoreni interval ne mora biti simetričan s obzirom na x_i, ali ćemo, zbog jednostavnosti, birati upravo takve intervale.

Opći algoritam za <u>točno</u> određivanje svih nultočaka realne funkcije jedne realne varijable ne postoji. U 10. poglavlju pokazat ćemo kako se MATLAB može koristiti u rješavanju nekih problema numeričke matematike, a jedan od tih problema bit će upravo približno određivanje nultočaka realne funkcije. Ovdje ćemo te nultočke odrediti pomoću MATLAB-ove ugrađene funkcije `solve`. Jedini ulazni argument te funkcije je jednadžba oblika $f(x) = c$, gdje je c bilo koji realan broj, a izlazni argumenti su (približno izračunana) realna rješenja navedene jednadžbe.

Iako, kako smo rekli, ne možemo stvoriti funkcijsku *m*-datoteku koja će računati i ispisivati sve lokalne ekstreme neke funkcije, ipak možemo donekle skratiti postupke navedene u Koracima 3. - 5. Stvorit ćemo funkcijsku *m*-datoteku *ekstrem*1.*m* čiji će ulazni podatci biti realna funkcija f, njezina stacionarna točka x i strogo pozitivna realna konstanta e, pa ćemo implementirati navedeni algoritam na sljedeći način:

```
function z=ekstrem1(f,x,e)
f1=diff(f,1);
y=subs(f,x);
a=x-e/2;
b=x+e/2;
c=subs(f1,a);
d=subs(f1,b);
if((c<0)&(d>0))
    sprintf('Funkcija ima lokalni minimum %d za x = %d.',y,x)
elseif ((c>0)&(d<0))
    sprintf('Funkcija ima lokalni maksimum %d za x = %d.',y,x)
else
    sprintf('Funkcija nema lokalni ekstrem za x = %d.',x)
end
```

Pogledajmo primjenu ove funkcije na primjeru.

***<u>Primjer</u>* <u>6.</u>** Ispitajmo postoje li, i ako postoje, odredimo sve lokalne ekstreme polinoma $p(x) = x^4 + 1$.

Krenimo redom. Radi preglednosti, „počistimo“ komandni prozor koristeći funkciju `clc`. U nova retka praznoga komandnoga prozora utipkajmo:

```
p=x^4+1;
p1=diff(p)
```

pa će MATLAB ispisati:

```
p1 =
4*x^3
```

Stacionarne točke polinoma $p_1(x)$ odredit ćemo tako da u novi redak komandnoga prozora utipkamo:

```
solve('4*x^3=0')
```

Pritisnimo *Enter*, pa će MATLAB (očekivano) ispisati:

```
ans =
[ 0]
[ 0]
[ 0]
```

jer je $x = 0$ očito jedina (i to trostruka) nultočka polinoma $p_1(x) = 4 \cdot x^3$. Stoga je jedina stacionarna točka zadanoga polinoma $x = 0$. Prirodno područje definicije bilo kojega polinoma je skup realnih brojeva **R**, pa ponašanje prve derivacije polinoma p (tj. polinoma p_1) možemo promatrati na bilo kojem otvorenom intervalu u **R** koji sadrži nulu. Nećemo komplicirati život: opredijelimo se za interval $\langle -1, 1\rangle$, tj. odaberimo $e = 1$. U sljedeći redak komandnoga prozora utipkajmo:

```
ekstrem1(p,0,1)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
Funkcija ima lokalni minimum 1 za x = 0.
```

Ovaj jednostavan primjer namjerno je odabran jer se lokalni ekstremi zadanoga polinoma ne mogu odrediti pomoću f'' – testa, kao što ćemo ubrzo vidjeti.

Preostaje opisati implementaciju f''–testa u MATLAB-u. Najprije podsjetimo na njegov algoritam (prešutno ćemo pretpostavljati da je f barem dvaput neprekidno derivabilna realna funkcija jedne realne varijable jer nećemo rješavati zadatke s funkcijama koje nemaju navedeno svojstvo):

Ulazni podatak: realna funkcija f
***Korak* 1**. Odrediti prvu derivaciju f'.
***Korak* 2.** Odrediti nultočke funkcije f'. Označimo te nultočke s: $x_1, x_2, \ldots, x_n$.
***Korak* 3.** Odrediti drugu derivaciju f''.
***Korak* 4.** Odrediti predznak vrijednosti $f''(x_i)$, za svaki $i = 1, \ldots, n$. Ako je $f''(x_i) > 0$, onda f poprima lokalni minimum $f(x_i)$ za $x = x_i$. Ako je $f''(x_i) < 0$, onda f poprima lokalni maksimum $f(x_i)$ za $x = x_i$. Ako je $f''(x_i) = 0$, onda f''– test ne daje odluku.[13]

Stvorimo funkcijsku *m*-datoteku *ekstrem*2.*m* čiji će ulazni podatci biti realna funkcija f i vrije-

[13] Studenti često posve pogrešno zaključuju da iz $f''(x_i) = 0$ slijedi da x_i nije lokalni ekstrem. Kako ćemo vidjeti, x_i može biti lokalni ekstrem i ako vrijedi jednakost $f''(x_i) = 0$.

dnost x. Pomoću f''–testa funkcija će ispisati je li x lokalni ekstrem funkcije f i, ako jest, je li riječ o lokalnom minimumu ili lokalnom maksimumu.

Otvorimo novu *m*-datoteku, pa u nju utipkajmo sljedeći niz naredbi:

```
function z=ekstrem2(f,x);
a=subs(f,x);
f1=diff(f,2);
if subs(f1,x)>0
    sprintf('funkcija ima lokalni minimum %d za x = %d',a,x)
else
    if subs(f1,x)<0
        sprintf('funkcija ima lokalni maksimum %d za x = %d',a,x)
    else
        sprintf('Test ne daje odluku!')
    end
end
```

***Zadatak*:** Objasnite svaki redak ove funkcijske *m*-datoteke.

Pohranimo navedeni niz naredbi pod imenom *ekstrem*2.*m*, pa zatvorimo dobivenu datoteku i vratimo se u komandni prozor. Pokušajmo riješiti Primjer 6. koristeći netom stvorenu funkcijsku datoteku. U novi redak komandnoga prozora utipkajmo:

```
ekstrem2(p,0)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
Test ne daje odluku!
```

Takav rezultat smo mogli i predvidjeti jer je očito $p''(x) = 12 \cdot x^2$ i $p''(0) = 0$. Stoga u ovakvim slučajevima zadatak možemo riješiti isključivo primjenom funkcijske *m*-datoteke *ekstrem*1.*m*.

***Primjer* 7.** Ispitajte postoje li i, ako postoje, odredite lokalne ekstreme realne funkcije $f(x) = 2 \cdot x^3 - 3 \cdot x^2 - 12 \cdot x + 11$.

Koristit ćemo ranije opisani algoritam, ugrađenu funkciju `solve` i obje *m*-datoteke *ekstrem*1.*m* i *ekstrem*2.m. Budući da argument funkcije `solve` mora biti „prava“ jednadžba (npr. $x + 2 = 0$), a ne samo izraz oblika $p(x) = 0$, moramo ispisati izraz za prvu derivaciju zadanoga polinoma, te se dodatno poslužiti *kopiraj – zalijepi* (engl.: *copy – paste*) tehnikom koja se u MATLAB-u primjenjuje kao i u MS Wordu, MS PowerPointu i drugim računalnim programima[14]. Međutim, argument funkcija *ekstrem*1 i *ekstrem*2 može biti simbolički objekt poput funkcije, pa nije nužno ispisivati sâm zadani polinom.

[14] Kako možemo izbjeći uporabu *copy – paste* tehnike, objasnit ćemo u točki 5.4.

Krenimo redom. U nova dva retka komandnoga prozora utipkajmo:

```
p=2*x^3-3*x^2-12*x+11;
p1=diff(p)
```

pa će MATLAB ispisati:

```
p1 =
6*x^2-6*x-12
```

Mišem označimo tekst `6*x^2-6*x-12`, pa istodobno pritisnimo tipke Ctrl i C (ako više volite raditi mišem, lijevom tipkom miša kliknite na izbornik *Edit*, pa odaberite opciju *Copy*). Vratimo se potom u tekući radni redak, pa utipkajmo:

```
solve('
```

Bez pritiskanja tipke *Enter* nakon što utipkamo znak ', pritisnimo istodobno tipke Ctrl i V (ako više volite raditi mišem, lijevom tipkom miša kliknite na izbornik *Edit*, pa odaberite opciju *Paste*), pa ćemo dobiti:

```
solve('6*x^2-6*x-12
```

Iza broja 12 u istom retku preostaje nadopisati:

```
=0')
```

i tek sad pritisnuti *Enter*. MATLAB će ispisati:

```
ans =
[ -1]
[  2]
```

Tako smo utvrdili da su stacionarne točke zadanoga polinoma $x_1 = -1$ i $x_2 = 2$. Preostaje provjeriti jesu li to doista lokalni ekstremi. Polinom p definiran je za svaki realan broj x, pa najprije trebamo odabrati otvoreni interval koji sadrži broj –1, a ne sadrži broj 2. Opredijelimo se za interval $\langle -2, 0\rangle$, tj. ponovno odaberimo $e = 1$. U novi redak komandnoga prozora utipkajmo:

```
ekstrem1(p,-1,1)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
funkcija ima lokalni maksimum 18 za x = -1
```

Pritisnimo tipku ↑ na tipkovnici i preuredimo posljednje utipkani redak ovako:

```
ekstrem2(p,-1)
```

Pritisnimo *Enter*, pa će MATLAB (ponovno) ispisati:

```
ans =
funkcija ima lokalni minimum 18 za x = -1.
```

Preostaje provjeriti ima li polinom p lokalni ekstrem za $x = 2$. Provjeru pomoću funkcije *ekstrem*1 napravite sami za vježbu. Ovdje ćemo napraviti provjeru pomoću funkcije *ekstrem*2. U novi redak komandnoga prozora utipkajmo:

```
ekstrem2(p,2)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
funkcija ima lokalni minimum -9 za x = 2
```

Za vježbu, prikažite grafički zadani polinom na segmentu [–3, 3] i uvjerite se u točnost dobivenih rezultata.

Primjer 8. Ispitajte postoje li i, ako postoje, odredite sve lokalne ekstreme realne funkcije $f(x) = \dfrac{\ln x}{x^3}$.

Postupimo analogno kao u Primjeru 7. U nova dva retka komandnoga prozora utipkajmo:

```
f=log(x)/x^3;
f1=simple(diff(f))
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
f1 =
(1-3*log(x))/x^4
```

Utipkajmo nadalje:

```
solve('
```

prekopirajmo izraz `(1-3*log(x))/x^4` i nadopišimo:

```
=0')
```

Sljedeći redak treba izgledati ovako:

```
solve('(1-3*log(x))/x^4=0')
```

Nakon što pritisnemo *Enter*, MATLAB će ispisati:

```
ans =
exp(1/3)
```

Dakle, jedina stacionarna točka je $x = \sqrt[3]{e}$. Provjerimo je li to lokalni ekstrem upisujući

```
ekstrem2(f,exp(1/3))
```

u sljedeći redak komandnoga prozora. Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
funkcija ima lokalni maksimum 1.226265e-001 za x = 1.395612e+000
```

Što se dogodilo? Je li MATLAB nešto „zabrljao“, pa smo dobili ovako čudne rezultate? Odgovor je: nije, sve je u najboljem redu. Broj 1.395612e+000 je zapravo znanstveni prikaz broja 1.395612, a taj je decimalan broj aproksimacija broja $\sqrt[3]{e}$ na šest decimalnih mjesta. Nadalje, 1.226265e-001 je zapravo znanstveni prikaz broja 0.1226265, a taj je broj aproksimacija broja $f\left(\sqrt[3]{e}\right) = \frac{1}{3 \cdot e}$ na sedam decimalnih mjesta. (Provjerite istinitost prethodnih dviju tvrdnji računajući približne vrijednosti brojeva $\sqrt[3]{e}$ i $\frac{1}{3 \cdot e}$.) Dakle, dobili smo posve ispravnu tvrdnju, samo što umjesto točnih vrijednosti $\sqrt[3]{e}$ i $\frac{1}{3 \cdot e}$ pišu njihove decimalne aproksimacije. Tu „pojavu“, nažalost, ne možemo izbjeći jer se ona javlja kad god rezultat dobivamo koristeći funkciju zapisanu u funkcijskoj *m*-datoteci.

<u>Napomena</u>: Ma koliko god vam se „moćna“ činila funkcija `solve`, zapamtite da ona (približno) može rješavati <u>isključivo jednadžbe</u>. Dakle, rješavanje bilo koje vrste nejednadžbi ne dolazi u obzir. U MATLAB-u ne postoji funkcija za rješavanje bilo kojega tipa nejednadžbi, pa se prigodom rješavanja zadataka s nejednadžbama moramo snalaziti kako god znamo i umijemo. Za „utjehu“ valja reći da se rješavanje nejednadžbi često svodi na rješavanje određenih tipova jednadžbi, a u tu nam svrhu vrlo solidno može poslužiti i spomenuta funkcija `solve`.

U nastavku ćemo riješiti još jedan problem iz primjene diferencijalnoga računa, a to je problem određivanja prijevojnih točaka (točaka infleksije) neke realne funkcije jedne realne varijable. Podsjetimo se, kandidati za prijevojne točke funkcije f su sve nultočke druge derivacije te funkcije. U *Matematici* 1 promatrali smo predznak druge derivacije na okolini svakoga pojedinoga kandidata i zaključivali mijenja li druga derivacija predznak pri prolasku kroz svoju nultočku. To smo učinili tako da smo uzeli „konkretne“ brojeve iz „male“ ε–okoline svakoga pojedinoga kandidata i računali predznake vrijednosti $f''(x)$ za svaki od tih brojeva. Analogno ćemo postupiti i sad, pri čemu ćemo prešutno pretpostaviti da je funkcija f barem dvaput neprekidno derivabilna u točki x.

Najprije ćemo formirati funkcijsku *m*-datoteku *infleksija.m*. Ulazni podatci za funkciju *infleksija* bit će: polazna funkcija f, apscisa „kandidatkinje“ za prijevojnu točku x i polovica širine okoline oko točke x (označimo je ponovno s e). Funkcija *infleksija* će ispisati ima li polazna realna funkcija *f* prijevojnu točku te, ako ima, ispisati i njezine koordinate.

Otvorimo novu *m*-datoteku, pa u odgovarajući prostor utipkajmo redom:

```
function z=infleksija(f,x,e)
f1=diff(f,2);
y=subs(f,x);
a=x-e/2;
b=x+e/2;
c=subs(f1,a);
d=subs(f1,b);
if c*d<0
    sprintf('Funkcija ima prijevojnu tocku (%d, %d)',x,y)
else
    sprintf('Funkcija nema prijevojnu tocku.')
end
```

Zadatak: Objasnite svaki pojedini redak ove funkcijske *m*-datoteke.

Pohranimo uneseni niz naredbi pod nazivom *infleksija.m*, pa se vratimo u komandni prozor. Potom riješimo sljedeći primjer.

Primjer 9. Ispitajmo ima li polinom $p(x) = x^4 + 1$ prijevojnu točku, pa, ako ima, odredimo je.

Ovaj je primjer namjerno odabran za „ispitivanje“ valjanosti rada funkcije *infleksija*. Naime, lako se vidi da je $p''(x) = 12 \cdot x^2 \geq 0$, za svaki $x \in \mathbf{R}$, pa sve vrijednosti koje može poprimiti polinom p'' imaju isti predznak. To znači da zadani polinom p nema prijevojnu točku.

Sad kad smo otkrili „tajnu“, tj. krajnji rezultat, prijeđimo na rješavanje zadatka. Ranije opisanim postupkom najprije ćemo doznati „kandidate“ za prijevojnu točku. Utipkajmo:

```
p=x^4+1;
p2=diff(p,2)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
p2 =
12*x^2
```

Formalno riješimo jednadžbu $p''(x) = 0$, tj. $p_2(x) = 0$ iako rješenja možemo otkriti i napamet:

```
solve('12*x^2=0')
```

MATLAB će ispisati:

```
ans =
[ 0]
[ 0]
```

Preostaje provjeriti ima li polazni polinom p prijevojnu točku u točki s apscisom 0. Promatrajmo „ponašanje" polinoma p'', tj. polinoma p_2 na segmentu [–0.0001, 0.0001], tj. zadajmo e = 0.0001. Utipkajmo:

```
infleksija(p,0,0.0001)
```

pa će MATLAB ekspresno ispisati očekivani rezultat:

```
ans =
Funkcija nema prijevojnu tocku.
```

***Primjer* 10.** Ispitajmo ima li realna funkcija $f(x) = \frac{\ln x}{x^2}$ prijevojnu točku. Ako ima, odredimo koordinate te točke.

U nova dva retka komandnoga prozora utipkamo:

```
f=log(x)/x^2;
f2=simple(diff(f,2))
```

pa će MATLAB ispisati:

```
f2 =
(-5+6*log(x))/x^4
```

U sljedeći redak najprije upišimo

```
solve('
```

potom prekopirajmo izraz `(-5+6*log(x))/x^4`, pa nadopišimo:

```
=0')
```

tako da dobijemo:

```
solve('(-5+6*log(x))/x^4=0')
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
exp(5/6)
```

Preostalo je još provjeriti je li točka čija je apscisa $x = \sqrt[6]{e^5}$ doista prijevojna točka. U tu svrhu opet uzmimo e = 0.0001, pa u sljedeći redak komandnoga prozora utipkajmo:

```
infleksija(f,exp(5/6),0.0001)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
Funkcija ima prijevojnu tocku (2.300976e+000, 1.573963e-001)
```

Ovdje imamo situaciju analognu onoj iz Primjera 8. Približna vrijednost broja $\sqrt[6]{e^5}$ je broj 2.300976, a približna vrijednost broja $f\left(\sqrt[6]{e^5}\right)=\frac{5\cdot\sqrt[3]{e}}{6\cdot e^2}$ je broj 0.1573963 čiji je zapis u znanstvenom obliku 1.573963e-001. (Provjerite istinitost navedenih aproksimacija.)

5.1.1. Kompozicija funkcija. Inverz funkcije. Derivacija kompozicije funkcije. Derivacija inverza funkcije

Pomoću MATLAB-a možemo određivati i kompoziciju dviju funkcija, inverz bilo koje bijekcije, te derivaciju tako dobivenih funkcija. Kompoziciju dviju realnih funkcija jedne realne varijable određuje funkcija `compose`, dok inverz neke realne bijekcije određuje funkcija `finverse`. Pogledajmo primjenu tih dviju funkcija na primjerima.

Primjer 1. Zadane su realne funkcije $f(x) = x^2$ i $g(x) = e^x$. Odredimo kompozicije funkcija $f\circ g$ i $g\circ f$.

Najprije „očistimo“ MATLAB-ov komandni prozor koristeći funkciju `clc`. Odredimo najprije kompoziciju $f\circ g$. U nova tri retka MATLAB-ova komandnoga prozora utipkajmo:

```
syms x f g;
f=x^2;g=exp(x);
simplify(compose(f,g,x))
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
exp(2*x)
```

Kako smo mogli i očekivati, rezultat je $(f\circ g)(x)=e^{2\cdot x}$. Odredimo kompoziciju $g\circ f$. Pritisnimo tipku ↑ i preuredimo posljednje utipkani redak ovako:

```
simplify(compose(g,f,x))
```

Ponovno pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
exp(x^2)
```

Dakle, $(g\circ f)(x)=e^{x^2}$. Kako dobro znamo iz *Matematike* 1, općenito vrijedi nejednakost $f\circ g\neq g\circ f$, u što nas je uvjerio i ovaj primjer.

Sljedeći će nam primjer pokazati što se događa kad neka jednadžba ima beskonačno mnogo međusobno različitih realnih rješenja.

Primjer 2. Zadane su realne funkcije $f(x)$ = sin x i $g(x)$ = $|x|$. Riješimo jednadžbu: $(g \circ f)(x) = 1$.

Označimo h:= $g \circ f$. Funkciju h odredimo kao u Primjeru 1. U nova dva retka MATLAB-ova komandnoga prozora utipkajmo:

```
f=sin(x);g=abs(x);
h=compose(g,f,x)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
h =
abs(sin(x))
```

Dakle, $h(x) = (g \circ f)(x) = |\sin x|$. Stoga trebamo riješiti jednadžbu:

$$|\sin x| = 1.$$

Riješimo li je analitički, dobivamo:

$$x_k = \frac{\pi}{2} + k \cdot \pi,\ k \in \mathbf{Z}.$$

MATLAB, međutim, kao rezultat ne može ispisati ovakav izraz, nego isključivo „konkretan“ realan ili kompleksan broj jer je izlazni argument te funkcije konačnodimenzionalna matrica. Stoga utipkavanjem:

```
solve('abs(sin(x))=1',x)
```

kao rezultat ne dobivamo gornji izraz, nego:

```
ans =
[  1/2*pi]
[ -1/2*pi]
```

Dobivena rješenja nisu slučajno odabrana. Naime, umjesto „prave“ funkcije $f(x)$ = sin x, prigodom rješavanja ove jednadžbe MATLAB je zapravo „promatrao“ restrikciju funkcije f na segment $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, tj. funkciju $F(x)$ = Sin x. Podsjetimo se da je ovakav segment odabran jer je na tom segmentu funkcija $F(x)$ bijekcija i ima inverz $F^{-1}(x)$ = arcsin x. Stoga je MATLAB zapravo riješio jednadžbu $(g \circ F)(x) = 1$, tj. jednadžbu $|\text{Sin } x| = 1$.

Napomena: Analogno razmatranje vrijedi zamijenimo li funkciju $f(x) = \sin x$ funkcijom $f_1(x) = \cos x$, samo što se umjesto segmenta $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ promatraju segment $[0, \pi]$ na kojem je funkcija f_1 bijekcija i pripadna restrikcija funkcije f_1 na taj segment, tj. funkcija $F_1(x) = \mathrm{Cos}\, x$. Uvjerite se u to.

Primjer 3. Odredimo inverz bijekcije $f : \mathbf{R} \to \langle 1, +\infty\rangle$ definirane propisom $f(x) = e^x + 1$. (Za vježbu provjerite da je funkcija f doista bijekcija.)

U nova dva retka MATLAB-ova komandnoga prozora utipkajmo:

```
f=exp(x)+1;
simplify(finverse(f))
```

pa će MATLAB (očekivano) ispisati:

```
ans =
log(-1+x)
```

Ovaj izraz valja pažljivo interpretirati:

$$f^{-1}(x) = \ln(x - 1).$$

Primjer 4. Zadane su realne funkcije $f(x) = x^2 + 1$ i $g(x) = \sqrt{x}$. Izračunajte $(f \circ g)^{-1}(1)$.

U ovom primjeru kombiniramo funkcije `compose`, `finverse` i `subs`. U nova tri retka MATLAB-ova komandnoga prozora utipkavamo redom:

```
f=x^2+1;g=sqrt(x);
h=finverse(compose(f,g,x));
subs(h,1)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
     0
```

Dakle, $(f \circ g)^{-1}(1) = 0$. Provjerite ovaj rezultat i analitičkim rješavanjem zadatka.

Primjer 5. Zadane su realne funkcije $f(x) = \ln x$ i $g(x) = \mathrm{arctg}\, x$. Riješite jednadžbu: $(g \circ f)^{-1}(x) = e$.

Najprije postupimo kao u prethodnom primjeru. U nova dva retka MATLAB-ova komandnoga prozora utipkajmo:

```
f=log(x);g=atan(x);
h=finverse(compose(g,f,x))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
h =
exp(tan(x))
```

Preostaje iskoristiti funkciju `solve`:

```
solve('exp(tan(x))=exp(1)',x)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
1/4*pi
```

Dakle, dobili smo $x = \frac{\pi}{4}$, i to je jedino rješenje polazne jednadžbe (zašto?).

Napomena: Primjer 5. smo mogli riješiti nešto kraće i „lukavije“. Naime, označimo li $h = g \circ f$, onda polaznu jednadžbu možemo zapisati u obliku:

$$h^{-1}(x) = e.$$

Na lijevu i desnu stranu jednadžbe djelujemo s funkcijom *h*, pa dobijemo:

$$h[h^{-1}(x)] = h(e),$$

Prema definiciji inverza funkcije, lijeva je strana jednaka *x*, pa gornja jednakost prelazi u:

$$x = h(e).$$

Stoga u nova tri retka MATLAB-ova komandnoga prozora utipkajmo:

```
f=log(x);g=atan(x);
h=compose(g,f,x);
x1=subs(h,exp(1))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
x1 =
    0.78539816339745
```

Lako se provjeri da je $\frac{\pi}{4} \approx 0.78539816339745$. Zbog primjene funkcije *subs*, rješenje zadatka nismo mogli dobiti u obliku $x = \frac{\pi}{4}$.

Primjer 6. Zadane su realne funkcije $f(x)=\sqrt[3]{x}$ i $g(x)$ = arcctg x. Odredimo derivacije funkcija $f\circ g$, $g\circ f$, f^{-1} i g^{-1}. (Za vježbu, analitički odredite te derivacije.)

Označimo tražene derivacije redom s f_1, f_2, f_3 i f_4. Radi preglednosti, sve rezultate zapisat ćemo u matričnom obliku. U sljedeća tri retka komandnoga prozora[15] utipkajmo:

```
f=x^(1/3);g=acot(x);
f1=diff(compose(f,g,x),1);f2=diff(compose(g,f,x),1);
f3=diff(finverse(f),1);f4=diff(finverse(g),1);F=[f1 f2 f3 f4]
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
F =
[-1/3/acot(x)^(2/3)/(x^2+1),-1/3/x^(2/3)/(1+x^(2/3)),3*x^2,-1-cot(x)^2]
```

Ovaj rezultat treba pažljivo interpretirati:

$$\left(f\circ g\right)'(x)=-\frac{1}{3\cdot\sqrt[3]{\operatorname{arcctg}^2 x}\cdot(x^2+1)},$$

$$\left(g\circ f\right)'(x)=-\frac{1}{3\cdot\sqrt[3]{x^2}\cdot\left(1+\sqrt[3]{x^2}\right)};$$

$$(f^{-1})'(x)=3\cdot x^2;\ (g^{-1})'(x)=-1-\operatorname{ctg}^2 x.$$

Primijetimo da je MATLAB prividno pogriješio u određivanju $(g^{-1})'$. Naime, očito vrijedi:

$$(g^{-1})'(x)=(\operatorname{ctg} x)'=-\frac{1}{\sin^2 x},$$

a MATLAB je ispisao:

$$(g^{-1})'(x)=f_4(x)=-1-\operatorname{ctg}^2 x\,.$$

Međutim, nije teško provjeriti jednakost funkcija $-\frac{1}{\sin^2 x}$ i $-1-\operatorname{ctg}^2 x$. Učinite to za vježbu.

Primjer 7. Zadane su realne funkcije $f(x) = \cos x$ i $g(x) = e^x$. Izračunajmo $\left[(f\circ g)^{-1}\right]'(0)$, pa riješimo jednadžbu $\left[(g\circ f)^{-1}\right]'(x)=-1$.

Prije rješavanja ovoga primjera napravite veliko spremanje i „očistite“ MATLAB-ov koman-

[15] Sve potrebne funkcije mogu se deklarirati i u samo jednom retku. Radi preglednosti, odlučili smo se za višeretčano deklariranje. Ista napomena vrijedi i za daljnja deklariranja varijabli, funkcija i sl.

dni prozor pomoću funkcije `clc`. Označimo $h_1 := \left[(f \circ g)^{-1}\right]'$ i $h_2 := \left[(g \circ f)^{-1}\right]'$ Potom u nova četiri retka komandnoga prozora utipkajmo:

```
f=cos(x);g=exp(x);
h1=diff(finverse(compose(f,g,x)),1);
h2=diff(finverse(compose(g,f,x)),1);
subs(h1,0)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
  -0.63661977236758
```

Riješimo li prvi dio zadatka analitički, dobit ćemo točnu vrijednost $\left[(f \circ g)^{-1}\right]'(0) = -\frac{2}{\pi}$.

Nije teško provjeriti jednakost $-\frac{2}{\pi} \approx -0.63661977236758$.

Za drugi dio zadatka utipkajmo najprije:

```
h2
```

pa će MATLAB ispisati:

```
h2 =
-1/x/(1-log(x)^2)^(1/2)
```

Preostaje primijeniti *kopiraj–zalijepi* tehniku i funkciju `solve`:

```
solve('-1/x/(1-log(x)^2)^(1/2)=-1',x)
```

pa će MATLAB ispisati prilično neočekivan rezultat:

```
ans =
exp(0)
```

Ovakav je zapis posljedica primjene funkcije `solve` i približnoga rješavanja nealgebarskih jednadžbi. Utipkamo li u sljedećem retku komandnoga prozora

```
simplify(ans)
```

dobit ćemo znatno ljepši oblik rješenja zadane jednadžbe:

```
ans =
1
```

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

5.2. Računanje graničnih vrijednosti

U ovoj ćemo točki ukratko objasniti kako pomoću MATLAB-a možemo računati različite granične vrijednosti funkcija. U tu nam svrhu služi MATLAB-ova ugrađena funkcija `limit`. Pomoću te funkcije možemo računati različite jednostrane ili obostrane granične vrijednosti funkcije u nekoj točki ili u „plus“ ili „minus beskonačnosti“. Pogledajmo to na primjerima.

***Primjer* 1.** Izračunajte graničnu vrijednost niza čiji je opći član $a_n = \dfrac{3^n + n^3}{2^n + n^2}$.

Ako to dosad niste učinili, svakako „počistite“ MATLAB-ov komandni prozor koristeći funkciju `clc`. Potom u nova dva retka „očišćenoga“ komandnoga prozora upišimo:

```
syms n;
limit((3^n+n^3)/(2^n+n^2),n,inf)
```

Uočimo sintaksu funkcije `limit` u ovom slučaju: najprije se upisuje opći član niza, potom ime nezavisne varijable, te naposljetku vrijednost prema kojoj teži ta nezavisna varijabla. Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
inf
```

Taj rezultat smo mogli i unaprijed očekivati. Naime, za „jako velike“ $n \in \mathbf{N}$ vrijedi asimptotska aproksimacija: $a_n = \dfrac{3^n + n^3}{2^n + n^2} \sim \dfrac{3^n}{2^n} = \left(\dfrac{3}{2}\right)^n$. Budući da za bilo koju eksponencijalnu funkciju čija je baza $a > 1$ vrijedi $\lim\limits_{x \to +\infty} a^x = +\infty$, polazni niz nema svoju graničnu vrijednost.

***Primjer* 2.** Izračunajte graničnu vrijednost niza čiji je opći član $a_n = \sqrt[3]{\left(1 + \dfrac{3}{4 \cdot n}\right)^{2 \cdot n}}$.

„Doajeni“ pismenih ispita iz *Matematike* 1 mogli bi vam potvrditi da su vrlo slični zadatci bili zadavani na tim pismenim ispitima i da je njihova riješenost bila – pristojno rečeno – vrlo skromna. Eh, da im je na pismenom ispitu još bio dostupan MATLAB, gdje bi im bio kraj! Ostavimo spomenute „doajene“ neka uživaju u svojim slatkim maštarijama, a mi riješimo postavljeni zadatak uz pomoć MATLAB-a. U novi redak komandnoga prozora utipkajmo:

```
limit(((1+3/(4*n))^(2*n))^(1/3),n,inf)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
exp(1/2)
```

Dakle, granična vrijednost zadanoga niza iznosi $e^{\frac{1}{2}} = \sqrt{e}$.

Primjer 3. Izračunajmo graničnu vrijednost niza čiji je opći član $a_n = \dfrac{\arcsin n}{n}$.

Još jedna „opaka zamka“ za MATLAB! Znamo da je prirodno područje definicije realne funkcije arcsin x segment [–1, 1]. Stoga zadani niz ima točno jedan član, i to je $a_1 = \dfrac{\pi}{2}$. Stoga bi valjalo očekivati da tražena granična vrijednost ne postoji. Utipkamo li, međutim,

```
limit(asin(n)/n,n,inf)
```

MATLAB će ispisati:

```
ans =
0
```

Neki će možda dobiti neodoljiv poriv otvoriti šampanjac i nazdraviti novom znanstveno-tehničkom otkriću *bug*-a u MATLAB-u, ali, na njihovu žalost, nema valjana razloga za zdravicu. Naime, u ovom slučaju MATLAB promatra arkussinus kao *kompleksnu* funkciju jedne *kompleksne* varijable, Preciznije, za $x \in [-1, 1]$ rezultat funkcije arcsin x je, kao i obično, neki realan broj iz segmenta $\left[-\dfrac{\pi}{2}, \dfrac{\pi}{2}\right]$, dok je za sve ostale realne brojeve x vrijednost arkussinusa određena izrazom

$$\arcsin x = -i \cdot \ln\left(x \cdot i + \sqrt{1 - x^2}\right)$$

i predstavlja kompleksan broj. Metodama i tehnikama kompleksne analize može se pokazati da je, uz tako definiranu funkciju arcsin x, tražena granična vrijednost doista jednaka 0.

Granična vrijednost neke realne funkcije određuje se potpuno analogno kao i granična vrijednost niza (uostalom, znamo da je svaki niz realnih brojeva poseban slučaj realne funkcije jedne realne varijable). Pogledajmo to na primjerima.

Primjer 4. Izračunajmo graničnu vrijednost $\lim\limits_{x \to 0} \dfrac{e^{\operatorname{arctg} x} - e^{\arcsin x}}{1 - \cos^3 x}$.

Ovu graničnu vrijednost relativno teško bismo izračunali na elementaran način (pokušajte to ipak učiniti!) ili uz uporabu L'Hôpitalova pravila, ali za MATLAB teškoća nema. Utipkavanje

```
limit((exp(atan(x))-exp(asin(x)))/(1-(cos(x))^3),x,0)
```

kao rezultat daje

```
ans =
0
```

Napomena: Isti rezultat dobili bismo i utipkavanjem

```
limit((exp(atan(x))-exp(asin(x)))/(1-(cos(x))^3))
```

jer izostavljanjem oznake varijable i vrijednosti prema kojoj teži ta varijabla MATLAB dogovorno pretpostavlja da dotična varijabla ima oznaku x i da teži prema nuli. Spretniji među vama prigodom računanja graničnih vrijednosti mogu primijeniti ovo pravilo, dok se onima manje spretnijima preporučuje neka zapisuju i oznaku varijable i vrijednost prema kojoj teži ta varijabla (kao što se, uostalom, mora učiniti u svim drugim slučajevima).

Primjer 5. Izračunajmo graničnu vrijednost $\lim\limits_{x\to\frac{\pi}{2}}\left[\operatorname{tg}\left(\frac{\pi}{4}\cdot\sin x\right)\right]^{\operatorname{ctg}(\pi\cdot\sin x)}$.

Ovoga puta utipkavanje

```
limit((tan(pi/4*sin(x)))^(cot(pi*sin(x))),x,pi/2)
```

kao rezultat daje

```
ans =
exp(1/2)
```

Pokušajte i ovu graničnu vrijednost izračunati bez uporabe L'Hôpitalova pravila.

Primjer 6. Izračunajmo graničnu vrijednost $\lim\limits_{x\to+\infty}\left[\frac{\ln(x^2+2\cdot x+8)}{\ln(x^2+4\cdot x+6)}\right]^{x\cdot\ln x}$.

Analogno kao i kod granične vrijednosti niza, u novi redak komandnoga prozora utipkavamo:

```
limit(((log(x^2+2*x+8))/log(x^2+4*x+6))^(x*log(x)),x,inf)
```

pa će MATLAB ispisati:

```
ans =
exp(-1)
```

Primjer 7. Izračunajmo graničnu vrijednost $\lim\limits_{x\to-\infty}\frac{\sqrt[3]{x}}{\sqrt[3]{x+\sqrt[3]{x+\sqrt[3]{x}}}}$.

Granična vrijednost u „minus beskonačnost“ računa se potpuno analogno kao i granična vrijednost u „plus beskonačnost“, pri čemu ispred konstante *inf* treba staviti znak -. Dakle, utipkamo:

```
limit(x^(1/3)/(x+(x+x^(1/3))^(1/3))^(1/3),x,-inf)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
1
```

Osim „tipičnih“ obostranih graničnih vrijednosti, pomoću funkcije `limit` mogu se računati i različite jednostrane granične vrijednosti. Evo nekoliko primjera.

***Primjer* 8.** Izračunajmo graničnu vrijednost $\lim\limits_{x \to 0+} \dfrac{\sqrt{1-e^{-x}} - \sqrt{1-\cos x}}{\sqrt{\sin x}}$.

U ovom slučaju u novi redak komandnoga prozora utipkavamo:

```
limit((sqrt(1-exp(-x))-sqrt(1-cos(x)))/sqrt(sin(x)),x,0,'right')
```

Uočimo da riječ *right* (engl.: desno) mora biti zapisana pod znakom jednostrukih navodnika. Ovdje, naravno, nismo mogli izostaviti nulu kao u napomeni iza Primjera 4. jer MATLAB-u treba dati do znanja da se radi o jednostranoj, a ne o obostranoj graničnoj vrijednosti.

Pritisnemo tipku *Enter*, pa će MATLAB ispisati:

```
ans =
1
```

***Primjer* 9.** Izračunajmo graničnu vrijednost $\lim\limits_{x \to 1-} \dfrac{\arcsin(1-x)}{\sin(\pi \cdot x)}$.

U ovom slučaju u novi redak komandnoga prozora utipkavamo:

```
limit(asin(1-x)/sin(pi*x),x,1,'left')
```

Budući da je riječ o graničnoj vrijednosti slijeva, posljednji ulazni podatak funkcije `limit` je riječ *left* (engl: lijevo). Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
1/pi
```

***Primjer* 10.** Pomoću MATLAB-a možemo izravno provjeriti valjanost derivacija elementarnih funkcija. Podsjetimo se, prema definiciji, derivacija neke realne funkcije jedne realne varijable u točki x je granična vrijednost $\lim\limits_{h\to 0}\dfrac{f(x+h)-f(x)}{h}$. Provjerimo je li doista derivacija funkcije sin x jednaka cos x.

U sljedeća dva retka komandnoga prozora utipkamo:

```
syms h
limit((sin(x+h)-sin(x))/h,h,0)
```

Pritisnemo *Enter*, pa će MATLAB ispisati rezultat koji smo i očekivali:

```
ans =
cos(x)
```

5.3. Integriranje u MATLAB-u

Integiranje u MATLAB-u izvodi se korištenjem funkcije `int`. Ta funkcija zahtijeva definiranje podintegralne funkcije kao simboličkoga objekta. Međutim, za razliku od funkcije `diff` koja ne omogućuje izravan izračun derivacije neke funkcije u „konkretnoj" točki, funkcija `int` omogućuje izravno računanje određenih i nepravih integrala. Osnovni razlog je mogućnost svođenja tzv. višestrukih integrala na konačan niz „običnih" (jednostrukih) određenih integrala (vidjeti Primjer 3. u točki 10.5., str. 238).

Pri određivanju *neodređenih* integrala u MATLAB-u treba upozoriti da MATLAB *ne* određuje neodređeni integral kao skup svih mogućih primitivnih funkcija, nego isključivo primitivnu funkciju čiji je slobodni član $C = 0$. Želimo li, pak, ispisati neodređeni integral, na MATLAB-ovo rješenje treba dodati nepoznatu realnu konstantu C. Ovdje se nećemo baviti pitanjima integrabilnosti pojedinih realnih funkcija, nego ćemo prešutno pretpostavljati da su sve funkcije integrabilne na svojim prirodnim područjima definicije.

***Primjer* 1.** Odredimo $\int \operatorname{arcctg} x \cdot dx$.

Pokušajte navedeni zadatak riješiti analitički primjenom djelomične (parcijalne) integracije. U novi redak komandnoga prozora utipkamo:

```
int(acot(x))
```

Uočite da nismo naveli varijablu po kojoj se integrira jer MATLAB dogovorno pretpostavlja da je ta varijabla x. Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
acot(x)*x+1/2*log(1+x^2)
```

Dakle, vrijedi:

$$\int \operatorname{arcctg} x \cdot dx = x \cdot \operatorname{arcctg} x + \frac{1}{2} \cdot \ln\left(1 + x^2\right) + C,\ C \in \mathbf{R}.$$

Primjer 2. Pogledajmo što se događa s funkcijama za koje znamo da imaju primitivnu funkciju, ali tu funkciju nije moguće zapisati u „zatvorenom“ (analitičkom) obliku, odnosno kao linearnu kombinaciju elementarnih funkcija. Odredimo $\int \operatorname{arctg}^2 x \cdot dx$.

U novi redak komandnoga prozora utipkamo:

```
int((atan(x))^2)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
Warning: Explicit integral could not be found.
> In d:\matlabr12\toolbox\symbolic\@sym\int.m at line 58
ans =
int(atan(x)^2,x)
```

Dakle, MATLAB je ispisao poruku da nije mogao eksplicitno odrediti traženi integral, te varijabli *ans* dodijelio „vrijednost“ jednaku početnom integralu.

U nekim slučajevima, međutim, integral koji nije prikaziv kao linearna kombinacija elementarnih funkcija moguće je prikazati kao linearnu kombinaciju nekih drugih funkcija koje je vrlo komplicirano (ili čak nemoguće) zadati uobičajenom zatvorenom formulom. Takav je npr. integral $\int e^{-x^2} \cdot dx$.

U novom retku komandnoga prozora utipkamo:

```
int(exp(-x^2))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
1/2*pi^(1/2)*erf(x)
```

U ovome je rješenju *erf* tzv. Gaussova funkcija pogrješke definirana s

$$\operatorname{erf}(x) = \frac{2}{\pi} \cdot \int_0^x e^{-t^2} \cdot dt,$$

pa smo zapravo dobili izraz koji nije jednostavniji od polaznoga, ali čija se *približna* vrijednost za svaki „konkretan“ realan broj x može izračunati s proizvoljnom točnošću.

Primjer 3. Izračunajmo određeni integral $\int_1^{e\cdot\sqrt{e}} \frac{\ln x}{\sqrt[3]{x}}\cdot dx$.

Pokušajte ovaj primjer najprije riješiti analitički pomoću djelomične integracije i Newton-Leibnizove formule. U MATLAB-u jednostavno utipkamo:

```
int(log(x)/(x^(1/3)),1,exp(3/2))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
3/2251799813685248*5045933306791233^(2/3)*1125899906842624^(1/3)*log(5045933306791233)-
75/1125899906842624*5045933306791233^(2/3)*1125899906842624^(1/3)*log(2)-
9/4503599627370496*5045933306791233^(2/3)*1125899906842624^(1/3)+9/4
```

Kakav li je ovo „užasan“ rezultat? Ako ste najprije računali analitički, rezultat je bio „pristojan“: točno $\frac{9}{4}$. Zašto se onda i ovdje nije pojavila ta vrijednost? Razlog je sljedeći.

Kad god je to moguće, MATLAB nastoji točno izračunati vrijednost integrala i u tom obliku ispisati rezultat. Odredimo li najprije neku primitivnu funkciju funkcije $f(x)=\frac{\ln x}{\sqrt[3]{x}}$ postupkom analognim onom iz Primjera 1., dobit ćemo npr.

$$F(x)=\frac{3}{2}\cdot\ln x\cdot\sqrt[3]{x^2}-\frac{9}{4}\cdot\sqrt[3]{x^2}\ .$$

U taj izraz MATLAB najprije uvrštava $x = e\cdot\sqrt{e}$, potom $x = 1$ i izračuna razliku dobivenih vrijednosti ne pokušavajući pojednostavniti dobiveni numerički izraz, niti ga izračunati na točno određen broj decimalnih mjesta. Stoga treba odrediti približnu vrijednost toga izraza.

Približan izračun *bilo kojega* numeričkoga izraza s „konkretnim“ brojevima u MATLAB-u obavlja funkcija `double`. Pritisnimo tipku ↑, pa posljednji upisani redak preuredimo ovako:

```
double(int(log(x)/x^(1/3),1,exp(3/2)))
```

i dobit ćemo približnu (zapravo, točnu) vrijednost polaznoga integrala:

```
ans =
   2.25000000000000
```

Primjer 4. Izračunajmo određeni integral $\int_0^{\frac{\pi}{2}} \cos^3 x\cdot dx$.

Na temelju iskustva iz prethodnoga primjera unaprijed nas može obuzeti nelagoda. Tko zna kakav li će se sad izraz pojaviti kao krajnji rezultat! Ipak, utipkamo li:

```
int(cos(x)^3,0,pi/2)
```

i pritisnemo *Enter*, MATLAB će ispisati:

```
ans =
2/3
```

Zbog čega i u ovom slučaju nismo dobili neki „grozan“ rezultat? Rad s eksponencijalnim i logaritamskim funkcijama u MATLAB-u nije baš potpuno isti kao i rad s trigonometrijskim funkcijama. Drugim riječima, nije svejedno računamo li cos 0 ili $e \cdot \sqrt{e}$, a pogotovo ne kad su u pitanju izrazi sa simboličkim objektima. Za vježbu provjerite valjanost dobivena rezultata analitičkim načinom.

Navedenim primjerima željeli smo ukazati kakvi se problemi i situacije mogu pojavljivati kod računanja određenih integrala. Preostaje nam utvrditi mogu li se pomoću MATLAB-a računati i teški nepravi integrali koje smo rješavali u *Matematici* 2. Odgovor je, naravno potvrdan. Evo nekoliko primjera.

Primjer 5. Izračunajmo (vjerojatno najpoznatiji) nepravi integral $\int\limits_{-\infty}^{+\infty} e^{-x^2} \cdot dx$.

Ovaj nepravi integral vjerojatno izaziva ne baš ugodna sjećanja svakome tko je ikad u životu imao nekakva posla s normalnom razdiobom (vidjeti stranicu 214.). MATLAB takvih sjećanja, niti problema, nasreću, nema. U novi redak komandnoga prozora utipkamo:

```
int(exp(-x^2),-inf,inf)
```

pritisnemo *Enter*, pa će MATLAB ispisati (točnu!) vrijednost zadanoga nepravog integrala:

```
ans =
pi^(1/2)
```

Doista, $\int\limits_{-\infty}^{+\infty} e^{-x^2} \cdot dx = \sqrt{\pi}$.

Primjer 6. Izračunajmo nepravi integral $\int\limits_{0}^{6} \frac{x^3 - 1}{x - 1} \cdot dx$.

Iskusniji i vještiji među nama odmah će uočiti da u segmentu integracije [0, 6] podintegralna racionalna funkcija $f(x) = \frac{x^3 - 1}{x - 1}$ ima uklonjiv pol $x = 1$, što znači da zadani integral konvergira. Pogledajmo što će ispisati MATLAB. U novi redak komandnoga prozora utipkajmo:

```
int((x^3-1)/(x-1),0,3)
```

i utvrdit ćemo da se MATLAB nije dao prevariti:

```
ans =
96
```

Provjerite ispravnost dobivenoga rezultata analitičkim određivanjem nepravoga integrala.

***Primjer* 7.** Izračunajmo nepravi integral $\int_0^1 x \cdot \ln x \cdot dx$.

Nekima od vas će se možda ponovno javiti nelagoda kao u Primjeru 4. Idemo najprije utvrditi zašto je zadani integral nepravi. Označimo li $f(x) = x \cdot \ln x$, onda lako vidimo da je prirodno područje definicije funkcije f interval $\langle 0, +\infty\rangle$, što znači da podintegralna funkcija nije definirana u donjem kraju područja integracije. Dakle, doista je riječ o nepravom integralu.

U novi redak komandnoga prozora utipkajmo:

```
int(x*log(x),0,1)
```

pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
-1/4
```

Provjerite ispravnost dobivenoga rješenja analitičkim određivanjem nepravoga integrala.

***Primjer* 8.** Izračunajmo nepravi integral $\int_{-\infty}^{0} \frac{\text{arctg } x}{(1+x^2)^2} \cdot dx$.

Analitičko određivanje ovoga integrala preporučujemo samo studentima koji su *Matematiku* 2 uspjeli položiti s ocjenom izvrstan (5). Pogledajmo hoće li taj integral MATLAB-u stvoriti probleme. Upit

```
int(atan(x)/(1+x^2)^ 2,-inf,0)
```

činit će mu se kao da ste ga pitali koliko je 2 + 2 jer je brzinom munje ispisao:

```
ans =
1/4-1/16*pi^2
```

Primjer 9. Izračunajmo nepravi integral $\int_0^{+\infty} \frac{\text{arcctg}\, x}{\sqrt{1+x^2}} \cdot dx$.

Početak je jasan: utipkajmo

```
int(acot(x)/sqrt(x^2+1),0,inf)
```

pritisnimo *Enter*, pa će MATLAB ispisati:

```
Warning: Explicit integral could not be found.
> In d:\matlabr12\toolbox\symbolic\@sym\int.m at line 58
ans =
int(acot(x)/(1+x^2)^(1/2),x = 0 .. inf)
```

Čini se da zadani integral ne konvergira i da je zadatak gotov. Međutim, ako ispred funkcije `int` zapišemo funkciju `double`, tj. utipkamo li:

```
double(int(acot(x)/sqrt(x^2+1),0,inf))
```

dobit ćemo:

```
Warning: Explicit integral could not be found.
> In d:\matlabr12\toolbox\symbolic\@sym\int.m at line 58
ans =
   1.83193118835444
```

Dakle, zadani integral ipak konvergira i približno je jednak 1.832! Razlog zbog kojega „u prvom pokušaju" nismo dobili taj rezultat jest relativno složen. Naime, zadani nepravi integral jednak je $2 \cdot K$, gdje je K tzv. Catalanova konstanta definirana kao zbroj reda

$$K = \sum_{k=0}^{+\infty} \frac{(-1)^k}{(2 \cdot k+1)^2} = 1 - \frac{1}{3^2} + \frac{1}{5^2} - \frac{1}{7^2} + .. \approx 0.915965594177.$$

Pitanje (i)racionalnosti Catalanove konstante danas je još uvijek neriješen problem. Njegovo bi ispravno rješenje originalnom autoru donijelo Fieldsovu medalju, svjetsku slavu i pristojnu svotu američkih dolara. (Ima li možda među vama zainteresiranih za rješavanje ovoga problema?)

Ovu točku zaključujemo s integralom poznatim iz *Matematike* 2.

Primjer 10. Neka je $p \in \mathbf{R}$. Ispitajte konvergenciju nepravoga integrala $\int_0^1 \frac{dx}{x^p}$.

U nova dva retka komandnoga prozora utipkajmo:

```
syms x p
int(1/x^p,0,1)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
limit((x-exp(p*log(x)))/(-1+p)/exp(p*log(x)),x = 0,right)
```

Kao rezultat smo dobili nekakvu jednostranu graničnu vrijednost. Može li je MATLAB barem približno izračunati? Kopirajmo dobiveni odgovor u novi redak našega komandnoga prozora i malo ga preuredimo tako da dobijemo:

```
limit((x-exp(p*log(x)))/(-1+p)/exp(p*log(x)),x,0,'right')
```

Pritisnimo *Enter*. Ništa od njega u ovom zadatku: opet je ispisao „stari" izraz. To je i razumljivo jer zadani integral konvergira za $p < 1$, a divergira za $p \geq 1$. Budući da MATLAB nema nikakve dodatne informacije o vrijednosti parametra p, a nije dovoljno mudar da nam ispisuje kad *bi* integral *mogao* konvergirati, dobili smo navedeni rezultat. U ovom je slučaju analitičko određivanje nepravoga integrala podesnije od računalnoga.

5.4. Dodatak: Preinačeno rješavanje (ne)algebarskih jednadžbi[16]

Jedan od ozbiljnijih nedostataka funkcije `solve` jest što prigodom njezina pozivanja moramo napisati cjelovitu jednadžbu koju želimo riješiti. Dosad smo taj problem rješavali staromodnom *kopiraj–zalijepi* (engl.: *copy – paste*) tehikom. U ovoj ćemo točki kratko opisati kako lukavo možemo izbjeći navedeni problem.

Za programerski nastrojene čitatelje kažimo da je osnovna ideja ovoga preinačenoga načina rješavanja (ne)algebarskih jednadžbi najprije pohraniti lijevu, odnosno desnu stranu polazne jednadžbe u tzv. *polje znakova* (engl: *string*), a potom „slijepiti" (konkatenirati) tako dobivene nizove znakova u jedan simbolički objekt i na njega primijeniti funkciju `solve`. MATLAB-ova funkcija koja simbolički objekt pretvara u polje znakova jest funkcija `char`, a funkcija koja horizontalno sljepljuje barem dva niza znakova jest `strcat`.[17]

Bez smanjenja općenitosti pretpostavit ćemo da je nepoznanica po kojoj rješavamo jednadžbu označena s x. Metodu za preinačeno rješavanje (ne)algebarskih jednadžbi implementirat ćemo u funkcijskoj *m*-datoteci `rijesi.m`. U MATLAB-ovu komandnom prozoru otvorimo novu *m*–datoteku i u nju upišimo sljedeće nizove naredbi:

[16] Ova točka nastala je na temelju ideja i poticaja kolege-nastavnika Luke Marohnića, na čemu mu iskreno zahvaljujem.
[17] Nizove znakova moguće je „slijepiti" i vertikalno koristeći funkciju `strvcat`.

```
function r=rijesi(f,g);
syms x;
F=char(f);
G=char(g);
h=strcat(F,'=',G);
r=double(solve(h));
```

Uočimo strukturu ovoga niza naredbi: ulazni podatci su lijeva i desna strana jednadžbe koju želimo riješiti, što znači da desna strana jednadžbe može biti različita od nule. Najprije lijevu stranu jednadžbe zapišemo kao niz znakova. Potom to isto učinimo i s desnom stranom jednadžbe, pa slijepimo tako dobivene nizove znakova i na dobiveni simbolički objekt primijenimo funkciju `solve`. Budući da kao rješenja želimo dobiti realne ili kompleksne brojeve, dobivene simboličke brojeve trebamo pretvoriti u „obične" realne brojeve pomoću funkcije `double`. Svi tako dobiveni brojevi bit će pohranjeni u izlaznu varijablu r.

Ovdje treba napomenuti da desna strana jednadžbe (tj. funkcija g) obavezno treba sadržavati varijablu x. Npr. ako je na desnoj strani jednadžbe prirodan broj n (tj. ako je $g(x) = n$), prigodom izvršenja funkcije `char` MATLAB će kao niz znakova G pohraniti znak čiji je ASCII-kod jednak n, što će uzrokovati nemogućnost rješavanja jednadžbe. No, takvu „opasnost" možemo lako ukloniti zapišemo li funkciju $g(x) = n$ u obliku $g(x) = 0 \cdot x + n$.

Pohranimo navedeni niz naredbi kao datoteku `rijesi.m` i vratimo se u komandni prozor. Pogledajmo primjenu dobivene funkcije na dvama primjerima.

***Primjer* 1.** Odredite zajedničke točke grafova polinoma $p_1(x) = x^3 + 6$ i $p_2(x) = 2 \cdot x^2 + 5 \cdot x$.

Apscise traženih točaka su rješenja jednadžbe $p_1(x) = p_2(x)$. Stoga u nova dva retka komandnoga prozora najprije utipkajmo:

```
p1=x^3+6;
p2=2*x^2+5*x;
```

Potom primijenimo funkciju *rijesi* na polinome p_1 i p_2. U novi redak komandnoga prozora utipkajmo:

```
rijesi(p1,p2)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
     1
     3
    -2
```

Dakle, apscise traženih točaka su $x_1 = -2$, $x_2 = 1$ i $x_3 = 3$. Ordinate tih točaka dobijemo uobičajeno koristeći funkciju `subs`. U novi redak komandnoga prozora utipkajmo:

```
y1=subs(p1,-2), y2=subs(p1,1), y3=subs(p1,3)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
y1 =
    -2
y2 =
     7
y3 =
    33
```

Dakle, tražene točke su redom $T_1 = (-2, -2)$, $T_2 = (1, 7)$ i $T_3 = (3, 33)$.

Primjer 2. Riješite jednadžbu: $4 \cdot \text{arctg}\, x + x - (\pi + 1) = 0$.

„Očistimo“ svoj komandni prozor pomoću funkcije `clc`. U sljedeća tri njegova retka utipkajmo:

```
f=4*atan(x)+x-(pi+1);
g=0*x;
rijesi(f,g)
```

Ovakav redoslijed smo namjerno odabrali tako da ilustriramo što upisati ako je desna strana jednadžbe 0. Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
      1
```

Isto rješenje dobili bismo i ako bismo polaznu jednadžbu zapisali u obliku

$$4 \cdot \text{arctg}\, x + x = \pi + 1.$$

U tom bismo slučaju upisali:

```
f=4*atan(x)+x;
g=0*x+pi+1;
rijesi(f,g)
```

pa bi nakon pritiska tipke *Enter* MATLAB ponovno ispisao:

```
ans =
      1
```

5.5. Zadatci za vježbu

1. Odredite prve tri derivacije sljedećih realnih funkcija:

a) $f(x)=\dfrac{x^2}{e^{2-x}}$;

b) $f(x)=\dfrac{\arccos x}{x}$;

c) $f(x)=\left(\dfrac{\ln x}{x}\right)^2$;

d) $f(x) = x \cdot \operatorname{arsh} x$;

e) $f(x)=\dfrac{\sqrt{1+x^2}}{\sin x}$;

f) $f(x)=\dfrac{x}{\sqrt[3]{\sin x}}$.

Pojednostavnite dobivene izraze što je više moguće.

2. Ispitajte ima li svaka od sljedećih realnih funkcija lokalne ekstreme i, ako ima, odredite ih:

a) $f(x) = 6 \cdot x^5 + 15 \cdot x^4 - 130 \cdot x^3 - 210 \cdot x^2 + 720 \cdot x - 400$;

b) $f(x)=\dfrac{\operatorname{tg} x}{x}$;

c) $f(x)=\left(\dfrac{1}{2}\cdot\ln x\right)^2-1$;

d) $f(x)=\dfrac{x}{e^{x^2}}$;

e) $f(x)=\arcsin\sqrt{x}-\arccos\sqrt{x}$;

f) $f(x)=\sqrt[3]{(\operatorname{ch} x-\operatorname{sh} x)^2}$.

3. Odredite (u eksplicitnom obliku) jednadžbu tangente i jednadžbu normale povučene na krivulju K u točki T, pa izračunajte njihove duljine s točnošću od 10^{-5} ako je:

a) $K\ldots\ y = \ln^2 x,\ T = (1, y)$;
b) $K\ldots\ y = e^{2-x},\ T = (x, e)$;
c) $K\ldots\ y = e^{x^2-1},\ T = (x > 0, e^3)$;
d) $K\ldots\ 4\cdot x^2+25\cdot y^2=100,\ T=\left(x>0,-\dfrac{8}{5}\right)$;
e) $K\ldots\ \sqrt[3]{x^2}+\sqrt[3]{y^2}=5,\ T=(x>0,\ 8)$;
f) $K\ldots\ y^2=\dfrac{x^3}{1-x},\ T=\left(\dfrac{1}{2},\ y<0\right)$.

4. Odredite (ako postoje) sve prijevojne točke sljedećih krivulja:

a) $y = x - \operatorname{tg} x$;

b) $y = \operatorname{arcctg} \dfrac{1}{x+1}$;

c) $y = x - 2 \cdot \operatorname{arctg} x$;

d) $y = \operatorname{arth}(x + 1)$;

e) $y = (2 \cdot x^3 - 6 \cdot x^2 + 7 \cdot x - 4) \cdot e^{2 \cdot x}$;

f) $y = x \cdot \ln\left(1 - \dfrac{1}{x}\right)$.

5. Odredite prvu derivaciju funkcija $f \circ g$, $(f \circ g)^{-1}$, $g \circ f$ i $(g \circ f)^{-1}$ ako je:

a) $f(x) = \operatorname{tg} x$, $g(x) = \ln x$;

b) $f(x) = \dfrac{1}{1+x^2}$, $g(x) = \sqrt{x}$;

c) $f(x) = \sqrt[3]{x^2+1}$, $g(x) = e^{x^3}$;

d) $f(x) = \arcsin\sqrt{x}$, $g(x) = \dfrac{1}{x^2}$;

e) $f(x) = e^{\sqrt{x}}$, $g(x) = \sin^2 x$.

6. Izračunajte $(f \circ g)^{-1}(1)$, $\left[(f \circ g)^{-1}\right]'(1)$, $\left[(f \circ g)'\right]^{-1}(1)$, $(g \circ f)^{-1}(1)$, $\left[(g \circ f)^{-1}\right]'(1)$ i $\left[(g \circ f)'\right]^{-1}(1)$ ako je:

a) $f(x) = \dfrac{1}{\sqrt{x^2+1}}$, $g(x) = \ln\left(x^2\right)$;

b) $f(x) = \arccos\sqrt{x-1}$, $g(x) = x^2 + 1$;

c) $f(x) = \operatorname{arctg}^2 x$, $g(x) = \sqrt[4]{x}$;

d) $f(x) = -\operatorname{arcctg}(e^{x+1})$, $g(x) = \ln x - 1$;

e) $f(x) = \arcsin(\ln x)$, $g(x) = e^{x-1}$.

7. Zadane su realne funkcije $f(x) = x^2$ i $g(x) = \ln x$. Odredite sve realne nultočke funkcije $h(x) = (f \circ g - g \circ f)(x)$. Provjerite svoje rješenje grafički.

8. Zadane su realne funkcije $f(x) = \sqrt{x}$ i $g(x) = e^x$. Odredite sve realne nultočke funkcije $h = \left[(f \circ g)'\right]^{-1}$. Provjerite svoje rješenje grafički.

9. Zadane su realne funkcije $f(x) = \sin(2 \cdot x)$ i $g(x) = \frac{x}{2}$. Odredite barem jedno rješenje jednadžbe $\left[(g \circ f)'\right]^{-1} = \frac{\pi}{2}$. Provjerite svoje rješenje grafički.

10. Zadane su realne funkcije $f(x) = \cos\frac{x}{4}$ i $g(x) = 8 \cdot x$. Odredite barem jedno rješenje jednadžbe $\left[(f \circ g)'\right]^{-1} = \pi$. Provjerite svoje rješenje grafički.

11. Izračunajte sljedeće granične vrijednosti:

a) $\lim\limits_{x\to 1} \frac{4 \cdot \operatorname{arcctg} x - \pi}{2 \cdot (x-1)}$;

b) $\lim\limits_{x\to 0} (1 + 2 \cdot \operatorname{tg}^2 x)^{\operatorname{ctg}^2 x}$;

c) $\lim\limits_{x\to +\infty} \frac{2 \cdot x^3 + x^2 + x + 1}{x^3 + 2 \cdot x^2 + x + 1}$;

d) $\lim\limits_{x\to -\infty} \frac{2 + 3 \cdot x - x^2 - x^3}{3 \cdot x^3 + 2 \cdot x^2 - x - 1}$;

e) $\lim\limits_{x\to 0+} x^{\frac{1}{2+\ln x}}$;

f) $\lim\limits_{x\to 2-} (2-x)^{\cos\left(\frac{\pi}{4} \cdot x\right)}$.

12. Izračunajte (ako postoji) graničnu vrijednost niza $(a_n)_{n\in\mathbf{N}}$ čiji je opći član:

a) $a_n = \frac{\sin(n^{2012} + n + 1)}{2 \cdot n^3 + 1}$;

b) $a_n = \frac{|\sin n|}{n^2}$

c) $a_n = \frac{\operatorname{arctg}\left(\frac{1}{n}\right) \cdot (n+1)}{n^2 + n + 1}$;

d) $a_n = \frac{\left|\operatorname{arctg}(n^n + n) + \operatorname{arcctg}(n^n - n)\right|}{2012 \cdot n^3 + 1}$;

e) $a_n = \frac{2}{n^2} + \left(\frac{2}{n}\right)^2 + \ldots + \frac{2 \cdot (n-1)}{n^2}$;

f) $a_n = \dfrac{\sqrt[6]{n+1} - \sqrt[3]{n-1}}{\sqrt[6]{n-1} + \sqrt{n}}$.

13. Za svaki $n \in \{1, 2, 3, 4, 5\}$ odredite derivaciju reda n sljedećih realnih funkcija:

a) $f(x) = \dfrac{2012}{(x-2011)^2}$;
b) $f(x) = \ln \dfrac{2012}{x^4}$;
c) $f(x) = e^{-x} \cdot \sin(2 \cdot x)$;
d) $f(x) = \arccos x$;
e) $f(x) = x \cdot (\text{arsh } x - \text{arch } x)$;
f) $f(x) = \text{arcth } x$.

14. Odredite sve kose asimptote (uračunavajući i horizontalne) sljedećih krivulja:

a) $y = x - 2011 + \dfrac{2011 \cdot x^2}{\sqrt{x^2 + 2011}}$;
b) $y = \dfrac{e^x}{1-x}$;
c) $y = \dfrac{2011 \cdot x}{2011 - e^{2011 \cdot x}}$;
d) $y = \text{arctg} \dfrac{x}{2-x}$;
e) $y = x - 2 \cdot \text{arcctg } x$;
f) $y = x + \text{cth } x$.

15. Kreirajte *m*–datoteku **horas.m** koja će sadržavati jedino funkciju *horas* čiji je ulazni parametar realna funkcija *f*. Funkcija *horas* treba ispisati eksplicitni oblik jednadžbi svih horizontalnih asimptota na graf realne funkcije *f*, pri čemu uz svaku pojedinu asimptotu treba ispisati je li riječ o lijevoj, desnoj ili obostranoj horizontalnoj asimptoti. Ako graf funkcije *f* nema horizontalnih asimptota, funkcija treba ispisati tekst *Nema niti jedne horizontalne asimptote*.

16. Kreirajte *m*–datoteku **kosas.m** koja će sadržavati jedino funkciju *kosas* čiji je ulazni parametar realna funkcija *f*. Funkcija *kosas* treba ispisati eksplicitni oblik jednadžbi svih pravih (nehorizontalnih) kosih asimptota na graf funkcije *f*, pri čemu uz svaku pojedinu asimptotu treba ispisati je li riječ o lijevoj, desnoj ili obostranoj kosoj asimptoti. Ako graf funkcije *f* nema kosih asimptota, funkcija treba ispisati tekst *Nema niti jedne kose asimptote*.

17. Provjerite je li funkcija F primitivna funkcija funkcije f ako je zadano:

a) $F(x) = \dfrac{x-1}{x+1}, f(x) = 2012 + \left(\dfrac{\sqrt{2}}{x+1}\right)^2$;

b) $F(x) = (x+2) \cdot \ln(2 \cdot x) + e^{2010}, f(x) = \dfrac{x \cdot (1+\ln 2) + \ln x + 2}{x}$;

c) $F(x) = \operatorname{ctg}^3(2012 \cdot x) + \ln(\sin 2012), f(x) = \dfrac{6036 \cdot \cos^2(2012 \cdot x)}{\sin^4(2012 \cdot x)}$;

d) $F(x) = e^x \cdot \operatorname{ctg}(2 \cdot x) - \operatorname{tg}(e^{2012}),\ f(x) = \dfrac{e^x \cdot [\sin(2 \cdot x) \cdot \cos(2 \cdot x) - 2]}{\sin^2(2 \cdot x)}$;

e) $F(u) = u \cdot \arcsin u + \sqrt{1-u^2}$, $f(u) = \arcsin u$;

f) $F(x) = x \cdot \operatorname{arccth} x + \dfrac{1}{2} \cdot \ln\left(x^2+1\right), f(x) = \operatorname{arccth} x$.

18. Odredite sljedeće neodređene integrale:

a) $\displaystyle\int x \cdot \sqrt[3]{x^2 \cdot \sqrt[4]{x^3}} \cdot \left(\frac{\sqrt{x}}{x} - \frac{\sqrt[3]{x^2}}{x^2}\right)^3 \cdot dx$;

b) $\displaystyle\int \frac{\left(\sqrt{2}+\sqrt{x}\right) \cdot (2+x) \cdot \left(\sqrt{x}-\sqrt{2}\right)}{\left(\sqrt{x}+1\right) \cdot (x+1) \cdot \left(1-\sqrt{x}\right)} \cdot dx$;

c) $\displaystyle\int \operatorname{arcctg} \sqrt[3]{x} \cdot dx$;

d) $\displaystyle\int \frac{72 \cdot x^5 + 36 \cdot x^4 - 135 \cdot x^3 - 65 \cdot x^2 - 85 \cdot x - 36}{36 \cdot x^5 - 65 \cdot x^3 - 36 \cdot x} \cdot dx$;

e) $\displaystyle\int \frac{x^4 + x^3 + x^2 + x + 1}{\sqrt{2 + x - x^2}} \cdot dx$;

f) $\displaystyle\int \frac{\cos x}{(3 - \sin x) \cdot (4 + \sin x)} \cdot dx$.

19. Izračunajte sljedeće određene integrale:

a) $\displaystyle\int_{-1}^{1} \sqrt{7 - 3 \cdot x^2} \cdot dx$;

b) $\displaystyle\int_{0}^{1} x \cdot e^{2 \cdot x} \cdot dx$;

c) $\int_{1}^{e} \ln^2 x \cdot dx$;

d) $\int_{\frac{\pi}{2}}^{\pi} \frac{\sin x \cdot \cos x \cdot dx}{\sqrt{\cos^2(2 \cdot x) + \cos(2 \cdot x)}}$;

e) $\int_{0}^{1} \text{sh}^3 x \cdot dx$;

f) $\int_{0}^{1} \text{arth}\, x \cdot dx$;

20. Izračunajte prosječnu vrijednost realne funkcije f na njezinu prirodnu području definicije ako je:

a) $f(x) = \arcsin x$;
b) $f(x) = \arccos(2 \cdot x)$;
c) $f(x) = \sqrt{4 - x^2}$;
d) $f(x) = \sqrt{2 \cdot x - x^2}$;
e) $f(x) = \sqrt{8 - 7 \cdot x - x^2}$;
f) $f(x) = x \cdot e^{-\sqrt{x}}$.

21. Izračunajte površinu ravninskoga lika omeđenoga krivuljama:

a) $y = \cos x$, $y = \sin x$, $x = 1$ i $x = \frac{\pi}{4}$;
b) $y = x^2 - x$ i $y = x - x^2$;
c) $y = \frac{1}{2} + \ln\sqrt{1 - x^2}$ i $y = 0$;
d) $y = \frac{1}{x^2 + 1}$, $y = 1 + \ln(x + 1)$ i $x = 2$;
e) $y = \arcsin\sqrt{1 - x^2}$ i $y = 0$
f) $y = \arcsin\frac{x}{2}$ i $\pi \cdot x - 4 \cdot y = 0$.

22. Izračunajte duljinu luka krivulje $y = \frac{a}{2}\left(e^{\frac{x}{a}} + e^{-\frac{x}{a}}\right)$ nad segmentom $[0, a]$, gdje je $a > 0$ proizvoljan, ali fiksiran realan broj.

23. Izračunajte obujam rotacijskoga tijela koje nastaje rotacijom lika omeđenoga krivuljama $y^2 = x^3$, $y = 0$ i $x = 1$ oko osi ordinata.

24. Izračunajte obujam rotacijskoga tijela koje nastaje rotacijom krivulje $x^2 + 4 \cdot y^2 = 4$ oko osi apscisa.

25. Ispitajte konvergenciju sljedećih nepravih integrala pa, ako konvergiraju, izračunajte ih:

a) $\int\limits_0^{+\infty} x \cdot e^{-x^2} dx$;

b) $\int\limits_{e^3}^{+\infty} \frac{dx}{x \cdot (\ln^2 x - \ln x - 2)}$;

c) $\int\limits_{-\infty}^{0} x^2 \cdot e^{x^3} dx$;

d) $\int\limits_{-\infty}^{-3} \frac{6 \cdot e^x \cdot dx}{5 - 4 \cdot e^x - e^{2 \cdot x}}$;

e) $\int\limits_{-\infty}^{+\infty} \frac{dx}{x^2 + x + 1}$;

f) $\int\limits_{-\infty}^{+\infty} \operatorname{arctg} x dx$.

26. Ispitajte konvergenciju sljedećih nepravih integrala, pa, ako konvergiraju, izračunajte ih:

a) $\int\limits_0^1 \frac{dx}{\sqrt[2012]{x}}$;

b) $\int\limits_1^e \frac{dx}{x \ln^2 x}$;

c) $\int\limits_0^2 \frac{dx}{\sqrt{x} - 1}$;

d) $\int\limits_{-1}^1 \frac{dx}{\sqrt[3]{x} - 1}$

e) $\int\limits_0^1 \frac{e^x \cdot dx}{\sqrt{e^x - 1}}$;

f) $\int\limits_0^1 \frac{dx}{\sqrt{x} - x}$.

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

§6. REDOVI BROJEVA. REDOVI FUNKCIJA.

U prethodnim smo se poglavljima upoznali s nizovima i računanjem granične vrijednosti niza (ako je postojala). U ovom ćemo se poglavlju pozabaviti s redovima brojeva (numeričkim redovima), te nekim poznatim redovima funkcija. Poseban naglasak stavit ćemo na implementaciju poznatih kriterija za ispitivanje konvergencije redova brojeva, određivanje Taylorova razvoja u red realne funkcije f oko neke točke iz njezina prirodnoga područja definicije, te određivanje konačno mnogo Fourierovih koeficijenata u razvoju periodične realne funkcije f u Fourierov red (ali bez provjere Dirichletovih uvjeta).

6.1. Redovi brojeva.

Općenito izračunavanje zbrojeva konvergentnih redova je vrlo teško. No, MATLAB ima implementiranu funkciju `symsum` koja nam omogućuje izračunavanje zbroja nekih konvergentnih redova definiranih kao simbolički objekti. Ista funkcija omogućuje i izračunavanje zbroja prvih n članova konvergentnoga reda.

Pogledajmo primjenu navedene funkcije na primjerima.

***Primjer* 1.** Analitički i pomoću MATLAB-a izračunajmo zbroj reda $\sum_{n=0}^{+\infty}\left(\frac{e+1}{\pi+1}\right)^{e\cdot n}$. Zapišimo dobiveni rezultat u znanstvenom obliku uz pretpostavku da mantisa ima točno 6 znamenaka.

Najprije primijetimo da vrijedi jednakost

$$\sum_{n=0}^{+\infty}\left(\frac{e+1}{\pi+1}\right)^{e\cdot n}=\sum_{n=0}^{+\infty}\left[\left(\frac{e+1}{\pi+1}\right)^{e}\right]^{n},$$

pa zbog nejednakosti (provjerite je!)

$$\frac{e+1}{\pi+1}<1$$

slijedi da je zadani red konvergentan geometrijski red kojemu je prvi član $a_1 = 1$, a količnik $q=\left(\frac{e+1}{\pi+1}\right)^{e}$. Stoga je zbroj toga reda jednak:

$$S=\frac{a_1}{1-q}=\frac{1}{1-\left(\frac{e+1}{\pi+1}\right)^{e}}\approx 3.93641\,.$$

Riješimo isti zadatak koristeći MATLAB. Ako dosad niste, obavezno „počistite“ svoj komandni prozor koristeći funkciju `clc`. U nova dva retka toga prostora utipkajmo:

```
syms n
symsum(((exp(1)+1)/(pi+1))^(exp(1)*n),n,0,inf)
```

Uočimo sintaksu simboličke funkcije `symsum`. Najprije pišemo opći član reda, potom oznaku nezavisne varijable (u ovom slučaju n) i, naposljetku, granice sumacije.

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
??? Error using ==> sym/maple
Error, (in sum/indefnew) integer too large in context
Error in ==> d:\matlabr12\toolbox\symbolic\@sym\symsum.m
On line 43  ==>    r = maple('map','sum',f,[x.s '=' a.s '..' b.s]);
```

Što se dogodilo i zašto smo dobili ovakvu poruku? Naime, brojevi e i π, definirani na gornji način, su „preglomazni“ za funkciju `symsum` i ona ne može ništa raditi s njima. Stoga ćemo se poslužiti uobičajenim „trikom“: količnik reda q definirat ćemo zasebno, pa ga uvrstiti kao argument funkcije `symsum`. Dakle, u nova dva retka komandnoga prozora utipkamo:

```
q=(((exp(1)+1)/(pi+1))^(exp(1)));
symsum(q^n,n,0,inf)
```

Nakon što pritisnemo *Enter*, slijedi ugodno iznenađenje:

```
ans =
140737488355328/35752781694167
```

Hitro.hr pozovemo funkciju `double` da nam dobiveni razlomak pretvori u decimalan broj. Utipkamo

```
double(ans)
```

i pritisnemo *Enter*, pa će MATLAB ispisati približnu vrijednost koju smo već ranije dobili:

```
ans =
   3.93641
```

Primjer 2. Izračunajmo zbroj reda $\sum_{n=0}^{+\infty}(-1)^n \cdot \left(\frac{\cos 1 + \operatorname{ch} 2}{\sin 3 + \operatorname{sh} 4}\right)^{(\operatorname{ctg} 5 + \operatorname{cth} 6)\cdot n}$. Zapišimo dobiveni rezultat u znanstvenom obliku uz pretpostavku da mantisa ima točno 6 znamenaka.

U ovom slučaju je ponovno $a_1 = 1$, dok je $q = (-1)\cdot\left(\frac{\cos 1 + \operatorname{ch} 2}{\sin 3 + \operatorname{sh} 4}\right)^{\operatorname{ctg} 5 + \operatorname{cth} 6}$. Analitičko rješenje ovoga zadatka prepuštamo čitateljima. I ovoga ćemo puta posebno zadati količnik q, pa postupiti kao u prethodnom primjeru. Dakle, u nova dva retka MATLAB-ova komandnoga prozora utipkamo:

```
q=((cos(1)+cosh(2))/(sin(3)+sinh(4)))^(cot(5)+coth(6));
symsum(q^n,n,0,inf)
```

Pritisnemo *Enter*, pa će MATLAB ispisati još jedan simpatičan razlomak:

```
ans =
18014398509481984/13127017543403183
```

Pomoću funkcije `double` dobivamo traženi znanstveni zapis rješenja. Dakle, utipkamo

```
double(ans)
```

pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
   1.37231
```

Primjer 3. Izračunajmo vrijednost zbroja reda $\sum_{n=0}^{+\infty} \frac{2^n}{n!}$.

Iz *Matematike* 2 znamo da za svaki $x \in \mathbf{R}$ vrijedi jednakost

$$e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!},$$

pa uvrštavanjem $x = 2$ u navedenu jednakost dobijemo:

$$\sum_{n=0}^{+\infty} \frac{2^n}{n!} = e^2.$$

Izvedimo ovaj rezultat koristeći MATLAB. Najprije razriješimo mali problem vezan uz zadavanje funkcije $f(n) = n!$ kao simboličke funkcije. Definirajmo je u novom retku radnoga prozora kao simboličku funkciju *nfact* s:

```
nfact=sym('n!');
```

Traženi zbroj reda dobivamo utipkavajući:

```
symsum(2^n/nfact,n,0,inf)
```

u novi redak komandnoga prozora. Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
exp(2)
```

Pomoću MATLAB-a možemo ne samo izračunati točne vrijednosti zbrojeva nekih redova brojeva, nego i brzo i jednostavno razriješiti neke aproksimacijske probleme vezane uz zamjenu zbroja reda zbrojem prvih n članova toga reda. Pogledajmo dva takva problema.

Primjer 4. Koristeći Cauchyjev kriterij (vidjeti točku 6.2.) može se pokazati da red $\sum_{n=1}^{+\infty}\frac{2^n\cdot n!}{n^n}$ konvergira. Odredimo apsolutnu vrijednost pogrješke koju činimo kad točnu vrijednost zbroja reda zamijenimo zbrojem prvih 100 članova toga reda.

U novi redak komandnoga prozora utipkamo

```
double(symsum(2^n*nfact/n^n,n,1,inf))-double(symsum(2^n*nfact/n^n,n,1,100))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
    3.355538069627073e-012
```

Dakle, pogrješka je reda veličine $3.35554\cdot 10^{-12}$. Provjerite da utipkavanjem

```
symsum(2^n*nfact/n^n,n,1,inf)
```

ne dobivamo nikakav „konkretan“ rezultat. Stoga utipkajmo:

```
double(symsum(2^n*nfact/n^n,n,1,inf))
```

u novi redak komandnoga prozora. Pritisnimo *Enter*, pa će MATLAB ispisati približnu vrijednost zbroja polaznoga reda:

```
ans =
  12.94895036267534
```

Primjer 5. Poznato nam je da harmonijski red $\sum\frac{1}{n}$ vrlo sporo divergira. Odredimo najmanji prirodan broj n tako da zbroj prvih n članova toga reda bude strogo veći od 17.

Ovaj je primjer bolje i praktičnije riješiti koristeći običnu m-datoteku. Nazovimo tu datoteku s **primjer5.m**. Otvorimo je u novom prozoru, pa utipkajmo sljedeći niz naredbi:

```
n=1;
z=0;
while z<=17
    z=z+1/n;
    n=n+1;
end
n-1
```

(Objasnite svaki redak ove *m*–datoteke.) Pohranimo upisane naredbe i vratimo se u komandni prozor. U njegov novi redak utipkamo

```
primjer5
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
n =
    13562027
```

Dakle, treba zbrojiti „samo“ prvih 13 562 027 članova zadanoga reda da se (prvi put) dobije zbroj strogo veći od 17.

6.2. Kriteriji konvergencije redova brojeva

U *Matematici* 2 naučili smo ispitivati konvergenciju redova brojeva pomoću nekoliko kriterija (Cauchyjev, D'Alembertov, Raabeov, Leibnizov, integralni kriterij itd.). Ovdje ćemo ukratko prikazati implementaciju prvih triju spomenutih kriterija u MATLAB-u.

***Primjer* 1.** Kreirajmo funkcijsku datoteku **cauchy.m** koja sadrži jedino funkciju *cauchy* čija je jedina ulazna varijabla opći član reda $\sum a_n$. Funkcija treba ispisati je li red konvergentan ili divergentan prema Cauchyjevu kriteriju, odnosno tekst „Nema odluke prema Cauchyjevu kriteriju.“ inače. Provjerimo ispravnost svojega rješenja na primjeru reda $\sum\left(\frac{2\cdot n+1}{3\cdot n}\right)^{\pi\cdot n}$.

Podsjetimo da red $\sum a_n$ konvergira prema Cauchyjevu kriteriju ako je $r := \lim\limits_{n\to+\infty}\sqrt[n]{|a_n|} < 1$, dok u slučaju $r = 1$ kriterij ne daje odluku. Stoga otvorimo novu *m*–datoteku, pa u nju utipkajmo:

```
function y=cauchy(red);
syms n;
b=red^(1/n);
r=double(abs(limit(b,n,inf)));
if r<1
    'Zadani red je konvergentan.'
else
    if r>1
        'Zadani red je divergentan.'
    else
        'Nema odluke prema Cauchyjevu kriteriju.'
    end
end
```

Detaljno objasnite svaki redak ove *m*–datoteke. Uočite korisnost funkcije `double` koja simbolički objekt (graničnu vrijednost) pretvara u numeričku varijablu s kojom se potom dalje može lako računati.

Pohranimo dobivenu datoteku pod nazivom **cauchy.m** i vratimo se u MATLAB-ov komandni prozor. Ako je potrebno, „očistite“ taj prozor od rezultata prethodnih primjera/zadataka koristeći funkciju `clc`. U nova tri retka toga prozora utipkajmo:

```
syms a n;
a=((2*n+1)/(3*n))^(pi*n);
cauchy(a)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
Zadani red je konvergentan.
```

(Za vježbu riješite ovaj primjer i analitički.) Potom prepravite izraz za *a* u:

```
a=1/n;
```

i ponovno pokrenite funkciju *cauchy*. Ovoga puta MATLAB će ispisati:

```
ans =
Nema odluke prema Cauchyjevu kriteriju.
```

Takav rezultat smo, naravno, mogli i predvidjeti jer se divergencija harmonijskoga reda ne može utvrditi pomoću Cauchyjeva (ali niti D'Alembertova i Raabeova!) kriterija.

Primjer 2. Kreirajmo funkcijsku datoteku **dalembert.m** koja sadrži jedino funkciju *dalembert* čija je jedina ulazna varijabla opći član reda $\sum a_n$. Funkcija treba ispisati je li red konvergentan ili divergentan prema D'Alembertovu kriteriju, odnosno tekst „Nema odluke prema zadanu kriteriju.“ inače. (Utvrdite zašto ne možemo ispisati tekst „Nema odluke prema D'Alembertovu kriteriju“.) Provjerimo ispravnost rada svojega rješenja na primjeru reda $\sum \frac{e \cdot n+\pi}{\pi \cdot n+e}$.

Podsjetimo da red $\sum a_n$ konvergira prema D'Alembertovu kriteriju ako je $r := \lim_{n\to+\infty}\left|\frac{a_{n+1}}{a_n}\right| < 1$, te da u slučaju $r = 1$ kriterij ne daje odluku. Otvorimo novu *m*–datoteku, pa u nju utipkajmo:

```
function r=dalembert(red);
syms n;
x=subs(red,n+1);
y=red;
```

```
a=double(abs(limit(x/y,n,inf)));
if a<1
    'Zadani red je konvergentan.'
else
    if a>1
        'Zadani red je divergentan.'
    else
        'Nema odluke prema zadanu kriteriju.'
    end
end
```

Detaljno objasnite svaki redak ove funkcijske *m*-datoteke. Uočimo korisnost funkcije `subs` pomoću koje možemo odrediti (n + 1)-vi član niza koji generira zadani red (tj. kraće i nepreciznije: (n + 1)-vi član reda).

Pohranimo dobivenu *m*-datoteku pod nazivom **dalembert.m** i vratimo se u MATLAB-ov komandni prozor. U nova dva retka toga prostora utipkajmo:

```
a=(exp(1)*n+pi)/(pi*n+exp(1));
dalembert(a)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
Zadani red je konvergentan
```

(Za vježbu riješite zadatak i analitički.) Potom prepravite izraz za a u:

```
a=1/n
```

pa ponovno pokrenite funkciju `dalembert`. Prema očekivanju, MATLAB će ispisati:

```
ans =
Nema odluke prema zadanu kriteriju.
```

Primjer 3. Kreirajmo funkcijsku datoteku **raabe.m** koja sadrži jedino funkciju *raabe* čija je jedina ulazna varijabla opći član reda $\sum a_n$. Funkcija treba ispisati je li red konvergentan ili divergentan prema Raabeovu kriteriju, odnosno tekst „Nema odluke prema zadanu kriteriju." inače. Provjerimo ispravnost rada svojega rješenja na primjeru reda $\sum \frac{2 \cdot n+3}{(n^2+5 \cdot n+4)^2}$.

Podsjetimo da red $\sum a_n$ konvergira prema Raabeovu kriteriju ako vrijedi $r := \lim\limits_{n \to +\infty} \left| n \cdot \left(1 - \frac{a_{n+1}}{a_n}\right) \right| > 1$, te da u slučaju r = 1 kriterij ne daje odluku. Otvorimo novu *m*–datoteku, pa u nju utipkajmo:

```
function y=raabe(red);
syms n;
b=subs(red,n+1);
c=red;
r=double(abs(limit(n*(1-a/b),n,inf)));
if r>1
    'Zadani red je konvergentan.'
else
    if r<1
        'Zadani red je divergentan.'
    else
        'Nema odluke prema Raabeovu kriteriju.'
    end
end
```

Detaljno objasnite svaki redak ove *m*–datoteke. Pohranimo dobivenu *m*–datoteku pod nazivom **raabe.m**, pa se vratimo u MATLAB-ov komandni prozor. U nova dva retka toga prostora utipkajmo:

```
a=(2*n+3)/(n^2+5*n+4)^ 2;
raabe(a)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
Zadani red je konvergentan.
```

(Za vježbu riješite zadatak i analitički.) Zadani je red primjer reda koji konvergira prema Raabeovu kriteriju, dok Cauchyjev i D'Alembertov kriterij ne daju odluku. Provjerite tu tvrdnju pokretanjem odgovarajućih funkcijskih datoteka.

Napomena: Često se pogrešno smatra da je kod Cauchyjeva i D'Alembertova kriterija valjana ekvivalencija: (red konvergira) $\Leftrightarrow$ ($r < 1$), a kod Raabeova kriterija ekvivalencija (red konvergira) $\Leftrightarrow$ ($r > 1$). Te ekvivalencije nisu istinite jer za svaki pojedini kriterij postoje primjeri konvergentnih redova takvih da je $r = 1$. Zbog toga smo pisali da slučaj $r = 1$ ne daje odluku. Detaljnije o tome može se naći u [2] i [8], a spomenuti primjeri u [3].

6.3. Redovi funkcija. Redovi potencija.

Osim zbroja reda brojeva, pomoću MATLAB-a možemo izračunavati i različite zbrojeve redova funkcija. U tu svrhu također koristimo dobro poznatu funkciju `symsum`, pri čemu moramo pripaziti da unaprijed deklariramo sve potrebne simboličke objekte (najčešće nezavisnu varijablu x i indeks sumacije n). Pogledajmo nekoliko primjera.

Primjer 1. Izračunajmo zbroj reda $\sum_{n=1}^{+\infty} \frac{x^n}{n}$, pa odredimo za koji $x \in \mathbf{R}$ taj zbroj iznosi ln 2.

Analitički bismo ovaj zadatak riješili integriranjem jednakosti $\sum_{n=0}^{+\infty} x^n = \frac{1}{1-x}$ član po član. U MATLAB-u je posao daleko jednostavniji. Prije rješavanja svakako „počistite" svoj komandni prozor. U nova dva retka toga prozora utipkajmo:

```
syms x n
symsum(x^n/n,n,1,inf)
```

Pritisnimo *Enter*, pa će MATLAB ispisati traženi zbroj:

```
ans =
-log(1-x)
```

Doista,

$$\sum_{n=1}^{+\infty} \frac{x^n}{n} = \int \sum_{n=0}^{+\infty} x^n = \int \frac{1}{1-x} \cdot dx = -\ln(1-x).$$

Preostaje riješiti jednadžbu

$$-\ln(1-x) = \ln 2.$$

Tu jednadžbu puno brže možemo riješiti analitički, negoli pomoću MATLAB-a, ali u ovom slučaju brzinu rješavanja zanemarujemo. U novom retku MATLAB-ova komandnoga prozora utipkajmo:

```
x=solve('-log(1-x)=log(2)')
```

Pritisnimo *Enter*, pa će MATLAB ispisati traženo rješenje:

```
x =
1/2
```

Dakle, za $x = \frac{1}{2}$ zbroj zadanoga reda jednak je ln 2.

Primjer 2. Izračunajmo zbroj reda $\sum_{n=1}^{+\infty} n \cdot x^n$, pa odredimo za koji $x \in \mathbf{R}$ taj zbroj iznosi 2.

Analitički bismo ovaj zadatak riješili deriviranjem jednakosti $\sum_{n=0}^{+\infty} x^n = \frac{1}{1-x}$ (lijevu stranu deriviramo član po član, a desnu kao racionalnu funkciju) i množenjem dobivene jednakosti s x. (Učinite to za vježbu!) U novom retku MATLAB-ova komandnoga prozora utipkajmo:

```
symsum(n*x^n,n,0,inf)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
x/(x-1)^2
```

Dakle,

$$\sum_{n=1}^{+\infty} n \cdot x^n = \frac{x}{(x-1)^2}.$$

Preostaje riješiti jednadžbu

$$\frac{x}{(x-1)^2} = 2.$$

U novi redak MATLAB-ova komandnoga prozora utipkajmo:

```
solve('x/(x-1)^2=2')
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
[ 1/2]
[   2]
```

Na prvi bismo pogled zaključili da je zbroj zadanoga reda jednak 2 za $x \in \left\{\frac{1}{2},\ 2\right\}$. Međutim, to nije točno. Lako vidimo da za x = 2 dobivamo red $\sum_{n=1}^{+\infty} n \cdot 2^n$ takav da je $\lim_{n\to+\infty} a_n = \lim_{n\to+\infty} n \cdot 2^n = +\infty$, pa primjenom nužnoga uvjeta konvergencije reda[18] zaključujemo da za x = 2 polazni red divergira, odnosno da njegov zbroj ne može biti jednak 2. Stoga je je-

[18] Više o nužnom uvjeru konvergencije reda vidjeti npr. u [8], str.187.

dino rješenje drugoga dijela ovoga primjera $x = \frac{1}{2}$.

Primjer 3. Izračunajmo zbroj reda $\sum_{n=1}^{+\infty} \frac{n}{x^n}$, pa odredimo za koji $x \in \mathbf{R}$ taj zbroj iznosi 2.

Postupimo analogno kao u Primjeru 2. U novi redak komandnoga prozora utipkajmo:

```
symsum(n/x^n,n,1,inf)
```

Pritisnimo *Enter*, pa će MATLAB – na naše sveopće iznenađenje – ispisati:

```
ans =
x/(x-1)^2
```

U prvi trenutak pomišljamo da nešto nije u redu jer, zbog tranzitivnosti relacije „biti jednak“, iz rezultata Primjera 2. i dobivenoga rezultata slijedi „čudna“ jednakost funkcija:

$$\sum_{n=1}^{+\infty} n \cdot x^n = \sum_{n=1}^{+\infty} \frac{n}{x^n} = \frac{x}{(x-1)^2}.$$

Pokažimo da ta jednakost nije istinita niti za jedan $x \in \mathbf{R}$. Npr. pomoću Cauchyjeva kriterija lako se provjeri (učinite to sami) da jednakost

$$\sum_{n=1}^{+\infty} n \cdot x^n = \frac{x}{(x-1)^2}$$

vrijedi za $x \in \langle -1, 1\rangle$, dok jednakost

$$\sum_{n=1}^{+\infty} \frac{n}{x^n} = \frac{x}{(x-1)^2}$$

vrijedi za $x \in \mathbf{R} \setminus [-1, 1]$. Dakle, polazna jednakost može biti valjana jedino za $x \in \{-1, 1\}$. Za $x = -1$ dobivamo red $\sum_{n=1}^{+\infty} (-1)^n \cdot n$, dok za $x = 1$ dobivamo red $\sum_{n=1}^{+\infty} n$. Nije teško provjeriti da su oba ta reda divergentna (divergencija prvoga reda najlakše se utvrđuje pomoću Leibnizova kriterija[19], a divergencija drugoga iz činjenice da očito ne vrijedi nužan uvjet konvergencije reda.) To znači da polazna jednakost nije valjana niti za $x \in \{-1, 1\}$. Tako zaključujemo da ne postoji niti jedan $x \in \mathbf{R}$ za koji vrijedi polazna jednakost, a to smo i željeli pokazati.

[19] Više o Leibnizovu kriteriju vidjeti u [8], str. 193.

Drugi dio primjera riješimo isto kao u Primjeru 2., ali uvažavajući uvjet $x \in \mathbf{R} \setminus [-1, 1]$. Ponovno dobivamo $x \in \left\{\frac{1}{2}, 2\right\}$, ali ovoga puta odbacujemo rješenje $x = \frac{1}{2}$. Stoga je zbroj polaznoga reda jednak 2 samo za $x = 2$.

6.4. Razvoj realne funkcije u Taylorov red

Odrediti Taylorov razvoj neke realne funkcije f jedne realne varijable u okolini točke c iz prirodnoga područja definicije te funkcije pomoću MATLAB-a vrlo je jednostavno. Naime, MATLAB-ova funkcijska „knjižnica“ sadrži ugrađenu funkciju `taylor` koja omogućuje ispis upravo ovoga razvoja. Argumenti te funkcije mogu biti:

– samo realna funkcija f (u tom slučaju dobivamo tzv. *MacLaurinov polinom* (u varijabli x) stupnja najviše 5, tj. aproksimaciju funkcije f Taylorovim polinomom stupnja najviše 5 u okolini točke $c = 0$) ;

– realna funkcija f i prirodan broj n (u tom slučaju dobivamo MacLaurinov polinom (u varijabli x) stupnja najviše $n - 1$);

– realna funkcija f, prirodan broj n i simbolička varijabla v (u tom slučaju dobivamo MacLaurinov polinom (u varijabli v) stupnja najviše $n - 1$);

– realna funkcija f i realan broj a (u tom slučaju dobivamo Taylorov polinom stupnja najviše 5 u okolini točke a);

– realna funkcija f, prirodan broj n i realan broj a (u tom slučaju dobivamo Taylorov polinom stupnja najviše $n - 1$ u okolini točke a)

– realna funkcija f, prirodan broj n, simbolička varijabla v i realan broj a (u tom slučaju dobivamo Taylorov polinom stupnja najviše $n - 1$ u varijabli v oko točke a).

Uobičajeno, Taylorov polinom stupnja n označavamo s T_n, dok MacLaurinov polinom stupnja n označavamo s M_n.

Pogledajmo svaki pojedini slučaj na konkretnom primjeru.

***Primjer* 1.** Aproksimirajmo realnu funkciju $f(x) = \sqrt{e^x + 1}$ MacLaurinovim polinomom stupnja najviše 5. Potom odredimo apsolutnu i relativnu pogrješku aproksimacije vrijednosti $f\left(\frac{1}{2}\right)$ s vrijednosti dobivenoga polinoma u točki $x = \frac{1}{2}$, pa na istoj slici prikažimo grafove obiju funkcija na segmentu $[-1, 1]$.

U ovome je primjeru, dakle, riječ o prvom od ranije spomenutih šest slučajeva. Ako to još nismo učinili, „očistimo“ MATLAB-ov komandni prozor, pa u nova tri retka toga prozora utipkajmo:

```
syms f x;
f=sqrt(exp(x)+1);
g=taylor(f)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
g =
2^(1/2)+1/4*2^(1/2)*x+3/32*2^(1/2)*x^2+7/384*2^(1/2)*x^3+3/2048*2^(1/2)*x^4
+1/122880*2^(1/2)*x^5
```

I ovaj rezultat treba pažljivo interpretirati. Traženi polinom je jednak:

$$M_5(x)=\frac{1}{122880}\cdot\sqrt{2}\cdot x^5+\frac{3}{2048}\cdot\sqrt{2}\cdot x^4+\frac{7}{384}\cdot\sqrt{2}\cdot x^3+\frac{3}{32}\cdot\sqrt{2}\cdot x^2+\frac{1}{4}\cdot\sqrt{2}\cdot x+\sqrt{2}\,.$$

Vrijednosti apsolutne, odnosno relativne pogrješke aproksimacije zadane funkcije MacLaurinovim polinomom u točki $x = \frac{1}{2}$ izračunat ćemo koristeći funkciju `subs`. Označimo s *ap* apsolutnu, a s *rp* relativnu vrijednost pogrješke aproksimacije. U novi redak MATLAB-ova komandnoga prozora utipkajmo:

```
ap=abs(subs(f,1/2)-subs(g,1/2)), rp=abs(ap/subs(f,1/2))*100
```

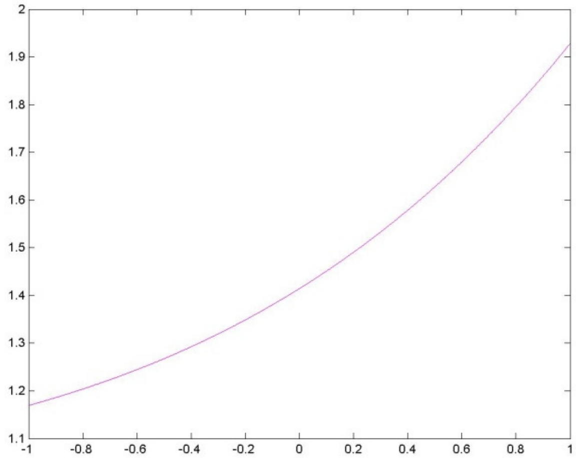
Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ap =
    1.038984905887830e-006
rp =
    6.383973986720478e-005
```

Dakle, apsolutna vrijednost pogrješke aproksimacije iznosi (približno) $1.03898 \cdot 10^{-6}$, dok je relativna vrijednost pogrješke aproksimacije (približno) $6.38397 \cdot 10^{-5}$ %. Stoga možemo zaključiti da se radi o vrlo dobroj aproksimaciji zadane funkcije MacLaurinovim polinomom. Provjerimo taj zaključak i grafički. U nova četiri retka MATLAB-ova komandnoga prozora utipkajmo:

```
x=-1:0.001:1;
y1=subs(f,x);
y2=subs(g,x);
plot(x,y1,x,y2)
```

Pritisnimo *Enter*, pa ćemo dobiti sljedeću sliku:

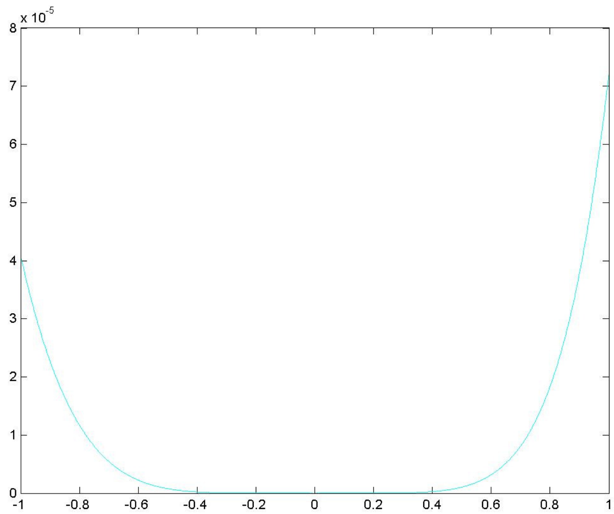


Slika 1.

Kvalitetu dobivene aproksimacije na cijelom segmentu [–1, 1] možemo provjeriti i tako da nacrtamo graf funkcije $h(x) = |f(x) - g(x)| = |f(x) - M_5(x)|$ na tom segmentu. U novi redak komandnoga prozora utipkajmo:

```
plot(x,abs(y1-y2))
```

Pritisnimo *Enter*, pa ćemo dobiti sljedeću sliku:



Slika 2.

Sa slike vidimo da je najveća vrijednost funkcije h na segmentu [–1, 1] strogo manja od $8 \cdot 10^{-5}$, pa možemo zaključiti da je riječ o vrlo dobroj aproksimaciji.

Primjer 2. Aproksimirajmo realnu funkciju $f(x)$ = arctg x MacLaurinovim polinomom stupnja najviše 8. Potom odredimo apsolutnu i relativnu pogrješku aproksimacije vrijednosti $f\left(\frac{1}{3}\right)$ s vrijednosti dobivenoga polinoma u točki $x = \frac{1}{3}$, pa na istoj slici prikažimo grafove obiju funkcija na segmentu [–1, 1].

Ulazni podatci za funkciju `taylor` su funkcija f i prirodan broj n = st(M) + 1 = 8 + 1 = 9. U nova dva retka MATLAB-ova komandnoga prozora utipkamo:

```
f=atan(x);
g=taylor(f,9)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
g =
x-1/3*x^3+1/5*x^5-1/7*x^7
```

Dakle, ne postoji MacLaurinov polinom stupnja 8 koji aproksimira zadanu funkciju, već je najbolja aproksimacija MacLaurinov polinom stupnja 7:

$$M_7(x) = -\frac{1}{7}\cdot x^7 + \frac{1}{5}\cdot x^5 - \frac{1}{3}\cdot x^3 + x\,.$$

Vrijednosti apsolutne, odnosno relativne pogrješke aproksimacije u točki $x = \frac{1}{3}$ računamo kao u prethodnom primjeru. U novi redak komandnoga prozora utipkamo:

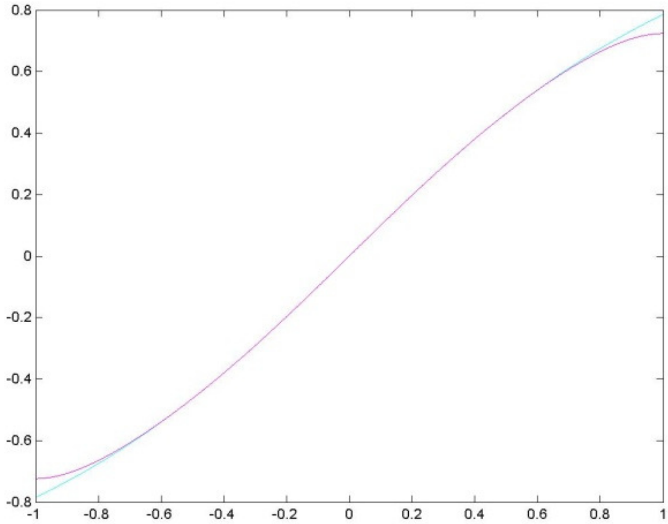
```
ap=abs(subs(f,1/3)-subs(g,1/3)), rp=abs(ap/subs(f,1/3))*100
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ap =
    5.175861140183091e-006
rp =
   0.00160865647921
```

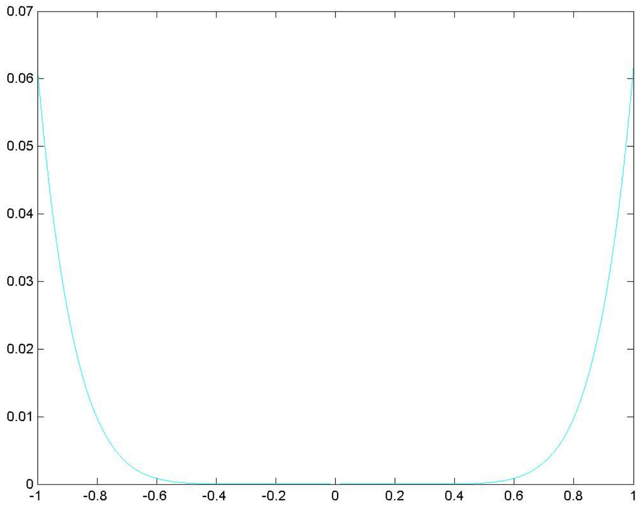
Dakle, apsolutna vrijednost pogrješke aproksimacije u točki $x = \frac{1}{3}$ iznosi približno 5.17586 · · 10^{-6}, dok relativna vrijednost pogrješke aproksimacije iznosi (približno) 0.00161%. Stoga zaključujemo da je riječ o vrlo dobroj aproksimaciji.

Provjerimo svoj zaključak i grafički. Potpuno analogno kao u prethodnom primjeru dobivamo sljedeću sliku:



Slika 3.

Kvalitetu dobivene aproksimacije na cijelom segmentu $[-1, 1]$ ponovno možemo provjeriti crtanjem grafa funkcije $h(x) = |f(x) - M_7(x)| = |f(x) - g(x)|$ na tom segmentu. Dobivamo:



Slika 4.

Sa slike vidimo da je najveća vrijednost funkcije h na segmentu $[-1, 1]$ približno jednaka 0.06 i da se postiže u krajevima segmenta (tj. za $x \in \{-1, 1\}$). Grubo možemo reći da smo dobili aproksimaciju *na cijelom segmentu* lošiju od one u Primjeru 1, ali odmah treba napomenuti da njezina kvaliteta bitno ovisi o ulaznim podatcima (funkciji, stupnju MacLaurinovoga polinoma, izboru segmenta itd.).

Primjer 3. Aproksimirajmo realnu funkciju $g(t) = \ln\left(\sqrt{t+1}+1\right)$ MacLaurinovim polinomom stupnja 7. Potom odredimo apsolutnu i relativnu pogrješku aproksimacije vrijednosti $f\left(-\frac{1}{2}\right)$ s vrijednosti dobivenoga polinoma u točki $t = -\frac{1}{2}$, pa na istoj slici prikažimo grafove obiju funkcija na segmentu [–1, 1].

U ovom primjeru moramo pripaziti i na oznaku nezavisne varijable: u ovom slučaju ta je varijabla označena slovom *t*. Vrijednost varijable *n* jednaka je *n* = 7 + 1 = 8. U nova tri retka komandnoga prozora utipkajmo:

```
syms g t;
g=log(sqrt(t+1)+1);
h=taylor(g,t,8)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
h =
log(2)+1/4*t-3/32*t^2+5/96*t^3-35/1024*t^4+63/2560*t^5-77/4096*t^6+429/28672*t^7
```

I ovdje treba biti oprezan s interpretacijom rezultata. Traženi je polinom jednak:

$$M_7(t) = \frac{429}{28672}\cdot t^7 - \frac{77}{4096}\cdot t^6 + \frac{63}{2560}\cdot t^5 - \frac{35}{1024}\cdot t^4 + \frac{5}{96}\cdot t^3 - \frac{3}{32}\cdot t^2 + \frac{1}{4}\cdot t + \ln 2\,.$$

Apsolutnu i relativnu pogrješku aproksimacije odredimo kao u prethodnim primjerima. Utipkavanje

```
ap=abs(subs(g,-1/2)-subs(h,-1/2)), rp=abs(ap/subs(g,-1/2))*100
```

u novi redak komandnoga prozora daje:

```
ap =
    8.336880340131803e-005
rp =
   0.01558878158369
```

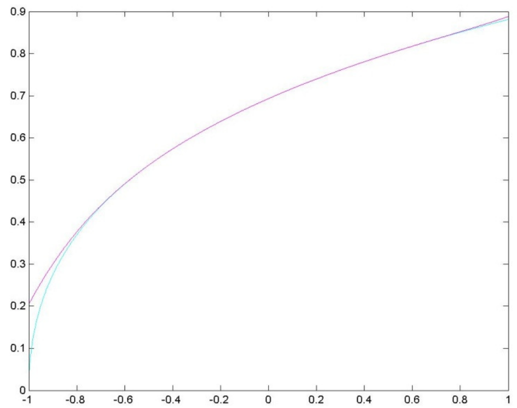
Dakle, apsolutna vrijednost pogrješke aproksimacije u točki $x = -\frac{1}{2}$ iznosi (približno) $8.33688 \cdot 10^{-5}$, dok je relativna vrijednost te pogrješke (približno) 0.01559%. Stoga zaključujemo da je riječ o vrlo dobroj aproksimaciji.

Provjerimo svoj zaključak i grafički. U nova četiri retka komandnoga prozora utipkajmo:

```
t=-1:0.001:1;
```

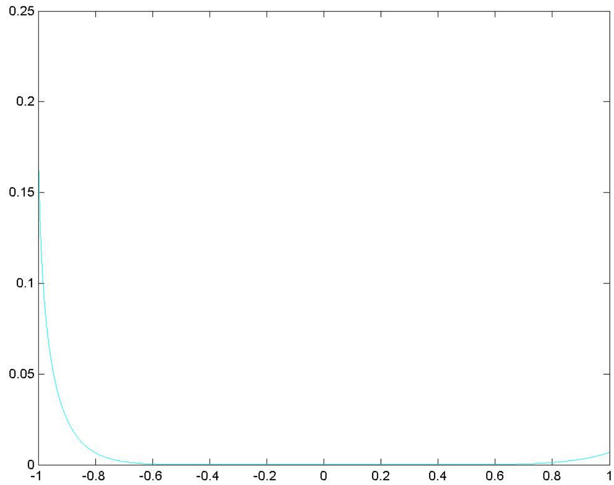
```
y1=subs(g,t);
y2=subs(h,t);
plot(t,y1,t,y2)
```

Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku:



Slika 5.

Analogno kao u prethodnim primjerima provjerimo kvalitetu dobivene aproksimacije na cijelom segmentu [–1, 1]. Dobivamo:



Slika 6.

Sa slike vidimo da je najveća vrijednost funkcije $h_1(t) = |g(t) - M_7(t)| = |g(t) - h(t)|$ na segmentu [–1, 1] približno jednaka 0.2 i da se postiže na lijevom kraju segmenta (tj. za $x =$ $= -1$). Da smo npr. odabrali segment [–0.6, 0.6], spomenuta najveća vrijednost bila bi približno $4.4 \cdot 10^{-4}$.

Primjer 4. Aproksimirajmo realnu funkciju $g(x)=\sqrt[3]{x+2}$ Taylorovim polinomom stupnja najviše 5 oko točke c = –1. Potom odredimo apsolutnu i relativnu pogrješku aproksimacije vrijednosti $f\left(-\frac{1}{3}\right)$ s vrijednosti dobivenoga polinoma u točki $x = -\frac{1}{3}$, pa na istoj slici prikažimo grafove obiju funkcija na segmentu [–2, 0].

U ovom su slučaju ulazni podatci za funkciju `taylor` realna funkcija g i točka c = –1. „Počistimo" komandni prozor, pa u nova tri njegova retka utipkajmo:

```
syms x
g=(x+2)^(1/3);
h=taylor(g,-1)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
h=
4/3+1/3*x-1/9*(x+1)^2+5/81*(x+1)^3-10/243*(x+1)^4+22/729*(x+1)^5
```

Stoga je traženi Taylorov polinom:

$$T_5(x)=\frac{22}{729}\cdot(x+1)^5-\frac{10}{243}\cdot(x+1)^4+\frac{5}{81}\cdot(x+1)^3-\frac{1}{9}\cdot(x+1)^2+\frac{1}{3}\cdot x+\frac{4}{3}.$$

Odmah primijetimo da je MATLAB pojednostavnio zapis posljednjih dvaju članova toga polinoma. Naime, umjesto „pravoga" izraza $\frac{1}{3}\cdot(x+1)+1$ MATLAB je ispisao reduciraniji izraz $\frac{1}{3}\cdot x+\frac{4}{3}$. Takvo se reduciranje „linearnih" (i samo „linearnih") članova Taylorova polinoma pojavljuje kod razvoja oko racionalnih točaka, pa to treba imati na umu prigodom zapisivanja polinoma.

Apsolutnu i relativnu pogrješku aproksimacije polazne funkcije u točki $x = -\frac{1}{3}$ dobivamo kao u prethodnim primjerima. Utipkavanje

```
ap=abs(subs(g,-1/3)-subs(h,-1/3)), rp=abs(ap/subs(g,-1/3))*100
```

u novi redak komandnoga prozora daje nešto slabiji rezultat nego u prethodnim primjerima:

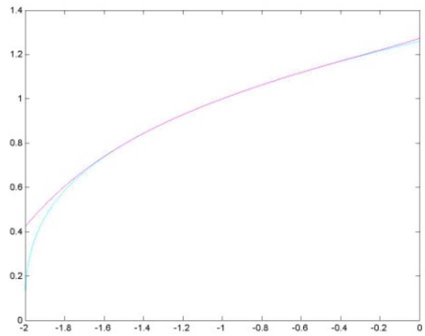
```
ap =
   0.00134355796692
rp =
   0.11332006770281
```

Dakle, apsolutna vrijednost pogrješke aproksimacije u navedenoj točki iznosi (približno) 0.00134, a relativna vrijednost te pogrješke 0.11332%. Stoga možemo zaključiti da je riječ o dobroj aproksimaciji (ali lošijoj nego npr. u prethodnom primjeru).

Provjerimo svoj zaključak i grafički. U nova četiri retka komandnoga prozora utipkajmo:

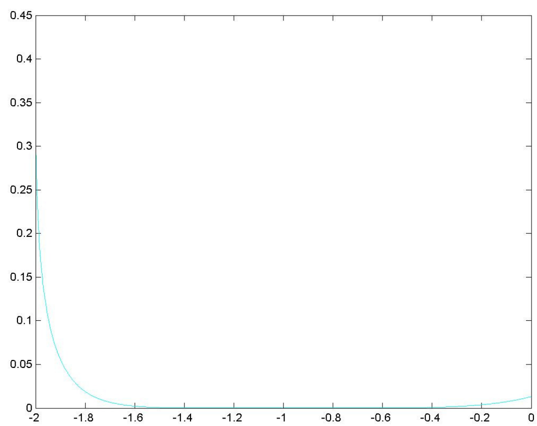
```
x=-2:0.001:0;
y1=subs(g,x);
y2=subs(h,x);
plot(x,y1,x,y2)
```

Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku:



Slika 7.

Provjerimo kvalitetu dobivene aproksimacije na cijelom segmentu [–2, 0]. Dobivamo:



Slika 8.

Sa slike vidimo da je najveća vrijednost funkcije $h_1(x) = |g(x) - T_5(x)| = |g(x) - h(x)|$ na segmentu [–2, 0] približno jednaka 0.43 i da se postiže na lijevom kraju segmenta. Da smo polazni segment „smanjili“ na [–1.6, –0.6], ta bi vrijednost iznosila približno $2.1 \cdot 10^{-3}$.

Primjer 5. Aproksimirajmo realnu funkciju $h(x) = \ln^2 x$ Taylorovim polinomom stupnja najviše 8 u okolini točke $c = 1$.

Ulazni podatci za funkciju `taylor` su funkcija h, prirodan broj $n = 8 + 1 = 9$ i točka $c = 1$. U nova tri retka MATLAB-ova komandnoga prozora utipkamo:

```
syms h;
h=(log(x))^ 2;
taylor(h,9,1)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
(x-1)^2-(x-1)^3+11/12*(x-1)^4-5/6*(x-1)^5+137/180*(x-1)^6-7/10*(x-1)^7+363/560*(x-1)^8
```

Stoga je traženi Taylorov polinom

$$T_8(x) = \frac{363}{560}\cdot(x-1)^8 - \frac{7}{10}\cdot(x-1)^7 + \frac{137}{180}\cdot(x-1)^6 - \frac{5}{6}\cdot(x-1)^5 + \frac{11}{12}\cdot(x-1)^4 - (x-1)^3 + (x-1)^2 .$$

Za vježbu prikažite grafički funkcije h, T_8 i $h_1 = h - T_8$ na segmentu $\left[\frac{1}{2}, \frac{3}{2}\right]$, pa provjerite da je najveća vrijednost funkcije h_1 na tom segmentu približno jednaka $2.3 \cdot 10^{-3}$.

Primjer 6. Aproksimirajmo realnu funkciju $f(u) = \sin(2 \cdot u)$ Taylorovim polinomom stupnja najviše 9 u okolini točke $c = \frac{\pi}{2}$.

U ovom su slučaju, dakle, ulazni podatci funkcije `taylor`: nezavisna varijabla u, vrijednost varijable $n = 9 + 1 = 10$ i točka $c = \frac{\pi}{2}$. Stoga u nova tri retka MATLAB-ova komandnoga prozora utipkamo:

```
syms u;
f=sin(2*u);
taylor(f,10,u,pi/2)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
-2*u+pi+4/3*(u-1/2*pi)^3-4/15*(u-1/2*pi)^5+8/315*(u-1/2*pi)^7-4/2835*(u-1/2*pi)^9
```

Preostaje zapisati dobiveni polinom u uobičajenom zapisu:

$$T_9(u) = -\frac{4}{2835}\cdot\left(u-\frac{\pi}{2}\right)^9 + \frac{8}{315}\cdot\left(u-\frac{\pi}{2}\right)^7 - \frac{4}{15}\cdot\left(u-\frac{\pi}{2}\right)^5 + \frac{4}{3}\cdot\left(u-\frac{\pi}{2}\right)^3 - 2\cdot u + \pi .$$

6.5. Razvoj periodične realne funkcije u Fourierov red

U *Matematici* 2 naučili smo kako periodičnu realnu funkciju (s temeljnim periodom $T > 0$ i definiranu na osnovnom segmentu $[a,\ a + T]$) aproksimirati Fourierovim (zapravo, trigonometrijskim) polinomom oblika:

$$F_n(x) = \sum_{k=0}^{n} \left[a_k \cdot \cos\left(\frac{2 \cdot k \cdot \pi}{T} \cdot x \right) + b_k \cdot \sin\left(\frac{2 \cdot k \cdot \pi}{T} \cdot x \right) \right]. \tag{1}$$

Najčešće je $T = 2 \cdot \pi$, pa relacija (1) tada prelazi u:

$$F_n(x) = \sum_{k=0}^{n} \left[a_k \cdot \cos(k \cdot x) + b_k \cdot \sin(k \cdot x) \right].^{20} \tag{2}$$

U općem su slučaju koeficijenti a_k i b_k dani sljedećim formulama:

$$\left. \begin{aligned} a_0 &= \frac{1}{T} \cdot \int_a^{a+T} f(x) \cdot dx, \\ a_k &= \frac{2}{T} \cdot \int_a^{a+T} f(x) \cdot \cos\left(\frac{2 \cdot k \cdot \pi}{T} \cdot x \right) \cdot dx, \text{ za svaki } k = 1, \dots, n; \\ b_k &= \frac{2}{T} \cdot \int_a^{a+T} f(x) \cdot \sin\left(\frac{2 \cdot k \cdot \pi}{T} \cdot x \right) \cdot dx, \text{ za svaki } k = 1, \dots, n. \end{aligned} \right\} \tag{3}$$

Za $T = 2 \cdot \pi$ te formule možemo zapisati u pojednostavljenom obliku:

$$\left. \begin{aligned} a_0 &= \frac{1}{2 \cdot \pi} \cdot \int_a^{a+2\cdot\pi} f(x) \cdot dx, \\ a_k &= \frac{1}{\pi} \cdot \int_a^{a+2\cdot\pi} f(x) \cdot \cos(k \cdot x) \cdot dx, \text{ za svaki } k = 1, \dots, n; \\ b_k &= \frac{1}{\pi} \cdot \int_a^{a+2\cdot\pi} f(x) \cdot \sin(k \cdot x) \cdot dx, \text{ za svaki } k = 1, \dots, n. \end{aligned} \right\} \tag{4}$$

Gornje je integrale često tehnički mukotrpno analitički računati, pa ćemo taj postupak implementirati u MATLAB-u. Pritom nećemo provjeravati tzv. Dirichletove uvjete (uz koje je moguć razvoj funkcije u Fourierov red), niti se baviti pitanjem periodičnosti, nego ćemo pretpostaviti da je periodična funkcija f definirana propisom na svojem osnovnom segmentu.

[20] U ovom se slučaju prirodan broj n naziva *stupanj Fourierova polinoma*.

***Primjer* 1.** Kreirajmo funkcijsku *m*-datoteku **fr.m** koja sadrži isključivo funkciju *fr* čije su ulazne vrijednosti periodična realna funkcija *f*, realni brojevi *a* i *b*, te prirodan broj *n*. Funkcija treba ispisati koeficijente a_0, a_1, …, a_n, b_1, …, b_n Fourierova polinoma koji aproksimira funkciju *f* definiranu na osnovnom segmentu $[a, b]$ s temeljnim periodom $T := b - a$. (Koeficijent b_0 ne treba ispisivati jer je taj koeficijent uvijek jednak 0.)

Odmah napomenimo da naziv funkcije ne može biti `fourier` jer MATLAB već ima implementiranu jednu „gotovu“ funkciju istoga imena. Nadalje, koeficijent a_0 moramo ispisati posebno jer se računa po posebnoj formuli, dok koeficijente uz „prave“ (netrivijalne) argumente funkcija sinus i kosinus možemo izračunati koristeći *for*-petlju.

Otvorimo novu *m*–datoteku, pa u nju utipkajmo:

```
function y=fr(f,a,b,n)
syms x;
a0=double(1/(b-a)*int(f,a,b));
for k=1:n
    c(k)=2/(b-a)*double(int(f*cos(2*k*pi/(b-a)*x),x,a,b));
    d(k)=2/(b-a)*double(int(f*sin(2*k*pi/(b-a)*x),x,a,b));
end;
'Slobodni clan je:'
a0
'Koeficijenti uz argumente funkcije sin su:'
d
'Koeficijenti uz argumente funkcije cos su:'
c
```

Detaljno objasnite svaki pojedini redak ove datoteke. Pohranimo tako stvorenu datoteku i vratimo se u MATLAB-ov komandni prozor. Preostaje „ispitati“ je na konkretnim primjerima.

***Primjer* 2.** $(2 \cdot \pi)$–periodična funkcija *f* definirana je propisom $f(x) = x^2$, za $x \in [-\pi, \pi]$. Aproksimirajmo zadanu funkciju Fourierovim polinomom stupnja 5.

Uočimo da je zadana funkcija parna, što znači da će „preživjeti“ samo koeficijenti uz argumente funkcije kosinus. Zbog $(2 \cdot \pi)$-periodičnosti argumenti funkcije kosinus bit će „prirodni višekratnici“ varijable *x*, tj. x, $2 \cdot x$, $3 \cdot x$, $4 \cdot x$ i $5 \cdot x$.

Ako to već niste napravili, obavezno „očistite“ komandni prozor. Zbog „kratkoće“ propisa funkcije *f*, nećemo deklarirati posebnu varijablu *f*, nego ćemo kao argument funkcije *fr* izravno uvrstiti simbolički objekt x^2. Dakle, u novim dvama retcima komandnoga prozora utipkamo:

```
syms x;
fr('x^2',-pi,pi,5)
```

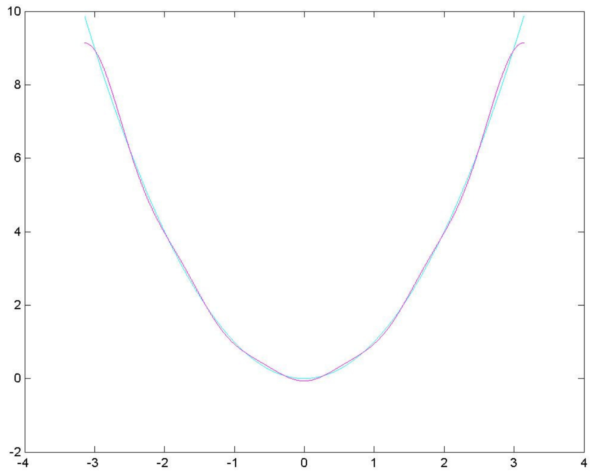
Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
Slobodni clan je:
a0 =
   3.28986813369645
ans =
Koeficijenti uz argumente funkcije sin su:
d =
     0     0     0     0     0
ans =
Koeficijenti uz argumente funkcije cos su:
c =
  -4.00000000000000       1.00000000000000     -0.44444444444444
0.25000000000000  -0.16000000000000
```

Ovakav smo rezultat mogli i očekivati jer smo u izrazima za određivanje Fourierovih koeficijenata koristili funkciju `double`. Kako god bilo, traženi Fourierov polinom stupnja 5 je:

$F_5(x) = 3.28986813369645 - 4 \cdot \cos x + \cos(2 \cdot x) - 0.44444444444444 \cdot \cos(3 \cdot x) + 0.25 \cdot$
$\cdot \cos(4 \cdot x) - 0.16 \cdot \cos(5 \cdot x)$.

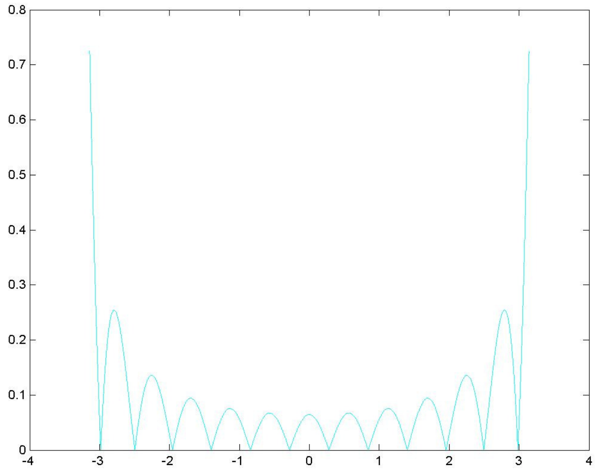
Grafove funkcija f i F_5 zgodno je prikazati na istoj slici. Dobivamo:



Slika 9.

(Svijetloplava krivulja je graf funkcije f, a roza krivulja je graf trigonometrijskoga polinoma F_5.) Kvalitetu aproksimacije već uobičajeno možemo provjeriti crtanjem grafa funkcije $h(x) =$

$= |f(x) - F_5(x)|$ na segmentu $[-\pi, \pi]$. Dobivamo:



Slika 10.

Primijetimo da najveća vrijednost funkcije h iznosi približno 0.7 i postiže se za $x \in \{-\pi, \pi\}$, tj. na krajevima segmenta.

<u>Primjer</u> **<u>3.</u>** $(2 \cdot \pi)$–periodična realna funkcija f definirana je propisom $f(x) = x^3$, za $x \in [-\pi, \pi\rangle$. Aproksimirajmo funkciju f Fourierovim polinomom stupnja 4.

Primijetimo da je zadana funkcija neparna, što znači da će u Fourierovu polinomu „preživjeti“ isključivo koeficijenti uz argumente funkcije sinus, kao i da će slobodni član biti jednak nuli. Za funkciju f vrijede Dirichletovi uvjeti, pa nas ne smeta što funkcija (prividno) nije definirana na segmentu, nego na poluzatvorenu intervalu. U ovakvim slučajevima rubna točka ne može utjecati na vrijednost određenoga integrala.

U novom retku komandnoga prozora utipkajmo:

```
fr('x^3',-pi,pi,4)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

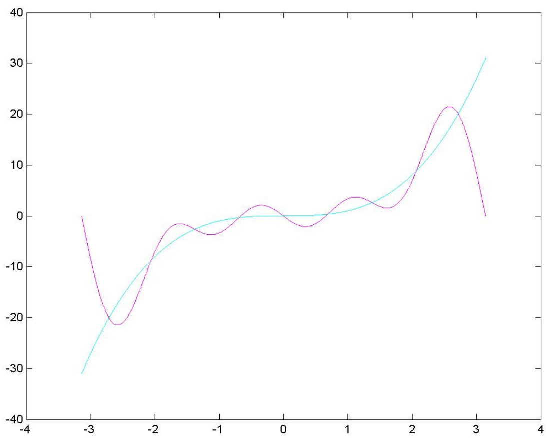
```
ans =
Slobodni clan je:
a0 =
     0
ans =
Koeficijenti uz argumente funkcije sin su:
d =
   7.73920880217872      -8.36960440108936          6.13529182294846
-4.74730220054468
```

```
ans =
Koeficijenti uz argumente funkcije cos su:
c =
     0     0     0     0
```

Prema tome, traženi Fourierov polinom je:

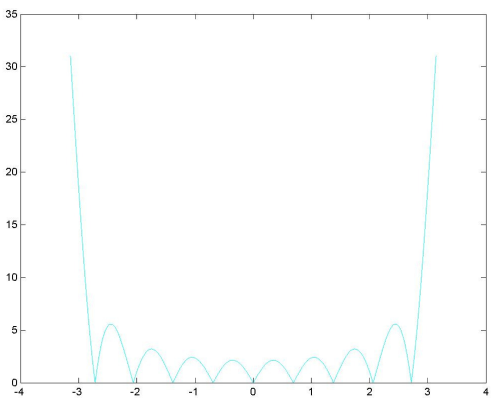
$F_4(x) = 7.73920880217872 \cdot \sin x - 8.36960440108936 \cdot \sin(2 \cdot x) + 6.13529182294846 \cdot \sin(3 \cdot x) - 4.74730220054468 \cdot \sin(4 \cdot x)$.

Prikažemo li grafički funkcije f i F_4 na istoj slici, dobit ćemo:



Slika 11.

Već na prvi pogled uočavamo da je ova aproksimacija bitno lošija od one u Primjeru 2. U to nas uvjerava i grafički prikaz funkcije $h = |f - F_4|$ na segmentu $[-\pi, \pi]$:

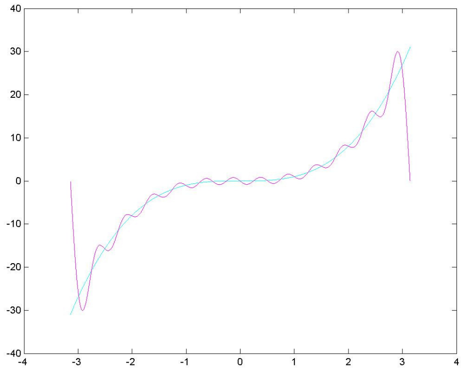


Slika 12.

Jedan od razloga za ovako lošu aproksimaciju jest i relativno mali stupanj Fourierova polinoma. Odlučimo li se za aproksimaciju Fourierovim polinomom stupnja 12, dobit ćemo:

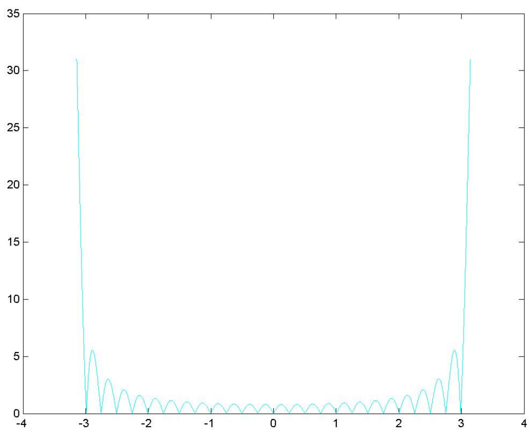
$F_{12}(x) = 7.73920880217872 \cdot \sin x - 8.36960440108936 \cdot \sin(2 \cdot x) + 6.13529182294846 \cdot \sin(3 \cdot x) - 4.74730220054468 \cdot \sin(4 \cdot x) + 3.85184176043574 \cdot \sin(5 \cdot x) - 3.2343125781409 \cdot \sin(6 \cdot x) + 2.78490154899929 \cdot \sin(7 \cdot x) - 2.44396360027234 \cdot \sin(8 \cdot x) + 2.17678451711451 \cdot \sin(9 \cdot x) - 1.96192088021787 \cdot \sin(10 \cdot x) + 1.78545774985997 \cdot \sin(11 \cdot x) - 1.63798962240378 \cdot \sin(12 \cdot x)$,

Uobičajenim postupkom nacrtajmo grafove funkcija f i F_{12} na istoj slici. Dobijemo:



Slika 13.

Graf funkcije $h_1 = |f - F_{12}|$ na segmentu $[-\pi, \pi]$ prikazan je na sljedećoj slici.



Slika 14.

„Problematičnost" aproksimacije funkcije Fourierovim polinomom u krajevima osnovnoga segmenta nismo uspjeli ukloniti, ali smo zato bitno poboljšali kvalitetu aproksimacije u apsolutnoj većini unutrašnjih točaka toga segmenta.

***Primjer* 4.** π – periodična realna funkcija definirana je propisom $f(x) = e^x$, za $x \in [0, \pi\rangle$. Aproksimirajmo tu funkciju Fourierovim polinomom stupnja 12.

Ovdje moramo pripaziti jer iz formula (3) slijedi da su argumenti funkcija sinus i kosinus „parni višekratnici“ varijable x, tj. $2 \cdot x$, $4 \cdot x$, $6 \cdot x$, $8 \cdot x$, $10 \cdot x$ i $12 \cdot x$. Zadana funkcija nije niti parna, niti neparna, pa će aproksimacija Fourierovim polinomom stupnja 12 sadržavati i kosinuse i sinuse naznačenih argumenata.

U novi redak MATLAB-ova komandnoga prozora utipkamo:

```
fr('exp(x)',0,pi,6)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

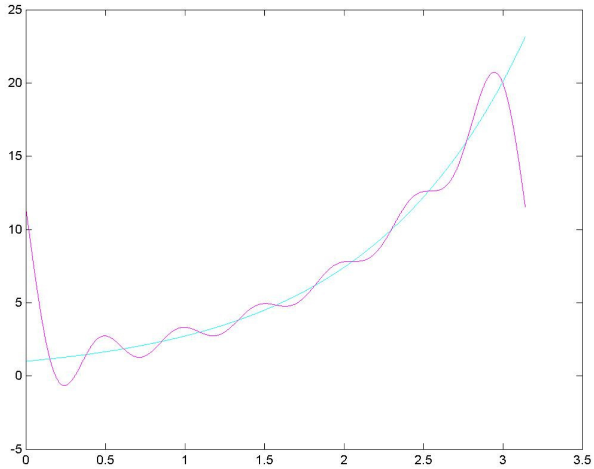
```
ans =
Slobodni clan je:
a0 =
   7.04760135197026
ans =
Koeficijenti uz argumente funkcije sin su:
d =
  -5.63808108157621   -3.31651828328012   -2.28570854658495   -
1.73479417894653  -1.39556462415253  -1.16649953411922
ans =
Koeficijenti uz argumente funkcije cos su:
c =
   2.81904054078810       0.82912957082003       0.38095142443082
0.21684927236832   0.13955646241525   0.09720829450993
```

Dakle, traženi Fourierov polinom je:

$F_{12}(x) = 7.04760135197026 + 2.8190405407881 \cdot \cos(2 \cdot x) - 5.63808108157621 \cdot \sin(2 \cdot x) + 0.82912957082003 \cdot \cos(4 \cdot x) - 3.31651828328012 \cdot \sin(4 \cdot x) + 0.38095142443082 \cdot \cos(6 \cdot x) - 2.28570854658495 \cdot \sin(6 \cdot x) + 0.21684927236832 \cdot \cos(8 \cdot x) - 1.73479417894653 \cdot \sin(8 \cdot x) + 0.13955646241525 \cdot \cos(10 \cdot x) - 1.39556462415253 \cdot \sin(10 \cdot x) + 0.09720829450993 \cdot \cos(12 \cdot x) - 1.16649953411922 \cdot \sin(12 \cdot x)$.

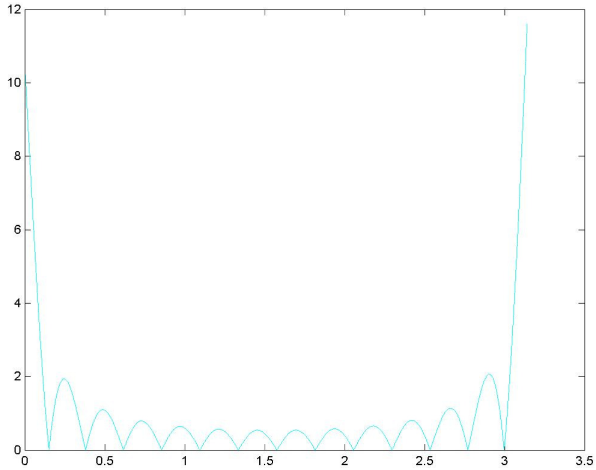
Iako je gornji zapis prilično „nezgrapan“ jer sadrži dvanaest decimalnih brojeva, točne vrijednosti Fourierovih koeficijenata relativno je teško izračunati (ali ne i nemoguće – pokušajte to učiniti analitički, pa provjerite svoje rezultate koristeći MATLAB, ali bez funkcije `double`), pa je gornji način posve primjeren za praktične svrhe (uz ne nužno toliku točnost).

Već uobičajeno na istoj slici prikažimo grafove funkcija f i F_{12}. Dobivamo:



Slika 15.

Kvalitetu aproksimacije provjerimo crtajući graf funkcije $h = |f - F_{12}|$. On je prikazan na Slici 16.



Slika 16.

I u ovom primjeru primjećujemo „problematičnost“ aproksimacije u rubnim točkama osnovnoga segmenta $[0, \pi]$, dok za vrijednosti koje zadana funkcija poprima u unutrašnjim točkama toga segmenta možemo govoriti o relativno dobroj aproksimaciji[21].

[21] Laici su možda očekivali da će u ovom slučaju slobodni član a_0 – koji je u Primjeru 3. bio jednak 0 - bitno popraviti kvalitetu aproksimacije vrijednosti funkcije u rubnim točkama. Aproksimacija Fourierovim polinomom zapravo se odnosi na aproksimaciju vrijednosti funkcije na *otvorenom* intervalu $\langle a, a + T\rangle$.

6.6. Zadatci za vježbu

1. Izračunajte zbrojeve (ako postoje) sljedećih redova i zapišite ih u znanstvenom obliku (pretpostavite da mantisa ima točno 6 znamenaka):

a) $\sum_{n=0}^{+\infty} \left(\frac{\sqrt[3]{e}}{\sqrt{\pi}} \right)^n$;

b) $\sum_{n=0}^{+\infty} \left(\frac{e}{\pi} \right)^{(\pi - e) \cdot n}$;

c) $\sum_{n=0}^{+\infty} \left(\frac{e-2}{\pi - 2} \right)^{(e+\pi) \cdot n}$;

d) $\sum_{n=0}^{+\infty} \left(\frac{\sin 2012 + \cos 2012}{\operatorname{sh} 2 + \operatorname{ch} 2} \right)^{(\operatorname{tg} 2012 + \operatorname{th} 2012) \cdot n}$;

e) $\sum_{n=0}^{+\infty} (-1)^n \cdot \left(\frac{\log 2 + \ln 3}{\log_2 3 + \log_3 2} \right)^{\ln^2 3 \cdot n}$;

f) $\sum_{n=0}^{+\infty} (-1)^n \cdot \left(\frac{\arcsin \frac{1}{3} + \operatorname{arsh} \frac{1}{3}}{\arccos \frac{1}{3} + \operatorname{arch} \frac{4}{3}} \right)^{\left(\operatorname{arctg} \frac{4}{3} + \operatorname{arth} \frac{1}{3} \right) \cdot n}$.

2. Odredite apsolutnu i relativnu vrijednost pogrješke koju činimo ako zamijenimo zbroj reda $\sum_{n=0}^{+\infty} \frac{2012^n}{n!}$ zbrojem prvih 100 članova reda. Zapišite rezultate u znanstvenom obliku.

3. Odredite apsolutnu i relativnu vrijednost pogrješke koju činimo ako zamijenimo zbroj reda $\sum_{n=1}^{+\infty} \frac{1}{n^4}$ zbrojem prvih 100 članova toga reda. Zapišite rezultate u znanstvenom obliku.

4. Odredite najmanji prirodan broj n tako da razlika zbroja reda $\sum_{k=1}^{+\infty} \frac{1}{k \cdot 3^k}$ i zbroja prvih n članova toga reda bude strogo manja od 10^{-5}.

5. Odredite najmanji prirodan broj n tako da razlika zbroja reda $\sum_{k=0}^{+\infty} \frac{1}{k!}$ i zbroja prvih n članova toga reda bude strogo manja od 10^{-5}.

6. Odredite najmanji prirodan broj n tako da zbroj prvih n članova harmonijskoga reda bude strogo veći od 12.

7. Primjenom svih triju kriterija obrađenih u ovom poglavlju ispitajte konvergenciju sljedećih redova:

a) $\frac{4}{10}+\frac{9}{20}+\frac{16}{30}+\frac{25}{40}+\frac{36}{50}+\ldots$; (*Naputak*: Najprije odredite opći član reda.)

b) $\frac{4}{2}+\frac{8}{6}+\frac{12}{12}+\frac{16}{20}+\frac{20}{30}+\ldots$; (*Naputak*: Najprije odredite opći član reda.)

c) $\sum\frac{\sqrt{5^n}}{n^2+2^n}$;

d) $\sum\frac{1}{2012}\cdot\left(\frac{1}{2}\right)^n$;

e) $\sum\frac{2012}{\sqrt[3]{n^2+1}}$;

f) $\sum\frac{1}{(3\cdot n-1)\cdot(3\cdot n+2)}$;

g) $\sum\frac{2012\cdot n-2011}{\sqrt[2013]{3^n}}$;

h) $\sum\left(\frac{n^2+1}{9\cdot n^2-1}\right)^{n+2}$;

i) $\sum\frac{(2\cdot n-1)^3}{(3\cdot n+1)\cdot(9\cdot n^2-3\cdot n+1)}$;

j) $\sum\frac{1}{(n!)^2}$.

8. Izračunajte zbroj reda $\sum_{n=1}^{+\infty}(-1)^n\frac{x^n}{n}$ i odredite sve $x\in\mathbf{R}$ za koje je taj zbroj jednak ln 2.

9. Izračunajte zbroj reda $\sum_{n=0}^{+\infty}\frac{(-1)^n}{x^n\cdot n!}$ i odredite sve $x\in\mathbf{R}$ za koje je taj zbroj jednak e.

10. Izračunajte zbroj reda $\sum_{n=0}^{+\infty}n^2\cdot x^n$ i odredite sve $x\in\mathbf{R}$ za koje je izračunani zbroj jednak 1.

11. Aproksimirajte sljedeće funkcije MacLaurinovim polinomom stupnja najviše n, izračunajte apsolutnu i relativnu pogrješku aproksimacije u točki c = 0.25, prikažite grafički funkciju f i dobiveni MacLaurinov polinom na segmentu [–0.5, 0.5] u istom pravokutnom koordinatnom sustavu u ravnini, te nacrtajte graf pripadne funkcije greške aproksimacije:

a) $f(x) = 3 \cdot \sqrt{x+1}$, $n = 5$;
b) $f(x) = \left(-\frac{1}{8}\right) \cdot \ln(x+1)$, $n = 5$;
c) $f(x) = \sin\left(\frac{\pi}{2} \cdot x\right)$, $n = 4$;
d) $f(x) = \operatorname{tg}\left(\frac{\pi}{4} \cdot x\right)$, $n = 6$;
e) $f(t) = \arcsin t$, $n = 4$;
f) $f(u) = \operatorname{arcctg} u$, $n = 4$.

12. Aproksimirajte sljedeće realne funkcije Taylorovim polinomom stupnja n u okolini točke c, odredite apsolutnu i relativnu vrijednost pogrješke aproksimacije u točki $c + 0.1$, prikažite grafički funkciju f i dobiveni Taylorov polinom na segmentu $[c - 1, c + 1]$ u istom pravokutnom koordinatnom sustavu u ravnini, pa nacrtajte graf pripadne funkcije greške aproksimacije:

a) $f(x) = \operatorname{ctg} x$, $n = 5$, $c = \frac{\pi}{2}$;
b) $f(x) = \operatorname{cth} x$, $n = 5$, $c = \ln 2$;
c) $f(x) = \log x$, $n = 4$, $c = 10$;
d) $f(x) = 2^x$, $n = 6$, $c = -1$;
e) $f(t) = \sin t + \cos^3 t$, $n = 4$, $c = \pi$;
f) $f(u) = \operatorname{th}^2 x$, $n = 6$, $c = -\ln 2$.

13. Fourierovim polinomom stupnja 8 aproksimirajte sljedeće $(2 \cdot \pi)$ – periodične funkcije, odredite apsolutnu i relativnu pogrješku aproksimacije u razrednoj sredini intervala, pa zadanu funkciju i dobiveni Fourierov polinom prikažite grafički na zadanom intervalu u istom pravokutnom koordinatnom sustavu u ravnini:

a) $f : [-\pi, \pi\rangle, f(x) = -x$;
b) $f : [-\pi, \pi], f(x) = \sin(x^2)$;
c) $f : [-\pi, \pi\rangle, f(x) = \operatorname{arctg} x$;
d) $f : [-\pi, \pi], f(x) = \operatorname{ch} x$;
e) $f : [0, 2 \cdot \pi\rangle, f(x) = x^2$;
f) $f : [0, 2 \cdot \pi], f(x) = \sin \frac{x}{2}$.

U svakom pojedinom slučaju nacrtajte i graf pripadne funkcije pogrješke aproksimacije.

14. Kreirajte funkcijsku m-datoteku **zbprvm.m** koja sadrži jedino funkciju *zbrprvm* čiji su argumenti opći član reda (u varijabli n) i prirodan broj m. Funkcija treba ispisati zbroj prvih m članova zadanoga reda.

15. Kreirajte funkcijsku *m*–datoteku **razlika.m** koja sadrži jedino funkciju *razlika* čiji su argumenti opći član reda (u varijabli n) i prirodan broj m. Funkcija treba odrediti koji je od prvih m članova reda najveći, a koji najmanji, te ispisati apsolutnu vrijednost njihove razlike.

16. Kreirajte funkcijsku *m*–datoteku **veciodk.m** koja sadrži jedino funkciju *veciodk* čiji su argumenti opći član reda (u varijabli n), te prirodni brojevi k i m. Funkcija treba generirati (ali ne i ispisati) prvih m članova reda, te ispisati ukupan broj generiranih članova koji su strogo veći od k.

17. Kreirajte funkcijsku *m*–datoteku **najveciodk.m** koja sadrži jedino funkciju *najveciodk* čiji su argumenti opći član reda (u varijabli n), te prirodni brojevi k i m. Funkcija treba generirati (ali ne i ispisati) prvih m članova reda, te ispisati najveći od svih generiranih članova koji je strogo manji od k. Ako takav član ne postoji, funkcija treba ispisati `NaN`.

18. Kreirajte funkcijsku *m*–datoteku **zbrmac.m** koja sadrži jedino funkciju *zbrmac* čiji su argumenti realna funkcija f (u varijabli x) i prirodan broj n. Funkcija treba ispisati zbroj svih koeficijenata MacLaurinova polinoma stupnja ne većega od n koji najbolje aproksimira funkciju f. Bez provjere smijete pretpostaviti da je funkcija f beskonačno puta derivabilna na nekoj okolini točke $c = 0$.

19. Kreirajte funkcijsku *m*–datoteku **zbrtay.m** koja sadrži jedino funkciju *zbrtay* čiji su argumenti realna funkcija f (u varijabli x), realan broj a i prirodan broj n. Funkcija treba ispisati zbroj svih koeficijenata Taylorova polinoma stupnja n koji najbolje aproksimira funkciju f u okolini točke a. Bez provjere smijete pretpostaviti da je funkcija f beskonačno puta derivabilna na nekoj okolini točke a.

20. Kreirajte funkcijsku *m*–datoteku **frparna.m** koja sadrži jedino funkciju *frparna* čiji su argumenti parna funkcija f definirana na segmentu $[0, a]$, strogo pozitivan realan broj a i prirodan broj n. Temeljni period funkcije f jednak je $2 \cdot a$. Funkcija treba ispisati koeficijente $a_0, a_1, \ldots, a_n$ odgovarajućega Fourierova polinoma koji aproksimira funkciju f na njezinu osnovnu segmentu. Bez provjere smijete pretpostaviti da funkcija f zadovoljava Dirichletove uvjete. (*Naputak*: Koristite preinačene formule za izračun Fourierovih koeficijenata parne funkcije.)

21. Kreirajte funkcijsku *m*–datoteku **frneparna.m** koja sadrži jedino funkciju *frneparna* čiji su argumenti neparna funkcija f definirana na intervalu $\langle 0, a \rangle$, strogo pozitivan realan broj a i prirodan broj n. Temeljni period funkcije f jednak je $2 \cdot a$. Funkcija treba ispisati koeficijente $b_1, b_2, \ldots, b_n$ Fourierova polinoma koji aproksimira funkciju f na njezinu osnovnu segmentu. Bez provjere smijete pretpostaviti da funkcija f zadovoljava Dirichletove uvjete. (*Naputak*: Koristite preinačene formule za izračun Fourierovih koeficijenata neparne funkcije.)

§7. RJEŠAVANJE OBIČNIH DIFERENCIJALNIH JEDNADŽBI

U ovom ćemo poglavlju opisati kako se pomoću MATLAB-a mogu rješavati obične diferencijalne jednadžbe i različite Cauchyjeve zadaće. Pritom ćemo dodatno opisati određivanje Laplaceovih transformata i njihovih inverza.

7.1. Rješavanje običnih diferencijalnih jednadžbi

Kao i prigodom deriviranja, odnosno integriranja, tako i prigodom rješavanja običnih diferencijalnih jednadžbi u MATLAB-u sa samim jednadžbama postupamo kao sa simboličkim objektima. Dakle, treba definirati nezavisnu varijablu pomoću funkcije `syms`, pa potom zadati običnu diferencijalnu jednadžbu koju želimo riješiti. Pritom koristimo funkciju `dsolve` koja rješava obične diferencijalne jednadžbe. Za pojednostavljivanje dobivenoga izraza dodatno možemo koristiti funkciju `simplify`.

Pogledajmo sve izrečeno na primjerima.

***Primjer* 1.** Riješimo jednadžbu $(1+x^2)\cdot y'-2\cdot\sqrt{y}=0$.

U nova dva retka komandnoga prozora utipkajmo

```
syms x y
dsolve('(1+x^2)*Dy-2*sqrt(y)=0','x')
```

Komentirajmo sintaksu funkcije `dsolve`. Kako vidimo, derivacija nepoznate funkcije *y* označava se s `Dy`. Nakon što napišemo običnu diferencijalnu jednadžbu koju treba riješiti, kao posljednji argument moramo upisati naziv nezavisne varijable. Dogovorno se kao oznaka za nezavisnu varijablu uzima slovo *t*.

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
atan(x)^2+2*atan(x)*C1+C1^2
```

Dobro nam je poznato da opće rješenje obične diferencijalne jednadžbe 1. reda sadrži točno jednu nepoznatu realnu konstantu. U ovome je slučaju ona označena s `C1`. Stoga je traženo opće rješenje polazne jednadžbe

$$y = \mathrm{arctg}^2 x + 2\cdot C\cdot \mathrm{arctg}\, x + C^2,\ C\in \mathbf{R}.$$

ili, ekvivalentno,

$$y = (\mathrm{arctg}\, x + C)^2,\ C\in \mathbf{R}.$$

Napomena 1.: Izostavimo li 'x' u sintaksi funkcije `dsolve`, MATLAB bi ispisao pogrešno rješenje:

```
ans =
(t+C1+C1*x^2)^2/(1+x^2)^2
```

Naime, kako smo istaknuli, ne navedemo li naziv nezavisne varijable, MATLAB će dogovorno pretpostaviti da je nezavisna varijabla t, a nepoznata funkcija $y = y(t)$. Zbog toga je u gornjem slučaju MATLAB varijablu x navedenu u izrazu – argumentu funkcije `dsolve` tretirao kao realnu konstantu, a ne kao nezavisnu varijablu i argument nepoznate funkcije y. Stoga pripazite da pri pozivu funkcije `dsolve` ne zaboravite naziv nezavisne varijable!

Primjer 2. Riješimo jednadžbu: $e^{x^2} \cdot y' + 2 \cdot x \cdot e^{x^2} \cdot y = x \cdot \cos x$.

U novi redak komandnoga prozora utipkamo:

```
dsolve('exp(x^2)*Dy+2*x*exp(x^2)*y=x*cos(x)','x')
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
exp(-x^2)*cos(x)+exp(-x^2)*x*sin(x)+exp(-x^2)*C1
```

Stoga je opće rješenje polazne jednadžbe

$$y = e^{-x^2} \cdot (\cos x + x \cdot \sin x + C),\ C \in \mathbf{R}.$$

Primjer 3. Riješimo jednadžbu: $y'' - y' - 30 \cdot y = 0$. (Pretpostavimo da nezavisnu varijablu uobičajeno označavamo s x.)

U ovom primjeru rješavamo običnu diferencijalnu jednadžbu drugoga reda. Prigodom poziva funkcije `dsolve` drugu derivaciju nepoznate funkcije označavamo s `D2y`. (Općenito, derivaciju n-toga reda nepoznate funkcije označavamo s D*n*.) Dakle, u novi redak komandnoga prozora utipkamo:

```
dsolve('D2y-Dy-30*y=0','x')
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
C1*exp(-5*x)+C2*exp(6*x)
```

Kako znamo, opće rješenje bilo koje homogene obične diferencijalne jednadžbe 2. reda sadrži točno dvije nepoznate realne konstante: C_1 i C_2. Stoga je opće rješenje polazne jednadžbe:

$$y = C_1 \cdot e^{-5 \cdot x} + C_2 \cdot e^{6 \cdot x},\ C_1, C_2 \in \mathbf{R}.$$

Primjer 4. Riješimo jednadžbu: $y''-y'=e^{-x}\cdot\left[(5\cdot x^2-10\cdot x+2)\cdot\sin x-x\cdot(5\cdot x-4)\cdot\cos x\right]$.

U točki 7.3. vidjet ćemo kako nehomogenu običnu diferencijalnu jednadžbu 2. reda možemo riješiti pomoću Laplaceovih transformata i njihovih inverza. Sad u novi redak komandnoga prozora utipkamo:

```
dsolve('D2y-Dy=exp(-x)*((5*x^2-10*x+2)*sin(x)-x*(5*x-4)*cos(x))','x')
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
2*exp(-x)*x*cos(x)+exp(-x)*cos(x)+2*exp(-x)*sin(x)*x^2-exp(-
x)*sin(x)+exp(-x)*x^2*cos(x)+C1+C2*exp(x)
```

Stoga je opće rješenje polazne jednadžbe zapisano u uobičajenom obliku:

$$y = C_2\cdot e^x + C_1 + e^{-x}\cdot[(x^2+2\cdot x+1)\cdot\cos x+(2\cdot x^2-1)\cdot\sin x],\ C_1, C_2\in \mathbf{R}.$$

Primjer 5. Riješimo jednadžbu: $y^{(IV)}-y''=e^{-x}$.

Ova jednadžba 4. reda analitički se rješava uvođenjem zamjene $u := y''$. (Pokušajte je riješiti i na taj način.) U novi redak MATLAB-a utipkamo:

```
dsolve('D4y-D2y=exp(-x)','x')
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
-1/2*x*exp(-x)-5/4*exp(-x)+C1+C2*x+C3*exp(x)+C4*exp(-x)
```

Stoga je opće rješenje polazne jednadžbe zapisano u uobičajenom obliku:

$$y=C_1+C_2\cdot x+C_3\cdot e^x+C_4\cdot e^{-x}-\frac{1}{4}\cdot(2\cdot x+5)\cdot e^{-x},\ C_1, C_2, C_3, C_4\in \mathbf{R}.$$

7.2. Rješavanje Cauchyjevih zadaća

Pomoću funkcije `dsolve` moguće je i rješavanje različitih Cauchyjevih zadaća. Poziv funkcije u tom slučaju istovjetan je kao u prethodnoj točki, osim što se kao argumenti moraju zadati i svi početni uvjeti. Pogledajmo navedeno na sljedećim primjerima.

Primjer 1. Riješimo Cauchyjevu zadaću: $\begin{cases} x^2\cdot y^2\cdot y'+x\cdot y^3=1, \\ y(1)=2. \end{cases}$

Ako to već ranije nismo učinili, „počistimo“ svoj komandni prozor. Time, dakako, MATLAB neće zaboraviti da smo deklarirali simboličke objekte x i y. U novi redak komandnoga prozora utipkajmo:

```
dsolve('x^2*y^2*Dy+x*y^3=1','y(1)=2','x')
```

Uočimo da se, kao i u postavci Cauchyjeve zadaće, pri pozivu funkcije `dsolve` najprije <u>zasebno</u> piše obična diferencijalna jednadžba, potom <u>zasebno</u> početni uvjet i naposljetku oznaka nezavisne varijable (ako ta oznaka nije t). Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
1/2*(12*x^2+52)^(1/3)/x
```

Rješenje svake Cauchyjeve zadaće (ako postoji) je jedinstveno, pa u ovom slučaju nemamo niti jednu nepoznatu realnu konstantu. Dakle, rješenje zadatka u uobičajenom zapisu je funkcija

$$y = \frac{\sqrt[3]{12 \cdot x^2 + 52}}{2 \cdot x}.$$

<u>Primjer 2.</u> Riješimo Cauchyjevu zadaću:

$$\begin{cases} y'' - 2 \cdot y' - 24 \cdot y + 4 \cdot \cos x + 27 \cdot \sin x + 25 \cdot x \cdot \cos x = 2 \cdot x \cdot \sin x; \\ y(0) = -1; \\ y'(0) = 26. \end{cases}$$

U novi redak komandnoga prozora utipkajmo:

```
dsolve('D2y-2*Dy-24*y+4*cos(x)+27*sin(x)+25*x*cos(x)=2*x*sin(x)','y(0)=-1','Dy(0)=26','x')
```

Uočimo da smo drugi početni uvjet zadali u obliku `Dy(0)=26`. Općenito, uvjet $y^{(n)}(x_0) = y_n$ zadaje se u obliku `D`n`y`(x_0)=yn. Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
sin(x)+x*cos(x)-3*exp(-4*x)+2*exp(6*x)
```

Dakle, rješenje zadatka (zapisano u uobičajenom obliku) je realna funkcija

$$y = 2 \cdot e^{6 \cdot x} - 3 \cdot e^{-4 \cdot x} + \sin x + x \cdot \cos x.$$

<u>Primjer 3.</u> Odredimo funkciju y iz jednakosti: $\begin{cases} (y')^2 - y^2 = 1, \\ y(0) = 0. \end{cases}$

Primijetimo da obična diferencijalna jednadžba koju treba riješiti ne pripada u „tipične“ obične diferencijalne jednadžbe kakve smo rješavali u *Matematici* 2. To je donekle i razlog zbog kojega se ovaj problem ne ubraja u klasu Cauchyjevih zadaća. No, MATLAB s tim nema problema. U novi redak komandnoga prozora utipkamo:

```
dsolve('(Dy)^2-y^2=1','y(0)=0','x')
```

Pritisnemo *Enter*, pa će MATLAB ispisati ono što smo mogli i naslutiti:

```
ans =
[  sinh(x)]
[ -sinh(x)]
```

Dakle, rješenje zadatka su dvije realne funkcije: $y_1(x) = \text{sh } x$ i $y_2(x) = -\text{sh } x$. Napomenimo da bi zadatak imao jedinstveno rješenje da smo kao dodatni početni uvjet zadali npr. $y(\ln 2) = \frac{3}{4}$. Provjerite to!

Primjer 4. Riješite Cauchyjev problem: $\begin{cases} u' = u - \sin t, \\ u\left(\dfrac{\pi}{2}\right) = 0. \end{cases}$

Ovaj je primjer ponešto drugačiji od ostalih jer je nepoznata funkcija označena slovom *u*, a njezina nezavisna varijabla slovom *t*. Stoga u novi redak komandnoga prozora najprije moramo zapisati deklaraciju novih simobličkih varijabli:

```
syms u t
```

Dalje postupamo na uobičajen način. U novi redak komandnoga prozora utipkamo:

```
dsolve('Du=u-sin(t)','u(pi/2)=0')
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
1/2*cos(t)+1/2*sin(t)-1/2*exp(t)/(cosh(1/2*pi)+sinh(1/2*pi))
```

Budući da za svaki $t \in \mathbf{R}$ vrijedi jednakost

$$\text{ch } t + \text{sh } t = e^t,$$

dobiveno rješenje možemo zapisati u jednostavnijem obliku:

$$u = \frac{1}{2} \cdot \left(\cos t + \sin t - e^{t - \frac{\pi}{2}} \right).$$

Primjer 5. Odredite jednadžbu krivulje koja prolazi točkom $T = (1, 2)$ i ima svojstvo da bilo koja njezina tangenta na osi ordinata odsijeca odsječak jednak kvadratu apscise dirališta iste tangente.

Ovaj zadatak svodimo na rješavanje Cauchyjeva problema. Neka je $y = y(x)$ tražena jednadžba krivulje. Iz podatka da krivulja prolazi točkom T slijedi $y(1) = 2$. Nadalje, jednadžba tangente u proizvoljnoj točki $A = (x_A, y_A)$ krivulje glasi:

$$y = y'(x_A) \cdot (x - x_A) + y_A,$$

Njezin odsječak na osi ordinata je:

$$l = y_A - y'(x_A) \cdot x_A.$$

Prma uvjetima zadatka slijedi da mora vrijediti jednakost:

$$y_A - y'(x_A) \cdot x_A = x_A^2,$$

i to za svaku točku A, pa dobivamo običnu diferencijalnu jednadžbu

$$y - y' \cdot x = x^2,$$

odnosno Cauchyjev problem

$$\begin{cases} y - y' \cdot x = x^2, \\ y(1) = 2. \end{cases}$$

U novi redak komandnoga prozora utipkamo:

```
dsolve('y-Dy*x=x^2','y(1)=2','x')
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
-x^2+3*x
```

Stoga je rješenje zadatka parabola $y = -x^2 + 3 \cdot x$.

7.3. Određivanje Laplaceovih transformata i njihovih inverza

U *Matematici* 2, a i u kasnijim kolegijima, zahtijeva se rješavanje Cauchyjevih problema u kojima se pojavljuje obična diferencijalna jednadžba 2. reda pomoću Laplaceovih transformata i njihovih inverza. Ovdje ćemo ukratko opisati kako pomoću MATLAB-a naći Laplaceov transformat neke funkcije ili inverz nekoga Laplaceova transformata[22].

[22] Definiciju Laplaceova transformata i njegova inverza vidjeti npr. u [8], poglavlje 9., str. 325.

Laplaceov transformat zadane realne funkcije određujemo pomoću MATLAB-ove funkcije `laplace`. Pogledajmo dva primjera.

***Primjer* 1.** Odredimo Laplaceov transformat funkcije $f(x) = x^2 \cdot e^{1-x}$.

U nova tri retka MATLAB-ova komandnoga prozora utipkajmo redom:

```
syms f F x s t
f=x^2*exp(1-x);
laplace(f)
```

Pritisnemo *Enter*, pa će MATLAB ispisati traženi Laplaceov transformat:

```
ans =
2*exp(1)/(s+1)^3
```

Zapišimo dobiveno rješenje u uobičajenom zapisu:

$$F(s) = \frac{2 \cdot e}{(s+1)^3}.$$

Primijetimo da se kao oznaka varijable u Laplaceovu transformatu dogovorno pojavljuje slovo *s*. Želimo li da varijabla u zapisu Laplaceova transformata bude označena slovom *t*, posljednji utipkani redak trebamo preinačiti u:

```
laplace(f,t)
```

***Primjer* 2.** Odredimo Laplaceov transformat funkcije $f(x) = x \cdot \sin^2 x$, pri čemu varijabla u rješenju treba biti *t*.

Postupamo analogno kao u prethodnom primjeru:

```
f=x*sin(x)^2;
laplace(f,t)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
2/t^2/(t^2+4)+4/(t^2+4)^2
```

Želimo li nešto ljepši i dotjeraniji zapis dobivenoga transformata, u novi redak komandnoga prozora utipkajmo:

```
pretty(simplify(ans))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
      2
   3 t  + 4
2 ------------
   2  2     2
  t  (t  + 4)
```

Dakle, traženi Laplaceov transformat je $F(t)=2\cdot\dfrac{3\cdot t^2+4}{t^2\cdot(t^2+4)^2}$.

Inverz Laplaceova transformata određujemo koristeći funkciju `ilaplace`. Dogovorno se za oznaku varijable Laplaceova transformata uzima slovo *s*, ali je moguće uzeti i druge oznake. Pogledajmo sljedeća dva primjera.

Primjer 3. Odredimo inverz Laplaceova transformata $F(s)=\dfrac{s^2}{s^3+1}$. Varijabla u inverzu treba biti *x*.

U nova dva retka komandnoga prozora utipkajmo:

```
F=s^2/(s^3+1);
ilaplace(F,x)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
1/3*exp(-x)+2/3*exp(1/2*x)*cos(1/2*3^(1/2)*x).
```

Ovaj izraz nije teško „dešifrirati“. Traženi inverz je $f(x)=\dfrac{1}{3}\cdot e^{-x}+\dfrac{2}{3}\cdot e^{\frac{1}{2}\cdot x}\cdot\cos\left(\dfrac{\sqrt{3}}{2}\cdot x\right)$.

Napomenimo da izostavljanjem varijable *x* u izrazu `ilaplace(F,x)` MATLAB dogovorno pretpostavlja da je nezavisna varijabla u traženu inverzu *t*.

Primjer 4. Odredimo inverz Laplaceova transformata $F(s)=\dfrac{s^3+s^2+1}{\left[s\cdot(s-1)\right]^2}$. Varijabla u inverzu treba biti *x*.

U nova dva retka komandnoga prozora utipkajmo redom:

```
F=(s^3+s^2+1)/(s*(s-1))^2;
ilaplace(F,x)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
x+2+3*x*exp(x)-exp(x)
```

Dakle, traženi inverz je $f(x) = x + 2 + (3 \cdot x - 1) \cdot e^x$.

7.4. Zadatci za vježbu

1. Riješite sljedeće obične diferencijalne jednadžbe:

a) $y'' + y' + y = \cos x$;
b) $y''' + y'' + y' + y + x = 0$;
c) $(\sqrt{x} - \sqrt{x \cdot y}) \cdot y' - \sqrt{y} = 0$;
d) $y' = y \cdot \operatorname{ch} x$;
e) $\cos(2 \cdot x) \cdot y' + \sin(2 \cdot x) \cdot y = y$;
f) $x^2 \cdot y' + x \cdot y = x + 1$;
g) $y' + \dfrac{y}{x} = \sqrt{x \cdot y}$;
h) $y'' + 4 \cdot y' + 4 \cdot y = 3 \cdot (3 \cdot x + 2) \cdot e^x$;
i) $y'' - 7 \cdot y' + 10 \cdot y + 5 \cdot [7 \cdot \cos(5 \cdot x) + 3 \cdot \sin(5 \cdot x)] = 0$;
j) $y'' - 2 \cdot y' + 2 \cdot y = 4 \cdot e^x \cdot (\cos x + \sin x + x \cdot \cos x - x \cdot \sin x)$.

2. Riješite sljedeće Cauchyjeve probleme:

a) $\begin{cases} y' + x \cdot y = 0; \\ y\left(\sqrt{2 \cdot \ln 2}\right) = \dfrac{1}{2}. \end{cases}$

b) $\begin{cases} y' + \dfrac{y}{x} = x^2; \\ y(1) = \dfrac{1}{4}. \end{cases}$

c) $\begin{cases} \dfrac{\ln x}{x} \cdot dx + \dfrac{e^y}{1 + e^y} \cdot dy = 0; \\ y(e^2) = \ln(e^2 - 1). \end{cases}$

d) $\begin{cases} y'' - 2 \cdot y' + y = \cos x; \\ y(0) = 0; \\ y'(0) = 1. \end{cases}$;

e) $\begin{cases} y''-3\cdot y'-4\cdot y+(6\cdot x+1)\cdot e^{x}=0; \\ y(0)=0; \\ y'(0)=6. \end{cases}$

f) $\begin{cases} y''+4\cdot y'+29\cdot y=20\cdot e^{-2\cdot x}\cdot\left[\sin(5\cdot x)-\cos(5\cdot x)\right]; \\ y(0)=4; \\ y'(0)=35. \end{cases}$

3. Odredite jednadžbu krivulje koja prolazi točkom $T(4,\ 1)$ tako da koeficijent smjera bilo koje njezine normale bude jednak kvadratu apscise sjecišta te normale s krivuljom.

4. Odredite jednadžbu krivulje takve da je odsječak na osi Ox bilo koje njezine normale trostruko veći od ordinate sjecišta te normale i tražene krivulje.

5. Odredite jednadžbu krivulje takve da koeficijent smjera bilo koje njezine normale bude trostruko manji od umnoška ordinate i apscise sjecišta te normale i krivulje.

6. Zadana je diferencijalna jednadžba $y'' - y' - 2\cdot y = 0$. Odredite integralnu krivulju koja dodiruje pravac $y = 3$ u točki $T = (0, y)$.

7. Odredite Laplaceove transformate sljedećih realnih funkcija:

a) $f(x)=\sqrt{4\cdot e^{x+2}}\cdot\sin\dfrac{x}{3}\cdot\cos\dfrac{x}{3}$;

b) $f(t)=\sqrt{t}\cdot\left(\cos\dfrac{t}{2}-\sin\dfrac{t}{2}\right)$;

c) $f(a)=(\cos a-\sin a)\cdot\sqrt[4]{e^{3-a}}$;

d) $f(w)=e^{w}\cdot\left[\operatorname{ch} w-\operatorname{sh}(2\cdot w)\right]$.

8. Odredite inverz sljedećih Laplaceovih transformata:

a) $F(s)=\dfrac{2+s-s^2}{(s-1)^3}$;

b) $F(s)=\dfrac{s^2+s+1}{s^3-1}$;

c) $F(s)=\dfrac{s^4+4\cdot s^3-8\cdot s^2+16\cdot s+16}{s^6-4\cdot s^5+12\cdot s^4-32\cdot s^3+48\cdot s^2-64\cdot s+64}$;

d) $F(s)=\dfrac{6\cdot s^3+18\cdot s^2-54\cdot s+162}{s^6-6\cdot s^5+27\cdot s^4-108\cdot s^3+243\cdot s^2-486\cdot s+729}$.

§8. OSNOVE DESKRIPTIVNE STATISTIKE

U svakodnevnome se životu gotovo "od malih nogu" susrećemo s obradom raznih tipova podataka. Primjerice, jedna od glavnih "briga" svakoga učenika nekoga od razreda osnovne ili srednje škole jest postići što bolji uspjeh na kraju svake školske godine. Što je uspjeh bolji, veće su mogućnosti za upis željene srednje škole (za osnovnoškolce), te više ili visoke škole, odnosno fakulteta (za srednjoškolce). Na temelju čega se određuje taj uspjeh? Na to pitanje dobro znamo odgovor: na temelju svih zaključnih ocjena iz odslušanih nastavnih predmeta. A *kako* se određuje taj uspjeh? I na to pitanje znamo odgovor: određuje se *prosjek* ocjena iz svih odslušanih nastavnih predmeta, i to tako da se zbroj svih ocjena podijeli ukupnim brojem odslušanih nastavnih predmeta. Zapravo možemo reći da smo skup od ukupno n ocjena *opisali* samo jednom ocjenom – prosjekom ocjena.

Ovaj primjer je samo jedan od slučajeva kada određeni skup nekih podataka (koji ne moraju nužno biti brojevi) trebamo opisati jednim jedinim podatkom. Razlog je najčešće "velik" broj elemenata početnoga skupa podataka pa nam je znatno lakše praktično baratati samo s jednim podatkom koji *dovoljno dobro* opisuje cijeli "veliki" skup. Obrada i analiza podataka, te interpretacija dobivenih rezultata dio su znanstvene discipline koja se naziva *statistika*.

U ovome ćemo poglavlju podrobnije upoznati načine obrade i analize *numeričkih* podataka – dakle, podataka čije su vrijednosti (tzv. *modaliteti*) iskazani brojevima. (Podatci, inače, mogu biti i *nenumerički*, kao što su npr. spol, bračno stanje, tip automobila itd.) Vidjet ćemo kako pomoću osnovnih i dopunskih funkcija MATLAB-a možemo grupirati podatke, prikazati ih grafički, te ih opisati *osnovnim parametrima statističkoga niza numeričkih podataka*: aritmetičkom sredinom (uobičajeno zvanom *prosjek*), varijancom i standardnom devijacijom. Detaljnija teorijska razmatranja ovdje izostavljamo, a mogu se naći u [6] ili [9].

8.1. Kvalitativna statistička obilježja

Osnovni pojam u statistici je *statistički skup* (kao, uostalom, i skup u matematici). Tvore ga *statističke jedinice* ili *elementi* (osobe, poslovni subjekti, regije, države, predmeti itd.) koji imaju barem jedno zajedničko svojstvo (*obilježje* ili *varijabla*[23]) koje od elementa do elementa očituje statističku promjenjivost. Ukupan broj elemenata statističkoga skupa naziva se *opseg statističkoga skupa*.[24] Svako obilježje se javlja u više pojavnih oblika (*modaliteta*). Skup svih modaliteta nekoga obilježja naziva se *skala*[25]. Tako je npr. skala modaliteta

[23] U opisnoj se statistici izraz *varijabla* obično poistovjećuje s izrazima *obilježje* i *svojstvo*, dok se u matematičkoj statistici odnosi isključivo na numerička obilježja.
[24] U teoriji skupova taj se broj, inače, naziva *kardinalni broj skupa*, dok se termin *opseg* više koristi u geometrijskim razmatranjima.
[25] Unatoč upornu zagovaranju glazbeno nadarenih statističara, sintagma *ljestvica modaliteta* (još) nije ušla u svakodnevnu uporabu.

obilježja "spol" *muški* i *ženski*, skala modaliteta obilježja "dan u tjednu" *ponedjeljak*, *utorak*, *srijeda*, *četvrtak*, *petak*, *subota* i *nedjelja*, skala modaliteta "mjesec u godini" *siječanj*, *veljača*, *ožujak*, *travanj*, *svibanj*, *lipanj*, *srpanj*, *kolovoz*, *rujan*, *listopad*, *studeni* i *prosinac* itd. Prilikom podjele elemenata statističkoga skupa prema barem jednom obilježju kad god je to moguće treba utvrditi skalu modaliteta. To se poglavito odnosi na obilježja čiji se modaliteti ne iskazuju numeričkim vrijednostima (tzv. *kvalitativna obilježja*).

Opća podjela obilježja razlikuje *kvalitativna*, *kvantitativna* i *vremenska obilježja*. *Kvalitativna obilježja* su obilježja kod kojih su svojstva elemenata statističkoga skupa iskazana određenim pridjevima ili atributima, kategorijama i sl. Ova obilježja mogu biti *nominalna* i *redosljedna* (*obilježja ranga*). Potonja se razlikuju od prvih po tome što pripadne modalitete možemo *rangirati* prema nekom prirodnom kriteriju, tj. poredati ih od najboljega prema najlošijemu ili obrnuto. *Kvantitativna obilježja* su obilježja kod kojih se svojstva elemenata statističkoga skupa izražavaju brojevima, a vezana su uz *intervalnu* i *omjernu skalu* iskazivanja modaliteta. Kvantitativna obilježja često se nazivaju i *numerička obilježja*, a dodatno se dijele na *diskretna* (*diskontinuirana*) i *kontinuirana*.

U ovoj ćemo točki ukratko opisati kako u MATLAB-u možemo grafički prikazati kvalitativna statistička obilježja i izračunati neke numeričke pokazatelje vezane uz njih. Nažalost, u MATLAB-u nije moguće *grupirati* kvalitativne statističke podatke (podatke koji se odnose na kvalitativna obilježja). Grupiranjem podataka za svaki pojedini modalitet utvrđujemo kolikom je broju elemenata statističkoga skupa pridružen taj modalitet. Dobiveni broj naziva se *apsolutna frekvencija* (*učestalost*). Neformalno, *apsolutna frekvencija* nekoga modaliteta jednaka je ukupnom broju pojavljivanja toga modaliteta u pripadnom statističkom nizu. Budući da je svakom elementu statističkoga skupa nužno pridružen točno jedan modalitet, zbroj apsolutnih frekvencija svih modaliteta mora biti jednak opsegu statističkoga skupa.

Relativna frekvencija nekoga modaliteta jednaka je količniku odgovarajuće apsolutne frekvencije toga modaliteta i opsega statističkoga skupa. Takav se račun provodi jer želimo vidjeti kolikom dijelu statističkoga skupa je pridružen određeni modalitet. Iako je relativna frekvencija općenito neki nenegativan realan broj iz segmenta [0, 1], radi praktičnih ga je potreba vrlo prikladno izraziti u proporcijama[26], postotcima ili promilima.

***Primjer* 1.** U sljedećoj je tablici prikazana podjela svih žena – magistara znanosti koje su tu akademsku titulu stekle u Republici Hrvatskoj u 2011. godini (stanje na dan 31.12.2011.) prema znanstvenom području magistarskoga rada.

Dakle, u ovom je slučaju statistički skup *žene – magistri znanosti koje su tu akademsku titulu stekle u Republici Hrvatskoj u* 2011. *godini*. Obilježje prema kojemu su podijeljeni elementi statističkoga skupa je *znanstveno područje magistarskoga rada* i to je kvalitativno nominalno statističko obilježje (svi modaliteti su međusobno ravnopravni). Npr. neki modaliteti toga obilježja su *prirodne znanosti*, *tehničke znanosti*, *društvene znanosti* itd.

[26] *Proporcija* označava određeni dio jedinične veličine. Ona je zapravo decimalan broj između 0 i 1.

Znanstveno područje	*Broj osoba*
prirodne znanosti	13
tehničke znanosti	40
biomedicina i zdravstvo	47
biotehničke znanosti	35
društvene znanosti	157
humanističke znanosti	20
Ukupno:	312

Izvor: Priopćenje Državnoga zavoda za statistiku Republike Hrvatske, Zagreb, ožujak 2012. (dostupno na www.dzs.hr)

S točnošću od 10^{-2} izračunajmo pripadne relativne frekvencije (iskazane u postotcima) i prikažimo dobivene podatke grafički strukturnim krugom. Uz grafikon navedimo sve potrebne oznake. Potom interpretirajmo dvije apsolutne frekvencije i njima odgovarajuće relativne frekvencije.

Ako to još nismo učinili, „očistimo“ MATLAB-ov komandni prozor. Apsolutne frekvencije iz drugoga stupca zadane tablice pohranit ćemo u matricu *A*. Dakle, u novi redak komandnoga prozora utipkajmo:

```
A=[13 40 47 35 157 20];
```

Matricu u kojoj će biti ispisane relativne frekvencije iskazane u postotcima označimo s *R*. Prema definiciji relativne frekvencije, matricu *R* dobit ćemo tako da matricu *A* podijelimo zbrojem svih njezinih elemenata. Taj zbroj je realan broj različit od nule, pa je dijeljenje dobro definirano i odvija se prema načelu „član po član“. Dobiveni količnik još treba pomnožiti sa 100 jer rezultat treba biti iskazan u postotcima. Stoga u novi redak komandnoga prozora utipkajmo:

```
R=(1/sum(A)*A)*100
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
R =
  Columns 1 through 4
   4.16666666666667  12.82051282051282  15.06410256410256  11.21794871794872
  Columns 5 through 6
  50.32051282051282   6.41025641025641
```

Tako smo dobili sljedeću tablicu:

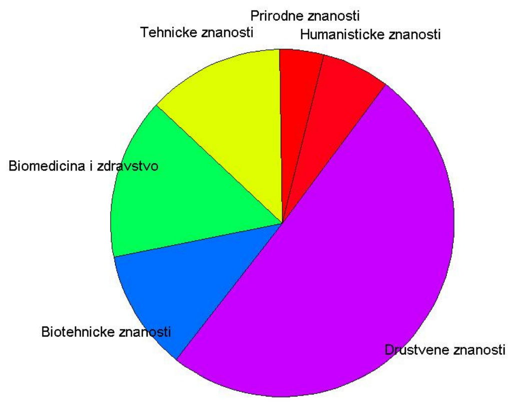
Znanstveno područje	*Struktura* [%]
prirodne znanosti	4.17
tehničke znanosti	12.82
biomedicina i zdravstvo	15.06
biotehničke znanosti	11.22
društvene znanosti	50.32
humanističke znanosti	6.41
Ukupno:	100

Iz dobivenih tablica slijedi npr. da je točno 13, odnosno približno 4.17% promatranih osoba magistriralo iz područja prirodnih znanosti, točno 40, odnosno približno 12.82% promatranih osoba magistriralo iz područja tehničkih znanosti itd. Uočite da je ovdje bitno istaknuti riječi *točno* i *približno*: apsolutne frekvencije 13 i 40 su točno određene (brojanjem pojavnosti pripadnih modaliteta u statističkom nizu), dok su relativne frekvencije 4.17% i 12.82% približno izračunane (s točnošću od 10^{-2}).

Grafički prikaz dobivenih relativnih frekvencija dobit ćemo koristeći funkciju `pie`. Njezin prvi argument bit će matrica s relativnim frekvencijama, dok će drugi argument biti skala modaliteta napisana u prvom stupcu tablice. Stoga u novi redak komandnoga prozora utipkamo:

```
pie(R,{'Prirodne znanosti', 'Tehnicke znanosti', 'Biomedicina i zdravstvo', 'Biotehnicke znanosti', 'Drustvene znanosti', 'Humanisticke znanosti'})
```

Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku:

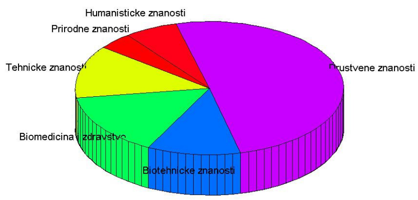


Slika 1.

Ovisno o estetskim ukusima, vizualno dojmljivija slika dobije se korištenjem funkcije `pie3` s istom sintaksom kao i funkcija `pie`. Dakle, utipkavanjem

```
pie3(R,{'Prirodne znanosti', 'Tehnicke znanosti', 'Biomedicina i zdravstvo', 'Biotehnicke znanosti', 'Drustvene znanosti', 'Humanisticke znanosti'})
```

dobijemo:



Slika 2.

Napomena: Česta pogrješka nestatističara je na grafikon uvrštavati različite numeričke podatke. To je potpuno pogrešno. Za kvantitativnu usporedbu numeričkih podataka prikladne su isključivo tablice, dok su grafikoni prikladni za usporedbe tipa „čega ima najviše, a čega najmanje“. U ovom slučaju iz gornjega grafikona možemo zaključiti da je najviše promatranih osoba magistriralo iz društvenih znanosti, a najmanje iz prirodnih znanosti i ništa drugo. Ako su nam bitne numeričke vrijednosti relativnih frekvencija, onda je tablica na str. 166. daleko primjerenija za prikaz takvih podataka od bilo kojega grafikona.

Primjer 2. U sljedećoj je tablici prikazana podjela studenata svih hrvatskih veleučilišta u akademskoj godini 2010/2011. prema godini studija.

Godina studija	*Broj studenata*
prva	10 959
druga	7 569
treća	5 594
Ukupno:	24 122

Izvor: Priopćenje Državnoga zavoda za statistiku Republike Hrvatske, Zagreb, ožujak 2012. (dostupno na www.dzs.hr)

Obilježje *godina studija* pripada u kvalitativno redosljedno obilježje, pa, osim relativnih frekvencija, ima smisla formirati i tzv. *kumulativne nizove* apsolutnih i relativnih frekvencija. Njihove formalne definicije su sljedeće:

Pretpostavimo da su modaliteti promatranoga redosljednoga obilježja (obilježja ranga) poredani od najslabijega prema najboljemu. Tada je:

– *kumulativna apsolutna frekvencija "manje od"* modaliteta x jednaka zbroju apsolutnih frekvencija svih modaliteta (elemenata niza) koji su jednaki ili slabiji od x;
– *kumulativna apsoltuna frekvencija "veće od"* modaliteta x jednaka zbroju apsolutnih frekvencija svih modaliteta (elemenata niza) koji su jednaki ili bolji od x;
– *kumulativna relativna frekvencija "manje od"* modaliteta x jednaka količniku apsolutne kumulativne frekvencije "manje od" toga modaliteta i opsega statističkoga skupa;
– *kumulativna relativna frekvencija "veće od"* modaliteta x jednaka količniku apsolutne kumulativne frekvencije "veće od" toga modaliteta i opsega statističkoga skupa.

Izračunajmo relativne frekvencije (iskazane u postotcima), kumulativne apsolutne frekvencije *manje od*, kumulativne relativne frekvencije *manje od* (iskazane u postotcima), kumulativne apsolutne frekvencije *veće od* i kumulativne relativne frekvencije *veće od* (iskazane u postotcima) za svaki modalitet promatranoga obilježja, pa interpretirajmo vrijednosti koje odgovaraju obilježju *druga*. (Relativne frekvencije izračunajmo s točnošću od 10^{-2}.) Naposljetku, prikažimo zadanu tablicu grafički jednostavnim stupcima, a niz kumulativnih relativnih frekvencija *manje od* grafički tzv. *kumulantom*.

Počinjemo kao i u prethodnom primjeru. Svakako „počistimo“ MATLAB-ov komandni prozor, pa u novi redak toga prozora utipkajmo matricu s apsolutnim frekvencijama:

```
A=[10959 7569 5594];
```

Matricu s relativnim frekvencijama iskazanima u postotcima dobijemo utipkavajući:

```
R=(1/sum(A)*A)*100;
```

u novi redak komandnoga prozora. Zasad tu matricu nećemo ispisivati nego ćemo njezin ispis ostaviti za kraj zadatka.

Matricu kumulativnih apsolutnih frekvencija *manje od* izravno računa MATLAB-ova funkcija `cumsum`. Ona pretpostavlja da su modaliteti poredani od najslabijega prema najboljemu (baš kao i mi maloprije), a njezin jedini argument je matrica apsolutnih frekvencija A. U novi redak radnoga prostora utipkamo:

```
AM=cumsum(A);
```

Analogno, matricu kumulativnih relativnih frekvencija *manje od* dobivamo primjenom funkcije `cumsum` na matricu relativnih frekvencija:

```
RM=cumsum(R);
```

Matricu kumulativnih apsolutnih frekvencija *veće od* nije moguće izravno izračunati jer se prigodom izračuna pretpostavlja da su modaliteti poredani od najboljega prema najslabijemu. Stoga elemente matrice A treba zapisati u obrnutom redoslijedu. Radi jednostavnosti, tu ćemo proceduru u MATLAB-u implementirati kao posebnu funkciju. Otvorimo novu *m*–datoteku i u nju utipkajmo sljedeći niz naredbi:

```
function y=obrnuta(x);
y=x(end:-1:1);
end
```

Pohranimo dobivenu *m*-datoteku pod nazivom **obrnuta.m** i vratimo se u komandni prozor. Daljnja strategija je jednostavna, a vrijedi i za izračun kumulativnih apsolutnih frekvencija *veće od* i za izračun kumulativnih relativnih frekvencija *veće od*:

- formiramo matricu B koja se dobije ispisom elemenata matrice A u obrnutom poretku (time zapravo slažemo apsolutne frekvencije od apsolutne frekvencije najboljega modaliteta do apsolutne frekvencije najslabijega modaliteta);
- primijenimo funkciju `cumsum` na matricu B i dobijemo matricu C;
- ispišemo elemente matrice C u obrnutom poretku (čime dobivamo polazni (tablični) poredak modaliteta kojima odgovaraju izračunane kumulativne frekvencije).

Stoga u nova dva retka radnoga prostora utipkamo:

```
AV=obrnuta(cumsum(obrnuta(A)));
RV=obrnuta(cumsum(obrnuta(R)));
```

Preostaje ispisati svih pet dobivenih matrica. Utipkavanjem

```
R, AM, RM, AV, RV
```

dobijemo:

```
R =
  45.43155625570019  31.37799519111185  23.19044855318796
AM =
       10959       18528       24122
RM =
  1.0e+002 *
   0.45431556255700   0.76809551446812   1.00000000000000
AV =
       24122       13163        5594
RV =
  1.0e+002 *
   1.00000000000000   0.54568443744300   0.23190448553188
```

Dobivene rezultate pregledno zapisujemo u sljedećoj tablici:

Godina studija	*Broj studenata*	*Relativna frekvencija* [%]	*Kumulativna apsolutna frekvencija „manje od“*	*Kumulativna relativna frekvencija „manje od“* [%]	*Kumulativna apsolutna frekvencija „veće od“*	*Kumulativna relativna frekvencija „veće od“* [%]
prva	10 959	45.43	10 959	45.43	24 122	100
druga	7 569	31.38	18 528	76.81	13 163	54.57
treća	5 594	23.19	24 122	100	5 594	23.19
Ukupno:	24 122	100	————	————	————	————

Interpretacije dobivenih podataka za modalitet *druga* su redom sljedeće:

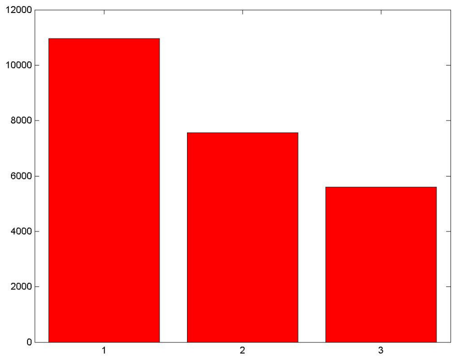
- Točno 7 569 promatranih studenata studira na drugoj godini studija.
- Približno 31.38% promatranih studenata studira na drugoj godini studija.

- Točno 18 528 promatranih studenata studira na prvoj ili drugoj godini studija.
- Približno 76.81% promatranih studenata studira na prvoj ili drugoj godini studija.
- Točno 13 163 promatranih studenata studira na drugoj ili trećoj godini studija.
- Približno 54.57% promatranih studenata studira na drugoj ili trećoj godini studija.

Preostaje grafički prikazati zadanu tablicu i niz kumulativnih relativnih frekvencija *manje od*. Jednostavne stupce dobijemo pomoću MATLAB-ove funkcije `bar`, a kumulantu pomoću funkcije `plot`. Stoga utipkavanjem

```
bar(A)
```

dobijemo:

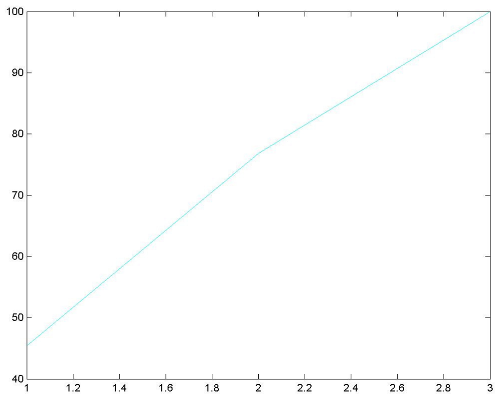


Slika 3.

a utipkavanjem

```
plot(RM)
```

dobijemo:



Slika 4.

8.2. Kvantitativna diskretna statistička obilježja

Pretpostavimo da imamo zadan statistički niz modaliteta nekoga kvantitativnoga diskretnoga obilježja (tj. podataka dobivenim brojanjem):

$$a_1, a_2, \ldots, a_n$$

(Tih podataka ima konačno mnogo, tj. $n \in \mathbf{N}$.) Želimo te podatke prikazati tablično i grafički, te ih opisati pomoću odgovarajućih statističkih pokazatelja.

Osnovna veličina pri tabličnomu prikazu podataka je (ponovno) *apsolutna frekvencija* nekoga podatka. Dakle, ako se modalitet a_i u zadanomu statističkomu nizu javlja ukupno f_i puta, onda kažemo da je *apsolutna frekvencija modaliteta* a_i *jednaka* f_i. Pripadna *relativna frekvencija* (označimo je s r_i). jednaka je količniku apsolutne frekvencije f_i i ukupnoga broja članova niza iskazanom u postotcima. Ako modalitet a_i ima relativnu frekvenciju r_i [%], to znači da je r_i% *svih članova niza jednako* a_i.

Pomoću apsolutnih i relativnih frekvencija formiraju se ranije definirani *nizovi kumulativnih frekvencija*. Da bismo uopće mogli računati bilo koji element niza kumulativnih frekvencija, zadane modalitete moramo *sortirati*, odnosno poredati ili od najvećega prema najmanjemu (tj. poredati *silazno*) ili od najmanjega prema najvećemu (tj. poredati *uzlazno*). MATLAB-ova funkcija *sort* uređuje niz numeričkih podataka od najmanjega prema najvećemu pa ćemo se i mi odlučiti na takav poredak podataka, tim više što MATLAB-ova funkcija `cumsum` za računanje elemenata kumulativnoga niza apsolutnih/relativnih frekvencija *manje od* pretpostavlja upravo takav poredak elemenata. (U slučaju suprotnoga poretka funkcija `cumsum` ispisuje kumulativni niz apsolutnih/relativnih frekvencija *veće od*.)

Dakle, pretpostavimo da je $a_1, a_2, \ldots, a_n$ niz numeričkih modaliteta sortiran od najmanjega prema najvećemu. Radi potpunosti razmatranja, navedimo definicije gore spomenutih kumulativnih frekvencija za slučaj kvantitativnoga obilježja:

– *Kumulativna apsolutna frekvencija "manje od "* modaliteta a_i jednaka je zbroju apsolutnih frekvencija svih modaliteta (elemenata niza) koji su jednaki ili manji od a_i;

- *kumulativna apsolutna frekvencija "veće od "* modaliteta a_i jednaka zbroju apsolutnih frekvencija svih modaliteta (elemenata niza) koji su jednaki ili veći od a_i;

- *kumulativna relativna frekvencija "manje od "* modaliteta a_i jednaka količniku apsolutne kumulativne frekvencije "manje od" toga modaliteta i zbroja apsolutnih frekvencija svih modaliteta, pri čemu se taj količnik obvezatno iskazuje u postotcima;

- *kumulativna relativna frekvencija "veće od "* modaliteta a_i jednaka količniku apsolutne kumulativne frekvencije "veće od" toga modaliteta i zbroja apsolutnih frekvencija svih modaliteta, pri čemu se taj količnik obvezatno iskazuje u postotcima.

Na kraju ovoga kratkoga pregleda statističkih veličina koje će nam kasnije trebati spomenimo i *osnovne numeričke parametre statističkoga niza*. Njihov je pregled dan u sljedećoj tablici.

Naziv parametra	*MATLAB-ova funkcija za računanje parametra*	*Interpretacija parametra*
aritmetička sredina	`mean`	prosječna vrijednost članova niza
varijanca	`var`	prosječno kvadratno odstupanje članova niza od aritmetičke sredine
standardno odstupanje (standardna devijacija)	`std`	prosječno odstupanje članova niza od aritmetičke sredine
koeficijent varijacije	*ne postoji*	relativna mjera raspršenosti podataka, jednaka je količniku standardne devijacije i aritmetičke sredine iskazanome u postotcima

Budući da ćemo sve podatke obrađivati i analizirati pomoću MATLAB-a, "sirove", odnosno neobrađene numeričke podatke zapisujemo matrično u obliku jednoretčane matrice. Razlog tomu je što sve gornje funkcije imaju matričnu ulaznu varijablu. Zbog raznih praktičnih razloga (najčešći takav razlog je pogrješka pri unosu podatka) prikladno je "sirove", odnosno neobrađene podatke zapisati u neku običnu *m*–datoteku koju u svakomu trenutku možemo pozvati iz komandnoga prozora jednostavnim navođenjem njezina imena.

Pogledajmo na primjerima kako se tablično i grafički obrađuju numerički nizovi "sirovih" podataka.

***Primjer* 1.** Državna revizorica Đurđica Knjižić kontrolira točnost knjiženja knjigovodstvenih zapisa. Odabirući 30 uzoraka od kojih se svaki sastoji od točno 20 knjigovodstvenih zapisa bilježila je broj pogrešnih zapisa u svakom uzorku. Dobila je sljedeće podatke:

3, 5, 2, 2, 5, 7, 6, 3, 1, 4, 8, 0, 0, 3, 6, 2, 5, 3, 0, 1, 6, 4, 4, 7, 2, 1, 6, 0, 3, 5

Prikažimo te podatke tablično, odredimo apsolutnu, relativnu, kumulativnu apsolutnu i kumulativnu relativnu frekvenciju "manje od" modaliteta 4, te sve osnovne numeričke parametre ovoga statističkoga niza, pa ih interpretirajmo.

Kako smo rekli, zadani niz "sirovih" podataka najprije zapišemo u obliku jednoretčane matrice u običnu *m*–datoteku. Nazovimo tu datoteku **pogreske.m**. U nju utipkamo (u jednomu retku):

```
x = [3, 5, 2, 2, 5, 7, 6, 3, 1, 4, 8, 0, 0, 3, 6, 2, 5, 3, 0,
1, 6, 4, 4, 7, 2, 1, 6, 0, 3, 5]
```

pohranimo unesene podatke i vratimo se u komandni prozor. U njegovu novome retku utipkajmo:

```
pogreske
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
x =
  Columns 1 through 13
     3     5     2     2     5     7     6     3     1     4     8     0     0
  Columns 14 through 26
     3     6     2     5     3     0     1     6     4     4     7     2     1
  Columns 27 through 30
     6     0     3     5
```

Prikažimo najprije te podatke tablično. Uočimo da se gornji niz sastoji od ukupno 9 različitih vrijednosti (modaliteta): 0, 1, 2, 3, 4, 5, 6, 7 i 8. Svaka od tih vrijednosti pojavljuje se u nizu određeni broj puta, pa najprije moramo odrediti te brojeve. U tu svrhu kreirajmo *m*–datoteku **af.m** koja sadrži jedino funkciju *af* čija je jedina ulazna varijabla matrica *x*, a izlazne varijable matrice *a* i *f* koje sadrže sve različite elemente niza poredane od najmanjega do najvećega, te njihove apsolutne frekvencije.

Otvorimo novu *m*-datoteku **af.m**. U njezin komandni prozor utipkajmo:

```
function [a,f]=af(x)
x=sort(x);
j=1;
n=1;
for k=2:size(x,2)
    if x(k)==x(k-1)
        n=n+1;
    else
        f(j)=n;
        a(j)=x(k-1);
        j=j+1;
        n=1;
    end
a(j)=x(k);
f(j)=n;
end
```

Objasnite svaki redak ove *m*–datoteke. Uočite na koji smo način odredili apsolutne frekvencije svakoga pojedinoga modaliteta.

Pohranimo utipkane naredbe i vratimo se u MATLAB-ov komandni prozor. Utipkajmo:

```
[a,f]=af(x)
```

i dobit ćemo:

```
a =
     0     1     2     3     4     5     6     7     8
f =
     4     3     4     5     3     4     4     2     1
```

Iz tablice očitamo da je apsolutna frekvencija modaliteta 4 jednaka 3, što interpretiramo ovako: *U točno* 3 *uzorka bilo je točno* 4 *pogrešna zapisa*. Izračunajmo sada relativnu frekvenciju toga modaliteta prema njezinoj definicijskoj formuli. Utipkajmo:

```
r=4/30*100
```

i dobit ćemo:

```
r =
  13.33333333333333
```

Taj broj interpretiramo ovako: *U približno* 13.333% *uzoraka bilo je točno* 4 *pogrešna zapisa*. Uočite koji su pokazatelji u toj interpretaciji točno, a koji približno izračunani.

Prijeđimo na izračunavanje elemenata niza kumulativnih frekvencija "*manje od* ". Kao i u prethodnoj točki, utipkajmo:

```
fm=cumsum(f)
```

i dobijemo:

```
fm =
     4     7    11    16    19    23    27    29    30
```

Modalitetu 4 odgovara kumulativna apsolutna frekvencija 19. Taj broj interpretiramo ovako: *U točno* 19 *promatranih uzoraka bilo je najviše* 4 *pogrešna zapisa*.

Pripadnu kumulativnu relativnu frekvenciju određujemo utipkavanjem:

```
krf=19/30*100
```

Dobijemo:

```
krf =
  53.33333333333334
```

Taj broj interpretiramo ovako: *U približno* 53.333% *promatranih uzoraka bilo je najviše* 4 *pogrešna zapisa*. Naravno, sve elemente niza kumulativnih relativnih frekvencija *manje od* dobili bismo utipkavanjem:

```
r=(1/sum(f)*f)*100;
rfm=cumsum(r)
```

Za vježbu, analogno kao u prethodnoj točki, odredite sve članove niza kumulativnih apsolutnih/relativnih frekvencija *veće od*.

Preostaje nam još izračunati i interpretirati osnovne numeričke parametre ovoga niza. Izračunat ćemo ih pomoću funkcijske *m*–datoteke **negrupirani.m** kako bismo izbjegli višekratno utipkavanja niza istih naredbi. Otvorimo novu *m*–datoteku i utipkajmo:

```
function [sv,v,sd,kv]=negrupirani(x);
sv=mean(x);
v=var(x);
sd=std(x);
kv=sd/sv*100;
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. U novi red utipkamo:

```
[sv,v,sd,kv]=negrupirani(x)
```

i dobijemo:

```
sv =
   3.46666666666667
v =
   5.42988505747126
sd =
   2.33021137613549
kv =
  67.21763585006228
```

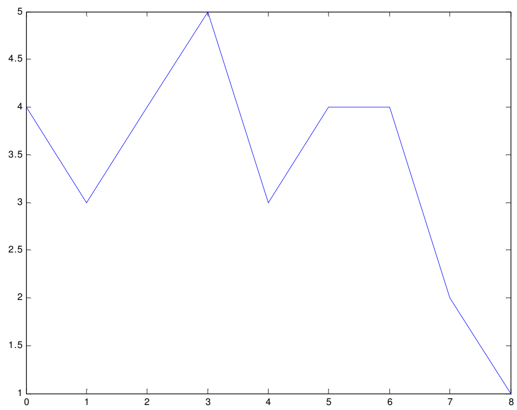
Te parametre redom interpretiramo ovako:

Prosječan broj pogrešaka u jednom uzorku (ili *po jednom uzorku*) *iznosi približno* 3.466667.
Prosječno kvadratno odstupanje <u>broja</u> pogrešaka od aritmetičke sredine iznosi približno 5.42989.
Prosječno odstupanje <u>broja</u> pogrešaka od aritmetičke sredine iznosi približno 2.33021.
Raspršenost pogrešaka oko prosječnoga broja pogrešaka je približno 67.21764% (prilično velika, što znači da se podatci slabo grupiraju oko aritmetičke sredine).

Želimo li obrađene podatke prikazati grafički, to možemo učiniti na više načina. Najčešći prikazi su *poligon* apsolutnih *frekvencija* (ili *mnogokut učestalosti*), već razmatrani jednostavni stupci i *jednostavni retci*. Dobivamo ih rabeći funkcije `plot`, `bar` i `barh`. Utipkajmo najprije:

```
plot(a,f)
```

i dobit ćemo sljedeći graf:



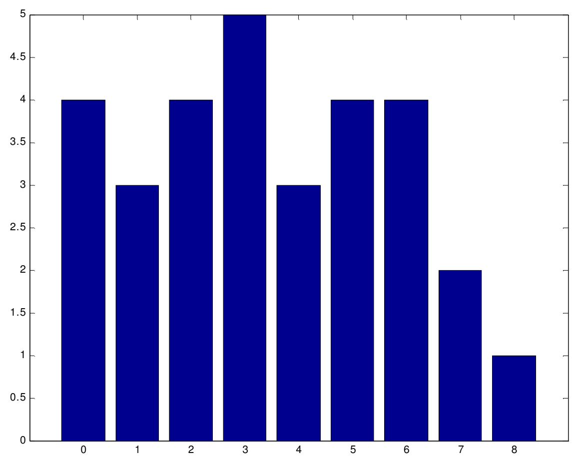
Slika 5.

Kako možemo vidjeti, na os x nanosimo vrijednosti modaliteta, a na os y vrijednosti apsolutnih frekvencija. Par (i, a_i) tvori točku grafa, a svake dvije neposredno susjedne točke spojene su dužinom. Zbog toga ovaj poligon pripada u tzv. *jednostavne izlomljene krivulje*.

Zatvorimo prethodnu sliku, vratimo se u komandni prozor i u njegov novi red utipkajmo:

```
bar(a,f)
```

Dobit ćemo sljedeću sliku:

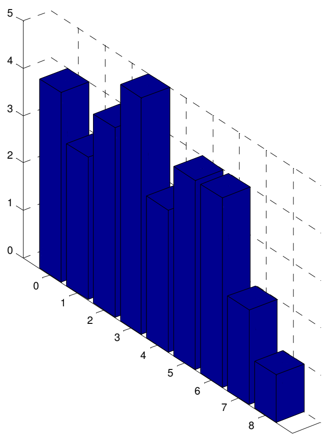


Slika 6.

Želimo li vizualno efektniji prikaz, možemo utipkati:

```
bar3(a,f)
```

što daje sljedeću trodimenzionalnu sliku:

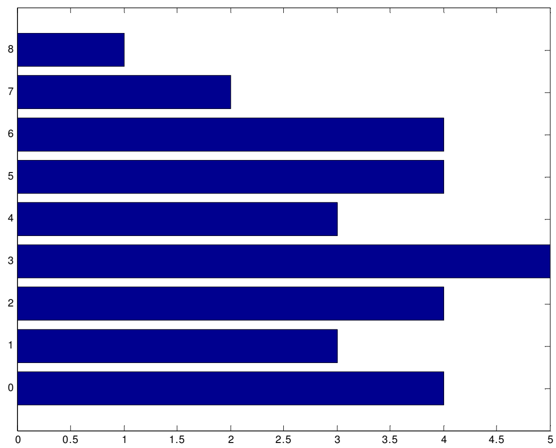


Slika 7.

Potpuno analogno postupamo i u slučaju prikazivanja pomoću jednostavnih redaka. Zatvorimo prethodne slike. U novi red komandnoga prozora utipkamo:

```
barh(a,f)
```

i dobijemo:

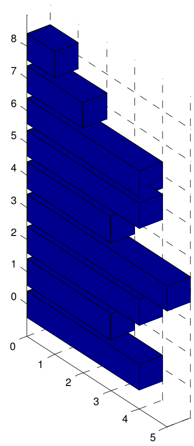


Slika 8.

Ako i ovdje želimo trodimenzionalni prikaz, utipkat ćemo:

```
bar3h(a,f)
```

i dobiti sliku:



Slika 9.

Ovime je obrada i analiza zadanoga niza završena.

Primjer 2. Brojanje poziva u nekoj je telefonskoj centrali vršeno automatski u jednakim vremenskim intervalima od po jedne minute. Kapacitet centrale jest šest poziva u minuti. Telefonski operater bilježio je broj poziva u svakoj minuti tijekom jednoga sata i dobio je sljedeće vrijednosti:

2,3,5,3,0,3,0,5,4,4,6,3,4,6,3,5,6,3,4,1,0,3,4,5,6,3,0,3,0,4,1,2,0,3,4,5,3,5,3,2,3,4,5,3,6,4,3,2,4,2,1,3,4,5,6,4,3,4,2,1.

a) Prikažimo dobiveni statistički niz tablično i grafički (poligonom frekvencija i jednostavnim 3D–stupcima).

b) Odredimo apsolutnu i relativnu frekvenciju modaliteta 0 i interpretirajmo te vrijednosti.

c) Izračunajmo postotak minuta u kojima broj poziva nije bio veći od 4.

d) Izračunajmo prosječan broj poziva u minuti, te odgovarajući pokazatelj raspršenosti brojeva poziva oko prosječnoga broja poziva.

Da bismo riješili sve navedene zadatke, navedene "sirove" podatke najprije moramo zapisati matrično u običnu *m*–datoteku. Nazovimo tu datoteku **pozivi.m**. Otvorimo tu datoteku pa preslikom gornjih vrijednosti (jednostavnom *Copy – Paste* tehnikom) unesimo:

```
x=[2,3,5,3,0,3,0,5,4,4,6,3,4,6,3,5,6,3,4,1,0,3,4,5,6,3,0,3,0,4
,1,2,0,3,4,5,3,5,3,2,3,4,5,3,6,4,3,2,4,2,1,3,4,5,6,4,3,4,2,1];
```

Na kraju gornjega retka stavili smo točku-zarez kako bismo spriječili MATLAB da nam iznova ispiše svih 60 dobivenih podataka. Pohranimo unesene podatke, vratimo se u komandni prozor i u njegov novi redak upišimo:

```
pozivi
```

Ovime smo "neprimjetno" deklarirali matricu x i u komandnomu prozoru. Sada možemo prijeći na rješavanje zadataka.

a) Najprije utipkajmo:

```
[a,f]=af(x)
```

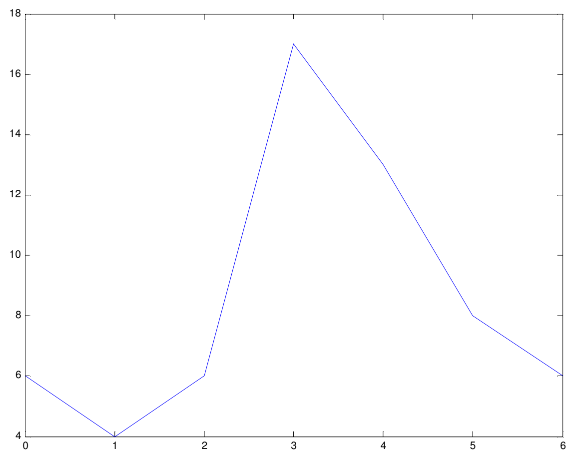
pa će MATLAB ispisati:

```
a =
     0     1     2     3     4     5     6
f =
     6     4     6    17    13     8     6
```

i to je traženi tablični prikaz. Za grafički prikaz u novi red komandnoga prozora najprije utipkajmo:

```
plot(a,f)
```

i dobit ćemo sljedeći poligon frekvencija:

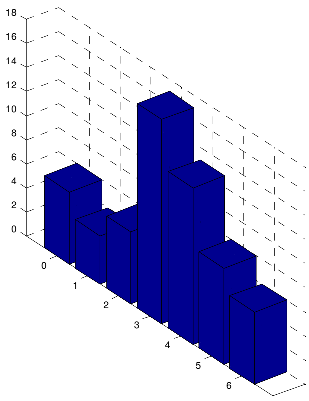


Slika 10.

Zatvorimo dobivenu sliku i vratimo se u komandni prozor. Za prikaz podataka pomoću jednostavnih 3D-stupaca u novi redak upišimo:

```
bar3(a,f)
```

i dobit ćemo traženi prikaz:



Slika 11.

b) Iz tablice dobivene u **a)** zadatku očitavamo da je apsolutna frekvencija modaliteta 0 jednaka 6. To znači da *u točno* 6 *minuta nije bilo niti jednoga poziva*. Relativnu frekvenciju računamo iz njezine definicijske formule:

```
r=6/60*100
```

pa će MATLAB ispisati:

```
r =
    10
```

To znači da *u <u>točno</u>* 10% *minuta nije bilo niti jednoga poziva*.

c) U ovome nam je zadatku zadana *interpretacija* jedne vrste frekvencija za jedan od modaliteta. Riječ "postotak" ukazuje da se radi o relativnim frekvencijama, a riječi "nije bio veći od 4" ukazuju da je riječ o kumulativnoj relativnoj frekvenciji "*manje od*" modaliteta 4. Dakle, tražimo kumulativnu relativnu frekvenciju "*manje od*" modaliteta 4. Da bismo je odredili, najprije moramo izračunati pripadnu kumulativnu apsolutnu frekvenciju. Ona se dobije pomoću funkcije `cumsum`. Dakle, u novi redak komandnoga prozora utipkamo:

```
rfm=(1/sum(f)*cumsum(f))*100
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
rfm =
  1.0e+002 *
  Columns 1 through 4
   0.10000000000000          0.16666666666667          0.26666666666667
0.55000000000000
  Columns 5 through 7
   0.76666666666667   0.90000000000000   1.00000000000000
```

Preostaje očitati traženu kumulativnu relativnu frekvenciju:

```
krf =
  76.66666666666667
```

Dakle, u <u>približno</u> 76.6667% minuta broj poziva nije bio veći od 4.

d) Tekst ovoga zadatka dulji je od njegova rješenja: traže se (samo) vrijednosti aritmetičke sredine i koeficijenta varijacije. Izračunat ćemo ih u novomu retku:

```
sv=mean(x),kv=std(x)/sv*100
```

pa dobivamo:

```
sv =
   3.25000000000000
kv =
  52.42107505975262
```

Dakle, prosječno je bilo 3.25 poziva u jednoj minuti, a raspršenost poziva oko prosječnoga broja poziva iznosi približno 52.421% (relativno velika).

8.3. Kvantitativna kontinuirana statistička obilježja

Za razliku od kvanititativnih diskretnih statističkih obilježja čije modalitete dobivamo brojanjem, modalitete kontinuiranih statističkih obilježja dobivamo mjerenjem. Zbog jako velikoga broja tako dobivenih modaliteta prikladno ih je grupirati u *razrede jednakih širina*. Ukratko ćemo opisati kako za zadane modalitete kvantitativnoga kontinuiranoga statističkoga obilježja formirati grupiranje u razrede jednakih širina, te kako dalje "računati" s takvim razredima. Postoji više takvih "tehnika", a mi ćemo opisati onu koju ćemo primjenjivati u MATLAB-u.

Najprije zadajemo ukupan broj razreda jednakih širina u koje želimo grupirati podatke. Označimo taj broj sa r. Kao najmanji element prvoga (ili početnoga) razreda uzmemo najmanji modalitet (označimo ga sa m), a kao najveći element zadnjega (ili završnoga) razreda uzmemo najveći modalitet (označimo ga sa M). Izračunamo vrijednost

$$d = \frac{M - m}{r}.$$

Veličinu d nazivamo *širina razreda*. Tako definiramo razrede kao sljedeće intervale:

$$[m, m + d\rangle, [m + d, m + 2 \cdot d\rangle, [m + 2 \cdot d, m + 3 \cdot d\rangle, \ldots, [m + (r - 1) \cdot d, M]$$

Uočimo da je razlika supremuma i najmanjega elementa svakoga razreda jednaka d, što opravdava terminologiju „*razredi jednakih širina*“. Sada svaki od modaliteta pridružimo odgovarajućem razredu, pa tako dobivamo *apsolutne frekvencije razreda* f_i. Te frekvencije govore koliko je podataka u kojemu razredu. Njihov ukupan zbroj mora biti jednak ukupnomu broju polaznih podataka (negrupiranih u razrede). Najmanji element svakoga razreda naziva se *donja granica razreda*, a supremum svakoga razreda *gornja granica razreda*.

Nameće se pitanje: kako računati s dobivenim razredima? Mi znamo zbrajati, oduzimati, množiti i dijeliti samo realne brojeve, ali ne i poluotvorene intervale, odnosno segmente. Taj se problem rješava definiranjem *razredne sredine* kao aritmetičke sredine najmanjega i maksimalnoga člana pojedinoga razreda. Drugim riječima, ako je $[A, B\rangle$ ili $[A, B]$ neki razred, onda je razredna sredina toga razreda realan broj s definiran formulom

$$s = \frac{A + B}{2}.$$

Tako svaki od dobivenih razreda "zamijenimo" s njegovom sredinom, pa dobivamo istu situaciju kao i u prethodnoj točki. Zapravo smo problem s kontinuiranim obilježjem "preveli" na problem s diskretnim obilježjem.

U MATLAB-u ćemo opisano grupiranje izvesti pomoću funkcije *sfd* čije su ulazne varijable matrica x s negrupiranim podatcima i ukupan broj razreda r, a izlazne varijable matrica razrednih sredina, matrica njihovih apsolutnih frekvencija i razredna širina svakoga razreda. Pritom prešutno pretpostavljamo da je vrijednost varijable r prirodan broj. Tu ćemo funkciju pohraniti u funkcijskoj *m*-datoteci **sfd.m**.

Otvorimo novu *m*–datoteku, pa u nju utipkajmo sljedeći niz naredbi:

```
function [s,f,d]=sfd(x,r)
m=min(x);
M=max(x);
d=(M-m)/r;
dg=m:d:(M-d);
gg=dg+d;
s=(dg+gg)/2;
f=zeros(1,r);
for k=1:size(x,2)
    j=1;
    while j<=r-1
        if and(x(k)>=dg(j),x(k)<gg(j))
            f(j)=f(j)+1;
        end;
        j=j+1;
    end;
```

```
    if and(x(k)>=dg(r),x(k)<=gg(r))
        f(r)=f(r)+1;
    end;
end
```

Objasnite svaki redak ove *m*–datoteke. Uočite kako smo pomoću logičkoga operatora `and` provjeravali kojem razredu pripada svaki pojedini negrupirani podatak.

Pohranimo upisani niz naredbi pod nazivom **sfd.m** i vratimo se u komandni prozor. Primijenimo dobivenu funkciju na rješavanje sljedećega primjera:

***Primjer* 1.** Mjerenjem masa (u kg) trideset purana u nekomu peradarniku dobiveni su sljedeći podatci:

8.22, 6.03, 5.81, 7.23, 5.67, 4.90, 6.97, 8.65, 8.44, 7.57, 6.89, 6.60, 6.85, 5.52, 5.14, 6.97, 7.98, 5.70, 5.09, 7.22, 5.54, 8.06, 7.33, 5.94, 6.47, 7.24, 6.59, 5.83, 7.58, 6.15

Grupirajmo dobivene podatke u točno $r = 5$ razreda jednakih širina. Izračunajmo osnovne numeričke parametre negrupiranih i grupiranih podataka, interpretirajmo ih i usporedimo. Naposljetku, dobivenu razdiobu podataka prikažimo grafički.

Podatke najprije zapišemo u običnu *m*–datoteku koju ćemo nazvati **purani.m**. Otvorimo je (iz komandnoga prozora) pa utipkajmo:

```
x=[8.22, 6.03, 5.81, 7.23, 5.67, 4.90, 6.97, 8.65, 8.44, 7.57,
6.89, 6.60, 6.85, 5.52, 5.14, 6.97, 7.98, 5.70, 5.09, 7.22,
5.54, 8.06, 7.33, 5.94, 6.47, 7.24, 6.59, 5.83, 7.58, 6.15];
```

Pohranimo unesene podatke i vratimo se u komandni prozor. U novi redak utipkamo:

```
purani
```

Tako smo inicijalizirali matricu x s masama purana. Grupirajmo ih u 5 razreda utipkavanjem:

```
[s,f,d]=sfd(x,5)
```

Dobit ćemo:

```
s =
  Columns 1 through 4
   5.27500000000000   6.02500000000000   6.77500000000000   7.52500000000000
  Column 5
   8.27500000000000
f =
     5     7     7     6     5
c =
   0.75000000000000
```

Dakle, razredne sredine su redom 5.275, 6.025, 6.775, 7.525 i 8.275, a pripadne apsolutne frekvencije redom 5, 7, 7, 6 i 5. Širina svakoga razreda jednaka je 0.75.

Za razliku od primjera iz prethodne točke u kojima smo sve osnovne numeričke parametre računali pomoću odgovarajućih MATLAB-ovih funkcija, ovdje te parametre moramo računati rabeći definicijske formule. Ipak, da ih ne bismo morali neprekidno utipkavati, stvorit ćemo funkcijsku *m*–datoteku **grupirani.m.** Ona će sadržavati jedino funkciju *grupirani* čije su ulazne varijable matrice *s* i *f*, a izlazne varijable vrijednosti aritmetičke sredine, varijance, standardnoga odstupanja i koeficijenta varijacije.

Otvorimo novu *m*–datoteku i u nju utipkajmo:

```
function [sv,v,sd,kv] = grupirani(s,f);
N=sum(f);
sv=s*f'/N;
v=(s.^2)*f'/N-sv^2;
sd=sqrt(v);
kv=sd/sv*100;
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. Utipkajmo:

```
[sv,v,sd,kv]=grupirani(s,f)
```

pa će MATLAB ispisati:

```
sv =
   6.75000000000000
v =
   0.99312499999999
sd =
   0.99655657139973
kv =
  14.76380105777383
```

Interpretacije dobivenih vrijednosti su sljedeće:

Prosječna masa jednoga purana iznosi 6.75 *kg*.
Prosječno kvadratno odstupanje masa purana od prosječne mase iznosi približno 0.993 kg^2.
Prosječno odstupanje masa purana od prosječne mase iznosi približno 0.997 *kg*.
Raspršenost masa purana oko prosječne mase iznosi približno 14.764% (relativno mala).

Izračunajmo sada vrijednosti istih veličina, ali iz negrupiranih podataka. Pritisnimo tipku ↑ i preuredimo posljednje upisanu naredbu ovako:

```
[sv,v,sd,kv]=negrupirani(x)
```

Dobit ćemo:

```
sv =
   6.67266666666667
v =
   1.08962712643678
sd =
   1.043852061566
kv =
  15.64370159206571
```

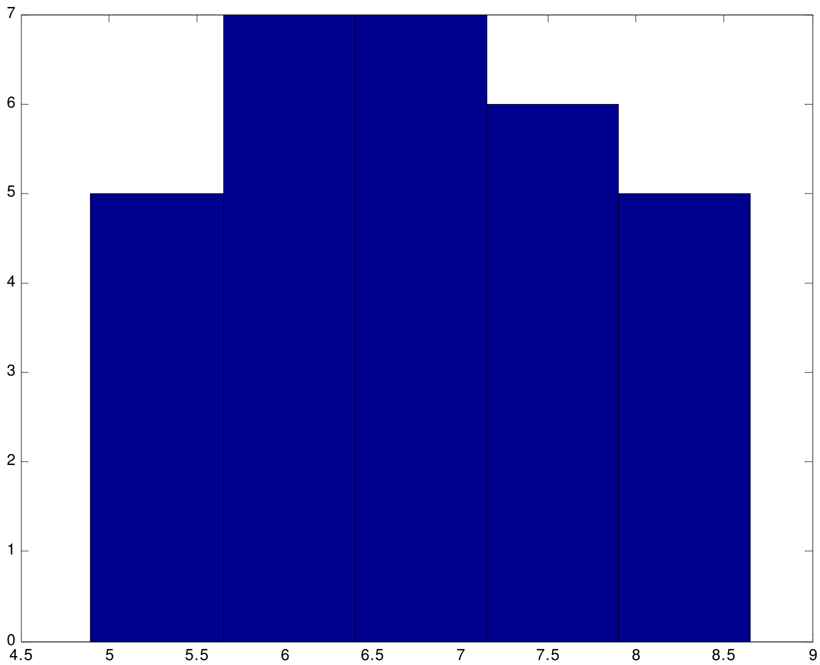
Interpretacije dobivenih vrijednosti su iste kao i za grupirane podatke.

Usporedimo dobivene rezultate. Primjećujemo da je prosječna masa izračunana iz negrupiranih podataka manja od prosječne mase izračunane iz grupiranih podataka, dok su varijanca, standardno odstupanje i koeficijent varijacije veći. Nameće se pitanje: koje vrijednosti *bolje* opisuju zadani niz masâ purana? Odgovor je uvijek: vrijednosti dobivene pomoću *negrupiranih* podataka. Grupiranjem u razrede i zamjenom razreda njegovom sredinom dobivamo na jednostavnosti baratanja s podatcima, ali gubimo na točnosti izračuna osnovnih numeričkih parametara statističkoga niza. Zbog toga je bolje – kad god nije odveć složeno – osnovne numeričke parametre statističkoga niza računati pomoću negrupiranih podataka.

Preostaje dobivene grupirane podatke prikazati grafički. Modaliteti kvantitativnih kontinuiranih statističkih obilježja uvijek se prikazuju *histogramom* (studenti vole reći da su to "slijepljeni jednostavni stupci"). Histogram dobijemo pomoću funkcije *hist* čije su ulazne varijable matrica x negrupiranih podataka i ukupan broj razreda r. U našem slučaju utipkamo:

```
hist(x,5)
```

i dobijemo sljedeći histogram:

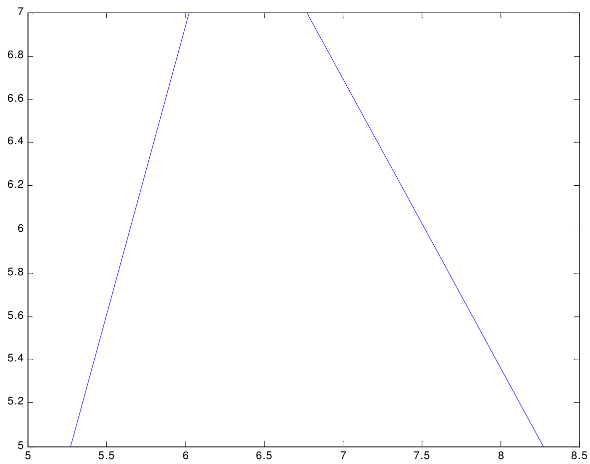


Slika 12.

Poligon frekvencija koji dobijemo utipkavanjem

```
plot(s,f)
```

u ovom slučaju izgleda ovako:



Slika 13.

Primjer 2. Mjerenjem trajanja rada (do prvoga kvara) određenoga broja računala istoga tipa dobiveni su sljedeći rezultati:

Trajanje rada [godina]	*Broj računala*
0.7 – 0.8	4
0.8 – 0.9	11
0.9 – 1.0	22
1.0 – 1.1	8
1.1 – 1.2	5
Ukupno:	50

Izračunajmo relativne frekvencije, pa formirajmo kumulativne nizove *manje od* i *veće od* apsolutnih, odnosno relativnih frekvencija. Objasnimo pokazatelje koji se odnose na treći razred. Potom izračunajmo osnovne numeričke pokazatelje promatrane razdiobe i prikažimo promatranu razdiobu grafički histogramom.

U ovome nam je primjeru već zadana razdioba podataka u prave razrede jednakih širina, pa ne moramo primjenjivati funkciju `sfd`. Budući da su razredi jednakih širina i te širine su jednake 0.1, razredne sredine tvore aritmetički niz kojemu je prvi član $s_1 = \frac{0.7+0.8}{2} = 0.75$, posljednji član $s_5 = \frac{1.1+1.2}{2} = 1.15$, a razlika d = 0.1. Stoga matricu razrednih sredina s generiramo utipkavajući

```
s=0.75:0.1:1.15;
```

u novi redak komandnoga prozora. Pripadnu matricu apsolutnih frekvencija f utipkamo „ručno“ u novi redak komandnoga prozora:

```
f=[4 11 22 8 5];
```

Izračunajmo najprije relativne frekvencije. U novi redak komandnoga prozora utipkajmo:

```
r=(1/sum(f)*f)*100
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
r =
     8    22    44    16    10
```

Zadanu tablicu nadopunimo s trećim stupcem u kojemu ćemo napisati izračunane relativne frekvencije razreda. Dobijemo sljedeću tablicu:

Trajanje rada [godina]	*Broj računala*	*Relativna frekvencija* [%]
0.7 – 0.8	4	8
0.8 – 0.9	11	22
0.9 – 1.0	22	44
1.0 – 1.1	8	16
1.1 – 1.2	5	10
Ukupno:	50	100

Pokazatelj koji se odnosi na treći razred je r_3 = 44. Zaključujemo da <u>*točno*</u> 44% *svih promatranih računala ima trajanje rada između* 0.9 *godina i jedne godine*. (Preciznije: točno 44% promatranih računala ima trajanje rada dulje od 0.9 godina, ali ne dulje od jedne godine.)

U nastavku formirajmo niz apsolutnih/relativnih frekvencija *manje od*. To ćemo učiniti primjenom funkcije `cumsum` na matricu s apsolutnim, odnosno relativnim frekvencijama. U sljedeći redak MATLAB-ova komandnoga prozora utipkajmo:

```
am=cumsum(f), rm=cumsum(r)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
am =
     4    15    37    45    50
rm =
     8    30    74    90   100
```

Tako smo dobili kumulativne nizove *manje od* apsolutnih, odnosno relativnih frekvencija, pa maloprijašnju tablicu nadopunjujemo s još dva stupca:

Trajanje rada [godina]	*Broj računala*	*Relativna frekvencija* [%]	*Kumulativna frekvencija manje od*	*Kumulativna frekvencija manje od* [%]
0.7 – 0.8	4	8	4	8
0.8 – 0.9	11	22	15	30
0.9 – 1.0	22	44	37	74
1.0 – 1.1	8	16	45	90
1.1 – 1.2	5	10	50	100
Ukupno:	50	100	————	————

Pripadni članovi dobivenih nizova koji se odnose na treći razred su $(am)_3 = 37$ i $(rm)_3 = 74$. To znači da <u>*točno*</u> 37, *odnosno* 74% *svih promatranih računala ima trajanje rada ne dulje od jedne godine*.

Odgovarajuće nizove apsolutnih, odnosno relativnih frekvencija *veće od* formirat ćemo koristeći ranije definiranu funkciju `obrnuta`. U novi redak MATLAB-ova komandnoga prozora utipkajmo:

```
av=obrnuta(cumsum(obrnuta(f))), rv=obrnuta(cumsum(obrnuta(r)))
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
av =
    50    46    35    13     5
rv =
   100    92    70    26    10
```

Tako smo dobili još dva niza kumulativnih frekvencija s kojima možemo nadopuniti polaznu tablicu:

Trajanje rada [godina]	*Broj računala*	*Relativna frekvencija* [%]	*Kumulativna frekvencija manje od*	*Kumulativna frekvencija manje od* [%]	*Kumulativna frekvencija veće od*	*Kumulativna frekvencija veće od* [%]
0.7 – 0.8	4	8	4	8	50	100
0.8 – 0.9	11	22	15	30	46	92
0.9 – 1.0	22	44	37	74	35	70
1.0 – 1.1	8	16	45	90	13	26
1.1 – 1.2	5	10	50	100	5	10
Ukupno:	50	100	———	———	———	———

Pripadni članovi tih nizova koji se odnose na treći razred su $(av)_3 = 35$ i $(rv)_3 = 70$. To znači da <u>*točno*</u> 35, *odnosno* 70% *svih promatranih računala ima trajanje rada dulje od* 0.9 *godina*.

Osnovne numeričke pokazatelje promatrane razdiobe (aritmetičku sredinu, varijancu, standardno odstupanje i koeficijent varijacije) izračunat ćemo pokretanjem ranije definirane funkcije `grupirani`. U novi redak komandnoga prozora utipkamo:

```
[sv,v,sd,kv]=grupirani(s,f)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
sv =
   0.94800000000000
v =
   0.01099600000000
sd =
   0.10486181383135
kv =
  11.06137276701963
```

Ove veličine interpretiramo ovako:

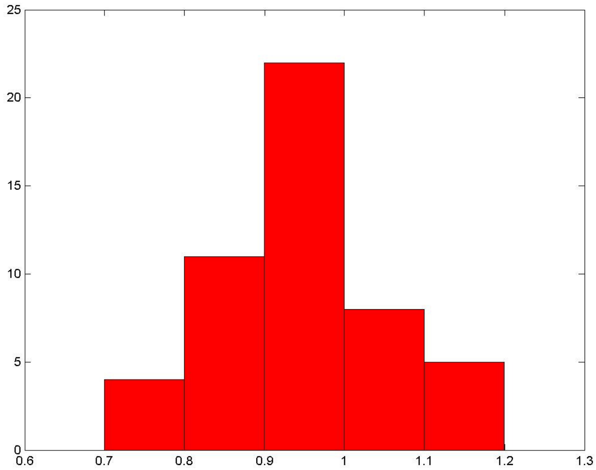
– Prosječno trajanje rada promatranih računala iznosi točno 0.948 godina.

- Pripadna varijanca iznosi (približno) 0.11 godina^2.
- Standardno odstupanje iznosi (približno) 0.10486 godina.
- Koeficijent varijacije iznosi (približno) 11.06%, što znači da je raspršenost podataka oko aritmetičke sredine relativno slaba.

Preostaje grafički prikazati podatke histogramom. U tu svrhu koristimo funkciju `bar`. Prigodom njezina poziva kao treći argument upisujemo 1. (Podsjetimo: funkcija `hist` zahtijeva negrupirane podatke.) Utipkavanjem

```
bar(s,f,1)
```

dobijemo traženi histogram:



Slika 14.

8.4. Zadatci za vježbu

1. U donjoj tablici prikazana je podjela svih obustavljenih pripremnih postupaka Državnoga odvjetništva Republike Hrvatske u 2011. godini (stanje na dan 31.12.2011.) prema razlogu obustave.

Razlog obustave	*Broj obustavljenih postupaka*
nije kazneno djelo	7
beznačajno djelo	6
okolnosti koje isključuju krivnju	10
nema dokaza	65
odustajanje državnoga odvjetnika	34
svrhovitost postupka	98
Ukupno	220

izvor: Mjesečno priopćenje Državnoga zavoda za statistiku Republike Hrvatske, Zagreb, ožujak 2012. (dostupno na: www.dzs.hr)

a) Odredite statistički skup i njegov opseg.
b) Prema kojemu su obilježju podijeljeni svi elementi statističkoga skupa? Kakvo je to obilježje: kvalitativno ili kvantitativno? Objasnite svoj odgovor.
c) Koliko modaliteta ima promatrano obilježje? Navedite ih.
d) Interpretirajte apsolutnu frekvenciju modaliteta *nema dokaza*.
e) S točnošću od 10^{-2} izračunajte relativnu frekvenciju svakoga pojedinoga modaliteta iskazanu u postotcima. Interpretirajte tako dobivenu relativnu frekvenciju modaliteta *svrhovitost postupka*.
f) Podatke iz tablice prikažite grafički jednostavnim stupcima, a dobivene relativne frekvencije prikažite tablično i grafički strukturnim krugom.

2. U donjoj tablici prikazana je podjela žena – istraživača s punim radnim vremenom u Republici Hrvatskoj na dan 31.12.2009. prema sektoru zapošljavanja.

Sektor zapošljavanja	*Broj osoba*
poslovni sektor	592
državni sektor	1 328
neprofitni sektor	3
visoko obrazovanje	2 927
Ukupno	4 850

izvor: Statističke informacije 2011., Državni zavod za statistiku, Zagreb, 2011. (dostupno na: www.dzs.hr)

a) Odredite statistički skup i njegov opseg.
b) Prema kojemu su obilježju podijeljeni svi elementi statističkoga skupa? Kakvo je to obilježje: kvalitativno ili kvantitativno? Objasnite svoj odgovor.
c) Koliko modaliteta ima promatrano obilježje? Navedite ih.
d) Interpretirajte apsolutnu frekvenciju modaliteta *državni sektor*.

e) S točnošću od 10^{-2} izračunajte relativnu frekvenciju svakoga pojedinoga modaliteta iskazanu u postotcima. Interpretirajte tako dobivenu relativnu frekvenciju modaliteta *poslovni sektor*.
f) Podatke iz tablice prikažite grafički jednostavnim stupcima. Dobivene relativne frekvencije prikažite tablično i grafički strukturnim krugom.

3. U donjoj tablici prikazana je podjela svih punoljetnih stanovnika Špičkovine (stanje na dan 31.03.2012.) prema najvišoj završenoj školi.

Najviša završena škola	*Broj osoba*
osnovna	520
srednja	460
veleučilište	200
fakultet	120
Ukupno	1300

a) Odredite statistički skup i njegov opseg.
b) Prema kojemu su obilježju podijeljeni svi elementi statističkoga skupa? Kakvo je to obilježje: kvalitativno ili kvantitativno? Objasnite svoj odgovor.
c) Koliko modaliteta ima promatrano obilježje? Navedite ih.
d) Interpretirajte apsolutnu frekvenciju modaliteta *veleučilište*.
e) S točnošću od 10^{-2} izračunajte relativnu frekvenciju svakoga pojedinoga modaliteta iskazanu u postotcima. Interpretirajte tako dobivenu relativnu frekvenciju modaliteta *osnovna škola*.
f) Formirajte nizove kumulativnih apsolutnih/relativnih frekvencija *manje od* i *veće od*, pa interpretirajte one članove tih nizova koji se odnose na modalitet *srednja*. Relativne frekvencije iskažite u postotcima i izračunajte s točnošću od 10^{-2}.
g) Podatke iz tablice prikažite grafički jednostavnim retcima. Relativne frekvencije prikažite tablično i grafički strukturnim krugom. Niz kumulativnih relativnih frekvencija *manje od* grafički prikažite odgovarajućom kumulantom.

4. U donjoj tablici prikazana je podjela svih stanovnika Frkljevaca prema stupnju zadovoljstva radom načelnika njihova mjesta (stanje na dan 31.03.2012.).

Stupanj zadovoljstva	*Broj osoba*
vrlo nezadovoljan	650
nezadovoljan	170
ni zadovoljan ni nezadovoljan	10
zadovoljan	7
vrlo zadovoljan	5
Ukupno	842

a) Odredite statistički skup i njegov opseg.

b) Prema kojemu su obilježju podijeljeni svi elementi statističkoga skupa? Kakvo je to obilježje: kvalitativno ili kvantitativno? Objasnite svoj odgovor.
c) Koliko modaliteta ima promatrano obilježje? Navedite ih.
d) Interpretirajte apsolutnu frekvenciju modaliteta *nezadovoljan*.
e) S točnošću od 10^{-2} izračunajte relativnu frekvenciju svakoga pojedinoga modaliteta iskazanu u postotcima. Interpretirajte tako dobivenu relativnu frekvenciju modaliteta *ni zadovoljan ni nezadovoljan*.
f) Formirajte nizove kumulativnih apsolutnih/relativnih frekvencija *manje od* i *veće od*, pa interpretirajte one članove tih nizova koji se odnose na modalitet *srednja*. Relativne frekvencije iskažite u postotcima i izračunajte s točnošću od 10^{-2}.
g) Podatke iz tablice prikažite grafički jednostavnim retcima. Relativne frekvencije prikažite tablično i grafički strukturnim krugom. Niz kumulativnih relativnih frekvencijama *manje od* grafički prikažite odgovarajućom kumulantom.

5. Tijekom protekle dvije godine Ivica je svakoga mjeseca bilježio iznos telefonskoga računa (u kn). Dobio je sljedeće vrijednosti:

250.45, 220.79, 235.75, 245.36, 224.46, 236.24, 180.42, 182.64, 228.26, 234.18, 226.89, 232.74, 260.10, 218.32, 225.48, 241.66, 214.83, 233.14, 182.26, 187.85, 227.67, 237.08, 232.49, 229.84.

a) Je li u ovome slučaju riječ o vrijednostima kvantitativnoga diskretnoga ili kvantitativnoga kontinuiranoga statističkoga obilježja? Obrazložite svoj odgovor.
b) Grupirajte dobivene vrijednosti u ukupno 6 razreda jednakih širina i prikažite dobivene rezultate grafički histogramom.
c) Izračunajte relativnu frekvenciju, kumulativnu apsolutnu/relativnu frekvenciju *manje od* i kumulativnu apsolutnu/relativnu frekvenciju *veće od* svakoga pojedinoga razreda dobivenoga u **b)** podzadatku. Sve relativne frekvencije izrazite u postotcima i izračunajte s točnošću od 10^{-2}. Interpretirajte statističke pokazatelje dobivene za četvrti razred.
d) S točnošću od 10^{-5} izračunajte osnovne numeričke parametre zadanoga statističkoga niza rabeći negrupirane i grupirane podatke. Interpretirajte svaki od tih parametara.

6. Tijekom protekloga mjeseca Janica je svakoga dana bilježila ocjene iz školskih ispita i dobila je sljedeće vrijednosti:

3, 2, 4, 4, 2, 5, 5, 2, 2, 4, 3, 4, 2, 2, 5, 4, 4, 3, 2, 4, 3, 3, 5, 4, 5, 3, 2, 5, 4, 2.

a) Je li u ovome slučaju riječ o vrijednostima kvantitativnoga diskretnoga ili kvantitativnoga kontinuiranoga statističkoga obilježja? Obrazložite svoj odgovor.
b) Prikažite dobivene podatke tablično i grafički jednostavnim retcima.
c) Za svaki modalitet dobiven u **b)** podzadatku izračunajte apsolutnu, relativnu, kumulativnu apsolutnu i kumulativnu relativnu frekvenciju "manje od". Relativne frekvencije izrazite u postotcima i izračunajte s točnošću od 10^{-2}. Interpretirajte svaki izračunani pokazatelj za modalitet 3.

d) Izračunajte postotak školskih ispita iz kojih je Janica dobila barem "četvorku".
e) Izračunajte Janičinu prosječnu ocjenu iz školskih ispita, te odgovarajući pokazatelj raspršenosti ocjena oko prosječne ocjene.

7. Rodilište u Vukovaru tijekom proteklih je mjeseci bilježilo porođajnu masu (u g) svih novorođenih beba. Dobivene su sljedeće vrijednosti:

3450, 4550, 3150, 3200, 2700, 4130, 2980, 3325, 3560, 3730, 2450, 3900, 2780, 3890, 4650, 3540, 2930, 3820, 3230, 4522, 4883, 2403, 2776, 3762, 2783, 3895, 4656, 3544, 2942, 3854, 3286, 4572, 3012, 3326, 3794, 3774, 2506, 3908, 2784, 4433, 4673, 3583, 2931, 3852, 3273, 4598, 4322.

a) Je li u ovome slučaju riječ o vrijednostima kvantitativnoga diskretnoga ili kvantitativnoga kontinuiranoga obilježja? Obrazložite svoj odgovor.
b) Grupirajte porođajne mase u ukupno 10 razreda jednakih širina i prikažite dobivene rezultate grafički histogramom.
c) Izračunajte relativnu frekvenciju, kumulativnu apsolutnu/relativnu frekvenciju *manje od* i kumulativnu apsolutnu/relativnu frekvenciju *veće od* svakoga pojedinoga razreda dobivenoga u **b)** podzadatku. Sve relativne frekvencije izrazite u postotcima i izračunajte s točnošću od 10^{-2}. Interpretirajte statističke pokazatelje dobivene za sedmi razred.
d) S točnošću od 10^{-5} izračunajte osnovne numeričke parametre zadanoga statističkoga niza rabeći negrupirane i grupirane podatke. Interpretirajte svaki od tih parametara.

8. Radi povećanja učinkovitosti rada odjela za kredite razvojni odjel banke "*Mufljuzbank*" bilježio je broj usluženih klijenata u jednome satu tijekom jednoga tjedna. Dobiveni su sljedeće vrijednosti:

6, 5, 3, 8, 7, 4, 2, 6, 9, 4, 5, 7, 4, 9, 1, 6, 8, 5, 3, 10, 7, 4, 5, 12, 3, 6, 4, 8, 5, 2, 11, 0, 6, 7, 4, 8, 5, 9, 3, 7, 1, 6, 4, 5, 7, 2, 9, 5, 12, 3, 8, 6, 10, 7, 4, 8, 6, 7, 5, 6.

a) Je li u ovome slučaju riječ o vrijednostima kvantitativnoga diskretnoga ili kvantitativnoga kontinuiranoga obilježja? Obrazložite svoj odgovor.
b) Zadane podatke prikažite tablično i grafički jednostavnim stupcima.
c) Izračunajte relativnu frekvenciju, kumulativnu apsolutnu/relativnu frekvenciju *manje od* i kumulativnu apsolutnu/relativnu frekvenciju *veće od* svakoga modaliteta dobivenoga u **b)** podzadatku. Sve relativne frekvencije izrazite u postotcima i izračunajte s točnošću od 10^{-2}. Interpretirajte statističke pokazatelje dobivene za modalitet 4.
d) Izračunajte prosječan broj usluženih klijenata u jednome satu, te odgovarajući pokazatelj rasipanja zadanih vrijednosti oko toga broja.

9. Na humanitarnoj utrci *Sportom protiv droge* koja se svake godine održava u Šupljoj Lipi ove godine je sudjelovalo ukupno 200 natjecatelja. Svi natjecatelji trčali su po stazi dugoj točno 5 km i počeli su s utrkom u isto vrijeme. Nakon što su svi stigli na cilj, izmjerena su vremena istrčavanja staze i dobiveni podatci su grupirani u sljedeće razrede:

Vrijeme istrčavanja [minuta]	*Broj natjecatelja*
22 – 24	20
24 – 26	30
26 – 28	40
28 – 30	50
30 – 40	60

a) Interpretirajte apsolutnu frekvenciju prvoga razreda.
b) Izračunajte pripadne relativne frekvencije. Interpretirajte relativnu frekvenciju drugoga razreda.
c) Formirajte kumulativne nizove *manje od* apsolutnih, odnosno relativnih frekvencija. Interpretirajte pokazatelje koji se odnose na treći razred.
d) Formirajte kumulativne nizove *veće od* apsolutnih, odnosno relativnih frekvencija. Interpretirajte pokazatelje koji se odnose na četvrti razred.
e) Izračunajte osnovne numeričke pokazatelje promatrane razdiobe. Interpretirajte aritmetičku sredinu, standardno odstupanje i koeficijent varijacije.
f) Promatranu razdiobu grafički prikažite histogramom.

10. Podatci o navršenoj dobi za svakoga od ukupno 40 štićenika Doma za umirovljenike *One Direction* u Ljeskovici grupirani su u razrede i dobivena je sljedeća tablica:

Navršena dob [godina]	*Broj štićenika*
70 – 80	7
80 – 82	12
82 – 84	10
84 – 86	8
86 – 100	3

a) Interpretirajte apsolutnu frekvenciju posljednjega razreda.
b) Izračunajte pripadne relativne frekvencije. Interpretirajte relativnu frekvenciju četvrtoga razreda.
c) Formirajte kumulativne nizove *manje od* apsolutnih, odnosno relativnih frekvencija. Interpretirajte pokazatelje koji se odnose na drugi razred.
d) Formirajte kumulativne nizove *veće od* apsolutnih, odnosno relativnih frekvencija. Interpretirajte pokazatelje koji se odnose na treći razred.
e) Izračunajte osnovne numeričke pokazatelje promatrane razdiobe. Interpretirajte aritmetičku sredinu, standardno odstupanje i koeficijent varijacije.
f) Promatranu razdiobu grafički prikažite histogramom.

11. U praksi se podatci vrlo često grupiraju u razrede čije su donje i gornje granice, te širine višekratnici brojeva 10, 100, 1000 itd. Kreirajte funkcijsku datoteku **sf1.m** koja sadrži jedino funkciju $sf1$ čije su ulazne varijable jednoretčana matrica x čiji su elementi negrupirani modaliteti kvantitativnoga obilježja, te prirodan broj n. Funkcija treba grupirati modalitete u razrede širine n takve da je donja granica prvoga razreda najveći prirodan broj djeljiv s n koji je manji ili jednak svakom od modaliteta, a gornja granica posljednjega razreda najmanji prirodan broj djeljiv s n koji je veći ili jednak svakom od modaliteta. Na izlazu treba ispisati razredne sredine i pripadne apsolutne frekvencije. Provjerite ispravnost rada funkcije na podatcima iz 7. zadatka i za $n = 1000$.

Naputak: U rješavanju zadatka rabite MATLAB-ove aritmetičke funkcije *mod*, *min* i *max*. Funkcija `min` ispisuje najmanji, a funkcija `max` najveći element neke jednoretčane matrice. Funkcija `mod(a,b)` vraća realan broj jednak $a - b \cdot \left\lfloor \frac{a}{b} \right\rfloor$ ako je $b \neq 0$, odnosno a inače[27].

12. Kreirajte funkcijsku *m*–datoteku **najcesci.m** koja sadrži jedino funkciju *najcesci* čija je jedina ulazna varijabla jednoretčana matrica x čiji su elementi negrupirani modaliteti kvantitativnoga diskretnoga obilježja. Na izlazu funkcija treba ispisati modalitet koji se najčešće pojavljuje u statističkome nizu, a ako takvih modaliteta ima više, onda treba ispisati najmanji od njih.

Napomena: Vrijednost koja se najčešće pojavljuje u nekom statističkom nizu naziva se *mod*. Zašto gornju funkcijsku datoteku nismo mogli nazvati **mod.m**?

13. Napišite funkcijsku *m*-datoteku **polozajne.m** koja sadrži jedino funkciju *polozajni* čija je jedina ulazna varijabla jednoretčana matrica x čiji su elementi negrupirani modaliteti kvantitativnoga diskretnoga obilježja. Funkcija treba uzlazno sortirati sve elemente matrice x, te ispisati 1. kvartil, medijan i 3. kvartil tako dobivenoga niza. Navedene vrijednosti definirane su formulama:

srednja vrijednost	*izračun iz negrupiranih podataka*
1. (donji) kvartil	$Q_1 = \begin{cases} x_{\left\lceil \frac{n}{4} \right\rceil}, & \text{ako } n \text{ nije djeljiv s 4;} \\ \frac{1}{2} \cdot (x_{\frac{n}{4}} + x_{\frac{n}{4}+1}), & \text{ako je } n \text{ djeljiv s 4.} \end{cases}$
2. kvartil (medijan)	$Me = Q_2 = \begin{cases} x_{\frac{n+1}{2}}, & \text{ako je } n \text{ neparan;} \\ \frac{1}{2} \cdot (x_{\frac{n}{2}} + x_{\frac{n}{2}+1}), & \text{ako je } n \text{ paran.} \end{cases}$
3. (gornji) kvartil	$Q_3 = \begin{cases} x_{\left\lceil \frac{3}{4} \cdot n \right\rceil}, & \text{ako } n \text{ nije djeljiv s 4;} \\ \frac{1}{2} \cdot (x_{\frac{3}{4} \cdot n} + x_{\frac{3}{4} \cdot n+1}), & \text{ako je } n \text{ djeljiv s 4.} \end{cases}$

Provjerite ispravnost rada funkcije koja određuje medijan korištenjem MATLAB-ove funkcije `median`.

[27] $\lfloor x \rfloor$ je najveći cijeli broj jednak ili manji od x. $\lceil x \rceil$ je najmanji cijeli broj jednak ili veći od x.

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

§9. PRILAGODBA TEORIJSKIH STATISTIČKIH RAZDIOBA EMPIRIJSKIM PODATCIMA

U ovome ćemo poglavlju razmatrati rješavanje sljedećega problema:

Pretpostavimo da su nam zadani skup modaliteta nekoga kvantitativnoga statističkoga obilježja i pripadne apsolutne frekvencije tih modaliteta. Kako već znamo, ti se podatci dobiju statističkim istraživanjem. U ovom se slučaju statističko istraživanje sastoji od izvođenja niza međusobno nezavisnih *slučajnih pokusa*, tj. pokusa čiji ishod ne možemo unaprijed predvidjeti. Kako bismo *s određenom vjerojatnošću* ipak procjenjivali moguće ishode, rabimo *slučajne varijable* koje svakom elementu skupa elementarnih događaja Ω pridružuju točno jedan realan broj.

Slučajne varijable mogu biti *diskretne* (takve da je *slika* slučajne varijable (tj. skup točno svih vrijednosti koje poprima ta varijabla) konačan ili prebrojiv skup), te *neprekidne* ili *kontinuirane* (takve da je slika slučajne varijable neprebrojiv skup (najčešće neki interval ili unija intervala)). Slučajne varijable opisujemo pomoću njihove *funkcije razdiobe* (*distribucije*). Vrijednost te funkcije u nekoj točki a jednaka je vjerojatnosti da slučajna varijabla poprimi vrijednosti (strogo) manje od a. Upoznat ćemo tri takve razdiobe: *binomnu*, *Poissonovu* i *normalnu*, te ćemo naučiti kako odrediti parametre tih razdioba tako da što bolje opisuju rezultate spomenutih pokusa. Kraće kažemo da ćemo *teorijske razdiobe prilagoditi empirijskim podatcima*. Budući da je praktično najteži problem "pogoditi" razdiobu koja najbolje opisuje rezultate slučajnoga pokusa, dat ćemo i kriterije kada uporabiti koju razdiobu.

9.1. Razdiobe diskretnih slučajnih varijabli

Postoji više vrsta razdioba diskretnih slučajnih varijabli, ali praktično najznačajnije i najčešće korištene su *binomna* i *Poissonova* razdioba.

9.1.1. Binomna razdioba

Binomna se razdioba obično javlja kod slučajnih pokusa koji imaju točno dva ishoda: "*uspjeh*" i "*neuspjeh*". Takvi se pokusi nazivaju *Bernoullijevi pokusi*. Primjeri Bernoullijevih pokusa su rođenje djeteta, određivanje je li neki proizvod "škart" ili nije itd. No, to nisu jedini primjeri uporabe binomne razdiobe. Općenito, binomna se razdioba rabi u slučajevima kada promatramo neki slučajan pokus i unaprijed definirani događaj A koji se može, ali i ne mora dogoditi u tom slučajnom pokusu. Nazovimo "*uspjeh*" ishod pokusa u kojemu se dogodio događaj A, a sa "*neuspjeh*" ishod pokusa u kojemu se nije dogodio događaj A. Neka je p vjerojatnost "*uspjeha*". (Uočite da smo na opisani način promatrani slučajni pokus zapravo sveli na Bernoullijev pokus.) Izvedemo li ukupno n slučajnih pokusa, onda događaju A

možemo pridružiti prirodan broj k koji označava ukupan broj "uspjeha" u tih n slučajnih pokusa (tj. koliko se puta kao ishod slučajnoga pokusa pojavio događaj A). Time je definirana *slučajna varijabla X*. Njezina slika je očito *konačan* skup $[n]_0 := \{0, 1, 2, \ldots, n\}$ pa je X diskretna slučajna varijabla.

Osnovni parametri binomne slučajne varijable su ukupan broj pokusa n i vjerojatnost "*uspjeha*" p. Zbog toga pišemo:

$$X \sim B(n, p)$$

Kod opisivanja razdioba uvijek želimo odrediti vjerojatnost da vrijednost slučajne varijable bude jednaka nekom „konkretnom" broju. Općenito, odredimo vjerojatnost da slučajna varijabla X poprimi vrijednost k, tj. da je ukupno k slučajnih pokusa rezultiralo "*uspjehom*", a ostatak slučajnih pokusa (njih $n - k$) "*neuspjehom*". Može se pokazati da vrijedi formula:

$$P(X = k) = \binom{n}{k} \cdot p^k \cdot (1 - p)^{n-k}. \tag{1}$$

Matematičko očekivanje binomne slučajne varijable s jedne je strane *aritmetička sredina empirijskih podataka*, a s druge umnožak $n \cdot p$. Slično, *varijanca* binomne slučajne varijable s jedne je strane *varijanca empirijskih podataka*, a s druge umnožak $n \cdot p \cdot (1 - p)$. Upravo jednakosti

$$sv = n \cdot p,$$
$$v = n \cdot p \cdot (1 - p)$$

omogućuju određivanje parametara prilagođene binomne razdiobe sljedećim algoritmom:

***Korak* 1.** Izračunamo aritmetičku sredinu sv i varijancu v empirijskih podataka. Ako je $v \le sv$, onda postoji binomna razdioba koja će dobro opisati zadane empirijske podatke. Ako je $v > sv$, takva binomna razdioba ne postoji (tj. podatci su distribuirani prema nekoj drugoj razdiobi).

***Korak* 2.** Izračunamo vrijednost parametra p iz jednakosti

$$p = 1 - \frac{v}{sv}.$$

***Korak* 3.** Izračunamo vrijednost parametra n iz jednakosti

$$n = \left\lfloor \frac{sv}{p} \right\rfloor.$$

(Često se dobije da je n jednak najvećemu od modaliteta.)

***Korak* 4.** Za svaki $k \in [n]$ izračunamo vjerojatnost $P(X = k)$ prema formuli (1).

***Korak* 5.** Prema statističkoj definiciji vjerojatnosti, vjerojatnost $P(X = k)$ jednaka je omjeru ukupnoga broja pokusa u kojima se dogodilo točno k "*uspjeha*" i ukupnoga broja izvedenih pokusa. No, ukupan broj pokusa u kojima se dogodilo točno k "*uspjeha*" je statistička interpretacija pojma (teorijske) *apsolutne frekvencije modaliteta k* (označavamo je sa ft_k). Stoga vrijedi jednakost:

$$P(X = k) = \frac{(f_t)_k}{N}, \tag{2}$$

pa teorijske frekvencije računamo pomoću relacije

$$(f_t)_k = N \cdot P(X = k). \tag{3}$$

Implementirajmo sada opisani algoritam u MATLAB-u. Pretpostavimo da su nam ulazne varijable ukupan broj slučajnih pokusa u jednoj seriji (n) i jednoretčana matrica x koja sadrži *ukupne brojeve pojavljivanja "uspjeha" u jednoj seriji*. Ako matrica x ima ukupno s stupaca, znači da smo izveli ukupno s serija slučajnih pokusa (tj. ukupno $s \cdot n$ slučajnih pokusa).

Korak 1. Pomoću funkcije *af* grupiramo elemente matrice x. Izlazne varijable su matrica a s modalitetima $0, 1, \ldots, n$ i matrica f s odgovarajućim apsolutnim frekvencijama tih modaliteta.

Korak 2. Pomoću funkcije *mean* odredimo prosječnu vrijednost *sv* elemenata matrice x.

Korak 3. Pomoću funkcije *var* odredimo varijancu v elemenata matrice x.

Korak 4. Ako je $v \leq sv$, prijeđimo na korak 5. Ako nije, ispišimo: *Tražena razdioba ne postoji*! i završimo postupak.

Korak 5. Izračunamo vrijednost parametra p iz jednakosti $p = 1 - v/sv$.

Korak 6. Izračunamo vrijednost parametra n iz jednakosti $n = \text{round}(sv/p)$.

Korak 7. Pomoću funkcije *sum* izračunamo zbroj svih elemenata matrice f. Taj broj označimo s N.

Korak 8. Izračunamo teorijske frekvencije rabeći funkcijsku *m*–datoteku **binomna.m**. Ta datoteka sadrži funkciju *binomna* čije su ulazne varijable redom N, n, p i a, a izlazna varijabla matrica *ft* čiji su elementi upravo teorijske frekvencije. Njezin sadržaj je sljedeći:

```
function ft=binomna(N,n,p,a);
for i=1:size(a,2)
    ft(i)=round(N*(nchoosek(n,a(i)))*p^(a(i))*(1-p)^(n-a(i)));
end
```

U ovoj smo datoteci koristili MATLAB-ovu ugrađenu funkciju `nchoosek` koja računa vrijednosti binomnih koeficijenata. Ona ima dva ulazna argumenta: nenegativne cijele brojeve n i k, te jedan izlazni argument: binomni koeficijent $\binom{n}{k}$.

Radi jednostavnosti, sve ove korake implementirat ćemo u vidu obične *m*–datoteke **bpr.m**. Otvorimo novu *m* – datoteku pa utipkajmo redom:

```
[a,f]=af(x);
sv=mean(x);
v=var(x);
if v > sv
    error('Trazena binomna razdioba ne postoji!')
end
p=1-v/sv;
n=round(sv/p);
N=sum(f);
n
p
a
f
ft=binomna(N,n,p,a)
```

U ovoj smo datoteci koristili MATLAB-ovu ugrađenu funkciju `error`. Ta funkcija ima jedan ulazni argument (obično neki prikladan tekst), a njezinim izvršenjem ispisuje se ulazni argument, prekida se rad funkcijske *m*-datoteke i vraća se u komandni prozor.

Pohranimo upisane naredbe i vratimo se u komandni prozor. Ako su matrice a i f već zadane (tj. ako su podatci već grupirani), datoteku **bpr.m** modificiramo u sljedeću datoteku **bprg.m**:

```
N=sum(f);
sv=a*f'/N;
v=(a.^2)*f'/N-sv^2;
if v > sv
    error('Trazena binomna razdioba ne postoji!')
end
p=1-v/sv;
n=round(sv/p);
n
p
a
f
ft=binomna(N,n,p,a)
```

Pohranimo upisane naredbe i vratimo se u komandni prozor. Ilustrirajmo primjenu opisanih algoritama na primjerima.

Primjer 1. Vratimo se na Primjer 1 iz točke 8.2. U tome je primjeru slučajan pokus *određivanje ispravnosti zapisa*. On ima točno dva ishoda: "*zapis je točan*" i "*zapis je pogrešan*", pa je riječ o Bernoullijevu pokusu. Provjerimo postoji li binomna razdioba koja dobro opisuje taj pokus.

U komandnomu prozoru utipkajmo redom:

```
x = [3, 5, 2, 2, 5, 7, 6, 3, 1, 4, 8, 0, 0, 3, 6, 2, 5, 3, 0,
1, 6, 4, 4, 7, 2, 1, 6, 0, 3, 5];
bpr
```

i dobit ćemo:

```
??? Error using ==> bpr
Trazena binomna razdioba ne postoji!
```

Dakle, ne postoji binomna razdioba koja bi dobro opisivala ove slučajne pokuse (varijanca je veća od artimetičke sredine).

Primjer 2. U tvornici električnih žarulja "*Žaruljčica*" žarulje se proizvode na pokretnoj traci. U pravilnim vremenskim razmacima obavlja se provjera proizvodnje tako da se slučajno izabere ukupno 8 žarulja i ispita koliko je od njih neispravnih. Rezultati velikoga broja ispitivanja dani su u donjoj tablici.

Broj "škartova"	*Broj žarulja*
0	95
1	238
2	322
3	288
4	121
5	81
6	32
7	16
8	4

a) Provjerimo jesu li gornji podatci raspoređeni prema binomnoj razdiobi i, ako jesu, odredimo njezine parametre.
b) Izračunajmo vjerojatnost da će u jednoj seriji žarulja biti najviše 3 neispravne žarulje rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.
c) Izračunajmo vjerojatnost da će u jednoj seriji žarulja biti barem 6 neispravnih žarulja rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.

a) U ovome je primjeru slučajan pokus *ispitivanje ispravnosti žarulje* koji ima točno dva moguća ishoda ("*ispravna*" i "*neispravna*"). Riječ je, dakle, o Bernoullijevu pokusu, pa postoji mogućnost da su podatci raspoređeni prema binomnoj razdiobi. Kako su oni već grupirani, matrice *a* i *f* morat ćemo unijeti "ručno". U komandnomu prozoru utipkamo:

```
a=0:8;
f=[95 238 322 288 121 81 32 16 4];
```

pa pokrenemo datoteku **bprg.m**:

```
bprg
```

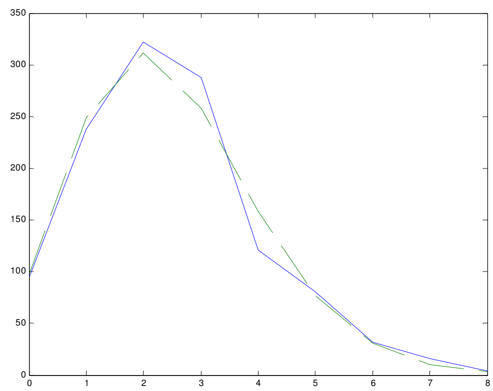
Dobijemo:

```
n =
    99
p =
   0.02495327289389
a =
     0     1     2     3     4     5     6     7     8
f =
    95   238   322   288   121    81    32    16     4
ft =
    98   249   312   258   158    77    31    10     3
```

Kvalitetu opisa rezultata pomoću binomne razdiobe provjerimo utipkavajući

```
plot(a,f,a,ft,'--')
```

pa dobijemo sljedeću sliku:



Slika 1.

(Plava crta je poligon empirijskih frekvencija, a zelena iscrtkana crta poligon teorijskih frekvencija.)

b) Događaj *"u jednoj seriji ima najviše* 3 *neispravne žarulje*" disjunktna je unija sljedećih četiriju događaja: "*u jednoj seriji nema neispravnih žarulja*", "*u jednoj seriji je točno jedna žarulja neispravna* ", "*u jednoj seriji su točno dvije žarulje neispravne*" i "*u jednoj seriji su točno tri žarulje neispravne*". Vjerojatnosti tih događaja su redom $P(X = 0)$, $P(X = 1)$, $P(X = 2)$ i $P(X = 3)$, pa je

$$P(X \leq 3) = P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3).$$

***Napomena*:** Za *diskretne* razdiobe X općenito vrijedi sljedeća formula:

$$P(X \leq k) = \sum_{i=1}^{k} P(X = i)\,. \tag{4}$$

Da ne bismo morali zasebno računati četiri vjerojatnosti, kreirajmo običnu *m*–datoteku **najvise3.m** u kojoj ćemo izračunati tu vjerojatnost koristeći *for*–petlju. Otvorimo novu *m* –datoteku pa u nju utipkajmo:

```
p3=0;
for i=0:3
    p3=p3+nchoosek(n,i)*p^i*(1-p)^(n-i);
end
p3
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. U novi redak utipkajmo:

```
najvise3
```

i dobit ćemo traženu vjerojatnost:

```
p3 =
   0.76533399606566
```

Sad istu vjerojatnost izračunajmo rabeći statističku definiciju vjerojatnosti. Primijetite da je p_3 zapravo kumulativna relativna frekvencija modaliteta 3. Stoga je:

$$p_3 = \frac{95+238+322+288}{95+238+322+288+121+81+32+16+4} = \frac{943}{1197} = 0.78780284043442.$$

c) Događaj "*u jednoj seriji ima barem* 6 *neispravnih žarulja*" disjunktna je unija sljedećih triju događaja: "*u jednoj seriji ima točno* 6 *neispravnih žarulja*", "*u jednoj seriji ima točno* 7 *neispravnih žarulja*" i "*u jednoj seriji ima točno* 8 *neispravnih žarulja*". Vjerojatnosti tih događaja su redom $P(X = 6)$, $P(X = 7)$ i $P(X = 8)$. Stoga je

$$P(X \geq 6) = P(X = 6) + P(X = 7) + P(X = 8).$$

***Napomena*:** Za *diskretne* razdiobe općenito vrijedi sljedeća formula:

$$P(X \geq k) = \sum_{i=k}^{n} P(X = i)\,. \tag{5}$$

Slično kao u prethodnom zadatku, kreirajmo običnu *m*–datoteku **barem6.m** u kojoj ćemo izračunati traženu vjerojatnost rabeći *for* – petlju. Otvorimo novu *m*–datoteku pa u nju utipkajmo:

```
p6=0;
for i=6:n
    p6=p6+nchoosek(n,i)*p^i*(1-p)^(n-i);
end
p6
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. U novi redak utipkajmo:

```
barem6
```

i dobit ćemo traženu vjerojatnost:

```
p6 =
   0.03800901472271
```

(Uočite da nas je MATLAB višekratno upozorio da rezultat može biti nedovoljno precizan zbog uporabe funkcije *nchoosek* koja računa vrijednosti binomnih koeficijenata!)

Rabimo li statističku definiciju vjerojatnosti, dobit ćemo:

$$p_6 = \frac{32+16+4}{95+238+322+288+121+81+32+16+4} = \frac{52}{1197} = 0.04344193817878.$$

Primjer 3. Radi povećanja kakvoće proizvodnje keksa u tvornici "*Keksić*", kontrolor proizvodnje uzeo je ukupno 50 uzoraka od po 10 kutija keksa i bilježio koliko je kutija s keksima nedovoljne kakvoće u svakom uzorku. Dobio je sljedeće podatke:

0,1,4,2,3,0,5,4,1,3,2,0,4,2,3,7,2,1,3,4,3,2,6,2,5,3,1,4,3,2,5,3,1,4,3,6,2,4,1,5,3,2,4,1,3,2,5,4,3,1.

a) Provjerimo jesu li gornji podatci raspoređeni prema binomnoj razdiobi. Ako jesu, odredimo njezine parametre.
b) Izračunajmo vjerojatnost da je u jednomu uzorku bilo točno 5 kutija keksa nedovoljne kakvoće rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.
c) Izračunajmo vjerojatnost da u jednomu uzorku postoji barem jedna kutija keksa nedovoljne kakvoće rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.

U ovome nam primjeru podatci nisu grupirani pa ćemo **a)** zadatak riješiti rabeći *m*–datoteku **bpr.m**.

a) Utipkajmo:

```
x=[0,1,4,2,3,0,5,4,1,3,2,0,4,2,3,7,2,1,3,4,3,2,6,2,5,3,1,4,3,2
,5,3,1,4,3,6,2,4,1,5,3,2,4,1,3,2,5,4,3,1];
bpr
```

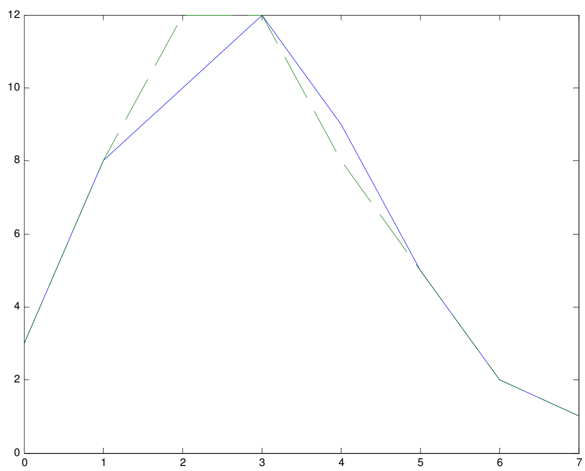
pa će MATLAB ispisati

```
n =
    41
p =
   0.06972789115646
a =
     0     1     2     3     4     5     6     7
f =
     3     8    10    12     9     5     2     1
ft =
     3     8    12    12     8     5     2     1
```

Vidimo da su podatci raspoređeni prema binomnoj razdiobi $B(41,\ 0.06972789115646)$. Utipkajmo nadalje:

```
plot(a,f,a,ft,'--')
```

i dobit ćemo sljedeću sliku:



Slika 2.

b) U ovome se zadatku traži vjerojatnost $P(X = 5)$. Izračunajmo je najprije koristeći klasičnu definiciju vjerojatnosti. U formulu (1) uvrstimo $k = 5$, $n = 41$, $p = 0.06972789115646$. U novomu retku komandnoga prozora utipkajmo:

```
p5=nchoosek(n,5)*p^5*(1-p)^(n-5)
```

(Sjetite se da varijable n i p već imaju željene vrijednosti dobivene izvršenjem naredbi zapisanih u *m*–datoteci **bpr.m**!) Pritisnimo *Enter*, pa će MATLAB ispisati:

```
p5 =
   0.09155987523158
```

Računamo li istu vjerojatnost rabeći statističku definiciju vjerojatnosti, dobit ćemo:

$$p_5 = \frac{5}{50} = 0.1\,.$$

c) Događaj "*u jednomu uzorku postoji barem jedna kutija keksa nedovoljne kakvoće*" jednak je disjunktnoj uniji sljedećih događaja: "*u jednomu uzorku postoji točno jedna kutija keksa nedovoljne kakvoće*", "*u jednom uzorku postoje točno dvije kutije keksa nedovoljne kakvoće*", … "*u jednom uzorku postoji točno deset kutija keksa nedovoljne kakvoće*". Stoga bismo mogli primijeniti formulu (5) i pomoću *for* – petlje izračunati traženu vjerojatnost. No, možemo postupiti brže i bolje. Zadanom događaju suprotan događaj jest "*u jednomu uzorku nema niti jedna kutija keksa nedovoljne kakvoće*". Njegova je vjerojatnost $P(X = 0)$. Zbog toga vrijedi:

$$P(X \geq 1) + P(X = 0) = 1.$$

Odatle je

$$P(X \geq 1) = 1 - P(X = 0).$$

Vjerojatnost $P(X = 0)$ jednaka je $(1 - p)^{41}$, pa utipkajmo:

```
p0 = 1-(1-p)^41
```

i dobit ćemo:

```
p0 =
   0.94835717271207
```

Rabimo li statističku definiciju vjerojatnosti, dobit ćemo:

$$p_0 = 1 - \frac{3}{50} = 0.94\,.$$

9.1.2. Poissonova razdioba

Za razliku od binomne razdiobe čija je slika uvijek konačan skup, slika Poissonove razdiobe može biti i prebrojiv skup. Budući da *prebrojiv* znači da postoji bijekcija između toga skupa i skupa prirodnih brojeva **N**, uvijek se uzima da je slika Poissonove razdiobe podskup skupa **N** bez obzira što stvarna slika te razdiobe možda sadržava i negativne cijele brojeve.

Osnovni parametar Poissonove razdiobe jest strogo pozitivan realan broj λ. Upravo je on "detektor" jesu li neki podatci raspoređeni prema Poissonovoj razdiobi ili nisu. Naime, može se pokazati da su i očekivanje i varijanca Poissonove razdiobe jednaki upravo λ. Kad utvrdimo da su podatci raspoređeni prema Poissonovoj razdiobi, onda – analogno kao i kod binomne razdiobe – za svaki $k \in \mathbf{N}_0 := \{0, 1, 2, \ldots\}$ određujemo vjerojatnost $P(X = k)$, gdje je s X označena Poissonova slučajna varijabla (tj. slučajna varijabla raspodijeljena prema Poissonovoj razdiobi). Vrijedi sljedeća formula:

$$P(X = k) = \frac{\lambda^k}{k!} \cdot e^{-\lambda}. \tag{6}$$

Poissonova razdioba ponekad se naziva i *zakon malih brojeva*, a rabi se kod tzv. "rijetkih procesa", kao što su npr.

- prometne nesreće na određenoj dionici autoceste u jednome danu;
- telefonski pozivi centrali u jednoj minuti;
- broj otkazivanja kočnica automobila tijekom njegova "radnoga vijeka";
- broj tiskarskih grešaka na jednoj stranici neke knjige;
- defekti po jedinici duljine bakrene žice;
- broj čestica kozmičkog zračenja detektiranih u sekundi;
- broj oboljelih stabala po aru šume itd.

Također, valja napomenuti da za velike vrijednosti parametra n, a vrlo male vrijednosti parametra p binomna razdioba "prelazi" u Poissonovu s parametrom $\lambda = n \cdot p$. Drugim riječima, u takvim slučajevima Poissonova razdioba vrlo dobro "zamjenjuje" (aproksimira) binomnu. O tome govori tzv. Poissonov poučak koji se uči u teoriji vjerojatnosti.

I ovdje ćemo promatrati problem prilagodbe Poissonove razdiobe empirijskim podatcima. U prvi se mah taj zadatak može činiti "nemogućom misijom" jer se postavlja uvjet da aritmetička sredina i varijanca moraju biti jednake, što je u praksi gotovo nemoguće. No, mi želimo *približno* opisati razdiobu empirijskih podataka pomoću Poissonove razdiobe pa nećemo zahtijevati da aritmetička sredina i varijanca budu jednake. Kvaliteta opisa bit će nam apsolutna vrijednost razlike artimetičke sredine i varijance. Ako je ta vrijednost velika, opis pomoću Poissonove razdiobe bit će relativno loš. Ako je ta vrijednost mala, opis pomoću Poissonove razdiobe bit će relativno dobar. Naravno, kvalitetu ćemo uvijek provjeravati i vizualno crtajući poligone empirijskih, odnosno teorijskih frekvencija na istoj slici.

Osnovni algoritam za određivanje parametra prilagođene Poissonove razdiobe jest:

Korak 1. Izračunamo aritmetičku sredinu *sv* zadanih podataka, te njihov ukupan broj *N*.

Korak 2. Definiramo parametar λ s $\lambda := sv$.

Korak 3. Odredimo teorijske frekvencije iz jednakosti

$$(f_t)_k = \lfloor N \cdot P(X = k) \rfloor$$

Opisani algoritam implementiran je u MATLAB-u kao funkcijska *m*–datoteka **poissonova.m**, a pretpostavlja da su podatci već grupirani. Sadržaj te datoteke je sljedeći:

```
function ft = poissonova(a,f)
N=sum(f);
sv=a*f'/N;
lambda=sv;
for i=1:size(f,2)
ft(i)=round(N*(sv^(a(i)))*exp(-sv)/factorial(a(i)));
end
lambda
a
f
```

Ako imamo negrupirane podatke, najprije ih moramo grupirati pomoću funkcije *af* u komandnomu prozoru, a potom pozvati spomenutu funkciju *poissonova*. Ilustrirajmo to na primjerima.

***Primjer* 4.** Radi poboljšanja kvalitete svojih usluga upravnik banke "*Drp–bank*" proveo je brojanje stranaka koje tijekom jednoga sata obave uplatu ili isplatu na jednom šalteru te banke. Dobiveni podatci navedeni su u donjoj tablici.

Broj stranaka po satu	*Apsolutna frekvencija*
0	1
1	4
2	8
3	12
4	15
5	17
6	13
7	9
8	5
9	4
10	2
11	1

a) Provjerimo jesu li gornji podatci raspoređeni prema Poissonovoj razdiobi i, ako jesu, prilagodimo Poissonovu razdiobu dobivenim podatcima.
b) Izračunajmo vjerojatnost da u jednome satu ne bude usluženo manje od četiri klijenta rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.
c) Izračunajmo vjerojatnost da u jednome satu broj klijenata bude djeljiv sa 3 rabeći klasičnu, odnosno statistčku definiciju vjerojatnosti.

Ovdje je očito riječ o "rijetkom" procesu pa je moguće da će Poissonova razdioba dobro opisati razdiobu dobivenih podataka. Utipkajmo najprije:

```
a=0:11;
f=[1 4 8 12 15 17 13 9 5 4 2 1];
```

Koristeći *m*–datoteku **grupirani.m** koju smo stvorili u 5. poglavlju izračunat ćemo vrijednosti aritmetičke sredine i varijance, pa ako su one približno jednake, Poissonova će razdioba dobro opisati razdiobu dobivenih podataka. Utipkajmo:

```
[sv,v,sd,k]=grupirani(a,f)
```

pa će MATLAB ispisati:

```
sv =
   4.93406593406593
v =
   5.13850984180654
d =
   2.26682814562696
k =
  45.94239671537928
```

Razlika između aritmetičke sredine i varijance iznosi približno 0.2 pa možemo očekivati dobar opis pomoću Poissonove razdiobe. Utipkajmo:

```
ft = poissonova(a,f)
```

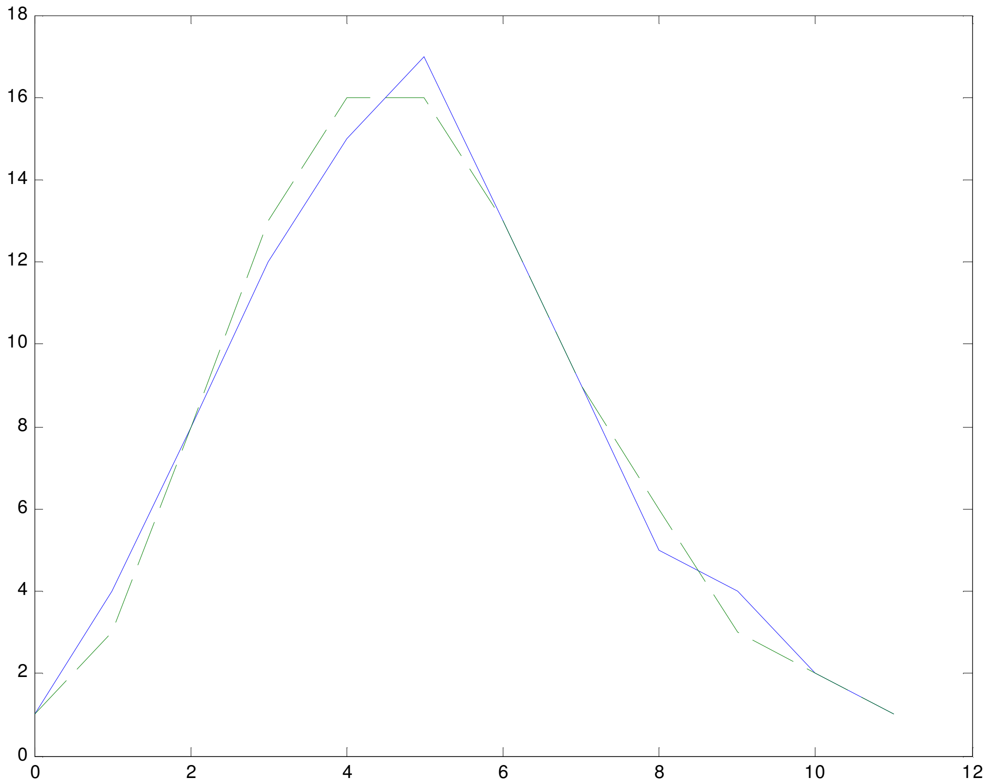
i dobit ćemo:

```
lambda =
   4.93406593406593
a =
     0     1     2     3     4     5     6     7     8     9    10    11
f =
     1     4     8    12    15    17    13     9     5     4     2     1
ft =
     1     3     8    13    16    16    13     9     6     3     2     1
```

Provjerimo kvalitetu dobivene prilagodbe i grafički utipkavajući:

```
plot(a,f,a,ft,'--')
```

Dobit ćemo sljedeću sliku:



Slika 3.

b) Događaj "*u jednome satu nije usluženo manje od četiri klijenta*" jednak je događaju "*u jednome su satu uslužena barem četiri klijenta*", pa je

$$P(X \geq 4) = \sum_{k=4}^{11} P(X = k).$$

No, ovdje nam je ponovno lakše i brže izračunati vjerojatnost zadanom događaju suprotnoga događaja. To je događaj "*u jednome je satu usluženo manje od četiri klijenta*", a taj je jednak događaju "*u jednome je satu usluženo najviše troje klijenata*". Prema tome je

$$P(X \geq 4) + P(X \leq 3) = 1,$$

otkuda je

$$P(X \geq 4) = 1 - P(X \leq 3),$$

odnosno

$$P(X \geq 4) = 1 - \sum_{k=0}^{3} P(X = k).$$

Kraće označimo:

$$p_4 := P(X \geq 4).$$

Izračunajmo najprije p_4 rabeći klasičnu definiciju vjerojatnosti. U tu ćemo svrhu iskoristiti MATLAB i u običnoj *m*–datoteci **p4.m** koristeći *for* – petlju izračunati željenu vrijednost. Otvorimo novu *m*–datoteku pa u nju utipkajmo:

```
p4=0;
lambda=4.93406593406593;
for i=0:3
    p4=p4+(lambda^i)*exp(-lambda)/factorial(i);
end
p4=1-p4
```

Pohranimo upisane naredbe i vratimo se u komandni prozor. U novi red utipkajmo:

```
p4
```

i MATLAB će ispisati:

```
p4 =
   0.72559635961475
```

Izračunajmo sada istu veličinu rabeći statističku definiciju vjerojatnosti. Uočimo da je zbroj na desnoj strani odredbenoga izraza za p_4 zapravo kumulativna relativna frekvencija modaliteta 3. Stoga je:

$$p_4 = 1 - \frac{1+4+8+12}{1+4+8+12+15+17+13+9+5+4+2+1} = \frac{66}{91} = 0.72527472527473.$$

c) Događaj "*broj usluženih klijenata u jednome satu djeljiv je sa* 3" je disjunktna unija sljedećih događaja: "*u jednome satu nema usluženih klijenata*", "*u jednome satu usluženo je točno troje klijenata*", "*u jednome satu usluženo je točno šest klijenata*" i "*u jednome satu usluženo je točno devet klijenata*". Označimo li traženu vjerojatnost sa p_3, vrijedi:

$$p_3 = \sum_{k=0}^{3} P(X = 3 \cdot k).$$

Izračunajmo najprije traženu vjerojatnost rabeći klasičnu definiciju vjerojatnosti. U tu ćemo svrhu ponovno uporabiti MATLAB i u običnoj *m*–datoteci **p3.m** izračunati traženu vjerojatnost. Otvorimo novu *m*–datoteku pa u nju utipkajmo:

```
p3=0;
lambda=4.93406593406593;
for i=0:3:9
    p3=p3+(lambda^i)*exp(-lambda)/factorial(i);
end
p3
```

Pohranimo upisane naredbe i vratimo se u komandni prozor. Utipkajmo:

```
p3
```

i dobit ćemo:

```
p3 =
   0.32989099176964
```

Sada istu vjerojatnost izračunajmo rabeći statističku definiciju vjerojatnosti. Iskoristimo podatak o ukupnome broju sati u kojima je vršeno promatranje dobiven u **b)** podzadatku (taj je broj jednak 91):

$$p_3 = \frac{1+12+13+4}{91} = \frac{30}{91} = 0.32967032967033.$$

9.2. Neprekidne (kontinuirane) slučajne varijable

Postoji više vrsta razdioba kontinuiranih slučajnih varijabli, ali najznačajnija i najprimjenjenija je *normalna* ili *Gaussova* slučajna varijabla. Toj varijabli odgovara istoimena razdioba koju opisujemo u nastavku.

9.2.1. Normalna ili Gaussova razdioba

Normalna ili Gaussova razdioba objašnjava najveći broj statističkih opažanja, kao što su npr. razdioba visina odraslih ljudi, razdioba rezultata mjerenja fizikalnih veličina itd. Kao svojevrsno objašnjenje normalne razdiobe ljudskih visina može poslužiti mnoštvo genetskih utjecaja i utjecaja okoliša na nečiju visinu. Neki utjecaji imaju jače djelovanje od drugih, dok neki djeluju udruženo, a ne nezavisno. Pokazuje se da ako je broj raznih čimbenika jako velik, Gaussova krivulja dobro opisuje takvu razdiobu.

Svaka neprekidna slučajna varijabla potpuno je određena svojom funkcijom gustoće. *Funkciju gustoće* normalne razdiobe obično zapisujemo u sljedećem obliku:

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} . \tag{7}$$

Pritom su realni brojevi μ (čitajte: mi) i σ^2 (čitajte: sigma na kvadrat) *parametri normalne* razdiobe. Te oznake nisu slučajne jer se može pokazati da je μ *očekivanje*, a σ^2 *varijanca* normalne razdiobe. Svaku slučajnu varijablu koja ima normalnu razdiobu nazivamo *normalna slučajna varijabla* i označavamo:

$$X \sim N(\mu, \sigma^2).$$

Posebno, za $\mu = 0$ i $\sigma^2 = 1$ dobivamo *standardnu* ili *jediničnu normalnu razdiobu*.

Budući da normalna slučajna varijabla pripada kontinuiranim slučajnim varijalama, njezina je slika uvijek neprebrojiv podskup skupa **R**. Tipični takvi podskupovi skupa **R** su otvoreni, poluotvoreni i zatvoreni intervali, pa se ova razdioba najčešće rabi kada su empirijski podatci grupirani u *razrede*. To je ujedno i kriterij prema kojemu ćete znati je li u nekom zadatku primjereno rabiti normalnu razdiobu ili nije.

Algoritam za određivanje parametara prilagođene normalne razdiobe je sljedeći:

Korak 1. Zadamo matrično donje granice razreda, gornje granice razreda, širine razreda (poželjno je, ali ne i nužno da svi razredi imaju istu širinu), razredne sredine i apsolutne frekvencije razreda.

Korak 2. Izračunamo ukupan broj podataka (N), aritmetičku sredinu (sv), varijancu (v) i standardno odstupanje (sd) iz grupiranih podataka.

Korak 3. Aritmetičku sredinu sv interpretiramo kao očekivanje μ, varijancu v kao varijancu σ^2, a standardno odstupanje sd kao standardno odstupanje σ normalne razdiobe.

Korak 4. Teorijske apsolutne frekvencije ft_k razreda $[a, b\rangle$ računamo prema formuli:

$$(f_t)_k = N \cdot \left[\Phi(b) - \Phi(a) \right], \tag{8}$$

gdje je

$$\Phi(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \int_{-\infty}^{x} e^{-\frac{(t-\mu)^2}{2 \cdot \sigma^2}} \cdot dt \tag{9}$$

funkcija razdiobe (distribucije) normalne razdiobe $N(\mu, \sigma^2)$. Ta funkcija računa vjerojatnost da neprekidna slučajna varijabla X raspodijeljena prema normalnoj razdiobi $N(\mu, \sigma^2)$ poprimi vrijednost ne veću od x. Dakle, po definiciji vrijedi:

$$\Phi(x) := P(X \leq x).$$

U prvi mah se možemo prilično zabrinuti jer ne znamo kako izračunati nepravi integral u definiciji funkcije $\Phi(x)$, čak ni kada x zamijenimo nekim „konkretnim" realnim brojem. No, nema potrebe za zabrinutošću. U davno doba (kad ljudi još nisu znali za MATLAB) vrijednosti funkcije $\Phi(x)$ standardne normalne razdiobe računale su se pomoću tablica nalik na logaritamske. U današnje doba za računanje vrijednosti funkcije razdiobe (i to za *bilo koju* normalnu razdiobu!) MATLAB ima već ugrađenu funkciju `normcdf`. Njezini su argumenti (redom) realan broj x, očekivanje μ i standardno odstupanje σ normalne razdiobe $N(\mu, \sigma^2)$. (Oprez: ne smijete zamijeniti redoslijed argumenata!). Funkcija `normcdf` vraća vrijednost $\Phi(x)$ izračunanu prema formuli (9).

Implementacija opisanoga algoritma u MATLAB-u je sljedeća:

Korak 1. Zadamo širinu razreda c, matricu donjih granica dg, matricu gornjih granica gg, matricu razrednih sredina s kao poluzbroj $\frac{1}{2}(dg + gg)$, te matricu apsolutnih frekvencija f.

Korak 2. Izračunamo ukupan zbroj svih apsolutnih frekvencija utipkavajući

```
N = sum(f);
```

Korak 3. U komandnomu prozoru utipkamo

```
grupirani(s,f)
```

i dobijemo vrijednosti `sv`, `d` i `sd`. (Vrijednost `kv` nam ne treba.)

Korak 4. Kreiramo jednorednu matricu teorijskih frekvencija `ft` utipkavanjem naredbe:

```
ft=round(N*(normcdf(gg,sv,sd)-normcdf(dg,sv,sd)))
```

Napomena: U Koraku 4. zapravo se „krije" formula (8), pri čemu se rezultat primjene te formule zaokružuje na najbliži cijeli broj primjenom funkcije `round`. Takvo je zaokruživanje potrebno jer su teorijske apsolutne frekvencije, prema definiciji apsolutne frekvencije, nenegativni cijeli brojevi, pa kao rezultat ne možemo dobiti decimalne brojeve.

Pogledajmo na primjerima kako odrediti parametre prilagođene normalne razdiobe.

***Primjer* 1.** Mjerenjem visina učenika srednje škole "Mirko S. Zlikovski" iz Donje Špičkovine ustanovljeno je da su one u rasponu od 160 cm do 200 cm. Podatci su grupirani u ukupno 10 razreda i prikazani tablično.

Visina [cm]	*Broj učenika*
160-165	2
165-170	15
170-175	26
175-180	45
180-185	42
185-190	30
190-195	17
195-200	3

Odredimo parametre prilagođene normalne razdiobe i izračunajmo pripadne teorijske frekvencije.

Zadatak ćemo riješiti opisanim algoritmom u MATLAB-u. U komandnomu prozoru utipkavamo:

```
dg=160:5:195;
gg=165:5:200;
s=(dg+gg)/2;
f=[2 15 26 45 42 30 17 3];
N=sum(f);
[sv,v,sd,kv]=grupirani(s,f);
ft=round(N*(normcdf(gg,sv,sd)-normcdf(dg,sv,sd)))
```

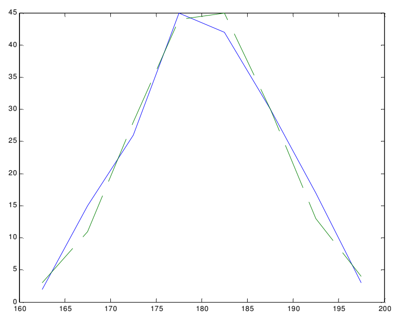
Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ft =
     3     11     28     44     45     30     13      4
```

Provjera pomoću funkcije *plot*

```
plot(s,f,s,ft,'--')
```

daje sljedeću sliku:



Slika 4.

***Primjer* 2.** Odredimo prilagođenu normalnu razdiobu za razdiobe masâ purana iz Primjera 1. u točki 8.3. Najmanja masa purana je 4.90 kg, a najveća 8.65 kg. U tom smo primjeru već izračunali da je širina svakoga razreda c = 0.75. Zato utipkavamo:

```
dg=4.90:0.75:(8.65-0.75);
gg=(4.90+0.75):0.75:8.65;
s=(dg+gg)/2;
f=[5 7 7 6 5];
N=sum(f);
[sv,v,sd,d]=grupirani(s,f);
ft=round(N*(normcdf(gg,sv,sd)-normcdf(dg,sv,sd)))
```

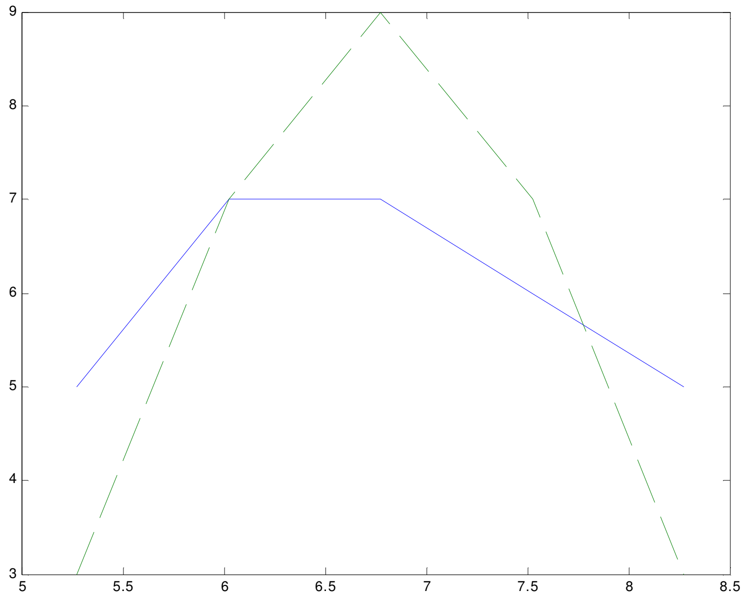
pa će MATLAB ispisati:

```
ft =
     3     7     9     7     3
```

Provjera pomoću naredbe *plot*

```
plot(s,f,s,ft,'--')
```

daje sljedeću sliku:



Slika 5.

Ovolika se nepreciznost pojavila zbog relativno maloga broja ukupnih podataka. Za veću točnost trebali bismo imati znatno više dobivenih podataka.

9.3. Zadatci za vježbu

1. Kreirajte funkcijsku *m*–datoteku **genbinomna.m** koja će sadržavati samo funkciju *genbinomna* čije su ulazne varijable prirodni brojevi N i n, te realan broj $p \in [0,1]$, a izlazna varijabla matrica apsolutnih frekvencija raspoređenih prema binomnoj razdiobi $B(n, p)$ čiji je zbroj jednak N. (Funkcija ne treba provjeravati jesu li N i n prirodni brojevi, te je li $p \in [0,1]$.)

2. Kreirajte funkcijsku *m*–datoteku **genpoissonova.m** koja će sadržavati samo funkciju *genpoissonova* čije su ulazne varijable nenegativan realan broj a i prirodni brojevi n i N, a izlazna varijabla matrica apsolutnih frekvencija modaliteta 0, 1, …, n raspoređenih prema Poissonovoj razdiobi $PO(a)$ čiji je zbroj jednak N. (Funkcija ne treba provjeravati je li $a \geq 0$ i jesu li $n, N \in \mathbf{N}$.)

3. Vršeći korekturu "*Velike dječje enciklopedije*" korektor je 70 puta slučajno izabrao uzorak od po 10 stranica te enciklopedije i bilježio ukupan broj otkrivenih tiskarskih grešaka. Dobiveni podatci prikazani su u sljedećoj tablici.

Broj grešaka	*Broj uzoraka*
0	6
1	15
2	18
3	15
4	9
5	4
6	2
7	1

a) Odredite i interpretirajte apsolutnu, relativnu, kumulativnu apsolutnu i kumulativnu relativnu frekvenciju modaliteta 5.
b) Izračunajte prosječan broj tiskarskih grešaka u jednom uzorku, te odgovarajući pokazatelj raspršenosti broja grešaka oko toga broja.
c) Izračunajte prosječan broj tiskarskih grešaka po jednoj stranici enciklopedije.
d) Odredite prilagođenu razdiobu koja najbolje opisuje dobivene podatke. Navedite njezine parametre i interpretirajte svaki od njih. Izračunajte pripadne teorijske frekvencije i interpretirajte jednu od njih.
e) Izračunajte vjerojatnost da će broj grešaka u jednom uzorku biti prost broj rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.
f) Izračunajte vjerojatnost da u jednom uzorku nema više od 4 greške rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.

4. U skladištu tvornice računala "*Macrohard*" nalazi se 100 000 gotovih čipova za računala. Želeći procijeniti postotak "škartova" među njima, kontrolor je ukupno 100 puta uzimao uzorke od po 20 komada čipova i bilježio ukupan broj neispravnih čipova po uzorku. Dobiveni podatci prikazani su u donjoj tablici.

Broj neispravnih čipova	*Broj uzoraka*
0	14
1	25
2	27
3	23
4	7
5	3
6	1

a) Odredite i interpretirajte apsolutnu, relativnu, kumulativnu apsolutnu i kumulativnu relativnu frekvenciju modaliteta 1.
b) Izračunajte prosječan broj neispravnih čipova u jednom uzorku, te odgovarajući pokazatelj raspršenosti frekvencija grešaka oko toga broja.
c) Odredite prilagođenu razdiobu koja najbolje opisuje dobivene podatke. Navedite njezine parametre i interpretirajte ih. Izračunajte i pripadne teorijske frekvencije, pa interpretirajte jednu od njih.
d) Izračunajte vjerojatnost da će slučajno odabrani uzorak sadržavati manje od 3 neispravna čipa rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.
e) Izračunajte vjerojatnost da će u slučajno odabranom uzorku biti više od 4 neispravna čipa rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.

5. Želeći utvrditi razloge katastrofalne igre svojih igrača trener nogometnoga kluba NK "*Stativa*" iz Muća odlučio je provjeriti njihovu fizičku spremnost mjereći vrijeme za koje će svaki od njih pretrčati 2 km. Dobiveni rezultati prikazani su u donjoj tablici.

Vrijeme (u min.)	*Broj igrača*
$[5.5, 5.7\rangle$	2
$[5.7, 5.9\rangle$	4
$[5.9, 6.1\rangle$	7
$[6.1, 6.3\rangle$	6
$[6.3, 6.5]$	3

a) Odredite i interpretirajte apsolutnu frekvenciju razreda $[5.5, 5.7\rangle$.
b) Izračunajte postotak igrača koji su 2 km uspjeli pretrčati za manje od 6 minuta i 6 sekundi.
c) Izračunajte prosječnu brzinu jednoga igrača NK "*Stativa*".
d) Odredite prilagođenu razdiobu koja najbolje opisuje dobivene podatke. Navedite njezine parametre i interpretirajte ih. Izračunajte i pripadne teorijske frekvencije, pa interpretirajte jednu od njih.
e) Izračunajte vjerojatnost da je slučajno odabrani igrač pretrčao zadanu udaljenost za točno

6 minuta rabeći klasičnu definiciju vjerojatnosti.

f) Izračunajte vjerojatnost da je slučajno odabrani igrač pretrčao zadanu udaljenost za barem 6 minuta i 6 sekundi.

6. U II. svjetskomu ratu London je bio podijeljen na ukupno 576 sektora koji su gađani bombama iz zrakoplova. Engleski vojni statističari bilježili su broj bombi koje su padale na svaki sektor i grupiranjem rezultata dobili su sljedeću tablicu.

Broj bombi po sektoru	*Broj sektora*
0	229
1	211
2	93
3	35
4	7
5	1

a) Odredite i interpretirajte apsolutnu, relativnu, kumulativnu apsolutnu i kumulativnu relativnu frekvenciju modaliteta 3.

b) Izračunajte postotak sektora na koje je pala barem jedna bomba.

c) Izračunajte prosječan broj bombi po sektoru, te odgovarajući pokazatelj raspršenosti frekvencija bombi oko toga broja.

d) Odredite prilagođenu razdiobu koja najbolje opisuje dobivene podatke. Navedite i interpretirajte sve parametre te razdiobe. Izračunajte i pripadne teorijske frekvencije, pa interpretirajte jednu od njih.

e) Izračunajte vjerojatnost da je slučajno odabrani sektor pogođen s najviše tri bombe rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.

f) Izračunajte vjerojatnost da broj bombi koje su pogodile slučajno odabrani sektor nije prost broj rabeći klasičnu, odnosno statističku definiciju vjerojatnosti.

7. Za normalnu se razdiobu definira i tzv. *funkcija pogrješke* (engl.: *error function*) izrazom

$$erf(x) = \frac{2}{\sqrt{\pi}} \cdot \int_0^x e^{-t^2} \cdot dt .$$

a) Pokažite da za sve realne brojeve a i b vrijedi jednakost:

$$\Phi(b) - \Phi(a) = \frac{1}{2} \cdot \left[erf\left(\frac{b}{\sqrt{2}}\right) - erf\left(\frac{a}{\sqrt{2}}\right) \right].$$

b) Vrijednosti funkcije *erf* u MATLAB-u ispisuje istoimena funkcija `erf`. Napišite Korak 4. algoritma kojim se u MATLAB-u implementira prilagodba normalne razdiobe empirijskim podatcima koristeći funkciju `erf` umjesto funkcije `normcdf`.

§10. OSNOVE NUMERIČKE MATEMATIKE

10.1. Numeričko rješavanje (ne)algebarskih jednadžbi s jednom nepoznanicom

U ovoj ćemo se točki pozabaviti problemom određivanja približnoga rješenja (ne)algebarske jednadžbe s jednom nepoznanicom. Formule po kojima se može *točno* odrediti rješenje neke algebarske jednadžbe postoje samo za jednadžbe 2., 3. i 4. stupnja – to su iz srednje škole poznata formula za računanje rješenja kvadratne jednadžbe, Cardanova formula (za algebarsku jednadžbu 3. stupnja), te Ferrarijeva formula (za algebarsku jednadžbu 4. stupnja). Abelov poučak[28] tvrdi da za algebarske jednadžbe stupnja barem 5 takve formule ne postoje. Stoga je najprije bilo u interesu naći algoritme za *približno* rješavanje takvih jednadžbi, a potom se problem poopćio i na znatno veću klasu jednadžbi, tzv. *nealgebarske jednadžbe*.

Općenito, nealgebarska jednadžba je svaka jednadžba oblika

$$f(x) = 0$$

gdje je $f(x)$ realna funkcija koja se ne može zapisati u obliku polinoma s realnim koeficijentima. Kako bi se i takve jednadžbe uspješno mogle (barem približno) riješiti, matematičari su problem rješavanja nelinearne jednadžbe sveli na problem proučavanja određenih svojstava funkcije $f(x)$ (neprekidnost, derivabilnost itd.). Razvitak *numeričke matematike* omogućio je pojavu brojnih metoda za određivanje približnih rješenja nealgebarske jednadžbe. Jedna od najčešće rabljenih jest *metoda raspolavljanja* ili *metoda bisekcije*.

Osnovna ideja te metode zasniva se na sljedećemu poučku:

***<u>Poučak</u>* 1. (*Bolzanov poučak*)** Neka je f realna funkcija neprekidna na segmentu $I = [a, b]$. Ako vrijedi nejednakost

$$f(a) \cdot f(b) < 0,$$

tada postoji barem jedan $c \in \langle a, b \rangle$ takav da je $f(c) = 0$.

Slobodno govoreći, Bolzanov poučak možemo interpretirati ovako: uzmemo li u pravokutnom koordinatnom sustavu u ravnini dvije točke s različitih strana osi apscisa (jednu ispod, a drugu iznad te osi) i spojimo ih jednom krivuljom ne dižući olovku s papira, onda ćemo pri tom spajanju barem jednom presjeći os apscisa.

Treba uočiti vrlo snažnu pretpostavku toga poučka, a to je neprekidnost funkcije f na segmentu I. To znači da Bolzanov poučak ne vrijedi za funkcije koje imaju (barem jedan) pre-

[28] Detaljnije o Abelovu poučku vidjeti npr. u [7], str. 161.

kid na tom segmentu. Također, može se pokazati da poučak ne vrijedi ako se segment zamijeni otvorenim ili poluotvorenim intervalom.

Na temelju Bolzanova poučka razvijen je sljedeći algoritam za približno određivanje nultočke *bilo koje neprekidne* realne funkcije (na nekom segmentu) *s točnošću* ε:

Korak 1. Odrediti realne brojeve a i b takve da vrijedi:

$$f(a) \cdot f(b) < 0.$$

Korak 2. Izračunati $c = \frac{1}{2}(a+b)$ i $f(c)$. Potonji broj sigurno postoji jer je $c \in \langle a, b \rangle$, a f je definirana u svakoj točki segmenta $[a, b]$.

Korak 3. Ako je $f(c) = 0$, traženo rješenje je jednako c. Kraj postupka.

Korak 4. Ako je $b - a \leq \varepsilon$, traženo približno rješenje je jednako c. Kraj postupka.

Korak 5. Ako je $f(a) \cdot f(c) > 0$, staviti $a := c$ i vratiti se na Korak 2. Ako je $f(b) \cdot f(c) > 0$, staviti $b := c$ i vratiti se na Korak 2.

Upravo opisani algoritam naziva se *metoda raspolavljanja* ili *bisekcije*. Može se pokazati da uz zadane brojeve a, b i ε broj raspolavljanja n mora zadovoljavati nejednakost

$$n \geq \frac{\ln(b-a) - \ln \varepsilon}{\ln 2}. \tag{1}$$

Kako bismo metodu raspolavljanja mogli rabiti u MATLAB-u, kreirajmo sljedeću funkcijsku m–datoteku **bis.m** čije su ulazne varijable funkcija *funkcija*, donja granica segmenta a, gornja granica segmenta b i željeni broj raspolavljanja n:

```
function y=bis(funkcija,a,b,n)
if feval(funkcija,a)*feval(funkcija,b)>0
    error('Pogrješka u ulaznim podatcima!')
end
for k=1:n
       x=(a+b)/2;
       if abs(feval(funkcija,x))==0
           break
       end
       if feval(funkcija,x)*feval(funkcija,a)>0
           a=x;
          else
              b=x;
          end
end
y=x;
```

Napomene: 1.) U strukturi funkcije *bis* pojavljuje se ugrađena MATLAB-ova funkcija `feval`. Ta se funkcija najčešće koristi za izračun vrijednosti funkcije definirane izvan dotične datoteke. U ovom slučaju, koristimo je za izračun vrijednosti funkcije *funkcija* koju smo pohranili u drugoj *m*–datoteci.

2.) Za određivanje nultočaka neke funkcije MATLAB posjeduje ugrađenu funkciju *fzero* koju ćemo upoznati malo kasnije.

Pohranimo tako dobivenu datoteku i vratimo se u MATLAB-ov komandni prozor. Pogledajmo primjenu funkcije *bis* na primjerima.

Primjer 1. Odredimo približnu vrijednost realnoga rješenja jednadžbe

$$\sin x + \ln x = 0$$

s točnošću $\varepsilon = 0.0001$, pa odredimo najmanji broj raspolavljanja potrebnih da bi se postigla ta točnost.

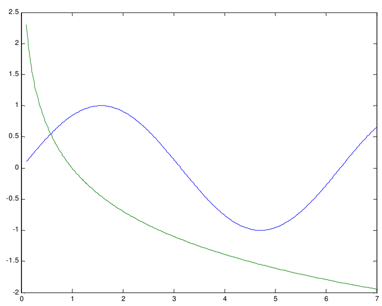
Da bismo mogli primijeniti metodu raspolavljanja, najprije moramo odrediti donju i gornju granicu segmenta u kojem se nalazi (bar jedno) rješenje zadane jednadžbe. U tu svrhu, zapišimo jednadžbu u obliku

$$\sin x = -\ln x$$

pa je riješimo grafički. Na istoj slici nacrtajmo grafove funkcija $f(x) = \sin x$ i $g(x) = -\ln x$. Budući da je funkcija $g(x)$ definirana samo za strogo pozitivne realne brojeve, nema smisla crtati graf funkcije $f(x)$ za $x \leq 0$. Stoga nacrtajmo grafove tih funkcija npr. na intervalu [0.1, 7]. U komandnom prozoru utipkajmo redom:

```
x=0.1:0.01:7;
y1=sin(x);
y2=-log(x);
plot(x,y1,x,y2)
```

pa ćemo dobiti sljedeću sliku:



Slika 1.

(Plava krivulja je graf funkcije $f(x)$, a zelena funkcije $g(x)$.) Odmah uočavamo da se tražena nultočka nalazi u segmentu [0.1, 1], pa ćemo uzeti $a = 0.1$, $b = 1$. Izračunajmo sada najmanji broj raspolavljanja potrebnih za postizanje točnosti $\varepsilon = 0.0001$ koristeći nejednakost (1). U novi redak komandnoga prozora utipkajmo:

```
n=ceil((log(1-0.1)-log(0.0001))/log(2))
```

pa će MATLAB ispisati:

```
n =
    14
```

Otvorimo novu *m*–datoteku pa utipkajmo:

```
function y=f(x);
y=sin(x)+log(x);
```

Pohranimo unesene naredbe pod imenom **f.m** i vratimo se u komandni prozor. Kad iz komandnoga prozora pozivamo neku funkciju kao argument druge funkcije, to činimo pomoću znaka @. U ovom slučaju, argument funkcije *bis* bit će funkcija *f* pohranjena u datoteci **f.m**, donja granica $a = 0.1$, gornja granica $b = 1$ i broj raspolavljanja $n = 14$. Stoga ćemo funkciju *bis* iz komandnoga prozora pozvati na sljedeći način:

```
bis(@f,0.1,1,14)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
   0.57872924804687
```

Možemo zaključiti da je $x = 0.57873$ približno rješenje zadane jednadžbe (s točnošću ε). Dobiveni rezultat možemo provjeriti tako da u novi redak komandnoga prozora utipkamo:

```
f(ans)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
    4.002732345431692e-005
```

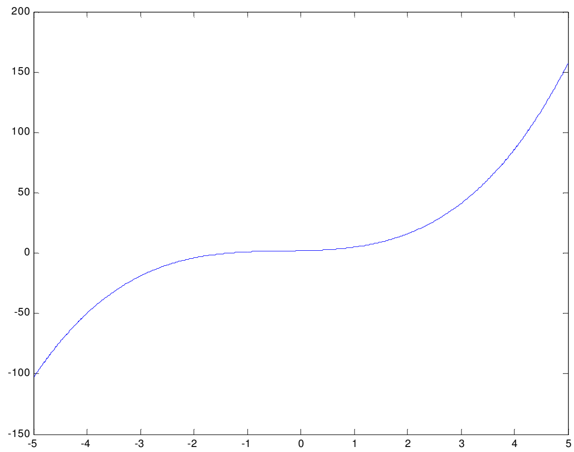
Dakle, $p(0.57872924804687) \approx 4 \cdot 10^{-5}$, pa smo dobili rješenje sa željenom točnošću.

Primjer 2. Zadan je polinom $p(x) = x^3 + x^2 + x + 2$. Odredimo sve njegove realne nultočke s točnošću $\varepsilon = 0.0001$.

Nacrtajmo najprije graf polinoma $p(x)$. Nije teško uočiti da je npr. $p(-5) < 0$, a da je $p(5) > 0$. Stoga graf polinoma p crtamo na segmentu [–5, 5]. Utipkajmo:

```
x=-5:0.01:5;
y=x.^3+x.^2+x+2;
plot(x,y)
```

pa ćemo dobiti sljedeću sliku:



Slika 2.

Uočavamo da je zadani polinom strogo rastući. On je neparnoga stupnja, pa prema osnovnom poučku algebre[29] mora imati barem jednu realnu nultočku. Zbog svojstva strogoga rasta zaključujemo da p ima točno jednu realnu nultočku. Budući da sa slike ne možemo precizno odrediti u kojemu se segmentu "užem" od segmenta [–5, 5] nalazi ta nultočka, uzet ćemo $a =$ $= -5$ i $b = 5$.

Izračunajmo najmanji broj raspolavljanja potrebnih da bi se postigla željena točnost. Zatvorimo dobivenu sliku, vratimo se u komandni prozor i u njegov novi redak utipkajmo:

```
n=ceil((log(5+5)-log(0.0001))/log(2))
```

MATLAB će ispisati:

```
n =
    17
```

Ponovno otvorimo *m*–datoteku **f.m** pa njezin sadržaj preoblikujmo u:

```
function y=f(x);
y=x.^3+x.^2+x+2;
```

Pohranimo unesene naredbe, vratimo se u komandni prozor pa u njegov novi redak utipkaj-

[29] Više o osnovnu poučku algebre vidjeti npr. u [7], str. 160-161.

mo:

```
bis(@f,-5,5,17)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
  -1.35322570800781
```

i to je jedina realna nultočka zadanoga polinoma. Provjerimo dobiveni rezultat utipkavajući:

```
f(ans)
```

u novi redak komandnoga prozora. Pritisnimo *Enter*, pa će MATLAB ispisati:

```
ans =
   -5.962431960782055e-005
```

Mada je algoritam metode raspolavljanja u suštini vrlo jednostavan, vidimo da ga je praktično relativno složeno primijeniti. Osnovni je problem, dakako, u određivanju granica segmenta u kojemu tražimo nultočku. Nameće se pitanje postoji li neka MATLAB-ova ugrađena funkcija čiji je ulazni argument jedino realna funkcija $f(x)$, a izlazni argumenti sve realne nultočke te funkcije. Nažalost, u općem je slučaju odgovor na to pitanje niječan. No, ako je $f(x)$ polinom stupnja n, $n \in \mathbf{N}$, onda je odgovor potvrdan. U nastavku ćemo razmotriti i taj problem.

Pozivajući funkciju *bis* koja određuje eventualnu nultočku unutar nekoga intervala morali smo navesti ukupno 4 ulazne varijable: funkciju, donju i gornju granicu intervala, te broj raspolavljanja. Stvari se donekle mogu pojednostavniti ako se metoda raspolavljanja kombinira s još nekim metodama numeričke matematike (metoda sekante i metoda inverzne kvadratične interpolacije). Na osnovi te kombinacije napravljen je algoritam za funkciju *fzero*. Ulazne varijable te funkcije su realna funkcija čije nultočke želimo odrediti i početna aproksimacija nultočke x_0. Drugim riječima, prije poziva te funkcije moramo odrediti neki realan broj "blizu" kojega se nalazi tražena nultočka. (Matematička formalizacija svojstva "biti blizu" zahtijeva poznavanje teorije mjere, pa u to ovdje nećemo ulaziti.) U tu svrhu ponovno moramo crtati graf funkcije i pomoću njega odrediti traženu aproksimaciju.

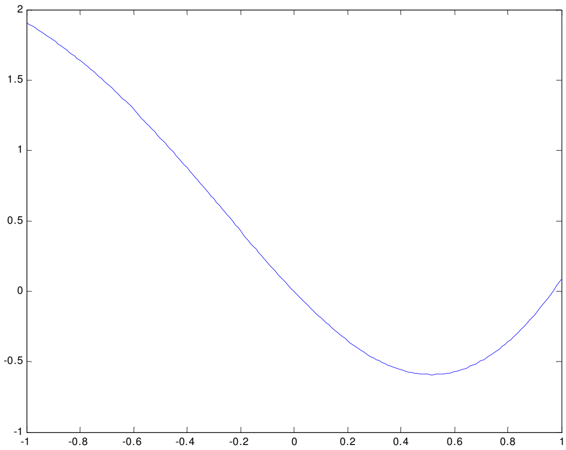
Ilustrirajmo primjenu funkcije *fzero* na primjerima.

Primjer 3. Zadana je funkcija $f(x) = x^2 - \sin(2 \cdot x)$. Odredimo sve realne nultočke te funkcije.

Kako smo rekli, najprije moramo nacrtati graf promatrane funkcije. Funkcija sinus može poprimiti jedino vrijednosti iz segmenta $[-1,1]$. Stoga i vrijednost x^2 mora pripadati tom segmentu, što znači da trebamo uzeti $x \in [-1, 1]$. Drugim riječima, ako nultočka uopće postoji, sigurno se nalazi u tom segmentu. Uobičajenom primjenom funkcije `clc` „počistimo“ komandni prozor, pa u njegova nova tri retka utipkajmo:

```
x=-1:0.01:1;
y=x.^2-sin(2*x);
plot(x,y)
```

Izvršenjem funkcije `plot` dobivamo sljedeći graf:



Slika 3.

Dakle, vidimo da imamo ukupno dvije nultočke: jedna se nalazi „blizu“ 0, a druga „blizu“ 1. Sad otvorimo *m*–datoteku **f.m** pa utipkamo:

```
function y=f(x);
y=x^2-sin(2*x);
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. Funkciju *fzero* pozivamo prema sljedećoj sintaksi:

```
fzero(@ime_funkcije, početna_aproksimacija)
```

Napomenimo još jednom da prigodom navođenja imena funkcije kao argumenta druge funkcije stavljamo znak @. U novomu retku komandnoga prozora utipkajmo:

```
fzero(@f,0)
```

pa će MATLAB ispisati

```
ans =
      0
```

To smo donekle mogli i očekivati jer je uistinu $f(0) = 0$. Odredimo i preostalu nultočku – onu koja se nalazi "blizu" 1. U novomu retku komandnoga prozora utipkajmo:

```
fzero(@f,1)
```

pa će MATLAB ispisati:

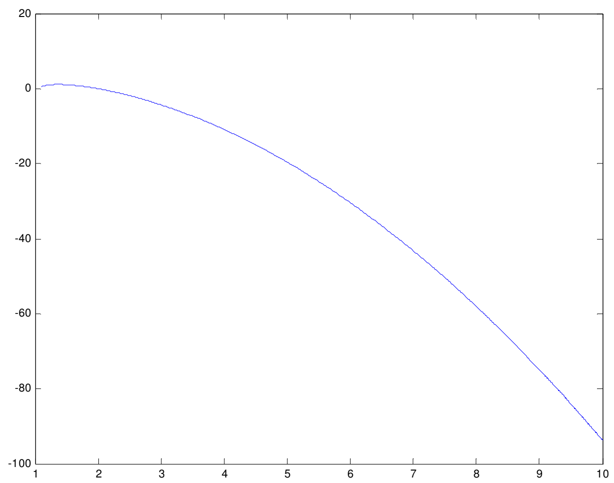
```
ans =
   0.96687688141351
```

***Primjer* 4.** Zadana je funkcija $f(x) = \ln(x - 1) - x^2 + 4$. Odredimo sve realne nultočke te funkcije.

Kao i u prethodnom primjeru, najprije nacrtajmo graf promatrane funkcije. Budući da je logaritamska funkcija $g(x) = \ln(x - 1)$ definirana za $x \in \langle 1, +\infty \rangle$, njezin ćemo graf nacrtati na segmentu [1.1, 10]. U novi redak komandnoga prozora utipkajmo:

```
x=1.1:0.01:10;
y=log(x-1)-x.^2+4;
plot(x,y)
```

pa ćemo dobiti sljedeći graf:

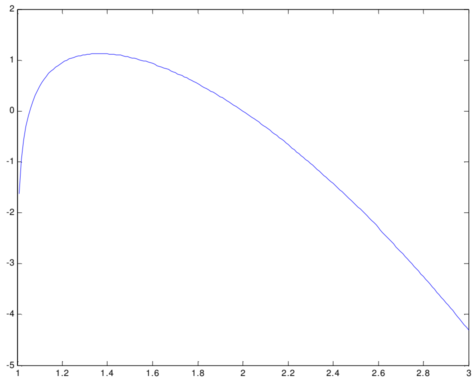


Slika 4.

Grubo govoreći, funkcija f je "uglavnom" padajuća (pokažite to "klasičnim" načinom rabeći prvu derivaciju!). Za određivanje njezine nultočke zanima nas njezino ponašanje na segmentu [1.1, 3]. U novi redak komandnoga prozora utipkajmo:

```
x=1.01:0.01:3;
y=log(x-1)-x.^2+4;
plot(x,y)
```

pa ćemo dobiti sljedeći graf:



Slika 5.

Vidimo da funkcija *f* ima dvije realne nultočke: prvu "blizu" 1.2 i drugu "blizu" 2. Otvorimo *m*–datoteku **f.m** i utipkajmo:

```
function y=f(x);
y=log(x-1)-x^2+4;
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. U njegov novi redak utipkajmo:

```
fzero(@f,1.2)
```

pa će MATLAB ispisati:

```
ans =
   1.05584398367821
```

Nakon toga, utipkajmo:

```
fzero(@f,2)
```

pa će MATLAB ispisati:

```
ans =
     2
```

i to su jedine dvije realne nultočke promatrane funkcije.

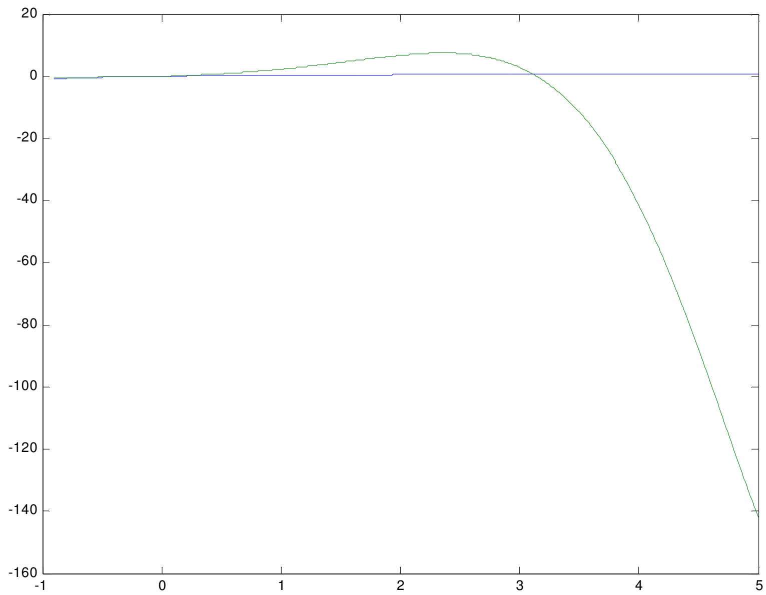
Napomena: Funkcija *fzero* dozvoljava da se umjesto početne aproksimacije unese i segment u kojemu se nalazi točno jedna nultočka funkcije $f(x)$. Takve primjere ovdje nećemo razmatrati jer se oni suštinski nimalo ne razlikuju od Primjera 1. i 2. koje smo rješavali metodom raspolavljanja.

Primjer 5. Zadane su funkcije $f(x) = \log(x+1)$ i $g(x) = e^x \cdot \sin x$. Odredimo sve zajedničke točke njihovih grafova na segmentu [–0.9, 5].

Ovdje nam zadani segment služi jedino kao putokaz na kojemu segmentu trebamo crtati grafove zadanih funkcija. Naime, ako se umjesto početne aproksimacije kao ulazna varijabla funkcije *fzero* navede segment, *fzero* će pretpostaviti da se u tom segmentu nalazi najviše jedna nultočka promatrane funkcije, što općenito ne mora biti točno. Zato moramo najprije nacrtati grafove zadanih funkcija na zadanom segmentu, a potom pomoću dobivenoga grafa odrediti sve eventualne početne aproksimacije nultočaka. U novomu retku komandnoga prozora utipkajmo:

```
x=-0.9:0.01:5;
y1=log10(x+1);
y2=exp(x).*sin(x);
plot(x,y1,x,y2)
```

pa ćemo dobiti sljedeću sliku:

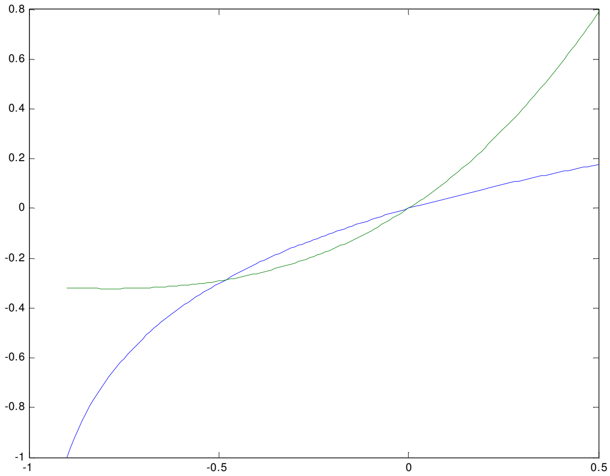


Slika 6.

Vidimo da se jedna zajednička točka nalazi "blizu" 3, ali slika ne daje precizan odgovor ima li zajedničkih točaka u segmentu [-0.9, 0.5]. Zato utipkajmo:

```
x=-0.9:0.01:0.5;
y1=log10(x+1);
y2=exp(x).*sin(x);
plot(x,y1,x,y2)
```

pa ćemo dobiti sljedeću sliku:



Slika 7.

Dakle, postoje ukupno 3 zajedničke točke promatranih funkcija. Jedna se nalazi "blizu" –0.5, druga "blizu" 0, a treća blizu 3. Za njihovo točno određivanje koristit ćemo funkciju *fzero*. Budući da ta funkcija ne rješava jednadžbe tipa $f(x) = g(x)$, već samo jednadžbe tipa $f(x) = 0$, zapišimo jednakost $f(x) = g(x)$ u obliku

$$f(x) - g(x) = 0.$$

Stavimo li

$$h(x) = f(x) - g(x),$$

onda vidimo da se zadani problem svodi na određivanje nultočaka funkcije $h(x)$. U komandnomu prozoru otvorimo *m*–datoteku **h.m** i utipkajmo:

```
function y=h(x);
y=log10(x+1)-exp(x)*sin(x);
```

Pohranimo upisane naredbe i vratimo se u komandni prozor. Odredimo najprije prvu nultočku funkcije $h(x)$. Utipkajmo:

```
fzero(@h,-0.5)
```

pa će MATLAB ispisati:

```
ans =
   -0.48302799677525
```

Potpuno analogno utipkavanjem

```
fzero(@f,0)
```

dobijemo:

```
ans =
     0
```

a utipkavanjem

```
fzero(@f,3)
```

dobijemo:

```
ans =
   3.11430884120450
```

Dakle, (približne) apscise traženih točaka su $x_1 = -0.483028$, $x_2 = 0$ i $x_3 = 3.114309$. Pripadne ordinate dobijemo izračunom vrijednosti ili funkcije *f* ili funkcije *g* za svaku pojedinu apscisu. Lako se dobiva (provjerite!) da su tražene točke $T_1 \approx (-0.483028, -0.28653)$, $T_2 = (0, 0)$ i $T_3 \approx$ $\approx (3.114309, 0.6143)$.

Razmotrimo sada poseban slučaj kad je funkcija $f(x)$ polinom stupnja n. Osnovni poučak algebre tvrdi da svaki polinom (s realnim ili kompleksnim koeficijentima) stupnja barem 1 ima barem jednu nultočku u skupu kompleksnih brojeva $\mathbf{C}$. Ako na skupu $\mathbf{C}^n[x]$ svih polinoma stupnja točno n čiji su koeficijenti kompleksni brojevi definiramo preslikavanje $k : \mathbf{C}^n[x] \to \mathbf{C}^{n+1}$ propisom

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0 \ \longmapsto (a_n, a_{n-1}, \ldots, a_2, a_1, a_0)$$

onda se lako može provjeriti da je preslikavanje k bijekcija. To znači da polinom možemo smatrati zadanim ako znamo sve njegove koeficijente. Zbog toga ima smisla definirati posebnu funkciju čije će ulazne varijable biti svi koeficijenti polinoma, a izlazne varijable sve (općenito, kompleksne) nultočke toga polinoma. U MATLABU postoji upravo takva ugrađena funkcija: to je funkcija *roots*. Njezina je sintaksa

```
roots(matrica_koeficijenata_polinoma)
```

Pogledajmo uporabu te funkcije na primjerima.

Primjer 6. Odredimo sve nultočke polinoma $p(x) = x^3 + x^2 + x + 1$. U novomu retku komandnoga prozora najprije zadajmo naš polinom pomoću jednoretčane matrice p čiji su elementi točno svi koeficijenti polinoma p. Utipkajmo:

```
p=[1 1 1 1];
```

Sada možemo pozvati funkciju *roots*:

```
roots(p)
```

pa će MATLAB ispisati:

```
ans =
 -1.00000000000000
 -0.00000000000000 + 1.00000000000000i
 -0.00000000000000 - 1.00000000000000i
```

Zaključujemo da su sve kompleksne nultočke zadanoga polinoma –1, i i $-i$. (Uočite da je predznak realnoga dijela čisto imaginarnih brojeva i i $-i$ negativan. To znači da ti realni dijelovi nisu identički jednaki 0, ali pogrješka aproksimacije je toliko malena da je praktički smijemo zanemariti.)

Primjer 7. Odredimo sve nultočke polinoma $p(x) = x^3 - x^2 - x + 1$. Ponovno najprije moramo zadati matricu p svih koeficijenata polinoma:

```
p=[1 -1 -1 1];
```

Pozivom funkcije *roots*

```
roots(p)
```

MATLAB će ispisati:

```
ans =
 -1.00000000000000
  1.00000000000000 + 0.00000000938807i
  1.00000000000000 - 0.00000000938807i
```

No, rastav zadanoga polinoma na faktore daje:

$$p(x) = x^3 - x^2 - x + 1 = x^2(x - 1) - (x - 1) = (x - 1) \cdot (x^2 - 1) = (x - 1) \cdot (x - 1) \cdot (x + 1) = \\ = (x-1)^2 \cdot (x+1)$$

pa vidimo da zadani polinom ima ukupno tri realne nultočke: jednu dvostruku (to je 1) i jednu jednostruku (to je –1). Zbog čega nam je onda primjena funkcije *roots* dala ovako čudne rezultate? Razlog je tolerancija računanja numeričke metode. Naime, MATLAB ne računa nultočke zadanoga polinoma rastavljanjem na faktore, nego primijenjuje neke metode numeričke matematike za približno određivanje nultočaka polinoma (u detalje ovdje ne ulazimo). Uporaba tih metoda nužno povlači pojavu pogrješaka aproksimacije nultočaka. Stoga ćemo imaginarne dijelove gornjih dvaju kompleksnih brojeva shvatiti kao aproksimacije reda veličine 10^{-9}, pa ih praktički možemo smatrati jednakim nuli, odnosno zanemariti.[30]

[30] U praksi je zanemarivanje „malih“ brojeva u ovakvim slučajevima nerijetko vrlo „nezgodno“ jer je teško razlikovati radi li se doista o „malom“ broju ili o pogrješci metode.

Želimo li izračunati vrijednost nekoga polinoma za „konkretnu“ vrijednost nezavisne varijable, osim ranije spomenute funkcije *feval* možemo koristiti funkciju *polyval*. Osnovna prednost funkcije *polyval* jest što kao prvi argument ima matricu koeficijenata polinoma, a ne cijeli polinom (kao funkcija *feval*). Njezina je sintaksa:

```
polyval(matrica_koeficijenata_polinoma, konkretna_vrijednost)
```

Ova je funkcija vrlo pogodna prigodom grafičkoga prikazivanja polinoma. Pogledajmo to na primjeru.

Primjer 7. Zadan je polinom $p(x) = x^3 - x^2 + x - 1$. Shvaćajući x kao kompleksnu varijablu, odredimo sve njegove nultočke i izračunajmo $p(1 - i) + p(1 + i)$. Potom shvaćajući x kao realnu varijablu[31] prikažimo grafički taj polinom na segmentu $[-3, 3]$.

Najprije zadajmo polinom definiranjem njegove matrice koeficijenata:

```
p=[1 -1 1 -1];
```

Njegove nultočke odredimo utipkavajući:

```
roots(p)
```

u novi redak komandnoga prozora. Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
  1.00000000000000
  0.00000000000000 + 1.00000000000000i
  0.00000000000000 - 1.00000000000000i
```

Dakle, nultočke zadanoga polinoma su 1, i i $-i$. Izračunajmo sada $p(1 - i) + p(1 + i)$. U novi redak komandnoga prozora utipkajmo:

```
polyval(p,1+i)+polyval(p,1-i)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

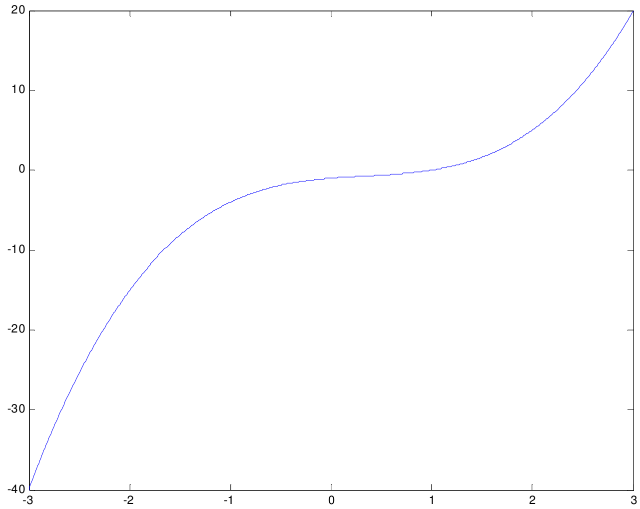
```
ans =
    -4
```

Napokon, nacrtajmo graf zadanoga polinoma iznad segmenta $[-3, 3]$. U nova dva retka komandnoga prozora utipkajmo:

```
x=-3:0.01:3;
plot(x,polyval(p,x))
```

[31] Grafove kompleksnih funkcija kompleksne varijable nije moguće crtati u MATLAB-u. Naime, svakoj točki (x, y) takvoga grafa bijektivno možemo pridružiti uređenu četvorku (Re x, Im x, Re y, Im y), pa bi graf morao biti „četverodimenzionalan“, a to je nemoguće praktično izvesti.

Dobit ćemo sljedeći graf:

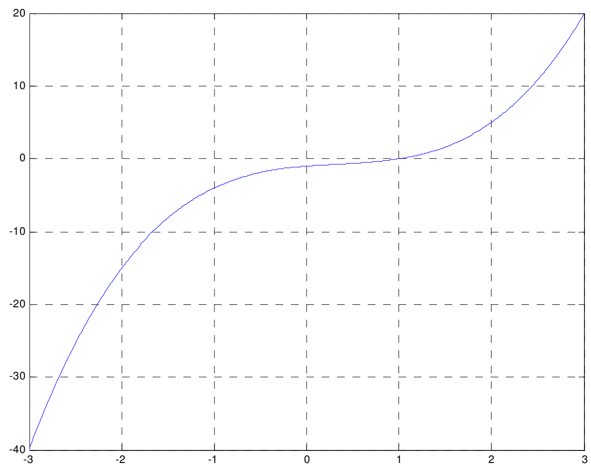


Slika 8.

Kako bismo točnije vidjeli određene „konkretne“ vrijednosti, utipkajmo još i

```
grid on
```

pa ćemo dobiti sljedeću sliku:



Slika 9.

Crtkane linije isključujemo utipkavanjem

```
grid off
```

10.2. Zadatci za vježbu

1. Bez korištenja funkcije *roots* pokažite da funkcija f ima (barem jednu) realnu nultočku u segmentu $[a, b]$, pa odredite tu nultočku s točnošću od 10^{-5} ako je:

a) $f(x) = \text{arsh}\, x$, $a = -1$, $b = 1$;
b) $f(x) = \ln(x - 1) - \text{tg}\, x$, $a = 3$, $b = 4$;
c) $f(x) = e^x + \text{ctg}\, x$, $a = -2$, $b = -1$;
d) $f(x) = \arcsin x - e^{-x}$, $a = 0$, $b = 1$.

2. Metodom raspolavljanja riješite sljedeće jednadžbe s točnošću $\varepsilon = 0.0001$:

a) $\sin x + \cos x = 0$ na segmentu $[0, 2 \cdot \pi]$;
b) $\cos x + \ln x = 0$;
c) $e^x - \sin(x + 1) = 0$;
d) $x^5 - x^4 + x^3 - x^2 + x - 1 = 0$.

U svakom od slučajeva odredite broj raspolavljanja potreban za postizanje zadane točnosti.

2. Odredite sve zajedničke točke grafova funkcija $f(x)$ i $g(x)$ ako je:

a) $f(x) = \dfrac{\ln(1-x)}{x}$, $g(x) = \dfrac{e^{x-1} - 1}{x+1}$;

b) $f(x) = \dfrac{\log(3 \cdot x + 1) - 1}{\sqrt{x+1}}$, $g(x) = \dfrac{\sin\left(\dfrac{\pi \cdot x}{3}\right)}{\sqrt{x+2}}$;

c) $f(x) = \text{tg}\, x$, $g(x) = 4 \cdot x - 1$ (na intervalu $\left\langle -\dfrac{\pi}{2}, \dfrac{\pi}{2} \right\rangle$);

d) $f(x) = \text{ctg}\, x$, $g(x) = -5 \cdot x + 7$ (na intervalu $\langle 0, \pi \rangle$).

3. Odredite sve realne nultočke (ako postoje) sljedećih funkcija:

a) $f(x) = x^3 - \sin(3 \cdot x)$;
b) $f(x) = x^2 - \cos(2 \cdot x)$;
c) $f(x) = x \cdot \text{th}(x) - 1$;
d) $f(x) = 3^x - 3 \cdot x$.

4. Odredite sve (realne i kompleksne) nultočke sljedećih polinoma:

a) $p(x) = x^4 - 10 \cdot x^3 + 35 \cdot x^2 - 50 \cdot x + 24$;
b) $p(x) = x^3 - 2 \cdot x^2 + 2 \cdot x - 1$;

c) $p(x) = x^5 - x^3 - 3 \cdot x - 18$;
d) $p(x) = x^3 - 6 \cdot x^2 + 9 \cdot x$.

Gdje god je to moguće, provjerite dobivene rezultate grafički.

5. Zadan je polinom $p(x) = x^3 + x^2 - x - 1$.

a) Odredite sve realne nultočke polinoma p i prikažite ga grafički na segmentu $[-3, 3]$.
b) Shvaćajući zadani polinom kao kompleksnu funkciju kompleksne varijable izračunajte $p(1 + i) + p(1 - i)$

6. Zadan je polinom $p(x) = x^5 - 2 \cdot x^4 + 3 \cdot x^2 - 5 \cdot x + 3$.

a) Odredite sve realne nultočke polinoma p i prikažite ga grafički na segmentu $[-1, 1]$.
b) Shvaćajući zadani polinom kao kompleksnu funkciju kompleksne varijable izračunajte $p(2 \cdot i) - p(1 - 2 \cdot i)$.

7. Kreirajte funkcijsku *m*–datoteku **polinom3.m** koja sadrži jedino funkciju *polinom*3 čije su ulazne varijable (ne nužno različiti) realni brojevi a, b i c, a jedina izlazna varijabla matrica p čiji su elementi točno svi koeficijenti polinoma $p(x)$ kojemu su točno sve nultočke jednake a, b i c.

8. Kreirajte funkcijsku *m*–datoteku **samorealni.m** koja sadrži jedino funkciju *samorealni* čija je jedina ulazna varijabla matrica p čiji su elementi točno svi koeficijenti polinoma $p(x)$, a izlazna varijabla jednaka 1 ako su sve nultočke polinoma $p(x)$ realne, a 0 inače.

9. Kreirajte funkcijsku *m*–datoteku **zopol.m** koja sadrži jedino funkciju *zopol* čije su ulazne varijable matrice p i q čiji su elementi točno svi koeficijenti redom polinoma $p(x)$ i $q(x)$, a izlazne varijable matrice Z i O čiji su elementi točno svi koeficijenti redom polinoma $p(x) + q(x)$ i $p(x) - q(x)$.

10. Kreirajte funkcijsku *m*–datoteku **umnozakpol.m** koja sadrži jedino funkciju *umnozakpol* čije su ulazne varijable matrice p i q čiji su elementi točno svi koeficijenti redom polinoma $p(x)$ i $q(x)$, a jedina izlazna varijabla matrica U čiji su elementi točno svi koeficijenti polinoma $p(x) \cdot q(x)$.

11. MATLAB-ova ugrađena funkcija *polyvalm* izračunava vrijednosti matričnoga polinoma p za konkretne vrijednosti kvadratne matrice x.

a) Proučite sintaksu navedene funkcije.
b) Neka je $p : M_3(\mathbf{R}) \to M_3(\mathbf{R})$ matrični polinom definiran propisom $p(X) = X^3 - 2 \cdot X^2 + 4 \cdot X$. Koristeći funkciju *polyvalm* izračunajte $p\left(\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}\right)$.

TEHNIČKO VELEUČILIŠTE U ZAGREBU
POLYTECHNICUM ZAGRABIENSE
ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

10.3. Numerička interpolacija

U ovoj točki promatramo sljedeći problem:

***Problem*:** Zadano je ukupno n točaka: $T_1 = (x_1, y_1)$, $T_2 = (x_2, y_2)$, …, $T_n = (x_n, y_n)$, pri čemu je $x_1 \neq x_2 \neq \ldots \neq x_n$. Odrediti polinom $p(x)$ (s realnim koeficijentima) što manjega stupnja takav da njegov graf prolazi svim zadanim točkama.

Može se pokazati da ako je zadano ukupno n točaka u ravnini među čijim apscisama nema međusobno jednakih, onda postoji jedinstveni polinom $L(x)$ s realnim koeficijentima čiji je stupanj najviše jednak $n - 1$ takav da njegov graf prolazi svim zadanim točkama. Taj je polinom $L(x)$ jedinstveno rješenje promatranoga problema. Nazivamo ga *Lagrangeov interpolacijski polinom* (skraćeno: *LIP*).

Analitičko određivanje *LIP*–a pomoću odgovarajućih formula je relativno složeno i mukotrpno već za "male" n (npr. $n = 4, 5, 6$), a za "velike" n ono je praktično nemoguće. Zbog toga ćemo pri određivanju *LIP*–a rabiti MATLAB. On posjeduje funkciju *polyfit* čije su ulazne varijable jednoredne matrice x i y koje sadrže redom točno sve apscise, odnosno ordinate zadanih točaka, te prirodan broj k, a jedina izlazna varijabla matrica L čiji su elementi točno svi koeficijenti polinoma stupnja *najviše* k (dakle, jednakoga ili manjega od k) koji najbolje opisuje zadani skup točaka u smislu *metode najmanjih kvadrata* (o tome ćemo govoriti nešto kasnije). Kako bismo kao rezultat dobili *LIP*, uzimat ćemo $k = n - 1$.

Ilustrirajmo primjenu te funkcije na primjeru.

***Primjer* 1.** Odredimo *LIP* čiji graf prolazi točkama $T_1 = (-1, 1)$, $T_2 = (2, -2)$, $T_3 = (-3, -1.5)$, $T_4 = (-4, 2.5)$ i $T_5 = (5, 4)$, te ga prikažimo grafički na segmentu $[-6, 6]$.

Najprije moramo zadati matrice x i y. U novi redak komandnoga prozora utipkajmo:

```
x=[-1 2 -3 -4 5];
y=[1 -2 -1.5 2.5 4];
```

Sada odredimo matricu L utipkavanjem:

```
L=polyfit(x,y,4)
```

Vrijednost 4 upisali smo zato što za $n = 5$ točaka imamo $k = n - 1 = 4$. Pritisnimo *Enter*, pa će MATLAB ispisati:

```
L =
  Columns 1 through 4
  0.05393518518519  -0.04305555555556  -1.02152777777778  -0.11898148148148
  Column 5
   1.80555555555558
```

Dakle, traženi je polinom

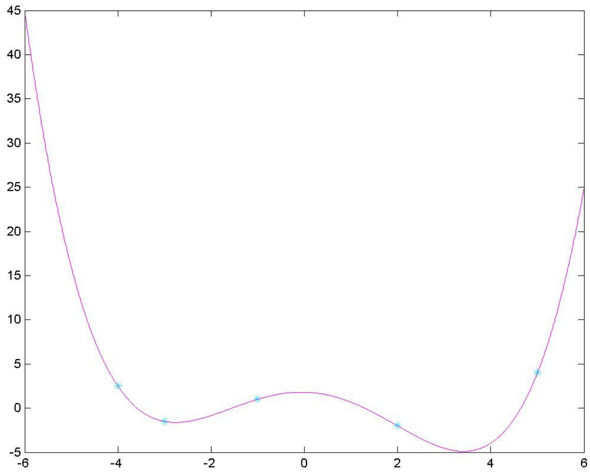
$L(x) = 0.05393518518519 \cdot x^4 - 0.04305555555556 \cdot x^3 - 1.02152777777778 \cdot x^2 - 0.11898148148148 \cdot x + 1.80555555555558.$

Njegovi su koeficijenti, ustvari, racionalni brojevi. Budući da MATLAB ne može računati s razlomcima, oni su aproksimirani decimalnim brojevima.

Prikažimo dobiveni polinom grafički. U novomu retku komandnoga prozora utipkajmo:

```
x1=-6:0.01:6;
plot(x,y,'*',x1,polyval(L,x1))
```

U posljednjoj smo naredbi zapisali da na istoj slici želimo ucrtati i zadane točke (označili smo ih znakom *) i *LIP* na zadanom segmentu. Dobivamo sljedeći graf:



Slika 10.

Najčešća primjena *LIP*–a je ipak u aproksimiranju transcendentnih (nealgebarskih) funkcija polinomima na određenim segmentima. Ilustrirajmo to na primjerima.

Primjer 2. Aproksimirajmo funkciju $f(x) = \sin(2 \cdot x)$ *LIP*–om u točkama čije su apscise redom jednake –3, –2, –1, 0, 1, 2 i 3 te ocijenimo apsolutnu pogrješku aproksimacije u točkama $x = -0.5$ i $x = 4$. Potom prikažimo grafove funkcije f i dobivenoga *LIP*–a na istoj slici.

Odredimo najprije matricu L čiji su elementi točno svi koeficijenti traženoga *LIP*-a. U novi redak komandnoga prozora utipkajmo:

```
x=-3:1:3;
y=sin(2*x);
L=polyfit(x,y,6)
```

MATLAB će ispisati:

```
L =
  Columns 1 through 4
   0.00000000000000   0.06078568014301  -0.00000000000000  -0.73316129220826
  Columns 5 through 7
   0.00000000000000   1.58167303889094  -0.00000000000000
```

Dakle, traženi je polinom

$$L(x) = 0.06078568014301 \cdot x^5 - 0.73316129220826 \cdot x^3 + 1.58167303889094 \cdot x.$$

Na prvi pogled možemo pomisliti da smo pogriješili jer stupanj *LIP*–a nije jednak 7 – 1 = 6. No, teorija ne tvrdi da je stupanj *LIP*–a jednak točno $n - 1$, nego *najviše* jednak $n - 1$, što znači da njegov stupanj može biti i strogo manji od $n - 1$. Upravo je to slučaj u ovom primjeru.

Ocijenimo sada apsolutnu pogrješku aproksimacije u zadanim točkama. Najprije za $x = 1$ izračunajmo

$$g_1 = |f(1) - L(1)|.$$

U novi redak komandnoga prozora utipkajmo:

```
g1=abs(sin(2*(-0.5)-polyval(L,-0.5))
```

pa će MATLAB ispisati:

```
g1 =
   0.14038007438399
```

Zaključujemo da je apsolutna pogrješka aproksimacije u točki $x = 1$ približno jednaka 0.14. Potpuno analogno, za $x = 4$ utipkajmo:

```
g1=abs(sin(2*4)-polyval(L,4)
```

pa će MATLAB ispisati:

```
g1 =
  20.65954767405372
```

U ovom je slučaju apsolutna pogrješka velika i iznosi gotovo 21, što znači da je aproksimacija funkcije $f(x)$ *LIP*–om u točki $x = 4$ katastrofalno loša. Objasnimo zašto je to tako. Naime, kada aproksimiramo neku funkciju *LIP*–om, mi je zapravo aproksimiramo na određenom segmentu. Ako je broj točaka relativno malen, pogrješka aproksimacije bit će „podnošljiva“, ali ako je broj točaka relativno velik, čak i na samom segmentu možemo dobiti velike pogrješke aproksimacije. U ovom smo primjeru funkciju $f(x) = \sin(2 \cdot x)$ aproksimirali *LIP*–

om na segmentu [–3, 3] uz relativno mali broj polaznih točaka. Točka $x = -0.5$ pripada tom segmentu pa je apsolutna pogrješka aproksimacije bila relativno mala. No, točka $x = 4$ ne pripada tom segmentu i zato je pripadna apsolutna pogrješka aproksimacije velika.

Napomena: Pogrješka aproksimacije može se smanjiti i pogodnim odabirom vrijednosti x_i, $i = 1, \ldots, n$. Tako se prigodom interpolacije neke funkcije f polinomom stupnja n na segmentu $[a, b]$ mogu uzeti tzv. *Čebiševljeve točke* definirane formulom

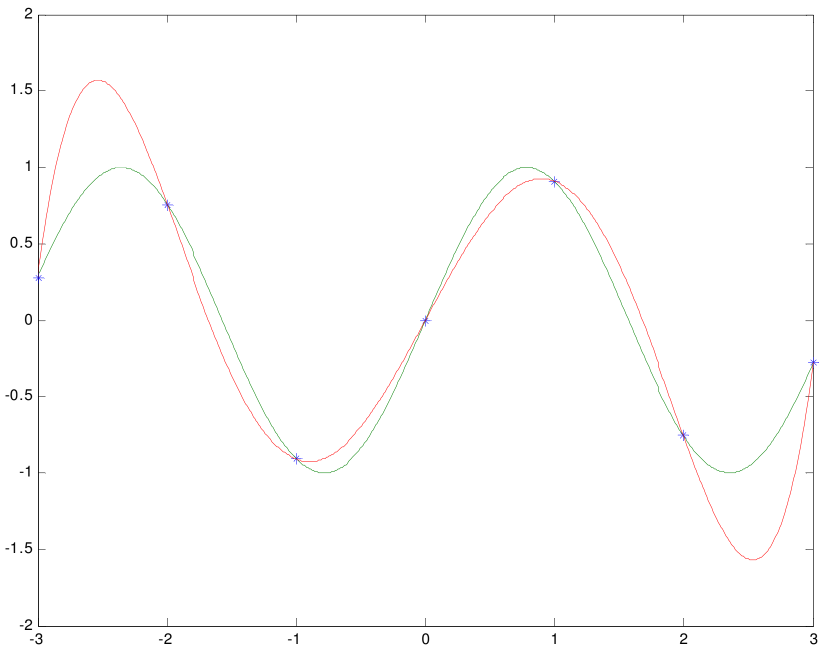
$$x_i = \frac{1}{2} \cdot \left[a + b + (a - b) \cdot \cos\left(\frac{2 \cdot i + 1}{2 \cdot n + 2} \right) \right], \text{ za } i = 0, \ldots, n,$$

pa se pokazuje da povećanjem stupnja n interpolacijskih polinoma niz tih polinoma konvergira prema funkciji f, tj. povećanjem stupnja n dobivamo sve bolje i bolje aproksimacije polazne funkcije. Detalje ovdje izostavljamo.

Pogledajmo kako naša aproksimacija izgleda grafički. Obavezno „očistimo“ komandni prozor, pa u njegova nova dva retka utipkajmo:

```
x1=-3:0.0001:3;
plot(x,y,'*',x1,sin(2*x1),x1,polyval(L,x1))
```

Dobit ćemo sljedeću sliku:

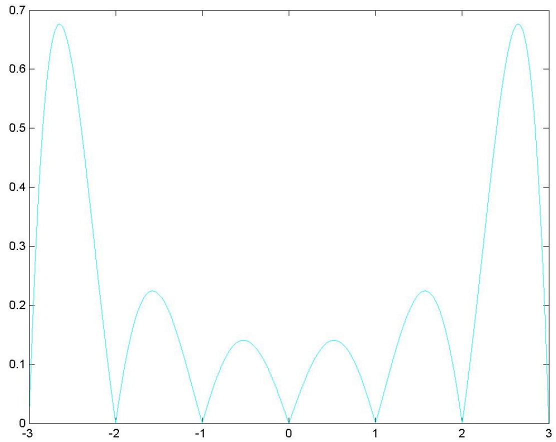


Slika 11.

Kvalitetu aproksimacije provjerit ćemo i crtanjem grafa funkcije $h = |f - L|$ na segmentu $[-3, 3]$. Pritisnimo tipku ↑ i preuredimo posljednje upisani redak ovako:

```
plot(x1,abs(sin(2*x1)-polyval(L,x1)))
```

Pritisnimo *Enter*, pa dobijemo sljedeću sliku:



Slika 12.

Primjer 3. Aproksimirajmo funkciju $f(x) = e^x$ *LIP* – om na segmentu $[-0.5, 0.5]$ koristeći krajnje točke segmenta i točke koje dijele taj segment na 4 dijela jednakih širina. Odredimo apsolutne pogrješke aproksimacija u točkama $x = 0.1$ i $x = 1$, te prikažimo grafove funkcija $f(x)$ i *LIP*–a na istoj slici.

Odredimo najprije apscise točaka kroz koje će prolaziti traženi *LIP*. Bilo koji segment možemo podijeliti na 4 dijela s ukupno $4 - 1 = 3$ unutrašnje točke segmenta. Iz činjenice sva četiri dijela moraju imati jednake širine zaključujemo da donja granica segmenta, spomenute tri unutrašnje točke segmenta i gornja granica segmenta tvore strogo rastući aritmetički niz. On očito ima ukupno 5 članova. Koristeći funkciju `linspace` generiramo konačan aritmetički niz kojemu je prvi član –0.5, posljednji član 0.5, a ukupan broj članova 5. U nova tri retka komandnoga prozora utipkajmo:

```
x=linspace(-0.5,0.5,5);
y=exp(x);
L=polyfit(x,y,4)
```

Nakon što pritisnemo *Enter*, MATLAB će ispisati:

```
L =
  Columns 1 through 4
   0.04210273467924   0.16928716669238   0.49997817715571   0.99986881931440
  Column 5
   1.00000000000000
```

Dakle, traženi *LIP* je:

$L(x) = 0.04210273467924 \cdot x^4 + 0.16928716669238 \cdot x^3 + 0.49997817715571 \cdot x^2 + 0.9998688193144 \cdot x + 1.$

Ocijenimo apsolutne pogrješke aproksimacija u točkama $x = 0.1$ i $x = 1$. U novi redak komandnoga prozora utipkajmo:

```
g1=abs(exp(0.1)-polyval(L,0.1)), g2=abs(exp(1)-polyval(L,1))
```

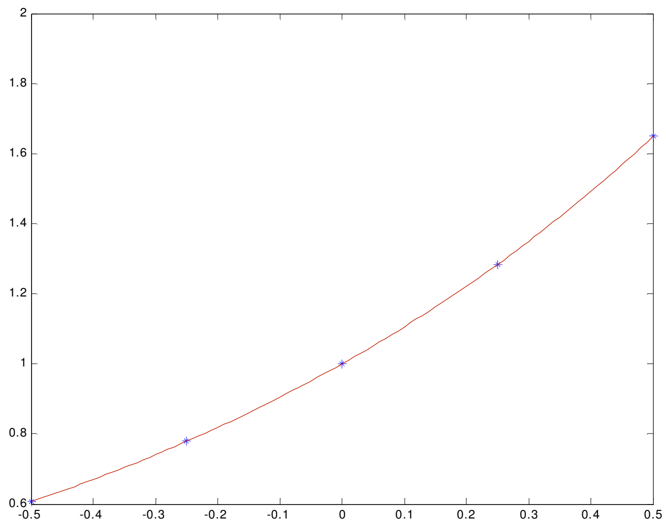
pa će MATLAB ispisati:

```
g1 =
    1.075693248986376e-005
g2 =
   0.00704493061731
```

U oba slučaja pogrješke su relativno male. Pogledajmo to na grafovima funkcija $f(x)$ i *LIP*–a prikazanima na istoj slici. U novi redak komandnoga prozora utipkajmo:

```
x1=-0.5:0.01:0.5;
plot(x,y,'*',x1,exp(x1),x1,polyval(L,x1))
```

pa ćemo dobiti sljedeću sliku:



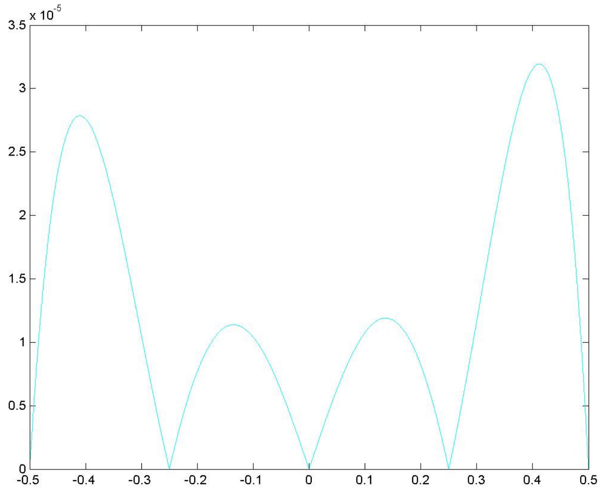
Slika 13.

Obrazložimo što se dogodilo i zašto na dobivenoj slici imamo samo jednu krivulju. Segment na kojemu smo aproksimirali funkciju $f(x) = e^x$ je relativno "uzak" (duljina toga segmenta je jednaka 1), pa su na tome segmentu apsolutne pogrješke aproksimacije tako male da ih je grafički nemoguće primijetiti. Zbog toga slobodno možemo reći da se u ovom slučaju grafovi funkcije $f(x)$ i *LIP*–a praktički "podudaraju".

Koliko se ti grafovi doista podudaraju, najbolje možemo vidjeti prikažemo li grafički funkciju $h = |f - L|$ na segmentu [–0.5, 0.5]. Ponovno pritisnimo tipku ↑, pa preuredimo posljednje upisani redak ovako:

```
plot(x1,abs(exp(x1)-polyval(L,x1)))
```

Pritisnemo *Enter*, pa dobivamo graf prikazan na Slici 14.:



Slika 14.

U praktičnim je primjenama (posebice u statistici) vrlo često potrebno naći polinom 1. ili 2. stupnja koji relativno dobro opisuje zadani skup točaka. Pritom je ukupan broj zadanih točaka strogo veći od 3 pa se kao rješenje takvoga problema općenito ne može uzeti *LIP*. Nameće se pitanje: Što učiniti u takvome slučaju? Rješenje toga problema nalazi se tzv. *metodom najmanjih kvadrata*. Grubo govoreći, osnovna ideja te metode jest odrediti polinom (općenito različit od *LIP*–a) takav da zbroj svih udaljenosti zadanih točaka od grafa toga polinoma bude što manji (idealno: 0, ali u praksi je to vrlo rijedak slučaj). Tako se npr. u statističkim regresijskim modelima nastoji odrediti polinom 1. stupnja takav da zbroj svih udaljenosti zadanih točaka od pravca (grafa toga polinoma) bude što manji. Mi ćemo takve probleme jednostavno i brzo riješiti rabeći MATLAB, točnije njegovu funkciju *polyfit*.

Pokažimo navedeno na primjerima.

Primjer 4. Aproksimirajmo funkciju $f(x) = \cos(2 \cdot x)$ na segmentu [–0.5, 0.5] polinomom 1. stupnja koristeći krajnje točke segmenta i točke koje dijele segment na ukupno 5 dijelova jednakih širina. Ocijenimo apsolutnu pogrješku aproksimacije u točkama $x = 0$ i $x = 1$, pa na istoj slici grafički prikažimo funkciju f i dobiveni polinom na zadanom segmentu.

Označimo traženi polinom s p. Analogno rješenju Primjera 3., najprije zadajemo točke koristeći funkciju `linspace` i računamo pripadajuće vrijednosti funkcije f u tim točkama. U novi redak komandnoga prozora utipkajmo:

```
x=linspace(-0.5,0.5,6);
y=cos(2*x);
```

Sada možemo primijeniti funkciju *polyfit*. Matrice x i y upravo smo deklarirali, a za prirodan broj k (koji je posljednja od triju ulaznih varijabli funkcije *polyfit*) uzet ćemo $k = 1$ jer tražimo polinom 1. stupnja. Utipkajmo:

```
p=polyfit(x,y,1)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
p =
   0.00000000000000   0.78190149953969
```

Dakle, traženi je polinom

$p(x) = 0.78190149953969$ (konstantni polinom).

Ocijenimo apsolutne pogrješke aproksimacije u točkama $x = 0$ i $x = 1$. U novi redak komandnoga prozora utipkajmo:

```
g1=abs(cos(2*0)-polyval(p,0)),g2=abs(cos(2*1)-polyval(p,1))
```

pa će MATLAB ispisati:

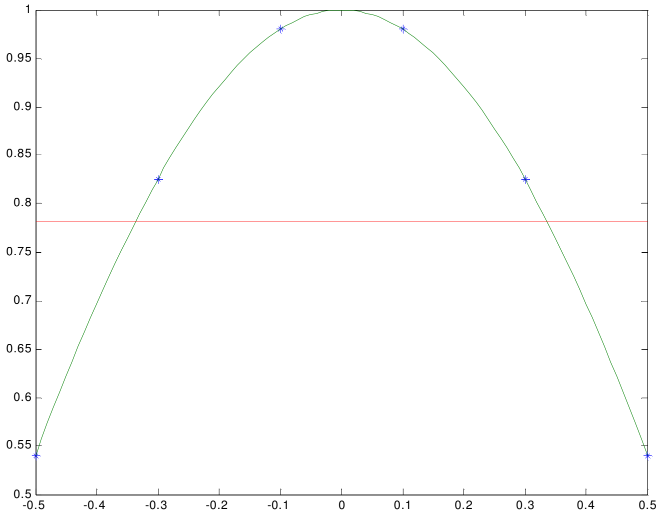
```
g1 =
   0.21809850046031
g2 =
   1.19804833608683
```

Takve rezultate smo mogli i očekivati jer je $x = 0$ element segmenta $[-0.5, 0.5]$, a $x = 1$ nije.

Grafove zadane funkcije i polinoma p na istom segmentu crtamo tako da u nova dva retka komandnoga prozora utipkamo:

```
x1=-0.5:0.0001:0.5;
plot(x,y,'*',x1,cos(2*x1),x1,polyval(p,x1))
```

Dobit ćemo sljedeću sliku:

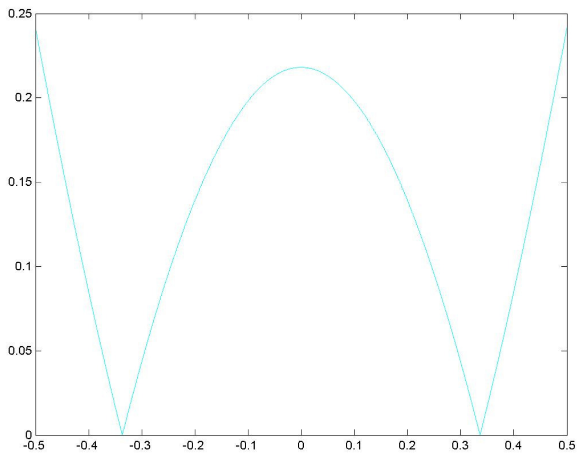


Slika 15.

Kvalitetu dobivene aproksimacije procjenjujemo na temelju grafičkoga prikaza funkcije $h = |f - P|$. Preuredimo posljednje upisani redak tako da dobijemo:

```
plot(x1,abs(cos(2*x1)-polyval(p,x1)))
```

Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku:



Slika 16.

Možemo zaključiti da je aproksimacija funkcije $f(x) = \cos(2 \cdot x)$ polinomom 1. stupnja na segmentu $[-0.5, 0.5]$ vrlo loša.

Primjer 5. Aproksimirajmo funkciju iz prethodnoga zadatka polinomima 2. i 3. stupnja u smislu metode najmanjih kvadrata. Kvalitetu dobivene aproksimacije provjerimo grafički.

U ovom primjeru najprije uzimamo $k = 2$, a potom $k = 3$. Traženi polinom 2. stupnja označit ćemo s p_2, a traženi polinom 3. stupnja s p_3. U novi redak komandnoga prozora utipkajmo:

```
p2=polyfit(x,y,2), p3=polyfit(x,y,3)
```

pa će MATLAB ispisati:

```
p2 =
  -1.82508071154816  -0.00000000000000   0.99482758255364
p3 =
   0.00000000000000  -1.82508071154817  -0.00000000000000   0.99482758255364
```

Stoga je

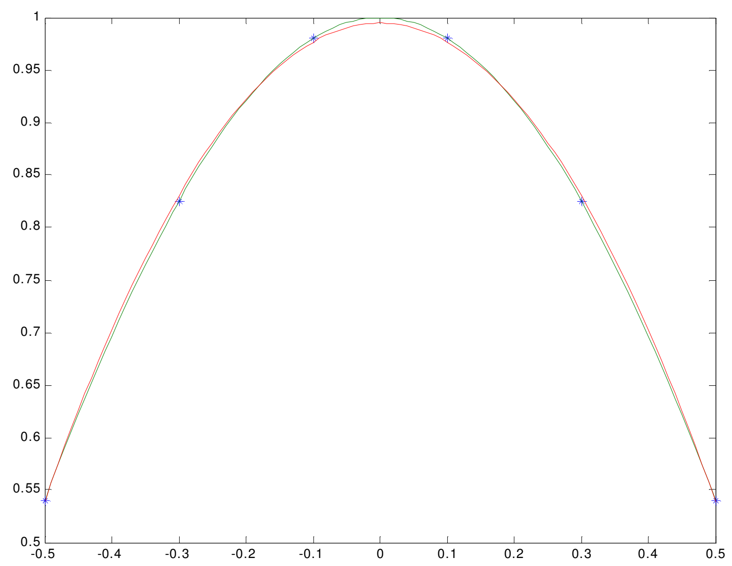
$$p_2(x) = p_3(x) = -1.82508071154816 \cdot x^2 + 0.99482758255364.$$

Primjećujemo da smo u oba slučaja dobili isti polinom, što znači da ne postoji polinom stupnja točno 3 koji dobro opisuje zadanu funkciju u smislu metode najmanjih kvadrata.

Prikažimo grafove funkcija f i p_2 na istoj slici. U novi redak komandnoga prozora utipkajmo:

```
plot(x,y,'*',x1,cos(2*x1),x1,polyval(p2,x1))
```

Pritisnemo *Enter*, pa dobivamo sljedeću sliku:



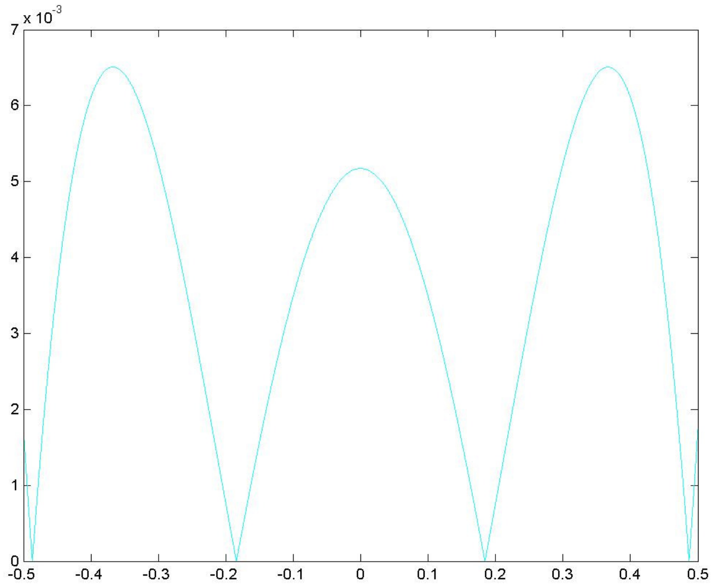
Slika 17.

Primjećujemo da se grafovi funkcija gotovo podudaraju, što znači da je aproksimacija polinomom 2. stupnja vrlo dobra.

Kvalitetu dobivene aproksimacije uobičajeno procijenjujemo crtajući graf funkcije $h = |f - p_2|$ na zadanom segmentu. Pritisnimo tipku ↑, pa preuredimo posljednje upisani redak ovako:

```
plot(x1,abs(cos(2*x1)-polyval(p2,x1)))
```

Pritisnemo *Enter*, pa ćemo dobiti sljedeću sliku:



Slika 18.

Dakle, najveća vrijednost funkcije h na zadanom segmentu je strogo manja od 0.007, što opravdava zaključak da se radi o vrlo dobroj aproksimaciji.

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

10.4. Zadatci za vježbu

1. Odredite *LIP* čiji graf prolazi točkama:

a) $T_1 = (-2, 1)$, $T_2 = (-1, 2)$, $T_3 = (0, 0)$ i $T_4 = (1, 1)$;
b) $T_1 = (\ln 2, e^{-2})$, $T_2 = (\cos 3, \operatorname{ctg} 5)$, $T_3 = (2 - \sqrt{2}, 2 + \sqrt{2})$, $T_4 = (-\sqrt[3]{3}, \log_2 3)$;
c) $T_1 = (-1, 2)$, $T_2 = (0, 1)$, $T_3 = (1, -1)$, $T_4 = (2, -3)$, $T_5 = (3, -5)$;
d) $T_1 = (-e, \ln 3)$, $T_2 = (-2, \ln 2)$, $T_3 = (-1, 0)$, $T_4 = (0, -\ln 2)$ i $T_5 = (1, -\ln 3)$.

U svakom od slučajeva odredite apsolutnu i relativnu pogrješku aproksimacije u točkama $x = 0.1$ i $x = 5$. Grafički provjerite kvalitetu dobivenih aproksimacija.

2. Aproksimirajte sljedeće funkcije *LIP*-om stupnja najviše 3 koristeći točke koje njihovo prirodno područje definicije dijele na 4 dijela jednakih širina ako je:

a) $f(x) = \arcsin x$;
b) $f(x) = \arccos x$;
c) $f(x) = \sqrt{4 - x^2}$;
d) $f(x) = \sqrt{x - x^2}$.

U svakom od slučajeva odredite apsolutnu i relativnu pogrješku aproksimacije u točkama $x = 0.\ 2$ i $x = 0.6$. Grafički prikažite funkciju f i dobiveni *LIP* na odgovarajućem segmentu. Potom grafički provjerite kvalitetu dobivenih aproksimacija.

3. **a)** Aproksimirajte funkciju $f(x) = \ln x$ *LIP*–om na segmentu $[0.5, 1.5]$ koristeći točke koje zadani segment dijele na točno 5 dijelova jednakih širina.
b) Odredite apsolutnu i relativnu pogrješku dobivene aproksimacije u točkama $x = 1$ i $x = 2$.
c) Grafički prikažite zadanu funkciju f i dobiveni *LIP* na zadanom segmentu. Potom grafički provjerite kvalitetu dobivene aproksimacije.

4. **a)** Aproksimirajte funkciju $f(x) = e^{-x}$ *LIP*–om na segmentu $[-1, 1]$ koristeći točke koje zadani segment dijele na točno 8 dijelova jednakih širina.
b) Odredite apsolutnu i relativnu pogrješku aproksimacije u točkama $x = -0.1$ i $x = 1.1$.
c) Grafički prikažite zadanu funkciju f i dobiveni *LIP* na zadanom segmentu. Potom grafički provjerite kvalitetu dobivene aproksimacije.

5. **a)** Aproksimirajte funkciju $f(x) = \operatorname{ctg}(2 \cdot x)$ *LIP*–om na segmentu $\left[-\frac{\pi}{8}, \frac{\pi}{8}\right]$ koristeći točke koje zadani segment dijele na 10 dijelova jednakih širina.
b) Odredite apsolutnu pogrješku aproksimacije u točkama $x = -0.1$ i $x = 0.4$.
c) Grafički prikažite zadanu funkciju f i dobiveni *LIP* na zadanom segmentu. Potom grafički provjerite kvalitetu dobivene aproksimacije.

6. Odredite polinome 1., 2. i 3. stupnja koji u smislu metode najmanjih kvadrata najbolje aproksimiraju funkciju $f(x) = \sin(3 \cdot x)$ na segmentu $\left[0, \frac{2}{3} \cdot \pi\right]$ koristeći točke koje zadani segment dijele na 6 jednakih dijelova. Grafički provjerite kvalitetu dobivenih aproksimacija.

7. Zadana je tablica vrijednosti funkcije *f*.

x	0	1	3	4	6	7
$f(x)$	3	6	24	48	192	384

a) Odredite polinome 1., 2. i 3. stupnja koji najbolje opisuju zadani skup podataka u smislu metode najmanjih kvadrata.
b) Odredite *LIP* koji najbolje opisuje zadani skup podataka.
c) Izračunajte vrijednost y pridruženu $x = 5$ koristeći svaki od polinoma iz **a)** i **b)** zadatka.

8. Zadana je tablica vrijednosti funkcije *f*.

x	0	1.5	2.5	3	4	4.5
$f(x)$	–1	3.2	13.6	25	79	138.3

a) Odredite polinome 1., 2. i 3. stupnja koji najbolje opisuju zadani skup podataka u smislu metode najmanjih kvadrata.
b) Odredite *LIP* koji najbolje opisuje zadani skup podataka.
c) Izračunajte vrijednost y pridruženu $x = 2$ koristeći svaki od polinoma iz **a)** i **b)** zadatka.

9. Kreirajte funkcijsku *m*–datoteku **stupanjlip.m** koja sadrži jedino funkciju *stupanjlip* čije su ulazne varijable jednoretčane realne matrice x i y istoga tipa takve da je elementu x_i pridružen element y_i, $i = 1, 2, \ldots$, a jedina izlazna varijabla stupanj *LIP*–a koji najbolje opisuje skup točaka $\{(x_i, y_i) : i = 1, 2, \ldots\}$.

10. <u>Bez uporabe MATLAB-</u>a odredite što će se ispisati utipkavanjem sljedećega niza naredbi:

a)
```
x=0:0.2:1;
y=x+1;
p=polyfit(x,y,2)
```

b)
```
x=-1:0.5:2;
y=x.^2+x-1;
p=polyfit(x,y,3)
```

c)
```
x=-2:1:5;
y=x-x.^2-x.^3;
p=polyfit(x,y,4)
```

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

10.5. Numerička integracija

U ovoj ćemo točki razmatrati sljedeći problem:

***Problem*:** Približno izračunati određeni integral

$$\int_a^b f(x) \cdot dx .$$

Pritom pretpostavljamo da je $f(x)$ realna funkcija realne varijable integrabilna na segmentu $[a, b]$.

Postoji više formula za približno računanje određenoga integrala (lijeva formula pravokutnika, desna formula pravokutnika, trapezna formula, Simpsonova formula itd.). Mi ćemo ovdje razmatrati dvije najpoznatije: *trapeznu* i *Simpsonovu*.

Grubo govoreći, *trapezna* se *formula* zasniva na zamjeni podintegralne funkcije $f(x)$ interpolacijskim polinomom 1. stupnja, odosno zamjeni krivocrtnoga trapeza običnim trapezom, i to na svakom od segmenata dobivenih razdiobom segmenta $[a, b]$ na n jednakih dijelova. Širina svakoga dijela jednaka je $h = \dfrac{b-a}{n}$, pa su diobene točke (tzv. *čvorovi*) dani izrazima:

$$\begin{cases} x_0 = a, \\ x_k = a + k \cdot h, \text{ za svaki } k \in [n] \end{cases} \tag{1}$$

Zbrajanjem površina svih trapeza dobiva se spomenuta trapezna formula:

$$\int_a^b f(x) \cdot dx \approx h \cdot \left[\frac{1}{2} \cdot f(x_0) + \sum_{k=1}^{n-1} f(x_k) + \frac{1}{2} \cdot f(x_n) \right] = \frac{b-a}{n} \cdot \left[\frac{1}{2} \cdot f(a) + \sum_{k=1}^{n-1} f(x_k) + \frac{1}{2} \cdot f(b) \right]. \tag{2}$$

Simpsonova se, pak, *formula* zasniva na razdiobi segmenta $[a, b]$ na *paran* broj dijelova (tj. n nužno mora biti paran broj), pa se u svakom od dobivenih segmenata podintegralna funkcija zamjenjuje s interpolacijskim polinomom drugoga stupnja (što znači da se površina krivocrtnoga trapeza na svakom od tih segmenata zamjenjuje s površinom ispod parabole). Izrazi za računanje čvorova su ponovno dani s (1). Zbrajanjem svih površina ispod dobivenih parabola dobiva se spomenuta Simpsonova formula:

$$\int_a^b f(x) \cdot dx \approx \frac{h}{3} \cdot \{ f(a) + 2 \cdot [f(x_2) + f(x_4) + \ldots + f(x_{n-2})] + 4 \cdot [f(x_1) + f(x_3) + \ldots + f(x_{n-1})] + f(b) \}. \tag{3}$$

MATLAB ne posjeduje "gotove" (standardne) funkcije koje određeni integral približno računaju prema trapeznoj, odnosno Simpsonovoj formuli. Ovdje ćemo navesti dvije posebno stvorene funkcije *trapez* i *simpson* koje to čine. Funkcija *trapez* kao ulazne varijable ima podintegralnu funkciju $f(x)$ (zapisanu u funkcijskoj *m*–datoteci), granice segmenta a i b, te broj dijelova n na koje treba podijeliti segment $[a, b]$, a kao jedinu izlaznu varijablu približnu vrijednost integrala $\int_a^b f(x)\cdot dx$. Ona glasi:

```
function z=trapez(funkcija,a,b,n)
sirina=(b-a)/n;
x=a:sirina:b;
for i=1:(n+1)
y(i)=feval(funkcija,x(i));
end
z=sirina*(y(1)/2+sum(y(2:n))+y(n+1)/2);
```

Funkcija *simpson* ima iste ulazne i izlazne varijable kao i funkcija *trapez*, ali se zahtijeva provjera je li n paran prirodan broj. Ona glasi:

```
function z=simpson(funkcija,a,b,n)
if mod(n,2)>0
error('Broj dijelova mora biti paran!'),
end
sirina=(b-a)/n;
x=a:sirina:b;
for i=1:(n+1)
     y(i)=feval(funkcija,x(i));
end
z=sirina*(y(1)+2*(2*sum(y(2:2:n))+sum(y(3:2:n)))+y(n+1))/3;
```

(Radi jednostavnosti, niti u jednoj od navedenih funkcija ne zahtijevamo provjeru je li n prirodan broj. Kao korisnu vježbu, možete doraditi obje funkcije tako da provjeravaju je li taj uvjet zadovoljen.)

Napomena: Ukupan broj dijelova (n) na koje treba podijeliti segment $[a, b]$ može se odrediti ovisno o točnosti aproksimacije (tzv. *toleranciji*) svake pojedine metode. Označimo li tu točnost s ε i ako je unaprijed zadamo (npr. $\varepsilon = 10^{-5}$), onda najmanji potreban broj dijelova n za trapeznu formulu možemo izračunati kao najmanji prirodan broj koji zadovoljava nejednakost

$$n \geq \sqrt{\frac{(b-a)^3}{12\cdot\varepsilon}\cdot M_2}\ , \tag{4}$$

pri čemu je M_2 najveća vrijednost (maksimum) koju funkcija $|f''|$ postiže na segmentu $[a, b]$. (Pritom pretpostavljamo da je podintegralna funkcija f dvaput neprekidno derivabilna na segmentu $[a, b]$.) Za Simpsonovu se formulu taj broj dijelova određuje kao najmanji prirodan broj koji zadovoljava nejednakost

$$n \geq \sqrt[4]{\frac{(b-a)^5}{180 \cdot \varepsilon} \cdot M_4}\,, \tag{5}$$

pri čemu je M_4 najveća vrijednost (maksimum) koju funkcija $|f^{(IV)}|$ postiže na segmentu $[a, b]$. (Pritom pretpostavljamo da je podintegralna funkcija f četiri puta neprekidno derivabilna na segmentu $[a, b]$.) Upravo zbog spomenutih maksimuma kao jedan od ulaznih argumenata funkcija *trapez* i *simpson* navodi se broj dijelova n, a ne točnost ε.

Sljedeći primjer riješit ćemo koristeći upravo nejednakosti (4) i (5).

***Primjer* 1.** S točnošću od 10^{-5} izračunajmo određeni integral $\int_0^1 \sin\left(x^2\right) \cdot dx$ koristeći:

a) trapeznu formulu;
b) Simpsonovu formulu.

a) Da bismo primijenili funkciju *trapez*, pomoću nejednakosti (4) najprije moramo odrediti ukupan broj dijelova na koje ćemo podijeliti segment [0, 1]. Prvi je korak odrediti najveću vrijednost apsolutne vrijednosti druge derivacije funkcije $f(x) = \sin(x^2)$. „Počistimo“ komandni prozor, pa u utipkajmo redom:

```
syms x
f=sin(x^2);
f2=diff(f,2);
f3=diff(f,3);
g=0*x;
rijesi(f3,g)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
ans =
     0
     0
     0
```

Dakle, jedina stacionarna točka funkcije f'' je $x = 0$, tj. donja granica segmenta [0, 1]. Stoga će funkcija $|f''|$ poprimiti maksimum ili za $x = 0$ ili za $x = 1$. U sljedeći redak komandnoga prostora utipkamo:

```
M2=max(abs(subs(f2,0)),abs(subs(f2,1)))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
M2 =
   2.28527932749531
```

Sada primijenimo nejednakost (4). Utipkajmo:

```
n=ceil(sqrt((1-0)^3/(12*10^(-5))*M2))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
n =
   138
```

Preostaje pozvati funkciju *trapez* i izračunati polazni određeni integral. Podintegralnu funkciju najprije zapišimo u posebnu funkcijsku *m*–datoteku. Otvorimo novu *m*-datoteku i utipkajmo:

```
function y=pif(x);
y=sin(x.^2);
```

Pohranimo dobivenu datoteku pod nazivom **pif.m** i vratimo se u komandni prozor. U novi redak toga prozora utipkamo:

```
I=trapez(@pif,0,1,n)
```

Pritisnemo *Enter*, pa dobivamo traženu vrijednost:

```
I =
   0.31027303032220
```

b) Najprije „počistimo“ komandni prozor, pa u njegova nova tri retka utipkajmo:

```
f4=diff(f,4);
f5=diff(f,5);
rijesi(f5,g)
```

(Funkciju $g(x) = 0 \cdot x$ imamo deklariranu u **a)** podzadatku, a „čišćenje“ pomoću funkcije `clc` nije utjecalo na pohranu te funkcije u memoriji.) Pritisnemo *Enter*, pa će MATLAB ispisati:

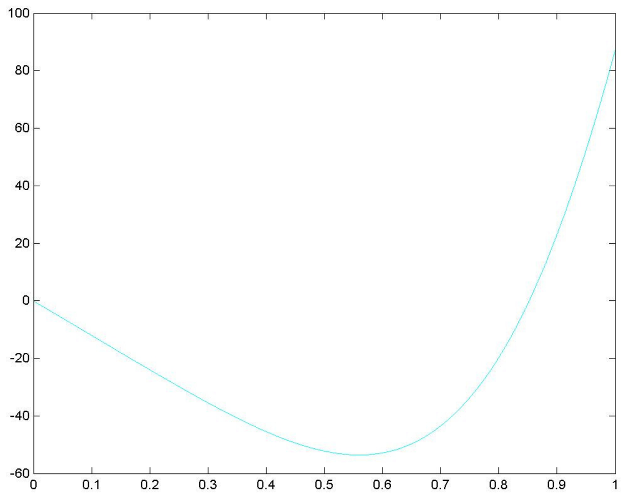
```
ans =
                    0
                    0 - 1.68676136346992i
                    0 + 1.68676136346992i
```

Iz dobivenoga bismo rješenja mogli pomisliti da je $x = 0$ jedina stacionarna točka funkcije $f^{(IV)}(x)$ u segmentu [0, 1]. Međutim, to nije točno. (Nemojte odmah kriviti funkciju `solve`: ni ona ne može uspješno riješiti *svaku* jednadžbu, tim više što je ovdje riječ o relativno složenoj nealgebarskoj jednadžbi.)

Nacrtajmo graf funkcije $f^{(V)}(x)$ na segmentu [0, 1]. U nova dva retka komandnoga prozora utipkajmo:

```
x1=0:0.0001:1;
plot(x1,subs(f5,x1))
```

Dobit ćemo sljedeću sliku:



Slika 19.

Iz nje vidimo da je jedna stacionarna točka funkcije $f^{(IV)}(x)$ na segmentu [0, 1] doista $x_1 = 0$, dok je druga približno jednaka 0.8. Njezinu točniju vrijednost dobit ćemo primjenom funkcije `fzero`. U novi redak komandnoga prozora utipkajmo:

```
f5
```

MATLAB će ispisati:

```
f5 =
32*cos(x^2)*x^5+160*sin(x^2)*x^3-120*cos(x^2)*x
```

Označimo izraz za f5, pa ga kopirajmo koristeći tipke Ctrl i C. U novi redak komandnoga prozora utipkajmo

```
x2=fzero
```

otvorimo običnu zagradu, zalijepimo kopirani izraz, stavimo znak zareza, nadopišimo 1 i zatvorimo običnu zagradu. Taj redak naposljetku treba izgledati ovako:

```
x2=fzero('32*cos(x^2)*x^5+160*sin(x^2)*x^3-120*cos(x^2)*x',1)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
x2 =
   0.85207666630662
```

Tako ćemo najveću vrijednost funkcije $|f^{(IV)}(x)|$ na segmentu [0, 1] odrediti kao najveću od triju vrijednosti: $f^{(IV)}(0)$, $f^{(IV)}(x_2)$ i $f^{(IV)}(1)$. U novi redak komandnoga prozora utipkajmo:

```
X=[ans(subs(f4,0)) abs(subs(f4,x2)) abs(subs(f4,1))];
M4=max(X)
```

Napomena: Kad tražimo najveći od barem tri zadana broja, moramo ih zapisati u obliku jednoretčane matrice, a potom primijeniti funkciju *max*. Matrični zapis nije potreban kad tražimo veći od dvaju zadanih brojeva.

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
M4 =
  28.42851540309637
```

Za određivanje broja n primijenimo nejednakost (5). U novi redak komandnoga prozora utipkamo:

```
n=ceil(sqrt((1-0)^5/(180*10^(-5))*M4))
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
n =
   126
```

Dobivena vrijednost varijable n je paran prirodan broj, pa preostaje izračunati polazni integral primjenom funkcije `simpson`. Njezini argumenti bit će funkcija f pohranjena u datoteci **pif.m**, $a = 0$, $b = 1$ i $n = 126$. U novi redak komandnoga prozora utipkajmo:

```
I1=simpson(@pif,0,1,n)
```

Pritisnemo *Enter*, pa će MATLAB ispisati:

```
I1 =
   0.31026830140552
```

Iz **a)** i **b)** podzadatka možemo zaključiti da je približna vrijednost polaznoga integrala (izračunana sa zadanom točnošću) 0.31027.

Za približno računanje određenoga integrala MATLAB posjeduje dvije ugrađene funkcije:

`quad` i `quadl`. Obje te funkcije kao ulazne varijable imaju podintegralnu funkciju $f(x)$, te granice segmenta a i b, a zasnivaju se na prilagođenoj Simpsonovoj, odnosno Lobattovljevoj formuli i imaju dogovornu točnost reda 10^{-6}. Ako želimo postići veću točnost (npr. `eps`) , tu točnost moramo navesti kao ulaznu varijablu pri pozivu funkcije. Mi ćemo koristiti funkciju `quadl` koja se (iz komandnoga prozora) može pozvati na ukupno tri načina:

1. način: `I = quadl(funkcija,donja_granica,gornja_granica)`

Npr. želimo li izračunati određeni integral $\int_0^{\frac{\pi}{2}} \sin x \cdot dx$, upisat ćemo:

```
I = quadl('sin(x)',0,pi/2)
```

MATLAB će ispisati:

```
I =
   0.99999999999175
```

Želimo li isti integral izračunati s točnošću od 10^{-12}, upisat ćemo:

```
I = quadl('sin(x)',0,pi/2,1e-12)
```

i dobiti:

```
I =
   1.00000000000000
```

2. način (koristeći funkciju *inline* za definiranje objekta u komandnomu prozoru):

```
F=inline(funkcija)
I=quadl(F,donja_granica,gornja_granica)
```

Integral iz 1.) tako možemo približno izračunati i ovako:

```
F=inline('sin(x)');
I = quadl(F,0,pi/2)
```

pa će MATLAB (očekivano) ispisati:

```
I =
   0.99999999999175
```

3. način: Podintegralnu funkciju zapišemo u obliku funkcijske *m*–datoteke **pif.m**, pa je iz komandnoga prozora pozivamo ovako:

```
I=quadl(@pif,donja_granica,gornja_granica)
```

Za početnike ili nedovoljno iskusne korisnike treći je način najpraktičniji jer se – u slučaju pogrešnoga zadavanja funkcije f – ispravke ne moraju vršiti u komandnomu prozoru. Stoga ćemo i mi primjenjivati taj način kad god to bude moguće. Za iskusne korisnike najpraktičniji je prvi način.

Pogledajmo nekoliko primjera.

***Primjer* 3.** Zadan je određeni integral

$$I = \int_0^1 \frac{\sqrt{\operatorname{arctg} x}}{1+x^2} \cdot dx .$$

Izračunajmo približne vrijednosti toga integrala pomoću trapezne, odnosno Simpsonove formule (uzimajući n = 1000), te pomoću funkcije `quadl` (s točnošću od 10^{-12}). Usporedimo dobivene rezultate s točnom vrijednošću integrala i ocijenimo pogrješku.

Podintegralnu funkciju najprije ćemo zapisati u funkcijsku m–datoteku **pif.m**. Otvorimo tu datoteku i utipkajmo:

```
function y=pif(x);
y=sqrt(atan(x))./(1+x.^2);
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. Izračunajmo najprije približnu vrijednost zadanoga integrala rabeći trapeznu formulu. U novi redak komandnoga prozora utipkajmo:

```
I1=trapez(@pif,0,1,1000)
```

pa će MATLAB ispisati:

```
I1 =
   0.46402073395717
```

Za primjenu Simpsonove formule u novomu retku utipkajmo:

```
I2=simpson(@pif,0,1,1000)
```

pa će MATLAB ispisati:

```
I2 =
   0.46402476578696
```

Uporabimo li funkciju `quadl`, nakon utipkavanja

```
I3=quadl(@pif,0,1,1e-12)
```

dobivamo:

```
I3 =
   0.46402733306852
```

Izračunajmo sada točnu vrijednost integrala koristeći funkciju `int`. U nova tri retka komandnoga prozora utipkajmo:

```
syms x
f=sqrt(atan(x))/(1+x^2);
I=int(f,x,0,1)
```

Pritisnimo *Enter*, pa će MATLAB ispisati:

```
I =
1/12*pi^(3/2)
```

Približnu vrijednost toga broja u MATLAB-u dobijemo koristeći funkciju `double`. Utipkajmo:

```
double(I)
```

pa će MATLAB ispisati:

```
I =
   0.46402733306931
```

Preostaje nam ocijeniti pogrješke približnih računa. To ćemo učiniti tako da izračunamo sljedeće vrijednosti:

$$g_1 = |I - I_1|,$$
$$g_2 = |I - I_2|,$$
$$g_3 = |I - I_3|.$$

U novi redak komandnoga prozora utipkajmo:

```
g1=abs(I-I1),g2=abs(I-I2),g3=abs(I-I3)
```

pa će MATLAB ispisati:

```
g1 =
    6.599112139127339e-006
g2 =
    2.567282352927691e-006
g3 =
    3.338995746560158e-012
```

Zaključujemo da smo pomoću funkcije `quadl` dobili najbolji (zapravo, najpribližniji) rezultat, a pomoću trapezne formule najlošiji rezultat.

***Primjer* 4.** Zadan je određeni integral

$$\int_1^2 e^{-x^2} \cdot dx .$$

Izračunajmo približnu vrijednost toga integrala rabeći trapeznu i Simpsonovu formulu (uz $n =$ $= 2000$). Pomoću funkcije `quadl` procijenimo točnu[32] vrijednost integrala s točnošću od 10^{-6}.

Najprije modificiramo *m*–datoteku **pif.m**:

```
function y=pif(x);
y=exp(-x.^2);
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. U njegov novi redak utipkajmo:

```
I1=trapez('pif',1,2,2000),I2=simpson('pif',1,2,2000),I3=quadl(@pif,1,2)
```

pa će MATLAB ispisati:

```
I1 =
   0.13525727175200
I2 =
   0.13525725794999
I3 =
   0.13525725727024
```

Možemo zaključiti da je prava vrijednost integrala (s točnošću od 10^{-6}) jednaka

$$I = 0.135257.$$

Napomena: Utipkavanjem

```
I3=quadl(@pif,1,2,1e-12)
```

dobili bismo:

```
I3 =
   0.13525725794999
```

tj. $I_3 = I_2$. Budući da *ne* možemo izračunati točnu vrijednost polaznoga integrala, ne smijemo zaključiti da u ovom slučaju Simpsonova formula daje točniji rezultat od funkcije `quadl`. Takve je usporedbe primjereno napraviti isključivo u slučajevima kad možemo izračunati točnu vrijednost određenoga integrala.

---

[32] Može se pokazati da nije moguće eksplicitno odrediti primitivnu funkciju funkcije $f(x) = e^{-x^2}$, a samim tim niti izračunati točnu vrijednost zadanoga integrala.

Želimo li približno izračunati neki dvostruki integral, rabimo MATLAB-ovu funkciju `dblquad` čija je sintaksa analogna funkciji `quadl`. Evo jednoga primjera.

Primjer 5. Približno izračunajmo dvostruki integral

$$I = \iint_D \sqrt{x^2 + y^2} \cdot dx \cdot dy$$

ako je područje integracije D jedinični kvadrat $[0,1]^2$. Prema Fubinijevu poučku[33] dvostruki integral možemo zamijeniti s dva "jednostruka" (obična) integrala:

$$\iint_D \sqrt{x^2 + y^2}\,dxdy = \int_0^1\int_0^1 \sqrt{x^2 + y^2}\,dxdy = \int_0^1 \left(\int_0^1 \sqrt{x^2 + y^2}\,dx \right) dy.$$

Modificirajmo sada funkciju **pif.m** ovako:

```
function z=pif(x,y);
z=sqrt(x.^2+y.^2);
```

Pohranimo unesene naredbe i vratimo se u komandni prozor. Utipkajmo:

```
I=dblquad(@pif,0,1,0,1,1e-12)
```

pa će MATLAB ispisati:

```
I =
   0.76519571646553
```

Poredak integrala pritom nije bitan jer se vrijednost integrala ne mijenja ako "zamijenimo integrale". No, moramo pripaziti da donja granica jednoga integrala bude zapisana točno ispred gornje granice toga istoga integrala jer u suprotnom nećemo dobiti ispravan rezultat.

Napomena: Može se pokazati da je zadani dvostruki integral jednak

$$I = \frac{1}{3} \cdot (\operatorname{arsh} 1 + \sqrt{2}) = \frac{1}{3} \cdot \left[\ln\left(1 + \sqrt{2}\right) + \sqrt{2} \right] \approx 0.765195716464212691344766016396 5.$$

Konstanta $P = \operatorname{arsh} 1 + \sqrt{2} = \ln\left(1 + \sqrt{2}\right) + \sqrt{2} \approx$ 2.295587149392638074034298049189 5

obično se naziva *opća parabolična konstanta* i predstavlja svojevrstan analogon broja π za slučaj parabole[34].

[33] Više o Fubinijevu poučku i njegovim primjenama vidjeti npr. u knjizi Š. Ungar: *Matematička analiza u* $\mathbf{R}^n$, Golden-marketing – Tehnička knjiga, Zagreb, 2005.

[34] Više o općoj paraboličnoj konstanti može se naći u knjizi R.S.Finch: *Mathematical Constants*,Cambridge University Press, 2003.

TEHNIČKO VELEUČILIŠTE U ZAGREBU

POLYTECHNICUM ZAGRABIENSE

ELEKTROTEHNIČKI ODJEL

Matematički alati u elektrotehnici

10.6. Zadatci za vježbu

Izračunajte približne vrijednosti sljedećih određenih integrala pomoću trapezne formule, Simpsonove formule (uzmite n = 2000) i funkcije `quadl`, usporedite ih s točnim vrijednostima, te ocijenite apsolutne i relativne vrijednosti pogrješke aproksimacije:

1. $\int_1^2 (1+x-\ln x)\cdot dx$.

2. $\int_0^{\frac{\pi}{2}} \frac{\sin x}{\cos^3 x}\cdot dx$.

3. $\int_0^1 \frac{1}{x^2-5\cdot x+6}\cdot dx$.

4. $\int_e^{e^2} \frac{\ln(\ln x)}{x}\cdot dx$.

5. $\int_{\ln 2}^{\ln 3} x^7\cdot e^{x^4}\cdot dx.$

6. $\int_0^1 e^{-\sqrt{x}}\cdot dx$.

7. $\int_0^{\frac{\pi}{2}} \sin^2 x\cdot\cos^2 x\cdot dx.$

8. $\int_0^1 \ln\left(x+\sqrt{1+x^2}\right)\cdot dx$.

9. $\int_0^1 \frac{e^{\operatorname{arcctg} x}-x\cdot\ln(1+x^2)-1}{1+x^2}\cdot dx$.

10. $\int_{-1}^1 |2\cdot x|\cdot dx.$

11. S točnošću od 10^{-6} približno izračunajte sljedeće određene integrale:

a) $\int_1^2 \sin(x^2)\cdot e^{-\log x}\cdot dx$,

b) $\int_0^1 \sqrt[3]{\ln(1+x^2)-\sqrt{1-x^2}}\cdot dx,$

c) $\int_2^3 \frac{x+\sin^2 x}{x^2+\cos^2 x}\cdot dx$,

d) $\int_1^3 \ln\frac{\sqrt{e^{2\cdot x}-e^x+1}}{\sin\left(\frac{\pi}{2}\cdot x^3\right)}\cdot dx.$

12. Približno izračunajte dvostruki integral

$$\iint_D \sqrt[3]{x^2+x\cdot y+y^2}\cdot dx\cdot dy$$

ako je D područje omeđeno krivuljom

$$4\cdot x^2+3\cdot y^2=48.$$

11. LITERATURA

1. S. Kurepa: *Matematička analiza* (1. dio), Školska knjiga, Zagreb, 1997.
2. S. Kurepa: *Matematička analiza* (2. dio), Tehnička knjiga, Zagreb, 1991.
3. B. P. Demidovič i suradnici: *Zadatci i riješeni primjeri iz matematičke analize za tehničke fakultete*, Golden marketing – Tehnička knjiga, Zagreb, 2003.
4. V. P. Minorski: *Zbirka zadataka iz više matematike*, Tehnička knjiga, Zagreb, 1971.
5. I. Ivanšić: *Numerička matematika*, Element, Zagreb, 2003.
6. S. Suljagić: *Vjerojatnost i statistika*, skripta, Tehničko veleučilište, Zagreb, 2003.
7. I. Vuković: *Matematika* 1, udžbenik za stručni studij elektrotehnike, Tehničko veleučilište u Zagrebu, Zagreb, 2011.
8. I. Vuković: *Matematika* 2, udžbenik za stručni studij elektrotehnike, Tehničko veleučilište u Zagrebu, Zagreb, 2011.
9. B. Kovačić: *Poslovna statistika*, skripta, Visoka poslovna škola PAR, Rijeka, 2012.
10. I. Vuković: *Osnove numeričke matematike*, udžbenik za stručni studij elektrotehnike, Tehničko veleučilište u Zagrebu, Zagreb, 2011.
11. *MATLAB Documentation* – Version 7., The MathWorks Inc., Natick, 2007.
12. M. Vrdoljak: *Uvod u MATLAB* (dostupno na: http://titan.fsb.hr/~mvrdolja/matlab)
13. R. L. Spencer, M. Ware: *Introduction to MATLAB*, Brigham Young University, 2011.

KAZALO POJMOVA