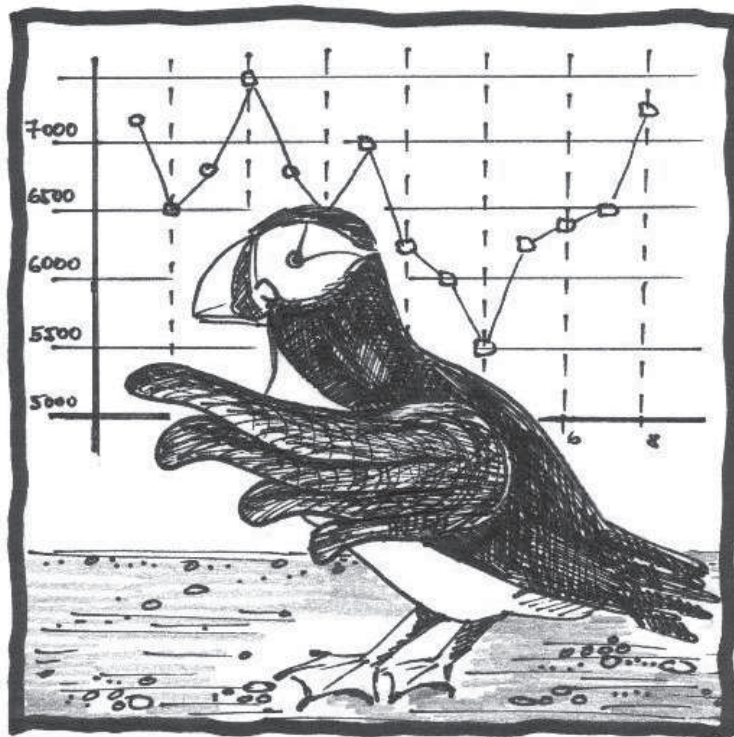


Poglavje 15

SCILAB - numerično reševanje in analiziranje podatkov



Numerično reševanje problemov s programiranjem
Kvantitativne metode napovedovanja
Primer: napoved prodaje vozil

15.1 Teoretično ozadje

15.1.1 Napovedovanje povpraševanja

Z napovedovanjem ocenjujemo verjetne razvoje v prihodnosti. Planiranje ne oznanja samo verjetnega, ampak na osnovi verjetnega postavlja tisto, kar je zaželeno. Pogosto se s pojmom "napovedovanje povpraševanja" pojavljata dva pojma, in sicer napovedovanje in predvidevanje. Pri čemer naj bi pojem "napovedovanje" ustrezal angleškemu izrazu *forecasting*, "predvidevanje" pa izrazu *prognosis*. Po pregledu različne literature ugotavljamo, da avtorji skoraj povsod govorijo o obeh izrazih, kot o enem samem.

Glede na časovni horizont za katerega napovedujemo povpraševanja ločimo dolgoročno, srednjeročno in kratkoročno napovedovanje. Za dolgoročno napovedovanje je značilno, da poteka daljše časovno obdobje in se praviloma izvaja za skupine proizvodov. Na podlagi tega napovedovanja se odločamo o fiksnih zmogljivostih. Srednjeročno napovedovanje je napovedovanje, ki zavzema obdobje nekje od 6 do 18 mesecev. Na podlagi tega poskušamo ugotoviti predvsem sezonska nihanja na ravni skupin proizvodov, katerim nato prilagajamo proizvodnjo. Kratkoročno napovedovanje se izvaja za obdobje od nekaj tednov pa do nekaj mesecev in služi (predvsem) kot osnova operativnemu planiranju.

Glede na dani problem se odločimo za vrsto in temu primerno metodo napovedovanja. Metode v osnovi delimo na dve skupini, in sicer: kvalitativne in kvantitativne. Za kvalitativne metode je značilno, da so običajno subjektivne narave. Kvantitativne metode napovedovanja povpraševanja temeljijo na podatkih iz preteklosti in statističnih orodjih. Med kvantitativne metode uvrščamo dva ključna pristopa. Prvi pristop so t.i. ekstrapolacijske metode, ki predpostavljajo, da je gibanje nekega pojava prvenstveno funkcija časa. Drugi pristop, imenovan vzročne metode temelji na predpostavki, da je gibanje nekega pojava v korelaciji z mnogimi zunanjimi faktorji, izmed katerih je eden lahko tudi čas. Kvalitativne metode so uporabne predvsem za dolgoročne napovedi ali pa za napovedi, ko nimamo na voljo dovolj podatkov o preteklem gibanju pojavov. V to vrsto uvrščamo Delfi metodo, ki jo uporabljamo za dolgoročno napovedovanje, oceno menagementa osebja, oceno prodajnega osebja, anketiranje, tržne raziskave ipd.

15.1.2 Kvantitativne metode napovedovanja

V izbranem primeru se osredotočimo na kvantitativne metode napovedovanja, ki kot osnovo napovedi, uporabljajo matematične modele, ki so zasnovani na podatkih iz preteklosti. Te metode temeljijo na predpostavki, da gibanja v

prihodnosti lahko predvidimo na podlagi gibanj v preteklosti. Govorimo o treh najpogosteje uporabljenih metodah, katere v nadaljevanju prikažemo na konkretnem primeru:

- Regresija meri odvisnost dveh slučajnih spremenljivk – kakšen vpliv ima ena na drugo. Linearna regresija je metoda, s katero ob znani spremenljivki X napovemo, koliko bo Y . Y je odvisna spremenljivka, X pa neodvisna (imenovana tudi prediktor).
- Holtova linearna metoda se imenuje po statistiku Holtu. Metoda uporablja dva faktorja glajenja - Alfa, ki predstavlja eksponentno glajenje in Beta, ki predstavlja trend v opazovanem obdobju. Obe konstanti se nahajata v razponu med 0 in 1. Praksa je pokazala, da mora biti vrednost Beta manjša od Alfe, Beta v razponu od 0,005 in 0,2 ter Alfa med 0,02 in 0,5.
- Brownova metoda eksponentnega glajenja je metoda pri kateri se upoštevajo vsi podatki, ki so na voljo, pri čemer se starejšim podatkom dodeli nižja utež. Gre za enokoračno metodo napovedovanja, kjer tvorimo napoved za en korak naprej v časovnem horizontu. Uporablja se v primerih, ko pri meritvah povpraševanja ni zaznati kakšnega spreminjanja trenda, niti ne cikličnih in sezonskih nihanj.

15.2 O programskem orodju

Scilab je programsko orodje za računanje numeričnih funkcij. Uporabno je za računanje z različnimi oblikami matematičnih zapisov in struktur z vsemi znanimi oblikami števil (z matrikami, polinomi, objekti ipd.). Z danim orodjem analiziramo linearne in nelinearne dinamične sisteme. Programsko orodje uporabimo kot pripomoček za numerično optimizacijo, kot sta npr. linearno in nelinearno programiranje. Razvili so ga na inštitutu INRIA v Franciji in je prosto odprto orodje. Prepozna funkcije kot podatkovne objekte, z njimi računa ali jih pretvori v druge podatkovne objekte. Funkcijo, ki je definirana znotraj Scilaba, lahko uporabljamo tudi kot vhodni ali izhodni argument druge funkcije. Podpira znakovne nize podatkovnega tipa, kar v posebnih pogojih omogoča sprotno tvorjenje funkcij.

Nekatere izmed možnosti Scilab so še: 2D in 3D grafika, animacija; polinomi, racionalne funkcije; linearna algebra, razpršene matrike; simulacija; statistika in mnoga druga področja. Deluje na številnih operacijskih sistemih (Unix, Linux, Windows ipd.). Vsebuje izvorno kodo, on-line pomoč in angleški uporabniški vodnik.



Slika 15.1: Uradna spletna stran Scilab

Prenos in namestitvev

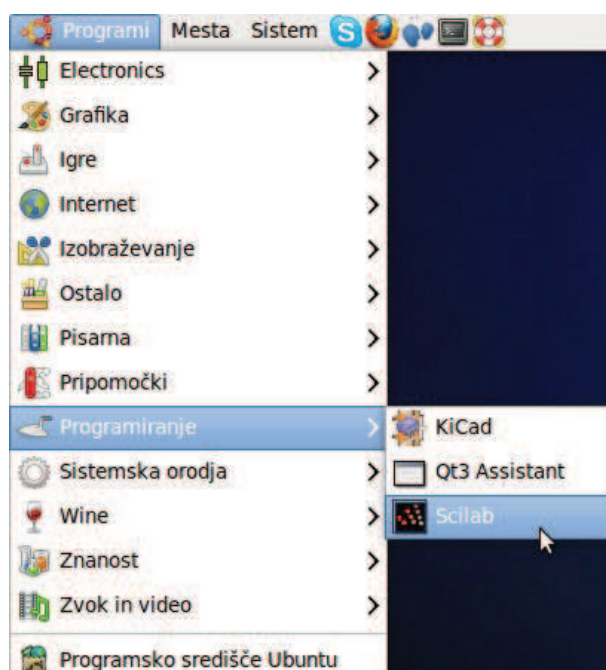
Programsko orodje Scilab namestimo na različne načine, odvisno od operacijskega sistema, ki ga uporabljamo. V okolju Windows programsko orodje namestimo iz uradne spletne strani Scilab [46] in ga nato brezplačno uporabljamo na lastnem računalniku, pri čemer moramo upoštevati avtorske pravice.

V operacijskem okolju Ubuntu programsko orodje namestimo po dveh različnih poteh, ki smo jih predhodno že opisali. Najenostavneje to naredimo z uporabo Synaptic orodja, kjer programsko orodje enostavno poiščemo, prenesemo in namestimo. Nameščen program nato odpremo v menijski vrstici v razdelku *Programiranje* (glej Sliko 15.2).

Program zaženemo z dvoklikom na ikono programa, pri čemer se odpreta dve okni. V konzolnem oknu (v nadaljevanju osnovno okno) se izvaja pomožni program Scilab, ki mora biti ves čas dostopen. V drugem oknu se prikaže vnosna vrstica, pred katero je zapisan simbol \rightarrow (glej Sliko 15.3).

Osnove za delo s programskim orodjem so predstavljene v *Introduction* v razdelku *Scilab Demonstration*. Ob zagonu programskega orodja se prikaže osnovno okno z zapisom menijske in orodne vrstice (glej Sliko 15.4).

Menijska vrstica je sestavljena iz 6 razdelkov. Razdelek *Datoteka (File)* omogoča zagon skriptne datoteke (Execute), odpretje že shranjene datoteke (*Open a file*), nalozitev shranjene spremenljivke (*Load environment*), shranitev



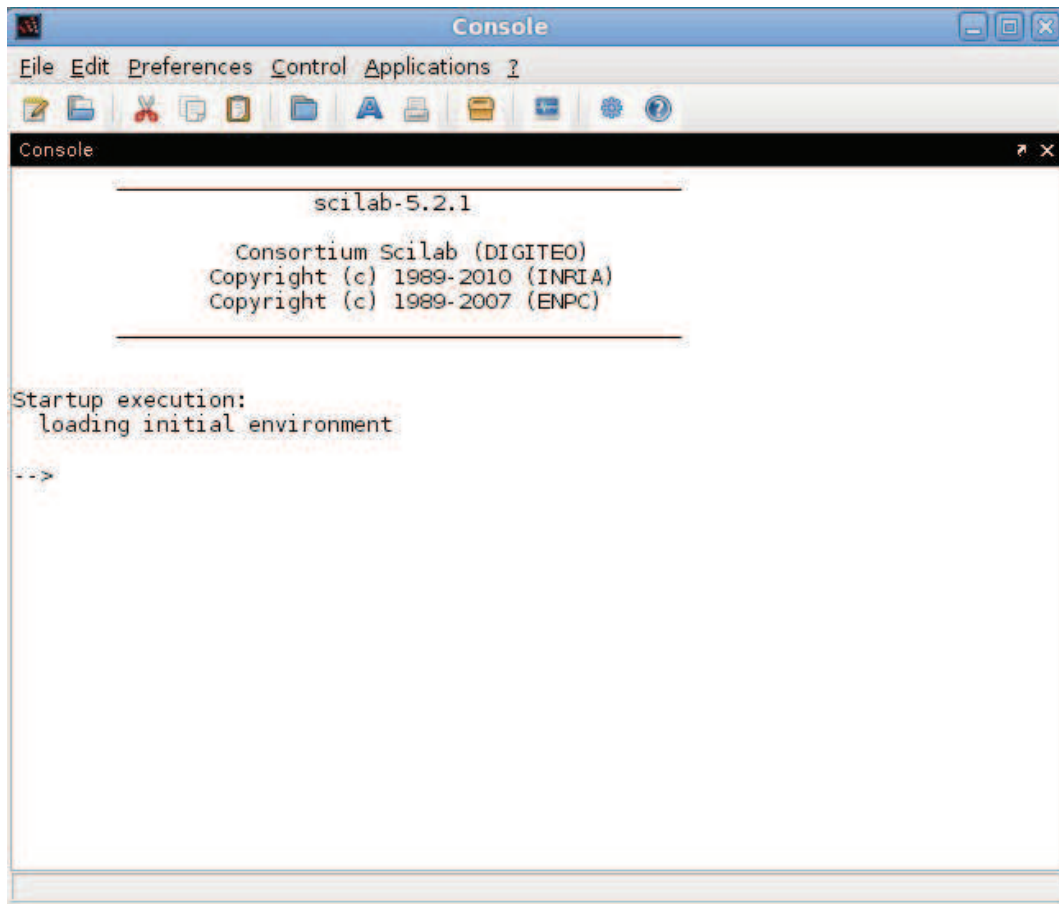
Slika 15.2: Odpiranje programskega orodja v Ubuntu

spremenljivke (*Save environment*), spremembo delovne mape (*Change current directory*), izpis delovne mape (*Display current directory*), tiskanje zapisanega programa (*Print*) in izhod iz programa (*Quit*). Razdelek *Uredi* (*Edit*) omogoča urejanje standardnih možnosti: rezanje (*Cut*), kopiranje (*Copy*), lepljenje podatkov iz drugih dokumentov (*Paste*) ipd. Razdelek *Preferences* ponuja oblikovne možnosti - poljubno lahko nastavimo barvo ozadja (*Colors*), velikost in vrsto pisave (*Font*). V primeru, ko imamo v okencu določene podatke, ki jih ne potrebujemo več, jih enostavno pobrišemo s *Clear History* (glej Sliko 15.5).

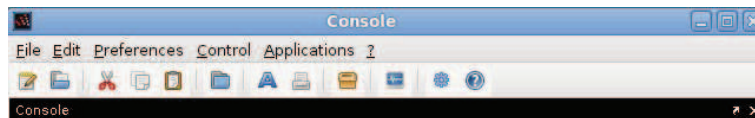
Razdelek *Kontrola* (*Control*) vsebuje določena kontrolna orodja in ponuja možnosti, kot so *Resume* (omogoča izvajanje programa po premoru), *Abort* (preneha z izvajanjem trenutnega programa) in *Interrupt* (prekine izvajanje trenutnega programa). Razdelek *Aplikacija* (*Application*) je pomemben predvsem zaradi možnosti *Editor*, s katero odpremo novo okno za pisanje programa, ki ga nato shranimo in izvozimo v osnovno programsko okno (glej Sliko 15.6).

Slika 15.7 prikazuje okno, ki je namenjeno zapisu programa, za izvoz v osnovno okno.

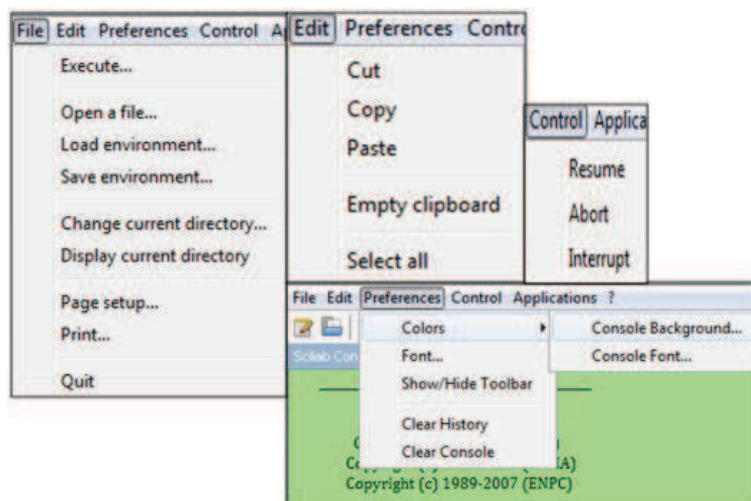
Razdelek *Xcos* uporabljamo za risanje blokovnih shem. Na voljo je večje število blokovnih shem ter prazno okno, namenjeno oblikovanju le teh (glej Sliko 15.8).



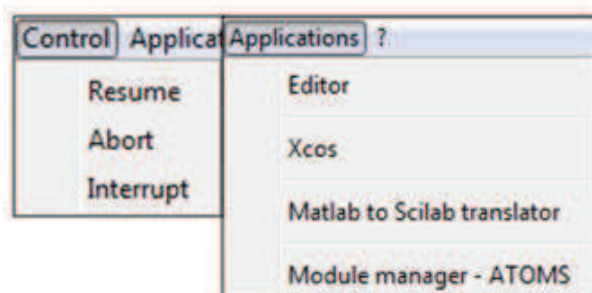
Slika 15.3: Zagon programa



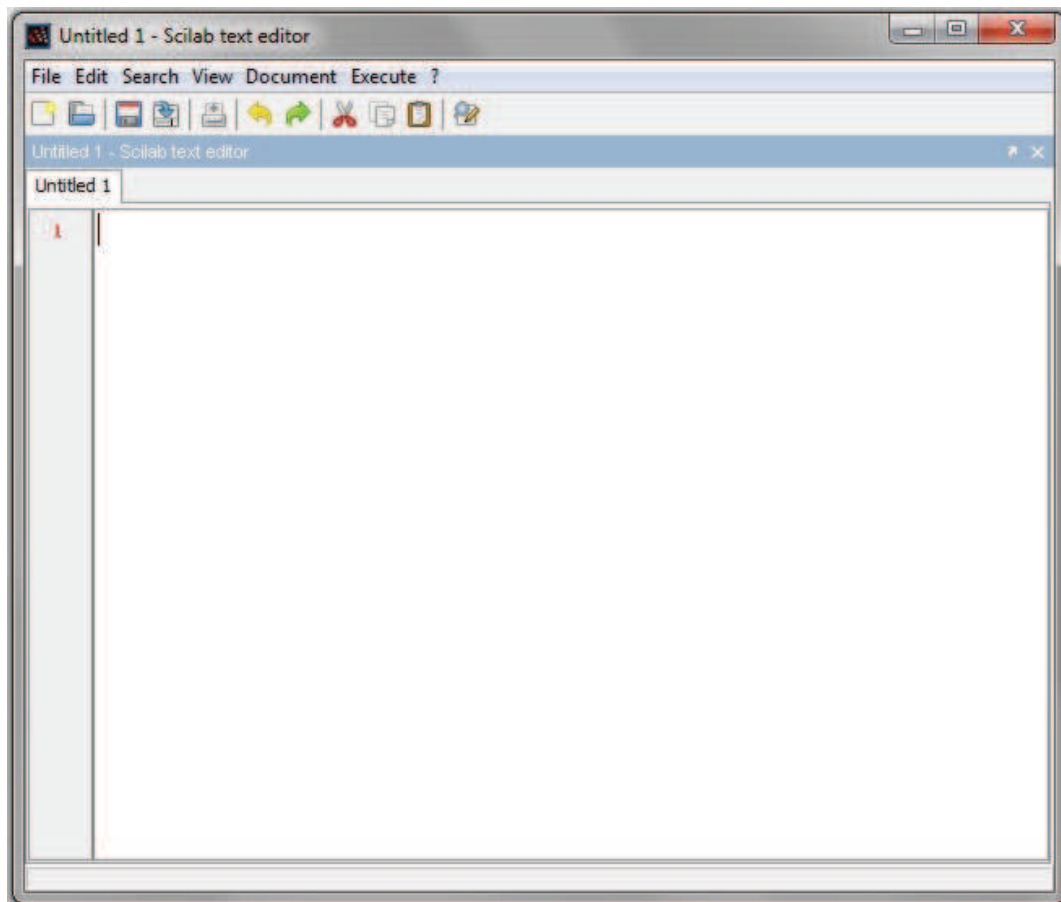
Slika 15.4: Menijska in orodna vrstica v Scilabu



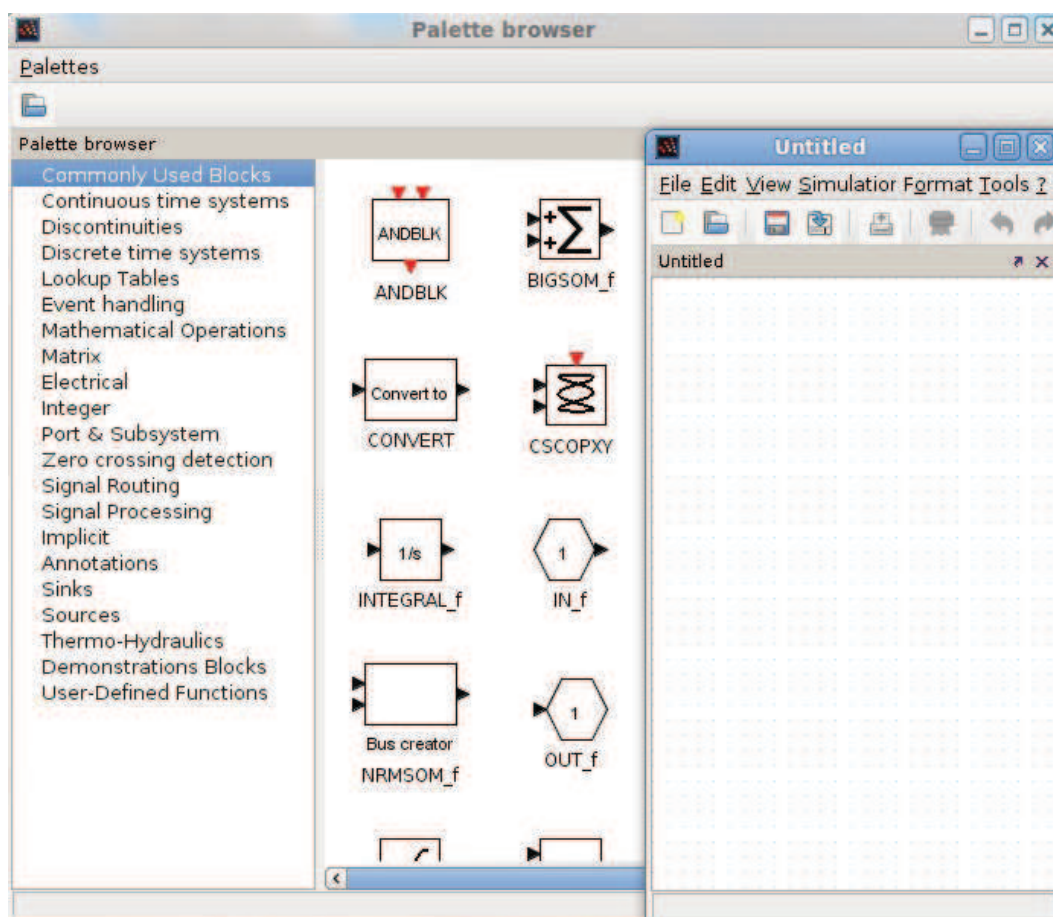
Slika 15.5: Menijska vrstica – 1. del



Slika 15.6: Menijska vrstica – 2. del



Slika 15.7: Okno za zapis programa



Slika 15.8: Razdelek Xcos

Ena izmed značilnih vrst blokov so bloki linearnih gradnikov, ki predstavljajo gradnike za opis linearnih dinamičnih sistemov. Nekateri izmed pomembnejših blokov linearnih gradnikov:

- *Sum* (seštevalnik) - blok, ki na izhodu da vsoto vseh vhodov. Bloku lahko določimo poljubno število vhodov, njihove uteži in njihov predznak. Poleg osnovnega bloka pravokotne oblike obstaja tudi inačica okrogle oblike. Pri slednji ne moremo spreminjati predznaka in uteži vhodov.
- *Integrator* - izhod tega gradnika je integral vhoda.
- *Numerator/Denominator* ($num(s)/den(s)$) - blok za predstavitev prenosne funkcije.
- *Gain* (ojačevalnik) uporabimo kadar moramo funkcijo množiti s konstanto. Na izhodu tako dobimo vhod z ojačenjem, ki ga določimo.

Uporabniki lahko v programskem orodju uporabljamo razdelek *Pomoč (?)*, ki omogoča pogled v opis posameznih blokov in ostalih funkcij s podrobnejšim opisom. Razdelek *?* omogoča pomoč pri uporabi programa (*Scilab Help*). *Scilab Demonstration* nudi možnost ogleda že izdelanih primerov programa. Dostopamo lahko tudi do njihove spletne strani (*Web links*), (*About Scilab*) pove več o programu. Preverimo lahko ali smo pravilno postavili blok, je potrebna kakršnakoli sprememba ipd. (glej Sliko 15.9).

Slika 15.10 prikazuje menijsko vrstico z vsemi vsebovanimi razdelki.

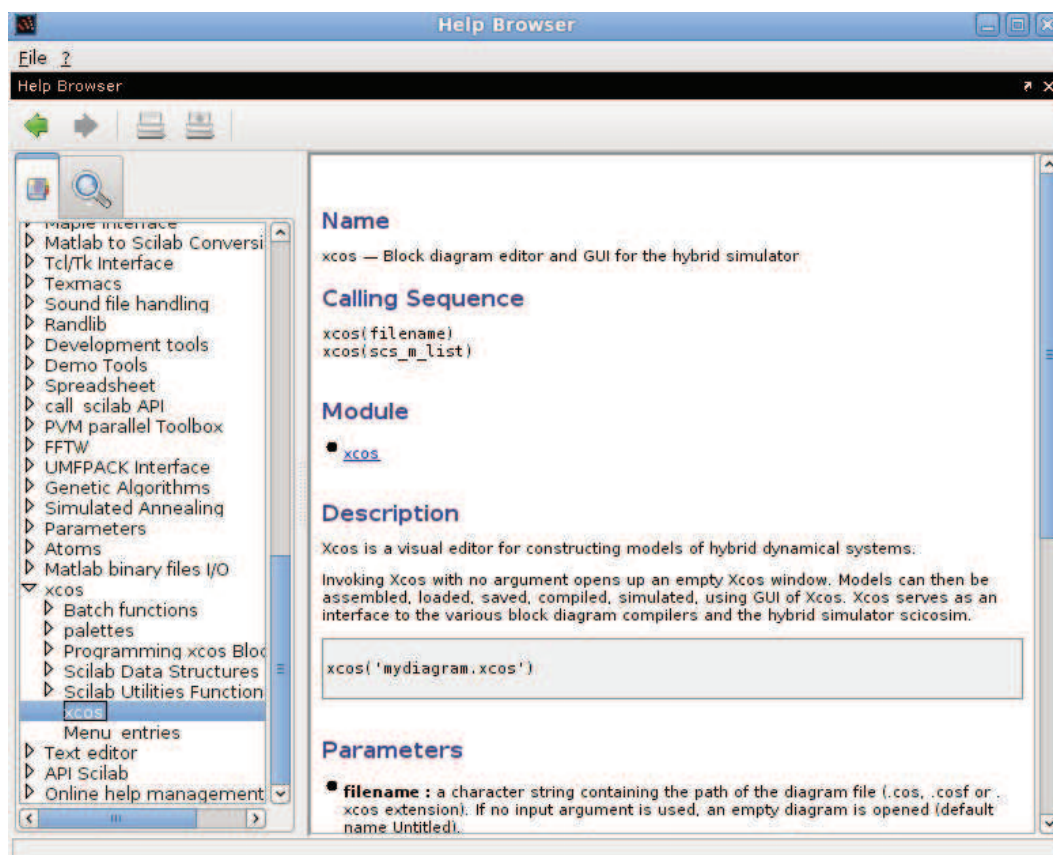
Scilab Demonstrations prikazuje nekaj izmed že zapisanih programov z rešitvami. Le-te lahko preoblikujemo glede na izbran problem in jih smiselno uporabimo (glej Sliko 15.11).

Orodno vrstico sestavlja 12 ikon. Glej Tabelo 15.1 in Sliko 15.12.

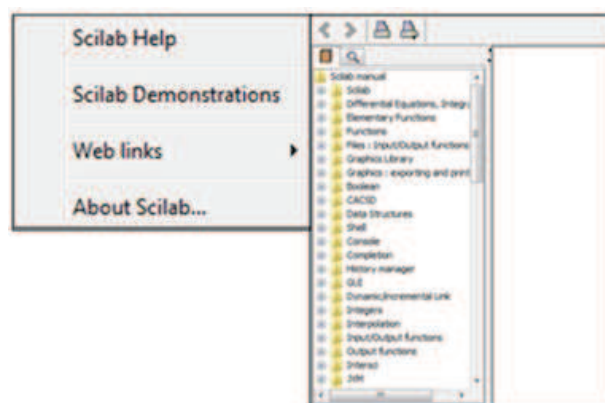
Scilab prepozna veliko podatkovnih tipov. Skalarni objekti so konstante, polinomi, znakovni nizi in racionalna števila.

Spremenljivke in osnovne računske operacije

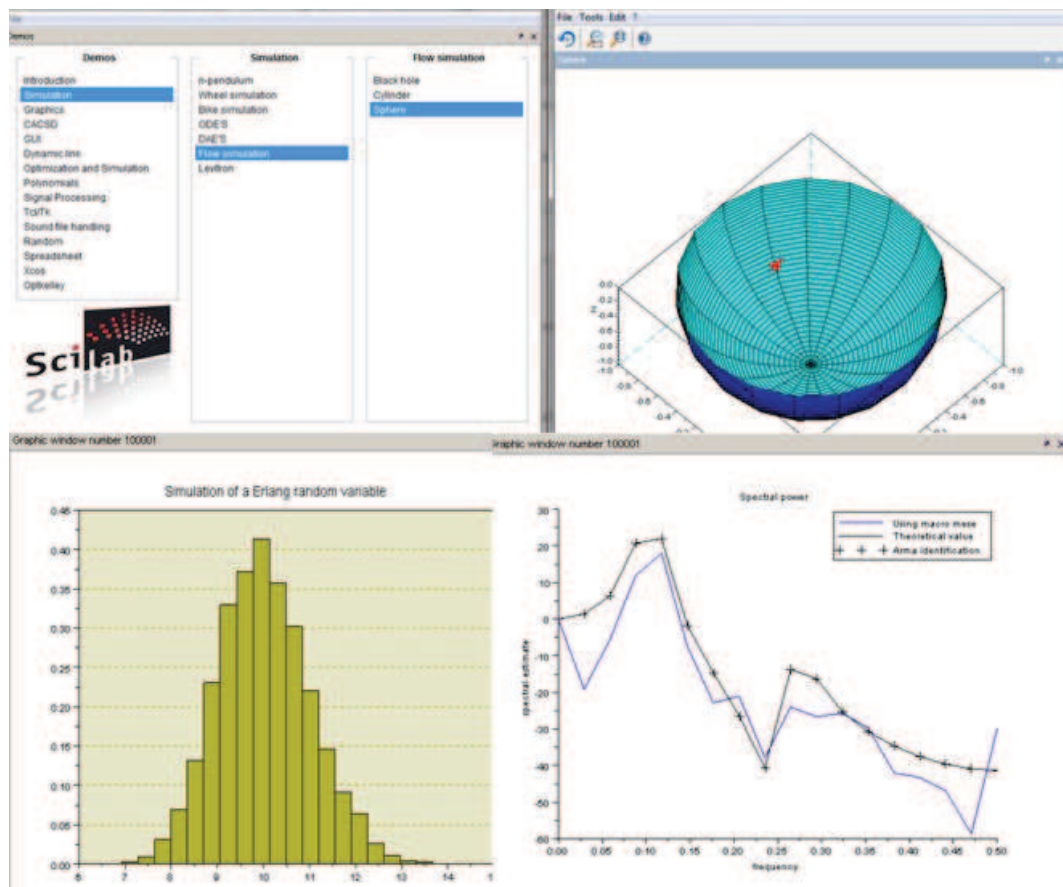
Programsko orodje je občutljivo na velike in male črke - b in B sta dve različni spremenljivki. V kolikor želimo izpis neke spremenljivke v matrični obliki, vpišemo ime pod katerim je shranjena. Na začetku spremenljivkama b in B določimo vrednosti ter za vsakim zapisom navedemo znak *;*, ki pomeni neizpis rezultata. Če znaka ni, se rezultati izpišejo. S simbolom *//* označimo komentarje.



Slika 15.9: Razdelek Pomoč



Slika 15.10: Menijska vrstica – 3. del



Slika 15.11: Demo programi



Slika 15.12: Orodna vrstica

Ikone orodne vrstice	
1	Odpiranje okenca za zapis programa
2	Shranjevanje dokumenta
3, 4, 5	Orodja za rezanje, kopiranje in lepljenje dokumentov
6	Spreminjanje delovne mape
7	Spreminjanje oblike in velikosti pisave
8	Tiskanje dokumenta
9	Namestitev posameznih modulov
10	Xcos za risanje blokovnih shem
11	Demo programi
12	Pomoč

Tabela 15.1: Ikone orodne vrstice

Matrike

Za vpisovanje matrik ali vektorjev upoštevamo nekaj osnovnih pravil. V vrstico matrike ali vektorja vpišemo število z vmesnim presledkom ali vejico. V novo vrstico matrike ali vektorja se pomaknemo s podpičjem ter celoten zapis matrike obdelamo z oglatimi oklepaji. Ko definiramo matriko, jo Scilab shrani pod oznako, s katero smo jo poimenovali in jo z vpisom oznake kadarkoli prikličemo - v danem primeru velika tiskana črka D.

Zapis spremenljivk in matrik

```
-->a=1
a =
  1.

-->A=2
A =
  2.

-->a+A
ans =
  3.

-->D=[16 3 2 1
-->5 10 11 8
-->9 6 7 12
-->4 15 15 1]
```

D =

16.	3.	2.	1.
5.	10.	11.	8.
9.	6.	7.	12.
4.	15.	15.	1.

Funkcije

Scilab ima veliko število vgrajenih funkcij: *srqt*, *sin*, *exp* ipd. Nekatere izmed vgrajenih funkcij so osnovne (*sum*, *prod*, *sqrt*, *diag*, *cos*, *max*, *round*, *sign*, *fft*), urejevalne (*sort*, *gsort*, *find*), za posebne matrike (*zeros*, *eye*, *ones*, *matrix*, *empty*), linearna algebra (*det*, *inv*, *gr* . . .), nizi znakov (*string*, *part*, *evstr* . . .), diferencialne enačbe (*ode*, *dassl*, *dassrt*, *odedc*) itd.

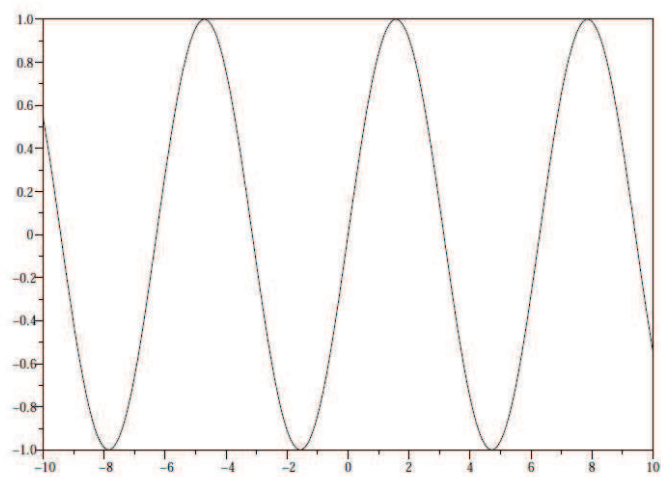
Grafi

S Scilab izrišemo številne grafe matrik, vektorjev in drugih funkcij v 2D in 3D, na različne načine in v obliki animiranega filma. Odprtih imamo lahko več grafičnih oken hkrati, vendar je v vsakem trenutku lahko aktivno le eno okno 15.13.

Programiranje v Scilab

Pogosto se v praksi srečujemo s problemi, ki jih ne moremo razrešiti na enostaven način, saj je predhodno potrebno zapisati določen računalniški program oz. model. Za izdelavo modela v programskem orodju Scilab je potrebna pripravljenost za raziskovanje. S pomočjo uporabe dodatnih priročnikov danega programa je programiranje veliko enostavneje.

```
-->y=sin(x);  
-->plot2d(x,y)
```



Slika 15.13: Graf funkcije narisane v Scilab

Leto	Nakup novih osebnih vozil	Leto	Nakup novih osebnih vozil
1999	77.217	2005	60.531
2000	55.120	2006	59.885
2001	52.316	2007	67.791
2002	50.859	2008	71.037
2003	59.881	2009	57.391
2004	62.697		

Tabela 15.2: Statistični podatki prodaje v obdobju 1999-2009

Problem

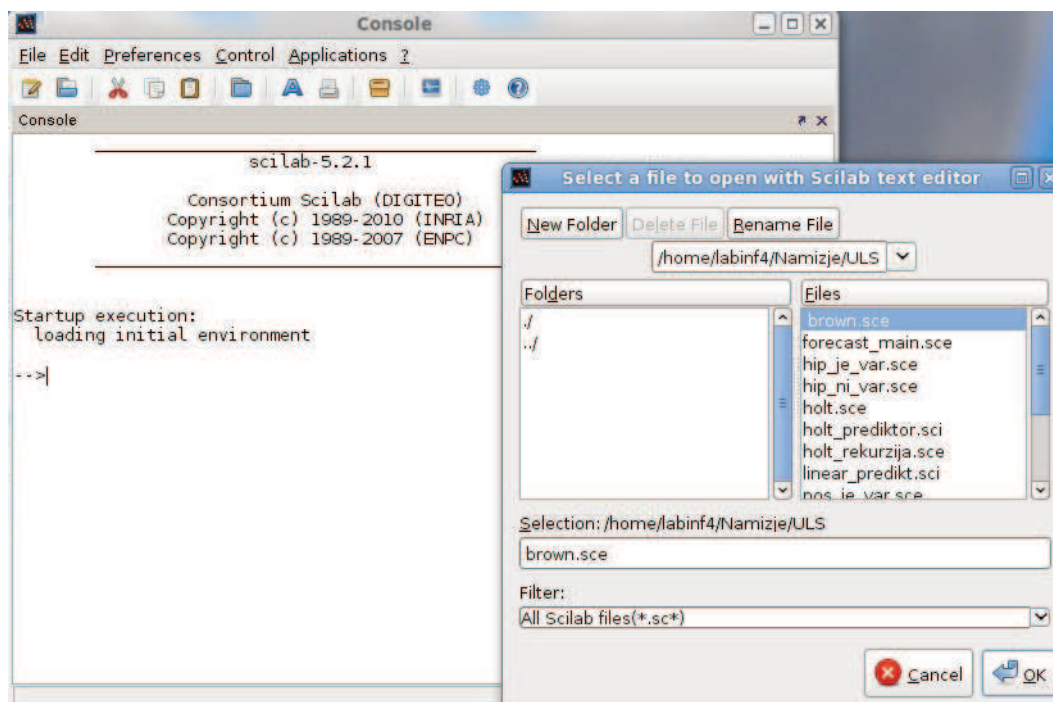
Podjetje OpenStorage se ukvarja s skladiščenjem treh vrst komponent (pnevmatik, platišč in vijakov), namenjenih za izdelavo osebnega avtomobila. Na podlagi zbranih podatkov o nakupu v preteklih 11 letih izvedemo napoved prodaje avtomobilov za obdobje prihodnjih 5 let. Ob zapisu danega programa in dobljenih rezultatih bomo preverimo ali je izbrana metoda pravilna oz. ali jo je za nadaljnjo delo potrebno spremeniti. Statistični podatki prodaje v obdobju 1999-2009 so podane v Tabeli 15.2.

Problem poskušamo rešiti na podlagi izdelanega modela linearne regresije, Holt in Brown metode. Dani program, že zapisan v programskem orodju Scilab pridobimo iz učnega gradiva dr. Dejana Dragana [80], ki program uporablja, kot študijsko gradivo na fakulteti.

15.3 Uporaba

Za razrešitev danega problema je v prvi fazi potreben zapis modela v Scilab. Prikažemo problem in njegovo rešitev na že izdelanih programih z Holt metodo (*holt.sce*), Brown metodo (*brown.sce*) in linearno regresijo (*regresija.sce*). Program je prikazan v Prilogi [80]. Matematičen model lahko zapišemo tudi sami, vendar je potrebno vložiti precej časa in truda. Na takšen način izvedemo marsikatero optimizacijo in napovedovanje. Zapisani programi delujejo znotraj skupnega programa imenovanega *forecast_main*, ki ga dopolnimo z izbranimi podatki. V Scilab odpremo osnovno okno in v razdelku *File* izberemo možnost *Open a file*. Izberemo program npr. z Brown metodo in ga s klikom na razdelek *Open* odpremo v novem oknu (glej Sliko 15.14).

Ko je program odprt, ga je potrebno izvoziti v osnovno okno, kar storimo



Slika 15.14: Odpiranje datoteke

tako, da v meniju *Execute* izberemo možnost *Excute File Into Scilab*. Postopek izvedemo še za vse ostale tri datoteke.

Zapis programa

```
//
// brown.m
//
//

function brown(d,alfa)

fakt = 5;

N = length(d);

t = [1:1:N];

// izhodi prediktor-filtra (projekcija prihodnosti za en korak naprej -
// napoved kolicine za cas t+1 ob casu t):
// (relevantno je k = 2:N)

p = [];
p(1) = d(1)
p(2) = d(1)

for k=3:N
    p(k) = alfa*d(k-1) + (1-alfa)*p(k-1)
end;
.....
```

Pozorni moramo biti predvsem pri datoteki *forecast_main*, kjer v program vstavimo podatke. V izbranem primeru vstavimo podatke namišljenega podjetja, ki se ukvarja s skladiščenjem komponent za izdelavo avtomobilov. Podatki prikazujejo število prodanih vozil v letih od 1999 do 2009. Ko izvozimo vse datoteke v osnovno okno zapišemo pogoj *forecast_main* ter pritisnemo *Enter*.

Klicanje programa

```
-->exec('C:\Users\Lab4\Desktop\ULS\brown.sce', -1)

-->exec('C:\Users\Lab4\Desktop\ULS\holt.sce', -1)
```

```
-->exec('C:\Users\Lab4\Desktop\ULS\regresija.sce', -1)
-->exec('C:\Users\Lab4\Desktop\ULS\forecast_main.sce', -1)
-->forecast_main
```

Program vpraša kako naj izvede povpraševanje: ročno (vpis naših podatkov), po "defaultu" (sam izbere vrednosti). Odločimo se za ročno povpraševanje. Ko program vpraša po podatkih, zapišemo podatke, ki so na voljo (prodaja novih avtomobilov od leta 1999 do 2009). Program vpraša še po metodi po kateri se bo izvedlo povpraševanje. Izberemo regresijsko metodo (lahko pa se odločimo za katero izmed preostalih) in določimo za koliko let izvajamo predikcijo. V danem primeru izberemo vrednost 5, ki pomeni predikcijo za naslednjih 5 let.

```
Povprasevanje rocno (1), default(2), nakljucno (3)1
povprasevanje d= ? npr. [7 9 12 14 12 30 8 12 14 15 25 40][77217 55120 52316
50859 59881 62697 60531 59885 67791 71037 57391]
regresija, mnk(1)/brown(2)/holt(3)1
Za koliko let predikcija =5
```

Ob vnosu izbranih podatkov za prikaz rezultatov, pritisnemo tipko *Enter* (glej Sliko 15.15).

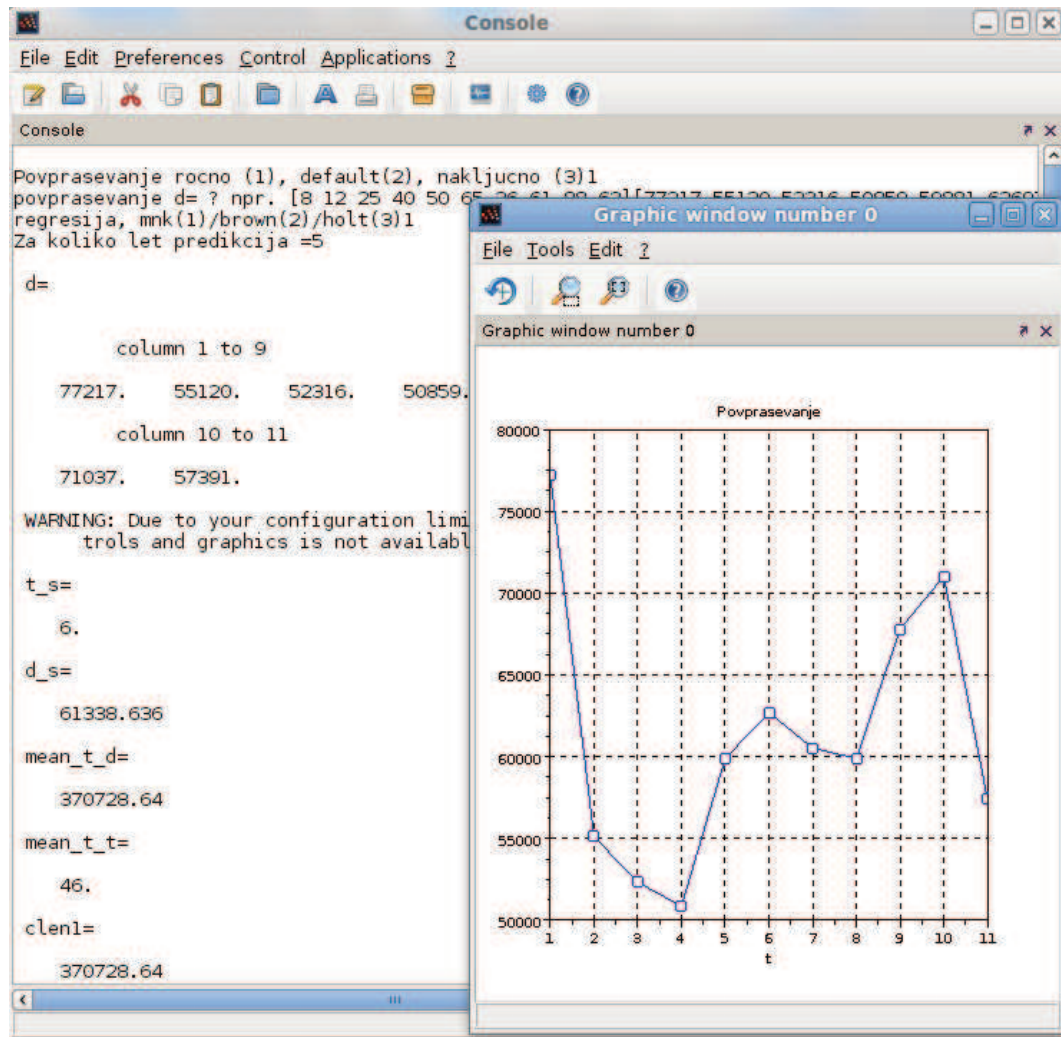
Program vpraša ali nadaljuje z izpisom rezultatov, za potrditev pritisnemo na tipko *Y*.

```
d_s=
61338.636

mean_t_d=
370728.64

mean_t_t=
46.

clen1=      370728.64
clen2=      368031.82
```



Slika 15.15: Izpis rezultatov

Razlaga rezultatov:

d - število podatkov = 77217 55120 52316 50859 59881 62697 60531 59885 67791
71037 57391

t_s - povprečen čas = 6

$$\bar{t} = \frac{1}{N} \sum t$$

d_s - povprečna vrednost = 61338,636

$$\bar{d} = \frac{1}{N} \sum d(t)$$

mean_t_d = 370728,64 - če bi računali analitično je to vrednost

$$\frac{1}{N} \sum t^2 \cdot d(t)$$

mean_t_t = 46 - če bi računali analitično je to vrednost

$$\frac{1}{N} \sum t^2$$

clen1 = 370728,64

clen2 = 368031,82 - če bi računali analitično je to vrednost

$$\bar{t} \cdot \bar{d}$$

clen3 = mean_t_t = 46

clen4 = 36 - če bi računali analitično je to vrednost

$$\bar{t}^2$$

a = 269,68182 - če bi računali analitično je to vrednost parametra

$$\hat{a} = \frac{\frac{1}{N} \sum t \cdot d(t) - \bar{t} \cdot \bar{d}}{\frac{1}{N} \sum t^2 - \bar{t}^2}$$

b = 59729,545 - če bi računali analitično je to vrednost parametra

$$\hat{b} = \bar{d} - \hat{a} \cdot \bar{t}$$

d_o - ocena modela za prvih 11 let = 59990,27 60259,909 60529,591... 61878

e - izračunani pogreški za prvih 1 let = 17226,773 -5139,9091 -8213,5909...
-5296,0455

e_sr = 1,984D-12

VAR - varianca = 63203478

$$VAR(e) = \frac{1}{N-1} \sum [e(t) - \bar{e}]^2$$

stde - standardni odklon = 7950,0615

$$STD(e) = \sqrt{VAR(e)}$$

d_o_pred - napoved povpraševanja za naslednjih 5 let = 62956,727 63226,
409... 64035,455

d_o_pred_zg - največje predvideno povpraševanje = 78856,85 79126,532...
79935,578 (če upoštevamo 2-kratno standardno deviacijo)

d_o_pred_sp - najmanjše predvideno povpraševanje = 47056,604
47326,286...
48135,332 (če upoštevamo 2-kratno standardno deviacijo)

Rezultati (Scilab izpis)

```
clen3=
  46.
clen4=
  36.
a=
  269.68182
b=
  59720.545

d_o=
```

```

column 1 to 4
  59990.227    60259.909    60529.591    60799.273
column 5 to 8
  61068.955    61338.636    61608.318    61878.
column 9 to 11
  62147.682    62417.364    62687.045

e=
column 1 to 4
  17226.773 - 5139.9091 - 8213.5909 - 9940.2727
column 5 to 9
  - 1187.9545  1358.3636 - 1077.3182 - 1993.    5643.3182
column 10 to 11
  8619.6364 - 5296.0455

e_sr=
  1.984D-12

VAR=
  63203478.

stde=
  7950.0615

d_o_pred=

column 1 to 4
  62956.727    63226.409    63496.091    63765.773
column 5
  64035.455

d_o_pred_zg=

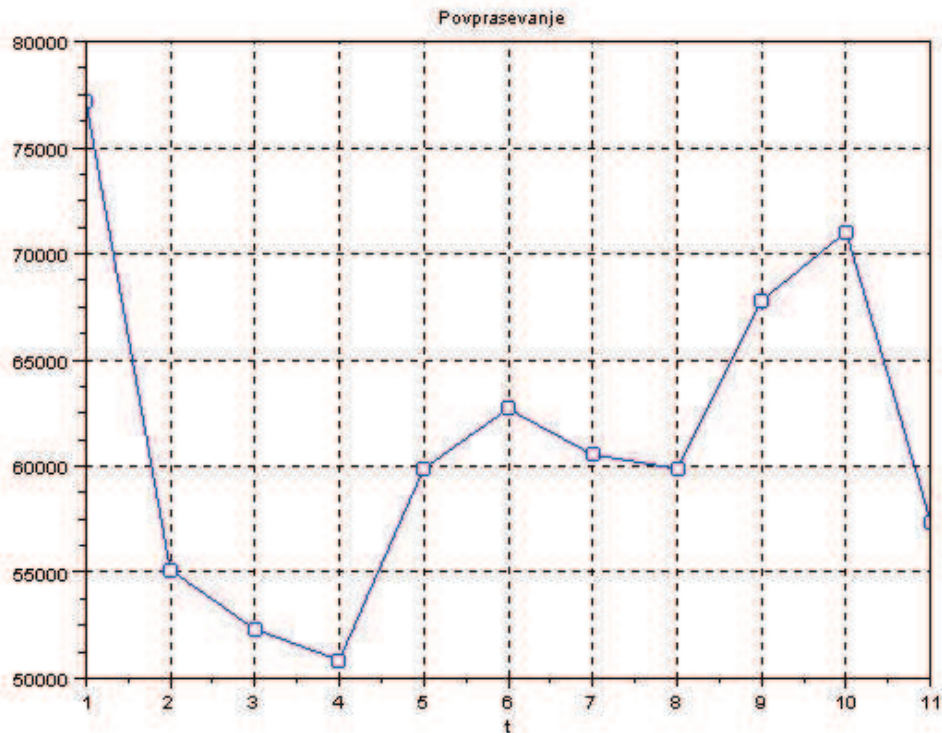
column 1 to 4
  78856.85    79126.532    79396.214    79665.896
column 5
  79935.578

d_o_pred_sp=

column 1 to 4
  47056.604    47326.286    47595.968    47865.65
column 5
  48135.332

```

Izračun pogreška podaja vrednosti odstopanja od linije regresijske premice, uporabimo pa tudi izračun standardne deviacije (STD), s pomočjo katere



Slika 15.16: Povpraševanje za sedanje obdobje

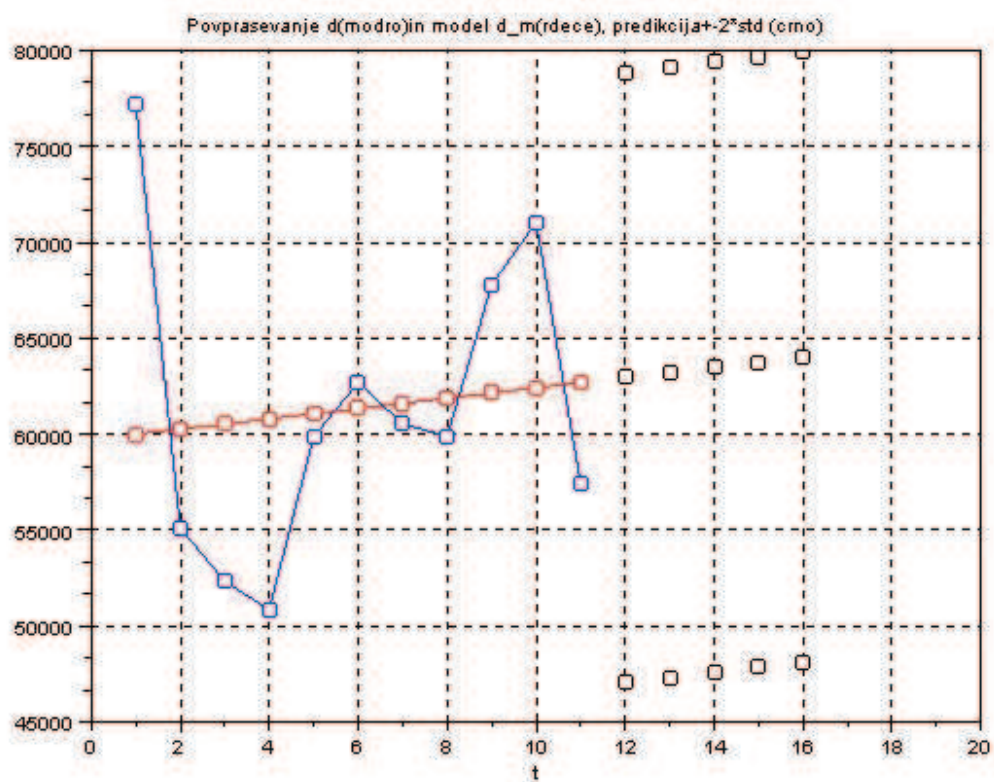
izračunamo povprečni odklon tako, da dobimo minimalno in maksimalno vrednost napovedi posameznega leta.

Program izriše dva grafa. Slika 15.16 prikazuje graf povpraševanja za sedanje obdobje, na podlagi podatkov iz preteklih obdobj.

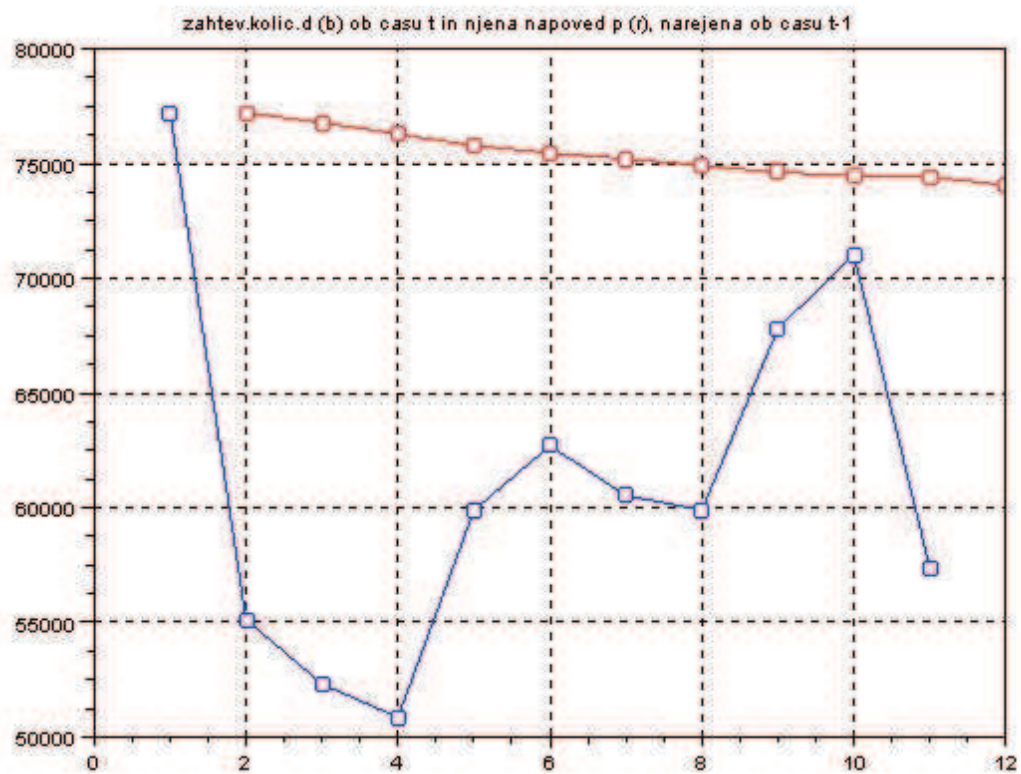
Sledi graf, ki prikazuje povpraševanje (modra barva), oceno modela d_m (rdeča barva) ter predikcijo z upoštevanjem 2-kratnega standardnega odklona (črna barva), za naslednjih 5 let (glej Sliko 15.17).

Iz grafa je razvidno, da se ocena modela razlikuje od realnih podatkov, zato kot primer prikazemo še rezultate, ki jih dobimo z uporabo Holt metode. Rezultati te metode prikazujejo še večje odstopanje, kar pomeni, da je za dani primer metoda še manj primerna kot linearna regresija (glej Sliko 15.18).

S Scilab prikazemo model napovedovanja povpraševanja nakupa novih vozil za naslednjih 5 let, pri čemer upoštevamo podatke preteklih 11 let. Iz rezultatov je razvidno, da je model linearne regresije primernejši od Holt metode, čeprav menimo, da nihče izmed njiju ni optimalen. Za reševanje kompleksnejših



Slika 15.17: Povpraševanje, model in predikcija z upoštevanjem standardnega odklona – regresija



Slika 15.18: Povpraševanje, model in predikcija z upoštevanjem standardnega odklona - Holt

problemov je vsekakor potrebno poglobljeno reševanje problema s pomočjo različnih priročnikov, navodil in dodatne literature. Preučiti in zapisati je potrebno model za nelinearne modele značilne v praksi. Potrebno je zapisati pogoje, ki bodo to tudi izpolnjevali.

Povzetek

Scilab je programsko orodje za računanje numeričnih funkcij. Uporabljamo ga za računanje z različnimi oblikami matematičnih zapisov in struktur. Z danim programom lahko analiziramo linearne in nelinearne dinamične sisteme. Prav tako programsko orodje uporabimo kot pripomoček za numerično optimizacijo, kot sta linearno in nelinearno programiranje.

S pomočjo izdelanega modela in zbranih statističnih podatkov preteklih 11 let, prikažemo model napovedovanja prodaje vozil prihodnjih 5 let. Pridobimo že izdelan program linearnega napovedovanja. Mnogokrat je za razrešitev kompleksnejših problemov potrebno izdelati samostojen program, vendar je samo programiranje zahtevno in vključuje bistveno več časa in truda.

S to vrstnimi programi si lahko pomagamo pri reševanju številnih problemov, v tako v praksi, kakor pri študiju. V praksi je poznan plačljiv program Matlab, ki vsebuje podobne funkcije kakor Scilab. Za namene študija je Scilab dovolj zanesljiv, da se naučimo osnove programiranja in reševanja določenih logističnih problemov.

Pri opisu programskega orodja Scilab smo uporabili še dodatne vire in literaturo: [109] [85] [10] [102] [84] [47] [48] [110] [81].