

Matlab Tutorial

Inteligentni sistemi
Vaje 2012/13, cikel 1
Emil Plesnik

Vsebina

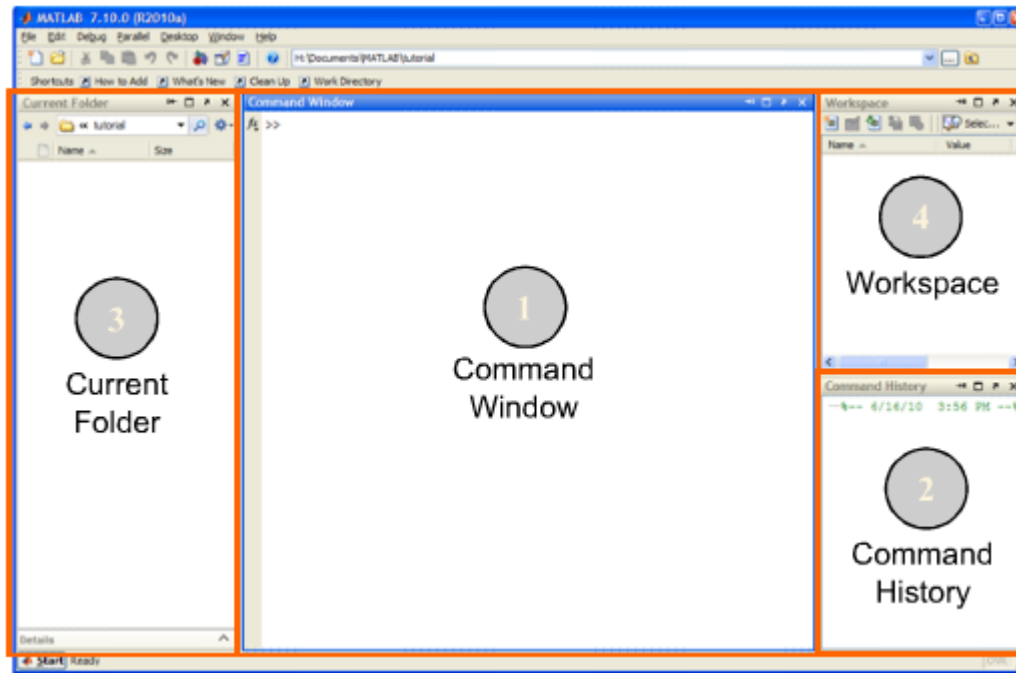
- Uvod
- Matlab okolje
- Matlab osnove
- Reševanje problemov z Matlabom
- Programiranje v Matlabu

Uvod

- Zmogljiv vsestranski grafični kalkulator
- Številna orodja (Toolboxes), knjižnice z vgrajenimi funkcijami
- Obdelava velikih zbirk podatkov
- Kompleksne numerične kalkulacije
- Vizualizacija podatkov z različnimi grafi

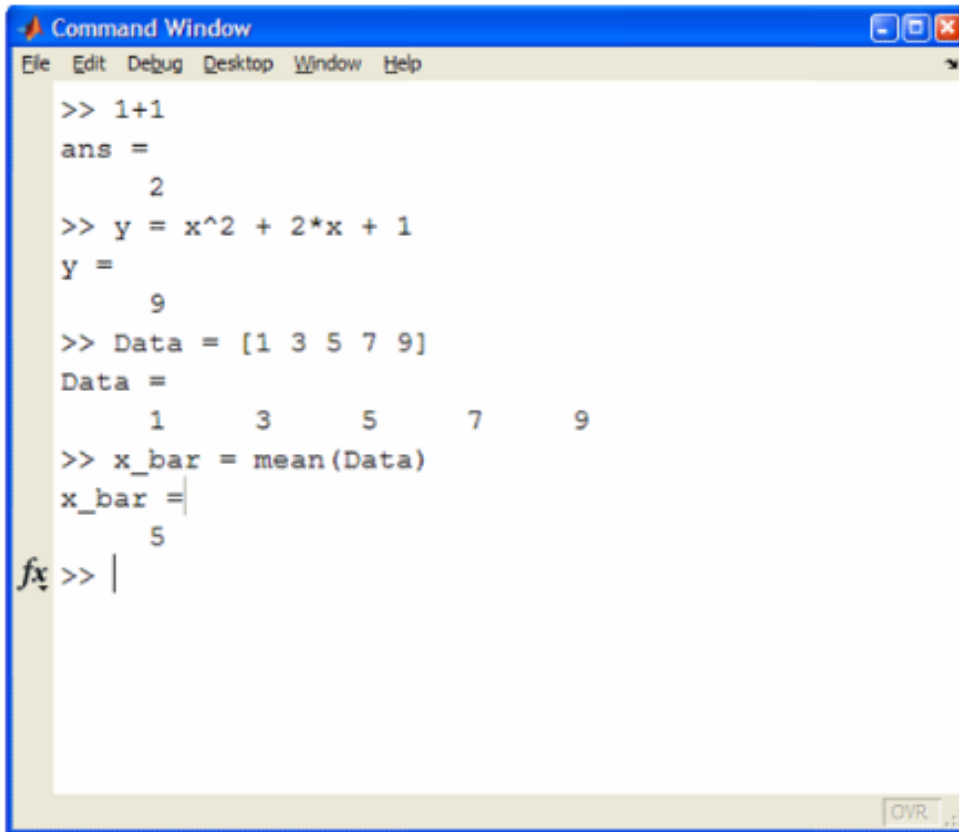
Matlab okolje

1. Ukazno okno (Command window)
2. Zgodovina (Command history)
3. Brskalnik (Current folder browser)
4. Delovno okolje (Workspace)

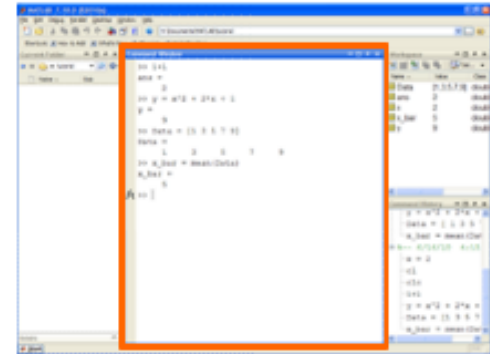


Ukazno okno (Command window)

- Osrednje okno programa
- Vnos ukazov



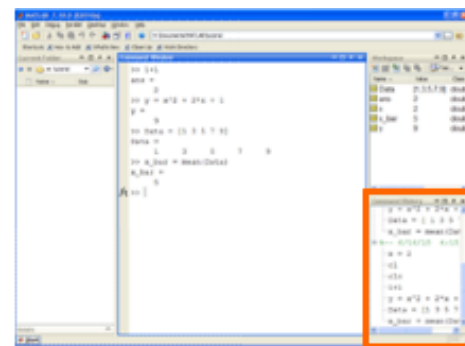
```
Command Window
File Edit Debug Desktop Window Help
>> 1+1
ans =
     2
>> y = x^2 + 2*x + 1
y =
     9
>> Data = [1 3 5 7 9]
Data =
     1     3     5     7     9
>> x_bar = mean(Data)
x_bar =
     5
fx >> |
```




Zgodovina (Command history)

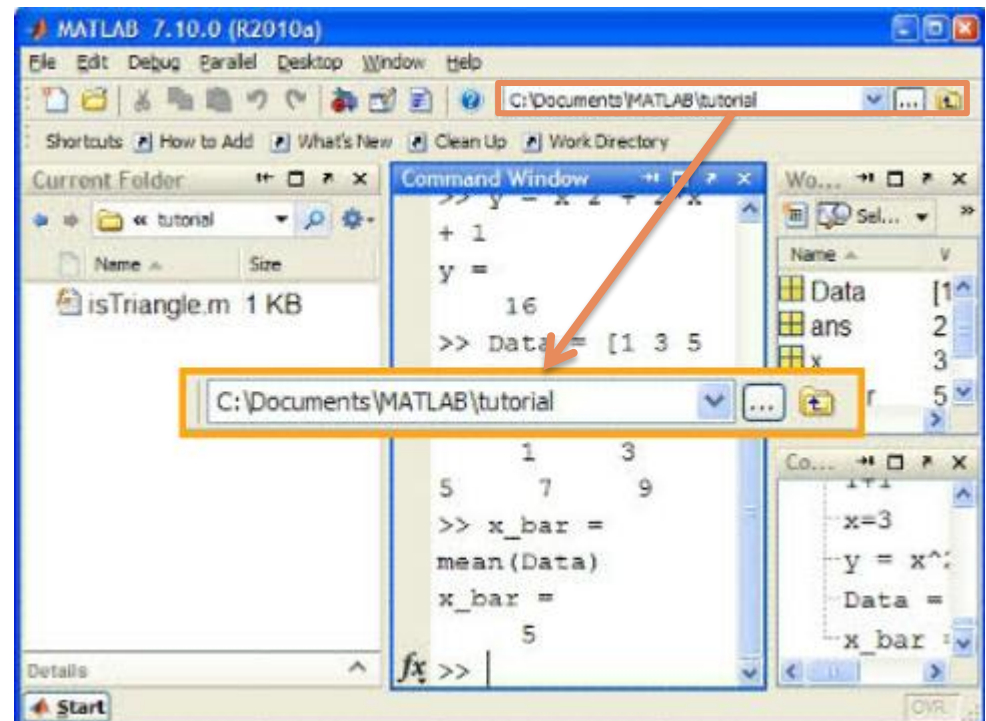
- Običajno spodaj desno
- Samodejno spremljanje/shranjevanje vnesenih ukazov
- Pregled nad predhodno vnesenimi ukazi (tudi za več sej nazaj)

```
Command History
File Edit Debug Desktop Window Help
1=1
a = [ 1 2 3 4 ]
x = 5
x = 2
clc
1+1
y = x^2 + 2*x + 1
Data = [1 3 5 7 9]
x_bar = mean(Data)
```



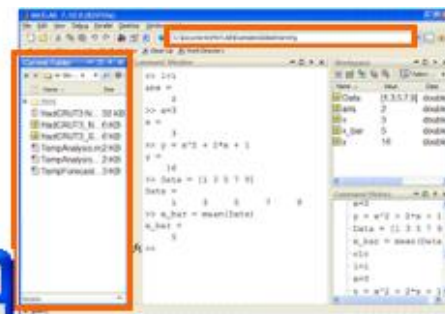
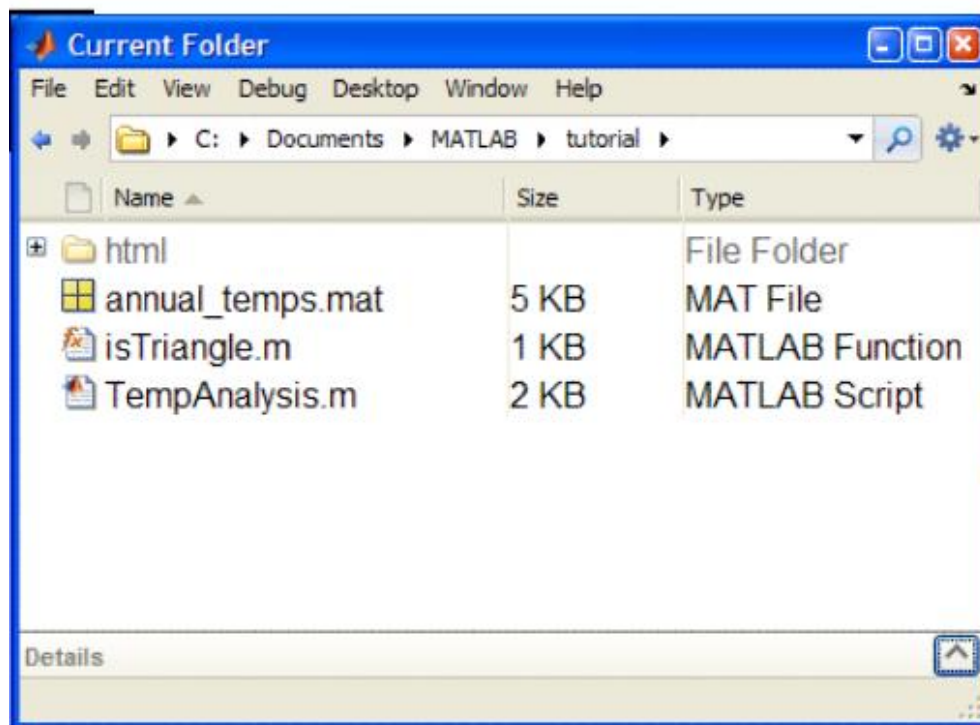
Trenutna mapa (Current folder)

- Okence v orodni vrstici
- Izpis lokacije in poti trenutne delovne mape
- Trenutna mapa = mapa, kamor se shranjuje delo (funkcije, datoteke, ...)
- Za spremembo te mape klikni na "Browse" gumb 



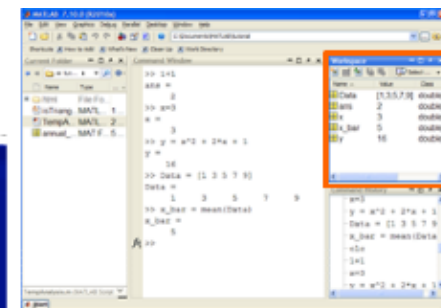
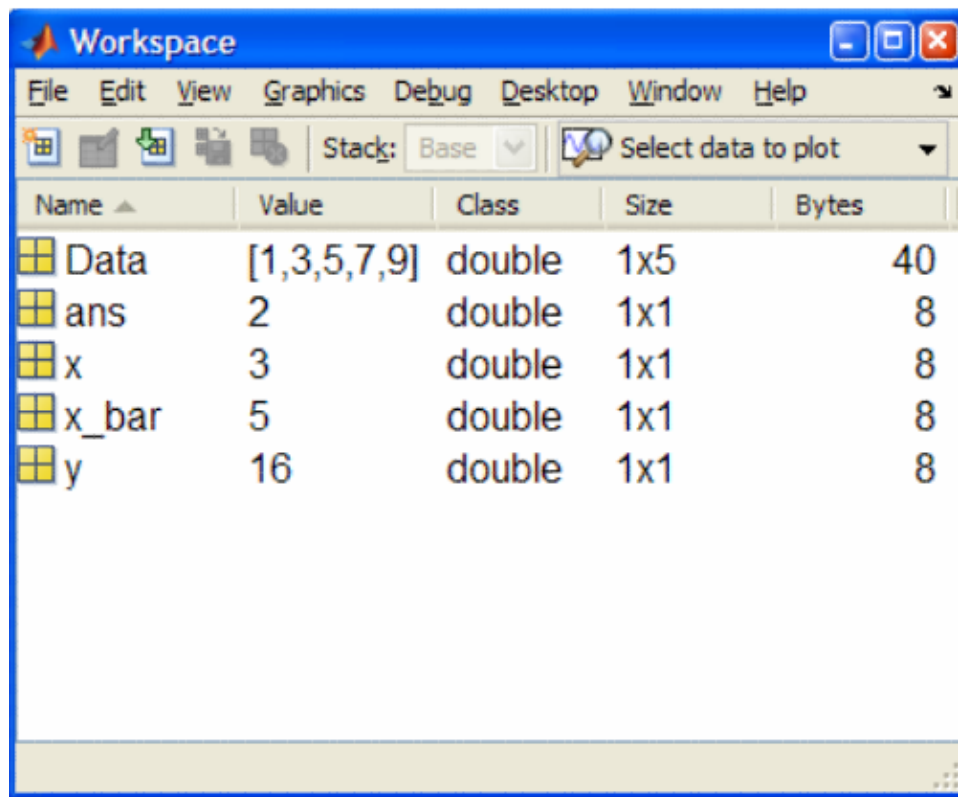
Brskalnik (Current folder browser)

- Prikazuje vsebino trenutne mape
- S spreminjanjem prikazane mape v brskalniku spreminjamo tudi trenutno mapo za shranjevanje našega dela
- Navigacija je enaka kot pri Win Explorerju



Delovno okolje (Workspace browser)

- Nad zgodovino
- V delovnem okolju so shranjeni podatki vaše trenutne seje – rezultati vnesenih ukazov
- Se zbríše, ko zapremo program



Matlab osnove

1. Osnovni računski postopki

- Matlab kot kalkulator
- Ustvarjanje spremenljivk
- Lociranje podatkov v Matlabu
- Pregledovanje vsebine spremenljivk

2. Ustvarjanje polj

- Ustvarjanje vektorjev
- Ustvarjanje matrik

3. Urejanje polj

- Indeksiranje elementov v poljih
- Operator podpičja (:)

4. Računanje s polji

- Matrične operacije
- Lastne vrednosti matrike
- Operacije nad polji

5. Vizualizacija matematičnih funkcij

Osnovni računski postopki

- Matlab kot kalkulator
 - Programabilen
 - Delo s funkcijami – vhodne parametre podajamo v okroglih oklepajih
 - Iskanje: Matlab Help! (tipka F1, ukaz *doc*, ikona z vprašajem v orodni vrstici)
 - Razporeditev po kategorijah (npr. Mathematics) in podkategorijah (npr. Linear Algebra)
 - Podana je sintaksa, osnovni primeri, itd.

Matlab kot kalkulator - primer

$$\rho = \frac{1 + \sqrt{5 - i}}{2}$$

$$z = e^{\rho}$$

$$a = |3 + 4i|$$

$$x = \sin\left(3t + \frac{\pi}{2}\right), \quad t = 0.2$$



```
Command Window
File Edit Debug Desktop Window Help
>> rho = (1 + sqrt(5-i))/2
rho =
    1.6236 - 0.1113i
>> z = exp(rho)
z =
    5.0397 - 0.5630i
>> a = abs(3+4i)
a =
    5
>> t = 0.2
t =
    0.2000
>> x = sin(3*t + pi/2)
x =
    0.8253
fx >> |
```

Naloga: poišči vgrajeno funkcijo s katero lahko izračunamo tretji koren spremenljivke a .

Matlab kot kalkulator - naloga

Izračunajte naslednje vrednosti

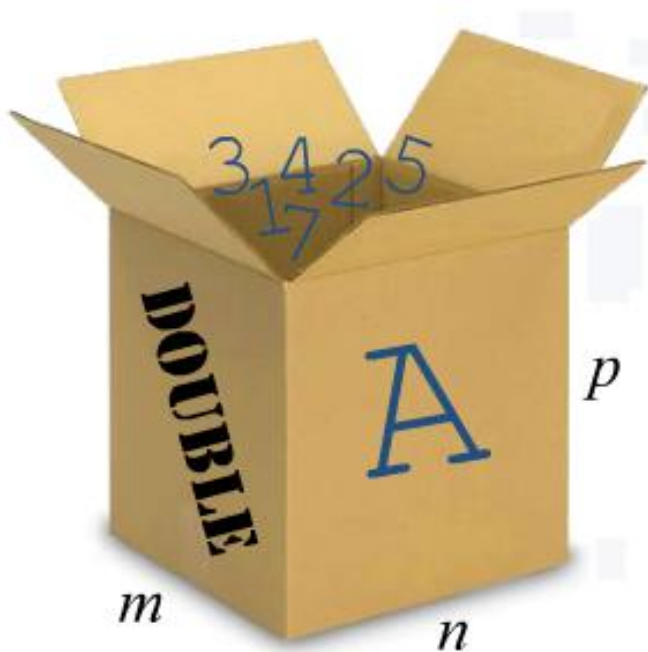
$$f = \frac{33 + \sqrt[2]{53}}{3 + \sqrt[3]{19}}$$

$$d = e^{-\frac{1}{5}}$$

$$r = e^3 \cdot \sin \frac{3\pi}{4}$$

$$z = f + d^2 + r^3$$

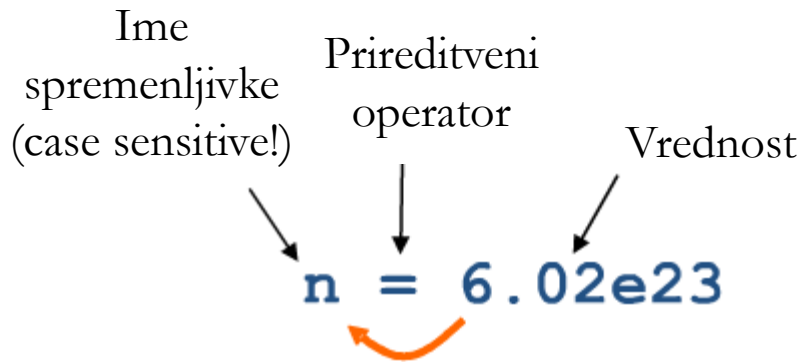
Kje so rezultati?



- Shranjeni v pomnilniku kot spremenljivke – podatkovni zabojniki
- Spremenljivke = matrike (polja)
- Atributi:
 - Velikost ($m \times n \times p$)
 - Tip (double, single, cell, ...)

Ustvarjanje spremenljivk

Sintaksa:



Primer:

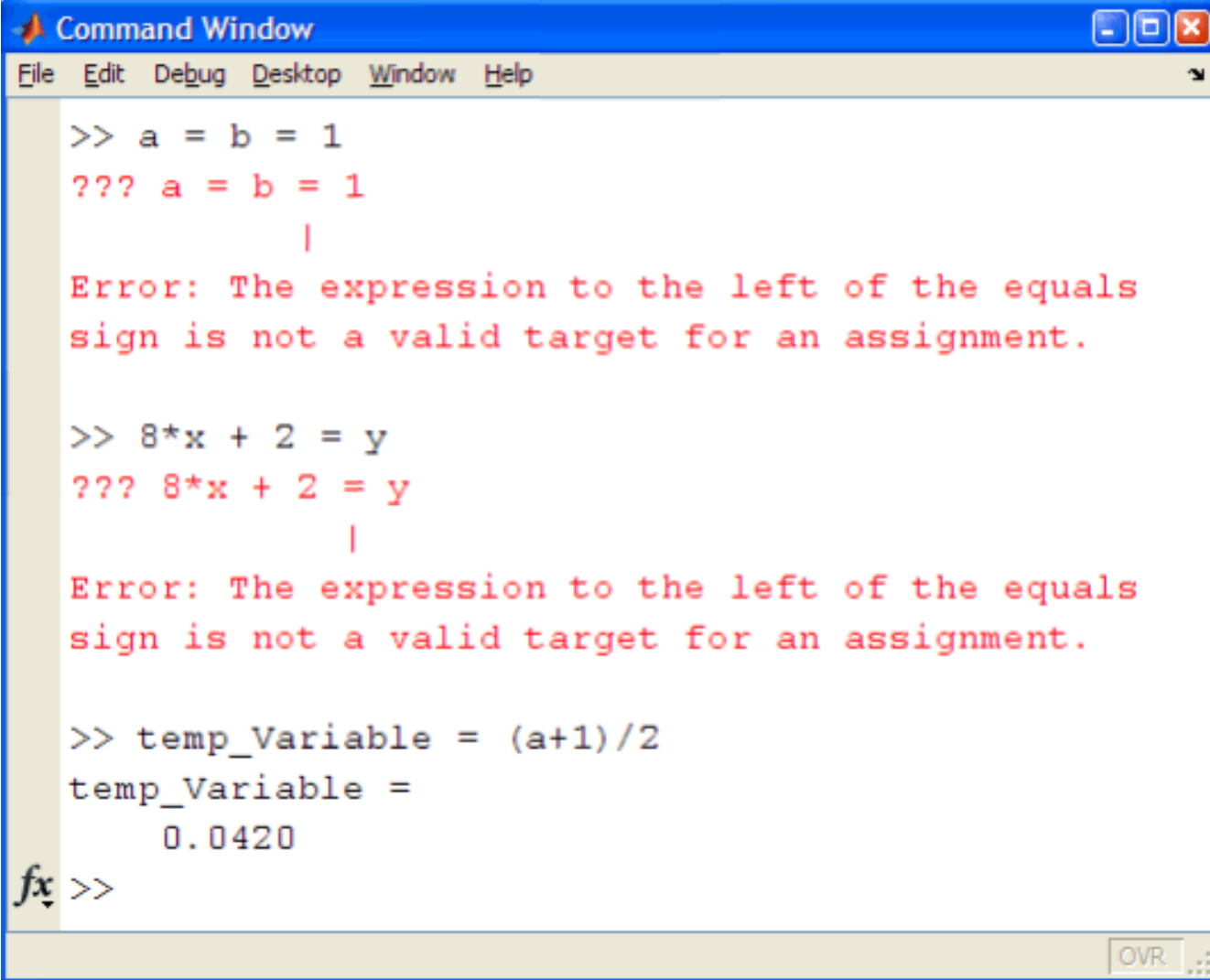
$$\theta = \frac{\pi}{2}$$

$$y = 2 + i \sin(\theta)$$

- Shranjene spremenljivke lahko uporabimo pri nadaljnjih izračunih

```
Command Window
File Edit Debug Desktop Window Help
>> theta = pi/2
theta =
    1.5708
>>
>> format long
>> theta
theta =
    1.570796326794897
>> format short
>>
>> y = 2 + i*sin(theta)
y =
    2.0000 + 1.0000i
fx >> |
```

Ustvarjanje spremenljivk – sintaktični primeri



```
Command Window
File Edit Debug Desktop Window Help
>> a = b = 1
??? a = b = 1
      |
Error: The expression to the left of the equals
sign is not a valid target for an assignment.

>> 8*x + 2 = y
??? 8*x + 2 = y
      |
Error: The expression to the left of the equals
sign is not a valid target for an assignment.

>> temp_variable = (a+1)/2
temp_variable =
    0.0420
fx >>
```


Lociranje in dostop do podatkov

Workspace Browser

The screenshot shows the MATLAB 7.10.0 (R2010a) interface. The Command Window displays the output of the `whos` command, listing variables in the workspace. The Workspace Browser shows a table of these variables, including their names, classes, sizes, and values. The Command History window shows the sequence of commands executed: `x = 0:0.1:100;`, `z = zeros(28,8,8);`, `clc`, and `whos`.

Name	Class	Size	Val
A	double	5x5	<5>
TF	logical	1x4	<1>
m	double	5x5	<5>
mycell	cell	2x3	<2>
s	char	1x6	'ma'
x	double	1x1001	<1>
z	double	28x8x8	<14>

```
x = 0:0.1:100;  
z = zeros(28,8,8);  
clc  
whos
```

Lociranje in dostop do podatkov

Urejevalnik spremenljivk:
Prikazuje vsebino spremenljivk, ki
jo lahko tudi spreminjamo.

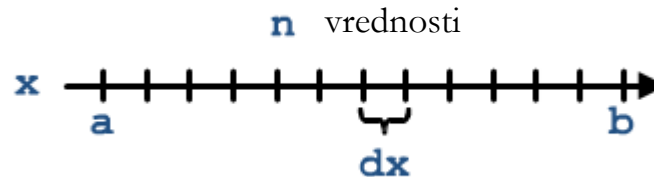
Z dvoklikom na spremenljivko jo
prikažemo v Urejevalniku
spremenljivk (Variable Editor)

The screenshot displays the MATLAB 7.10.0 (R2010a) environment. The Variable Editor window is open, showing a 5x5 double matrix 'A'. The Workspace window shows a list of variables: A (double, 5x5), TF (logical, 1x4), m (double, 5x5), mycell (cell, 2x3), s (char, 1x6), and x (double, 1x1001). The Command Window shows the execution of 'x = 0:0.1:100;' and 'z = zeros(28,8,8);'. The Command History window shows the sequence of commands: 'x = 0:0.1:100;', 'z = zeros(28,8,8);', 'clc', and 'whos'.

Name	Class	Size	Val
A	double	5x5	<5>
TF	logical	1x4	<1>
m	double	5x5	<5>
mycell	cell	2x3	<2>
s	char	1x6	'ma'
x	double	1x1001	<1>

```
x = 0:0.1:100;
z = zeros(28,8,8);
clc
whos
```

Ustvarjanje polj – vektorjev



- Vektor = polje ena od dimenzij je enaka 1
- 2 osnovni metodi za ustvarjanje vrstičnih vektorjev $1 \times n$:
 - Operator :

```
>> x = a:dx:b
```
 - Ukaz `linspace`

```
>> x = linspace(a,b,n)
```
- Stolpčni vektor ($n \times 1$) dobimo s transponiranjem – operator `'`

```
>> x = x'
```
- Namig: podpičje na koncu izraza prepreči izpis ustvarjenega polja v ukaznem oknu → preglednejše okno in hitrejšo delovanje
- Primer: na oba načina ustvarite vektor x med 0 in 6 z dolžino intervala 2.

Ustvarjanje matrik

- 2D polja
- Matriko lahko vnesemo ročno:
 - vrednosti podamo v oglatih oklepajih
 - Vrednosti se vnašajo po vrsticah
 - Ločevanje stolpcev: **,** ali **presledek**
 - Ločevanje vrstic: **;** ali **enter**
- Vnos z uporabo funkcij `ones`, `zeros`, `rand`

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

```
>> A = [1,2,3; 4,5,6; 7,8,9]
```

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

```
>> A = [1 2 3  
4 5 6  
7 8 9]
```

Urejanje polj

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

- Indeksiranje

```
>> k = A(2,3)
```

```
>> blok1 = A(2, [1 2])
```

- Operator dvopičja (colon) – :

```
>> blok2 = A(2, 1:2)
```

```
>> vrstica2 = A(2, :)
```

- Sestavljanje matrik

```
>> B = [A ; A]
```

- Transponiranje

```
>> Atrans = A'
```

- Vgrajeni ukazi: `sort`, `reshape`, `flipud`

Matrične operacije

- Vse spremenljivke so matrike!
- Pravila pri uporabi matematičnih operacij:
 - Množenje: ujemanje notranjih dimenzij faktorjev
 - Seštevanje in odštevanje: popolno ujemanje dimenzij
- Poseben primer: računske operacije z matriko (1 x 1), ki je obravnavana kot skalar

Osnovne operacije s skalarno vrednostjo



```
>> A
A =
    1  2
    4  5
    7  8

>> B = 2 * A
B =
     2  4
     8 10
    14 16
```

Množenje s skalarjem

```
>> A
A =
    1  2
    4  5
    7  8

>> B = A + 2
B =
     3  4
     6  7
     9 10
```

Prištevanje skalarja

```
>> A
A =
    1  2
    4  5
    7  8

>> B
B =
    1  0  1
    1  1  0

>> C = A * B
C =
     3  2  1
     9  5  4
    15  8  7
```

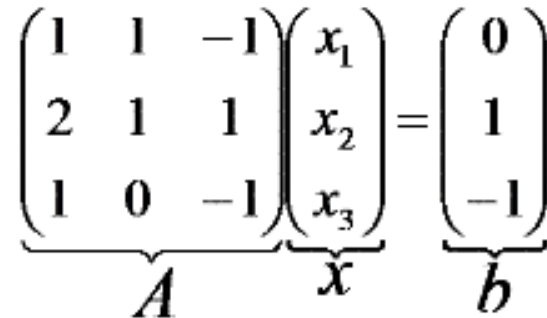
Množenje matrik

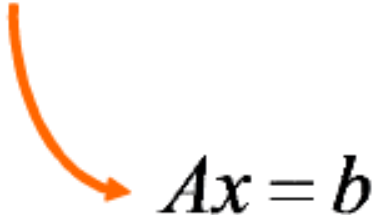
Primer: sistem linearnih enačb

$$x_1 + x_2 - x_3 = 0$$

$$2x_1 + x_2 + x_3 = 1$$

$$x_1 - x_3 = -1$$


$$\underbrace{\begin{pmatrix} 1 & 1 & -1 \\ 2 & 1 & 1 \\ 1 & 0 & -1 \end{pmatrix}}_A \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}}_x = \underbrace{\begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}}_b$$


$$Ax = b$$

Določite vrednosti vektorja x !

Primer: lastne vrednosti matrike

$$A = \begin{bmatrix} 2 & -1 \\ 1 & -3 \end{bmatrix}$$



$$\det(A - \lambda I) = 0$$



$$c_n \lambda^n + \dots + c_2 \lambda^2 + c_1 \lambda + c_0 = 0$$

- Za vsako matriko lahko določimo njen karakteristični polinom
- Funkcija `poly` – koeficienti karakterističnega polinoma
- Funkcija `roots` – poišče korene karakterističnega polinoma = lastne vrednosti
- Funkcija `eig` – direktna pot do lastnih vrednosti in vektorjev

Operacije nad polji

```
>> A
A =
    1 2 4
    5 7 8

>> B
B =
    1 0 1
    1 1 0

>> C = A + B
C =
    2 2 5
    6 8 8
```

Seštevanje polj

```
>> A
A =
    1 2 4
    5 7 8

>> B
B =
    1 0 1
    1 1 0

>> C = A .* B
C =
    1 0 4
    5 7 0
```

Množenje polj

```
>> A
A =
    1 2 4
    5 7 8

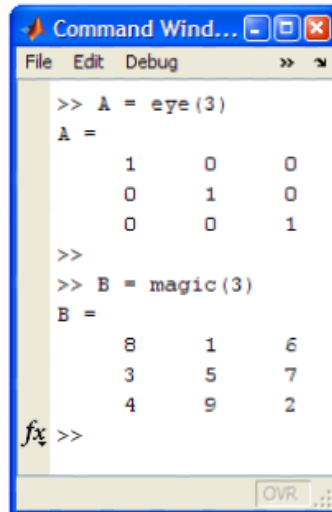
>> B = A.^2
B =
    1 4 16
    25 49 64
```

Potenciranje polj



- Delujejo na nivoju istoležnih elementov
- Ujemanje velikosti mora biti popolno
- Pred računski operator vstavimo piko

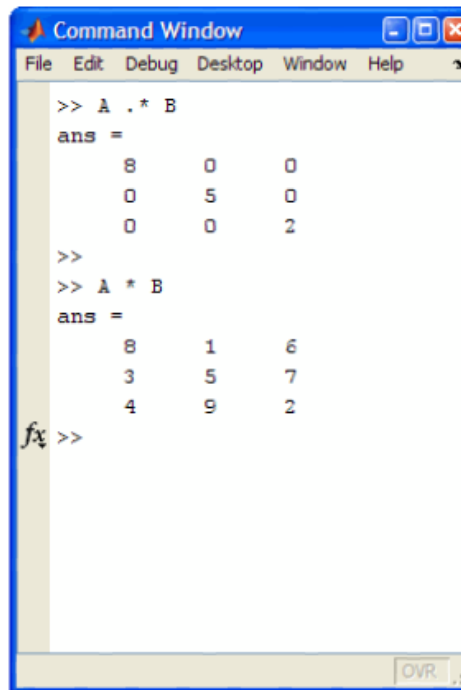
Množenje polj vs množenje matrik



```
>> A = eye(3)
A =
     1     0     0
     0     1     0
     0     0     1

>>
>> B = magic(3)
B =
     8     1     6
     3     5     7
     4     9     2

fx >>
```



```
>> A .* B
ans =
     8     0     0
     0     5     0
     0     0     2

>>
>> A * B
ans =
     8     1     6
     3     5     7
     4     9     2

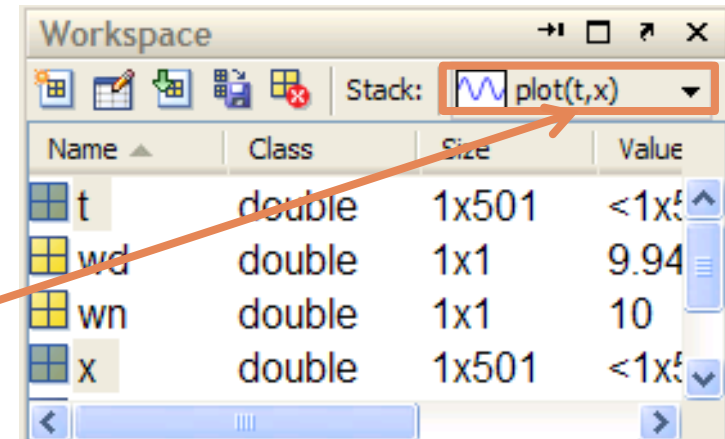
fx >>
```

Vizualizacija matematičnih funkcij

Za izris funkcije x sta 2 možnosti:

- Interaktivno prek delovnega okolja
 - Z uporabo `Ctrl` označimo med spremenljivkami tisti dve, ki ju želimo izrisati na 2D diagram – t in x
 - Nato v orodni vrstici delovnega okolja uporabimo bližnico za izris
- Z uporabo vgrajene funkcije `plot` prek ukaznega okna:

```
>> plot(t,x)
```



Uporaba vgrajenih funkcij

■ Sintaksa:

- `Ime_funkcije(parameter1, parameter2, ...)`
- Vhodne parametre funkcij zmeraj podajamo v oklepajih
 - Primer: `>> plot(t, x)`
- Rezultat funkcije lahko priredimo poljubni izhodni spremenljivki
 - Primer: `>> y = cos(pi/4)`

Naloga

Primer dušenega nihanja:

$$z = 0.1$$

$$w_n = 10$$

$$w_d = w_n \sqrt{1 - z^2}$$

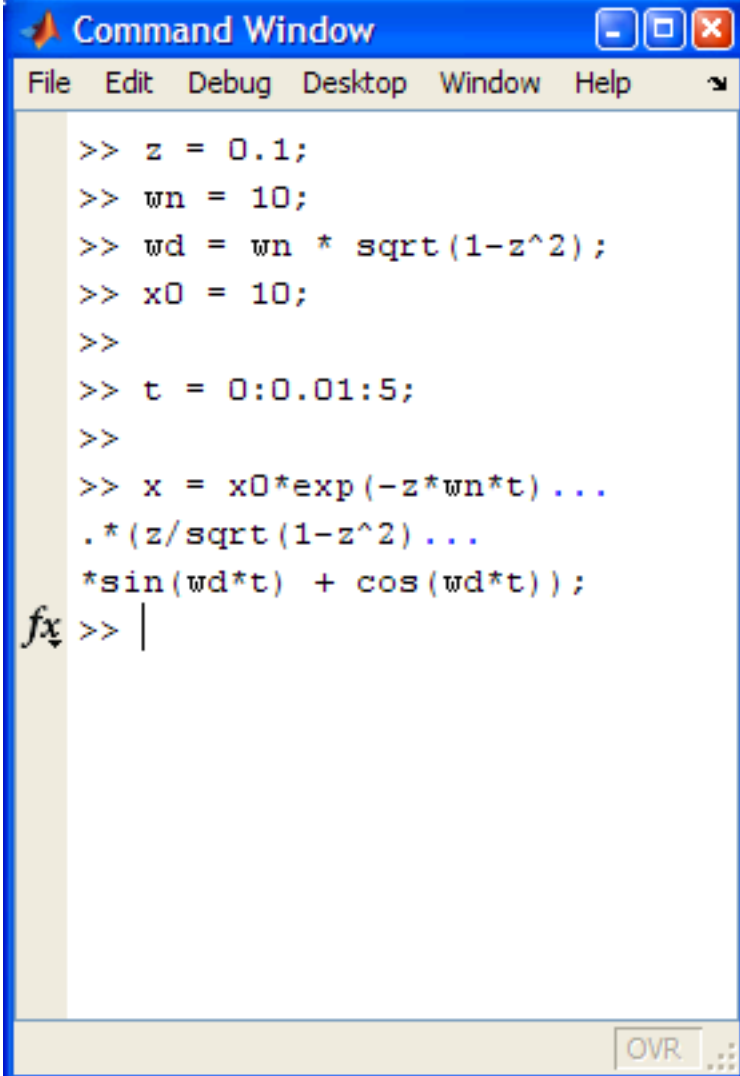
$$x_0 = 10$$

Nariši:

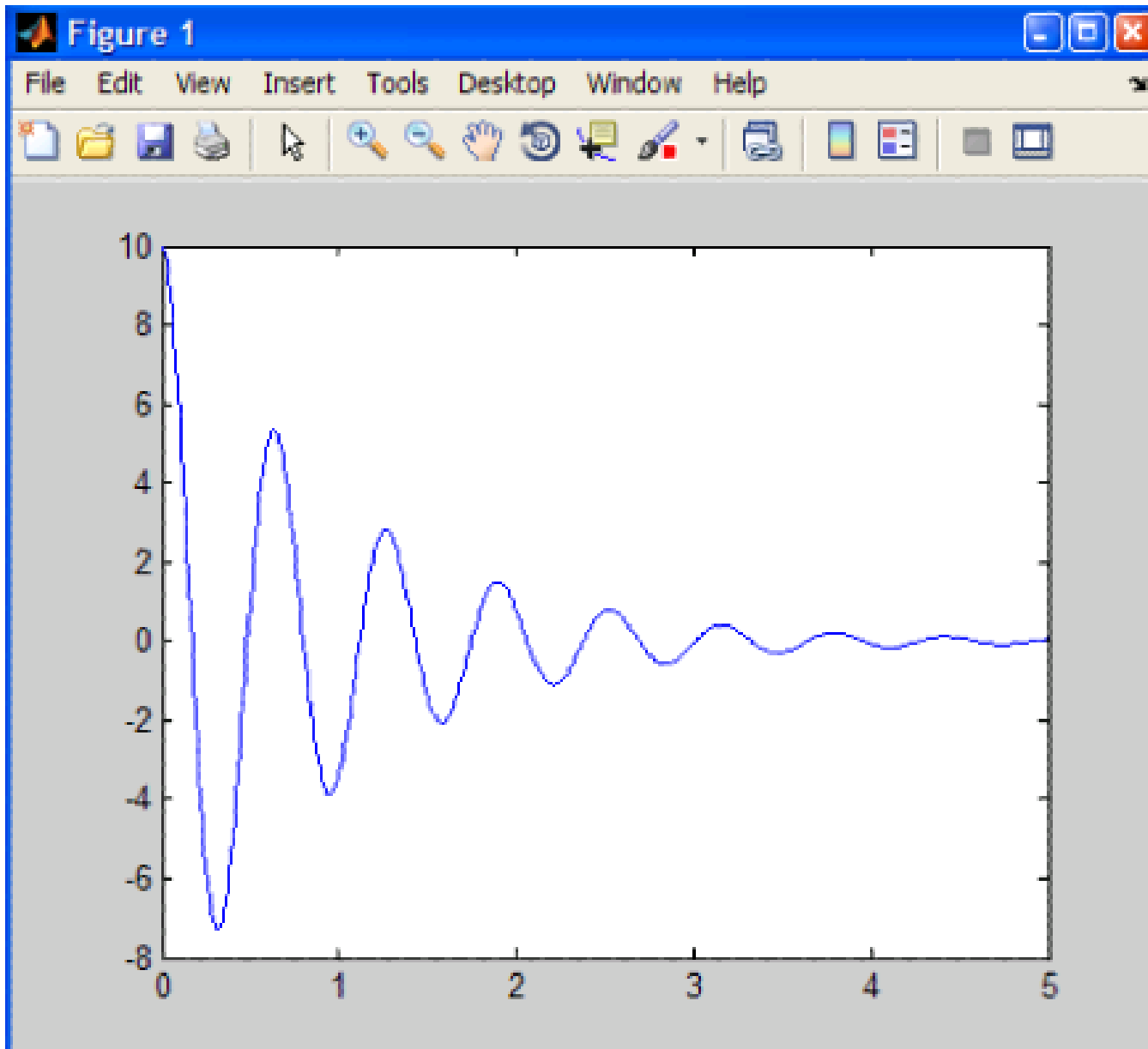
$$x(t) = x_0 e^{-zw_n t} \left(\frac{z}{\sqrt{1 - z^2}} \sin(w_d t) + \cos(w_d t) \right)$$

$$t = 0 \div 5 \text{ s}$$

Izriši graf (t, x(t))!



```
Command Window
File Edit Debug Desktop Window Help
>> z = 0.1;
>> wn = 10;
>> wd = wn * sqrt(1-z^2);
>> x0 = 10;
>>
>> t = 0:0.01:5;
>>
>> x = x0*exp(-z*wn*t) ...
.* (z/sqrt(1-z^2) ...
.*sin(wd*t) + cos(wd*t));
fx >> |
OVR
```



Matlab osnove - povzetek

- Osnovni računski postopki

- Delo s polji – matrikami
 - Ustvarjanje
 - Urejanje
 - Računanje

- Vizualizacija rezultatov (matematičnih funkcij)

Reševanje problemov z Matlabom

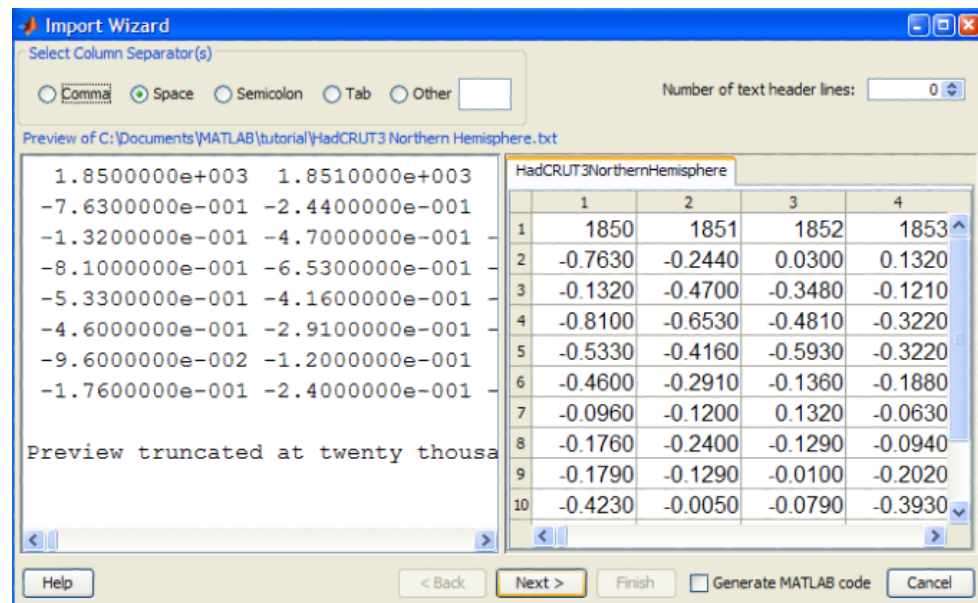
- Uvoz podatkov
(wav, txt, csv, slikovne datoteke itd.)
- Vizualizacija podatkov
- Numerična analiza
 - Prilagajanje krivulj
 - Avtomatska analiza s skriptami
 - Objava rezultatov
- Primeri:
 - Simulacije (omrežij, posameznih delov ali nivojev omrežij, prenosa podatkov itd.)
 - Modeliranje in generiranje signalov, šuma
 - Implementacija različnih modulacijskih tehnik
 - Obdelava signalov (slike, video itd.)
 - Fuzija podatkov
 - Vizualizacija

Uvoz podatkov

- Čarovnik za uvoz podatkov
- Programirani uvoz podatkov
- Pridobivanje zanimivih podatkov
 - Naslavljanje vrstic in stolpcev
- Shranjevanje in odpiranje podatkov

Čarovnik za uvoz podatkov

- Zagon:
 - File>Import Data ...
 - Desni klik na izbrano datoteko v brskalniku trenutne mape>Import Data ...
- Parametri
 - Separator stolpcev, imena spremenljivk
- Uvoženi podatki se shranijo v delovno okolje trenutne seje
- Uporabno za enkratni uvoz
- Osveževanje podatkov → sprogramiran ponavljajoč se UVOZ



Programirani uvoz podatkov

- Avtomatizacija uvažanja podatkov
- Primerno za podatke, ki se nenehno osvežujejo (senzorji, podatkovne baze itd.)
- Funkcija `importdata` (čarovnik brez UI)
- Podprti formati za uvoz:
 >> `doc fileformats`
- Druge funkcije za specifičen tip podatkov:

- `textread`
- `imread`
- `wavread`
- `uiimport`
- `xlsread`
- `input`

```
Average hourly temperature readings for each  
month in Metcalf, MA, home of The MathWorks Inc.  
hour  jan  feb  mar  apr  may  jun  jul  aug  sep  oct  nov  dec  
0000  2  19  37  42  53  67  74  75  62  52  42  34  
0100  9  18  36  41  53  67  74  74  61  51  42  33  
0200  8  16  35  41  52  66  73  73  60  49  41  32  
0300  9  17  35  40  51  64  74  74  59  50  42  32  
0400  10  18  36  40  50  67  74  75  49  50  43  33  
0500  10  18  37  41  50  69  76  75  53  51  43  33  
0600  11  19  37  42  50  69  76  75  62  53  44  33  
0700  11  19  38  44  59  70  77  77  63  54  44  34  
0800  12  21  40  56  70  79  77  63  56  45  34  
0900  15  23  41  47  60  72  79  79  66  57  47  34  
1000  14  23  41  48  62  73  80  79  64  57  47  37  
1100  19  27  42  51  63  74  82  79  64  57  48  39  
1200  22  30  45  55  63  74  82  80  64  60  49  40  
1300  25  33  47  53  64  74  82  81  67  62  49  40  
1400  27  35  47  53  64  76  83  81  68  62  50  41  
1500  27  36  46  55  66  78  82  82  68  63  52  40
```

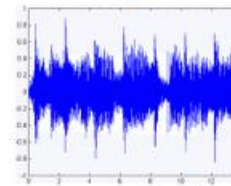
`textread`



`imread`
`importdata`



`input`



`wavread`
`uiimport`

	A	B	C	D
1	Dates	Min	Case	Lu
2				
3	1/2/1997	2.04E+01	7.00E+00	1.10E+01
4	1/3/1997	2.12E+01	7.00E+00	1.10E+01
5	1/4/1997	2.11E+01	8.00E+00	1.10E+01
6	1/5/1997	2.13E+01	8.00E+00	1.10E+01
7	1/6/1997	2.08E+01	7.00E+00	1.10E+01
8	1/9/1997	2.08E+01	7.00E+00	1.20E+01
9	1/10/1997	2.11E+01	8.00E+00	1.20E+01
10	1/13/1997	2.03E+01	8.00E+00	1.20E+01
11	1/14/1997	2.13E+01	8.00E+00	1.20E+01
12	1/15/1997	2.12E+01	8.00E+00	1.20E+01
13	1/16/1997	2.15E+01	8.00E+00	1.20E+01
14	1/17/1997	2.18E+01	8.00E+00	1.20E+01
15	1/20/1997	2.29E+01	8.00E+00	1.20E+01
16	1/21/1997	2.33E+01	8.00E+00	1.30E+01
17	1/22/1997	2.43E+01	8.00E+00	1.30E+01
18	1/23/1997	2.39E+01	8.00E+00	1.30E+01
19	1/24/1997	2.42E+01	8.00E+00	1.20E+01
20	1/27/1997	2.42E+01	8.00E+00	1.20E+01

`xlsread`

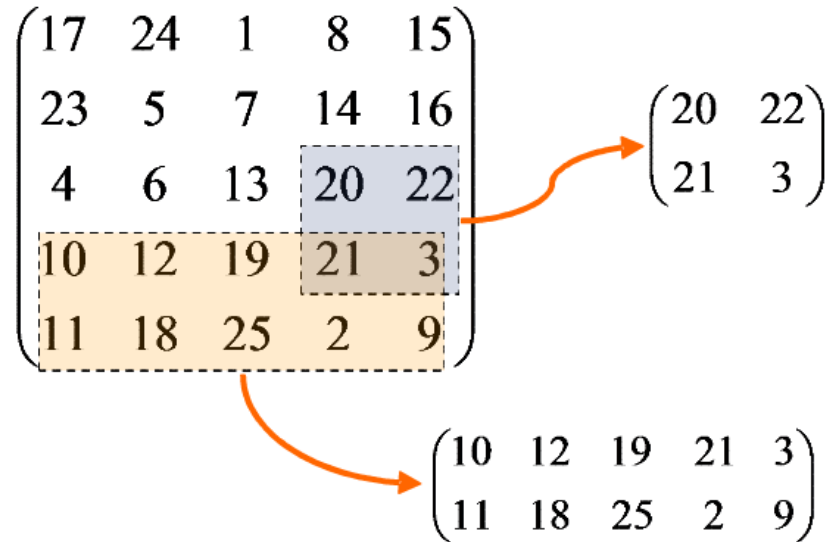
Naloga

Uvozite podatke iz datoteke *L_cikel1_data_antena.txt*
(med gradivi predmeta na spletu).

Uporabite interaktivni način in način preko ukaznega okna.

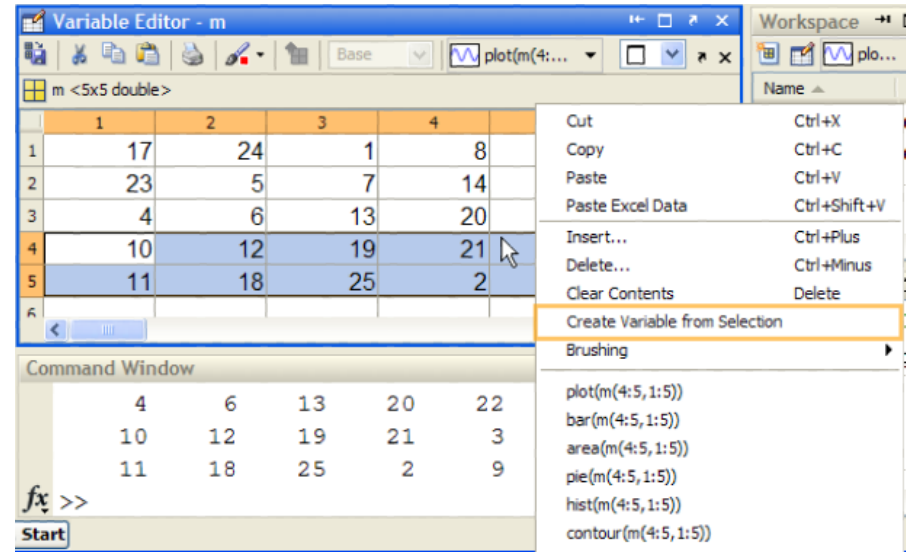
Pridobivanje zanimivih podatkov

- Podatki podani v obliki večjih večdimenzionalnih matrik
- Delo s podatki:
 - Zanima nas le del vseh podatkov
 - Podatke želimo razvrstiti za lažje nadaljnje delo
- Dve možnosti:
 - Urejanje v urejevalniku spremenljivk
 - Urejanje s funkcijami prek ukaznega okna



Urejevalnik spremenljivk

- Izbrano spremenljivko odpremo v urejevalniku (dvoklik)
- Izberemo del, ki nas zanima in ga želimo prenesti v novo spremenljivko
- Z desnim klikom na izbrani del odpremo meni → **Create Variable from Selection**
- Nazadnje izberemo še ime nove spremenljivke
- Izberemo lahko celoten stolpec ali vrstico (podobno kot v Excel-u)



Urejanje prek ukaznega okna

- Naslavljanje po vrsticah in stolpcih – dostop do posameznih elementov matrik
- Začetek štetja z 1

```
>> m = magic(5)
```

m =		Indeksi	Podatki						
1,1	17	1,2	24	1,3	1	1,4	8	1,5	15
2,1	23	2,2	5	2,3	7	2,4	14	2,5	16
3,1	4	3,2	6	3,3	13	3,4	20	3,5	22
4,1	10	4,2	12	4,3	19	4,4	21	4,5	3
5,1	11	5,2	18	5,3	25	5,4	2	5,5	9

↑
end,end

Naslavljanje po vrsticah in stolpcih

Naslavljanje posameznega elementa

```
>> m = magic(5)
```

m =		Indeksi	Podatki						
1,1	17	1,2	24	1,3	1	1,4	8	1,5	15
2,1	23	2,2	5	2,3	7	2,4	14	2,5	16
3,1	4	3,2	6	3,3	13	3,4	20	3,5	22
4,1	10	4,2	12	4,3	19	4,4	21	4,5	3
5,1	11	5,2	18	5,3	25	5,4	2	5,5	9

`m(3,2)`

`end,end`

Ime_spremenljivke (vrstica, stolpec)

Naslavljanje po vrsticah in stolpcih

Z uporabo dvopičja lahko naslovimo izbrani obseg vrstic ali stolpcev

```
>> m = magic(5)
```

m =		Indeksi	Podatki						
1,1	17	1,2	24	1,3	1	1,4	8	1,5	15
2,1	23	2,2	5	2,3	7	2,4	14	2,5	16
3,1	4	3,2	6	3,3	13	3,4	20	3,5	22
4,1	10	4,2	12	4,3	19	4,4	21	4,5	3
5,1	11	5,2	18	5,3	25	5,4	2	5,5	9

`m(2:4,3)`

`end,end`

Ime_spremenljivke(vrstica, stolpec)

Naslavljanje po vrsticah in stolpcih

Z uporabo dvopičja lahko naslovimo celotne vrstice in/ali stolpce

```
>> m = magic(5)
```

m =		Indeksi		Podatki					
1,1	17	1,2	24	1,3	1	1,4	8	1,5	15
2,1	23	2,2	5	2,3	7	2,4	14	2,5	16
3,1	4	3,2	6	3,3	13	3,4	20	3,5	22
4,1	10	4,2	12	4,3	19	4,4	21	4,5	3
5,1	11	5,2	18	5,3	25	5,4	2	5,5	9

`m(4, :)`

`end, end`

Ime_spremenljivke(vrstica, stolpec)

Naloga

Podana je naslednja matrika:

$$\begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix}$$

- Z naslavljanjem izpiši zadnji element matrike
- V urejevalniku spremenljivk ustvari spremenljivko, ki bo zajemala vrstic 3 in 4 ter stolpce 3 – 5. Nalogo ponovi še z uporabo naslavljanja.

Shranjevanje in odpiranje podatkov

The image shows two MATLAB windows. The 'Workspace' window on the left lists variables: A (5x5 double), TF (1x4 logical), m (5x5 double), myCell (2x3 cell), s (string), str (1x2 struct), x (1x1001 double), and z (28x8x8 double). The 'Current Directory' window on the right shows a file named 'myVariables.mat' selected, which contains variables A, TF, and m. An orange arrow labeled 'Shrani' points from the workspace to the file, and another orange arrow labeled 'Odpri' points from the file back to the workspace.

Shrani

Odpri

Name	Value	Min	Max
A	<5x5 double>	-0.2...	1
TF	<1x4 logical>		
m	<5x5 double>	1	25
myCell	<2x3 cell>		
s	'string'		
str	<1x2 struct>		
x	<1x1001 double>	0	
z	<28x8x8 double>	0	

Name	Value
A	<5x5 double>
TF	<1x4 logical>
m	<5x5 double>

Shranjevanje in odpiranje podatkov

- Shranjevanje preko delovnega okolja
 - Izberemo eno ali več spremenljivk v delovnem okolju
 - Desni klik → Save as ...
 - Spremenljivka/-e se shranijo kot .mat datoteka v trenutno mapo (brskalnik)
- Shranjevanje preko ukaznega okna
 - `>> save (ime_datoteke, ime_spremenljivke1, ...)`
 - Če uporabimo samo `save`, se shranijo vse spremenljivke v delovnem okolju kot datoteka `matlab.mat`
- Odpiranje
 - Dvoklik na izbrano datoteko v brskalniku trenutne mape
 - Uporabimo `load(ime_datoteke)` v ukaznem oknu

Uvoz podatkov - povzetek

- Uporaba čarovnika za uvoz podatkov
- Uporaba programiranega uvoza podatkov
- Pridobivanje zanimivih podatkov z naslavljanjem vrstic in stolpcev
- Shranjevanje in odpiranje podatkov

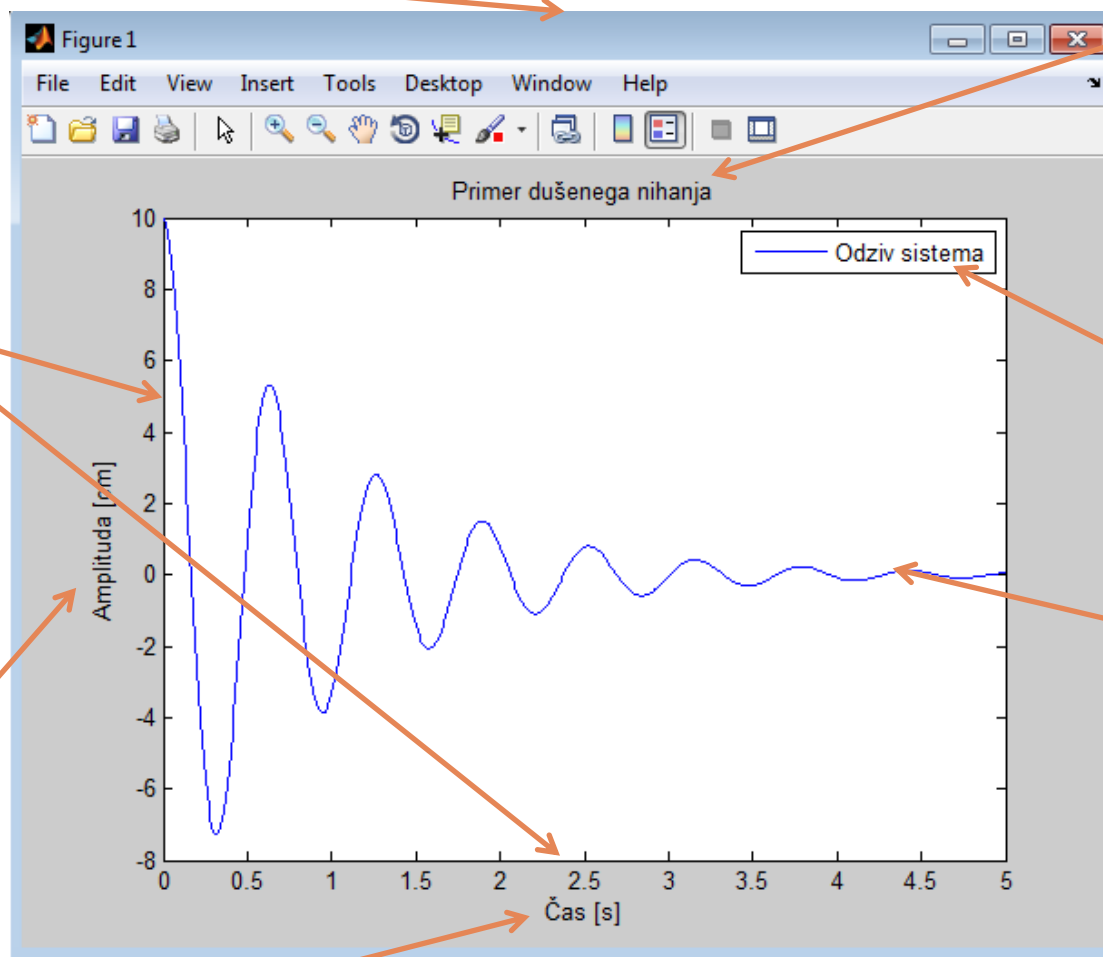
Vizualizacija podatkov

„Slika pove več kot 1000 besed!“

- Boljše razumevanje konceptov
- Bolj učinkovito sporočanje rezultatov in ugotovitev
- Dva osnovna načina za ustvarjanje in spreminjanje grafov:
 - Uporaba interaktivnih orodij
 - Uporaba ukaznega okna (`plot`)

Komponente grafa

Okno grafa



title

Osi

legend

ylabel

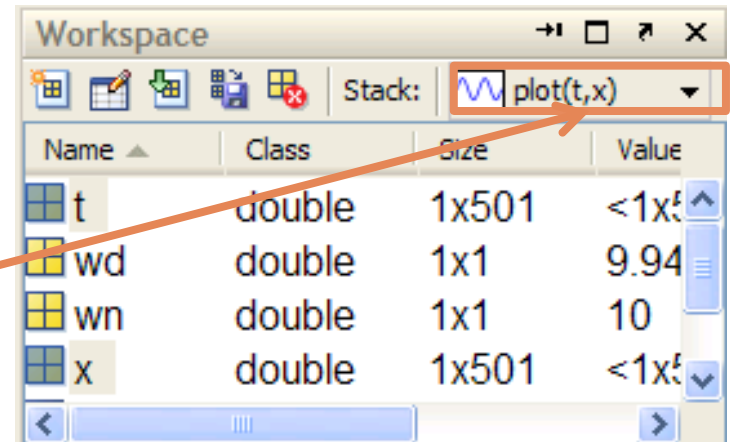
Črtni graf

xlabel

Vizualizacija matematičnih funkcij

Za izris funkcije x sta 2 možnosti:

- Interaktivno prek delovnega okolja
 - Z uporabo `Ctrl` označimo med spremenljivkami tisti dve, ki ju želimo izrisati na 2D diagram – t in x
 - Nato v orodni vrstici delovnega okolja uporabimo bližnico za izris



- Z uporabo vgrajene funkcije `plot` prek ukaznega okna:

```
>> plot(t,x)
```

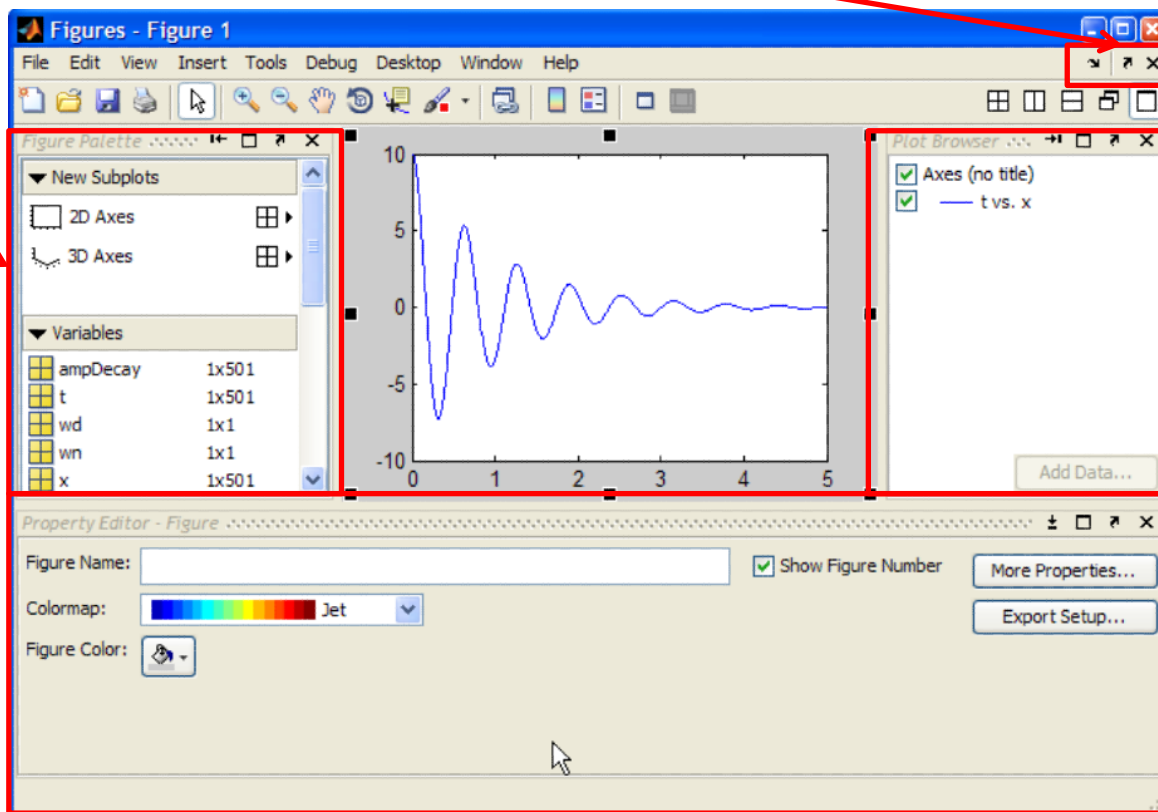
Interaktivno urejanje - plottools

- Interaktivna orodja za urejanje grafov: ukaz `plottools` v ukaznem oknu

Dodajanje podoken, pregled spremenljivk

Dock/Undock
(ko končamo z urejanjem)

Legenda, sledenje posameznim grafom, dodajanje novih grafov

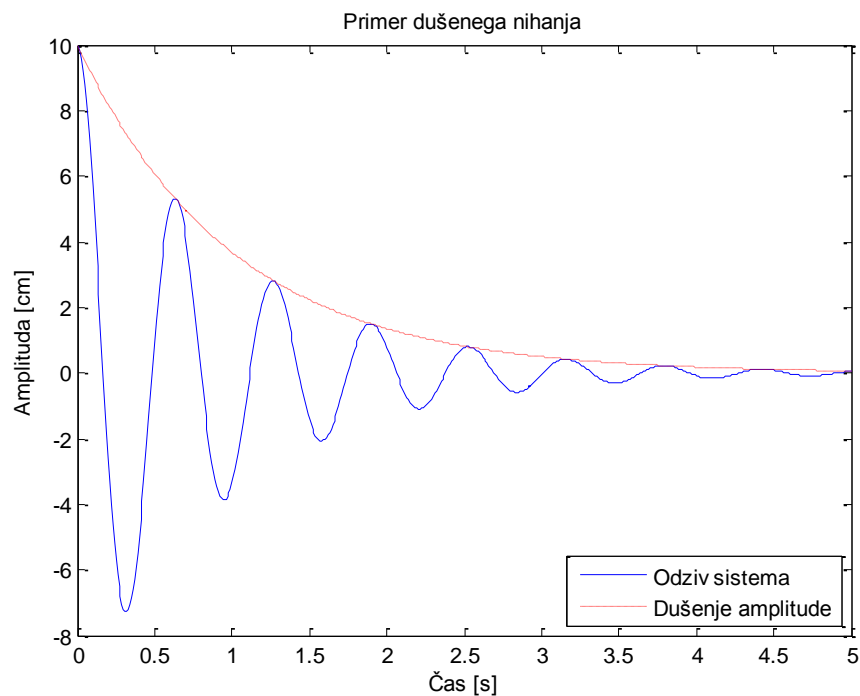


Naloga

- Na graf iz prejšnje naloge dodaj graf poteka dušenja:

$$dusenje = x_0 * e^{-z w_n t}$$

- Graf dušenja naj bo označen z rdečo črtkano črto
- Ustrezno označite naslov, osi in legendo grafa.



Urejanje s funkcijo `plot`

- Osnovni graf:
`>> plot(t,x)`
- Za dodajanje novih grafov na obstoječega moramo uporabiti ukaz `hold on/off`
- Za urejanje oblike črt v grafih uporabimo dodatne parametre v funkciji `plot` (`LineStyle` parametri), npr:
`>> plot(t,x, 'r:')`
- Za urejanje naslova, imen osi in legende imamo na voljo funkcije `title`, `xlabel`, `ylabel`, `legend`
Primer: `>> xlabel('Čas[s]')`
- Dodatni parametri – dokumentacija (Help)
- Druge funkcije za risanje grafov: `surf`, `imagesc`, `colormap`, `contour`

Vizualizacija – povzetek

- Komponente grafa v Matlabu
- Uporaba interaktivnih orodij za risanje grafov
- Risanje grafov s funkcijo `plot`

Numerična analiza

- Vgrajene funkcije
 - Uporaba

- Analiza podatkov
 - Prilagajanje krivulj
 - Avtomatizacija s skriptami
 - Objava rezultatov

Vgrajene funkcije

- Seznam vgrajenih funkcij:

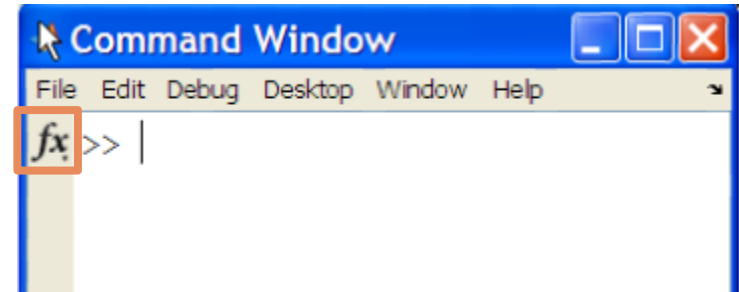
- Help>Functions>

- By Category

- Alphabetical

- Ukaz doc v ukaznem oknu

- Klik na simbol f_x v ukaznem oknu



- Razdelitev na kategorije:

- Elementarne matematične (Elementary Math Functions) – doc elfun

- Elementarne statistične (Elementary Statistical Functions) – doc datafun

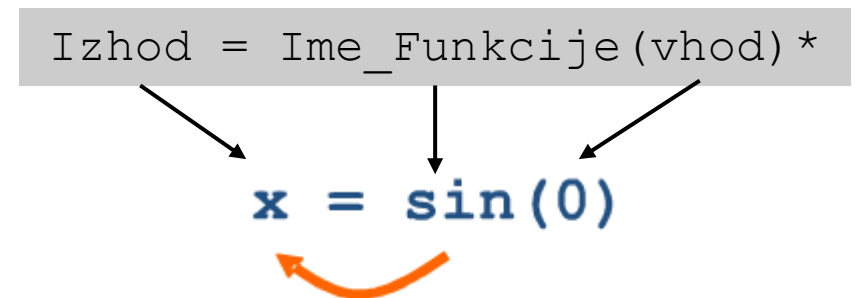
- Matrične (Matrix Functions) – doc elmat

- Napredne matematične (Advanced Math Functions) – doc specfun

- Glede na temo (All functions by topic) – helpwin

Uporaba funkcij – osnovna sintaksa

- Vhodne parametre funkcije vedno podajamo v (okroglih) oklepajih za njenim imenom
- Vhodni parametri so lahko konkretne vrednosti (števila, besedilni nizi, matrike itd.) ali pa spremenljivke iz delovnega okolja
 - Gnezdenje funkcij: rezultat neke funkcije je lahko vhodni parameter druge funkcije
npr: `>> a=abs(cos(pi))`
- Izhodno vrednost funkcije lahko priredimo novi/izbrani spremenljivki
- Če nismo prepričani o sintaksi → Help



* Vhodnih parametrov je lahko tudi več. Med sabo jih ločimo z vejicami.

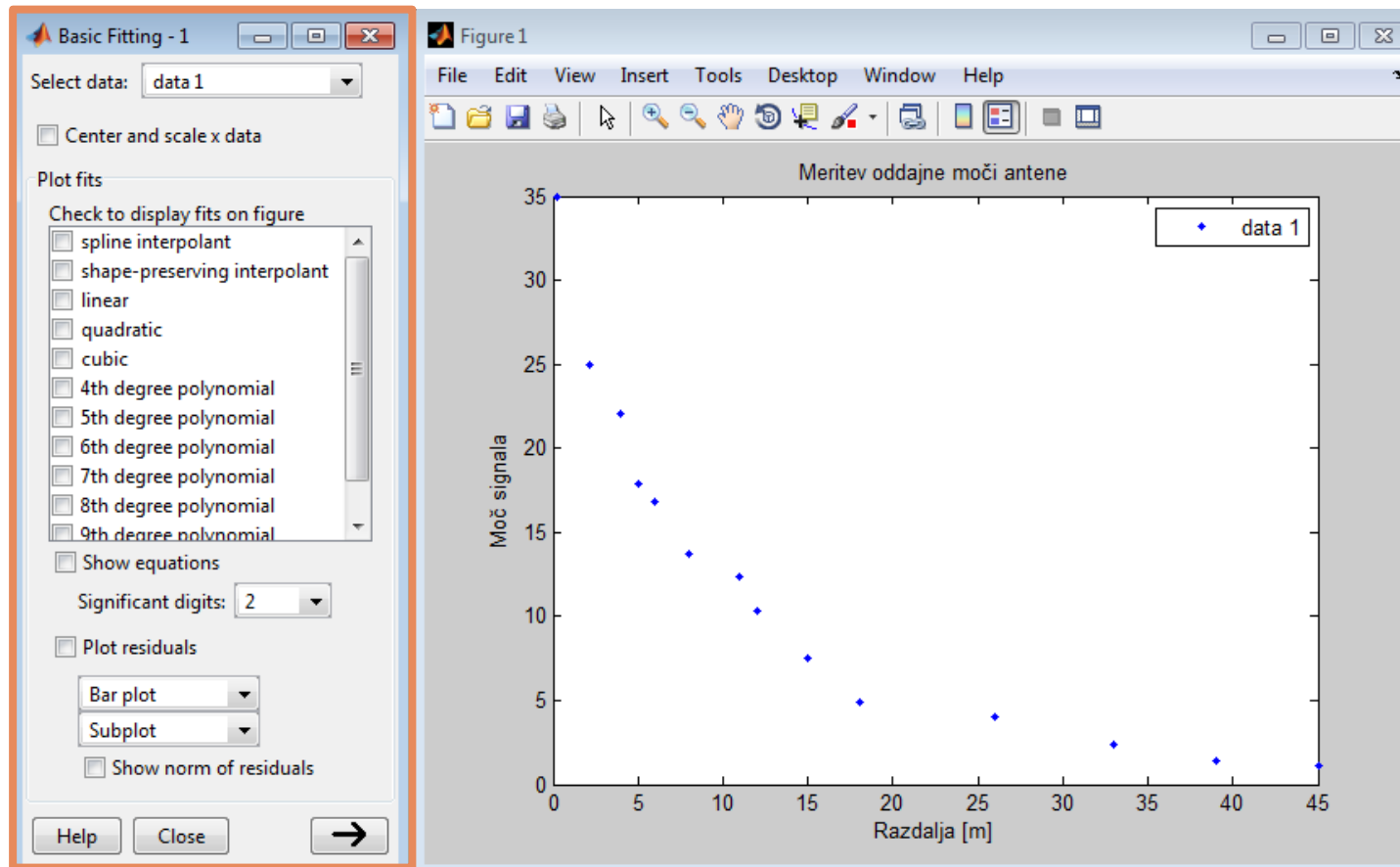
Prilagajanje krivulj

- Interpolacija nad izmerjenimi podatki – ocena stanja med izmerjenimi vrednostmi
- Dokazovanje, vrednotenje (matematičnih) predpostavk
- Modeliranje empiričnih podatkov

- Možnosti v Matlabu:
 - Basic Fitting Tool
 - Curve Fitting Toolbox
 - Spline Toolbox (spline = zlepek)

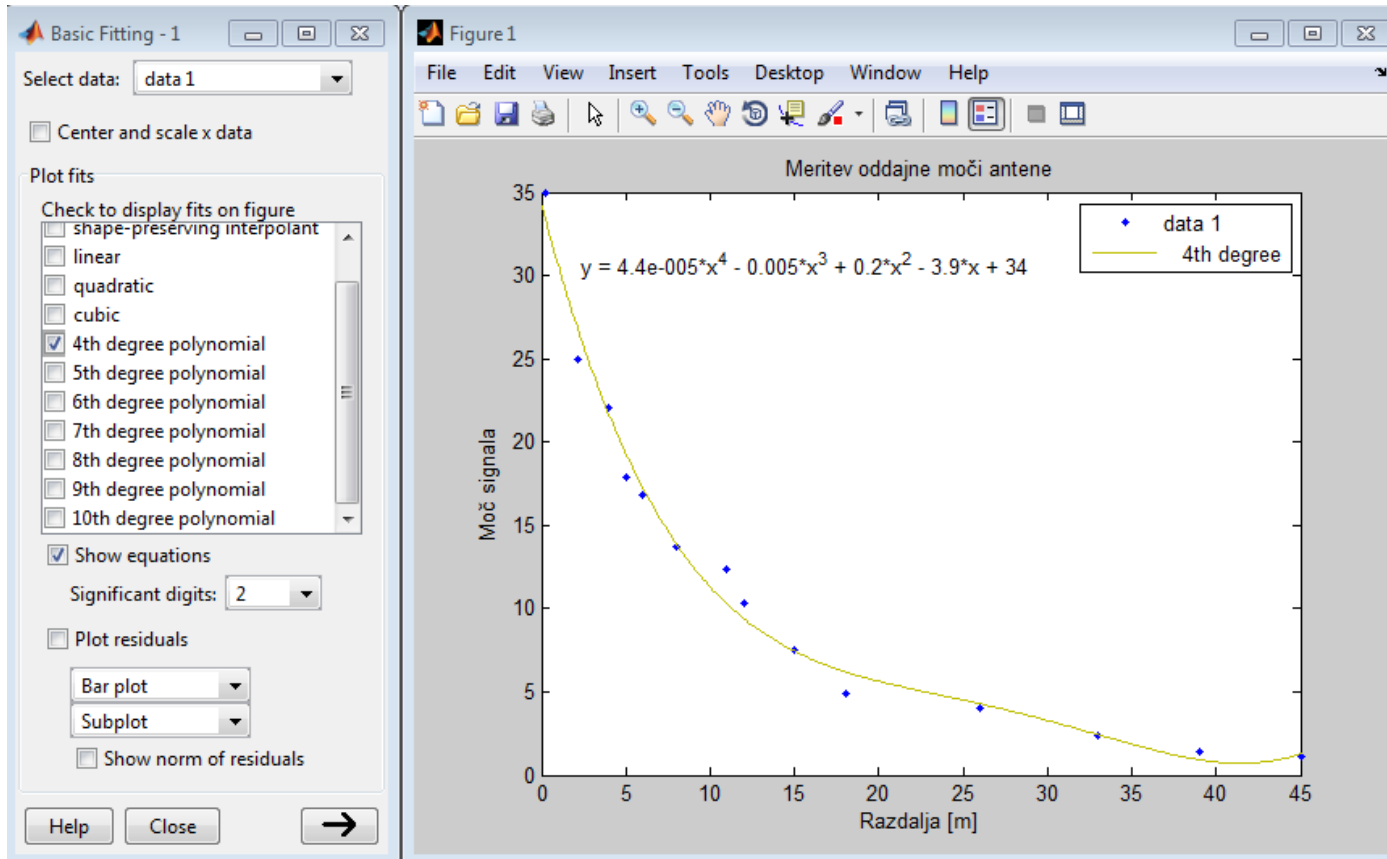
- Spoznali bomo:
 - Basic Fitting Tool
 - Interaktivno prilagajanje krivulj
 - Pridobivanje koeficientov
 - Osnovni ukazi prilagajanja krivulj

Basic Fitting Tool (BFT)



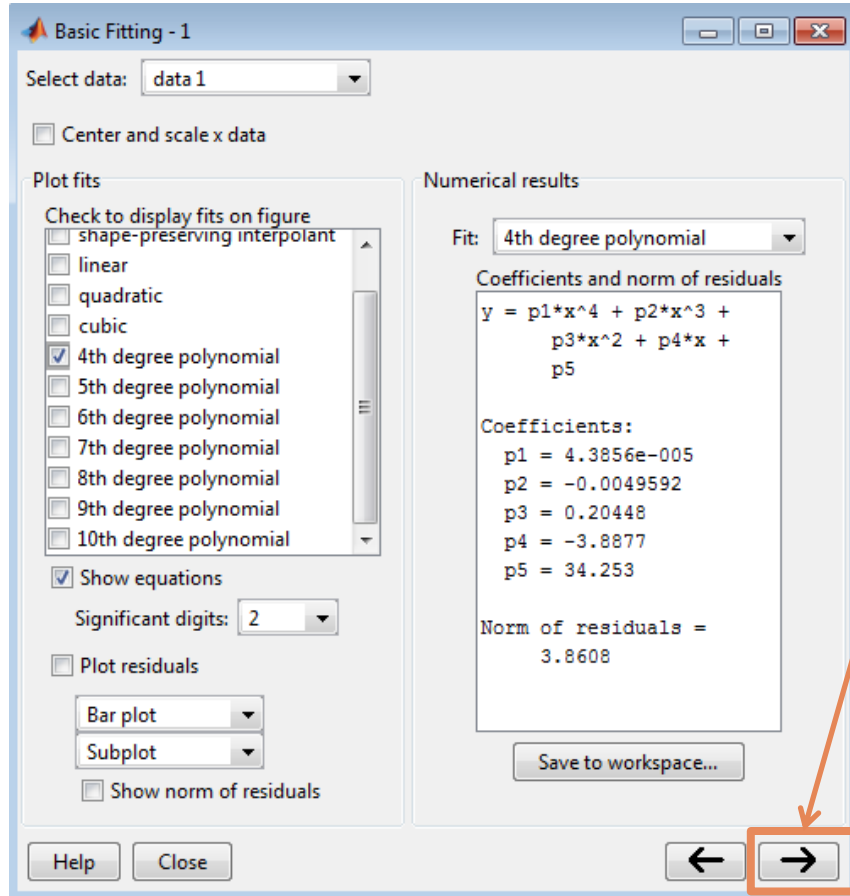
- V oknu grafa, nad katerim želimo izvesti prilagajanje krivulj izberemo: **Tools>Basic Fitting**

Basic Fitting Tool (BFT)

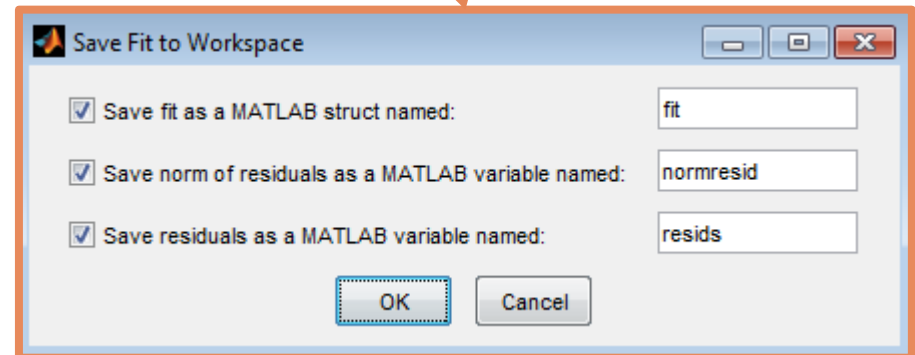


- Izbiramo lahko med različnimi možnostmi prilagajanja krivulj (linearno, kvadratično, kubično, s polinomi višjih redov)
- Možen je tudi izpis enačbe prilagojene krivulje in s tem njena nadaljnja uporaba

BFT – pridobivanje koeficientov



- V spodnjem desnem kotu kliknemo na puščico in razširimo vmesnik BFT
- prikažejo se tudi koeficienti polinoma
- Izberemo Save to workspace – v novem oknu izberemo imena → OK
- Izbrane vrednosti se shranijo v delovno okolje



Ukazi osnovnega prilagajanja

- Polinomsko prilagajanje iz orodja BFT lahko nadomestimo s funkcijama `polyfit` in `polyval` v ukaznem oknu

```
>>  
>> plot(razdalja,moc,'r.')
```

```
>> hold on
```

```
>> p=polyfit(razdalja,moc,4)
```

```
p =
```

```
0.0000 -0.0050 0.2045  
-3.8877 34.2527
```

- `Polyfit`: pridobitev koeficientov

- `Polyval`: določitev funkcijskih vrednosti za določene koeficiente in podano funkcijsko območje

```
>> y=polyval(p,razdalja);
```

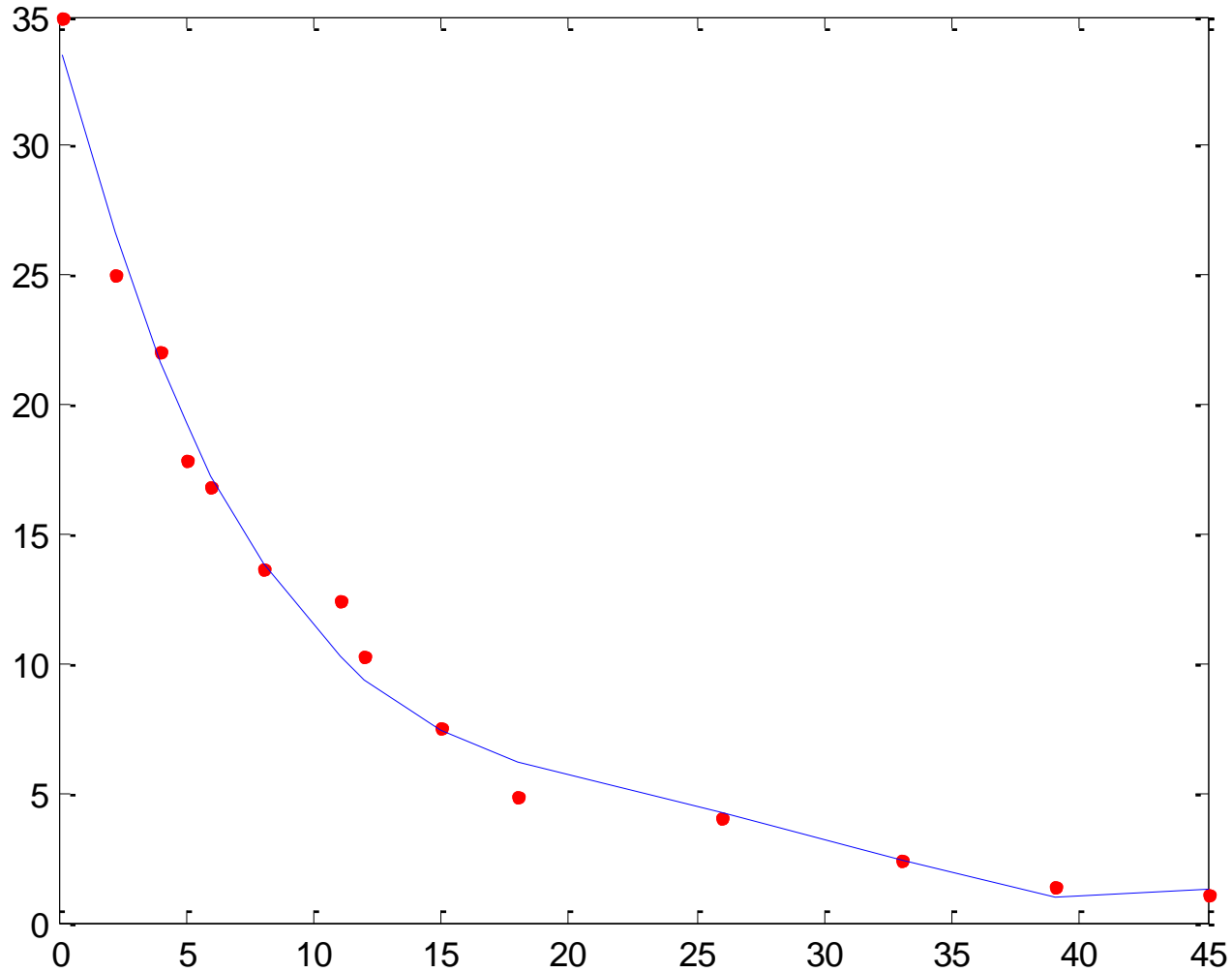
```
>> plot(razdalja,y,'b-')
```

Ukazi osnovnega prilagajanja

■

■

■



Naloga

- Izvedite polinomsko interpolacijo za dano tabelo
- Interpolacijski polinom naj bo 4. stopnje
- Določite koeficiente interpolacijskega polinoma
- Narišite polinom in napišite njegov analitični izraz
- Izračunajte $y(-0.2)$

x	y
-1	-1
0.5	3
1	2
2.3	4.5
2.5	6

Numerična analiza - povzetek

- Uporaba vgrajenih funkcij
- Uporaba pomoči
- Analiza podatkov
 - Prilagajanje krivulj
 - Uporaba BFT
 - Interaktivno prilagajanje krivulj
 - Pridobivanje koeficientov polinomskega prilagajanja
 - Osnovne funkcije za prilagajanje krivulj

Avtomatizacija s skriptami

- Različni tipi Matlab programov

- Matlab Editor
 - Uporaba zgodovine ukazov
 - Uporaba komentarjev
 - Shranjevanje in poganjanje skript


Matlab programi

- Uporaba:
 - Lažje urejanje ukazov
 - Avtomatizacija opravil
 - Ustvarjanje lastnih aplikacij

- Dva osnovna tipa:
 - Skripte (več ukazov združenih v skupino)
 - Funkcije

- Sestavljanje programov – Matlab Editor
 - File > Open > Script

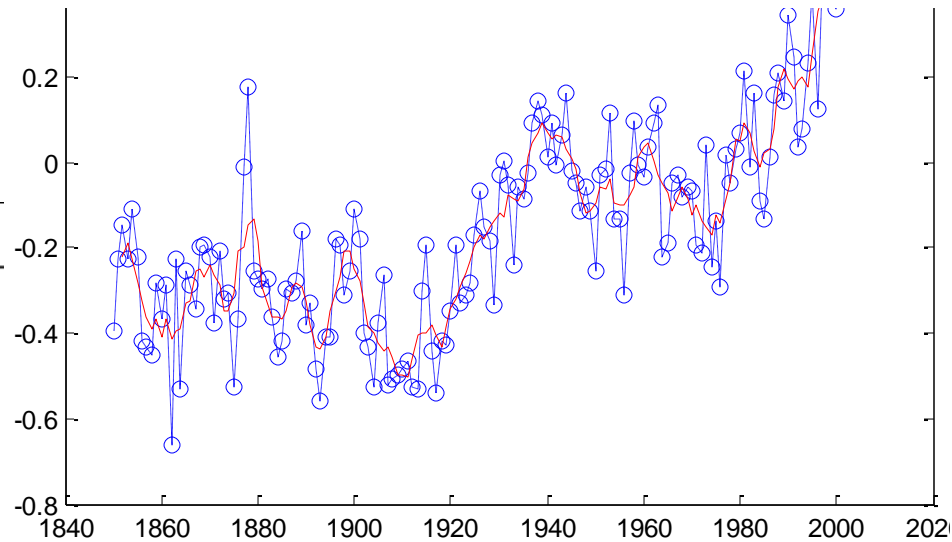
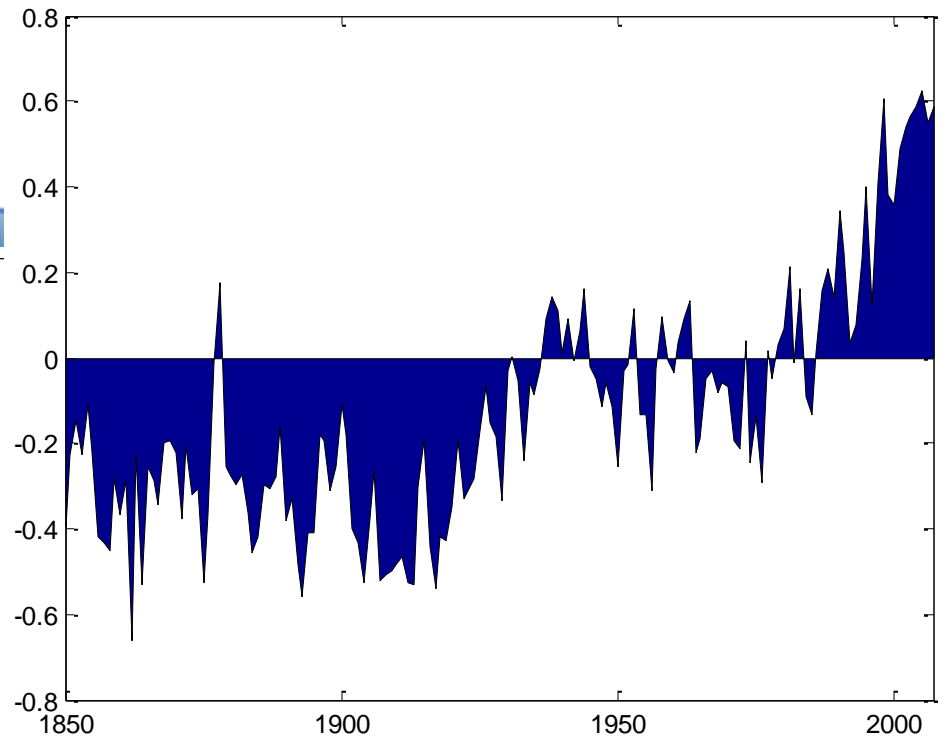
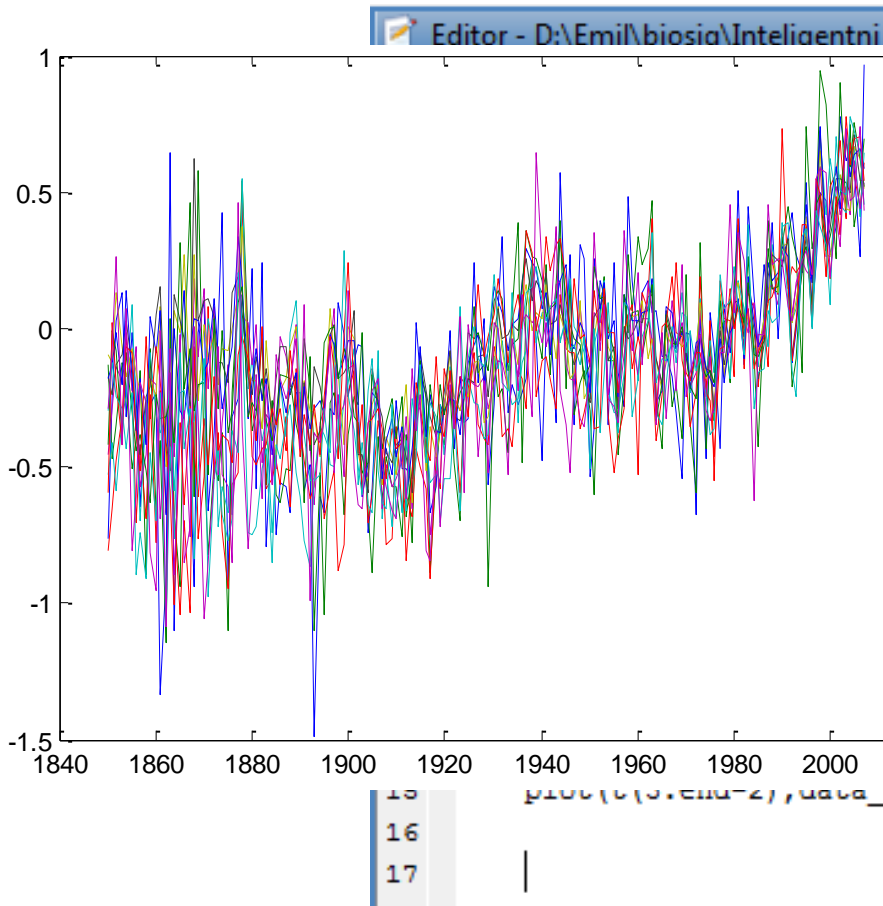
Matlab Editor

- Vnos ukazov je enak kot v ukaznem oknu
 - Veljajo enaka sintaktična pravila
- Pri sestavljanju programov uporabite zgodovino ukazov (kopiranje ukazov)
- Uporaba komentarjev na primernih mestih – začnejo se z oznako `%`, označeni so z **zeleno** barvo
 - Razlaga kode → bolj razumljivo za druge in (po daljšem času neuporabe kode) tudi za avtorja
- Shranjevanje programov:
 - File > Save / Save As ...
 - Shranjeni programi imajo končnico `.m`
 - Privzeta lokacija je trenutna mapa, ki jo lahko spremenimo
- Poganjanje skripte:
 - Pritisni F5 ali klik na gumb 
 - Ime skripte vpišemo v ukazno okno brez končnice

Primer skripte

```
1      % Uvoz podatkov
2      |
3      plot(t,data)
4
5      % Izračun povprečja uvoženih podatkov
6      data_avg = mean(data);
7      figure
8      area(t,data_avg)
9
10     % Izračun drseče sredine (moving average) za 5 enot
11     % z uporabo N-dimenzioanlne konvolucije
12     k=1/5*ones(1,5);
13     data_avg_5=convn(data_avg,k,'valid');
14     plot(t,data_avg,'--o')
15     hold on
16     plot(t(3:end-2),data_avg_5,'r')
17
18
```

Primer skripte



Avtomatizacija s skriptami - povzetek

- Različni tipi Matlab programov

- Uporaba Matlab Editor-ja
 - Uporaba zgodovine ukazov
 - Uporaba komentarjev
 - Shranjevanje in poganjanje skript

Objava rezultatov

- Objava Matlab programov in njihovih rezultatov
 - HTML, XML, DOC, PPT, LaTeX formati
 - Uporaba celic pri programiranju (Cell feature) za avtomatsko objavo datoteke
 - Koda, komentarji in rezultati so predstavljeni v obliki strukturiranega poročila
- HTML poročila
 - Vsebuje kodo s sintakso, komentarje in rezultate za vsako celico

Generiranje poročila

- Odpremo ustrezno skripto – vzemimo zadnji primer
- Omogočiti moramo uporabo celic
 - V orodni vrstici Matlab Editorja izberemo:
Cell > Enable Cell Mode
- Sledi delitev kode na sekcije – celice
 - Začetek celice označimo s simbolom `%%`, ki mu sledi naslov celice
 - Vstavimo jih kolikor se nam zdi potrebno
 - Vsaka celica bo v poročilu predstavljala podpoglavje
- Po končani delitvi lahko končamo poročilo za objavo
 - V orodni vrstici Matlab Editorja izberemo: **File > Publish**
 - Izvrši se koda in ustvari se poročilo v izbranem formatu

HTML poročila

- Vsebuje kodo s sintakso, komentarje in rezultate za vsako celico
- Naslovi celic so predstavljeni kot naslovi poglavij poročila, ustvari se ustrezno kazalo s povezavami
- S klikanjem na povezave kazala se premikamo po poročilu v spletnem brskalniku
- Koda je predstavljena v obarvanih odsekih za boljšo predstavo
- Poročilo se shrani v trenutni delovni mapi v podmapo HTML
- HTML je privzeti format poročil, ki ga lahko spremenimo:
Matlab Editor > File > Publish Configuration > Edit Publish Configuration

Objava rezultatov - povzetek

- Objava Matlab programov in njihovih rezultatov
 - HTML, XML, DOC, PPT, LaTeX formati
- Uporaba celic pri programiranju (Cell feature) za avtomatsko objavo datoteke
- Izdelava HTML poročila
 - Vsebuje kodo s sintakso, komentarje in rezultate za vsako celico

Programiranje v Matlabu

- Pisanje funkcij
 - Modularnost – problem razdelimo na manjše obvladljive dele
 - Ponovna uporaba kode za druge projekte

- Uporaba zank in vejitev
 - Ponavljanje določenih odsekov kode
 - Logično odločanje

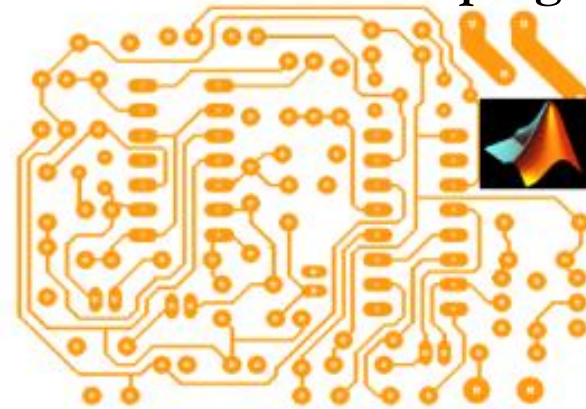
Integracija komponent

Vgrajene funkcije
in orodja



+

Lastni programi



=

Aplikacije



Pisanje funkcij

- Sintaksa
- Upravljanje z vhodnimi in izhodnimi spremenljivkami
- Razlike med skriptami in funkcijami

Sintaksa pisanja funkcij

- Sintaksa pisanja je zelo podobna sintaksi klicanja funkcij

```
function izhod = Ime_funkcije(vhod)
```

```
function [izhod1, izhod2] = Ime_funkcije(vhod1, vhod2, ...)
```

- Izrazu s ključno besedo `function` pravimo deklaracija funkcije
 - Vedno je v prvi vrstici kode funkcije
- Ime shranjene .m datoteke mora biti enako kot ime funkcije v deklaraciji!
- Vhodne spremenljivke so uporabljene le kot vhodni parametri izračunov v funkciji!

Primer

window,'valid'); % Posodobimo časovni vektor, da se ujema z izračunanim povprečjem avg. tout = time(ceil(span/2):length(avg)+ceil(span/2)-1); end" data-bbox="165 192 832 857"/>

```
1 function [ avg, tout ] = movingAverage( data,time,span )
2 %MovingAverage
3 % Za izračun drseče sredine (moving average) uporabimo funkcijo CONVN.
4 % S tem "zgladimo" vhodne podatke.
5 % Vhodni parametri funkcije [avg,tout] = movingAverage(data,time,span)
6 % so podatki za glajenje, pripadajoča časovna os in perioda drseče
7 % sredine. Kot izhodne spremenljivke vrne zglajene podatke in ustrezno
8 % časovno os.
9
10 % Ustvarimo okno za izračun drseče sredine.
11 window = ones(1,span)/span;
12
13 % Uporabimo CONVN
14 avg = convn(data>window,'valid');
15
16 % Posodobimo časovni vektor, da se ujema z izračunanim povprečjem avg.
17 tout = time(ceil(span/2):length(avg)+ceil(span/2)-1);
18
19
20 end
21
```


Skripta vs. funkcija

- Skripta
 - Zaporedje ukazov - enakovredno ukaznemu oknu
 - Avtomatizacija ponavljajočih se opravil
 - Vsi uporabljeni/generirani podatki so shranjeni v delovnem okolju
- Funkcija:
 - Avtomatizacija opravil
 - Ob klicu moramo vnesti vhodne parametre
 - Rezultat so vedno izhodne spremenljivke
 - Uporabimo jih lahko za različne vhode brez spreminjanja kode – modularnost in splošna raba kode
 - v delovno okolje se shranijo samo izhodne spremenljivke, vmesne spremenljivke (uporabljene samo v funkciji) se ne shranijo

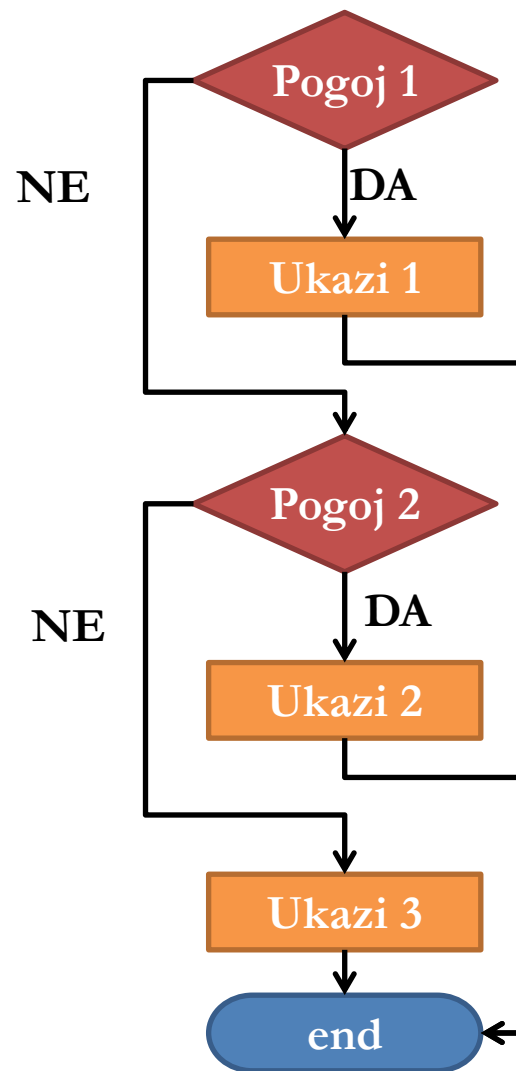
Zanke in vejitve

- **Vejitve**
 - Pogojni stavek `if - elseif - else`

- **Zanke, ponovitveni stavki**
 - `While` zanka
 - `For` zanka

Pogojni stavek `if - elseif - else`

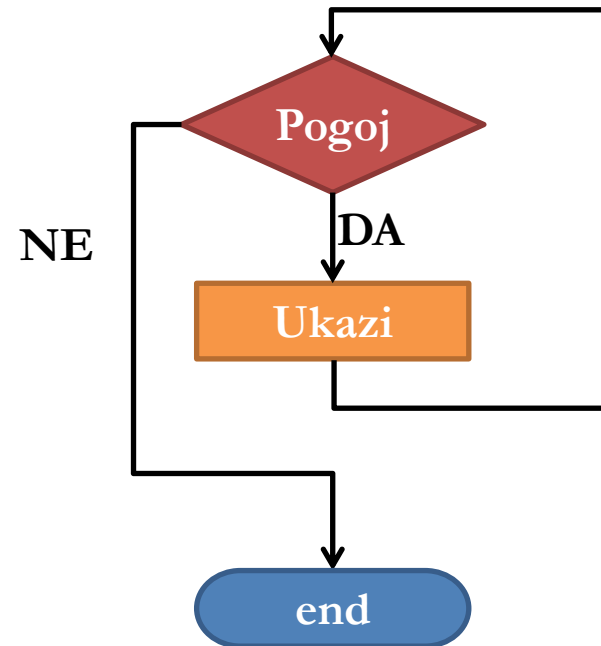
```
if pogoj 1 == true  
    ukazi 1  
elseif pogoj 2 == true  
    ukazi 2  
else  
    ukazi 3  
end
```



While zanka

- Ponavljanje zaporedja ukazov dokler je izpolnjen dani pogoj

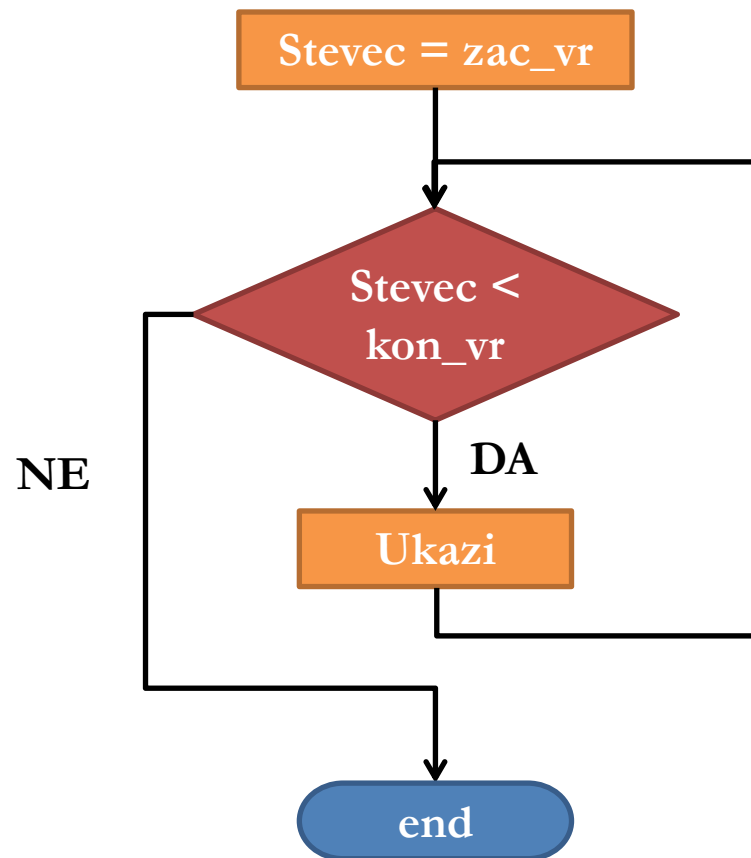
```
while pogoj  
    ukazi  
end
```



For zanka

- Ponavljanje zaporedja ukazov za točno število iteracij
- Možna je predčasna prekinitev

```
for stevec = zac_vr:korak:kon_vr  
    ukazi  
end
```



Naloga

- Z uporabo zank izriši animirano sinusno funkcijo:
 - x os: 0 – 2 pi, s korakom 0.1
 - $y = \sin(x)$
 - Funkcija naj se izriše zaporedoma od prve do zadnje točke na x osi.

- Namig:
 - Sliko je potrebno predhodno odpreti, definirati x in y skalo ter poskrbeti, da jo lahko dopolnjujemo: funkcije `figure`, `axis` in `hold on`
 - Uporabite funkcijo `pause` za zakasnitev

Programiranje – ključne besede

- Spoznali smo osnovne ključne besede: for, if, elseif, else, end, while
- Celoten seznam ključnih besed, ki so nam lahko v pomoč pri programiranju dobimo z ukazom `iskeyword` v ukaznem oknu

```
>> iskeyword
ans =
    'break'
    'case'
    'catch'
    'classdef'
    'continue'
    'else'
    'elseif'
    'end'
    'for'
    'function'
    'global'
    'if'
    'otherwise'
    'parfor'
    'persistent'
    'return'
    'spmd'
    'switch'
    'try'
    'while'
```

Programiranje v Matlabu – povzetek

- Sintaksa za pisanje funkcij
- Upravljanje z vhodnimi in izhodnimi spremenljivkami
- Primerjava skript in funkcij
- Uporaba odločitvenih/vejitvenih stavkov
- Uporaba zank `for` in `while`

Hvala za pozornost

- Matlab okolje
- Matlab osnove
- Reševanje problemov z Matlabom
- Programiranje v Matlabu