

Z Matlabom ali Octave v Numerične metode

Andrej Perne

Fakulteta za elektrotehniko

pomlad 2006 - jesen 2012

Vsebina

UNIX

Osnovni ukazi

Elementarne funkcije

Funkcije za delo z vektorji

Funkcije za delo z matrikami

Programiranje, m-datoteke

Risanje grafov in razno

Ukazi v UNIX-u I

- `mkdir`: nova mapa

Z ukazom `mkdir MojaMapa` ustvarimo novo mapo z imenom `MojaMapa`.

- `cd`: prehajanje med mapami

Z ukazom `cd MojaMapa` se iz mape, ki vsebuje mapo `MojaMapa`, preselimo v mapo `MojaMapa`.

Z ukazom `cd ..` se iz mape `MojaMapa` preselimo v mapo, ki vsebuje mapo `MojaMapa`.

- `rmdir`: izbris mape

Z ukazom `rmdir MojaMapa` izbrišemo mapo z imenom `MojaMapa`.

- `ls` ali `dir`: izpis vseh map in datotek, ki se nahajajo v trenutni mapi

- `man`: pomoč

Ukazi v UNIX-u II

- `cp`: kopiranje datoteke

Z ukazom `cp vaja.txt Vaja` skopiramo datoteko `vaja.txt`, ki se nahaja v trenutni mapi, v mapo `Vaja`, ki se nahaja v trenutni mapi.

- `rm`: brisanje datoteke

Z ukazom `rm vaja.txt` izbrišemo datoteko `vaja.txt`, ki se nahaja v trenutni mapi.

- `mv`: preimenovanje datoteke

Z ukazom `mv vaja.txt naloga.txt` preimenujemo datoteko `vaja.txt` v `naloga.txt`.

- `matlab`: zagon Matlab

- `octave`: zagon Octave

Osnovni ukazi `help`, `clear`, `exit`, `who`, `what`

- `help`: pomoč v Matlabu oz. Octave
Z ukazom `help exp` izpišemo uporabo in primere za ukaz `exp`.
- `clear`: izbris spremenljivk
Z ukazom `clear all` izbrišemo vse spremenljivke.
Z ukazom `clear x` izbrišemo spremenljivko `x`.
- `exit` ali `quit`: izhod iz Matlabu oz. Octave
- `who`: informacija o trenutno uporabljenih spremenljivkah
- `whos`: informacija o trenutno uporabljenih spremenljivkah v daljši obliki
- `what`: izpis vseh m-datotek v trenutni mapi

Ukaza format in diary

- `format`: oblikovanje izpisa na zaslonu

Z ukazom `format long` spremenimo izpis števil v daljši izpis na 14 decimalk.

Z ukazom `format short` spremenimo izpis števil v krajši izpis na 4 decimalke. Ta izpis je privzet.

- `diary`: zapis v datoteko

Z ukazom `diary vaja.txt` ustvarimo tekstovno datoteko `vaja.txt`, kamor se zapisuje vse kar se izpiše na zaslon.

Z ukazom `diary off` izklopimo zapis na datoteko.

Z ukazom `diary on` ponovno vklopimo zapis na datoteko.

Algebrske operacije, decimalna števila, konstante

Za računanje z realnimi (in kompleksnimi) števili imamo na voljo naslednje algebrske operacije.

- $+$, $-$: seštevanje, odštevanje
- $*$, $/$: množenje, deljenje
- $^$: potenciranje

Decimalno število zapišemo z **decimalno piko**.

Najpogostejše konstante:

- pi : število π
- $\text{exp}(1)$: osnova naravnega logaritma e
- i ali j : imaginarna enota i

Relacijski in logični (Boolovi) operatorji

Relacijski operatorji:

- `==` : enako
- `~=` : različno
- `<` : manjše
- `<=` : manjše ali enako
- `>` : večje
- `>=` : večje ali enako

Logični operatorji:

- `&` : logični in (in hkrati)
- `|` : logični ali
- `~` : logični ne (negacija)

Funkcije `sqrt`, `abs`, `exp`, `log`, `log10`

Z ukazom `help elfun` izpišemo spisek vseh elementarnih funkcij (samo v Matlabu). Argumente funkcij pišemo v okroglih oklepajih ().

- `sqrt`: kvadratni koren (\sqrt{x})
- `abs`: absolutna vrednost ($|x|$)
- `exp`: eksponentna funkcija (e^x)
- `log`: naravni logaritem ($\ln x$)
- `log10`: desetiški logaritem ($\log x$)

- `sqrt(9) → 3`
- `abs(-1) → 1`
- `exp(0) → 1`
- `log(1) → 0`
- `log10(10) → 1`

Trigonometrične in krožne funkcije

- `sin`: sinus ($\sin x$)
- `cos`: kosinus ($\cos x$)
- `tan`: tangens ($\operatorname{tg} x$)
- `cot`: kotangens ($\operatorname{ctg} x$)

- `asin`: arcus sinus ($\arcsin x$)
- `acos`: arcus kosinus ($\arccos x$)
- `atan`: arcus tangens ($\operatorname{arctg} x$)
- `acot`: arcus kotangens ($\operatorname{arcctg} x$)

```
• sin(pi) → 0
• cos(0) → 1
• tan(pi/4) → 1
• cot(pi/2) → 0
```

```
• asin(1) → 1.5708
• acos(-1) → 3.1416
• atan(1) → 0.7854
• acot(-1) → -0.7854
```

Funkcije nad kompleksnimi števili

- `real`: realna komponenta
- `imag`: imaginarna komponenta
- `conj`: konjugirana vrednost kompleksnega števila
- `abs`: absolutna vrednost
- `angle`: kot

```
• real(4+3*i) → 4
• imag(4+3*i) → 3
• conj(4+3*i) →
  4-3*i
• abs(4+3*i) → 5
• angle(4+3*i) →
  0.6435
```

Zaokrožitvene funkcije in ostanek pri deljenju

- `fix`: zaokrožanje na najbližje celo število proti 0
- `floor`: zaokrožanje navzdol
- `ceil`: zaokrožanje navzgor
- `round`: zaokrožanje k najbližjemu celemu številu
- `mod(x, y)`: ostanek pri deljenju x z y
- `rem(x, y)`: ostanek pri deljenju x z y

```
• fix(1.8) → 1
  fix(-1.8) → -1
• floor(1.8) → 1
• ceil(1.8) → 2
• round(-1.8) → -2
```

```
• mod(5, 2) → 1
  mod(-5, 2) → 1
• rem(5, 2) → 1
  rem(-5, 2) → -1
```

Specialne funkcije sign, gamma in beta

- Funkcija `sign(x)` vrne predznak števila x .

$$\text{sign}(x) = \begin{cases} 1; & x > 0 \\ 0; & x = 0 \\ -1; & x < 0 \end{cases}$$

- `gamma(x)`: funkcija gama

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt, x > 0$$

- `beta(x, y)`: funkcija beta

$$B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt, \\ x > 0, y > 0$$

```
sign(-5) → -1
sign(5) → 1
```

```
gamma(1/2) →
1.7725
sqrt(pi) →
1.7725
```

```
beta(1, 2) → 0.5
```

Faktorji in praštevila

- `primes(N)`: izpis vseh praštevil, ki so manjša ali enaka N

```
primes(10) → 2 3 5 7
```

- `factor(N)`: izpis vseh praštevilskih faktorjev števila N

```
factor(10) → 2 5
```

- `isprime(p)`: ali je število p praštevilo?

```
isprime(2) → 1
```

```
isprime(4) → 0
```

Vektorji in polja

- Vektor pišemo v oglatih oklepajih `[]`. Decimalno število zapišemo z **decimalno piko**. Komponente vektorja ločimo z vejicami ali presledki.

```
x = [1.65, 2.23, 1.99]
```

```
y = [1 2 3 4 5]
```

- Do komponent vektorja dostopamo z okroglimi oklepaji `()`. Več komponent izpišemo z operatorjem `:`. Kot primer izpišimo prvo komponento vektorja x ter drugo do četrto komponento vektorja y .

```
x(1) → 1.65
```

```
y(2:4) → [2 3 4]
```

Algebrske operacije s polji

Pri poljih je potrebno pri nekaterih operacijah pred operatorjem uporabljati **PIKO**.

- $+$, $-$: seštevanje, odštevanje
- $.*$, $.^$: množenje, potenciranje
- $./$, $.\backslash$: desno deljenje, levo deljenje
- $'$: transponiranje

Skalarni in vektorski produkt vektorjev izračunamo s funkcijama **dot** in **cross**.

- $\text{dot}(x, y)$: skalarni produkt vektorjev x in y
- $\text{cross}(x, y)$: vektorski produkt vektorjev x in y

Ukaz linspace in operator :

- Z ukazom `linspace(a,b,n)` zgradimo vektor z n ekvidistantnimi komponentami med a in b . Kot primer definirajmo vektor s sedmimi ekvidistantnimi komponentami med 0 in 3.

```
z = linspace(0,3,7) → z=[0 0.5 1 1.5 2 2.5 3]
```

- Z operatorjem `:` zgradimo vektor z ekvidistantnimi komponentami. Privzeti korak je 1. Kot primer definirajmo vektora med 0 in 6 s korakoma 1 in 2.

```
z = 0:6 → z = [0 1 2 3 4 5 6]
```

```
z = 0:2:6 → z = [0 2 4 6]
```

Vektorske funkcije `length`, `min`, `max`, `sum`, `prod`

- `length`: število komponent
- `min`: najmanjša komponenta
- `max`: največja komponenta
- `sum`: vsota komponent
- `prod`: produkt komponent
- `cumsum`: kumulativna vsota komponent
- `cumprod`: kumulativen produkt komponent

```
• length(x) → 3
• min(y) → 1
• max(y) → 5
• sum(x) → 5.87
• prod(y) → 120
• cumsum(x) →
  [1.65 3.88 5.87]
• cumprod(y) →
  [1 2 6 24 120]
```

Funkcija `norm`

Z ukazom `norm` izračunamo normo vektorja. Privzeta je evklidska ali druga norma.

- `norm(x)` ali `norm(x, 2)`: evklidska norma

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$$

- `norm(x, 1)`: prva norma

$$\|x\|_1 = \sum_{i=1}^n |x_i|$$

- `norm(x, inf)`: neskončna norma

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

- `norm(y)` → 7.4162
- `norm(y, 1)` → 15
- `norm(y, inf)` → 5

Vektorske funkcije `diff`, `sort`, `unique`, `find`

- `diff(x)`: vektor razlik med sosednjima komponentama vektorja x

```
diff(y) → [1 1 1 1]
```

- `sort(x)`: urejen vektor x
- `issort(x)`: ali je vektor x urejen?
- `unique(x)`: urejen vektor x , kjer se vsako število pojavi samo enkrat
- `find(x)`: iskanje neničelnih elementov v vektorju x , kjer kot rezultat dobimo vektor indeksov, kjer se nahajajo neničelni elementi

```
z = [1 0 2];  
find(z) → [1 3]
```

Statistične funkcije mean, var, std, cov

- **mean** (**x**): srednja vrednost

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- **var** (**x**): disperzija

$$\text{var}(x) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

- **std** (**x**): standardna deviacija

$$\text{std}(x) = \sqrt{\text{var}(x)}$$

- **cov** (**x**, **y**): kovarianca

```
• mean(y) → 3
  mean(x) → 1.9567
• var(y) → 2.5
  var(x) → 0.0849
• std(y) → 1.5811
  std(x) → 0.2914
• cov(x, y(1:3)) →
  0.0849 0.1700
  0.1700 1.0000
```

Množice

Vektorje v primeru uporabe teh funkcij razumemo kot množice komponent vektorjev.

- `union(x, y)`: unija množic x in y
- `intersect(x, y)`: presek množic x in y
- `setdiff(x, y)`: razlika množic x in y
- `ismember(p, x)`: ali je število p element množice x ?

```
ismember(2, x) → 0
```

```
ismember(2, y) → 1
```

Matrike

- Matriko pišemo v oglatih oklepajih $[]$. Decimalno število napišemo z **decimalno piko**. Elemente v vrstici ločimo s presledkom ali vejico, vrstice pa ločimo s podpičjem.

$$A = [1 \ 2 \ 3; \ 4 \ 5 \ 6; \ 7 \ 8 \ 9]$$

- Do elementov matrike dostopamo z okroglimi oklepaji $()$. Več elementov izpišemo z operatorjem $:$. Kot primer izpišimo element matrike A , ki leži v drugi vrstici in tretjem stolpcu, ter podmatriko matrike A od prve do druge vrstice in od drugega do tretjega stolpca.

$$A(2, 3) \rightarrow 6$$

$$A(1:2, 2:3) \rightarrow \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$$

Algebrske operacije z matrikami

- $+$, $-$: seštevanje, odštevanje
- $*$, $^$: množenje, potenciranje
- $/$, \backslash : desno deljenje, levo deljenje
- $'$ ali `transpose` : transponiranje

Nekatere druge operacije z matrikami:

- `fliplr(A)` : zrcaljenje matrike A levo — desno
- `flipud(A)` : zrcaljenje matrike A gor — dol

Matrične funkcije `size`, `max`, `min`, `sum`

- `size(A)`: dimenzije matrike A (število vrstic in stolpcev)

```
size(A) → 3 3
```

- `max(A)`: vektor največjih elementov v stolpcih matrike A

```
max(A) → [7 8 9]
```

```
max(max(A)) → 9
```

- `min(A)`: vektor najmanjših elementov v stolpcih matrike A

```
min(A) → [1 2 3]
```

```
min(min(A)) → 1
```

- `sum(A, 1)`: vektor vsot elementov po stolpcih matrike A (privzeto)
`sum(A, 2)`: vektor vsot elementov po vrsticah matrike A

Matrična funkcija `diag`

- `diag(A)`: diagonalna matrika A (privzeto $n = 0$)
`diag(A, n)`: n -ta naddiagonalna (poddiagonalna) matrika A
`diag(A, 1)`: prva naddiagonalna matrika A
`diag(A, -1)`: prva poddiagonalna matrika A

```
diag(A) → [1 5 9]
```

- `diag(x)`: sestavimo diagonalno matriko A , ki ima na diagonalni elemente vektorja x
`diag(diag(A))`: diagonalna matrika, ki ima na diagonalni diagonalne elemente matrike A

```
diag([1 2 3]) →  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$      diag(diag(A)) →  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{bmatrix}$ 
```

Matrične funkcije `triu`, `tril`

- `triu(A)`: zgornji trikotnik matrike A (privzeto $n = 0$)
`triu(A, n)`: zgornji trikotnik od n -te naddiagonale naprej
`triu(A, 1)`: strogi zgornji trikotnik matrike A

$$\text{triu}(A) \rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 6 \\ 0 & 0 & 9 \end{bmatrix} \quad \text{triu}(A, 1) \rightarrow \begin{bmatrix} 0 & 2 & 3 \\ 0 & 0 & 6 \\ 0 & 0 & 0 \end{bmatrix}$$

- `tril(A)`: spodnji trikotnik matrike A (privzeto $n = 0$)
`tril(A, -n)`: spodnji trikotnik od n -te poddiagonale naprej
`tril(A, -1)`: strogi spodnji trikotnik matrike A

$$\text{tril}(A) \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 4 & 5 & 0 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{tril}(A, -1) \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 0 \\ 7 & 8 & 0 \end{bmatrix}$$

Matrične funkcije za konstrukcijo matrik

- `ones(n, m)`: matrika dimenzije $n \times m$ iz samih enic

$$\text{ones}(2, 3) \rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- `zeros(n, m)`: matrika dimenzije $n \times m$ iz samih ničel

$$\text{zeros}(2, 2) \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- `eye(n)`: identična matrika dimenzije $n \times n$

$$\text{eye}(2) \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- `rand(n, m)`: naključna matrika dimenzije $n \times m$

Ukaza reshape in sparse

- Z ukazom `reshape(x, n, n)` iz vektorja dolžine n^2 konstruiramo matriko dimenzije $n \times n$ tako, da elemente zložimo po stolpcih.

```
x = [1 2 3 4 5 6 7 8 9];  
reshape(x, 3, 3) →  $\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$ 
```

- Z ukazom `sparse(i, j, a, m, n)` sestavimo razpršeno matriko dimenzije $m \times n$, kjer so v vektorju i shranjeni indeksi vrstic, v vektorju j indeksi stolpcev in v vektorju a elementi matrike.

```
i = [1 2]; j = [3 1]; a = [9 5];  
sparse(i, j, a, 3, 3) →  $\begin{bmatrix} 0 & 0 & 9 \\ 5 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ 
```

Linearna algebra I

- $\det(A)$: determinanta matrike A
- $\text{eig}(A)$: lastne vrednosti in lastni vektorji matrike A
- $\text{poly}(A)$: karakteristični polinom matrike A
- $\text{norm}(A)$: norma matrike A
- $\text{rank}(A)$: rang matrike A
- $\text{inv}(A)$: inverz matrike A
- $\text{cond}(A)$: pogojenostno število matrike A
- $\text{trace}(A)$: sled matrike A (vsota diagonalnih elementov)

Linearna algebra II

- $\text{rref}(A)$: reducirana oblika matrike A
- $\text{null}(A)$: ničelni prostor matrike A
- $\text{orth}(A)$: ortonormalna baza matrike A
- $\text{lu}(A)$: LU razcep matrike A
- $\text{chol}(A)$: razcep Choleskega matrike A
- $\text{qr}(A)$: QR razcep matrike A
- $\text{svd}(A)$: singularni razcep matrike A

Sistem linearnih enačb

Linearen sistem $Ax = b$ rešimo z ukazom $x = A \setminus b$ (levo deljenje). Kot primer vzemimo

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 5 \\ 4 \end{bmatrix}.$$

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

```
b = [4 5 4]';
```

```
x = A \ b → [1 1 1]'
```


Zanki for in while

Za programiranje imamo na voljo zanki `for` in `while`. For zanka izvede določeno število ponovitev, while zanka pa se izvaja dokler je pogoj izpolnjen. Iz zanke lahko izstopimo z ukazom `break`.

- `for` spremenljivka = pogoj
stavki
`end`

- `while` pogoj
stavki
`end`

```
x = zeros(1,5);  
for i=1:5  
    x(i) = i ^ 2;  
end;
```

```
a = 100;  
while a > 10  
    a = a/2;  
end;
```

Stavka `if in switch`

Za programiranje imamo na voljo imamo na voljo pogojna stavka `if` in `switch`, ki izvajata ukaze glede na pogoj.

- `if` pogoj
stavki
`elseif` pogoj
stavki
`else` pogoj
stavki
`end`
- `switch` pogoj
`case` vrednost
stavki
`case` vrednost1, vrednost2
stavki
`otherwise`
stavki
`end`

```
if n > 5
    a = 1;
else
    a = 0;
end;
```

m-datoteke

- Poznamo dva tipa m-datotek: **skripte** in **funkcije**.
- Skripte so zgolj daljše kode, ki jih ne želimo pisati v ukazno vrstico.
- Funkcije pa se vedno začnejo z ukazom **function**. Funkcijam podamo vhodne in izhodne argumente. Tako napisano funkcijo kličemo na enak način kot vgrajene funkcije. Ime m-datoteke naj bo enako imenu funkcije. Kot primer napišimo funkcijo, ki izračuna obseg in ploščino kroga z danim polmerom r .

```
function [o,p] = krog(r)
o = 2*pi*r;
p = pi*r ^ 2;
```

```
[o,p] = krog(3)
o = 18.8496
p = 28.2743
```

Ukaza `inline` in `eval`

- Z ukazom `inline` definiramo notranje funkcije. Kot primer definirajmo funkcijo $f(x) = x^2$ in izračunajmo njeno vrednost v točki 2.

```
f = inline('x ^ 2', 'x');  
f(2) → 4
```

- Z ukazom `eval` izračunavamo funkcije. Argument mora biti podan kot niz. Funkcijo $f(x) = x^2$ zapišimo kot niz in izračunajmo njeno vrednost v točki 2.

```
f = 'x ^ 2';  
x = 2;  
eval(f) → 4
```

Ukaz `plot`: risanje grafov funkcij

Z ukazom `plot(x, y)` narišemo graf funkcije, ki je podana z vektorjema `x` (x koordinate) in `y` (y koordinate). Kot primer narišimo funkcijo $y = \sin x$ na intervalu $[0, 2\pi]$.

```
x = 0:0.01:2*pi;  
y = sin(x);  
plot(x, y)
```

Sliki lahko dodamo naslov, oznake na obeh oseh in legendo.

- `title`: naslov
- `xlabel`: oznake na x -osi
- `ylabel`: oznake na y -osi
- `legend`: legenda

Ukazi za delo s slikami

- `axis`: osi

Z ukazom `axis equal` določimo, da sta enoti na obeh oseh enaki.

Z ukazom `axis([-5, 5, -3, 3])` določimo, da se graf funkcije izriše na intervalu $[-5, 5]$ na x -osi in $[-3, 3]$ na y -osi.

- `hold`: zadržanje slike

Z ukazom `hold on` zadržimo sliko, da lahko na isto sliko narišemo nov graf.

Z ukazom `hold off` izklopimo zadržanje slike.

- `grid`: mreža

Z ukazom `grid on` vklopimo mrežo.

Z ukazom `grid off` izklopimo mrežo.

Poleg osnovnega ukaza `plot` imamo za izris slik na voljo še druge ukaze, npr. `semilogy` (logaritemska skala na y -osi) in `plot3` (grafi v 3 dimenzijah).

Opcije za izris grafov funkcij

Obsežen seznam opcij dobimo z ukazom `help plot`. Nekatere pomembnejše so našteje spodaj. Zapišemo jih v enojnih navednicah ' ' .

- **Barve:** `b` modra, `g` zelena, `r` rdeča, `y` rumena, `k` črna, `w` bela.
- **Črte:** `-` polna, `:` pikčasta, `-` črtkasta, `-.` pikčasto-črtkasta.
- **Debelina črt:** `'LineWidth'`, `2` dvojna debelina.
- **Markerji:** `.` točka, `o` krogec, `x` x-znak, `+` plusek, `*` zvezdica.

Primer opremljenega grafa

Narišimo na isto sliko grafa funkcij $y = \sin x$ z rdečo črtkasto črto ter $y = \cos x$ z modro polno črto na intervalu $[-\pi, 2\pi]$. Maksimume in minumume obeh funkcij označimo z zelenimi krogi, ničle pa s črnimi x -znaki. Dodajmo še naslov, oznake na obeh oseh, mrežo ter legendo.

```
x = -pi:0.01:2*pi; y = sin(x); z = cos(x);
plot(x,y,'r-',x,z,'b-'); hold on; grid on;
axis equal; axis([-pi 2*pi -1.1 1.1]);
title('Grafa funkcij sinus in kosinus');
xlabel('x'); ylabel('y');
xx = [-pi -pi/2 0 pi/2 pi 3*pi/2 2*pi];
plot(xx,[-1 -1 1 1 -1 -1 1],'go');
plot(xx,zeros(1,7),'kx');
legend('sinus','kosinus','ekstremi','nicle');
```


Ukazi za delo s polinomi

- Z ukazom `polyval(p, x)` izračunamo vrednost polinoma p v točki x . Polinom podamo z vektorjem koeficientov, ki so urejeni po padajočih potencah x -a.
- Z ukazom `roots(p)` izračunamo ničle polinoma p . Kot primer izračunajmo vrednost polinoma $p(x) = x^3 - 3x + 2$ v točki $x = 2$ ter njegove ničle.

```
polyval([1 0 -3 2], 2) → 4
```

```
roots([1 0 -3 2]) → [-2 1 1]
```

- Z ukazom `polyfit(x, y, n)` določimo koeficiente interpolacijskega polinoma stopnje n , kjer za interpolacijske točke vzamemo pare komponent iz vektorjev x in y .
- Z ukazom `spline(x, y)` določimo kubični zlepek, kjer za interpolacijske točke vzamemo pare komponent iz vektorjev x in y .

Ukaz quad: numerično integriranje

- Z ukazom `quad(f, a, b)` numerično izračunamo določeni integral funkcije f na intervalu $[a, b]$. Kot primer izračunajmo integral $\int_0^1 x^2 dx$.

```
f = inline('x ^ 2');  
quad(f, 0, 1) → 1/3
```

- Z ukazom `trapz(y)` z uporabo trapeznega pravila izračunamo določeni integral funkcije, ki je podana z vektorjem funkcijskih vrednosti y za enotske razmike. Kot primer izračunajmo integral $\int_0^3 (x + 1) dx$, kjer je vhodni podatek vektor $y = (1, 2, 3, 4)$.

```
y = [1 2 3 4];  
trapz(y) → 7.5
```