

Octave

Numerične metode (NTF)

Karla Ferjančič, Vito Vitrih

4. oktober 2018

Vsebina

Uvod

Elementarne funkcije

Vsebina

Uvod

Elementarne funkcije

Funkcije za delo z vektorji

Vsebina

Uvod

Elementarne funkcije

Funkcije za delo z vektorji

Funkcije za delo z matrikami

Vsebina

Uvod

Elementarne funkcije

Funkcije za delo z vektorji

Funkcije za delo z matrikami

Programiranje, m-datoteke

Vsebina

Uvod

Elementarne funkcije

Funkcije za delo z vektorji

Funkcije za delo z matrikami

Programiranje, m-datoteke

Risanje grafov in razno

Octave

- ▶ Gre za **brezplačen klon** bolj znanega programskega jezika **Matlab**.
- ▶ Dostopen je na www.gnu.org/software/octave/download.html.
- ▶ Je dokaj **preprost** za uporabnika.
- ▶ Omogoča **hitro** in **pregledno** programiranje.
- ▶ prilagojen je za **matrični račun**
 - ▶ osnovni element je matrika,
 - ▶ skalarji in vektorji so le poseben primer matrik.

Octave

- ▶ Ima veliko število **vgrajenih** funkcij.
- ▶ Ima precej zmogljivo **grafiko**.
- ▶ **Loči** med malimi in velikimi črkami.
- ▶ Vsak ukaz končamo s tipko **Enter**.
- ▶ Če na konec vrstice napišemo **;**, se vsebina izračuna ne bo izpisala.
- ▶ **Komentar** napišemo tako, da na začetek vrstice damo znak **%**.

Ukaz `format`

- ▶ `format`: oblikovanje izpisa na zaslonu.
- ▶ `format long`: spremeni izpis števil v daljši izpis na 14 decimalk.
- ▶ `format short`: spremeni izpis števil v krajši izpis na 4 decimalke.

Ta izpis je privzet.

Ukaz diary

- ▶ `diary`: zapis v datoteko
Z ukazom `diary dnevnik.txt` ustvarimo tekstovno datoteko `dnevnik.txt`, kamor se zapisuje vse kar se izpiše na zaslon.
- ▶ `diary off`: izklopimo zapis
- ▶ `diary on`: ponovno vklopimo zapis

Izpis na zaslon

- ▶ `disp`: izpis spremenljivke ali teksta na zaslon.

Primer: `disp('niz')` izpiše besedo niz.

Primer: `disp(a)` izpiše vrednost spremenljivke a.

- ▶ `error`: izpiše nek niz (običajno opozorilo) in konča izvajanje.

Primer: `error('niz')` izpiše besedo niz in konča izvajanje.

Algebrske operacije

Za računanje s števili imamo naslednje algebrske operacije.

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

Definiranje spremenljivke

- ▶ V Octavu za definiranje neke spremenljivke uporabljamo znak `=`.
- ▶ *Primer:* $x = 2$.

Relacijski operatorji

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

Logični operatorji

- ▶ $\&$: logični in (in hkrati)
- ▶ $|$: logični ali
- ▶ $\sim$: logični ne (negacija)

Funkcije `sqrt`, `abs`

- ▶ Argumente funkcij pišemo v okroglih oklepajih `( )`.
- ▶ `sqrt`: kvadratni koren ($\sqrt{x}$)
Primer: `sqrt(9) → 3`
- ▶ `abs`: absolutna vrednost ($|x|$)
Primer: `abs(-1) → 1`

Funkcije $\exp$, $\log$, $\log_{10}$

- ▶ $\exp$: eksponentna funkcija (e^x)
Primer: $\exp(0) \rightarrow 1$
- ▶ $\log$: naravni logaritem ($\ln x$)
Primer: $\log(1) \rightarrow 0$
- ▶ $\log_{10}$: desetiški logaritem ($\log x$)
Primer: $\log_{10}(10) \rightarrow 1$

Posebna števila

- ▶ i : imaginarna enota i .
- ▶ π : ploščina enotskega kroga.
- ▶ e : Eulerjevo število.
- ▶ ∞ : znak za neskončno.

Funkcije nad kompleksnimi števili

- ▶ `real`: realni del kompleksnega števila

Primer: `real(2+3*i)` → 2

- ▶ `imag`: imaginarni del kompleksnega števila

Primer: `imag(2+3*i)` → 3

- ▶ `conj`: konjugiranje kompleksnega števila

Primer: `conj(2+3*i)` → $2-3*i$

Zaokrožitvene funkcije

- **fix**: zaokrožanje na najbližje celo število proti 0

Primer: `fix(1.8)` → 1

Primer: `fix(-1.8)` → -1

- **floor**: zaokrožanje navzdol

Primer: `floor(1.8)` → 1

- **ceil**: zaokrožanje navzgor

Primer: `ceil(1.8)` → 2

- **round**: zaokrožanje k najbližjemu celemu številu

Primer: `round(-1.8)` → -2

Ostanek pri deljenju

- ▶ `mod(x, y)` : ostanek pri deljenju x z y

Primer: `mod(5, 2) → 1`

Primer: `mod(-5, 2) → 1`

- ▶ `rem(x, y)` : ostanek pri deljenju x z y

Primer: `rem(5, 2) → 1`

Primer: `rem(-5, 2) → -1`

Trigonometrične in krožne funkcije

- ▶ `sin`: funkcija sinus ($\sin x$)
- ▶ `cos`: funkcija kosinus ($\cos x$)
- ▶ `tan`: funkcija tangens ($\tan x$)
- ▶ `asin`: funkcija arcus sinus ($\arcsin x$)
- ▶ `acos`: funkcija arcus kosinus ($\arccos x$)
- ▶ `atan`: funkcija arcus tangens ($\arctan x$)

Funkcija `sign`

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

Primer: `sign(-5)` → -1

Primer: `sign(5)` → 1

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

Vektorji in polja

- ▶ Vektor pišemo v oglatih oklepajih. Decimalno število zapišemo z decimalno piko.

Primer: $x = [1.65, 2.23, 1.99]$

Primer: $x = [1.65, 2.23, 1.99]'$

- ▶ Komponente vektorja ločimo z vejicami ali presledki.

Primer: $y = [1 \ 2 \ 3 \ 4 \ 5]$

- ▶ Do **komponent** vektorja dostopamo z **okroglimi** oklepaji.

Primer: **prva** komponenta vektorja x : $x(1) \rightarrow 1.65$

Primer: **druga do četrta** komponenta vektorja y :

$y(2:4) \rightarrow [2 \ 3 \ 4]$

Ukaz `linspace` in operator `:`

- ▶ `linspace(a,b,n)` : zgradi vektor z n ekvidistantnimi komponentami med a in b

Primer:

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

- ▶ Operator `:` se prav tako uporablja pri generiranju vektorja. Privzeti korak je 1.

Primer: `x=0:6` → `x=[0 1 2 3 4 5 6]`

Primer: `x=0:2:6` → `x=[0 2 4 6]`

Algebrske operacije s polji

- ▶ Pri poljih je potrebno pri nekaterih operacijah pred operatorjem uporabljati **PIKO**.
- ▶ $+$, $-$: seštevanje, odštevanje
- ▶ $\cdot$, $\wedge$: množenje, potenciranje
- ▶ $\cdot /$, $\cdot \backslash$: desno deljenje, levo deljenje

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

- ▶ `length(x)` : število komponent vektorja x
Primer: `length(y)` $\rightarrow$ 5
- ▶ `min(x)` : najmanjša komponenta vektorja x
Primer: `min(y)` $\rightarrow$ 1
- ▶ `max(x)` : največja komponenta vektorja x
Primer: `max(y)` $\rightarrow$ 5
- ▶ `sum(x)` : vsota komponent vektorja x
Primer: `sum(x)` $\rightarrow$ 5.87
- ▶ `prod(x)` : produkt komponent vektorja x

Funkcija `norm`

- ▶ Z ukazom `norm` dobimo (evklidsko) normo vektorja.
- ▶ `norm(x)` ali `norm(x, 2)`: evklidska norma vektorja:

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

Primer: `norm(y)` $\rightarrow$ 7.4162

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

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

Primer: `norm(y, 1)` $\rightarrow$ 15

- ▶ `norm(x, inf)`: neskončna norma vektorja:

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

Primer: `norm(y, inf)` $\rightarrow$ 5

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

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

Primer: `diff(y) → [1 1 1 1]`.

- ▶ `sort(x)`: urejen vektor x .
- ▶ `unique(x)`: urejen vektor x , kjer se vsako število pojavi samo enkrat.
- ▶ `find(x)`: iskanje neničelnih elementov v vektorju x .

Vektorski funkciji `dot`, `cross`

- ▶ `dot(x, y)`: skalarni produkt vektorjev x in y .
- ▶ $x' * y$: tudi skalarni produkt vektorjev x in y (x in y sta stolpca).
- ▶ `cross(x, y)`: vektorski produkt vektorjev x in y .

Množice

- ▶ Vektorje v primeru 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 komponenta vektorja x ?

Primer: `ismember(2, x)` $\rightarrow$ 0

Primer: `ismember(2, y)` $\rightarrow$ 1

Matrike

- ▶ Matrike pišemo v **oglatih** oklepajih.
- ▶ Elemente v vrstici ločimo s **presledkom** ali **vejico**.
- ▶ Vrstice pa ločimo s **podpičjem**.

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

- ▶ j-ti stolpec matrike A: $A(:, j)$.
- ▶ i-ta vrstica matrike A: $A(i, :)$.

Matrike

- ▶ Do **elementov** matrike dostopamo z **okroglimi** oklepaji.

Primer: **element** matrike A v drugi vrstici in tretjem stolpcu:

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

Primer: **podmatrika** matrike A : $A(1:2, 2:3)$

$$\rightarrow \begin{bmatrix} 2 & 3 \\ 7 & 9 \end{bmatrix}$$

Primer: **podmatrika** matrike A : $A([1 \ 3], [1 \ 3])$

$$\rightarrow \begin{bmatrix} 1 & 3 \\ 7 & 9 \end{bmatrix}$$

Algebrske operacije z matrikami

- ▶ $+$, $-$: seštevanje, odštevanje
- ▶ $*$, $^$: množenje, potenciranje
- ▶ $'$ ali `transpose` : transponiranje

Algebrske operacije z matrikami

- ▶ .* : množenje po komponentah.
- ▶ .^ : potenciranje po komponentah.

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

- ▶ `size(A)` : dimenzije matrike A (število vrstic in število stolpcev).
- ▶ `max(A)` : vektor največjih elementov v stolpcih matrike A .
- ▶ `min(A)` : vektor najmanjših elementov v stolpcih matrike A .
- ▶ `max(max(A))` in `min(min(A))` : največji in najmanjši element matrike A .
- ▶ `sum(A, 1)` : vektor vsot elementov po stolpcih matrike A .
- ▶ `sum(A, 2)` : vektor vsot elementov po vrsticah matrike A .

Matrična funkcija `diag`

- ▶ `diag(A)` : diagonalna matrike A
- ▶ `diag(A, n)` : n -ta naddiagonalna (poddiagonalna) matrike A
Primer: `diag(A, 1)` - prva naddiagonalna matrike A
Primer: `diag(A, -1)` - prva poddiagonalna matrike A
- ▶ `A = diag(x)` : sestavimo diagonalno matriko A , ki ima na diagonalni elemente vektorja x
Primer: `A = diag([1 2 3 4 5])`
- ▶ `diag(diag(A))` : diagonalna matrika, ki ima na diagonalni diagonalne elemente matrike A

Matrične funkcije `triu`, `tril`

- ▶ `triu(A)`: zgornji trikotnik matrike A
- ▶ `triu(A, n)`: zgornji trikotnik matrike A od n -te nad diagonale naprej

Primer: `triu(A, 1)` - strogi zgornji trikotnik matrike A

- ▶ `tril(A)`: spodnji trikotnik matrike A
- ▶ `tril(A, -n)`: spodnji trikotnik matrike A od n -te pod diagonale naprej

Primer: `tril(A, -1)` - strogi spodnji trikotnik matrike A

Matrične funkcije za konstrukcijo matrik

- ▶ `ones(n)` : matrika dimenzije $n \times n$ iz samih enic
- ▶ `zeros(n)` : matrika dimenzije $n \times n$ iz samih ničel
- ▶ `eye(n)` : identična matrika dimenzije $n \times n$
- ▶ `rand(n)` : naključna matrika dimenzije $n \times n$
- ▶ `fliplr(A)` : zrcaljenje matrike A levo – desno
- ▶ `flipud(A)` : zrcaljenje matrike A gor – dol

Linearna algebra

- ▶ `det(A)` : determinanta matrike A
- ▶ `eig(A)` : lastne vrednosti in lastni vektorji matrike A
- ▶ `norm(A)` : norma matrike A
- ▶ `rank(A)` : rang matrike A
- ▶ `inv(A)` : inverz matrike A
- ▶ `trace(A)` : sled matrike A (vsota diagonalnih elementov)

Sistem linearnih enačb

► Sistem $Ax = b$ rešimo z ukazom $x = A \setminus b$.

► *Primer:*

$$A = [1 \ 2 \ 1; \ 2 \ 1 \ 2; \ 1 \ 1 \ 2], \quad b = [4 \ 5 \ 4]'$$

$$x = A \setminus b \rightarrow [1 \ 1 \ 1]'$$

Polinomi

- ▶ `poly(a)` : koeficienti polinoma, ki ima ničle zapisane v vektorju `a`.
- ▶ `poly(A)` : karakteristični polinom matrike `A`.
- ▶ `roots(a)` : ničle polinoma, ki ima koeficiente podane v vektorju `a`.
- ▶ `polyval(a, x)` : izračuna vrednost polinoma s koeficienti `a` v točki `x`.

Polinomi: ukaz `polyfit`

► `polyfit`

Z ukazom `polyfit` določimo koeficiente interpolacijskega polinoma.

Primer:

`p = polyfit(x, y, n)` nam da koeficiente polinoma stopnje n , ki gre skozi točke sestavljene iz parov vektorjev x in y .

Zanka for

- ```
▶ for pogoj
 stavki
end
```

- *Primer*:

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

# Zanka `while`

▶ `while` pogoj  
    stavki  
end

▶ *Primer:*

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

# Stavek `if`

- ▶ `if` izraz  
stavki  
`elseif` izraz  
stavki  
`else`  
stavki  
`end`

- ▶ *Primer:*

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

# Ukazi `return`, `break` in `continue`

- ▶ `return`: konča izvajanje funkcije
- ▶ `break`: konča izvajanje zanke
- ▶ `continue`: skoči na začetek zanke

# m-datoteke

- ▶ Poznamo dva tipa m-datotek: opisne in funkcijske.
- ▶ Pišemo jih v obliki tekstovnih datotek (uporabi ukaz edit).
- ▶ Opisne m-datoteke so zgolj daljše kode, ki jih ne želimo pisati v ukazno vrstico.
  - ▶ nimajo vhodnih in izhodnih podatkov
  - ▶ vračajo le rezultate vrstic, ki jih ne končamo s podpičjem

# m-datoteke

- ▶ **Funkcijske m-datoteke** se vedno začnejo z ukazom `function`.
- ▶ Funkcijam podamo **vhodne** in **izhodne** argumente.
- ▶ Spremenljivke v funkciji so **lokalne** (jih ne poznamo zunaj funkcije).
- ▶ Obstaja možnost **spremenljivega** števila vhodnih in izhodnih podatkov.

# m-datoteke

- ▶ Tako napisano funkcijo kličemo na **enak** način kot vgrajene funkcije.
- ▶ kličemo lahko **le** funkcije, ki so v **izbranem** direktoriju (mapi).
- ▶ **Ime** m-datoteke naj bo **enako** kot je **ime funkcije**.

# Primer funkcijske m-datoteke

Napišimo funkcijo, ki izračuna **koren** števila  $x$ .

```
▶ function rezultat = koren(x)
 rezultat = sqrt(x);
```

▶ **Uporaba** funkcije:

```
r=koren(3)
```

```
r = 1.7321
```

# Primer funkcijske m-datoteke

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;
```

▶ **Uporaba** funkcije:

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

## Primer funkcijske m-datoteke

- ▶ Ukaz `nargin` pove število vhodnih argumentov funkcije.

Napišimo funkcijo, ki vrne **naključno celoštevilsko** matriko dimenzije  $m \times n$ .

- ▶ 

```
function A = nakljucna(m,n)
```

```
 if nargin < 2
```

```
 n = m;
```

```
 end
```

```
 A = round(10*rand(m,n));
```

- ▶ Dva primera **uporabe** funkcije:

```
A = nakljucna(3,4)
```

```
A = nakljucna(3)
```

# Ukaz inline

- ▶ Z ukazom `inline` definiramo **notranje funkcije**.

- ▶ *Primer:* funkcija  $f(x) = x^2$ :

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

Uporaba:  $f(2) \rightarrow 4$

# Ukaz `plot`: risanje grafov funkcij

- ▶ S funkcijo `plot(x, y)` narišemo graf funkcije, ki je podana z vektorjema `x` (x koordinate) in `y` (y koordinate).
- ▶ *Primer:* narišimo funkcijo  $y = \sin x$  na intervalu  $[0, 2\pi]$ :  

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

## Ukaz `plot`: risanje grafov funkcij

- ▶ Pri funkciji `plot(y)` privzamemo, da je vektor `x` oblike `[1,2,3,...,n]`, kjer je `n` dolžina vektorja `y`.
- ▶ Pri funkciji `plot(x,y,'parametri')` lahko določimo še nekatere parametre:
  - ▶ `barva`: `'g', 'b', 'r', ...`
  - ▶ `tip črt`: `'-', '.', ...`
  - ▶ `oblika točk na grafu`: `'+', 'o', 'x', '*', ...`

# Ukaz `plot`: risanje grafov funkcij

- ▶ Rišemo lahko **več** grafov **skupaj**: `plot(x, y1, x, y2)`.
- ▶ Za risanje **vgrajenih** funkcij imamo ukaz `fplot('f', [Xmin, Xmax])`

*Primer:* `fplot('sin', [-pi, pi])`

# Grafi funkcij

- ▶ `title`: naslov
- ▶ `xlabel`: oznake na x-osi
- ▶ `ylabel`: oznake na y-osi
- ▶ `legend`: legenda
- ▶ `text(x,y,'besedilo')`: napiše besedilo na mesto (x,y).

# Grafi funkcij

## ► `grid`: mreža

Primer: `grid on` - prikaže pomožne črte

Primer: `grid off` - skrije pomožne črte

## ► `axis`: osi

Primer: `axis equal` - enaki enoti na obeh oseh

Primer: `axis [-5, 5, -3, 3]` - graf na intervalu  $[-5, 5]$  na  $x$ -osi in  $[-3, 3]$  na  $y$ -osi

# Grafi funkcij

- ▶ Če želimo, da se **nova** slika nariše na **isto** sliko, kjer smo narisali **prejšnjo** sliko: `hold on`.
- ▶ Če tega ne želimo: `hold off`.
- ▶ Če želimo narisati **več** slik, vsako zase, potem pred vsako napišemo ukaz `figure`.

# Grafi 3D funkcij

- ▶ `plot3(x, y, z)`: nariše graf 3D krivulje  $(x(t), y(t), z(t))$
- ▶ `mesh(x, y, z)`: nariše ploskev  $z = f(x, y)$
- ▶ `surf(x, y, z)`: nariše ploskev  $z = f(x, y)$
- ▶ `meshgrid`: generira mrežo.

# Grafi 3D funkcij

## Primer uporabe:

- ▶ `xx = [1 2 3 4 5 6]; yy = xx;`
- ▶ `[x, y] = meshgrid(xx,yy);`
- ▶ `mesh(x, y, x.^2 + y.^2);`

## Še nekaj ostalih funkcij:

- ▶ `A(:)`: matriko A zapiše v en sam stolpec (najprej 1. stolpec, nato 2. stolpec ...)
- ▶ `control C`: prekinemo delovanje kakšne funkcije.
- ▶ `...`: pisanje nadaljujemo v novi vrstici.
- ▶ `pause`: premor.
- ▶ `num2str`: za izpis spremenljivke v isti vrstici kot teksta.  
*Primer:* `disp(['stevilo x je enako', 'num2str(x)'])`