Tvorba funkcií v programovom prostredí MATLAB
a vysvetlenie pojmu funkcie funkcií

- *funkcie funkcií*
- *niektoré základné funkcie funkcií*
- *tvorba funkcií v programovom prostredí MATLAB*
- *grafické zobrazenie funkcií*
- *použitie funkcií na príkladoch*

⇨ Funkcie pracujúce s funkciami umožňujú prácu s matematickými funkciami namiesto číselných premenných.

⇨ Tieto funkcie zahŕňajú nasledovnú problematiku:

- numerickú integráciu
- riešenie diferenciálnych rovníc
- optimalizáciu a riešenie nelineárnych rovníc

Programové funkcie pracujúce s matematickými funkciami sú umiestnené v adresári **funfun** simulačného jazyka MATLAB.

funkcia		**popis**
fmin	→	*minimalizácia funkcie s jednou premennou*
fmins	→	*minimalizácia funkcie s niekoľkými premennými*
fplot	→	*zobrazenie priebehu funkcie*
fzero	→	*nájdenie núl funkcie s jednou premennou*
ode23	→	*riešenie DR* $R-K$ *3. rádu*
ode45	→	*riešenie DR* $R-K$ *5. rádu*
quad	→	*numerický integrál v tvare nižšieho rádu*
quad8	→	*numerický integrál v tvare vyššieho rádu*
quadl		

1. Vytváranie funkcií v programovom prostredí MATLAB

<u>*m-funkcia*</u>

⇨ umožňuje naprogramovať určité výpočty a po odladení ich používať ako súčasť iných programov

⇨ na rozdiel od skriptu (súboru) musí obsahovať slovo **function** v 1. príkazovom riadku a môže obsahovať definíciu vstupnej a výstupnej premennej

⇨ meno funkcie musí byť totožné s menom súboru, kde je funkcia uložená

⇨ syntax funkcie: **function [výstupné arg.] = meno_funkcie(vstupné arg.)**

Sú premenné, ktoré vracia funkcia a nemusia byť vždy len hodnotou.

Sú premenné, ktoré vstupujú do funkcie a musia byť pri volaní funkcie pri nej definované(priradené v zátvorkách).

- premenné skriptu sú GLOBÁLNE
- premenné funkcie sú LOKÁLNE (po použití funkcie už nie sú dostupné)

PRÍKLAD 1

Uvažujeme matematickú funkciu $f(x) = \frac{1}{(x-0,3)^2+0,01} + \frac{1}{(x-0,9)^2+0,04} - 6$. Táto funkcia môže byť použitá ako vstup pre už uvedené funkcie. Vykonáme zápis funkcie do m-súboru menom **humps.m**

```
function y=humps(x)
y=1./((x-0,3).^2+0.01)+1./((x-0.9).^2+0.04)-6;
```

x → vstupná premenná

y → výstupná premenná

- simulačný jazyk MATLAB môže reprezentovať funkcie pomocou m-súborov typu funkcia alebo priamymi objektmi ~ pomocou funkcie ***inline***
- ***inline*** je funkcia vytvorená na krátky čas, bez jej zadefinovania do m-súboru

Definícia inline funkcií je dočasná a pri novom spustení programového prostredia MATLAB je neznáma!

PRÍKLAD 2

Vytvorte v simulačnom jazyku Matlab inline funkciu. Funkciu použite z príkladu 1.

```
>> f=inline('1./((x-0.3).^2+0.01)+1./((x-0.9).^2+0.04)-6')

f =

    Inline function:

    f(x) = 1./((x-0.3).^2+0.01)+1./((x-0.9).^2+0.04)-6
```

- Následne vypočítajte hodnotu funkcie v bode 2:

```
>> f(2)

ans =

   -4.8552
```

Vytvorenie funkcie viacerých premenných:

$$inline('funkcia','arg1','arg2',\dots)$$

Reťazec znakov vyjadrujúcich funkciu

argument 1

argument 2

PRÍKLAD 3

Vytvorte inline funkciu viacerých premenných $f(x,y) = y * \sin(x) + x * \cos(y)$.

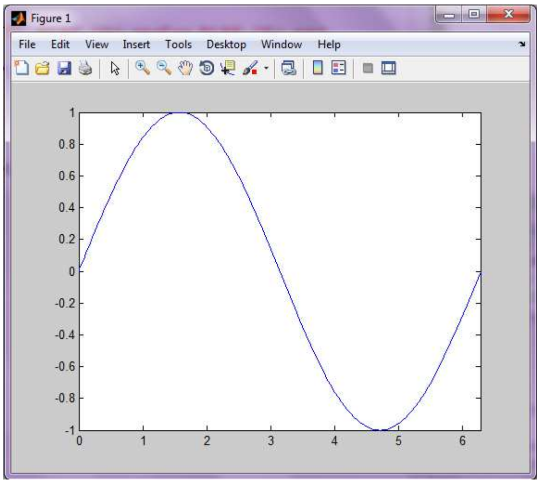
```
>> f=inline('y*sin(x)+x*cos(y)', 'x', 'y')

f =

     Inline function:

     f(x,y) = y*sin(x)+x*cos(y)
```

2. Grafické zobrazenie funkcií

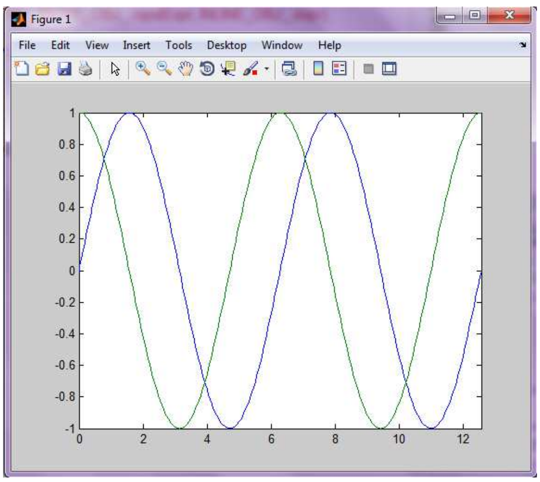
Zobrazenie matematických funkcií pomocou funkcie $fplot('fmeno', limit)$ vieme zobraziť priebeh funkcie s názvom ***fmeno*** v zadanom intervale ***limit*** (parametrom funkcie je funkcia).

PRÍKLAD 4

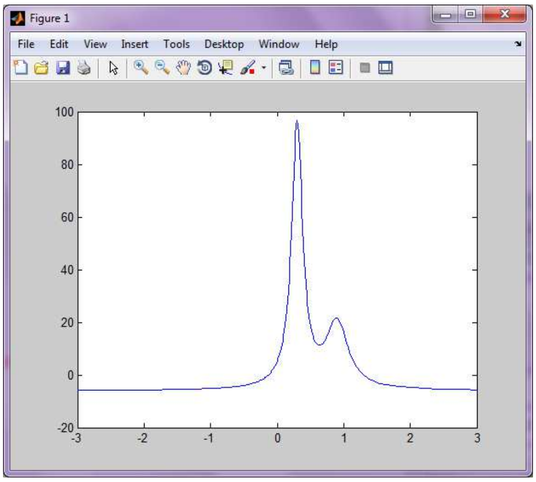
```
>> fplot('sin',[0,2*pi]);
```




```
>> fplot('[sin(x),cos(x)]',[0,4*pi]);
```

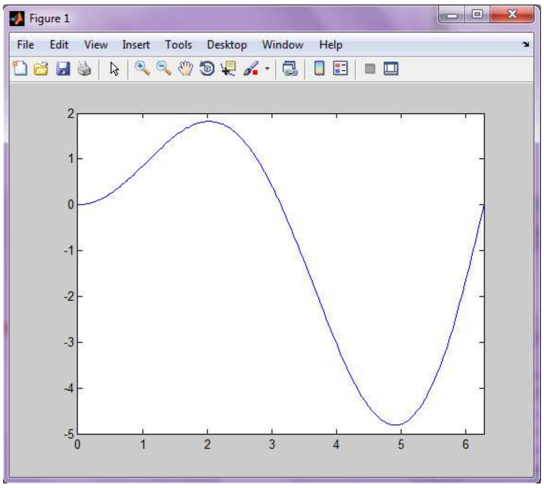


```
>> fplot('humps',[-3,3]);
```



⇨ podobne je možné použiť ako vstupné argumenty funkcie aj funkciu definovanú ako inline objekt

```
>> f=inline('x.*sin(x)','x');

>> fplot(f,[0,2*pi])
```



3. Minimum funkcie a hľadanie nulových bodov

Na minimalizáciu funkcie simulačný jazyk Matlab využíva optimalizačné prostriedky a umožňuje:

- ⇨ minimalizácia funkcie s jednou premennou;
- ⇨ minimalizáciu funkcie s viacerými premennými;
- ⇨ hľadanie nulového bodu funkcie s jednou premennou:

$$fminbnd('fun', x_1, x_2, options)$$

fun – reťazec znakov, pomocou ktorého je zapísaná funkcia alebo názov premennej, v ktorej je funkcia zapísaná pomocou príkazu inline

x_1,x_2 – začiatok a koniec intervalu na ktorom hľadáme minimum

***options* –** voľby pre hľadanie minima

PRÍKLAD 5

```
>> k=fminbnd('sin',0,2*pi)

k =

   4.7124
```

```
>> min=sin(k)

min =

  -1.0000
```

Od verzie 6.0 je možný aj zápis

```
>> k=fminbnd(@sin,0,2*pi)
```

Práca s rovnicami s jednou premennou

Na riešenie koreňov algebraických rovníc → $\boldsymbol{roots}$ (ľavá strana polynómu)

Na hľadanie numerického riešenia rovnice, ktorá nie je algebraická, používame funkciu → $\boldsymbol{fzero}$ **-** nájdenie núl funkcie(nulových bodov)

$$fzero('fun', x_0)$$

Funkcia predstavujúca ľavú stranu rovnice

Počiatočný odhad riešenia alebo interval, kde sa má nachádzať koreň

PRÍKLAD 6

Nájdite riešenie rovnice $\cos(2x) * \sin(3x)$.

```
>> x=fzero(inline('cos(2*x)*sin(3*x)'),2)

x =

   2.0944
```

```
>> x=fzero(inline('cos(2*x)*sin(3*x)'),[1.5 2.2])

x =

   2.0944
```

```
>> x=fzero('f1',2)

x =

   2.0944
```

Ak je funkcia zadefinovaná v samostatnom m-súbore s názvom f1

PRÍKLAD 7

Riešte nasledujúce rovnice:

1. $x * e^x = 1$
2. $x^2 + x + 1 = 0$
3. $\sin(x) = \frac{x}{10}$

Upraviť na tvar f(x)=0

Problém: minimum funkcie viacerých premenných, koreň funkcie, riešenie sústavy diferenciálnych rovníc, numerická integrácia -> často vyskytujúce sa problémy pri riešení inžinierskych úloh

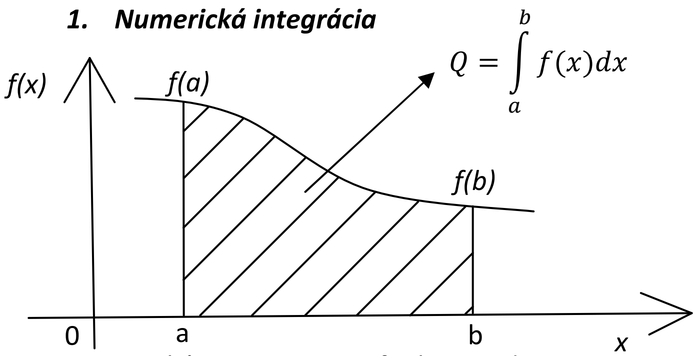
Simulačný jazyk Matlab podporuje riešenie týchto problémov:

a) nájdenie nulovej hodnoty funkcie jednej premennej,
b) nájdenie minima funkcie jednej alebo premenných,
c) nájdenie hodnoty určitého integrálu jednej premennej,
d) riešenie sústavy diferenciálnych rovníc (LDR,NDR)

Podmienkou pre používanie štandardných funkcií simulačného jazyka Matlab je znalosť tvorby m-funkcií. Štandardné funkcie majú ako prvý parameter meno novovytvorenej funkcie, ďalšie parametre sú dané syntaxou.

- Funkcie nepracujú s číselnými poľami, ale s funkciami programovacieho prostredia MATLAB. Adresár funfun (obsahuje programové funkcie vykonávané nad matematickými fukciami ako napríklad hľadanie bodov funkcie, vykresľovanie, funkcie pre integráciu ...).

1. Numerická integrácia



Na numerické integrovanie funkcie jednej premennej v tvare $\int_a^b f(x)dx$ ponúka MATLAB dve štandardné funkcie

- quad -> adaptívne Simpsonovo pravidlo
  Pre výpočet Simpsonovým pravidlom rozdelíme interval <a,b> na párny počet podintervalov a v každom intervale vykonávame náhradu pôvodnej funkcie parabolou. Vzorec pre výpočet Simpsonového pravidla je: $\int_a^b f(x)dx \approx \frac{h}{3}\big(f(x_0) + f(x_{2n}) + 4(f(x_1) + f(x_3) + \ldots + f(x_{2n-1})) + 2(f(x_2) + f(x_4) + \cdots + f(x_{2n-2}))\big)$.

Vzorec pre odhad chyby vyzerá: $\frac{max_{<a,b>}|f^{(4)}(x)|}{180n^4}(b-a)^5$

- quadl -> adaptívne Lobattovo pravidlo

  pre výpočet integrál pomocou Lobattovho pravidla použijeme vzorec: $\int_a^b f(x)dx \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}z_i + \frac{b+a}{2}\right)$, kde $w_i = \frac{2}{(1-x_i{}^2)[P'_n(x_i)]^2}$, P_n je Lagrangeov polynóm.

<u>Základná syntax funkcií quad (quadl):</u>

$$hodnota_{int} = quad('fname', dol_{hr}, horna_{hr});$$

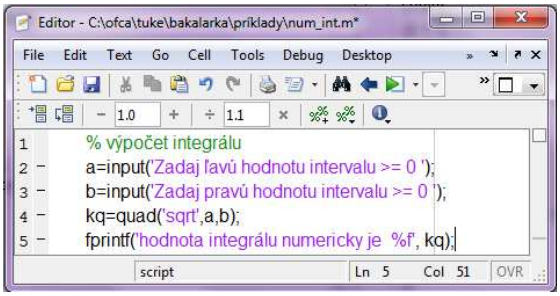
fname – meno užívateľom vytvorenej funkcie, môže obsahovať aj meno funkcie programového prostredia MATLAB, tzv. *built-in-funkcie* takými to funkciami sú napríklad: sin, sqrt ...

dol_hr, horna_hr – hranice intervalu, na ktorých je funkcia integrovaná

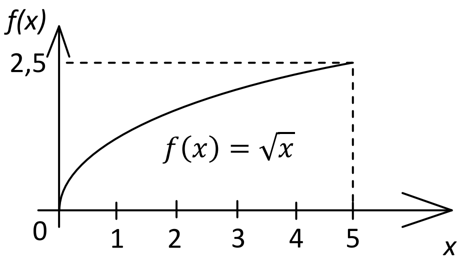
<u>PRÍKLAD 8</u>

Vypočítajte v simulačnom jazyku MATLAB integrál $\int_0^5 \sqrt{x}\, dx; f(x) = \sqrt{x}$.

<u>Riešenie v simulačnom jazyku Matlab:</u>



Zadaj ľavú hodnotu intervalu >= 0 0
Zadaj pravú hodnotu intervalu >= 0 5
hodnota integrálu numericky je 7.453556



Ďalšie voliteľné požiadavky pre riešenie integrálu:

$$Q = quad('fun', a, b, tol, trace)$$

tol - presnosť integrálu ($1e^{-6}$)

trace – nenulový výpis výpočtovej rekurzie

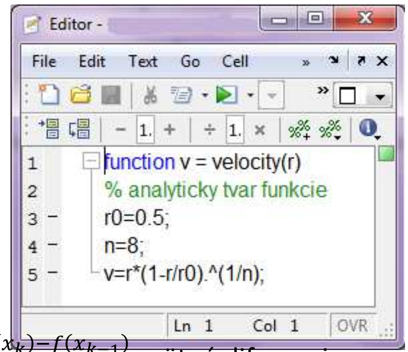
<u>PRÍKLAD 9</u>

Vypočítajte integrál, ak analytická funkcia sa definuje užívateľom a nepatrí medzi štandardné vybavenie simulačného jazyka Matlab.

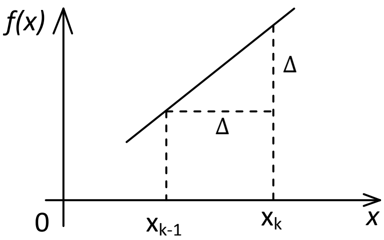
$$v_{av} = \frac{\int_0^{r_0} v(r) 2\pi dr}{\pi r_0^2} = \frac{2 v_{max}}{r_0^2} \int_0^{r} r \left(1 - \frac{r}{r_0}\right)^{\frac{1}{n}} dr$$

$$v(r) = v_{max} \left(1 - \frac{r}{r_0}\right)^{\frac{1}{n}}$$

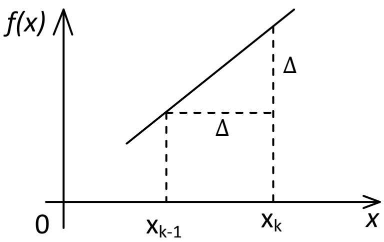
- ➔ Vytvorte m-file, ktorý zadefinuje analytický tvar funkcie, z ktorej integrál treba vypočítať:
- ➔ **<u>Riešenie v simulačnom jazyku Matlab:</u>**



2. Numerická derivácia



$f´(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$ spätná diferencia



$f´_{KM}(x_k) = \frac{f(x_{KM}) - f(x_K)}{x_{KM} - x_K}$ dopredná diferencia

*Funkcia **<u>diff:</u>***

- ⇨ vypočíta diferencie medzi hodnotami vo vektore
- ⇨ vytvára nový vektor

<u>PRÍKLAD 10</u>

Predpokladajme, že funkcia f(x) má tvar polynómu: $f(x) = x^5 - 3x^4 - 11x^3 + 27x^2 + 10x - 24$

Vypočítajte deriváciu tejto funkcie na intervale [-4,5].

<u>Riešenie v simulačnom jazyku Matlab:</u>

```
>> x=-4:0.1:5;%generovanie hodnôt nezávislej premennej
>> f=x.^5-3*x.^4-11*x.^3+27*x.^2+10*x-24;
>> df=diff(f)./diff(x);
```

Príklad na cvičenie:

Vypočítajte integrál $\int_0^{3\pi} \sqrt{4\cos(2t)^2 + \sin(t)^2 + 1}\, dt$ pomocou vytvorenia funkcie ako m-file a ako objekt.

3. Riešenie diferenciálnych rovníc

- Fyzikálne systémy: elektrické, mechanické, tepelné, hydraulické sú vo všeobecnosti popísané systémom DR.
    - lineárne dynamické systémy sú popísané LDR s konštantnými koeficientmi.
- Analytické riešenie LTI (Linear Time Invariant)systému je vykonávané Laplaceovou transformáciou a spätnou Laplaceovou transformáciou (L^{-1})

⇨ Lineárne a časovo premenné systémy (LTV) a nelineárne systémy

Diferenciálne rovnice

⇨ DR riešime numerickými metódami. Funkciu (jej hodnotu), ktorú dostávame po každom kroku pri numerickom výpočte je aproximácia analytického riešenia. Takúto funkciu nevieme analyticky vyjadriť.

Riešenie v simulačnom jazyku Matlab

⇨ Je nutné si uvedomiť, že simulačný jazyk Matlab neumožňuje riešiť DR vyššieho rádu, ale iba systém DR 1.rádu

⇨ Preto využívame transformáciu DR n-tého rádu na DR 1.rádu:

$$\frac{dy(t)}{dt} = f(t,y), \ \ PP: y(0) = y_0$$

*Funkcia **ODE*** – rieši systém DR 1.rádu

f - lineárna alebo nelineárna funkcia

Syntax funkcie na riešenie systému obyčajných DR:

[t,y]=solver(‘odefun’, cas_int, poc_pod)

solver – riešiteľ , napríklad *ODE45, ODE23* ...

‘odefun’ – meno m-filu funkcie , ktorá obsahuje analytický popis systém DR

$dy = odefun(t,y)$

dy,y – stĺpcové vektory

t - skalár

výstupný stĺpcový vektor dy

cas_int – [t0,tF] riadkový vektor – integračný interval

poc_pod – stĺpcový vektor počiatočných podmienok

<u>*Výstup z funkcie ODE:*</u>

t – stĺpcový vektor časových bodov

y – matica riešenia, kde v každom riadku je riešenie v čase, ktoré je uvedené v príslušnom riadku stĺpcového vektora t.

<u>Ukážka tvorby substitučného kanického tvaru pre LDR 3 rádu:</u>

Prepis DR vyššieho rádu na systém DR 1.rádu (napr. 3 rád)

$$\boxed{y'''(t) + a_2 y''(t) + a_1 y'(t) + a_0 y(t) = b_0 * u(t)}$$

<u>Kanonicky tvar</u>

označme $\boxed{y(t) = x_1(t)}$ - stavová premenná

Prepis do substitučného kanonického tvaru: $y(t) = x_1(t)$

$$y'(t) = x_1'(t) = x_2(t) \qquad (1)$$

$$y''(t) = x_2'(t) = x_3(t) \qquad (2)$$

$$y'''(t) = x_3'(t) = b_0 u(t) - a_0 x_1 - a_1 x_2 - a_2 x_3 \qquad (3)$$

Záver: DR 3rádu -> 3DRovnice 1rádu (1)÷(3)

<u>*PRÍKLAD 11*</u>

Uvažujeme diferenciálnu rovnicu 2. rádu (nelineárna diferenciálna rovnica) reprezentujúcu Van-der-Polov oscilátor. Namodelujte riešenie tejto DR v simulačnom jazyku Matlab.

$$y''(t) = g(t, y, y') = y'(1 - y^2) - y$$

$$y''(t) - y'(t) * \left(1 - y^2(t)\right) + y(t) = 0$$

Substitúcia: $y(t) = x_1(t)$

Vytvorenie substitučného kanonického tvaru:

dve diferenciálne rovnice 1.rádu

$y'(t) = x_1'(t) = x_2(t)$

$y''(t) = x_2' = x_2(t) * \left(1 - x^2(t)\right) - x_1(t)$

vander.m:

```
% Matematicky zápis DR2.rádu prepísaný
%do stavového priestoru
function xder = vander(t,x)
xder=[x(2);x(2).*(1-x(1).^2)-x(1)+0.5];
return
```

Pozn. $xder = [x(2); x(2).* (1 - x(1).\hat{}2) - x(1)]$ - maticový zápis

difrov2r.m :

```
% program na riešenie nelineárnej diferenciálnej rovnice
% 2.rádu -  Van - Der - Pol
% oscilátor - zadaný v stavovom priestore
x0=[0 0.25]'; % inicializácia počiatočných podmienok
t0=0; tf=10;  % definícia časového intervalu
[t,x]=ode23('vander',[t0,tf],[0;0.25]);
[t,x]=ode45('vander',[t0,tf],[0;0.25]);
subplot(211); plot(t,x(:,1)); ...
title('riešenie y(t)'); xlabel('t'); grid; ...
subplot(212); plot(t,x(:,2)); ...
title('prvá derivácia y(t)'); xlabel('t'); grid; ...
plot(t,x); % dva grafy v jednom obrázku
title('Van-der-Pol rovnica - časová história ');
pause
```

Hlavný program :

```
plot(t,x(:,1),'k-',t,x(:,2),'k--')
```

viacparametrový plot -> $y(t), y'(t)$ v jednom grafe