Základy práce s textovými reťazcami

Doteraz sme v MATLABe pracovali s datovými typmi: reálne číslo, vektor, matica.
Veľmi dôležitým a často používaným dátovým typom je textový reťazec. Ako si môžeme predstaviť textový reťazec? Predstavte si, že do premennej potrebujete uložiť názov ovocia, ktoré je nevyhnutné na prípravu jablkového koláča. V MATLABe je to možné vykonať nasledovne:

```
ovocie = 'jablko'
```

ovocie =

jablko

Toto je v podstate najjednoduchší spôsob akým je možné vytvoriť textový reťazec. Skôr ako si ukážeme komplexnejšie spôsoby tvorby reťazcov, vysvetlíme si niektoré funkcie na prácu s reťazcami.

Porovnanie dvoch reťazcov.

Na porovnanie reťazcov v MATLABe slúži funkcia **strcmp**.

```
ovocie1 = 'jablko'
ovocie2 = 'melon'
strcmp(ovocie1,ovocie2)
```

ans =

0

V prípade, že reťazce sú rovnaké funkcia vráti hodnotu 1, v prípade že sa reťazce líšia funkcia vráti hodnotu 0.

Od teraz si budeme pamätať !!!

Na porovnanie dvoch textových reťazcov slúži funkcia **strcmp**

Príklad použitia:

```
meno1='Peter';
meno2='Udo';
vysledok=strcmp(meno1,meno2)
```

<u>Príklad 1</u>

V kapitole, v ktorej sme sa venovali riadeniu toku programu sme pomocou cyklu vytvorili program, ktorý opakoval cyklus do tej chvíle kým užívateľ nezadal správne heslo. No v danom príklade sme mohli použiť len numerické heslo. Teraz prerobíme príklad tak, aby akceptoval aj textové reťazce. Pýtate sa PREČO? Lebo sa učíme porovnávať reťazce!

```
textoveheslo=input('Zadaj heslo','s');
while (strcmp(textoveheslo,'tajneheslo')==0)
        disp('Zadal si zle HESLO skus znova');
        textoveheslo=input('Zadaj heslo','s');
end
disp('Tvoje heslo je spravne');
```

Poznámka k príkladu: Možno Vás trochu zarazilo použitie funkcie input, kde ako druhý parameter funkcie bol použitý operátor 's'. Tento parameter informuje funkciu aby akceptovala textový reťazec. Ako si to predstaviť? Pár riadkov vyššie je uvedený spôsob, akým tvoriť textové reťazce. To znamená, že ak chceme do premennej meno uložiť krsné meno máme dve možnosti:

I. možnosť

>> meno=input('Zadaj krsne meno:')

Zadaj krsne meno: 'Udo'

meno =
Udo

Týmto ale nútime užívateľa aby poznal syntax MATLABu a svoje krsné meno zadal v úvodzovkách.

II. možnosť

>> meno=input('Zadaj krsne meno:','s')

Zadaj krsne meno: Udo

meno =

Udo

Je oveľa elegantnejšia, pretože MATLAB dopredu očakáva textový reťazec a užívateľ nemusí zadať svoje meno v úvodzovkách.

MATLAB má ešte niekoľko funkcií na porovnanie dvoch reťazcov:

strcmpi – porovná reťazce *r1* a *r2* a nerozlišuje veľké a malé písmena; strncmpi(*r1*,*r2*)
strncmp – porovná prvých *n*-znakov reťazcov *r1* a *r2;* strncmp(*r1*,*r2*,*n*)
strncmpi – porovná prvých *n*-znakov reťazcov *r1* a *r2* a nerozlišuje veľké a malá písmena; strncmpi(*r1*,*r2*,*n*)

Konverzia číselnej premennej na textovú

Často sa pri programovaní stáva, že potrebujeme z určitého čísla vytvoriť reťazec, prípadne naopak z textového reťazca vytvoriť číslo.

Na konverziu čísla na reťazec slúži funkcia **num2str**.

Príklad použitia:

```
A=25.4;
retazec=num2str(A)
```

Naopak konverziu reťazca na číslo je možné vykonať pomocou funkcie **str2num.**

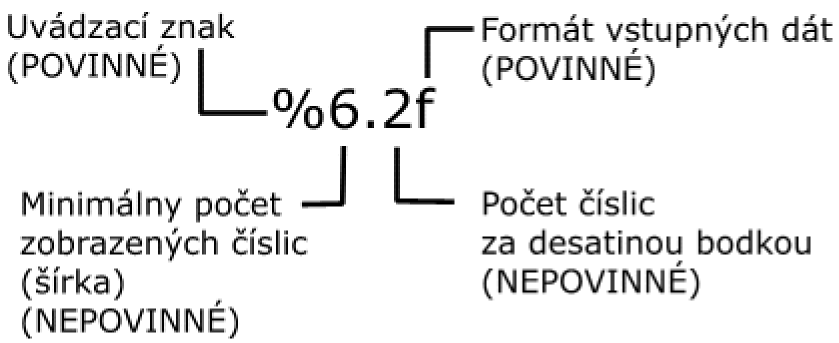
Formátovaný zápis do reťazca

Obrovský komfort a univerzálnosť pri práci s reťazcami poskytuje funkcia **sprintf**. Slúži na formátovaný zápis dát do reťazca. Jej všeobecný prototyp je nasledovný:

retazec = sprintf('format',data...)

Spôsob akým sú dáta formátované pri vytváraní reťazca sa špecifikuje pomocou formátovacieho reťazca format. Tento formátovací reťazec musí vždy začínať uvádzacím znakom %. Ďalej je nutné zadať typ dát, ktoré budeme formátovať, prípadne presnosť (počet číslic), ktorú chceme zobraziť.

Schematický znázornený formátovací reťazec je na nasledujúcom obrázku:



Formát vstupných dát je jedno písmeno charakterizujúce typ dát, ktoré sa zapíšu do textového reťazca. Význam niektorých písmen je uvedený v nasledujúcej tabuľke:

symbol	význam
f	zánam reálneho čísla
d	záznam celého čísla
e	záznam exponenciálneho čísla
s	záznam reťazca

Príkaz sprintf('\n') ukončí riadok.

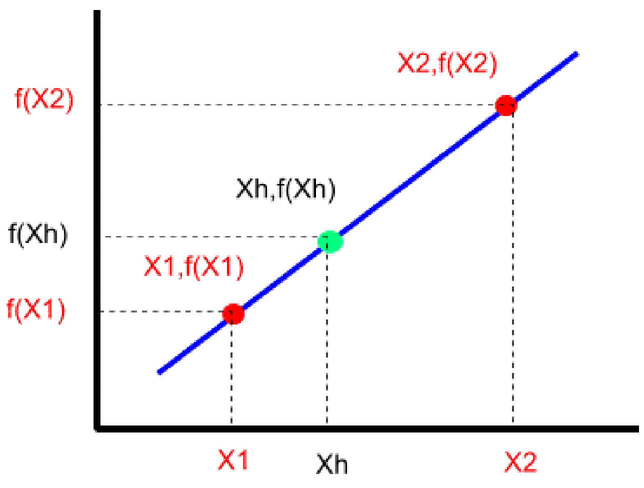
Niekoľko príkladov vysvetľujúcich použitie funkcie sprintf:

```
>> A=10.1
A =
10.1000

>> B=5.4
B =
5.4000
```

```
>> s=sprintf('%f',A)
s =
10.100000
>> s=sprintf('%1.2f',A)
s =
10.10

>> s=sprintf('%1.25f',A)
s =
10.1000000000000000000000000

>> s=sprintf('Avogardova konstanta je %1.6e',6.02214e23)
s =
Avogardova konstanta je 6.022140e+023

>> s=sprintf('Sucet cisla A=%1.2f s cislom B=%1.5f je cislo %f',A,B,A+B)
s =
Sucet cisla A=10.10 s cislom B=5.40000 je cislo 15.500000

>> krsne='Arnold';
>> priezvisko='Schwarzneger';
>> vek=54;
>> info=sprintf('Herec %s %s ma %d rokov.',krsne,priezvisko,vek)
info =

Herec Arnold Schwarzneger ma 54 rokov.
```

Numerické okienko I: Lineárna interpolácia

V inžinierskej praxi sa veľmi často stretnete s pojmom interpolácia. Interpoláciu používame v prípade, keď potrebujeme poznať hodnotu v mieste, kde sme ju nemerali. Preto sa ju snažíme odhadnúť na základe hodnôt v jej bezprostrednom okolí. Geometrický význam lineárnej interpolácie je znázornený na nasledujúcom obrázku:



$$f(x_h) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x_h - x_1) \qquad (I)$$

Príklad 1

Tepelná kapacita vodných roztokov glycerínu v závislosti od teploty je uvedená v nasledujúcej tabuľke:

Teplota [°C]	0	20	40	60	80	100
Tepelná kapacita [$kJkg^{-1}K^{-1}$]	2,261	2,368	2,475	2,583	2,690	2,797

Predstavte si situáciu, že potrebujete určiť (odhadnúť) tepelnú kapacitu pri 65 °C.

Dosadením do vzorca (I) dostávame

$$Cp(65) = Cp(60) + \frac{Cp(80) - Cp(60)}{80-60}(65-60) =$$
$$= 2.583 + \frac{2.690 - 2.583}{80-60}(65-60) = 2.6098 kJ.kg^{-1}.K^{-1}$$

V MATLABe je možné vykonať jednorozmernú lineárnu interpoláciu príkazom **interp1**.

Prototyp funkcie:

Fhladane=interp1(hodnotyX, hodnotyY, X)

V našom prípade by bolo možné vykonať danú interpoláciu nasledovne:

```
X=[0 20 40 60 80 100]
Y=[2.261 2.368 2.475 2.583 2.690 2.797]
CP_65=interp1(X,Y,65)
```

CP_65 =

2.6098

Príklad 2

Závislosť hustoty roztoku KOH vo vode od zloženia roztoku v hmot.% pri teplote 20 °C vyjadruje nasledujúca tabuľka:

ρ [$kg.m^{-3}$]	1090	1288	1396	1510
w(KOH)	0.1	0.3	0.4	0.5

Ako na potvoru, v labáku pracujete s roztokom KOH so zložením 15 mol.% a potrebujete vedieť jeho molárnu koncentráciu v mol.dm^{-3}. A čo teraz?

Riešenie

Označíme si A=KOH, B=voda a ideme odvádzať:

$$c_A = \frac{n_A}{V} = \frac{\frac{m_A}{M_A}}{V} = \frac{m_A}{M_A V} = \frac{w_A m}{M_A V} = \frac{w_A \rho}{M_A} \quad \text{(I)}$$

Dostali sme vzťah na výpočet molárnej koncentrácie pomocou hustoty roztoku. Nebol by žiaden problém, ak by sa hustota ρ so zložením nemenila, t.j. bola by konštantná. Za *wA* by sme dosadili nasledujúci vzťah:

$$w_A = \frac{x_A M_A}{x_A M_A + (1 - x_A) M_B} \quad \text{(II)}$$

A pre zadanie x_A=0,15 by sme tak mohli určiť c_A.

Avšak hodnota ρ sa so zložením mení (viď tabuľka), teda nevieme aká bude hustota roztoku so zložením 15 mol.%. Preto najprv zostrojíme graf závislosti hustoty roztoku od zloženia v mol.%, potom odčítame hodnotu hustoty pre xA=0,15, dosadíme do vzťahu (I) a vypočítame cA .

Zostrojenie grafu v MATLABe:

```
clc
wA=[0.10 0.30 0.40 0.50];
ro=[1090 1288 1396 1510]; % [kg/m3]
MA=56.11; % [g/mol]
MB=18.015; % [g/mol]
for i=1:4
        xA(i)=(wA(i)/MA)/(wA(i)/MA+(1-wA(i))/MB);
end

% alebo mozeme pouzit bodkovy sucin
% xA=(wA/MA)./(wA/MA+(1-wA)./MB);

plot(xA,ro);
xlabel('mol. zlomok');
ylabel('hustota [kg/m3]');
```

Pre hodnotu x_A=0,15 odčítaním z grafu vychádza hustota roztoku cca ρ =1345 kg.m^{-3}. Nakoľko už vieme vykonať lineárnu interpoláciu, môžeme do príkladu pridať riadok, ktorý nám určí hodnotu hustoty roztoku:

RO_preXA_15=interp1(xA,ro,0.15)

Dostávame hodnotu veľmi podobnú s hodnotou odčítanou z grafu 1.3447e+003 kg.m^{-3}.
Dosadením do rovnice, ktorá vznikla spojením rovníc (I) a (II), doplníme program:

```
xA_zadane=0.15;
cA=xA_zadane*MA/(xA_zadane*MA+(1-xA_zadane)*MB)*RO_preXA_15/MA;
```

Mól. koncentrácia daného roztoku je c_A = 8,5001 mol.dm^{-3}.

Numerické okienko II: numerický výpočet prvej derivácie

Na tomto mieste sa nebudeme podrobne venovať teórii, akým spôsobom boli odvodené vzťahy na výpočet prvej derivácie. Ukážeme si iba najjednoduchší vzťah na výpočet prvej derivácie spojitej funkcie.

Najjednoduchšia aproximácia prvej derivácie je tzv. dvojbodová formula:

$$\left.\frac{df}{dx}\right|_{x_0} \doteq \frac{f(x_0+\Delta)-f(x_0)}{\Delta}$$

Δ je čo možno najmenšie číslo. Pre zaujímavosť: pri tejto formule „vyrábame" chybu rádovo 0,5. Δ (preto je vhodné túto hodnotu voliť čo najmenšiu).

Na výpočet prvej derivácie v bode *x0* nám teda stačí vypočítať funkčnú hodnotu v bode *x0* a funkčnú hodnotu v tesnej blízkosti bodu *x0* (teda v bode *x0*+ Δ).

Ukážka

Nech máme funkciu:

$$y = 5x^2 - 4x + 2$$

a chceme vypočítať numerickú deriváciu funkcie v bode x=3:

$$\left.\frac{dy}{dx}\right|_{x=3} \doteq \frac{y(x_0+\Delta)-y(x_0)}{\Delta} = \frac{y(3+\Delta)-y(3)}{\Delta}$$

Nech hodnota Δ je číslo 1e-6, potom:

$$\left.\frac{dy}{dx}\right|_{x=3} \doteq \frac{y(3+1e^{-6})-y(3)}{1e^{-6}} = \frac{5(3+1e^{-6})^2-4(3+1e^{-6})+2-(5.(3)-4.(3)+2)}{1e^{-6}}$$

$$\left.\frac{dy}{dx}\right|_{x=3} \doteq \mathbf{26.00000500763144}$$

Skutočnú hodnotu vieme získať, nakoľko danú funkciu je veľmi jednoduché analytický zderivovať:

$$\frac{dy}{dx} = 10.x - 4$$

po dosadení za x=3 dostávame presnú hodnotu:

$$\left.\frac{dy}{dx}\right|_{x=3} = 26$$

Ako môžete vidieť, nami vypočítaná numerická hodnota sa len trochu líši od skutočnej hodnoty.

Príklad 1

Zostrojte graf funkcie y=x^4-15x^2+10x+24 a graf prvej derivácie funkcie *y*.

Dve možnosti:

Použitie cyklu:

```
X=linspace(-5,5,100)
d=1e-6;
for i=1:100
    Y(i)=X(i)^4-15*X(i)^2+10*X(i)+24;
    FX0plusD(i)=(X(i)+d)^4-15*(X(i)+d)^2+10*(X(i)+d)+24;
    FX0(i)= X(i)^4-15*X(i)^2+10*X(i)+24;
    DER=(FX0plusD-FX0)/d;
end
plot(X,Y,X,DER)
legend('Funkcia Y','Prva derivacia funkcie')
```

Použitie bodkového súčinu:

```
X=linspace(-5,5,100)
Y=X.^4-15*X.^2+10*X+24;
d=1e-6;
FX0plusD=(X+d).^4-15*(X+d).^2+10*(X+d)+24;
FX0=X.^4-15*X.^2+10*X+24;
DER=(FX0plusD-FX0)/d;
plot(X,Y,X,DER)
legend('Funkcia Y','Prva derivacia funkcie')
```

Výsledok bude samozrejme totožný.

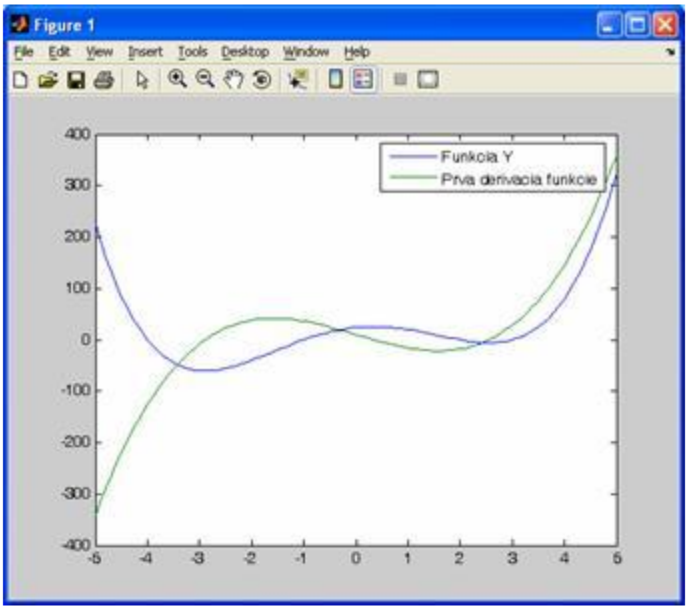


Figure 1
File Edit View Insert Tools Desktop Window Help
Funkcia Y
Prva derivacia funkcie
400
300
200
100
0
-100
-200
-300
-400
-5
-4
-3
-2
-1
0
1
2
3
4
5