Lineární algebra s Matlabem – cvičení 1

Skripty

1. Neakceptují vstupní a nevracejí výstupní argumenty.
2. Operují na datech v pracovním prostoru. Všechny proměnné definované ve skriptu jsou do tohoto prostoru uloženy a zůstávají v něm i po jeho ukončení.
3. Vhodné pro spuštění dávky příkazů, které potřebujeme spouštět víckrát.
4. Skript je složen z řádků s příkazy a komentáři.
5. Volání skriptu pomocí jména souboru (bez „.m“).

Příklad 1.

1. Vytvořte skript „goniom_script.m“ s následujícím obsahem:

```
% Priklad skriptu na vypocet a vypis hodnot
% goniometrickych funkci v bode x=pi/2
x=pi/2;
s=sin(x); c=cos(x);
disp([s, c]); % Vypise hodnoty goniom. fci. na obrazovku
```

2. Přidejte do skriptu výpočet a výpis funkcí tg a cotg (použijte `help elfun` k nalezení příslušných goniometrických funkcí).

Příklad 2.

1. Stáhněte si do Current folder skript http://homel.vsb.cz/~mer126/LAM/statistics.m
2. Skript počítá a vypisuje základní statistiky zadaného vektoru. Doplňte skript o výpočet a výpis následujících hodnot: rozptyl, minimum, maximum, modus. Použijte metody `var`, `min`, `max`, `mode`.

Příklad 3.

1. Vytvořte skript „magic_matrix.m“ který vytvoří „magickou matici“ řádu 5 a uloží do příslušných proměnných a vypíše:
    a. Prvek na pozici [2, 3].
    b. První řádek.
    c. Poslední sloupec.
    d. Hlavní diagonálu.
    e. Submatici obsahující pouze první až třetí řádek a druhý a třetí sloupec dané matice.

Příklad 4.

1. Vytvořte skript string.m s následujícím obsahem:

```
% Priklad prace s textovym retezcem.
str1 = 'Hello world! ';
str2 = 'This is Matlab.';
```

2. Spojte řetězce `str1` a `str2` do jednoho řetězce `str` a vypište jej.
3. Vypište délku řetězce `str`.
4. Pomocí vhodné funkce nahraďte v řetězci `str` slovo „world“ za „FEI“.
5. Pomocí vhodné funkce rozdělte řetězec opět na dva. Jako oddělovač použijte symbol „!“.

Funkce

1. Akceptují vstupní argumenty a vracejí výstupní argumenty
2. Operují na datech v lokálním pracovním prostoru, vnitřní proměnné jsou defaultně lokální, po ukončení funkce se zruší.
3. Vhodné pro rozšíření možností MATLABu, resp. pro tvorbu vlastních projektů.
4. Funkce se skládá z těchto řádků:
    a. Řádek s definicí funkce uvozený klíčovým slovem function, pr.
    `function[r, s, t]=jmeno_funkce(x,y)`
    kde x, y jsou vstupní argumenty a r,s,t výstupní argumenty. Jméno funkce se doporučuje volit shodně s názvem souboru.
    b. H1 řádek (hned po definici funkce). Jednořádkový co nejvýstižnější popis funkce začínající znakem `%`. Tento řádek se zobrazí při volání funkce `lookfor` a vypíše se na obrazovku při volání příkazu `help` se jménem adresáře.
    c. Help text – řádky, které následují hned po H1 řádku a začínají symbolem `%`. Tento text se vypíše na obrazovku i s H1 řádkem při volání příkazu `help` se jménem funkce.
    d. `% See also` – následuje za Help textem. Všechny Matlabovské funkce uvozené těmito klíčovými slovy se vypíší zvýrazněně a je možno kliknutím myši přejít na help text těchto funkcí.
    e. Řádky s příkazy a komentáři, kontrola vstupních a výstupních argumentů, vlastní tělo funkce, možnost definování podfunkcí.
    f. V případě vnořených (nested) funkcí je nutno primární funkci i všechny do ní vnořené funkce ukončit klíčovým slovem `end`. Obecně není nutné funkci takto ukončovat.
5. Volání funkce pomocí jména souboru, který by měl být shodný se jménem funkce.

Podfunkce
Uvádějí se za definicí vlastní funkce. Začínají opět klíčovým slovem `function` a následují řádky s příkazy a komentáři. Podfunkce ovšem není viditelná vně funkce, v níž je definována.

Jména funkcí
- Rozlišuje se 31 znaků v závislosti na systému.
- Musí začínat písmenem a pokračuje dalšími písmeny, čísly nebo znakem podtržítko.
- Jméno funkce a jméno souboru se může lišit. Doporučuje se ovšem, aby byly stejné.

Rozlišení jména proměnné a funkce
Při kontrole se postupně ptáme, zda se jedná o:
1. proměnnou,
2. podfunkci,
3. privátní funkci,
4. funkci existující v nastavených cestách.

Vezme se vždy první výskyt z důvodu jednoznačnosti.

Příklad 1.

1. Vytvořte funkci sincar.m s následujícím obsahem:
```
function y=sincar(x)
% SINCAR Sinus cardinalis
% Uziti y=sincar(x)
% See also sin

y = sin(x) / x;
```
2. Upravte ji tak, aby vypisovala text "Hodnota funkce sinus cardinalis v bode *x* je *s*."

Příklad 2.

1. Vytvořte funkci „goniom_fce.m" vycházející ze skriptu „goniom_script.m" a upravte ji tak, aby x byl vstupní argument a hodnoty goniometrických funkcí byly výstupními argumenty.
2. Nezapomeňte na správnou dokumentaci (viz body b. – d.)
3. Kontrolu správnosti vstupních argumentů doplníme, až budeme znát podmínkové bloky.

Příklad 3.

1. Vytvořte funkci „compute_statistics.m" vycházející ze skriptu „statistics.m". Upravte ji tak, aby vektor x byl vstupním argumentem a základní statistiky byly výstupními argumenty.
2. Nezapomeňte na správnou dokumentaci (viz body b. – d.)
3. Za tělem funkce compute_statistics vytvořte podfunkci mean, která vrátí upravenou střední hodnotu:
```
% Podfunkce MEAN (upraveny vypocet stredni hodnoty)
function f=mean(x)
% Vypocet stredni hodnoty
f=sum(x)/(length(x) + 1);
```
Která verze funkce `mean` se volá v těle funkce `compute_statistics`?
4. Kontrolu správnosti vstupních argumentů doplníme, až budeme znát podmínkové bloky.

Příklad 4.

1. Vytvořte funkci matrix_info.m, která bude mít na vstupu obecnou obdélníkovou matici. Výstupem budou:
    a. Vektor obsahující rozměry matice.
    b. Vektor obsahující součet všech prvků v matici, maximální prvek v matici a průměr všech prvků v matici.
    c. Vektor hlavní diagonály matice.
    d. První řádek matice.
    e. Transponovanou vstupní matici.
2. Výsledky bodu b. vypište na obrazovku formátovaným výstupem.