

Základy programování v MATLABu

(Pomocný text verze 0.2 část 1 — 23.2.2004 — Michal Hajžman)

Tento materiál obsahuje stručné a subjektivně vybrané informace potřebné pro základní programování v MATLABu. Měl by sloužit jako přehled při cvičeních a také jako pomůcka při samostatném studiu :-). U každého tématu je vždy uveden příkaz pro *help*, kde lze získat podrobnější informace. Pro další studium doporučuji elektronickou dokumentaci (*Getting Started with MATLAB* a *Using MATLAB*).

V přehledu jsou ve většině případů uváděny příkazy bez parametrů, přestože vstupní či výstupní parametry existují. Jestliže není uveden dostatečně názorný příklad pro použití příkazu, lze pomocí příkazu

```
help <jmeno_prikazu>
```

získat okamžitou podrobnou nápovědu.

1 Obecné problémy

help	Vypisuje základní nápovědu.
quit	Ukončení programu.
who, whos	Informace o proměnných přítomných v paměti.
clear	Vyčistí paměť, popřípadě vybrané objekty.
dir, ls	Výpis obsahu adresáře.
cd	Změna aktivního adresáře.
pwd	Vypíše jméno aktuálního adresáře.
type	Výpis souboru na obrazovku.
edit	Editace souboru v matlabovském editoru. Příkaz edit <jmeno> otevírá M-soubor jmeno.m
format	Nastavuje formát výstupu na obrazovku. (Ale všechny výpočty jsou dále prováděny v dvojité přesnosti.) Lze nastavit například zkrácený výpis (format short), dlouhý výpis (format long) či krátký výpis s mantisou (format short e) atd.
disp	Vypisuje proměnnou (pole) na obrazovku (disp(x)).
;	Píše se za výraz či příkaz a potlačuje výstup na obrazovku.

Další informace, včetně jiných obecných příkazů, lze získat například příkazem **help general**.

2 Práce s proměnnými

V této části jsou uvedeny základní informace pro práci s maticemi a vektory.

2.1 Typy

V MATLABu lze používat mnoho datových typů, od číselných formátů, přes řetězce a různá pole, až po datové struktury. Pro naši práci je nejdůležitější *číselný typ*. Proměnné mohou být *skalární* nebo mohou mít formát vícerozměrných polí. My budeme používat *jednorozměrná pole* (řádkové a sloupcové *vektory*) a *dvourozměrná pole* (*matice*). Číselné hodnoty mají implicitně dvojitou přesnost a mohou být reálné nebo komplexní. Matice nemusí být pouze plné, ale lze využít také řídké reprezentace. V desetinných číslech se píše desetinná tečka. Podrobný popis je možno najít například v elektronické dokumentaci či použít příkaz **help datatypes**.

2.2 Speciální proměnné a konstanty

ans	Výsledek posledního (nikam nepřirazeného) výrazu (funkce, ...).
eps	Relativní přesnost čísel v pohyblivé řádové čárce.
pi	Konstanta π .
i, j	Imaginární jednotka (lze ji předefinovat na klasickou proměnnou).
inf	Reprezentace pro nekonečno (například výsledek operace $1/0$).
nan	Reprezentace pro ne-číslo (výsledek operace $0/0$).

Další informace pomocí příkazu `help elmat`.

2.3 Vytváření matic a vektorů

Při vytváření matic a vektorů pracujeme s přiřazovacím příkazem „`=`“. Vlastní prvky píšeme mezi dvě hranaté závorky a oddělujeme mezerami nebo čárkami (v případě prvků v řádce) a dále řádky matic mezi sebou oddělujeme středníky (nebo píšeme každý řádek matice na zvláštní řádek).

Příkaz `a = [1 2 3 4 5 6 7 8 9 10]` vytvoří řádkový vektor dimenze 10, zatímco příkaz `B = [1 2 3 ; 4 5 6 ; 7 8 9]` vytvoří čtvercovou matici řádu 3.

2.4 Generování vektorů

Použitím schématu `[od:krok:do]` lze generovat vektor s prvky od čísla *od* s krokem *krok* do čísla *do*, přičemž fantazii při výběru *od*, *krok* a *do* se meze nekladou. Například příkaz `[1:0.2:2]` vygeneruje vektor `[1 1.2 1.4 1.6 1.8 2]`. Vynecháním kroku vygenerujeme příkazem `[od:do]` posloupnost čísel od čísla *od* do čísla *do* s krokem 1.

2.5 Indexování prvků

Pro odkazování se na prvky vektorů či matic platí v MATLABu stejná pravidla jako v lineární algebře. Na *k*-tý prvek vektoru *a* se odkážeme příkazem `a(k)`, přičemž *k* může být libovolný vektor, který obsahuje indexy označující libovolné prvky z *a*. Zde lze s oblibou využít vygenerovaných vektorů nebo použít znak „`:`“ a odkázat se tak příkazem `a(:)` na všechny prvky vektoru *a*. Podobně na prvek matice *B* na *r*-tém řádku a v *s*-tém sloupci se lze odkázat příkazem `B(r,s)`. Opět lze využít vektorů a znaku „`:`“. Existuje také operátor `end`, kterým se lze odkázat na poslední index pole.

2.6 Operace s maticemi a vektory

V MATLABu lze s maticemi a vektory provádět všechny operace známé z matematiky. Označení operátorů odpovídá zažitým konvencím (+, −, *, /, ^). Operátory s tečkou (například „`.*`“) označují prvkové operace. V případě „`.*`“ by se tedy neprováděl maticový nebo vektorový součin, ale součin korespondujících prvků. Následuje seznam některých dalších základních příkazů.

'	Hermitova transpozice matice či vektoru.
.'	Transpozice matice či vektoru (non-conjugate).
length	Délka vektoru.
size	Rozměry matice či vektoru.
zeros	Vytváří nulovou matici.
ones	Vytváří matici, která má všude jedničky.
eye	Vytváří jednotkovou matici.
diag	Vytváření diagonálních matic nebo získávání diagonál z matic.
find	Najde indexy nenulových prvků nebo prvků s jistou vlastností. Například příkaz <code>find(X>1)</code> najde indexy všech prvků v matici <i>X</i> , které jsou větší než 1.

Další informace lze získat příkazy `help ops` a `help elmat`.

3 Základní matematické funkce

MATLAB obsahuje mnoho základních předdefinovaných matematických funkcí. Kompletní seznam lze vypsat pomocí příkazu `help elfun`. Následující tabulka obsahuje pouze několik ukázkových funkcí.

<code>sin</code>	Sinus. Vstupní argumenty goniometrických funkcí jsou v radiánech.
<code>sinh</code>	Hyperbolický sinus.
<code>exp</code>	Mocnina Eulerova čísla e^x .
<code>log</code>	Přirozený logaritmus.
<code>log10</code>	Logaritmus se základem 10.
<code>abs</code>	Absolutní velikost.
<code>imag</code>	Imaginární složka komplexního čísla.
<code>real</code>	Reálná složka komplexního čísla.
<code>sqrt</code>	Druhá odmocnina.
<code>round</code>	Zaokrouhlení k nejbližšímu celému číslu.

4 Vytváření skriptů

Skript je textový soubor s příponou „`m`“ (M-soubor), který obsahuje posloupnost libovolných příkazů MATLABu. Spouští se voláním jména souboru. Všechny proměnné zůstávají po provedení skriptu v globální paměti. Skript může obsahovat komentáře uvozené znakem procenta „`%`“. Jestliže několik prvních řádek skriptu jsou komentáře, lze tyto řádky vypsat příkazem `help <jmeno_skriptu>`. To platí také pro funkce.

5 Vytváření funkcí

Funkce je posloupnost libovolných příkazů MATLABu uvozená klíčovým slovem `function` podle schématu

```
function [vystup] = jmeno_fce(vstup)
```

Vstupními i výstupními parametry mohou být libovolné proměnné (skaláry, vektory, matice, struktury, řetězce, ...), které mezi sebou oddělujeme čárkami. Rozlišujeme funkce *globální*, které jsou zapsány v M-souboru se shodným jménem jako jméno funkce, a funkce *lokální*, které mohou být zapsány za tělem nadřazené globální funkce. Funkci lze v jejím těle ukončit příkazem `return`.

Další informace lze získat příkazem `help function` nebo příkazem `help lang`, kde lze najít seznam příkazů vztahujících se k tomuto tématu.

6 Řídící struktury

Stejně jako v každém programovacím jazyku, také v MATLABu existují řídicí struktury typu různých cyklů a podmíněných příkazů, jejichž použití je víceméně konvenční. Více informací opět v nápovědě `help lang`.

6.1 Podmíněný příkaz `if`

Základní schéma příkazu je

```
if <logicky_vyraz>
    <prikazy>
elseif <logicky_vyraz>
    <prikazy>
```

```

else
    <prikazy>
end

```

přičemž v logický výrazech je možné používat klasické relační a logické operátory (`==`, `=`, `>`, `>`, `&`, `|`, ...). Další informace pomocí příkazu `help ops`.

6.2 Podmíněný příkaz switch

Umožňuje podle hodnoty výrazu `vyraz` vykonat různé příkazy.

```

switch <vyraz>
    case <hodnota1>
        <prikazy>
    case <hodnota2>
        <prikazy>
    ...
    otherwise
        <prikazy>
end

```

6.3 Cyklus for

Provádí určité příkazy po daný počet opakování. Schéma klasického použití je

```

for <promenna> = <vektor>
    <prikazy>
end

```

kde `<vektor>` je obvykle nějaký vygenerovaný vektor typu `[od:krok:do]`.

6.4 Cyklus while

Provádí určité příkazy pokud platí daná podmínka.

```

while <logicky_vyraz>
    <prikazy>
end

```

7 Ostatní

Uvedené shrnutí postihuje velmi stručně pouze vybraná témata ze základů programování v MATLABu, proto bych ještě jednou připomenul existenci velmi dobré elektronické dokumentace. Vřele doporučuji rovněž text prof. Baldy.

Kromě výše uvedených témat doporučuji dále projít

```

help strfun    Funkce pro práci s řetězci.
help timefun   Funkce pro práci s datumem a časem.
help iofun     Funkce pro vstup a výstup z a do souboru.
help demos     :-)

```