Cvičení 2 - zadání a řešení úloh

Základy numerické matematiky - NMNM201 Verze z 8. října 2024

---

1 Úvod do MATLABu

1.1 Úvodní komentáře

Pro práci s MATLABem je vhodnější anglická klávesnice. Budeme potřebovat zejména hranaté závorky `[`, `]`, apostrof `'`, zpětné lomítko `\`, symbol pro mocnění `^`.

Pokud si chcete nainstalovat MATLAB na svůj počítač (což doporučujeme), informace jsou zde: `www.mff.cuni.cz/cs/math/vnitrni-zalezitosti/pocitace-a-technika/matlab`.

Připravili jsme pro vás „tahák“ se základními příkazy. Ke stažení je zde: `www.karlin.mff.cuni.cz/~blechta/znm/assets/MATLAB_tahak.pdf`.

1.2 Proměnné

- Vytváří se za pochodu (tj. není je třeba předem definovat).
- Lze změnit typ (`p = 1, p = true, p = 1+i`).
- Příkazy pro výpis: `who`, `whos`, podívat se do panelu „Workspace“.
- Jména musí začínat písmenem, mohou následovat písmena, čísla nebo podtržítka. Pozor, lze si předefinovat vestavěnou funkci, pak nás zachraňuje příkaz `clear`.
- Existují některé předdefinované konstanty, se kterými se časem blíže potkáme: `ans`, `eps`, `Inf`, `NaN`, komplexní jednotka `i` (respektive `1i`).

1.3 Vektory a matice

Pole, tj. vektory a matice jsou základní data v MATLABu (skalár je vnímán jako pole 1×1). Při vytváření pole začínáme `[` a ukončujeme `]`, prvky zadáváme po řádcích. Mezera nebo čárka oddělují prvky na řádku, středník řádek ukončuje.

Úloha 1. *Vytvořte si jednotkovou matici řádu 3 pojmenovanou jako* `E`*.*

Řešení. `E = [1 0 0; 0 1 0; 0 0 1];` □

Transpozice `'` se může hodit při vytváření vektorů, `v = [1 2 3]'` je sloupcový vektor.

Existují příkazy pro vytvoření některých speciálních vektorů a matic, viz tahák. Zejména dvojtečková notace je užitečná.

Úloha 2. *Mějme vektory* `u = [6 2 4]`, `v = 1:4`, `w = [3; -4; 2; -6]` *a matici* `M = ones(4)`*. Určete, které z následujících operací jsou definovány v MATLABu. Definované operace vyhodnoťte:*

```
u*v    u'*v    u*v'    v*w    w*v    M*v    M*w    v*M
```

Pomocí hranatých závorek a středníků lze pole spojovat, pokud mají vhodný rozměr. Například pro `v=[1 2 3]'` funguje

```
B = [rand(3) v]      C = [ones(5,3); v']
```

narozdíl od `D = [eye(7) v]`. Sami vyzkoušejte.

1.4 Indexování a operace s maticemi

Pro přístup k prvkům ve vektorech a maticích používáme kulaté závorky. Můžeme přistupovat k více prvkům najednou, jestliže do kulaté závorky zadáme vektor.

Úloha 3. *Mějme vektor* `v = [1 3 8 9 7 9 21 3.5]`. *Jak se vyhodnotí následující příkazy:*
`v(3)` `v(end)` `v(1:2:4)` `v([1:3,end-2:end])`.

Úloha 4 (Navíc)**.** *Pomocí dvojtečkové notace otočte pořadí prvků ve* `v` *z předchozí úlohy.*

Řešení. `v = v(end:-1:1)` □

Dvojtečka má zvláštní význam; `A(1:end,3)` lze nahradit `A(:,3)` a podobně.

Úloha 5. *Odhadněte, co dělají příkazy*
`M(1,end)` `M(:,3)` `M(1,:)` `M(1:2:end,:)` `M(1:2,2:4)`
a vyzkoušejte pro náhodnou matici `M=rand(6)`.

1.5 Matematické a vestavěné funkce

MATLAB umožňuje obvykle definované sčítání, odčítání, násobení vektorů a matic (odpovídajících rozměrů). Kromě toho umožňuje provádět specifické operace s jednotlivými prvky v poli. Takto jsou definovány takzvané tečkové operace `.*`, `./`, `.^`.

Úloha 6. *Pro vektory* `v = [1 2 3]'` *a* `w = [3 -1 2]'` *vyzkoušejte operace*
`v .* w` `v ./ w` `v.^3` `3.^w`

MATLAB dále obsahuje řadu vestavěných matematických funkcí (sin, log, apod.) jejichž vstupem mohou být i matice a vektory. Tyto operace se pak provedou pro každou jednotlivou složku a výstupem je opět matice či vektor. Vyzkoušejte pro `sin(pi*(0:0.5:2))`.

Úloha 7 (Navíc)**.** *Odhadněte numericky limitu*

$$\lim_{x\to 0} \frac{e^x - 1}{x}.$$

Vytvořte vektor $x = (0.1,\ 0.01, \ldots,\ 0.000001)$ *pomocí dvojtečkové notace a operace po složkách. Poté vyjádřete výraz* $\frac{e^x-1}{x}$. *Výsledek porovnejte s analytickým řešením.*

Řešení. `a = [-1:-1:-6]; x = 10.^a; y = (exp(x)-1)./x` □

Řada funkcí pro matice, které znáte z lineární algebry, je v MATLABu naimplementována. Vyzkoušejte například
`rank` `det` `norm` `eig`
Více informací o každé funkci naleznete v nápovědě (například `help norm`).

1.6 Vykreslování

MATLAB nevykresluje funkce nebo spojité křivky, ale pouze body. Základním příkazem je `plot`. Vyzkoušejte si příkazy
`x=0:0.5:6;`
`plot(sin(x));` a srovnejte s `plot(x,sin(x));`
Styl čáry lze změnit, vyzkoušejte `plot(x,sin(x),'m:');` a další možnosti si nejděte v dokumentaci příkazem `doc plot`.

Výsledný graf lze doplnit např. názvem, legendou, popisky os: `title`, `legend`, `xlabel`.

Úloha 8. *Vykreslete funkci* $\cos(x)$ *na intervalu* $(\pi, 3\pi)$ *zelenou a přerušovanou („dashed“) čárou. Přidejte legendu a název grafu.*

1.7 Skripty a vlastní funkce

Skripty umožňují zapsat posloupnost příkazů, ideálně pro opakované spouštění. Příkazy jsou uloženy v textovém souboru s koncovkou `.m`. Lze vytvořit například příkazem `edit muj_skript`

Pro spuštění skriptu napište v Command window jeho název nebo v Editoru stiskněte tlačítko „Run“

Úloha 9. *Vytvořme m-file (MATLAB skript), ve kterém postupně zrealizujeme rotaci čtverce na základě následujících bodů:*

1. *Zadefinujme v MATLABu čtverec s vrcholy* $[1,0],[0,1],[-1,0],[0,-1]$ *jako matice* A *typu* 2×4. *(První řádek bude obsahovat x-ové souřadnice a druhý řádek y-ové souřadnice vrcholů.) Pomocí příkazu* `fill` *můžeme vykreslit např. červeně vyplněný polygon určený dvěma řádkovými vektory obsahující x-ové a y-ové souřadnice vrcholů.*

   Řešení.
   ```
   A = [1 0 -1 0; 0 1 0 -1];
   fill(A(1,:), A(2,:), 'r')
   ```
   □

2. *Proveďme rotaci tohoto čtverce o úhel* $\theta = \pi/4$ *a výsledný objekt opět vykresleme pomocí příkazu* `fill` *např. s modrou výplní. Abychom oba čtverce mohli porovnat, vykreslíme je do jednoho grafu pomocí příkazu* `hold on`.

   Řešení.
   ```
   theta = pi/4;
   R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
   B = R*A;
   hold on
   fill(B(1,:), B(2,:), 'b')
   ```
   □

3. *Aby měřítko na obou osách bylo stejné, použijme příkaz* `axis equal tight`. *Můžeme také zapnout mřížku na pozadí pomocí příkazu* `grid on`.

Pro zadefinování vlastní funkce je třeba vytvořit zvláštní `m`-file (jedna funkce, jeden soubor) se jménem nové funkce, např. `NazevFunkce.m`. Hlavička souboru pak vypadá:

```
function [out1,...,outN] = NazevFunkce(in1,...inM)
% NazevFunkce: Stručný popis (volitelně)
% ...
příkazy;
```

Funkci poté spustíme takto:

```
[výstup1,...,výstupN] = NazevFunkce(vstup1,...vstupM)
```

Úloha 10 (Navazující na úlohu 9)**.** *Upravte skript z úlohy 9 jako funkci, která bude mít na vstupu matici („čtverec“) a úhel* θ *a výstupem bude zrotovaný čtverec.*

1.8 Logické operátory, cykly a větvení kódu

Logické proměnné v MATLABu nabývají hodnot `true` a `false`. (MATLAB nicméně jako `false` vnímá i nulové číslo a jako `true` jakoukoli nenulovou hodnotu.)

Pro přehled relačních a logických operátorů (tj. těch, jejichž výstupem je `true` nebo `false`) se podívejte na tahák.

Logické proměnné se obvykle používají pro cykly a větvení kódu (podmínka `if`). V taháku najdete, jak se zapisují.

Úloha 11. *Vytvořte matici* 8×6, *která bude mít na pozici* (i,j) *hodnotu* $i * (-1)^j$.

Řešení.
```
for i=1:8
  for j=1:6
    A(i,j) = i*(-1)^j;
  end
end
```
Nebo také `A=[1:8]'*((-1).^(1:6));` □

Úloha 12 (navíc, navazující na úlohu 9)**.** *Nechť R je matice, která rotuje objekty proti směru hodinových ručiček o úhel $\theta = \pi/8$ a nechť $S = 0.9 * R$. Potom matice S současně zrotuje a zmenšuje objekt. Modifikujme kód z úlohy 9, abychom dvacetkrát rotovali a zmenšili (ve stejném míře v každém kroku) čtverec A s vrcholy $[1,0],[0,1],[-1,0],[0,-1]$. Vykresleme obrázek obsahujících všech 21 čtverců.*

Řešení.
```
A = [1 0 -1 0; 0 1 0 -1];
fill(A(1,:), A(2,:), 'r')
hold on

theta = pi/8;
R = [cos(theta) -sin(theta); sin(theta) cos(theta)];

for i = 1:20
  S = 0.9*R;
  A = S*A;
  fill(A(1,:), A(2,:), 'b');
end

axis equal tight, grid on
```
□

1.9 Omezení počítače: čas a paměť

Úloha 13. *Mějme čtvercovou matici řádu n a vektor s n prvky. Spočtěte, kolik operací násobení a sčítání stojí operace násobení matice vektorem.*

Řešení. Potřebujeme celkem $2n^2 - n$ operací. (Výsledek má n prvků a každý spočítáme díky n násobení a $n-1$ sčítání.) □

Úloha 14. *Pomocí příkazů* `tic` *a* `toc` *(start a konec stopek) změřte násobení náhodné čtvercové matice vektorem s řádem $n = 100, 1000, 10\,000$. S využitím výpočtu v předchozí úloze přepočtěte výsledek jako počet elementárních operací (*flop*ů) za vteřinu.*

Řešení.
```
n_all = [100 1000 10000];
for n = n_all
   A = rand(n);
   b = rand(n,1);
   tic;
   A*b;
   time = toc;
   flops = 2*n*n-n;
   flops/time
end
```
□

Současný nejvýkonnější superpočítač dosahuje teoretického výkonu přibližně 10^{18} flopů za vteřinu. I tak to pro některé aplikace nestačí.

Úloha 15. *Vyzkoušejte, jakou největší náhodnou čtvercovou matici dokážete v MATLABu vytvořit (a uložit). Pomocí příkazů* `whos` *zjistěte, kolik paměti zabírá.*

Řešení. Tato velikost závisí na paměti RAM počítače a nastavení MATLABu (u mě to bylo například matice řádu 32 272, která zabrala 7.75 GB). Lze zjistit například pomocí metody bisekce. □