Vytěžování dat, cvičení 1: Úvod do Matlabu

Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Fakulta elektrotechnická, ČVUT

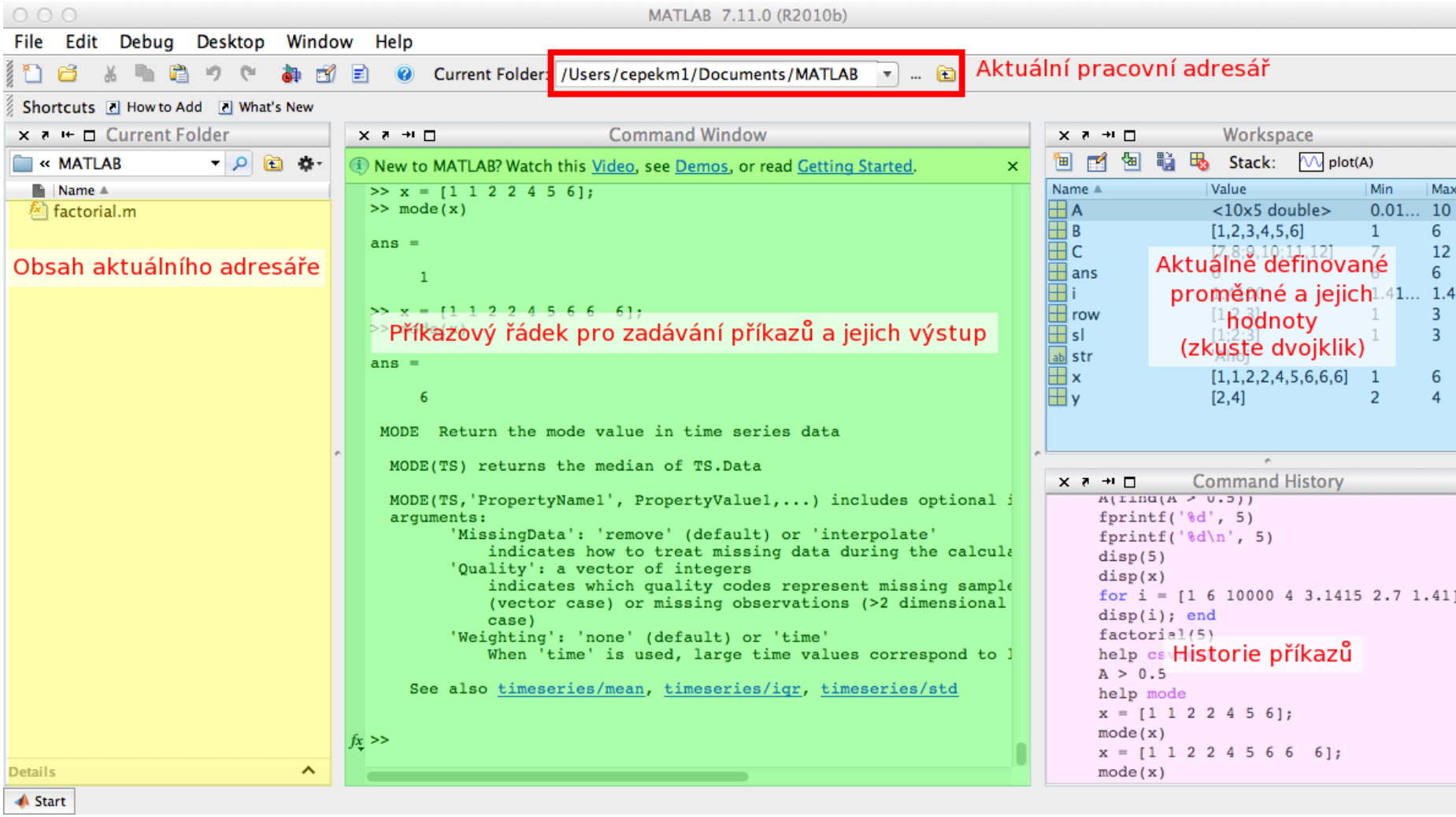
Proč proboha Matlab?

- Matlab je SW pro vědecko-technické výpočty a de facto průmyslový standard v mnoha odvětvých.
- Vhodný pro rychlé prototypování a zkoušení aplikací.
- Obsahuje skriptovací jazyk se spoustou knihoven pro různé oblasti (včetně vytěžování dat).
- Matlab je v tomto předmětu kompromis mezi "klikacími" nástroji typu RapidMiner a programováním v jazycích typu Java.
- Dovolí nám hrát si s implementací algoritům, ale spoutu věcí řeší interně za vás.

Kde získat Matlab?

- FEL má multilicenci, kterou můžete využít.
- Kopii instalačního DVD můžete získat na `http://www.fel.cvut.cz/user-info/matlab.html`
- Zde se musíte přihlásit hlavním přístupovým heslem a pak už můžete stahovat a instalovat.

Základní uživatelské rozhraní



Jednoduché příkazy

Do příkazového okna zkuste zadat následující příkazy. Jaký je jejich význam a co je výsledkem?

- `1 + 5`
- Ukládá se někam výsledek?
- `x = 3`
- `y = 1 * 6`
- `x + y`
- `z = x + y`
- `z = x + y;`
- Jaký je rozdíl mezi předchozími dvěma příkazy?

Matice

- Hlavní síla Matlabu spočívá v práci s maticemi.
- Jak vypadají matice z následujících příkazů?
- Jak zjistíme rozměry matic - funkce `shape` `length`

```
A = [1 2; 3 4; 5 6]
B = [1 2 3; 4 5 6]
```

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Matcové operace

- Jaké znáte maticové operace?
- Maticové sčítání, odčítání, násobení, dělení, transpozice
- `C = [7 8; 9 10; 11 12]`
- `A * 2`
- `A + C`
- `A * B`
- `A .* C`
- Jak se liší výsledky posledních dvou příkazů?
- Co dělá `A'` ?

Funkce pro práci s maticemi

- Matice náhodných čísel `rand(<počet prvků v 1. dimenzi>, <v 2. dimenzi>, ...)`
  - například `A = rand(10, 5)`
- Velikost matice, počet prvků v jednotlivých dimenzích – `size(A)`
- Matice samých nul – `zeros(10, 6)`
- Inverzní matice – `inv(A)`
- Indikace splnění podmínky – `A > 0.5`
- Vybere indexy z pole na základě podmínky – `find(A > 0.5)`
- Spočítá indexy z pole na základě podmínky – `sum(A > 0.5)`

Nápověda v Matlabu

Existuje několik možnosti získání nápovědy pro příkaz Matlabu.

- Jednoduchá textová nápověda – příkaz `help <příkaz>`
- Hypertextová nápověda – příkaz `doc <příkaz>`
- Mathworld Knowledge Base – `http://www.mathworks.com/help/techdoc/`

A co vektory?

- Vektory jsou matice, které mají jen jeden řádek/sloupec.
- Vytvořte řádkový a sloupcový vektor hodnot 1, 2, 3, 4, 5, 6.
- `row = [1 2 3 4 5 6]`
- `sl = [1; 2; 3; 4; 5; 6]`
- Jak vytvořím z řádkového vektoru sloupcový?
- Transpozicí – `row'`

Generování posloupností čísel

Pokud potřebujete vytvořit posloupnost čísel, můžete využít příslušný operátor "**:**"(dvojtečka).
Pomocí něj vygeneruje vektor, který obsahuje posloupnost čísel v zadaném rozmezí – zkuste

- `[1:10]; [-5:5]; [5.4:15.6]`
- můžete zadat i krok, se kterým se posloupnost mění. Zkuste: `[1:0.5:10]; [5:-1:5]; [5.4:0.2:15.6]`

Přístup k prvkům matice

Zopakujte `A = rand(10, 5)`, ať máme všichni stejné rozměry matice.

- Přístup k jednomu konkrétnímu prvku – `A(1,2)`
- POZOR – indexy se číslují od **1**!
- Přístup k podmaticím – uvedu rozsahy indexů, které chci v mít podmatici `A(3:5, 1:3)`
- Místo čísel můžu uvést i vektory indexů, které chci zobrazit.
- Zkuste zobrazit prvky ve řádcích 1., 5., 3., 8. a sloupcích 3., 2., 1.
- `x = [1 5 3 8]; y = [3 2 1]; A(x,y)`

Přiřazování hodnot do matice

- Úplně stejně jako výběr prvků na minulém slajdu, jen výběr umístím na levou stranu přiřazení.
- Přiřazení jedné hodnoty – přiřaďte 10 do levého horního rohu matice.
- `A(1,1) = 10`
- Přiřazení do podmatice – přiřadte hodnoty 1, 2, 3, 4 na souřadnice (2,2), (2,4), (4,2), (4,4)
- `x=[2 4]; y=[2 4]; A(x,y) = [1 3; 2 4]`

Datové typy

- V Matlabu, stejně jako v jiných skriptovacích jazycích, proměnné nemají pevný datový typ.
- Základní datové typy jsou:
  - Čísla, Logické hodnoty, Řetězce
  - Matice – matice hodnot jednoho datového typu
  - Struktury – skupina několika pojmenovaných hodnot zabalených do jedné proměnné
  - Buňková pole (Cell arrays) – pole hodnotu různých datových typů
  - Odkazy (Handles)
  - Objekty

Více o datových typech se lze dozvědět na `http://www.mathworks.com/help/techdoc/matlab_prog/f2-43934.html`

Programování pro Matlab

Doteď jsme zkoušeli interaktivní práci se systémem Matlab. Teď zkusíme programování.
Programy se zadávají do tzv. M-souborů (M-file) což je jen textový soubor s koncovkou `.m`. Existují dva typy M-souborů

- Skripty – obsahují jen posloupnost příkazů Matlabu,
- Soubory definující funkci – obsahuje definici funkce, kterou lze využít při interaktivní práci s Matlabem nebo v jiných M-souborech.

Pro edtaci obou typů M-souborů můžete použít:

- libovolný textový editor,
- editor integrovaný přímo do Matlabu.

My použijeme druhou možnost. Interní editor se spouští buď příkazem `edit` nebo z menu File > New Script resp. File > New Function.

Struktura funkce

Základní struktura funkce vypadá takto:

```
function [soucet,rozdil] = SlozitaFunkce(a,b)
%
% Funkce se jmenuje SlozitaFunkce a ma dva vstupni
% parametry - a, b. A dva vystupni parametry -
% soucet, rozdil.
% Tohle je help k funkci SlozitaFunkce. Vypisuje se
% zadanim prikazu 'help SlozitaFunkce'.
%
%Naplneni vystupniho parametru se deje prirazenim
     soucet = a+b;
     rozdil = a-b;
end % Nepovinne end
```

Funkce **musí** být uložena v souboru `slozitafunkce.m`.

Volání funkce

- Funkci můžete zavolat pomocí jejího jména.
- `[s, r] = slozitafunkce(10, 4)`
- Jak Matlab zjistí, kterou funkce voláte?
  - Matlab se podívá do aktualního adresáře, zda v aktuálním adresáři existuje soubor `slozitafunkce.m`
  - Pokud Matlab funkci nenajde v aktualním adresáři, podívá se na vnitřní proměnné `PATH` na seznam adresářů, které se mají prohledávat a zkusí, zda některý z nich funkci neobsahuje.
- Z toho plyne, že je důležité, ve kterém adresáři se nacházíte :).

Příkazy řízení programu

Provádění funkce můžete ovlivnit pomocí standardních konstrukcí, které znáte z jiných programovacích jazyků.

- Podmínky
  - if – then – else
  - switch – case
- Cykly
  - for
  - while

Podmínky

if – then – else

```
if x > 11
     disp('x je vetsi nez 11.');
elseif x < 5
     disp('x je mensi nez 11 a take mensi nez 5.')
else
     disp('Je to nejak uplne jinak.');
end
```

Podmínky (II)

switch – case

```
switch x
     case {0, 1, 2, 3}
          disp('x je v intervalu 0-3.');
     case 4
          disp('x je 4.');
     otherwise
          disp('x je uplne jine.')
end
```

Cykly – for cyklus

For cyklus funguje trochu jinak než jste zvyklí z Javy. For cyklus v Matlabu iteruje přes všechny hodnoty vektoru.

```
for i = 1:10
     fprintf('%d ',i); %muzete pouzit take disp(i)
end

for i = [1 6 10000 4 3.1415 2.7 1.41]
     disp(i);
end
```

POZOR – použití cyklů je extrémně nevýhodné, protože Matlab parsuje každý řádek znovu $\Rightarrow$ pokud můžete zkuste cykly obejít (například maticovými operacemi, příkazy `find`, `sum`, `arrayfun`...).

Cykly – While cyklus

While cyklus je mnohem blíž tomu, co znáte z jiných jazyků.

```
i = 1;
while i < 10
     fprintf('%d',i);
     i = i+1;
end
```

Existují i příkazy na přerušení cyklů

- break – ukončuje cyklus if i = 5, break, end
- continue – přeskakuje zbytek iterace if i = 5, continue, end

Příklad - faktorial

```
function fac = factorial(n)
if n < 0
     disp('n musi byt vetsi nez 0!');
     fac = 0;
     return;
end
if n == 0
     fac = 1;
     return;
end
fac = 1;
for i = 1:n
     fac = fac * i;
end
end
```

Další zdroje

Zde jsou odkazy na další výukové materiály o Matlabu

- `http://labe.felk.cvut.cz/~posik/y33aui/uvod-do-matlabu/`
- `http://www.mathworks.com/help/techdoc/matlab_product_page.html`
- `http://www.mathworks.com/moler/intro.pdf`
- `http://www.maths.dundee.ac.uk/~ftp/na-reports/MatlabNotes.pdf`
- a mnoho dalších...