

ÚVOD DO MATLABU A ZÁKLADNÍ OPERACE SE SIGNÁLY

Jan Černocký, FIT VUT Brno

MATLAB (MATrix LABoratory) – software pro vědecké výpočty a zobrazování.

1 Několik praktických rad

- po startu Windows spusťte Matlab z adresáře `Q:\matlab6p1` dvojitým kliknutím na jeho ikonu. Chybová hlášení ignorujte.
- v Matlabu můžete zadávat příkazy pouze, je-li kurzor na příkazové řádce `>>`
- ve vedlejší okně si otevřete v libovolném editoru (můžete použít i vestavěný editor Matlabu, který umí “obarvovat”) **textový soubor**, kam si **vše zapisujete** (příkazy, výsledky, Vaše poznámky). Příkazy můžete do Matlabu kopírovat pomocí copy (`Ctrl-c`) and paste (`Ctrl-v`). Po ukončení cvičení Vám pak zbyde přesný záznam toho, co jste dělali, a co (snad) chodilo.
- Obsahuje-li řádek znak `%`, bere se jeho část za tímto znakem jako *poznámka* a není Matlabem prováděna.
- v Matlabu je možné se k předchozím příkazům vracet pomocí `↑`. Návrat může být i “selektivní”, např. k již použitým příkazům `plot` se můžete vrátit pomocí:

`pl ↑ ↑ ↑ ...`

2 Stručný přehled základních dovedností Matlabu

1. Proměnné:

- skalární: `a=1;`
- sloupcový vektor: `vecsl=[1; 2; 3]`
- řádkový vektor: `vecrad=[2.5 16e-3 85]`
- matice: `mat=[1 2 3; 4 5 6; 7 8 9]`

2. Zobrazení obsahu proměnné:

- textové: jméno proměnné bez středníku.
- grafické s x-ovými souřadnicemi bodů `1,2,3,...`: `plot(vec)`
- grafické se zadáním x-ové i y-ové souřadnice: `plot(vecx,vecy)`
- zjištění velikosti proměnné:
`length(vec)` – délka vektoru
`size(mat)` – počet řádků a sloupců matice (dvouprvkový vektor)

3. Pomoc !!!

- `help jméno_funkce`
- `lookfor klíčové_slovo`

4. Case sensitivity: Matlab rozlišuje v názvech proměnných malá a velká písmena, takže `mujvektor` \neq `MujVektor` \neq `MUJVEKTOR`.

Příklady

1. Napište skalární, vektorovou a maticovou proměnnou. Zobrazte jejich obsah jako text a graficky.
2. Jak zadáte komplexní číslo ? Co je výsledkem grafického zobrazení vektoru komplexních čísel ?

3 Základní operace s vektory a maticemi

- přičtení/odečtení/násobení/dělení skalárem vektoru nebo matice: $v+s$, $v-s$, $v*s$, v/s
- skalární součin dvou vektorů: `rad_v*sl_v`
- součin dvou vektorů nebo matic prvek po prvku: `rad_v.*rad_v`, `sl_v.*sl_v`, `mat.*mat`
- generování aritmetické posloupnosti:
s krokem 1: `min_hodnota:max_hodnota`
s volitelným krokem: `min_hodnota:krok:max_hodnota`
- transpozice vektoru/matice: `vektor'`, `matice'`
- funkce (`abs`, `sin`, `cos`, `log`,...) pracují většinou i s vektory či s maticemi prvek po prvku.
- vytvoření matice nul či jedniček:
`zeros(počet_řádků,počet_sloupců)`, `ones(počet_řádků,počet_sloupců)`
- přístup k prvkům vektorů a matic: `vec(1)`, `vec(2:5)`, `mat(3,4)`, `mat(3,:)`, `mat(1:5,4)`
- nalezení indexů pro prvky vyhovující nějaké podmínce: `find`, např: `n = -10:10; ii = find(n<10);`

Příklady

1. Vytvořte matici o rozměrech 4×5 se všemi prvky rovnými jedné. Přičtěte ke všem prvkům 3.25, pak vynásobte všechny prvky -1.5. Zobrazte třetí a čtvrtý sloupec matice.
2. Naplňte vektor hodnotami funkce $\cos(x)$ pro $0 \leq x < 2\pi$. Zobrazte se správnými hodnotami na osách.
3. Najděte všechna místa, kde je funkce $\cos(x)$ záporná, obarvte je zeleně.

4 Generování a zobrazování signálů

Diskrétní signály se budou generovat velice snadno - diskrétní signál si totiž můžeme přímo představit jako vektor hodnot. Vždy musíme předem nadefinovat rozsah nezávislé proměnné (nejčastěji n). Zobrazovat budeme pomocí funkce `stem`. Diskrétní jednotkový impuls $\delta(n)$ tak třeba “naprogramujeme” takto:

```
n = -10:10;
delta = zeros(size(n));
delta(find(n==0)) = 1;
stem(n,delta)
```

Všimněte si použití podmínky v argumentu – tento trik budeme velice často využívat!

Se **signály se spojitým časem** se teoreticky v žádném programovém prostředí nedá pracovat – časová osa obsahuje ∞ hodnot. Musíme si pomoci vzorkováním s malým krokem (o teorii vzorkování se dozvíte později). Zobrazovat budeme pomocí `plot`, abychom dosáhli zdání spojitého času. Příklad pro pravoúhlý impuls:

$$x(t) = \begin{cases} 2 & \text{pro } -1 \leq t \leq 2 \\ 0 & \text{jinde} \end{cases}$$

```
t = -10:0.01:10;
s = zeros(size(t));
s(find(t>=-1 & t<=2)) = 2;
plot(t,s)
```

Příklady

1. Vytvořte a zobrazte a pro $n \in [-5, 5]$ zobrazte diskrétní jednotkový skok $\sigma(n)$.
2. Vygenerujte trojúhelníkový impuls se spojitým časem:

$$x(t) = \begin{cases} t+1 & \text{pro } -1 \leq t \leq 0 \\ -t/3+1 & \text{pro } 0 < t \leq 3 \\ 0 & \text{jinde} \end{cases}$$

5 Transformace nazávislé proměnné – času

Na přednášce jste se dozvěděli o časovém posunutí (zpoždění, předběhnutí), o obrácení časové osy či o kontrakci či dilataci času (neboli zrychlení či zpomalení signálu). V Matlabu se tyto operace dají provádět a zobrazovat tak, že si nejprve vygenerujeme standardní časovou osu a poté modifikovanou. Podle ní vygenerujeme signál. Příklad pro zpožděný jednotkový impuls $\delta(n - 1)$:

```
n = -10:10;
nminus1 = n-1;
deltaminus1 = zeros(size(nminus1));
deltaminus1(find(nminus1==0)) = 1;
stem(n,deltaminus1)
```

Všimněte si, že musíme zobrazovat proti **původní ose!** Podobný příklad pro pravoúhlý impuls (viz výše), předběhnutý o 2 vteřiny: $x(t + 2)$:

```
t = -10:0.01:10;
tplus2 = t+2;
splus2 = zeros(size(tplus2));
splus2(find(tplus2>=-1 & tplus2<=2)) = 2;
plot(t,splus2)
```

Příklady

Pracujte s jednotkovým skokem $\sigma(n)$ a trojúhelníkovým impulsem $s(t)$ z předcházejících příkladů.

1. Zobrazte diskrétní jednotkový skok s následujícími transformacemi časové osy: $\sigma(n - 2)$, $\sigma(n + 2)$, $\sigma(-n - 2)$, $\sigma(-n + 2)$. Popište slovy, co vidíte.
2. Vygenerujte trojúhelníkový impuls se spojitým časem pro tyto modifikace časové osy: $s(t - 2)$, $s(t + 2)$, $s(-t - 2)$, $s(t + 2)$, $s(t/2)$, $s(2t)$. Popište slovy, co vidíte.