FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

MATLAB: Vizualizace

Jaroslav Čmejla
Martin Vrátný
Miroslav Holada
ITE FM TUL

jaroslav.cmejla@tul.cz martin.vratny@tul.cz miroslav.holada@tul.cz

Vizualizace

- 2D grafy: `plot`, `bar`, `stem`, ...
- Bitmapy: `image`, `imagesc`, `imread`, `imwrite`, ...
- 3D grafy: `plot3`, `mesh`, `surf`, ...
- Možnosti exportování obrázků do různých formátů
- Úpravy obrázků příkazy z konzole nebo pomocí GUI
- Animace

Část 1

2D Vizualizace

2D: Základní používání příkazu `plot`

```
>> a=randn(1,10);
>> plot(a)
```

- Otevře se nové okno (`figure`), pokud již není nějaké otevřené
- Vykreslení se provede do aktivního okna
- osa x: indexy prvků a
- osa y: hodnoty proměnné a
- čárový graf: spojnice párů hodnot na osách x a y
- meze grafu se nastavují automaticky podle maximálních a minimálních hodnot

2D: Základní používání příkazu `plot`

```
>> x=0:0.1:0.9;
>> y=randn(1,10);
>> plot(x,y)
```

- Opět: vykreslení se provede do aktivního okna, není-li žádné, vytvoří se nové.
- Starý obsah okna je zrušen, není-li okno v režimu `hold on` (viz dále)
- osa x: hodnoty proměnné x
- osa y: hodnoty proměnné y
- čárový graf: spojnice párů hodnot na osách x a y
- x a y musí mít stejnou délku (vektory)

2D: Základní používání příkazu `plot`

- Příklad použití k vizualizaci průběhu funkce $\sin(2x)$

```
>> x=0:0.01:2*pi;
>> plot(x,sin(2*x)) % využíváme skládání výrazů
                    % a vektorizaci
                    % žádný cyklus for!!!
```

- Je třeba si uvědomit: jedná se pouze o spojnici bodů. Graf ve skutečnosti není dokonale hladký a nemusí vypovídat vše o vizualizované funkci.
- Příklad: funkce $\sin(\frac{1}{x})$ vizualizovaná kolem bodu 0

```
>> x=-2:0.1:2;
>> plot(x,sin(1./x))
```

- Příklad: trojúhelník s vrcholy [-1;0], [1;0] a [0;1]

```
>> plot([-1 1 0 -1],[0 0 1 0])
```

Příkaz `plot` a matice

```
>> A=randn(10,5);
>> B=repmat((0:0.1:0.9)',1,5);
```

- `plot(A)` vykreslí sloupce matice A
- `plot(B,A)` vykreslí po sloupcích dvojice hodnot B (osa x) a A (osa y). A a B tedy musí mít stejné rozměry.

Další používání příkazu `plot`

- `plot(x,y,p)` vykreslení grafu s parametry p. Např

```
>> plot(x,y,'ro:'); % červená, tečkovaná,
                    % ve spojnicích kolečka
                    % viz help plot
```

- `plot(x1,y1,x2,y2,…)` vykreslí více čar s automatickou volbou barev. Např.

```
>> plot(x,y,x,y.^2,x,y.^3)
```

- `plot(x1,y1,p1,x2,y2,p2,…)` vykreslí více čar najednou se zvolenými parametry.
- NaN se nevykresluje - použijeme chceme-li mezeru či prázdné místo

Více obrázků a více grafů v jednom

- Příkaz `subplot` - dlaždicové rozmístění os v obrázku

```
>> subplot(2,1,1)
>> plot(x1,y1)
>> subplot(2,1,2)
>> plot(x2,y2)
```

- Vkládání dalších os: příkaz `axes` nebo GUI.
- Přepínání režimu (ne)překreslování: příkaz `hold`

```
>> hold on
>> hold off
```

- Přepínání aktivních os příkazem `axes` nebo kliknutím.
- Nový obrázek a přepínání aktivního obrázku: příkaz `figure` nebo kliknutím.
- Zjišťování aktivního obrázku a os: `gcf`, `gca`.

Úpravy grafu příkazy z konzole

- Ovládání příkazy z konzole je potřebné, chceme-li něco automatizovat
- Vkládání popisů: `title`, `xlabel`, `ylabel`
- Legenda: `legend`

- způsob 1:

```
>> x = 0:0.01:2*pi
>> plot(x,sin(x));
>> hold on;
>> plot(x,cos(x));
>> legend('sin(x)','cos(x)')
```

- způsob 2:

```
>> x = 0:0.01:2*pi
>> plot(x,sin(x),'DisplayName','sin(x)');
>> hold on;
>> plot(x,cos(x),'DisplayName','cos(x)');
>> legend
```

- Zoom: `axis`
- ...a další
- Nastavování vlastností čáry již v příkazu `plot` např.

```
>> plot(1:10,5*rand(10,1),'marker','square',...
          'linestyle','-','color','r','linewidth',2);
```

Úpravy grafu příkazy z konzole

- Příkaz `plot`, a podobně i jiné, vrací ukazatel(e) na právě vložené nebo vykreslené objekty (osy - axes, čáry, sloupce, bitmapy, ...). Tedy je-li jich více, vrací vektor ukazatelů.
- Použití ukazatele (handle) a univerzálních příkazů `set` a `get`. Např.

```
>> h=plot(1:10,5*rand(10,1))
h =
  171.0076
>> set(h,'linestyle',':','color','g');
>> get(h,'linestyle') % vrací hodnotu parametru
ans =
    ':'
```

- `get(h)` vypíše všechny vlastnosti objektu h, které můžeme měnit příkazem `set`.

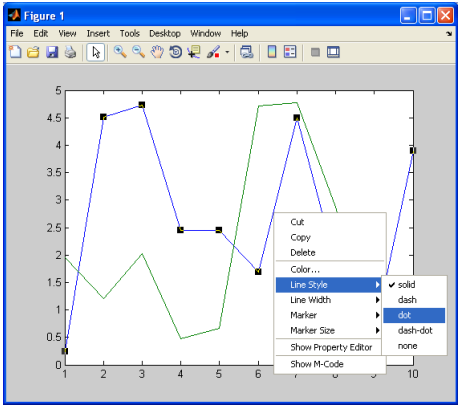
Propojování vlastností objektů: `linkprop`, `linkaxes`, `linkdata`

- Některé vlastnosti objektů lze propojit tak, že změna propojené vlastnosti u některého z objektů se automaticky kopíruje do ostatních propojených objektů.
- Typické využití:

```
>> figure
>> t = -10:0.1:10;
>> ax1 = subplot(2,1,1);
>> plot(t,sin(t));
>> ax2 = subplot(2,1,2);
>> plot(t,tan(t));
>> linkaxes([ax1,ax2],'x');
```

  Nyní mají osy "x" obou grafů vždy stejný rozsah (i když lupou měníme zobrazovanou oblast).
- Pomocí `linkdata` lze propojit aktuální obsah proměnné s grafem.

Úpravy skrze GUI



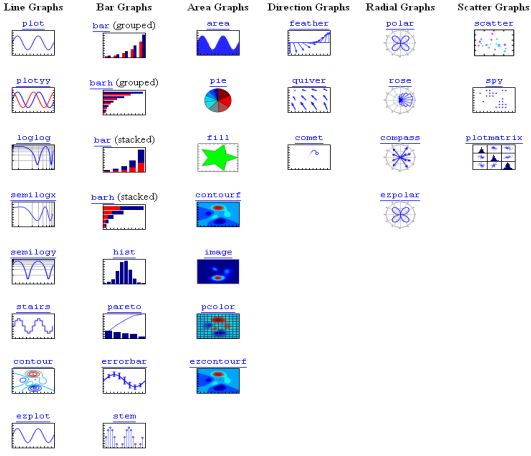
- Klikáme a klikáme...S každou verzí Matlabu možnosti přibývají a zlepšují se vlastnosti.

Další 2D grafy (1)

- `semilogx`, `semilogy`, `loglog`
- `stem`
- `bar`
- `polar`
- `histogram`

Další 2D grafy (2)

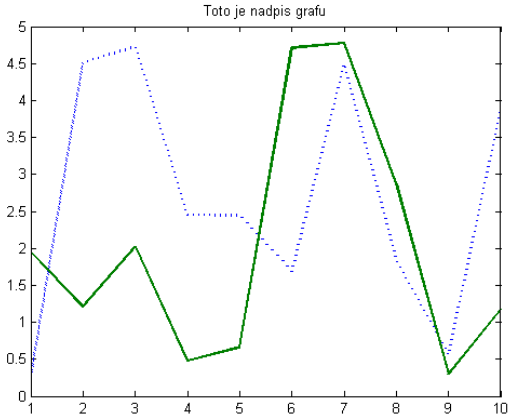
Product Help → MATLAB → User Guide → Graphics → Plots and Plotting Tools



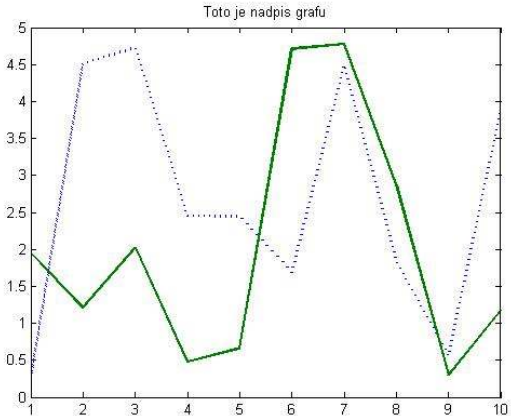
Export obrázku

- File → Save As...
- Bitmapové formáty s bezeztrátovou (ne)kompresí: "*.bmp", "*.png", "*.gif". Pozor na zbytečnou velikost.
- Bitmapový formát "*.jpg" se ztrátovou kompresí: nepoužíváme na grafy či texturovou grafiku! Používáme na reálné fotografie nebo na obrázky, kde je ztráta nepozorovatelná.
- Vektorové formáty "*.eps", "*.pdf", "*.emf" - zachovávají 100% kvalitu čárových grafů při zvětšování/zmenšování. Nejlepší pro tisk. Nehodí se na ukládání bitmap (jsou pak příliš veliké a těžko je zpracovávají některé tiskárny).
- EPS doporučuji pro LaTeX (čárové grafy). Pro konverzi EPS do PDF doporučuji konzolový příkaz `epstopdf`, který je standardní součástí distribucí LaTeXu.

Obrázek ve formátu PNG (7kb)

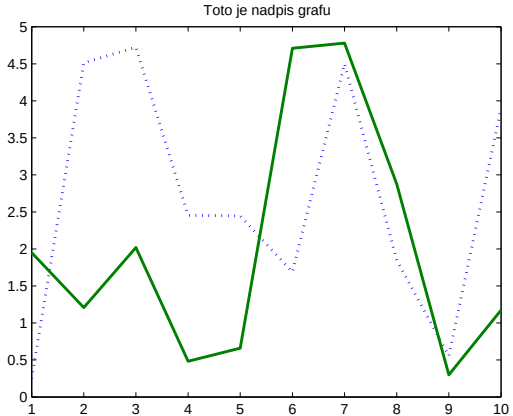


Obrázek ve formátu JPG (20kb)

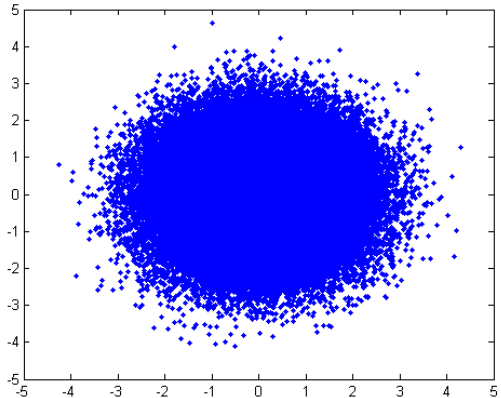


Běda tomu, kdo v bakalářce, diplomce,...

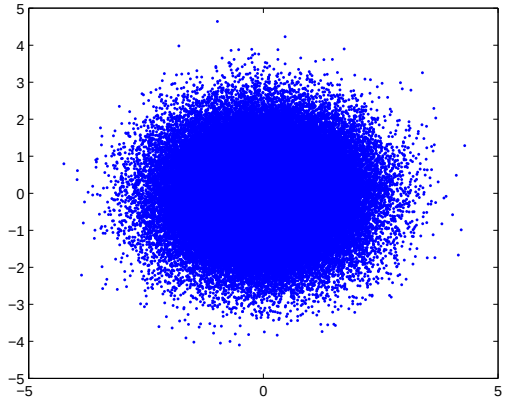
Obrázek ve formátu EPS po konverzi do PDF (4kb)



Obrázek obsahující 100000 bodů ve formátu PNG (12kb)



Stejný obrázek ve formátu PDF (5Mb)!!



Pošlete-li tuto stránku na tiskárnu, hrozí, že ji zahltíte.

Bitmapové obrázky

- Příkaz `image` - hodnoty $0 \ldots 128$, barva bodu podle palety barev
- Příkaz `imagesc` - jako `image` ale vyškáluje rozpětí barev na maximální rozsah
- Příkaz `colormap` mění paletu
- Příkazy pro práci se standardními formáty `imread`, `imwrite`, `imshow` používají speciální 24bit datový typ (RGB).

Ruční kreslení

- Příkazy `line`, `rectangle`, `text`, ...
- Lze takto naprogramovat vlastní specializovaný graf či obrázek.
- Viz `http://www.mathworks.com/matlabcentral`, kde je možné sdílet funkce, skripty až celé balíky příkazů (toolboxy).

Část 2

3D Vizualizace

Příkaz `plot3`

- Vykresluje čárový graf ve 3D, tedy spojnici bodů, kdy každý bod je určený trojicí souřadnic
- Používá se stejně jako `plot`

```
>> x=1:0.1:4*pi;
>> y=sin(x);
>> z=cos(x);
>> h=plot3(x,y,z,'r')
```

- Tedy neslouží k vykreslování 3D ploch nýbrž křivek.

Kreslení 3D ploch

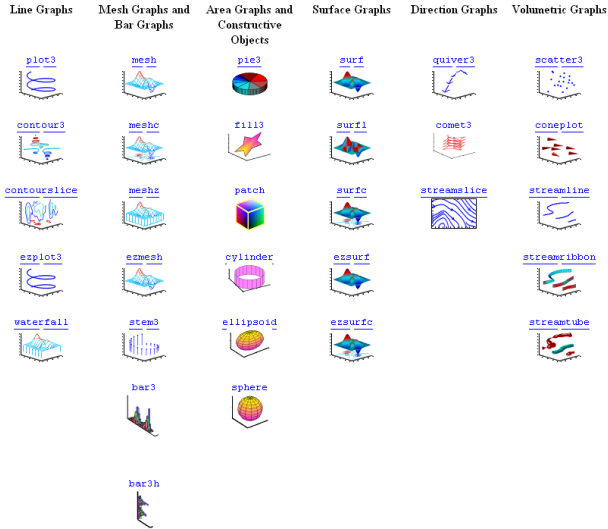
- Příkazy `surf` nebo `mesh`
- `mesh(X,Y,Z)`, kde X, Y a Z jsou matice stejných rozměrů, pospojuje čtyřúhelníky všechny čtveřice sousedních bodů, jejichž souřadnice jsou v X, Y a Z. Vykreslené čtyřúhelníky pak tvoří plochu. Např. vykreslení funkce $e^{-x^2-y^2}$

```
>> x=-2:0.1:2; % nejprve vytvoříme síť bodů
>> y=-2:0.1:2;
>> X=repmat(x,length(x),1);
>> Y=repmat(y',1,length(y));
>> Z=exp(-X.^2-Y.^2); % funkční hodnoty pro celou síť bodů
>> mesh(X,Y,Z)
```

- Síť bodů lze vytvořit pohodlně příkazem `meshgrid`

```
>> [X Y]=meshgrid(-2:0.1:2,-2:0.1:2);
>> Z=exp(-X.^2-Y.^2);
>> mesh(X,Y,Z)
```

Další 3D grafy



Část 3

Animace

Tvorba animací AVI (R2013b)

```
x=1:0.1:4*pi;
y=sin(x);
z=cos(x);

fig=figure;
writerObj = VideoWriter('example.avi');
open(writerObj);

for k=1:length(x)
      h = plot3(x(1:k),y(1:k),z(1:k));
      axis([1 4*pi -1 1 -1 1])
      F = getframe(fig);
      writeVideo(writerObj,F);
end
close(fig)
close(writerObj);
```

FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Děkuji za pozornost

Jaroslav Čmejla
Martin Vrátný
Miroslav Holada
ITE FM TUL

jaroslav.cmejla@tul.cz martin.vratny@tul.cz miroslav.holada@tul.cz