MATLAB
ver. 4

Petr Hora
CDM, ÚT AV ČR
Veleslavínova 11
301 14 Plzeň
hora@cdm.it.cas.cz

Popis prostředí a základní koncepce MATLABu

Příkazové okno a jeho menu

File	Edit	Options	Windows	Help

Grafické okno a jeho menu

File	Edit	Windows	Help

Použití schránky Windows

Vyvolávání běžných dialogových boxů

uisetfont - pro výběr fontu

uigetfile - pro volbu souboru

uiputfile - pro volbu souboru

uisetcolor - pro výběr barvy

Struktura adresářů MATLABu

Adresář/soubor	Význam
`\BIN`	Binární soubory MATLABu
`\TOOLBOX`	Nadstavby MATLABu
`\EXTERN`	Pomocné programy pro sestavení MEX-souborů
`\GHOSTSCR`	GhostScript – program pro převod PS-souborů
`MATLABRC.M`	Inicializační soubor MATLABu
`PRINTOPT.M`	Uživatelsky nastavitelné volby pro tisk

Vyhledávací cesta MATLABu, **path**

Kroky k určení způsobu zpracování textového řetězce:

1. proměnná,

2. vestavěná funkce,

3. soubor s příponou MEX, DLL nebo M v aktuálním adresáři.

Vstup matic

Jeden typ dat – matice (reálná nebo komplexní). Matice může být do MATLABu zavedena několika způsoby:

- napsána jako seznam prvků,
- vygenerována příkazem nebo funkcí,
- vytvořena vsouboru lokálním editorem a načtena,
- načtena zexterního datového souboru nebo aplikace.

Jazyk MATLABu neobsahuje žádný příkaz pro nastavení dimenze nebo typu matice. Potřebnou paměť alokuje MATLAB automaticky až do velikosti využitelné na konkrétním počítači.

Zadání malých matic:

- Jednotlivé prvky matice v řádce oddělit mezerou nebo čárkou.
- Jednotlivé řádky matice oddělit středníkem nebo znakem konce řádky.
- Celý seznam vložit mezi hranaté závorky.

Zadání malé matice

Zadání matice načtením souborů s příponou M

Zadání ze souboru

Prvky matice

Prvky matice mohou být libovolné výrazy MATLABu.

Jednotlivé prvky matice mohou být zpřístupněny indexy uvnitř kulatých závorek.

Velké matice můžete vytvářet pomocí malých matic, na které pohlížíte jako na prvky.

Operace s dvojtečkou (:).

Prvky matice

Příkazy a proměnné

MATLAB je *výrazový* jazyk (výrazy jsou interpretovány a vyhodnoceny).

Výsledkem vyhodnocení výrazu je matice, která je přiřazena do proměnné.

Pokud název proměnné a rovnítko chybí, je vytvořena proměnná **ans**, do které je výsledek uložen.

Jeli posledním znakem příkazu středník (;), je potlačeno zobrazení výsledku.

Pokud jsou před znak konce řádky vloženy tři tečky (...), znamená to, že příkaz pokračuje na následující řádce.

Např.

```
s = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 ...
      - 1/8 + 1/9 - 1/10 + 1/11 - 1/12;
```

Několik příkazů na jediné řádce. Např.

```
a = 3; b = 10; c = a/b;
```

Názvy proměnných a funkcí musí začínat písmenem, za kterým následuje libovolný počet písmen, číslic a podtržítek. MATLAB však rozlišuje pouze prvních 19 znaků jména.

MATLAB rozlišuje malá a velká písmena.

Názvy všech funkcí musí být s malými písmeny.

Funkce může být zastíněna proměnnou, která má stejný název.

Proměnná ans

Použití středníku

Rozlišování velikosti písmen

Zastínění funkce proměnnou

Informace o pracovním prostoru

who
whos

Každý prvek reálné matice vyžaduje 8 bytů paměti.

Stálé proměnné (nejdou smazat)

ans
eps

Nápověda

help
lookfor
who
what
which

Ukončení práce

quit
exit

Uložení pracovního prostoru

save
load

Čísla a aritmetické výrazy

Příklady přípustných čísel jsou

```
3 -99 +0.0001

9.6397238 1.60210E-20 6.02252e23

2i -3.14159i 3e5i
```

POZOR, na omylem vložené mezery před a za symbolem exponentu a imaginární jednotky.

Výrazy můžete sestavovat pomocí obvyklých aritmetických operátorů a pravidel o prioritě operací.

\+ - sčítání

\- - odčítání

* - násobení

/ - dělení zprava

\ - dělení zleva

^ - mocnění

Ke změně pravidel o prioritě se používají standardním způsobem závorky.

MATLAB má vestavěny elementární matematické funkce.

Funkce **pi** vrací π.

Funkce **Inf** zastupuje $nekonečno$ (1/0).

Proměnná **NaN** ($Not\ a\ Number$) je IEEE 'číslo' vytvořené výpočty jako Inf/Inf nebo 0/0.

Komplexní čísla a matice

i nebo **j**

Zadání komplexních čísel

Výstupní formát - vliv pouze na zobrazení matic

format

Funkce

- vnitřní, neboli vestavěné,
- funkce v knihovnách distribuovaných s MATLABem (TOOLBOX),
- funkce vytvořené uživatelem.

Kategorie analytických funkcí využitelných v MATLABu:

- Elementární matematické funkce
- Elementární matice
- Rozklad matic
- Polynomy
- Nelineární rovnice a optimalizace
- Zpracování signálů
- Speciální funkce
- Speciální matice
- Analýza dat
- Řešení diferenciálních rovnic
- Numerická integrace

Operace a manipulace s maticemi

- Transpozice matic
- Sčítání, odčítání, násobení a dělení matic
- Elementární maticové funkce
- Prvkové sčítání, odčítání, násobení a dělení
- Prvkové použití mocnin
- Relační operace
- Logické operace
- Matematické funkce

Transpozice matic

Transpozici matice označuje apostrof (').

Pro komplexní matici:

Z' - vytvoří komplexně sdruženou transponovanou matici,

Z.' - vytvoří nekonjugovanou transpozici.

Sčítání, odčítání, násobení a dělení matic

Znaménka plus (+) a minus (-) označují sčítání a odčítání matic.

Matice musí mít shodné dimenze.

Sčítání a odčítání je také definováno, je-li jeden operand skalár.

Symbol hvězdička $(*)$ označuje násobení matic.

Operace je definována, pokud vnitřní rozměry dvou operandů jsou stejné.

Mezi nejběžnější patří vnitřní součin (skalární).

V MATLABu existují dva symboly pro dělení matic, lomítko (/) a zpětné lomítko (\).
X = A\B je řešením A*X = B
X = A/B je řešením X*A = B

Elementární maticové funkce

poly -charakteristický polynom,

det - determinant,

trace - stopa,

kron - Kroneckerův tenzorový součin.

Prvkové sčítání, odčítání, násobení a dělení

U sčítání a odčítání jsou maticové operace totožné s operacemi prvkovými, takže plus (+) a mínus (-) může být považováno buď za maticovou nebo prvkovou operaci.

Symbol tečka-hvězdička (.*) označuje prvkové násobení.

Symbol tečka-lomítko (./) resp. tečka-zpětné lomítko (.\) označuje prvkové dělení.

Prvkové použití mocnin

Symbol tečka-stříška (.^) označuje prvkové mocniny.

Relační operace

Pro porovnání dvou matic shodných rozměrů existuje šest relačních operátorů.

$<$ - menší než,

$<=$ - menší nebo rovno,

$>$ - větší než,

$>=$ - větší nebo rovno,

$==$ - rovno,

$\sim=$ - nerovno.

MATLAB porovnává dvojice odpovídajících prvků; výsledkem je matice jedniček a nul (jednička znamená splnění podmínky, nula nesplnění podmínky).

find - nachází nenulové prvky v matici, což mohou být datové prvky vyhovující nějaké relační podmínce.

Pro testování hodnot NaN nejsou relační operátory vhodné s ohledem na specifikaci IEEE aritmetiky. Pro testování těchto hodnot slouží funkce **isnan**, která vrací jedničky v místech prvků rovnajících se hodnotě NaN a nuly jinde.

Další užitečnou funkcí je funkce **finite**, která vrací jedničky, pokud $-\infty < x < \infty$.

Logické operace

Pro logické porovnání dvou matic shodných rozměrů existují tři logické operátory.

& - operátor logického součinu ('and'),

| - operátor logického součtu ('or'),

$\sim$ - operátor negace ('not'),

any - vrací jedničku, pokud alespoň jeden z prvků vektoru je nenulový, a jinak nulu.

all - vrací jedničku, pokud všechny prvky vektoru jsou nenulové, a jinak nulu.

Při maticových argumentech pracuje **any** a **all** sloupcově, tj. vrací řádkový vektor s výsledky za každý sloupec. Použijete-li tyto funkce dvakrát, např. any(any(A)), zredukujete tím maticovou podmínku na podmínku skalární.

Další relační a logické funkce jsou:

exist - kontrola existence proměnných a souborů,

isempty - detekce prázdných matic,

isstr - detekce řetězcových proměnných,

isglobal - detekce globálních proměnných,

issparse - detekce řídkých matic.

Matematické funkce

Základní matematické funkce se aplikují na každý prvek matice.

MATLAB obsahuje tyto trigonometrické funkce:

sin - sinus

cos - kosinus

tan - tangens

asin - arkussinus

acos - arkuskosinus

atan - arkustangens

atan2 - čtyř-kvadrantový arkustangens

sinh - hyperbolický sinus

cosh - hyperbolický kosinus

tanh - hyperbolický tangens

asinh - argument hyperbolického sinu

acosh - argument hyperbolického kosinu

atanh - argument hyperbolické tangenty

MATLAB obsahuje tyto základní funkce:

abs - absolutní hodnota nebo modul komplexního čísla

angle - fáze komplexního čísla

sqrt - druhá odmocnina

real - reálná část komplexního čísla

imag - imaginární část komplexního čísla

conj - komplexně sdružené číslo

round - zaokrouhlení k nejbližšímu celému číslu

fix - zaokrouhlení na celé číslo bližší k nule

floor - zaokrouhlení na celé číslo bližší k $-\infty$

ceil - zaokrouhlení na celé číslo bližší k ∞

sign - funkce signum

rem - zbytek po celočíselném dělení

gcd - největší společný dělitel

lcm - nejmenší společný násobek

exp - exponenciální funkce

log - přirozený logaritmus

log10 - dekadický logaritmus

Seznam některých speciálních funkcí v MATLABu:

bessel - Besselova funkce

beta - funkce beta

gamma - funkce gama

rat - racionální aproximace

erf - chybová funkce

erfinv - inverzní chybová funkce

ellipke - eliptický integrál prvního a druhého druhu

ellipj - Jacobiho eliptická funkce

Manipulace s vektory a maticemi

Indexovací schopnosti MATLABu umožňují manipulaci s řádky, sloupci, jednotlivými prvky a submaticemi matic.

Vytváření posloupností:

dvojtečka (:)

linspace - umožní určit počet bodů vektoru namísto volby kroku,

logspace - vytvoří vektor s logaritmickým rozložením.

Indexace a užití logických vektorů při indexaci

- Skalár nebo vektor v kulatých závorkách,
- Logický vektor (vektor obsahující pouze nuly a jedničky), které většinou vzniknou při aplikaci relačních operátorů.

Prázdné matice

Prázdná matice ([ ]) = matici řádu nula

Odstraňování řádek a sloupců matice

Speciální matice

Speciální matice, které se vyskytují v lineární algebře a zpracování signálů, generují následující funkce:

compan - matice přidružená k charakteristickému polynomu

diag - diagonální matice

gallery - testovací matice

hadamard - Hadamardova matice

hankel - Hankelova matice

hilb - Hilbertova matice

invhilb - inverzní Hilbertova matice

kron - Kroneckerův tenzorový součin

magic - magický čtverec

pascal - Pascalův trojúhelník

toeplitz - Töplitzova matice

vander - Vandermondeova matice

Další funkce generují ne tak zajímavé, ale o to užitečnější matice:

zeros - nulová matice

ones - matice jedniček

rand -matice náhodných čísel s rovnoměrným rozdělením

randn - matice náhodných čísel s normálním rozdělením

eye - jednotková matice

linspace - vektor s lineárním rozložením

logspace - vektor s logaritmickým rozložením

Tvorba velkých matic

Velké matice můžete tvořit z malých matic, když je vložíte do hranatých závorek.

Manipulace s maticemi

Několik funkcí slouží k rotaci, překlápění, změně tvaru nebo vyjímání určitých částí z matice.

rot90 - rotace

fliplr - horizontální překlopení

flipud - vertikální překlopení

diag - vyjmutí nebo vytvoření diagonály

tril - dolní trojúhelníková část matice

triu - horní trojúhelníková část matice

reshape - změna tvaru

' nebo **.'** - transponování

Tvorba programů v MATLABu

M-soubory: Skripty a funkce

MATLAB se obvykle používá v příkazovém módu; když zadáte jednořádkový příkaz, MATLAB ho okamžitě provede a zobrazí výsledky.

Kromě toho může MATLAB také spouštět posloupnosti příkazů, které jsou uloženy v souborech. Soubory, které obsahují příkazy MATLABu, se nazývají *M-soubory*, neboť mají příponu M.

M-soubory obsahují posloupnost normálních příkazů MATLABu, které se mohou dále odkazovat na jiné M-soubory. M-soubor může volat rekursivně sám sebe.

Rozlišujeme dva typy M-souborů: *skripty* a *funkce*.

Cyklus FOR

Cyklus **for** slouží pro předem daný počet opakování příkazu nebo skupiny příkazů.

Obecný tvar cyklu **for** je

```
for v = výraz
  příkazy
end
```

Výraz je ve skutečnosti matice, neboť nic jiného v MATLABu neexistuje. Sloupce této matice jsou postupně přiřazovány proměnné **v** a následně jsou provedeny *příkazy*.

Jasněji lze celou záležitost vyjádřit jako

```
E=výraz;
[m,n]=size(E);
for j=1:n
  v=E(:,j);
  příkazy
end
```

Obvykle je výraz ve tvaru m:n nebo m:i:n, což je matice s jednou řádkou, takže sloupce jsou skaláry. V tomto speciálním případě se chová cyklus **for** MATLABu jako cykly FOR a DO v jiných jazycích.

Cyklus WHILE

Cyklus **while** umožňuje opakovat příkaz nebo skupinu příkazů v závislosti na logické podmínce.

Obecný tvar cyklu **while** je

```
while výraz
  příkazy
end
```

Příkazy se opakují tak dlouho, dokud jsou všechny prvky ve výrazu (výraz je matice) nenulové. Výraz je téměř vždy skalárním relačním výrazem, takže nenulové hodnoty odpovídají logické hodnotě TRUE. Pokud výraz není skalár, můžete ho redukovat funkcí **any** nebo **all**.

Příkazy IF a BREAK

Příkaz **if** slouží k větvení algoritmu.

Obecný tvar příkazu **if** je

```
if výraz
  příkazy
[elseif výraz
  příkazy]
[else
  příkazy]
end
```

Skriptové soubory

Když je spuštěn **skript**, MATLAB jednoduše spouští příkazy, které nalezne v souboru.

Příkazy ve *skriptovém souboru* operují globálně s daty v pracovním prostoru.

Skripty jsou užitečné k provedení analýz, řešení problémů nebo konstruování dlouhých posloupností příkazů, které se interaktivně dají dělat jenom těžkopádně a zdlouhavě.

Když spustíte MATLAB, automaticky se spustí skript s názvem **startup.m**. Do něho si můžete zadat fyzikální konstanty, inženýrské konverze nebo cokoli jiného, co chcete mít předdefinováno ve vašem pracovním prostoru.

Funkční soubory

M-soubor, který obsahuje slovo **function** na začátku první řádky, je *funkční soubor*. *Funkce* se liší od *skriptu* v následujícím:

- funkci mohou být předány vstupní parametry,
- ve funkci mohou být definovány proměnné, které jsou lokální,
- funkce může předat výstupní parametry.

Funkční soubory jsou významné pro rozšíření MATLABu, tj. vytvoření nových funkcí MATLABu za použití jazyka MATLABu samotného.

nargin - obsahuje počet vstupních argumentů

nargout - obsahuje počet výstupních argumentů

Tvorba nápovědy pro vaše M-soubory

Nápovědu pro vaše M-soubory vytvoříte zadáním jedné nebo několika komentářových řádek počínaje druhou řádkou souboru.

Funkce **help** zobrazuje první spojitý blok komentářových řádek. Systém nápovědy ignoruje komentářové řádky, které se objeví v souboru později po nějakém proveditelném příkazu nebo prázdné řádce.

Globální proměnné

Obvykle každá funkce MATLABu, definovaná jako M-soubor, má své vlastní lokální proměnné, které jsou oddělené od lokálních proměnných jiných funkcí a od proměnných základního pracovního prostoru.

Pokud však několik funkcí popř. základní prostor deklarují název proměnné jako globální, potom společně sdílení jednu kopii této proměnné.

Globální proměnné se obvykle z důvodu lepší čitelnosti M-souborů píší velkými písmeny.

global ALPHA BETA

Textové řetězce

Textové řetězce se do MATLABu zadávají v apostrofech.

Text je uložen ve vektoru, co znak to prvek.

Funkce **disp** zobrazí text v proměnné, funkce **isstr** detekuje řetězce a funkce **strcmp** řetězce porovnává.

Použitím hranatých závorek můžete textové proměnné spojovat do velkých řetězců.

Čísla se převádí na řetězce funkcemi **sprintf**, **num2str** a **int2str**.

Funkce EVAL

Funkce **eval** pracuje s textovými proměnnými a patří mezi nejvýkonnější (ale také nejzákeřnější) funkce MATLABu.

eval(t) provede vyhodnocení textu uloženého v proměnné **t.**

Jak zvýšit rychlost a ušetřit paměť

Operace s vektory a maticemi, které jsou vestavěné v MATLABu, jsou daleko rychlejší než operace vyžadující kompilaci a interpretaci. To znamená, že chcete-li získat co nejvyšší rychlost zpracování vašich M-souborů, musíte se pokusit vaše algoritmy **vektorizovat**. Kdekoli je to možné, nahraďte cykly **for** a **while** vektorovými či maticovými operacemi.

Pokud část svého kódu vektorizovat nemůžete, máte ještě jednu možnost jak provádění svých cyklů **for** urychlit: provést předběžnou alokaci vektorů, do kterých se v cyklu ukládají výsledky.

Manipulace se soubory

MATLAB	MS-DOS	UNIX	VAX/VMS
dir	**dir**	**ls**	**dir**
type	**type**	**cat**	**type**
delete	**del**	**rm**	**delete**
cd	**chdir**	**cd**	**set default**

Ve většině příkazů můžete používat obvyklé označení disku, název adresáře a *žolíkové* znaky (* a ?).

Příkaz **type** se liší od systémového příkazu **type**. Pokud zadáte za příkazem **type** název souboru bez přípony, MATLAB použije implicitně příponu M. Tato odlišnost je velice výhodná, neboť nejčastější použití příkazu **type** v MATLABu je zobrazení M-souborů na obrazovce.

Spouštění externích programů (! notepad)

Import dat

- Zadání dat jako explicitní seznam prvků.
- Vytvoření dat v M-souboru.
- Načtení dat z textového (ASCII) souboru nebo binárního souboru MATLABu (*.MAT) příkazem **load**.
- Použití nízkoúrovňových funkcí (**fopen**, **fread** a ostatní).
- Vytvoření MEX-souboru pro načtení dat.

Export dat

- Uložení dat do textovém (ASCII) souboru nebo binárního souboru MATLABu (*.MAT) příkazem **save**.
- Použití nízkoúrovňových funkcí (**fopen**, **fwrite** a ostatní).
- Vytvoření MEX-souboru pro uložení dat.

Grafické výstupy v MATLABu

Grafický systém MATLABu je vybudován na základě sady grafických objektů (line, surface, ...), jejichž vzhled lze řídit nastavením parametrů jejich vlastností.

- funkce vyšší úrovně,
- funkce pro práci s grafickými objekty.

Dvourozměrná grafika

plot	vytváří graf užitím lineární stupnice pro obě osy
loglog	vytváří graf užitím logaritmické stupnice pro obě osy
semilogx	vytváří graf užitím logaritmické stupnice pro x-ovou osu a lineární stupnice pro y-ovou osu
semilogy	vytváří graf užitím logaritmické stupnice pro y-ovou osu a lineární stupnice pro x-ovou osu

title	přidá nadpis do grafu (doprostřed nad graf)
xlabel	přidá popis x-ové osy (doprostřed pod osu)
ylabel	přidá popis y-ové osy (doprostřed podél osy)
text	přidá textový řetězec na určenou pozici
gtext	umístí text do grafu na místo vybrané myší
grid	zobrazí síť

symbol	**barva**	**symbol**	**typ čáry**
y	žlutá	**.**	bod
m	fialová	**o**	kroužek
c	tyrkysová	**x**	značka x
r	červená	**+**	plus
g	zelená	*****	hvězdička
b	modrá	-	plná
w	bílá	**:**	tečkovaná
k	černá	**-.**	čerchovaná
		--	čárkovaná

Přidání čar do existujícího grafu a zmrazení měřítka

hold on

hold off

axis(axis)

Imaginární a komplexní data

Jsou-li argumenty funkce **plot** komplexní, tj. mají nenulové imaginární části, jsou tyto imaginární části ignorovány. Pouze v případě, kdy argument funkce **plot** je jediný, tj. **plot(Z)**, kde **Z** je komplexní vektor nebo matice, je vykreslena závislost imaginárních částí prvků **Z** vzhledem k reálným částem.

Kreslení matic

- **plot(MATICE)** - co sloupec, to průběh,
- **plot(vektor,MATICE)** - kreslí řádky nebo sloupce **MATICE** vzhledem k **vektoru** a pro každou čáru použije jinou barvu nebo jiný typ čáry. O tom, zda budou kresleny řádky nebo sloupce rozhoduje počet prvků **vektoru**. Řádková nebo sloupcová orientace je vybrána podle shody počtu prvků řádků nebo sloupců **MATICE** s počtem prvků **vektoru**. Pokud je **MATICE** čtvercová, jsou vykresleny její sloupce.
- **plot(MATICE,vektor)** - kreslí každý řádek nebo sloupec **MATICE** vzhledem k **vektoru**,
- **plot(MATICE_X,MATICE_Y)** - zobrazí sloupce **MATICE_X** vůči sloupcům **MATICE_Y**.

Samozřejmě lze též použít funkci **plot** s několika dvojicemi maticových argumentů

```
plot(X1,Y1,X2,Y2, ...)
```

Speciální funkce pro kreslení grafů

bar	vytváří sloupcový graf
compass	vytváří graf komplexních čísel ve formě šipek vycházejících zpočátku
errorbar	vytváří sloupcový graf chyb
feather	vytváří graf komplexních čísel ve formě šipek vycházejících z ekvidistantně rozložených bodů podél horizontální osy
fplot	vykreslí graf funkce
hist	vytváří histogram
polar	vytváří graf v polárních souřadnicích
quiver	vytváří graf gradientu nebo jiného vektorového pole
rose	vytváří úhlový histogram
stairs	vytváří graf ve tvaru schodů
fill	vykreslí mnohoúhelník a vyplní jej

Vykreslení matematických funkcí

fplot - přizpůsobuje periodu vzorkování funkce průběhu funkce

Třírozměrná grafika

plot3	kreslí čáry a body v prostoru
contour **contour3**	vytváří vrstevnice grafu 2-D vytváří vrstevnice grafu 3-D
pcolor	vytvoří obdélníkové pole buněk, kterým jsou přiřazeny barvy podle velikosti prvků v matici
image	zobrazí matici jako obraz, každý prvek matice určuje barvu políčka v obraze. Prvky matice jsou užity jako indexy aktuální mapy barev k určení barvy
mesh **meshc** **meshz**	vytváří síť v prostoru, prvky matice zobrazí jako výšky nad základnou
surf **surfc** **surfl**	jako mesh, ale vytvoří plochu složenou ze čtyřúhelníků, jejichž vrcholy tvoří prvky zadané matice. Jednotlivé čtyřúhelníky jsou vybarveny odpovídající barvou.
fill3	vytvoří mnohoúhelník v prostoru a vyplní jej

Kromě funkcí uvedených u 2-D grafiky používá MATLAB k popisu os navíc funkce:

zlabel	vytvoří popis z-ové osy
clabel	přidá popis vrstevnic

MATLAB dovoluje určit bod pohledu na graf. Následující dvě funkce umožňují jednoduchým způsobem definovat bod pohledu

view	nastaví aktuální bod pohledu pomocí azimutu a elevace nebo pomocí transformační matice
viewmtx	vypočte transformační matici čtvrtého řádu jak pro pravoúhlou, tak pro perspektivní transformaci

Kreslení čar

- **plot3(x,y,z)** -vytvoří v prostoru čáru procházející body, jejichž souřadnice jsou prvky vektorů **x**, **y** a **z**, a provede 2-D projekci této čáry na obrazovku.

- **plot3(X, Y, Z)** - vykreslí čáry získané ze sloupců matic **X**, **Y** a **Z**.

Funkce meshgrid

Ke znázornění funkce dvou proměnných $z = f(x, y)$ musíme nejprve vytvořit matice **X** a **Y** (definiční oblast funkce). Funkce **meshgrid** transformuje oblast určenou dvěma vektory **x** a **y** na matice **X** a **Y**, které pak využijeme k vyhodnocení a znázornění funkce dvou proměnných. Řádky matice **X** jsou kopiemi vektoru **x**, sloupce matice **Y** jsou kopiemi vektoru **y**.

Kreslení vrstevnic

contour a **contour3**

Funkce pcolor

Název funkce **pcolor** je zkratkou slova pseudocolor. Každý bod matice je transformován podle rozsahu aktuální mapy barev (**colormap**); maximální hodnota matice se nastaví na maximální hodnotu indexu mapy barev.

Objekty image

MATLAB vytváří obraz (image) tím, že pro každý prvek v matici vyhledá přímo hodnotu barvy v mapě barev. Pro objekty image je příznačné, že mají své vlastní mapy barev.

Získání správného poměru vzhledu

Ve většině případů je pro vytvoření objektu image důležité použít vhodný poměr vzhledu obrazu, a tím předejít případnému zkreslení. Příkaz

```
axis('equal')
```

zajistí, že jsou objekty image zobrazovány ve správném poměru bez ohledu na změnu velikosti okna figure. Nechceme-li, aby byly osy a jejich popis viditelné, můžeme viditelnost os a jejich popisů vypnout příkazem **axis('off')**.

Porovnání objektů image a grafů, kreslených funkcí pcolor

- funkce **image** slouží pro zobrazování fotografií, obrazů apod.
- funkce **pcolor** slouží pro zobrazování abstraktních matematických objektů

Následující seznam uvádí rozdíly mezi těmito dvěma funkcemi pro zobrazení matice **A** typu (m,n).

- **image(A)** vytvoří pole o m*n buňkách, zatímco **pcolor(A)** vytváří m*n čar sítě, a tudíž pole o počtu pouze (m-1)*(n-1) buněk.

- **image(A)** používá číslování os pomocí 'ij', **pcolor(A)** implicitně používá číslování os 'xy'.

- **image(A)** užívá vždy obdélníkovou rovnoměrnou síť, zatímco **pcolor(X,Y,A)** může vytvořit parametrickou síť v jiném souřadnicovém systému.

- **image(A)** tvoří 2-D objekty, na které se můžeme dívat pouze ze standardního bodu pohledu (azimut=0°, elevace=90°), **pcolor(A)** tvoří plochu, na kterou se můžeme dívat z libovolného úhlu.

- funkce **image** používá prvky matice **A** pro vyhledání hodnot barev přímo v mapě barev. Tyto prvky jsou celá čísla v rozsahu od 1 do počtu prvků v mapě barev (length(colormap)). Vstupní matice ve funkci **pcolor** je transformována podle rozsahu barevné osy (**caxis**), funkce **image** není rozsahem **caxis** ovlivnitelná**.**

- **image** zmrazí osy tak, aby obraz úplně vyplnil celý rozsah os, **pcolor** ne.

Kreslení ploch

mesh a **surf**

shading

Mají-li funkce **mesh** a **surf** jako argument jedinou matici **Z**, pak tato matice definuje jak výšku, tak barvu plochy.

Následující příkazy se dvěma maticovými argumenty

mesh(Z, C)

surf(Z, C)

specifikují barvu použitou jako druhý argument. Hodnoty prvků matice **C** jsou transformovány a použity jako indexy v aktuální mapě barev (podobně jako u funkce **pcolor(C)**).

Vyříznutí části plochy hodnotami NaN

Jestliže data plochy (nebo čáry) obsahují prvky **NaN**, nejsou tyto prvky zobrazeny.

Parametrické plochy

Funkce **mesh**, **surf** a **pcolor** mohou mít ještě dalších dva vektorové nebo maticové argumenty popisující plochu určenou k zobrazení. Je-li **Z** matice typu (m,n), **x** vektor délky **n** a **y** vektor délky m, potom

```
mesh(x, y, Z, C)
```

popisuje drátový model plochy, jejíž vrcholy mají barvu **C(i,j)** a jsou umístěny v bodech

(x(j), y(i), Z(i,j)). **x** odpovídá sloupcům matice **Z** a **y** jejím řádkům.

Obecněji, jsou-li **X**, **Y**, **Z** a **C** matice téhož typu, pak

```
mesh(X, Y, Z, C)
```

generuje drátový model plochy, jejíž vrcholy mají barvu **C(i,j)** a jsou umístěny v bodech (X(i,j), Y(i,j), Z(i,j)).

Totéž platí pro funkce **surf(X, Y, Z, C)**, atd.

Obměny funkcí surf a mesh

surfc

meshz

surfl

Obecné grafické funkce

Funkce view

azimut a elevace

Ovládání os funkcí axis

axis

```
axis([xmin xmax ymin ymax])
axis([xmin xmax ymin ymax zmin zmax])
axis('auto')
v = axis
axis(axis)
axis('ij')
axis('xy')
axis('square') a axis('equal')
axis('on') a axis('off')
```

Odstranění skrytých čar

hidden

Funkce subplot

subplot

Funkce figure

Funkce **figure** bez argumentů otevře nové grafické okno.

figure(N) zaktualizuje **N**-té grafické okno, grafické příkazy budou nyní zobrazovat data do tohoto okna.

Animace (movie)

moviein, **getframe** a **movie**

Grafický vstup

ginput

Tisk grafických oken

print

Ghostscript

Mapy barev a ovládání barev

MATLAB definuje mapu barev ve tvaru matice o třech sloupcích. Každá řádka matice definuje jednotlivou barvu na základě tří hodnot v rozsahu od 0 do 1 (RGB hodnoty). RGB hodnoty mají význam intenzity složek červené (R), zelené (G) a modré (B).

Mapy barev mohou být zadány přímo ve tvaru matice nebo mohou být generovány operacemi MATLABu (**hsv**, **cool**, **pink**, **cooper**, **flag**).

Mapu barev používají funkce **mesh**, **surf**, **pcolor** a **image** a funkce z nich odvozené.

V MATLABu je k dispozici deset map barev:

hsv	Mapa barev hsv (Hue-saturation-value)
gray	Lineární šedá mapa barev
white	Bílá mapa barev
hot	Černo-červeno-žluto-bílá mapa barev
cool	Mapa barev s odstíny tyrkysové a fialové
bone	Šedá mapa barev se zabarvením do modra
copper	Mapa barev s lineárními tóny mědi
pink	Mapa barev s pastelovými odstíny růžové

prism	Mapa barev prism
jet	Varianta mapy barev hsv (přechod od fialové přes modrou a žlutou k červené; jako v kartografii)
flag	Mapa barev tvořená střídavě červenou, bílou, modrou a černou

Ovládání barevné osy

caxis([cmin cmax])

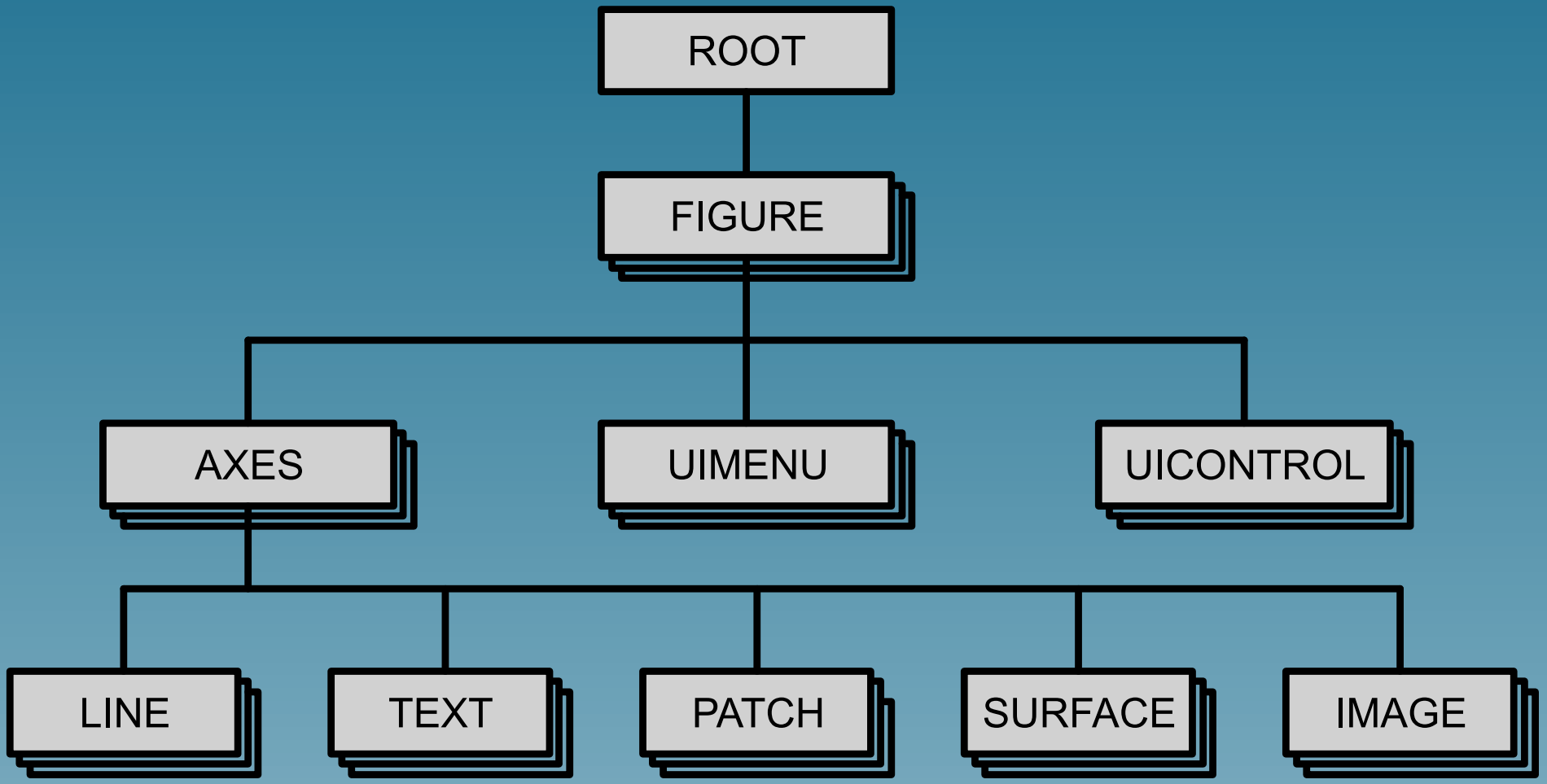
Funkci **caxis** můžeme použít i k dosažení následujících dvou efektů:

- Nastavíme-li **cmin**, popř. **cmax** na hodnoty, které jsou menší než rozsah dat plochy, potom data menší než **cmin** a větší než **cmax** se nebudou transformovat do barev.
  POZOR! Od verze MATLABu 4.2 se data menší než **cmin** resp. větší než **cmax** budou transformovat do krajních hodnot mapy barev, tj. **cmin**, resp. **cmax**.

- Nastavíme-li **cmin**, popř. **cmax** na hodnoty, které jsou větší než rozsah dat plochy, MATLAB transformuje mapu barev do většího rozsahu, jako kdyby data byla rozprostřena od **cmin** až do **cmax**. V důsledku toho jsou aktuální data zobrazena použitím pouze části mapy barev.

Objektová grafika

MATLAB definuje grafické objekty jako základní grafické jednotky svého grafického systému a organizuje je do stromově strukturované hierarchie. Tyto objekty zahrnují:



Identifikátory objektů (handle)

Každý samostatný grafický objekt má svůj vlastní identifikátor, tzv. handle, který je tomuto objektu přiřazen při jeho vytvoření. Některé grafy, např. vrstevnice, jsou složeny z několika objektů a každý z nich má svůj vlastní identifikátor, tj. každá vrstevnice má svůj identifikátor.

Identifikátor objektu root je vždy nulový. Identifikátor objektu figure je celé kladné číslo, které je implicitně zobrazeno v názvu grafického okna. Identifikátory ostatních objektů jsou reálná čísla, která obsahují informace používané MATLABem.

K jednoduchému přístupu k identifikátorům objektů definuje MATLAB následující funkce:

gcf - vrací identifikátor aktuálního objektu figure

gca - vrací identifikátor aktuálního objektu axes

Tyto funkce můžeme použít jako vstupní argumenty pro jiné funkce, které požadují identifikátor objektů figure nebo axes.

Libovolný objekt lze zrušit funkcí **delete** s použitím identifikátoru tohoto objektu jako argumentu. Např. můžeme vymazat aktuální osy, a tím i všechny jejich děti, příkazem

```
delete(gca)
```

Všechny funkce MATLABu, které vytvářejí objekty, vrací identifikátory (nebo vektor identifikátorů) vytvořených objektů. A to jak funkce vyšší úrovně jako **surfc** (generuje jak plochu, tak čáry), tak i funkce nižší úrovně, které generují pouze jeden objekt, např. funkce **surface**.

Funkce vytvářející objekty

Všechny objekty mohou být generovány funkcemi, které mají stejné jméno jako jimi generovaný objekt (funkce **text** vytvoří objekt text, funkce **figure** vytvoří objekty figure atd.).

Vlastnosti objektů

Všechny objekty mají vlastnosti, které rozhodují o tom, jak budou tyto objekty zobrazeny. Tyto vlastnosti zahrnují jak obecné informace (typ objektu, jeho rodiče a děti, zda je nebo není objekt viditelný), tak i informace jedinečné pro jednotlivý typ objektu (např. rozsah x-ové osy objektu axes).

Tvořený grafický objekt je inicializován množinou implicitních hodnot vlastností.

Aktuální hodnoty všech vlastností můžeme získat a většinu z nich specifikovat. Některé vlastnosti jsou nastaveny MATLABem a jsou určeny pouze ke čtení. Hodnoty vlastností se aplikují jednoznačně na konkrétní objekt, nastavení hodnoty pro jeden objekt neovlivňuje hodnotu u ostatních objektů téhož typu.

Poznámka o názvech vlastností

Podle zvyklostí dává MATLAB u názvů vlastností objektů vždy první písmeno každého slova velké, např. **LineStyle** nebo **XMinorTickMode**. Tento způsob je vhodný pro jejich lepší čtení. MATLAB nekontroluje v názvech vlastností velikost písmen, proto lze pro správnou identifikaci názvu vlastnosti použít písmena libovolné velikosti. Lze dokonce použít i zkrácených názvů, ale tak, aby tato zkratka jednoznačně určovala danou vlastnost.

POZOR!!! Název vlastnosti uvedený v apostrofech nesmí obsahovat žádné mezery.

Nastavení vlastností objektů

- určit vlastnost objektu v době, kdy voláme funkci, která příslušný grafický objekt generuje

- nastavit hodnotu vlastnosti po vytvoření objektu pomocí funkce **set**

Funkce set a get

Vlastnosti objektu můžeme nastavit také až po jeho vytvoření. K tomu využijeme identifikátory, které vracejí vytvářející funkce.

Funkce **set** umožňuje nastavit vlastnosti objektu pomocí identifikátoru objektu a dvojice PropertyName/PropertyValue.

Chceme-li znát hodnoty nastavených vlastností, použijeme funkci **get**.

Jestliže máme objekt jednou již identifikovaný, můžeme změnit libovolnou jeho vlastnost bez „roztržení“ celého grafu. Tuto techniku můžeme použít pro přístup k jednotlivým grafickým objektům, a tím modifikovat chování funkcí MATLABu, jak ukazuje následující příklad.

Vlastnosti objektů

Transformace ploch typu texture

Texture je technika transformace 2-D obrazu na 3-D plochu, kdy se barevná data přizpůsobí tvaru 3-D plochy. Tím lze na 3-D plochu aplikovat různé textury, jako např. povrchy materiálů, bez složitého 3-D geometrického modelování výsledné plochy s těmito rysy. Barevná data mohou také být v podobě libovolného obrazu nebo fotografie.

MATLAB převede barevná data textury do vlastnosti **CData** objektu surface. Zatímco barva objektu surface je vždy určena hodnotami obsaženými v jeho vlastnosti **CData**, je transformace texture odlišná v tom, že rozměr pole **CData** může být u tohoto objektu surface jiný než rozměr jeho pole **ZData**. Tím je umožněna aplikace obrazu libovolné velikosti na jakoukoliv plochu. MATLAB interpoluje barevná data textury tak, aby pokryla úplnou plochu objektu surface.

Textury

Poloautomatické meze os

Pokud chceme použít režim poloautomatických mezí os, definujeme jednu mez rozsahu souřadnic nebo barevné osy (vlastnost **XLim**, **YLim**, **ZLim** nebo **CLim**) a druhou mez nastavíme do automatického režimu zadáním hodnoty plus nebo minus **inf**.

Poloautomatické meze

Logaritmická stupnice

MATLAB vykresluje v logaritmické stupnici také záporná data. Nemůže ale zobrazit záporná a kladná data současně do jediných os. Obsahují-li data kladná i záporná čísla, jsou záporná čísla ignorována a dolní mez osy je nastavena automaticky tak, aby byla znázorněna nejmenší kladná hodnota dat. Zápornou logaritmickou osu vytváří MATLAB pouze tehdy, pokud jsou všechna kreslená data záporná.

Logaritmická stupnice os

Funkce grafického okna závislé na akci tlačítka myši

Funkce grafického objektu závislé na akci tlačítka myši

Implicitní hodnoty vlastností

Všechny vlastnosti objektů mají své implicitní hodnoty vestavěné v MATLABu (factory settings). Navíc ale můžeme definovat své vlastní implicitní hodnoty v libovolném bodu hierarchie objektů.

Hledání implicitních hodnot začíná u aktuálního objektu a pokračuje přes předky do té doby, dokud není nalezena implicitní hodnota definovaná uživatelem nebo dokud není dosaženo vestavěných implicitních hodnot. Proto je hledání implicitních hodnot vždy úspěšné.

Implicitní hodnoty můžeme nastavit pomocí řetězce začínajícího slovem **Default**, za kterým následuje typ objektu a nakonec vlastnost objektu. Např. nastavení implicitní barvy čáry na bílou barvu v úrovni aktuálního objektu figure provede příkaz

```
set(gcf, 'DefaultLineColor', 'w')
```

Bod hierarchie, ve kterém definujeme implicitní hodnotu, určuje, které objekty tuto hodnotu použijí.

Zadáme-li hodnotu **factory**, nastaví se vlastnost na svou hodnotu vestavěnou v MATLABu.

Řetězcem **remove** můžeme odstranit implicitní hodnoty nastavené uživatelem.

Implicitní hodnoty jsou respektovány **pouze** grafickými funkcemi nejnižší úrovně (figure, axes, line, text, surface, patch a image) !!!

Poznámky:

MATLAB má vestavěn implicitní font písma Helvetica. Pokud tento font není na našem počítači, nebude otočení textu provedeno. V tomto případě je vhodné nastavit pro text na úrovni figure implicitní název fontu (z dostupných fontů) pomocí vlastnosti **DefaultTextFontName**.

Chceme-li mít v MATLABu definované určité hodnoty vždy, je vhodné je definovat v m-souboru **startup.m**.

Užitečné funkce

MATLAB obsahuje některé funkce, které zjednodušují proces získávání a nastavování hodnot vlastností v aktuálním objektu. Ve všech případech provádějí funkce **get** a **set** tutéž činnost, ale musíme jim určit identifikátory cílových objektů. Následující seznam uvádí přehledně tyto funkce i vlastnosti, které jimi jsou ovlivňovány, popř. které jsou jimi získány.

Funkce	**Typ objektu**	**Nastavené nebo vrácené vlastnosti**
axis	*axes*	XLim, YLim, ZLim, XLimMode, YLimMode, ZLimMode, View, YDir, Position, Visibility, AspectRatio
caxis	*axes*	CLim, CLimMode
cla	*axes*	odstraní děti
clf	*figure*	odstraní děti
colormap	*figure*	ColorMap
gca	*figure*	CurrentAxis
gcf	*root*	CurrentFigure
grid	*axes*	XGrid, YGrid, ZGrid

hold	***axes***	NextPlot
	figure	NextPlot
orient	***figure***	PaperOrientation, PaperPosition
reset	***axes, figure***	vše kromě Position
subplot	***figure***	NextPlot, CurrentAxis
	axes	Position
view	***axes***	View, XForm