Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázku knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukázka má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázku jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umisťováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz

2 Základy práce v prostředí MATLAB

Jak bylo zmíněno výše, pro efektivní práci je třeba mít spuštěný MATLAB. Nebudeme se věnovat detailnímu popisu instalace, požadavků na vybavení počítače atd. Pravdou je, že na pomalém počítači s malou operační pamětí bude práce vyžadovat pevné nervy. Budeme předpokládat, že pracujete na legálně zakoupené či přidělené licenci.

Spuštění MATLABu:

Nemáte-li na pracovní ploše počítače (předpokládáme operační systém MS Windows 95 a vyšší) příslušnou ikonu, pak spustitelným souborem je soubor **MATLAB.exe**. Je umístěn ve složce (adresáři) *...../MATLAB6p1/bin/win32*. Je však třeba hned poznamenat, že to platí pro aktuální verzi 6.6. a v případě, že při instalaci nebyla změněna cílová složka *...../MATLAB6p1*. V jiných verzích, zejména nižších, je cesta k souboru MATLAB.exe pozměněna, soubor sám je však stejného jména. Poznáte jej podle typického grafického symbolu oranžové barvy.

2.1 Stručný průvodce pracovní plochou

Než se pustíme do vlastní práce, poznamenejme, že současnou aktuální verzí MATLABu je verze 6.6. Ta bude předmětem popisu ve skriptu. Verze 6 je oproti předchozím typická také poněkud modifikovaným pracovním prostředím. To je problémem všech systémů podobného typu. Občas je uživatel postaven před nutnost přizpůsobit se sice efektivnějšímu a výkonnějšímu prostředí, ale za cenu změn svých návyků.

Z důvodů výroby knihy jsme nuceni používat jen černobílé obrázky, resp. obrázky s odstíny šedi. Bohužel, někdy je to právě barva či barevnost, která odlišuje různé varianty příkazů a povelů a která činí vizualizaci dat efektnější. Sami jsme při práci používali operační systémy MS Windows XP.

Vlastní popis pracovní plochy:

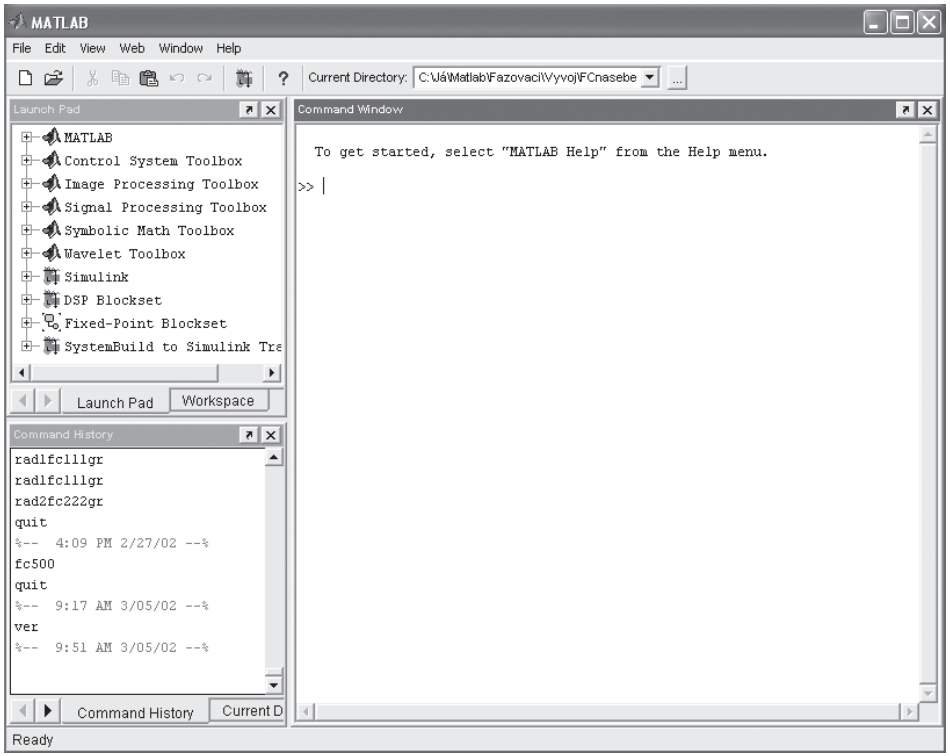
Po spuštění MATLABu se nám otevře pracovní plocha nebo prostředí. Jak je vidět z *obr. 2.1*, sestává celkem ze tří otevřených oken. Jedno okno je podstatně větší, je v horní části nazváno *Command Window*. Další dvě menší okna v levé části jsou nazvána *Launch Pad* a *Command History*. Všimněte si, že tato menší okna mohou

být přepnuta na jiná. Okno *Launch Pad* může být přepnuto na *Workspace* a okno *Command History* na *Current Directory*. Přepínání se děje kliknutím na záložky ve spodních částech těchto oken. Vysvětleme si stručně funkci těchto oken.

Funkce oken pracovní plochy:

Okno Command Window – je hlavní a nejdůležitější částí pracovní plochy. Sem zapisuje uživatel své příkazy a povely, zde je vidět odezva MATLABu a zde se zobrazují systémová hlášení. Toto okno si můžeme představit jako chytrou kalkulačku s mnoha funkcemi a možnostmi.

Okno Launch Pad slouží k rychlému spuštění řady nabídek. Jde o interaktivní *help*, demonstrační samoběžící výukové programy *demos*, podporu tvorby grafických prostředků atd. Stačí si je vybrat a poklepat na ně myší.



Obr. 2.1 *Pracovní plocha po spuštění MATLABu verze 6.6.*

Okno Command History je velmi užitečným oknem. Zobrazují se zde všechny příkazy a povely, zapsané a potvrzené uživatelem v hlavním okně *Command Window*. Je-li potřeba již jednou zapsaný a potvrzený příkaz znovu použít, stačí jej v tomto okně nalistovat a poklepáním znovu aktivovat či jej přetáhnout myší do hlavního okna. Totéž je však možné v hlavním okně pomocí kurzorových šipek nahoru a dolů.

Okno Workspace bude při prvním spuštění prázdné. Při používání proměnných v hlavním okně bude v tomto okénku přehled všech vámi použitých proměnných. Poklepete-li na symbol některé proměnné, zobrazí se detailní informace o ní (rozměr, struktura apod.). To je užitečné zejména při používání většího množství proměnných pro udržení přehledu.

Okno Current Directory ukazuje seznam souborů v aktuální (current) složce (adresáři). Poklepáním myší na některý ze zobrazených souborů jej otevřete ve vestavěném editoru apod.

Current Directory (aktuální složka) je zobrazena těsně nad hlavním oknem (Command Window). Je možné ji změnit podle potřeby. Implicitní nastavení je do složky *...MATLAB6p1/Work*, kam MATLAB předpokládá umístění výsledků vaší práce.

Ukončit práci se systémem MATLAB lze zápisem **quit** s následným **ENTER** nebo myší kliknutím na křížek vpravo nahoře (standardní ukončení v operačním systému Windows).

Funkce oken plně oceníte později, až budete zkušenějšími uživateli. Pak vám budou velmi šetřit čas.

Menu nad okny pracovní plochy:

Nad výše popsanými okny pracovní plochy jsou dvě řady nabídek. Ta níže uvedená je ve formě tlačítek. Slouží k vytvoření nového tzv. *m-souboru*, otevření již existujícího, ke spuštění simulačního systému Simulink a k dalším funkcím, potřebným k editaci.

Nejvýše jsou umístěny rozbalovací nabídky – menu. Můžete si je prohlédnout, věnovat se jim zatím nebudeme, neboť to není nutné. Slouží např. k nastavení cesty k *m-souborům* či jiným souborům, k detailnímu nastavení pracovní plochy, vyvolání helpu atd.

Co jsou to m-soubory:

Tzv. *m-soubory*, zmíněné výše, jsou textové soubory s příponou *.m. Slouží k zápisu posloupnosti příkazů MATLABu a jejich uložení např. na disk. Jsou tedy zdrojovým kódem, který umí MATLAB vykonat. Pokud si používané příkazy neuložíte, při zavření MATLABu o ně přijdete. Jde tedy o soubory, které zatím nevyužijete, ale které jsou nezbytné např. při programování. Více informací prozatím není třeba. Pro ilustraci, jenom popisu nastavení pracovní plochy věnují autoři MATLABu v manuálu několik desítek stran.

2.2 Základní operace s čísly

2.2.1 Elementární operace s reálnými čísly

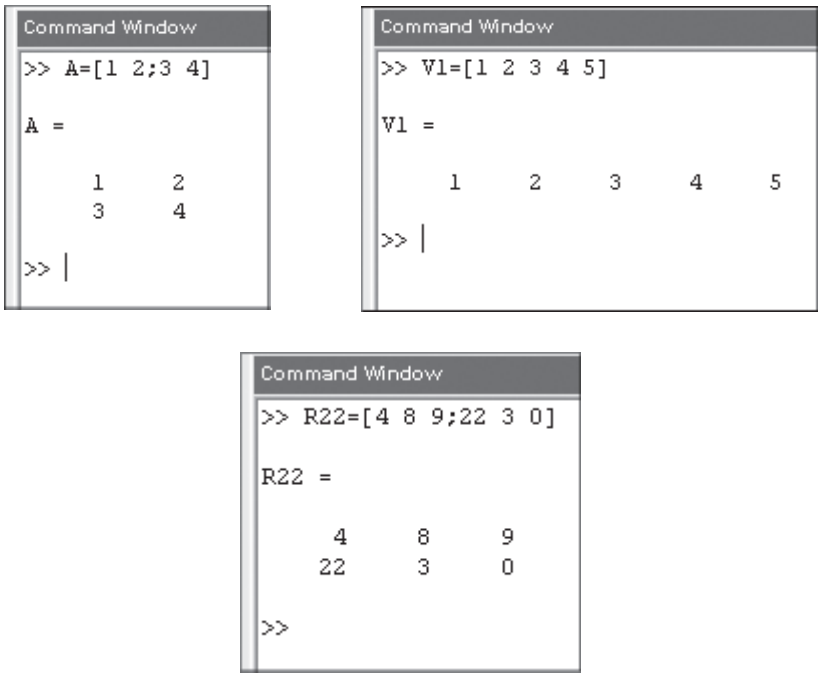
MATLAB je svou podstatou nástroj pro práci s čísly (pomiňme možnost dokoupení toolboxu pro symbolickou matematiku). Začněme prací s reálnými čísly.

3 Práce s maticemi a řešení soustavy rovnic

Programový systém MATLAB je orientován maticově. Matice je základním objektem s mnoha možnými podobami a tomu je přizpůsobena struktura celého systému. Kapitola 3 proto tvoří jádro knihy. Ukážeme si v ní zejména možnosti základní definice matic a práce s nimi.

3.1 Vytvoření matic a vektorů

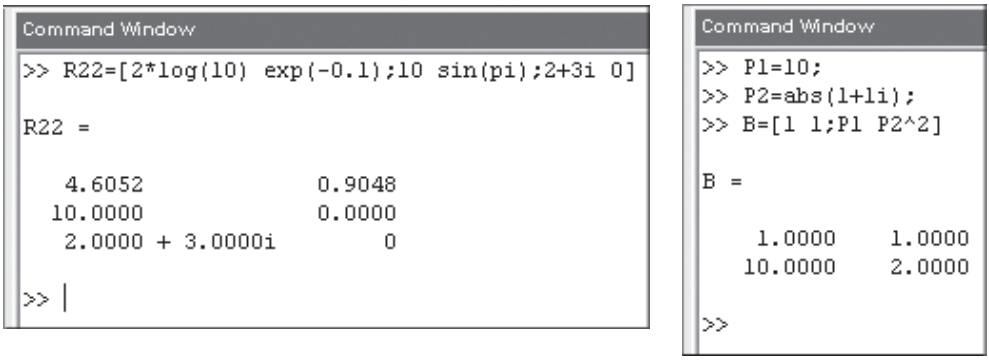
Možností, jak vytvořit matici či vektor, je v MATLABu několik. Zapište v hlavním okně MATLABu zápis: **A=[1 2;3 4]** a potvrďte pomocí **ENTER**. Zopakujme, že použijete-li středníku za výrazem, matice se nadefinuje, ale nevypíše. Zapište dále ostatní výrazy podle *obr. 3.1*.



Obr. 3.1 *Základní způsob vytvoření matice a vektoru*

Z *obr. 3.1* je zřejmé, že matice se zapisují po řádcích, které jsou odděleny středníkem. Prvky každého řádku jsou odděleny mezerou nebo čárkou (méně často). Je-li matice jen jednořádková, nazýváme ji řádkovým vektorem, v našem případě *V1*. Přitom počet řádků, resp. sloupců matice není v principu nijak omezen.

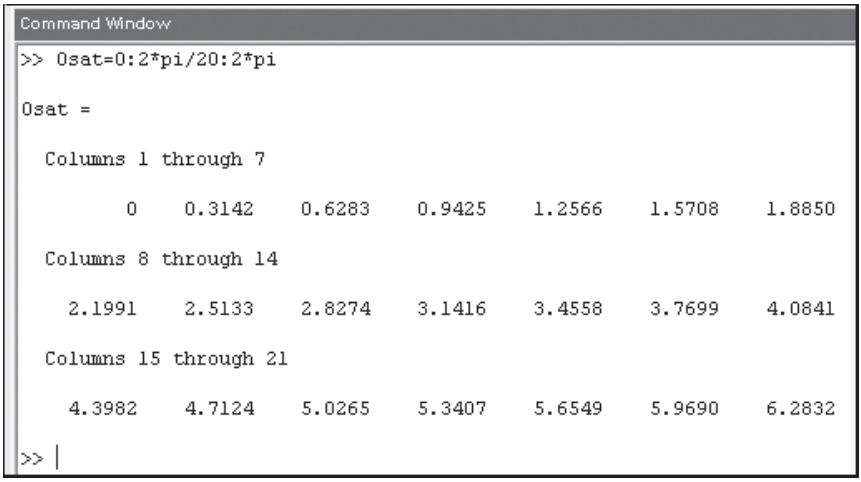
Pro vytvoření matic a vektorů můžete však použít také libovolné matematické výrazy a rovněž proměnné, jak ukazuje *obr. 3.2*.



Obr. 3.2 *Využití matematických výrazů a proměnných k definici matic a vektorů*

Tvorba dlouhých vektorů a matic:

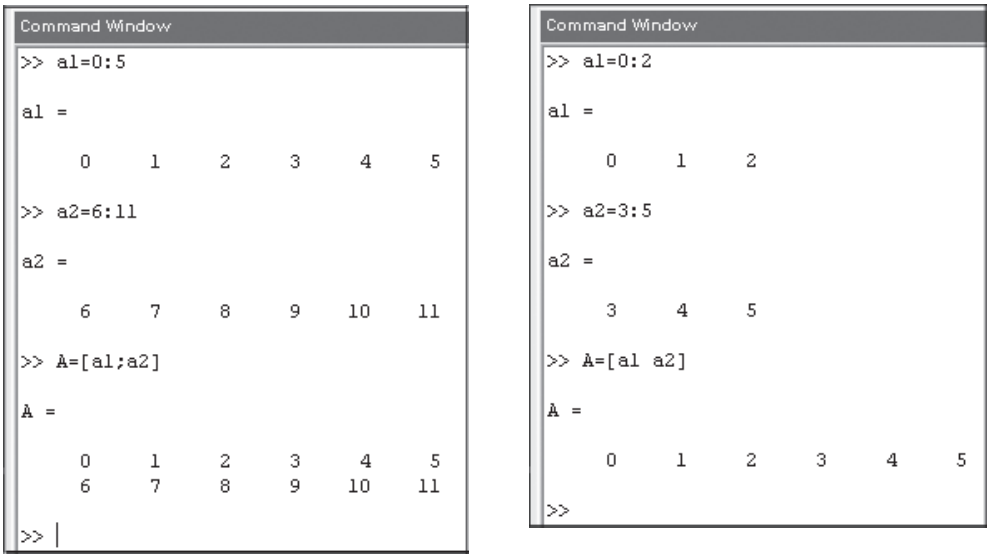
Popsaný způsob tvorby matic, resp. vektorů, je doporučen pro nepříliš velký počet řádků či sloupců. Při potřebě generovat dlouhé vektory, např. s mnoha desítkami či stovkami prvků, je třeba využít jiný způsob. Předpokládejme, že potřebujete vytvořit časovou osu pro nějaký graf. Půjde tedy o vektor, nazvaný např. **Osat**. Zapište: **Osat=0:2*pi/20:2*pi** a potvrďte pomocí **ENTER**. Odezvu ukazuje *obr. 3.3*. Čísla v zápisu vektoru představují startovací prvek, krok a cílový prvek vektoru **Osat** a jsou odděleny dvojtečkou.



Obr. 3.3 *Tvorba „dlouhého“ řádkového vektoru*

Obr. 3.3 ukazuje výpis všech 21 prvků celého vektoru **Osat**. Je třeba poznamenat, že u definic velmi dlouhých vektorů či matic je vhodné používat středník na konci výrazu. Jinak totiž dojde k často dlouhému vypisování čísel, což jen zdržuje od práce. Pokud se to stane, stačí stisknout klávesy **Ctrl-C** a zdlouhavý výpis je zastaven.

Dlouhé vektory, vytvořené popsaným způsobem, lze jednoduše sdružovat do matic. Vyzkoušejte si to podle *obr. 3.4*.



Obr. 3.4 *Sdružení dvou vektorů do matice*

Uvedené možnosti tvorby matic a vektorů představují základní způsob jejich definice. MATLAB má i jiné způsoby, jak naplnit matice jejich číselnými, příp. jinými obsahy. Pro základní seznámení se systémem však tyto postupy postačí.

Stručné shrnutí:

- relativně malé matice a vektory definujeme zápisem na dialogovém řádku hlavního okna; matice jsou zapisovány po řádcích, oddělených středníky, prvky jednotlivých řádků se oddělují mezerami nebo čárkami, např. **Mat=[1 2 3;4 5 6;7 8 9]**, **Vek=[11 12 13 14]**,
- k definici delších vektorů je vhodné použít specifický způsob s využitím dvojteček, např. **Osax=0:4*pi/100:4*pi**,
- při zápisu delších matic či vektorů je vhodné používat středníky, aby nedocházelo k zdlouhavým výpisům prvků, stane-li se to, pomocí **Ctrl-C** lze výpis přerušit,
- vektory lze jednoduše sdružovat do matic.

4 Základní použití 2D grafiky

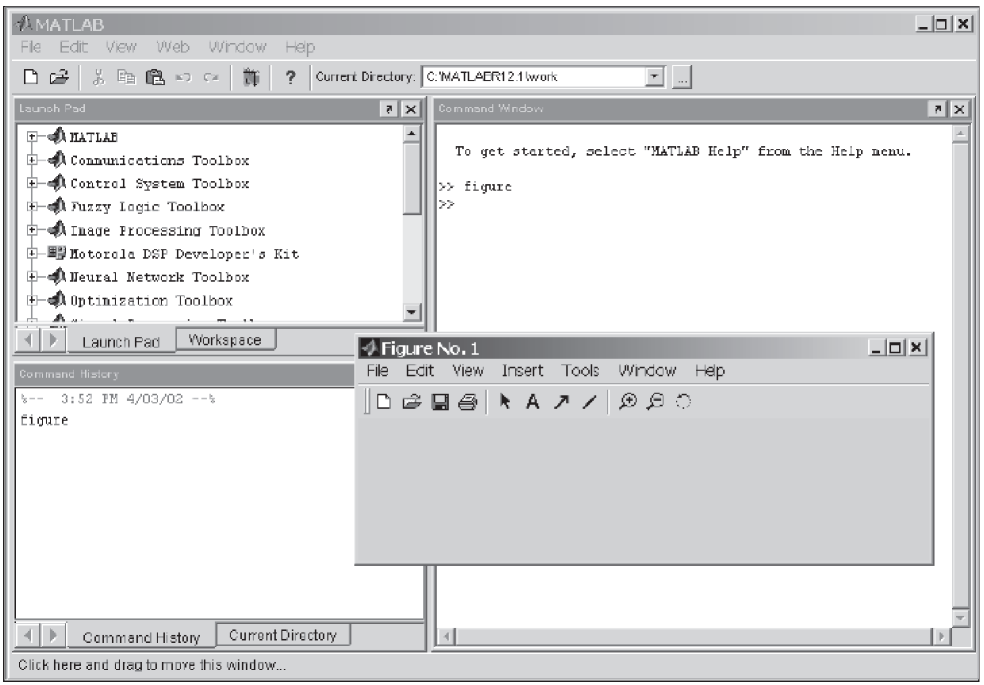
Grafický výstup je pro technika nezbytnou potřebou, neboť „obrázek poví za tisíc slov“, jak pravili již staří Číňané. MATLAB v tomto směru poskytuje značné možnosti jak dvojrozměrné (2D), tak i trojrozměrné (3D) vizualizace výsledků.

4.1 Stručný průvodce pracovní plochou

Nové grafické okno vytvoříme příkazem

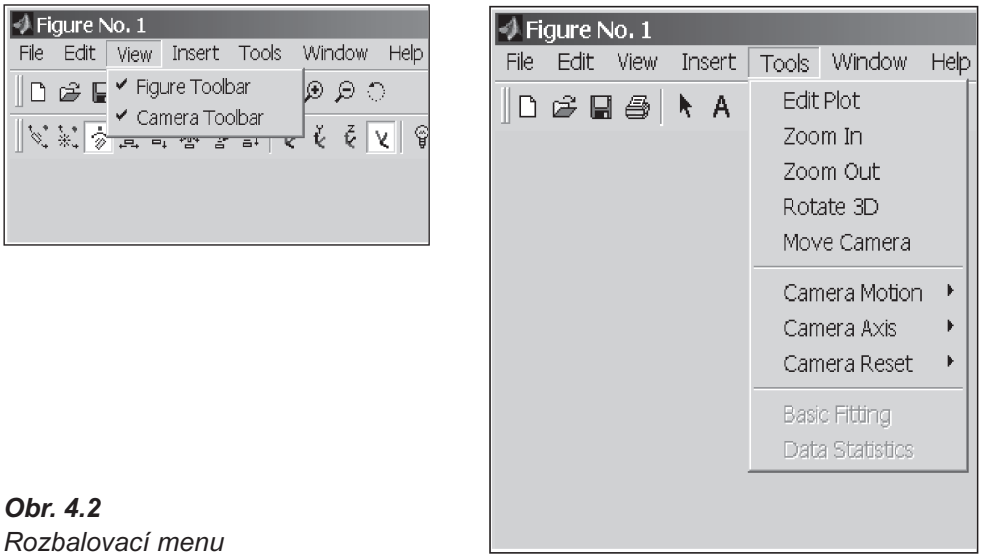
figure

zapsaným do hlavního okna Command Window. Založí se grafické okno **Figure No. 1** jako šedá plocha. Okno se opět chová jako běžné „windowsovské“ – lze jej posouvat, zmenšovat atp.



Obr. 4.1 *Vzhled pracovní plochy po založení grafického okna*

Součástí okna jsou rozbalovací menu a „implicitně“ i řada ikon *Figure Toolbar* – *Pruh nástrojů* obrázku. Pro ilustraci jsou na *obr. 4.2* uvedena dvě menu v rozbaleném tvaru a to menu *View* a *Tools*. Především proto, že zobrazují druhý přístupný nástrojový pruh *Camera Toolbar* a některé funkce nástroje *Camera*. Asi tušíte, že jde o nástroj využitelný plně až ve 3D grafice.

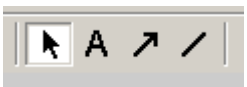


Obr. 4.2
Rozbalovací menu

Ikony mají formu typickou z windows; po najetí kurzoru se ve žlutém políčku objeví její název.

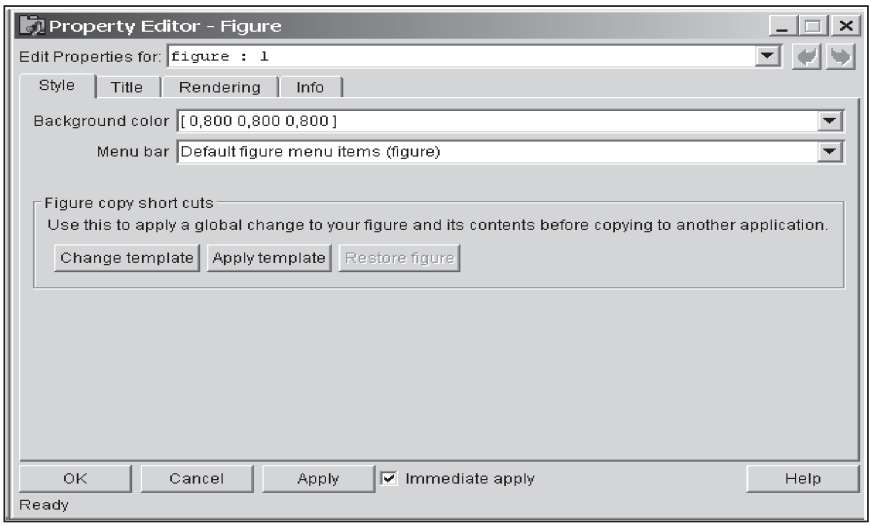
Ikonka *New Figure* – Nový Soubor  je stále plně funkční. Z prázdného obrázku *Figure 1* si takto hned můžeme založit prázdný obrázek *Figure 2*, atd. I další ikonky v první sekci jsou, jak vzhledem, tak funkcí identické s ostatními ve „windowsovských“ programech: *Open File* – Otevři soubor, *Save Figure* – Ulož obrázek, *Print Figure* – Vytiskni obrázek.

Nyní si povšimněte druhé sekce ikon ve Figure Toolbar:



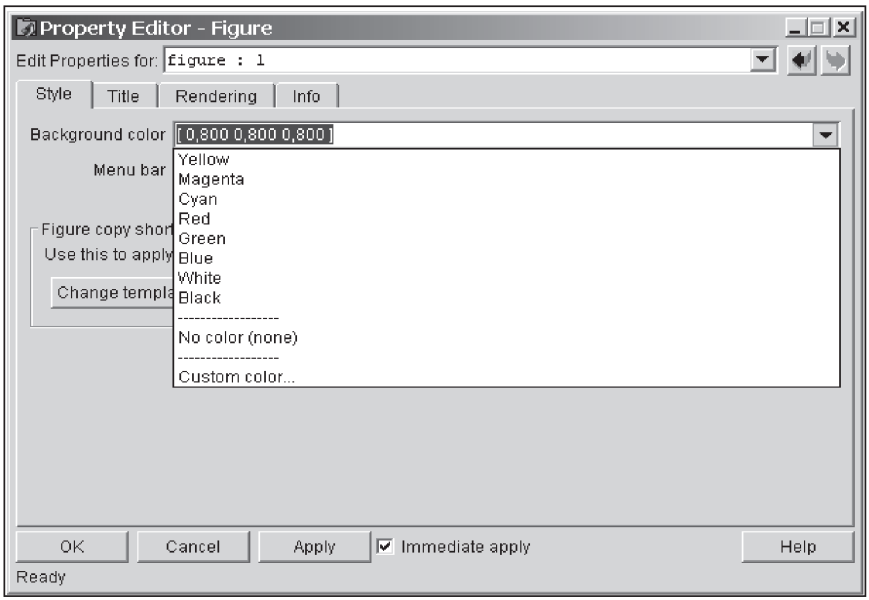
Ťukneme-li levým tlačítkem myši na ikonu šipky (*Edit Plot*) a následně provedeme na ploše obrázku *Figure 1* „dvojklik“, otevře se okno Editoru vlastností obrázku „*Property editor – Figure*“.

Pouhým pohledem je patrné, že zde je možno nastavovat a měnit veškeré atributy a nastavení příslušného budoucího obrázku. Náš obrázek *Figure 1* je prázdný a představuje jen šedou plochu pozadí. Ale i tu můžeme ovlivnit. Těm trochu obeznámeněj-



Obr. 4.3 *Editor vlastností obrázku*

ším s barevnou stupnicí je jasné, že škála [0,800 0,800 0,800] v rozbalovacím menu *Background color* (barva pozadí), představuje RGB souřadnice šedé barvy. Můžeme přímo rozbalit toto menu a změnit si barvu podkladu. Máme-li zaškrtnuto políčko *Immediate apply*, tak se vše dle tohoto příkazu okamžitě provede



Obr. 4.4 *Možnosti ovlivnění barvy podkladu obrázku*

5 Základní použití 3D grafiky a další typy grafů

Základní princip použití 3D (třírozměrné) grafiky je v souladu s tím, co bylo vysvětleno v kapitole 4 o práci s 2D grafikou. Mnohé příkazy pro práci s 3D grafy jsou analogií těch dvourozměrných, často jsou jim podobné i syntaxí. Rovněž editace 3D grafů je analogická jejich 2D ekvivalentům. Graf 3D má celkem tři osy, dvě nezávisle a jednu závisle proměnnou. Z toho vyplývají i nutná rozšíření možností nastavení grafů, ale i jejich základních typů. Tak jako v ostatních kapitolách, je i zde cílem uvést uživatele do problematiky. Prostudování všech možností a kombinací je pak na vaší píli. *Čtenář nechť se důkladně seznámí s kapitolou 4, kde je vysvětleno použití a editace 2D grafů*.

5.1 Vytvoření spojitého 3D grafu

Principiálně se tato činnost neliší od tvorby 2D ekvivalentu. Je třeba mít k dispozici:

- tři vektory, popisující osy x, y, z (dvě nezávisle a jednu závisle proměnnou),
- klíčový příkaz pro vykreslení 3D grafu, příp. další povely pro editaci grafu (nebo pomocí menu).

Zapište a potvrďte:

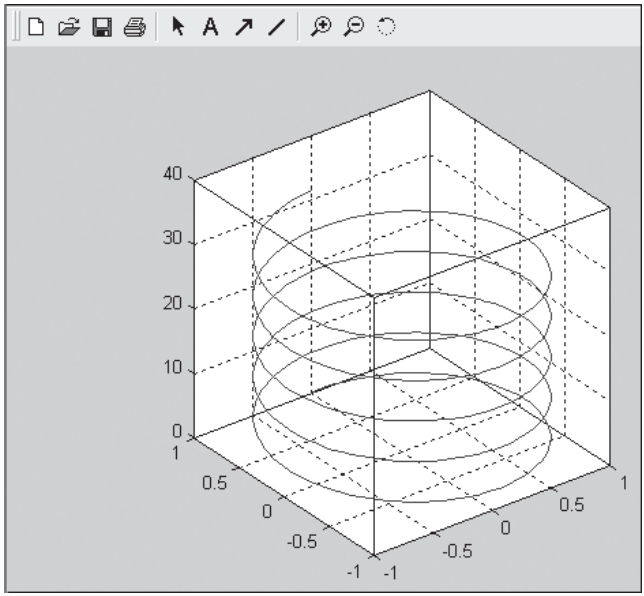
t=0:pi/50:10*pi;

plot3(sin(t),cos(t),t).

Dojde k vykreslení základního typu spojitého grafu (parametricky zadaná šroubovice). Aby graf vypadal lépe, je možné ještě dodat: **grid on** a **axis square** nebo **box on**. Vytvoří se mřížka a upraví osy. Tyto jednoduché editační povely jsou stejné jako u 2D grafu. Obecně je příkaz **plot3** tvořen syntaxí: **plot3(X,Y,Z)**, kde X, Y a Z jsou vektory nebo matice. V případě matic se kreslí více křivek pro každý sloupec jednotlivých matic. Jako u 2D grafu je možné přidat další parametry povelu **plot3** pro bližší určení typu kreslených bodů a čar, barev apod. Odezva na uvedené příkazy je na *obr. 5.1*.

Další editaci vytvořeného 3D grafu lze provést analogicky podle popisu v kapitolách 4.2 až 4.5 (popis os, natočení grafu, změna tloušťky čar, editace měřítka na osách apod.). Lze jen doporučit provádět toto pomocí tlačítek nad obrázkem. Využívejte nápovědu **help plot3** nebo okno pro interaktivní help (viz kapitolu 2.5).

```
Command Window
>> t=0:pi/50:10*pi;
>> plot3(sin(t),cos(t),t)
>> grid on
>> axis square
>> box on
>>
```



Obr. 5.1 *Vytvoření základního spojitého 3D grafu*

5.2 Matice jako plocha

Kromě základního spojitého grafu je však v Matlabu zakomponována možnost *kreslení plošných 3D grafů*. Protože celý systém je orientován maticově, je matice základním objektem k dosažení takových velmi efektních grafů.

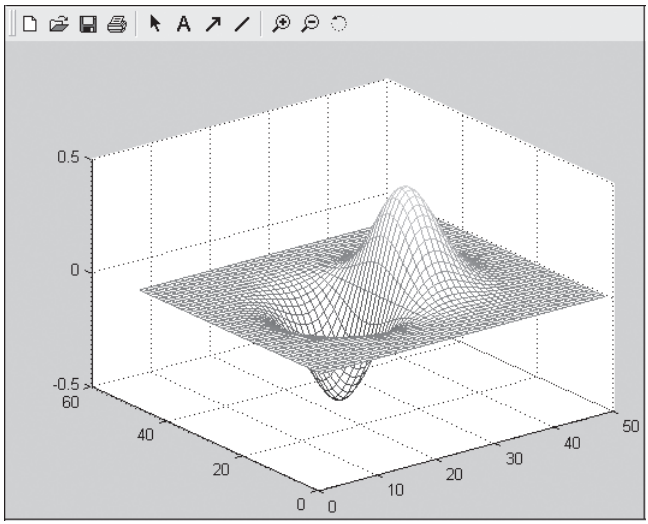
3D síťovaný graf:

Zapište a potvrďte:

x=–3:0.125:3;	vytvoření vektoru na ose x (jedna nezávisle proměnná),
y=x;	vytvoření vektoru na ose y (druhá nezávisle proměnná),
[X,Y]=meshgrid(x,y);	vytvoření zvláštní matice (mřížky), která umožní následné 3D kreslení,
Z=X.*exp(–X.^2–Y.^2);	definice osy závisle proměnné,
mesh(Z)	klíčový povel pro vykreslení 3D síťovaného grafu.

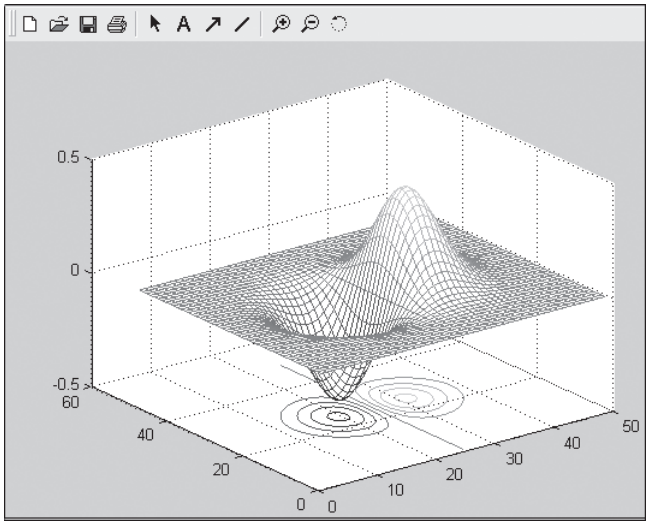
Obrázek, dosažený povelem **mesh**, je velmi efektní. Došlo k vykreslení tzv. síťovaného, barevně kolorovaného grafu. Barevně jsou rozlišeny hodnoty závisle proměnné podle své velikosti. *Hustota sítě grafu* závisí na počtu bodů definovaných vektorů **x** a **y**. Vyzkoušejte místo povelu **mesh** povel **meshc**. Výsledný graf je pak kombinací grafu plošného síťovaného a tzv. *contour* (vrstevnicového) grafu, viz *obr. 5.2* a *obr. 5.3*.

```
Command Window
>> x=-3:0.125:3;
>> y=x;
>> [X,Y]=meshgrid(x,y);
>> Z=X.*exp(-X.^2-Y.^2);
>> mesh(Z)
>>
```



Obr. 5.2 *Tvorba základního typu 3D síťovaného grafu*

```
Command Window
>> x=-3:0.125:3;
>> y=x;
>> [X,Y]=meshgrid(x,y);
>> meshc(Z)
>>
```



Obr. 5.3 *Varianta základního typu 3D síťovaného grafu*

Opět lze použít již známé editační povely (axis, grid on, box on atd.). Povel **axis** bude mít však nikoli čtyři, nýbrž šest parametrů (tři osy místo dvou).