1

Matematické výpočetní systémy[1]

Doc. RNDr. Vítězslav Veselý, CSc.

Katedra aplikované matematiky
PřF MU Brno
vesely@math.muni.cz

16. listopadu 2003

[1]Zpracováno v rámci projektu G0142 "Distanční a kombinovaná forma bakalářského a magisterského studia aplikovaných matematických disciplin na MU v Brně".

1 Úvod

Pod pojmem *Matematický výpočetní systém* (MVS) rozumíme obecně jakýkoliv softwarový prostředek využívající matematický aparát pro řešení problémů v oblasti matematiky samotné nebo i v jiných oborech, které aplikace takového aparátu vyžadují. Tyto systémy jsou zpravidla koncipovány jako zcela autonomní s možností práce v interaktivním i dávkovém režimu.

Vzhledem k šíři nabídky takových systémů se v této úvodní části omezíme spíše na výčet a základní charakteristiku nejznámějších nebo nejpoužívanějších z nich.

Systémy lze rozčlenit do tří kategorií:

Systémy počítačové algebry: Mezi tři nejznámější patří:

- **Maple** kanadské společnosti *MapleSoft* (`http://www.maplesoft.com`), v České republice distribuovaný firmou *Czech Software First s.r.o.*
- **Mathematica** společnosti *Wolfram Research* (`http://www.wolfram.com`), v České republice distribuovaný firmou *ELKAN s.r.o.* (`http://www.mathematica.cz`).
- **MathCad** americké společnosti *Mathsoft* (`http://www.mathsoft.com`), v České republice distribuovaný firmou *Software Factory s.r.o.* (`http://www.mathsoft.cz`).

Jedná se o systémy zaměřené na formální algebraické výpočty neboli symbolické manipulace s algebraickými výrazy. Počítačovou algebru lze charakterizovat jako obor vědeckého počítání, který vyvíjí, analyzuje, implementuje a používá algebraické algoritmy. Provádění numerických výpočtů je samozřejmě rovněž možné, vesměs dokonce se zvolenou přesností, tj. na zvolený počet desetinných míst. Další podrobnější informace lze nalézt například na
`http://www.math.muni.cz/~plch/difer/cela/node2.html`,
`http://www-troja.fjfi.cvut.cz/~liska/poalg/` nebo
`http://www.can.nl/`.

Numerické výpočetní systémy: Opět se omezíme jen na stručný výběr několika nejznámějších:

- **MATLAB** americké společnosti *MathWorks* (`http://www.mathworks.com`), v České republice distribuovaný firmou *Humusoft, s r.o.* (`http://www.humusoft.cz`).
- **NAG** anglické společnosti *Numerical Algorithms Group* (`http://www.nag.com`).
- **S-plus** americké společnosti *Insightful Corporation* (`http://www.insightful.com`), v České republice distribuovaný firmou *TriloByte s.r.o.* (`http://www.trilobyte.cz/splus`).
- **SPSS** americké společnosti *SPSS, Inc.* (`http://www.spss.com`), v České republice zastoupené firmou *SPSS CR s.r.o.* (`http://www.spss.cz`).
- **Statistica** americké společnosti *StatSoft, Inc.* (`http://www.statsoft.com`), v České republice zastoupené firmou *StatSoft Czech Republic s.r.o.* (`http://www.statsoft.cz`).

- **Statgraphics** americké společnosti *Statistical Graphics Corporation* (`http://www.sgcorp.com`).
- **SAS** americké společnosti *SAS Institute, Inc.* (`http://www.sas.com/`), v České republice zastoupené firmou *SAS Institute ČR, s.r.o.* (`http://www.sas.com/offices/europe/czech/sas`).

Jedná se o systémy zaměřené na numerické výpočty v pohyblivé řádové čárce. Na rozdíl od systémů počítačové algebry není obvykle možné volit počet desetinných míst přesnosti. Standardně jsou výpočty prováděny s přesností na cca 15 desetinných míst, což odpovídá 64-bitové reprezentaci čísel v pohyblivé řádové čárce, neboli uložení každého reálného čísla spotřebuje 8 bajtů vnitřní paměti (u komplexních čísel dvojnásobek). Jak je zřejmé z předchozího výčtu, je většina těchto systémů zaměřena na statistické výpočty a modelování. Pokud je záběr systému širší (MATLAB), pak statistické výpočty bývají podporovány volitelnými specializovanými knihovnami funkcí. Ucelený přehled komerčních i volně šířených systémů umožňujících (mimo jiné) provádění statistických výpočtů lze nalézt na `http://ourworld.compuserve.com/homepages/Rainer_Wuerlaender/statsoft.htm`.

Specializované systémy: Typickým představitelem je

- **PARI-GP** — volně šířený systém specializovaný na výpočty v teorii čísel. Podrobnější informace lze nalézt na `http://pari.math.u-bordeaux.fr/`.

V tomto kurzu se budeme nadále podrobněji zabývat pouze systémem MATLAB. Důvodem pro tuto volbu je několik skutečností:

- Ve standardní výbavě MATLAB nabízí velmi široké spektrum funkcí napříč nejrůznějšími oblastmi matematiky doplněný mnoha volitelnými speciálními knihovnami jak pro potřeby matematiky tak i jiných oborů. Namátkově lze zmínit například: statistické výpočty a modelování, optimalizace, splajny, wavelety, číslicové zpracování signálů a obrazů, dynamické systémy a řízení, fuzzy logika, finanční matematika a mnoho dalších. Je takto dokonce zpřístupněno i jádro systému MAPLE, takže MATLAB není uzavřen ani symbolickým manipulacím typickým pro systémy počítačové algebry.

- Intuitivní programovací jazyk se syntaxí blízkou matematickému stylu výrazně zvyšující produktivitu práce při vývoji numericky náročných algoritmů.

- Propracované algoritmy numericky kvalitně ošetřené, což vede k celkové vysoké stabilitě systému i při řešení rozsáhlých výpočetně i paměťově náročných úloh.

- MATLAB se po letech vyvinul v celosvětový standard v oblasti analýzy, zpracování a modelování dat jak na akademických pracovištích tak i v komerční sféře.

- Je vhodný jako podpůrný prostředek do výpočetních praktik k mnoha dalším matematickým kurzům — mimo jiné také proto, že algoritmy většiny funkcí jsou přístupné k podrobnému studiu či dalším modifikacím.

- Kvalitní grafický subsystém pro snadnou vizuální prezentaci dat i výsledků zpracování.

- Existence celé řady obdobných avšak cenově dostupnějších systémů, nabízejících kompatibilitu svého jazyka s jazykem MATLABu a nebo prostředky pro automatickou konverzi programů mezi oběma jazyky. Jako typické představitele lze uvést tyto systémy:
  - **O-matrix**, komerční systém, avšak podstatně levnější než MATLAB (`http://www.omatrix.com`).
  - **Octave**, volně šířený systém dostupný i pod Linuxem (`http://www.octave.org`).
  - **Scilab**, volně šířený systém dostupný i pod Linuxem (`http://www-rocq.inria.fr/scilab`).

Naším cílem však nebude a ani nemůže být vyčerpávající popis všech funkcí a možností systému MATLAB. K tomu jsou určeny firemní manuály (dostupné i v elektronické podobě) a řada dalších specializovaných publikací (viz kapitolu 2).

V následujících kapitolách tohoto textu se proto zaměříme spíše na výklad základní filozofie a principů, na nichž systém pracuje. Po jejich prostudování pak posluchač může přistoupit k získání základních praktických dovedností, které mu umožní využít poznatky získané v jiných matematických předmětech k efektivnímu a tvůrčímu řešení reálných úloh a problémů.

Pro tento účel byla zvolena netradiční metoda interaktivního tutoriálu. Posluchač si jej spustí jako samostatnou aplikaci přímo po systémem MATLAB. (viz přílohu A). Aplikace jej postupně v 11 lekcích provádí jednotlivými tétmatickými okruhy. Každá lekce systematicky vysvětluje a na živých ukázkách přímo ilustruje syntaxi a funkci probíraných příkazů. V interaktivním režimu `keyboard` má student možnost si sám prakticky vyzkoušet předtím ilustrované příkazy.

V příloze B lze ke každé lekci nalézt zadání praktických úloh k vyřešení. Jejich vypracováním si student dále fixuje získané poznatky a učí se MATLAB tvůrčím způsobem využívat.

2 Charakteristika systému MATLAB

2.1 Základní fakta

Název systému je odvozen z angličtiny: MATLAB=MATrix LABoratory. Systém produkuje americká společnost *MathWorks, Inc.* (`http://www.mathworks.com`) a jejím výhradním distributorem pro Českou republiku je pražská firma *Humusoft, s.r.o.* (`http://www.humusoft.cz`). Systém MATLAB představuje maticově orientované výpočetní prostředí s vlastním programovacím jazykem vhodným pro efektivní implementaci, provádění a vizualizaci numerických výpočtů vyžadujících náročný aparát z nejrůznějších oblastí matematiky. Je dostupný na všech běžných platformách (WINDOWS, LINUX, UNIX, atd.) při zachování takřka stoprocentní přenositelnosti vytvořených programů a datových souborů.

2.2 Programovací jazyk

- Programovací jazyk vyšší generace umožňuje automatické deklarování a přetypování používaných proměnných. Podporované datové typy jsou tyto:

  Základním datovým typem je *matice* nebo obecnější *vícerozměrné pole (N-D array)*, jehož prvky mohou být jakékoliv reálné nebo komplexní skalární veličiny.

  Odvozené datové typy jsou *buněčná pole (cell arrays)* svojí strukturou podobné základnímu datovému typu s tím rozdílem, že za prvky mohou mít proměnné libovolného typu (tedy nejen skalární veličiny), dále *struktury* a *objekty* s položkami libovolného přípustného typu umožňující využívat techniky strukturovaného a objektového programování běžné v klasických programovacích jazycích jakým je například C++.

- Na rozdíl od *skalárního programování* známého z klasických programovacích jazyků jako FORTRAN, PASCAL nebo C++ je pro MATLAB typické *vektorově-maticové programování*, neboť základními proměnnými jsou právě vektory a matice. Syntaxe jazyka je pak blízká matematickým zápisům používaným v lineární algebře, což usnadňuje přepis i komplikovaných matematických výrazů při zachování srozumitelnosti. Efektivní využívání všech konstrukcí jazyka tak vyžaduje opustit zaběhaný *skalární* přístup, neboli již ve fázi návrhu algoritmu je třeba provést tzv. *vektorizaci* řešeného problému, tj. již při jeho formulaci je třeba se snažit v maximálně možné míře využít formalismu maticové algebry. Vynaložená námaha se vrátí v přehlednosti, spolehlivosti a rychlosti výsledného kódu.

2.3 M-soubory

Základními programovými jednotkami jsou buď dávkové (skripty) nebo funkční procedury zapsané v jazyce MATLABu. Jsou to soubory s příponou `.m`, neboli tzv. **`M`-soubory**, které lze připravit jako otevřený text libovolným textovým editorem. Systém MATLAB má pro tento účel vlastní editor, který je součástí instalace a nabízí další specifické funkce. Spustí se příkazem `edit`. Mimo jiné automaticky barevně rozlišuje klíčová slova jazyka MATLAB a upozorňuje tak na možné překlepy již ve fázi psaní programu, v režimu ladění (debugging) pak nabízí prostředky pro krokování běžícího programu a sledovaní mezivýsledků v lokálních proměnných.

- Program uložený v dávkovém M-souboru se spustí v příkazovém okně MATLABu jednoduše zadáním povelu, jehož jméno je totožné se základním jménem M-souboru (bez přípony .m). Všechny proměnné použité programem jsou sdíleny s příkazovým oknem a tedy jsou z něj interaktivně přístupné.

- Programy funkčního M-souboru vyžadují speciální deklaraci vstupních a výstupních parametrů v jeho záhlaví. Funkční program se opět spouští obdobně jako u dávek s tím rozdílem, že povel je doplněn o seznamy skutečně přenášených vstupních a výstupních proměnných, jejichž hodnoty korespondují v daném pořadí s příslušnými vstupními a výstupními parametry. Ostatní proměnné používané funkční procedurou mají lokální charakter a nejsou tedy běžně přístupné v příkazovém okně. U tzv. *globálních* proměnných speciálně takto deklarovaných toto omezení neplatí.

- MATLAB je rychlý interpret. Programy v M-souborech není tedy třeba předem kompilovat. Jsou přímo prováděny (interpretovány) po spuštění ve tvaru připraveném editorem. Interpretace je velmi efektivní, neboť zahrnuje skrytou automatickou předkompilaci, která se provádí jen jednou při prvním spuštění. Opakovaná spouštění pak užívají předkompilovaný kód uložený ve vnitřní paměti a běží proto výrazně rychleji. Tento předkompilovaný kód lze z paměti odstranit pomocí příkazu `clear`. Užívá se v případech, kdy potřebujeme modifikovat dříve spuštěný M-soubor uložený mimo aktuální adresář.

2.4 Vestavěné příkazy a MEX-soubory

Většina příkazů jazyka MATLAB je realizována pomocí M-souborů. Použité algoritmy jsou tak otevřeny ke studiu i modifikaci uživatelem. Výjimkou jsou jen některé *vestavěné (built-in)* příkazy jádra systému, které jsou jeho plně kompilovanou součástí s cílem maximalizovat výkon celého systému nebo některých jeho funkcí vyžadujících vysokou výpočetní rychlost.

Uživatel MATLABu může užít externí kompilátor jazyka C nebo FORTRAN k tvorbě vlastních programových modulů, které skýtají obdobné možnosti jako vestavěné příkazy. Představují tak alternativu k M-souborům v případech náročných na výpočetní rychlost. Takto vytvořené binární soubory se nazývají `MEX-soubory`. Pod operačním systémem Windows tuto roli přebírají soubory s příponou `.dll` (Dynamically Linked Library). MEX-soubor vytvořený pod jednou platformou (operačním systémem) není ale přímo přenositelný. Na jiné platformě je tak třeba příslušný modul znovu zkompilovat.

Systém nabízí i možnost využít MATLAB jako výpočetní stroj pro samostatný a souběžně spuštěný program zkompilovaný z jazyka C nebo FORTRAN. Takový program může prostřednictvím speciálního rozhraní využívat libovolné funkce MATLABu.

2.5 Specializované knihovny funkcí

Tématicky zaměřené knihovny M-souborů (a případně i MEX-souborů) soustředěné v jednom adresáři jsou v terminologii MATLABu nazývány *toolboxy*. V současné době existuje široká nabídka nejrůznějších specializovaných toolboxů ať již volně ke stažení na internetu nebo komerčních (firemních od *MathWorks* nebo od jiných subjektů). Systémové firemní toolboxy jsou integrální součástí základního systému. Pro větší přehlednost třídí dostupné funkce do příbuzných tematických skupin. Zvláštní pozornost zaslouží následující tři volitelné firemní toolboxy:

- *Symbolický toolbox (Extended Symbolic Math Toolbox)*, který poskytuje rozhraní k jádru známého systému MAPLE zmíněnému již v kapitole 1. Jádro systému MAPLE je přímo integrální součástí toolboxu a neinstaluje se tedy odděleně od MATLABu. MAPLE je na rozdíl od MATLABu orientován na symbolické výpočty a výpočty s vysokou přesností. Naopak silnou stránkou MATLABu je široká nabídka numericky stabilních výpočetních algoritmů z nejrůznějších oblastí. Symbolický toolbox propojuje obě prostředí a umožňuje využít předností každého z nich. Navíc symbolické výrazy mohou vystupovat přímo jako prvky v maticích, mohou být manipulovány při symbolicky prováděných maticových operacích a nakonec jednoduše numericky vyhodnoceny po dosazení konkrétních hodnot za symbolické proměnné.

- *Statistický toolbox* nabízející základní funkce potřebné při modelování a výpočtech ve statistice. Namátkou vyjmenujme některé z nich. Pro většinu běžných rozložení jsou k dispozici funkce pro výpočet hustoty, distribuční funkce a funkce k ní inverzní (výpočet kvantilů), odhad parametrů rozdělení z histogramu, generátory pseudonáhodných čísel, funkce popisné statistiky, lineární a nelineární modely a mnoho dalších.

- *Optimalizační toolbox* zahrnující běžné techniky lineárního i nelineárního programování.

2.6 Grafický subsystém

Výborně navržený grafický subsystém nabízí funkce z těchto dvou hlavních oblastí:

- Funkce pro kreslení 2-D i 3-D grafů a řadu dalších nástrojů pro názornou vizualizaci dat. Vyznačuje se vysokou flexibilitou (vlastnosti grafických objektů lze měnit nastavováním velkého množství řídících parametrů) při zachování jednoduchosti a intuitivnosti ovládání (většina parametrů má nastaveny implicitní hodnoty).

- Nástroje pro interaktivní návrh uživatelského rozhraní (GUI=Graphical User Interface) umožňují vytvářet pro koncové uživatele aplikace ovládané pomocí menu a dalších běžných grafických ovládacích prvků (tlačítka, seznamy, zatrhávací položky aj.). Spouští se příkazem `guide`. Tvůrce aplikace se tak může vyhnout příkazovému režimu, který bývá pro většinu koncových uživatelů nevhodný.

2.7 Kompilace z MATLABu do jazyka C

Kompilátor MATLAB $\longrightarrow$ C je nadstandardní výbava systému umožňující automatický převod programů zapsaných v jazyce MATLABu do jazyka C. Po převodu je možno takto získaný zdrojový C kód samostatně přeložit vhodným kompilátorem jazyka C a sestavit pomocí knihoven MATLABu dodaných speciálně pro tento účel.

Obdržíme tak aplikaci (obvykle spustitelný soubor .EXE), který pak lze užívat či dále distribuovat ke koncovým uživatelům zcela nezávisle na prostředí systému MATLAB. Pořízení kompilátoru je dosti nákladné a vyplatí se proto pouze větším softwarovým firmám, kterým umožní výrazně redukovat náklady na vývoj jejich vlastních aplikací. Přímý vývoj a testování takových aplikací v jazyce C totiž vyžaduje několikanásobně větší nároky na čas a lidské zdroje, než v prostředí MATLABu.

2.8 Spuštění a ukončení MATLABu

MATLAB spustíme pod operačním systémem provedením příkazu `matlab.exe`, který se nachází v podadresáři `bin` kořenového adresáře instalovaného MATLABu. Pod operačním systémem WINDOWS je spuštění možné též prostřednictvím nabídky START a nebo kliknutím na vytvořeného zástupce MATLABu na pracovní ploše. Po spuštění se objeví *příkazové okno*, v němž symbol » (angl. *prompt*) signalizuje připravenost systému přijímat příkazy.
Práci MATLABu ukončíme příkazem `exit` nebo `quit`.

2.9 Online nápověda, manuály a učebnice

Online nápověda je nápomocná uživateli přímo během práce v příkazovém okně. Nápověda pracuje strukturovaně v několika úrovních podrobností:

- příkaz `help` zobrazí seznam tématických okruhů (toolboxů).
- příkaz `help jméno_toolboxu` vypíše seznam funkcí zvoleného tématického okruhu (toolboxu).
- příkaz `help jméno_funkce` nebo `help operátor` dá podrobnou nápovědu k danému konkrétnímu příkazu, resp. operátoru; zejména `help help` vypíše nápovědu k samotnému příkazu `help`.
- příkaz `doc` zobrazí pomocí rezidentního internetového prohlížeče (zpravidla *Netscape* nebo *Internet Explorer*) samostatné okno tzv. *HelpDesk*. Toto okno představuje nejpodrobnější zdroj informací o systému MATLAB i všech nainstalovaných toolboxech. Lze v něm dokonce nalézt všechny tištěné manuály v elektronické podobě jako soubory s příponou .PDF pro rozšířený prohlížeč *Acrobat Reader*. Jsou zde také přímé odkazy na domovskou stránku firmy *MathWorks*, kde lze nalézt například informace o dalších nabízených produktech a učebnicích (viz odkaz `http://www.mathworks.com/support/books/`).

Pro samostatné studium lze samozřejmě využít i originální anglicky psané manuály v tištěné podobě vztahující se jak k základnímu systému (viz např. [1, 2]) tak i k jednotlivým toolboxům, případně některou z učebnic nebo příruček vydávaných jinými autory v nejrůznějších světových nakladatelstvích.

Publikací tohoto druhu jsou i česky psaná skripta vydaná na Masarykově univerzitě v Brně [3] a na Západočeské univerzitě v Plzni [4, 5].

A Interaktivní průvodce systémem MATLAB

V příkazovém okně MATLABu nejprve zkontrolujeme, zda adresář `MATLABtutorial` je nastaven jako aktuální. Pokud tomu tak není, provedeme nastavení příkazem `cd`. Alternativně lze pro tento účel zvolit v menu *File* příkazového okna MATLABu položku *Set Path* a pak tlačítkem *Browse* aktivovat adresář `MATLABtutorial`. Poté již můžeme průvodce spustit zadáním povelu `tutorial`. Objeví se hlavní nabídka jako na obrázku 1 s těmito ovládacími prvky:

Seznam témat: Zvolíme požadovaný tématický okruh z nabídky jedenácti lekcí.

Obsah tématu: Udává stručný obsah zvoleného tématu.

Subtémata: Vybereme dílčí téma, případně dvojklikem je rovnou aktivujeme stejně jako při stisku tlačítka "Spustit".

Tlačítko "Konec": Ukončí činnost průvodce.

Tlačítko "Nápověda": Stručný popis ovládání hlavní nabídky.

Tlačítko "Spustit": Vyvolá druhé nabídkové okno jako na obrázku 2 umožňující ovládat výběr demonstračních příkladů pro zvolené subtéma.

Nabídkové okno pro volbu demonstračních příkladů má následující ovládací prvky:

Záhlaví zleva doprava: Vybraný tématický okruh (lekce), vybrané subtéma a pořadové číslo jeho části.

Okno pod záhlavím: Obsahuje stručný vysvětlující popis jednoho nebo více příkazů zvolené části subtématu.

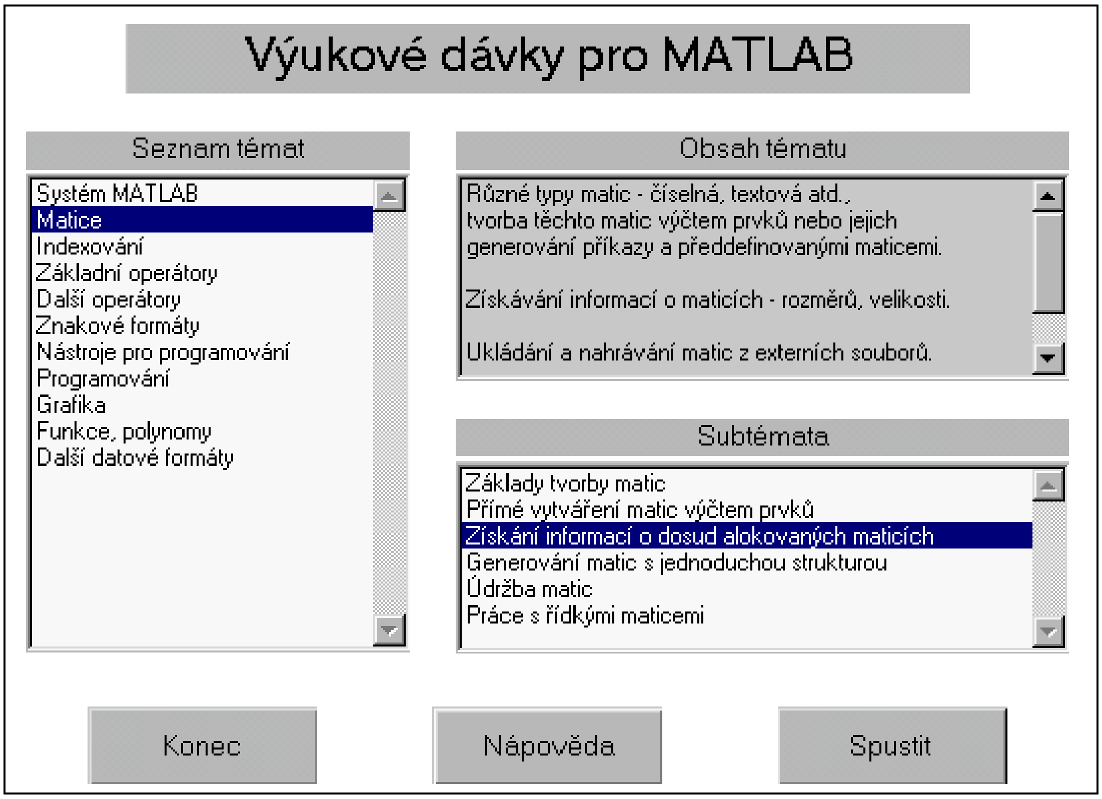
Tlačítko "Příklad": V příkazovém okně MATLABu spustí demonstrační příklad pro zvolenou část subtématu. Demonstrace probíhá postupně tak, aby bylo možno sledovat provádění automaticky prováděných příkazů. V režimu `pause` zajistíme pokračování stiskem libovolné klávesy. V interaktivním režimu `keyboard` je po K» možno zadávat jakékoliv příkazy a vyzkoušet si tak probírané povely podle vlastního uvážení. Po zadání příkazu `return` je interaktivní režim ukončen a demonstrační příklad pokračuje. Takto postupujeme až do ohlášení konce příkladu. Příklad je nutno v každém případě projet až do konce. Pak je možno jej spustit buď znovu a nebo přejít na další subtéma.

Tlačítko "Dále": Vybere v pořadí další část subtématu.

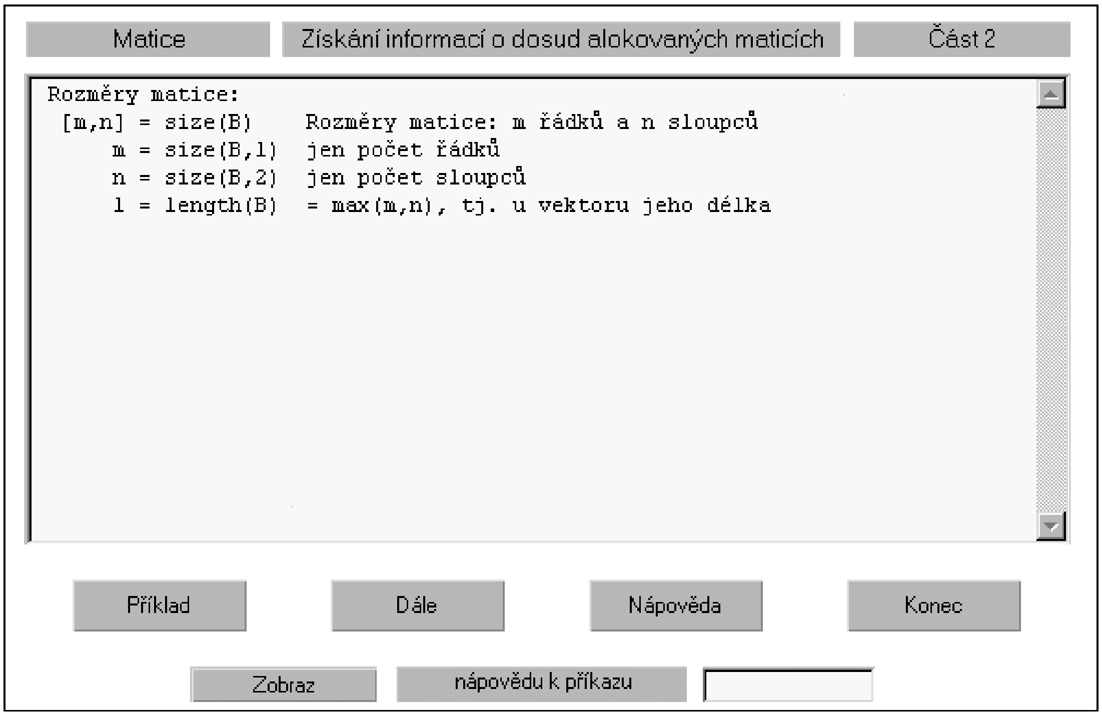
Tlačítko "Nápověda": Stručný popis ovládání nabídky demonstračních příkladů.

Tlačítko "Konec": Zavře okno pro volbu demonstračních příkladů.

Tlačítko "Zobraz": V rezidentním internetovém prohlížeči zobrazí firemní online nápovědu k příkazu zadanému do editačního pole napravo.



Obrázek 1: Hlavní nabídkové okno



Obrázek 2: Nabídkové okno pro volbu demonstračních ukázek

B Praktické úlohy k řešení v MATLABu

Ke každé lekci následuje seznam praktických úloh. Po zvládnutí příslušné lekce za pomoci interaktivního průvodce se doporučuje vypracovat příslušné úlohy k této lekci a dále si tak upevnit získané poznatky. Student si takto postupně tvůrčím způsobem osvojuje používání MATLABu jako účinného a efektivního nástroje k řešení nejrůznějších problémů vyžadujících matematický aparát.

Lekce 1: syntaxe jazyka, ovládání příkazového okna, práce se soubory a operačním systémem, nápověda a kontextové vyhledávání, seznam připojených knihoven (tzv. MATLABPATH).

1. Nastavte adresář `MATLABtutorial` jako aktuální a vytvořte v něm podadresář `MATLABcviceni`, který pak nastavte také jako aktuální. V něm budete nadále řešit úlohy z této a dalších lekcí.
2. Zajistěte vytvoření záznamu řešení následujících úloh tohoto cvičení.
3. Nalezněte jméno funkce, která počítá stopu matice (angl. *trace*).
4. Zjistěte, v kterém adresáři (knihovně) je uložena.
5. Vypište nápovědu a tělo této funkce. Porovnáním obou výpisů vydedukujte, jak se realizuje nápověda v m-souborech.
6. Zkopírujte pod týmž jménem tuto funkci do sveho aktuálního adresáře a pomocí editoru v ní nápovědu počeštěte.
7. Zopakujte postup z kroků 3 až 5 — co jste zjistili?
8. Vypište všechny m-soubory ve Vašem aktuálním adresáři.
9. Připojte na konec MATLABPATH Vaši knihovnu `MATLABcviceni`.
10. Odstraňte tuto knihovnu pomocí příkazu `rmpath`.
11. Uložte stávající MATLABPATH do proměnné `p`.
12. Znovu knihovnu `MATLABcviceni` přidejte, ale tentokrát na začátek MATLABPATH.
13. Obnovte stav z kroku 11 s využitím proměnné `p`.
14. Ukončete záznam řešení zahájený v kroku 2.

Lekce 2: prázdná matice, přímé vytváření matic výčtem prvků, generování matic se specifickou strukturou, získání informací o dosud alokovaných maticích, údržba matic.

1. Vytvořte náhodnou matici `A` rozměru 10×8, jejíž všechny prvky jsou celá čísla v intervalu $[-50, 40]$.
2. Do `B` uložte submatici z `A` tvořenou jejími řádky na sudých a sloupci na lichých pozicích.
3. Na tytéž pozice uložte v původní matici `A` samé nuly.
4. Zjistěte velikost matice `B`, resp. jen počet jejích řádků a sloupců.
5. Co se stane, když provedete přiřazení `A(1,9)=1`?

6. Uložte matici `B` do souboru v binárním tvaru a pak ji vymažte z paměti (zkontrolujte vypsáním obsahu paměti).
7. Načtěte uloženou matici zpět a opět proveďte kontrolní výpis.
8. Zopakujte kroky 6 a 7 s tím rozdílem, že matici uložíte jako textovou, obsah tohoto souboru vypište.
9. Vypočtěte délku druhého řádku matice `A`.
10. Utvořte matici 8×8, která bude mít na hlavní diagonále tytéž prvky jako zde má matice `A`, jinak ale samé nuly.
11. Nalezněte ekvidistantní dělení intervalu $[-2, 10]$ s krokem $0,5$.

Lekce 3: techniky indexování a práce se submaticemi, submatice na levé straně přiřazovacího příkazu, změna rozměru matice.

1. Vygenerujte sloupcový vektor $\boldsymbol{u} = [-5, -3, -1, \dots]$ délky 20.
2. Doplňte $\boldsymbol{u}$ na vektor $\boldsymbol{v} = [-5, -4, -3, -2, -1, 0, \dots]$ délky 40 tvořící souvislou posloupnost.
3. Utvořte z vektoru $\boldsymbol{v}$ matici $\boldsymbol{V}$ rozměru 5×8, v níž prvky vektoru $\boldsymbol{v}$ budou uloženy po sloupcích.
4. Odstraňte 7. sloupec matice $\boldsymbol{V}$.
5. Pomoci $\boldsymbol{V}$ vytvořte matici $\boldsymbol{A}$, jež obsahuje řádky matice $\boldsymbol{V}$ v obráceném pořadí.
6. Zjistěte počet prvků matice $\boldsymbol{A}$ a uložte jej do proměnné n.
7. Na náhodně vybraných $(n-1)/2$ pozic v matici $\boldsymbol{A}$ uložte hodnotu kladného nekonečna, ostatní prvky ponechte beze změny.
8. Kladná nekonečna v matici $\boldsymbol{A}$ nahraďte zápornými nekonečny.

Lekce 4: základní maticové operace, speciální maticové operace (operace po složkách, levé a pravé dělení, umocňování, Kroneckerův součin aj.).

1. Vygenerujte nahodně matice $\boldsymbol{A}, \boldsymbol{B}$ téhož rozměru 3×5, jejichž prvky jsou komplexní čísla s celočíselnou reálnou i imaginární částí v intervalu -5 až 5.
2. Vytvořte z $\boldsymbol{B}$ transponovanou matici $\boldsymbol{C}$ a hermitovsky sdruženou matici $\boldsymbol{D}$.
3. Prověřte, zda submatice vytvořená z prvých tří řádků a sloupců v $\boldsymbol{B}$ je unitární.
4. Na maticích $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ vyzkoušejte všechny binární operace, které znáte.
5. Spočtěte matici $\boldsymbol{E}$ jako součin submatic tvořených prvými třemi sloupci matice $\boldsymbol{A}$ a prvými třemi řádky matice $\boldsymbol{C}$. Výpočtem ověřte, že determinant součinu $\boldsymbol{E}$ je součinem determinantů výše popsaných submatic.
6. Spočtěte determinant a stopu matice $\boldsymbol{E}$ pomocí jejích vlastních čísel.
7. Sestavte složený příkaz, který za poslední řádek matice $\boldsymbol{A}$ doplní lineární kombinaci jejích řádků, kde každý řádek je vždy násoben svým řádkovým indexem.
8. Najděte vektor $\boldsymbol{v}$, který je ortogonální projekcí prvého sloupce matice $\boldsymbol{B}$ na prvý sloupec matice $\boldsymbol{A}$.

9. Doplňte ortogonální vektory získané v předešlém kroku na ortogonální bázi v trojrozměrném vektorovém prostoru nad tělesem komplexních čísel.
10. Vytvořte z $\boldsymbol{A}$ matici, kde na místě každého prvku bude submatice 3×2 s tímto prvkem na každé pozici.

Lekce 5: relační a logické maticové operátory, smíšené aritmeticko-logické výrazy, logické funkce, hledání v maticích.

1. Vygenerujte náhodně reálné celočíselné matice $\boldsymbol{A}, \boldsymbol{B}$ rozměru 3×4 s prvky v intervalu -5 až 5.
2. Vytvořte textovou matici `T` téhož rozměru, kde znak plus(+), resp. minus(-) na (i, j)-té pozici indikuje, ze $A(i, j)$ a $B(i, j)$ jsou nenulová čísla téhož, resp. různého znaménka; je-li některý z prvků $A(i, j)$ nebo $B(i, j)$ nulový uložíme znak tečky(.).
3. Jedním složeným příkazem zjistěte, zda součet záporných prvků v $\boldsymbol{A}$ je v absolutní hodnotě menší než součet kladných prvků v $\boldsymbol{B}$.
4. Zjistěte celkový počet nulových prvků v maticích $\boldsymbol{A}$ a $\boldsymbol{B}$ dohromady.
5. Vytvořte matici $\boldsymbol{C}$ stejného rozměru jako $\boldsymbol{A}$, která ma na (i, j)-té pozici rozdíl řádkového a sloupcového indexu. Pomocí této matice rozložte matici $\boldsymbol{A}$ na její diagonálovou, dolní trojúhelníkovou a horní trojúhelníkovou část.
6. Zjistěte, zda v $\boldsymbol{A}$ existuje nulový prvek.
7. Zjistěte odmocniny vlastních čísel matice $\boldsymbol{A}\boldsymbol{A}^T$ a porovnejte výsledek s výstupem příkazu `svd(A)` — viz též `help svd`.
8. Nalezněte řádkové a sloupcové pozice kladných prvků v $\boldsymbol{A}$.
9. Zopakujte příklad 8 s maticí $\boldsymbol{B}$, kde využijete jednorozměrného indexovaní a všechny kladné prvky přepíšete jejich desetinásobkem.
10. Zkontrolujte existenci proměnné `A` před a po vymazáni matice $\boldsymbol{A}$ z paměti.

Lekce 6: práce s textovými řetězci, znakové konverze čísel, vyhodnocení příkazu zadaného textovým řetězcem, zpracování chybových hlášení, operace s datumem a časem.

1. Zajistěte, abyste na konci tohoto cvičení mohli zjistit Váš celkový čas spotřebovaný na řešení a porovnat jej s čistým časem procesoru.
2. Uložte následující dvě věty do matice `T` na samostatné řádky: *Hlavní náplní třídenního pobytu byly diskuse. Účastnilo se ho 110 středoškoláků.*
3. Věty z `T` uložte bez koncových mezer do řádkových vektorů `v1` a `v2`.
4. Zjistěte počet mezer, písmen a číslic ve `v1` a `v2`.
5. Spojte věty `v1` a `v2` za sebe do jednoho řádkového vektoru `t` tak, aby za tečkou prvé věty následovala mezera.
6. V celém textu `t` vyhledejte výskyt slabiky *la*.
7. V `t` přidejte za slova *byly* a *ho* dvě mezery.
8. Najděte pozice prvého písmena každého slova.
(Návod: porovnejte `t` s řetězcem `t` posunutým o 1 pozici vpravo)

9. Počáteční písmena slov nalezená v předchozím příkladu nahraďte velkými písmeny.
10. Jednotlivá slova v `t` uložte postupně do proměnných `slovo1`, `slovo2`, atd.
11. Vypište datum a čas, kdy jste cvičení řešili, kolik minut Vám to trvalo a jaký čas byl spotřebován procesorem.

Lekce 7: funkční a dávkové m-soubory, větvení běhu programu, cykly v programu.

Podle následujících pokynů vytvořte funkční m-soubor `cramer.m`. pro řešení systému lineárních rovnic $\boldsymbol{Ax} = \boldsymbol{b}$ Cramerovým pravidlem:

1. V samostatném okně spusťte editor se souborem `cramer.m`.
   Jako vzorovou kostru procedury lze použít soubor `matsem7f.m` z adresáře `MATLABtutorial`.
2. Vytvořte úvodní deklaraci a nápovědu pro tři vstupní parametry `A,b,e` a tři vstupní parametry `x,e1,e2` s následujicím významem:
   - `A` ... čtvercová matice soustavy rozměru $n \times n$ (povinný parametr)
   - `b` ... matice m pravých stran rozměru $n \times m$ (nepovinný parametr při výpočtu inverze matice `A`)
   - `e` ... tolerance (malé číslo) pro detekci skoro singulární matice (nepovinný parametr, implicitní hodnota `10*eps`)
   - `x` ... matice rozměru $n \times m$, jejíž sloupce po řadě odpovídají řešením soustav `A*x(:,j)=b(:,j)`, $j = 1, ..., m$
   - `e1` ... maximální odchylka (v absolutní hodnotě) od řešení spočteného pomocí násobení zleva inverzní maticí
   - `e2` ... maximální odchylka (v absolutní hodnotě) od řešení spočteného pomocí operátoru levého dělení
3. Popis činnosti funkce:
   a) Program zpracujte tak, aby zadání prázdné matice za nepovinný parametr mělo stejný efekt, jako jeho vynechání.
   b) Pokud byla zadána pravá strana, pak `x` je řešení maticové rovnice `A*x=b`.
   c) Pokud nebyla zadána pravá strana, pak `x` je matice inverzní k `A`.
   d) Odchylky `e1`, resp. `e2` počítejte jen, když tyto parametry byly při volání funkce skutečně požadovány.
4. Zastavte program s výpisem chybových hlášení v těchto případech:
   a) Počet vstupních parametrů je menší než počet povinných parametrů nebo větší než počet deklarovaných parametrů.
   b) Matice `A` není čtvercová nebo je prázdná.
   c) Matice `b` nemá stejný počet řádků jako `A`.
   d) Matice `A` je skoro singulární, tj. její determinant je v absolutní hodnotě menší než tolerance `e`.
5. Program řádně otestujte pro všechny kombinace vstupních a vystupních parametrů:
   a) V případě spolehlivě regulární matice `A` (například ortogonální), by odchylky `e1` a `e2` měly být zanedbatelné.

b) V případě skoro singulární matice: tu konstruujte pomocí regulární matice ad a) ortogonalizací (příkaz `orth`) následovanou postupným nahrazováním některého sloupce jeho stále se zmenšujícím skalárním násobkem zvětšeným o nějakou lineární kombinaci ostatních k němu kolmých $n-1$ sloupců. Dostanete tak posloupnost matic blížících se k singulární matici, na níž můžete monitorovat chování chyb `e1` a `e2` a detekci singulárnosti při různých tolerancích `e`.

Lekce 8: interakce se spuštěným programem, řízení výpisů na obrazovku, prostředky pro ladění správné funkce programů.

Na funkční proceduře `cramer` zpracované v předchozím cvičení vyzkoušejte různé způsoby ladění.

Lekce 9: grafický subsystém, základy kreslení dvourozměrných a trojrozměrných grafů.

1. Nakreslete na intervalu $[0, 4\pi]$ graf funkce $tsin(at)$ pro hodnotu parametru $a = 1$.
2. Graf vhodně otitulkujte a popište osy.
3. Zmeňte u grafu dodatečně rozsah os, barvu, typ a tloušťku čáry.
4. Nakreslete jedním příkazem do tychž os m grafu téže funkce pro hodnoty parametru $a = 1, \ldots, m$ při předem zadaném $m = 4$.
5. Totéž, ale s postupným přikreslováním grafu do téhož okna.
6. Totéž, ale do 6-ti samostatných obrázků ve dvou řadách po třech.
7. Nakreslete 3 periody cykloidy s poloměrem kružnice $r = 2$;
   Cykloida je křivka zadaná parametricky:
   $x(t) = r(t - \sin(t)), y(t) = r(1 - \cos(t))$.
8. Obrázek s cykloidou pojmenujte *Cykloida* namísto implicitního jména *Figure No.*
9. Vykreslete na intervalu $[-5, 5]$ graf tzv. Dirichletovy funkce:
   $y(x) = \sin(a\pi x)/(a \sin(\pi x))$ pro hodnotu parametru $a = 5$;
   pro vykresleni užijte ekvidistantní 101-bodovou síť.
10. Graf se Vám asi moc nelíbí. Důvodem je, že $y(x)$ je periodická funkce s periodou 1, přičemž pro celočíselné hodnoty x dostáváme výrazy typu 0/0. L'Hospitalovým pravidlem spočtěte správné hodnoty a graf opravte.
11. Do samostatných obrázků vykreslete grafy níže uvedených funkcí $z = f(x, y)$, přičemž pro každou použijte jinou metodu pro vykreslení:
    a) $f(x, y) := |x - y|$ pro x, y z oblasti $[-5, 5] \times [-6, 6]$.
    b) $f(x, y) := 1/(1 + (x + y)^2)$ pro x, y z oblasti $[-5, 5] \times [-5, 5]$.
    c) $f(x, y) := xe^{-x^2-y^2}$ pro x, y z oblasti $[-2, 2] \times [-2, 2]$.
    d) $f(x, y) := (x^2+y^2)(e^{-(x-y)^2}+e^{-(x+y)^2})+80/(1+x^2+y^2)$ pro x, y z oblasti $[-5, 5] \times [-5, 5]$.
    e) V ploše funkce ad d) zvýrazněte barevnou tlustou čarou trajektorii odpovídající řezu rovinou kolmou na rovinu $x - y$ a protínající ji v přímce procházející body $[0, 4]$ a $[4, 0]$.

f) Meňte u obrázků různými způsoby detaily grafického provedení (úhel pohledu, směr a barevné složení světelného zdroje apod.).
g) Prohlédněte si další firemní demonstrace grafiky.

Lekce 10: elementární funkce a jejich průběhy, příkazy vztahující se k náročnějším tématickým okruhům z maticové a polynomiální algebry.

1. Jedním příkazem `fplot` vykreslete průběh dvou funkcí: sinus s amplitudou 8 a kosinus s amplitudou 6 na intervalu $[0, 2\pi]$.
2. Do proměnné `r` spočtěte kořeny polynomu $P(x) = 2x^3 - 12x^2 - 144x - 54$.
3. Pomocí `r` sestavte polynom $p(x)$ s týmiž kořeny; vysvětlete vztah mezi oběma polynomy $p(x)$ a $P(x)$.
4. Nalezněte matici `A` asociovanou s polynomem $p(x)$ a ověřte, že $p(x)$ je skutečně jejím charakteristickým polynomem.
5. Vytvořte náhodnou matici `X` rozměru 2×3 a sestavte matici `Y` téhož rozměru, která je vyhodnocením polynomu $p(x)$ na `X`, tj. $Y(i,j) = p(X(i,j))$.
6. Ověřte platnost Cayley-Hamiltonovy věty pro matici `A`, tj. `A` musí být v maticovém smyslu kořenem svého charakteristického polynomu (použijte příkaz `polyvalm`).
7. Nalezněte polynom $c(x)$, který je součinem polynomů $a(x) = x^3+2x^2+3x+4$ a $b(x) = 10x^2 + 20x + 30$.
8. Proveďte dělení polynomu $d(x) = c(x) + 2(x-1)$ polynomem $b(x)$ a určete současně nejen podíl, ale i zbytek po dělení.
9. Nalezněte rozklad na parciální zlomky racionální lomené funkce $d(x)/b(x)$ nejprve nad tělesem komplexních čísel a odtud pak spočtěte jeho tvar nad tělesem reálných čísel.
10. Nalezněte interpolační polynom funkce $\sin(x)$ procházející jejími body pro $x = 0, \pi/2, \pi, 3\pi/2, 2\pi$; oba průběhy znázorněte graficky a do samostatného obrázku vykreslete průběh chybové funkce (rozdíl mezi oběma funkcemi).

Literatura

[1] The MathWorks, Inc., 24 Prime Park Way, Natick, Mass. 01760. *MATLAB. High-Performance Numeric Computation and Visualization Software – Reference Guide*, October 1992.

[2] The MathWorks, Inc., 24 Prime Park Way, Natick, Mass. 01760. *MATLAB. High-Performance Numeric Computation and Visualization Software – User's Guide*, August 1992.

[3] Arnošt Svoboda a Leonard Walletzký. *Informatika pro ekonomy: Základy práce v Matlabu.* Ekonomicko-správní fakulta Masarykovy univerzity, Brno, 2001. Skripta.

[4] Blanka Heringová a Petr Hora. *MATLAB 4.0 – Popis grafického systému, grafická nadstavba a práce se soubory*, volume I–II. Institut technologie a spolehlivosti, Západočeská univerzita, Plzeň, červen 1994. Skripta.

[5] Blanka Heringová a Petr Hora. *MATLAB pro Windows. Díl I. – Práce s programem*, volume I. Institut technologie a spolehlivosti, Západočeská univerzita, Plzeň, 1995. Skripta.

[6] P. Lancaster. *Theory of matrices.* Academic Press, New York, 1969.

[7] F. R. Gantmakher. *Teorija matric.* Nauka, Moscow, 1988.

[8] E. Pärt-Enander, A. Sjöberg, Bo Melin, and P. Isaksson. *The MATLAB Handbook.* Addison–Wesley, Harlow, England, 1996.