

Elektronický manuál pro program FreeMat

Electronic manual for FreeMat

Pavol TURAN

Bakalářská práce
2008



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2007/2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Pavol TURAN**

Studijní program: **B 3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Téma práce: **Elektronický manuál pro program FreeMat**

Zásady pro vypracování:

1. Provedte literární rešerši na zadané téma.
2. V teoretické části práce popište základy ovládání programového prostředí FreeMat, zadávání příkazů a možnosti programu v oblasti práce s maticemi a oblasti vizualizace výsledků.
3. Vytvořte elektronický manuál k programovému prostředí FreeMat ve formě HTML stránek. Manuál bude sloužit pro výukové účely a bude obsahovat popis ovládání programu, základy práce s programem, rozdělení příkazů do jednotlivých skupin.
4. U jednotlivých vybraných příkazů z daných skupin uveďte jejich podrobný popis a jejich použití demonstруйте na vhodně zvolených příkladech.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Freemat [online]. [2003], 24 September 2007 [cit. 2008-01-21]. Dostupný z WWW: http://freemat.sourceforge.net/wiki/index.php/Main_Page.
2. BASU, Samit. FreeMat v3.4 Documentation. [s.l.] : [s.n.], 2007. 561 s. Dostupný z WWW: <http://freemat.sourceforge.net/FreeMat-3.5.pdf>.
3. KOSEK, Jiří. Html -- tvorba dokonalých stránek: podrobný průvodce. Ilustroval Ondřej Tůma. 1. vyd. Praha: Grada, 1998. 291 s. ISBN 80-7169-608-0.

Vedoucí bakalářské práce:

Ing. Petr Navrátil, Ph.D.

Ústav řízení procesů

Datum zadání bakalářské práce:

20. února 2008

Termín odevzdání bakalářské práce:

5. května 2008

Ve Zlině dne 20. února 2008

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Cieľom bakalárskej práce je oboznámiť čitateľa s programovým prostredím FreeMat. Zároveň má za úlohu popísať základy práce s týmto programovým prostredím, objasniť možnosti použitia matematických a štatistických funkcií, práce s maticami, vektormi a reťazcami, tvorby vizualizácií a základov programovania. Spracovanie manuálu je vo forme html stránok a vo forme elektronického pomocníka vo formáte *.chm. Členenie práce a štruktúra manuálu sú tvorené s prihliadnutím na jednoduchosť a zrozumiteľnosť. Súčasťou práce je podrobný popis jednotlivých funkcií a taktiež množstvo doplňujúcich a vysvetľujúcich ilustračných príkladov.

Kľúčové slová: FreeMat, funkcia, matica, vektor, vizualizácia, programovanie, manuál

ABSTRACT

The goal of bachelor thesis is to apprise the reader of the software environment of FreeMat. At the same time it has a goal to describe basics of work with this software environment. It makes also clear possibilities of using mathematic and statistic functions, work with matrices, vectors and strings, generating visualization and basics of programming. Execution of this reference is in form of html pages and in form of electronic help in *.chm format. Segmentation of thesis and structure of manual are composed with regards to simplicity and comprehensibility. Included are detailed description of particular functions and also a number of supplemental and declarative illustration examples.

Keywords: FreeMat, function, matrix, vector, visualization, programming, user manual

Touto cestou by som sa chcel poďakovať vedúcemu mojej bakalárskej práce Ing. Petrovi Navrátilovi, Ph.D. za čas venovaný mojej práci, za tvorivé pripomienky a profesionálne vedenie pri tvorbe mojej bakalárskej práce.

Čestne prehlasujem, že som bakalársku prácu vypracoval samostatne a použítú literatúru som citoval. V prípade publikácie výsledkov, ak je to uvoľnené na základe licenčnej zmluvy, budem uvedený ako spoluautor.

V Zlíně

.....

Podpis bakalára

OBSAH

ÚVOD	8
I TEORETICKÁ ČASŤ	9
1 ÚVOD	10
2 ČO JE FREEMAT	12
3 INŠTALÁCIA FREEMAT	14
3.1 INŠTALÁCIA PRE LINUX	14
3.2 INŠTALÁCIA PRE WINDOWS	14
3.3 INŠTALÁCIA PRE MAC OS X	15
3.4 INŠTALÁCIA ZO ZDROJOVÉHO KÓDU	15
4 PROSTREDIE FREEMAT	16
4.1 SPUSTENIE FREEMAT	16
4.2 UKONČENIE PRÁCE S FREEMAT	16
4.3 PRACOVNÁ PLOCHA	16
4.4 PRÍKAZOVÝ RIADOK	18
4.5 POMOCNÍK	20
4.6 PREMENNÉ A FUNKCIE	20
4.7 ŠPECIÁLNE KONŠTANTY	21
4.8 OPERÁTORY	22
4.9 SKUPINY FUNKCIÍ	23
5 MATICE	25
5.1 VYTVÁRANIE MATÍC	26
5.2 ZÁKLADNÉ OPERÁCIE S MATICAMI	29
5.2.1 Transpozícia	29
5.2.2 Operátorové aritmetické operácie	30
5.2.3 Funkcie cross, dot, sum, size a det	32
5.2.4 Ďalšie funkcie pre prácu s maticami	33
5.3 INDEXOVANIE	36
5.4 POLYNÓMY	38
5.4.1 Funkcie pre prácu s polynómami	39
6 MATEMATICKÉ FUNKCIE	41
6.1 TRIGONOMETRICKÉ FUNKCIE	41
6.2 CYKLOMETRICKÉ FUNKCIE	42
6.3 ĎALŠIE MATEMATICKÉ FUNKCIE	43
7 ELEMENTÁRNE FUNKCIE	45
8 PRÁCA S REŤAZCAMI	47

8.1	VYTVORENIE POĽA REŤAZCOV.....	47
8.2	FUNKCIE PRE ÚPRAVU REŤAZCOV.....	48
8.3	TESTOVACIE FUNKCIE.....	48
8.4	KONVERZNÉ FUNKCIE.....	49
8.5	ĎALŠIE FUNKCIE PRE PRÁCU S REŤAZCAMI.....	50
9	VIZUALIZÁCIA.....	52
9.1	OBJEKT FIGURE.....	52
9.2	OBJEKT AXES.....	53
9.3	POTOMKOVIA OBJEKTU AXES	53
9.4	ZÁKLADNÉ TYPY VYKRESLENÍ.....	54
10	ĎALŠIE MOŽNOSTI PROSTREDIA	59
II	PRAKTICKÁ ČASŤ.....	60
11	ELEKTRONICKÝ MANUÁL.....	61
	ZÁVER	63
	ZOZNAM POUŽITEJ LITERATÚRY.....	65
	ZOZNAM POUŽITÝCH SYMBOĽOV A SKRATIEK.....	67
	ZOZNAM OBRÁZKOV	68
	ZOZNAM TABULIEK	69
	ZOZNAM PRÍLOH	70

ÚVOD

V praxi sa človek často ocitne v situácii, keď potrebuje výkonný softvérový nástroj pre matematické, štatistické, inžinierske a vedecké výpočty, simuláciu, vizualizáciu a vyhodnocovanie. V súčasnosti je k dispozícii veľké množstvo programov umožňujúcich vykonávať vyššie spomenuté činnosti. Najkvalitnejšie a najobsiahlejšie programy poskytujúce široké spektrum funkcií sú však licencované a finančne náročné vývojové prostredia ako Matlab, Mathematica, Maple a i.. Alternatívou ku komerčným systémom sú voľne šíriteľné balíky. Vzhľadom na to, že sa jedná o neziskové projekty, sú tieto vývojové prostredia buď jednostranne zamerané alebo neposkytujú rozsah funkcií porovnateľný s licencovanými produktmi. Avšak i v tomto segmente sa nájdu projekty na vysokej úrovni, ktoré vo väčšine prípadov dokážu adekvátne nahradiť platené vývojové prostredia. Jedným zo všestranne zameraných a komplexne vypracovaných voľne šíriteľných vývojových prostredí je FreeMat, ktorý podobne ako iné Open Source projekty vznikli práve ako odpoveď na komerčné vývojové balíky. Ďalšími voľne dostupnými prostrediami sú Octave, Scilab, R, SciPy, Maxima, Euler, Math GV, MuPAD Light, Tanagra a i.. [7]

Tvorcom FreeMatu je vývojár a programátor Samit Basu, ktorý spolu s množstvom prispievateľov z celého sveta vytvoril tento Open Source projekt šíriteľný pod GPL licenciou. Jedná sa o interpretované maticovo orientované vývojové prostredie pre inžinierske a vedecké aplikácie. Poskytuje vizualizáciu, vyhodnocovanie, numerické výpočty, paralelné programovanie a spracovanie dát. [6] Okrem množstva podporovaných funkcií a bezplatného šírenia je ďalšou výhodou tohto prostredia multiplatformita. Medzi podporovanými platformami sú Windows, Linux a MAC OS X. [4]

Keďže sa jedná o nekomerčný projekt, jeho podpora nie je porovnateľná s platenými projektmi a pre mnohých potenciálnych užívateľov je odstrašujúcim faktorom neexistencia návodov alebo príručiek v zrozumiteľnom jazyku. Jediným dostupným užívateľským manuálom je totiž dokumentácia v angličtine vytvorená tvorcom programu. Z tohto dôvodu je cieľom bakalárskej práce priblížiť prostredie FreeMat budúcim používateľom a zoznámiť ich s prácou v tomto prostredí. Súčasťou práce je vytvorenie elektronického manuálu podrobne popisujúceho základné funkcie a ovládacie prvky programu. Manuál je vytvorený s prihliadnutím na užívateľa začiatočníka a je doplnený o množstvo príkladov z oblasti práce s maticami a vektormi, polynómami, reťazcami, vizualizácie a základov programovania.

I. TEORETICKÁ ČASŤ

1 ÚVOD

Program FreeMat nie je vo svojej oblasti určenia jedinečný a má mnoho priamych – voľne šíriteľných a nepriamych – licencovaných a platených konkurentov v zmysle rovnako zameraných vývojových prostredí. Obsahom tejto kapitoly bude priblížiť najznámejšie z nich.

Za vzorové prostredie FreeMat-u bol zvolený spoplatnený Matlab. Dnes je Matlab považovaný za štandardný a univerzálny výpočtový nástroj v oblasti techniky, vedy a výskumu. Určený je pre platformy Linux, Mac OS X, Solaris a Windows. Matlab je základná výpočtová platforma, ktorá sa dá rozširovať o ďalšie funkcie z rôznych oblastí pomocou rozširujúcich balíkov – toolboxov. [9] Integruje v sebe výpočty, vizualizáciu a programovanie do jedného použiteľného rozhrania. Typické oblasti použitia Matlabu sú:

- Inžinierske výpočty
- Vývoj algoritmov
- Modelovanie, simulácia a tvorba prototypov
- Analýza dát a ich vizualizácia
- Inžinierska grafika
- Vývoj aplikácii vrátane tvorby grafického užívateľského prostredia
- Podpora pri vyučovaní technických predmetov [10]

Ďalším rovnako zameraným prostredím je Scilab. Scilab bol vyvinutý v Scilab Group INRIA - Rocquencourt Metalau Project a jedná sa o integrované prostredie na 2D a 3D grafiku, matematické a technické výpočty, modelovanie a simuláciu, spracovanie signálov, analýzu a vizualizáciu dát a množstvo iného. Jedná sa o nekomerčnú voľne šíriteľnú verziu určenú pre operačné systémy UNIX – Linux, HP – UX, MAC OS X a Windows. Podobne ako FreeMat vychádza z už spomínaného Matlabu. [11]

Iným, dá sa povedať jedným z prvých a najrozšírenejších klonov prostredia Matlab, je Octave. Služi k algebrickým i numerickým výpočtom, riešeniu rovníc a sústav rovníc, vrátane diferenciálnych. [8] Octave patrí medzi Open Source voľne šíriteľné programy a podobne ako predchádzajúce spomenuté sa vyznačuje multiplatformitou. Je určený pre operačné systémy Linux, MAC OS X, Sun Solaris, OS/2 alebo eComStation i Windows. [12]

Pri informovaní o najznámejších a najkomplexnejších prostrediach nemožno zabudnúť na softvér Mathematica. Zahŕňa množstvo funkcií a algoritmov z oblastí všeobecnej algebry, kombinatoriky, maticového a tenzorového počtu, štatistiky a regresnej analýzy, riešenia rovníc a ich sústav (algebraických i diferenciálnych), diferenciálneho a integračného počtu a i. [13] Rovnako slúži na riešenie lineárnych optimalizačných úloh, vizualizácie problémov a pod. Nevýhodou je finančná náročnosť jej obstarania.

Vo vymenovávaní vývojových prostredí pre inžinierske a vedecké úlohy by bolo možné ešte pokračovať, avšak to nie je náplňou tejto bakalárskej práce. Preto boli spomenuté len najvýznamnejšie z nich pre vytvorenie obrazu o ponúkaných riešeniach v tejto oblasti. Existuje mnoho iných, z ktorých niektoré boli spomenuté v úvode práce, avšak určite sa nájdu i také, ktoré neboli spomenuté z dôvodu veľkého množstva softvérových prostredí určených na riešenie úloh technického, matematického a iného charakteru.

2 ČO JE FREEMAT

FreeMat je moderné vývojové prostredie pre inžinierske a vedecké výpočty. Jedná sa o voľne šíriteľnú verziu, ktorá vznikla ako odpoveď na spoplatnený balík Matlab. [8] Za jeho vývojom stojí Samit Basu, ktorý spolu s pomocou množstva prispievateľov roztrúsených po celej planéte vytvoril počas 4 rokov vývoja prostredie do podoby, v akej ho poznáme dnes. [4]

Jedná sa o prostredie určené nielen na numerické výpočty, ale taktiež podporujúce aj programovanie. Navyše podporuje mnoho funkcií z prostredia Matlabu a niektoré IDL (Interactive data language) funkcie. [5] Aktuálna verzia podporuje približne 95% funkcií Matlabu a neustále sa pracuje na zvýšení kompatibility a každá nová verzia zvyšuje počet podporovaných vlastností. Čo sa týka kompatibility s IDL, podporuje základné spôsoby ako predávanie odkazom alebo kľúčové slová. FreeMat je taktiež určený na vedecké modelovanie a spracovanie dát. Program je distribuovaný pod licenciou GNU/GPL ako Open Source. V súčasnosti je vo verzii 3.6, ktorá vyšla 17.3.2008. Predchádzajúce verzie však boli vydávané pod MIT licenciami.

FreeMat podporuje všetky bežne dostupné platformy, aktuálne sú to: Windows, Linux a MAC OS X. Funkčnosť v iných UNIXových prostrediach ako sú IRIX/SOLARIS nie je zaručená. Inštalčné balíky sú k dispozícii k stiahnutiu na domovskej stránke prostredia: <http://freemat.sourceforge.net/> .

Podporované vlastnosti prostredia FreeMat:

- Spracovanie N rozmerných polí (štandardne N obmedzené na 6).
- Podpora 8, 16, 32 a 64 bitových celočíselných typov (so znamienkom a bez znamienka), 32 a 64 bitové typy s pohyblivou desatinnou čiarkou, 64 a 128 bitové komplexné typy.
- Vstavaná aritmetika pre spracovanie všetkých podporovaných dátových typov.
- Podpora pre riešenie systémov lineárnych rovníc pomocou operátorov delenia.
- Rozklad na vlastné hodnoty a singulárne hodnoty.
- Plná podpora riadiacich štruktúr vrátane for, while, break, continue a iných.
- 2D vykresľovanie a zobrazovanie.

- Plná podpora nehomogénnych polí (tzv. cell arrays).
- Podpora ľubovoľne veľkých FFT.
- Podpora predávania odkazom.
- Podpora kľúčových slov.
- Podpora riedkych matíc.
- Ukazatele funkcií.
- Triedy, preťaženie operátorov.
- 3D vykresľovanie a vizualizácia pomocou OpenGL.
- Súbežné spracovanie v MPI. [3],[4]

3 INŠTALÁCIA FREEMAT

Ako už bolo spomenuté v úvode a v kapitole Čo je FreeMat, jedná sa o multiplatformné prostredie určené pre prácu pod operačnými systémami Linux, Windows i MAC OS X.

3.1 Inštalácia pre Linux

Pre Linux je FreeMat poskytovaný ako binárna inštalácia. Pre inštaláciu je potrebné stiahnuť z domovskej stránky programu súbor a rozpakovať ho:

```
tar xvfz FreeMat-3.6-Linux-Binary.tar.gz
```

FreeMat je možné spustiť priamo bez väčšej námahy a to nasledovne:

```
FreeMat-3.6-Linux-Binary/Contents/bin/FreeMat
```

Predchádzajúce spustí FreeMat ako X – window aplikáciu. V prípade, že chceme pracovať v príkazovom riadku je potrebné použiť **nogui** značku:

```
FreeMat-3.6-Linux-Binary/Contents/bin/FreeMat -nogui
```

V prípade, že nechceme vôbec použiť grafiku je potrebné doplniť **noX** značku:

```
FreeMat-3.6-Linux-Binary/Contents/bin/FreeMat -noX
```

Pre uľahčenie je možné pridať FreeMat do vašej cesty. Existuje viacero spôsobov ako si prispôbiť cestu v závislosti od používaného „shell“. Jednotlivé spôsoby sú opísané v pôvodnej originálnej dokumentácii (ďalej len „OD“) umiestnenej na domovskej stránke produktu a taktiež v elektronickom manuáli (ďalej len „EM“), ktorý je súčasťou tejto práce. V prípade, že pripravený binárny balíček nebude pracovať správne pod danou distribúciou Linuxu je potrebné použiť zostavenie zo zdrojového kódu. Jedná sa však o pomerne zložitý proces, ktorý sa odporúča iba užívateľom ovládajúcim inštaláciu zo zdrojového kódu, postup je opäť možno nájsť v EM a OD.

3.2 Inštalácia pre Windows

Inštalácia FreeMat prebieha pre Windows pomocou binárneho inštaláčného programu. Pre inštaláciu je potrebné stiahnuť z domovskej stránky prostredia FreeMat nastavovací program:

```
FreeMat-3.6-Setup.exe
```

Po stiahnutí a otvorení stačí nasledovať inštrukcie počas priebehu inštalácie a nastaviť cestu pomocou nástroja **pathtool**.

3.3 Inštalácia pre MAC OS X

Šírenie FreeMat pre MAC OS X prebieha v podobe aplikačného balíčka. K inštalácii je potrebné stiahnuť tento balíček, ktorý je v podstate súbor so skomprimovaným obrazom disku:

`FreeMat-3.6.dmg`

Po stiahnutí je potrebné obraz disku namontovať a následne skopírovať prostredie **FreeMat-3.6** na určené miesto. Ďalej je možné aplikáciu štandardne spustiť. Pre nastavenie FreeMat cesty je opäť potrebné použiť nástroj **pathtool**.

3.4 Inštalácia zo zdrojového kódu

FreeMat je možné zostaviť zo zdrojového kódu. Jedná sa o zložitejší postup, ktorý je doporučený skúseným užívateľom, resp. jeho použitie je nevyhnutné, pretože štandardný postup inštalácie pre danú platformu spomenutý vyššie zlyhal. Zostavenie FreeMat zo zdrojového kódu je možné použiť pre všetky platformy. Postup inštalácie je opísaný v EM a OD.

4 PROSTREDIE FREEMAT

FreeMat bol vytvorený ako voľne šíriteľná odpoveď na komerčné vývojové prostredie Matlab. Tomuto faktu zodpovedá aj pracovné prostredie, ktoré je navrhnuté tak, aby bolo v čo najväčšej možnej miere podobné práve aplikácii Matlab. Už po spustení je vidno podobnosti medzi oboma prostrediami a taktiež samotná práca a filozofia FreeMat je založená na filozofii svojho komerčného vzoru.

4.1 Spustenie FreeMat

V graficky orientovaných operačných systémoch spustíme aplikáciu dvojklikom na ikonu na ploche, poprípade výberom z ponuky programov. V prípade spúšťania pomocou príkazového riadka systému je potrebné zadať názov aplikácie, resp. spúšťacieho súboru.

4.2 Ukončenie práce s FreeMat

Existuje viacero spôsobov ukončenia práce s prostredím FreeMat, jednotlivé spôsoby sú si rovnocenné a záleží len na používateľovi, ktorý si vyberie.

Štandardným ukončením je výber záložky File z roletového menu a následné kliknutie na položku Quit.

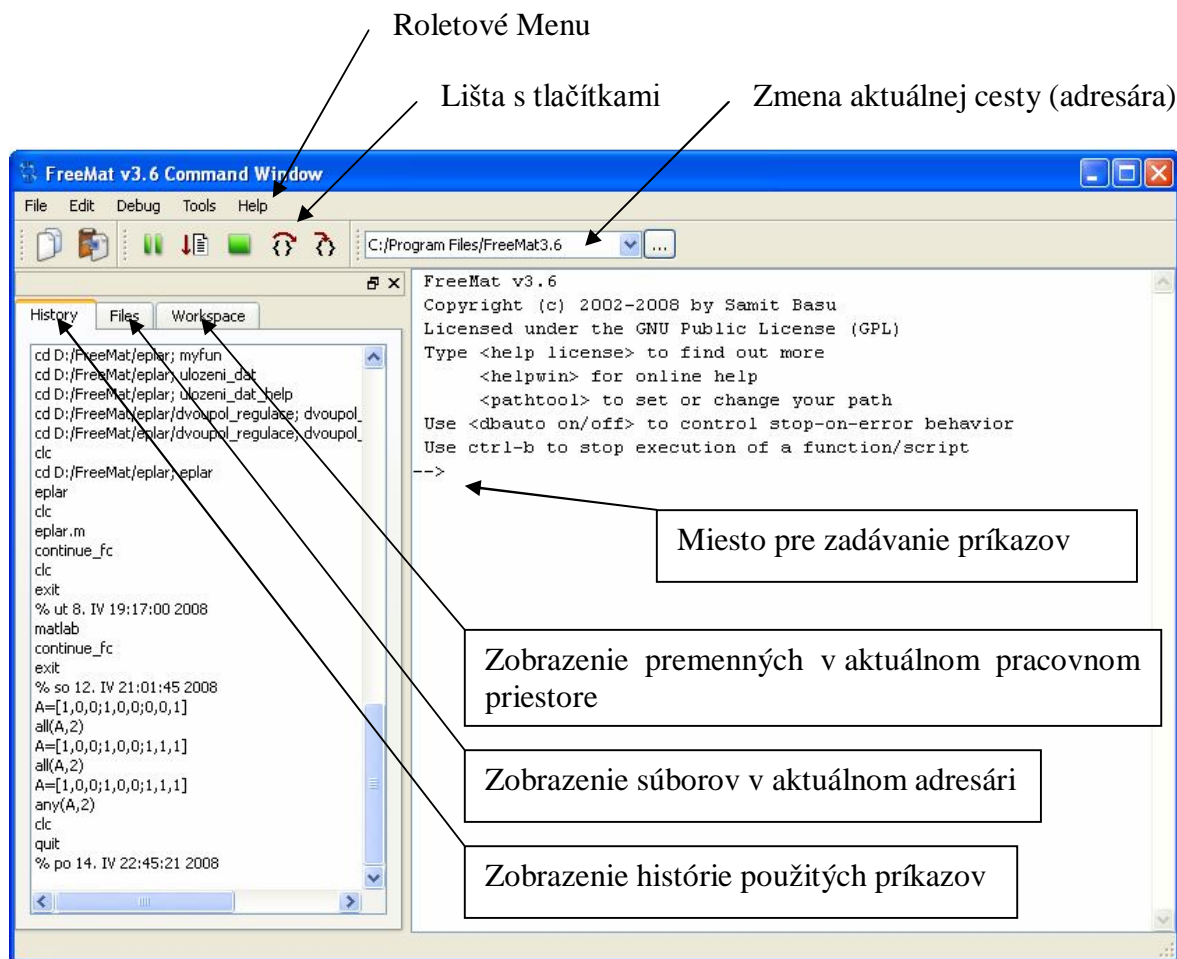
Taktiež je možné ukončenie programu pomocou klávesovej skratky Ctrl+Q.

Samozrejmosťou je ukončenie práce pomocou príkazu zadaného do príkazového riadka programu. K dispozícii máme dva samostatné príkazy, ktoré sú si však ekvivalentné: **exit**, **quit**.

Oba príkazy slúžia pre okamžité ukončenie aplikácie FreeMat.

4.3 Pracovná plocha

Po spustení programu sa nám otvorí okno aplikácie rozdelené do niekoľkých častí podľa vzoru Matlab. Jednotlivé časti sú prehľadne umiestnené a neobsahujú nadbytočné funkcie vedúce k zhoršeniu prehľadnosti pri práci. Pracovná plocha zobrazená nižšie na Obr. 1 je štandardná plocha so štandardným rozložením a veľkosťami jednotlivých častí. Vzhľad pracovného okna je možné zmeniť pomocou zmeny veľkosti a umiestnenia jednotlivých blokov tvoriacich pracovnú plochu.



Obrázok 1: Pracovná plocha FreeMat

Ako je možno vidieť na Obr. 1 základnou časťou je priestor pre zadávanie príkazov. Okrem zadaných príkazov sa po potvrdení príkazu stlačením klávesy Enter zobrazujú i výsledky daného príkazu, ak nie je potlačený výpis výsledkov znakom (;) bodkočiarka, ukončujúcim zadaný príkaz. Potlačenie výpisu neznamená nevykonanie príkazu. Príkaz je štandardne vyhodnotený.

```
--> a=b+c
--> a=b+c;
```

V prvom prípade dôjde k výpisu výsledku priradenia obsahu premenných b, c do premennej a. V druhom prípade je však výpis potlačený ukončujúcim znakom (;).

Možnosť zadania príkazu je signalizovaná úvodnou postupnosťou znakov -->. Podrobnejší popis príkazového riadka a práce s ním ako aj zadávanie príkazov je predmetom nasledujúcich kapitol.

Ďalšou časťou pracovnej plochy je roletové menu, ktoré obsahuje základné podmenu potrebné pre prácu v prostredí FreeMat. Pomocou jednotlivých položiek v menu je možné

uložit' prepis časti pre zadávanie príkazov a výpis výsledkov do súboru, ďalej nám umožňuje základné operácie editácie, ladenia programov, prispôsobenia vzhľadu okna aplikácie a nakoniec informácie o programe a spustenie pomocníka.

Pod roletovým menu sa štandardne nachádza lišta s tlačítkami, ktorých funkcia je ekvivalentná ponúkaným možnostiam z roletového menu a to konkrétne zo záložiek Edit a Debug. Umožňujú tak rýchly a prehľadný prístup k funkciám ponúkaným voľbami z roletového menu.

V hornej časti pracovného okna sa nachádza priestor zobrazujúci aktuálny pracovný adresár a taktiež umožňujúci zmenu aktuálneho adresára.

Ak nie je nastavené inak, ľavú časť okna tvorí informačný panel s tromi záložkami umožňujúcimi zobrazenie histórie zadávaných príkazov, zobrazenie obsahu aktuálneho adresára a nakoniec výpis premenných, definovaných v aktuálnom pracovnom priestore, a ich vlastností.

4.4 Príkazový riadok

Načrtnutie práce s príkazovým riadkom obsahuje už predchádzajúca podkapitola, avšak len vo vzťahu rozvrhnutia pracovného okna. Pre prehľadnosť a celistvosť budú niektoré údaje spomenuté i v tejto podkapitole.

Po spustení programu sa v okne s príkazovým riadkom zobrazia základné informácie o programe a tipy pre ovládanie. Pod touto informačnou časťou sa nachádza samotný príkazový riadok očakávajúci zadanie príkazu, ktoré je indikované úvodnou postupnosťou znakov -->. Táto postupnosť sa objaví vždy, keď je možné zadávať príkazy. Zoznam podporovaných príkazov a výrazov je uvedený v OD a veľká väčšina z nich taktiež v EM. V prípade nepodporovaného príkazu alebo nekorektného zadania je po potvrdení príkazu stlačením klávesy Enter na nasledujúcom, riadku resp. riadkoch, zobrazené chybové hlásenie, vysvetľujúce príčinu vzniku chyby.

```
--> a
Error: Undefined function or variable a
--> b=c+
Error: Unexpected input
      a=b+
      ^
```

Príkazy zadávame priamo z klávesnice a je možné pohybovať sa po riadku s príkazom pomocou šípok vľavo a vpravo. Šípka hore slúži pre listovanie v histórii naposledy použitých príkazov. Okrem samotných použitých príkazov z minulosti zobrazuje aj tzv. časovú značku, ktorá indikuje dátum spustenia relácie, v ktorej boli jednotlivé príkazy použité.

Pri zadávaní si treba uvedomiť, že FreeMat je case-sensitive, čo znamená, že rozlišuje veľké a malé písmená. Z tohto dôvodu neukazujú názvy Premenna, premenna, PREMENNA a pod. na tú istú premennú, ale predstavujú tri rôzne premenné. Rovnaký princíp sa dodržiava i pri písaní príkazov. Zatiaľ čo príkaz `clc` vymaže obrazovku, zadanie príkazu `Clc` povedie k vypísaniu chybového hlásenia.

V prípade potreby je možné potlačiť výpis výsledku zadaného príkazu alebo výrazu ukončením daného výrazu bodkočiarkou (;). V prípade ukončenia príkazu bodkočiarkou sa príkaz vykoná, ale jeho výsledok sa nevypíše, v inom prípade dôjde po vykonaní príkazu k vypísaniu výsledku.

Ak nie je v príkaze alebo výraze uvedená výstupná premenná, výstup je uložený do tzv. ans premennej. Typ tejto premennej a dáta v nej obsiahnuté sú určené príkazom. Treba si však uvedomiť, že obsahuje vždy výsledok posledného správne vyhodnoteného príkazu. V prípade použitia ďalšieho príkazu bez určenia výstupnej premennej, dochádza k prepisu dát uložených v ans.

```
--> a=5;b=6;a+b
ans =
    11
--> b+4
ans =
    10
```

Predchádzajúci príklad znázorňuje použitie potlačeného výpisu v prípade priradenia hodnôt do premenných a, b a zároveň demonštruje využitie premennej ans ako i fakt, že dochádza k prepisu obsahu premennej ans.

Podobne ako iné prostredia, aj FreeMat umožňuje použitie komentárov. Komentáre sú uvedené znakom percento (%). Ľubovoľný text nachádzajúci sa v príkazovom riadku alebo riadku kódu za znakom % je braný ako komentár a nevyhodnocuje sa. Treba si však uvedomiť rozdiel, ak je znak % súčasťou príkazu, v tomto prípade dochádza k jeho vyhodnoteniu v závislosti od príkazu, v ktorom sa nachádza.

```
--> A=[1,2,3;4,5,-1]; % vytvorenie matice 3x3
```

Text za znakom % je považovaný za komentár a nevyhodnocuje sa

```
--> f='pekne'; printf('Dnes bude %s\n',f)
```

```
Dnes bude pekne
```

V tomto prípade je % súčasťou príkazu printf a pri vyhodnocovaní bude nahradené obsahom premennej f.

4.5 Pomocník

FreeMat rozoznáva dva základné druhy pomocníka. Prvý druh pomocníka zobrazí pomoc k ľubovoľnej funkcii podporovanej FreeMat priamo v okne s príkazovým riadkom. Funkcia **help** podporuje jeden argument - **topic**, ktorým je funkcia, pre ktorú chceme pomoc zobrazit'. Syntax funkcie je nasledovný: **help topic**.

V prípade skriptu je výsledkom volania funkcie help obsah poznámok, komentára na začiatku súboru. Ak súbor neobsahuje žiadne komentáre, zobrazí deklaráciu funkcie.

Druhý typ pomocníka je štandardný help browser alebo pomocník prehliadač, ktorý spustíme stlačením F1 pri aktívnom okne aplikácie FreeMat, zadaním názvu funkcie **helpwin** do príkazového riadka alebo výberom z roletového menu: Help → Online Manual. Vo všetkých prípadoch sa nám otvorí okno pomocníka skladajúce sa z viacerých častí. Vľavo máme na výber z dostupných funkcií, ku ktorým sa nám po dvojkliku na názov funkcie zobrazí samotný pomocník v pravej časti. Ten obsahuje názov funkcie, kategóriu, vysvetlenie použitia a príklady. Máme k dispozícii výber z abecedne zoradených funkcií, výber podľa tematických celkov a nakoniec štandardné hľadanie podľa názvu. Obsiahlejší popis práce s pomocníkom sa nachádza v EM.

4.6 Premenné a funkcie

Názov premennej nesmie začínať špeciálnym znakom :, &, ^, %, \$, #, @, !, ?, bodkou (.), pomlčkou (-), apostrofom alebo úvodzovkami ('', '''), \, akoukoľvek zátvorkou ([,], (,), {, }, <, >), =, ľubovoľnou číslicou (0-9), pomlčkou (-), znakom „rúra“ (|), operátorom (+, -, /, *). Názov môže obsahovať ľubovoľnú kombináciu malých a veľkých písmen anglickej abecedy. FreeMat rozoznáva v názvoch premenných, funkcií, príkazoch a výrazoch veľké a malé písmená a rozlišuje ich. Táto skutočnosť je opísaná v podkapitole 4.4 Príkazový riadok. Zoznam aktuálne definovaných premenných je možno získať výberom položky z roletového menu: Tools → Workspace Tool alebo kliknutím na

záložku Workspace v ľavej časti pracovného okna pri štandardnom rozmiestnení okna. Taktiež je možné získať informácie o všetkých alebo určených aktuálne definovaných premenných pomocou funkcie `who`. Po jej použití sa nám v okne s príkazovým riadkom zobrazia podrobnosti o premenných. Vo výpise sú pre každú vypísanú premennú zobrazené nasledovné vlastnosti: názov premennej, dátový typ, veľkosť, či sa jedná o globálnu alebo trvalú premennú. Pri zobrazení podrobností o premenných pomocou roletového menu a záložky Workspace je zobrazená aj hodnota danej premennej.

FreeMat podporuje dva formáty zadávania funkcie `who`:

```
--> who a1 a2 ...
--> who 'a1' 'a2' ...
--> who ('a1', 'a2', ...)
```

Všetky tri spôsoby zápisu sú ekvivalentné a poskytujú rovnaké výsledky a síce výpis podrobností k premenným `a1` až `aN` napísaným v liste za názvom funkcie `who`. Forma pre výpis podrobností o všetkých premenných je bezargumentová a má tvar: **who**.

Funkcie slúžia vo FreeMat pre uľahčenie práce pri opakovanom vyhodnocovaní a spracovaní dát. Jedná sa o prvok, ktorý nám po zavolaní vráti určitú hodnotu, nastaví premennú, prípadne volá inú funkciu a pod.. FreeMat podporuje viacero druhov funkcií a taktiež viacero spôsobov zápisu funkcií. Okrem zabudovaných funkcií sú podporované i užívateľom definované funkcie a skripty. Funkcie a skripty sú ukladané v súboroch s príponou `*.m`, s výnimkou tzv. anonymných funkcií, ktoré sú ekvivalentné s inline funkciami známymi s iných prostredí. Tieto funkcie môžu byť definované kdekoľvek, v skripte, funkcii alebo príkazovom riadku. V prípade funkcií a skriptov uložených v tzv. `m-files` (súbory s príponou `*.m`) vystupuje funkcia vo FreeMat pod menom `m-file` a nie pod menom použitým v deklarácii.

4.7 Špeciálne konštanty

Podporovaných je niekoľko základných špeciálnych konštánt použiteľných vo funkciách ako i v bežných výpočtoch.

Tabuľka 1: Špeciálne konštanty

Názov	Hodnota
<code>e</code>	Základ prirodzených logaritmov – $e \cong 2,718281828459045$
<code>i</code>	Imaginárna jednotka – $i^2 = -1$

j	Rovnaké ako i
inf	Kladné nekonečno (Infinity)
nan	Nie je číslo (Not a Number)
true	Logická 1 alebo pravda
false	Logická 0 alebo nepravda
pi	$\pi \cong 3,141592653589793$

4.8 Operátory

Podporované operátory možno rozdeliť do štyroch skupín: logické operátory, operátory prvok po prvku, štandardné operátory a operátory porovnania. Tabuľka obsahuje prehľad operátorov, spôsob ich zápisu a ich popis. Podrobný popis možno nájsť v EM alebo OD.

Tabuľka 2: Operátory

Symbol	Syntax	Popis
+	$y = a + b$	Sčítanie dvoch numerických polí prvok po prvku
-	$y = a - b$	Odočítanie dvoch numerických polí prvok po prvku
*	$y = a * b$	Násobenie dvoch numerických polí
.*	$y = a .* b$	Násobenie dvoch numerických polí prvok po prvku
/	$Y = A / B$	Pravé delenie dvoch numerických polí – matíc, totožné s \
\	$Y = A \setminus B$	Ľavé delenie dvoch numerických polí – matíc
./	$y = a ./ b$	Pravé delenie dvoch numerických polí prvok po prvku
.\	$y = a .\setminus b$	Ľavé delenie dvoch numerických polí prvok po prvku
^	$y = a ^ b$	Umocňovanie skalárov a štvorcových matíc
.^	$y = a .^ b$	Umocňovanie prvok po prvku
.'	$y = a .' $	Transpozícia matice (nevytvára komplexne združené čísla)
'	$y = a ' $	Transpozícia matice (vytvára komplexne združené čísla)
~	$y = \sim x$	NOT – negácia
&	$y = a \& b$	AND – logický súčin
	$y = a b$	OR – logický súčet
<	$y = a < b$	Menšie než
<=	$y = a <= b$	Menšie alebo rovné než
>	$y = a > b$	Väčšie než
>=	$y = a >= b$	Väčšie alebo rovné než
~=	$y = a ~= b$	Nerovná sa

==	$y = a == b$	Rovná sa
:	$y = a : c$	Vytvorenie vektora od a po c po kroku 1
	$y = a : b : c$	Vytvorenie vektora od a po c po kroku b

Základné príklady sú uvedené v kapitole 5.2 Základné operácie s maticami

4.9 Skupiny funkcií

Vychádzajúc z rozdelenia funkcií v originálnej príručke k prostrediu FreeMat je možné rozdeliť podporované funkcie do nasledujúcich kategórií:

- FreeMat funkcie – obsahuje základné funkcie pre prispôsobenie a správu prostredia.
- Matematické operátory – obsahuje operátory zobrazené v kapitole 4.8 Operátory
- Ladiace funkcie – obsahuje funkcie slúžiace pre ladenie programov
- Podpora riedkych matíc – obsahuje funkcie určené pre prácu s riedkymi maticami
- Matematické funkcie – obsahuje goniometrické, hyperbolické, exponenciálne a logaritmické funkcie
- Elementárne funkcie – obsahuje funkcie zaokrúhľovania, prevodov medzi sústavami, sumačné funkcie a pod.
- Testovacie funkcie – obsahuje funkcie pre zisťovanie vlastností premenných
- Konverzné funkcie – obsahuje funkcie pre prevody medzi dátovými typmi
- Tvorba a práca s poľami – obsahuje funkcie pre generovanie a prácu s poľami
- Tvorba náhodných čísel – obsahuje funkcie pre generovanie náhodných čísel
- Vstupne - výstupné funkcie – obsahuje funkcie pre prácu so súbormi a pod.
- Reťazcové funkcie – obsahuje funkcie pre prácu s reťazcami
- Transformácie a rozklady – obsahuje funkcie určené pre rozklady matíc a ich transformáciu
- Signal Processing funkcie – obsahuje funkcie pre spracovanie signálov
- Funkcie operačných systémov – obsahuje funkcie pre prácu so súbormi a zložkami

- Optimalizácia a aproximácia – obsahuje funkcie pre vyhodnocovanie a aproximáciu
- MPI funkcie – obsahuje funkcie MPI
- Vizualizácia – obsahuje funkcie pre tvorbu a správu grafov a základnej grafiky
- Objektovo orientované programovanie – obsahuje funkcie objektového programovania
- Bitové operátory – obsahuje funkcie pre bitové logické operácie
- Threads – obsahuje funkcie pre tvorbu a prácu s vláknami
- Related functions – obsahuje funkcie inline a symvar
- Externé rozhranie FreeMat – obsahuje funkcie programovacieho jazyka C

5 MATICE

Filozofia prostredia FreeMat je založená na práci s poľami resp. množinami. Špeciálnym typom n - rozmerného poľa je dvojrozmerné pole – matica, kde $n = 2$. Matica je základným typom vo FreeMat. Matica je chápaná ako dvojrozmerné pole s rozmermi $m \times n$, kde m je počet riadkov a n je počet stĺpcov. V prípade, že $m = 1$ sa jedná o jednoriadkovú maticu s rozmermi $1 \times n$, ktorú nazývame riadkový vektor. Obdobne existuje matica, pre ktorej rozmetry platí $m \times 1$. V tomto prípade sa jedná o jednotlípčovú maticu nazývanú stĺpcový vektor. Špeciálnym typom matice, je matica s rozmermi 1×1 , tiež nazývaná skalár.

Každý prvok matice môže byť skalár alebo iná matica pod podmienkou, že výsledná matica dáva zmysel. Obecne to znamená, že všetky prvky patriace do riadka musia mať rovnaký počet riadkov sami o sebe a zároveň všetky definície riadkov musia mať rovnaký počet stĺpcov.

Viacrozmerné polia, kde $n > 2$ nemôžu byť tvorené pomocou „zátvorkového zápisu“ používaného na tvorbu matíc. Dátový typ matice je pri použití „zátvorkového zápisu“ určený vyšetrením typov jednotlivých prvkov matice. Výsledný typ je následne vybraný ako typ s najväčším rozsahom, z typov prítomných v prvkoch matice. Výsledkom je, že nie sú stratené žiadne informácie z ktoréhokolvek prvku matice. Tento spôsob platí len pri kombinácii číselných dátových typov.

Je možné vytvoriť maticu reťazcov, avšak je potrebné sa uistiť, že v prípade, že má matica viac ako jeden riadok, majú reťazce v stĺpci rovnakú dĺžku. V prípade nezhodnosti je nutné doplniť reťazce na rovnakú dĺžku, napr. medzerami. Jedná sa však o pomerne nepraktické riešenie, preto sa používajú pre prácu s reťazcami tzv. polia reťazcov, ktoré sú špeciálnym prípadom bunkových (cell) polí. V prípade bunkových polí je možné ľubovoľne kombinovať dátové typy a taktiež nie je potrebné dohliadať na veľkosť jednotlivých prvkov.

Všeobecný zápis matice $m \times n$ je:

$$M = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \quad (1)$$

5.1 Vytváranie matíc

Prostredie FreeMat podporuje niekoľko spôsobov vytvárania matíc. Základným je vytvorenie matice s explicitného listu prvkov: $A = [\text{riadok}_1; \text{riadok}_2; \dots; \text{riadok}_m]$, kde platí: $\text{riadok}_i = \text{prvok}_1, \text{prvok}_2, \dots, \text{prvok}_n$, pre $i = \langle 1, m \rangle$. Rozmery výslednej matice budú $m \times n$. Zo zápisu vidno, že matice sa vytvárajú pomocou hranatých zátvoriek `[]` a jednotlivé riadky v definícii sú oddelované bodkočiarkou, zatiaľ čo na oddelenie prvkov v riadku slúži medzera alebo čiarka.

```
--> c=[123 256;512 895]
```

```
C =  
 123  256  
 512  895
```

Rovnaký výsledok dostaneme zápisom:

```
--> c=[123,256;512,895]
```

Ďalším spôsobom je využitie operátora dvojbodky, ktorý slúži na vytváranie riadkových vektorov v tvare: $y = a : c$, kedy je vytvorený vektor od počiatkovej hodnoty a po koncovú hodnotu c s krokom 1 , resp. v tvare $y = a : b : c$. Výsledkom druhého zápisu je vektor od počiatkovej hodnoty a po koncovú hodnotu c s krokom b . Pre tvorbu rastúceho vektora platí $a \leq c$, $b > 0$. Pre klesajúci vektor potom platí $a > c$, $b < 0$.

```
--> c=[1:3;4:0.5:5]
```

```
C =  
 1.0000  2.0000  3.0000  
 4.0000  4.5000  5.0000
```

V príklade možno vidieť vytvorenie matice použitím oboch spôsobov vytvorenia vektora pomocou operátora dvojbodka a zároveň je vidieť zmena celočíselného typu, v prípade vektora pre prvý riadok, na typ s desatinnou čiarkou.

Matice je možné vytvárať taktiež pomocou zabudovaných funkcií, načítaním matíc z externých súborov alebo pomocou užívateľom definovaných funkcií.

Vytvorenie matice náhodných prvkov je možné pomocou niektorej zo zabudovaných funkcií určených pre generovanie náhodných čísel. Podľa požiadaviek je možné vytvoriť maticu náhodných čísel s rozdelením: rovnomerným, beta, binomickým, chi – kvadrát, exponenciálnym, Fischerovým F - rozdelením, gama, Gaussovým normálnym, Poissonovým, záporným binomickým a inými. Príklad matice prvkov s Gaussovým normálnym rozdelením:

```
--> matica=randn(2,6)
```

```
matica =
```

```
-0.6907    0.7553    2.2822   -0.9318    1.0221    0.1581
 0.0849   -0.0437   -0.3667    0.5529    0.5368    0.4418
```

FreeMat podporuje tvorbu špeciálnych druhov matíc pomocou vstavaných funkcií. Jednou z takýchto funkcií je aj **diag**. Tá nám umožňuje vytvoriť diagonálnu maticu z vektora alebo získať prvky na diagonále z matice v podobe vektora. Všeobecný zápis funkcie je: **y = diag(x,n)**. Ak je **x** matica, potom výstupom **y** bude n-tá diagonála. V prípade, že je **n** vynechané, je štandardne predpokladané **n=0**. Ak je **x** vektor, potom **y** je matica s n-tou diagonálou rovnou prvkom vektora **x**. Diagonály nad hlavnou diagonálou sú označované kladne – $n > 0$, zatiaľ čo diagonály pod hlavnou diagonálou sú označené záporne – $n < 0$.

```
--> a=[2,4,6]; y=diag(a) % parameter n=0 - prvky sú na hlavnej diagonále
```

```
Y =
```

```
2 0 0
0 4 0
0 0 6
```

```
--> b=[2,4,6;1,3,5;7,8,9]
```

```
b =
```

```
2 4 6
1 3 5
7 8 9
```

```
--> z=diag(b,-1) % n= -1 - vybraté prvky 1.diagonály pod hlavnou
```

```
z =
```

```
1
8
```

Príklady demonštrujú oba spôsoby využitia funkcie **diag**.

Ďalším špecifickým typom matice je matica jednotková. Tú možno vo FreeMat vytvoriť pomocou vstavanej funkcie **eye**, ktorej syntax je: **y = eye(n)**, kde **n** je veľkosť vzniknutej jednotkovej štvorcovej matice.

```
--> eye(4)
```

```
ans =
```

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

Inou užitočnou funkciou pre tvorbu matic je funkcia **ones**, ktorej výstupom je pole so špecifikovanou veľkosťou vyplnené samými 1. Podporované sú dva spôsoby zápisu funkcie. Zápis pomocou postupnosti skalárnych hodnôt určujúcich rozmery v jednotlivých dimenziách: $y = \text{ones}(d_1, d_2, \dots, d_n)$, kde d_i je celočíselná veľkosť v príslušnej dimenzii. Druhým spôsobom je zápis rozmerov poľa pomocou vektora, ktorého každý prvok určuje veľkosť jednej dimenzie: $y = \text{ones}([d_1, d_2, \dots, d_n])$. Druhý spôsob je vhodný pre volanie funkcie s premennou – vektorom v argumente. V oboch prípadoch je pri zadaní len jedného rozmeru výstupom štvorcová matica.

```
--> ones(2,4) % rovnaký výsledok dostaneme zápisom ones([2,4])
ans =
    1 1 1 1
    1 1 1 1
```

V prípade potreby je možno vytvoriť nulovú maticu resp. maticu so všetkými prvkami rovnými 0. Tomuto účelu slúži vstavaná funkcia **zeros**. Tá nám umožňuje vytvoriť maticu núl danej veľkosti. K dispozícii máme dve formy zápisu zhodné so zápisom predchádzajúcej funkcie ones. Na viac je možné doplniť parameter určujúci dátový typ výstupnej premennej. Štandardný zápis je: $y = \text{zeros}(d_1, d_2, \dots, d_n)$, v prípade doplneného parametra je prvý spôsob zápisu nasledovný: $y = \text{zeros}(d_1, d_2, \dots, d_n, \text{classname})$. V prípade použitia vektorového zápisu je syntax nasledovná: $y = \text{zeros}([d_1, d_2, \dots, d_n])$, resp. $y = \text{zeros}([d_1, d_2, \dots, d_n], \text{classname})$. V oboch prípadoch predstavujú prvky d_i rozmery výsledného poľa v daných dimenziách. Štandardne je výstupným typom double. Ak potrebujeme iný, použijeme parameter **classname**, ktorý môže byť jedným z nasledovných reťazcov: 'double', 'single', 'int8', 'uint8', 'int16', 'uint16', 'int32', 'uint32', 'int64', 'uint64', 'float', 'logical'. Zadanie jedného rozmeru vedie na vznik štvorcovej matice daných rozmerov.

```
--> zeros(2,5,'uint16') % adekvátne zápisu zeros([2,5],'uint16')
ans =
    0 0 0 0 0
    0 0 0 0 0
```

5.2 Základné operácie s maticami

Práca s maticami je vo FreeMat realizovaná pomocou funkcií a operácií s operátormi.

5.2.1 Transpozícia

FreeMat rozoznáva dva druhy transpozície. Oba vykonajú transponovanie matice v argumente. Argument **a** je matica typu $M \times N$. Výstupom je taktiež matica rovnakého typu ako argument avšak s rozmermi $N \times M$. V prípade reálnych prvkov matice sa výstupy oboch druhov navzájom od seba nelíšia. Rozdiel je vidno pri použití argumentu obsahujúceho komplexné číslo. Zápis jednej funkcie je: **y = a.'** Jedná sa o transpozíciu nevytvárajúcu komplexne združené číslo.

```
--> a=[1 2 3;4 5 6]
a =
  1 2 3
  4 5 6
--> a.'
ans =
  1 4
  2 5
  3 6
--> b=[6+4i,3-j] % je možné použiť i alebo j ako imaginárnu jednotku
b =
  6.0000 + 4.0000i      3.0000 - 1.0000i
--> b.'
ans =
  6.0000 + 4.0000i
  3.0000 - 1.0000i
```

Druhý spôsob transponovania matíc má všeobecnú syntax: **y = a'**. Po použití tejto funkcie na komplexné číslo dochádza k vytvoreniu komplexne združených čísel.

```
--> a=[1 2 3;4 5 6]
a =
  1 2 3
  4 5 6
--> a'
ans =
  1 4
  2 5
  3 6
```

```
--> b=[6+4i,3-i]
b =
  6.0000 + 4.0000i    3.0000 - 1.0000i
--> b'
ans =
  6.0000 - 4.0000i
  3.0000 + 1.0000i
```

Porovnaním príkladov je zreteľný rozdiel pri použití oboch funkcií na komplexné čísla a zhodnosť ich výstupov pri použití na reálne čísla.

5.2.2 Operátorové aritmetické operácie

Sčítovanie (Odčítavanie) n – rozmerných číselných polí prvok po prvku je možné v prípade, že sú oba argumenty **a** a **b** rovnakej veľkosti, potom výstup **y** má veľkosť zhodnú s oboma vstupmi a rovná sa sume (rozdielu) **a** a **b** prvok po prvku. V druhom prípade je jeden z argumentov skalár a výstup **y** má veľkosť zhodnú s väčším argumentom a je sumou (rozdielom) skalára a každého prvku druhého argumentu.

```
--> A=[1 2 3;4 5 6]
A =
  1 2 3
  4 5 6
--> B=[1 0 1;2 10 6]
B =
  1 0 1
  2 10 6
--> C=A+B
C =
  2 2 4
  6 15 12
--> D=B-A
D =
  0 -2 -2
 -2 5 0
```

Pri násobení dvoch číselných polí je výsledok závislý od rozmerov oboch argumentov. V prípade, že jeden z argumentov je skalár a druhý n – rozmerné pole, výstup je súčin všetkých prvkov poľa so skalárom prvok po prvku. V prípade násobenia matíc si treba uvedomiť, že pri násobení **A** * **B** musia byť rozmery matíc nasledujúce: **A**(**M** x **N**)

a matica $\mathbf{B}(N \times K)$. Výsledná matica bude mať potom rozmery $\mathbf{M} \times \mathbf{K}$ a bude súčinom \mathbf{A} a \mathbf{B} .

```
--> A=[1 2 3;4 5 6]           % matica M x N = 2 x 3
A =
  1 2 3
  4 5 6
--> B=[1 0;1 2;10 6]         % matica N x K = 3 x 2
B =
  1 0
  1 2
 10 6
--> C=A*B                     % výsledná matica M x K = 2 x 2
C =
  33 22
  69 46
--> D=B*A                     % výsledná matica 3 x 3
D =
  1 2 3
  9 12 15
 34 50 66
```

Pri „klasickom“ násobení záleží na poradí násobených matic!!!

Pri násobení (ľavom i pravom delení) prvok po prvku platia obdobné zákonitosti ako pri sčítavaní a odčítavaní prvok po prvku. Násobenie (delenie) n – rozmerných číselných polí prvok po prvku je možné v prípade, že sú oba argumenty \mathbf{a} a \mathbf{b} rovnakej veľkosti, potom výstup \mathbf{y} má veľkosť zhodnú s oboma vstupmi a rovná sa súčinu (podielu) \mathbf{a} a \mathbf{b} prvok po prvku. V druhom prípade je jeden z argumentov skalár a výstup \mathbf{y} má veľkosť zhodnú s väčším argumentom a je súčinom (podielom) skalára a každého prvku druhého argumentu.

```
--> A=[1 2 3;4 5 6];B=[1 0 1;2 10 6];
--> E=A.*B                    % totožné s B.*A
E =
  1 0 3
  8 50 36
--> F=A./B                    % pravé delenie prvok po prvku
F =
  1.0000  1.#INF  3.0000
  2.0000  0.5000  1.0000
```

```
--> G=A.\B                                % ľavé delene prvok po prvku
G =
  1.0000         0  0.3333                 % A .\ B = B ./ A
  0.5000  2.0000  1.0000
```

5.2.3 Funkcie cross, dot, sum, size a det

Nasledovať budú základné funkcie pre prácu s maticami.

Funkcia cross

Vypočíta vektorový súčin dvoch vektorov. Syntax zápisu je: $\mathbf{c} = \text{cross}(\mathbf{a}, \mathbf{b})$, kde argumenty \mathbf{a} a \mathbf{b} sú 3 prvkové vektory.

```
--> a=[1,2,3];b=[4,5,6];
--> cross(a,b)
ans =
  -3  6 -3
```

Funkcia dot

Vypočíta skalárny súčin dvoch vektorov. Syntax zápisu je: $\mathbf{y} = \text{dot}(\mathbf{x}, \mathbf{z})$, kde argumenty \mathbf{x} a \mathbf{z} sú číselné vektory rovnakej dĺžky. V prípade, že sú argumentmi viacrozmerne polia rovnakej veľkosti, je vypočítaný skalárny súčin podľa prvej nejednoprvkovej dimenzie. Taktiež je možné špecifikovať pomocou argumentu **dim** dimenziu, podľa ktorej sa má skalárny súčin počítať. Syntax funkcie bude potom nasledovný: $\mathbf{y} = \text{dot}(\mathbf{x}, \mathbf{z}, \text{dim})$.

```
--> a=[1,2,3];b=[4,5,6];
--> dot(a,b)
ans =
  32
```

Funkcia sum

Slúži na výpočet sumy poľa podľa danej dimenzie. Štandardný zápis funkcie je: $\mathbf{y} = \text{sum}(\mathbf{x}, \mathbf{d})$, kde \mathbf{x} je n – rozmerne pole číselného typu. Argument \mathbf{d} je voliteľný a určuje dimenziu, podľa ktorej sa vykoná sumácia.

```
--> A=[1,2,3;4,5,6];
--> sum(A)
ans =
  5  7  9
--> sum(A,2)
```



```
ans =  
    6  
   15
```

Funkcia size

Je určená pre zistenie veľkosti premennej. Podporované sú dve formy zápisu: **[d1,d2,...,dn] = size(x)**, **d = size(x,n)**. Prvá forma vráti vektor celých čísel **d1** až **dn**, jedno číslo pre každú dimenziu, zatiaľ čo druhá forma vráti veľkosť argumentu **x** podľa argumentom **n** určenej dimenzie.

```
--> A=[1 2 3;4 5 6;7 8 9;0 1 2;3 4 5;4.2 0.8 9;2+5i 0.25f 5];  
--> [riadkov,stlpcov]=size(A)  
riadkov =  
    7  
stlpcov =  
    3
```

Funkcia det

Slúži na výpočet determinantu štvorcovej matice. Syntax funkcie je: $y = \det(A)$.

```
--> A=2*rand(5);det(A)  
ans =  
   -0.6256
```

5.2.4 Ďalšie funkcie pre prácu s maticami

Vzhľadom na to, že základom prostredia FreeMat je práca s maticami existuje veľké množstvo iných funkcií, okrem vyššie spomenutých, ktoré slúžia práve pre prácu s maticami. Niektoré ďalšie základné budú preto predmetom tejto podkapitoly, avšak ani tá ani zďaleka nepokryje všetky ponúkané funkcie. Omnoho viac možno nájsť v EM resp. všetky v OD.

Funkcia Rank

Výstupom funkcie je hodnosť matice. Pri použití sú možné dva spôsoby syntaxe: $y = \text{rank}(A,\text{tol})$, kde **tol** je tolerancia použitá pri výpočte, resp. tvar bez tol: $y = \text{rank}(A)$. V tomto prípade je použitá tolerancia vypočítaná podľa zákonitostí FreeMat. Hodnosť je počítaná pomocou rozkladu na singulárne hodnoty.

```
--> A=[3 7 9;15 450 8]; rank(A)
```

```
ans =  
    2  
--> B=[3 7 9;6 14 18]; rank(B)  
ans =  
    1
```

Funkcia Linspace

Funkcia slúži na vytvorenie riadkového vektora s určeným počtom prvkov, ktoré sú rovnomerne rozložené medzi dvoma určenými koncovými bodmi. Výsledkom je lineárne rozložený vektor. V prípade, že nepoužijeme argument **count** na určenie počtu prvkov výsledného vektora má funkcia tvar: $\mathbf{y} = \text{linspace}(\mathbf{a}, \mathbf{b})$, kde **a** a **b** sú koncové body intervalu a počet prvkov je štandardne určený **count = 100**. Ak potrebujeme zadať iný počet prvkov tvar zápisu je nasledovný: $\mathbf{y} = \text{linspace}(\mathbf{a}, \mathbf{b}, \text{count})$.

```
--> v = linspace(5,8,7)  
v =  
    5.0000    5.5000    6.0000    6.5000    7.0000    7.5000    8.0000
```

Funkcia Logspace

Výstupom je riadkový vektor s určeným počtom prvkov, ktoré sú logaritmicky rozložené medzi dvoma určenými koncovými bodmi. Podporované sú tri formy zápisu: $\mathbf{y} = \text{logspace}(\mathbf{a}, \mathbf{b})$, kde sú určené len koncové body intervalu $\langle 10^{\mathbf{a}}, 10^{\mathbf{b}} \rangle$ a počet prvkov vektora je daný **count = 100** alebo $\mathbf{y} = \text{logspace}(\mathbf{a}, \mathbf{b}, \text{count})$, kde je možné špecifikovať okrem hraníc intervalu **a** a **b** taktiež počet výstupných prvkov **count**. Tretia forma má tvar: $\mathbf{y} = \text{logspace}(\mathbf{a}, \mathbf{pi})$ a jej výstupom sú prvky z intervalu $\langle 10^{\mathbf{a}}, \mathbf{pi} \rangle$, ak platí $10^{\mathbf{a}} > \mathbf{pi}$ výsledkom je zostupný vektor, v prípade nerovnosti $10^{\mathbf{a}} < \mathbf{pi}$ bude vektor vzostupný.

```
--> v = logspace(0,1,4)  
v =  
    1.0000    2.1544    4.6416   10.0000
```

Funkcia Inv

Vykoná inverziu matice **x** v argumente, ktorá musí byť štvorcová a invertovateľná. Použitie: $\mathbf{y} = \text{inv}(\mathbf{x})$.

```
--> C=[0 2 1;3 3 3;5 2 4]; inv(C)  
ans =  
   -2.0000    2.0000   -1.0000  
   -1.0000    1.6667   -1.0000
```

3.0000 -3.3333 2.0000

Funkcia Sort

Slúži na zotriedenie n - rozmerných polí podľa určenej dimenzie. Zápis $\mathbf{B} = \text{sort}(\mathbf{A})$ zoradí pole \mathbf{A} podľa prvej nesingulárnej dimenzie. Dimenzia, podľa ktorej bude uskutočnené triedenie môže byť explicitne uvedená ako argument **dim** v tvare: $\mathbf{B} = \text{sort}(\mathbf{A}, \mathbf{dim})$. Taktiež je možné zadať režim zotriedenia ako argument **mode**: $\mathbf{B} = \text{sort}(\mathbf{A}, \mathbf{dim}, \mathbf{mode})$. Pre vzostupné triedenie platí **mode='ascend'** (štandardne) a režim **mode='descend'** pre zotriedenie dáť v zostupnom zmysle. Funkcia sort môže poskytovať dva výstupy a to okrem zotriedenej matice \mathbf{B} je to pole \mathbf{IX} rovnakej veľkosti ako vstupné pole \mathbf{A} , kde \mathbf{IX} predstavuje indexy \mathbf{A} podľa triedenej dimenzie odpovedajúce výstupnému poľu \mathbf{B} . Syntax je nasledujúci: $[\mathbf{B}, \mathbf{IX}] = \text{sort}(\mathbf{A})$ alebo $[\mathbf{B}, \mathbf{IX}] = \text{sort}(\mathbf{A}, \mathbf{dim})$, resp. $[\mathbf{B}, \mathbf{IX}] = \text{sort}(\mathbf{A}, \mathbf{dim}, \mathbf{mode})$. Argument **dim** nie je podporovaný ako vektor. V prípade potreby viacnásobného triedenia podľa rôznych dimenzií je potrebné viacnásobné volanie funkcie sort so zmenenými parametrami. Funkciu sort je možné použiť i na cell polia avšak len v prípade, že obsahujú reťazce. V tomto prípade je aplikované abecedné triedenie. Pri triedení komplexných čísel sa vychádza z faktu, že komplexné číslo \mathbf{a} je menšie než komplexné číslo \mathbf{b} ak platí: $|a| < |b|$. Ak sa absolútne hodnoty rovnajú, dochádza k porovnaniu podľa fázového uhla.

```
--> A=[4 3 6;2 9 3;5 0 4];
--> [B,IX] = sort(A)
B =
     2     0     3
     4     3     4
     5     9     6
IX =
     2     3     2
     1     1     3
     3     2     1
--> C = sort(A,2)
C =
     3     4     6
     2     3     9
     0     4     5
--> [D,E] = sort(A,1,'descend')
D =
     5     9     6
```

```

4 3 4
2 0 3
IX =
3 2 1
1 1 3
2 3 2
--> a={'ahoj','kamarat','vietor','slnko','motorka'}; sort(a)
ans =
 ['ahoj'] ['kamarat'] ['motorka'] ['slnko'] ['vietor']
--> b=[6+i;3-4i];sort(b)
ans =
 3.0000 - 4.0000i
 6.0000 + 1.0000i

```

5.3 Indexovanie

FreeMat podporuje tri základné triedy indexovania: `()`, `{}`, `..`. Každá je určená pre prácu s určitými objektmi v prostredí FreeMat.

Indexovanie polí

Jedná sa o najbežnejšiu formu indexácie použiteľnú na všetky polia. Prípustné sú dva spôsoby zápisu indexového výrazu, n – rozmerná forma **variable(index_1,index_2,...,index_n)** a vektorová forma **variable(index)**. Každý indexový výraz môže byť skalár, rada celočíselných hodnôt alebo špeciálny znak `:`, ktorý je skrátanou verziou výrazu **1 : end**. Kľúčové slovo **end**, v prípade, že je obsiahnuté v indexovom výraze predstavuje dĺžku poľa v danej dimenzii.

```

--> A=[4 3 6 6;2 9 3 9;5 0 4 8;3 2 1 9]
A =
 4 3 6 6
 2 9 3 9
 5 0 4 8
 3 2 1 9
--> B=A(2:3,2:3)           % použitie indexovania na pravej strane výrazu
B =
 9 3
 0 4
--> A(1:2,2:3)=zeros(2)   % použitie indexovania na ľavej strane výrazu
A =

```

```

4 0 0 6
2 0 0 9
5 0 4 8
3 2 1 9
--> D=A(4,1:end)           % ekvivalent D=A(4,:)
D =
3 2 1 9

```

Bunkové indexovanie (cell indexovanie)

Je učené len pre prácu s bunkovými – cell poľami vytvorenými pomocou syntaxe **A={riadok_1;riadok_2;...;riadok_m}**, kde platí: **riadok_i** = **prvok_i1,prvok_i2,...,prvok_in**, pre $i = \langle 1,m \rangle$. Každý prvok môže byť ľubovoľná premenná rôzneho typu vrátane matice, poľa, bunkového poľa, štruktúrovaného poľa, reťazca atď.. Jediným obmedzením je, že každý riadok musí mať rovnaký počet prvkov. Podobne ako v predchádzajúcom prípade existujú dva spôsoby zápisu indexovania: **variable{index_1,index_2,...,index_n}** ako n – rozmerná forma a **variable{index}** ako vektorová forma. Platia rovnaké pravidlá ako pre klasické indexovanie polí. Rozdiel však vzniká pri použití () indexácie na bunkové pole. V prípade použitia () je výsledkom pole, ktoré je podmnožinou pôvodného, zatiaľ čo použitím { } získame obsah bunkového poľa. Rozdiel je viditeľný z príkladu.

```

--> A={3.14, 'ahoj', [0.25:5]}
A =
[3.14]    ['ahoj']    [[1 5] double]
--> A(1:2)
ans =
[3.14]    ['ahoj']
--> A{1:2}
ans =
1 of 2:
    3.1400

2 of 2:
    ahoj

```

Indexovanie štruktúr

Je použiteľné iba na polia štruktúr, napr. vytvorené pomocou syntaxe: **y = struct(n_1,v_1,n_2,v_2,...)**, kde **n_i** sú mená polí v poli štruktúr a **v_i** sú hodnoty, ktoré

musia byť buď skalárom alebo bunkové polia rovnakých rozmerov. Zápis indexovania štruktúr má tvar: **variable.fieldname**, kde **fieldname** je jedno z polí štruktúry **variable**.

```
--> y = struct('cisla',{1,2},'pismena',{'a','b'},'znaky',{'>','|'})
y =
  Fields
  cisla
  pismena
  znaky
--> y.pismena
ans =
  1 of 2:
    a

  2 of 2:
    b
```

Komplexné indexovanie

Tri základné druhy indexovania spomenuté vyššie môžu byť skombinované do jedného indexového výrazu, za účelom vytvorenia komplikovaných indexových výrazov.

```
--> pokus{3}.struktura(4)= e
```

Vyššie spomenutý zápis má nasledovný význam. Pole **pokus** je bunkové (cell) pole s dĺžkou tri, kde tretí prvok je pole štruktúr s jedným prvkom a toto obsahuje pole menom **struktura** so štyrmi prvkami, kde štvrtému z nich je priradená hodnota e.

5.4 Polynómy

FreeMat poskytuje základné funkcie pre prácu s polynómami. Polynómy sú v prostredí FreeMat reprezentované riadkovým vektorom obsahujúcim koeficienty zoradené v zostupnom poradí mocnín. Všeobecne možno polynóm $p(x)=a_nx^n + a_{n-1}x^{n-1} + \dots + a_2x^2 + a_1x^1 + a_0x^0$ zapísať ako riadkový vektor koeficientov jeho mocnín v nasledovnom tvare: $p=[a_n \ a_{n-1} \ a_{n-2} \ \dots \ a_2 \ a_1 \ a_0]$.

```
--> p=[3 0 2 5] % predstavuje zápis polynómu p(x)=3x3 + 0x2 + 2x1 + 5x0
```

5.4.1 Funkcie pre prácu s polynómami

Funkcia Poly

Slúži na výpočet koeficientov polynómu zo zadaných koreňov. Použitie: $\mathbf{p} = \text{poly}(\mathbf{r})$, kde \mathbf{r} je vektor koreňov, \mathbf{p} je vektor, ktorého prvky sú koeficienty polynómu, ktorého korene sú prvkami vektora \mathbf{r} . Funkciu je možné použiť taktiež na matice: $\mathbf{p} = \text{poly}(\mathbf{A})$, kde \mathbf{A} je $n \times n$ štvorcová matica a \mathbf{p} je riadkový vektor s $n+1$ prvkami, ktoré sú koeficienty charakteristického polynómu.

```
--> r=[-5 -3]; p=poly(r)
p =
    1 8 15
    % hľadáme polynóm, ktorého korene sú: x1=-5, x2=-3
    % výsledkom je polynóm 1x2 + 8x + 15
--> A=[2 4 6;1 3 5;7 8 7]; s=poly(A)
s =
    1.0000    -12.0000   -45.0000     4.0000
```

Funkcia Polyder

Výsledkom funkcie **polyder** sú koeficienty polynómu vyplývajúceho z diferenciácie (derivácie) polynómu \mathbf{p} v argumente funkcie: $\mathbf{pder} = \text{polyder}(\mathbf{p})$ alebo spočíta deriváciu súčinu polynómov $\mathbf{p1}$, $\mathbf{p2}$ ($\mathbf{p1}*\mathbf{p2}$) v argumente: $\mathbf{prodp1p2der} = \text{polyder}(\mathbf{p1},\mathbf{p2})$. Taktiež je možné pomocou funkcie **polyder** určiť deriváciu podielu polynómov $\mathbf{n,d}$ ($\mathbf{n/d}$) z argumentu: $[\mathbf{nder},\mathbf{dder}] = \text{polyder}(\mathbf{n},\mathbf{d})$, kde \mathbf{nder} je čitateľ a \mathbf{dder} je menovateľ.

```
--> q=polyder([5 0 4 3])
q =
    15 0 4
    % požadujeme deriváciu polynómu 5x3 + 0x2 + 4x + 3
    % výsledkom je polynóm q = 15x2 + 0x + 4

--> a=[7 2];r=polyder([2 3 4],a)
r =
    42 50 34
    % požadujeme r=[(2x2 + 3x + 4)*(7x + 2)]' = 42x2 + 50x + 34

--> [citateľ,menovateľ]=polyder([2 3 4],a)
citateľ =
    14 8 -22
    % požadujeme [(2x2 + 3x + 4)/(7x + 2)]'

menovateľ =
    49 28 4
```

Funkcia Polyfit

Slúži pre aproximovanie množiny dát polynómom pomocou metódy najmenších štvorcov. Syntax funkcie je: **p = polyfit(x,y,n)**, kde **x** a **y** sú vektory rovnakej veľkosti obsahujúce **x** a **y** hodnoty a **n** je stupeň aproximujúceho polynómu. Výsledný vektor **p** obsahuje koeficienty mocnín optimálneho polynómu v zostupnom poradí.

```
--> x=[1 2 3 4 5];y=[3.4 7.6 8.9 9.5 9.9];p=polyfit(x,y,2)
p =
-0.5929    5.0471    -0.7600    % p = -0.5929x2 + 5.0471x - 0.7600
```

Funkcia Polyint

Výstupom funkcie **polyint** sú koeficienty polynómu **pint** vzniknutého integráciou polynómu **p** v argumente funkcie: **pint = polyint(p,k)**, kde **p** je vektor koeficientov polynómu v zostupnom poradí mocnín a **k** je integračná konštanta. Pre **k=0** platí: **pint = polyint(p)**.

```
--> pint=polyint([2 3 4])
pint =
0.6667    1.5000    4.0000    0    % pint = 0.6667x3 + 1.5x2 + 4x
```

Funkcia Polyval

Slúži pre vyhodnotenie polynómu, ktorého koeficienty mocnín v zostupnom zmysle obsahuje vektor **p**, pre data **x**. Syntax funkcie je: **y = polyval(p,x)**. Ak je **x** matica, polynóm je vyhodnotený v maticovom zmysle - **x** musí byť štvorcová matica.

```
--> p=[1 0 0];x=[0:4];y=polyval(p,x)
y =
0 1 4 9 16    % vyhodnocujeme p = x2 pre hodnoty x = 0,1,2,3,4
              % pre jednotlivé x nadobúda y hodnoty: y = 0,1,4,9,16
```

Funkcia Roots

Slúži pre získanie koreňov polynómu. Výstupom funkcie je stĺpcový vektor **z** obsahujúci korene polynómu, ktorého koeficienty sú v zoradene zostupne podľa mocnín vo vektore **p**. Syntax je: **z = roots(p)**.

```
--> p=[1 8 15];z=roots(p)
z =
-5          % hľadáme korene polynómu p = 1x2 + 8x + 15
-3          % vypočítané korene sú: x1=-5, x2=-3
```


6 MATEMATICKÉ FUNKCIE

Obsahom tejto kapitoly bude zoznámenie sa so základnými a najpoužívanejšími matematickými funkciami podporovanými prostredím FreeMat. Funkcie budú rozdelené do skupín podľa ich matematického zadelenia.

6.1 Trigonometrické funkcie

Obsahom podkapitoly trigonometrické funkcie bude priblíženie použitia a práce so štyrmi základnými goniometrickými funkciami – sin, cos, tan, cot. Pre všetky funkcie platia rovnaké vlastnosti, preto budú spomenuté iba pri prvej z nich a to pri funkcii sin.

Funkcia Sin

Spočíta funkciu sínus pre daný argument zadaný v radiánoch. Syntax je: $y = \sin(x)$, kde x je n – rozmerné pole číselného typu. Výstup y je rovnakej veľkosti a typu ako argument x okrem prípadu celočíselného integer vstupu, ktorý je pretypovaný na typ double. V prípade potreby je možné použiť funkciu: $y = \text{sin}d(x)$, ktorá sa líši od $\sin(x)$ tým, že argument je zadávaný v stupňoch.

```
--> sin(pi/2)           % ekvivalent: sind(90)
ans =
    1
```

Funkcia Cos

Pre výpočet funkcie kosínus je syntax: $y = \cos(x)$ pre argument x zadaný v radiánoch a syntax: $y = \text{cos}d(x)$ pre argument zadávaný v stupňoch.

```
--> cos(pi/3)           % ekvivalent: cosd(60)
ans =
    0.5000
```

Funkcia Tan

Slúži pre výpočet tangens pre daný argument x zadaný v radiánoch: $y = \tan(x)$. Pre prácu s argumentom x zadaným v stupňoch slúži funkcia: $y = \text{tand}(x)$.

```
--> tan(pi/6)           % ekvivalent: tand(30)
ans =
    0.55774
```

Funkcia Cot

Výstupom je funkcia kotangens spočítaná pre argument x zadaný v radiánoch v prípade syntaxe: $y = \mathbf{cot}(x)$ a pre argument x zadaný v stupňoch v tvare: $y = \mathbf{cotd}(x)$.

```
--> cot(pi/4)           % ekvivalent: cotd(45)
ans =
    1
```

6.2 Cyklometrické funkcie

Samozrejmosťou prostredia FreeMat je podpora inverzných trigonometrických funkcií tzv. cyklometrických funkcií, ktoré sú predmetom tejto podkapitoly. Všetky funkcie sú určené pre argument x , ktorý je n – rozmerným poľom číselného typu. Výstup y je rovnakého typu a veľkosti ako vstup až na prípad celočíselného integer vstupu, kedy dôjde k pretypovaniu na typ double.

Funkcia Asin

Spočíta inverzný sínus y v radiánoch: $y = \mathbf{asin}(x)$, resp. inverzný sínus v stupňoch: $y = \mathbf{asind}(x)$ pre argument x .

```
--> asin(1)
ans =
    1.5708
--> asind(1)           % 1.5708 rad = 90°
ans =
    90
```

Funkcia Acos

Výstupom je inverzný kosínus y v radiánoch: $y = \mathbf{acos}(x)$, resp. inverzný kosínus v stupňoch: $y = \mathbf{acosd}(x)$ pre argument x .

```
--> acos(sqrt(2)/2)
ans =
    0.7854
--> acosd(sqrt(2)/2)   % 0.7854 rad = 45°
ans =
    45
```

Funkcia Atan

Služi na výpočet inverzného tangens y v radiánoch: $y = \mathbf{atan(x)}$, resp. v stupňoch: $y = \mathbf{atan(x)}$ pre argument x .

```
--> atan(sqrt(3))
ans =
    1.0472
--> atand(sqrt(3))      % 1.0472 rad = 60°
ans =
    60.0000
```

Funkcia Acot

Je určená na výpočet inverzného kotangens y v radiánoch: $y = \mathbf{acot(x)}$, resp. v stupňoch: $y = \mathbf{acotd(x)}$ pre argument x .

```
--> acot(sqrt(3))
ans =
    0.5236
--> acot(sqrt(3)) % 0.5236 rad = 30°
ans =
    30.0000
```

6.3 Ďalšie matematické funkcie

Okrem vyššie spomenutých funkcií podporuje FreeMat veľké množstvo iných matematických funkcií, z ktorých najpoužívanejšie sú obsahom tejto podkapitoly.

Funkcia Exp

Je určená na výpočet exponenciálnej funkcie pre argument x , ktorý je n – rozmerné číselné pole. Syntax je: $y = \mathbf{exp(x)}$.

```
--> y=exp(2)
y =
    7.3891
```

Funkcia Fix

Služi pre zaokrúhľovanie čísel smerom k nule. Je určená pre číselné polia. Pre kladné prvky x je výstupom y najväčšie celé číslo menšie než x . Pre záporné prvky x je výstup y

najmenšie celé číslo väčšie než x . Pre komplexné čísla x je zaokrúhlenie aplikované oddelene na reálnu i imaginárnu časť. Syntax funkcie je: $y = \mathbf{fix}(x)$.

```
--> y=fix([1.4,5.6,-3.8,-8.1])
y =
    1    5   -3   -8
--> fix([2.3+5.9i,-7.7-0.02i])
ans =
    2.0000 + 5.0000i    -7.0000 + 0.0000i
```

Funkcie pre výpočet logaritmu

Funkcia $y = \mathbf{log}(x)$ vypočíta prirodzený logaritmus so základom e pre argument x . Pre kladné reálne vstupy je výstup y rovnaký ako vstup, pre záporné a komplexné argumenty x je výstupom y komplexné číslo. Argument x je n – rozmerné číselné pole. Na výpočet dekadického logaritmu so základom 10 slúži funkcia: $y = \mathbf{log10}(x)$. FreeMat podporuje i výpočet dvojkového logaritmu so základom 2 pomocou vstavanej funkcie: $y = \mathbf{log2}(x)$.

```
--> y=log(e)
y =
    1
--> y=log10(100)
y =
    2
--> y=log2(256)
y =
    8
```

Funkcia Sqrt

Vypočíta druhú odmocninu n – rozmerného poľa v argumente. Zápis: $y = \mathbf{sqrt}(x)$.

```
--> y=sqrt([64,-1,3+4i])
y =
    8.0000 + 0.0000i    0.0000 + 1.0000i    2.0000 + 1.0000i
```

7 ELEMENTÁRNE FUNKCIE

Obsahom tejto kapitoly budú funkcie, ktoré nie sú typicky matematické, ale slúžia na prácu s číslom, konkrétne sa jedná o funkcie pre zaokrúhľovanie a určenie absolútnej hodnoty čísla a funkciu pre vytvorenie komplexne združeného čísla.

Funkcia Abs

Vráti absolútnu hodnotu pre všetky prvky vstupného n – rozmerného číselného poľa x v argumente. Zápis jej použitia je: $y = \text{abs}(x)$.

```
--> y=abs([3+4i,2.0-3.0i,i])
y =
    5.0000    3.6056    1.0000
--> z=abs([-5,8,0,10.45,-2.75])
z =
    5.0000    8.0000    0    10.4500    2.7500
```

Funkcia Round

Je určená pre zaokrúhlenie prvkov n – rozmerného poľa x k najbližšiemu celému číslu prvok po prvku. Pre celočíselné argumenty nemá použitie funkcie round význam a pre komplexné čísla nie je definovaná. Zápis funkcie je: $y = \text{round}(x)$.

```
--> y=round([-5,8,0,10.45,8.89,-2.75,-9.01])
y =
    -5    8    0    10    9    -3    -9
```

Funkcia Ceil

Prvok po prvku zaokrúhli smerom nahor všetky prvky vstupného n – rozmerného poľa x . Zaokrúhlenie čísla smerom nahor je definované ako najmenšie celé číslo väčšie alebo rovné ako argument. Podobne ako pri predchádzajúcej funkcii celočíselné argumenty nie sú zmenené a funkcia nie je definovaná pre komplexné čísla. Syntax je: $y = \text{ceil}(x)$.

```
--> y=ceil([-5,8,0,10.45,8.89,-2.75,-9.01])
y =
    -5    8    0    11    9    -2    -9
```

Funkcia Floor

Je určená pre zaokrúhlenie všetkých prvkov vstupného n – rozmerného poľa x smerom nadol prvok po prvku. Zaokrúhlenie čísla smerom nadol je definované ako najväčšie celé

číslo, ktoré je menšie alebo rovné číslu zaokrúhľovanému. Pri celočíselných argumentoch nedochádza k ich zmene a funkcia nie je definovaná pre komplexné čísla v argumente.

Zápis je: $y = \text{floor}(x)$.

```
--> y=floor([-5,8,0,10.45,8.89,-2.75,-9.01])
```

```
y =
```

```
-5 8 0 10 8 -3 -10
```

Funkcia Conj

Vráti komplexne združené čísla pre všetky prvky vstupného n – rozmerného poľa x .

Použitie funkcie conj nemá žiadny vplyv na celočíselné a reálne argumenty. Všeobecný

zápis funkcie je: $y = \text{conj}(x)$.

```
--> y=conj([-5,8,0,10.45,-2.75])
```

```
y =
```

```
-5.0000 8.0000 0 10.4500 -2.7500
```

```
--> z=conj([3+4i,-2+i,1-9i])
```

```
z =
```

```
3.0000 - 4.0000i -2.0000 - 1.0000i 1.0000 + 9.0000i
```

8 PRÁCA S REŤAZCAMI

Samostatnou časťou prostredia FreeMat je práca s reťazcami. Ako už bolo spomenuté v kapitole č. 5 Matice je možné vytvoriť maticu reťazcov. Je však potrebné dodržať, aby všetky reťazce v stĺpci mali rovnakú dĺžku, čo je pomerne nepraktické. Z tohto dôvodu sa používajú na prácu s reťazcami bunkové polia tzv. bunkové polia reťazcov. Práve týmto poliam reťazcov a prácou s nimi bude venovaná nasledovná kapitola.

8.1 Vytvorenie poľa reťazcov

Vytvorenie bunkového poľa reťazcov je možné pomocou štandardného postupu pri tvorení bunkového poľa a to zápisom: $A = \{\text{riadok}_1; \text{riadok}_2; \dots; \text{riadok}_m\}$, kde platí: $\text{riadok}_i = \text{prvok}_{i1}, \text{prvok}_{i2}, \dots, \text{prvok}_{in}$, pre $i = \langle 1, m \rangle$. Každý prvok môže byť ľubovoľná premenná rôzneho typu vrátane matice, poľa, bunkového poľa, štruktúrovaného poľa, reťazca atď.. Jediným obmedzením je, že každý riadok musí mať rovnaký počet prvkov. Druhým spôsobom je prevedenie matice znakov na pole reťazcov funkciou `cellstr`.

Funkcia `Cellstr`

Slúži na zmenu matice znakov x na bunkové pole jednotlivých reťazcov y . Každý reťazec matice je umiestnený do jednej bunky poľa a doplnené medzery sú odstránené. Zápis funkcie je: $y = \text{cellstr}(x)$, kde x je $M \times N$ pole znakov (matica znakov) ako reťazec. Najskôr vytvoríme maticu znakov. Je potrebné, aby mali všetky reťazce v stĺpci rovnaký počet znakov – kratšie slová je potrebné doplniť medzerami na dĺžku najdlhšieho reťazca v stĺpci.

```
--> A=['skola';'ahoj ';'pes ';'jedlo']
A =
    skola
    ahoj
    pes
    jedlo
```

Pomocou funkcie `cellstr` prevedieme maticu A na pole reťazcov a .

```
--> a=cellstr(A)
A =
    ['skola']
    ['ahoj']
    ['pes']
    ['jedlo']
```

8.2 Funkcie pre úpravu reťazcov

Jedná sa predovšetkým o úpravu formou odstránenia resp. pridania medzier na konci alebo začiatku reťazca.

Funkcia Deblank

Slúži na odstránenie koncových medzier z reťazca. Funkciu je možné použiť na jeden reťazec i bunkové pole: $y = \text{deblank}(x)$, kde x je reťazec, z ktorého budú odstránené medzery na jeho konci alebo x je bunkové pole reťazcov, v ktorom z každého reťazca v ňom obsiahnutého budú odstránené koncové medzery.

```
--> y=deblank('slnko  ')
y =
  slnko
--> z=deblank({'kamen ','spadol ',' do ',' rieky '})
z =
  ['kamen'] ['spadol'] [' do'] [' rieky']
```

Treba si uvedomiť, že funkcia odstráni iba koncové medzery!! Na odstránenie všetkých medzier slúži funkcia strtrim.

Funkcia Strtrim

Slúži pre odstránenie medzier zo začiatku i konca reťazca. Je určená pre prácu s reťazcom ako argumentom: $y = \text{strtrim}(str)$, kde str je reťazec alebo bunkové pole reťazcov. V prípade bunkového poľa reťazcov je funkcia aplikovaná na každý reťazec v poli.

```
--> y=strtrim('  privela medzier v texte  ')
y =
  privela medzier v texte
--> z=strtrim({'  privela ','medzier ',' v ',' texte ',' '})
z =
  ['privela'] ['medzier '] ['v'] ['texte'] []
```

8.3 Testovacie funkcie

Testujú reťazec na prítomnosť určitých znakov – medzier, číslíc alebo písmen. Výstupom všetkých testovacích funkcií je logické pole, ktoré obsahuje log 1 na mieste výskytu testovaného znaku (medzera, číslica, písmeno) a log 0 na mieste výskytu iného ako testovaného znaku. Vstupom pre všetky funkcie je reťazec s .

Funkcia Isalpha

Testuje reťazec na prítomnosť písmen. Nerozoznáva veľkosť písmen a vráti log 1 pre všetky výskyty anglických písmen a až z resp. A až Z. Syntax je: **x = isalpha(s)**.

```
--> isalpha('Film Cislo 5 zije ma 2. diel')
ans =
  1 1 1 1 0 1 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 0 0 1 1 1 1
```

Funkcia Isdigit

Testuje reťazec na prítomnosť číslíc. Zápis funkcie je: **x = isdigit(s)**.

```
--> isdigit('Film Cislo 5 zije ma 2. diel')
ans =
  0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
```

Funkcia Isspace

Testuje reťazec na prítomnosť prázdnych znakov. Za prázdny znak je považovaná medzera, znak pre nový riadok, znak tabulátora, znak pre návrat vozíka na začiatok riadku, znak novej stránky a vertikálny tabulátor. Syntax funkcie je: **x = isspace(s)**.

```
--> isspace('Film Cislo 5 zije ma 2. diel')
ans =
  0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0
```

8.4 Konverzné funkcie

Slúžia na zmenu písmen v reťazci na veľké resp. malé. Zmenu všetkých malých písmen (a až z) na veľké dosiahneme funkciou: **y = upper(x)**, kde **x** je reťazec resp. bunkové pole reťazcov. V prípade bunkového poľa je funkcia aplikovaná na každý reťazec v ňom obsiahnutý. Na zmenu všetkých veľkých písmen (A až Z) v reťazci na malé písmená je potrebné použiť funkciu: **y = lower(x)**, pričom pre **x** platia rovnaké vlastnosti ako pri funkcii upper.

```
--> upper('ToTo je LUBOvolny text s pismenami a cislami 124589')
ans =
  TOTO JE LUBOVOLNY TEXT S PISMENAMI A CISLAMI 124589
--> lower('ToTo je LUBOvolny text s pismenami a cislami 124589')
ans =
  toto je lubovolny text s pismenami a cislami 124589
```

8.5 Další funkce pro práci s řetězci

Funkce Strcmp

Služi pro porovnání dvou řetězců, resp. řetězců v dvou bunkových poliach **x** a **y** na shodnost. Zápis funkce je: **p = strcmp(x,y)**. Výstup je log 1 ak mají oba argumenty rovnakú veľkosť a sú zhodné ako reťazce, inak je výstup log 0. Funkciu je taktiež možné použiť na maticu znakov. Funkcia rozoznáva a vyhodnocuje veľkosť písmen. Obmenou funkcie strcmp je funkcia: **y = strncmp(x,y,n)**, pre ktorú platia rovnaké zákonitosti ako pre strcmp avšak porovnáva len prvých **n** znakov z oboch reťazcov.

```
--> retazec1='jablko';retazec2='hruska';retazec3='jablko';
--> strcmp(retazec1,retazec2)
ans =
    0
--> strcmp(retazec1,retazec3)
ans =
    1
--> poleRetazcov={'jablko','stolicka','128','jablko '};
--> strcmp(retazec1,poleRetazcov)
ans =
    1 0 0 0
--> poleRetazcov2={'hruska','stolica','128','jablko '};
--> strcmp(poleRetazcov,poleRetazcov2)
ans =
    0 0 1 1
```

Funkce Strcmpi

Ekvivalent predchádzajúcej funkcie, avšak ignoruje veľkosť písmen. Syntax je: **y = strcmpi(x,y)**.

```
--> retazec1='jablko';retazec2='Jablko';
--> strcmpi(retazec1,retazec2)
ans =
    0
--> strcmpi(retazec1,retazec2)
ans =
    1
```

Funkcia Strfind

Prehľadáva reťazec alebo pole reťazcov **string**, v ktorom hľadá vzor **pattern**. Výsledkom je pole **ndx** obsahujúce začiatkové pozície výskytu vzoru v reťazci. V prípade poľa reťazcov sa vyhľadávanie vzoru vykonáva pre každý reťazec v poli. Funkcia má tvar:

ndx = strfind(string,pattern).

```
--> a='ahoj, moj stroj presiel loj';b=strfind(a,'oj')
b =
    3 7 13 25
```

Funkcia Strrep

Sa používa na nahradenie každého výskytu reťazca **find** reťazcom **replace** v reťazci **source**. Každý z reťazcov môže byť taktiež bunkovým poľom reťazcov, pričom sa funkcia aplikuje na každý prvok. Syntax funkcie je: **p = strrep(source,find,replace)**.

```
--> strrep('V zime casto prsi','prsi','snezi')
ans =
    V zime casto snezi
```

9 VIZUALIZÁCIA

Podpora vizualizácie výpočtov pomocou 2D, 3D grafov a základnej grafiky je v prostredí FreeMat natoľko obsiahla, že by ňou bolo možné pokryť rozsah celej tejto práce. Z tohto dôvodu bude predmetom tejto kapitoly iba stručný výber funkcií a popis základných možností práce s grafikou, názorné príklady a podrobnejšie vysvetlenia budú v EM, ktorý je súčasťou práce.

Filozofia tvorby grafiky vo FreeMat vychádza zo základnej myšlienky hierarchickej štruktúry rodiča a jeho potomkov. Rodičom je v tomto prípade grafický objekt Figure(diagram, schéma, obrázok) a ďalšie grafické objekty neskôr pridávané sú jeho potomkami alebo potomkami iných grafických objektov, ktorých priamym rodičom je práve Figure. Priamym potomkom objektu Figure je objekt Axes (osi), ďalšie vkladané objekty ako sú obrázky, čiary, texty a pod sú jeho potomkami. Každý grafický objekt má určitý rozsah svojich vlastností, ktorých hodnoty je možno meniť a taktiež má svoj manipulátor, ktorý sa používa pre prácu s daným objektom a nastavovanie už spomenutých vlastností.

9.1 Objekt Figure

Vytvorí sa vždy pri zavolaní funkcie vykresľujúcej grafiku, resp. je možné ho vytvoriť príkazom **figure(number)**, kde **number** je číslo slúžiace ako manipulátor objektu. Týmto príkazom je zároveň možno určiť, ktoré okno je aktívne v prípade spustenia viacerých okien – objektov Figure. V mnohých príkazoch pre prácu s oknom je argumentom funkcie číslo okna, ktoré je zároveň aj jeho manipulátorom. Na získanie čísla aktívneho okna slúži bezargumentový príkaz **gcf**. Zatvorenie okna uskutočníme príkazom **close(handle)**, kde **handle** je práve manipulátor okna, ktoré chceme zavrieť. Alternatívou je príkaz **close** bez argumentu, ktorý zatvorí aktívne okno, resp. príkaz **close('all')**, ktorý slúži na zatvorenie všetkých práve otvorených okien. Obsah aktívneho okna Figure možno v prípade potreby vyčistiť príkazom **clf**. Grafické okno určené manipulátorom okna ako argumentom **fignum** je možné pomocou príkazov **figlower(fignum)** resp. **figraise(fignum)** skryť do úzadia za iné okná, resp. zviditeľniť do popredia pred iné okná. Ďalším základným príkazom je funkcia **sizefig(width,height)**, ktorá slúži na zmenu veľkosti aktuálne vybraného okna. Argumenty **width** (šírka) a **height** (výška) sú požadované rozmery okna.

Druhú skupinu funkcií tvoria funkcie určené pre prácu s obsahom okna a sú tu zahrnuté napr. **copy**, ktorá skopíruje aktívne Figure okno do schránky vo formáte *.bmp alebo funkcia **print(filename)**, resp. **print filename**, ktorá slúži na uloženie práve aktívneho okna do súboru určeného jeho názvom – **filename** vo formáte daného príponou. Podporované prípony sú *.jpg, *.jpeg, *.pdf, *.png.

9.2 Objekt Axes

Novú sadu osí vytvoríme pomocou funkcie **axes**, ktorá má tri rôzne podoby: bez argumentu: **h = axes**, kedy sa vytvorí nový set osí a pridruží sa k aktuálnemu diagramu – objektu figure, rozšírený zápis: **h = axes(propertyname,value,propertyname,value, ...)**, kde **propertyname** je meno vlastnosti a **value** nasledujúce za ním, je hodnota nastavená tejto vlastnosti. Nastavenie správania sa osí je možné pomocou funkcie **axis** a jej rôznych možností. Takto je možné prispôbiť si rozsahy, farbu osí, určiť polohu začiatku súradného systému, tvar osí, ich viditeľnosť a podobne. Jednotlivé osi je taktiež možno prispôbovať nastavovaním hodnôt jednotlivým vlastnostiam. Vo FreeMat je možno nastaviť objektu osi takmer sto rôznych vlastností. Za zmienku stoja vlastnosti pre úpravu rozsahu jednotlivých osí a zobrazenie ich popisu. Tieto vlastnosti je možné meniť pomocou samostatných funkcií na to určených a nielen pomocou dvoch vyššie uvedených spôsobov. Popis osí vložíme pomocou príkazu **?label('label')**, resp. niektorej z jeho verzií **?label 'label'**, **?label label** alebo **?label('label',properties...)**, kde znak **?** nahradíme označením osi **x**, **y** alebo **z** (napr. **xlabel label**). Označenie osi určuje, ktorej osi sa daný príkaz týka. Rozsah jednotlivých osí je možné nastaviť pomocou príkazu **?lim([lo,hi])** alebo niektorej z jeho verzií **?lim('auto')**, **?lim('manual')**. Opäť je potrebné nahradiť znak **?** označením osi **x**, **y** alebo **z**. Na prácu s osou pomocou príkazov vyžadujúcich manipulátor slúži funkcia **gca**, ktorá vráti práve manipulátor aktuálneho objektu typu osi. Všetky ďalšie vytvorené objekty budú potomkami aktuálnych osí. Na odstránenie aktuálnych osí slúži funkcia **cla**. Obe posledne zmienené funkcie sú bezargumentové.

9.3 Potomkovia objektu Axes

Priamymi potomkami objektu osi sú základné objekty ako obrysový objekt, obrázok, čiara, text, plocha a iné. Tieto objekty je možné vytvoriť samostatnými funkciami. Všeobecný

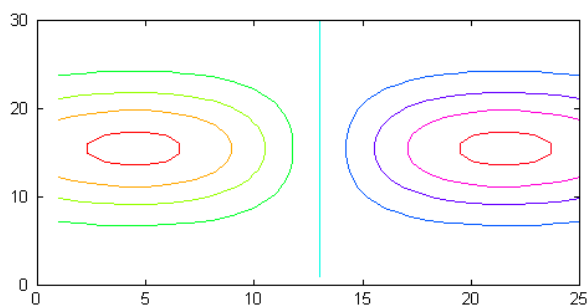
tvar funkcie je: **handle** = **objectname(property, value, property, value,...)**, kde výstupom **handle** bude manipulátor takto vytvoreného objektu a **objectname** je potrebné nahradiť kľúčovým slovom pre daný objekt: **hcontour** pre vytvorenie obrysového objektu, **himage** pre obrázok, **hline** pre čiaru, **htext** pre text, **hsurface** pre rovinný objekt. Každému z takto vytvorených objektov je opäť možné ponastavovať veľké množstvo vlastností a tak si prispôbiť vzhľad a správanie sa daného objektu. Medzi základné vlastnosti patria napr. zobrazenie a nastavenie mriežky, legendy, nadpisu, titulkov, hrúbky a štýlu čiar a písma, vlastností farby a pod..

9.4 Základné typy vykreslení

Na vykresľovanie priebehov funkcií slúžia 2D a 3D grafy, ktoré sú potomkami objektu *osi*.

Contour Plot

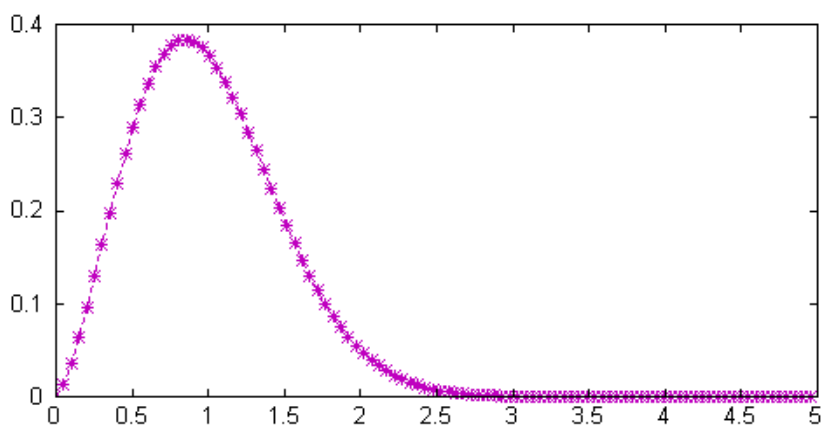
Slúži na vykreslenie obrysového grafu. Existuje viacero prevedení základnej funkcie: **contour(Z)**, ktorá vytvorí graf z dát v matici **Z**. Zápis **contour(Z,n)** je ekvivalentný predchádzajúcemu s tým rozdielom, že argument **n** slúži na špecifikáciu úrovní obrysov. Ak potrebujeme zadať vektory hodnôt **x** a **y** syntax je nasledovná: **contour(X,Y,Z,n)**. Ďalšie spôsoby zápisu slúžia pre bližšiu špecifikáciu vlastností výsledného vykreslenia. Forma pre vytvorenie 3D obrysového grafu má tvar: **contour3(X,Y,Z,n)**. Pridanie popisov jednotlivých úrovní obrysov do grafu je možné samostatnou funkciou **clabel(contourhandle,property,value,property,value,...)**, kde **contourhandle** je manipulátor obrysového grafu, do ktorého budú popisy vložené a dvojice **property, value** predstavujú vlastnosť a jej nastavené hodnoty.



Obrázok 2: Obrysový graf

Plot

Služi pre vykreslenie štandardného grafu a syntax funkcie je: **plot(<data 1>,{linespec 1},<data 2>,{linespec 2},...,properties,...)**, kde <data> argumenty môžu mať rôznu formu: vektor-matica pár – hodnoty **x** sú číselný vektor a hodnoty **y** sú v stĺpcoch matice, nespárovaná matica – hodnoty **y** sú dodané v matici a hodnoty **x** sú dopočítané automaticky a nakoniec prípad komplexnej matice, kedy sú reálne časti komplexných čísel vykreslené voči imaginárnym. Argument **linespec** je voliteľný a jedná sa o reťazec pozostávajúci z troch znakov – **colorspec**, **symbolspec** a **linestylespec**. Znak **colorspec** určuje farbu. Podporované sú: **r** – červená, **g** – zelená, **b** – modrá, **k** – čierna, **c** – tyrkysová, **m** – purpurová a **y** – žltá. Druhým v poradí je znak **symbolspec** určujúci značku vykreslenú v každom dátovom bode. Na výber máme z: „**.**“ – bodka, „**o**“ – kružnica, „**x**“ – symbol krát, „**+**“ – symbol plus, „*****“ – symbol hviezdička, „**s**“ – štvorček, „**d**“ – diamant, „**v**“ – klesajúci trojuholník, „**^**“ – stúpajúci trojuholník, „**<**“ – ľavý trojuholník, „**>**“ – pravý trojuholník. Posledný symbol špecifikuje štýl vykresľovanej čiary: „**-**“ – plná čiara, „**:**“ – bodkovaná čiara, „**-.**“ – bodkočiarkovaná čiara, „**- -**“ – čiarkovaná čiara.



Obrázok 3: Štandardný graf

Plot3

Jedná sa o 3D verziu predchádzajúceho príkazu plot. Syntax funkcie je: **plot3(X,Y,Z,{linespec 1},X,Y,Z,{linespec 2},...,properties,...)**. **X**, **Y** a **Z** sú súradnice bodov v 3D priestore. Ďalšie parametre sú zhodné s plot príkazom.

Semilogx

Služi na vytvorenie štandardného vykreslenia zhodného s príkazom `plot` avšak nastaví osi **x** logaritmickú mierku. Syntax použitia: `semilogx(<data 1>,{linespec 1},<data 2>,{linespec 2},...,properties,...)`.

Semilogy

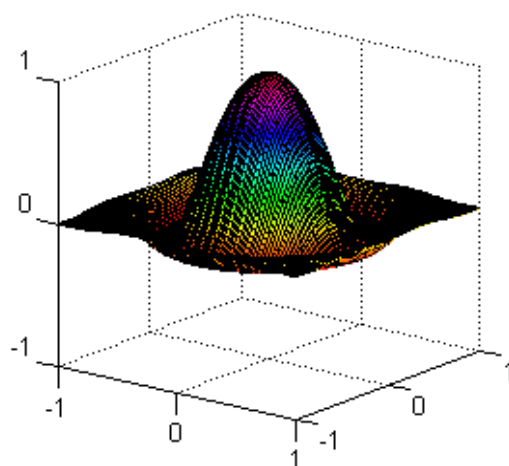
Zhodné s predchádzajúcim spomenutým avšak logaritmická mierka je nastavená osi **y**. Zápis: `semilogy(<data 1>,{linespec 1},<data 2>,{linespec 2},...,properties,...)` .

Log-Log

Kombinácia predchádzajúcich dvoch vykreslení. Služi na vykreslenie klasického grafu s oboma osami v logaritmickej mierke. Syntax: `loglog(<data 1>,{linespec 1},<data 2>,{linespec 2},...,properties,...)` .

Surf

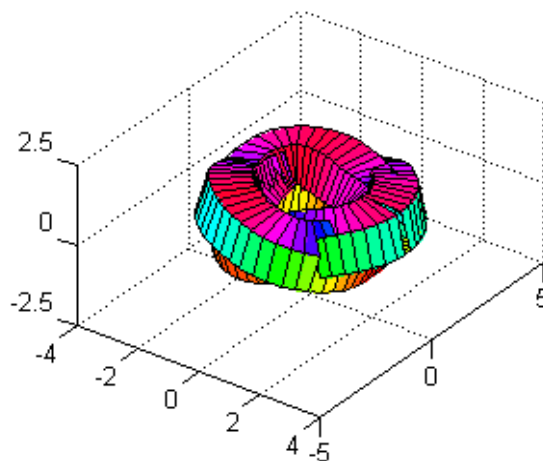
Používa sa na vytvorenie plošného 3D grafu, ktorý je definovaný vektormi **X**, **Y** a **Z** obsahujúcimi príslušné súradnice bodov. Voliteľnými argumentmi sú matica **C** obsahujúca farebnú mapu aplikovanú na vykreslenie a možnosť nastaviť vlastnosti pomocou **properties** argumentu. Syntax funkcie je: `h = surf(X,Y,Z,C,properties...)`.



Obrázok 4: 3D graf

Tubeplot

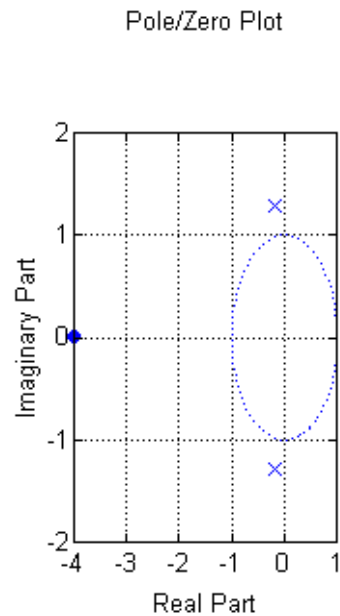
Funkcia **tubeplot** slúži na vykreslenie tubového grafu. Zápis **tubeplot(x,y,z,r,v,s)** obsahuje základné argumenty **x**, **y** a **z**, ktoré sú vektory opisujúce tubu a voliteľné argumenty **r** – vektor alebo skalár definujúci polomer tuby, **v** – vektor určujúci farebnú mapu použitú pri vykreslení a **s** – číslo definujúce počet tangenciálnych sekcií.



Obrázok 5: Tubový graf

Zplane

Je určená na vykreslenie grafu nuly – póly pre diskretný systém určený nulami a pólmi. Štandardný zápis je: **zplane(z,p)**, kde **z** a **p** sú vektory obsahujúce nuly (**z**) a póly (**p**) systému. Možný je zápis pomocou vektorov obsahujúcich koeficienty čitateľa **a** a menovateľa **b**: **zplane(a,b)**.



Obrázok 6: Zplane graf

Subplot

Nejedná sa o štandardnú funkciu zabezpečujúcu vykreslenie určitého typu diagramu. Služi predovšetkým na rozdelenie plochy určenej pre vykreslenie na dvojrozmernú mriežku, kde každá z častí mriežky môže obsahovať ľubovoľný typ vykreslenia z vyššie spomenutých. Existuje viacero spôsobov zápisu funkcie subplot: **subplot(row,col, num)** – vytvorí mriežku s rozmermi **row x col** a aktivuje časť určenú argumentom **num**. Číslovanie polí je po riadkoch. Ak potrebujeme vykreslenie grafu zaberajúceho viacero buniek mriežky je tvar funkcie: **subplot(row,col,[vec])**, kde **row** je počet riadkov a **col** je počet stĺpcov novej mriežky a argument **vec** je vektor určujúci bunky pokryté grafom.

10 ĎALŠIE MOŽNOSTI PROSTREDIA

Rozsah ponúkaných funkcií prostredia FreeMat je pomerne komplexný a okrem štandardnej práce s maticami a vizualizácie výsledkov inžinierskych výpočtov obsahuje celý rad iných funkcií. Kompletné rozdelenie podporovaných funkcií je v podkapitole 4.9 Skupiny funkcií.

I / O funkcie

Jedná sa o rozsiahlu kapitolu zahŕňajúcu vstupne výstupné funkcie určené pre prácu so zvukom, so súbormi – CSV i štandardnými, s HTML a XML dokumentmi, s URL, s konzolou, so zobrazením výstupov funkcií a pod..

Flow Control

Táto časť funkcií obsahuje základné konštrukcie spracovania chybových hlásení, programovania a riadenia toku a spracovania informácií ako sú podmienky a cykly a k nim doplnkové funkcie. Obsiahnuté sú funkcie – for, while, switch, if-else if-else, break, continue, error, lasterr, return, try-catch a warning.

Inšpekčné funkcie

Skupina inšpekčných funkcií zahŕňa celú radu funkcií určených na testovanie rôznych parametrov, vlastností objektov, zisťovanie prítomnosti určitých prvkov v štruktúrach, určovanie dátových typov a získavanie informácií o funkciách a premenných.

Konverzné funkcie

Sú určené na typovú konverziu medzi dátovými typmi podporovanými vo FreeMat.

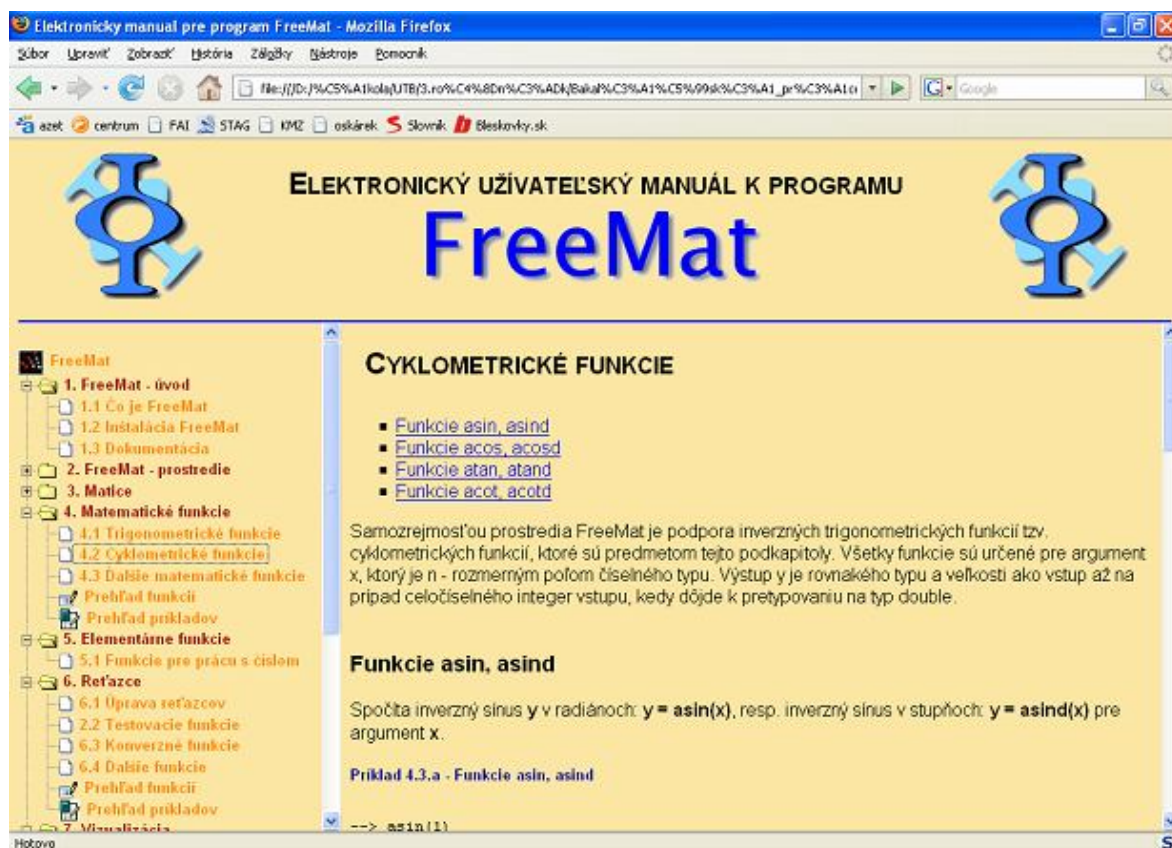
II. PRAKTICKÁ ČASŤ

11 ELEKTRONICKÝ MANUÁL

Elektronický manuál pre program FreeMat bol vytvorený vo forme *.html stránok. Teoretické podklady na tvorbu som získal z publikácie [2]. Graficky sú stránky rozčlenené na tri základné rámce. Táto forma bola použitá pre zvýšenie prehľadnosti stránok. Taktiež neboli použité žiadne nevyhnutné prvky, ktoré by mohli pôsobiť rušivo. Vrchný rámec má funkciu hlavičky a jeho obsah je nemenný. Tvorí ho názov a po stranách logá prostredia FreeMat. Ďalšie dva rámce sú „aktívne“ a ich obsah nie je stály. V ľavom rámci sa nachádza navigačné menu stránok vo forme stromovej štruktúry. Toto menu bolo vytvorené pomocou javaskriptu dTree 2.05. Tento javaskript je voľne k stiahnutiu z <http://www.destroydrop.com/javascripts/tree/>. Menu je vytvorené vo forme klasického rozbaľovacieho menu ako ho môžeme poznať z iných aplikácií. Položka menu, ktorá obsahuje nižšie úrovne je označená znamienkom plus pred jej názvom indikujúcim možnosť rozbalenia stromovej štruktúry tejto položky. Jej opätovné zbalenie vykonáme kliknutím na znamienko mínus pred názvom položky. Najnižšia úroveň v stromovej štruktúre, ktorú nie je možno rozbaľiť je uvedená iba názvom. Jednotlivé úrovne sú graficky rozlíšené, čo umožňuje ľahšiu orientáciu v menu. Pravý rámec stránky je hlavný rámec, v ktorom sa objavuje obsah jednotlivých kapitol po kliknutí na ich názov v menu. Jednotlivé stránky majú spoločnú úpravu, ktorá bola dosiahnutá použitím kaskádových štýlov, ktoré definujú základné vlastnosti jednotlivých objektov a prvkov použitých pri tvorbe stránok.

Samotný manuál je tvorený ôsmymi kapitolami rozdelenými podľa tematických okruhov. Toto delenie zodpovedá deleniu použitému v teoretickej časti práce. V jednotlivých kapitolách sú teoretické podklady doplnené množstvom názorných a vysvetľujúcich príkladov graficky oddelených od bežného textu. Zároveň je na konci väčšiny kapitol prehľadný zoznam funkcií opísaných v danej kapitole, resp. jej podkapitolách aj s ich krátkym popisom. Tento prehľad funkcií je možné zobraziť v abecednom usporiadaní alebo usporiadaní podľa tematických celkov. Taktiež sa v závere väčšiny kapitol nachádza zoznam príkladov s popisom ozrejmujúcim obsah príkladu. V prípade, že podkapitola obsahuje viacero funkcií sú v úvode podkapitoly pod nadpisom umiestnené odkazy na tieto funkcie. Prvá kapitola je úvodná a obsahuje podkapitoly venované inštalácii prostredia FreeMat na rôzne platformy a priblíženie a zoznámenie sa s prostredím. V druhej kapitole človek preniká hlbšie do podstaty práce vo FreeMat a zoznamuje sa s princípmi práce v ňom a to s členením okna, ovládaním príkazového riadka, s pomocníkom, dátovými

typmi používanými v programe, ponúkacími funkciami, špeciálnymi konštantami a premennými. Tretia kapitola tvorí jadro práce, pretože pojednáva o základnom prvku prostredia a to o maticiach. Venuje sa ich vytváraniu, indexácii, úprave a funkciám určeným pre prácu s maticami a pod.. Taktiež je v tejto kapitole obsiahnutá práca s polynómami. Obsahom štvrtej kapitoly je priblíženie matematických funkcií ponúkaných prostredím FreeMat. Obsiahnuté sú tu goniometrické funkcie, cyklometrické funkcie ako aj iné užitočné matematické funkcie. Piatu kapitolu tvoria tzv. elementárne funkcie predovšetkým určené pre prácu s číslom, jeho zaokrúhľovaním, určením absolútnej hodnoty a komplexne združeného čísla. Ďalšou pomerne rozsiahlou kapitolou je kapitola šiesta venujúca sa reťazcom. V jej jednotlivých kapitolách sú opísané základné funkcie pre prácu s reťazcami, ich tvorenie, úpravu a pod.. Siedma kapitola sa venuje vizualizácii. Zoznamuje záujemcu s filozofiou tvorby vizualizácii vo FreeMat, so základnými grafickými prvkami, ich vlastnosťami a použitím. Záverečná ôsma kapitola je venovaná priblíženiu problematiky programovania. Obsahuje podkapitoly venované tvorbe funkcií, skriptov ako aj priblíženie základných riadiacich konštrukcií, ktorými sú cykly a podmienky.



Obrázok 7: Ukážka elektronického manuálu

ZÁVER

Jedným z cieľov tejto práce bolo ukázať čitateľovi, že v súčasnej dobe je vďaka internetu k dispozícii mnoho prostredí určených pre inžinierske a vedecké výpočty, ktoré sú dostupné bežnému používateľovi z radov študentov a jednotlivcov, ktorí si nemôžu dovoliť obstaráť výkonné licencované a platené prostredia. Jedným z takýchto programov je FreeMat, ktorý je voľne stiahnuteľný z domovskej stránky. Ten rozsahom ponúkaných funkcií dokáže konkurovať plateným prostrediam. Ovládaním a spôsobom práce je podobný prostrediu Matlab, ktoré je v dnešnej dobe považované za štandard. Z tohto dôvodu nebude pre používateľa prípadný prechod medzi prostrediami problémový. Na rozdiel od svojho vzoru je nevýhodou FreeMat, ktorá často krát odradí potenciálnych záujemcov o prácu v ňom, neexistencia dostatočného množstva materiálov zaoberajúcich sa prácou v tomto prostredí. A práve táto skutočnosť bola kľúčová pri vytváraní témy tejto práce.

Podstatou tejto práce bolo totiž vytvoriť elektronický manuál zaoberajúci sa základmi práce vo FreeMat a priblížiť tak toto prostredie potenciálnym užívateľom, ktorí nie sú natoľko jazykovo zdatní, aby mohli pracovať s originálnou dokumentáciou napísanou tvorcom prostredia v anglickom jazyku.

Výsledkom je jednoducho a prehľadne, avšak poučne vytvorený manuál, v ktorom sú obsiahnuté a popísané základné funkcie a možnosti prostredia FreeMat. Manuál je tematicky rozdelený do viacerých kapitol zameraných na objasnenie jednotlivých stavebných prvkov prostredia. Pre lepšie pochopenie je uvedených množstvo príkladov.

Vďaka mnohým nadšencom z celého sveta, spolupracujúcim na neziskových projektoch typu FreeMat, čaká takéto prostredia jasná budúcnosť.

ZÁVER V ANGLIČTINE

One of the main goals of this paper work was to show to the reader that in present are due to internet available a lot of environments intended for engineering and scientific calculations, which are available for ordinary users including students and individuals, which can't effort effective licensed and paid environments. One of this software is FreeMat, which can be tax free downloaded from his home page. This one can be serious rival to the paid environments in size of offered built in functions. The way of work is similar to Matlab environment, which is in present considered as standard. Hence it won't be a problem for the user to change the environments. Unlike the Matlab, the FreeMat has one disadvantage for possible users and it is absence of sufficient number of suitable materials. This was the key factor by creating the theme of this bachelor thesis.

The basis of this paper work was to create electronic manual about basics of work in FreeMat. It was also the way how to approach this environment to possible users, which aren't able to work with original documentation written by the author.

The result of this bachelor thesis is simply and clearly understanding manual with fundamental functions and possibilities of FreeMat environment. The manual consists of chapters focused on particular elements of FreeMat and for better understanding it is supplemented with a number of examples.

Due to amateurs from all over the world contributing on non-profit projects like FreeMat, we can predict a bright future for these environments.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] BASU, Samit. *FreeMat v3.6 Documentation*. [s.l.] : [s.n.], 2008. 566 s. Dostupný z WWW: <<http://freemat.sourceforge.net/FreeMat-3.6.pdf>>.
- [2] KOSEK, Jiří. *HTML tvorba dokonalých WWW stránek : Podrobný průvodce*. Ondřej Tůma. 1. vyd. Praha : Grada Publishing, spol. s. r. o., 1998. 296 s. ISBN 80-7169-608-0.
- [3] BASU, Samit, et al. *FreeMat : Home* [online]. 2004 , Monday, March 17, 2008, 01:03:19 PM (CET) [cit. 2008-04-14]. Dostupný z WWW: <<http://freemat.sourceforge.net/index.html>>.
- [4] BASU, Samit, et al. *FreeMat : FAQ* [online]. 2004 , Monday, March 17, 2008, 01:03:19 PM (CET) [cit. 2008-04-14]. Dostupný z WWW: <<http://freemat.sourceforge.net/faq.html> >.
- [5] *Wikipedia, the free encyclopedia : FreeMat* [online]. 2006 , 1 May 2008, at 11:26 [cit. 2008-04-14]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/FreeMat>>.
- [6] DEMETRIOS, Kyriakis, SAMIT, Basu. *Freshmeat : FreeMat 3.6* [online]. 2002 , Wed, Mar 19th 2008 08:09 PDT [cit. 2008-04-14]. Dostupný z WWW: <http://freshmeat.net/projects/freemat/?branch_id=52138&release_id=273989>.
- [7] PERINGER. *Náměty na projekty* [online]. 2007 , 5. listopadu 2007 [cit. 2008-04-14]. Dostupný z WWW: <<http://www.fit.vutbr.cz/~peringer/UNOFFICIAL/vyuka/Projekty.html>>.
- [8] FRIEDRICH, Václav. *Moodle.vsb.cz : FREE statistické a matematické programy* [online]. 2008 IIV [cit. 2008-04-14]. Dostupný z WWW: <<http://moodle.vsb.cz/moodle/mod/glossary/view.php?id=25707&mode=date&hook=&sortkey=CREATION&sortorder=asc&fullsearch=0&page=2>>.
- [9] VRÁBEL, Martin, et al. *MacPlanet : Matlab - matematika pre Mac OS X* [online]. 2006 [cit. 2008-04-14]. Dostupný z WWW: <http://www.macplanet.sk/portal/?page_id=1>.
- [10] *PULIB* [online]. 2005 , 17-Mar-2006 13:06 [cit. 2008-04-14]. Dostupný z WWW: <<http://www.pulib.sk/elpub/FHPV/Pavelka1/5.pdf>>.
- [11] BUŠA, Ján. *SCILAB* [online]. 2007 [cit. 2008-04-14]. Dostupný z WWW: <<http://people.tuke.sk/jan.busa/kega/scilab/scilab.pdf>>.

- [12] BUŠA, Ján. *OCTAVE* [online]. 2007 [cit. 2008-04-14]. Dostupný z WWW: <<http://people.tuke.sk/jan.busa/kega/octave/octave.pdf>>.
- [13] BELÁKOVÁ, Stanislava. *Citadel.ukf : konferencia* [online]. 2007 , 20-Mar-2007 17:00 [cit. 2008-04-14]. Dostupný z WWW: <http://citadel.ukf.sk/konferencia/papers/PDF_Matematika/Belakova.pdf>.

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

napr.	například
atd.	a tak d'alej
EM	elektronický manuál
OD	originálna dokumentácia
ans	answer
GPL	General Public License
OS	operačný systém
IDL	Interactive Data Language
MIT	Massachusetts Institute of Technology
a i.	a iné
a pod.	a podobne

ZOZNAM OBRÁZKOV

Obrázok 1: Pracovná plocha FreeMat	17
Obrázok 2: Obrysový graf	54
Obrázok 3: Štandardný graf	55
Obrázok 4: 3D graf.....	56
Obrázok 5: Tubový graf	57
Obrázok 6: Zplane graf.....	58
Obrázok 7: Ukážka elektronického manuálu	62

ZOZNAM TABULIEK

Tabuľka 1: Špeciálne konštanty	21
Tabuľka 2: Operátory	22

ZOZNAM PRÍLOH

Príloha P I: Zoznam rozobraných funkcií

Príloha P II: CD-ROM

PRÍLOHA P I: ZOZNAM ROZOBANÝCH FUNKCIÍ

abs	det	log	sizefig
acos	diag	log10	sort
acosd	dot	log2	sqrt
acot	exit	loglog	strcmp
acotd	exp	logspace	strcmpi
asin	eye	lower	strfind
asind	figlower	ones	strncmp
atan	figraise	pathtool	strrep
atand	figure	plot	strtrim
axes	fix	plot3	subplot
axis	floor	poly	sum
ceil	gca	polyder	surf
cellstr	gcf	polyfit	tan
clf	hcontour	polyint	tand
close	help	polyval	tubeplot
conj	helpwin	print	upper
contour	himage	quit	who
contour3	hline	rank	xlabel
copy	hsurface	roots	xlim
cos	htext	round	ylabel
cosd	inv	semilogx	ylim
cot	isalpha	semilogy	zeros
cotd	isdigit	sin	zlabel
cross	isspace	sind	zlim
deblank	linspace	size	zplane

PRÍLOHA P II: CD – ROM

Súčasťou práce je priložený CD disk obsahujúci samotný elektronický užívateľský manuál ako i bakalársku prácu vo formáte *.doc i vo formáte *.pdf. Elektronický manuál je uložený v zložke FreeMat_EM, ktorá obsahuje ďalšie podzložky potrebné pre správne fungovanie manuálu. Samotný manuál sa spúšťa pomocou súboru index.html nachádzajúceho sa priamo v zložke FreeMat_EM.