

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Porovnání grafických výstupů z Matlabu a GNUPLoTtu

Jaroslav Beza

Bakalářská práce

2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jaroslav Beza**
Osobní číslo: **I10014**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Porovnání grafických výstupů z Matlabu a GNUPLOTu**
Zadávající katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

V teoretické části BP budou popsány možnosti grafických výstupů z výpočtového softwaru Matlab. Dále bude popsána práce s programem GNUPLOT, zásady pro jeho ovládání a popis vlastností grafického výstupu. Na závěr teoretické části budou grafické výstupy z Matlabu a GNUPLOTu vzájemně porovnány v závislosti na parametrech, složitosti vykreslení a pod. Cílem implementační části je vytvoření uživatelské příručky pro generování grafických výstupů v Matlabu a v GNUPLOTu zahrnující zkrácený popis teoretických výsledků včetně ilustračních příkladů. V rámci porovnání Matlabu a GNUPLOTu a tvorby ilustračních příkladů budou generovány různé grafické výstupy a bude sledována jejich náročnost na naprogramování a časová náročnost vykreslení.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

DUŠEK, F. Matlab a Simulink ? úvod do používání. Pardubice, 2000. ISBN 80-7194-273-1.

JANERT, Philipp K. Gnuplot in Action - Understanding Data with Graphs, Manning Publications, 2009. ISBN 978-1933-988-399.

GNU PLOT homepage <http://www.gnuplot.info/>

Vedoucí bakalářské práce:

Mgr. Jana Heckenbergerová, Ph.D.

Katedra matematiky a fyziky

Datum zadání bakalářské práce:

21. prosince 2012

Termín odevzdání bakalářské práce:

10. května 2013



prof. Ing. Simeon Karamazov, Dr.
děkan



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 08. 05. 2013

Jaroslav Beza

Poděkování

Tímto bych chtěl poděkovat Mgr. Janě Heckenbergerové, Ph.D., vedoucí mé bakalářské práce za rady a pomoc s touto prací. Dále bych chtěl poděkovat rodině a spolužákům za rady a podporu, kterých si moc vážím.

Anotace

Tato práce se zabývá možnostmi grafických výstupů z výpočtového softwaru Matlab s porovnáním grafických výstupů z výpočtového softwaru Gnuplot. Dále zde bude popsána práce s programem Gnuplot a jeho zásady pro ovládání.

Výsledkem práce bude uživatelská příručka pro generování grafických výstupů v Matlabu a Gnuplotu včetně ilustračních příkladů a popisu.

Klíčová slova

Matlab, Gnuplot, Grafické výstupy, Vizualizace

Title

Comparison of graphical outputs from Matlab and Gnuplot

Annotation

This work deals with the possibilities of graphical outputs of computational software Matlab comparing with the graphical output of the calculation software Gnuplot. In addition, it will describe the work with the program Gnuplot and its principles for control.

Result of this work is a user's manual for generating graphical output in Matlab and Gnuplot including illustrative examples and description.

Keywords

Matlab, Gnuplot, Graphical outputs, Visualization

Obsah

Seznam obrázků.....	8
Seznam tabulek.....	8
1 Úvod.....	9
2 Matlab.....	10
3 Gnuplot.....	10
3.1 Spuštění programu.....	11
3.2 Základní informace.....	11
3.3 Náповěda.....	12
3.4 Výstupní soubor	12
3.5 Komentáře	12
3.6 Souřadnice a kreslení základních objektů	12
3.7 Datové soubory.....	13
3.8 Výrazy	15
3.9 Operátory.....	16
3.9.1 Unární	16
3.9.2 Binární	16
3.9.3 Ternární	17
3.10 Funkce – sum, random	17
3.10.1 Sum.....	17
3.10.2 Random.....	17
3.11 Uživatelsky definované proměnné a funkce	17
3.11.1 Uživatelsky definované proměnné	18
3.11.2 Uživatelsky definované funkce	18
3.12 Blok „if / else“, cyklus „while / do“	18
3.12.1 Bloky „if / else“	18
3.13 Čas / datum.....	19
3.14 Typy čar, barvy, styly, barva pozadí, barva v proměnné	19
3.15 Rozdíl mezi linetypes a linestyles	21
3.16 Vykreslování grafu	21
3.17 Styly vykreslování.....	23

3.17.1	Boxes	23
3.17.2	Boxerrorbars	24
3.17.3	Boxxyerrorbars	24
3.17.4	Boxplot	25
3.17.5	CandleSticks	25
3.17.6	Circles	26
3.17.7	Ellipses	26
3.17.8	Dots.....	27
3.17.9	Filledcurves	27
3.17.10	FinanceBars	28
3.17.11	Steps, Fsteps.....	28
3.17.12	Histeps	29
3.17.13	Histograms	30
3.17.14	Newhistogram	32
3.17.15	Impulses, Lines, Linespoints, Points	33
3.17.16	Labels	35
3.17.17	Vectors	35
3.17.18	Xerrorbars, Yerrorbars	36
3.17.19	Xyerrorbars	36
3.17.20	Xerrorlines, Yerrorlines, Xyerrorlines	37
4	Praktická část:	38
4.1	Porovnání grafických výstupů včetně zdrojových kódů	38
4.1.1	Teplotní závislosti	38
4.1.2	Histogramy	41
4.2	Časová náročnost výpočtů.....	43
4.2.1	Teplotní závislosti	43
4.2.2	Histogramy	44
5	Závěr.....	45
	Literatura	46

Seznam obrázků

Obrázek 1 - Vykreslovací styl Boxes	23
Obrázek 2 - Vykreslovací styl Boxerrorbars	24
Obrázek 3 - Vykreslovací styl Boxyerrorbars	24
Obrázek 4 - Vykreslovací styl Boxplot	25
Obrázek 5 - Vykreslovací styl CandleSticks	25
Obrázek 6 - Vykreslovací styl Circles	26
Obrázek 7 - Vykreslovací styl Ellipses.....	26
Obrázek 8 - Vykreslovací styl Dots.....	27
Obrázek 9 - Vykreslovací styl Filledcurves	27
Obrázek 10- Vykreslovací styl Financebars	28
Obrázek 11 - Vykreslovací styl Steps.....	29
Obrázek 12 - Vykreslovací styl Fsteps	29
Obrázek 13 - Vykreslovací styl Histeps	29
Obrázek 14 - Vykreslovací styl Histograms (1)	30
Obrázek 15 - Vykreslovací styl Histograms (2)	30
Obrázek 16 - Vykreslovací styl Histograms (3)	31
Obrázek 17 - Vykreslovací styl Histograms(4)	31
Obrázek 18 - Vykreslovací styl Newhistogram.....	32
Obrázek 19 - Vykreslovací styl Impulses.....	33
Obrázek 20 - Vykreslovací styl Lines	33
Obrázek 21 - Vykreslovací styl Linespoints.....	34
Obrázek 22 - Vykreslovací styl Points	34
Obrázek 23 - Vykreslovací styl Labels.....	35
Obrázek 24 - Vykreslovací styl Vectors.....	35
Obrázek 25 - Vykreslovací styl Yerrorbars	36
Obrázek 26 - Vykreslovací styl Xerrorbars	36
Obrázek 27 - Vykreslovací styl Xyerrorbars	36
Obrázek 28 - Vykreslovací styl Xyerrorlines	37
Obrázek 29 - Grafická metoda určení ztráty napětí vodiče v tahu (Matlab)	39
Obrázek 30 - Grafická metoda určení ztráty napětí vodiče v tahu (Gnuplot)	41
Obrázek 31 - Zmenšená verze – Gnuplot	42
Obrázek 32- Zmenšená verze - Matlab.....	41
Obrázek 33 - Frekvenční histogram proudové zatížitelnosti vodiče (Matlab)	42
Obrázek 34 - Frekvenční histogram proudové zatížitelnosti vodiče (Gnuplot)	43

Seznam tabulek

Tabulka 1- Unární operátory	16
Tabulka 2 - Binární operátory	16
Tabulka 3 - Ternární operátory.....	17

1 Úvod

Bylo zjištěno, že softwarový program Matlab, který slouží mimo jiné ke generování grafických výstupů, si v některých situacích nebyl schopen poradit tak, jak by to běžný uživatel očekával. Některé, pro uživatele podstatné informace, nebyly tak zřetelné, jak by měly být, z tohoto důvodu vzniká téma této bakalářské práce.

Snahou této práce je představit program Gnuplot, který se touto problematikou vizualizace výsledků taktéž zabývá. Program Gnuplot je k dispozici jako free software, bezproblémově běží na všech hlavních operačních systémech a standardně je součástí většiny distribucí Linuxových systémů.

Cílem bude popsat zadávání příkazů a následné generování grafických výstupů jak v programu Matlab tak v programu Gnuplot, celá práce by měla být postavena na porovnání těchto programů.

Hlavním cílem je vytvořit jednoduchou a přehlednou uživatelskou příručku k softwaru Gnuplot, která bude sloužit jak pro uživatele, kteří nemají s programem žádné zkušenosti, tak i pro uživatele pokročilejší.

Příručka nabídne uživatelům příkazy, které jsou potřeba k ovládní Gnuplotu spolu s ukázkovými příklady, v některých případech doplněných i ilustračními obrázky.

2 Matlab

Program Matlab, vyvíjen a spravován firmou MathWorks, bezesporu patří mezi nejvíce rozšířený skriptovací a programovací jazyk, který je využíván pro vědecké a výzkumné účely v odlišných odvětvích vědy a techniky a pro studentské a zaměstnanecké účely na vysokých školách. Pomocí něj jsme v dnešní době schopni zpracovat jakákoliv data do podoby grafů, které nám slouží pro vizualizaci našich prací.

Nejen tímhle je program Matlab celosvětově známý, dále uživatelům nabízí i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Tyto knihovny mohou uživatelé využít v podstatě ve všech okruzích lidské činnosti.

Za nejsilnější stránku Matlabu považují vědci jeho mimořádně rychlé výpočetní jádro, které je opatřeno optimálními algoritmy, které jsou po celá léta testovány a prověřovány specialisty v tomto oboru.

Implementace v dnešních nejznámějších operačních systémech (Windows, Linux, Mac, Solaris) je pro tento program samozřejmostí.

3 Gnuplot

Gnuplot se zabývá problematikou vizualizace výsledků. Disponuje jak zpracováním funkcí (knihovných či uživatelem definovaných), tak i zpracováním dat ze souborů. Číselné hodnoty v datových souborech mohou být v podstatě v jakékoliv struktuře, Gnuplotu stačí říct, jaká čísla reprezentují jaké hodnoty a program si je sám přebere a použije. Dále umožňuje uživateli navrhnout vzhled grafu co možná nejvíce podle jeho představ.

Původně byl vytvořen pro studenty a vědce k vizualizaci jejich výsledků, v dnešní době už se rozrostl pro podporu web skriptů. Dále je využit jako kreslicí nástroj do aplikací třetích stran jako je Octave. Jeho vývoj se počítá od roku 1986.

Podporuje mnoho typů vizualizace ve 2D a 3D prostorech. Umí vykreslovat pomocí čar, bodů, různých tvarů, pracuje s vektory, povrchy i s různými variacemi textu. Dále podporuje mnoho typů výstupních zařízení jako obrazovkové terminály, přímý výstup na plottery nebo moderní tiskárny. Samozřejmě nabízí veliké spektrum výstupních formátů – EPS, EMF, JPEG, LaTeX, PDF, PNG, POSTSCRIPT, atd.

3.1 Spuštění programu

Nezávisle na druhu operačního systému (Linux, Windows) je možné pustit program ve dvou režimech – Interaktivní nebo Dávkový. Je-li program spuštěn s parametry, tak se tyto parametry budou brát jako názvy vstupních souborů a budou se vykonávat postupně, v „dávkách“. Interaktivní mód lze zapnout přepínačem: pomlčka „-“, který dává uživateli možnost zásahu do průběhu vykonávání makra.

Standardně se program spouští bez parametrů, kde se objeví tzv. prompt (pro pokročilejší uživatele), standardně známý jako příkazový řádek. Pro zajímavost: při spuštění Gnuplotu v standardním příkazovém řádku se nám program pustí právě uvnitř a dále s ním pracujeme pouze v rámci příkazové řádky (CMD).

Spuštění v interaktivním režimu:

```
gnuplot
```

Spuštění v dávkovém režimu:

```
gnuplot input1 input2
```

Předání příkazů přímo do příkazového řádku, příkaz „-persist“ dává možnost zanechání výsledku na obrazovce, příkaz“-e“ slouží ke zvýšení priority příkazu:

```
gnuplot -persist -e "set title 'sin(x)'; plot sin(x)"
```

3.2 Základní informace

- použitím znaku vykřičník „!“ je možné využít příkazů klasického shellu, který je znám z Unixových systémů (příklad: „!ls“ pro vypsání aktuálního adresáře).
- program Gnuplot je „Case sensitive“ – v příkazech jsou rozlišována malá a velká písmena,
- řetězce mohou být uvedeny jak do apostrofů, tak i do uvozovek (příklad: ‚data‘ nebo „data“), jejich zpracování je však odlišné. V apostrofech znamenají posloupnost znaků; Uvozovky jsou nahrazovány znakem „\n“ jako indikátor pro nový řádek,
- příkazy gnuplotu mívají povinné parametry, jejichž pořadí a správnost je nutno dodržet. Toto neplatí u nepovinných parametrů, které není nutno uvádět,
- jeden řádek může obsahovat libovolný počet příkazů, tyto příkazy je nutno oddělit středníkem,
- neukončený příkaz je indikován zpětným lomítkem („\“) na konci každého neukončeného řádku,
- složené závorky „{ }“ označují volitelné argumenty a vertikální bar „|“ odděluje vzájemně exkluzivní výběr.

3.3 Nápověda

S instalací je dodán obsáhlý manuál, který můžete nalézt ve složce „/docs“ jako gnuplot.pdf.

K nápovědě se dá dostat i přes příkazy „h“, „help“, „?“ . Nápověda je zde velice dobře popsána, pomoc pro určitý příkaz lze vyvolat dopsáním hledaného příkazu za klíčové slovo „help/h“ . Příklad: „h plot“, „help plot“ apod.

3.4 Výstupní soubor

Následujícím příkazem lze nastavit velikost výstupního souboru, v tomto případě bude celé plátno vyplněno grafem.

```
set term <terminal type> size <XX>, <YY>
```

Doplněním následujícím příkazem lze určit velikost grafu vůči plátnu.

```
set size <XX>, <YY>
```

Pro představu:

```
set size 1,1 -          vykreslí graf přes celé plátno (defaultní volba)
set size 0.5,0.5 -     Zmenší X, Y souřadnice na polovinu, graf se tedy vykreslí
                        do levé spodní čtvrtiny celého plátna.
```

Příklad:

```
set size 0.5,0.5
set term png size 600, 400
set output "vystup.png"
plot sin(x)
```

Příkaz zajistí, že grafem bude „sin (x)“, vykreslen bude do souboru „vystup.png“, velikost souboru bude 600x400 px a graf bude umístěn v levé spodní polovině plátna.

3.5 Komentáře

Komentáře jsou v Gnuplotu reprezentovány znakem „#“, vše co se nachází za tímto znakem, bude Gnuplot ignorovat.

3.6 Souřadnice a kreslení základních objektů

Souřadnicové systémy jsou děleny na systém `first`, `second`, `graph`, `screen` a `character`.

Systém „first“ je definován levou a spodní osou (x, y), systém „second“ je definován pravou a horní osou (x2,y2). Systém „graph“ specifikuje oblast mezi jednotlivými osami – [0,0] je levá spodní osa a [1,1] je pravá horní osa. „Screen“ systém je definován pro celou oblast, tím je však myšlena celá oblast, ne jen oblast zvolená příkazem „set size“. „Character“ dává pozici od počátku [0,0] podle šířky a výšky znaku, zde záleží na zvolené velikosti fontu. Ke kreslení v polárních souřadnicích slouží příkaz „set polar“. Za pomoci zvolených souřadnicových systémů máme možnost kreslit na jakoukoliv pozici různé objekty:

- Kreslení šipek pomocí příkazu „set arrow“. V následujícím příkazu je šipka kreslena relativně k počátečnímu bodu a její vzdálenost je uvedena v souřadnicovém systému „graph“.

```
set arrow from 0,-5 rto graph 0.1,0.1
```

- Při vykreslení grafu je u něj automaticky generována legenda, která tento graf popisuje. Legendu můžeme vypnout a nastavit si svoji podle potřeby. Vypneme ji příkazem „unset key“. Pomocí příkazu „set key“ si pak nastavíme svoji vlastní. Následující příkaz zajistí, že legenda bude obsahovat titulek „Legenda“ zarovnaný doprava, bude umístěna vpravo dole a bude mít ohraničení čarou o tloušťce 3.

```
set key right bottom Left title 'Legenda' box 3
```

- Příkaz „set label“ slouží k vypsání popisku. Následující příkaz vypíše „y = 6“ přesně na pozici [2,5] v souřadnicovém systému „first“.

```
set label "y = 6" at 2,5
```

- Příkaz „set object“ umožňuje vykreslit jakýkoliv objekt, buďto svůj vlastní nebo již v Gnuplotu přednastavený, mezi které se řadí obdélník, kruh, elipsa a polygon. Každý objekt pak může být ohraničen i vyplněn. Následující příkaz vykreslí obdélník přes celé plátno, parametrem „behind“ určíme, že se obdélník vykreslí za všechny objekty, které jsou již vykresleny.

```
set object rectangle from screen 0,0 to screen 1,1 behind
```

3.7 Datové soubory

Obecný zápis příkazu plot s použitím datového souboru vypadá následovně:

```
plot [rozsahy] "soubor" specifikace dat [axes] typ souřadnic [title]
"popis" [with] styl vykreslení
```

Data v souboru mohou být v jakékoliv struktuře, vše co je zapotřebí, je použít pár prepínačů, kterými se určí, jak s daty pracovat. Každý řádek by měl specifikovat jeden bod, který je určen souřadnicemi [x, y], kde souřadnice x není povinná, Gnuplot si ji dopočítá sám, souřadnice y však musí být stanovena. Prázdným řádkem v souboru určíme, že ve výsledku bude funkce nespojitá. Dvěma prázdnými řádky po sobě pak rozdělíme soubor na dva podsoubory, kde se s daty pracuje jako ve dvou odlišných souborech.

Příkladem je soubor „data.txt“, který má 4 sloupce, přičemž ve druhém sloupci je text:

```
1.000 "text" 20.000 5.000
```

Vykreslení grafu pak může vypadat následovně:

```
plot 'datafile' using 1:2:4 with labels
```

Ke složitěji uloženým datům jsou pak potřeba různé prepínače, které umožní jejich zpracování. Jednotlivé prepínače:

- `index` - umožňuje vykreslit jen uživatelem požadované podsoubory („`index M:N:K`“), pokud je zadán pouze parametr „`M`“, zpracuje se podsoubor na `M` - tém místě. Při zadání „`M:N`“ jsou určeny hraniční soubory, které se mají zpracovat. Přidáním kroku, tedy „`M:N:K`“, se zpracují soubory od indexu `M` po `N` s krokem `K`,
- `every` - umožňuje vybrat jak podsoubory, tak jednotlivé body. („`every inkrement_bod : blok : první_bod : blok : poslední_bod : blok`“),
- `using` - asi nejsložitější prepínač, který umožňuje vybrat i složitěji zapsaná data v souboru. Ukázka nejlépe rovnou na příkladu:

```
plot "data.txt" using 1:($2+$3) `7lf,%lf,%lf`
```

vybere z prvního sloupce souřadnici x, součet druhého a třetího řádku nám dá souřadnici y a všechny hodnoty jsou oddělené čárkou. Další parametry:

- `%*lf` – přeskočí jedno číslo,
- `%lf` – načte číslo
- `%*10[^\n]` – přeskočí 10 znaků, které nejsou koncem řádku,
- `Smooth` - slouží k proložení bodů křivkou. Každá křivka potřebuje určitý počet bodů k proložení. Příkaz („`plot „data“ smooth csplines`“) proloží body přirozenou kubickou spline křivkou. Dále je možnost proložení Bezierovou křivkou stupně `N`, polynom 1. stupně, atd.

3.8 Výrazy

S výrazy v Gnuplotu se pracuje podobně jako v programovacích jazycích Pascal, jazyk C, Basic apod. Mezery a tabulátory mezi jednotlivými výrazy jsou ignorovány.

Gnuplot pracuje s číselnými typy „integer“ a „real“, kde integer je definován kupříkladu jako „20“, „-350“ atd., zatímco real má podobu jako „20.0“, „-350.0“, „25e-10“ atd.

Použití odlišného číselného se nám odrazí například u dělení čísel:

Integer: 3/2 = 1

Real: 3.0/2.0 = 1.5

Při použití obou číselných typů má typ real přednost před typem integer:

3/2e0 = 1.5

Komplexní čísla jsou vyjádřena reálnou a imaginární složkou, zapisujeme je do složených závorek, kde reálná a imaginární složkou musí být číselná konstanta. Ať tato čísla zadáme v jakékoliv formě, vždy budou brána jako reálný typ.

{5,3} - zápis komplexního čísla „5 + 3i“

{0,1} - zápis komplexního čísla „i“

Mezi výrazy můžeme řadit i práci s řetězci. V následujícím příkladu vidíme zřetězení dvou řetězců „A“ a „B“ pomocí znaku „“, které jsou pak rovny řetězci „AB“ reprezentováno příkazem „eq“:

„A“ . „B“ eq „AB“

S řetězcem, který obsahuje číslo, se pracuje stejně jako s číselnou hodnotou:

„3“ + „2“ = 7

Použití číselného typu integer nám může pomoci například při skládání názvu souboru apod. Reálná a komplexní čísla nám to nepovolují.

„soubor“ . 4 eq „soubor4“

Podřetězce můžou být dále vybrány pomocí vzoru „řetězec“[začátek:konec].

Příklad:

„ABCDEFGH“[5:6] vybere řetězec „EF“

„ABCDEFGH“[6:*] vybere řetězec „FGH“

3.9 Operátory

Operátory jsou odvozeny od programovacího jazyku C. Obecně platí, že při použití závorek vznikají priority při výpočtech. Pro příklad:

$$-4*2 = -8 \text{ zatímco } (-4)*2 = 8$$

3.9.1 Unární

Tabulka 1- Unární operátory

Symbol	Použití	Popis
+	+ a	Unární plus
-	- a	Unární mínus
~	~ a	Doplňěk (*)
!	! a	Logická negace (*)
!	a !	Faktoriál (*)
\$	\$3	Volá argument / sloupec (*)

(*) – takto značené operátory vyžadují číselný typ „Integer“

3.9.2 Binární

Tabulka 2 - Binární operátory

Symbol	Použití	Popis
**	a**b	Mocnina
*	a*b	Násobení
/	a/b	Dělení
%	a%b	Dělení se zbytkem (*)
+	a+b	Sčítání
-	a-b	Odčítání
==	a==b	Rovnost
!=	a!=b	Nerovnost
<,<=	a<b, a<=b	Menší než, menší nebo rovno
>,>=	a>b, a>=b	Větší než, větší nebo rovno
&	a&b	Bitový AND (*)
^	a^b	Bitový exkluzivní OR (*)
	a b	Bitový inkluzivní OR (*)
&&	a&&b	Logický AND (*)
	a b	Logický OR (*)
=	a=b	Přiřazení
,	(a,b)	Postupný výpočet
.	A . B	Sjednocení řetězců
eq	A eq B	Rovnost řetězců
ne	A ne B	Nerovnost řetězců

(*) – takto značené operace vyžadují číselný typ „Integer“

3.9.3 Ternární

Tabulka 3 - Ternární operátory

Symbol	Použití	Popis
?:	a?b:c	Ternární operátor

Příklad vykreslí funkci „sin (x)“ v rozmezí $0 \leq x < 1$, f-ci $1/x$ v rozmezí $1 \leq x < 2$. Zbytek nebude definovaný.

```
f (x) = 0<=x && x<1 ? sin(x) : 1<=x && x<2 ? 1/x : 1/0
```

```
plot f(x)
```

3.10 Funkce – sum, random

3.10.1 Sum

Syntaxe: `sum [<proměnná> = <začátek> : <konec>] <výraz>`

<proměnná> je číslo typu Integer, které nabývá v intervalu <začátek> : <konec>, každým cyklem se přičte hodnota právě procházené položky a ve výsledku se pak zobrazí celkový součet (suma).

Příklad: Interval pro proměnnou „i“ je <1:10>. V každém cyklu je proměnná „i“ zvýšena o 1. Výsledek je 55:

```
sum [i=1:10] i
```

3.10.2 Random

Funkce „rand ()“ generuje pseudo - náhodná čísla v intervalu 0 až 1.

`rand (0)` – vrací pseudo – náhodné číslo v intervalu <0:1> generované z aktuální hodnoty dvou 32bitových jader

`rand (-1)` – resetuje obě jádra na standardní hodnoty

`rand (x)` – pro $0 < x < 2^{31}-1$ nastaví obě hodnoty jako „x“

`rand ({x,y})` – pro $0 < x, y < 2^{31}-1$ nastaví hodnotu prvního jádra jako „x“ a hodnotu druhého jádra jako „y“

3.11 Uživatelsky definované proměnné a funkce

Uživatelsky definované proměnné a funkce mohou být deklarovány a použity kdekoliv, kde je zapotřebí. Zde stojí za zmínku proměnné „pi – 3.14159...“ a „NaN – Not a Number“, které jsou v Gnuplotu již definovány. Pro vlastní potřeby je samozřejmě můžeme měnit, k obnovení původních hodnot slouží nastavení: `pi = GPVAL_pi`, `NaN = GPVAL_NaN`.

3.12.2 Cyklus „while / do“

```
while (<expr>) {  
    <commands>  
}
```

Příklad:

```
x=0  
while (x! =5) {print x; x = x+1}
```

3.13 Čas / datum

Gnuplot pracuje s časovými údaji jako se vstupními daty. Veškeré časy a data jsou převáděny na sekundy, které jsou počítány od roku 2000. K nastavení časových os slouží příkazy „set xdata time“ a „set ydata time“. Příkazem „set timefmt“ definujeme formát všech vstupních dat. Pokud je v datovém souboru uvedena osa X a osa Y jako čas nebo datum, musí být ve stejném formátu.

Příklad: Předpokládejme následující podobu datového souboru:

```
05/15/98 20:00 7.08e24
```

Tento soubor můžeme vykreslit následovně:

```
set xdata time                #nastaví časovou osu X  
set timefmt "%m/%d/%y"      #format mesic/den/rok  
set xrange ["05/15/98 ":"05/16/98 "] #rozsah osy X  
set format x "%m/%d"        #popis osy X  
set timefmt "%m/%d/%y %H:%M" #mesic/den/rok hodina:minuta  
plot "data" using 1:3       #vykresleni pomoci „using“, osa  
                             X bude „05/15/98 20:00, osa Y  
                             bude „7.08e24“
```

3.14 Typy čar, barvy, styly, barva pozadí, barva v proměnné

V Gnuplotu jsou k dispozici předdefinované typy čar, které se mohou lišit jejich barvou, tloušťkou, mohou být tečkované či čárkované. K výběru typu čar může pomoci příkaz „test“, který nám zobrazí veškeré možnosti typu čar. Typ čar se může lišit podle toho, jaký terminál je zrovna použit, avšak každý terminál obsahuje speciální typ plné čáry „-1“ v černé barvě. Typ čáry je možné předdefinovat jak v inicializačním souboru tak i přímo v terminálu. Defaultně bude nastavení typu čáry spojeno se všemi funkcemi, které chceme vykreslit. Jestliže je zapotřebí jednotlivé funkce od sebe odlišit, je potřeba nastavit typ čáry pro každou funkci zvlášť.

Příklad:

```
plot sin (x) linestyle 3           #vybere typ čáry č. 3
plot sin (x) lt -1                 #vybere černou čáru
```

Příkaz „show colornames“ vypíše všechny předdefinované barvy, které jsou pak možno zadávat jak pomocí jejich názvu, tak pomocí složek RGB. Další možností je zadání hodnoty barvy, která odkazuje na aktuální paletu.

```
plot sin(x) lt rgb "gray"         #název barvy gray
plot sin(x) lt rgb "#FF00FF"     #barva hexadecimálně
plot sin(x) lt palette cb -45    #barva odkazuje na cbrange
                                  místo v paletě

plot sin(x) lt palette frac 0.3  #dílčí hodnota z celé palety
                                  z hodnot mezi 0 a 1
```

Pokud je požadován konkrétní styl barvy a konkrétní barvu, je potřeba první změnit typ čáry a následně změnit její barvu pomocí příkazu „linecolor - lc“

```
plot sin(x) lt 3 lc rgb "violet"
```

Barvu pozadí změníme příkazem „bgnd“, v našem případě bude pozadí šedé:

```
set term wxt background rgb "gray75"
```

Příkaz „linecolor variable“ říká programu, aby bral jeden sloupec ze souboru jako index k určení typu čáry spojené s barvou. Přes příkaz „using“ si pak vybere, který sloupec to bude.

Příklad: Vykreslí každou položku jinou barvou, kterou získá ze sloupce č. 3

```
plot 'data' using 1:2:(column(3)) with lines lc variable
```

Příkaz „lc rgbcolor variable“ vykreslí různou barvou jakýkoliv bod, úsečku či popis. Program dostane RGB informaci pro každý řádek zvlášť. Zde je potřeba speciální sloupec, který bude reprezentován jako 24 - bitové číslo.

V příkazu „using“ je pak potřeba určit hodnoty pro každý sloupec, viz příklad, který vykreslí body x, y, z odpovídající barvám „red“, „green“, „blue“.

```
rgb(r,g,b) = 65536 * int(r) + 256 * int(g) + int(b)
splot "data" using 1:2:3:(rgb($1,$2,$3)) with points lc rgb variable
```

3.15 Rozdíl mezi `linetypes` a `linestyles`

Jako u úseček, tak i u bodů je možné nastavovat vlastnosti, jako jsou například tloušťka apod. Jedná se o „`linestyles`“, které může uživatel definovat pomocí příkazu „`set style line`“. Nově definovaný styl lze pak použít kdekoliv je potřeba. Hlavní rozdíl je, že „`linestyles`“ jsou pouze dočasné a zanikají při ukončení programu, zatímco „`linetypes`“ jsou stálé, dokud je výslovně nepředefinujeme. Jako příklad je uveden styl č. 2, který bude mít zelenou barvu, tloušťku čáry 2 a typ bodu č. 5:

```
set style line 2 lt rgb "green" lw 2 pt 5
plot sin(x) with linespoints ls 2
```

3.16 Vykreslování grafu

Pro vykreslení grafu slouží v podstatě 3 příkazy. Příkaz „`plot`“ kreslí grafy ve 2D, příkaz „`splot`“ kreslí grafy ve 3D a příkaz „`replot`“ slouží k překreslení grafu např. při změně jeho parametrů apod. Příkaz „`plot`“ pracuje jak v obdélníkových souřadnicích tak i v polárních, pro jejich nastavení slouží příkaz „`set polar`“. Kreslení ve 3D se týká pouze souřadnic obdélníkových, avšak lze příkazem „`set mapping`“ zvolit ze tří dalších souřadnic – kartézské, sférické a válcové. Při použití příkazu „`using`“ lze pak u kreslení ve 2D a 3D definovat jakýkoliv souřadnicový systém.

Při zadání „`set mapping`“ se defaultně zvolí kartézský souřadnicový systém (cartesian). Pro sférický (spherical) platí, že soubor s daty obsahuje dva nebo tři sloupce. V prvních dvou jsou úhly „`theta`“ a „`phi`“ („zeměpisná délka“ a „zeměpisná šířka“), ve třetím sloupci se může nacházet `radius`. Pokud zde třetí sloupec není, jsou hodnoty `radius` brány jako číslo jedna.

```
x = r * cos(theta) * cos(phi)
y = r * sin(theta) * cos(phi)
z = r * sin(phi)
```

Válcové (cylindrical) můžou být definovány taktéž třemi sloupci, v prvním se nachází „`theta`“, ve druhém „`z`“ a ve třetím může být zadán `radius` jako je tomu v předchozím případě.

```
x = r * cos(theta)
y = r * sin(theta)
z = z
```

Obecný zápis příkazu `plot` vypadá následovně:

```
plot [rozsah X] [rozsah Y] f (x) [axes] typ souřadnic [title] „popis
funkce“ [with] styl vykreslení
```

rozsah X, rozsah Y	#nastaví interval osy x a osy y
axes	#přepínače x1y1(bottom, left), x1y2(bottom, right), x2y1(top, left), x2y2(top, right), určují, která z os se vztahuje k určitému grafu
title	#přidá legendu ke grafu funkce
with	#určí styl, jakým se daná funkce vykreslí

Pro představu vzorový příklad pro zobrazení paraboly s využitím všech přepínačů:

```
plot [-5:5][[0:100] x**2 axes x1y2 title „parabola“ with lines
```

Podrobnější popis přepínačů rozsahů, axes, title, with:

Nastavení rozsahů – implicitně je rozsah osy x v intervalu <-10; 10> a pro osu y automatický, Gnuplot dopočítává tak, aby byl obrázek viditelný celý. Ručně je možno nastavit rozsah do hranatých závorek „[min:max]“, pokud chce mít uživatel v grafu více křivek, příkazem „set [xyz]{2}range“ se nastaví zbývající osy podle potřeby. U tohoto nastavení jsou dále používané příkazy „reverse“, který změni orientaci osy, a příkaz „writeback“, který uchovává nastavení předchozí osy a přenesse jej do právě nastavované. Příkazem „set autoscale“ se zapne dopočítávání osy.

Přepínač TITLE – při vykreslení grafu je u něj automaticky generována legenda, která tento graf popisuje. Příkazem „set nokey“ se dá samozřejmě vypnout. Jestliže není uživatel spokojen s popisem grafu a požadujeme popisek jiný, použitím právě příkazu „title ,název“ lze tento popisek změnit.

Přepínač AXES – umožňuje mít v jednom obrázku více křivek, kde bude mít každá křivka jiné měřítko. Gnuplot zná v základu pouze dvě osy, a to osu x1 a y1, je nutné mu tedy nějakým způsobem říct, že chceme pracovat s více. Slouží k tomu příkaz „set [xyz]{2}[md]tics“. Parametry [md] umožňují převést čísla na názvy měsíců (m – month) nebo na názvy dnů (d – day). Může to být užitečné při zobrazování delších časových průběhů.

Příkazem „set xtics border“ nebo „set xtics axis“ se umístí popisek os buď ke kraji grafu, nebo na nulové osy. Jsou-li požadovány značky os i na protější ose, slouží k tomu příkaz „set xtics mirror“, pro změnu hustoty těchto značek slouží příkaz „set xtics min, krok, max“. Je-li požadován přesný výčet hodnot u jednotlivých značek, je možno je výčtem vypsát do kulatých závorek, takže příkaz je potom „set xtics (1, 2, 4, 8, 16, 32,...)“, pro textový výpis značek se před hodnotu, kterou je potřeba nahradit, jednoduše vypíše jeho znění, tedy „set xtics („málo“ 1, „“ 8, „hodně“ 32)“. Možnosti toho příkazu jsou velmi rozsáhlé, k získání více informací poslouží příkaz „help xtics“.

Přepínač WITH – nabízí mnoho stylů, jak jednotlivé grafy vykreslit. Implicitně je vykreslování nastaveno na lines (čáry), možnost změny nám dává příkaz „set f (x) with STYLE“, kde místo parametru style uvedeme jednu z následujících možností: lines, dots, steps, points, impulse, linespoints, labels, histeps, atd. Veškeré styly jsou pak uvedeny v nápovědě Gnuplotu, dostupné příkazem „help style“. Vzhled čáry (např. barvu) měníme parametrem „linetype x“ kde za x dosadíme číselnou hodnotu, která odpovídá danému typu. Seznam typů lze získat příkazem „test“. Šířka čáry je pak ovlivnitelná příkazem „linewidth x“, kde za x dosadíme číslo, které je násobkem původní velikosti.

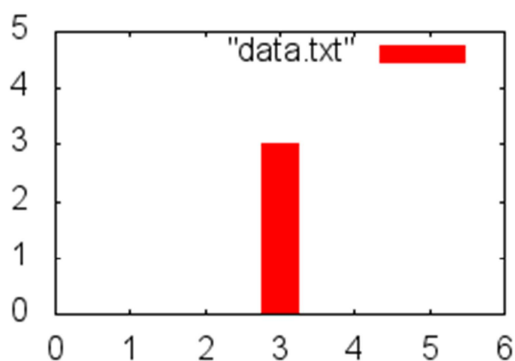
Je-li požadavkem provést kompletní nastavení jednotlivých os, ať už se jedná například o měřítko nebo jejich popisky, poslouží nám příkaz „set“. Většina názvů příkazů je odvezena od jejich činnosti, např.: popisek osy „x“ nastavíme příkazem „set xlabel“, logaritmické měřítko osy „x“ a „y“ nastavíme příkazem „set logscale xy“, atd.

3.17 Styly vykreslování

3.17.1 Boxes

Tento styl se používá pouze ve 2D vykreslování. Soubor může obsahovat dva nebo 3 sloupce, první dva pro základní x a y souřadnici, třetí pro doplňující informace jako je šířka vykreslovaného sloupce. Pokud není třetí sloupec zadán, šířka obdélníku se bude počítat automaticky. Dále je možno pomocí příkazu „set style fill“ kreslit vyplněný obdélník. Příkaz „set style fill empty“ obdélník nevyplní, „set style fill solid“ vykreslí obdélník vyplněný aktuální barvou, „set style fill pattern“ vykreslí obdélník aktuální barvou a aktuálním vzorem.

Příklad: Vykreslí k souřadnici [3:3] vyplněný obdélník o šířce 0.5



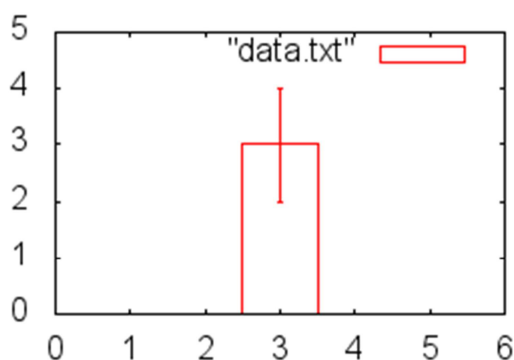
```
Datový soubor „data.txt“: 3 3 0.5
set style fill solid 1
plot „data.txt“ with boxes
```

Obrázek 1 - Vykreslovací styl Boxes

3.17.2 Boxerrorbars

Podobný styl jako Boxes, také se používá jen ve 2D vykreslování. Tento styl je složen ze stylu „Boxes“ a stylu „yError-bars“. Datový soubor může obsahovat tři, čtyři nebo pět sloupců. První dva jsou základní souřadnice x a y. Pokud obsahuje soubor tři sloupce, pak je třetí sloupec jako „ydelta“ (velikost chybového ukazatele). Pokud jsou zadány čtyři sloupce, pak je třetí jako „ydelta“ a čtvrtý jako „xdelta“ nebo je třetí jako minimum souřadnice y pro chybový ukazatel (ylow) a čtvrtý jako maximum souřadnice y pro chybový ukazatel (yhigh). Pokud je zadáno 5 sloupců, pak je to v následujícím pořadí: „x y ylow yhigh xdelta“, kde poslední sloupec nastaví šířku obdélníku (boxu).

Příklad: Vykreslí k souřadnici [3:3] obdélník o šířce 1 a chybový ukazatel od 2 do 4



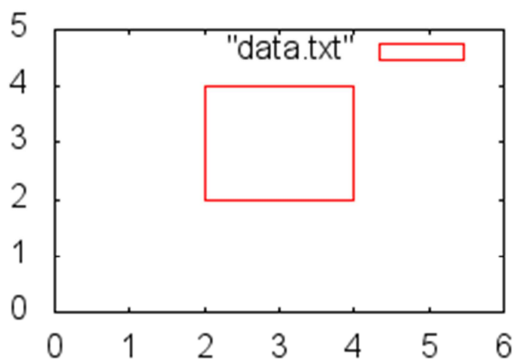
Datový soubor „data.txt“: 3 3 2 4 1
plot „data.txt“ with boxerrorbars

Obrázek 2 - Vykreslovací styl Boxerrorbars

3.17.3 Boxyerrorbars

Podobný styl jako Boxerrorbars, zde se však vykresluje obdélník. Datový soubor musí obsahovat čtyři nebo šest sloupců. První dva sloupce jsou základní souřadnice x a y. Pokud jsou v souboru čtyři sloupce, tak třetí reprezentuje „xdelta“ (šířka obdélníku od základních souřadnic) a čtvrtý reprezentuje „ydelta“ (výška obdélníku od základních souřadnic). Pokud máme šest sloupců, pak je to v následujícím pořadí: „x y xlow xhigh ylow yhigh“.

Příklad: Vykreslí k souřadnici [3:3] obdélník o šířce 2 a výšce 2



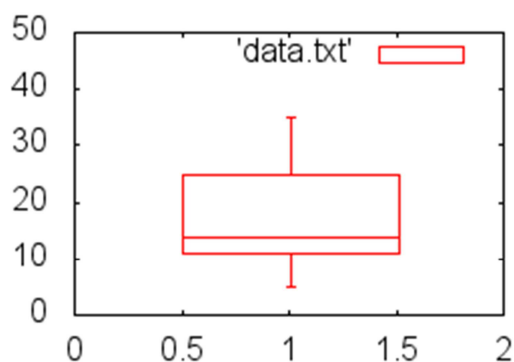
Datový soubor „data.txt“: 3 3 2 4 2 4
plot „data.txt“ with boxxyerrorbars

Obrázek 3 - Vykreslovací styl Boxyerrorbars

3.17.4 Boxplot

Tento styl vykreslování se opět používá jen ve 2D vykreslování. Slouží k reprezentaci statistických hodnot.

Příklad: Vykreslí obdélník se svislým ukazatelem. Rozsah hodnot na souřadnici y je od 5 do 35 a v zobrazeném obdélníku je vidět rozdělení hodnot podle mediánů. Datový soubor obsahuje dva sloupce, v prvním je hodnota souřadnice x rovna jedné, která je konstantním a ve druhém sloupci jsou hodnoty souřadnic y v následující podobě: „5 10 11 12 13 14 15 20 25 30 35“. Medián rozdělí hodnoty mezi dvě stejně početné skupiny. Zde je medián roven 14, který dělí obdélník v grafu na dvě části. Z těchto rozdělených skupin, tedy „5 10 11 12 13“ a „15 20 25 30 35“ jsou pak mediány 11 a 25, což je pak znázorněno jako vrchní a spodní hranice obdélníku. Šířka obdélníku je 1.

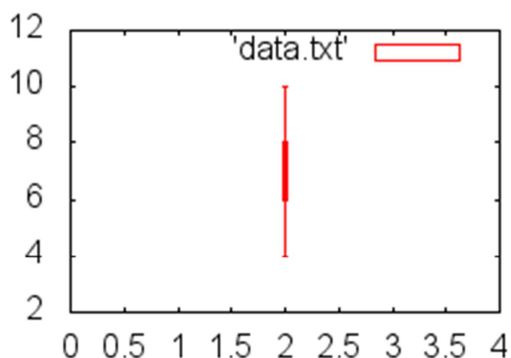


```
set boxwidth 1
plot „data.txt“ with boxplot
```

Obrázek 4 - Vykreslovací styl Boxplot

3.17.5 CandleSticks

Styl vykreslování určený pro 2D, může být využit pro vizualizaci financí nebo pro generování statistických dat zobrazených pomocí vertikálního ukazatele a obdélníku. Datový soubor je ve formátu „x box_min whisker_min whisker_high box_high“, kde hodnota „whisker_min“ a „whisker_high“ je velikost vertikálního ukazatele a hodnoty „box_min“ a „box_high“ jsou hranice obdélníku. Pokud je hodnota „box_min“ menší jako „box_high“, tak je obdélník prázdný, v opačném případě je vybarven. Pomocí příkazu „set boxwidth“ se nastaví šířka obdélníku. Koncové horizontální ukazatele se nastaví příkazem „whiskerbars“.



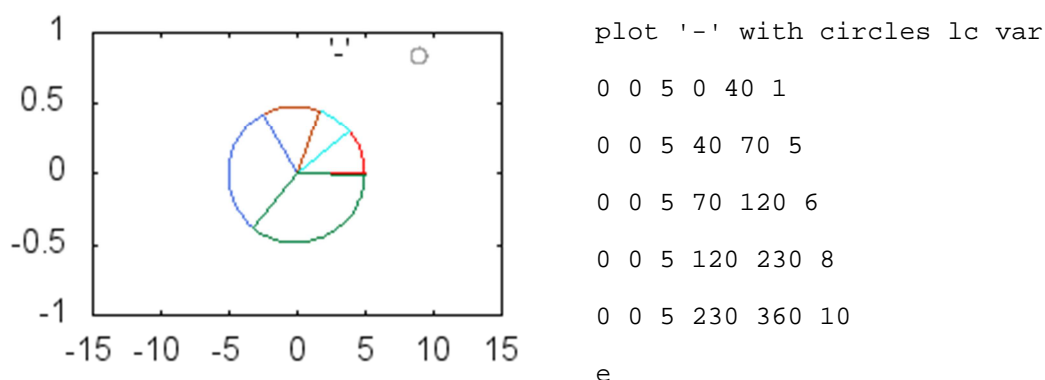
```
Datový soubor „data.txt“: 2 8 4 10 6
plot „data.txt“ with candlesticks
whiskerbars 0.5
```

Obrázek 5 - Vykreslovací styl CandleSticks

3.17.6 Circles

Styl vykreslování spočívá ve vykreslení kruhu kolem každého bodu. Datový soubor může mít podobu dvou nebo třech sloupců. Pokud jsou zadány tři sloupce, tak první dva reprezentují základní souřadnice x a y , třetí pak reprezentuje rádius kruhu, který je vyjádřen v jednotkách odpovídajících vodorovné ose x . Pokud jsou zadány pouze dva sloupce, pak je rádius určen nastaveným stylem příkazem „set style circle“. V tomto případě je pak rádius počítán v souřadnicích obrazovky nebo grafu. Přidáním dalších dvou sloupců do datového souboru lze nastavit vykreslení oblouku, kde čtvrtý sloupec představuje počáteční úhel a pátý sloupec představuje koncový úhel. Přidáním posledního (tedy 4. nebo 6. sloupce) lze nastavit barvu pro každý segment kruhu.

Příklad: Vykreslí různobarevné segmenty podle zadaných „linecolor variables“

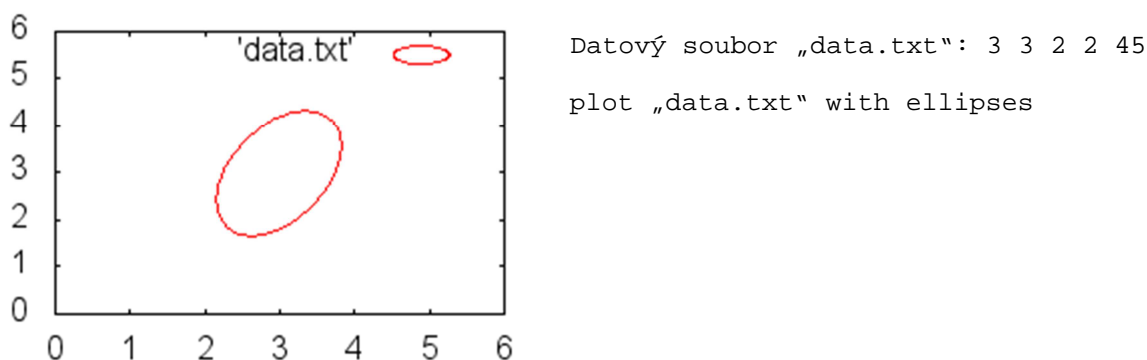


Obrázek 6 - Vykreslovací styl Circles

3.17.7 Ellipses

Styl vykreslování spočívá ve vykreslení elips kolem každého bodu. Používá se pouze u 2D vykreslování. Datový soubor může obsahovat dva, tři, čtyři nebo pět sloupců. První dva sloupce reprezentují základní souřadnice x a y , třetí sloupec určuje velikost hlavní poloosy, čtvrtý sloupec určuje velikost vedlejší poloosy, pátý určuje úhel, který svírá hlavní poloosa s vodorovnou osou x .

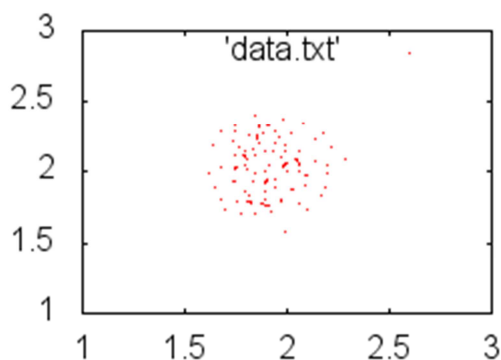
Příklad: Vykreslí z bodu [3:3] elipsu, kde hlavní a vedlejší poloosa bude rovna 2 a celá elipsa bude potočena o úhel 45° oproti ose x .



Obrázek 7 - Vykreslovací styl Ellipses

3.17.8 Dots

Styl vykreslování spočívá ve vykreslení bodu (pixelu) na každé zadané souřadnici. Používá se jak u 2D vykreslování tak i u 3D vykreslování. Datový soubor může obsahovat jeden, dva nebo tři sloupce. Pokud je zadán pouze jeden, bere se jako základní souřadnice y a souřadnice x je automaticky brána jako odpovídající řádek. Dva zadané sloupce slouží jako základní souřadnice x a y pro vykreslení 2D grafu pomocí „plot“ funkce. Tři sloupce jsou pak souřadnice x, y a z pro vykreslení 3D grafu pomocí funkce „splot“.



Obrázek 8 - Vykreslovací styl Dots

3.17.9 Filledcurves

Styl Filledcurves se používá pouze u 2D vykreslování. Datový soubor musí obsahovat funkci nebo dva sloupce. Potom může být vykreslování doplněno o další nastavení.

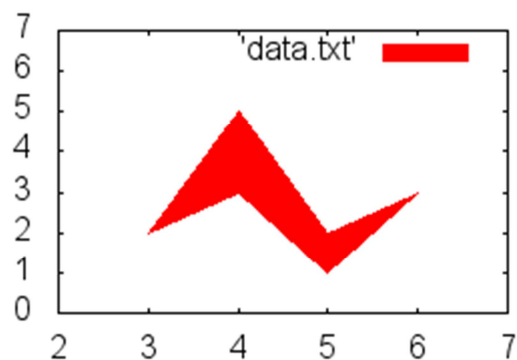
```
„plot ‚data.txt‘ with filledcurves [Možnosti]
```

```
Možnosti: [closed | {above | below}]
```

```
{x1 | x2 | y1 | y2 | r}[=<a>] | xy=<x>,<y>]
```

Pokud obsahuje datový soubor tři sloupce, potom je první sloupec brán jako základní souřadnice x a zbylé dva sloupce představují souřadnici y1 a y2, mezi kterými je vyplněn prostor.

Příklad: Vyplní prostor mezi zadanými souřadnicemi.



Datový soubor „data.txt“:

```
3 2 2
```

```
4 3 5
```

```
5 1 2
```

```
6 3 3
```

```
plot „data.txt“ with
```

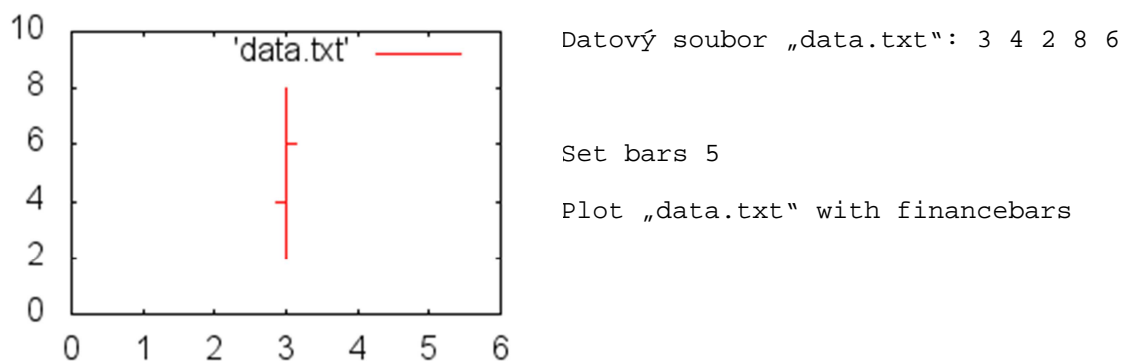
```
filledcurves closed
```

Obrázek 9 - Vykreslovací styl Filledcurves

3.17.10 FinanceBars

Styl FinanceBars se používá pouze u 2D vykreslování pro vizualizaci finančních dat. Datový soubor se musí skládat z pěti sloupců v této podobě: „date open low high close“. První sloupec reprezentuje základní souřadnici x, nejčastěji vyjádřenou jako datum. Položky „open“ a „close“ reprezentují počáteční a koncovou částku. Položky „low“ a „high“ jsou pak vertikální hranice ukazatele. Dodatečný sloupec může sloužit ke změně barev. Velikost ukazatelů „open“ a „close“ lze změnit příkazem „set bars“.

Příklad: Na souřadnici x = 3 zobrazí počáteční (4) a koncovou (6) částku na vertikálním ukazateli od 2 do 8. Velikost ukazatelů je 5.



Obrázek 10- Vykreslovací styl Financebars

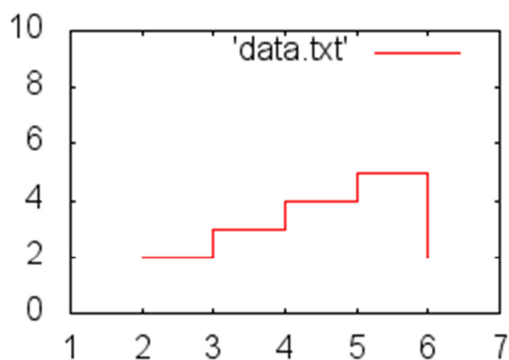
3.17.11 Steps, Fsteps

Další styl určený pro 2D vykreslování. Datový soubor požaduje pouze základní souřadnice x a y. Vykreslovat je možno pomocí funkce „steps“, u které se první vykresluje podle osy x, potom podle osy y. U funkce „fsteps“ je tomu přesně naopak. K vyplnění vykreslených obdélníků slouží vykreslovací styl Fillsteps.

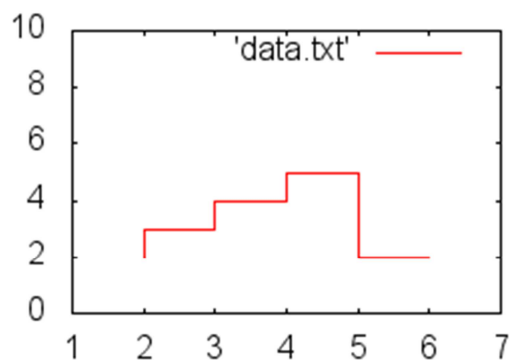
Příklad: Vykreslí podle zadaného datového souboru graf prvně pomocí funkce „steps“, potom pomocí funkce „fsteps“.

Datový soubor „data.txt“:

```
2 2
3 3
4 4
5 5
6 2
```



Obrázek 11 - Vykreslovací styl Steps

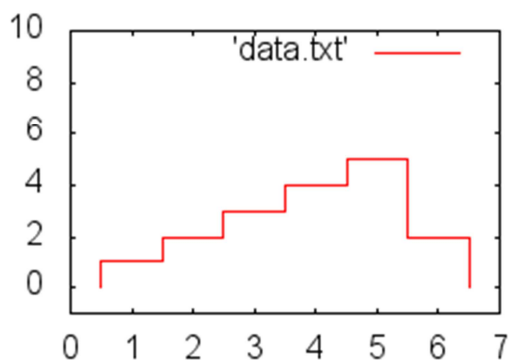


Obrázek 12 - Vykreslovací styl Fsteps

3.17.12 Histeps

Styl histeps se používá pouze u 2D vykreslování. Je to základní styl, který se používá k vykreslování histogramů (styl „histograms“). Datový soubor požaduje pouze základní souřadnice X a Y. Způsob vykreslování spočívá v tom, že bod na souřadnici „x1“ je reprezentován horizontální čarou. Počáteční bod této čáry je počítán jako „ $((x_0+x_1)/2, y_1)$ “, koncový bod jako „ $((x_1+x_2)/2, y_1)$ “. K odpovídající souřadnici x je vykreslena vertikální čára počítaná od „ $((x_1+x_2)/2, y_1)$ “ do „ $((x_1+x_2)/2, y_2)$ “.

Příklad: Vykreslí graf stylem: První souřadnice x se rovná „ $0+1/2$ “, vykreslena bude vertikální čára od „0“ do „1“, druhá souřadnice x bude „ $1+2/2$ “ a k ní odpovídající vertikální čára od „1“ do „2“, atd.



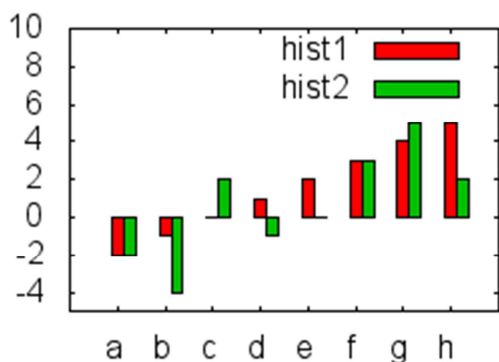
Datový soubor „data.txt“:

```
1 1
2 2
3 3
4 4
5 5
6 2
```

Obrázek 13 - Vykreslovací styl Histeps

3.17.13 Histograms

Styl histograms slouží pouze pro 2D vykreslování. Histogram je složen z paralelních sloupců, vykreslovaných buďto svisle nebo vodorovně podle nastaveného typu. Datový soubor musí obsahovat pouze jeden sloupec, který představuje výšku (respektive šířku) daného sloupce. Základní styl vykreslování je nastaven jako „set style histogram clustered gap 2“, vykreslování začíná na $x = 1$, k této souřadnici odpovídá první záznam v datovém souboru, další záznam se vykreslí na souřadnici $x = 2$ atd. Parametr „clustered“ vyjadřuje seskupený graf, parametr „gap 2“ znamená, že mezi jednotlivými vykreslenými sloupci bude mezera o velikost dvojnásobku šířky obdélníku.



Datový soubor „data.txt“:

```
-2 -2 „a“
-1 -4 „b“
0 2 „c“
1 -1 „d“
2 0 „e“
3 3 „f“
4 5 „g“
5 2 „h“
```

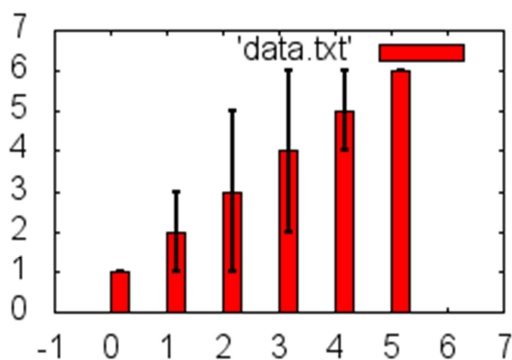
Obrázek 14 - Vykreslovací styl Histograms (1)

```
set style histogram cluster
set style fill solid 1.0 border lt -1
plot "data.txt" using 1:xtic(3) title "hist1", "data.txt" using
2:xtic(3) title "hist2"
```

Dalšími typy histogramů jsou:

```
set style histogram errorbars {gap <gapsize>} {<linewidth>}
set style histogram rowstacked
set style histogram columnstacked
```

Styl „errorbars“ vyžaduje navíc jeden nebo dva sloupce, které vyjadřují buďto velikost nebo minimální a maximální souřadnici y u chybového ukazatele.



Datový soubor „data.txt“:

```
1 0
2 1
3 2
4 2
5 1,5
6 0
```

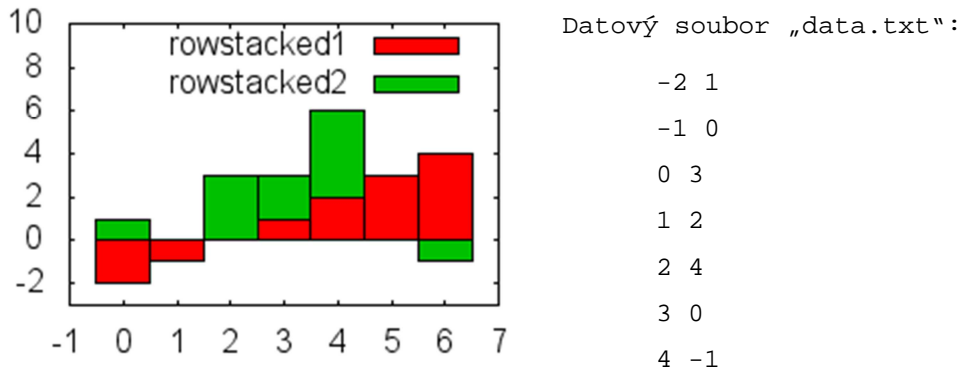
Obrázek 15 - Vykreslovací styl Histograms (2)

```

set style fill solid 1 border lt -1
set style histogram errorbars gap 2 lw 2
plot "data.txt" with histogram

```

Styly „rowstacked“ a „columnstacked“ jsou založeny na skládání jednotlivých obdélníků do tzv. zásobníku.

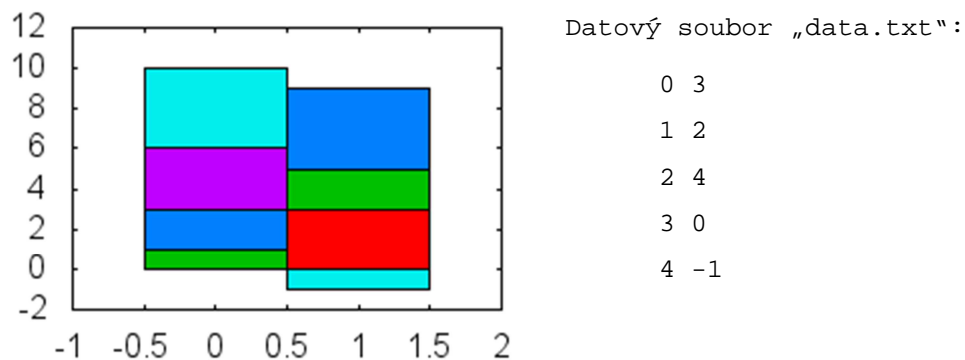


Obrázek 16 - Vykreslovací styl Histograms (3)

```

set style fill solid 1 border lt -1
set style histogram rowstacked
plot "data.txt" using 1 title "rowstacked1" with histogram, '' using
2 title "rowstacked2" with histogram

```



Obrázek 17 - Vykreslovací styl Histograms(4)

```

set style fill solid 1 border lt -1
set style histogram columnstacked
plot "data.txt" using 1 with histogram, '' using 2 with histogram

```

3.17.14 Newhistogram

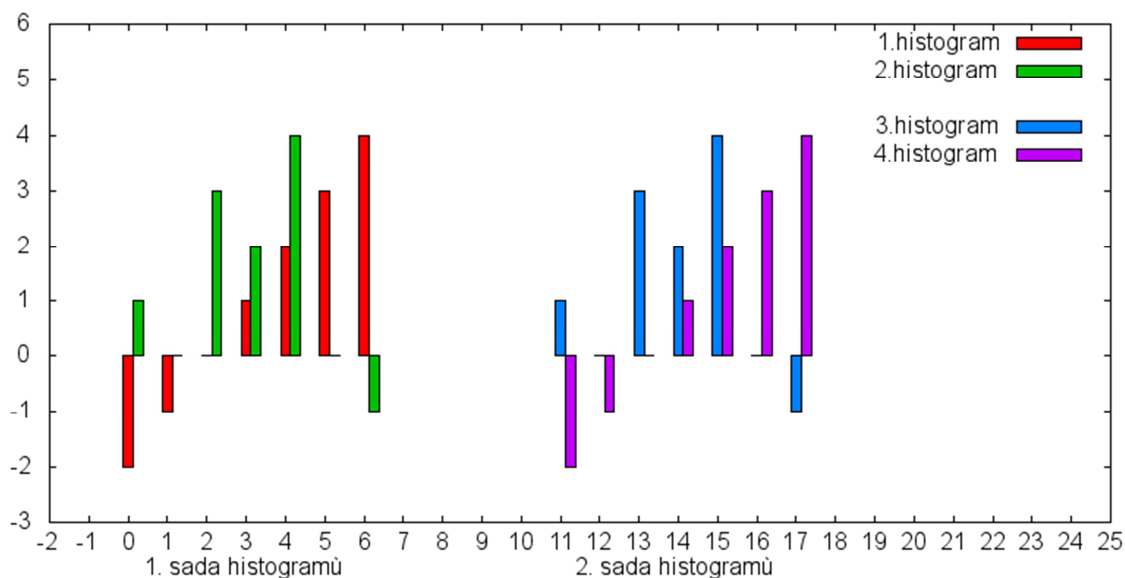
Tento styl slouží k zobrazení více histogramů v jednom grafu, které mohou mít naprosto různé vlastnosti. Základní syntaxe vypadá následovně:

```
newhistogram {"<title>"} {lt <linetype>} {fs <fillstyle>}  
{at <x-coord>}
```

Každému histogramu lze nastavit pomocí názvu pomocí příkazu „title“, „lt“ nastaví typ čáry, „fs“ nastaví, jestli bude obdélník vyplněn či nikoliv, „at“ pak určuje, od jaké souřadnice X se začne vykreslovat následující histogram.

Příklad: Vykreslí dvě sady histogramů, každé sadě je nastaven popisek a v každé sadě je potom popsán každý histogram jednotlivě. Druhá sada histogramů se vykreslí od souřadnice X = 11. Datový soubor „data.txt“: -2 1 \ -1 0 \ 0 3 \ 1 2 \ 2 4 \ 3 0 \ 4 -1

```
set style histogram cluster  
set style data histogram  
set style fill solid 1.0 border -1  
plot newhistogram "1. sada histogramů", "data.txt" using 1 title "1.  
histogram", "" using 2 title "2. histogram", newhistogram "2. sada  
histogramů" at 11, "data.txt" using 2 title "3. histogram", "" using  
1 title "4. histogram"
```



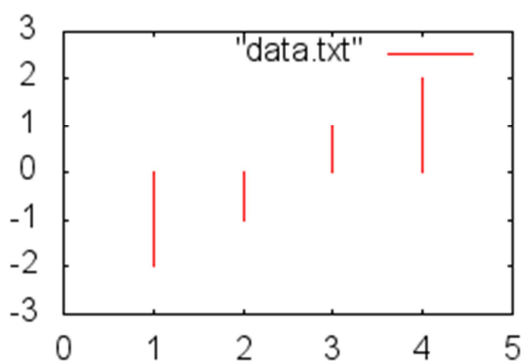
Obrázek 18 - Vykreslovací styl Newhistogram

3.17.15 Impulses, Lines, Linespoints, Points

Styly pro vykreslování ve 2D i 3D. Datový soubor může obsahovat jeden, dva nebo tři sloupce. Jeden sloupec reprezentuje souřadnici y, pokud máme zadány dva, tak se jedná o souřadnici x a y, pokud máme tři sloupce, pak se jedná o souřadnice x, y a z. Dodatečné sloupce mohou sloužit k určení barev pro každý impuls.

Impulses

Styl, který kreslí vertikální čáry od souřadnice $y = 0$ (u 2D kreslení) nebo od $z = 0$ (u 3D kreslení).



Datový soubor „data.txt“:

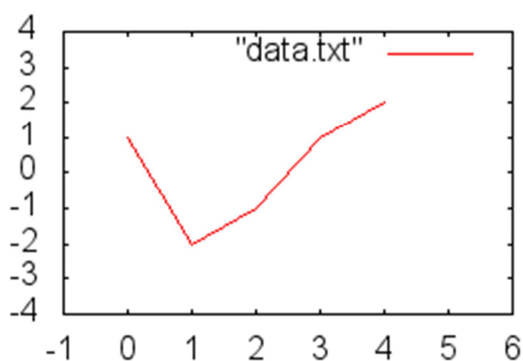
```
-2  
-1  
1  
2
```

plot „data.txt“ with impulses

Obrázek 19 - Vykreslovací styl Impulses

Lines

Styl spojuje jednotlivé body rovnou čárou.



Datový soubor „data.txt“:

```
1 \ -2 \ -1 \ 1 \ 2
```

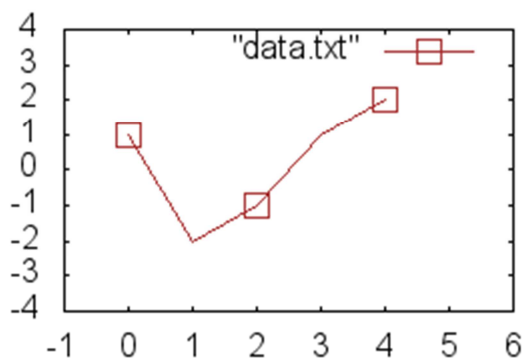
plot „data.txt“ with lines

Obrázek 20 - Vykreslovací styl Lines

Linespoints

Styl spojuje jednotlivé body rovnou čarou a od stylu Lines se liší tím, že u každého bodu vykresluje symbol. Velikost toho symbolu je možno změnit příkazem „set pointsize“. Pomocí příkazu „pointinterval“ můžeme určit četnost vykreslování symbolů.

Příklad: Vykreslí graf, který bude mít nastaven typ čary číslo 17, bude se vykreslovat každý druhý symbol a velikost symbolu bude 2.



Datový soubor „data.txt“:

```
1 \ -2 \ -1 \ 1 \ 2
```

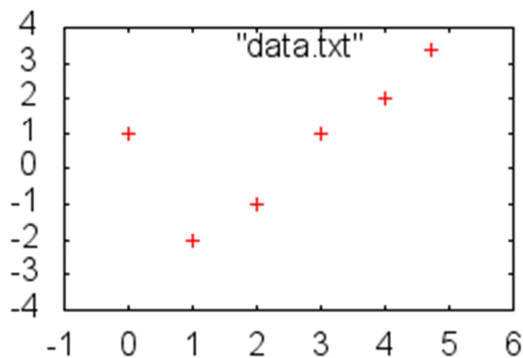
```
set pointsize 2
```

```
plot „data.txt“ with linespoints ls  
17 pointinterval 2
```

Obrázek 21 - Vykreslovací styl Linespoints

Points

Styl vykreslí symbol kolem každého bodu. Velikost symbolu lze určit pomocí příkazu „set pointsize“.



Datový soubor „data.txt“:

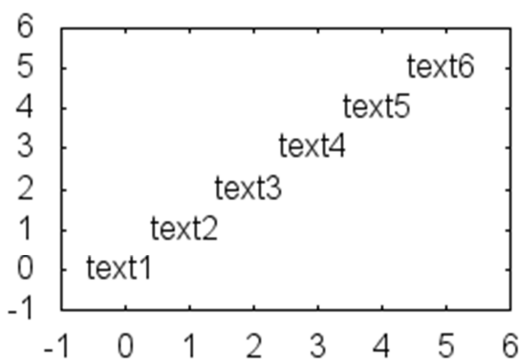
```
1 \ -2 \ -1 \ 1 \ 2
```

```
plot „data.txt“ with points
```

Obrázek 22 - Vykreslovací styl Points

3.17.16 Labels

Styl vykresluje ve 2D i 3D. Datový soubor požaduje tři nebo čtyři sloupce. Pokud máme zadány tři sloupce, pak první dva představují základní souřadnice x a y , třetí sloupec je určen pro text, který se bude na konkrétních souřadnicích vykreslovat. U vykreslování ve 3D je potřeba zadat ještě souřadnici z . Dodatečné sloupce mohou určovat velikost nebo barvu textu.



Datový soubor „data.txt“:

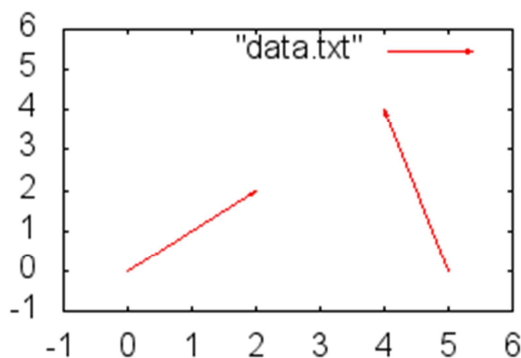
```
0 0 "text1"
1 1 "text2"
2 2 "text3"
3 3 "text4"
4 4 "text5"
5 5 "text6"
```

plot „data.txt“ with labels title „“

Obrázek 23 - Vykreslovací styl Labels

3.17.17 Vectors

Vykreslovací styl pro 2D i 3D, který kreslí vektory od souřadnice $[x, y]$ do souřadnice $[x+xdelta, y+ydelta]$. Datový soubor požaduje čtyři nebo šest sloupců, podle toho, jestli kreslíme ve 2D nebo 3D. Pokud máme zadány čtyři sloupce, pak první dva představují základní souřadnici x, y a další dva sloupce představují souřadnici $x+xdelta, y+ydelta$. Obdobně u 3D s přidáním souřadnice z . Dodatečné sloupce mohou sloužit pro změnu barvy každého vektoru. Každý vektor je zakončen šipkou, která může být pomocí příkazu „head filled“ vyplněna. Styl šipky můžeme měnit příkazem „arrowstyle <číslo>“.



Datový soubor „data.txt“:

```
0 0 2 2
5 0 -1 4
```

plot „data.txt“ with vectors filled

Obrázek 24 - Vykreslovací styl Vectors

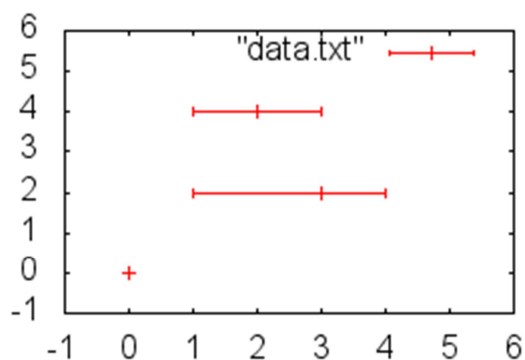
3.17.18 Xerrorbars, Yerrorbars

Styly pouze pro vykreslování ve 2D. Jsou založeny na vykreslování bodu a horizontálního (vertikálního) chybového ukazatele. Datové soubory požadují tři nebo čtyři sloupce. První dva představují základní souřadnice x, y, pokud máme tři sloupce, pak poslední představuje velikost chybového ukazatele centricky od bodu (x_{delta}), pokud máme čtyři sloupce, pak si můžeme určit minimum a maximum chybového ukazatele (x_{low} , x_{high} resp. y_{low} , y_{high}). Dodatečné sloupce mohou sloužit pro změnu barvy.

Datový soubor „data.txt“:

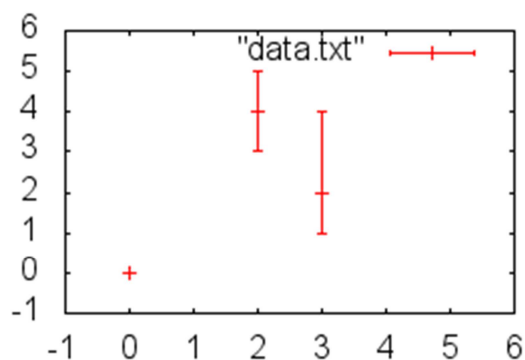
```
0 0 0
2 4 1
2 1 4
```

plot „data.txt“ with xerrorbars



Obrázek 26 - Vykreslovací styl Xerrorbars

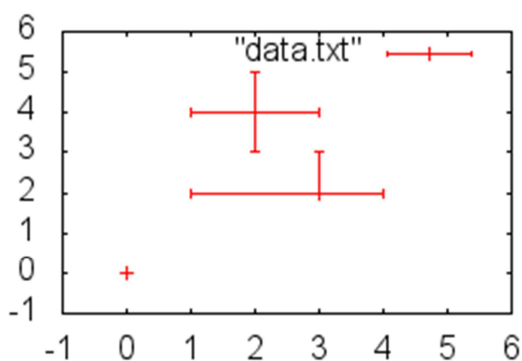
plot „data.txt“ with yerrorbars



Obrázek 25 - Vykreslovací styl Yerrorbars

3.17.19 Xyerrorbars

Styl, který je kombinací stylů Xerrorbars a Yerrorbars, vykresluje vertikální i horizontální chybové ukazatele. Jediná změna je v datovém souboru, kde jsou požadovány základní souřadnice x, y. Pokud zadáme čtyři sloupce, pak třetí a čtvrtý představuje x_{delta} a y_{delta} , pokud zadáme šest sloupců, pak si určíme minimum a maximum jednotlivých ukazatelů (x_{low} , x_{high} , y_{low} , y_{high}). Dodatečné sloupce mohou sloužit pro změnu barvy.



Obrázek 27 - Vykreslovací styl Xyerrorbars

Datový soubor „data.txt“:

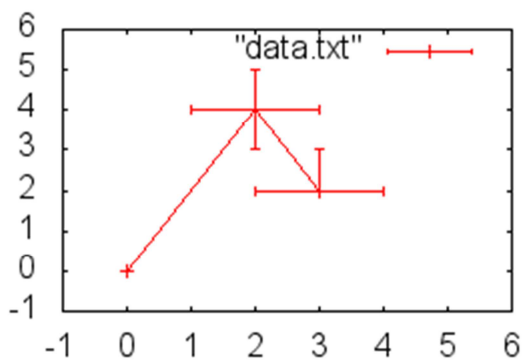
```
0 0 0 0
2 4 1 1
3 2 1 4 2 3
```

plot „data.txt“ with xyerrorbars

3.17.20 Xerrorlines, Yerrorlines, Xyerrorlines

Styly jsou obdobné jako předchozí Xerrorbars, Yerrorbars, Xyerrorbars s tím rozdílem, že jsou jednotlivé body spojeny rovnou čarou. Datové soubory jsou taktéž naprosto totožné.

Příklad: Pro ukázkou vykreslovací styl Xyerrorlines.



Datový soubor „data.txt“:

```
0 0 0 0
```

```
2 4 1 1
```

```
3 2 1 4 2 3
```

plot „data.txt“ with xyerrorlines

Obrázek 28 - Vykreslovací styl Xyerrorlines

4 Praktická část:

Cílem této kapitoly je ukázat, jaké jsou hlavní rozdíly v grafických výstupech, které jsou generovány jak Matlabem, tak Gnuplotem. Z praktických zkušeností, které jsme mohli získat z projektu, který byl zaměřen na problematiku tepelného stárnutí vedení vysokého napětí, jsme narazili na pár věcí, se kterými si samozřejmě Matlab dokázal poradit, ale výsledek nebyl tak čitelný a tak čistý při zmenšených rozměrech, jako tomu bylo u Gnuplotu. Co se týkalo jednodušších grafických výstupů, zde tyto rozdíly nebyly až tak patrné, jako tomu bylo u složitějších případů.

4.1 Porovnání grafických výstupů včetně zdrojových kódů

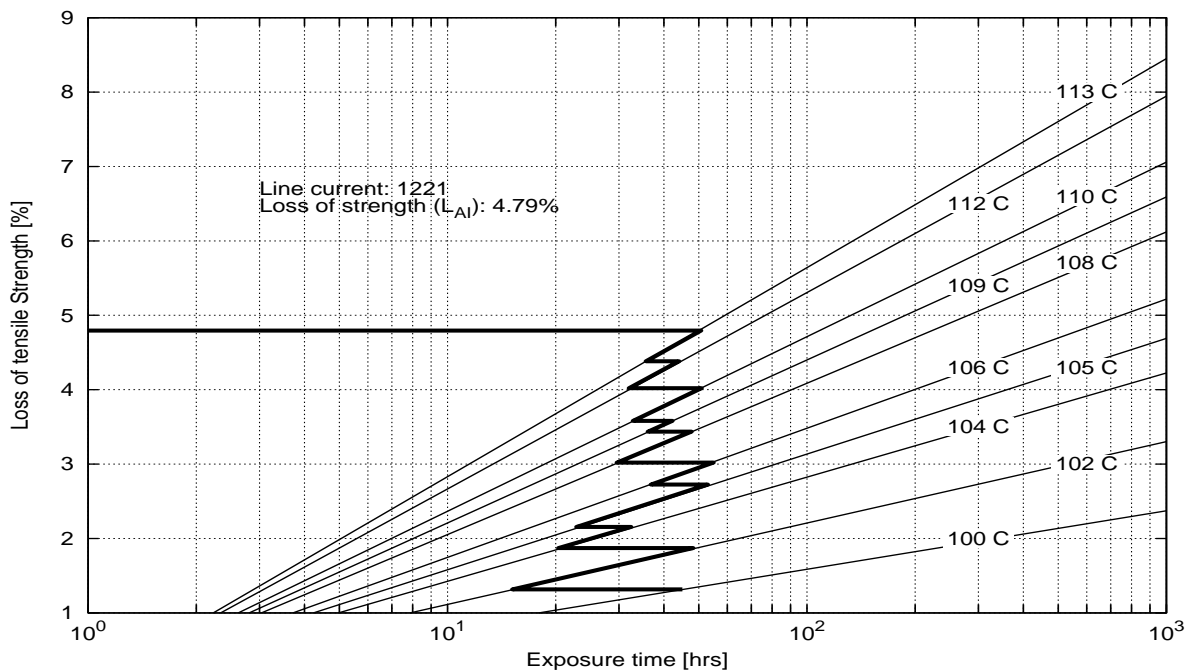
Problém, se kterým jsme se potýkali a po té se rozhodli využít program Gnuplot, se naskytl u tvorby grafických výstupů tepelných závislostí. V tomto případě si byl Gnuplot schopný poradit s čitelností a viditelností lépe, než tomu bylo u Matlabu. Matlab generoval grafické výstupy s písmem o velikosti 16b a při zmenšených verzích celého grafu to bylo hůře čitelné. Pro porovnání jsme dále využili grafy histogramů.

4.1.1 Teplotní závislosti

Matlab

Zdrojový kód:

```
>> cd /... /data.txt
>> load data.txt
>> sourX1 = data(:,1);
>> sourY1= data(:,2);
>> plot(sourX1,sourY1,'black-','LineWidth',3);
>> xlabel('Exposure Time[hrs]')
>> ylabel('Loss of tensile strength [%]')
>> set(gca,'xscale','log')
>> xlim([1 1000])
>> ylim([1 9])
>> grid on
>> hold on
>> >> cd /... /data1.txt
>> load data1.txt
>> sourX2 = data1(:,1);
>> sourY2 = data1(:,2);
>> plot(sourX2,sourY2,'black-');
```



Obrázek 29 - Grafická metoda určení ztráty napětí vodiče v tahu (Matlab)

Gnuplot

Zdrojový kód:

```

set terminal postscript eps size 8,4 enhanced color font
'Helvetica,20'
set output "prvni1.eps"
set logscale x
set xrange[1:1000]
set yrange [1:9]

set xtics ("10^{0}" 1, "" 2, "" 3, "" 4, "" 5, "" 6, "" 7, "" 8, "" 9, \
"10^{1}" 10, "" 20, "" 30, "" 40, "" 50, "" 60, "" 70, "" 80, "" 90, "10^{2}"
100, \
"" 200, "" 300, "" 400, "" 500, "" 600, "" 700, "" 800, "" 900, "10^{3}"
1000)

set grid xtics ytics
set xlabel "Exposure time [hrs]"
set ylabel "Loss of tensile Strength [%]"

set style rect fillcolor lt -3 fillstyle solid 1.0 noborder

lb1 = "113 C"
set label 1 lb1 at 600,8 center front
set obj 1 rect center 600,8 size char strlen(lb1), char 1
set obj 1 front

lb2 = "110 C"
set label 2 lb2 at 600,6.6 center front
set obj 2 rect center 600,6.6 size char strlen(lb2), char 1
set obj 2 front

```

```

lb3 = "108 C"
set label 3 lb3 at 600,5.7 center front
set obj 3 rect center 600,5.7 size char strlen(lb3), char 1
set obj 3 front

lb4 = "105 C"
set label 4 lb4 at 600,4.3 center front
set obj 4 rect center 600,4.3 size char strlen(lb4), char 1
set obj 4 front

lb5 = "102 C"
set label 5 lb5 at 600,3 center front
set obj 5 rect center 600,3 size char strlen(lb5), char 1
set obj 5 front

lb6 = "112 C"
set label 6 lb6 at 300,6.5 center front
set obj 6 rect center 300,6.5 size char strlen(lb6), char 1
set obj 6 front

lb7 = "109 C"
set label 7 lb7 at 300,5.4 center front
set obj 7 rect center 300,5.4 size char strlen(lb7), char 1
set obj 7 front

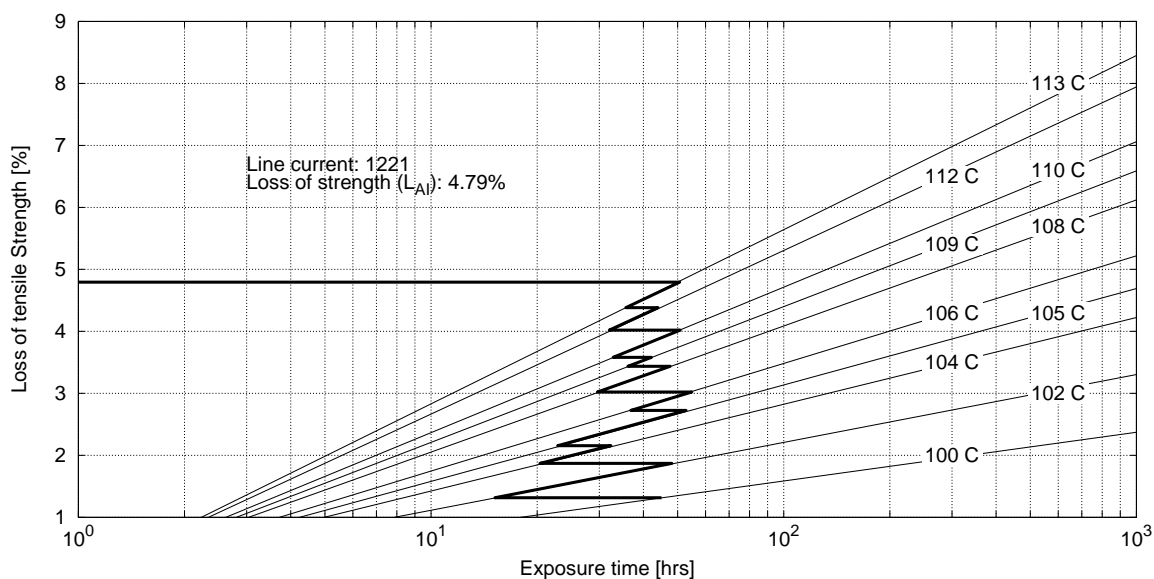
lb8 = "106 C"
set label 8 lb8 at 300,4.3 center front
set obj 8 rect center 300,4.3 size char strlen(lb8), char 1
set obj 8 front

lb9 = "104 C"
set label 9 lb9 at 300,3.5 center front
set obj 9 rect center 300,3.5 size char strlen(lb9), char 1
set obj 9 front

lb10 = "100 C"
set label 10 lb10 at 300,2 center front
set obj 10 rect center 300,2 size char strlen(lb10), char 1
set obj 10 front

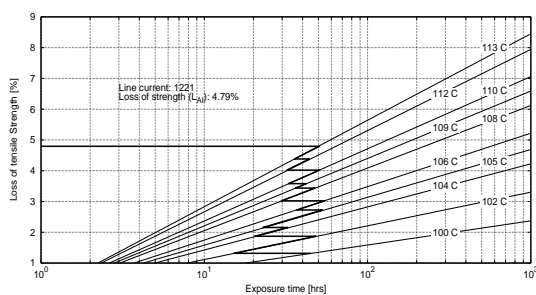
set label "Line current: 1221\nLoss of strength (L_{AI}): 4.79%" at
3,6.7
plot "data.txt" with lines lt -1 lw 3 title "", "data1.txt" with
lines lt -1 title ""

```

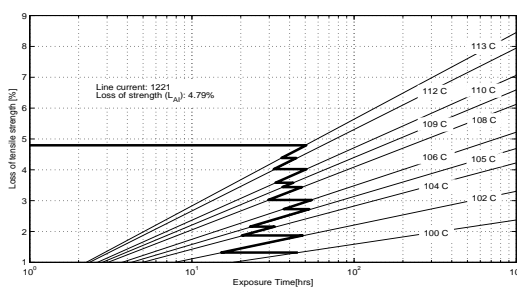


Obrázek 30 - Grafická metoda určení ztráty napětí vodiče v tahu (Gnuplot)

Ukázka zmenšených verzí



Obrázek 31 - Zmenšená verze – Gnuplot



Obrázek 32- Zmenšená verze - Matlab

4.1.2 Histogramy

Matlab

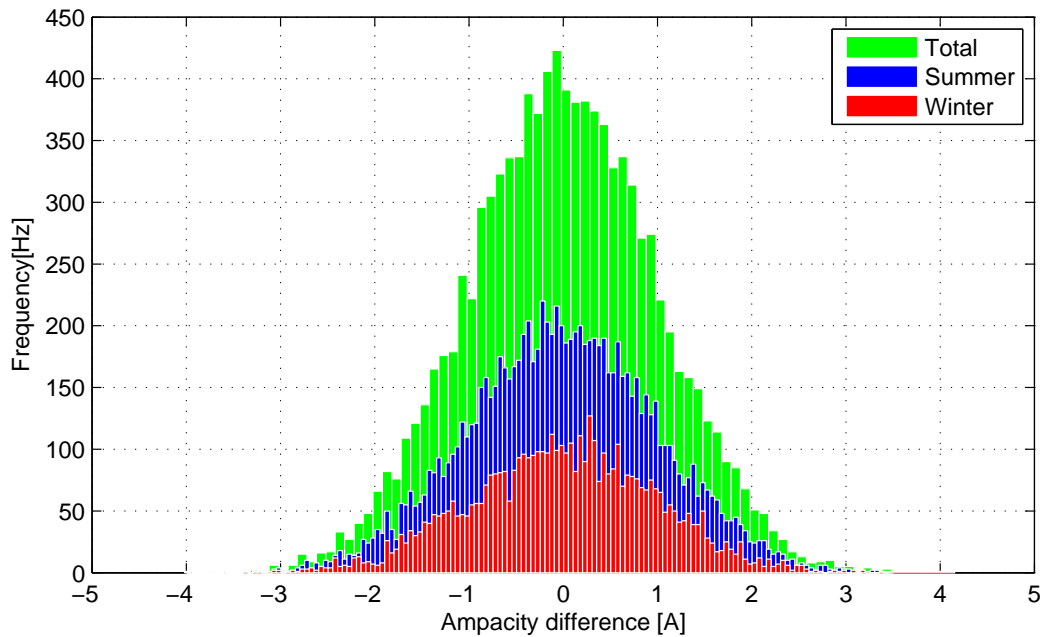
Zdrojový kód:

```
>> yn = randn(10000,1);
>> x = min(yn):.1:max(yn);
>> hist(yn,x)
>> hold on
>> yn1 = randn(5000,1);
>> x = min(yn):.05:max(yn);
>> hist(yn,x)
>> yn1 = randn(5000,1);
>> hold on
```

```

>> hist(yn1,x)
>> grid on
>> xlabel('Ampacity difference [A]')
>> ylabel('Frequency[Hz]')
>> legend('Total', 'Summer', 'Winter')
>> h = findobj(gca, 'Type','patch');
>> set(h(1), 'FaceColor','r', 'EdgeColor','w')
>> set(h(2), 'FaceColor','b', 'EdgeColor','w')
>> set(h(3), 'FaceColor','g', 'EdgeColor','w')

```



Obrázek 33 - Frekvenční histogram proudové zatížitelnosti vodiče (Matlab)

Gnuplot

Zdrojový kód:

```

set terminal postscript eps size 8,4 enhanced color font
'Helvetica,20'
set output "druhy.eps"

set xrange [-5:5]
set yrange [0:450]

set grid
set xlabel "Ampacity difference [A]"
set ylabel "Frequency[Hz]"

set key at 4.8,430 box
set xtics -5,1,5

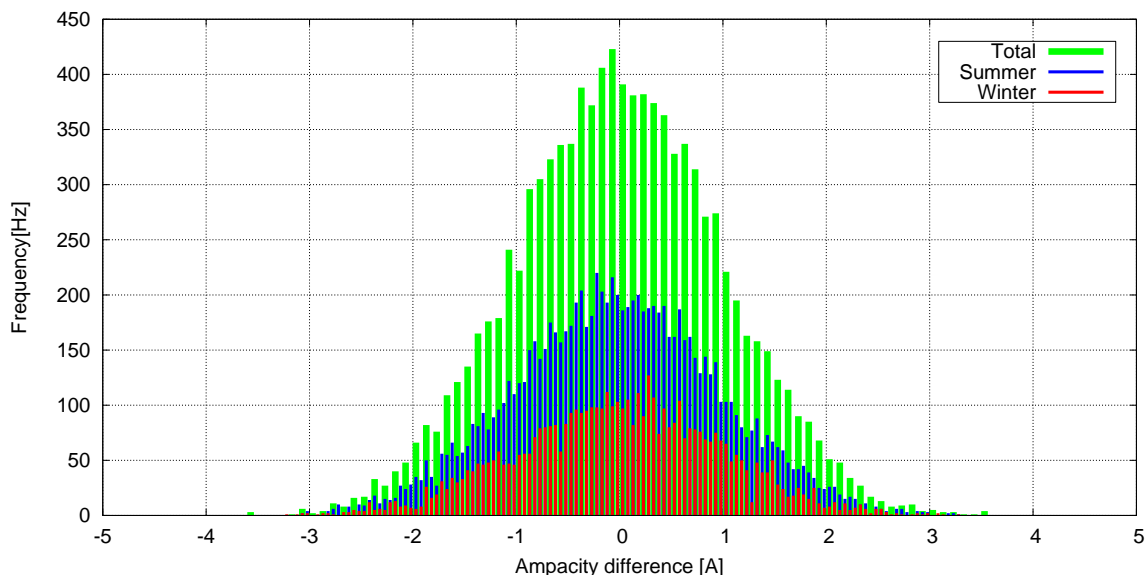
set linestyle 1 lt 1 lc 1 lw 6
set linestyle 2 lt 1 lc 2 lw 13
set linestyle 3 lt 1 lc 3 lw 6

```

```

plot "data2.txt" using 1:2 with impulses ls 2 title " Total",
"druhy_pomocny.txt" u 1:2 with impulses ls 3 title "Summer",
"druhy_pomocny.txt" u 1:3 with impulses ls 1 title "Winter"

```



Obrázek 34 - Frekvenční histogram proudové zatížitelnosti vodiče (Gnuplot)

4.2 Časová náročnost výpočtů

Matlab nabízí funkci „tic, toc“, která nám vrací celkový čas výpočtu v sekundách. Příklad použití je následující:

```
tic <sekvence příkazů> toc
```

Gnuplot nenabízí žádný příkaz, který by změřil časovou náročnost vykreslení, k dispozici je však příkaz přímo z Linuxové distribuce. Příkaz se jmenuje „time“ a použití je následující.

```
time <příkaz>
```

4.2.1 Teplotní závislosti

Matlab: 0.5495 s

Gnuplot: real 0m0.041s

user 0m0.012s

sys 0m0.024s

4.2.2 Histogramy

Matlab:	0.2606 s
Gnuplot:	real 0m0.355s
	user 0m0.020s
	sys 0m0.024s

Časová náročnost se u teplotních závislostí lišila podstatně více, než to bylo u vykreslení histogramů. Celkově se dá říct, že co se týče výpočetního času, tak je na tom Gnuplot o něco lépe, než Matlab.

5 Závěr

V rámci této práce byla vytvořena příručka, která je psaná od samotného spuštění programu, přes základní informace o Gnuplotu, dále je zde popsáno, jak správně pracovat s datovými soubory, jak začít vykreslovat základní objekty, základní popisky, nastavování jednotlivých os, atd. Snažil jsem se demonstrovat zde všechny možné vykreslovací typy, aby si mohl uživatel v první řadě udělat obrázek o tom, jakým množstvím možností Gnuplot disponuje. V práci jsem nakonec nepopsal všechno do větších detailů, jak jsem původně plánoval a to z důvodu její obsahové omezenosti. Na druhou stranu však bohatě poslouží k tomu, aby zde uživatel našel potřebné informace pro jeho vizualizaci výsledků. Do budoucna by se samozřejmě patřilo popsat veškeré možnosti, které Gnuplot nabízí.

Tato práce si žádala naučit se pracovat pro mě s úplně neznámým programem. Z vlastních zkušeností musím říct, že i mně tato příručka pomohla v řešení úloh v praktické části.

V praktické části jsem demonstroval vykreslení zcela totožných grafů, které srovnává grafické výstupy z Matlabu a Gnuplotu. Připojil jsem k tomu i zdrojové kódy, aby měl budoucí uživatel představu o tom, jak se celkové vykreslení grafu provádí. Dále zde byly zobrazeny menší velikosti jednotlivých grafů, aby vynikl rozdíl mezi porovnávanými softwary. Závěrem praktické části jsem pak zobrazil časovou náročnost vykreslování grafických výstupů.

Zpracování této práce mi přineslo seznámení se s úplně, pro mě neznámým programem, který mě překvapil v mnoha věcech, pro uživatele určitě přívětivých. Práce v Gnuplotu je po seznámení jednoduchá a výsledné grafy si uživatel může upravit přesně podle jeho představ.

Literatura

PHILIPP K. JANERT. *Gnuplot in Action: Understanding Data with Graphs*. United States: Manning Publications, 2009. ISBN 978-1933988399.

PINKAS, Petr. Seriál Gnuplot. [online]. 2001 [cit. 2013-05-07]. Dostupné z: <http://www.root.cz/serialy/gnuplot/>.

DUŠEK, František. *MATLAB a Simulink: úvod do používání*. Pardubice, 2000. ISBN 80-7194-273-1.

MUSILEK, Petr, Jana HECKENBERGEROVÁ a Md Mafijul Islam BHUIYAN. Spatial Analysis of Thermal Aging of Overhead Transmission Conductors. *IEEE Transactions on Power Delivery*. 2012, vol. 27, no. 3, 1196 - 1204.

HECKENBERGEROVÁ, Jana, Petr MUSILEK, Md Mafijul Islam BHUIYAN a Don O. KOVAL. Analysis of spatial and seasonal distribution of power transmission line thermal aging. *IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES)*. Waltham, MA, 2010, 22 - 27.