

Ján Buša

MAXIMA

Open source systém počítačovej algebry



<http://www.skosi.org>

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 1 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Táto publikácia vznikla s príspevím grantovej agentúry SR KEGA v tematickej oblasti „Nové technológie vo výučbe“ – projekt: 3/2158/04 – „Využitie OPENSOURCE softvéru vo výučbe na vysokých školách“.

Recenzovali: Michal Kaukič
Miloš Šrámek

ISBN 80-8073-641-3

Sadzba programom pdfT_EX

Copyright © 2006 Ján Buša

Ktokoľvek má dovolenie vyhotoviť alebo distribuovať doslovný opis tohoto dokumentu alebo jeho časti akýmkoľvek médiom za predpokladu, že bude zachované oznámenie o copyrighte a oznámenie o povolení, a že distribútor príjemcovi poskytne povolenie na ďalšie šírenie, a to v rovnakej podobe, akú má toto oznámenie.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

[Strana 2 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Obsah

Úvod	7
1 Prvé kroky	9
1.1 Inštalácia programu MAXIMA	9
1.1.1 Inštalácia v OS Windows	9
1.1.2 Inštalácia v OS GNU/Linux	11
1.1.3 Prostredia programu MAXIMA	13
1.2 Spustenie programu MAXIMA	13
1.3 Help programu MAXIMA	17
1.4 Ukončenie činnosti programu MAXIMA	21
2 Čísla, výrazy a funkcie	23
2.1 Výrazy a priradenia	23
2.2 Rôzne spôsoby zobrazenia výrazov	23
2.2.1 Príkaz kill	27
2.3 Zápis známych konštánt	28
2.4 Voľba presnosti výpočtu a zobrazovania reálnych čísel	28
2.5 Komplexné čísla	31
2.6 Operátory priradovania	36
2.7 Funkcie	37

2.7.1	Práca s reálnymi číslami	37
2.7.2	Trigonometrické funkcie	38
2.7.3	Špeciálne funkcie	40
2.8	Prostredie ev	41
2.9	Funkcie assume a forget	43
2.10	Príkaz declare	45
3	Zjednodušovanie výrazov a substitúcie	47
3.1	Funkcia ratsimp	47
3.2	Funkcia radcan	50
3.3	Funkcie expand, factor a gfactor	50
3.4	Funkcie subst a ratsubst	54
3.4.1	Zadanie pravidiel zjednodušovania výrazov	55
4	Riešenie úloh matematickej analýzy	57
4.1	Výpočet limit	59
4.2	Výpočet derivácií a parciálnych derivácií	62
4.3	Výpočet Taylorovho polynómu funkcií	66
4.4	Vyšetrovanie priebehu funkcie	70
4.4.1	Definičný obor	70
4.4.2	Nulové body	71
4.4.3	Body nespojitosti a asymptoty bez smernice	72
4.4.4	Asymptoty so smernicou v bodoch $\pm\infty$	72
4.4.5	Určovanie stacionárnych bodov	73
4.4.6	Určovanie inflexných bodov	73

Domovská stránka

Titulná strana

Obsah

◀◀ ▶▶

◀ ▶

Strana 4 z 156

Späť

Celá strana

Zatvoriť

Koniec

4.4.7	Zobrazenie grafu funkcie jednej premennej	74
4.5	Výpočet neurčitých a určitých integrálov	75
4.5.1	Substitúcia v integráloch	81
4.6	Výpočet extrémov funkcií viac premenných	82
4.6.1	Zobrazenie grafu funkcie dvoch premenných	86
4.7	Vyšetrovanie konvergenie číselných a mocninových radov .	91
4.8	Riešenie obyčajných diferenciálnych rovníc	96
4.8.1	Riešenie začiatočných úloh pre diferenciálne rovnice 1. rádu	96
4.8.2	Riešenie začiatočných a okrajových úloh pre diferen- ciálne rovnice 2. rádu	106
4.8.3	Riešenie sústav lineárnych diferenciálnych rovníc s kon- štantnými koeficientmi	113
4.9	Laplaceova transformácia	117
5	Riešenie algebrických úloh	123
5.1	Mnohočleny	123
5.2	Racionálne funkcie	127
5.3	Maticová algebra	129
5.3.1	Vytváranie matíc	131
5.3.2	Maticové operácie	131
5.3.3	Maticové funkcie	133
5.4	Riešenie lineárnych a nelineárnych (algebrických) rovníc a sústav	135
5.4.1	Funkcia solve	136

Domovská stránka

Titulná strana

Obsah



Strana 5 z 156

Späť

Celá strana

Zatvoriť

Koniec

5.4.2	Riešenie sústav lineárnych algebrických rovníc pomocou funkcie <code>linsolve</code>	137
5.4.3	Funkcia <code>algsys</code>	139
5.4.4	Numerické riešenie rovníc pomocou funkcie <code>find_root</code>	139
6	Štatistické funkcie	140
7	Tvorba grafov	144
7.1	Voľby príkazov <code>plot</code>	144
8	Programovanie v SPA MAXIMA	150
	Záver	153
	Použitá literatúra	154

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Strana 6 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Úvod

Program MAXIMA patrí medzi tzv. systémy počítačovej algebry (SPA), ktoré umožňujú vykonávanie symbolických aj numerických výpočtov (riešenia rovníc, derivovania, integrovania, a pod.) na počítači. Jedným z prvých SPA bol program Macsymba (projekt MAC's SYmbolic MANipulator), ktorého vývoj sa začal v roku 1968 v MIT (Massachusetts Institute of Technology), pozri (SOUZA, FATEMAN, MOSES a YAPP, 2004). Medzi ďalšie SPA patrili od začiatku systémy Reduce, CAMAL, Mathlab-68, PM a ALTRAN. Výrazný skok nastal, až keď sa objavili programy Maple (1985) a Mathematica (1988), inšpirované SPA Macsymba. Spomeňme ešte MuPAD a Derive.

Maxima je pokračovateľom SPA Macsymba. Bola navrhnutá a udržiavaná profesorom Williamom F. Schelterom z Univerzity v Texase od roku 1982, až do jeho smrti v roku 2001. V roku 1998 získal od Oddelenia energie (Department of Energy) súhlas na zverejnenie zdrojového kódu programu DOE Macsymba pod licenciou GNU Public License a v roku 2000 inializoval na SourceForge projekt MAXIMA, na údržbu a vývoj SPA DOE Macsymba pod terajším názvom MAXIMA. MAXIMA teda patrí medzi programy s otvoreným zdrojovým kódom – OPENSOURCE softvér. Program je možné kompilovať v rôznych OS, vrátane Windows, GNU/Linuxu a MacOS X. Predkompilovaný sa dá pre GNU/Linux a Windows bezplatne získať na stránke [SourceForge](#).

Maxima je systém na manipuláciu so symbolickými a numerickými výrazmi, vrátane derivovania, integrovania, výpočtu Taylorových polynó-

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 7 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

mov, Laplaceovej transformácie, riešenia obyčajných diferenciálnych rovníc, systémov lineárnych rovníc. Pracuje s polynómami, množinami, zoznamami, vektormi, maticami a tenzormi. Umožňuje získať veľmi presné výsledky použitím presných zlomkov a celých i desatinných čísel s ľubovoľnou presnosťou. Zobrazuje grafy funkcií jednej aj dvoch premenných.

Na stránke <http://maxima.sourceforge.net/> nájdete množstvo zaujímavých informácií, týkajúcich sa nielen programu MAXIMA, ale aj ďalších OPENSOURCE SPA a OPENSOURCE matematického softvéru. Rozsiahla dokumentácia, uvedená aj v zozname použitej literatúry, sa nachádza na stránke <http://maxima.sourceforge.net/docs.shtml>.

Z vyššie uvedeného vyplýva, že SPA MAXIMA sa dá využiť v rôznych oblastiach výskumu a výučby. Každý môže tento systém využívať za rovnakých podmienok (bezplatne) aj po ukončení štúdia. Cieľom tejto učebnice je oboznámiť čitateľa s inštaláciou programu a so základnými úlohami, ktoré sa s jeho pomocou dajú efektívne riešiť. Podrobný popis poskytuje rozsiahly (MAXIMA MANUAL, 2005). Je zrejmé, že mnohé príkazy v stručnej učebnici nebudú ani spomenuté, nielen opísané. Dôležité je urobiť prvý krok. Veríme, že táto knižka Vám tento prvý (ale rozhodný) krok uľahčí.

Chcem sa poďakovať recenzentom Ladislavovi Ševčovičovi a Milošovi Šrámkovi za pozorné prečítanie textu a cenné pripomienky, ktoré zlepšili čitateľnosť textu.

Košice 6. októbra 2006

autor

Domovská stránka

Titulná strana

Obsah



Strana 8 z 156

Späť

Celá strana

Zatvoriť

Koniec

1. Prvé kroky

1.1. Inštalácia programu MAXIMA

V tomto oddiele popíšeme inštaláciu programu MAXIMA v operačných systémoch Windows a GNU/Linux. Na stránke

http://sourceforge.net/project/showfiles.php?group_id=4933

je možné nájsť a stiahnuť si inštalačné rpm súbory pre OS GNU/Linux a súbor maxima-5.10.0.exe pre OS Windows.

1.1.1. Inštalácia v OS Windows

Proces inštalácie v OS Windows odštartujeme spustením inštalačného súboru. Po začiatocnom odsúhlasení licenčných podmienok a nastavení pracovného priečinka sa na obrazovke objaví výzva:

Domovská stránka

Titulná strana

Obsah



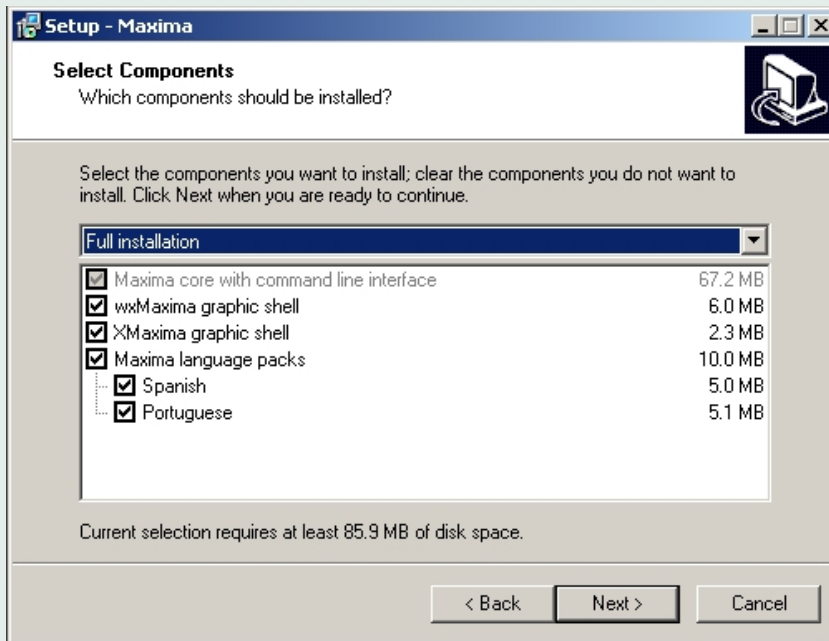
Strana 9 z 156

Späť

Celá strana

Zatvoriť

Koniec



Potvrdíme voľbu grafického prostredia WXMAXIMA, ktoré zjednodušuje prácu s programom MAXIMA. Objaví sa ďalšie okno, v ktorom môžeme potvrdiť umiestnenie ikonky programu na pracovnej ploche:

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



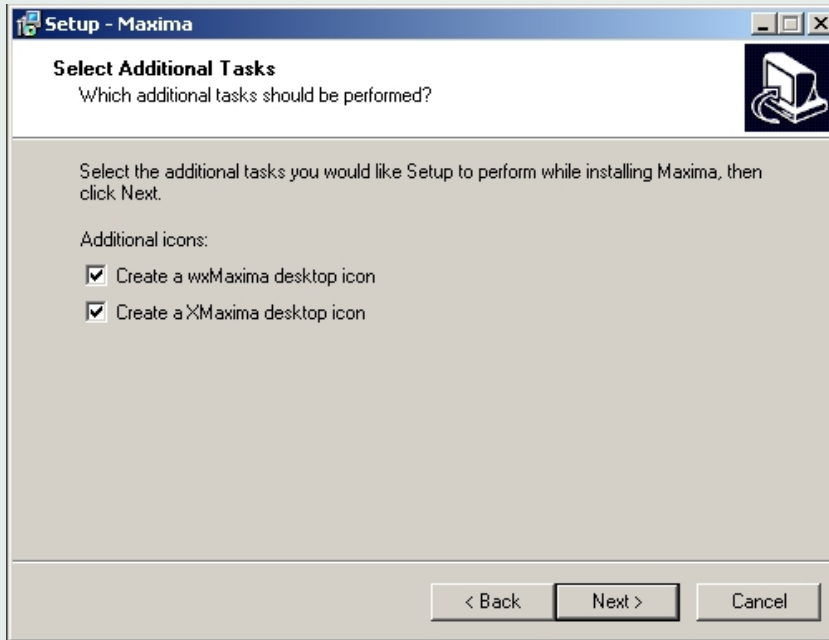
[Strana 10 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



Ďalej budeme predpokladať, že ste si program nainštalovali a máte možnosť jednotlivé ukážky overiť samostatne. Nie je to síce nutné, ale takáto štúdiom bude určite efektívnejšie.

1.1.2. Inštalácia v OS GNU/Linux

Pri inštalácii programu wxMAXIMA v OS GNU/Linux postavených na Debiane je problém, že program MAXIMA skompilovaný s GNU Lisp-om ne-

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Strana 11 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

bude fungovať s programom WXMAXIMA. Tento nedostatok je možné riešiť skompilovaním programu MAXIMA buď s balíkom clisp alebo s balíkom cmucl. Uvedieme postup inštalácie s balíkom cmucl:

1. odinštalujeme starú verziu programu MAXIMA aj balík gcl:

```
sudo apt-get remove --purge gcl maxima
```

2. nainštalujeme balík cmucl:

```
sudo apt-get install cmucl
```

3. stiahneme najnovšie zdrojáky programu MAXIMA:

<http://prdownloads.sourceforge.net/maxima/maxima-5.10.0.tar.gz?download>

(momentálne je najnovšia verzia 5.10.0)

4. skompilujeme program MAXIMA s balíkom cmucl namiesto gcl:

```
tar xzf maxima-5.10.0.tar.gz
cd maxima-5.10.0
./configure --enable-cmucl
make install
```

5. nainštalujeme program WXMAXIMA (akceptujte všetky závislosti!):

```
apt-get install wxmaxima
```

Týmto je inštalácia ukončená, teraz už môžeme spustiť program MAXIMA s grafickou nadstavbou wx, napr. z príkazového riadku zapísaním príkazu wxmaxima a odoslaním klávesom ENTER.

Domovská stránka

Titulná strana

Obsah



Strana 12 z 156

Späť

Celá strana

Zatvoriť

Koniec

1.1.3. Prostredia programu MAXIMA

V knihe The Maxima Book autorov (SOUZA, FATEMAN, MOSES a YAPP, 2004) sú popísané viaceré prostredia programu MAXIMA. Okrem vyššie spomenutých grafických prostredí WXMAXIMA a XMAXIMA je možné použiť prostredia:

- terminál – pôvodné prostredie, práca v príkazovom riadku,
- editor Emacs – prepracované negrafické prostredie, umožňuje kombinovať vstupy a výstupy programu MAXIMA s textom a po ich kompilácii programom $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ vznikne typograficky kvalitný dokument,
- $\text{T}_{\text{E}}\text{X}$ macs – vedecký WYSIWYG editor, spolupracujúci s rôznymi SPA. Dá sa využívať možnosť výstupu programu MAXIMA vo formáte $\text{T}_{\text{E}}\text{X}$,
- ďalšie prostredia nie je odporúčané používať, nie sú kvalitne udržiavané.

1.2. Spustenie programu MAXIMA

V Linuxe spustíme program MAXIMA v príkazovom riadku terminálu príkazom `maxima` alebo `xmaxima`. Po potvrdení stlačením klávesu ENTER sa otvorí okno programu MAXIMA alebo grafického prostredia `xMaxima`. V OS Windows klikneme na ikonku WXMAXIMA (ďalej budeme popisovať túto

možnosť) alebo XMAXIMA, čím zvolíme grafické prostredie programu MAXIMA. Príkazy môžeme zadávať v príkazovom riadku programu alebo použijeme položky menu, ku ktorým sa dostaneme použitím tlačidiel na hornej lište.

Po kliknutí na ikonku program Wxmaxima sa na obrazovke objaví okno zobrazené na obrázku 1, v hornej časti ktorého sa nachádza lišta so základnou ponukou (obrázok 2).

Položky na lište prostredia Wxmaxima preskúmame neskôr. Nič Vám však nebráni pootvárať ich a presvedčiť sa o bohatých možnostiach SPA MAXIMA.

V dolnej časti sa nachádza vstupný riadok (obrázok 3), v ktorom sa zadávajú príkazy programu MAXIMA a pod ním sú umiestnené tlačidlá vybraných funkcií programu MAXIMA.

Môžeme ho vyskúšať a postupne zadať, napríklad, príkazy uvedené v nasledujúcej ukážke v riadkoch, začínajúcich sa znakmi (%i*), kde * je poradové číslo vstupného riadku. Zadanie každého riadku potvrdíme klávesom ENTER. V prvom riadku zadávame príkaz na riešenie kvadratickej rovnice $x^2 + 2x + 3 = 0$, ktorá má dva komplexné korene $x_{1,2} = -1 \pm i\sqrt{2}$. Výpis je v symbolickom tvare, a preto ďalej vypisujeme 2. riešenie v desatinnom tvare s nastavenými rôznymi presnosťami.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



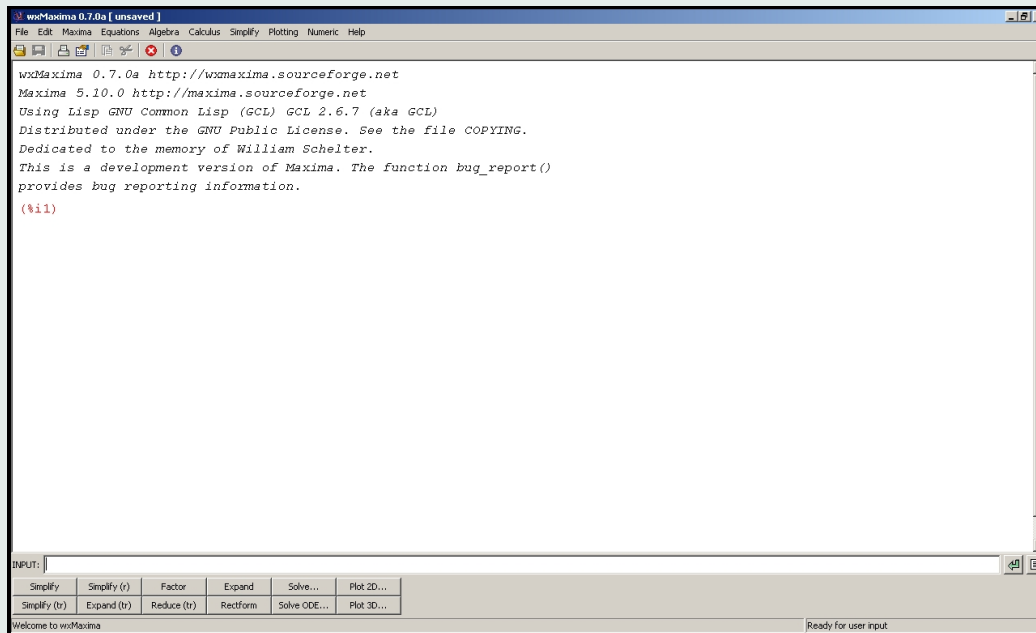
Strana 14 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



Obr. 1: Okno grafického prostredia wxMAXIMA

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

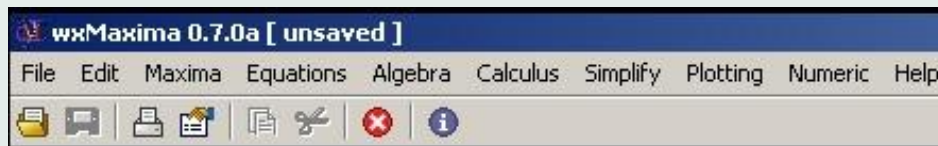
[Strana 15 z 156](#)

[Späť](#)

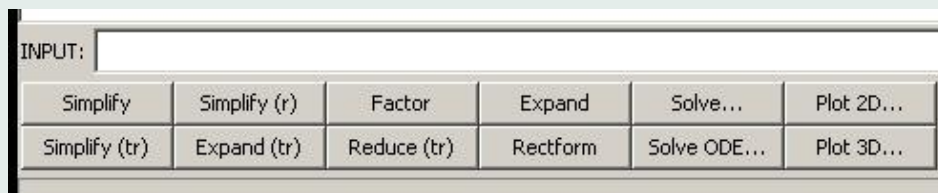
[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



Obr. 2: Položky menu programu MAXIMA



Obr. 3: Príkazový riadok

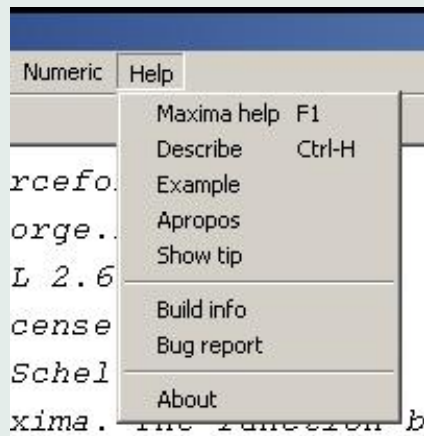

```
(%i1) solve(x^2+2*x+3);
(%o1) [ x = -√2 %i - 1 , x = √2 %i - 1 ]
(%i2) %o1[2],float;
(%o2) x = 1.414213562373095 %i - 1
(%i3) fpprec : 4;
(%o3) 4
(%i4) %o1[2],float;
(%o4) x = 1.414213562373095 %i - 1
(%i5) %o1[2],bfloat;
(%o5) x = 1.414b0 %i - 1.0b0
(%i6) fpprec : 50;
(%o6) 50
(%i7) %o1[2],bfloat;
(%o7) x = 1.414213562373095048801688724209698078569671875377b0 %i - 1.0b0
```

Vidíme, že z dvoch príkazov na výpis čísla s plávajúcou desatinnou bodkou (anglicky floating point) – float a bfloat, len druhý reaguje na zmenu nastavenia počtu desatinných miest príkazom fpprec (floating point precision).

1.3. Help programu MAXIMA

Grafické prostredie wxMAXIMA poskytuje pomocnú informáciu užívateľovi prostredníctvom položky Help (obrázok 4).

Online Maxima help v prvom riadku otvoríme aj stlačením klávesu F1. Príkazy v ďalších riadkoch – describe, example a apropos – je možné zadávať aj v príkazovom riadku, teda sa dajú využívať aj v terminálovom režime programu MAXIMA. Vyskúšajme ich:¹



Obr. 4: Možnosti položky Help

¹Pri dlhých výpisoch pokračuje MAXIMA v ďalšom riadku. Náš výpis môže byť rozdelený v inom mieste.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

[Strana 18 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

```
(%i8) apropos('pl');  
(%o8) [pl,playback,plog,plot,plot2d,plot2dopen,plot2d_ps,  
plot3d,plotheight,plotmode,plotting,plot_format,plot_options,  
plot_realpart,plus]
```

Všimnite si apostrof ' za otváracou zátvorkou príkazu apropos. Netradične je len jeden, bez párového kamaráta na konci výrazu pl. Iste ste už pochopili, že funkcia apropos nám pomáha spomenúť si na presný názov príkazu. Túto funkciu v niektorých systémoch plní kláves TAB, umožňujúci výpis všetkých funkcií, ktorých názvy majú na začiatku daný výraz.

[Domovská stránka](#)[Titulná strana](#)[Obsah](#)[Strana 19 z 156](#)[Späť](#)[Celá strana](#)[Zatvoriť](#)[Koniec](#)

```
(%i9) describe(plot2d);
0: plot2d :(maxima.info)Definitions for Plotting.
1: plot2d_ps :Definitions for Plotting.
Enter space-separated numbers, 'all' or 'none':
Still waiting: 0;
-- Function: plot2d (<expr>, <range>, ..., <options>, ...)
-- Function: plot2d (<parametric_expr>)
-- Function: plot2d (<discrete_expr>)
...
Displays a plot of one or more expressions as a function
of one variable.
...
See also 'plot_options', which describes plotting options
and has more examples.
(%o9) false
```

Funkcia `describe("názov")` vypíše popis zadaného príkazu.² V tejto ukážke program čaká na špecifikáciu príkazu, keďže výraz `plot2d` je na začiatku názvu dvoch funkcií. Po zadaní voľby a potvrdení klávesom ENTER nasleduje výpis.³

Funkcia `example(názov)` poskytne ukážky použitia zadaného príkazu.

²Názov je možné zadať aj bez úvodzoviek.

³Vynechali sme vyše 80 riadkov rôznych užitočných rád týkajúcich sa použitia funkcie `plot2d`.

Nasledujúci príklad ukazuje výsledok rôznych substitúcií, uskutočnených príkazom `subst`.⁴ K opisu jeho syntaxe sa ešte vrátíme neskôr:

```
(%i10) example(subst);
(%i11) subst(a,y+x,y+(y+x)^2+x)
(%o11) y+x+a^2
(%i12) subst(-%i,%i,b*%i+a)
(%o12) a-%i*b
(%i13) subst(x,y,y+x)
(%o13) 2*x
(%i14) subst(x = 0,diff(sin(x),x))
(%o14) 1
```

1.4. Ukončenie činnosti programu MAXIMA

V terminálovom režime ukončíme prácu programu príkazom `quit()`; . Prázdne zátvorky na konci sú *povinné*:

```
(%i15) quit;
(%o15) quit
(%i16) quit();
CLIENT: Lost socket connection ...
Restart maxima with 'Maxima->Restart maxima'.
```

⁴Zobrazili sme len neúplný výpis príkladov.

V grafickom režime Wxmaxima ukončíme činnosť programu MAXIMA prostredníctvom menu – File/Exit alebo zadaním Ctrl Q.

Domovská stránka

Titulná strana

Obsah



Strana 22 z 156

Späť

Celá strana

Zatvoriť

Koniec

2. Čísla, výrazy a funkcie

2.1. Výrazy a priradenia

Výrazy sa zadávajú pomocou obvyklých znakov operácií a okrúhlych zátvoriek. Umocnenie sa zadáva znakom \wedge alebo pomocou $**$. Na konci príkazu sa zadáva *bodkočiarka* ; alebo, ak chceme potlačiť zobrazenie výstupu, znak dolára \$. V jednom riadku môžeme zadať niekoľko príkazov.

```
(%i1) a:3$ b:4$ a+b;
(%i2)
(%i3)
(%o3) 7
(%i4) a**3;
(%o4) 27
```

Ako je vidieť, znakom *priradenia* slúži netradične *dvojbodka* : a nie =! V riadku %i1 sme premenným *a* a *b* priradili hodnoty bez ich zobrazenia, potom sme vypočítali a zobrazili hodnotu $a + b$.

2.2. Rôzne spôsoby zobrazenia výrazov

Zadajme výraz $\frac{x}{\sqrt{x^2 + 1}}$:

```
(%i5) x/sqrt(x^2+1);
(%o5) 
$$\frac{x}{\sqrt{x^2 + 1}}$$

(%i6)
```

Medzi podpoložkami menu v záložke Maxima (obrázok 5) nájdeme možnosť zmeny zobrazovania výstupu kliknutím na Change 2d display. Otvorí sa ďalšie okienko, zobrazené na obrázku 6.

Vyberieme, napríklad, voľbu ascii a potvrdíme ju. Po opätovnom zadaní vstupu (%i5) sa výstup zmení na nasledujúci:

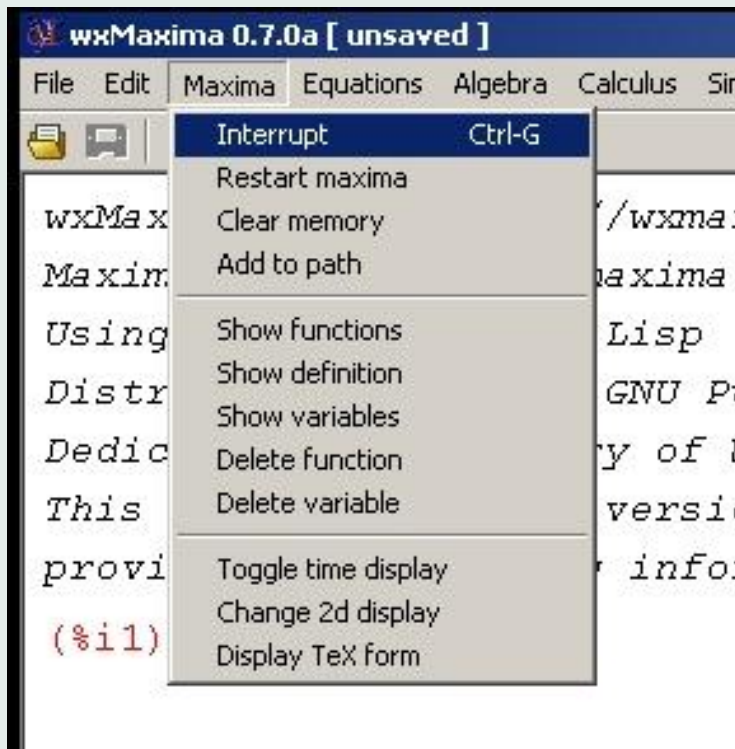
```
(%i6) set_display('ascii)$
(%i7) %i5;
(%o7) 
$$\frac{x}{\sqrt{x^2 + 1}}$$

```

Pri zadaní voľby none bude zobrazovanie lineárne (jednorozmerné), keď sa výrazy zapisujú do jedného riadku:

```
(%i8) set_display('none)$
(%i9) %i5;
(%o9) x/sqrt(x^2+1)
```

K pôvodnému krajšiemu a príjemnejšiemu zobrazovaniu výsledkov sa



Obr. 5: Voľby programu MAXIMA

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

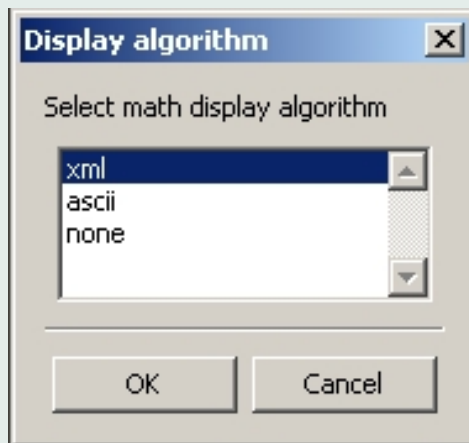
[Strana 25 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Konec](#)



Obr. 6: Režim zobrazovania výsledkov

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 26 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

v grafickom prostredí Wxmaxima vrátíme voľbou `xml`:

```
(%i10) set_display('xml)$
```

Na záver spomeňme ešte možnosť zapísať výstup vo formáte typografického programu $\text{T}_\text{E}_\text{X}$:

```
(%i11) tex(x/sqrt(x^2+1));
$$${{x}\over{\sqrt{x^2+1}}}$$$
(%o11) false
(%i12) tex(%o5);
$$${{x}\over{\sqrt{x^2+1}}}\leqno{\tt (\%o5)}$$$
(%o12) (%o5)
```

Na vstupe funkcie `tex` môžeme zadať buď samotný výraz alebo jeho odkaz na neho, prípadne jeho názov. Pri volaní funkcie `tex` je možné výstup zapísať do zadaného súboru.

2.2.1. Príkaz `kill`

Príkazom `kill` môžeme odstrániť premenné s ich všetkými priradeniami a vlastnosťami z aktuálneho pracovného prostredia programu MAXIMA:

```
(%i13) kill(a);
(%o13) done
(%i14) a;
(%o14) a
```

2.3. Zápis známych konštánt

V nasledujúcej tabuľke uvádzame zoznam konštánt používaných programom MAXIMA:

konštanta	MAXIMA
e – Eulerovo číslo	%e
i – imaginárna jednotka	%i
π – Ludolfovo číslo	%pi
∞ – reálne kladné nekonečno	inf
$-\infty$ – reálne záporné nekonečno	minf
∞ – komplexné nekonečno	infinity
lož – logická konštanta	false
pravda – logická konštanta	true

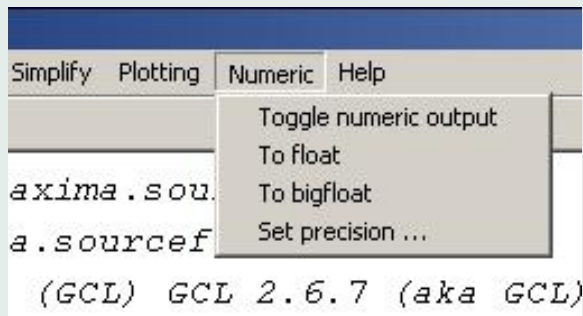
Aj konštanty, ktoré sú súčasťou vypočítaných výsledkov, uvádza MAXIMA znakom %.

2.4. Voľba presnosti výpočtu a zobrazovania reálnych čísel

MAXIMA dokáže pracovať s presnými reálnymi číslami, zapísanými v symbolickom tvare. O tom, či sa čísla zapisujú v symbolickom tvare alebo v tvare desatinných čísel rozhoduje nastavenie premennej numer.

V grafickom prostredí WXMAXIMA môžeme formát výpisu ovplyvniť v záložke Numeric, kde sa dá voľbou Toggle numeric output prepínať

medzi symbolickým a dekadickým zápisom:⁵



Obr. 7: Položka Numeric

⁵Z neznámych dôvodov sa to netýka Eulerovho čísla e ! V desatinnom tvare ho vypíšeme príkazom `%e`, `numer` alebo `float(%e)` (sú aj ďalšie možnosti).

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



[Strana 29 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

```
(%i15) numer:true;  
(%o15) true  
(%i16) sqrt(5);  
(%o16) 2.23606797749979  
(%i17) numer:false;  
(%o17) false  
(%i18) %e;  
(%o18) %e  
(%i19) %e,numer;  
(%o19) 2.718281828459045
```

Maxima zobrazuje štandardne 16 cifier desatinných čísel (medzi nimi sa ešte zobrazuje desatinná bodka). Zmenou hodnoty premennej `fpproc` môžeme dosiahnuť inú presnosť, ktorá sa však prejaví len pri použití výstupu `bfloat` (Big Float, pozri obr. 7), nie `float` – ten zobrazuje vždy rovnako. Presnosť môžeme prakticky neohraničene zvýšiť aj znížiť. Hodnotu `fpprec` je možné zmeniť aj lokálne (pozri riadok `(%i23)`) v rámci jedného príkazu⁶:

⁶Zhodou okolností by sme rovnaký výsledok získali pri nastavení `fpprec=12`, program MAXIMA po zaokrúhlení nevypísal poslednú nulu.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 30 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

```
(%i20) fpprec:40;
(%o20) 40
(%i21) %pi,bfloat;
(%o21) 3.141592653589793238462643383279502884197b0
(%i22) float(%pi);
(%o22) 3.141592653589793
(%i23) bfloat(%pi), fpprec=13;
(%o23) 3.14159265359b0
(%i24) %e, bfloat;
(%o24) 2.718281828459045235360287471352662497757b0
```

2.5. Komplexné čísla

Komplexné čísla môžeme zadať v algebrickom tvare použitím konštanty $%i$ – imaginárnej jednotky programu MAXIMA:

```
(%i25) z1:2+3*%i;
(%o25) 3 %i + 2
(%i26) z2:4-5*%i;
(%o26) 4 - 5 %i
```

Výsledok súčinu dvoch čísel by sme radšej videli priamo v zjednodušenom tvare. Toto dosiahneme až použitím príkazu `expand` alebo príkazu `rectform`:

```
(%i27) z1*z2;
(%o27) (4 - 5 %i) (3 %i + 2)
(%i28) expand(%);
(%o28) 2 %i + 23
(%i29) rectform(z1*z2);
(%o29) 2 %i + 23
```

V prípade podielu dvoch komplexných čísel však pomôže už len príkaz `rectform`⁷:

```
(%i30) z1/z2;
(%o30) 3 %i + 2
-----
4 - 5 %i
(%i31) expand(%);
(%o31) 3 %i 2
----- + -----
4 - 5 %i 4 - 5 %i
(%i32) rectform(z1/z2);
(%o32) 22 %i 7
----- - ---
41 41
```

Samozrejme máme k dispozícii reálnu a imaginárnu zložku komplexných čísel:

⁷Vyskúšajte samostatne, ako v prípade súčinu aj podielu komplexných čísel funguje funkcia `trigrat`.


```
(%i33) realpart(z1);
(%o33) 2
(%i34) imagpart(z1/z2);
(%o34) 22
--
41
```

Komplexné číslo v algebrickom tvare môžeme vypísať v exponenciálnom tvare:

```
(%i35) z3:2+2*%i;
(%o35) 2 %i + 2
(%i36) polarform(z3);
(%o36) 3/2 4
         %i %pi
         -----
         2 %e
```

Príkazmi `abs` a `carg` získame absolútnu hodnotu (modul, veľkosť) a argument komplexného čísla:

```
(%i37) abs(z3);
                                3/2
(%o37)                          2
(%i38) carg(z3);
                                %pi
(%o38)                          ---
                                4
```

Ak zadáme komplexné číslo v exponenciálnom tvare, môže byť transformované na algebraický tvar, ak sa to dá urobiť presne (bez použitia funkcií sínus a kosínus). Všimnite si, ako sa číslo z_5 (s iracionálnymi zložkami) zmenilo na približné s racionálnymi zložkami po priradení v riadku (%i42). Porovnajte ešte rozdiel pri použití funkcií numer a float:

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 34 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

```
(%i39) z4:3*exp(2*i*pi/3);
(%o39)          sqrt(3) %i  1
              3 (----- - -)
                  2      2

(%i40) rectform(z4);
(%o40)          3/2
              3  %i  3
              ---- - -
                  2      2

(%i41) z5:exp(2*i*pi/17);
(%o41)          %e
              2 %i %pi
              ----
                  17

(%i42) z5:exp(2*i*pi/17),numer;
(%o42)          0.36124166618715 %i + 0.93247222940436

(%i43) z5;
(%o43)          0.36124166618715 %i + 0.93247222940436

(%i44) exp(2*i*pi/17),float;
(%o44)          0.11764705882353 %i %pi
              %e
```



2.6. Operátory priradovania

Operátor `:` používame na priradenie hodnôt alebo výrazov premenným. Týmto spôsobom však nedefinujeme funkcie:

```
(%i45) y:x^2-x+1;
(%o45)
          2
         x  - x + 1
(%i46) y(2);
          2
y evaluates to x  - x + 1
Improper name or value in functional position.
-- an error. Quitting. To debug this try debugmode(true);
```

Ako vidíme, $y = x^2 - x + 1$ nie je funkcia, je to len výraz. Ak chceme získať hodnotu výrazu napr. pre $x = 2$, musíme do výrazu y dosadiť číslo 2 namiesto x :

```
(%i47) subst(2,x,y);
(%o47)
          3
```

Treba si dávať pozor na poradie premenných vo funkcii `subst`. Porozmýšľajte, čo bude výsledkom príkazov `subst(y,x,2)` alebo `subst(x,2,y)`:

```
(%i48) subst(y,x,2);
(%o48)
          2
(%i49) subst(x,2,y);
          x
(%o49)
         x  - x + 1
```

Ak chceme vytvoriť funkciu argumentu x , musíme použiť priradenie s operátorom $:=$. Napríklad:

```
(%i50) f(x) := x^2 - x + 1;
```

```
(%o50)                2
                    f(x) := x  - x + 1
```

```
(%i51) f(2);
```

```
(%o51)                3
```

2.7. Funkcie

MAXIMA, podobne ako Mathematica alebo Maple, má oveľa väčší počet rôznych funkcií ako štandardné programovacie jazyky. Ich zoznam je uvedený na stranách 441–451 v knihe ([MAXIMA MANUAL, 2005](#)), v ktorej nájdete aj ich popis. Nižšie uvádzame len niektoré z nich. Nezabudnite, že na získanie cenných informácií o príkazoch a funkciách môžete použiť funkcie `describe` a `example`.

2.7.1. Práca s reálnymi číslami

Na prácu s reálnymi číslami má MAXIMA nasledujúce:

funkcie: `bffac`, `bfloat`, `bfloatp`, `bfpsi`, `bfpsi0`, `cbffac`, `float`, `floatnump`, `?round`, `?truncate` a



premenné: `algepsilon`, `bftorat`, `bftrunc`, `float2bf`, `fpprec`,
`fpprintprec`.

Popis nájdete v kapitole 10 knihy ([MAXIMA MANUAL, 2005](#)).

2.7.2. Trigonometrické funkcie

MAXIMA pozná nasledujúce *trigonometrické* funkcie: `acos`, `acosh`, `acot`, `acoth`,
`acsc`, `acsch`, `asec`, `asech`, `asin`, `asinh`, `atan`, `atan2`, `atanh`, `cos`, `cosh`, `cot`,
`coth`, `csc`, `csch`, `sec`, `sech`, `sin`, `sinh`, `tan`, `tanh`. Na manipuláciu s vý-
razmi obsahujúcimi trigonometrické funkcie slúžia funkcie `trigexpand`,
`trigreduce`, `trigsimp`, `trigrat`. Balíky `atrig1`, `ntrig` a `spangl` obsahujú
ďalšie pravidlá zjednodušovania trigonometrických funkcií. Ak sa chcete
dozvedieť viac, prečítajte si kapitolu 15 knihy ([MAXIMA MANUAL, 2005](#)).

MAXIMA pozná aj niektoré hodnoty trigonometrických funkcií. Naprí-
klad:



```
(%i52) ratsimp(sqrt((1-cos(%pi/4))/2));
          sqrt(sqrt(2) - 1)
(%o52)  -----
          3/4
          2

(%i53) tan(%pi/8);
(%o53) tan(%pi/8)

(%i54) ratsimp(%);
(%o54) tan(%pi/8)

(%i55) load(spangl);
(%o55) C:/PROGRA~1/MAXIMA~1.0/share/maxima/5.10.0/share/
      /trigonometry/spangl.mac

(%i56) tan(%pi/8);
(%o56) sqrt(2)-1
```

Niekedy by sme očakávali, že MAXIMA vydá výsledok trigonometrickej funkčnej hodnoty v algebrickom tvare (čo je síce krajšie a vhodné na ďalšie úpravy, ale možno nie jednoduchšie z hľadiska výpočtovej náročnosti). Napríklad:

```
(%i57) sin(a/2);
(%o57) sin(a/2)

(%i58) sin(%pi/8);
(%o58) sin(%pi/8)
```

Nasledujúcim príkazom sprístupníme použitie vzorcov pre polovičný uhol

(treba si však uvedomiť, že vzorec je platný len pre určité uhly):

```
(%i59) halfangles:true$
(%i60) sin(a/2);
(%o60) sqrt(1-cos(a))/sqrt(2)
(%i61) sin(%pi/8);
(%o61) sin(%pi/8)
(%i62) subst(%pi/4,a,%o60);
(%o62) sqrt(1-1/sqrt(2))/sqrt(2)
(%i63) radcan(%o62);
(%o63) sqrt(sqrt(2)-1)/2^(3/4)
```

Hoci program MAXIMA už „pozná“ vzorce pre polovičné uhly, nenapadne ho, že ich má v riadku 61 použiť.⁸ Poradíme si však tak, že použijeme substitúciu za a v riadku 62. Na záver sme ešte použili jeden zo základných spôsobov zjednodušovania výrazov obsahujúcich odmocniny (radikály).

Ak by ste pracovali s trigonometrickými funkciami, je potrebné nastudovať ďalšie bohaté možnosti, na začiatok môžeme odporučiť knihu ([MAXIMA MANUAL, 2005](#)).

2.7.3. Špeciálne funkcie

Hoci špeciálne funkcie samozrejme tvoria súčasť programu MAXIMA, nebudeme sa venovať ich popisu. Táto tématika vychádza za rámec tejto

⁸Pri príprave textu sa nám podarilo dosiahnuť aj stav, keď program MAXIMA samostatne upravil výraz $\sin(\pi/8)$. Nepodarilo sa nám však tento stav zopakovať znova.



úvodnej učebnice. Ak na svoju prácu potrebujete špeciálne funkcie, otvoríte kapitoly 16 a 18 knihy ([MAXIMA MANUAL, 2005](#)).

2.8. Prostredie ev

Všetky operácie programu MAXIMA sa uskutočňujú v nejakom prostredí, v ktorom systém predpokladá platnosť určitých podmienok, ktoré sa dajú meniť. Často potrebujeme zmeniť správanie systému pri nejakých výpočtoch bez toho, aby sme vykonali globálne zmeny. Na to MAXIMA poskytuje príkaz `ev`⁹, ktorý je jedným z najvýkonnejších a ktorý umožňuje definovať lokálne prostredie v rámci jedného príkazu. Ak užívateľ zvládne prácu s funkciou `ev` čo najskôr, získa veľkú výhodu a úžitok. V tomto oddieli ukážeme len niektoré možnosti tohoto príkazu. Podľa príručky ([RAND, 2005](#)) má táto funkcia nasledujúcu syntax:

$$\text{ev}(a, b_1, b_2, \dots, b_n)$$

Výraz a sa vyhodnotí pri platnosti podmienok b_1, b_2, \dots, b_n . Týmito „podmienkami“ môžu byť aj rovnice, priradenia, slová (napríklad `numer` alebo `diff`).

Ako prvý uveďme jednoduchý príklad riešenia lineárnej algebrickej rovnice:

⁹ev je asi skratka od evaluate.

```
(%i64) ev(solve(a*x+b=0),x,a:3,b=12);  
(%o64) [x=-4]  
(%i65) a;  
(%o65) a
```

Vidíme, že (rôzne) priradenia hodnôt premenným a a b vo vnútri prostredia `ev` sú len lokálne. V nasledujúcom riadku 65 nemá premenná a priradenú hodnotu.

Nasledujúci príklad je prevzatý z príručky (SOUZA, FATEMAN, MOSES a YAPP, 2004), kde nájdete aj ďalšie príklady:

```
(%i66) a:9/4;  
(%o66) 9/4  
(%i67) exp(a);  
(%o67) %e^(9/4)  
(%i68) ev(exp(a),float);  
(%o68) 9.487735836358526
```

a v podobnom duchu ďalej:

```
(%i69) ev(exp(a*x));
(%o69) %e^((9*x)/4)
(%i70) ev(exp(a*x),float);
(%o70) %e^(2.25*x)
(%i71) ev(exp(a*x),x=2);
(%o71) %e^(9/2)
(%i72) ev(exp(a*x),numer,x=2);
(%o72) 90.01713130052181
```

2.9. Funkcie `assume` a `forget`

V niektorých situáciách je dobré predpokladať splnenie určitých podmienok. Pomocou príkazu `assume` – „predpokladajme“ – oznámime programu MAXIMA potrebnú informáciu.¹⁰ Syntax príkazu je

```
assume(predikát_1,predikát_2,...,predikát_n)
```

Predikáty `predikát_1`, `predikát_2`, ..., `predikát_n` môžu byť len výrazy zadávané pomocou relačných operátorov `<`, `<=`, `equal`, `notequal`, `>=` a `>`.¹¹

Platnosť predpokladu zadaného príkazom `assume(predikát)` zrušíme príkazom `forget(predikát)`:

¹⁰V nápovede SPA MAXIMA sa odporúča zoznámiť sa aj s pojmi `is`, `facts`, `context` a `declare`.

¹¹Podmienku $n \neq -1$ zadáme v tvare `notequal(n,-1)`, podobne sa použije operátor `equal`.



```
(%i73) sqrt(a^2);
(%o73) |a|
(%i74) assume(a<0);
(%o74) [a<0]
(%i75) assume(a>=0);
(%o75) [inconsistent]
(%i76) assume(a<=0);
(%o76) [redundant]
(%i77) sqrt(a^2);
(%o77) -a
(%i78) log(a^2);
(%o78) 2*log(a)
(%i79) log(-1), numer;
(%o79) 3.141592653589793*i
(%i80) float(log(1+i));
(%o81) 0.78539816339745*i+0.34657359027997
(%i81) forget(a<0)$
(%i82) sqrt(a^2);
(%o82) |a|
```

Riadky 75 a 76 ukazujú reakciu SPA MAXIMA na pokus predefinovať už existujúci predpoklad. Zmenu predpokladu môžeme uskutočniť až po jeho „zabudnutí“ príkazom `forget` – zabudni. Riadok 78 svedčí o tom, že MAXIMA neoveruje podmienku kladnosti vstupného argumentu logaritmu. Vysvetlením môže slúžiť riadok 79. Program MAXIMA totiž dokáže pra-

covať s logaritmiami záporných (ale aj komplexných – pozri riadok 80) čísel.¹²

2.10. Príkaz declare

Podobnú funkciu ako príkaz assume plní príkaz declare($a_1, v_1, a_2, v_2, \dots$), kde „atóm“ a_i má vlastnosť v_i . Tento príkaz umožňuje definovať veľký počet rôznych vlastností nielen pre premenné, pre ktoré uvedieme niekoľko príkladov. Zrušenie predpokladu (i premennej) môžeme uskutočniť príkazom kill.¹³

¹²Bez hlbšieho štúdia programu MAXIMA je nemožné povedať, ako sa dá zúžiť oblasť vstupných argumentov logaritmu len na kladné reálne čísla, aby sme získali „klasický“ logaritmus.

¹³Iný spôsob zrušenia platnosti deklarácie sme zatiaľ nenašli.

Domovská stránka

Titulná strana

Obsah



Strana 45 z 156

Späť

Celá strana

Zatvoriť

Koniec

```
(%i83) (-1)^n;  
(%o83) (-1)^n  
(%i84) declare(n,even)$  
(%i85) (-1)^n;  
(%o85) 1  
(%i86) declare(n,odd)$  
Inconsistent Declaration: declare(n,odd)  
-- an error. Quitting. To debug this try debugmode(true);  
(%i87) kill(n)$  
(%i88) declare(n,odd)$  
(%i89) (-1)^n;  
(%o89) -1  
(%i90) kill(n)$  
(%i91) limit(sin(%pi*n),n,inf);  
(%o91) ind  
(%i92) declare(n,integer)$  
(%i93) limit(sin(%pi*n),n,inf);  
(%o93) 0  
(%i94) abs((-1)^n);  
(%o94) |(-1)^n|
```

Riadok 86 svedčí o tom, že deklarácia sa nedá zmeniť opätovným použitím príkazu `declare`. Hoci výsledok v riadku 93 je očakávaný pre celočíselnú hodnotu n , nie je jasné, prečo MAXIMA nedokáže určiť absolútnu hodnotu v riadku 94.



3. Zjednodušovanie výrazov a substitúcie

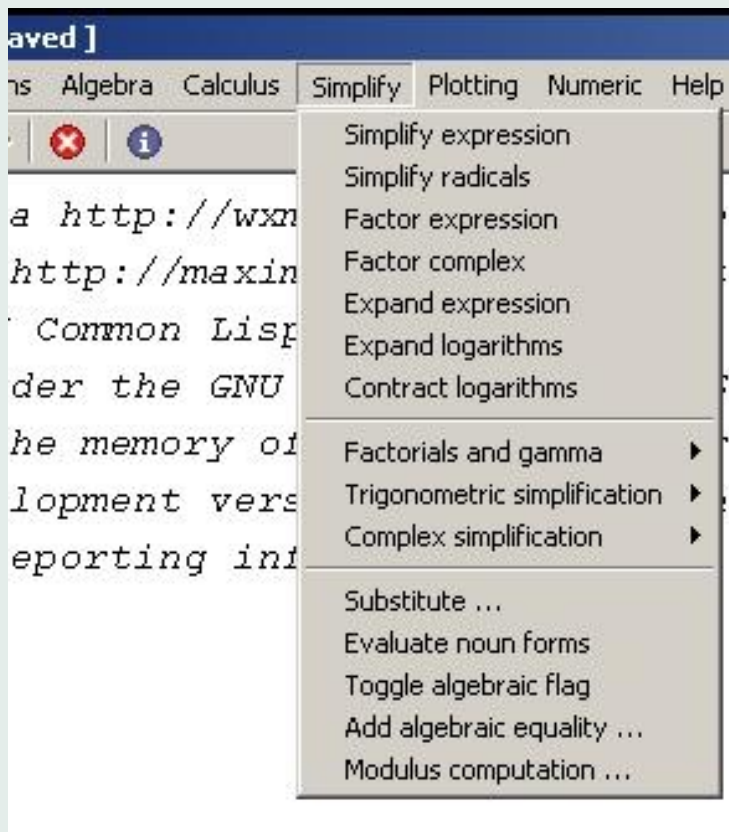
Na obrázku 8 vidíme výpis podpoložiek položky Simplify. Niektoré z týchto podpoložiek sú umiestnené aj pod príkazovým riadkom SPA MAXIMA. Podrobný popis všetkých možností nie je možný v rámci tejto úvodnej učebnice. Ak potrebujete získať podrobnejšie informácie o jednotlivých možnostiach, využite funkcie `describe` a `example`, prípadne nazrite do odporúčanej literatúry.

3.1. Funkcia `ratsimp`

Nahliadnime do príkladov, ktoré ponúka príkaz `example(ratsimp)`:

```
(%i1) example(ratsimp);
(%i2) sin(x/(x+x^2)) = %e^((1+log(x))^2-log(x)^2)
(%o2)
          x      (log(x) + 1)2 - log(x)2
      sin(-----) = %e
          2
         x  + x

(%i3) ratsimp(%)
(%o3)
          1      2
      sin(-----) = %e x
          x + 1
```



Obr. 8: Položka Simplify hlavního menu

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

[Strana 48 z 156](#)

[Spät](#)

[Celá strana](#)

[Zatvorit'](#)

[Konec](#)


```
(%i4) a+b*x+b*(a/b-x)
(%o4)
          a
      b x + b (- - x) + a
          b

(%i5) ratsimp(%)
(%o5)

          2 a
(%i6) ((x-1)^(3/2)-(1+x)*sqrt(x-1))/sqrt(x-1)/sqrt(1+x)
          3/2
      (x - 1)  - sqrt(x - 1) (x + 1)
-----
          sqrt(x - 1) sqrt(x + 1)

(%o6)

          2
(%i7) ratsimp(%)
          - -----
          sqrt(x + 1)

(%i8) ev(x^(1/a+a),ratsimpexpons)
          2
          a + 1
          -----
          a

(%o8)
          x
```

3.2. Funkcia radcan

S príkazom radcan sme sa už stretli vyššie v oddiele 2.7.2 na strane 40. Táto funkcia slúži na zjednodušovanie výrazov obsahujúcich logaritmy, exponenty a odmocniny (radikály), ktoré konvertuje na kanonický tvar.

3.3. Funkcie expand, factor a gfactor

Uvedme jednoduchuchý príklad použitia príkazov expand, factor a gfactor:

```
(%i9) v:expand((x+1)*(x-2)*(x^2+9));
(%o9)          4      3      2
              x  - x  + 7 x  - 9 x - 18
(%i10) factor(v);
(%o10)          2
              (x - 2) (x + 1) (x  + 9)
(%i11) gfactor(v);
(%o11)          (x - 2) (x + 1) (x - 3 %i) (x + 3 %i)
```

Ako vidíme, príkaz expand dokáže, napríklad, roznásobiť (expandovať) výrazy uvedené v zátvorkách. Opačným príkazom, ktorý dokáže výrazy napísať v tvare súčinu (faktorizovať ich)¹⁴, je príkaz factor. V uvedenom príklade sme príkazom factor(v) získali rozklad polynómu v na súčin koreňových činiteľov **nad poľom reálnych čísel**. Príkaz gfactor vykonal

¹⁴Koeficienty po faktorizácii musia byť racionálne alebo jednoduché radikály.

rozklad polynómu na súčin koreňových činiteľov **nad poľom komplexných čísel**.

Funkcia `factor` pre celočíselný vstupný argument vráti rozklad celého čísla na súčin prvočísel:

```
(%i12) factor(1440);  
(%o12) 2^5*3^2*5
```

Popis funkcie `factor` pomocou príkazu `describe` je rozsiahly – má 20 podpoložiek.

Príkazom `expand(v, p, n)` rozvineme len členy, ktorých mocniny sú od $-n$ po p .¹⁵

```
(%i13) expand((x+2)^2*(x-1)^3*(x+3)^4/(x-2)^2/(x+1), 3, 1);  
(%o13) (x^5*(x+3)^4)/((x-2)^2*x+(x-2)^2)+  
(x^4*(x+3)^4)/((x-2)^2*x+(x-2)^2)-  
(5*x^3*(x+3)^4)/((x-2)^2*x+(x-2)^2)-  
(x^2*(x+3)^4)/((x-2)^2*x+(x-2)^2)+  
(8*x*(x+3)^4)/((x-2)^2*x+(x-2)^2)-  
(4*(x+3)^4)/((x-2)^2*x+(x-2)^2)
```

Činnosť príkazu `expand`, resp. správanie programu MAXIMA ovplyvňujú nastavenia rôznych systémových premenných. Pozrime sa, napríklad, na spracovanie výrazov obsahujúcich logaritmy pri rôznych hodnotách premennej `logexpand`.¹⁶

¹⁵Výpis v okne Wxmaxima je na jednom riadku.

¹⁶Implicitná hodnota tejto premennej je `true`.

```
(%i14) log(a^b);  
(%o14) log(a)*b  
(%i15) log(x*b^2/y^3);  
(%o15) log((b^2*x)/y^3)
```

Po zmene premennej logexpand dostaneme:

```
(%i16) logexpand:false$  
(%i17) log(a^b);  
(%o17) log(a^b)  
(%i18) log(x*b^2/y^3);  
(%o18) log((b^2*x)/y^3)
```

a nakoniec nastavenie na hodnotu super:

```
(%i19) logexpand:super$  
(%i20) log(a^b);  
(%o20) log(a)*b  
(%i21) log(x*b^2/y^3);  
(%o21) -3*log(y)+log(x)+2*log(b)
```

Poznámka 3.1. Podobne ako pri iných úpravách, aj v tomto prípade si treba uvedomiť, že aplikácia daného vzorca nemusí byť správna. V uvedenom príklade je nesprávna napríklad v prípade $b < 0$, keď vstup riadku 21 má zmysel a výstup nie (vymyslíte ešte iný prípad, kedy je táto úprava nesprávna).

Funkcia `logcontract` slúži na spájanie viacerých logaritmov do jedného:

```
(%i22) log(a)+log(b);
(%o22) log(b)+log(a)
(%i23) logcontract(%);
(%o23) log(a*b)
```

Podobne môžeme ovplyvniť správanie činnosti SPA MAXIMA zmenou nastavenia hodnoty `trigexpand`¹⁷, ktorej prednastavená hodnota je `false`:

```
(%i24) trigexpand;
(%o24) false
(%i25) sin(3*x+y);
(%o25) sin(y+3*x)
(%i26) trigexpand(sin(3*x+y));
(%o26) cos(3*x)*sin(y)+sin(3*x)*cos(y)
(%i27) trigexpand:true$
(%i28) sin(2*x+y);
(%o28) (cos(x)^2-sin(x)^2)*sin(y)+2*cos(x)*sin(x)*cos(y)
```

Informácie o ďalších systémových premenných (napríklad `besselexpand`, `expandwrt_denom`, `faceexpand`, `psexpand`, `radexpand`, `ratexpand`, `sumexpand`, `taylor_logexpand`, `trigexpandplus`, `trigexpandtimes`) a rôznych funkciách súvisiacich s úpravami výrazov nájdete v knihe ([MAXIMA MANUAL, 2005](#)).

¹⁷Rovnaký názov `trigexpand` majú systémová premenná aj funkcia!





3.4. Funkcie subst a ratsubst

Funkciu `subst` sme použili v oddiele 2.7.2 na strane 40 na výpočet hodnoty $\sin(\pi/8)$ a predtým pri definovaní výrazov obsahujúcich premenné. Príkaz má nasledujúcu syntax:

```
subst(vyraz1, vyraz2, vyraz3)
```

kde `vyraz1` bude dosadený za `vyraz2` do výrazu `vyraz3`. V najjednoduchšom prípade dostávame napríklad:

```
(%i29) subst(2*b, a, a^2+b^2);
(%o29) 5*b^2
```

Nahradený môže byť celý výraz, nielen premenná:

```
(%i30) subst(p, x+y, x+(x+y)^2+y);
(%o30) y+x+p^2
(%i31) subst(p, x+y, x+y+(x+y)^2);
(%o31) y+x+p^2
(%i32) ratsubst(p, x+y, x+(x+y)^2+y);
(%o32) p^2+p
```

V riadku 30 sme sa pokúsili nahradiť výraz $x + y$ premennou p . Keďže premenné x a y boli oddelené výrazom v zátvorke, pokúsili sme sa situáciu zlepšiť zmenou zápisu vstupného výrazu. Ako vidieť, nepomohlo to. Funkcia `ratsubst` si s touto situáciou poradila.

Substitúciu môžeme použiť aj na vytvorenie komplexne združeného čísla:

```
(%i33) subst(-%i,%i,a-3*%i);  
(%o34) a+3*%i
```

Na mieste výrazu `vyraz3` môže byť aj nejaký operátor, ktorý sa najskôr vykoná a potom bude vykonaná substitúcia:

```
(%i35) subst(%pi/2,x,diff(sin(2*x),x));  
(%o35) -2
```

V časti o integrovaní sa ešte vrátíme k substitúciám pri výpočte integrálov. Ďalšie informácie o substitúciách nájdete v odporúčanej literatúre.

3.4.1. Zadanie pravidiel zjednodušovania výrazov

V predošlom sme pomocou príkazu `ratsubst` zjednodušili výraz (%i32). Ak chceme podobnú operáciu vykonávať častejšie, môžeme zadať pravidlo zjednodušovania výrazov, ktoré potom využije funkcia `ratsimp`.¹⁸

```
(%i36) tellrat(x+y=p);  
(%o36) [y+x-p]  
(%i37) ratsimp(x^2-y^2+x+(x+y)^2+2*y);  
(%o37) (2*p-1)*x+2*p
```

V tomto prípade nie je jednoduché dopredu získať rovnaký výsledok, aký nachádzame v riadku 37. O jeho správnosti sa presvedčíme tým, že vykonáme to isté iným spôsobom:

¹⁸Príkaz `unte11rat` by mal SPA MAXIMA pomôcť zabudnúť na zadané pravidlo. Žiaľ, zabudnúť nám pomohol až príkaz `kill(a11)`!

```
(%i38) subst(p-x,y,x^2-y^2+x+(x+y)^2+2*y);  
(%o38) x^2+x-(p-x)^2+2*(p-x)+p^2  
(%i39) ratsimp(%);  
(%o39) (2*p-1)*x+2*p
```

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 56 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



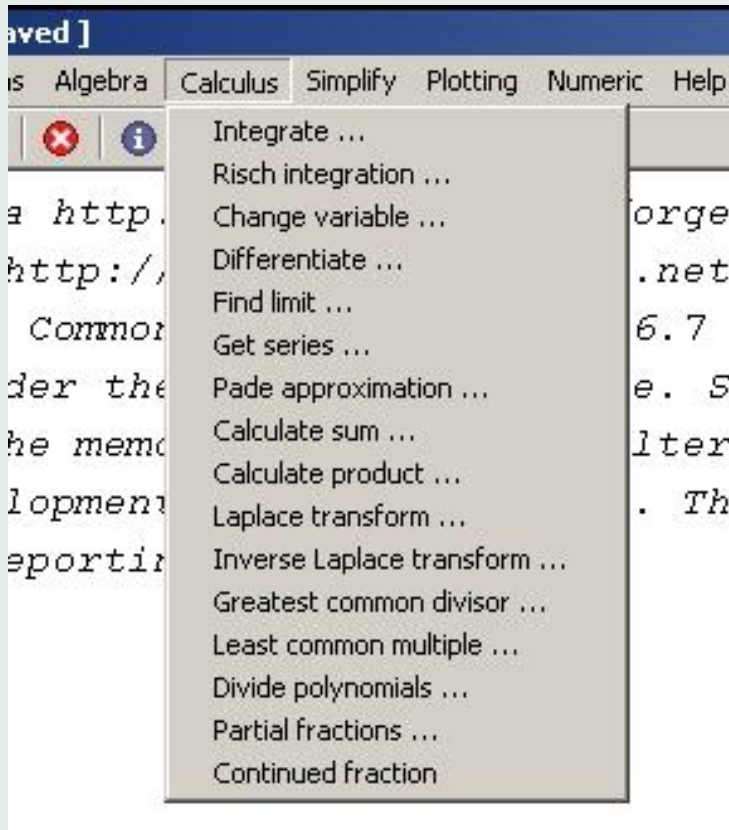
4. Riešenie úloh matematickej analýzy

V tejto kapitole ukážeme, ako sa pomocou SPA MAXIMA riešia štandardné úlohy matematickej analýzy, anglicky nazývané tiež calculus (u nás tiež diferenciálny a integrálny počet).

Funkcie, určené na riešenie týchto úloh, nájdeme v položke menu Calculus (pozri obrázok 9).

Tieto úlohy sa učia riešiť najmä študenti prvých ročníkov vysokých škôl (niektoré sa čiastočne preberajú aj na gymnáziách). Sú to:

- výpočet limít,
- derivovanie funkcií,
- výpočet Taylorovho polynómu funkcií,
- vyšetrowanie priebehu funkcií,
- integrowanie funkcií,
- vyšetrowanie extrémov funkcií viacerých premenných,
- posudzovanie konvergenie číselných radov,
- riešenie diferenciálnych rovníc,
- Laplaceova transformácia.



Obr. 9: Položka Calculus hlavného menu

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

[Strana 58 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



4.1. Výpočet limit

Príklad 1. Vypočítajte limitu

$$\lim_{x \rightarrow 1} \frac{x^2 - 1}{x - 1}.$$

Riešenie. Daná limita je typu $\left[\frac{0}{0}\right]$, preto rozložíme čitateľa na súčin koreňových činiteľov:

$$\lim_{x \rightarrow 1} \frac{x^2 - 1}{x - 1} = \lim_{x \rightarrow 1} \frac{(x - 1)(x + 1)}{x - 1} = \lim_{x \rightarrow 1} (x + 1) = 2.$$

V programe MAXIMA použijeme funkciu `limit` (funkcia, premenná, bod):

```
(%i1) limit((x^2-1)/(x-1), x, 1);
(%o1) 2
```

Príklad 2. Vypočítajte limitu

$$\lim_{x \rightarrow -1} \frac{x^2 + 1}{x + 1}.$$

Riešenie. Na výpočet použijeme funkciu `limit`:

```
(%i2) limit((x^2+1)/(x+1), x, -1);
(%o2) und
```



SPA MAXIMA nám oznámila, že táto limita neexistuje (skratka anglického *undefined*). O dôvode neexistencie limity sa presvedčíme výpočtom jednostranných limit (minus znamená „zo strany mínusu“, teda zľava):

```
(%i3) limit((x^2+1)/(x+1),x,-1,minus);
(%o3) -inf
(%i4) limit((x^2+1)/(x+1),x,-1,plus);
(%o4) inf
```

Teda limity zľava aj sprava sú nevlastné ($-\infty$ resp. $+\infty$) a rôzne.

Na záver tejto časti uveďme ešte jeden príklad na výpočet limity typu $[1^\infty]$.

Príklad 3. Vypočítajte limity

$$\lim_{n \rightarrow \infty} \left[\frac{n+2}{n+3} \right]^{n+4}$$

a

$$\lim_{n \rightarrow \infty} \left[\frac{1+2n}{1+3n} \right]^{1+4n}.$$

Riešenie.

```
(%i5) limit(((n+2)/(n+3))^(n+4),n,inf);
(%o5) %e^(-1)
(%i6) limit(((1+2*n)/(1+3*n))^(1+4*n),n,inf);
(%o6) 0
```

Poznámka 4.2. Apostrof na začiatku príkazu spôsobí, že sa príkaz nevykoná, len sa (v prostredí Wxmaxima) zobrazí:



```
(%i7) 'limit(((1+2*n)/(1+3*n))^(1+4*n),n,inf);
(%o7)  lim_{n -> inf} \left( \frac{2n+1}{3n+1} \right)^{4n+1}
```

Toto zobrazenie môžeme uložiť na spracovanie programom T_EX príkazom:

```
(%i8) tex(%)$
$$\lim_{n \rightarrow \infty} \left\{ \left( \frac{2n+1}{3n+1} \right)^{4n+1} \right\} \leq \text{tt}(\%o7)$$
```

Po kompilácii programom pdfT_EX získame:

```
(%o7)  lim_{n \rightarrow \infty} \left( \frac{2n+1}{3n+1} \right)^{4n+1}
```



4.2. Výpočet derivácií a parciálnych derivácií

Výpočet derivácii (obyčajných i parciálnych) sa využíva pri riešení viacerých úloh, napríklad pri určovaní extrémov funkcií. Pozrime si na úvod zopár príkladov.

Príklad 4. Vypočítajme deriváciu funkcie $\sqrt{\sin(x^2) + 1}$.

Riešenie. MAXIMA poskytuje na výpočet derivácií funkciu `diff`:¹⁹

```
(%i9) diff(sqrt(sin(x^2)+1),x);
```

```
(%o9) 
$$\frac{x \cos x^2}{\sqrt{\sin x^2 + 1}}$$

```

Príklad 5. Vypočítajme deriváciu funkcie $\ln(x + \sqrt{x^2 + a})$.

Riešenie. Zjednodušený výsledok získame až po niekoľkých úpravách.²⁰

¹⁹Výstupy programu MAXIMA budeme zobrazovať v \LaTeX u, čo sa podobá na výstup v prostredí `WXMAXIMA`.

²⁰Je možné, že by stačilo správne nastaviť niektoré systémové premenné.

```
(%i10) diff(log(x+sqrt(x^2+a)),x);
```

```
(%o10)
```

$$\frac{\frac{x}{\sqrt{x^2+a}} + 1}{\sqrt{x^2+a} + x}$$

```
(%i11) subst(b,x+sqrt(x^2+a),%);
```

```
(%o11)
```

$$\frac{\frac{x}{\sqrt{x^2+a}} + 1}{b}$$

```
(%i12) radcan(%);
```

```
(%o12)
```

$$\frac{\sqrt{x^2+a} + x}{b \sqrt{x^2+a}}$$

```
(%i13) subst(b,x+sqrt(x^2+a),%);
```

```
(%o13)
```

$$\frac{1}{\sqrt{x^2+a}}$$

Poznámka 4.3. Prirodzený logaritmus sa v programe MAXIMA označuje log, podobne ako v jazyku C alebo v MATLABe. Iné logaritmy nie sú k dispozícii.

Príklad 6. Vypočítajme 2. deriváciu funkcie $x \cdot \sin(x)$.

Riešenie. Porovnajme nasledujúce dva spôsoby:

```
(%i14) diff(diff(x*sin(x),x),x);  
(%o14) 2*cos(x)-x*sin(x)  
(%i15) diff(x*sin(x),x,2);  
(%o15) 2*cos(x)-x*sin(x)
```

Príklad 7. Uvažujme funkciu $f(z) = 1/z$ komplexnej premennej $z = x + iy$. Overme, že reálna zložka $u(x, y)$ funkcie f je harmonická funkcia, t. j. $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$.

```
(%i16) z:x+%i*y$  
(%i17) u:realpart(1/z);
```

```
(%o17) 
$$\frac{x}{y^2 + x^2}$$

```




```
(%i18) diff(u,x,2)+diff(u,y,2);
```

```
(%o18)
```

$$-\frac{8x}{(y^2+x^2)^2} + \frac{8xy^2}{(y^2+x^2)^3} + \frac{8x^3}{(y^2+x^2)^3}$$

```
(%i19) radcan(%);
```

```
(%o19) 0
```

Príklad 8. Vypočítajme zmiešanú parciálnu deriváciu $\frac{\partial^2 u}{\partial x \partial y}$ druhého rádu funkcie u .

```
(%i20) diff(diff(u,x),y);
```

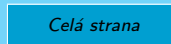
```
(%o20)
```

$$\frac{8x^2y}{(y^2+x^2)^3} - \frac{2y}{(y^2+x^2)^2}$$

```
(%i21) diff(u,x,1,y,1);
```

```
(%o21)
```

$$\frac{8x^2y}{(y^2+x^2)^3} - \frac{2y}{(y^2+x^2)^2}$$



4.3. Výpočet Taylorovho polynómu funkcií

Výpočet Taylorovho polynómu hladkej funkcie v okolí daného bodu je síce teoreticky priehľadný, ale aj dostatočne náročný, vzhľadom na nutnosť výpočtu derivácií vyšších rádov. Taylorov polynóm pritom poskytuje efektívny prostriedok na aproximáciu funkcie v okolí daného bodu (výpočtovo) jednoduchým polynómom, čo má význam, napríklad, pri riadení v reálnom čase.

Ako je známe, ak funkcia $f(x)$ má $n + 1$ spojité derivácie na intervale $\langle a, b \rangle$, potom jej Taylorov polynóm n -tého stupňa so stredom v bode $x_0 \in (a, b)$ je daný vzťahom

$$T_n(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k, \quad (1)$$

kde $f^{(k)}(x_0)$ je k -ta derivácia funkcie $f(x)$ v bode x_0 a $f^{(0)}(x) \stackrel{\text{def}}{=} f(x)$.

Otázke odhadu absolútnej hodnoty odchýlky Taylorovho polynómu od funkcie $f(x)$ sa v tejto učebnici nebudeme venovať.

Príklad 9. Vypočítajme Taylorov polynóm 5. stupňa so stredom v bode

$$x_0 = 0 \text{ funkcie } f(x) = \frac{\sin(x^2)}{x^2 + 2x + 3}.$$

Riešenie. Príkaz `taylor`, ktorý použijeme na riešenie daného príkladu, sa dá zadať rôznymi spôsobmi. Nebudeme však presne popisovať jeho syntax, veríme, že je jasná zo zadania úlohy a samotného príkazu.

```
(%i22) taylor(sin(x^2)/(x^2+2*x+3),x,0,5);
(%o22)      x^2      2 x^3      x^4      4 x^5
            3      - 9      + 27      - 81      + ...
```

Poznámka 4.4. Ak si predstavíme výpočet derivácie danej funkcie $f(x)$, je zrejmé, že vyššiu deriváciu ako 2. by sme „ručne“ len neradi počítali.

Príklad 10. Vypočítajme rozvoj polynómu $p(x) = x^5 - 4x^3 + 2x$ so stredom v bode $x_0 = -2$.

Riešenie.

```
(%i23) t:taylor(x^5-4*x^3+2*x,x,-2,7);
(%o23)
-4 + 34 (x + 2) - 56 (x + 2)^2 + 36 (x + 2)^3 - 10 (x + 2)^4 + (x + 2)^5 + ...
```

Bodky na konci výrazu t vyvolávajú dojem, že sa nejedná o polynóm.²¹ Na druhej strane príkaz `expand` svedčí o tom, že by to polynóm mohol byť:

```
(%i24) expand(t);
(%o24) x^5-4*x^3+2*x
```

Ku jednotlivým koeficientom polynómu t sa dostaneme použitím funkcie `coeff`. Napríklad:

²¹Ak chceme získať polynóm bez bodiek, môžeme po načítaní balíka `load(revert)` použiť príkaz `revert` dvakrát za sebou. Zmysel tohoto príkazu autorovi uniká, skúste sami šťastie pri jeho našťudovaní.

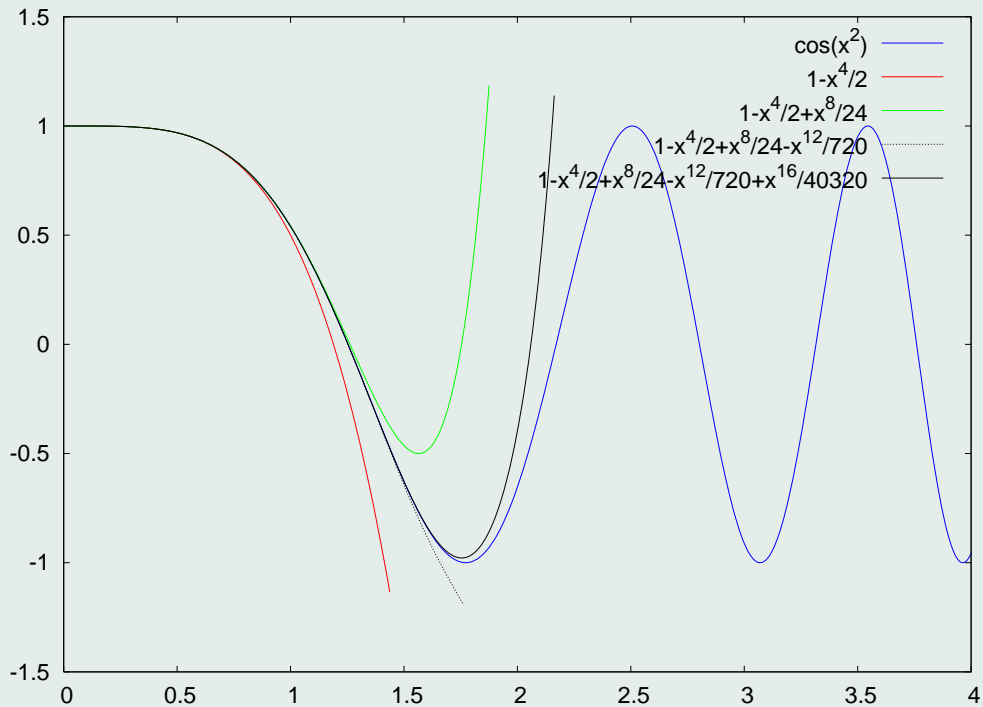
```
(%i25) coeff(t,x+2,2);  
(%o25) -56
```

Príklad 11. Porovnajme kvalitu aproximácie funkcie $\cos(x^2)$ na intervale $\langle 0, 4 \rangle$ pomocou Taylorových polynómov stupňov 4, 8, 12 a 16.

Riešenie. Po výpočte Taylorových polynómov na zobrazenie funkcií využijeme príkaz `plot2d`. Na to môžeme najprv použiť ponuku menu `Plotting`. Obrázok uložíme do súboru vo formáte Encapsulated PostScript (rozšírenie `eps`) a zaradíme ho do textu (výsledok vidíte na obrázku 10):

```
(%i26) t4:taylor(cos(x^2),x,0,4);  
(%o26) 1-x^4/2+...  
(%i27) t8:taylor(cos(x^2),x,0,8);  
(%o27) 1-x^4/2+x^8/24+...  
(%i28) t12:taylor(cos(x^2),x,0,12);  
(%o28) 1-x^4/2+x^8/24-x^12/720+...  
(%i29) t16:taylor(cos(x^2),x,0,16);  
(%o29) 1-x^4/2+x^8/24-x^12/720+x^16/40320+...  
(%i30) plot2d([cos(x^2),t4,t8,t12,t16], [x,0,4],  
[y,-1.2,1.2], [gnuplot_term, ps], [gnuplot_out_file,  
"D:/Users/Busa/Kega/Maxima/cosx2.eps"])$
```

Ako vidieť, polynóm stupňa 16 aproximuje dobre funkciu na intervale $\langle 0, 1.5 \rangle$.



Obr. 10: Taylorove polynómy funkcie $\cos(x^2)$ stupňov 4, 8, 12 a 16

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

◀ ▶

◀ ▶

Strana 69 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



4.4. Vyšetrowanie priebehu funkcie

Jednou z najdôležitejších úloh, ktorej riešenie by mali zvládať študenti vysokej školy, je vyšetrowanie priebehu funkcie. SPA môže zjednodušiť riešenie viaceré čiastočné úlohy.

Príklad 12. Vyšetrimo priebeh funkcie

$$f(x) = \frac{x^2 - x - 2}{x^2}.$$

Riešenie. Pri jednotlivých krokoch využijeme rôzne funkcie programu MAXIMA.

Funkciu najprv zdefinujeme:

```
(%i31) f(x) := (x^2 - x - 2) / x^2;
```

```
(%o31) 
$$f(x) := \frac{x^2 - x - 2}{x^2}$$

```

4.4.1. Definičný obor

Keďže funkcia má tvar zlomku, zistíme, kedy sa menovateľ, ktorý určíme príkazom denom (anglicky denominator), rovná nule:

```
(%i32) men:denom(f(x));  
(%o32) x^2  
(%i33) solve(men=0);  
(%o33) [x=0]
```

Vidíme, že bod $x = 0$ nepatrí do definičného oboru.

4.4.2. Nulové body

Priesečníky grafu funkcie so súradnicovými osami určíme, keď zistíme, kedy sa čitateľ rovná nule. Čitateľa určíme príkazom num (anglicky numerator).

```
(%i34) cit:num(f(x));  
(%o34) x^2-x-2  
(%i35) factor(cit);  
(%o35) (x-2)*(x+1)  
(%i36) solve(cit=0);  
(%o36) [x=2,x=-1]
```

Keďže čitateľ je polynóm, môžeme sa pokúsiť určiť jeho rozklad na súčin koreňových činiteľov. Ten však nie je možný pre každú kvadratickú rovnicu. Riešením rovnice $x^2 - x - 2 = 0$ dosiahneme to isté, ako o tom svedčí príkaz solve. Týmto spôsobom však úlohu vyriešime vždy.

4.4.3. Body nespojitosti a asymptoty bez smernice

Ako sme zistili vyššie, bodom nespojitosti je bod $x = 0$. Vyšetříme správanie sa funkcie v okolí bodu nespojitosti:

```
(%i37) limit(f(x),x,0);  
(%o37) -inf
```

V bode $x = 0$ existuje nevlastná (obojsstranná) limita rovná $-\infty$. Teda priamka $x = 0$ bude **asymptotou bez smernice**.

4.4.4. Asymptoty so smernicou v bodoch $\pm\infty$

Asymptoty so smernicou v tvare $y = k_{\pm}x + q_{\pm}$ určujeme pomocou vzorcov:

$$k_{\pm} = \lim_{x \rightarrow \pm\infty} \frac{f(x)}{x}, \quad q_{\pm} = \lim_{x \rightarrow \pm\infty} [f(x) - k_{\pm}x].$$

```
(%i38) km:limit(f(x)/x,x,minf);  
(%o38) 0  
(%i39) qm:limit(f(x)-km*x,x,minf);  
(%o39) 1  
(%i40) kp:limit(f(x)/x,x,inf);  
(%o40) 0  
(%i41) qp:limit(f(x)-kp*x,x,inf);  
(%o41) 1
```


Všimnite si, že bod $-\infty$ sa v limite zadáva skratkou `minf` (anglicky minus infinity). Teda funkcia má spoločnú **asymptotu so smernicou** (nulovou) $y = 1$.

4.4.5. Určovanie stacionárnych bodov

```
(%i42) g:=diff(f(x),x);
```

```
(%o42) 
$$\frac{2x - 1}{x^2} - \frac{2(x^2 - x - 2)}{x^3}$$

```

```
(%i43) factor(g);
```

```
(%o43) 
$$\frac{x + 4}{x^3}$$

```

Vidíme, že funkcia $f(x)$ má jediný stacionárny bod $x = -4$. Jeho charakter vyšetríme v ďalšej časti.

4.4.6. Určovanie inflexných bodov

Na určenie inflexných bodov je potrebná druhá derivácia funkcie. Ak je druhá derivácia v stacionárnom bode kladná, funkcia má v tomto bode minimum a ak je záporná, funkcia má maximum.

```
(%i44) h:diff(f(x),x,2);  
(%o44) 
$$\frac{6(x^2 - x - 2)}{x^4} - \frac{4(2x - 1)}{x^3} + \frac{2}{x^2}$$
  
(%i45) h:factor(h);  
(%o45) 
$$-\frac{2(x + 6)}{x^4}$$

```

Ako vidíme, funkcia má jediný inflexný bod $x = -6$.²² Vyšetříme charakter stacionárneho bodu $x = -4$:

```
(%i46) subst(-4,x,h);  
(%o46) -1/64  
(%i47) f(-4);  
(%o47) 9/8
```

Keďže druhá derivácia v stacionárnom bode je záporná, funkcia nadobúda v bode $x = -4$ lokálne maximálnu hodnotu $f(-4) = 9/8 > 1$. Maximálny bod grafu sa teda nachádza nad asymptotou so smernicou $y = 1$.

4.4.7. Zobrazenie grafu funkcie jednej premennej

Po predchádzajúcich výpočtoch je zrejmé, že graf funkcie stačí zobraziť na intervale $\langle -8, 4 \rangle$, na ktorom sa nachádzajú všetky nulové body, bod

²²Dokázať by sme to mohli pomocou 3. derivácie.

nespojivosti, lokálne maximum aj inflexný bod. Hodnoty y môžeme zvoliť, napríklad, z intervalu $\langle -4, 4 \rangle$, ktorý zrejme obsahuje všetky „zaujímavé“ funkčné hodnoty. Spolu s grafom funkcie $f(x)$ zobrazíme aj asymptotu so smernicou a osi. Predtým, ako graf uložíme do súboru, môžeme si ho pozrieť na obrazovke programu Gnuplot:

```
(%i48) plot2d([f(x),p(x)], [x,-8,4], [y,-4,4],
             [gnuplot_preamble, "set zeroaxis;"],
             [gnuplot_term, ps], [gnuplot_out_file,
             "D:/Users/Busa/Kega/Maxima/TeX/fx.eps"])$
```

Na obrázku 11 je znázornený graf funkcie $f(x)$.²³

4.5. Výpočet neurčitých a určitých integrálov

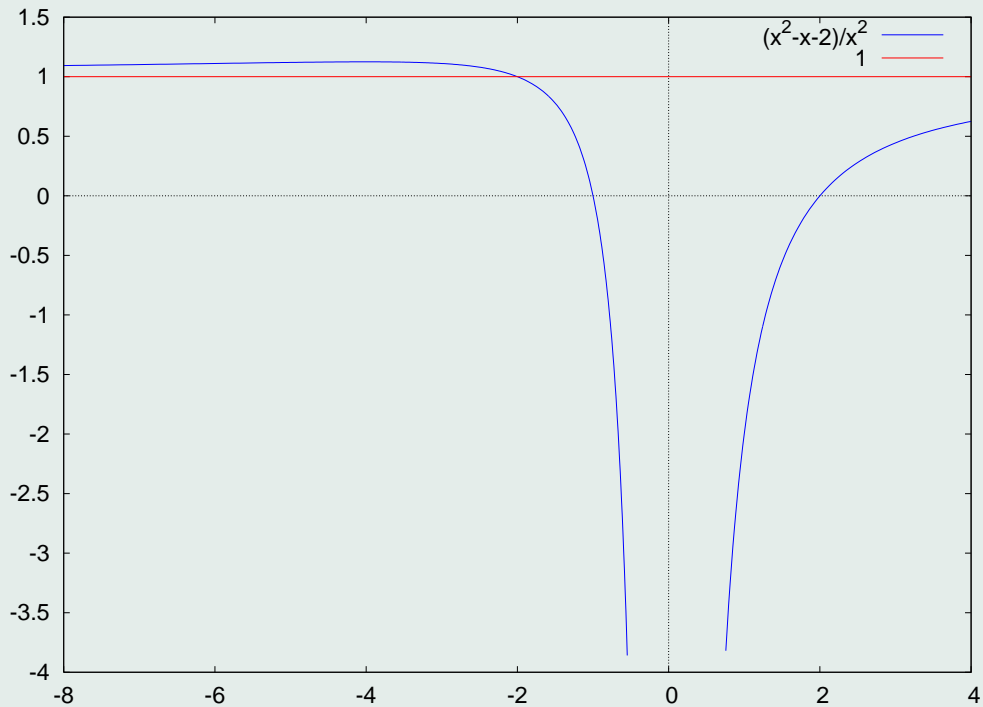
V tomto oddiele len uvedieme niekoľko príkladov výpočtu integrálov.²⁴

Príklad 13. Vypočítajme neurčitý integrál $\int x \cdot \ln(x^2 + 3x + 4) dx$.

Riešenie. Tento integrál už iste nepatrí medzi triviálne, aj keď je to štandardný príklad na použitie metódy „per partes – po častiach“ a následného integrovania racionálnej funkcie.

²³Z neznámeho dôvodu je maximálna hodnota na osi y rovná 1.5 a nie 4, ako sme zadali :(!

²⁴V prostredí WXMAXIMA môžeme integrál (ale aj ďalšie funkcie v ponuke menu) zadať príjemnejším spôsobom cez položku menu Calculus/Integrate.



Obr. 11: Graf funkcie $(x^3 - x - 2)/x^2$

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

◀◀ ▶▶

◀ ▶

Strana 76 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

```
(%i49) integrate(x*log(x^2+3*x+4), x);
```

$$(\%o49) \frac{x^2 \log(x^2 + 3x + 4)}{2} - \frac{\log(x^2 + 3x + 4)}{2} + \frac{21 \arctan\left(\frac{2x+3}{\sqrt{7}}\right)}{\sqrt{7}} + \frac{2x^2 - 6x}{2}$$

Poznámka 4.5. Vo výsledku aj v zadaní je namiesto funkcie ln funkcia log, o čom sme už písali vyššie. V našich krajoch tiež namiesto označenia arctan používame označenie arctg. Navyše vo výsledku chýba konštanta C, čo však nie je nejaká vážna chyba.

SPA MAXIMA je pozorný a ak mu niečo nie je jasné, môže Vás požiadať o doplňujúce informácie. Výsledkom môže byť, napríklad, nasledujúci „dialóg“:

```
(%i50) integrate(x^n, x);
Is n+1 zero or nonzero?
      zero;
(%o50) log(x)
(%i51) integrate(x^n, x);
Is n+1 zero or nonzero?
      nonzero;

(%o51) \frac{x^{n+1}}{n+1}
```

Poznámka 4.6. V tomto príklade sme naše odpovede umiestnili na samostatný riadok, na obrazovke prostredia Wxmaxima sú zobrazené hneď za

otázkou, ktorú vypíše MAXIMA. Odpoveď zadávame v príkazovom riadku.

Poznámka 4.7. Príkazom `assume(notequal(n, -1))` popísaným v oddiele 2.9 na strane 43 by sme v tomto prípade predišli zadaniu otázky, či je $n + 1$ nulové.

Príklad 14. Určme dĺžku oblúka krivky, ktorý tvorí časť grafu funkcie $\ln x$ pre x od 0,5 po 3.

Riešenie. Na výpočet dĺžky oblúka pre $x \in \langle a, b \rangle$ môžeme použiť známy vzorec:

$$L = \int_a^b \sqrt{1 + [f'(x)]^2} dx.$$

Pre funkciu $f(x) = \ln x$ a zadaný interval po úprave dostaneme:

$$L = \int_{0,5}^3 \frac{\sqrt{x^2 + 1}}{x} dx = \int_{\sqrt{5}/2}^{\sqrt{10}} \frac{t^2}{t^2 - 1} dt = \left[t + \frac{1}{2} \ln \left| \frac{t - 1}{t + 1} \right| \right]_{\sqrt{5}/2}^{\sqrt{10}}.$$

V programe MAXIMA riešime úlohu nasledujúcim spôsobom:

```
(%i52) f(x):=log(x)$
(%i53) integrate(sqrt(1+(diff(f(x),x))^2), x, 0.5, 3);
'rat' replaced 2.5 by 5//2 = 2.5
...
'rat' replaced 2.5 by 5//2 = 2.5
(%o53) 
$$-\frac{\log(\sqrt{10}+1)}{2} + \frac{\log(\sqrt{10}-1)}{2} + \sqrt{10} + 0.32560148642892$$

(%i54) float(%o31);
(%o54) 3.160428996360036
```

Ak porovnáme náš výsledok s výsledkom v riadku 53, vidíme, že sa prakticky zhodujú. Niektoré členy však SPA MAXIMA nahradil ich číselnými hodnotami. Po dosadení by sme sa však presvedčili, že číselné hodnoty obidvoch výsledkov sú zhodné.

O správnosti výsledku sa ešte presvedčíme numerickým integrovaním:²⁵

```
(%i55) romberg((sqrt(x^2+1))/x, x, 0.5, 3);
(%o55) 3.160429989460405
```

Príklad 15. Vypočítajte hodnotu $\int_0^2 e^{x^2} dx$.

²⁵K príkazu romberg sme sa dopracovali cez menu Calculus/Integrate.

Riešenie. Integrál sa nedá vypočítať pomocou elementárnych funkcií. Pozrime sa, ako si s touto situáciou poradí MAXIMA:

```
(%i56) integrate(exp(x^2), x, 0, 2);
```

```
(%o56) 
$$-\frac{\sqrt{\pi}i \operatorname{erf}(2i)}{2}$$

```

SPA MAXIMA použil špeciálnu funkciu erf – z anglického Error function. Vo výsledku sa objavila imaginárna jednotka, pretože funkcia chyby je definovaná vzťahom²⁶

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt.$$

Znamienko mínus v exponente funkcie teda získame po substitúcii $x = i \cdot t$ v našom integrále. Žiaľ, tento výsledok je nepoužiteľný, pretože MAXIMA nám neprezradí číselnú hodnotu svojho výsledku²⁷:

```
(%i57) float(%o56);
```

```
(%o57) -0.88622692545276*i*erf(2.0*i)
```

V tejto situácii nás zachráni numerické integrovanie:

```
(%i58) romberg(exp(x^2), x, 0, 2);
```

```
(%o58) 16.45262913575137
```

²⁶Ak sa o funkciu chyby chcete dozvedieť viac, otvorte si internetovskú stránku <http://mathworld.wolfram.com/Erf.html>.

²⁷Možno však existuje nejaká možnosť sa k tomuto číslu dopracovať :).

4.5.1. Substitúcia v integráloch

Niekedy je vhodné použiť určitú substitúciu na transformáciu integrálu²⁸ z jedného tvaru na druhý. Nato slúži funkcia

`changevar (<integrál>, <f(x,y)>, <y>, <x>),`

kde $f(x, y)$ je výraz určujúci substitúciu zadaný v explicitnom alebo v implicitnom tvare, y je nová premenná a x je pôvodná premenná. K príkazu `changevar` sme sa dopracovali cez menu Calculus/Change variable. Na potlačenie výpočtu integrálu použijeme apostrof ' pred príkazom `integrate`:

```
(%i59) changevar('integrate(sqrt(x^2+1)/x, x), t^2-x^2-1, t, x);
```

```
(%o59) 
$$\int \frac{t |t|}{t^2 - 1} dt$$

```

Poznámka 4.8. Keďže danému implicitnému tvaru vyhovuje viac substitúcií (napríklad $t = \pm\sqrt{x^2 + 1}$), vo výsledku sa správne objavila absolútna hodnota $|t|$. Absolútna hodnota sa vo výsledku neobjaví, ak predtým zadáme príkaz `assume(t>0)`.

Podobne v prípade určitého integrálu dostávame:

²⁸Alebo súčtu.

```
(%i60) changevar('integrate(x^3*sin(x^2), x, 2,3),t=x^2,t,x);
(%o60) integrate(t*sin(t),t,4,9)/2
```

alebo

```
(%o60) 
$$\frac{\int_4^9 t \sin t \, dt}{2}$$

```

v závislosti od nastaveného režimu zobrazovania výsledkov.

4.6. Výpočet extrémov funkcií viac premenných

Pri vyšetrowaní extrémov funkcií viac premenných sa využije celá škála rôznych funkcií systému MAXIMA. Stretávame sa tu s parciálnymi deriváciami funkcie viac premenných, s riešením **nelineárnych** sústav (algebraických) rovníc, ako aj s hlavnými minormi (subdeterminantmi) Hessovej matice druhých derivácií.

Príklad 16. Určme lokálne extrém y funkcie $f(x, y) = \ln x - y^2 - \frac{y}{x}$.

Riešenie. Začneme definíciou funkcie a výpočtom parciálnych derivácií funkcie podľa jednotlivých premenných:

```
(%i61) f(x,y):=log(x)-y^2-y/x$
```

```
(%i62) diff(f(x,y),x);
```

```
(%o62) 
$$\frac{y}{x^2} + \frac{1}{x}$$

```

```
(%i63) diff(f(x,y),y);
```

```
(%o63) 
$$-2y - \frac{1}{x}$$

```

Ďalej určíme **stacionárne body** riešením sústavy **nelineárnych** rovníc $\text{grad } f = \mathbf{0}$:

```
(%i64) stac:algsys([%o62, %o63], [x,y]);
```

```
(%o64) 
$$\left[ \left[ x = -\frac{1}{\sqrt{2}}, y = \frac{1}{\sqrt{2}} \right], \left[ x = \frac{1}{\sqrt{2}}, y = -\frac{1}{\sqrt{2}} \right] \right]$$

```

```
(%i65) xs:last(stac[2][1]);
```

```
(%o65) 1/sqrt(2)
```

```
(%i66) ys:last(stac[2][2]);
```

```
(%o66) -1/sqrt(2)
```

Z dvojice stacionárnych bodov, ktoré určil program MAXIMA, jeden nepatrí do definičného oboru funkcie $f(x, y)$, vzhľadom na podmienku $x > 0$.

Poznámka 4.9. Všimnite si príkaz last v riadkoch 65 a 66. Výsledkom riešenia sústavy je zoznam (anglicky list) stac, ktorý obsahuje rovnosti, ako sa o tom môžete presvedčiť pri pohľade na riadok (%o64). Premennú xs získame ako pravú stranu rovnosti prvku [2][1] tohoto zoznamu (anglicky last term – posledný člen).

Ďalej vypočítame Hessovu maticu²⁹ druhých parciálnych derivácií funkcie $f(x, y)$, určíme jej hlavné minory Δ_1, Δ_2 a rozhodneme o type stacionárneho bodu:

```
(%i67) H:matrix([diff(f(x,y),x,2),diff(f(x,y),x,1,y,1)],
                [diff(f(x,y),x,1,y,1),diff(f(x,y),y,2)]);
```

```
(%o67)      [ -2y/x^3 - 1/x^2   1/x^2 ]
             [ 1/x^2         -2     ]
```

```
(%i68) d1:subst(ys,y,subst(xs,x,H[1][1]));
```

```
(%o68) 2
```

```
(%i69) d:determinant(H);
```

```
(%o69)      -2 * ( -2y/x^3 - 1/x^2 ) - 1/x^4
```

```
(%i70) d2:subst(ys,y,subst(xs,x,d));
```

```
(%o70) -8
```

²⁹K príkazu matrix sme sa prepracovali cez položku menu Algebra/Enter Matrix.

Keďže $\Delta_2 < 0$, funkcia $f(x, y)$ nemá žiaden lokálny extrém.

Príklad 17. Určme lokálne extrémymy funkcie $f(x, y) = x^3 + y^3$ pri väzbe $x + y - 4 = 0$.

Riešenie. Úlohu budeme riešiť metódou Lagrangeovych multiplikátorov. Najprv zadefinujeme Lagrangeovu funkciu a určíme jej stacionárne body:

```
(%i71) f(x,y):=x^3+y^3$
(%i72) v(x,y):=x+y-4$
(%i73) l(x,y,lambda):=f(x,y)+lambda*v(x,y)$
(%i74) diff(l(x,y,lambda),x);
(%o74) lambda+3*x^2
(%i75) diff(l(x,y,lambda),y);
(%o75) lambda+3*y^2
(%i76) diff(l(x,y,lambda),lambda);
(%o76) y+x-4
(%i77) stac:algsys([%o74,%o75,%o76],[x,y,lambda]);
(%o77) [[x=2,y=2,lambda=-12]]
```

Lagrangeova funkcia má teda jediný stacionárny bod $(x^*, y^*, \lambda^*) = (2, 2, -12)$. Zostavíme podmaticu druhých derivácií Lagrangeovej funkcie podľa premenných x a y a určíme jej hlavné minory:



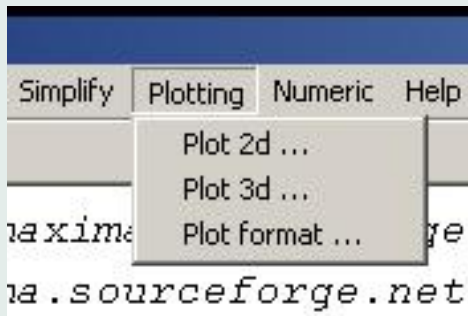
```
(%i78) H:matrix([diff(l(x,y,lambda),x,2),
                diff(l(x,y,lambda),x,1,y,1)],
                [diff(l(x,y,lambda),x,1,y,1),
                diff(l(x,y,lambda),y,2)]);
(%o78) matrix([6*x,0],[0,6*y])
(%i79) xs:last(stac[1][1])$
(%i80) ys:last(stac[1][2])$
(%i81) d1:subst(ys,y,subst(xs,x,H[1][1]));
(%o81) 12
(%i82) d2:subst(ys,y,subst(xs,x,determinant(H)));
(%o82) 144
(%i83) f(xs,ys);
(%o83) 16
```

Keďže $\Delta_2 = 144 > 0$, funkcia $f(x, y)$ má pri danej väzbe lokálny extrém. Vzhľadom na to, že $\Delta_1 = 12 > 0$, funkcia nadobúda na priamke $x + y - 4 = 0$ lokálne minimum v bode $(2, 2)$, pričom sa minimálna funkčná hodnota rovná 16.

4.6.1. Zobrazenie grafu funkcie dvoch premenných

V prostredí Wxmaxima sa nemusíme obávať vytvorenia grafu funkcie dvoch premenných. Môžeme znova využiť menu, tentoraz klikneme na položku Plotting, zobrazenú na obrázku 12.

Vyššie sme túto položku použili pri vytvorení grafu funkcie jednej pre-



Obr. 12: Položka menu Plotting

mennej. Teraz využijeme položku Plotting/Plot 3D a zadáme vstupné údaje. Ďalej už môžeme modifikovať údaje aj v príkazovom riadku.

Príklad 18. Zobrazme funkciu $f(x, y) = \ln x - y^2 - y/x$ v okolí stacionárneho bodu $(x^*, y^*) = (1/\sqrt{2}, -1/\sqrt{2})$.

Riešenie. Po zadaní funkcie pomocou menu sme získali zobrazenie grafu. Všimli sme si, že tento graf je možné rotovať a našli sme vhodný pohľad. Ak si uvedomíme, že MAXIMA pri vytváraní grafov spolupracuje s programom Gnuplot, je jasné, že nastal čas nazrieť do učebnice Gnuplotu (DOBOŠ, 2006). Na strane 35 objavíme príklad použitia príkazu view.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Strana 87 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

```
(%i84) plot3d(log(x)-x^2-y/x, [x,0.6,0.8], [y,-0.8,-0.6],  
  [gnuplot_preamble, "set view 37,26"], [gnuplot_term, ps],  
  [gnuplot_out_file, "D:/Users/Busa/Kega/Maxima/logxyn.eps"])$
```

Na obrázku 13 sa môžeme uistiť, že stacionárny bod $(1/\sqrt{2}, -1/\sqrt{2}) \approx (0,707; 0,707)$ nie je bodom lokálneho extrémumu ale môže byť len sedlovým bodom.

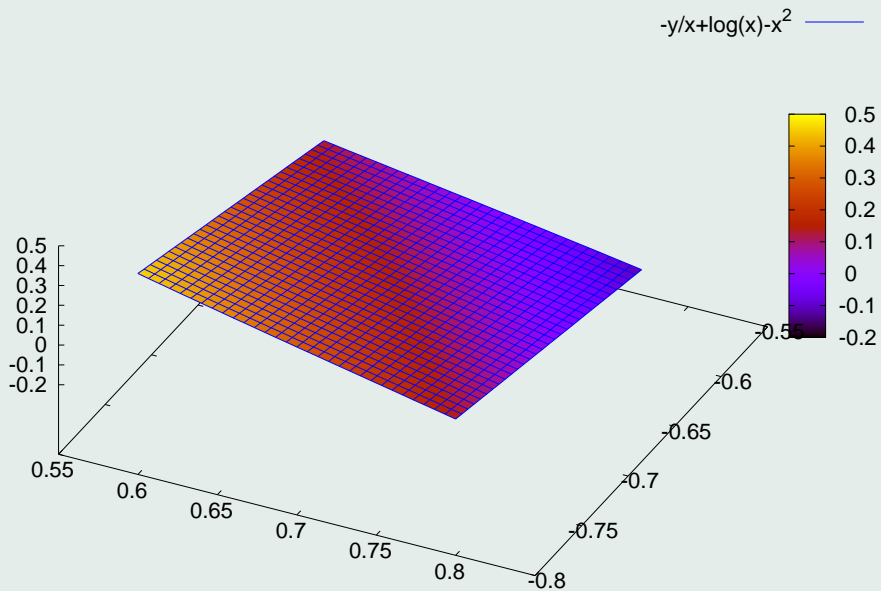
Príklad 19. Znázorníme funkciu $f(x, y) = x^3 + y^3$ v okolí lokálneho minima pri väzbe $x + y - 4$.

Riešenie. Postupujeme podobne ako pri riešení predchádzajúceho príkladu:

```
(%i85) plot3d(x^3+y^3, [x,0,4], [y,0,4],  
  [gnuplot_preamble,"set pm3d at s;unset surf;unset colorbox"],  
  [gnuplot_term, ps],  
  [gnuplot_out_file, "D:/Users/Busa/Kega/Maxima/x3py3.eps"])$
```

Väzba je rovnica priamky $x + y - 4 = 0$, ktorá v rovine \mathcal{O}_{xy} spája body $(0, 4)$ a $(4, 0)$. Ako je vidieť na obrázku 14, nad touto priamkou nadobúda funkcia $f(x, y)$ najmenšiu hodnotu nad bodom $(2, 2)$. Ak porovnáme obrázok 14 s obrázkom 13, uvidíme, že chýba farebná škála funkčných hodnôt. Toto sme dosiahli pridaním príkazu `unset colorbox`.

Ešte lepší pohľad získame, ak zobrazíme vrstevnice funkcie $f(x, y)$.



Obr. 13: Graf funkcie $f(x, y) = \ln x - y^2 - y/x$

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

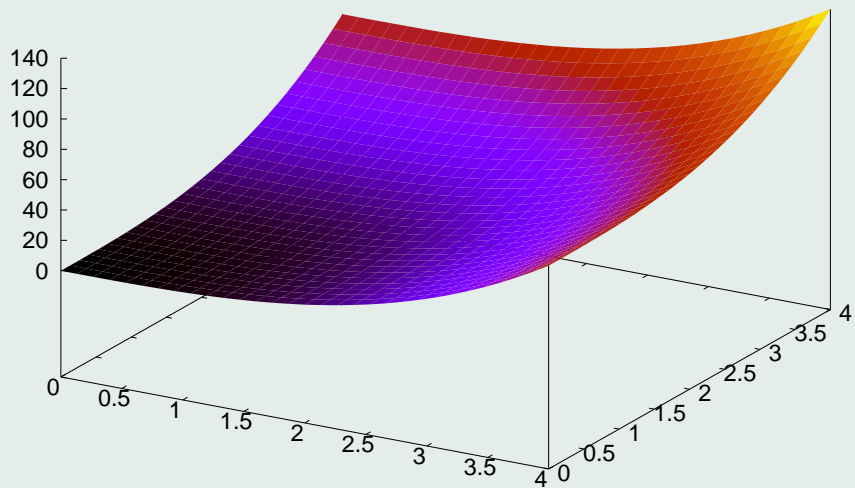
Strana 89 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



Obr. 14: Graf funkcie $f(x, y) = x^3 + y^3$

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

[Strana 90 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

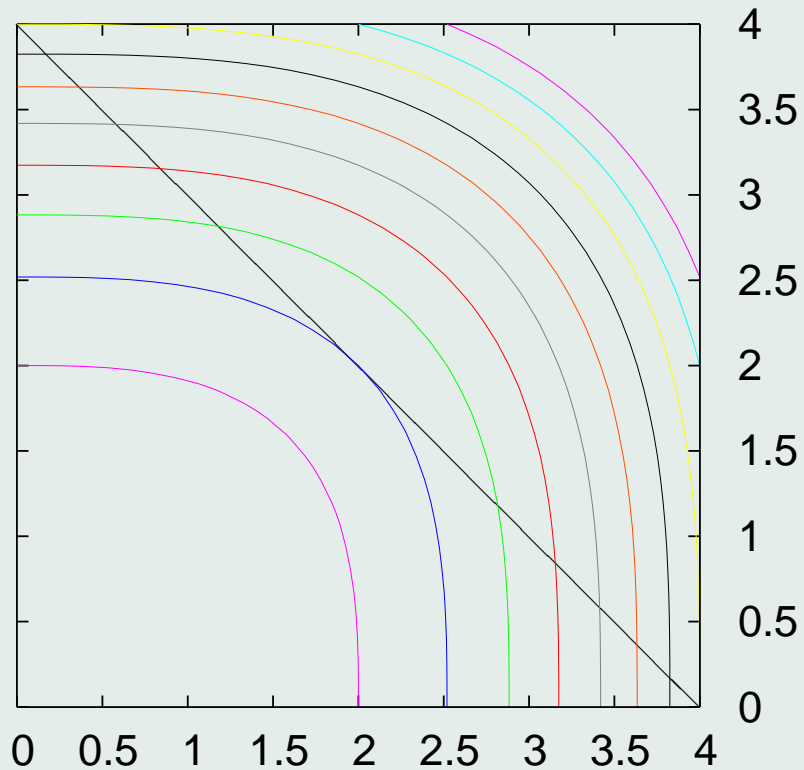
```
(%i86) plot3d(x^3+y^3, [x,0,4], [y,0,4],  
  [gnuplot_preamble, "unset pm3d; set contour base;  
  set cntrparam levels incremental 0,8,80; set nokey;  
  set size square; unset surf; set view 0,0],  
  [gnuplot_term, ps], [gnuplot_out_file,  
  "D:/Users/Busa/Kega/Maxima/vrstevnice.eps"])$
```

V tomto príklade sme vypli farby voľbou `unset pm3d`, nastavili výstup grafu na typ izolínií voľbou `set contour base`, nastavili sme hodnoty vrstevníc, ktoré sme zobrazili – `set cntrparam levels incremental 0,8,80`, vypli sme legendu príkazom `set nokey`, nastavili sme pomer jednotiek osí x a y na 1:1 príkazom `set size square`, apod.³⁰ Časť priamky $x + y - 4 = 0$ (uhlopriečku štvorca) sme dokreslili samostatne. Výsledok vidíte na obrázku 15. Minimálna hodnota pri väzbe sa dosahuje tam, kde sa priamka daná väzbou dotýka druhej izolínie, ktorej odpovedá hodnota $f = 16$.

4.7. Vyšetrovanie konvergenzie číselných a mocninových radov

SPA MAXIMA dokáže vypočítať presný súčet niektorých radov. Napríklad, ak jej zadáme známy **harmonický** rad $\sum_{n=1}^{\infty} \frac{1}{n}$, nezaváha a odpovie:

³⁰Popri učebnici (DOBOŠ, 2006) sme čerpali z podrobnejšej príručky (KAWANO, 2005).



Obr. 15: Izolínie funkcie $f(x, y) = x^3 + y^3$

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

[Strana 92 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

```
(%i87) sum(1/n, n, 1, inf), simpsum;
(%o87) inf
```

Teda súčet radu je nekonečný, čo znamená divergenciu.

Podobne dopadne výpočet súčtu radu $\sum_{n=1}^{\infty} \frac{1}{n^2}$:

```
(%i88) sum(1/n^2, n, 1, inf), simpsum;
(%o88)  $\frac{\pi^2}{6}$ 
```

Teda aj v tomto prípade sme získali presný súčet.

Príklad 20. Vyšetrite konvergenciu radu $\sum_{n=1}^{\infty} \frac{1}{\sqrt{n+1}}$.

Riešenie. Ak požiadame o výpočet tohoto radu, MAXIMA nám ako výsledok vráti len jeho zápis:

```
(%i89) sum(1/sqrt(n+1), n, 1, inf), simpsum;
(%o89)  $\sum_{n=1}^{\infty} \frac{1}{\sqrt{n+1}}$ 
```

Skúsime preto použiť D'Alembertovo kritérium:

```
(%i90) a(x) := 1/sqrt(x+1)$
(%i91) limit(a(n+1)/a(n), n, inf);
(%o91) 1
```

Dostali sme práve výsledok, ktorý nám neumožňuje urobiť záver o konvergencii. Preto vyskúšame Cauchyho integrálne kritérium:

```
(%i92) integrate(a(x),x,1,inf);
Integral is divergent
-- an error. Quitting. To debug this try debugmode(true);
```

Keďže integrál diverguje, diverguje aj samotný rad.

Príklad 21. Overme splnenie nutnej podmienky konvergenie pre rad

$$\sum_{n=1}^{\infty} \frac{(-1)^n \cdot n}{\sqrt{n^2 + 1}}.$$

Riešenie. Vypočítame $\lim_{n \rightarrow \infty} a_n$:

```
(%i93) a(n):=(-1)^n*n/sqrt(n^2+1)$
(%i94) limit(a(n),n,inf);
(%o94) ind
(%i95) limit(abs(a(n)),n,inf);
(%o95) -ind
(%i96) b(n):=n/sqrt(n^2+1)$
(%i97) limit(b(n),n,inf);
(%o97) 1
```

V predošlom sme sa už stretli s výsledkom und – anglicky undefined – nedefinovaný. Tentoraz je výsledok v riadku 94 ind. Žeby preklep? Nie, v knihe (MAXIMA MANUAL, 2005) na strane 167 nachádzame odpoveď:



ind – undefined but bounded – nedefinované ale ohraničené. Vidíme, že sa nás MAXIMA snaží čo najpresnejšie informovať o vzniknutej situácii. Prečo však je aj odpoveď v riadku 95 podobná? Vari MAXIMA nevie, že $|(-1)^n| = 1$? Po hlbšej úvahe prichádzame na to, že sme program MAXIMA neupozornili na to, že predpokladáme, že n je prirodzené.³¹ A v prípade kladného reálneho n , nemusí byť hodnota $(-1)^n$ vôbec definovaná! Tento problém prekonáme tým, že zadáme novú kladnú postupnosť (b_n) už bez člena $(-1)^n$. Limita tejto postupnosti je rovná 1 (riadok 97). Teda nutná podmienka konvergencie nie je splnená a rad nekonverguje.

Podobne môžeme použiť SPA Maxima na určovanie polomeru konvergencie mocninových radov v tvare $\sum_{n=0}^{\infty} a_n \cdot (x - x_0)^n$, keď je potrebné použiť Cauchyho-Hadamardov vzorec:

$$R = \lim_{n \rightarrow \infty} \left| \frac{a_n}{a_{n+1}} \right|$$

alebo

$$R = \left[\lim_{n \rightarrow \infty} \sqrt[n]{|a_n|} \right]^{-1}.$$

³¹Nie je jasné, či to MAXIMA umožňuje.



4.8. Riešenie obyčajných diferenciálnych rovníc

MAXIMA poskytuje niekoľko funkcií na riešenie diferenciálnych rovníc podobne ako, napríklad, Maple (pozri knihy (ĎJAKONOV, 2003; KREYSZIG a NORMINTON, 2006)). V tomto oddiele budeme využívať najmä príručku (SOUZA, FATEMAN, MOSES a YAPP, 2004).

Základ výbavy na riešenie diferenciálnych rovníc tvoria dve funkcie:

`ode2` – rieši obyčajné diferenciálne rovnice 1. a 2. rádu;

`desolve` – rieši systémy obyčajných lineárnych diferenciálnych rovníc s konštantnými koeficientmi, na riešenie MAXIMA využíva Laplaceovu transformáciu.

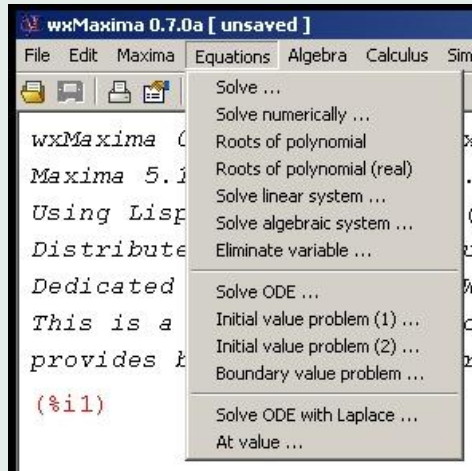
V prostredí Wxmaxima sa ku funkciám na riešenie diferenciálnych rovníc dostaneme cez položky menu Equations/Solve ODE a pod. (pozri obrázok 16).

4.8.1. Riešenie začiatočných úloh pre diferenciálne rovnice 1. rádu

MAXIMA dokáže riešiť nasledujúce typy diferenciálnych rovníc 1. rádu, ktoré sa obyčajne vyučujú na univerzitách:³²

- separovateľné,

³²Žiaľ, namiesto by tu bol skôr minulý čas, keďže v bakalárskom stupni výučby už na takú rozkoš nie je priestor.



Obr. 16: Ponuka menu Equations

- homogénne,
- lineárne,
- Bernoulliho,
- exaktné,
- zovšeobecnené homogénne.

Na zápis diferenciálnej rovnice, napríklad:

$$x^2 y' + 3xy = \frac{\sin x}{x}, \quad (2)$$

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



[Strana 97 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

existujú tri rôzne spôsoby, uvedené v knihe (SOUZA, FATEMAN, MOSES a YAPP, 2004). My uvedieme len dva z nich, ktoré autori odporúčajú používať pre funkcie ode2 a desolve.

Prvý z nich je vhodný na používanie vo funkcii ode2:

```
(%i98) depends(y,x);
(%o98) [y(x)]
(%i99) x^2*diff(y,x)+3*x*y=sin(x)/x;

(%o99)          x^2 ( d/dx y ) + 3 x y = sin x / x
```

Druhý spôsob je odporúčaný na použitie s funkciou desolve:

```
(%i100) x^2*diff(y(x),x)+3*x*y=sin(x)/x;

(%o100)          3 x y + x^2 ( d/dx y(x) ) = sin x / x
```

Predpokladajme, že sme na zadanie rovnice využili riadky 98–99. Môžeme pristúpiť k riešeniu pomocou funkcie ode2:³³

```
(%i101) ode2(%o99,y,x);

(%o101)          y = %c - cos x / x^3
```

³³Dávajte pozor na použitie (%oXX) a nie (%iXX)! Ak však použijete ako argument funkcie ode2 priamo vstup riadku (%i99), všetko bude v poriadku.

Vidíme, že vo výsledku sa objavila **konštanta** %c.

MAXIMA nás môže informovať o type rovnice, ktorú sme zadali:

```
(%i102) method;
(%o102) linear
```

Zadaná rovnica (2) je teda lineárna.

Poznámka 4.10. Ak MAXIMA nedokáže vyriešiť zadanú rovnicu, ako výsledok dostaneme false.

Ak chceme získať jedno partikulárne riešenie, môžeme zvoliť hodnotu konštanty %c. Treba si však uvedomiť, že výsledok je rovnosť a ak chceme získať funkciu, musíme použiť len pavú stranu:

```
(%i103) depends(g,x)$
(%i104) g:subst(1,%c,last(%o101));

(%o104) 
$$\frac{1 - \cos x}{x^3}$$

```

Teraz môžeme overiť správnosť riešenia:

```
(%i105) ratsimp(x^2*diff(g,x)+3*x*g);

(%o105) 
$$\frac{\sin x}{x}$$

```

Pomocou funkcie ic1 dokážeme riešiť **Cauchyho začiatočnú úlohu pre diferenciálnu rovnicu 1. rádu.**

Príklad 22. Určme riešenie diferenciálnej rovnice

$$y' + x y = x y^2, \quad (3)$$

vyhovujúce začiatočnej podmienke $y(1) = 2$.

Riešenie. Rovnica (3) je zapísaná v tvare Bernoulliho rovnice. Pozrime sa, ako ju vidí MAXIMA:

```
(%i106) depends(y,x)$
(%i107) rov:diff(y,x)+x*y=x*y^2;
(%o107) 'diff(y,x,1)+x*y=x*y^2
(%i108) ries:ode2(rov,y,x);
(%o108) log(y-1)-log(y)=x^2/2+%c
(%i109) method;
(%o109) separable
```

Rovnica (3) je teda separovateľná (ak na pravú stranu presunieme všetky členy okrem derivácie y , dá sa x vybrať pred zátvorku). Riešenie je zapísané v implicitnom tvare.

Pokračujme v riešení začiatočnej úlohy.

```
(%i110) riesz:ic1(ries,x=1,y=2);
```

$$(\%o110) \quad \log(y-1) - \log y = \frac{x^2 - 2 \log 2 - 1}{2}$$

Trochu SPA MAXIMA pomôžeme. Oddelíme ľavú a pravú stranu implicitného zadania riešenia začiatočnej úlohy a samostatne ich exponujeme:

```
(%i111) l:first(riesz)$
(%i112) r:last(riesz)$
(%i113) l:radcan(exp(l));
(%o113) (y-1)/y
(%i114) r:radcan(exp(r));
```

```
(%o114) 
$$\frac{e^{\frac{x^2-1}{2}}}{2}$$

```

Zbavili sme sa logaritmov a môžeme úlohu doriešiť:³⁴

```
(%i115) f:last(solve(l=r,y)[1]);
```

```
(%o115) 
$$-\frac{2}{e^{\frac{x^2}{2}-\frac{1}{2}}-2}$$

```

Na záver ešte overme správnosť riešenia:

³⁴Všimnite si, čo sa všetko vykonalo v riadku 115!



```
(%i116) diff(f,x)+x*f-x*f^2;
```

```
(%o116)
```

$$\frac{2 x e^{\frac{x^2}{2}-\frac{1}{2}}}{\left(e^{\frac{x^2}{2}-\frac{1}{2}}-2\right)^2}-\frac{2 x}{e^{\frac{x^2}{2}-\frac{1}{2}}-2}-\frac{4 x}{\left(e^{\frac{x^2}{2}-\frac{1}{2}}-2\right)^2}$$

```
(%i117) radcan(%o116);
```

```
(%o117) 0
```

Uved'me ešte zopár príkladov na ostatné typy diferenciálnych rovníc 1. rádu.

Príklad 23. Určme typ a riešenie diferenciálnej rovnice

$$y' = \sqrt{1 + \frac{y}{x}}. \quad (4)$$

Riešenie. Bez komentára vypíšme:

```
(%i118) diff(y,x)=sqrt(1+y/x);
```

```
(%o118)
```

$$\frac{d}{dx} y = \sqrt{\frac{y}{x} + 1}$$

```
(%i119) ode2(%,y,x);
```

```
(%119)
```

$$c x = e^{-\frac{\sqrt{5} \log\left(-x \sqrt{\frac{y+x}{x}} - y\right) + \log\left(\frac{2 \sqrt{\frac{y+x}{x}} - \sqrt{5} - 1}{2 \sqrt{\frac{y+x}{x}} + \sqrt{5} - 1}\right)}{\sqrt{5}}}$$

```
(%i120) method;
```

```
(%o120) homogeneous
```

Rovnica (4) je teda homogénna.³⁵

Príklad 24. Určme typ a riešenie diferenciálnej rovnice

$$y' = x y + x^3. \quad (5)$$

³⁵V riadku 119 sme v príkaze ode2 namiesto %o118 použili jednoducho %, čo znamená výstup z predchádzajúceho riadku.

```
(%i121) ode2(diff(y,x)-x*y=x^3,y,x);
(%o121)      y = \left( \frac{(-2x^2 - 4) e^{-\frac{x^2}{2}}}{2} + \%c \right) e^{\frac{x^2}{2}}
```

```
(%i122) method;
(%o122) linear
```

Príklad 25. Určme typ a riešenie diferenciálnej rovnice

$$x^2 y' \cos(x y) + \sin(x y) + x y \cos(x y) = 0. \tag{6}$$

```
(%i123) ode2((x^2*diff(y,x)+x*y)*cos(x*y)+sin(x*y)=0,y,x);
(%o123) x*sin(x*y)=%c
(%i124) method;
(%o124) exact
```

Poznámka 4.11. Diferenciálna rovnica 1. rádu sa nazýva **exaktná**, ak sa dá napísať v tvare $p(x, y) y' + q(x, y) = 0$, kde $p(x, y) = \partial M(x, y) / \partial y$ a $q(x, y) = \partial M(x, y) / \partial x$ pre nejakú funkciu $M(x, y)$. Niekedy sa dá získať exaktná rovnica z neexaktnej pre násobením tzv. integračným faktorom (SOUZA, FATEMAN, MOSES a YAPP, 2004).



Príklad 26. Určme typ a riešenie diferenciálnej rovnice^a

$$y' + \frac{2y}{x} = \frac{y^3}{x^2}. \quad (7)$$

^aPríklad je prevzatý z knihy (SOUZA, FATEMAN, MOSES a YAPP, 2004).

```
(%i125) ode2(diff(y,x)+2*y/x=y^3/x^2,y,x);
```

```
(%o125)
```

$$y = \frac{1}{\sqrt{\frac{2}{5}x^5 + \%c x^2}}$$

```
(%i126) method;
```

```
(%o126) bernoulli
```

Príklad 27. Určme typ a riešenie diferenciálnej rovnice

$$x y' = y \sqrt{1 + y x^4}. \quad (8)$$

```
(%i127) ode2(x*diff(y,x)=y*sqrt(1+y*x^4),y,x);
```

```
(%o127)      x = %c e- $\frac{8 \log(\sqrt{x^4 y+1+4}) - 5 \log(\sqrt{x^4 y+1+1}) - 3 \log(\sqrt{x^4 y+1-1})}{15}$ 
```

```
(%i128) method;
```

```
(%o128) genhom
```

Poznámka 4.12. V knihe The MAXIMA Book (SOUZA, FATEMAN, MOSES a YAPP, 2004) sa uvádza, že rovnica sa nazýva **zovšeobecnená homogénna** (anglicky general homogeneous), ak sa dá napísať v tvare

$$y' = \frac{y}{x} G(y x^n).$$

Riešenie takejto rovnice sa dá zapísať v implicitnom tvare pomocou primitívnej funkcie k funkcii $1/(x(n + G(x)))$, ktorá sa však nemusí dať vyjadriť pomocou elementárnych funkcií.

4.8.2. Riešenie začiatočných a okrajových úloh pre diferenciálne rovnice 2. rádu

MAXIMA rieši nasledujúce typy obyčajných diferenciálnych rovníc 2. rádu:

- lineárne s konštantnými a s nekonštantnými koeficientmi,
- exaktné,

- Eulerove,
- Besselove,
- rovnice bez závislej premennej,
- rovnice bez nezávislej premennej.

Podobne, ako v prípade rovníc 1. rádu, je možné riešiť **Cauchyho začiatočnú úlohu** ale v prípade rovníc 2. rádu je navyše možné riešiť aj **okrajovú úlohu**.

Príklad 28. Určme riešenie diferenciálnej rovnice

$$x^2 y'' + x y' - y = 0, \quad (9)$$

vyhovujúce začiatočným podmienkam $y(1) = 2, y'(1) = -3$.

Riešenie. Vidíme, že túto rovnicu môžeme zaradiť medzi Eulerove rovnice bez pravej strany tvaru

$$x^2 y'' + a x y' + b y = 0.$$

Postupujeme analogicky s prípadom riešenia rovnice 1. rádu. Rozdielny je typ začiatočnej podmienky. Začiatočná úloha sa rieši pomocou funkcie ic_2 :

```
(%i129) ode2(x^2*diff(y,x,2)+x*diff(y,x)-y=0,y,x);
(%o129) y=%k2*x-%k1/(2*x)
(%i130) method;
(%o130) exact
(%i131) ic2(%o129,x=1,y=2,diff(y,x)=-3);
```

$$(\%o131) \quad y = \frac{5}{2x} - \frac{x}{2}$$

SPA MAXIMA teda rovnicu zaradil medzi exaktné.

Príklad 29. Určme riešenie diferenciálnej rovnice (9) vyhovujúce okrajovým podmienkam $y(1) = 2$ a $y(2) = 1$.

Riešenie. V tomto prípade použijeme funkciu bc2:

```
(%i132) bc2(%o129,x=1,y=2,x=2,y=1);
```

$$(\%o132) \quad y = \frac{2}{x}$$

Príklad 30. Riešme Besselovu diferenciálnu rovnicu $x^2 y'' + x y' + (x^2 - 4)y = 0$.

Poznámka 4.13. Besselove rovnice majú tvar $x^2 y'' + x y' + (x^2 - n^2)y = 0$, kde n je konštanta. Príklad sme prevzali z knihy (SOUZA, FATEMAN, MOSES a YAPP, 2004).

```
(%i133) ode2(x^2*diff(y,x,2)+x*diff(y,x)+(x^2-4)*y-0,y,x);  
(%o133) y=bessel_y(2,x)*%k2+bessel_j(2,x)*%k1  
  
alebo  
  
(%o133) y = Y2(x)%k2 + J2(x)%k1
```

V zápise výsledku sa vyskytujú Besselove funkcie prvého resp. druhého druhu – J_n resp. Y_n .

Príklad 31. Určme riešenie diferenciálnej rovnice

$$y y'' + (y')^2 = 0 \tag{10}$$

vyhovujúce **začiatocným podmienkam** $y(1) = 2$ a $y'(1) = 3$.

Riešenie. Použijeme funkcie ode2 a ic2:³⁶

³⁶Aj túto rovnicu (bez doplňujúcich podmienok) sme prevzali z učebnice (SOUZA, FATEMAN, MOSES a YAPP, 2004).

```
(%i134) ode2(y*diff(y,x,2)+diff(y,x)^2=0,y,x);
(%o134) y^2/(2*k1)=x+k2
(%i135) method;
(%o135) freeofx
(%i136) ic2(%o134,x=1,y=2,diff(y,x)=3);

(%o136) 
$$\frac{y}{6} = x - \frac{3y-2}{3y}$$


(%i137) solve(%o136,y);

(%o137) 
$$\left[ y = -\sqrt{9x^2 - 18x + 13} + 3x - 3, y = \sqrt{9x^2 - 18x + 13} + 3x - 3 \right]$$

```

SPA MAXIMA určil, že sa jedná o rovnicu bez nezávislej premennej x . Ak porovnáme všeobecné riešenie v riadku (%i134) s riešením z riadku (%o137), vidíme, že sú nekonzistentné, nie je jasné, odkiaľ sa mohol nabrat' člen x^2 pod odmocninou. Ak preveríme splnenie začiatočných podmienok z rovnice (%o136), zistíme, že sú splnené. Ale aj rovnica (%o136) sa líši od rovnice (%o134) (ak by sme poslednú vydelili premennou y , pri premennej x by mal byť výraz $1/y$).

Takže podozrivá je rovnica (%o136). Preto určíme potrebné konštanty %k1 a %k2 „ručne“ pomocou programu MAXIMA:

```
(%i138) subst(2,y,subst(1,x,%o134));  
(%o138) 2/%k1=%k2+1  
(%i139) diff(%o134,x);
```

(%139)
$$\frac{y \left(\frac{d}{dx} y \right)}{\%k_1} = 1$$

```
(%i140) subst(2,y,subst(3,diff(y,x),%o139));  
(%o140) 6/%k1=1  
(%i141) algsys([%o138,%o140], [%k1,%k2]);  
(%o141) [[%k1=6,%k2=-2/3]]  
(%i142) subst(%o141,%o134);
```

(%o142)
$$\frac{y^2}{12} = x - \frac{2}{3}$$

```
(%i143) solve(%o142,y);  
(%o143) [y=-2*sqrt(3*x-2),y=2*sqrt(3*x-2)]  
(%i144) subst(9,x,%o143);  
(%o144) [y=-10,y=10]
```

Do rovnice (%134) sme dosadili začiatočnú podmienku. Potom sme rovnicu (%134) zderivovali a do tejto rovnice sme dosadili hodnoty zo začiatočných podmienok (riadky 139–140). Získanú sústavu nelineárnych rovníc sme zadali do funkcie algsys a získali sme konštanty k_1 a k_2 . Po ich dosadení

do rovnice (134) a jej opätovnom riešení vzhľadom na neznámu funkciu y sme získali riešenie (z dvoch riešení v riadku 143 len pravé vyhovuje začiatočnej podmienke). Nakoniec sme určili funkčnú hodnotu riešenia v bode $x = 9$.

Príklad 32. Určme riešenie diferenciálnej rovnice (10) vyhovujúce **okrajovým podmienkam** $y(1) = 2$ a $y(9) = 10$.

```
(i145) bc2(o134, x=1, y=2, x=9, y=10) ;  
(o145) y^2/12=x-2/3
```

Dostali sme rovnaké riešenie ako v riadku 142.

Poznámka 4.14. Príklady, ktoré sme teraz doriešili, nás vedú k záveru, že ani taký zaujímavý systém počítačovej algebry akým MAXIMA nesporne je, nie je bez chýb. Ako v mnohých ďalších oblastiach ľudskej činnosti, aj tu platí známe: „Dvakrát meraj a až potom rež!“ – je dôležité naučiť sa overovať aj výsledky, ktoré nám prezentuje program. Najhoršie je, že program MAXIMA nás neupozornil na žiaden problém!

Na záver vyriešme nejaký typický školský príklad diferenciálnej rovnice s konštantnými koeficientmi.



Príklad 33. Určme riešenia diferenciálnej rovnice

$$y'' + y = \frac{1}{\cos x}$$

vyhovujúce: a) začiatočným podmienkam $y_1(0) = -1$ a $y_1'(0) = 1$;
b) okrajovým podmienkam $y_2(0) = -1$ a $y_2(\pi/4) = 0$.

```
(%i146) ode2(diff(y,x,2)+y=1/cos(x),y,x);
(%o146) y=cos(x)*log(cos(x))+x*sin(x)+%k1*sin(x)+%k2*cos(x)
(%i147) ic2(%146,x=0,y=-1,diff(y,x)=1);
(%o147) y=cos(x)*log(cos(x))+x*sin(x)+sin(x)-cos(x)
(%i148) bc2(%146,x=0,y=-1,x=%pi/4,y=0);
(%o148) y = cos x log cos x + x sin x + (2 log 2 - pi + 4) sin x / 4 - cos x
```

4.8.3. Riešenie sústav lineárnych diferenciálnych rovníc s konštantnými koeficientmi

Pri zadaní sústavy lineárnych diferenciálnych rovníc s konštantnými koeficientmi, ktorú sa chystáme riešiť pomocou funkcie `desolve`, musia byť funkcionálne vzťahy explicitne vyznačené (MAXIMA MANUAL, 2005, strana 211). Napríklad:

```
(%i149) dr1: 'diff(f(x),x)='diff(g(x),x)+sin(x)$
(%i150) dr2: 'diff(f(x),x)+x^2-f(x)=2*'diff(g(x),x);
(%i151) desolve([dr1,dr2],[f(x),g(x)]);
(%o151) [f(x)=sin(x)-cos(x)+(f(0)-1)*%e^(-x)+x^2-2*x+2,
        g(x)=sin(x)+(f(0)-1)*%e^(-x)+x^2-2*x+g(0)-f(0)+1]
(%i152) subst(-3,g(0),subst(2,f(0),%o151));
(%o152) [f(x)=sin(x)-cos(x)+%e^(-x)+x^2-2*x+2,
        g(x)=sin(x)+%e^(-x)+x^2-2*x-4]
```

Ak vynecháme apostrofy ', ktoré potláčajú vykonanie derivácie, nič zvláštneho sa nestane. Všimnite si, že vo výsledku sú nedefinované hodnoty $f(0)$ a $g(0)$. V príručkách (MAXIMA MANUAL, 2005; SOUZA, FATEMAN, MOSES a YAPP, 2004) sa uvádza, že začiatkové hodnoty je možné zadať

1. len v bode $x = 0$,
2. len pred použitím funkcie desolve pomocou funkcie atvalue.

V riadku 152 sa nám podarila substitúcia za $f(0)$ a $g(0)$. Je ťažké povedať, čo je na nej nevyhovujúce.

Vyskúšajme príklad, uvedený v knižke (SOUZA, FATEMAN, MOSES a YAPP, 2004):

```
(%i153) atvalue(f(x),x=0,1)$
(%i154) atvalue(g(x),x=0,2)$
(%i155) atvalue(diff(g(x),x),x=0,3)$
(%i156) dr3:diff(f(x),x)=diff(g(x),x)+sin(x)$
(%i157) dr4:diff(g(x),x,2)=diff(f(x),x)-cos(x)$
(%i158) sol:desolve([dr3,dr4],[f(x),g(x)]);
(%o158) [f(x)=3*e^x-2,g(x)=cos(x)+3*e^x-2]
```

Zdá sa, že sme dospeli k podobnému výsledku, ako v predchádzajúcom príklade, kde sme vopred nepoužili príkaz `atvalue`. Nie je však jasné, ako by sa dala dosadiť napríklad hodnota $g'(0)$ pomocou príkazu `subst`. Po ďalšom overovaní sme zistili, že pomocou príkazu `atvalue` je nutné zadať len hodnoty potrebných derivácií v bode $x = 0$, funkčné hodnoty neznámych funkcií v bode $x = 0$ je možné dodať pomocou príkazu `subst` aj po vyriešení sústavy diferenciálnych rovníc.

Ako sa dostaneme ku funkciám $f(x)$ a $g(x)$? Nasledujúce riadky ukazujú, že tieto funkcie nie sú k dispozícii hneď po riešení sústavy. Je potrebné im priradiť výrazy na pravej strane rovností (%158). Ale aj po priradení sa funkcie správajú zvlášťne, ako keby boli len výrazy (pozri riadok 164) – je možné ich derivovať, ale funkčné hodnoty získame len pomocou substitúcie:

```
(%i159) f(0);
(%o159) 1
(%i160) f(1);
(%o160) f(1)
(%i161) diff(f(x),x);
(%o161) 'diff(f(x),x,1)
(%i162) g(0);
(%o162) 2
(%i163) f(x):=last(sol[1])$
(%i164) f(1);
(%o164) 3*e^x-2
(%i165) subst(1,x,f(x));
(%o165) 3*e-2
(%i166) diff(f(x),x);
(%o166) 3*e^x
(%i167) g(x):=last(sol[2])$
(%i168) diff(g(x),x,3);
(%o168) sin(x)+3*e^x
(%i169) float(subst(3,x,g(x)));
(%o169) 57.26661827296256
(%i170) float(subst(3,x,diff(g(x),x)));
(%o170) 60.11549076150314
```

4.9. Laplaceova transformácia

MAXIMA ponúka výpočet Laplaceovej transformácie základných funkcií – \sin , \cos , \exp , \sinh , \cosh , \log , δ a erf . V porovnaní s programom Maple chýba napríklad Heavisideova funkcia, ako aj možnosť pracovať s funkciami, ktoré sú definované po častiach. V knihe (KREYSZIG a NORMINTON, 2006, strana 60) nájdete príklad takej funkcie a jej Laplaceovu transformáciu uskutočnenú programom Maple.

Pri výpočte Laplaceovej transformácie pracuje MAXIMA s deriváciami, integrálmi a súčtami. Popri funkcii `laplace` poskytuje funkciu `ilt` – inverznú Laplaceovu transformáciu.

Funkcia `laplace` rozozná konvolúciu v tvare

```
integrate(f(x)*g(t-x), x, 0, t)
```

iné typy konvolúcie nerozozná (MAXIMA MANUAL, 2005).

Príklad 34. Určme Laplaceovu transformáciu posunutej Diracovej funkcie $\delta(x - a)$.

Riešenie. Použijeme príkaz `laplace`, všimnite si, že je potrebné zadať aj premenné pred a po transformácii:

```
(%i171) laplace(delta(t-a),t,p);  
Is a positive, negative, or zero?  
negative;  
(%o171) 0  
(%i172) laplace(delta(t-a),t,p);  
Is a positive, negative, or zero?  
zero;  
(%o172) 1  
(%i173) laplace(delta(t-a),t,p);  
Is a positive, negative, or zero?  
positive;  
(%o173) %e^(-a*p)
```

Príklad 35. Vypočítajte Laplaceovu transformáciu (Laplaceov obraz) funkcie

$$f(t) = t^2 \cdot \sin(3t) \cdot \cos(4t) \cdot e^{-5t}.$$

```
(%i174) laplace(t^2*sin(3*t)*cos(4*t)*exp(-5*t),t,s);
(%o174) 6/(s^4+20*s^3+200*s^2+1000*s+1924)-
(2*(6*s+30)*(4*s^3+60*s^2+400*s+1000))/(s^4+20*s^3+200*s^2+1000*s+1924)^2-
((3*s^2+30*s+54)*(12*s^2+120*s+400))/(s^4+20*s^3+200*s^2+1000*s+1924)^2+
(2*(3*s^2+30*s+54)*(4*s^3+60*s^2+400*s+1000)^2)
/(s^4+20*s^3+200*s^2+1000*s+1924)^3
```

alebo

```
(%o174)
```

$$\frac{6}{s^4 + 20s^3 + 200s^2 + 1000s + 1924} -$$

$$\frac{2(6s + 30)(4s^3 + 60s^2 + 400s + 1000)}{(s^4 + 20s^3 + 200s^2 + 1000s + 1924)^2} -$$

$$\frac{(3s^2 + 30s + 54)(12s^2 + 120s + 400)}{(s^4 + 20s^3 + 200s^2 + 1000s + 1924)^2} +$$

$$+ \frac{2(3s^2 + 30s + 54)(4s^3 + 60s^2 + 400s + 1000)^2}{(s^4 + 20s^3 + 200s^2 + 1000s + 1924)^3}$$

Príklad 36. Určme spätnú Laplaceovu transformáciu (predmet) pre funkciu

$$F(s) = \frac{s - 5}{s^2 + 4s + 20}.$$



$$\text{ilt}((s-5)/(s^2+4s+20), s, t);$$
$$e^{-2t} \left(\cos(4t) - \frac{7 \sin(4t)}{4} \right)$$

Príklad 37. Nech $f(t) = t^2$ a $g(t) = \sin(3t)$. Vypočítajme Laplaceov obraz funkcie $\int_0^t f(x) \cdot g(t-x) dx$ a overme platnosť vety o konvolúcii.


```
(%i176) f(t):=t^2$
(%i177) g(t):=sin(3*t)$
(%i178) F(p):=laplace(f(t),t,p)$
(%i179) G(p):=laplace(g(t),t,p)$
(%i180) laplace(integrate(f(x)*g(t-x),x,0,t),t,p);
Is t positive, negative, or zero?
    positive;
Is cos(3*t) positive, negative, or zero?
    zero;
Is sin(3*t) positive, negative, or zero?
    negative;

(%o180) 
$$\frac{2p}{27(p^2+9)} + \frac{\frac{18}{p^3} - \frac{2}{p}}{27}$$

(%i181) F(p);
(%o181) 2/p^3
(%i182) G(p);
(%o182) 3/(p^2+9)
(%i183) F(p)*G(p);
(%o183) 6/(p^3*(p^2+9))
(%i184) radcan(%o180-%o183);
(%o184) 0
```

Program MAXIMA nereaguje na odpovede na otázky, ktoré sám zadával v riadku 180.

Príklad 38. Zobrazme vzorec Laplaceovej transformácie tretej derivácie funkcie $h(x)$.

```
(%i185) laplace(diff(h(x),x,3),x,s);
```

```
(%o185)
```

$$-\frac{d^2}{dx^2} h(x) \Big|_{x=0} - s \left(\frac{d}{dx} h(x) \Big|_{x=0} \right) + s^3 \mathcal{L}(h(x), x, s) - h(0) s^2$$

5. Riešenie algebrických úloh

V tejto kapitole uvedieme niektoré metódy riešenia rôznych algebrických úloh. Začneme úpravou polynómov a racionálnych funkcií, ďalej prejdeme na matice a úlohy lineárnej algebry a na záver sa budeme venovať riešeniu rôznych typov rovníc a ich sústav. S týmito úlohami je možné stretnúť sa v každodennom živote ako na vysokej škole, tak aj v praxi.



5.1. Mnohočleny

Podrobný popis funkcií na prácu s mnohočlenmi (polynómami) nájdete v knižke ([MAXIMA MANUAL, 2005](#)), množstvo informácií získate ako obvyčajne použitím príkazov `apropos`, `describe` a `example`, prípadne z položky menu `Help`.



Zadefinujme polynóm pomocou jeho rozkladu na súčin koreňových činiteľov, roznásobme členy a určme späťne rozklad na súčin koreňových činiteľov:

```
(%i1) p: (x+3)^2*(x-2)*3*(3*x+2);
(%o1) 3*(x-2)*(x+3)^2*(3*x+2)
(%i2) pe: expand(p);
(%o2) 9*x^4+42*x^3-3*x^2-180*x-108
(%i3) q: factor(pe);
(%o3) 3*(x-2)*(x+3)^2*(3*x+2)
```

V tomto prípade si SPA MAXIMA poradil s danou úlohou bez problémov.

Koeficienty polynómu pri jednotlivých mocninách zadanej premennej získame príkazom `coeff(p, x, n)`. Napríklad:

```
(%i4) c3:coeff(p,x,3);
(%o4) 0
(%i5) c3:coeff(pe,x,3);
(%o5) 42
(%i6) q:expand((x^2+x*y+3*y^2+7)^3);
(%o6) 27*y^6+27*x*y^5+36*x^2*y^4+189*y^4+19*x^3*y^3+
126*x*y^3+12*x^4*y^2+147*x^2*y^2+441*y^2+3*x^5*y+
42*x^3*y+147*x*y+x^6+21*x^4+147*x^2+343
(%i7) coeff(q,y,3);
(%o7) 19*x^3+126*x
```

Zdá sa, že polynóm musí byť v rozvinutom (expandovanom) tvare, aby sme sa dostali k jeho koeficientom. MAXIMA nerobí rozdiely medzi rôznymi premennými, ako sa o tom môžeme presvedčiť v riadku 7, kde sme požiadali o koeficient pri 3. mocnine premennej y .

V niektorých situáciách sa zíše funkcia na výpočet podielu dvoch polynómov. Funkcia `divide` má na vstupe dva polynómy (delenec a deliteľ), na výstupe vracia dvojprvkový zoznam (podiel polynómov a zvyšok):





```
(%i8) divide(p,x^2+x-6,x);
(%o8) [9*x^2+33*x+18,0]
(%i9) factor(%o8[1]);
(%o9) 3*(x+3)*(3*x+2)
(%i10) divide(p,x^2-x-6,x);
(%o10) [9*x^2+51*x+102,228*x+504]
(%i11) (x^2-x-6)*%o8[1]+%o8[2];
(%o11) (x^2-x-6)*(9*x^2+51*x+102)+228*x+504
(%i12) ratsimp(%);
(%o12) 9*x^4+42*x^3-3*x^2-180*x-108(%i11)
```

V riadku (%i8) sme delili zadaný polynóm p súčinom jeho dvoch koreňových činiteľov $(x - 2)(x + 3) = x^2 + x - 6$. Delenie teda logicky skončilo bez zvyšku. V riadku 10 už deliteľ nebol súčinom koreňových činiteľov, a tak sa v odpovedi okrem podielu objavil aj zvyšok $228x + 504$. Roznásobením (všimnite si, ako sme sa dostali ku jednotlivým zložkám zoznamu – anglicky list – odpovede %o8) v riadkoch 11–12 sme overili správnosť výsledku delenia. V tomto prípade sme namiesto príkazu `expand`, použili príkaz `ratsimp` (z anglického rational simplification – racionálne zjednodušenie) s rovnakým výsledkom.

Pozrime sa, ako dokáže MAXIMA riešiť algebrické rovnice:

```
(%i13) p:x^3+x^2+2;
(%o13) x^3+x^2+2
(%i14) factor(p);
(%o14) x^3+x^2+2
(%i15) solve(p);
(%o15) [x=((sqrt(3)*%i)/2-1/2)/(9*(3^(-3/2)*sqrt(29)-28/27)^(1/3))+
(3^(-3/2)*sqrt(29)-28/27)^(1/3)*(-(sqrt(3)*%i)/2-1/2)-1/3,
x=(3^(-3/2)*sqrt(29)-28/27)^(1/3)*((sqrt(3)*%i)/2-1/2)+
(-(sqrt(3)*%i)/2-1/2)/(9*(3^(-3/2)*sqrt(29)-28/27)^(1/3))-1/3,
x=(3^(-3/2)*sqrt(29)-28/27)^(1/3)+
1/(9*(3^(-3/2)*sqrt(29)-28/27)^(1/3))-1/3]

(%i16) ev(bfloat(rectform(%o20)),fpprec=5);
(%o16) [x=1.0276b0*%i+3.4716b-1,x=-1.6943b0,x=3.4716b-1-1.0276b0*%i]
(%i17) ev(bfloat(rectform(solve(x^4+x+1))),fpprec=6);
(%o17) [x=-4.30014b-1*%i-7.27136b-1,x=4.30014b-1*%i-7.27136b-1,
x=9.34099b-1*%i+7.26737b-1,x=7.27535b-1-9.34099b-1*%i]
(%i18) ev(bfloat(rectform(solve(x^5+x^2+1))),fpprec=6);
(%o18) [0.0b0=x^5+x^2+1.0b0]
(%i19) ev(bfloat(allroots(x^5+x^2+1)),fpprec=4);
Warning: Float to bigfloat conversion of 0.75151923237895168
...
Warning: Float to bigfloat conversion of -1.1938591113212367
(%o19) [x=7.846b-1*%i+7.515b-1,x=7.515b-1-7.846b-1*%i,
x=8.281b-1*%i-1.546b-1,x=-8.281b-1*%i-1.546b-1,x=-1.194b0]
(%i20) ev(bfloat(realroots(x^5+x^2+1)),fpprec=5,realonly=true);
(%o20) [x=-1.1939b0]
```

Vidíme, že funkcia `factor` neslúži na určovanie všeobecných koreňov algebrických rovníc. V tomto prípade sme použili funkciu `solve`. Táto dokáže riešiť rovnice tretieho aj štvrtého rádu (výsledok v prípade polynómu $x^4 + x + 4$ sme radšej nevypísali), MAXIMA zrejme pozná Cardanove i Ferrariho vzorce. V prípade rovnice 5. stupňa sme sa už presné riešenie samozrejme nedozvedeli. Numerické riešenie algebrických rovníc však umožňujú funkcie `allroots` a `realroots`.³⁷

Na záver tohoto oddielu uveďme ešte jednu ukážku použitia funkcií `expand` a `factor`, ktorá svedčí o tom, že ako „premenná“ polynómu môže vystupovať, napríklad, aj funkcia `sinus`:

```
(%i21) p: ((sin(x))^3+8)*((sin(x))^2+4*sin(x)+4)$
(%i22) expand(p);
(%o22) sin(x)^5+4*sin(x)^4+4*sin(x)^3+8*sin(x)^2+32*sin(x)+32
(%i23) factor(%);
(%o23) (sin(x)+2)^3*(sin(x)^2-2*sin(x)+4)
```

5.2. Racionálne funkcie

Z oblasti racionálnych funkcií uveďme len jeden príklad rozkladu racionálnej funkcie (podielu dvoch polynómov) na súčet polynómu a parciálnych

³⁷Treba povedať, že na numerické riešenie algebrických rovníc je lepšie použiť MATLAB, Pylab, Scilab alebo Octave, ktoré majú funkciu `roots`. Pozri učebnice (KAUKIČ, 2006; PRIBIŠ, 2006; BUŠA, 2006).

zlomkov. Riešenie tejto úlohy sa využíva pri integrovaní racionálnych funkcií alebo pri výpočte spätnej Laplaceovej transformácie. MAXIMA poskytuje viaceré funkcie na prácu s racionálnymi funkciami, vrátane Padeho aproximácie.

```
(%i24) q: (x^2+4*x+8)^2*(x-3)^2*(x+5)$
```

```
(%i25) expand(q);
```

```
(%o25) x^7+7*x^6+3*x^5-91*x^4-312*x^3+32*x^2+1536*x+2880
```

```
(%i26) r: (3*x^8-4*x^7+6*x^5+2*x+1)/(%o25);
```

$$(\%o26) \frac{3x^8 - 4x^7 + 6x^5 + 2x + 1}{x^7 + 7x^6 + 3x^5 - 91x^4 - 312x^3 + 32x^2 + 1536x + 2880}$$

Niekedy je potrebné získať menovateľa alebo čitateľa (anglicky denominator resp. numerator) zlomku. V takom prípade je možné použiť funkcie denom resp. num:

```
(%i27) denom(r);
```

```
(%o27) x^7+7*x^6+3*x^5-91*x^4-312*x^3+32*x^2+1536*x+2880
```

```
(%i28) num(r);
```

```
(%o28) 3*x^8-4*x^7+6*x^5+2*x+1
```

```
(%i29) gfactorsum(r);
```

$$(\%o29) \frac{(x+1)(3x^7 - 7x^6 + 7x^5 - x^4 + x^3 - x^2 + x + 1)}{(x-3)^2(x+5)(x-2i+2)^2(x+2i+2)^2}$$

Funkcia factorum resp. funkcia factor sa dá použiť nielen na rozklad

polynómov na súčin činiteľov, ale aj pre racionálnu funkciu (a iné). Funkcie `gfactorum` resp. `gfactor` vykonajú rozklad nad poľom komplexných čísel.

Na záver rozložme racionálnu funkciu na súčet polynómu a parciálnych zlomkov nad poľom reálnych čísel. V prostredí `WXMAXIMA` túto funkciu nájdete tiež v položke menu `Calculus`:

```
(%i30) partfrac(r, x);
```

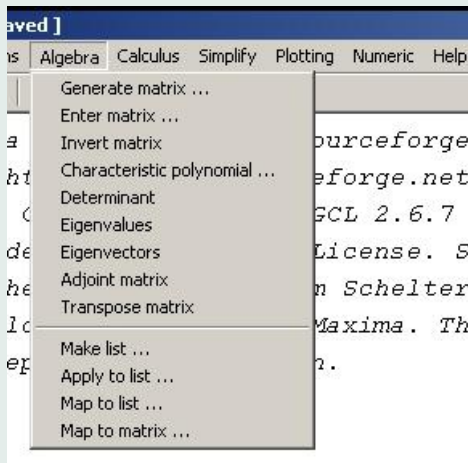
```
(%o30)
```

$$\frac{110742914x - 147276320}{4121741(x^2 + 4x + 8)} + \frac{325939x + 2358249}{10933(x^2 + 4x + 8)^2} + \frac{91601}{676(x + 5)} + 3x + \frac{353891}{97556(x - 3)} + \frac{1550}{841(x - 3)^2} - 25$$

5.3. Maticová algebra

Na obrázku 17 vidíte položky, ktoré poskytujú rôzne maticové funkcie, od zadávania matic až po určovanie vlastných hodnôt a vlastných vektorov matic.

Ako sme už písali vyššie, na prácu s maticami a na riešenie úloh lineárnej algebry je možné odporučiť programy `MATLAB`, `PyLab`, `Scilab` alebo `Octave` (KAUKIČ, 2006; PRIBIŠ, 2006; BUŠA, 2006) – posledné tri menované patria medzi programy s otvoreným zdrojovým kódom a je možné ich bezplatne získať z internetu. Ich popis nájdete na adrese <http://people.tuke.sk/jan.busa/kega/>.



Obr. 17: Položka menu Algebra

Tieto programy pracujú s maticami ako so svojimi základnými objektami a poskytujú funkcie založené na algoritmoch overených dlhoročným používaním. Z tohoto dôvodu popíšeme maticové funkcie programu MAXIMA len čiastočne a navyše iba stručne. Na druhej strane MAXIMA umožňuje riešiť tieto úlohy so symbolickými maticami. Táto črta môže byť využitá napríklad pri výučbe lineárnej algebry.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

[Strana 130 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

5.3.1. Vytváranie matíc

Na zadanie matíc používame príkaz `matrix`. V prostredí `WXMAXIMA` môžete na pohodlné zadávanie prvkov matice použiť položku menu Algebra/Enter Matrix. Prvky matíc sa zadávajú po riadkoch:

```
(%i31) A:matrix([1,-2], [a,4]);
```

```
(%o31)
```

$$\begin{pmatrix} 1 & -2 \\ a & 4 \end{pmatrix}$$

```
(%i32) B:matrix([1,-2,3], [-4,b,6]);
```

```
(%o32)
```

$$\begin{pmatrix} 1 & -2 & 3 \\ -4 & b & 6 \end{pmatrix}$$

5.3.2. Maticové operácie

Ako príklad uvidíme len súčin matíc. Ak ho zadáme tak, ako sme zvyknutí z iných programov, dočkáme sa nasledujúceho prekvapenia:

```
(%i33) A*B;  
'fullmap' found arguments with incompatible structure.  
-- an error. Quitting. To debug this try debugmode(true);
```

Operátor * sa totiž používa na tzv. zložkové násobenie dvoch matíc rovnakého rozmeru $m \times n$, keď sa operácia násobenia vykoná postupne pre všetky prvky na odpovedúcich pozíciách.³⁸ Znakom násobenia je v SPA MAXIMA **bodka** .:

```
(%i34) A.B;  
  
(%o34) 
$$\begin{pmatrix} 9 & -2b - a & -9 \\ a - 16 & 4b - 2a & 3a + 24 \end{pmatrix}$$
  
  
(%i35) A*A;  
(%o35) matrix([1,4],[a^2,16])  
(%i36) A.A;  
(%o36) matrix([1-2a,-10],[5a,16-2a])
```

³⁸V MATLAbE sa na takéto „zložkové“ násobenie matíc používa „bodkový“ operátor A.*B.

5.3.3. Maticové funkcie

Medzi základné maticové funkcie patria: determinant, hodnosť (anglicky rank), transponovanie a inverzná matica:

```
(%i37) C:matrix([a,b],[c,d]);
```

```
(%o37)          ( a  b )
                  ( c  d )
```

```
(%i38) determinant(C);
```

```
(%o38) a*d-b*c
```

```
(%i39) rank(A);
```

```
(%o39) 2
```

```
(%i40) transpose(A);
```

```
(%o40) matrix([1,a],[-2,4])
```

```
(%i41) inverse(C);
```

```
(%o41)          (  d      b      )
                  (  a d-bc  - a d-bc )
                  ( - a d-bc  a      )
                  (  a d-bc  a d-bc )
```

Ďalej sa využíva charakteristický polynóm štvorcovej matice A , definovaný ako determinant matice $A - r \cdot I$, kde r je premenná charakteristického polynómu a I je jednotková matica (anglicky identity matrix).

```
(%i42) charpoly(A, r), expand;
(%o42) r^2-5*r+2*a+4
(%i43) I:ident(2);
(%o43) matrix([1,0],[0,1])
(%i44) determinant(A-r*I);
(%o44) (1-r)*(4-r)+2*a
(%i45) ratsimp(%);
(%o45) r^2-5*r+2*a+4
```

Vidíme, že sme dostali rovnaký výsledok.

Vlastné hodnoty a vlastné vektory (anglicky eigenvalues a eigenvectors) **symbolickej** matice vypočítame nasledujúcim spôsobom:

```
(%i46) eigenvalues(A);
(%o46) [[[-(sqrt(9-8a)-5)/2, (sqrt(9-8a)+5)/2], [1,1]]]
(%i47) eigenvectors(A);
(%o47) [[[[[-(sqrt(9-8a)-5)/2, (sqrt(9-8a)+5)/2], [1,1]], [1, (sqrt(9-8a)-3)/4], [1, -(sqrt(9-8a)+3)/4]]]]
```



Funkcia `eigenvalues` vráti dva zoznamy. Prvý obsahuje vlastné hodnoty matice a druhý ich násobnosti. Na prvý pohľad zbadáme, že funkcia `eigenvectors` vráti zoznam, obsahujúci vlastné čísla, ich násobnosti a vlastné vektory matice. V prípade násobných vlastných čísel sa počet vlastných vektorov môže zmenšiť:

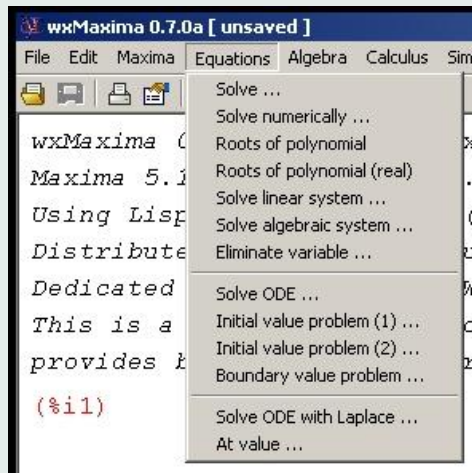
```
(%i48) eigenvectors(matrix([3,0],[0,3]));
(%o48) [[[3],[2]],[1,0],[0,1]]
(%i49) eigenvectors(matrix([3,-2],[0,3]));
(%o49) [[[3],[2]],[1,0]]
```

V oboch prípadoch má matica dvojnásobnú vlastnú hodnotu 3, ale v druhom prípade má len jeden vlastný vektor.

5.4. Riešenie lineárnych a nelineárnych (algebraických) rovníc a sústav

V niekoľkých nasledujúcich oddieloch popíšeme stručne možnosti riešenia sústav rovníc. V prostredí `WXMAXIMA` tieto možnosti môžeme skúmať použitím podpoložiek menu položky `Equations`, zobrazenej na nasledujúcom obrázku 18.

V tejto učebnici sme sa už na viacerých miestach stretli s použitím funkcií `solve` a `algsys`.



Obr. 18: Položka menu Equations

5.4.1. Funkcia solve

Po zadaní príkazu `describe(solve)` a použití voľby

```
12: solve :Definitions for Equations
```

sa dozvedáme:

Funkcia `solve(<výraz>, <x>)` rieši algebraickú rovnicu `<výraz>` s neznámou `<x>` a vracia zoznam riešení tejto rovnice. Ak `<výraz>` nie je rovnica, predpokladá sa rovnica „`<výraz>=0`“. `<x>` môže

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

◀

▶

◀

▶

Strana 136 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

byť aj funkcia (napríklad „ $f(x)$ “) alebo nejaký iný (neatomárny) výraz s výnimkou súčtu alebo súčinu. Ak <výraz> obsahuje len jednu premennú, <x> nie je potrebné zadávať. <výraz> môže byť racionálny a môže obsahovať trigonometrické funkcie, exponenty atď. ...

Funkcie `solve([<r_1>, ..., <r_n>], [<x_1>, ..., <x_n>])` rieši sústavu (lineárnych alebo nelineárnych) algebrických rovníc volaním funkcie `linsolve` alebo `algsys`. ...

Zvedavému čitateľovi odporúčame podrobne sa oboznámiť s úplným znením výpisu príkazu `describe(solve)`.

Nasledujúci príklad svedčí o tom, že funkcia `solve` si poradí aj s ďalšími typmi rovníc:

```
(%i50) solve(sin(x)=1/2);  
'solve' is using arc-trig functions to get a solution.  
Some solutions will be lost.  
(%o50) [x=%pi/6]
```

V tomto prípade nás SPA MAXIMA upozornil, aby sme sa zamysleli nad ďalšími možnými riešeniami danej rovnice.

5.4.2. Riešenie sústav lineárnych algebrických rovníc pomocou funkcie `linsolve`

Na riešenie sústav lineárnych algebrických rovníc slúži funkcia `linsolve`:

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

◀◀ ▶▶

◀ ▶

Strana 137 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

```
(%i51) linsolve([x+y=2, x+y=3], [x,y]);  
Inconsistent equations: (2)  
-- an error. Quitting. To debug this try debugmode(true);  
(%i52) linsolve([x+y=2, 2*x+2*y=4], [x,y]);  
Dependent equations eliminated: (2)  
(%o52) [x=2-%r1,y=%r1]
```

Vidíme, že program MAXIMA si poradil s problematickými sústavami.

Riešenie symbolických rovníc je samozrejmosťou:

```
(%i53) linsolve([a*x+y=2, (2-a)*x+a*y=4], [x,y]);
```

$$(%o53) \quad \left[x = \frac{2a - 4}{a^2 + a - 2}, y = \frac{6a - 4}{a^2 + a - 2} \right]$$

```
(%i54) subst(-2,a,%i53);
```

```
(%o54) linsolve([y-2*x=2,4*x-2*y=4], [x,y])
```

```
(%i55) subst(-2,a,%o53);
```

```
Division by 0
```

```
-- an error. Quitting. To debug this try debugmode(true);
```

```
(%i56) subst(2,a,%i53);
```

```
(%o56) linsolve([y+2*x=2,2*y=4], [x,y])
```

```
(%i57) subst(2,a,%o53);
```

```
(%o57) [x=0,y=2]
```

Aj v tomto prípade môžeme byť spokojní s riešením programu MAXIMA.

5.4.3. Funkcia `algsys`

Funkcia `algsys([<r_1>, ..., <r_m>], [<x_1>, ..., <x_n>])` rieši (najmä) sústavy algebrických rovníc, ktoré majú tvar mnohočlenov obsahujúcich súčiny rôznych mocnín neznámych.

Aj v tomto prípade je potrebné dávať si pozor, pretože výsledok nemusí byť vždy taký, ako by sme očakávali. Napríklad nasledujúca sústava má nekonečne veľa riešení, ale MAXIMA vydá len dve z nich:

```
(%i58) algsys([x^2-y^2=0, (x+2)*sin(y)=0], [x, y]);
(%o58) [[x=-2, y=2], [x=-2, y=-2]]
```

Ako používateľa SPA MAXIMA na úrovni začiatočníka môžeme len hádať, prečo sa program nevenoval „jasnej“ možnosti $\sin(y) = 0$.

5.4.4. Numerické riešenie rovníc pomocou funkcie `find_root`

Príklad 39. Určme numerické (približné) riešenie rovnice $\cos(x) = x$.

Riešenie. Ak načrtneme grafy funkcií $y = \cos(x)$ a $y = x$ uvidíme, že tieto majú jediný priesečník, ktorý sa nachádza na intervale $\langle 0, \pi/2 \rangle$. Určíme ho príkazom:

```
(%i59) find_root(cos(x)=x, x, 0, 5pi/2);
(%o59) 0.73908513321516
```

Podmienky použitia tejto funkcie si naštudujte samostatne.

6. Štatistické funkcie

MAXIMA poskytuje aj najdôležitejšie štatistické funkcie, žiaľ, v uvedenej literatúre chýba ich systematický opis. Preto ich uvedieme len stručne.

Začneme načítaním balíka `distrib`, ktorý nájdete v podpriechniku `share/contrib/distrib`.³⁹

```
(%i1) load(distrib)$  
(%o1) .../maxima/5.10.0/share/contrib/distrib/distrib.mac
```

Na prvé (a teda povrchné) oboznámenie sa s obsahom balíka môžeme použiť príkaz `apropos`.⁴⁰

³⁹V podpriechnikoch `share` resp. `share/contrib` nájdete ďalšie užitočné balíky, ktoré ešte len čakajú na opis v uvedeých učebniciach.

⁴⁰Vieme, že v MATLAbE sa distribučná funkcia (anglicky `cumulative density function`) nazýva `menocdf`.



```
(%i2) apropos(cdf);(%o2) [cdf,cdf_bernoulli,cdf_beta,
cdf_binomial,cdf_cauchy,cdf_chi2,cdf_continuous_uniform,
cdf_discrete_uniform,cdf_exp,cdf_f,cdf_gamma,cdf_geometric,
cdf_gumbel,cdf_hypergeometric,cdf_laplace,cdf_logistic,
cdf_lognormal,cdf_negative_binomial,cdf_normal,cdf_pareto,
cdf_poisson,cdf_rayleigh,cdf_student_t,cdf_weibull]
(%i3) apropos(pdf);
(%o3) [pdf,pdf_bernoulli,pdf_beta,pdf_binomial,pdf_cauchy,
pdf_chi2,pdf_continuous_uniform,pdf_discrete_uniform,pdf_exp,
pdf_f,pdf_gamma,pdf_geometric,pdf_gumbel,pdf_hypergeometric,
pdf_laplace,pdf_logistic,pdf_lognormal,pdf_negative_binomial,
pdf_normal,pdf_pareto,pdf_poisson,pdf_rayleigh,pdf_student_t,
pdf_weibull]
```

Teda balíček `distrib` poskytuje distribučné funkcie (`cdf`) a funkcie rozdelenia pravdepodobnosti alebo jej hustoty (`pdf` – z anglického *probability density function*) pre všetky najčastejšie používané rozdelenia.

Pozrime sa, napríklad, na opis distribučnej funkcie normálneho rozdelenia (`cdf_normal`):

```
(%i4) describe(cdf_normal)$
-- Function: cdf_normal (<x>,<m>,<s>)
Returns the value at <x> of the distribution function of a
Normal(m,s) random variable, with s>0. This function is
defined in terms of Maxima's built-in error function 'erf'.
(%i1) load (distrib)$
(%i2) assume(s>0)$ cdf_normal(x,m,s);
                                x - m
                                erf(-----)
                                sqrt(2) s
(%o3) ----- + -
                                2          2
See also 'erf'.
```

Vstupy tejto funkcie sú teda štandardné – hodnota x , stredná hodnota m a smerodajná odchýlka s . Vyskúšajme ju:

```
(%i5) cdf_normal(3*2,0,2);
(%o5)
                                erf( (3/√2) )
                                ----- + 1/2
                                2
(%i6) float(%);
(%o6) 0.99865010196837
```

Dostali sme najprv výsledok v tvare chybovej funkcie, v podstate nepoužiteľný. Našťastie v tomto prípade nebol problém získať jeho numerickú hodnotu a overiť, že výsledok je taký, aký sme očakávali.

Aj keď balíček neobsahuje špeciálne funkcie na testovanie štatistických hypotéz, rozhodujúca je predsa len dostupnosť distribučných funkcií. Tieto môžeme použiť na určenie kritických hodnôt pre jednotlivé testy.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 143 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

7. Tvorba grafov

V predošlých častiach sme na dvoch príkladoch ukázali, ako MAXIMA v spolupráci s externým programom Gnuplot, vytvorí zobrazenie dvoj- a trojrozmerných plôch (grafov funkcií jednej a dvoch premenných). V knihe The MAXIMA book (SOUZA, FATEMAN, MOSES a YAPP, 2004) nájdete ďalšie pekné obrázky, rovnako ako aj inde – napríklad (STOPKA, 2006) alebo na stránke [maxima-gnuplot](#). Ďalej len opíšeme voľby príkazu `plot`.

7.1. Voľby príkazov `plot`

Tento oddiel je úplne prevzatý z knihy The MAXIMA book (SOUZA, FATEMAN, MOSES a YAPP, 2004).

MAXIMA definuje zoznam, ktorý sa nazýva `PLOT_OPTIONS` a ktorý riadi podstatnú časť správania sa grafických možností SPA MAXIMA. Existujú dve cesty nastavenia týchto volieb – jednotlivú voľbu môžete nastaviť globálne použitím príkazu `SET_PLOT_OPTION`, alebo môžete voľbu zadať ako argument príkazu `plot`. Ak chcete zistiť aktuálne globálne nastavenia,⁴¹ stačí napísať `PLOT_OPTIONS`; a vypíše sa Vám stav systému.

⁴¹Globálne nastavenia sú platné len pre práve spustený režim programu MAXIMA.

Domovská stránka

Titulná strana

Obsah

◀◀ ▶▶

◀ ▶

Strana 144 z 156

Späť

Celá strana

Zatvoriť

Koniec


```
(%i1) PLOT_OPTIONS;  
(%o1) [[x, -3, 3], [y, -3, 3], [t, -3, 3], [GRID, 30, 30],  
[VIEW_DIRECTION, 1, 1, 1], [COLOUR_Z, FALSE], [TRANSFORM_XY,  
FALSE], [RUN_VIEWER, TRUE], [PLOT_FORMAT, GNUPLOT],  
[GNUPLOT_TERM, DEFAULT], [GNUPLOT_OUT_FILE, FALSE],  
[NTICKS, 10], [ADAPT_DEPTH, 10], [GNUPLOT_PM3D, FALSE],  
[GNUPLOT_PREAMBLE, ], [GNUPLOT_CURVE_TITLES, [DEFAULT]],  
[GNUPLOT_CURVE_STYLES, [with lines 3, with lines 1, with  
lines 2, with lines 5, with lines 4, with lines 6, with  
lines 7]], [GNUPLOT_DEFAULT_TERM_COMMAND, ],  
[GNUPLOT_DUMB_TERM_COMMAND, set term dumb 79 22],  
[GNUPLOT_PS_TERM_COMMAND, set size 1.5, 1.5;  
set term postscript eps enhanced color solid 24]]
```

OK, pozrime sa na každú z týchto volieb.

`[x, -3, 3]` – toto definuje oblasť hodnôt premennej X . Na globálnu zmenu použite príkaz v tvare `SET_PLOT_OPTION([x, -5, 5]);`

`[y, -3, 3]` – toto definuje oblasť hodnôt premennej Y . Na globálnu zmenu použite príkaz v tvare `SET_PLOT_OPTION([y, -5, 5]);`

`[t, -3, 3]` – t implicitne nastavená premenná v grafoch parametricky zadaných funkcií. Na globálnu zmenu použite príkaz v tvare `SET_PLOT_OPTION([t, -5, 5]);`

`[GRID, 30, 30]` – voľba v prípade 3D grafov riadi počet bodov použitých na nakreslenie obrázku. Funkčné hodnoty sú počítané len v určitých bodoch,

v ostatných sa používa lineárna aproximácia. Na globálnu zmenu použite príkaz v tvare

```
SET_PLOT_OPTION([GRID,40, 35]);
```

[VIEW_DIRECTION, 1, 1, 1] – táto voľba sa týka prípadu, keď sa príkaz plot používa na výstup priamo do postscriptu v 3D (viď PLOT_FORMAT). Určuje, z ktorého bodu sa „kamera“ pozerá na graf funkcie, pozdĺž priamky paralelnej spojnici VIEW_DIRECTION a počiatku. V ostatných príkazoch sa ignoruje. Na globálnu zmenu použite príkaz v tvare

```
SET_PLOT_OPTION([VIEW_DIRECTION,1.4,1.4, 1.4]);
```

[COLOUR_Z, FALSE] – táto voľba sa tiež týka výstupu do postscriptu – ak je nastavená na TRUE, zabezpečí jemné farebné tieňovanie na výstupe.

Tvar je SET_PLOT_OPTION([COLOUR_Z, TRUE]);

[TRANSFORM_XY, FALSE] – voľba sa týka možnosti vytvárania grafov v rôznych súradnicových systémoch, ale nie som si istý, ako to funguje. Tu potrebujem pomoc.

[RUN_VIEWER, TRUE] – ak si prajete len výstup do súboru bez spustenia grafického okna, nastavte túto voľbu na FALSE. Avšak zapamätajte si, že ak budete potrebovať zadať násobné príkazy na vygenerovanie dát, budete zakaždým musieť obnoviť informáciu zo súboru maxout.PLOT_FORMAT, pretože každý nový príkaz plot ju prepíše. Tvar je SET_PLOT_OPTION([RUN_VIEWER, FALSE]);

[PLOT_FORMAT, GNUPLOT] – voľba určuje program, ktorý zobrazí výstup programu MAXIMA. Momentálne sú možné nasledujúce voľby – GNUPLOT,

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

[Strana 146 z 156](#)

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

OPENMATH, MGNUPLOT, GEOMVIEW a PS. PS je jednoducho priamy výstup do postscriptového súboru `maxout.ps`.

Tvar je `SET_PLOT_OPTION([PLOT_FORMAT, GEOMVIEW])`; MGNUPLOT je interface ku GNUPLOTu (dodaný s programom MAXIMA), ktorý poskytuje základné prostredie GUI, ale je málo rozvinuté. Všetky tieto programy sú voľne dostupné. Geomview je momentálne len pre Unix, dostupný na <http://geomview.sourceforge.net> (zdrojáky aj binárky). Openmath je súčasťou distribúcie SPA MAXIMA. Gnuplot je široko dostupný, jeho domovská stránka je <http://www.gnuplot.info> a aktuálna verzia na stránke <http://sourceforge.net/projects/gnuplot>. Môže bežať v oboch operačných systémoch Windows a Linux a momentálne je implicitne nastavený.

`[GNUPLOT_TERM, DEFAULT]` – táto voľba sa týka formátu gnuplot. Nastavuje typ výstupného terminálu (zariadenia) pre Gnuplot. Voľby sú — `DEFAULT`: osobitné grafické okno, `DUMB` – vytvorí ASCII aproximáciu grafu a `PS` – použijete túto voľbu spoločne s `GNUPLOT_OUT_FILE` ak potrebujete zapísať výstup Gnuplotu do postscriptového súboru namiesto na obrazovku. Tvar je `SET_PLOT_OPTION([GNUPLOT_TERM, PS])`;

`[GNUPLOT_OUT_FILE, FALSE]` – táto voľba sa týka formátu gnuplot. Používa sa na nastavenie názvu výstupného súboru.

Tvar je `SET_PLOT_OPTION([GNUPLOT_OUT_FILE, "myplot.ps"])`;

`[NTICKS, 10]` – riadi začiatkový počet bodov použitých pre adaptívnu postup kreslenia. Tvar je `SET_PLOT_OPTION([NTICKS, 200])`;

`[ADAPT_DEPTH, 10]` – riadi maximálny počet delení pri adaptívnom režime kreslenia. Tvar je `SET_PLOT_OPTION([ADAPT_DEPTH, 5])`;

Domovská stránka

Titulná strana

Obsah



Strana 147 z 156

Späť

Celá strana

Zatvoriť

Koniec

`[GNUPLOT_PM3D, FALSE]` – táto voľba sa týka formátu gnuplot. Riadi použitie režimu PM3D, ktorý má doplnkové 3D črty. PM3D je súčasťou gnuplotu od verzie 3.7. Tvar je `SET_PLOT_OPTION([GNUPLOT_PM3D, TRUE]);`

`[GNUPLOT_PREAMBLE,]` – táto voľba sa týka formátu gnuplot. Vkladá príkazy Gnuplotu pred nakreslením grafu. Môže byť použitý ľubovoľný platný príkaz Gnuplotu. Príkazy musia byť oddelené bodkočiarkou. Môžu byť nastavené ja voľby, ktoré nedokáže nastaviť MAXIMA.
Tvar je `SET_PLOT_OPTION([GNUPLOT_PREAMBLE, "set log y; set log x"]);`

`[GNUPLOT_CURVE_TITLES, [DEFAULT]]` – táto voľba sa týka formátu gnuplot. Riadi výpisy zadané v legende grafu. Implicitná hodnota je DEFAULT. Automaticky pridáva názov pre každú kreslenú krivku. V prípadoch iných ako DEFAULT, musí `GNUPLOT_CURVE_TITLES` obsahovať zoznam reťazcov. (Na vypnutie legendy pridajte voľbu "set nokey" do `GNUPLOT_PREAMBLE`.)
Tvar je `SET_PLOT_OPTION([GNUPLOT_CURVE_TITLES, ["my first function","my second function"]]);`

`[GNUPLOT_CURVE_STYLES, [with lines 3, with lines 1, with lines 2, with lines 5, with lines 4, with lines 6, with lines 7]]` – táto voľba sa týka formátu gnuplot. Je to zoznam reťazcov, určujúcich vlastnosti kriviek, to znamená farbu, šírku, prerušovanie čiar a pod., ktorý sa pošle do príkazu Gnuplotu. Na ďalšiu informáciu pozri dokumentáciu gnuplotu pre "plot".
Tvar je `SET_PLOT_OPTION([GNUPLOT_CURVE_STYLES, ["with lines 7", "with lines 2"]]);`

`[GNUPLOT_DEFAULT_TERM_COMMAND,]` – táto voľba sa týka formátu gnuplot. Je to príkaz Gnuplotu na nastavenie predvoleného typu terminálu. Implicitná

hodnota je "" – použitie implicitného nastavenia Gnuplotu. Tvar je
`SET_PLOT_OPTION([GNUPLOT_DEFAULT_TERM_COMMAND, "set term x11"]);`

`[GNUPLOT_DUMB_TERM_COMMAND, set term dumb 79 22]` – táto voľba sa týka formátu gnuplot. Nastavuje terminál na typ DUMB. Tvar je
`SET_PLOT_OPTION([GNUPLOT_DUMB_TERM_COMMAND, "set term dumb 80 24"]);`

`[GNUPLOT_PS_TERM_COMMAND, set size 1.5, 1.5; set term postscript eps enhanced color solid 24]` – táto voľba sa týka formátu gnuplot. Je to príkaz Gnuplotu nastavujúci typ terminálu na postscriptový terminál. Na podrobnejšiu informáciu sa pozri do dokumentácie Gnuplotu pre "set term postscript".

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 149 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

8. Programovanie v SPA MAXIMA

V SPA MAXIMA je možné písať programy (alebo vlastné funkcie). MAXIMA poskytuje štandardné príkazy, akými sú napríklad príkazy cyklu typu `for` alebo `while`. Keďže sa jedná o systém počítačovej algebry, jeho možnosti sú širšie, ako možnosti programovacích jazykov typu FORTRAN, zameraných skôr na numerické výpočty.

Na podrobný opis prvkov programovacieho jazyka SPA MAXIMA nemáme dostatok priestoru, navyše tento opis vychádza za rámec našej učebnice. V knihe ([MAXIMA MANUAL, 2005](#)) je tejto problematike venovaných niekoľko kapitol.

Uvedieme preto len jeden príklad definovania a použitia funkcie, ktorý je uvedený v príručke [The MAXIMA Book \(SOUZA, FATEMAN, MOSES a YAPP, 2004\)](#).

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

◀◀ ▶▶

◀ ▶

Strana 150 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)



```
(%i1) test(x,y):=block([],
if x>y then print(x, "is greater than", y)
else print(x, "is not greater than", y),return(alldone))$
(%i2) test(4,3);
4 is greater than 3
(%o2) alldone
(%i3) test(3,4);
3 is not greater than 4
(%o3) alldone
(%i4) test(y^2+1,-y);
Maxima was unable to evaluate the predicate:
y^2+1>-y
#0: test(x=y^2+1,y=-y)
-- an error. Quitting. To debug this try debugmode(true);
```

V knihe The MAXIMA Book (SOUZA, FATEMAN, MOSES a YAPP, 2004) je vysvetlené, že MAXIMA nedokáže rozhodnúť o platnosti nerovnice, pretože nemá informáciu o type premennej y , ktorá by mohla byť napríklad maticou. Ak budeme predpokladať, že y je kladné, situácia sa zmení:

```
(%i5) assume(y>0);
(%o5) [y>0]
(%i6) test(y^2+1,-y);
y^2+1 is greater than -y
(%o6) alldone
```

Skúsme však predpokladať, že y je záporné:

```
(%i7) forget(y>0)$
(%i8) assume(y<0);
(%o8) [y<0]
(%i9) test(y^2+1,-y);
Maxima was unable to evaluate the predicate:
y^2+1>-y
#0: test(x=y^2+1,y=-y)
-- an error. Quitting. To debug this try debugmode(true);
```

Hoci nerovnosť platí aj pre záporné y , program MAXIMA nedokázal rozhodnúť. Bez ďalšieho uvažovania totiž po dosadení do nerovnosti dostaneme na obidvoch stranách kladné číslo, z hľadiska pohľadu programu MAXIMA zrejme neurčité.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)

[◀](#)

[▶](#)

[◀](#)

[▶](#)

Strana 152 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Záver

Milá čitateľka, vážený čitateľ!

Nastal čas na rozlúčku. Verím, že ste sa pri čítaní tejto knižky pobavili aspoň tak, ako som sa pobavil ja pri jej písaní. Dúfam, že Vás možnosti systému počítačovej algebry MAXIMA zaujali a stanete sa jeho používateľmi.

Snažil som sa zároveň ukázať, že SPA MAXIMA je vhodný nástroj na využitie vo výučbe (najmä) matematických predmetov, predovšetkým matematickej analýzy, lineárnej algebry i matematickej štatistiky. Dá sa použiť aj na výučbu numerických metód. Na druhej strane na výučbu vymenovaných predmetov s výnimkou matematickej analýzy je možné a vhodné použiť iné OPENSOURCE programy – Pylab, Scilab, Octave (KAUKIČ, 2006; PRIBIŠ, 2006; BUŠA, 2006), R apod.

Vôbec som sa nedotkol ďalších oblastí použitia programu MAXIMA, medzi nimi napríklad teórie čísel, kombinatoriky a iných. Snáď sa do toho pustí niekto ďalší, komu môžu priniesť použitie SPA MAXIMA v týchto oblastiach väčší úžitok. Podobne, žiaľ, zrejme väčšina príkazov programu MAXIMA, ktoré sú dostatočne popísané napríklad v knihe (MAXIMA MANUAL, 2005), nebola v tejto učebnici ani spomenutá. Verím, že i napriek týmto nedostatkom bude učebnica užitočná, čítaná a používaná.

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 153 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)

Literatúra

Buša, J. 2006. *Octave. Rozšírený úvod*, ISBN 80-8073-596-4,
online na <http://people.tuke.sk/jan.busa/kega/octave/>. 127, 129,
153

Doboš, J. 2006. *GNU PLOT*, ISBN 80-8073-637-5, 52 s.,
online na <http://people.tuke.sk/jan.busa/kega/gnuplot/>. 87, 91

Ђјakonov, V. P. 2003. *Maple 8 v matematike, fizike i obrazovanii*, Moskva,
SOLON-Press, ISBN 5-98003-038-7, 656 s. 96

Glöckner, R. 2006. *Einführung in Maxima*, 23 s.

Gräbe, H.-G. 2005. *Skript zum Kurs Einführung in das symbolische Rechnen*,
Wintersemester 2004/2005, Institut für Informatik, Leipzig, 158 s.,
<http://www.informatik.uni-leipzig.de/~graebe>.

Micro introduction into Maxima, Computing Harvard Math department, on-
line na <http://www.math.harvard.edu/computing/maxima/>.

Kawano, T. 2005. *Gnuplot – not so Frequently Asked Questions*,
<http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>. 91

Kaukič, M. 2006. *Základy programovania v Pylabe*, ISBN 80-8073-635-9,
<http://people.tuke.sk/jan.busa/kega/pylab/>. 127, 129, 153

Domovská stránka

Titulná strana

Obsah



Strana 154 z 156

Späť

Celá strana

Zatvoriť

Koniec

Kreyszig, E. — Norminton, E. J. 2006. *Maple Computer Guide, A Self-Contained Introduction*, for E. Kreyszig Advanced Engineering Mathematics, 9. vydanie, John Wiley & Sons, Inc., ISBN 0-471-72645-1, 300 s. 96, 117

2006. *Maxima Manual*, 461 s. 8, 37, 38, 40, 41, 53, 94, 113, 114, 117, 123, 150, 153

2006. *Maxima Manual*, online na

<http://maxima.sourceforge.net/docs/manual/en/maxima.html>.

Pribiš, J. 2006. *Scilab*, ISBN 80-8073-655-3, <http://people.tuke.sk/jan.busa/kega/scilab/>. 127, 129, 153

Rand, R. H. 2005. *Introduction to Maxima*, Cornell University. 41

Souza, P. N. de — Fateman, R. J. — Moses, J. — Yapp, C. 2004. *The Maxima Book*, 154 s. 7, 13, 42, 96, 98, 104, 105, 106, 108, 109, 114, 144, 150, 151

Stopka M. *Hra s písmenky wxMaxima*, ABC Linuxu, 18. 5. 2006,

online na <http://www.abclinuxu.cz/clanky/programovani/hra-s-pismenky-wxmaxima>. 144

Domovská stránka

Titulná strana

Obsah



Strana 155 z 156

Späť

Celá strana

Zatvoriť

Koniec

NÁZOV: MAXIMA. Open source systém počítačovej algebry
AUTOR: Ján Buša
POČET STRÁN:
VYDANIE: prvé
SADZBA: elektronická, programom pdfT_EX

ISBN 80-8073-641-3

[Domovská stránka](#)

[Titulná strana](#)

[Obsah](#)



Strana 156 z 156

[Späť](#)

[Celá strana](#)

[Zatvoriť](#)

[Koniec](#)