

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA MATEMATIKY

SYSTÉM POČÍTAČOVÉ ALGEBRY MAXIMA
BAKLÁŘSKÁ PRÁCE

Mgr. Lucie Olšavská

Vedoucí práce: Mgr. Lukáš HONZÍK, Ph.D.

Plzeň, 2015

Prohlašuji, že jsem práci vypracoval(a) samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni dne

.....

Poděkování

Chtěla bych poděkovat vedoucímu bakalářské práce Mgr. Lukáši HONZÍKOVÍ, Ph.D. za metodické vedení a věcné připomínky při zpracování bakalářské práce.

Obsah

Část I	3
Úvod	3
Část II	5
Teoretická část	5
1 Základní ovládání programu Maxima	5
2 Grafické prostředí wxMaxima	8
2.1 Rovnice	8
2.2 Algebra	9
2.3 Analýza	9
2.4 Numerické výpočty	10
2.5 Grafy	11
2.6 Ovládání pro tvorbu grafů	12
2.7 Prohlížeč	13
2.8 Nápověda	13
3 Online webové rozhraní programu Maxima	14
3.1 Maxima online	14
3.2 Zápis příkazů	15
Část III	28
Praktická část	28
4 Ukázka řešených příkladů	28
5 Zhodnocení dovedností programu	42
Část IV	44
Závěr	44
Resumé	45

Část V	46
Literatura	46
Seznam obrázků	47
Seznam příkladů	50

Část I

Úvod

V dnešní době je program Maxima svobodný počítačový algebraický systém pro manipulaci se symboly a numerickými hodnotami. Je dostupný pro všechny platformy Posix, jako Unix, BSD nebo Linux. Dostupná je také verze binárních souborů pro MS Windows. WxMaxima je multiplatformová verze s grafickým uživatelským rozhraním.

Pomocí tohoto programu se dají vypočítávat diferenciály, integrály, derivace, Taylorovy řady, Laplaceovy obrazy, obyčejné diferenciální rovnice, soustavy lineárních rovnic, polynomy, vektory, matice a tenzory (Objekt, který je zobecněním pojmu vektor. Složky vektoru je možné označit jedním indexem, tenzor může mít více indexů). Maxima poskytuje vysoce přesné číselné výsledky pomocí zlomků, libovolných přesných celých čísel a čísel variabilních s plovoucí řádovou čárkou. Maxima umí vykreslit funkce a data ve dvou a třech rozměrech.

Maxima je potomek Macsyma, legendárního počítačového systému pro algebru, vyvinutého na konci roku 1960 na Massachusetts Institute of Technology. Je to jediný systém veřejně k dispozici s aktivní komunitou uživatelů, díky open source povaze (Všichni uživatelé mohou posílat své připomínky a podněty ke zkvalitnění programu). Macsyma byl revoluční ve vývoji a mnoho systémů, jako je Maple a Mathematica, jím byly inspirovány.

Program Maxima byl udržován Williamem Schelterem od roku 1982 do roku 2001, kdy zemřel. V roce 1998 získal povolení ke zveřejnění a ochraně zdrojového kódu pod GNU General Public License (Česky „všeobecná veřejná licence GNU“ je licence pro svobodný software, GPL je nejznámějším příkladem silně copyleftové licence, tzn., že díla odvozená jsou dostupná pod toutéž licencí). Toto jeho úsilí a dovednosti vedly k přežití Maximy dodnes. Od jeho smrti skupina uživatelů a vývojářů vytvořila program Maxima pro širší publikum.

Program Maxima je velmi často aktualizován, vývojáři opravují chyby a snaží se zlepšit kód a dokumentaci. Ve finální podobě nebude nejspíš nikdy, jelikož na něm bude stále co zlepšovat.

Cílem této práce je popsání programu Maxima a zhodnocení grafického prostředí wxMaxima a online webového rozhraní programu.

Bakalářská práce je rozdělena do dvou částí. V teoretické části se zabývám ovládním samotného programu, co dokáže, případně nedokáže. V praktické části jsou ukázky typových příkladů.

Část II

Teoretická část

1 Základní ovládání programu Maxima

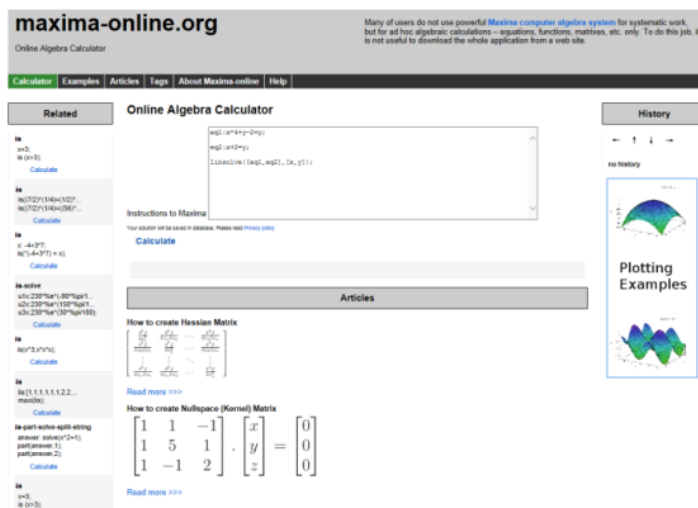
Zdrojový kód Maximy je možné přečíst na mnoha systémech, včetně Windows, Linux a Mac OS X (aktuální operační systém pro počítače Macintosh). Zdrojový kód pro všechny systémy a předkompilované verze pro Windows a Linux jsou k dispozici ve správci souborů na webové stránce programu Maxima.

Základní ovládání programu Maxima:

- Každý příkaz musí končit středníkem - příkaz se může rozdělit na více řádků, provede se až po zadání středníku a stisknutí [Enter], program také rozlišuje klávesu [Enter] na klávesnici, tzv. velký [Enter], a klávesu [Enter] na numerické klávesnici, tzv. malý [Enter]. Malý [Enter] slouží jako zadání středníku a potvrzení příkazu, velký [Enter] jen odřádkuje a uživatel může pokračovat v psaní příkazu na další řádku.
- Nejčastěji se zadávají příkazy v posledním vstupním řádku, který se objeví v textovém okně. Pokud se stiskne klávesa [Enter], aniž by byl napsán ukončovací znak ; na konci řádku, text zůstává modrý a může se pokračovat v psaní příkazu na více řádků. Výsledek příkladu v programu wxMaxima se vypíše v černé barvě.
- Funkce Maximy zabraňuje pokusu o zadání příkazu na špatném místě.
- Struktura příkazů je následující: klíčové slovo, které říká, jaká operace se má provést a do závorek zadat proměnnou, jež chceme vyřešit. Pak napsat středník a stisknout [Enter].
- Proměnnou je možné na začátku symbolicky pojmenovat. Následně se již zadává klíčové slovo a do závorek jen symbol.
- Maxima si ukládá do paměti vždy aktuální hodnotu a s tou pracuje. Pokud chceme pracovat stále s původní hodnotou, musíme ji zadat znovu, nebo si ji označit myší anebo znakem procenta s číslem původní hodnoty.

- Maxima je case sensitive, to znamená, že rozlišuje velká a malá písmena.
- Pro násobení se musí použít znak hvězdička „*“ - $2*a$ je správně, $2a$ je špatně.
- Pro umocňování můžeme používat symbol stříška $^$ nebo dvě hvězdičky $**$.
- Je-li potřeba najít nějaký příkaz v historii, může se použít kombinace kláves Alt-P a Alt-N. Slouží to zejména k obnově předchozích nebo následujících příkazů, které byly zadány. Každé stisknutí těchto kombinací kláves načte předchozí respektive následující příkaz. Takto se může listovat celou historií příkazů, které byly zadány.
- Je-li dosaženo prvního příkazu a opakovaně se stiskne kombinace kláves Alt-P, načte se poslední příkaz, který byl zadán.
- Tyto dvě klíčové kombinace kláves mohou být také použity k hledání předchozího vstupního řetězce s aktuálním řetězcem. Nejprve se zapíše řetězec pro vyhledávání, poté se stiskne Alt-P pro hledání dozadu nebo Alt-N pro hledání dopředu. Stisk těchto kombinací kláves umožňuje procházení všech řádků, které obsahují hledaný řetězec.
- Při zápisu konstanty se musí použít znak %, aby se rozlišily od parametrů a neznámých.

Ve výchozím nastavení Maxima vytváří okna horizontálně a rozděluje je do dvou dílčích oken. První je textové okno, kam se uvádějí všechny příkazy pro program Maxima a jeho vrácené vstupy a výstupy budou vytištěny. Druhé okno prohlížeče Maxima se používá k zobrazení rychlého základu, jako je kopie manuálu Maxima a jakékoli jiné HTML odkazy, které mohou být staženy z webu Maxima.



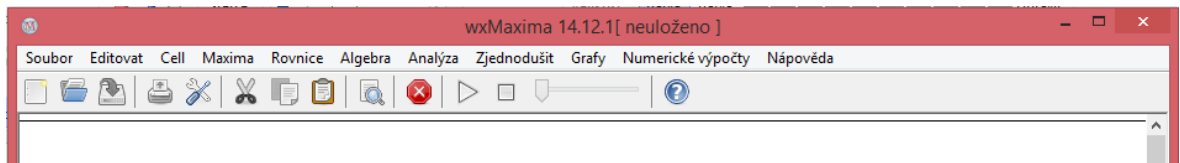
Obrázek 1.1: Maxima – online prohlížeč

Velikost obou dílčích oken lze nastavit přetažením vodorovného pruhu, který je odděluje, pomocí myši. Spodní okno prohlížeče lze odstranit, poté zůstane jen větší textové okno. Může se zpět vrátit pomocí sekce „Toggle Browser zviditelnění“ v menu Volby.

Uživatelé si také mohou vybrat různé typy a velikosti písma v sekci „Preferences“ v menu Volby. Tato nastavení lze uložit pro budoucí relace.

2 Grafické prostředí wxMaxima

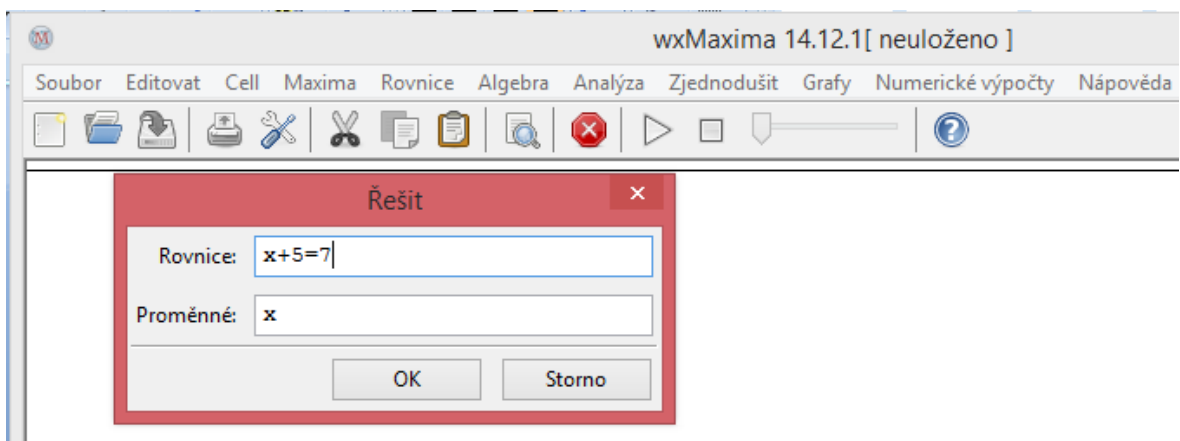
Pro jednodušší ovládání programu wxMaxima je v aplikaci v horním menu rozdělení na jednotlivé podskupiny. Nejpoužívanější jsou rovnice, algebra, analýza, grafy a numerické výpočty.



Obrázek 2.1: wxMaxima – hlavní nabídka a panel nástrojů

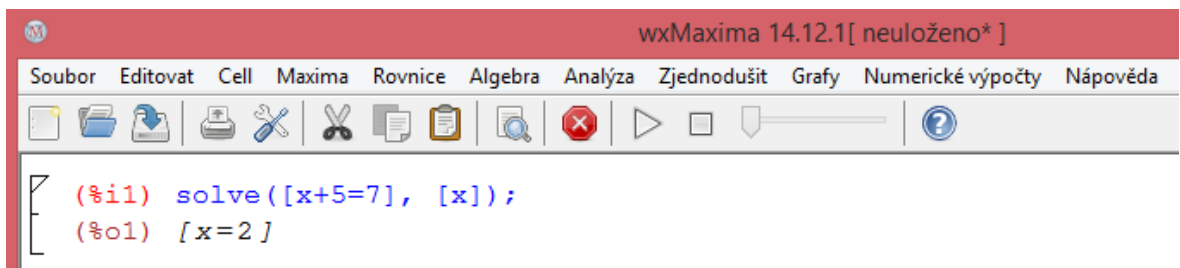
2.1 Rovnice

V hlavní nabídce v podskupině rovnice se nachází možnosti pro zjednodušení zadávání příkazů. Pro vyřešení rovnice se vybere příkaz řešit, zadá se rovnice a její proměnná, ta se potvrdí tlačítkem OK.



Obrázek 2.2: Rovnice zadání

Program wxMaxima sám vyplní vstupní příkaz a vrátí výsledek.

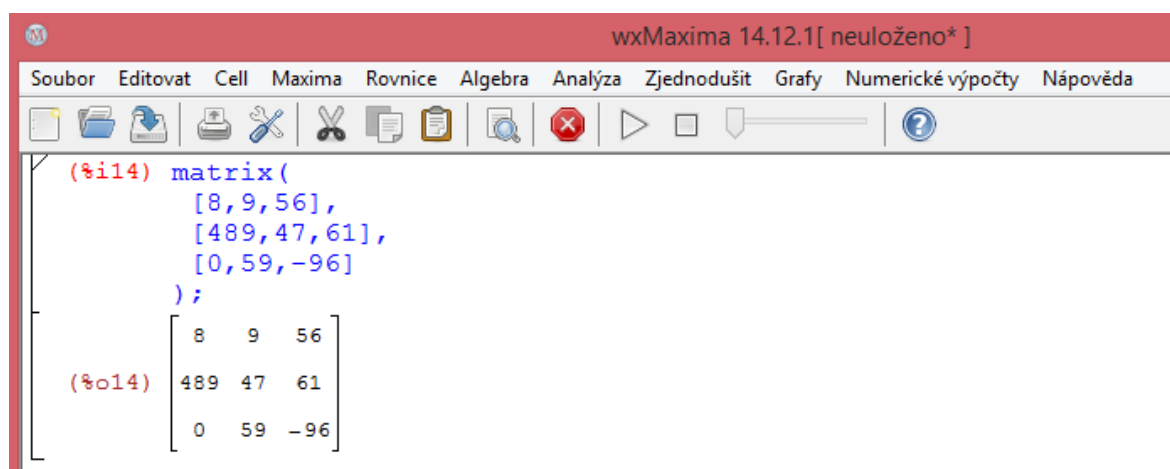


Obrázek 2.3: Rovnice řešení

Dále se pomocí této nabídky může najít kořen rovnice nebo reálné kořeny polynomu. U řešení lineárního systému se zadává počet rovnic. Poté se jednotlivé rovnice vypíší a zadají se proměnné. WxMaxima opět sama zapíše vstupní příkaz a vrátí výsledek. Obdobně je to u řešení algebraického systému.

2.2 Algebra

Zjednodušená nabídka algebry nabízí generování a základní operace s maticemi. Pro zadání matice si uživatel vybere počet řádků a sloupců a po potvrzení zadá jednotlivé hodnoty matice.

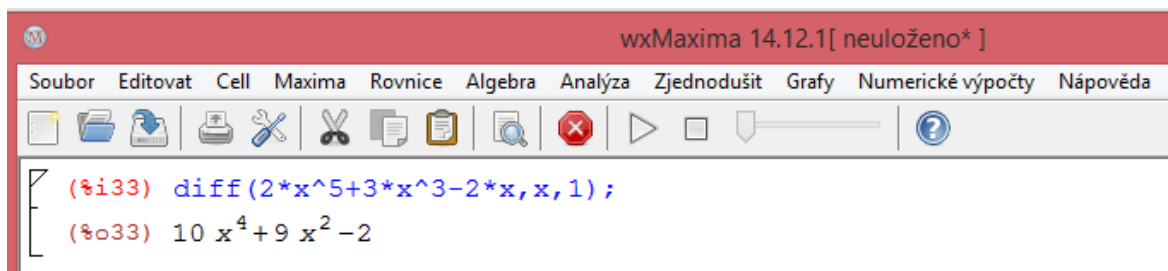


Obrázek 2.4: Algebra – Matice

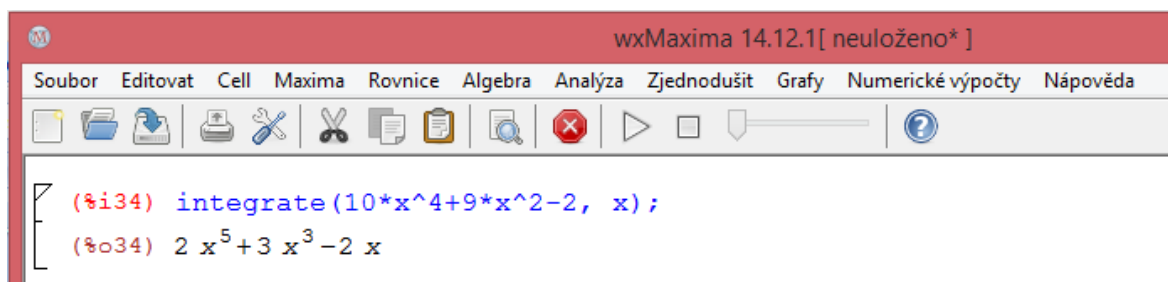
Program snadno vypočte inverzní matici a její determinant nebo matici převede na transponovanou matici.

2.3 Analýza

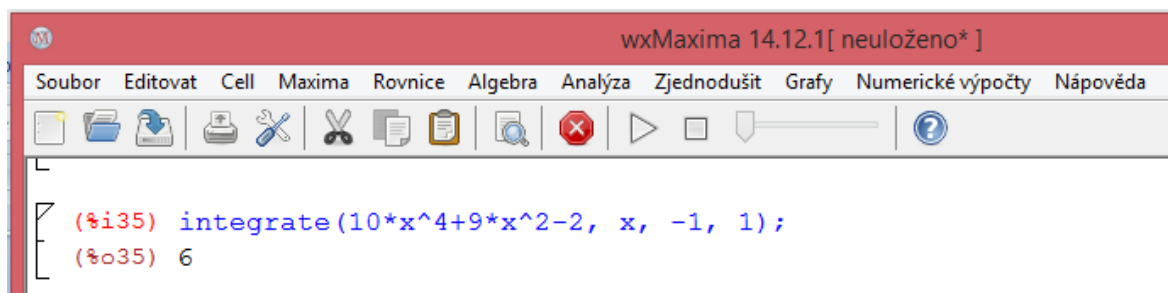
Tato nabídka z panelu nástrojů je nejobsáhlejší a taky nejvíce využitelná. Od základních početních operací jako derivace a integrace, samozřejmostí je výpočet určitého integrálu, přes hledání limit, a to jak v nekonečnu a v bodě, tak i v obou směrech nebo jen zleva a zprava, se dá dále využít pro hledání minima a maxima, pro výpočet sumy nebo součinu, zjištění parciálních zlomků, ale i nalezení největšího společného dělitele a nejmenšího společného násobku.



Obrázek 2.5: Analýza derivace



Obrázek 2.6: Analýza integrace



Obrázek 2.7: Analýza určitý integrál

2.4: Numerické výpočty

Díky této nabídce si může uživatel nastavit novou přesnost výpočtů. V základním nastavení je to šestnáct míst. Dá se nastavit až do hodnoty typu pozitivní integer. Ale aplikaci trvá několik desítek minut až hodinu, než nastaví novou přesnost. Skoro všechny výsledky vrací wxMaxima v desetinných číslech, což je u některých periodických čísel dost nepřehledné. Přepnutím nastavení v numerických výpočtech – Přepnout numerický výstup, se změní výstupní čísla ve zlomky.

```

wxMaxima 14.12.1[ neuloženo* ]
Soubor Editovat Cell Maxima Rovnice Algebra Analýza Zjednoduřit Grafy Numerické výpočty Nápověda

(%i6) solve(x^3-6*x+9=0);
rat: replaced 1.732050807568877 by 37220045/21489003 = 1.732050807568876
rat: replaced 1.732050807568877 by 37220045/21489003 = 1.732050807568876
(%o6) [x=-2.326771511921702 10^-8 (37220045 %i-64467009) , x=2.326771511921702 10^-8 (37220045 %i+64467009) , x
=-3 ]

(%i5) if numer#false then numer:false else numer:true;
(%o5) true

(%i8) solve(x^3-6*x+9=0);
(%o8) [x=- $\frac{\sqrt{3} %i-3}{2}$  , x= $\frac{\sqrt{3} %i+3}{2}$  , x=-3 ]

```

Obrázek 2.8: Přepnutí numerického výstupu

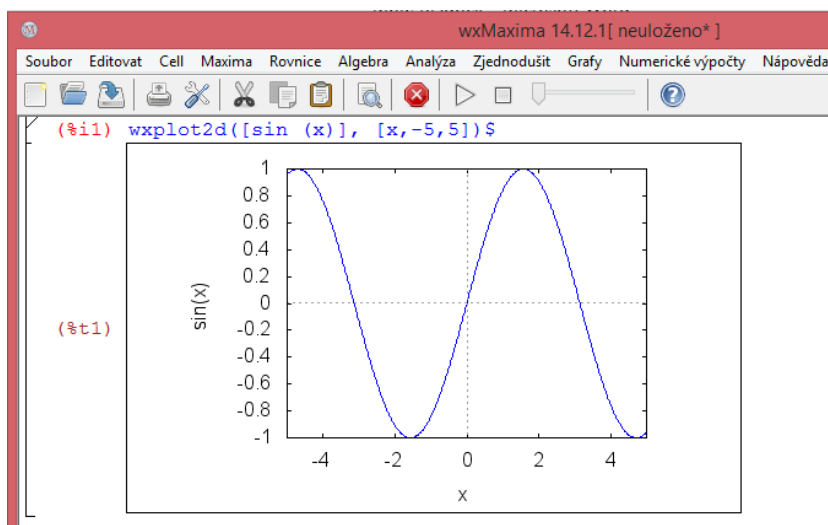
2.5 Grafy

Pro nabídku grafů je stěžejní výběr zobrazení grafu ve 2D nebo 3D formátu. Zadávání je pro oba formáty stejné.

OpenMath grafy

OpenMath grafy lze použít přímo z programu wxMaxima pro vykreslení 2D a 3D grafu. Příkazy pro vykreslení 2D a 3D OpenMath grafů se používají plot2d a plot3d pro formát [plot_format, OpenMath]. Existují i jiné kreslicí funkce v programu wxMaxima, které fungují pouze v OpenMath, a to plotdf, openplot_curves a graph2d.

Lišta menu nabízí různé možnosti pro úpravu grafů. Pomocí config lze změnit některé konfigurace grafu. Po změně konfigurace je třeba aktivovat nové nastavení pomocí replot. Pro přiblížení grafu v určité oblasti se použije zoom.

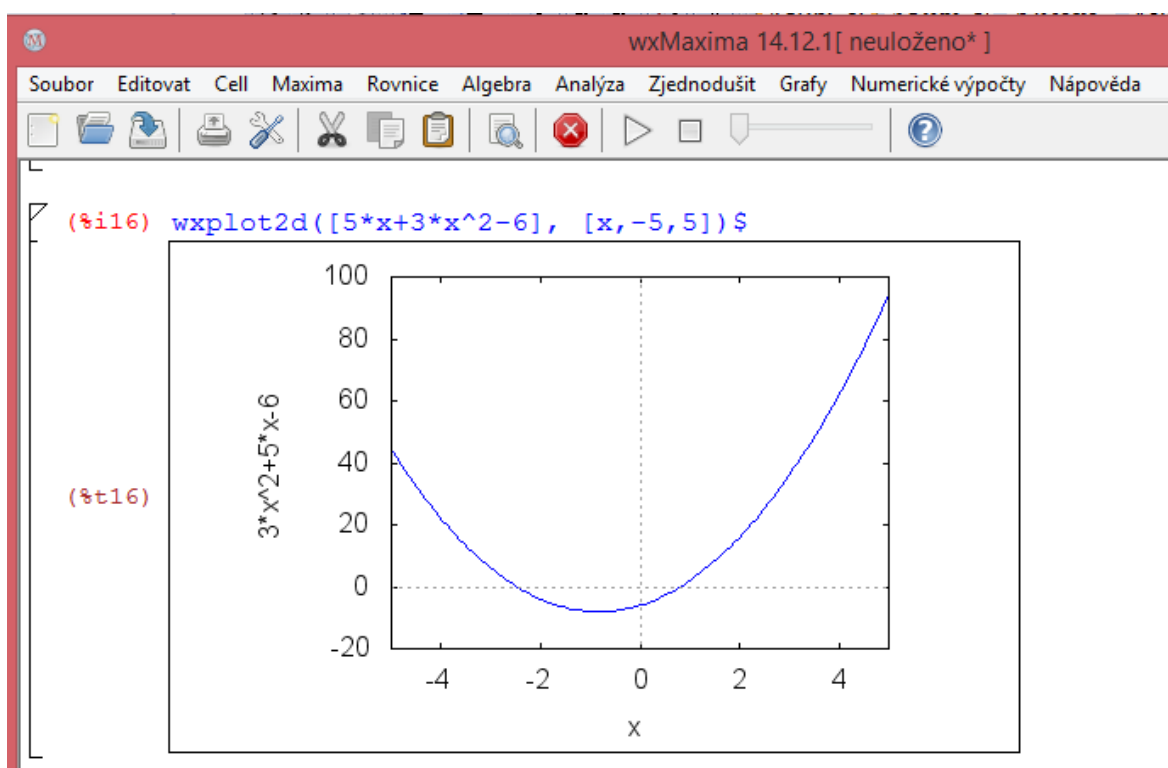


Obrázek 2.9: Graf 2D sin(x)

Ve výchozím nastavení bude každý graf otevřen v novém samostatném okně. Dialogová okna grafů lze vkládat do textového okna X programu wxMaxima pomocí volby Plot v systému Windows: Embended v menu Volby.

2.6 Ovládání pro tvorbu grafů

U dvou aspektů tvorby grafu vyskakují obavy. Prvním problémem je, že kruhový vzhléd rozhodně není sférický. Je to proto, že wxMaxima nepředpokládá, že osa x a osa y mají stejné jednotky, tím se obrázek roztáhne. Svislá osa představuje více jednotek než je podél horizontální osy. Lze to upravit zadáním stejného rozmezí na x-vé a na y-vé souřadnici, poté bude graf symetrický.



Obrázek 2.10: 2D graf

Druhým problémem je legenda u grafu. Chce-li uživatel přidat obsah do legendy, musí přidat volbu legendy. Legenda se objeví automaticky vždy, je-li zadáno dvě a více funkcí.

3D graf má menu velmi podobné 2D grafu, navíc obsahuje jednu dodatečnou možnost. Otočit graf. Při kliknutí na graf není rozšířený jako 2D graf a kliknutím levého tlačítka při tažení myši se bude otáčet graf celý. Uživatel může ovládat dva úhly: azimut a elevaci.

Azimut je orientovaný úhel, který svírá určitý směr se směrem severním. Záleží na směru měření úhlu – měří se po směru pohybu hodinových ručiček. Naměřená hodnota se udává ve stupních.

Elevace označuje výškový úhel paraboly ve stupních při pohledu z horizontu. Nabídka Zoom v menu může změnit výchozí chování myši a slouží ke zvětšení shluku kolem určitého bodu v grafu, jako je tomu v případě 2D grafů. Pokud chce uživatel vrátit výchozí chování, použije tlačítko Otočit.

Pokud spouští uživatel program wxMaxima z příkazového řádku, jako parametr se užívá název souboru. Tento soubor se otevře a interpretuje jako příkaz plot se syntaxí specifik OpenMath. Graf bude vygenerován pomocí příkazů v souboru a bude vykreslen v okně Plot. Hlavní okno wxMaxima se neotevře a nebudou zřízeny žádné komunikace s wxMaxima.

2.11 Prohlížeč

Maxima je aplikace, která je spustitelná na všech webových prohlížečích, což postrádá mnoho funkcí, které jsou k dispozici v každém moderním prohlížeči, jako je například styl listů a podpora tabulky. Na druhé straně má tento prohlížeč dvě další funkce, které nejsou součástí HTML kódu, ale jsou velmi užitečné pro předávání příkazů do wxMaxima z HTML stránky.

První funkce upozorní na zadaný příkaz program wxMaxima a pokud uživatel dvakrát klikne na daný příkaz, bude vyhodnocen, ale jeho výsledek nebude vložen zpět do původní webové stránky. Druhá funkce pracuje podobně jako první, ale výsledek z programu wxMaxima bude nyní vložen zpět do webové stránky do dalšího tagu výsledku, který bude zobrazen na stránce. Příkazy je možné opakovaně editovat.

2.12 Nápověda

Program wxMaxima může být použit pro přístup k dokumentům v několika různých formách. V programu samotném je to nabídka Help, kde je uložena kopie manuálu, který je distribuován programem wxMaxima. V této nabídce je také odkaz na webové stránky Maxima, kde lze nalézt i jiné dokumenty a manuály v různých jazycích.

3 Online webové rozhraní programu Maxima

Vzhledem k širokému využití webové aplikace Maxima budou dále rozebrány jen některé její části – analýza a algebra.

3.1 Maxima online

Originální Maxima on-line webové rozhraní bylo navrženo na základě Jacomax, pro použití počítačového programu. Maxima on-line rozhraní a příkazy jsou stejné jako u wxMaxima. Seznam příkazů nalezne uživatel v nápovědě. Jednoduché příklady algebraických operací, jako jsou rovnice, matice nebo funkce, jsou demonstrovány v sekci příkladů.



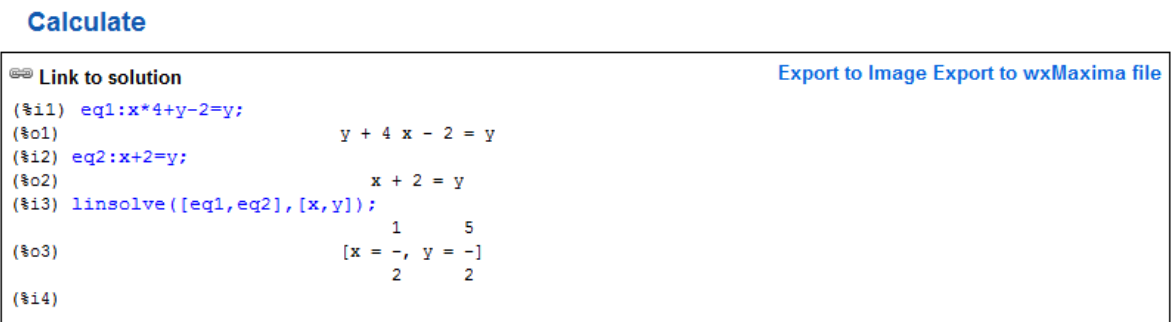
Obrázek 3.1: Maxima online

Online Algebra Calculator

Instructions to Maxima

```
eq1:x*4+y-2=y;  
eq2:x+2=y;  
linsolve([eq1,eq2],[x,y]);
```

Obrázek 3.2: Horní okno prohlížeče - zadání



Obrázek 3.3: Dolní okno prohlížeče – řešení

3.2 Zápis příkazů

Zápis příkazů se hodně podobá programování v Pascalu nebo Delphi.

Příkaz `abs` vrací uživateli absolutní hodnotu čísla nebo výrazu na jeden řádek.



Obrázek 3.4: Absolutní hodnota – výstup

Příkaz `absboxchar` se používá k výpočtu absolutní hodnoty kolem výrazu, který je více než jeden řádek dlouhý.

Příkaz `matrix (1,n)` vrací obdélníkovou matici, která má počet řádků 1 a počet sloupců n . Každý řádek je seznam výrazů. Všechny řádky musí být stejně dlouhé. Operace $+$ (sčítání), $-$ (odčítání), $*$ (násobení), a $/$ (dělení), se provádí pro každý prvek, jehož operandy jsou dvě matice, skalár a matice, nebo matice a skalár. Operace $^$ (umocňování, ekvivalentně `**`) se provádí pro každý prvek, jestliže operandy jsou skalár a matice nebo matice a skalár. Ne v případě, že jsou dva operandy matice. Všechny operace jsou prováděny v plném rozsahu. Existuje mnoho funkcí, které lze použít pro zjednodušení příkazů matic. Jedná se například o `addmatrices`, `eigenvectors`, `determinant`, `charpoly`, `genmatrix`, `addcol`, `addrow`, `copymatrix`, `transpose`, `echelon` a `rank`.

Příkaz `determinant` vypočítá determinant způsobem podobným Gaussově eliminaci.

Hodnost matice se vypočte příkazem `rank`. `Rank` může vrátit špatnou odpověď, pokud nelze určit, zda matice obsahuje lineárně závislé vektory. Analogicky `transpose` vrací uživateli transponovanou matici k aktuální matici.

Calculate

[↔ Link to solution](#)

```
(%i1) x: matrix ([17, 3], [-8, 11]);
                                [ 17  3 ]
(%o1)                                [      ]
                                [ - 8 11 ]
(%i2) y: matrix ([%pi, %e], [a, b]);
                                [ %pi %e ]
(%o2)                                [      ]
                                [ a  b ]
(%i3) x + y;
                                [ %pi + 17 %e + 3 ]
(%o3)                                [      ]
                                [ a - 8  b + 11 ]
(%i4) x - y;
                                [ 17 - %pi 3 - %e ]
(%o4)                                [      ]
                                [ - a - 8 11 - b ]
(%i5) x * y;
                                [ 17 %pi 3 %e ]
(%o5)                                [      ]
                                [ - 8 a 11 b ]
(%i6) x / y;
                                [ 17      - 1 ]
                                [ --- 3 %e ]
                                [ %pi      ]
(%o6)                                [      ]
                                [ 8 11 ]
                                [ - - -- ]
                                [ a b ]
(%i7)
```

Obrázek 3.5: Početní operace s maticemi 1 – výstup

Calculate

[Link to solution](#)

```
(%i1) x: matrix ([17, 3], [-8, 11]);
      [ 17  3 ]
(%o1) [      ]
      [- 8 11 ]
(%i2) y: matrix ([%pi, %e], [a, b]);
      [ %pi %e ]
(%o2) [      ]
      [ a  b ]
(%i3) x ^ 3;
      [ 4913  27 ]
(%o3) [      ]
      [- 512 1331 ]
(%i4) exp(y);
      [ %pi %e ]
      [ %e %e ]
(%o4) [      ]
      [ a  b ]
      [ %e %e ]
(%i5) x ^ y;
      [ %pi %e ]
      [      ]
      [ a  b ]
      [ 17  3 ]
(%o5) [      ]
      [- 8 11 ]
(%i6) x . y;
      [ 3 a + 17 %pi  3 b + 17 %e ]
(%o6) [      ]
      [ 11 a - 8 %pi  11 b - 8 %e ]
(%i7) y . x;
      [ 17 %pi - 8 %e  3 %pi + 11 %e ]
(%o7) [      ]
      [ 17 a - 8 b  11 b + 3 a ]
(%i8)
```

Obrázek 3.6: Početní operace s maticemi 2 – výstup

Calculate

```
Link to solution
(%i1) x: matrix ([17, 3], [-8, 11]);
      [ 17  3 ]
(%o1) [      ]
      [- 8 11]
(%i2) y: matrix ([%pi, %e], [a, b]);
      [ %pi %e ]
(%o2) [      ]
      [ a  b ]
(%i3) x ^^ 3;
      [ 3833 1719 ]
(%o3) [      ]
      [- 4584 395 ]
(%i4) %e ^^ y;
      [ %pi %e ]
      [ %e %e ]
(%o4) [      ]
      [ a  b ]
      [ %e %e ]
(%i5) x ^^ -1;
      [ 11  3 ]
      [ --- - --- ]
      [ 211 211 ]
(%o5) [      ]
      [ 8  17 ]
      [ --- --- ]
      [ 211 211 ]
(%i6) x . (x ^^ -1);
      [ 1 0 ]
(%o6) [      ]
      [ 0 1 ]
(%i7)
```

Obrázek 3.7: Početní operace s maticemi 3 – výstup

Příkaz `addmatrices` se hodí pro různé operace s maticemi, navíc do proměnné je možné zapsat libovolný počet argumentů. Struktura příkazu: klíčové slovo (požadovaná operace, výpis symbolů zastupujících matice). Snadno se porovnávají jednotlivé hodnoty matic.

Instructions to Maxima

```
m1 : matrix([1,2],[3,4])$
m2 : matrix([7,8],[9,10])$
addmatrices(max,m1,m2);
addmatrices(max,m1,m2,5*m1);
```

Obrázek 3.8: Součet matic - vstup

☞ **Link to solution**

```
(%i1) m1 : matrix([1,2],[3,4]);
      [ 1 2 ]
(%o1)  [      ]
      [ 3 4 ]
(%i2) m2 : matrix([7,8],[9,10]);
      [ 7 8 ]
(%o2)  [      ]
      [ 9 10 ]
(%i3) addmatrices(max,m1,m2);
      [ 7 8 ]
(%o3)  [      ]
      [ 9 10 ]
(%i4) addmatrices(max,m1,m2,5*m1);
      [ 7 10 ]
(%o4)  [      ]
      [ 15 20 ]
(%i5)
```

Obrázek 3.9: Součet matic – výstup

Potřebuje – li uživatel matici převést na diagonální, použije příkaz `diag_matrix`.

Calculate

```
☞ Link to solution
(%i1) diag_matrix(diag_matrix(1,2),diag_matrix(3,4));
      [ [ 1 0 ] [ 0 0 ] ]
      [ [      ] [      ] ]
      [ [ 0 2 ] [ 0 0 ] ]
(%o1)  [      ]
      [ [ 0 0 ] [ 3 0 ] ]
      [ [      ] [      ] ]
      [ [ 0 0 ] [ 0 4 ] ]
(%i2) diag_matrix(p,q);
      [ p 0 ]
(%o2)  [      ]
      [ 0 q ]
(%i3)
```

Obrázek 3.10: Diagonální matice - výstup

Příkaz `solve` řeší algebraické rovnice s jednou proměnnou. Výraz může být racionální nebo může obsahovat goniometrické funkce anebo exponenciály. Systém polynomiálních rovnic (lineárních nebo nelineárních) se řeší použitím příkazu `linsolve` nebo `algsys`. Struktura příkazu: klíčové slovo (výpis všech rovnic, výpis všech proměnných);

Calculate

Link to solution
Export to

```

(%i1) solve (asin (cos (3*x))*(f(x) - 1), x);

solve: using arc-trig functions to get a solution.
Some solutions will be lost.

(%o1)

$$[x = \frac{\pi}{6}, f(x) = 1]$$

(%i2) ev (solve (5^f(x) = 125, f(x)), solveradcan);
(%o2)

$$[f(x) = \frac{\log(125)}{\log(5)}]$$

(%i3) [4*x^2 - y^2 = 12, x*y - x = 2];
(%o3)

$$[4x^2 - y^2 = 12, xy - x = 2]$$

(%i4) solve (% , [x, y]);
(%o4)

$$[]$$

(%i5) solve (x^3 - 1);
(%o5)

$$[x = \frac{\sqrt{3}i - 1}{2}, x = \frac{\sqrt{3}i + 1}{2}, x = 1]$$

(%i6) solve (x^6 - 1);
(%o6)

$$[x = \frac{\sqrt{3}i + 1}{2}, x = \frac{\sqrt{3}i - 1}{2}, x = -1, x = \frac{\sqrt{3}i + 1}{2}, x = -\frac{\sqrt{3}i - 1}{2}, x = 1]$$

(%i7) expand (%);
(%o7)

$$x^2 - 1;$$

solve (% , x);
(%o8)

$$x^2 - 1$$

(%i9) ev (%th(2), %[1]);
(%o9)

$$[]$$

(%i10)

```

Obrázek 3.11: Algebraické rovnice – výstup

Příkaz `linsolve` řeší soustavu lineárních rovnic pro n proměnných. V případě, že by příkaz `linsolve` obsahoval jeden argument řešení, nastává problém. Příkaz trvá na dvou argumentech. První představuje rovnice, které je třeba řešit. Druhý je seznam neznámých, jež budou stanoveny. V případě, že celkový počet proměnných v rovnicích je roven počtu rovnic, může být druhý argument, tedy seznam neznámých, vynechán.

Calculate

```

@@ Link to solution
(%i1) e1: x + z = y;
(%o1) z + x = y
(%i2) e2: 2*a*x - y = 2*a^2;
(%o2) 2 a x - y = 2 a2
(%i3) e3: y - 2*z = 2;
(%o3) y - 2 z = 2
(%i4) [globalsolve: false, programmode: true];
(%o4) [false, true]
(%i5) linsolve ([e1, e2, e3], [x, y, z]);
(%o5) [x = a + 1, y = 2 a, z = a - 1]
(%i6) [globalsolve: false, programmode: false];
(%o6) [false, false]
(%i7) linsolve ([e1, e2, e3], [x, y, z]);
Solution:
(%t7) z = a - 1
(%t8) y = 2 a
(%t9) x = a + 1
(%o9) [%t7, %t8, %t9]
(%i10)

```

Obrázek 3.12: Soustava lineárních rovnic – výstup

Příkaz `algsys` řeší současně polynomy nebo polynomiální rovnice pro n proměnných. Může být více rovnic než veličin a naopak. `Algsys` vrátí seznam řešení s uvedením hodnot proměnných x , které splňují soustavu rovnic. Pokud `algsys` nemůže najít řešení, vrátí prázdný seznam. V případě že `algsys` vytváří řešení, které má méně platných číslic než je požadováno, může uživatel změnit hodnotu `algepsilon` na vyšší přesnost. Jestliže `algsys` narazí s proměnnými na rovnici, která obsahuje plovoucí řádovou čárkou přiblížení (protože se obvykle nedaří nalézt přesné řešení v raném stadiu), nebude se Maxima pokoušet aplikovat přesné metody pro tyto rovnice a místo toho vytiskne zprávu: "algsys nemůže vyřešit - systém příliš složitý".

```

e1: 2*x*(1 - a1) - 2*(x - 1)*a2;
e2: a2 - a1;
e3: a1*(-y - x^2 + 1);
e4: a2*(y - (x - 1)^2);
algsys ([e1, e2, e3, e4], [x, y, a1, a2]);
e1: x^2 - y^2;
e2: -1 - y + 2*y^2 - x + x^2;
algsys ([e1, e2], [x, y]);

```

Obrázek 3.13: Algebraický systém - vstup

Calculate

[Link to solution](#)

[Export](#)

```
(%i1) e1: 2*x*(1 - a1) - 2*(x - 1)*a2;
(%o1) 2 (1 - a1) x - 2 a2 (x - 1)
(%i2) e2: a2 - a1;
(%o2) a2 - a1
(%i3) e3: a1*(-y - x^2 + 1);
(%o3) a1 (- y - x^2 + 1)
(%i4) e4: a2*(y - (x - 1)^2);
(%o4) a2 (y - (x - 1)^2)
(%i5) algsys ([e1, e2, e3, e4], [x, y, a1, a2]);
(%o5) [[x = 0, y = %r1, a1 = 0, a2 = 0], [x = 1, y = 0, a1 = 1, a2 = 1]]
(%i6) e1: x^2 - y^2;
(%o6) x^2 - y^2
(%i7) e2: -1 - y + 2*y^2 - x + x^2;
(%o7) 2 y^2 - y + x^2 - x - 1
(%i8) algsys ([e1, e2], [x, y]);
(%o8) [[x = - 1/sqrt(3), y = - 1/sqrt(3)], [x = 1/sqrt(3), y = - 1/sqrt(3)],
[x = - 1/3, y = - 1/3], [x = 1, y = 1]]
(%i9)
```

Obrázek 3.14: Algebraický systém – výstup

Příkaz `allroots` vypočítá numerické aproximace reálných a komplexních kořenů polynomu nebo polynomické rovnice jedné proměnné. `Allroots` může poskytovat nepřesné výsledky v případě vícenásobných kořenů.

Příkaz `desolve` řeší soustavy obyčejných lineárních diferenciálních rovnic pomocí Laplaceovy transformace. V programu Maxima jsou diferenciální rovnice v závislých proměnných. Funkční závislost na nezávislé proměnné, například x , musí být výslovně uvedeno v proměnných. Je-li $x = 0$, jsou známé počáteční podmínky, které mohou být dodány před voláním `desolve` pomocí příkazu `atvalue`. V tomto případě Maxima nezvládne výpočet a program vrátí `error`.

Calculate

[Link to solution](#)

```
(%i1) diff(f(x),x)=diff(g(x),x)+sin(x);
      d      d
(%o1)  -- (f(x)) = -- (g(x)) + sin(x)
      dx      dx
(%i2) diff(g(x),x,2)=diff(f(x),x)-cos(x);
      2
      d      d
(%o2)  --- (g(x)) = -- (f(x)) - cos(x)
      2      dx
(%i3) atvalue(diff(g(x),x),x=0,a);
(%o3)  a
(%i4) atvalue(f(x),x=0,1);
(%o4)  1
(%i5) desolve([%o1,%o2],[f(x),g(x)]);

desolve: can't handle this case.
-- an error. To debug this try: debugmode(true);
(%i6) [%o1,%o2],%o5,diff;

Improper argument to ev:
%o5
-- an error. To debug this try: debugmode(true);
(%i7)
```

Obrázek 3.15: Obyčejné diferenciální rovnice – výstup

Příkaz `diff` vrací derivaci nebo diferenciál s ohledem na některé nebo všechny proměnné. Struktura příkazu: Klíčové slovo `diff` (výraz, který se bude derivovat, proměnná, podle které se derivuje);. `Diff(výraz, x1, n1, ..., xm, nm)`; vrátí smíšené parciální derivace s ohledem na x₁, ..., x_m. `Diff(výraz, x, n)`; vrátí n-tou derivaci s ohledem na x. `Diff(výraz, x)`; vrátí první derivaci vzhledem k proměnné x. `Diff(výraz)`; vrací totální diferenciál (tj. aplikovaný na funkci několika proměnných). Podstatné jméno ve formě `diff` je nutné uvádět v některých z dalších výpočtů, jako je tomu u diferenciálních rovnic.

Příkaz `derivabbrev` má nastavenou výchozí hodnotu na `false`, tedy nepravda. Je-li `derivabbrev` nastaven na hodnotu `true`, tedy pravda, symboly derivací jsou zobrazeny jako indexy. V opačném případě jsou derivace zobrazeny v Leibnizově notaci dy/dx .

Calculate

```
Link to solution
(%i1) diff (exp (f(x)), x, 2);
(%o1)
      f(x) d      f(x) d
      (--- (f(x))) + %e (--- (f(x)))
      2      dx
      dx
(%i2) derivabbrev: true;
(%o2) true
(%i3) 'integrate (f(x, y), y, g(x), h(x));
(%o3)
      h(x)
      /
      [
      I   f(x, y) dy
      ]
      /
      g(x)
(%i4) diff (%o3, x);
(%o4) 0
(%i5)
```

Obrázek 3.16: Derivace – výstup

Příkaz `integrate` se pokusí vypočítat integrál podle zadané proměnné. Struktura příkazu: `integrate (výraz, proměnná, podle které integrujeme)` je neurčitý integrál a `integrate (výraz, proměnná, podle které integrujeme, a, b)` je určitý integrál, kde hodnota `a` a hodnota `b` označují meze integrálu. Meze by neměly obsahovat proměnnou `x`, i když příkaz `integrate` toto omezení nevyžaduje. Hodnota `a` nemusí být menší než hodnota `b`. Je-li `b` rovno `a`, `integrate` vrátí nulu. Integrál (výraz bez integrace) je vrácen, pokud `integrate` uspěje. `integrate` se zabývá výpočtem určitých integrálů odděleně od neurčitých. Zvláštní případy určitých integrálů jsou u mezí integrace rovno nule nebo nekonečno (`inf` nebo `minf`), goniometrické funkce, jejichž meze integrace se rovnají nule a π nebo 2π , racionální funkce, integrály týkající se definice beta a psi funkce, a některé logaritmické a goniometrické integrály. Zpracování racionální funkce může zahrnovat výpočet zbytků.

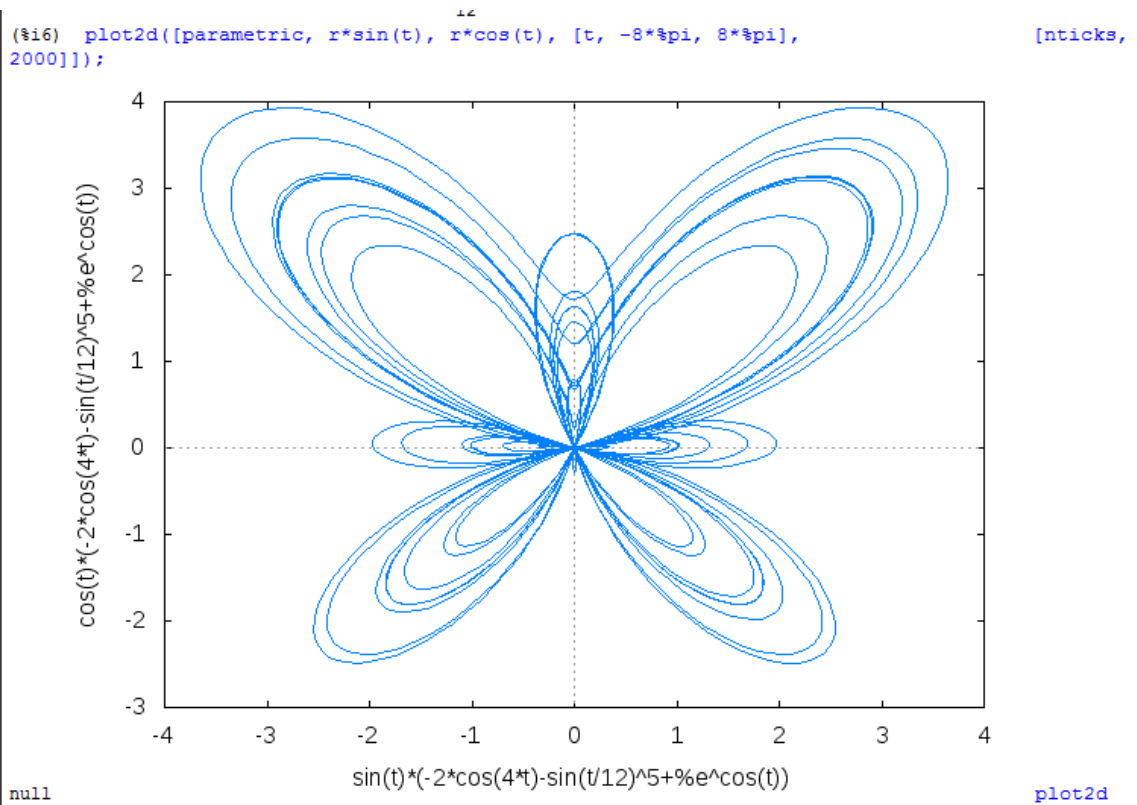
Zvláštní případy neurčitých integrálů jsou goniometrické funkce, exponenciální a logaritmické funkce a racionální funkce. `integrate` vypočítá metody integrací *per partes* jen v několika málo zvláštních případech.

Calculate

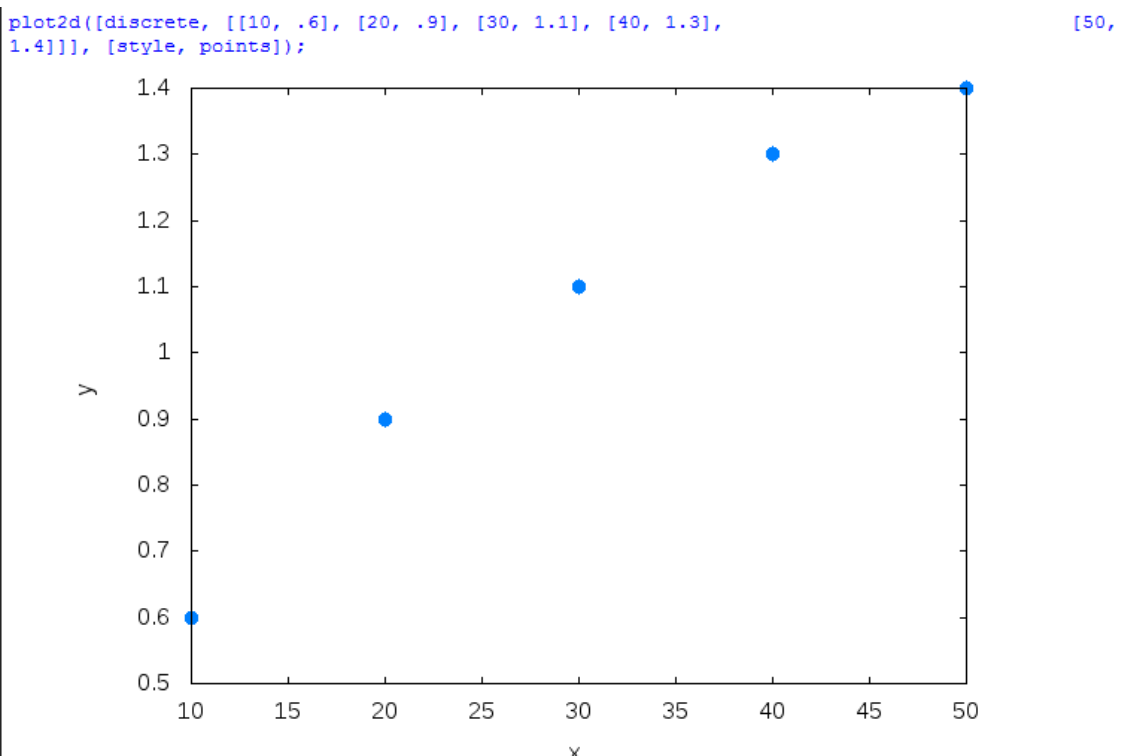
```
🔗 Link to solution
(%i1) integrate (sin(x)^3, x);
(%o1)
      3
      cos (x)
      ----- - cos(x)
      3
(%i2) integrate (x/ sqrt (b^2 - x^2), x);
(%o2)
      2      2
      - sqrt(b  - x )
(%i3) integrate (cos(x)^2 * exp(x), x, 0, %pi);
(%o3)
      %pi
      3 %e      3
      ----- -
      5      5
(%i4) integrate (x^2 * exp(-x^2), x, minf, inf);
(%o4)
      sqrt(%pi)
      -----
      2
(%i5)
```

Obrázek 3.17: Integrál – výstup

Příkaz `plot2d` zobrazuje jeden nebo více grafů ve dvou rozměrech. Graf je možné definovat také v diskretních nebo parametrických formách. Diskrétní forma je používána k vykreslení souboru bodů se zadanými souřadnicemi. Diskrétní graf je definován jedním nebo dvěma seznamy hodnot. Jsou-li uvedeny dva seznamy, musí mít stejnou délku. První seznam bude interpretován jako souřadnice bodů x a druhý seznam jako souřadnice y . Pokud by byl uveden jen jeden seznam, každý prvek musí mít v seznamu dvě hodnoty, které odpovídají x -vé a y -vé souřadnice bodu. Parametrický graf je definován dvěma výrazy pro názvy funkcí a rozsahu pro parametr. Rozsah parametru musí být v rozmezí jeho minimální a maximální hodnoty. Rozmezí svislé osy je volitelný argument. Není-li rozsah na svislé ose uveden, bude zřízen v souladu s minimální a maximální hodnotou danou x -vou souřadnicí bodů. Pokud bude vykresleno více grafů než jeden, použije se legenda pro jednotlivé grafy.

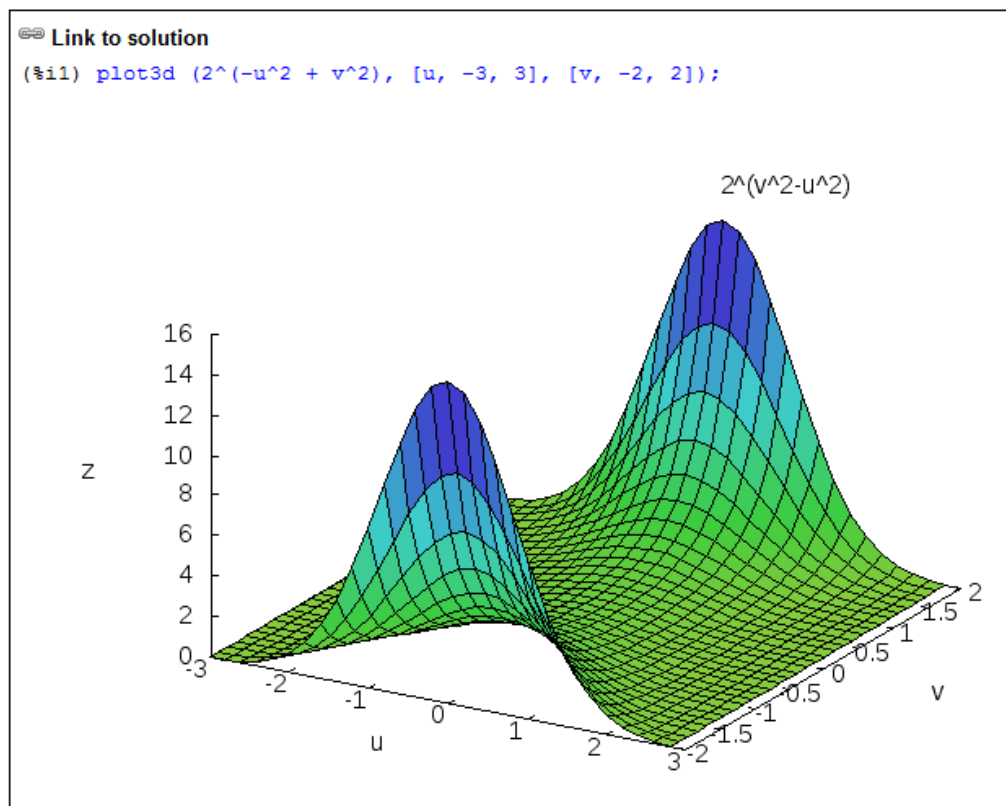


Obrázek 3.18: Parametrický 2D graf – výstup



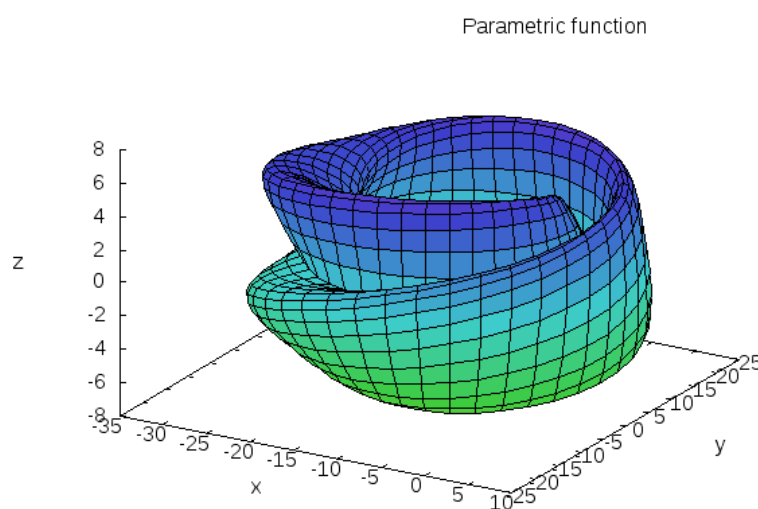
Obrázek 3.19: Diskrétní 2D graf – výstup

Příkaz `plot3d` zobrazuje graf ve třech rozměrech. Mezi funkce, které budou vyneseny do grafu, mohou být výrazy nebo názvy funkcí. Myši lze otáčet graf na pohled z různých stran.



Obrázek 3.20: 3D graf – výstup

```
(%i5) plot3d ([e_1, e_2, e_3], [x, -%pi, %pi], [y, -%pi, %pi], [grid, 40, 40]);
```



Obrázek 3.21: Parametrický 3D graf – výstup

Část III

Praktická část

4 Ukázka řešených příkladů

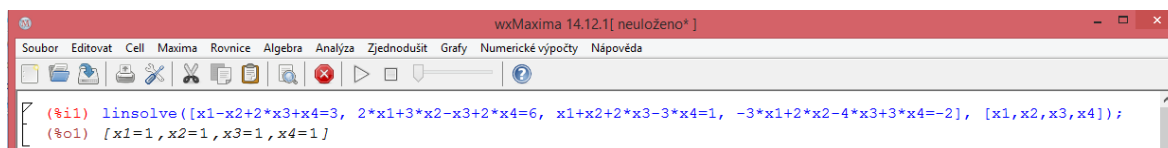
V praktické části bych ráda použila stejné zadání některých příkladů, jako měla slečna Gabriela Kaufnerová a provedla srovnání výpočtů v aplikaci wxMaxima a ve Wolfram Alpha, který zpracovala již zmíněná slečna Kaufnerová. Důvodem pro srovnání je široké použití v oblasti matematiky, volná dostupnost na internetu a je otázkou, která z aplikací zvládne více věcí vyřešit.

Zadání příkladů 4.1, 4.2, 4.3, 4.4, 4.5 a 4.6 byla převzata z bakalářské práce Gabriely Kaufnerové a původ jde dohledat v její práci. Název bakalářské práce: VYUŽITÍ WOLFRAMU ALPHA V MATEMATICE, práce byla obhájená 19. května 2014.

Příklad 4.1: Určení všech řešení soustavy lineárních rovnic I

Zadání: „Určete všechna řešení soustavy lineárních rovnic $x_1 - x_2 + 2x_3 + x_4 = 3$, $2x_1 + 3x_2 - x_3 + 2x_4 = 6$, $x_1 + x_2 + 2x_3 - 3x_4 = 1$, $-3x_1 + 2x_2 - 4x_3 + 3x_4 = -2$ “

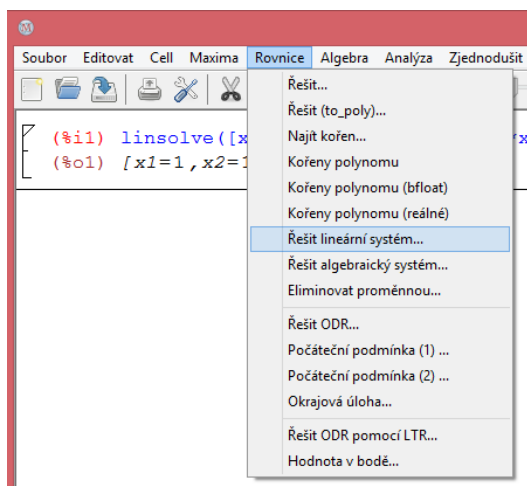
Řešení:



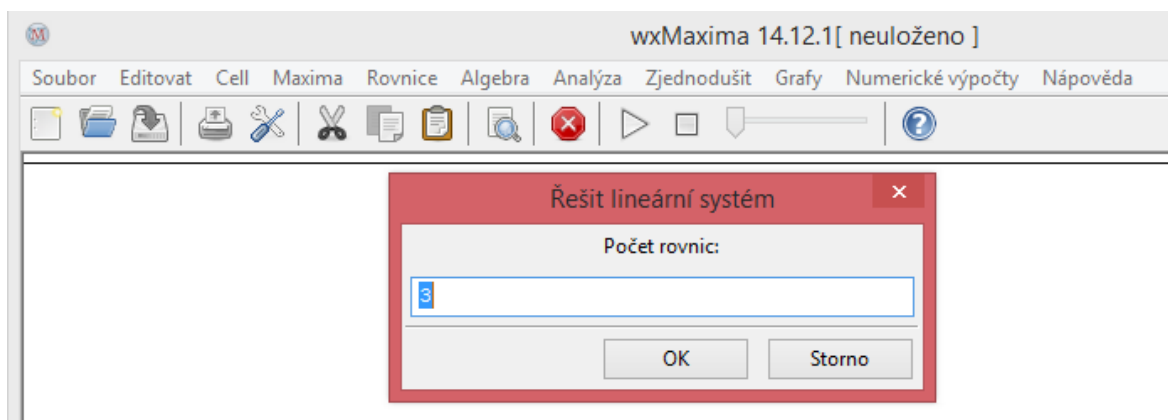
```
wxMaxima 14.12.1[ neuloženo* ]
Soubor  Editovat  Cell  Maxima  Rovnice  Algebra  Analýza  Zjednodušit  Grafy  Numerické výpočty  Nápověda
[ (%i1) linsolve([x1-x2+2*x3+x4=3, 2*x1+3*x2-x3+2*x4=6, x1+x2+2*x3-3*x4=1, -3*x1+2*x2-4*x3+3*x4=-2], [x1,x2,x3,x4]);
(%o1) [x1=1, x2=1, x3=1, x4=1]
```

Obrázek 4.1: Výsledek soustavy lineárních rovnic I

Soustava lineárních rovnic má jedno řešení a to $[1,1,1,1]$, jak je vidět na obrázku 4.1. V programu wxMaxima se použije příkaz linsolve, nebo je možné použít hlavní nabídku, jak je zachyceno na následujících obrázcích.

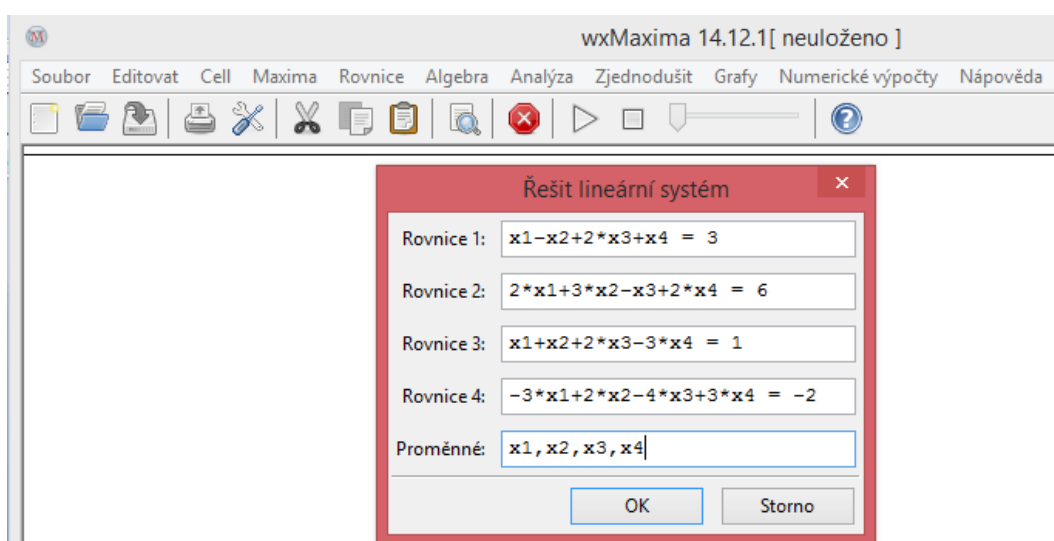


Obrázek 4.2: Výběr Rovnice – Řešit lineární systém



Obrázek 4.3: Výběr počtu rovnic s defaultní hodnotou 3

Nastavíme počet na čtyři a stiskneme tlačítko OK.



Obrázek 4.4: Podokno – zadané lineární rovnice a proměnné

Po zadání lineárních rovnic opět stiskneme tlačítko OK a necháme program vyřešit soustavu. Nesmíme zapomenout na psaní znaku * místo násobení. Pokud znak vynecháme, program soustavu lineárních rovnic nevypočítá.

Příklad 4.2: Určení všech řešení soustavy lineárních rovnic II

Zadání: „Určete všechna řešení soustavy lineárních rovnic $x_1 + 2x_2 - 4x_3 + 3x_4 + x_5 = -5$, $2x_1 + 4x_2 - 11x_3 + 13x_4 = -13$, $x_1 + 2x_2 - 3x_3 + 2x_4 + 7x_5 = 4$, $-3x_1 - 6x_2 + 13x_3 - 9x_4 + 7x_5 = 30$, $-2x_1 - 4x_2 + 9x_3 - 11x_4 - 7x_5 = 20$ “.

Řešení: Tato soustava obsahuje lineárně závislou rovnici, kterou program vypustí. WxMaxima vrátí komentář o eliminaci rovnice a za ní vypíše výsledek s parametrem. Výsledkem soustavy lineárních rovnic je tedy uspořádaná pětice výrazů $[-2t-108, t, -35, -14, 5]$, kde t je parametr. Soustava má nekonečně mnoho řešení v závislosti na parametru t .

Postup pro zadání rovnic do programu je obdobný jako v příkladu 4.1.

```

wxMaxima 14.12.1[ neuloženo* ]
Soubor Editovat Cell Maxima Rovnice Algebra Analýza Zjednodušit Grafy Numerické výpočty nápověda
--> linsolve([x1+2*x2-4*x3+3*x4+x5=-5, 2*x1+4*x2-11*x3+13*x4=-13, x1+2*x2-3*x3+2*x4+7*x5=4,
-3*x1-6*x2+13*x3-9*x4+7*x5=30, -2*x1-4*x2+9*x3-11*x4-7*x5=20], [x1, x2, x3, x4, x5]);
solve: dependent equations eliminated: (4)
(%o1) [x1=-2 %r1-108, x2=%r1, x3=-35, x4=-14, x5=5]

```

Obrázek 4.5: Řešení soustavy lineárních rovnic II

Příklad 4.3: Určení vlastních čísel, vlastních vektorů, Jordanův kanonický tvar J matice a ověření rovnosti $A = TJT^{-1}$.

Zadání: „Určete vlastní čísla, vlastní vektory a Jordanův kanonický tvar J matice

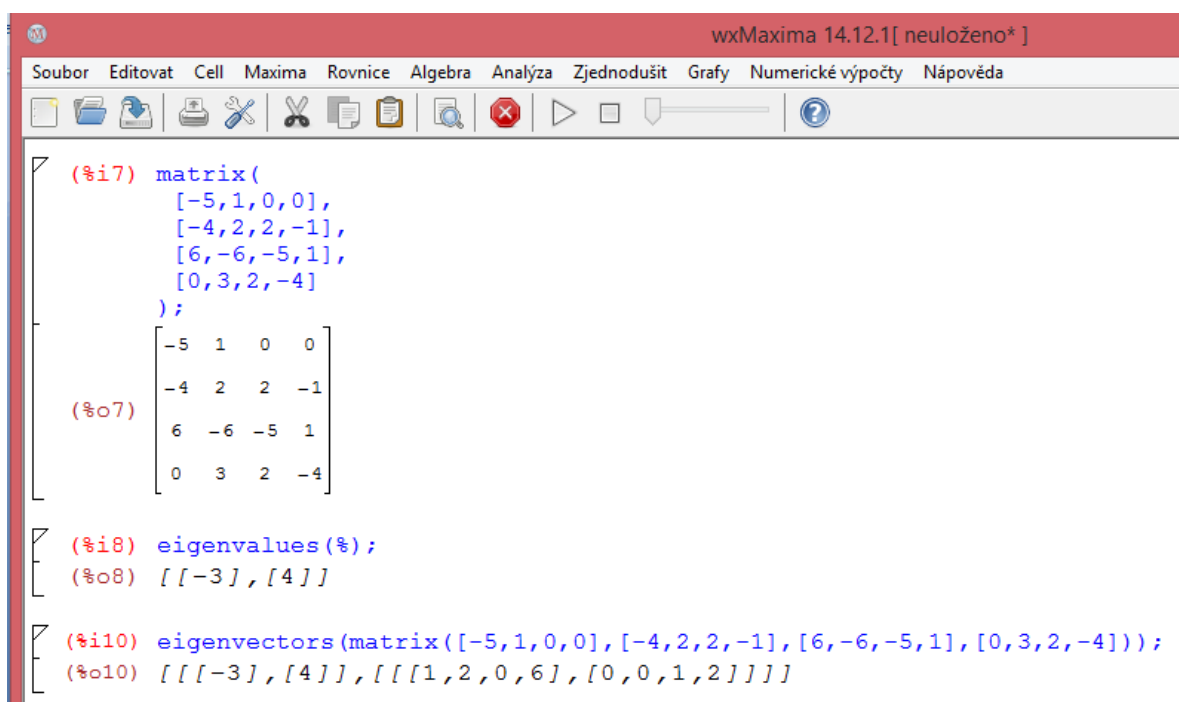
$$A = \begin{pmatrix} -5 & 1 & 0 & 0 \\ -4 & 2 & 2 & -1 \\ 6 & -6 & -5 & 1 \\ 0 & 3 & 2 & -4 \end{pmatrix}$$

Ověřte, že platí rovnost $A = TJT^{-1}$.

Řešení: Nejprve si spočteme vlastní čísla a vlastní vektory zadané matice. V hlavní nabídce klikneme na Algebra a vybereme vlastní hodnoty a vlastní vektory.

Nebo se můžeme použít příkaz `eigenvalues` pro zjištění vlastních čísel a příkaz `eigenvectors` pro vlastní vektory. Za oběma příkazy musíme uvést do závorek příslušnou matici.

V případě vlastních čísel nám program wxMaxima vrátil hodnoty `[-3]` a `[4]`. Kde první hodnota, může jich být více, znamená vlastní číslo a druhá hodnota určuje, kolikanásobné vlastní číslo je. Druhých hodnot může být také více. U vlastních vektorů program vrátí napřed vlastní čísla a jejich násobné hodnoty a dále až vlastní vektory. Zajímavé je, že vrátí jiné hodnoty než wolfram alpha. WxMaxima udává vlastní vektory v jiné bázi. Můžeme je přepočítat na stejné hodnoty, jaké udává wolfram alpha.

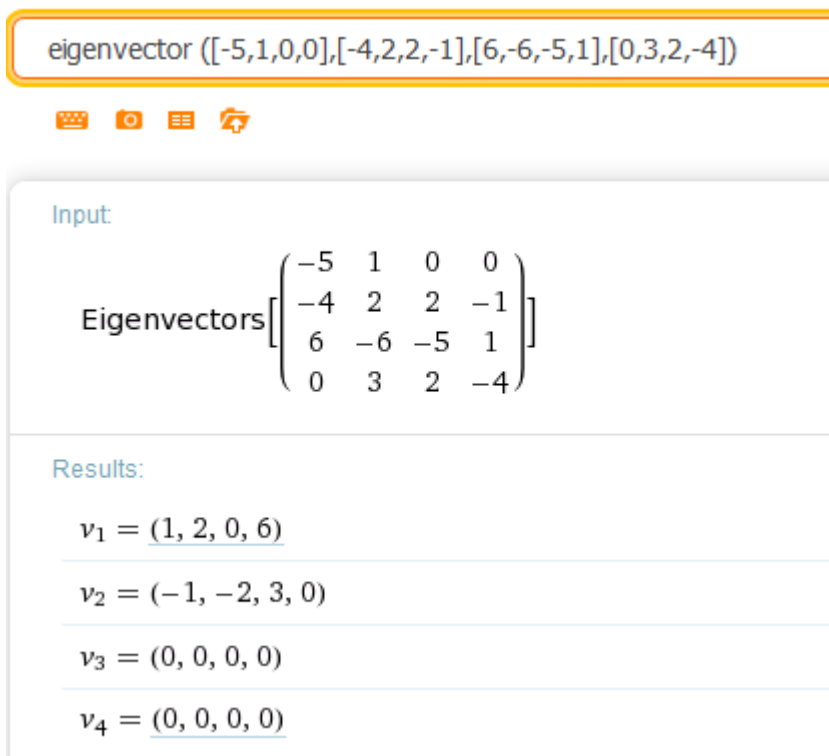


```
(%i7) matrix(
  [-5,1,0,0],
  [-4,2,2,-1],
  [6,-6,-5,1],
  [0,3,2,-4]
);
(%o7)

$$\begin{bmatrix} -5 & 1 & 0 & 0 \\ -4 & 2 & 2 & -1 \\ 6 & -6 & -5 & 1 \\ 0 & 3 & 2 & -4 \end{bmatrix}$$

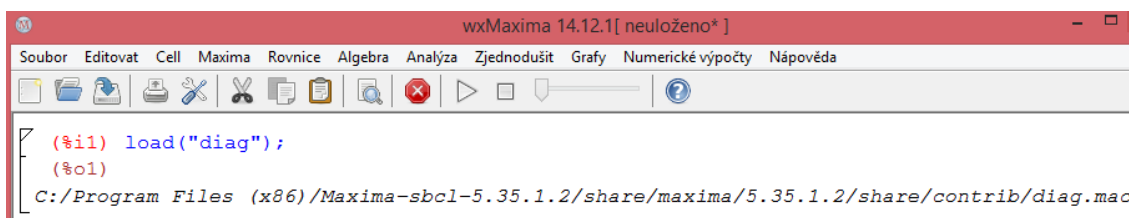
(%i8) eigenvalues(%);
(%o8) [[-3],[4]]
(%i10) eigenvectors(matrix([-5,1,0,0],[-4,2,2,-1],[6,-6,-5,1],[0,3,2,-4]));
(%o10) [[[-3],[4]], [[1,2,0,6],[0,0,1,2]]]
```

Obrázek 4.6: Vlastní čísla a vlastní vektory matice v programu wxMaxima



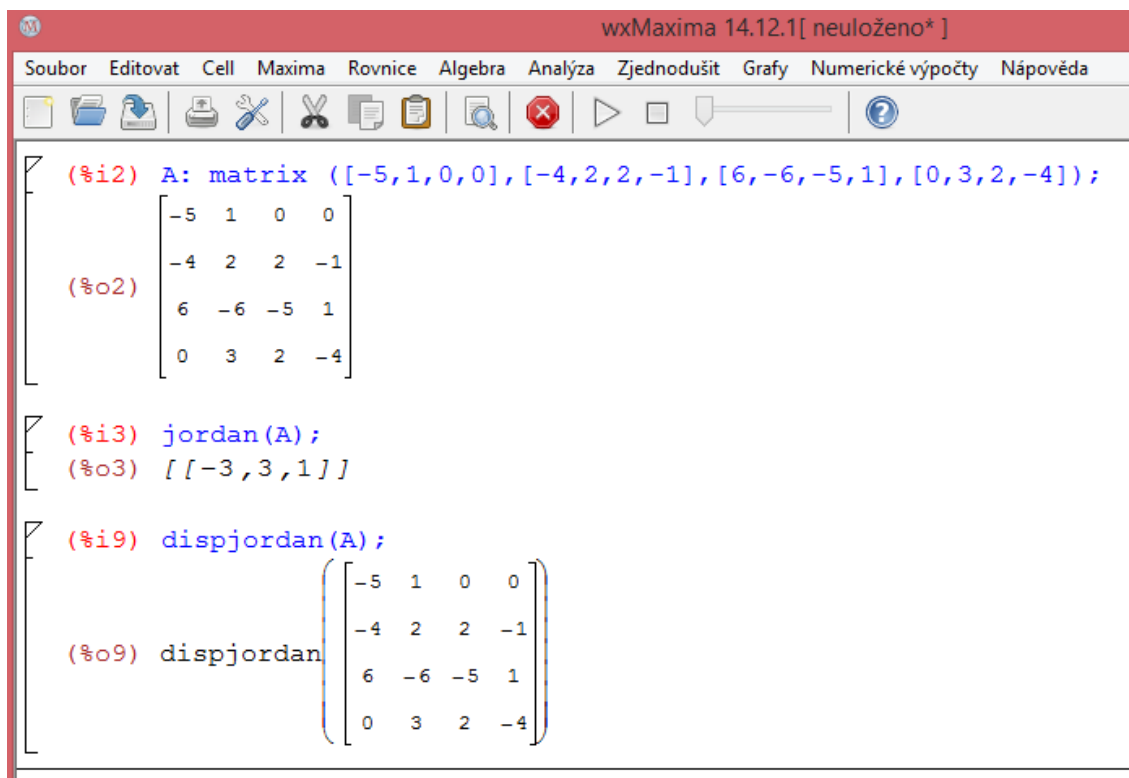
Obrázek 4.7: Vlastní čísla a vlastní vektory v aplikaci wolfram alpha

Další částí příkladu je Jordanův kanonický tvar J matice a ověření rovnosti $A = TJT^{-1}$. Výpočet nejde udělat přímo, ale Jordanovu matici je možné spočítat načtením balíčku "diag" (příkaz `load("diag");`); a dále příkazy `jordan(A)`; a následně `dispJordan(%)`;



Obrázek 4.8: Načtení balíčku diag

Oproti wolframu alpha, který vypíše i mezikroky při řešení, wxMaxima jen vrátí Jordanovu matici. Je ale zřejmé, že rovnost platí.

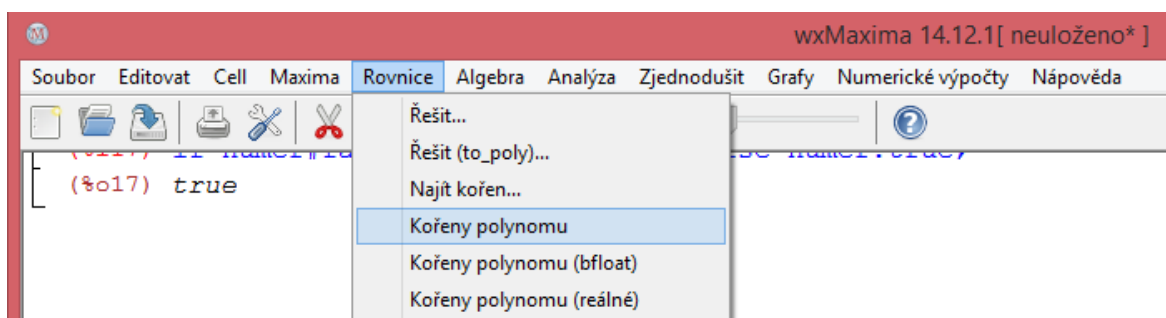


Obrázek 4.9: Ověření rovnosti matice A a Jordanovo matice

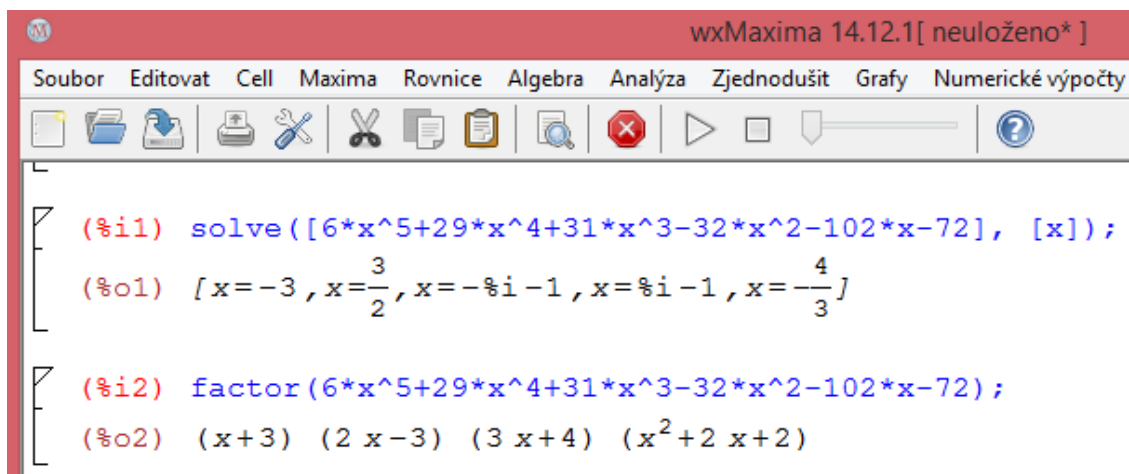
Příklad 4.4: Určení všech kořenů polynomu a ověření roznásobením všech kořenů

Zadání: „Určeme všechny kořeny polynomu $f(x) = 6x^5 + 29x^4 + 31x^3 - 32x^2 - 102x - 72$, jestliže je známo, že některé z jeho kořenů jsou celočíselné, resp. racionální (a neceločíselné).“

Řešení: V programu wxMaxima musíme nejprve zadat samostatně polynom, aby si ho program uložil do paměti, a pak můžeme použít nabídku panelu nástrojů: Rovnice – kořeny polynomu. Nebo můžeme použít příkaz allroots a do závorky vypsát polynom, jehož kořeny chceme najít.

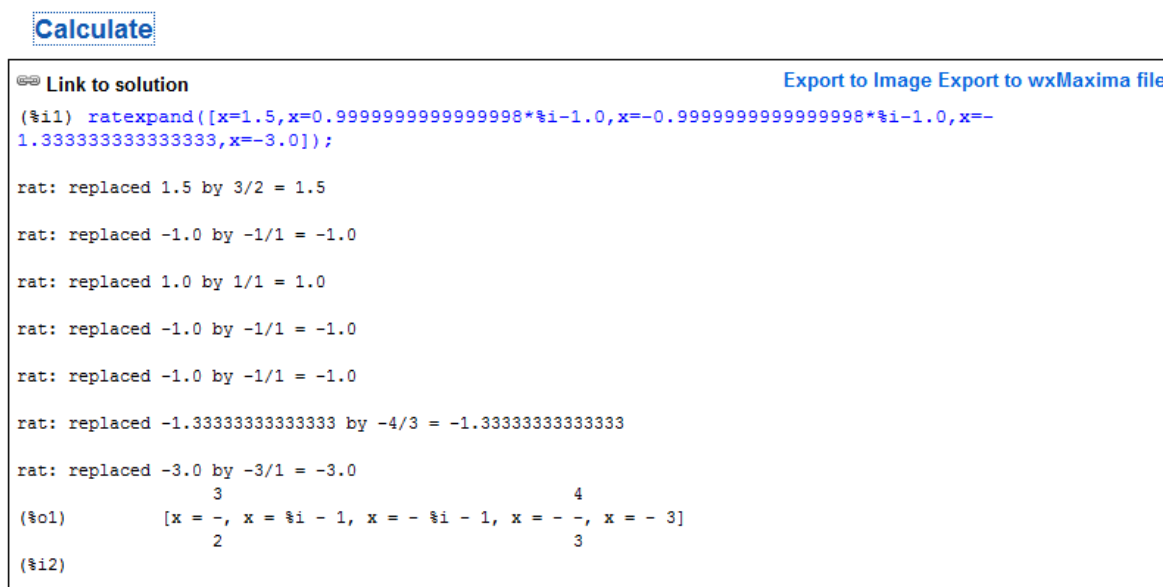


Obrázek 4.10: Program wxMaxima – Kořeny polynomu



Obrázek 4.11: Kořeny polynomu – výsledek

Program wxMaxima vrátí dosti nepřehledný výsledek se zaokrouhlením na šestnáct míst. Přepnutím numerického výstupu upravíme výsledek na zlomky. Použijeme-li v hlavní nabídce Zjednodušit možnost Na součin, vypíše wxMaxima výsledek rozkladu polynomu v součinném tvaru (obrázek 4.11). Na rozdíl od wolfram alpha, který zapíše výsledek ve tvaru rozkladu na součin, wxMaxima vypíše hned jednotlivé kořeny. Ve webově aplikaci Maxima se použitím příkazu ratexpand zobrazí výsledek všech kořenů v přívětivější formě (obrázek 4.14). Bohužel ověření pomocí roznásobení všech kořenů neumí ani webová aplikace Maxima ani program wxMaxima.



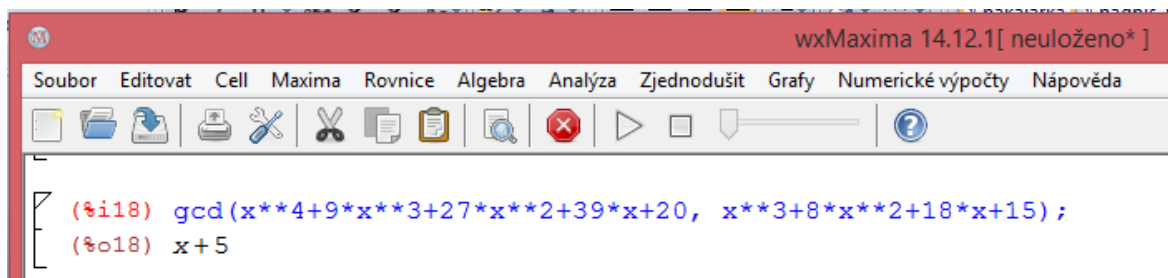
Obrázek 4.12: Příkaz ratexpand – výpis kořenů polynomu

Příklad 4.5: Nalezení největšího společného dělitele polynomů

Zadání: „Nad oborem integrity polynomů jedné proměnné nad komutativním tělesem

$(\mathbb{Q}; +, \cdot)$ najděte největší společný dělitel polynomů $f(x) = x^4 + 9x^3 + 27x^2 + 39x + 20$,
 $g(x) = x^3 + 8x^2 + 18x + 15$.“

Řešení: Pro zjištění největšího společného dělitele polynomů použijeme příkaz gcd nebo opět panel nástrojů – nabídka Největší společný dělitel...



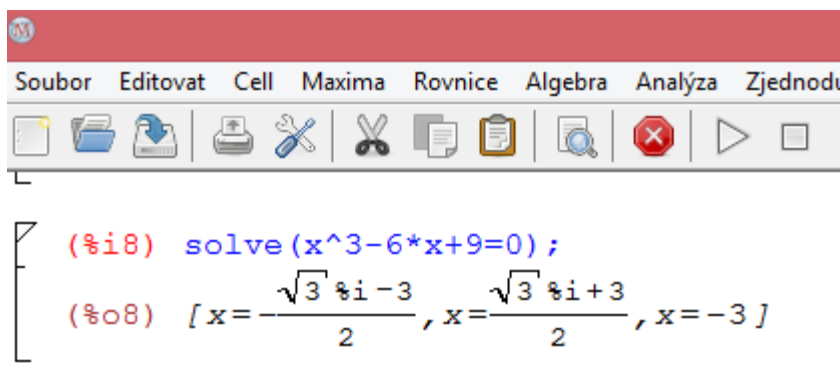
Obrázek 4.13: Největší společný dělitel polynomů

Z obrázku je zřejmé, že výsledek největšího společného dělitele zadaných polynomů je $x + 5$. Tedy $D(f(x), g(x)) = x + 5$.

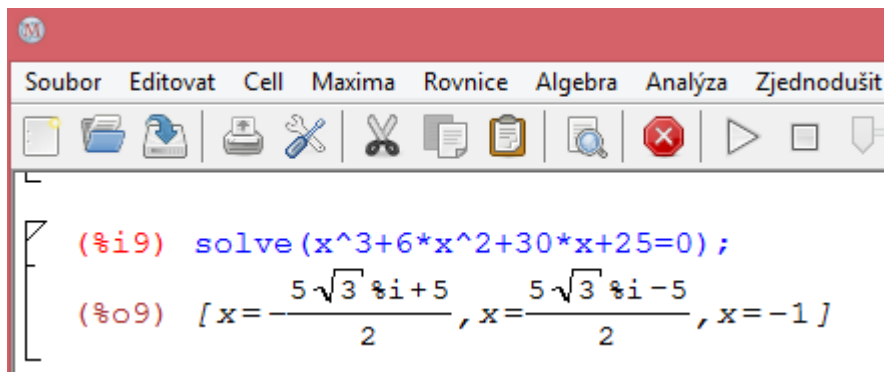
Příklad 4.6: Řešení kubických rovnic

Zadání: „Řešte kubické rovnice a) $x^3 - 6x + 9 = 0$, b) $x^3 + 6x^2 + 30x + 25 = 0$,
c) $x^3 + 9x^2 + 19x + 11 = 0$, d) $x^3 + 3x + 8 = 0$.“

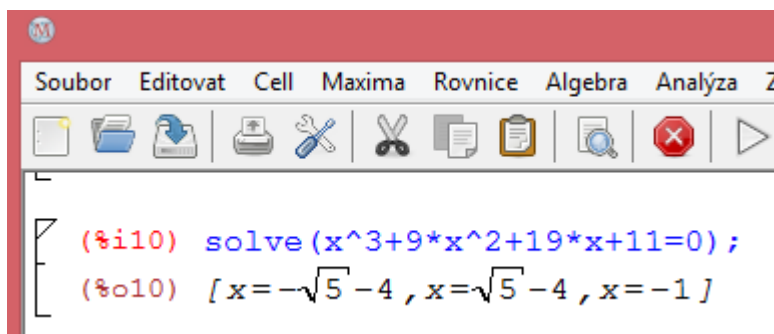
Řešení: Pro získání řešení v programu wxMaxima zadáme příkaz solve nebo použijeme nabídku v panelu nástrojů – Rovnice: Řešit. Výsledky jednotlivých rovnic jsou uvedeny na obrázcích 4.14 až 4.17. Po přepnutí numerického výstupu jsou řešení zobrazena ve zlomcích.



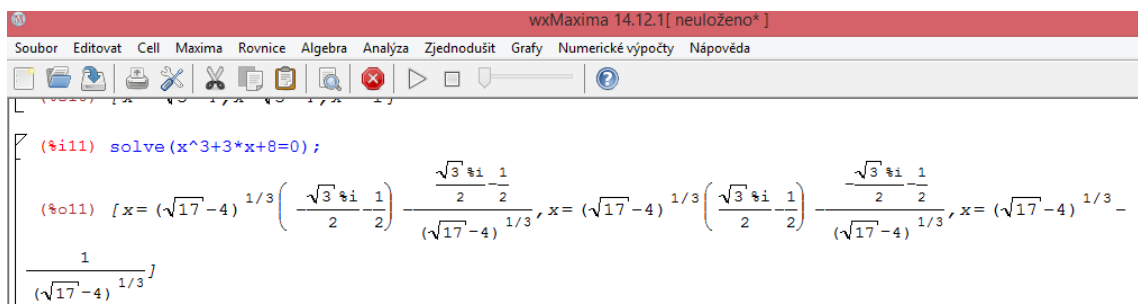
Obrázek 4.14: Program wxMaxima – Řešení kubické rovnice I



Obrázek 4.15: Program wxMaxima – Řešení kubické rovnice II



Obrázek 4.16: Program wxMaxima – Řešení kubické rovnice III



Obrázek 4.17: Program wxMaxima – Řešení kubické rovnice IV

Příklad 4.7: Derivace některých funkcí

Zadání: „Řešte derivace funkcí a) $f(x) = e^x \cdot x^2$, b) $g(x) = \ln(\ln(\ln(x)))$, c) $h(x) = \sin^3(x)$.“

Řešení: Záměrně jsem si vybrala pro ukázkou derivací tyto funkce, protože si s nimi wxMaxima sice poradí, ale mají specifické zadávání. Pro výpočet derivace použijeme příkaz diff nebo nabídku panelu nástrojů – derivovat.

a) $f(x) = e^x \cdot x^2$

Jak je vidět na obrázku 4.24 program wxMaxima funkci zderivuje, jen se nesmí zapomenout připsat znak % před funkci e^x .

```
(%i12) diff(%e^x*x^2);
(%o12) (x^2 %e^x+2 x %e^x) del (x)
```

Obrázek 4.18: Řešení derivace I

b) $g(x) = \ln(\ln(\ln(x)))$

S touto funkcí nemá při derivaci wxMaxima žádné potíže, jen přirozený logaritmus zapisuje příkazem log a ne ln. Maxima zná pouze přirozený logaritmus, ostatní musíme zadat výpočtem.

```
(%i27) diff(ln(ln(ln(x))), x, 1);
(%o27)  $\frac{d}{dx} \ln(\ln(\ln(x)))$ 
```

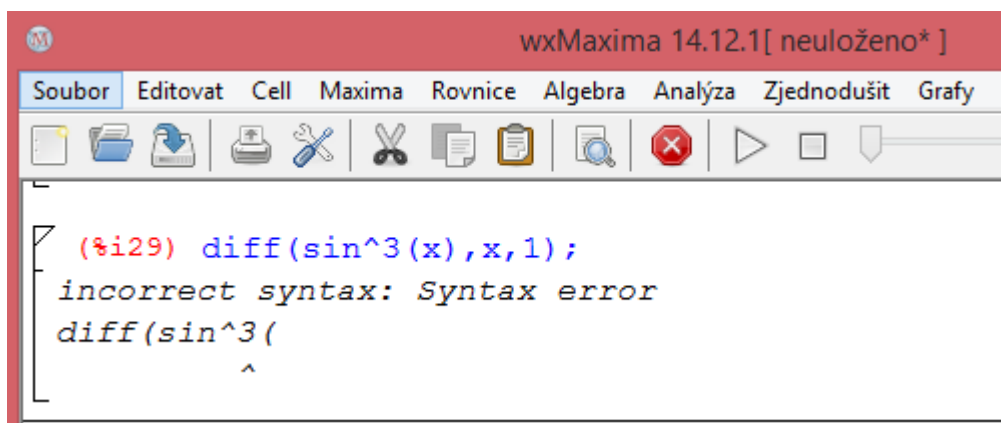
Obrázek 4.19: Řešení derivace IIa

```
(%i28) diff(log(log(log(x))), x, 1);
(%o28)  $\frac{1}{x \log(x) \log(\log(x))}$ 
```

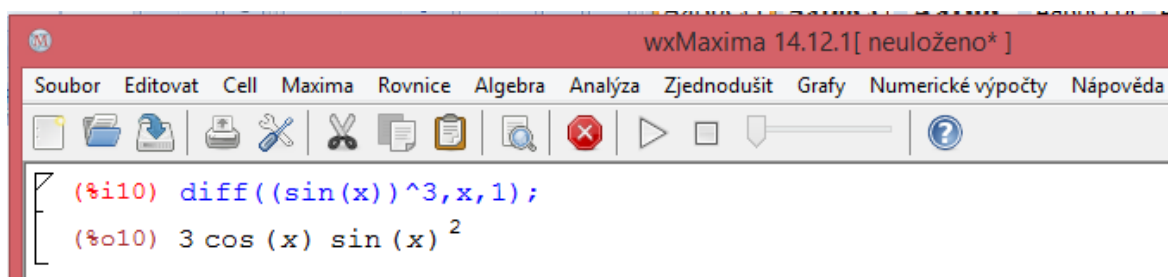
Obrázek 4.20: Řešení derivace IIb

c) $h(x) = \sin^3(x)$

Funkci sinus (x) dokáže wxMaxima derivovat bez obtíží. Problém nastává ve chvíli, kdy potřebujeme derivaci n-té mocniny některé z goniometrických funkcí. Tohle wxMaxima spočítat sice umí, ale uživatel si musí dát pozor na správný zápis. Syntaxe $\sin^3(x)$, což wolfram alpha přečte a vypočítá bez problémů, wxMaxima nevezme. Správná syntaxe je: $(\sin(x))^3$.



Obrázek 4.21: Řešení derivace III – chybný zápis funkce



Obrázek 4.22: Řešení derivace III – správný zápis funkce

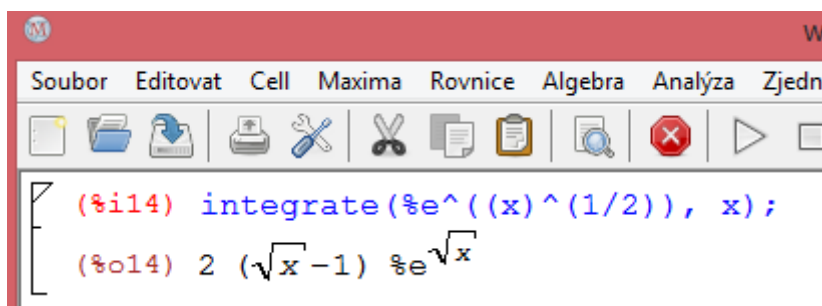
Příklad 4.8: Integrace – neurčité integrály

Zadání: „Vyřešte neurčitý integrál funkce a) $f(x) = e^{\sqrt{x}}$, b) $g(x) = \frac{e^{2x}}{\sqrt[4]{e^x+1}}$,

$h(x) = \frac{\sqrt{x}}{\sqrt{x}-\sqrt[3]{x}}$, i) $x) = \frac{\sqrt{2x+x^2}}{x^2}$.“

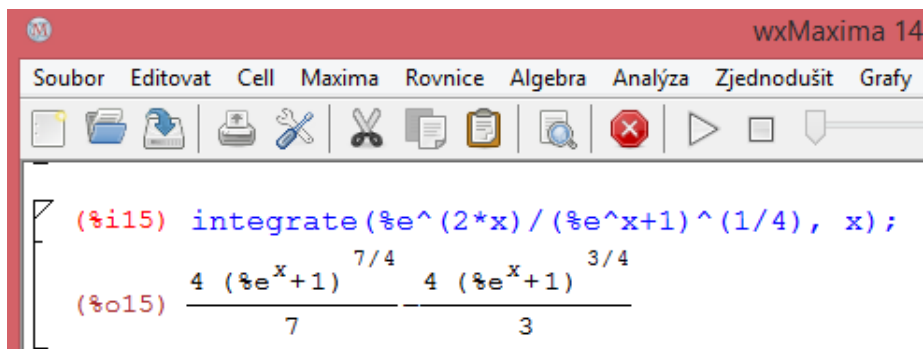
Řešení: WxMaxima si poradí i se složitějšími integrály, které bychom intuitivně řešili jako racionálně lomené funkce nebo pomocí Eulerovy substituce.

a) $f(x) = e^{\sqrt{x}}$



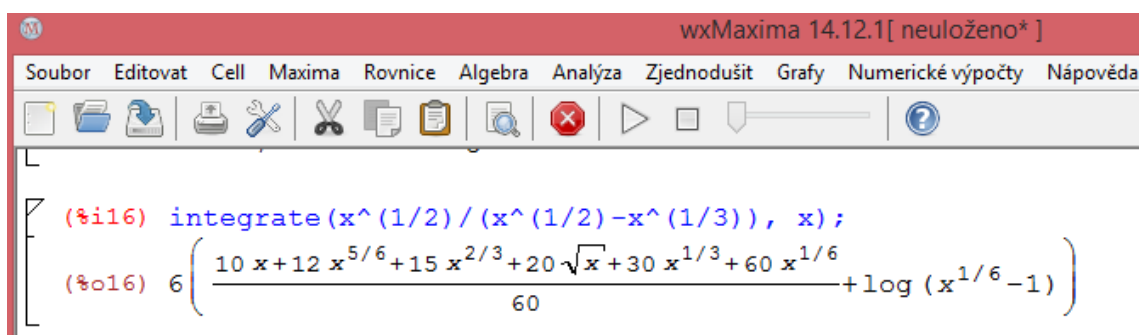
Obrázek 4.23: Řešení integrace funkce $f(x) = e^{\sqrt{x}}$

b) $g(x) = \frac{e^{2x}}{\sqrt[4]{e^x+1}}$



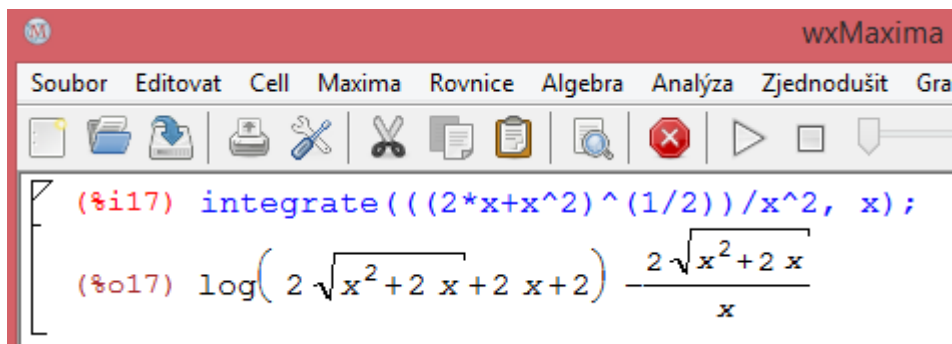
Obrázek 4.24: Řešení integrace funkce $g(x) = \frac{e^{2x}}{\sqrt[4]{e^x+1}}$

c) $h(x) = \frac{\sqrt{x}}{\sqrt{x} - \sqrt[3]{x}}$



Obrázek 4.25: Řešení integrace funkce $h(x) = \frac{\sqrt{x}}{\sqrt{x} - \sqrt[3]{x}}$

$$d) i(x) = \frac{\sqrt{2x+x^2}}{x^2}$$

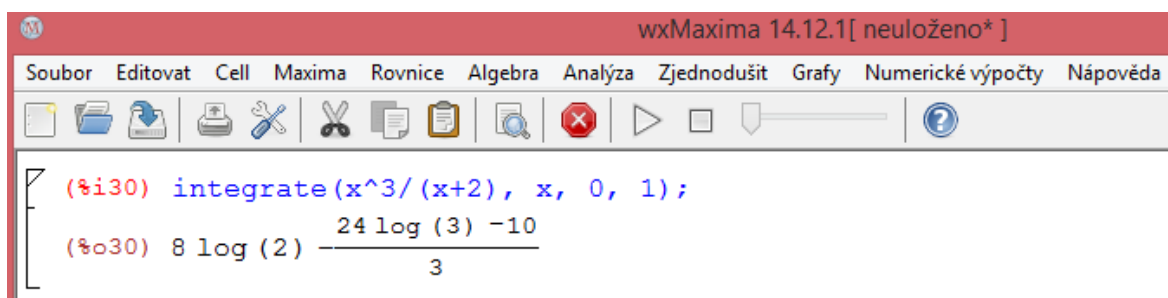


Obrázek 4.26: Řešení integrace funkce $i(x) = \frac{\sqrt{2x+x^2}}{x^2}$ – Eulerova substituce

Příklad 4.9: Integrace – určité integrály I

Zadání: „Vyřešte určitý integrál s vlastními mezemi od 0 do 1 funkce $f(x) = x^3/(x+2)$.“

Řešení: WxMaxima integrál v pořádku vyřeší.

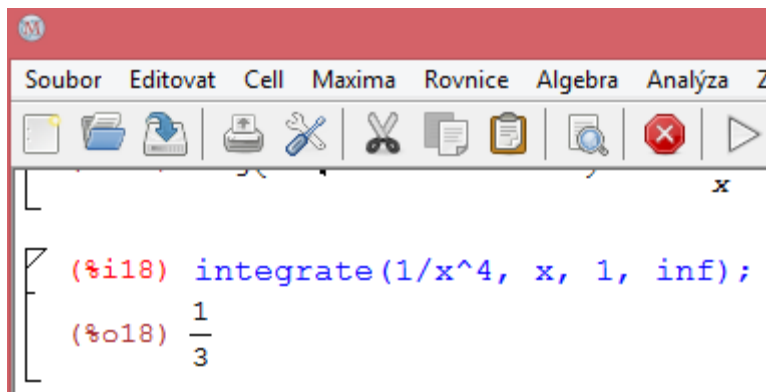


Obrázek 4.27: Určitý integrál I

Příklad 4.10: Integrace – určité integrály II

Zadání: „Vyřešte určitý integrál s nevlastní mezí od 1 do nekonečna funkce $f(x) = \frac{1}{x^4}$.“

Řešení: Tento integrál také v pořádku vyřeší.



Obrázek 4.28: Určitý integrál II

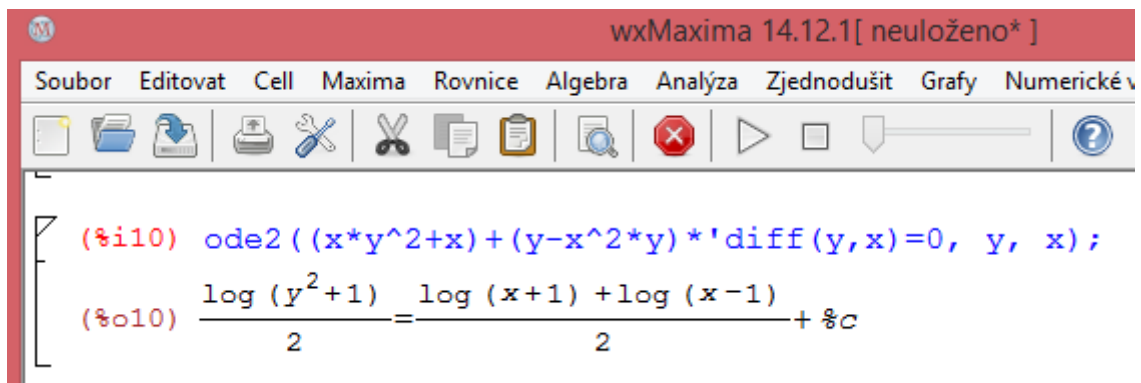
Příklad 4.11: Jednoduché diferenciální rovnice

Zadání: „Vyřešte obecnou diferenciální rovnici $(x \cdot y^2 + x) + (y - x^2 \cdot y) \cdot (y)' = 0$.“

Řešení: Program wxMaxima nabízí možnost vyřešení diferenciálních rovnic, ale je potřeba napsat správnou syntaxi příkazu. Nesmíme zapomenout na klíčové slovo 'diff u diferencovaného y a připsání proměnné x. Za rovnici zapíšeme znak funkce a její proměnné:

ode2 ((x * y^2 + x) + (y - x^2 * y) * 'diff(y,x) = 0, y, x);

V případě, že není schopen rovnici vyřešit, vrátí hlášku false.



Obrázek 4.29: Diferenciální rovnice

5 Zhodnocení dovedností programu

Program wxMaxima i aplikace Maxima nabízí širokou škálu použití a pro její celé prozkoumání a sestavení manuálu nestačí rozsah bakalářské práce, ale vyžadovalo by to několik samostatných knih. Pro svou práci jsem se zaměřila na nejzákladnější použití programu jako takového a hlubší prozkoumání části zaměřující se na analýzu a algebru.

Po celkem dlouhém prozkoumávání a vyzkoušení různých příkladů bych zhodnotila program wxMaxima takto:

Jedná se o velmi zajímavý a určitě užitečný program, bohužel jen pro určitou skupinu uživatelů. Program wxMaxima zvládá základní početní operace, jako je sčítání, odčítání, násobení, dělení, umocňování a odmocňování čísel a výrazů. Docela obstojně zvládne vyřešit různé lineární i nelineární rovnice a soustavy rovnic. Dobře si poradí i s integrací složitějších funkcí, které bychom intuitivně řešili například Eulerovo substitucí. To samé platí i u derivací funkcí, Maxima zvládne vše vyřešit, jen mi chvíli trvalo, než jsem zjistila, jak má být příkaz správně zadán. Velký nedostatek vidím v logaritmech. Maxima rozpozná jen přirozený logaritmus a zadává se \log a ne \ln . Ostatní logaritmy, jako například dekadický logaritmus, musí být zadány výpočtem. To je časově náročné.

Program wxMaxima sám o sobě funguje celkem dobře, ale občas se mi stalo, že přestal pracovat, nebo vrátil chybovou hlášku: se zadaným argumentem nelze pracovat. Po zadání vstupního příkazu a stisknutí klávesy enter se příklad nevyřešil a kurzor naskočil opět na vstupní příkaz. Program jsem musela ukončit a spustit znovu, poté fungovalo vše v pořádku.

Hodně mě také překvapila časová prodleva s jakou program wxMaxima nastavuje novou přesnost. Přednastavená je na šestnáct míst. Vyzkoušela jsem ji několikrát změnit. Čím vyšší přesnost jsem nastavila, tím byla delší doba zpracování.

Webová aplikace Maxima nabízí daleko více možností než sám program wxMaxima, což je pro začínající uživatele zbytečné a navíc nepřehledné. V porovnání s wolframem alphou nenabízí možnosti příkazů v příkazové řádce. Existuje několik stovek příkazů, které jsou sice v nabídce help, ale než si uživatel všechny nápovědy a poučky přeloží, zabere mu to spousty času. Aplikace Maxima umí být přehlednější a uživatelsky příjemnější, až po zadání určitého příkazu nebo přepnutí numerického výstupu.

A to opět znamená pro uživatele čas, aby si našel správné zadání příkazů. Jako plus vidím, že u většiny příkazů, které jsou obsažené v nápovědě, je uvedeno i několik příkladů. Také je mezi nimi jistá provázanost, což vede k jednoduššímu hledání potřebných příkazů. Zajímavé je také vykreslování, přibližování a oddalování a různé posouvání pohledu 3D grafů, toto je poněkud náročnější na zadávání.

Závěrem bych zmínila, že tento program využije jen určitá skupina uživatelů, i když některé možnosti má lepší než wolfram alpha. Co je ale složitější, je veškeré zadávání příkazů a u některých výpočtů se musí navíc dotahovat tzv. balíčky, bez nichž by Maxima příklad nevyřešila. Doufejme, že tito uživatelé program udrží a budou ho dále vylepšovat a zdokonalovat. Bylo by škoda, kdyby měl zaniknout.

Část IV

Závěr

Cílem této práce bylo prozkoumání a popsání systému počítačové algebry Maxima. Práce je rozdělena do dvou hlavních částí, teoretické a praktické. Jelikož je Maxima velmi rozsáhlý program a umí toho , zaměřila jsem se hlavně na algebru a analýzu.

Teoretická část popisuje základní ovládání programu Maxima, grafické prostředí wxMaxima a online webové rozhraní programu Maxima.

V první kapitole jsou popsány příkazy, jak se zadávají do Maximy a je v ní upozorněno na důležité detaily, na které nesmí uživatel při zadání zapomínat.

V druhé kapitole jsou zhodnocené poznatky z práce z wxMaxima, kterou bylo nutné si nainstalovat do počítače. Blíže je popsáno několik možností z hlavní nabídky, které se nejvíce využijí při výuce analýzy a algebry.

Třetí kapitola popisuje online webové rozhraní programu Maxima a ukazuje malé rozdíly oproti programu wxMaxima.

Dalo by se zhodnotit, že oba prostředky zvládnou stejné výpočty, jen se k některým zobrazením dostaneme jiným příkazem.

Stěžejním pro tuto práci bylo vypracování řešených příkladů a zhodnocení toho, co systém dokáže a co nezvládne. Dalším cílem bylo malé porovnání s wolframem alphou, což je popsáno v páté kapitole.

Vzhledem k výše uvedenému se domnívám, že jsem cíl práce splnila. Program Maxima mě docela příjemně překvapil, kolik toho zvládne vyřešit. Ale nemyslím si, že by se rozšířil a byl používán stejně efektivně jako wolfram alpha. Naštěstí díky skupince nadšenců pro tento program, kteří jej budou nadále používat a zlepšovat, si určitě udrží své místo na dost dlouho.

Resumé

The objective of my thesis was to describe Maxima which is a system of a computer algebra. The thesis is split into two main parts, theoretical and practical. Theoretical part describes how to operate software Maxima, graphical interface wxMaxima and on-line web interface of Maxima. Key deliverable of this thesis was to work out mathematical problems and evaluate what the software can achieve and what it can't. Another deliverable was to perform a light comparison to wolfram alpha. My conclusion is that software Maxima is appropriate for university students and specialists rather than for public usage.

Část V

Literatura

- *Maxima, a Computer Algebra System* [online]. Dostupné z: <http://maxima.sourceforge.net/>
- *Maxima*. In Wikipedia: Otevřená encyklopedie [online]. San Francisco (CA): Wikimedia Foundation, 2001. Dostupné z: <http://cs.wikipedia.org/wiki/Maxima>
- Majewski, A. Maxima: free sophisticated computer algebra system (CAS). *Fractals* [online]. 2010-07-09. Dostupné z: <http://fraktal.republika.pl/maxima.html>
- Vodopivec, A. *wxMaxima* [online]. 2011. Dostupné z: <http://andrejv.github.io/wxmaxima>
- Online Algebra Calculator. *Maxima Online* [online]. Dostupné z: <http://maxima-online.org/index.html>
- WOLFRAM, Stephen. WolframAlpha: computational knowledge engine [online]. Dostupné z: <http://www.wolframalpha.com>
- KAUFNEROVÁ, Gabriela. *Využití Wolramu alpha v matematice*. Bakalářská práce. Vedoucí práce: HONZÍK Lukáš, Plzeň, 2014. Dostupné z: <https://portal.zcu.cz/portal/studium/prohlizeni.html>
- Hora, J. *Poznámky z přednášek a cvičení z předmětu Vybrané kapitoly z matematické analýzy*, 2013.

Seznam obrázků

Obrázek 1.1: Maxima – online prohlížeč	7
Obrázek 2.1: wxMaxima – nabídka panel nástrojů	8
Obrázek 2.2: Rovnice zadání	8
Obrázek 2.3: Rovnice řešení	8
Obrázek 2.4: Algebra – Matice	9
Obrázek 2.5: Analýza derivace	10
Obrázek 2.6: Analýza integrace	10
Obrázek 2.7: Analýza určitý integrál	10
Obrázek 2.8: Přepnutí numerického výstupu	11
Obrázek 2.9 Graf 2D $\sin(x)$	11
Obrázek 2.10: 2D graf	12
Obrázek 3.1: Maxima online	14
Obrázek 3.2: Horní okno prohlížeče – zadání	14
Obrázek 3.3: Dolní okno prohlížeče – řešení	14
Obrázek 3.4: Absolutní hodnota – výstup	15
Obrázek 3.5: Početní operace s maticemi 1 – výstup	16
Obrázek 3.6: Početní operace s maticemi 2 – výstup	16
Obrázek 3.7: Početní operace s maticemi 3 – výstup	18
Obrázek 3.8: Součet matic – vstup	18
Obrázek 3.9: Součet matic – výstup	19
Obrázek 3.10: Diagonální matice – výstup	19
Obrázek 3.11: Algebraické rovnice – výstup	20
Obrázek 3.12: Soustava lineárních rovnic – výstup	21
Obrázek 3.13: Algebraický systém – vstup	21

Obrázek 3.14: Algebraický systém – výstup	22
Obrázek 3.15: Obyčejné diferenciální rovnice – výstup	23
Obrázek 3.16: Derivace – výstup	24
Obrázek 3.17: Integrál – výstup	25
Obrázek 3.18: Parametrický 2D graf – výstup	26
Obrázek 3.19: Diskrétní 2D graf – výstup	26
Obrázek 3.20: 3D graf – výstup	27
Obrázek 3.21: Parametrický 3D graf – výstup	27
Obrázek 4.1: Výsledek soustavy lineárních rovnic I	28
Obrázek 4.2: Výběr Rovnice – Řešit lineární systém	29
Obrázek 4.3: Výběr počtu rovnic s defaultní hodnotou 3	29
Obrázek 4.4: Podokno – zadané lineární rovnice a proměnné	29
Obrázek 4.5: Řešení soustavy lineárních rovnic II	30
Obrázek 4.6: Vlastní čísla a vlastní vektory matice v programu wxMaxima	31
Obrázek 4.7: Vlastní čísla a vlastní vektory v aplikaci wolramalpha	32
Obrázek 4.8: Načtení balíčku diag	32
Obrázek 4.9: Ověření rovnosti matice A a Jordanovo matice	33
Obrázek 4.10: Program wxMaxima – Kořeny polynomu	33
Obrázek 4.11: Kořeny polynomu – výsledek	34
Obrázek 4.12: Příkaz ratexpand – výpis kořenů polynomu	35
Obrázek 4.13: Největší společný dělitel polynomů	35
Obrázek 4.14: Program wxMaxima – Řešení kubické rovnice I	36
Obrázek 4.15: Program wxMaxima – Řešení kubické rovnice II	36
Obrázek 4.16: Program wxMaxima – Řešení kubické rovnice III	36
Obrázek 4.17: Program wxMaxima – Řešení kubické rovnice IV	37

Obrázek 4.18: Řešení derivace I	37
Obrázek 4.19: Řešení derivace IIa	38
Obrázek 4.20: Řešení derivace IIb	38
Obrázek 4.21: Řešení derivace III – chybný zápis funkce	38
Obrázek 4.22: Řešení derivace III – správný zápis funkce	39
Obrázek 4.23: Řešení integrace funkce $f(x) = e^{\sqrt{x}}$	39
Obrázek 4.24: Řešení integrace funkce $g(x) = \frac{e^{2x}}{\sqrt[4]{e^x+1}}$	39
Obrázek 4.25: Řešení integrace funkce $h(x) = \frac{\sqrt{x}}{\sqrt{x} - \sqrt[3]{x}}$	40
Obrázek 4.26: Řešení integrace funkce $i(x) = \frac{\sqrt{2x+x^2}}{x^2}$ – Eulerova substituce	40
Obrázek 4.27: Určitý integrál I	40
Obrázek 4.28: Určitý integrál II	41
Obrázek 4.29: Diferenciální rovnice	41

Seznam příkladů

Příklad 4.1: Určení všech řešení soustavy lineárních rovnic I	28
Příklad 4.2: Určení všech řešení soustavy lineárních rovnic II	30
Příklad 4.3: Určení vlastních čísel, vlastních vektorů, Jordanův kanonický tvar J matice a ověření rovnosti $A = TJT^{-1}$	30
Příklad 4.4: Určení všech kořenů polynomu a ověření roznásobením všech kořenů	33
Příklad 4.5: Nalezení největšího společného dělitele polynomů	35
Příklad 4.6: Řešení kubických rovnic	36
Příklad 4.7: Derivace některých funkcí	37
Příklad 4.8: Integrace – neurčité integrály	39
Příklad 4.9: Integrace – určité integrály I	40
Příklad 4.10: Integrace – určité integrály II	41
Příklad 4.11: Jednoduché diferenciální rovnice	41