Úvodní seznámení s matematickými programy Maxima 5.21, wxMaxima 0.8.5

1 Základy používání programu

Pozor! Jelikož Maxima je určena hlavně pro Linux, některé funkce programu nefungují pod operačním systém Windows - např. animace, řešení nerovnic, podmínky při řešení rovnic.

Jelikož je wxMaxima jen uživatelské rozhraní, které má zpřehlednit výstupy programu Maxima může se stát, že některé funkce nebudou dostupné - např. balíček vect.

Pozor! Tento dokument byl vytvořen ve Windows a je v něm použitá diakritika. V systému Linux jej proto není možné otevřít v programu wxMaxima.

Je možné použít alternativně rozhraní xmaxima, případně v Linuxu také terminál a další. Příkazy wxMaximy - grafy začínající wx a příkazy s with_slider nebudou v těchto rozhraních fungovat.
Výpočty se ukončují středníkem (není nutné pro jeden výpočet v jedné sekci označené -->, tam je dopněn automaticky).
Shift + Enter provede výpočet.
Enter posune na další řádek.
Klávesová zkratka CTRL + G přeruší výpočet. Případně lze použít v nabídce Maxima možnost Interupt, nebo červené tlačítko na horní liště.
Při opětovném otevření dokumentu Maximy, je potřeba znovu provést výpočty, jejikož v souboru jsou uloženy pouze zadání, nikoli výstupy.
Výpočty jsou prováděny v časovém pořadí v jakém jste spustili výpočty.
Je tedy možné přiradit nějakou proměnnou na konci notebooku a pokud poté provede s proměnnou výpočet např. na počátku, tak již bude program počítat s vaší zadanou hodnotou proměné.
Pokud chcete nechat propočítat celý pracovní list stiskněte CTRL + R, případně zvolte v nabídce Cell možnost Evaluate All Cells.
Pokud konec zadání ukočíme $, výsledek se provede, ale nevypíše.
Pro většinu složitějších funkcí je nutné zavolat balíčky příkazem load např. load(draw).
Pokud kliknete na danou funkci a stiknete klávesu F1, zobrazí se k dané funkci nápověda.
Pro zápis příkazů jako např. x^2, integrate(sin(x),x), je možné využít klávesnici případně si nechat vypsat syntax příkazu pomocí příkazů v horní liště (např. Calculus/Integrate). Poslední možností je nechat si nejprve zobrazit palety v nabídce Maxima/Panes a poté jen nakliknout příslušný příkaz.
Hodí se na počátku pracovního listu použít příkaz kill(all). Jednak se tím odstraní přiřazení proměnných a také načtené balíčky. Předejde se tím např. zavolání funkce z balíčku ještě před jeho načtením.
Program standardně nezalamuje psaný text, je potřeba to udělat ručně.
Pro práci s programem doporučuji požívat anglickou klávesnici, jelikož se snadněji zadávají např. mocniny (^) apod..
Má-li program nějaké speciální funkce vůči ostatním je za příslušným názvem odstavce "(navíc)".

```
(%i1) kill(all);
(%o0) done
```

2 Základní operace

2.1 Základní početní operace

Základní operace +-*/ se používají stejně jako v jakémkoli jiném programovacím jazyku. U tototo progamu se musí strinkně dodržovat operace násobení - např. 2x se musí zapsat jako 2*x, jinak program napíše chybovou hlášku a výpočet neprovede.
U dělení je dobré si dát pozor, aby složitější výrazy byly v závorkách, jejikož dělení má přednost před sčítáním.
Poslední výpočet se zavolá %.

```
(%i1) 1+2;
(%o1) 3

(%i2) 2^5-10*3;
(%o2) 2

(%i3) 2/(5+3);
(%o3) 1/4
```

2.2 Číselný výpočet

U definovaných funkcí a zlomků se výrazy automaticky nepřevádějí na číselnou hodnotu (z důvodu přesnosti). Pokud ale je nějaké z čísel je desetiné, dostaneme číselnou hodnotu, ne zlomek (což se dá využít i jako trik, pro rychlé získání číselné hodnoty).
Pozor! Pro čísla z intervalu (-1,1) zlobí počet cifer zaokrouhlování, nastavených příkazem fpprintprec. Příkaz float a numer nevypíše počet platných cifer, ale počet znaků (tzn. počet platných čísel - 2).
Příkaz fpprintprec:0$ vrací zpět přesnost vypsaného čísla na standardní hodnotu (16 cifer). Maximální hodnota příkazu fpprintprec se dá zvýšit příkazem fpprec.

```
(%i4) sin(3);
(%o4) sin(3)

(%i5) float(sin(3));
      sin(3),numer;
      bfloat(sin(3));
(%o5) 0.14112000805987
(%o6) 0.14112000805987
(%o7) 1.411200080598672b-1

(%i8) fpprintprec:13$
      float(sin(3));
      sin(3),numer;
      bfloat(sin(3));
      fpprintprec:0$
(%o9) 0.14112000806
(%o10) 0.14112000806
(%o11) 1.411200080598b-1
```

Trik

```
(%i13) sin(3.0);
(%o13) 0.14112000805987
```

2.3 Matematické konstanty

Matematické konstanty se píší pomocí klávesnice se znakem %
na počátku.
U nekonečna se znak % nepřidává a rozlišuje se mezi reálným
a komplexním nekonečnem.

```
(%i14) %pi;
       %e;
       %i;
       inf;
(%o14) π
(%o15) %e
(%o16) %i
(%o17) ∞
```

2.4 Definování proměnné

U definování názvů proměnné se hodí nepoužívat háčky a čárky,
případně speciální znaky.

```
(%i18) promenna:3;
(%o18) 3
```

```
(%i19) promenna;
(%o19) 3
```

2.5 Definování vlastní funkce

```
(%i20) funkce(x):=(x+1)/(x^2-1);
       funkce(2);
(%o20) funkce(x):=(x+1)/(x^2-1)
(%o21) 1
```

Definovani funkce pokud je výraz funkce v jiné proměnné

```
(%i22) vyraz:x+1$
       funkce2(x):=''vyraz;
(%o23) funkce2(x):=x+1
```

3 Řešení rovnic a nerovnic

3.1 Řešení rovnice

```
(%i24) solve(x+3=4,x);
(%o24) [x=1]
```

Příkaz solve řeší pouze algebraické rovnice, na složitější
rovnice hodí balík to_poly_solver s funkcí to_poly_solve.

```
(%i25) load("to_poly_solver")$
       to_poly_solve([sin(x)+cos(x)=1], [x]);
Loading maxima-grobner $Revision: 1.6 $ $Date: 2009/06/02 07:49:49 $
define: warning: redefining the built-in function prog1
define: warning: redefining the built-in function symbolcheck
define: warning: redefining the built-in function push
define: warning: redefining the built-in function pop
define: warning: redefining the built-in function tr_ev
```

(%o26) $\%union\left([x=2\,\pi\,\%z6],[x=2\,\pi\,\%z8+\frac{\pi}{2}]\right)$

3.2 Řešení soustavy rovnic

```
(%i27) solve([x+y=1,x-y=2],[x,y]);
```

(%o27) $[[x=\frac{3}{2},y=-\frac{1}{2}]]$

3.3 Řešení nerovnice

Maxima umí řešit nerovnice za pomoci balíku solve_rat_ineq, který ale nefunguje pod Windows.
U nerovnic s neznámými v první mocnině je možné využít balík fourier_elim, který pod Windows funguje.

```
(%i28) load(solve_rat_ineq)$
       solve_rat_ineq(x^2-1>5);
```

(%o29) []

```
(%i30) load(fourier_elim)$
       fourier_elim(x-2> 5*x, [x]);
```

(%o31) $[x<-\frac{1}{2}]$

3.4 Řešení rovnice s podmínkami

Řešení rovnic s podmínkami nefunguje pod Windows.

```
(%i32) to_poly_solve(x^3-1=0, [x]);
```

(%o32) $\%union\left([x=1],[x=\frac{\sqrt{3}\,\%i-1}{2}],[x=-\frac{\sqrt{3}\,\%i+1}{2}]\right)$

```
(%i33) realonly:true$
       to_poly_solve(x^3-1=0, [x]);
       realonly:false$
```

(%o34) %union( )

```
(%i36) to_poly_solve([x^2+x-6=0,x>0], x);
```

(%o36) %union([$x=2$])

4 Úprava výrazů

4.1 Zjednodušení výrazu

Maxima používá více příkazů na zjednodušení výrazů - mnohé příkazy nejsou tak komplexní jako např. v Maple případně Mathematice.

```
(%i37) ratsimp((x^3-1)/(x-1));
```

$$(\%o37)\quad x^2+x+1$$

```
(%i38) radcan((x^3-1)/(x-1));
```

$$(\%o38)\quad x^2+x+1$$

4.2 Rozklad výrazu

```
(%i39) expand((x-1)^3);
```

$$(\%o39)\quad x^3-3\,x^2+3\,x-1$$

4.3 Složení výrazu

```
(%i40) factor(x^3-3*x^2+3*x-1);
```

$$(\%o40)\quad (x-1)^3$$

4.4 Převedení na společného jmenovatele

```
(%i41) factor(1/(x+1)+1/(x-1));
```

$$(\%o41)\quad \frac{2\,x}{(x-1)(x+1)}$$

4.5 Rozklad na parciální zlomky

```
(%i42) partfrac(2*x/(x^2-1),x);
```

$$(\%o42)\quad \frac{1}{x+1}+\frac{1}{x-1}$$

5 Vektory a matice

Pro práci s vektory a maticemi budeme pro některé funkce potřebovat balíček Lineární algebry a balíček pro práci s vektory.
Pozor! Balíček vect v programu wxMaxima nefunguje.
Funkce související z maticemi se dají najít v nabídce Algebra.

```
load(vect)$
```

```
(%i43) load(linearalgebra)$
```

5.1 Zápis vektoru

Na elementární práci s vektory, je možné použít jen zápis v hranatých závorkách. Hranaté závorky reprezentují seznam, který nemění pořadní členů.

```
(%i44) vektor:[3,1,2];
       vektor2:matrix([3,1,2]);
(%o44) [3,1,2]
(%o45) [3 1 2]
```

5.2 Složky vektoru

```
(%i46) vektor[1];
       vektor2[1][1];
(%o46) 3
(%o47) 3
```

5.3 Velikost vektoru

Bohužel Maxima nema funkci na velikost vektoru. Musí se tedy počítat standardně podle vzorce.

```
(%i48) sqrt([3,1,2].[3,1,2]);
(%o48) √14
```

5.4 Skalární součin

```
(%i49) [1,2,0].[0,1,2];
       matrix([1,2,0]).matrix([0,1,2]);
       dotproduct(transpose(matrix([1,2,0])),transpose(matrix([0,1,2])));
(%o49) 2
(%o50) 2
(%o51) 2
```

5.5 Vektorový součin

Pozor! Vektorový součet v programu wxMaxima nefunguje, jelikož je součástí balíčku vect.

```
express([1,2,0]~[0,1,2]);
```

5.6 Zápis matice

Základní funkce pro práci s maticemi je možné najít v nabídce Algebra.

```
(%i52) matice:[[1,2],[3,1]];
       matrix([1,2],[3,1]);
(%o52) [[1,2],[3,1]]
(%o53) matrix([1,2],[3,1])
```

Zápis matice II - pomocí nabídky Algebra/Enter Matrix

```
(%i54) matrix(
         [1,2],
         [3,1]
       );
(%o54) matrix([1,2],[3,1])
```

5.7 Složky matice

```
(%i55) matice[1][2];
(%o55) 2
```

5.8 Determinant matice

```
(%i56) determinant(matrix([1,2],[3,1]));
(%o56) -5
```

5.9 Vlastní čísla

```
(%i57) eigenvalues(matrix([1,2],[3,1]));
(%o57) [[1-sqrt(6),sqrt(6)+1],[1,1]]
```

5.10 Stopa matice

Pro určení stopy se nejdřív musí zavolat balík functs.

```
(%i58) load(functs)$
       tracematrix(matrix([1,2],[3,1]));
define: warning: redefining the built-in function lcm
(%o59) 2
```

5.11 Vlastní vektory

```
(%i60) eigenvectors(matrix([1,2],[3,1]));
(%o60) [[[1-sqrt(6),sqrt(6)+1],[1,1]],[[[1,-sqrt(6)/2]],[[1,sqrt(6)/2]]]]
```

6 Limity

Funkce související s integrálním a diferenciálním počtem lze nalést v nabídce Calculus, v případě diferenciálních rovnic v nabídce Equations.

6.1 Limita

```
(%i61) limit((x^2-x-2)/(x^3+1),x,-1);
(%o61) -1
```

6.2 Limita zleva

```
(%i62) limit(1/x^3,x,0,minus);
(%o62) -∞
```

6.3 Limita zprava

```
(%i63) limit(1/x^3,x,0,plus);
(%o63) ∞
```

6.4 Pozor!

Výsledkem limity níže je komplexní nekonečno.

```
(%i64) limit(1/x^3,x,0);
(%o64) infinity
```

7 Derivace

7.1 První derivace

```
(%i65) diff(x^3,x);
(%o65) 3 x^2
```

7.2 Vyšší derivace

```
(%i66) diff(x^3,x,3);
(%o66) 6
```

7.3 Derivace podle více proměnných

```
(%i67) diff(y^2*(x^3+y),x,1,y,1);
(%o67) 6 x^2 y
```

8 Integrály

8.1 Neurčitý integrál

```
(%i68) integrate(sin(x),x);
(%o68) -cos(x)
```

8.2 Určitý integrál

```
(%i69) integrate(sin(x),x,0,%pi/2);
(%o69) 1
```

8.3 Numerické počítání integrálů

```
(%i70) quad_qags(sin(x)/x, x, 0, %pi/2);
(%o70) [1.370762168154488,1.5218517203702136 10^-14,21,0]
```

8.4 Vícerozměrné integrály

```
(%i71) integrate(integrate(y*sin(x),x,0,%pi/2),y,0,1);
(%o71) 1/2
```

8.5 Taylorův rozvoj řady

```
(%i72) taylor(log(x),x,1,5);
```

$$(\%o72)/T/\quad x-1-\frac{(x-1)^2}{2}+\frac{(x-1)^3}{3}-\frac{(x-1)^4}{4}+\frac{(x-1)^5}{5}+\ldots$$

```
(%i73) powerseries(log(x),x,1);
```

$$(\%o73)\quad -\sum_{i1=1}^{\infty}\frac{(-1)^{i1}(x-1)^{i1}}{i1}$$

9 Diferenciální rovnice

Pro řešení diferenciálních rovnic není jen jeden příkaz, ale příkazy se liší podle typu diferenciálních rovnic a počátečních podmínek.

9.1 Diferenciální rovnice

```
(%i74) ode2('diff(y,x)+y/(1+x)=exp(-x), y, x);
```

$$(\%o74)\quad y=\frac{(-x-1)\%e^{-x}-\%e^{-x}+\%c}{x+1}$$

9.2 Diferenciální rovnice s počátečními podmínkami

```
(%i75) ode2('diff(y,x)+y/(1+x)=exp(-x), y, x);
```

$$(\%o75)\quad y=\frac{(-x-1)\%e^{-x}-\%e^{-x}+\%c}{x+1}$$

```
(%i76) ic1(ode2('diff(y,x)+y/(1+x)=exp(-x), y, x), x=5, y=1);
```

$$(\%o76)\quad y=\frac{\%e^{-x}((6\,\%e^{5}+7)\%e^{x}-\%e^{5}\,x-2\,\%e^{5})}{\%e^{5}\,x+\%e^{5}}$$

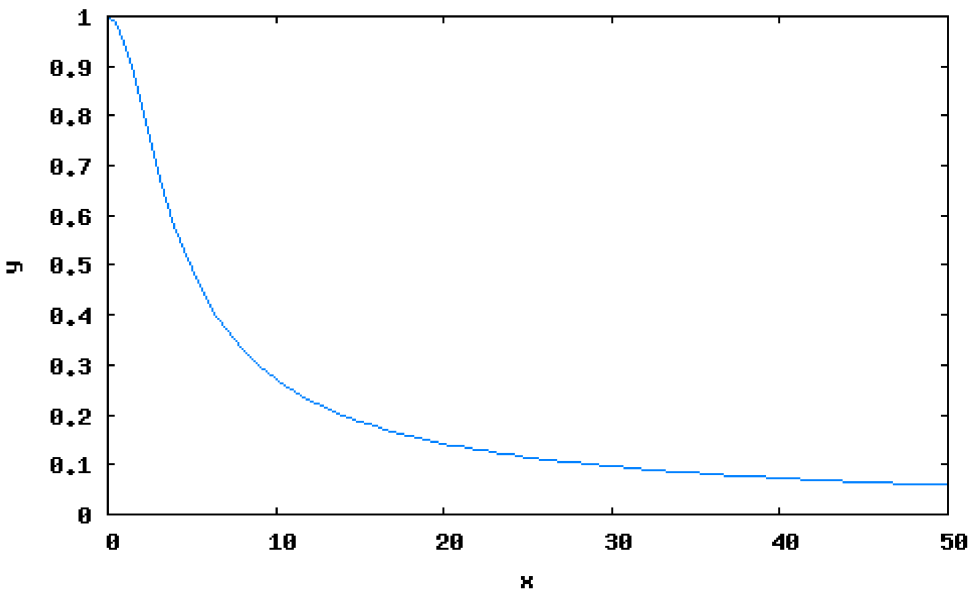
9.3 Numerické řešení diferenciálních rovnic

Pro numerické řešení diferenciálních rovnic se musí nejprve zavolat balíček dynamics, přičemž je možno řešit jen rovnice prvního stupně a počáteční podmínky se dají zadat jen v počátečním bodě.
Do funkce pro řešení (rk) diferenciálních rovnic se zadává vztah pro první derivaci tzn. ze vztahu y'=f(y(x),x) se zadává do funkce rk jen f(y(x),x).

```
(%i77) load(dynamics)$
```

```
(%i78) reseni:rk([-y/(1+x)+exp(-x)],y,1,[x,0,50,0.5])$
       wxplot2d([discrete,reseni]);
```

(%t79)



(%o79)

10 Posloupnosti a řady

10.1 Posloupnosti

```
(%i80) makelist(i^2,i,1,10);
(%o80) [1,4,9,16,25,36,49,64,81,100]
```

10.2 Řady

```
(%i81) sum(1/n^2,n,1,10);
(%o81) 1968329/1270080
```

```
(%i82) simpsum:true$
       sum(1/n^2,n,1,inf);
(%o83) π^2/6
```

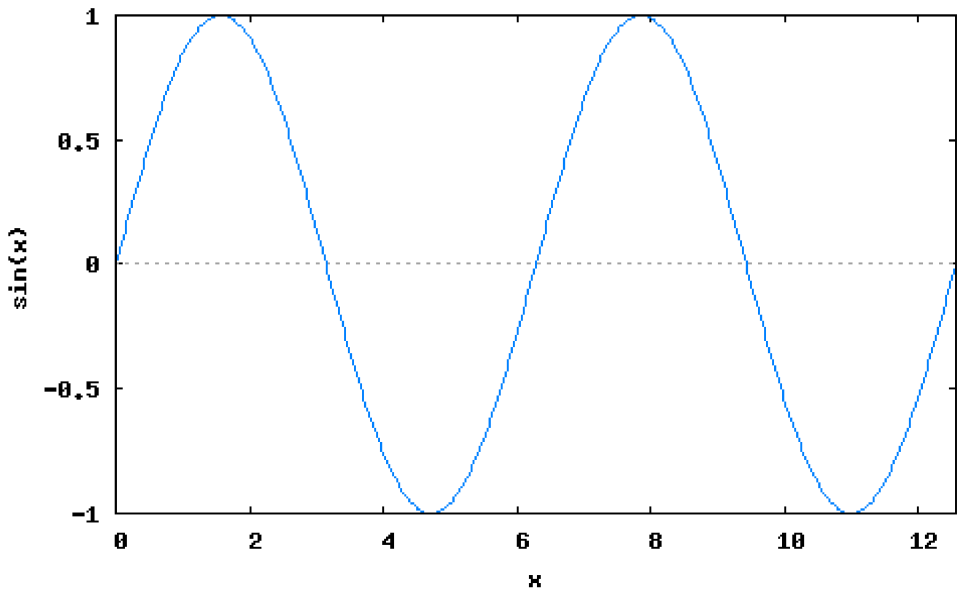
11 Grafy 2D a 3D

Maxima umožňuje kreslit grafy dvěmi typy příkazů. První je pomocí příkazů typu plot. Druhým způsobem je použití balíčku draw a příslušných příkazů pro grafy v něm. Příkazy typu plot, dělají přehlednější grafy (barevně, volí velikost a typ bodů), grafy přes balík draw na druhou stranu umožňují více nastavení grafu. Výhodou otevírání grafů v novém okně je možnost interaktivně otáčet 3D grafy.
Přidáním příkazu wx před plot se ve wxMaximě zobrazí graf přímo v programu.
Pozor! Pokud si necháte nějaký z grafů vykreslit (v novém okně), je důležité toto okno následně zavřít, jinak není možné v programu dále pracovat.

```
(%i84) load(draw)$
```

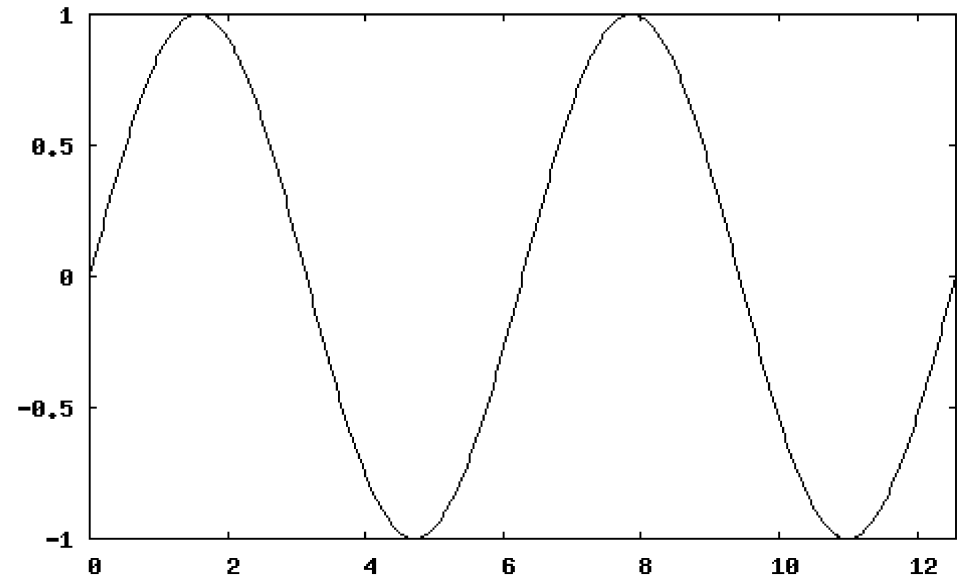
11.1 Grafy 2D - graf x,y

```
(%i85) wxplot2d(sin(x),[x,0,4*%pi]);
```




```
(%o85)
```

```
(%i86) wxdraw2d(explicit(sin(x),x,0,4*%pi));
```

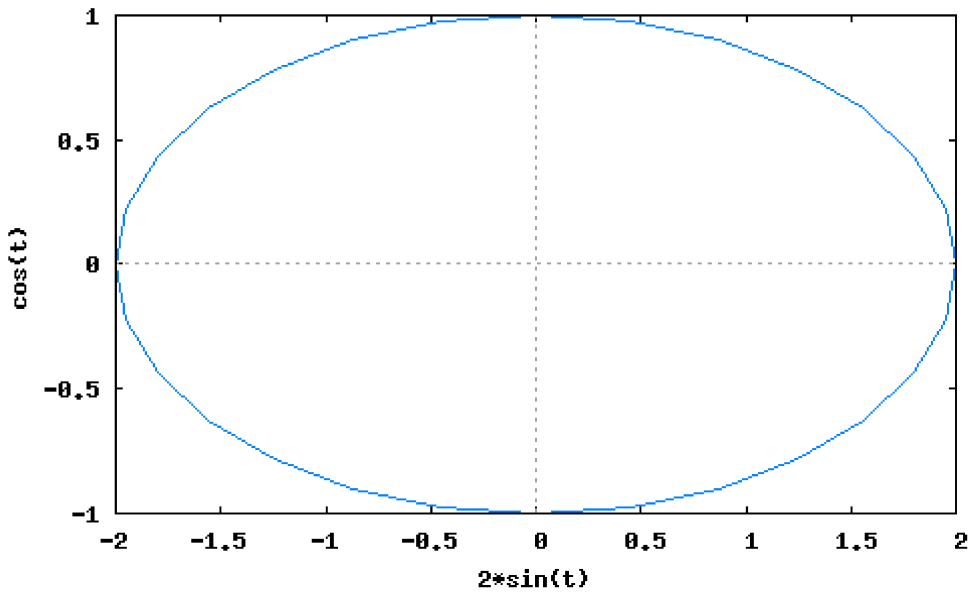



```
(%o86) [gr2d(explicit)]
```

11.2 Grafy 2D - parametrický graf

(%i87)
```
wxplot2d([parametric,2*sin(t),cos(t)],[t,0,2*%pi]);
```

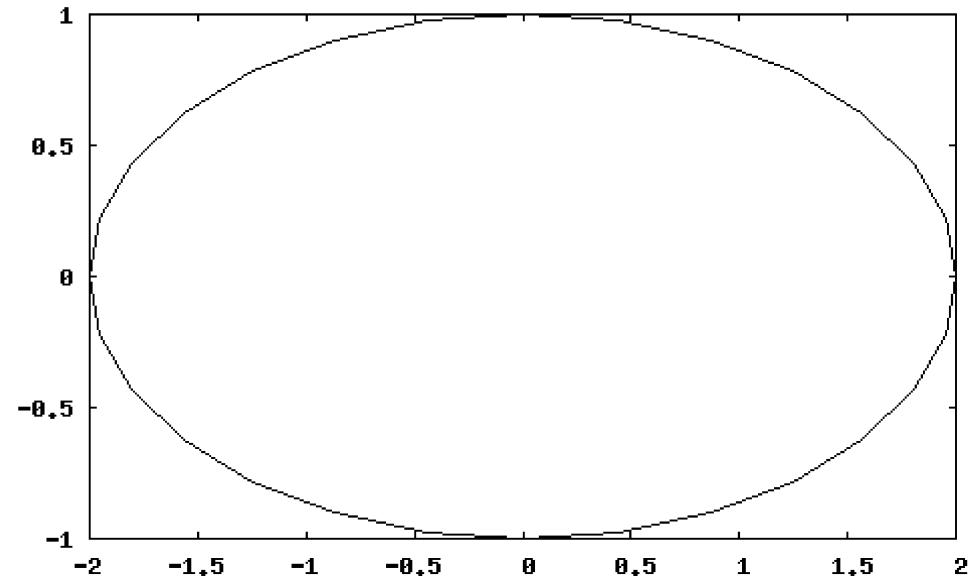
(%t87)



(%o87)

(%i88)
```
wxdraw2d(parametric(2*sin(t),cos(t),t,0,2*%pi));
```

(%t88)



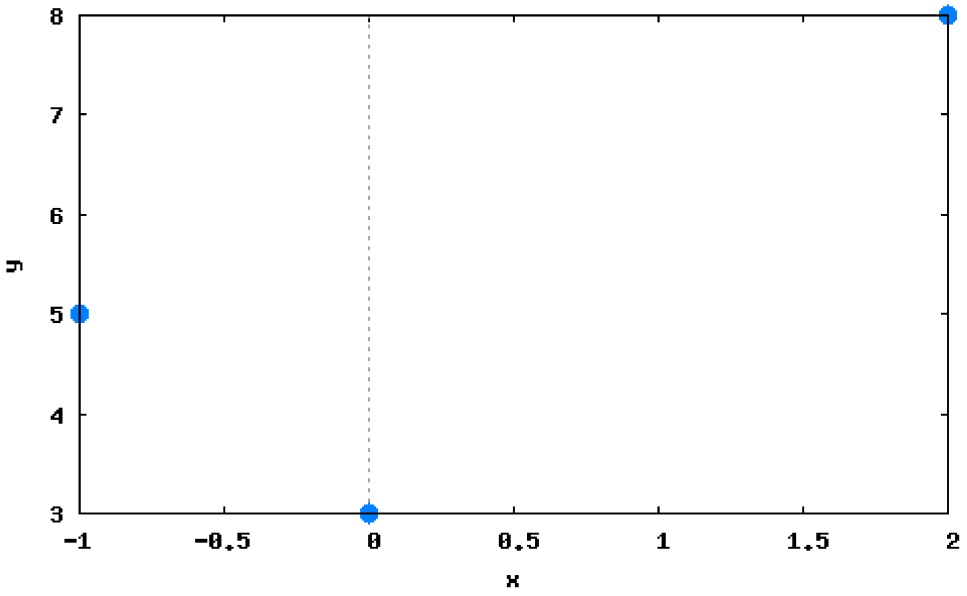
(%o88) [gr2d(*parametric*)]

11.3 Grafy 2D - graf na zobrazení dat

```
Pozor! U příkazu draw2d se musí uvést jak má příslušný bod
v grafu vypadat příkazem point_type. Jinak se body nevykreslí.
```

(%i89)
```
wxplot2d([discrete,[[0,3],[2,8],[-1,5]]],[style, points]);
```

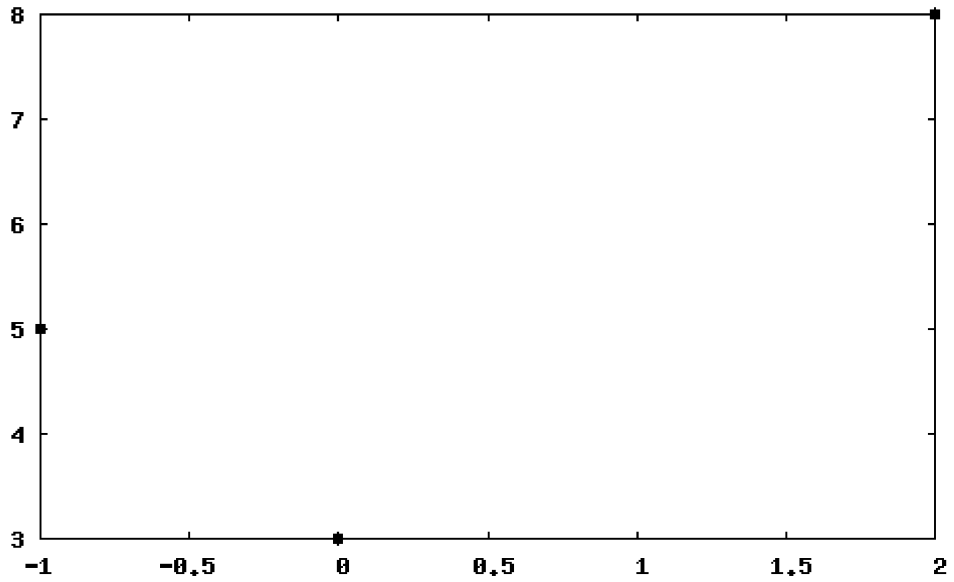
(%t89)



(%o89)

(%i90)
```
wxdraw2d(point_type=7,points([[0,3],[2,8],[-1,5]]));
```

(%t90)

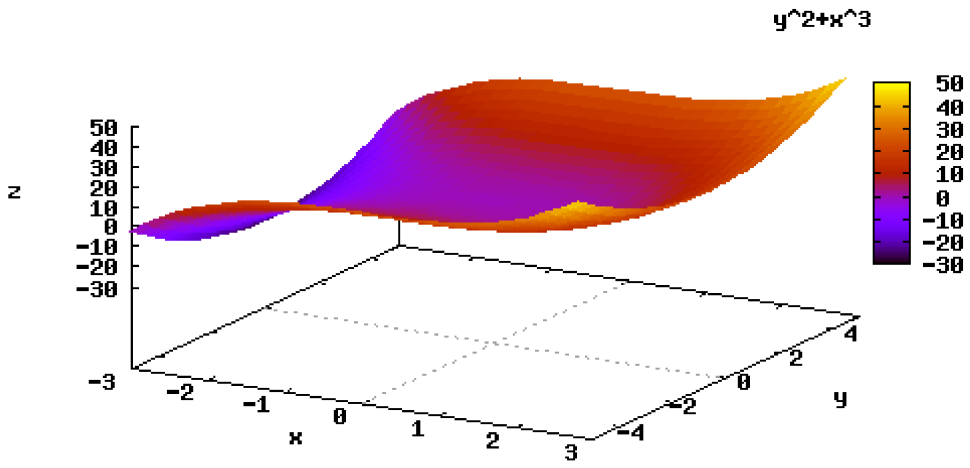


(%o90) [gr2d(*points*)]

11.4 Grafy 3D - graf x,y,z

(%i91)
```
wxplot3d(x^3+y^2,[x,-3,3],[y,-5,5]);
```

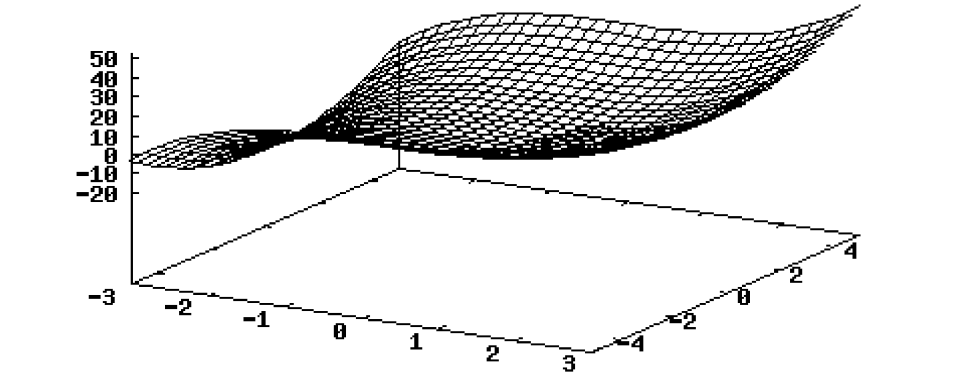
(%t91)



(%o91)

(%i92)
```
wxdraw3d(explicit(x^3+y^2,x,-3,3,y,-5,5));
```

(%t92)



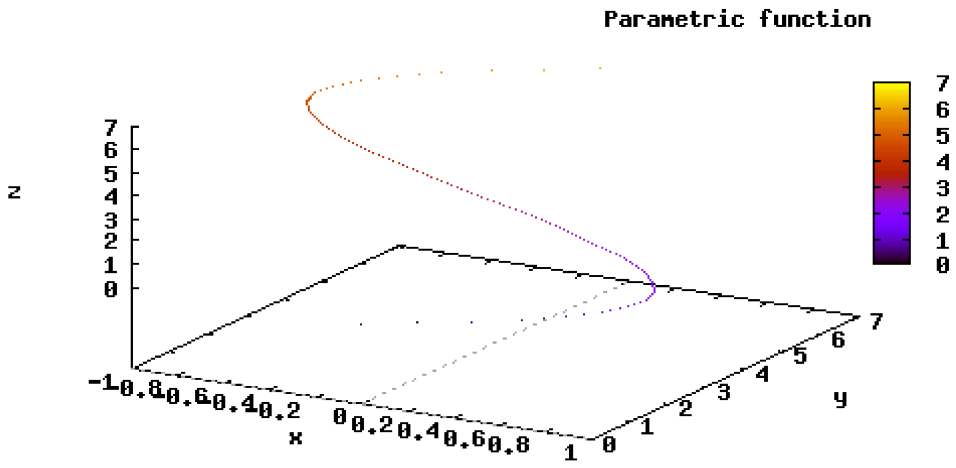
(%o92) [gr3d(*explicit*)]

11.5 Grafy 3D - parametrický graf - křivka

Příkaz plot neumožňuje vykreslení 3d křivky. Křivku je ale možné nakreslit trikem jakožto plochu, přičemž hodnoty druhého parametru jsou malé.

(%i93) `wxplot3d([sin(t),t,t+u],[t,0,2*%pi],[u,0,0.001]);`

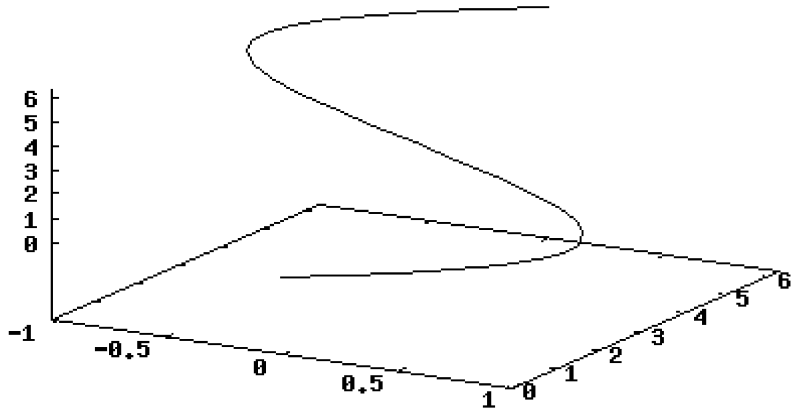
(%t93)



(%o93)

(%i94) `wxdraw3d(parametric(sin(t),t,t,t,0,2*%pi));`

(%t94)

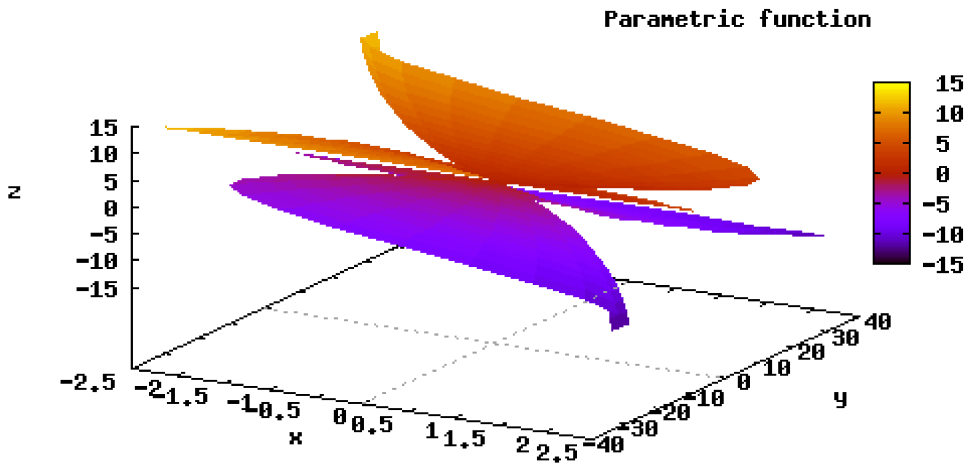


(%o94) [gr3d(*parametric*)]

11.6 Grafy 3D - parametrický graf - plocha

```
(%i95) wxplot3d([u*sin(t),t*u^2,t*u],[t,-5,5],[u,-2.5,2.5]);
```

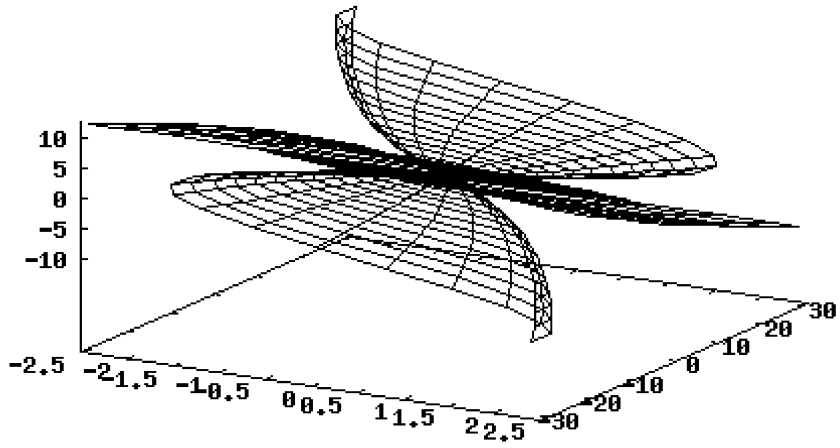
(%t95)



(%o95)

```
(%i96) wxdraw3d(parametric_surface
       (u*sin(t),t*u^2,t*u,t,-5,5,u,-2.5,2.5));
```

(%t96)



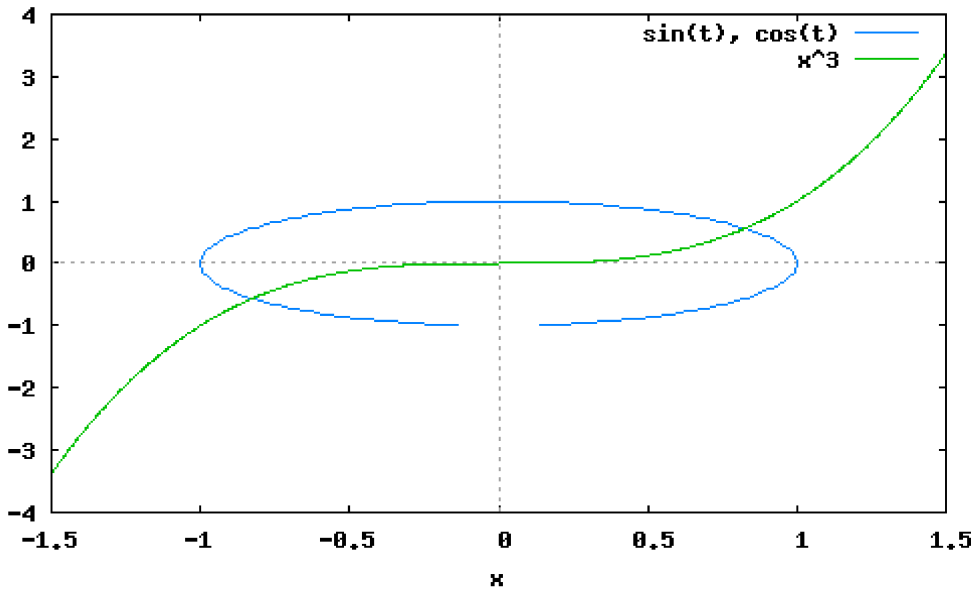
(%o96) [gr3d(*parametric_surface*)]

11.7 Zobrazení více typů grafů v jednom

Pomocí příkazu typu plot je možné zobrazit více typů grafů v jednom jen u plot2d. Křivka parametrického grafu se ale nemusí zobrazit celá.

```
(%i97) wxplot2d([[parametric,sin(t),cos(t)],[t,0,2*%pi],x^3],
       [x,-1.5,1.5],[nticks, 200]);
plot2d: expression evaluates to non-numeric value somewhere in plotting rang
```

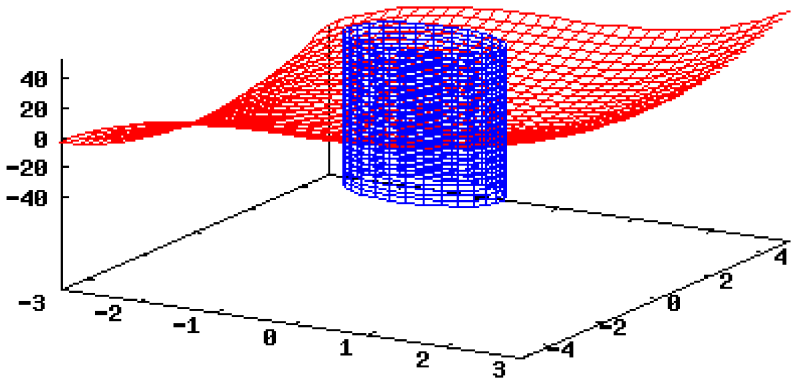
(%t97)



(%o97)

```
(%i98) wxdraw3d(
       color=red,
       explicit(x^3+y^2,x,-3,3,y,-5,5),
       color=blue,
       parametric_surface(sin(t),cos(t),u,t,0,2*%pi,u,-50,50)
       );
```

(%t98)



(%o98) [gr3d(*explicit*,*parametric_surface*)]

12 Fyzikální jednotky

Pro práci s jednotkami se dají použít 2(3) balíčky - unit,(units), ezunits. Každý pracuje s jednotkami svým vlastním způsobem, přičemž balíček unit a units k sobě mají blízko typem zápisu, ale balíček units není podporován nápovědou. Ne všechny jednotky jsou však v balíčcích definovány (např. balíček unit nemá hodinu). Je dobré zvolit jeden ze stylu zápisu jednotek - unit nebo ezunits.

12.1 Fyzikální jednotky - balíček unit

```
(%i99) kill(all);
(%o0) done
```

```
(%i1) load(unit);
*************************************************************************
*                          Units version 0.50                           *
*          Definitions based on the NIST Reference on                   *
*              Constants, Units, and Uncertainty                        *
*       Conversion factors from various sources including               *
*                    NIST and the GNU units package                     *
*************************************************************************

Redefining necessary functions...
Initializing unit arrays...
Done.
(%o1)
C:/PROGRA~1/Maxima/share/maxima/5.21.1/share/contrib/unit/unit.mac
```

```
(%i2) convert(50.0*km/s,[m,s]);
rat: replaced 50.0 by 50/1 = 50.0
(%o2) 50000 m/s
```

```
(%i3) mass:10*kg;
      v:7.9*km/s;
      Energie:1/2*mass*v^2;
      convert(Energie,[J]);
(%o3) 10 kg
(%o4) 7900 m/s
(%o5) 312050000 kg m^2/s^2
rat: replaced 312.05 by 6241/20 = 312.05
(%o6) 312050000 J
```

12.2 Fyzikální jednotky - balíček ezunits

```
(%i7) kill(all);
remvalue: warning: cannot remove value of:
structures
(%o0) done
```

```
(%i1) load(ezunits)$
Compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
End of Pass 1.
End of Pass 2.
OPTIMIZE levels: Safety=1 (No runtime error checking), Space=3, Speed=3
Finished compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
Compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
End of Pass 1.
End of Pass 2.
OPTIMIZE levels: Safety=1 (No runtime error checking), Space=3, Speed=3
Finished compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
Compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
End of Pass 1.
End of Pass 2.
OPTIMIZE levels: Safety=1 (No runtime error checking), Space=3, Speed=3
Finished compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
Compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
End of Pass 1.
End of Pass 2.
OPTIMIZE levels: Safety=1 (No runtime error checking), Space=3, Speed=3
Finished compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
Compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
End of Pass 1.
End of Pass 2.
OPTIMIZE levels: Safety=1 (No runtime error checking), Space=3, Speed=3
Finished compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
Compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
End of Pass 1.
End of Pass 2.
OPTIMIZE levels: Safety=1 (No runtime error checking), Space=3, Speed=3
Finished compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
Compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
End of Pass 1.
End of Pass 2.
OPTIMIZE levels: Safety=1 (No runtime error checking), Space=3, Speed=3
Finished compiling C:/Users/Radim/AppData/Local/Temp/gazonk_4676_0.lsp.
```

```
(%i2) 50.0`km/hour``m/s;
ezunits: computing conversions to base units; may take a moment.
                             m
(%o2) 13.88888888888889  `   -
                             s
```

```
(%i3) mass:10`kg;
      v:7.9`km/s;
      Energie:1/2*mass*v^2;
      Energie``[J];
```

(%o3) $10 \ ` \ kg$

(%o4) $7.9 \ ` \ \frac{km}{s}$

(%o5) $312.05 \ ` \ \frac{kg\ km^2}{s^2}$

(%o6) $[3.1205\,10^8 \ ` \ J]$

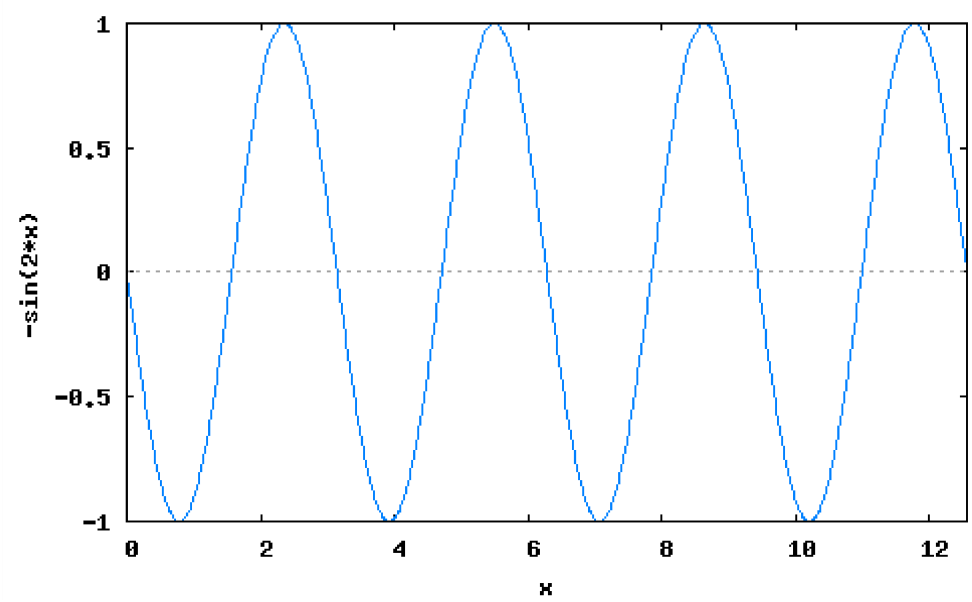
13 Zajímavosti pro učitele

13.1 Interaktivní změna dokumentu

wxMaxima umožňuje měnit interaktivně pouze grafy.
Použité příkazy jsou with_slider (využívající plot2d),
with_slider_draw a with_slider_draw3d.

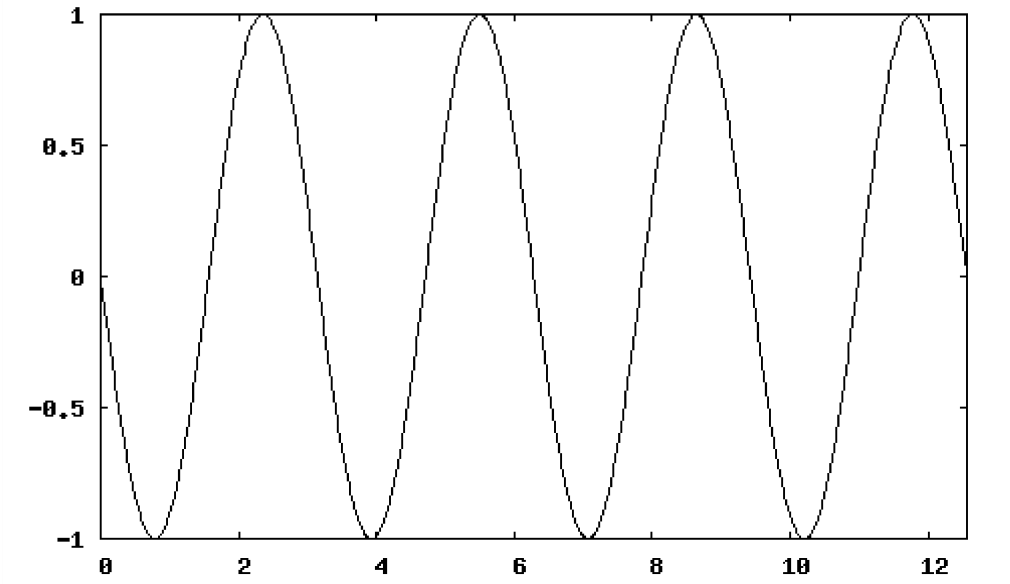
```
(%i7) with_slider(
          a, makelist(i, i, -20, 20)/10,
          sin(a*x),[x,0,4*%pi])$
```

(%t7)



```
(%i8) with_slider_draw(
        a, makelist(i, i, -20, 20)/10,
       explicit(sin(a*x),x,0,4*%pi))$
```

(%t8)



13.2 Převedení na html

Soubor je možné převést na html stránku za pomoci nabídky File/Export.

13.3 Převedení do pdfLaTeXu

Je možné převést jen určitý výraz příkazem tex, případně převést celý dokument za pomocí File/Export. Při exportu celého dokumentu se špatně vyexportují znaky $ a mezer \quad.

14 Závěrečné shrnutí

Výše nastíněné funkce jsou jen základní funkce programu, které by měl student znát, pro snadnější práci s programem během studia. Samozřejmě, že Maxima nabízí mnohem více funkcí. Mnoho funkcí jako např. plot, případně to_poly_solve mají množství nastavení, které můžeme nastavit tak, aby výsledek odpovídal našemu očekávání (např. změnit vzhled grafu, případně nastavit podmínky pro řešené rovnice).
Problémem tohoto programu ale je, že nemá až tak přehlednou nápovědu (případně k danému příkazu není) a tudíž pro řešení problémů je někdy lepší zkusit internetový prohlížeč. Taktéž se může stát, že některé funkce nemusí dobře fungovat - nedokreslený graf, špatně spočtený integrál s odmocninami apod..
Na druhou stranu se jedná o projekt typu GNU, takže je možné jej zdarma využívat při výuce a také je možné upravit si program a balíčky podle svých představ.

15 Doporučená literatura

Jelikož Maxima je GNU projekt, literatura je zdarma ke stažení ve formátu .pdf. Všem autorům patří velký dík.
http://wxmaxima.sourceforge.net/wiki/index.php/Animations
http://maxima.sourceforge.net/docs/manual/en/maxima.pdf
http://maxima.sourceforge.net/docs/tutorial/en/minimal-maxima.pdf
http://www.telefonica.net/web2/biomates/maxima/gpdraw/index.html

16 Odkazy

http://maxima.sourceforge.net/
http://wxmaxima.sourceforge.net/
http://www.ma.utexas.edu/pipermail/maxima/