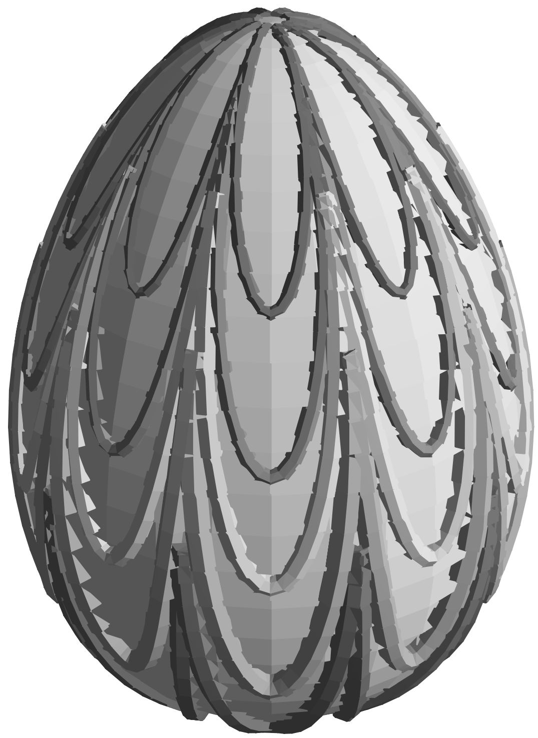
GRAFIKA V MAPLU -- I. CAST

Uvod

Velikonocni vajicko

```
> with(plots):setoptions3d(scaling=constrained,project
  ion=.5,style=patchnogrid):
Warning, the name changecoords has been redefined
```

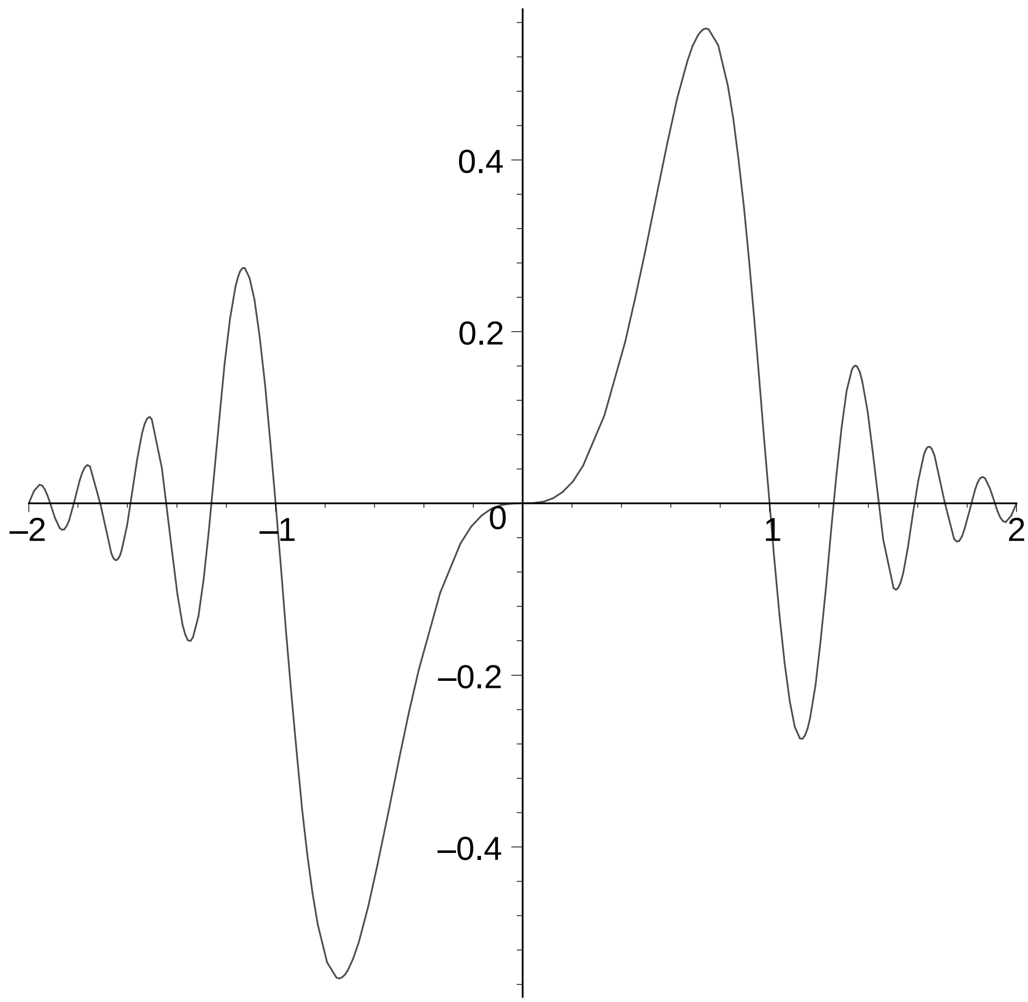
```
> X:=cos(x)*cos(y)*(1+.2*sin(y)):
  Y:=sin(x)*cos(y)*(1+.2*sin(y)):
  Z:=1.5*sin(y):
  a:=plot3d([X,Y,Z],x=-Pi..Pi,y=-Pi/2..Pi/2,color=[1,.
  8,.5],ambientlight=[.4,.4,.4],light=[75,50,.8,.8,.7]
  ,grid=[30,30]):
  p:=t+.1*sin(10*t):
  q:=.5*cos(10*t):
  X0t:=cos(p)*cos(q-.3)*(1+.2*sin(q-.3)):
  Y0t:=sin(p)*cos(q-.3)*(1+.2*sin(q-.3)):
  Z0t:=1.5*sin(q-.3):
  X1t:=cos(p)*cos(q-.6)*(1+.2*sin(q-.6)):
  Y1t:=sin(p)*cos(q-.6)*(1+.2*sin(q-.6)):
  Z1t:=1.5*sin(q-.6):
  X2t:=cos(p)*cos(q+.1)*(1+.2*sin(q+.1)):
  Y2t:=sin(p)*cos(q+.1)*(1+.2*sin(q+.1)):
  Z2t:=1.5*sin(q+.1):
  X3t:=cos(p)*cos(q+.6)*(1+.2*sin(q+.6)):
  Y3t:=sin(p)*cos(q+.6)*(1+.2*sin(q+.6)):
  Z3t:=1.5*sin(q+.6):
  X4t:=cos(p)*cos(q-1)*(1+.2*sin(q-1)):
  Y4t:=sin(p)*cos(q-1)*(1+.2*sin(q-1)):
  Z4t:=1.5*sin(q-1):
  b:=tubeplot({[X0t,Y0t,Z0t,radius=.04,color=[.9,.8,.2
  ]],[X1t,Y1t,Z1t,radius=.03,color=[.0,.7,1]],[X2t,Y2t
  ,Z2t,radius=.05,color=[.3,.9,.2]],[X3t,Y3t,Z3t,radiu
  s=.05,color=[.1,.6,.6]],[X4t,Y4t,Z4t,radius=.025,col
  or=[.8,.3,.2]]},t=-Pi..Pi,numpoints=300,tubepoints=7
  ):
  display3d({a,b},orientation=[0,-120]);
```



```
> restart;
> f:=x->exp(-x^2)*sin(Pi*x^3);
```

$$f := x \rightarrow e^{(-x^2)} \sin(\pi\, x^3)$$

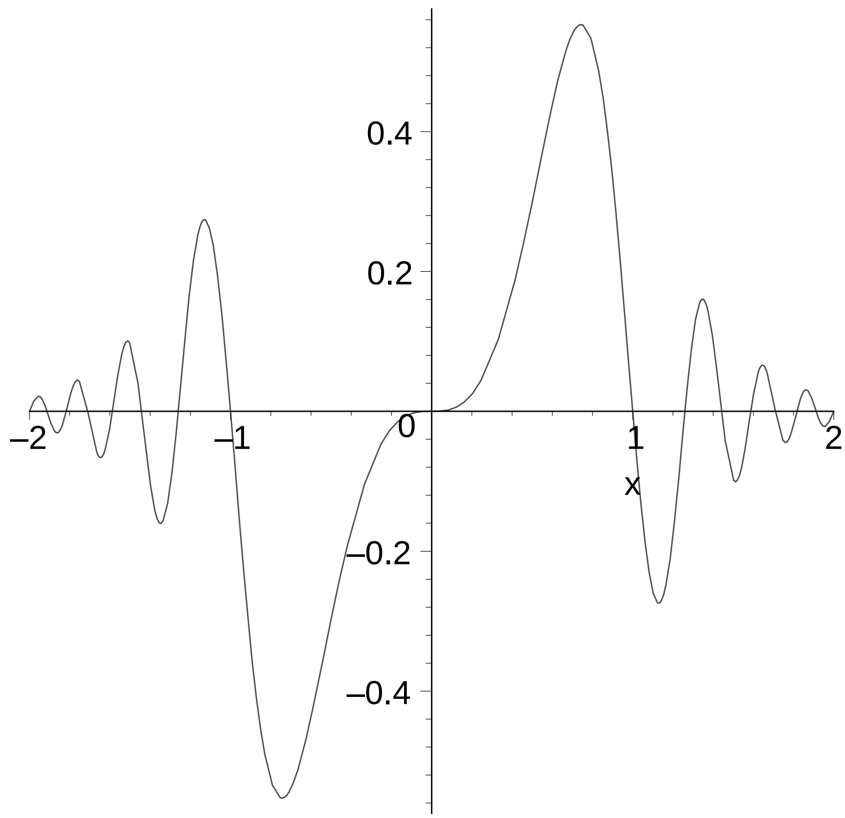
```
> plot(f, -2..2);
```



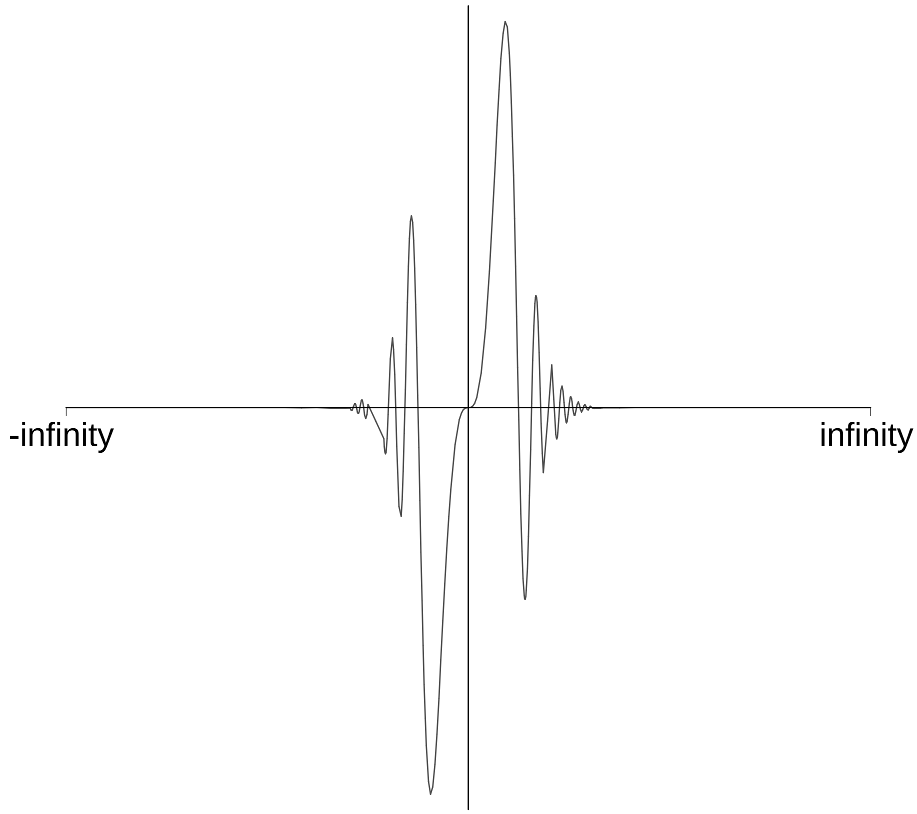
Prikaz pro nakresleni funkce plot(f, a..b, options), a..b interval na ose x, options jsou nepovinne.

Pri kresleni formule je treba pouzit plot(f(x), x=a..b, options).

```
> plot(f(x), x=-2..2);
```



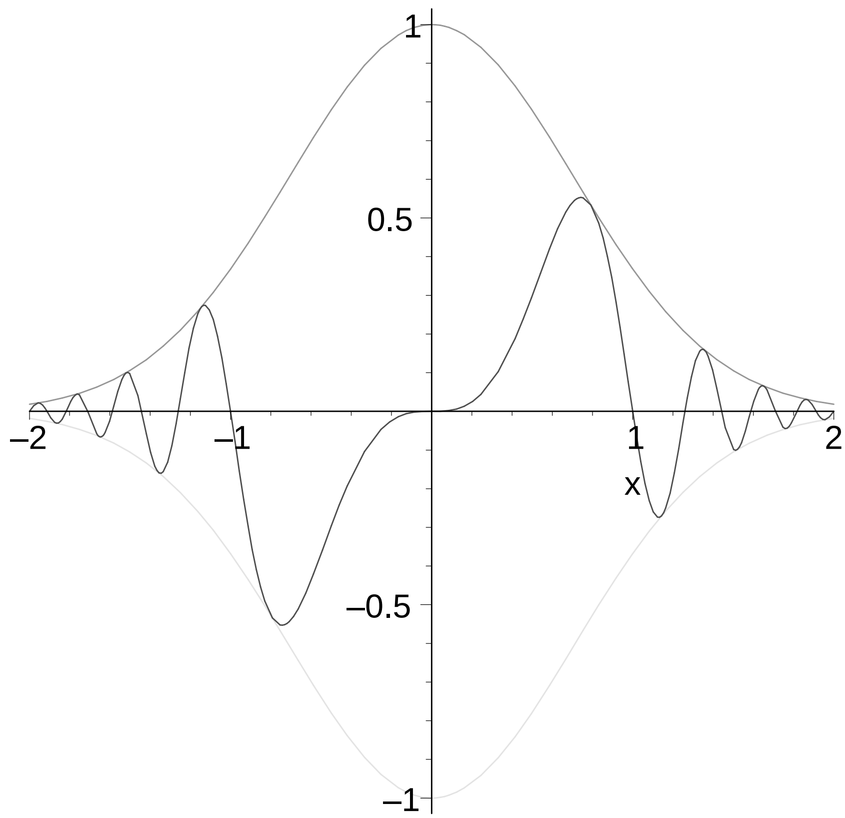

```
> plot(f, -infinity..infinity);
```



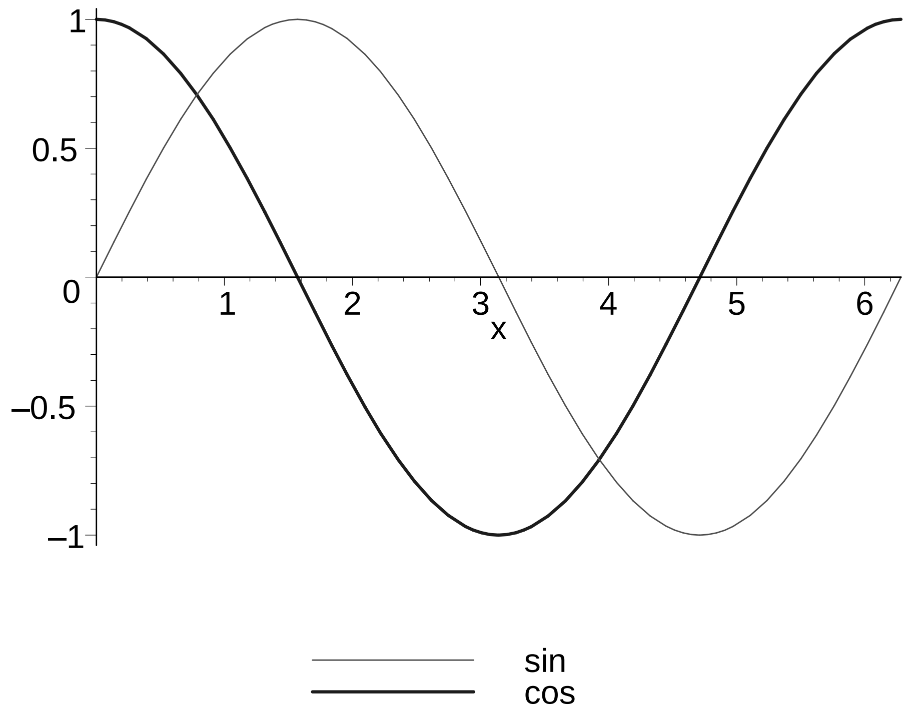
V tomto pripade Maple transformuje realnou osu na interval (-1,1).

Vice funkci v jednom obrazku:

```
> plot({f(x), exp(-x^2), -exp(-x^2)}, x=-2..2);
```

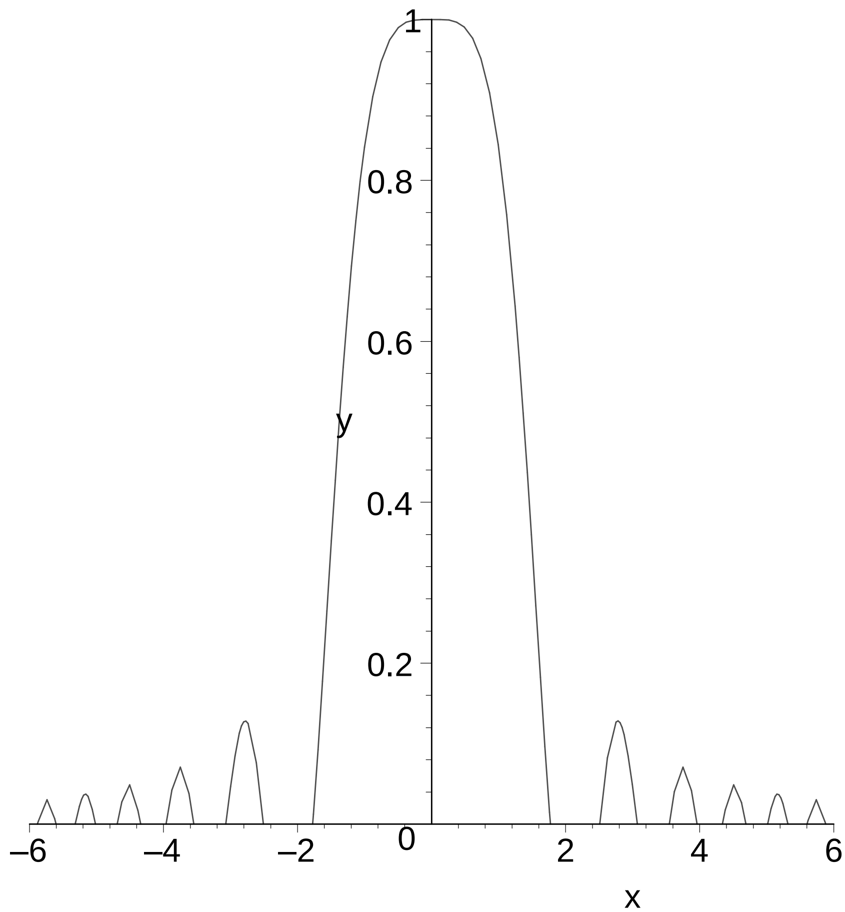



```
> plot([sin(x), cos(x)], x=0..2*Pi, color=[red, blue],
  thickness=[2,3], legend=["sin", "cos"]);
```



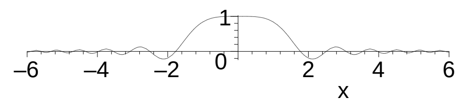
Pokud chceme omezit rozsah zobrazovanych hodnot na ose y, musime to Maplu sdelit:

```
> plot(sin(x^2)/x^2, x=-6..6, y=0..1);
```



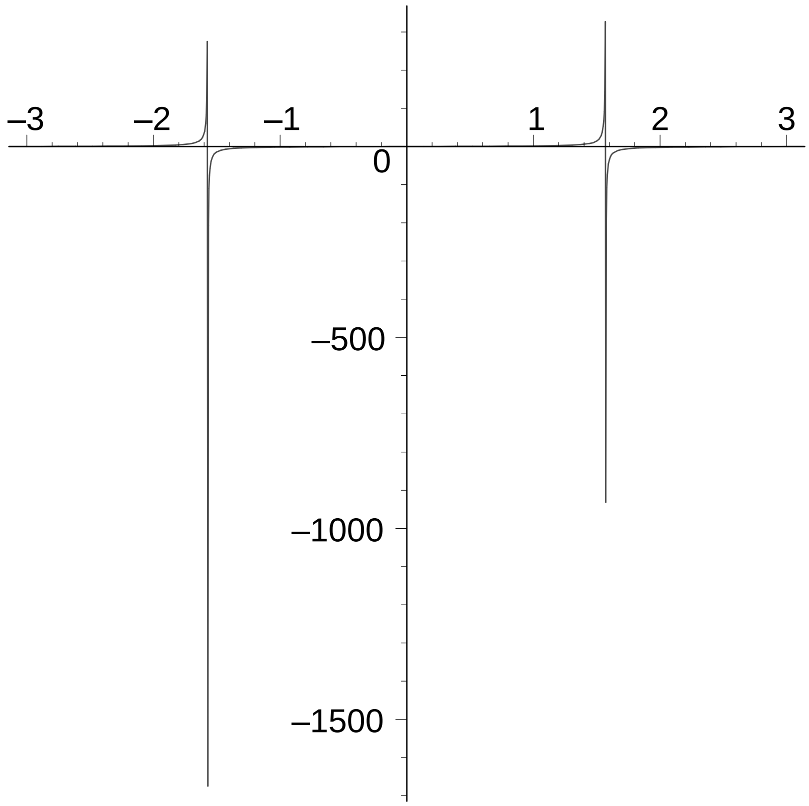
Maple voli meritko na obou osach tak, aby obrazek co nejlepe "zaplnil" display. Pokud chceme stejne meritko na osach, pouzijeme volbu scaling=constrained.

```
> plot(sin(x^2)/x^2, x=-6..6, scaling=constrained);
```

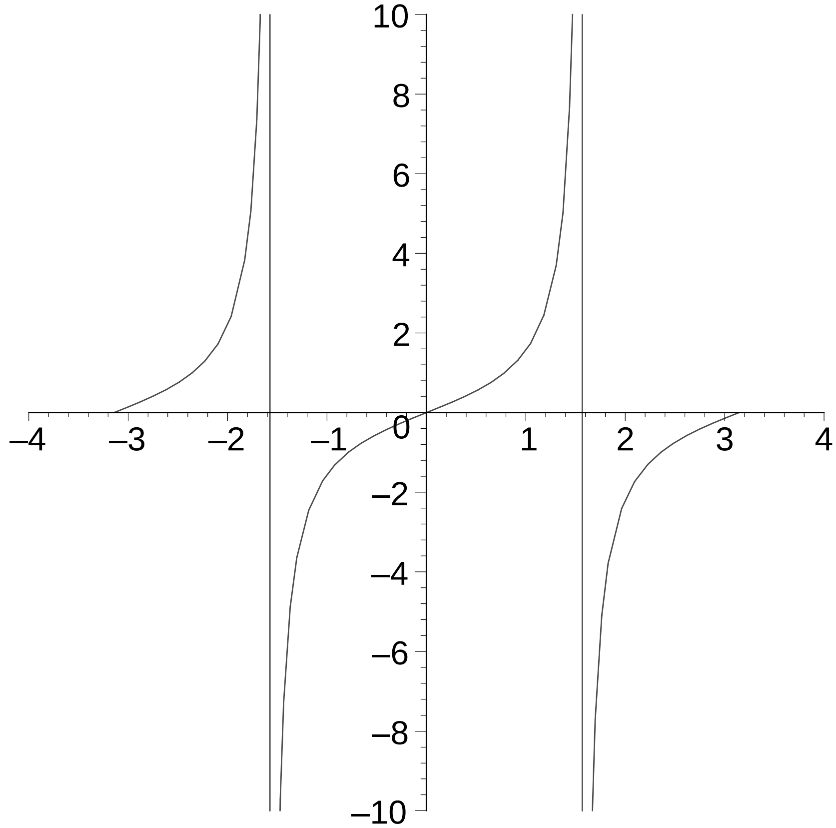


Nekdy je vymezeni rozsahu zobrazovanych hodnot nutne:

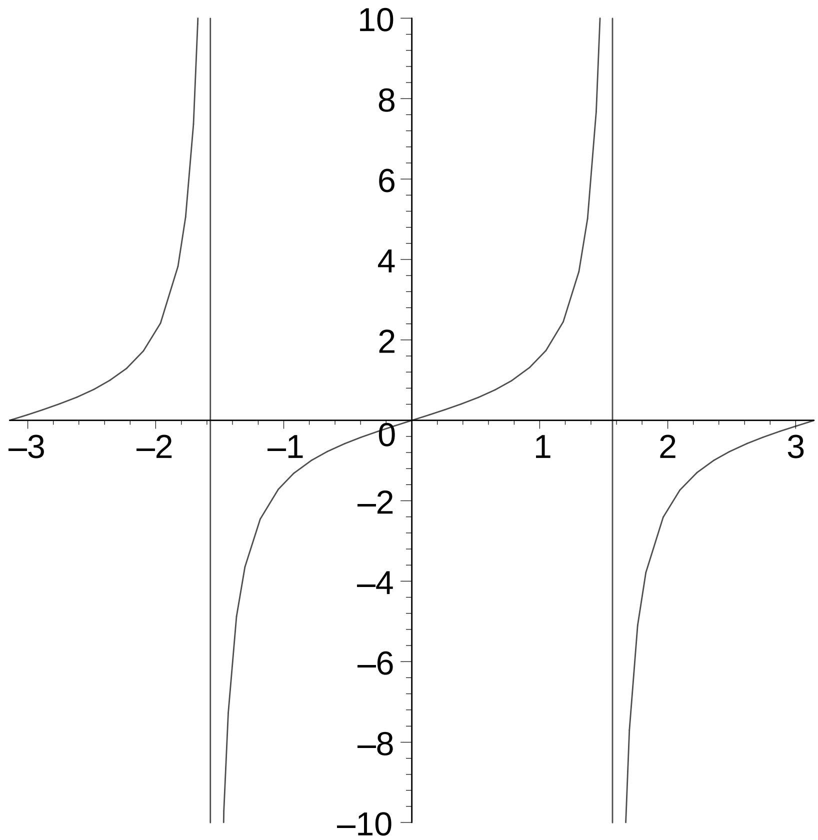
```
> plot(tan, -Pi..Pi);
```




```
> with(plots):
Warning, the name changecoords has been redefined
> display(%%, view=[-4..4,-10..10]);
```

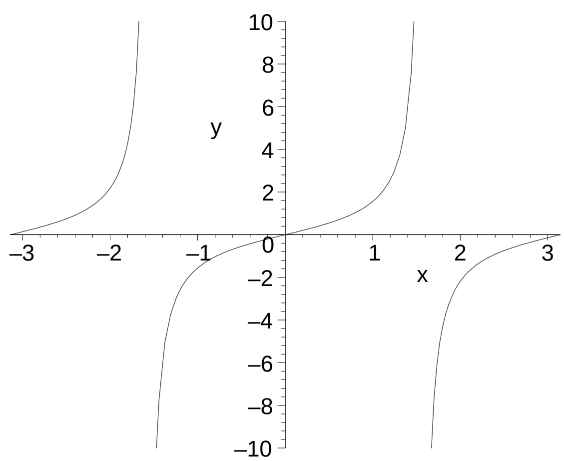



```
> plot(tan, -Pi..Pi, -10..10);
```

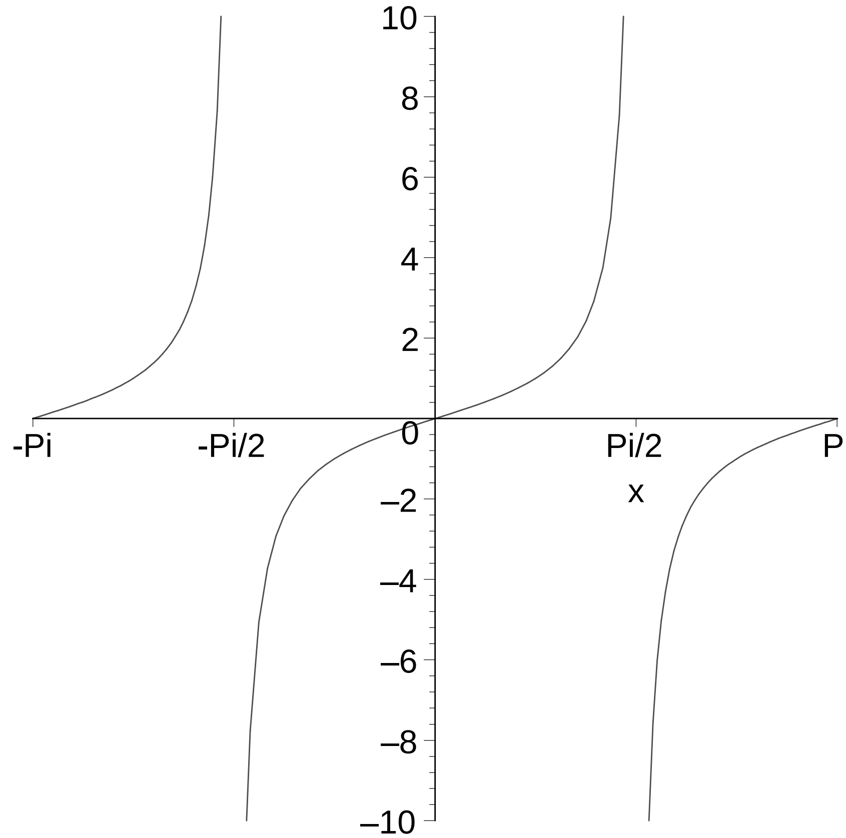


Zamezeni "spojovani" nespojitych funkci provedeme pomoci volby discont=true.

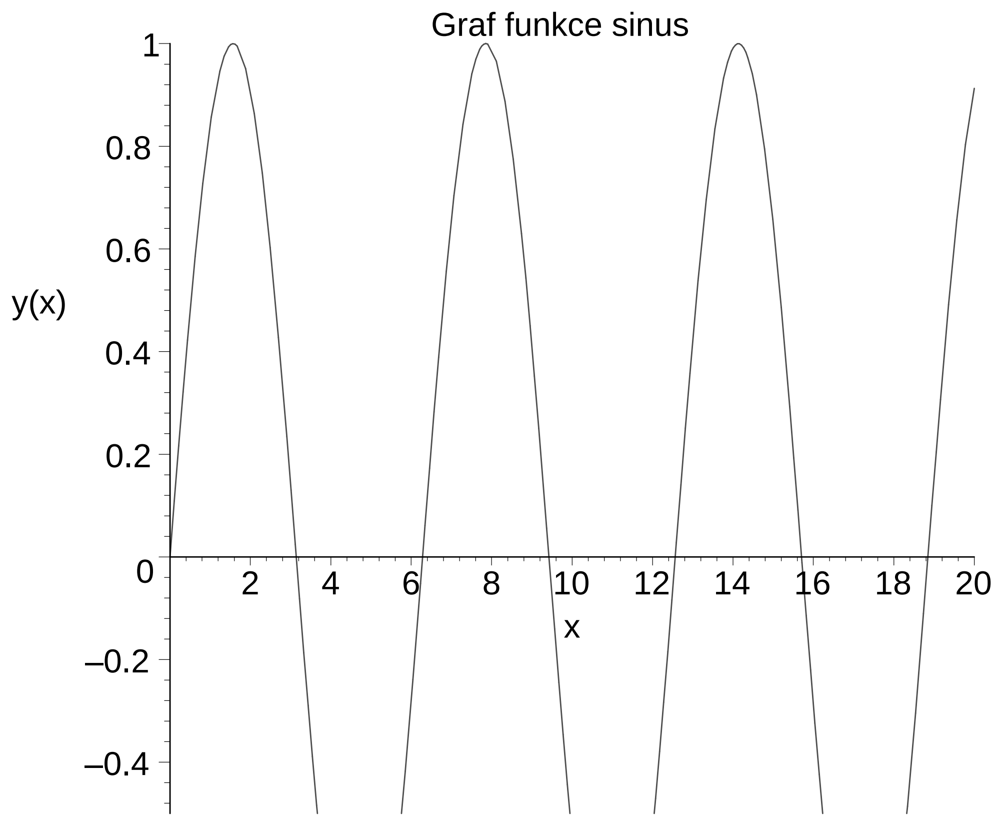
```
> plot(tan(x), x=-Pi..Pi, y=-10..10, discont=true);
```



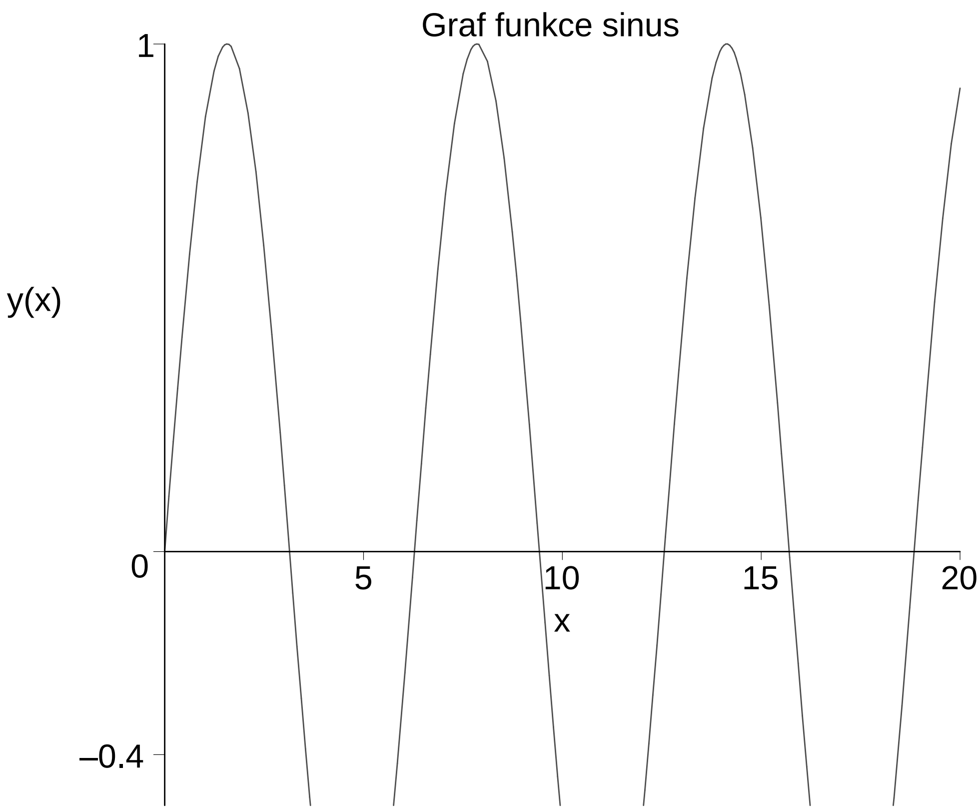
```
> plot(tan(x), x=-Pi..Pi, -10..10, discont=true,
  xtickmarks=[-3.14=‘-Pi‘, -1.57=‘-Pi/2‘, 1.57=‘Pi/2‘,
  3.14=‘Pi‘]);
```




```
> plot(sin(x), x=0..20, ‘y(x)‘=-0.5..1, xtickmarks=8,
  ytickmarks=4, title=‘Graf funkce sinus‘);
```



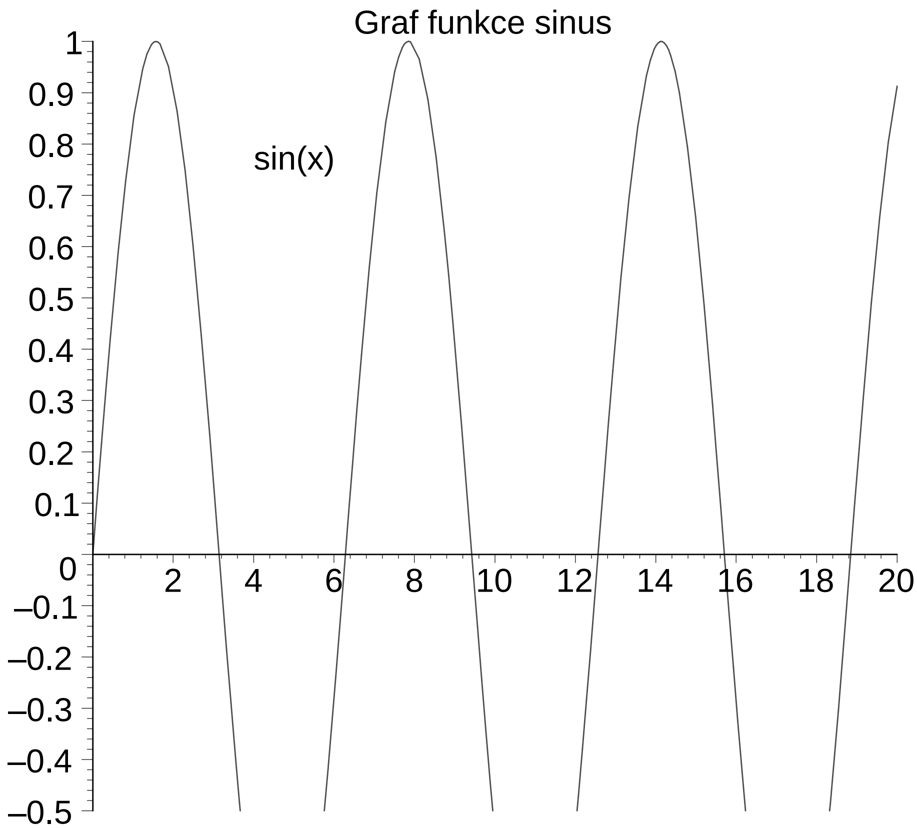

```
> plot(sin(x), x=0..20, ‘y(x)‘=-0.5..1,
  xtickmarks=[5,10,15,20], ytickmarks=[-0.4,0,1],
  title=‘Graf funkce sinus‘);
```




```
> with(plots):
> plot1:=plot(sin, 0..20, -0.5..1, tickmarks=[5,9],
  title= ‘Graf funkce sinus‘):
> plot2:=textplot({[4,0.75,‘sin(x)‘]}, align={ABOVE,
  RIGHT}):
```

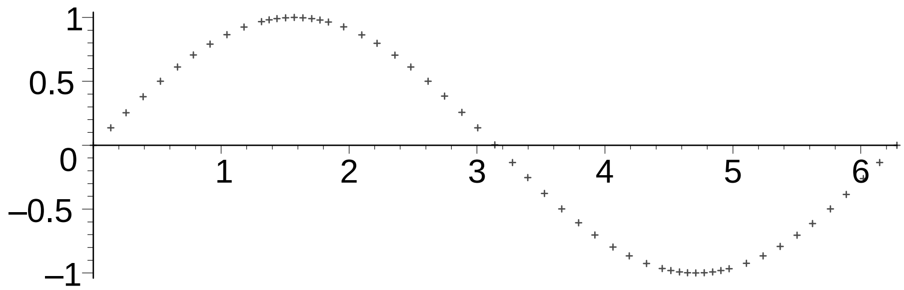
Implicitni nastaveni je centrovani horozontalni i vertikalni. Zarovnani menime parametrem align=t, kde t muze byt BELOW,RIGHT,ABOVE a LEFT.

```
> display({plot1, plot2});
```

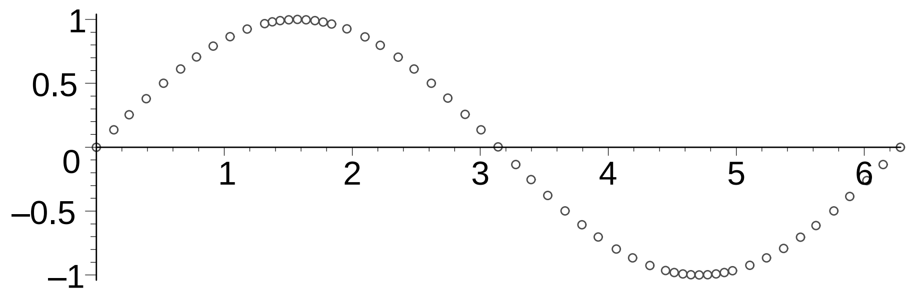


Pro znazornovani dat se pouziva style=point.

```
> plot(sin, 0..2*Pi, scaling=constrained, style=point);
```



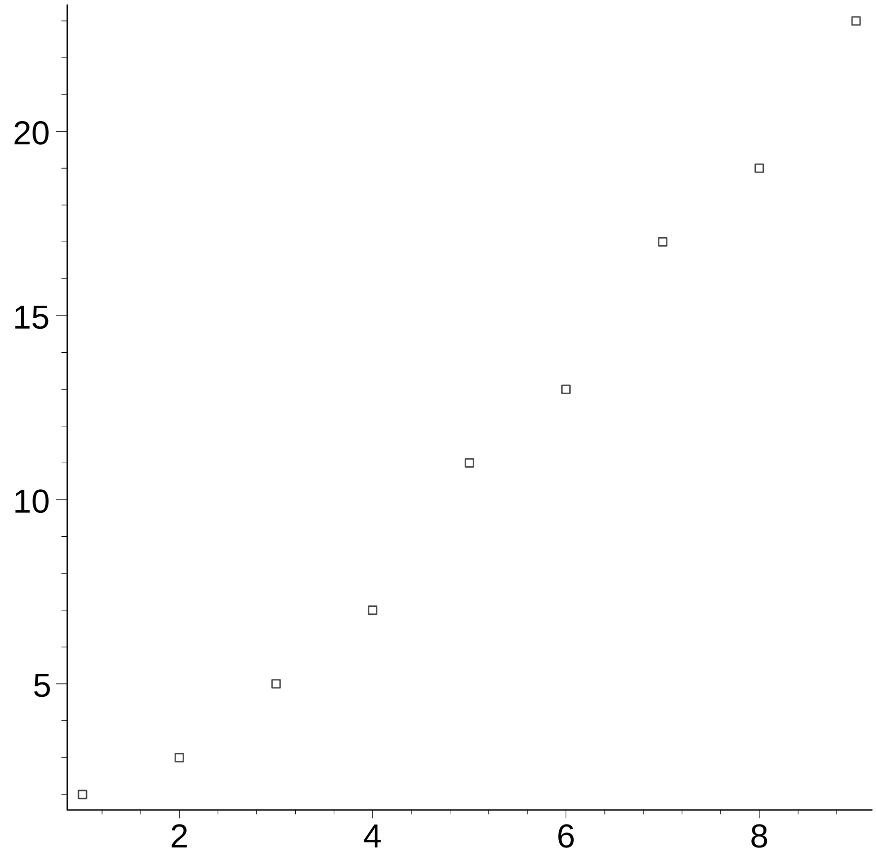

```
> plot(sin, 0..2*Pi, scaling=constrained, style=point,
  symbol=circle);
```



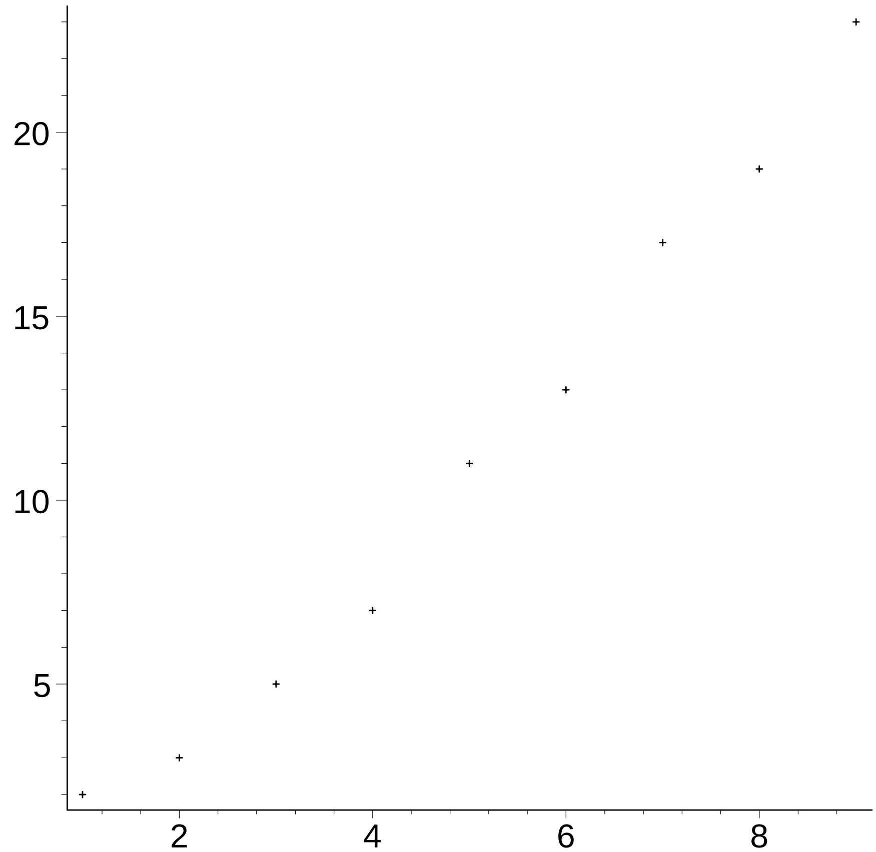

```
> plotpoints:=[seq([i, ithprime(i)], i=1..9)];
```

$$plotpoints := [[1, 2], [2, 3], [3, 5], [4, 7], [5, 11], [6, 13], [7, 17], [8, 19], [9, 23]]$$

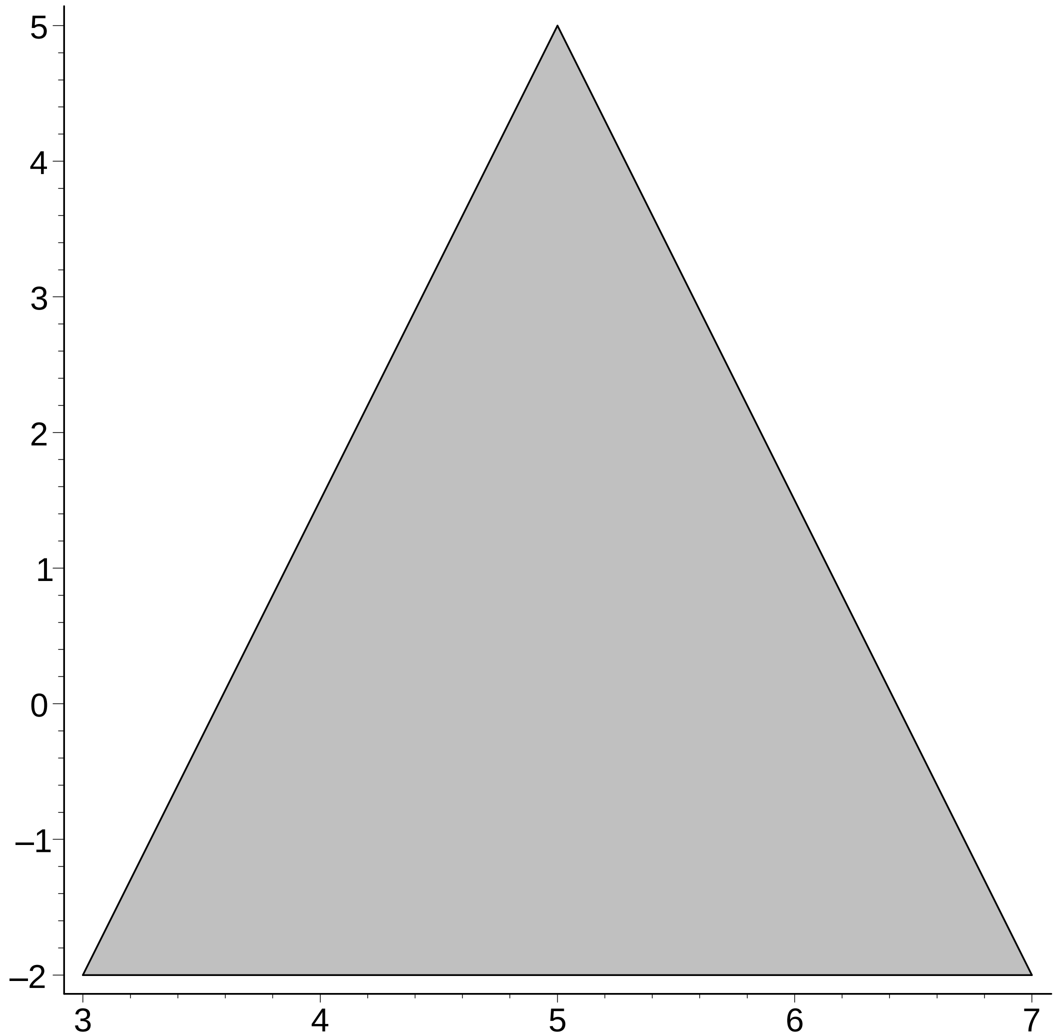
```
> plot(plotpoints, style=point, symbol=box);
```




```
> pointplot(plotpoints);
```




```
>
> plots[polygonplot]([[3,-2],[7,-2],[5,5]],color=grey,ax
  es=framed);
```

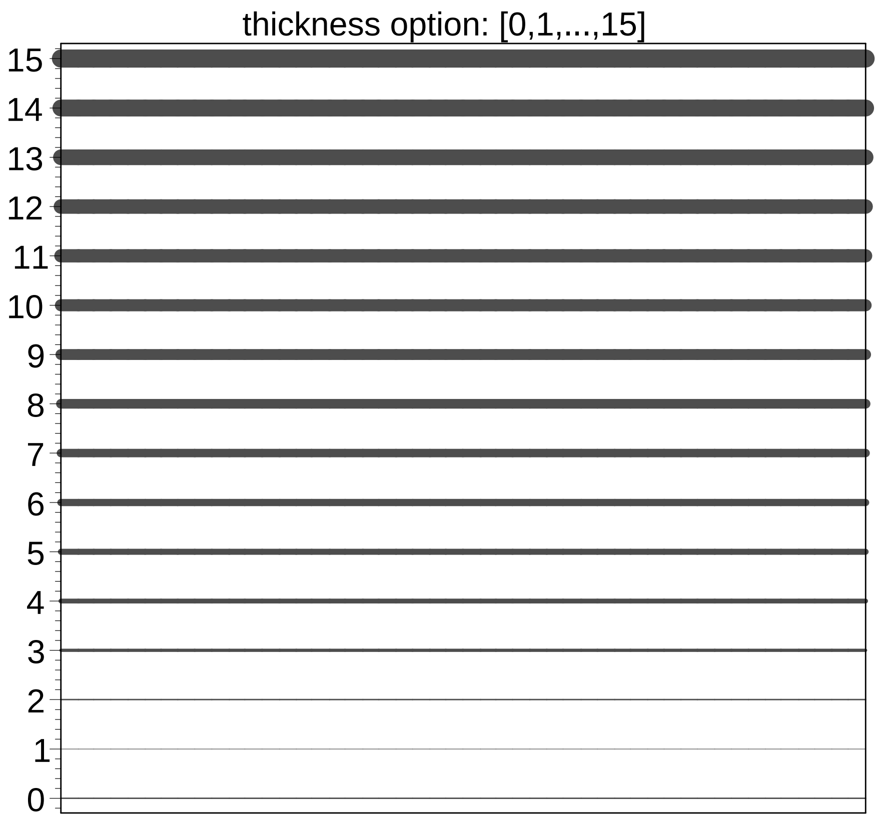


```
> ?plot[options]
```

Styl a tloustka car

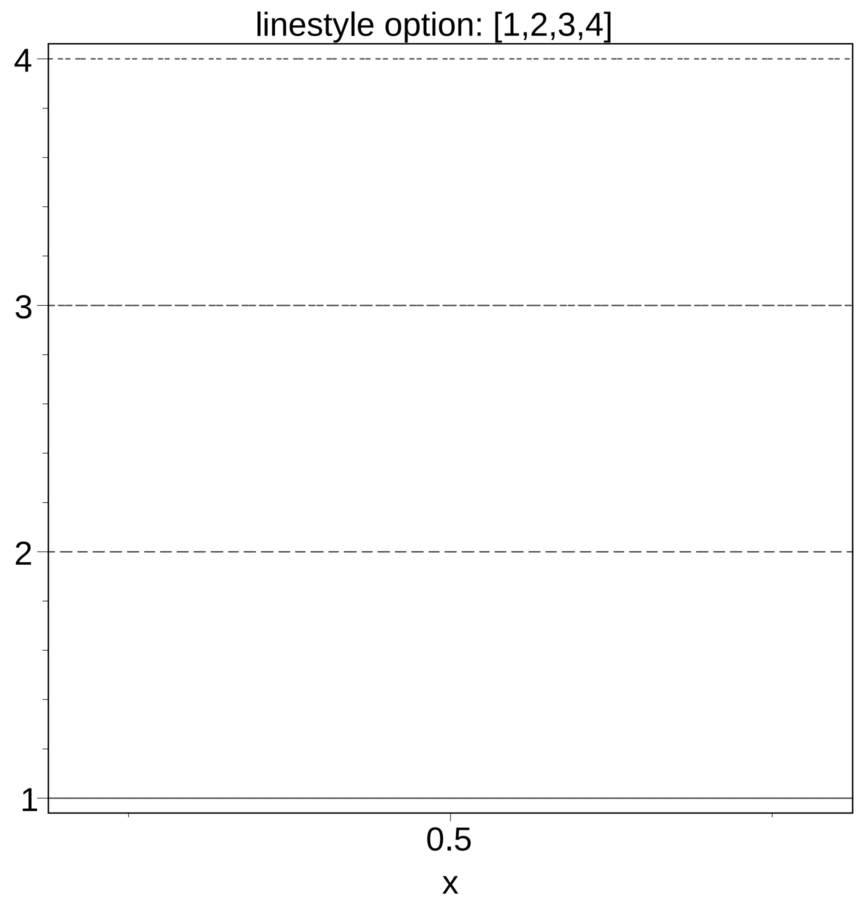
```
> for i from 0 to 15 do
  thick[i]:=plot(i, x=0..1, thickness=i):
  od:
```

```
> display(convert(thick, set), axes=box,
  tickmarks=[0,15], title=‘thickness option:
  [0,1,...,15]‘ );
```




```
> for i from 1 to 4 do
  line[i]:=plot(i, x=0..1, linestyle=i):
  od:
```

```
> display(convert(line, set), axes=box, tickmarks=[1,4],
  title=‘linestyle option: [1,2,3,4]‘);
```

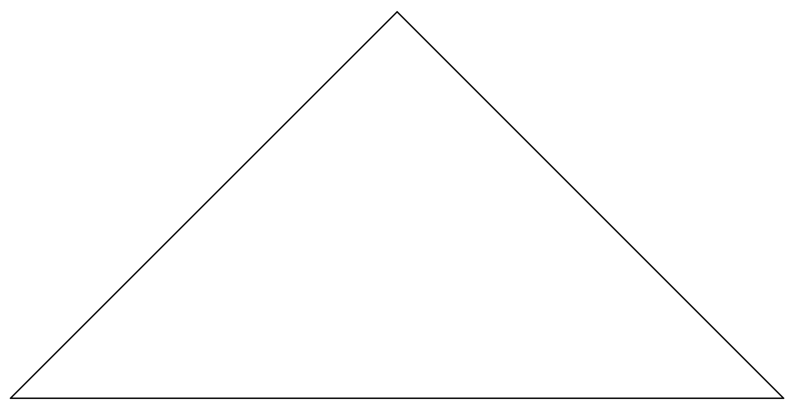



```
> ?linestyle
> interactive(sin(x), x);
Initializing Java runtime environment.
```

Struktura dvoj-dimenzionalni grafiky

Vytvareni obrazku probiha ve dvou fazich:
1) Jsou spocitany funkcni hodnoty v referencnich bodech a tyto jsou ulozeny do objektu datoveho typu PLOT.
2) Objekt je vykreslen na obrazovce.

```
> PLOT(CURVES([[1,1],[2,2], [3,1], [1,1]]),
  AXESSTYLE(NONE), SCALING(CONSTRAINED));
```



```
> P:=plot([[1,1], [2,2], [3,1], [1,1]], axes=none,
  scaling=constrained);
```

$$P := \mathrm{PLOT}(\mathrm{CURVES}($$

$$[[1., 1.], [2., 2.], [3., 1.], [1., 1.]],$$

$$\mathrm{COLOUR}(RGB, 1.0, 0., 0.)),$$

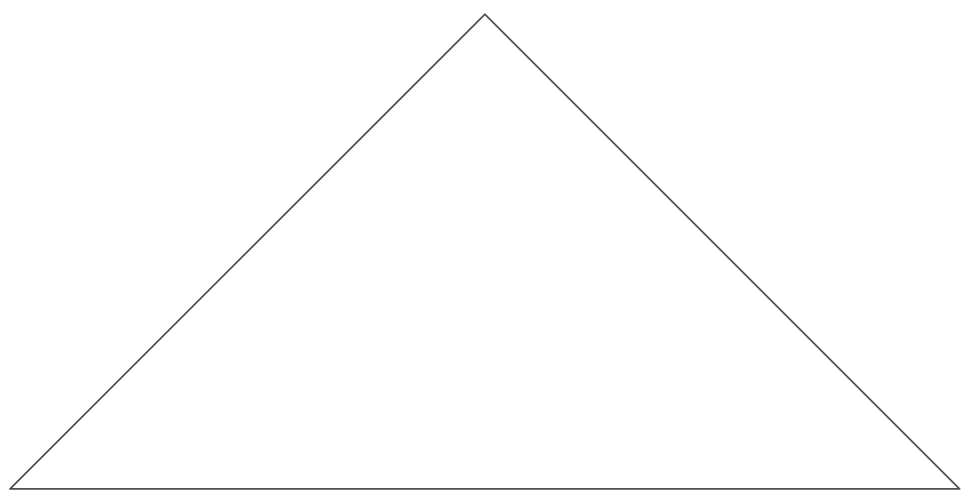
$$\mathrm{SCALING}(CONSTRAINED),$$

$$\mathrm{AXESSTYLE}(NONE),$$

AXESLABELS("", ""),

VIEW(*DEFAULT*, *DEFAULT*))

```
> P;
```



```
> f:=x->sqrt(2)-sqrt(2*sqrt(x)):
> P:=plot(f(x), x=0..1, y=0..sqrt(2));
```

P := PLOT(CURVES([

[0., 1.4142135623730951], [

0.00068116106770833304,

1.1857444721970180], [

0.0013623221354166661,

1.1425164947775061], [

0.0020434832031249991,

1.1135313300294607], [

0.0027246442708333321,

1.0911094764630453], [

0.0040869664062499982,

1.0566401123151436], [

0.0054492885416666643,

1.0299758845224136], [

0.0081739328124999965,

0.98898467142810908], [

0.010898577083333329,

0.95727538202094120 1], [

0.016347865624999993,

0.90852833975660163 3], [

0.021797154166666657,

0.87081942718191707 2], [

0.031279947812500000 2,

0.81946759943148395 2], [

0.040762741458333334 8,

0.77876435265115373 5], [

0.062091527916666666 7,

0.70826495707936976 1], [

0.083561695416666664 8,

0.65385785782303207 4], [

0.10492981895833332 3,

0.60931763727380949 8], [

0.12474080479166666 0,

0.57375339913244549 6], [

0.14525393312499998 4,

0.54114776878358905 6], [

0.16646863979166665 1,

0.51088008465893575 1], [

0.18761531062499997 6,

0.48346565808545483 0], [

0.20936726083333334 6,

0.45758753967870746 7], [

0.22852639041666667 0,

0.43641569023022475 1], [

0.25009503249999998 1,

0.41411854341685616 6], [

0.27175223749999999 3,

0.39313706489956845 6], [

0.29262296749999999 7,

0.37407283836706817 2], [

0.31157558979166666 3,

0.357625123681850176], [

0.334112090833333320,

0.339016557617175084], [

0.353203463333333301,

0.323975760718647754], [

0.375411268124999975,

0.307228338718146032], [

0.395068938333333286,

0.293013226572828689], [

0.416636518125000022,

0.278014749677451211], [

0.437173954375000006,

0.264264594371818040], [

0.458602614583333345,

0.250424880036826503], [

0.478280813958333329,

0.238136613844476264], [

0.49950641041666658,

0.22530004620292776 1], [

0.52155383979166585,

0.21239257280346435 2], [

0.54074619187499949,

0.20148567300800435 1], [

0.56147445791666614,

0.19002730745293039 4], [

0.58288878499999965,

0.17851818311469269 7], [

0.60383852624999896,

0.16756166467584199 3], [

0.62410847812499921,

0.15722878451484634 3], [

0.64661480374999911,

0.14604667692899409 0], [

0.66683773833333263,

0.13624538297169830 3], [

0.68843003749999998 0,

0.12602348186487577 5], [

0.70799586979166662 0,

0.11696655086911889 1], [

0.72938649249999998 4,

0.10727724337491229 0], [

0.74951340145833322 8,

0.09835306794166975 84], [

0.77055182062499993 9,

0.08921483678266617 37], [

0.79112072083333329 1,

0.08045970725675721 80], [

0.81265445312499995 4,

0.07147495045080520 80], [

0.83339399666666669 2,

0.06298881774569831 12], [

0.854603175416666638,

0.0544727801778668308], [

0.875636731458333317,

0.0461823989272109880], [

0.894964384999999973,

0.0386950615003713860], [

0.917116057083333325,

0.0302614205047448959], [

0.936928776666666629,

0.0228467429151371082], [

0.958053268125000023,

0.0150695701281506889], [

0.978272101875000023,

0.00774536984891516233], [1., 0.]],

COLOUR(*RGB*, 1.0, 0., 0.)),

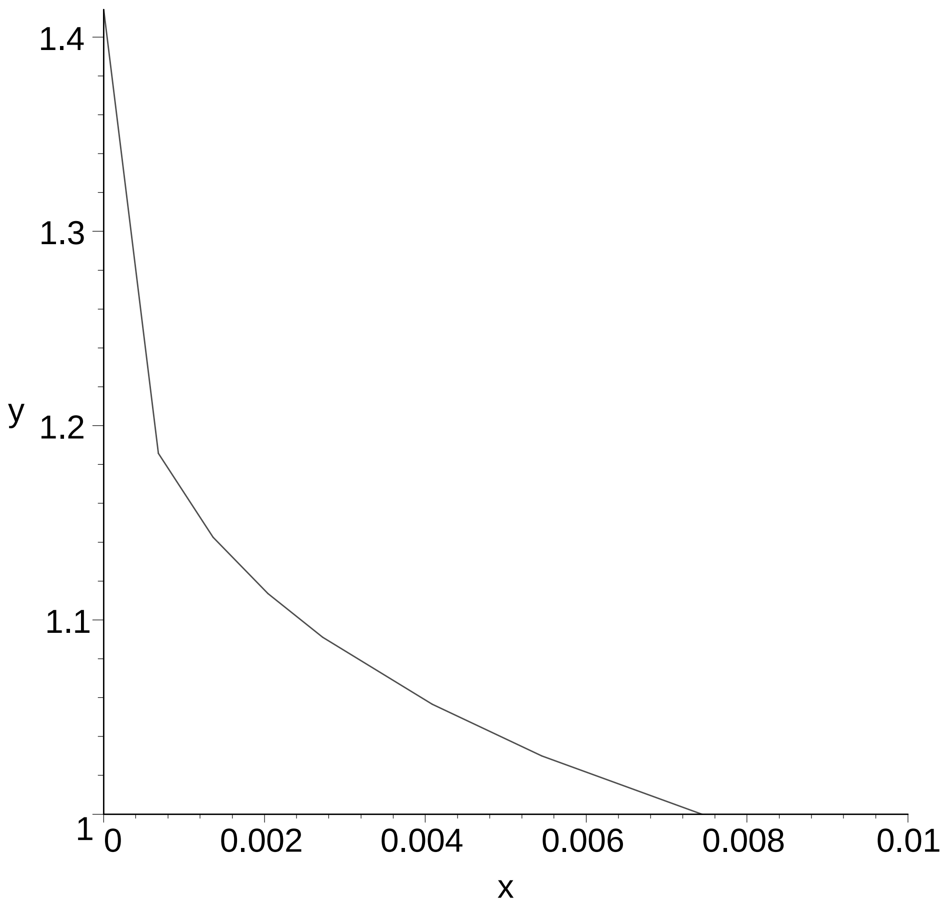
AXESLABELS("x", "y"),

VIEW(0. .. 1., 0. .. 1.414213562))

Nejdrive se vyhodnoti prvni argument prikazu plot (popisuje zadanou funkci), dale jsou vybrany equidistantni body ze zadaneho intervalu (implicitne 49 bodu) a numericky jsou v techto bodech spocteny funkcni hodnoty. Dale se maple diva na tyto body, jako by byly spojeny useckami a kontroluje, jestli mezi nekterymi useckami neni prilis velky uhel. Pokud ano, prida do teto oblasti dalsi referencni body. Maximalni pocet referencnich bodu je urcovan promennou resolution, jejiz implicitni hodnota je 200. Tedy pocet referencnich bodu je cele cislo z intervalu [49,200].

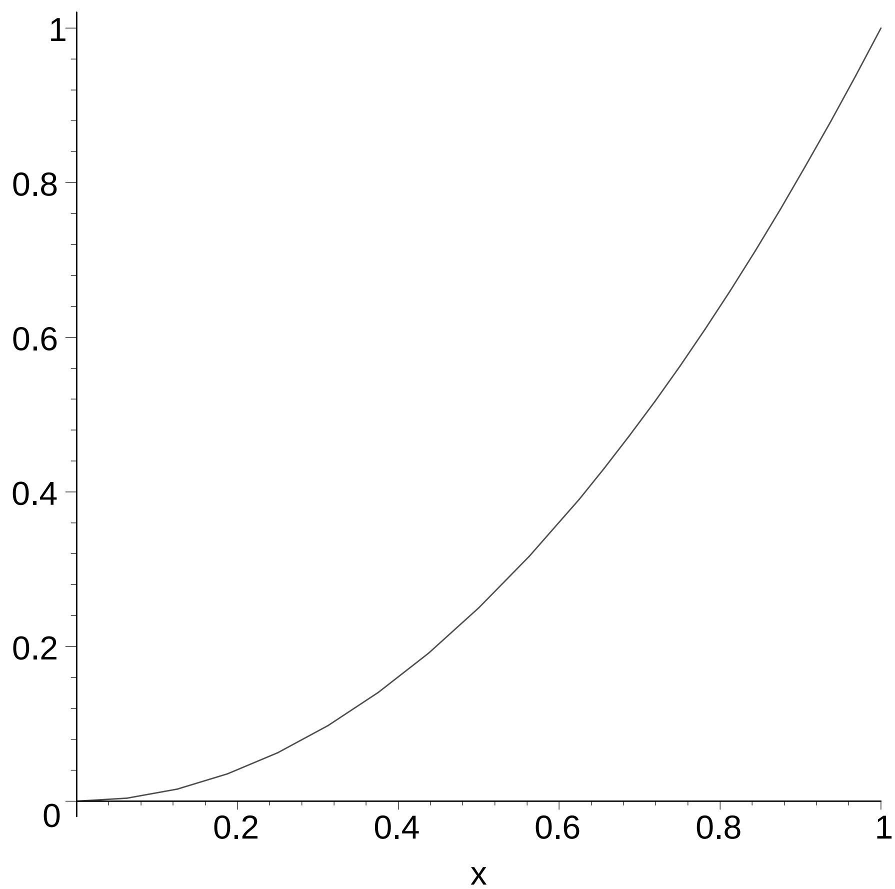
Pomoci funkce replot z balicku plots muzeme zvetsovat ci zmensovat drive nakresleny obrazek a ilustrovat vyse uvedenou skutecnost.

```
> plots[replot] (P, x=0..0.01, y=1..sqrt(2));
```

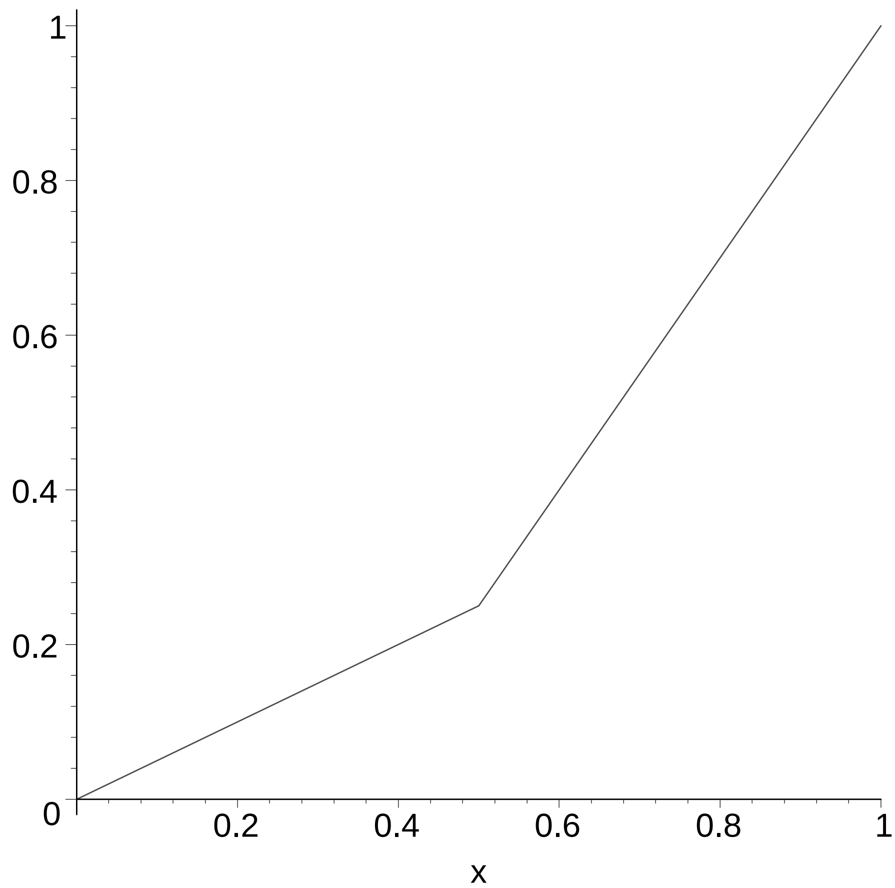



```
> infolevel[plot] := 2:
> plot(f(x), x=0..1, y=0..sqrt(2)):

plot/adaptive:   evalhf succeeded
plot/adaptive:   produced    59.    output segments
```

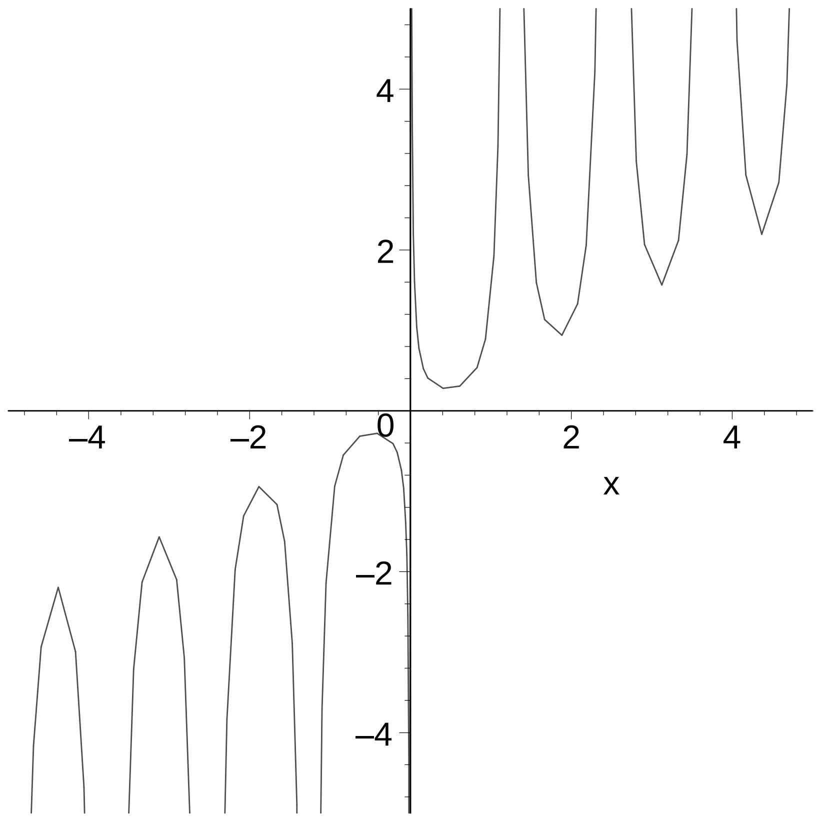
```
plot/adaptive:   using    59.    function evaluations
> plot(f(x), x=0..1, y=0..sqrt(2), adaptive=false):

plot/adaptive:   evalhf succeeded
plot/adaptive:   produced    49.    output segments
plot/adaptive:   using    49.    function evaluations
> infolevel[plot]:=1:
> plot(x^2, x=0..1, sample=[0,1/2,1], adaptive=true);
```



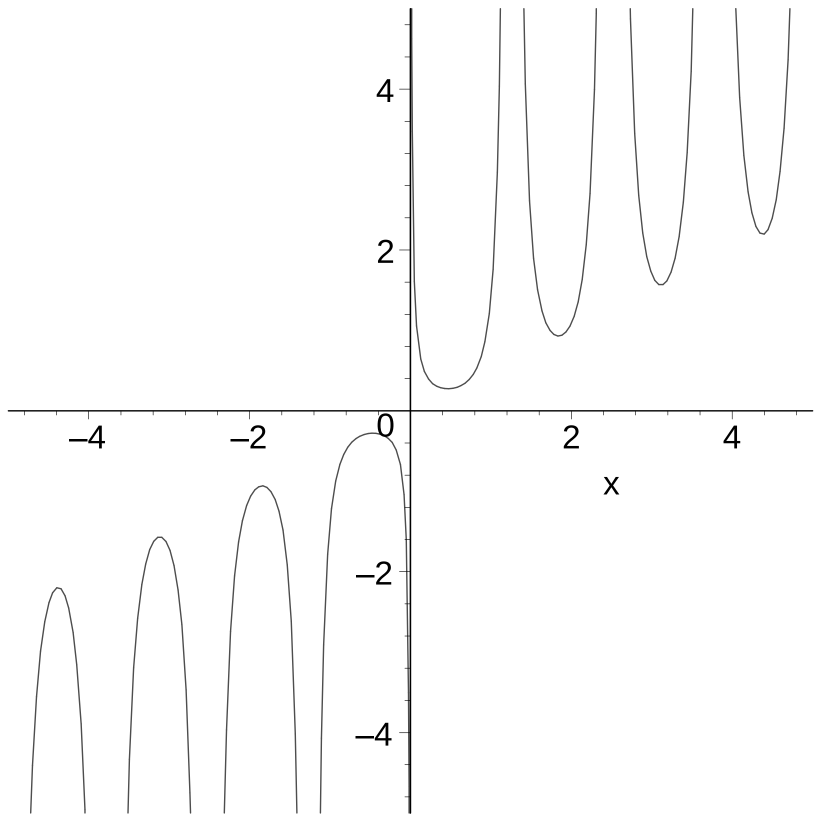

```
> plot(x^2, x=0..1, sample=[0,1/2,1], adaptive=false);
```



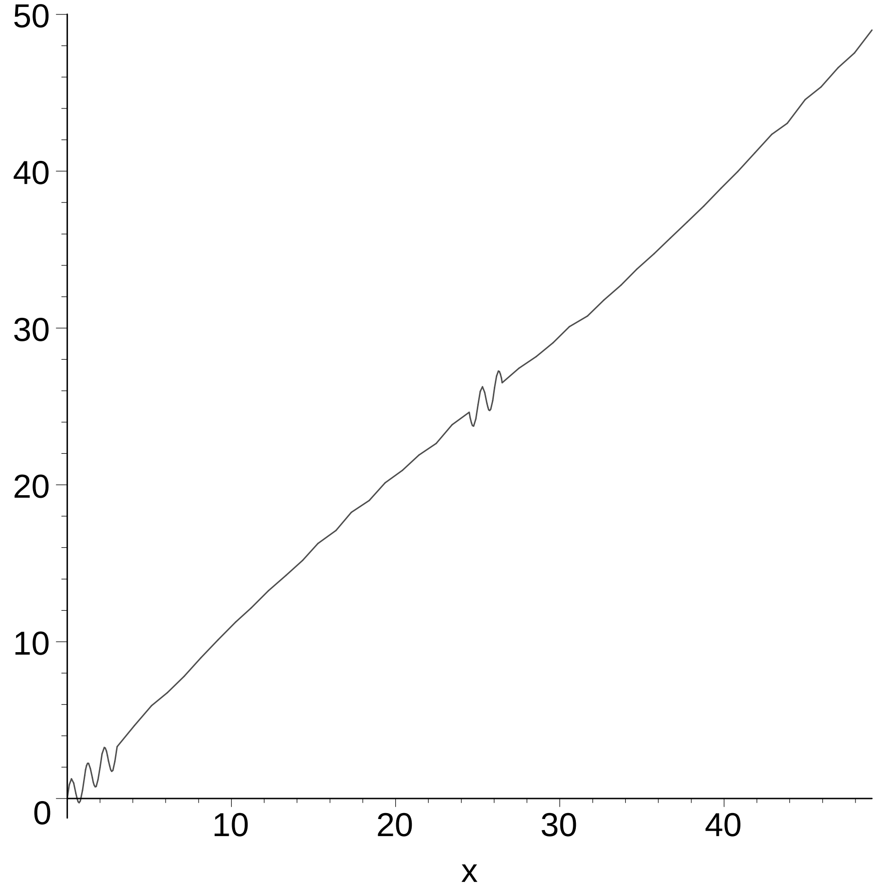

```
> plot(x/(1-cos(5*x)), x=-5..5, -5..5);
```



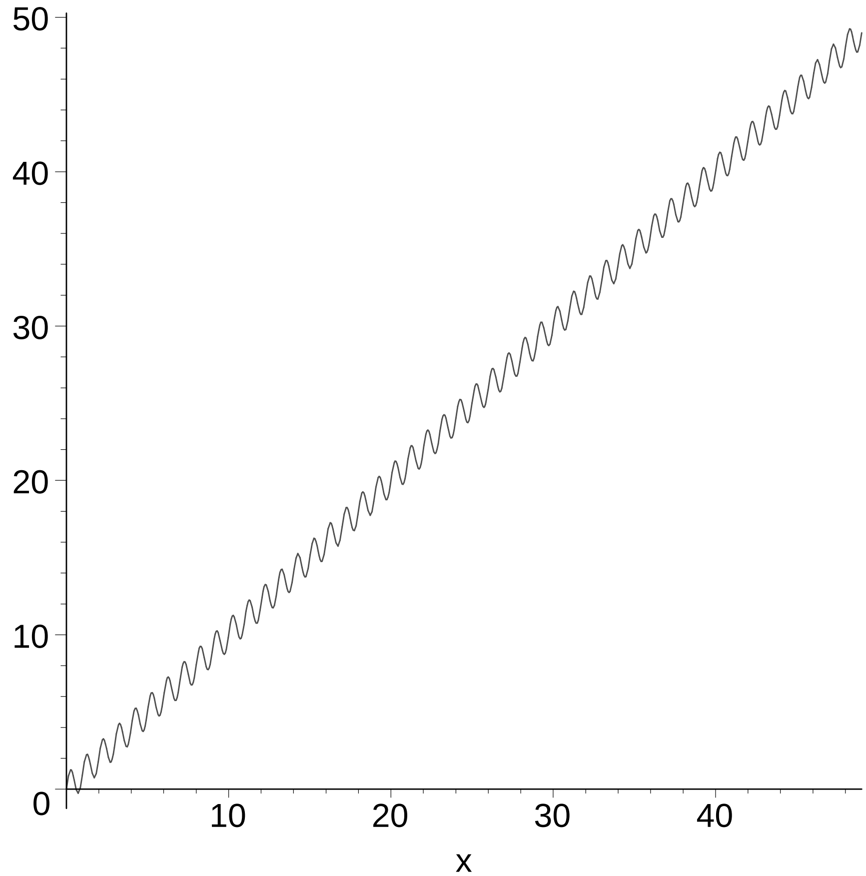

```
> plot(x/(1-cos(5*x)), x=-5..5, -5..5, numpoints=200);
```




```
> plot(x+sin(2*Pi*x), x=0..49);
```




```
> plot(x+sin(2*Pi*x), x=0..49, numpoints=200);
```



Pokud chceme nejakou option zmenit pro celou session, musime jeji hodnotu nastavit pomoci procedury setoptions.

```
> setoptions(scaling=constrained);
> setoptions(numpoints);
```

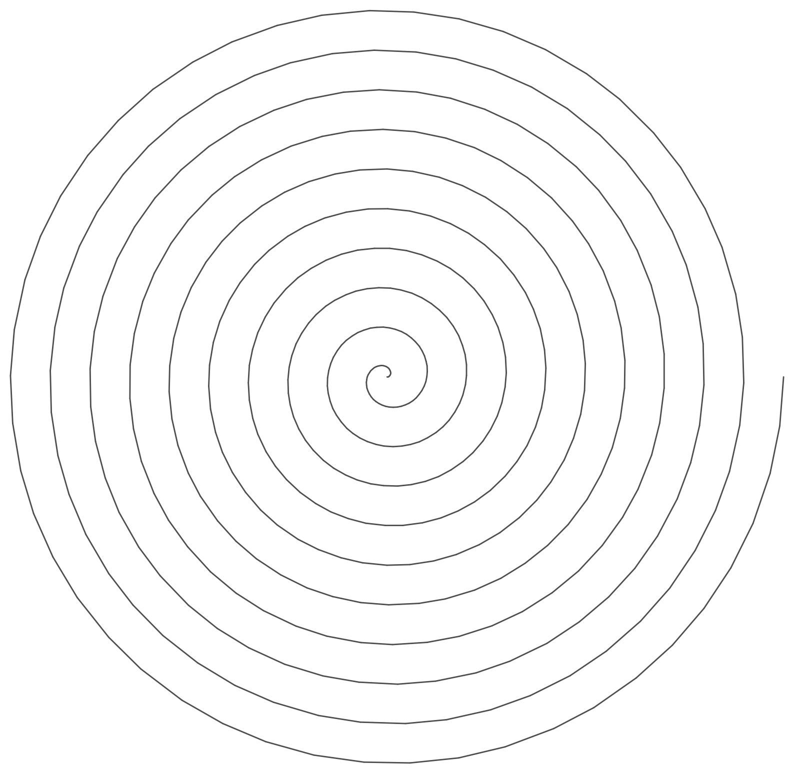
49

```
> ?plot[structure]
```

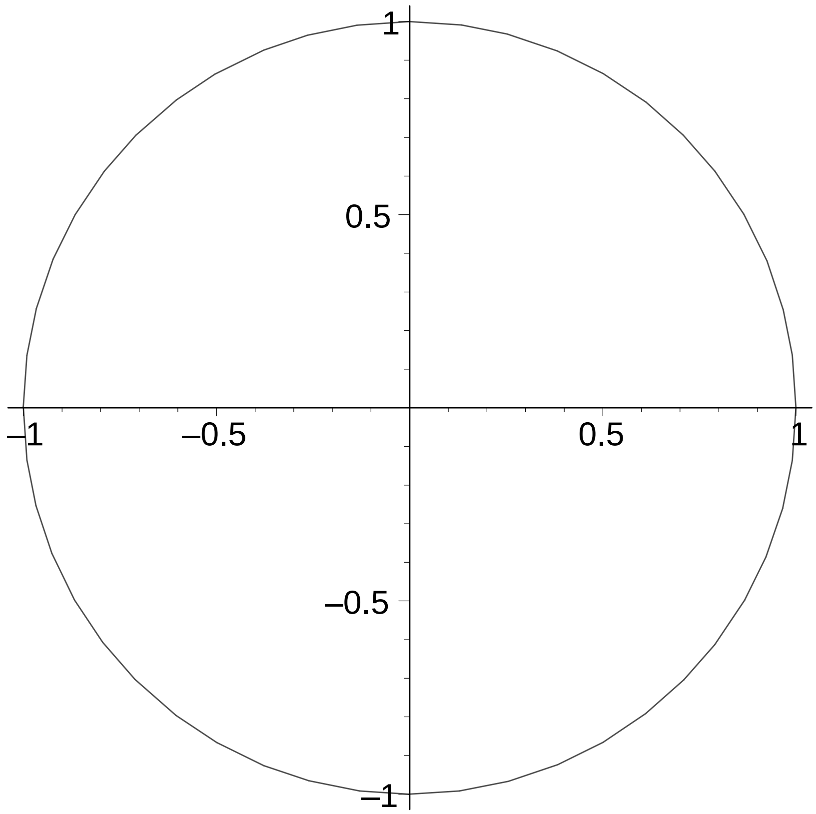
Specialni dvoj-dimenzionalni obrazky

Funkce dana parametricky:
syntaxe: plot([f(t), g(t), t=a..b], options)

```
> plot([t*cos(2*Pi*t), t*sin(2*Pi*t), t=0..10],
  numpoints=500, scaling=constrained, axes=none);
```



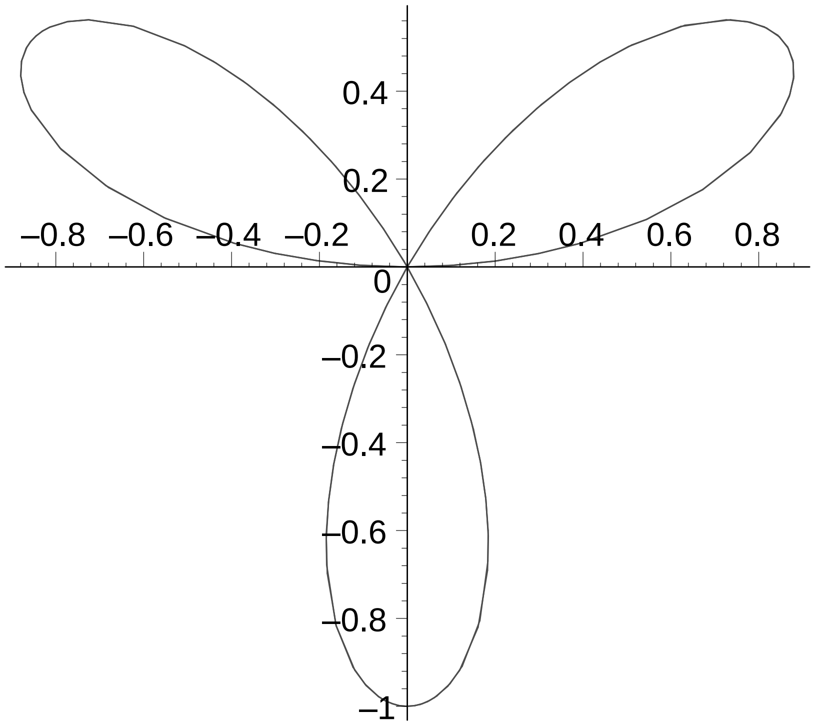
```
> plot([cos, sin, 0..2*Pi], scaling=constrained);
```



Krivka v polarnich souradnicich:

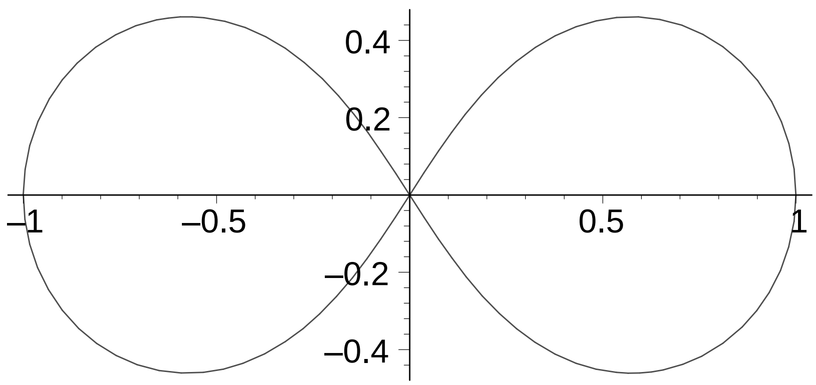
polarplot(r-expr,angle=range)

```
> plots[polarplot](sin(3*theta), theta=0..2*Pi);
```

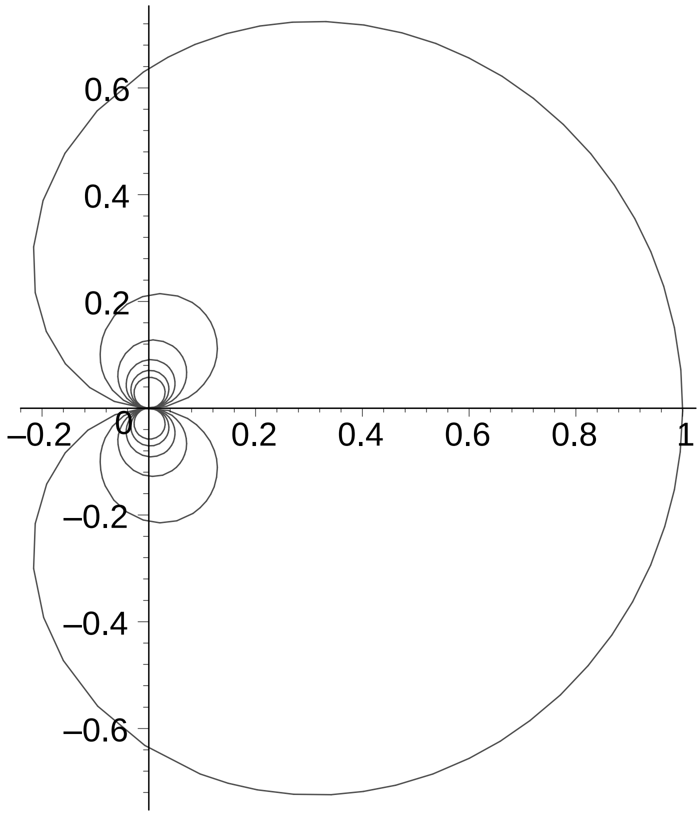


polarplot([r-expr, angle-expr, parameter=range)

```
> plots[polarplot]([sin(t), cos(t), t=0..2*Pi]);
```

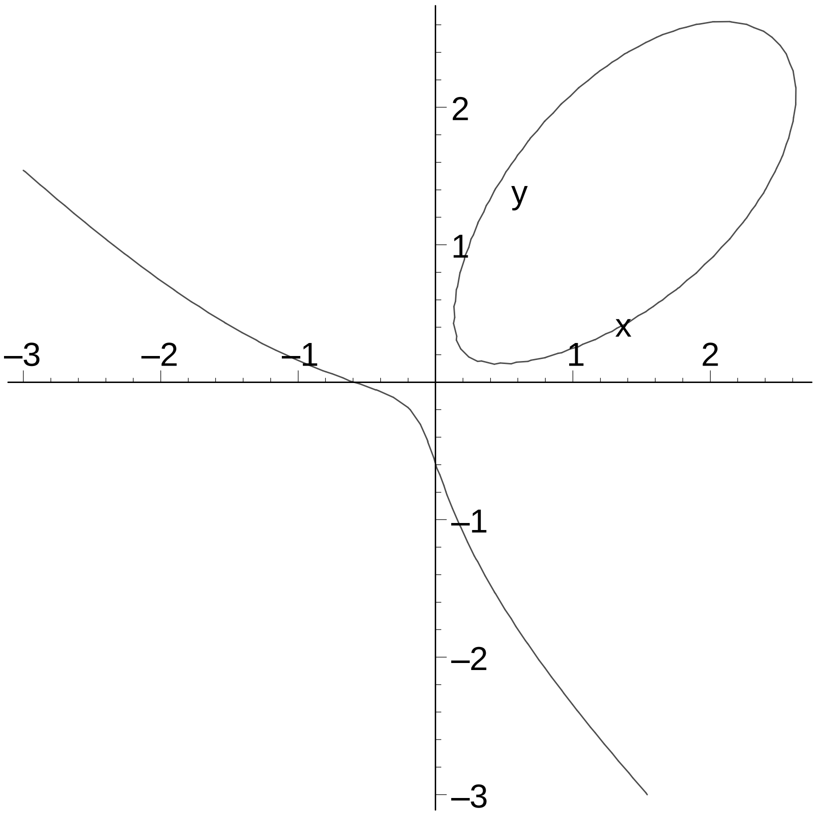



```
> plot([sin(t)/t, t, t=-6*Pi..6*Pi], coords=polar,
  numpoints=250);
```



Krivka dana implicitne rovnici

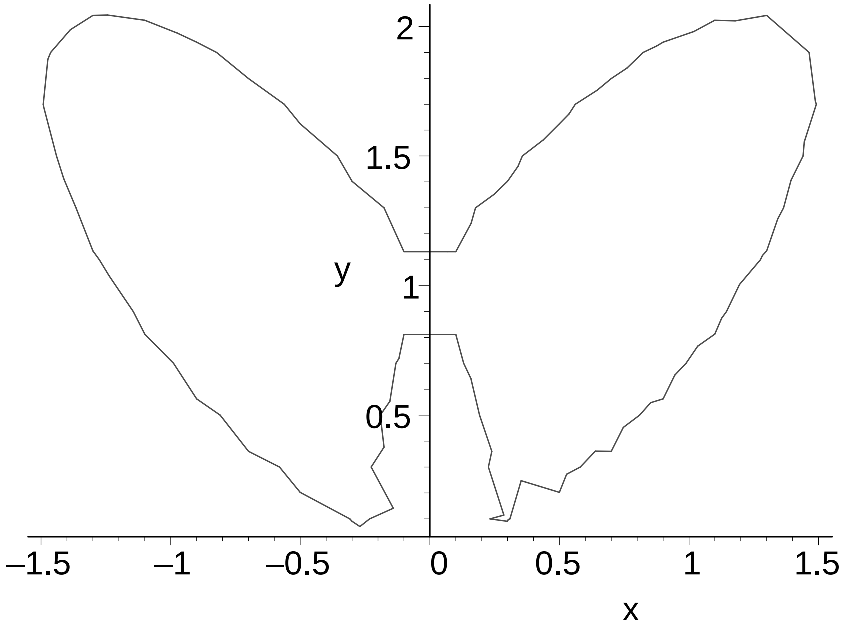
```
> implicitplot(x^3+y^3-5*x*y+1/5=0, x=-3..3, y=-3..3,
  grid=[50,50]);
```



Maple pracuje se zadanou rovnici jako s funkci dvou promennych a generuje vrstevnici teto funkce na hladine z=0 (rez rovinou z=0).

Nevyhodou teto metody je, ze ziskane obrazky mohou byt "kostrbate" v okoli singularit a mistech, kde krivka protina sebe samu.

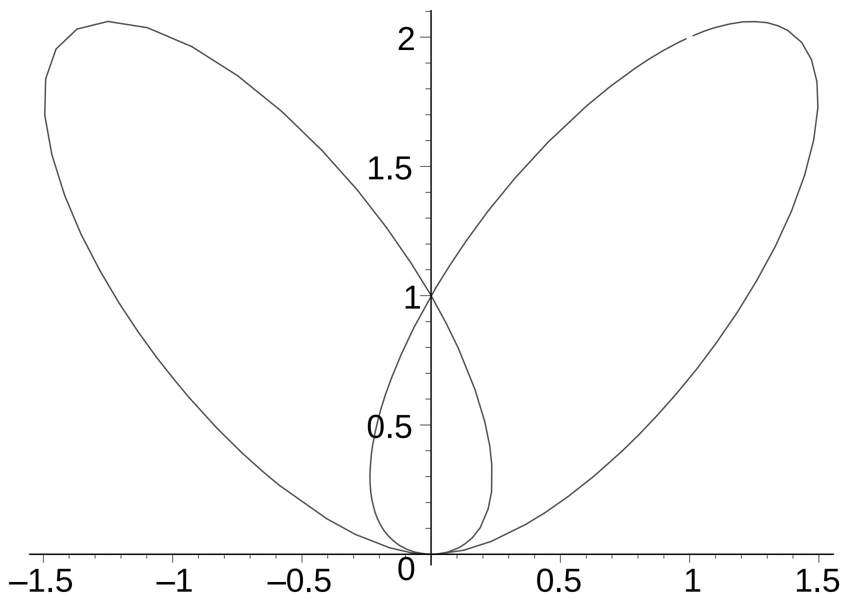
```
> implicitplot(2*x^4+y^4-3*x^2*y-2*y^3+y^2, x=-5/2..5/2,
  y=-5/2..5/2,scaling=constrained);
```




```
> v:=algcurves[parametrization](2*x^4+y^4-3*x^2*y-2*y^3+
  y^2,x,y,t);
```

$$v := \left[\frac{-8470 - 6883\,t - 1548\,t^2 - 15\,t^3 + 18\,t^4}{18\,t^4 + 336\,t^3 + 5056\,t^2 + 26696\,t + 43257}, \frac{4900 + 6020\,t + 2689\,t^2 + 516\,t^3 + 36\,t^4}{18\,t^4 + 336\,t^3 + 5056\,t^2 + 26696\,t + 43257} \right]$$

```
> plot([op(v), t=-infinity..infinity],
  scaling=constrained);
```




```
> restart;
```

Prokladani krivek (linearni regrese, metoda nejmensich ctvercu) - pouzijeme balik statistic:

```
> datax:=seq(i, i=1..9);
```

$$datax := 1, 2, 3, 4, 5, 6, 7, 8, 9$$

```
> datay:=seq(ithprime(i), i=1..9);
```

$$datay := 2, 3, 5, 7, 11, 13, 17, 19, 23$$

```
> with(stats); Digits:=5;
```

$$[anova, describe, fit, importdata, random, statevalf, statplots, transform]$$

$$Digits := 5$$

```
> aproximace:=fit[leastsquare[[x,y], y=a*x+b,
  {a,b}]]([[datax],[datay]]);
```

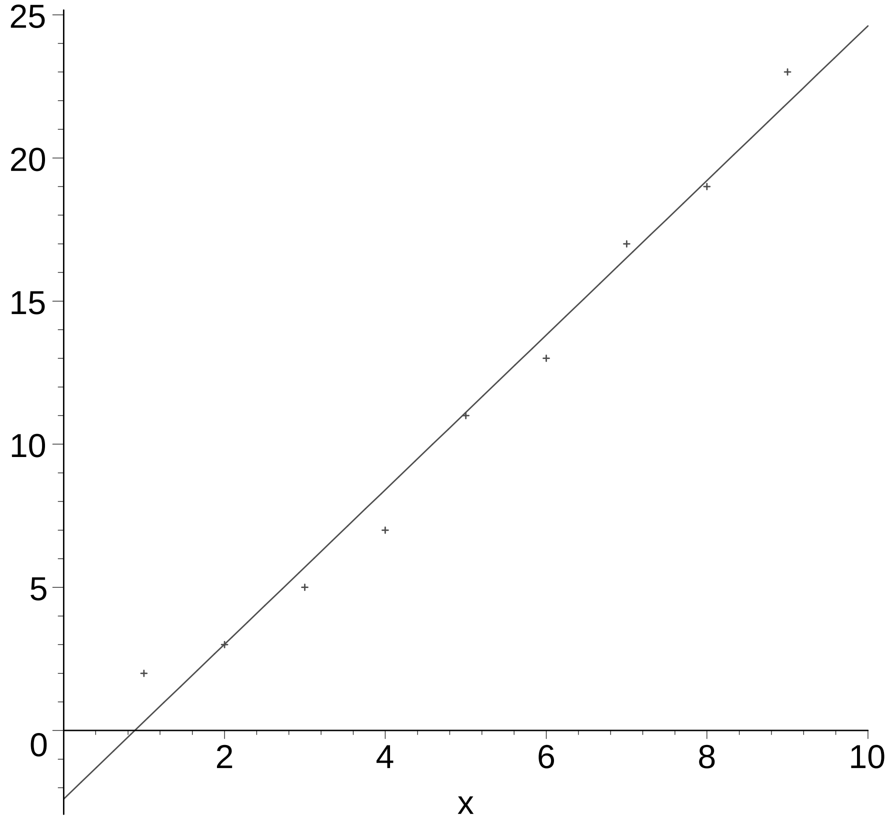
$$aproximace := y = \frac{27\,x}{10} - \frac{43}{18}$$

```
> pair:=(x,y)->[x,y];
  plotpoints:=zip(pair, [datax],[datay]);
```

$$pair := (x, y) \rightarrow [x, y]$$

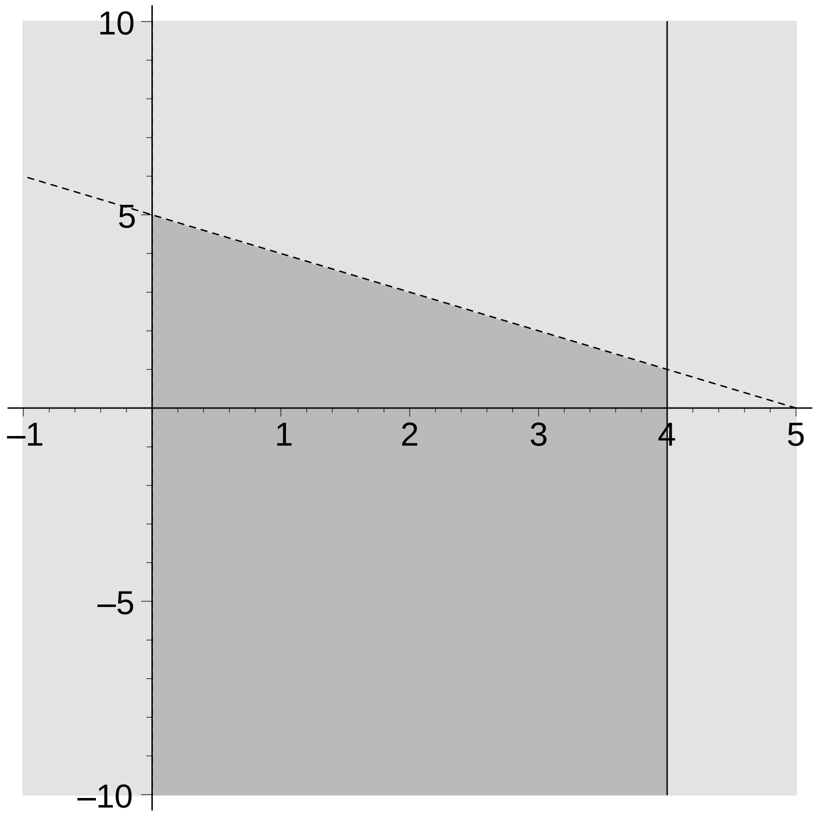
$$plotpoints := [[1, 2], [2, 3], [3, 5], [4, 7], [5, 11], [6, 13], [7, 17], [8, 19], [9, 23]]$$

```
> plot1:=plot(plotpoints, style=point):
> plot2:=plot(rhs(aproximace), x=0..10):
> plots[display]({plot1,plot2});
```



Nasledujici obrazek ukazuje oblast, vyhovujici nerovnicim x+y<5, 0<x, x<=4.

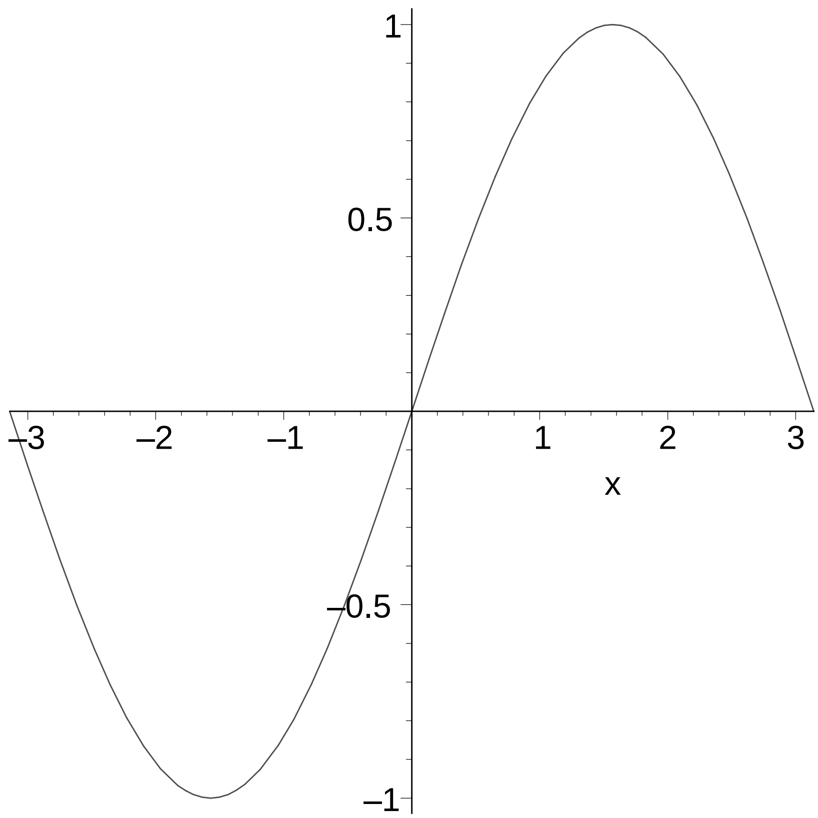
```
> plots[inequal]({x+y<5, 0<x, x<=4}, x=-1..5, y=-10..10,
  optionsexcluded=(color=yellow));
```



Ukladani grafiky

Ulozeni obrazku do postscriptu:

```
> plot(sin(x), x=-Pi..Pi);
```




```
> G:=%:
```

Nejprve prikazem

```
> plotsetup(cps, plotoutput="sin.eps", plotoptions=
  "portrait,noborder,leftmargin=0,bottommargin=0");
```

Maplu sdelime, ze ma graficky vystup ukladat do postscriptoveho souboru sin.eps. Pote vygenerujeme obrazek, jehoz vystup se neobjevi na obrazovce, ale ulozi do souboru sin.eps v aktulanim adresari.

```
> G;
```

Zamezit kresleni ramecku kolem obrazku muzeme pouzitim doplnujiciho parametru plotoptions=noborder. Vystup grafiky zpet do zapisniku vratime prikazem:

```
> plotsetup(default);
> ?plot[device]
>
```

>