

wxdraw2d examples.wxm

Examples from the worksheet wxdraw2d.wxm in the webpage section **Economic Analysis with Maxima**.

Ted Woollett

<http://web.csulb.edu/~woollett/>

Feb., 2021

MAIN PLOTTING FUNCTIONS

explicit (expr, var, varmin, varmax) plots an expression (which may be a Maxima function).

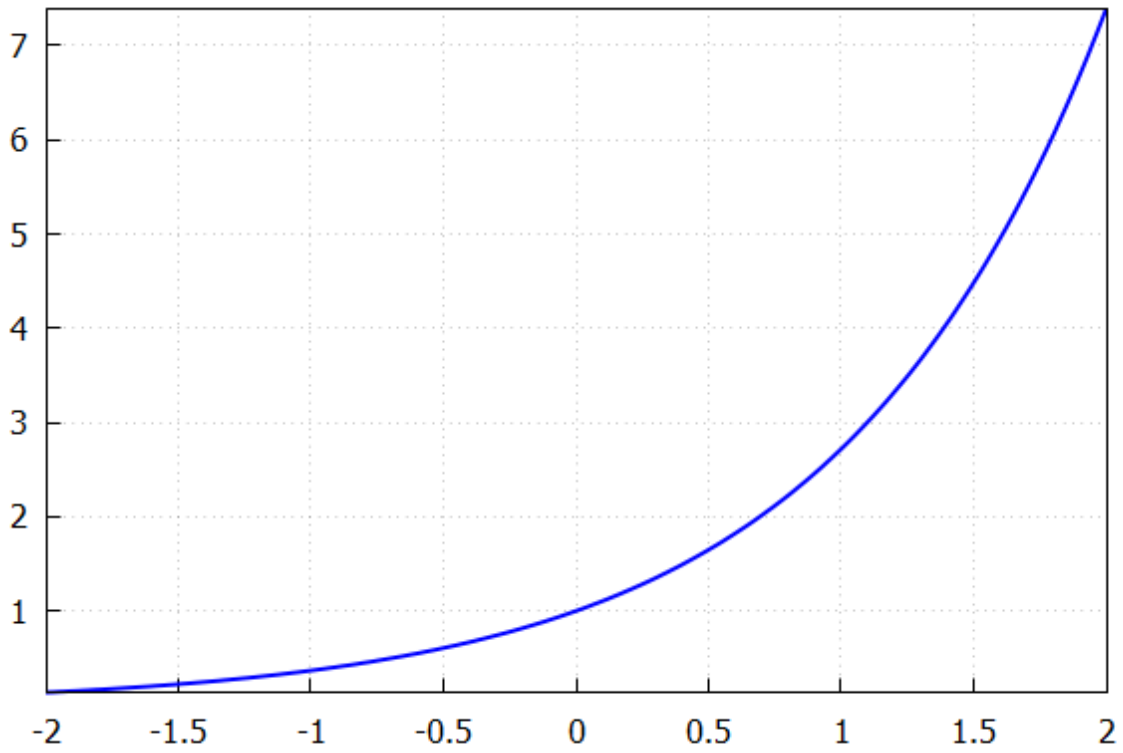
implicit plots all points for which an equation is true.

points ([x1, x2, x3], [y1, y2, y3]) plots the points (x1, y1), (x2, y2), and (x3, y3) (connected by lines if the option points_joined was set to true in a previous line).

parametric (fx, fy, t, t1, t2) allows to specify separate expressions that calculate the x coordinate and the y coordinate when drawing a curve that depends on some parameter t. In general, both fx and fy depend on the parameter t, but they can each also just be some constant.

1 plot one expression

```
(%i1) wxdraw2d ( grid = true, line_width = 2,  
                explicit ( exp ( u ), u, -2, 2))$
```



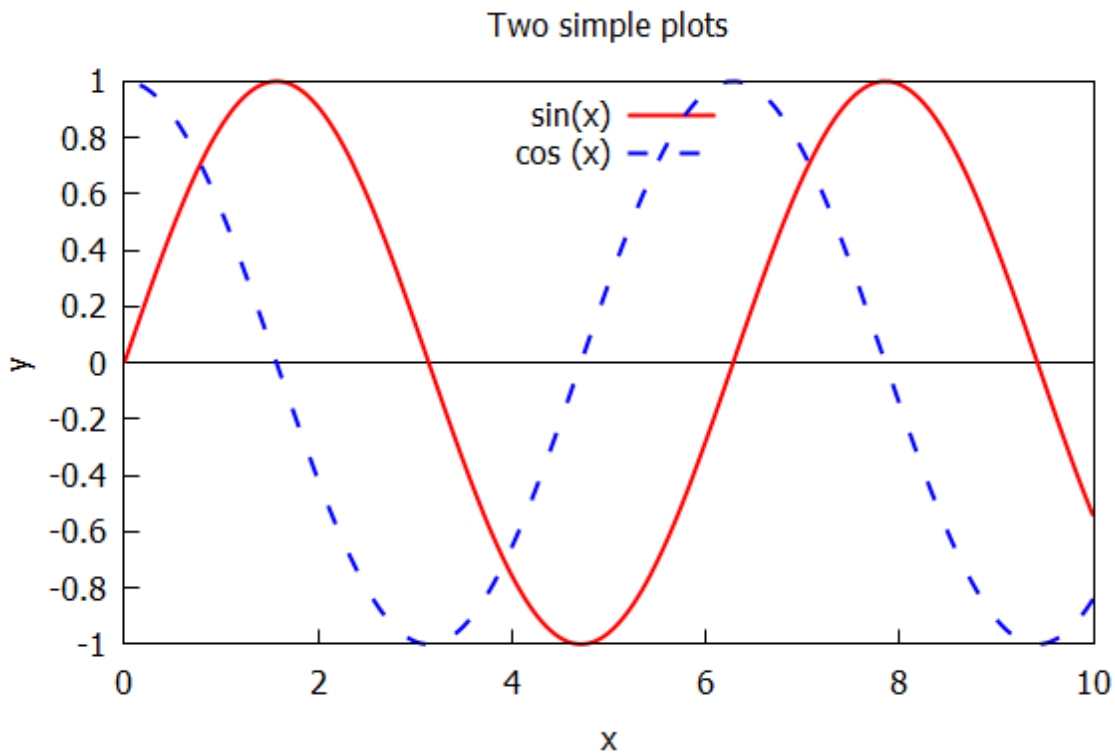
Add background color:

```
(%i2) wxdraw2d ( grid = [2, 2],  
                background_color = light_gray,  
                color = black,  
                explicit ( 0, x, 0, 2*%pi),  
                line_width = 2,  
                color = dark_green,  
                explicit ( sin ( x ), x, 0, 2*%pi) )$
```



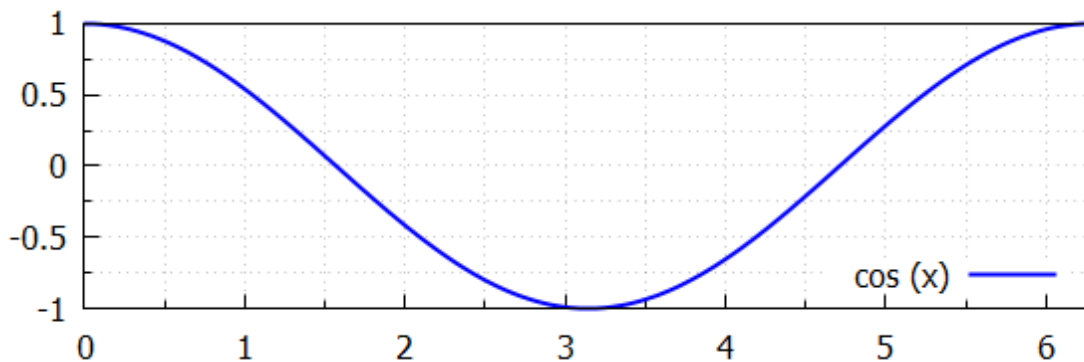
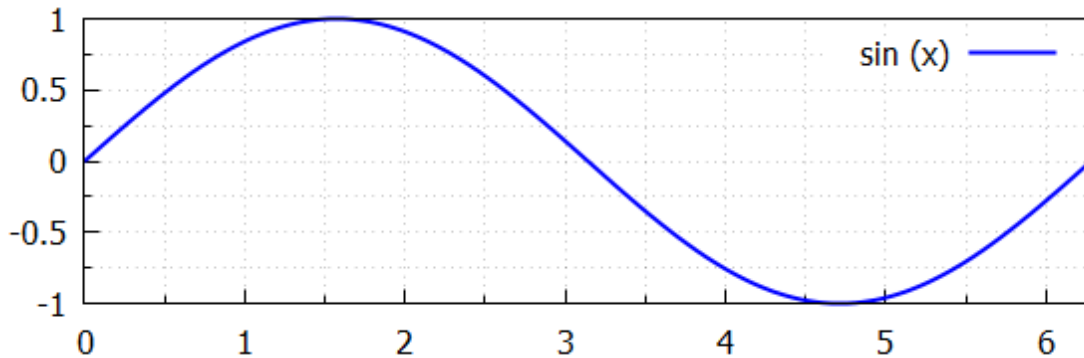
2 plot sin and cos curves together

```
(%i3) wxdraw2d(  
    title = "Two simple plots",  
    xlabel = "x", ylabel = "y",  
    key_pos = top_center,  
    color = black,  
    explicit ( 0, x, 0, 10),  
    color = red, key = "sin(x)",  
    line_width = 2,  
    explicit ( sin ( x ), x, 0, 10),  
    color = blue, line_type = dashes, key = "cos (x)",  
    explicit ( cos ( x ), x, 0, 10) )$
```



3 sin and cos plots as two rows

```
(%i4) wxdraw (  
    gr2d ( key="sin (x)", grid=[2,2],  
          line_width = 2,  
          explicit ( sin(x), x,0,2*%pi ) ),  
    gr2d ( key="cos (x)", grid=[2,2],  
          key_pos = bottom_right,  
          line_width = 2,  
          explicit( cos(x), x,0,2*%pi ) ) )$
```



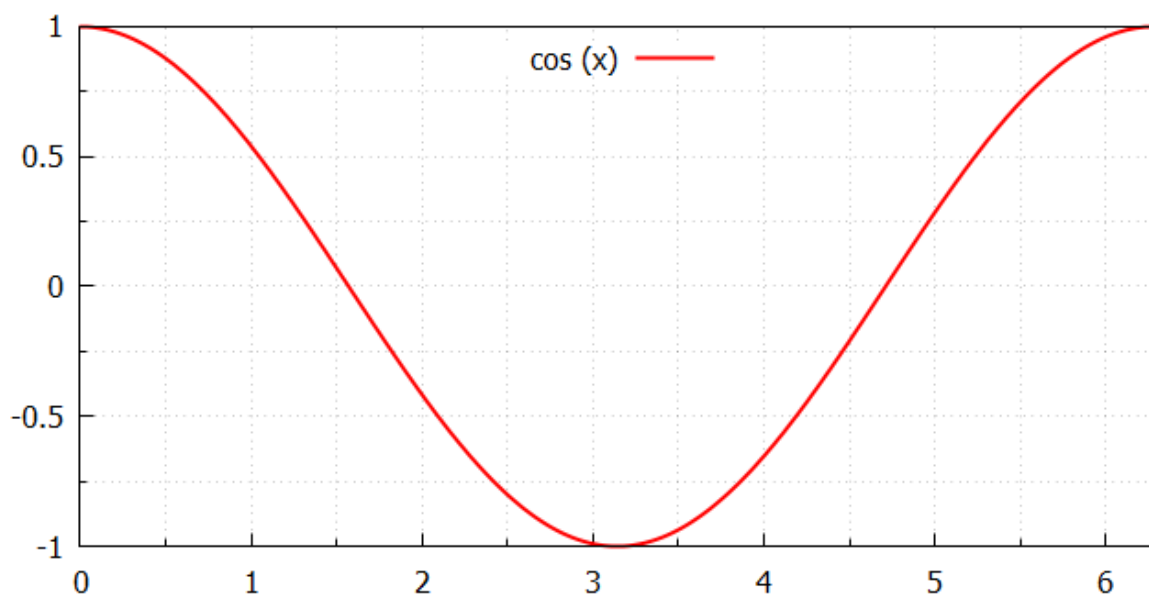
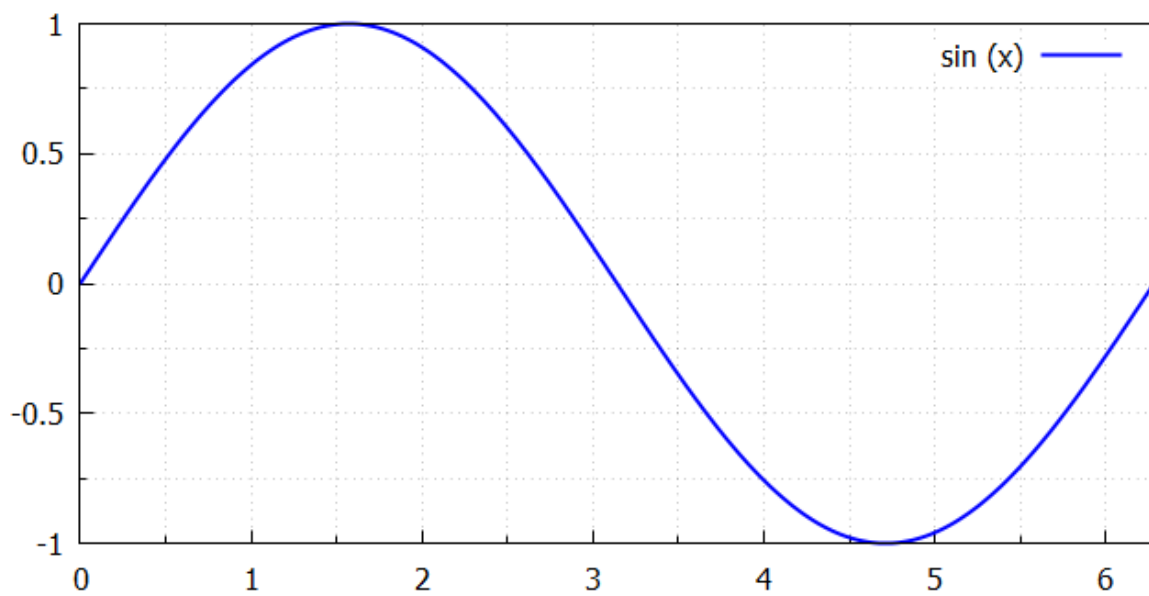
Example of using `grid = [nx,ny]` option with `draw` pkg.
This shows two (2) grid intervals between each numbered tick mark on both axes. Also example of using the external modifying option `wxplot_size` to get larger than the default size.

(%i7)

```
sinplot : gr2d(line_width = 2, color = blue,  
key="sin (x)",grid=[2,2], explicit( sin(x), x,0,2*%pi )$
```

```
cosplot : gr2d( line_width = 2, color = red,  
key_pos = top_center, key="cos(x)",grid=[2,2],  
explicit(cos(x), x,0,2*%pi ) )$
```

```
wxdraw ( sinplot, cosplot ), wxplot_size = [680,680]$
```

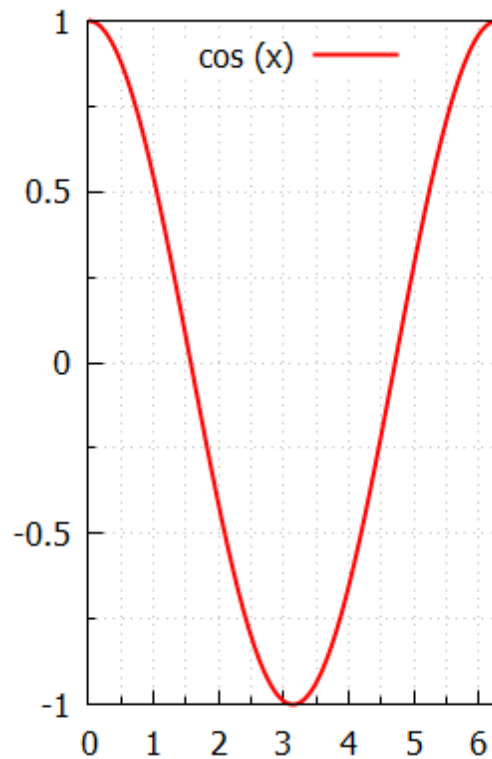
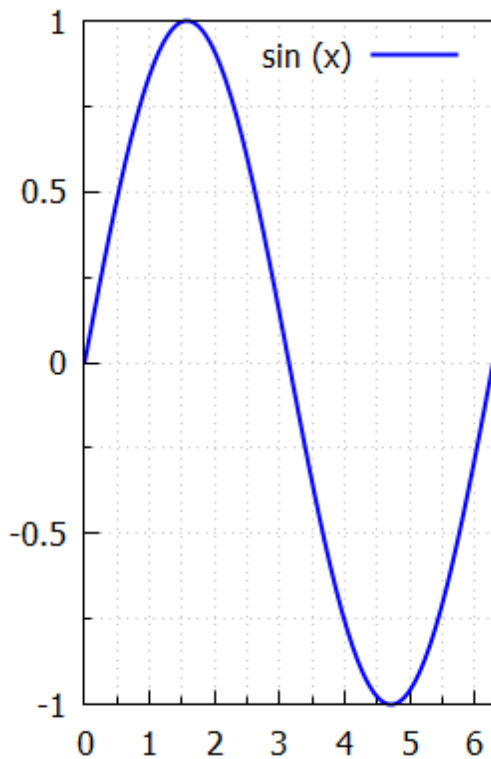


4 sin and cos plots as two columns

default size, next to each other

(%i8)

```
wxdraw ( sinplot, cosplot, columns = 2);
```

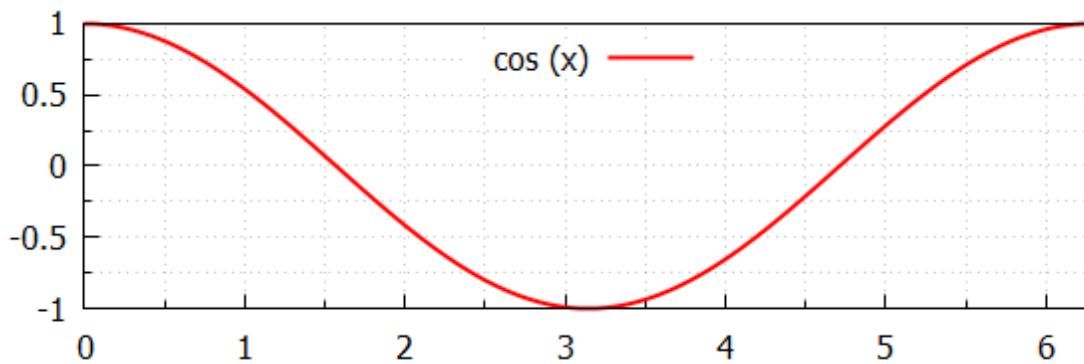
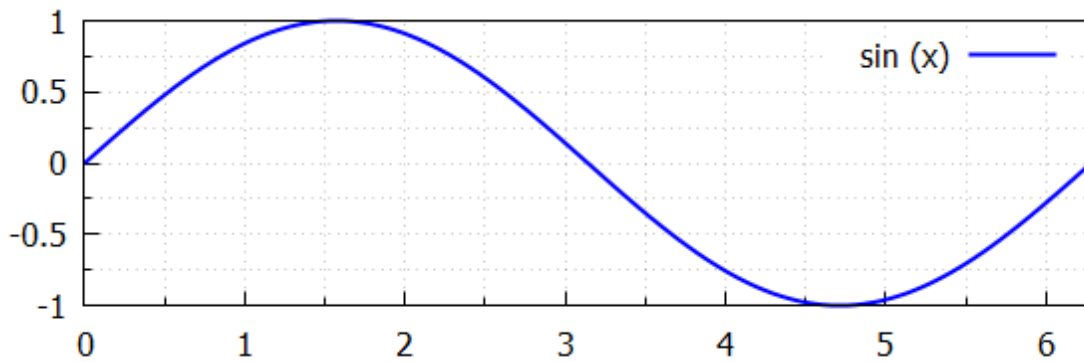


(%o8)(%o8)

default size, on top of each other (two rows)

(%i9)

wxdraw (sinplot, cosplot);

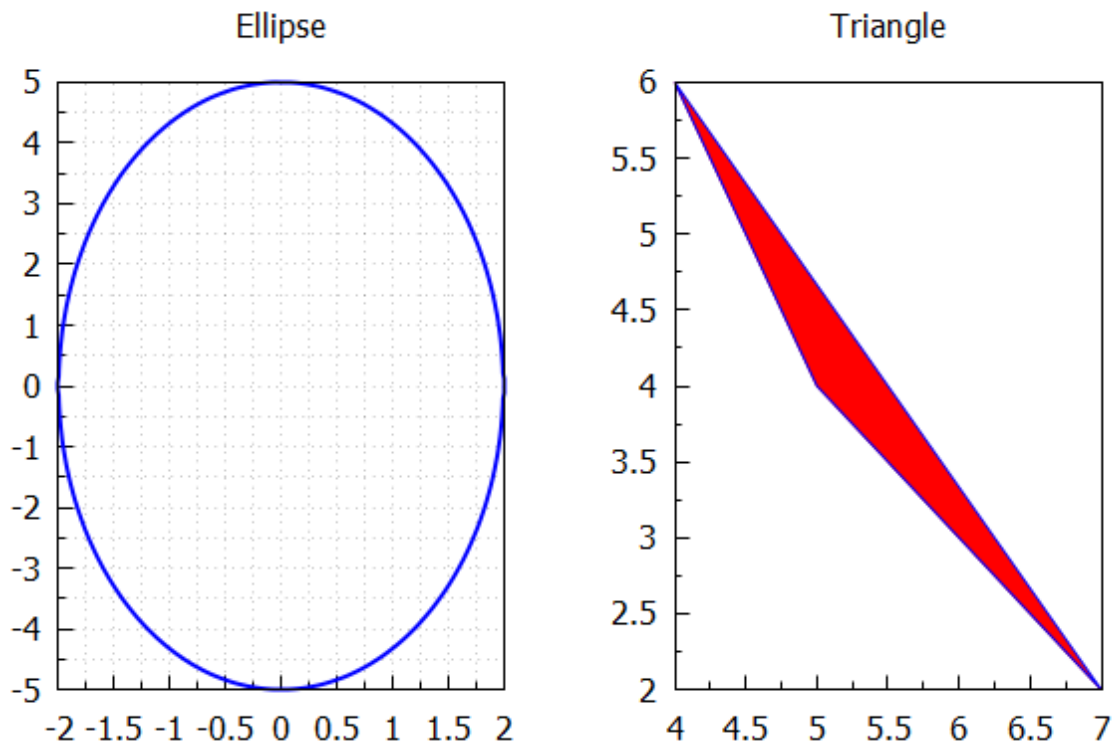


5 ellipse and triangle side by side

```
(%i10) scene1: gr2d(title="Ellipse",  
    nticks=300, line_width = 2,  
    grid = [2,2],  
    parametric(2*cos(t),5*sin(t),t,0,2*%pi))$
```

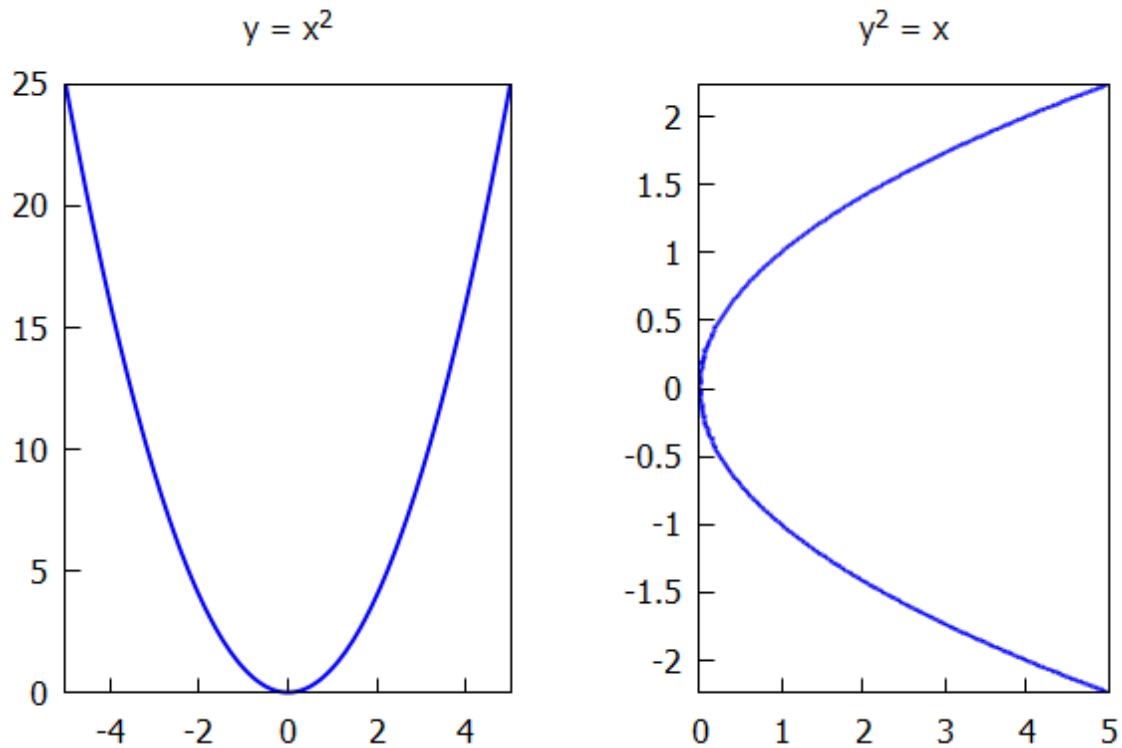
```
(%i11) scene2: gr2d(title="Triangle",  
    polygon([4,5,7],[6,4,2]))$
```

```
(%i12) wxdraw(scene1, scene2, columns = 2)$
```



6 explicit and implicit curves side by side

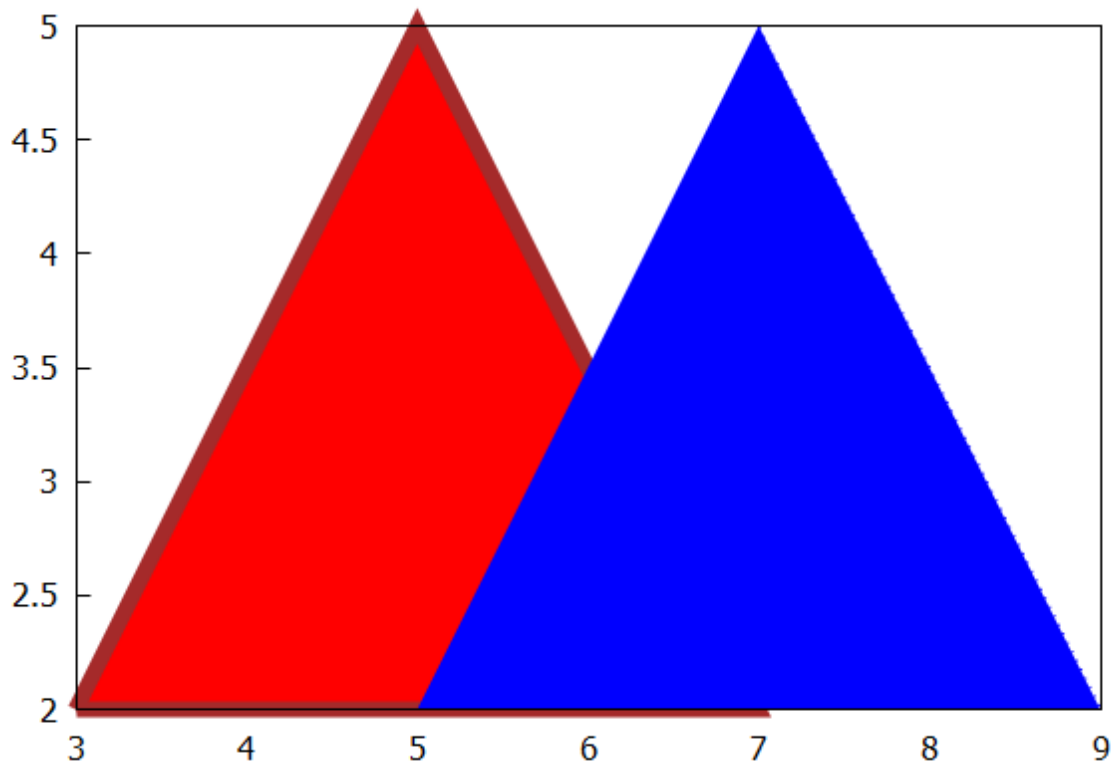
```
(%i15) first : gr2d (title = "y =  
x^2", line_width = 2, explicit (x^2, x, -5,5) )$  
second : gr2d (title = "y^2 = x", line_width = 2,  
implicit ( y^2 = x, x, 0, 5, y, -sqrt (5), sqrt (5) ) )$  
wxdraw ( first, second, columns = 2 )$
```



7 two filled triangles in one plot

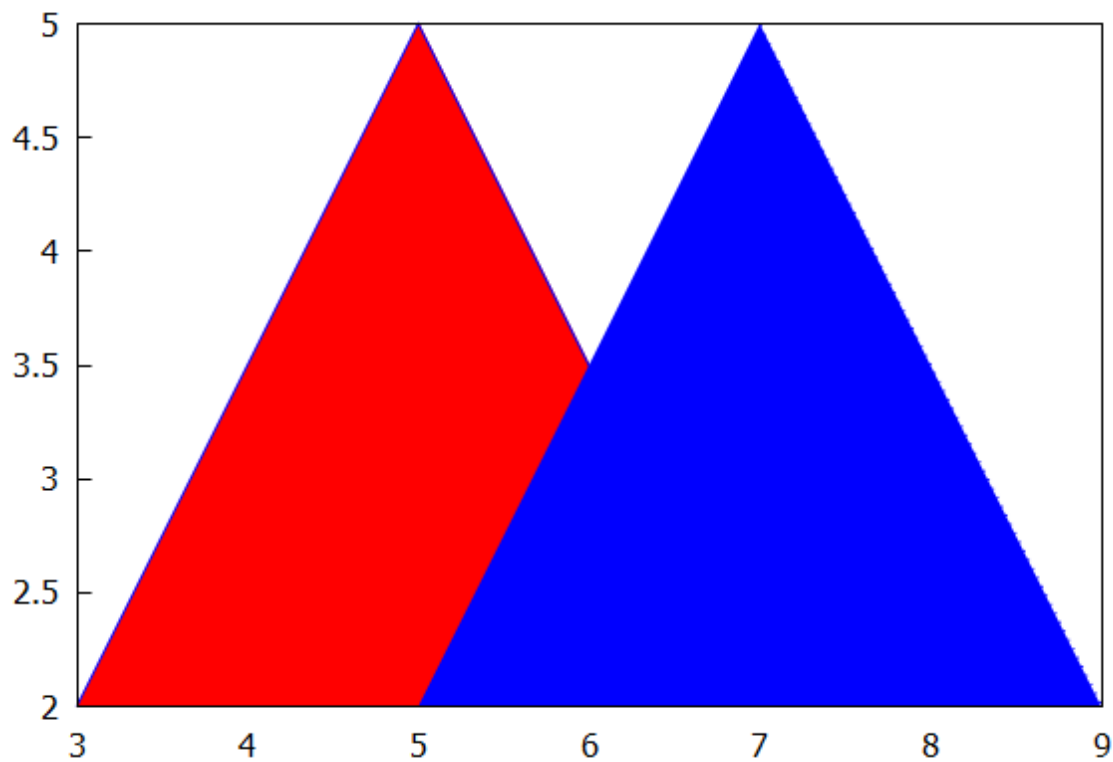
red triangle has border, blue doesn't

```
(%i16) wxdraw2d(color = brown,  
                line_width = 8,  
                polygon([[3,2],[7,2],[5,5]]),  
                border = false,  
                fill_color = blue,  
                polygon([[5,2],[9,2],[7,5]]) )$
```



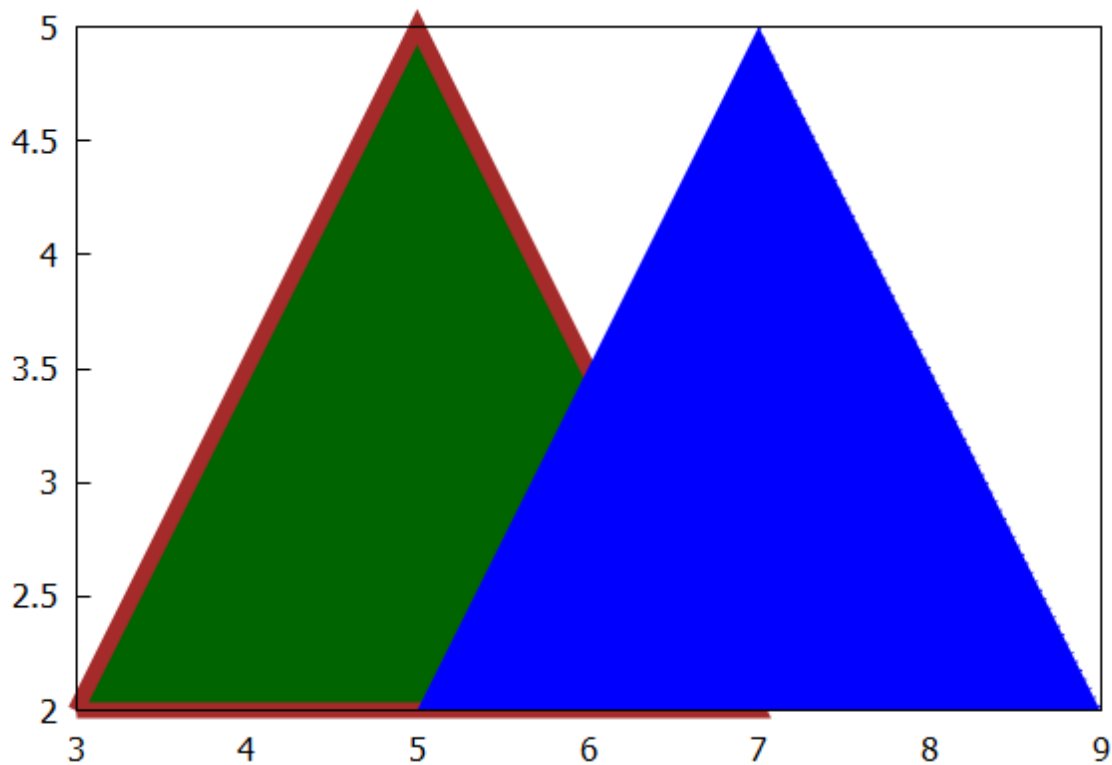
Thin default border

```
(%i17) wxdraw2d(  
    polygon([[3,2],[7,2],[5,5]]),  
    border = false,  
    fill_color = blue,  
    polygon([[5,2],[9,2],[7,5]]) )$
```



Thicker border

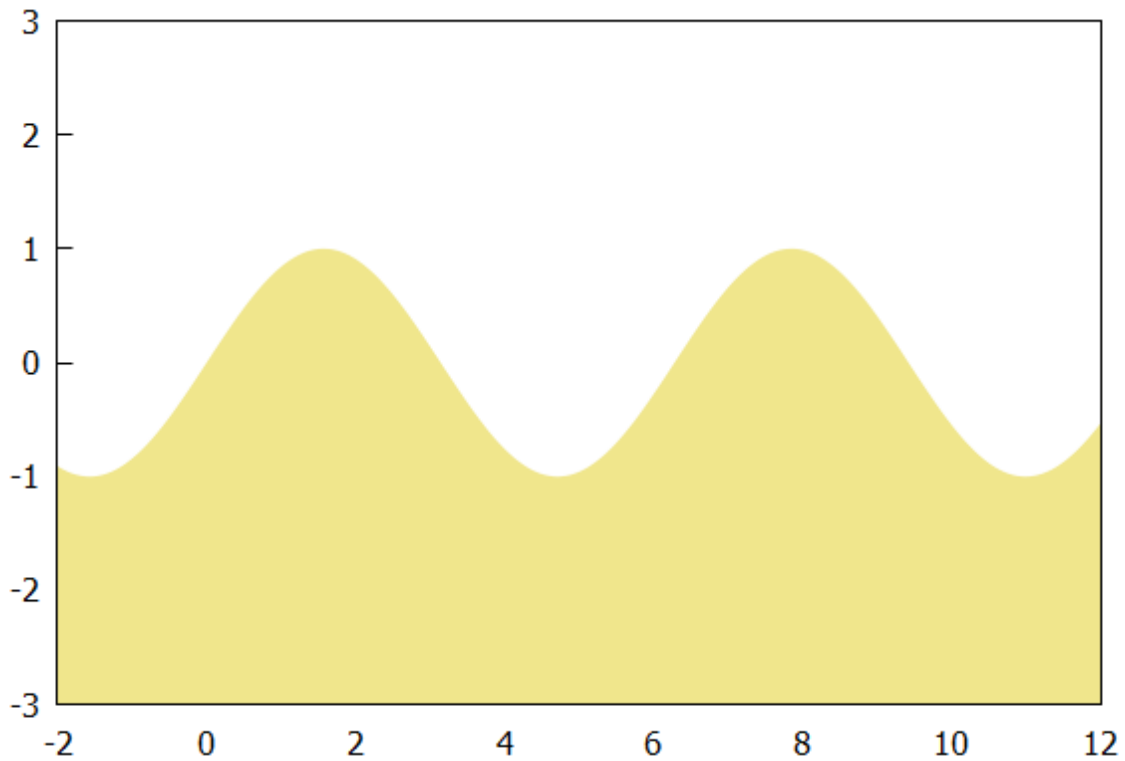
```
(%i18) wxdraw2d(color = brown,  
                line_width = 8,  
                fill_color = dark_green,  
                polygon([[3,2],[7,2],[5,5]]),  
                border = false,  
                fill_color = blue,  
                polygon([[5,2],[9,2],[7,5]]) )$
```



8 plot a function with a fill

Function is $\sin(x)$

```
(%i19) wxdraw2d ( xrange = [-2, 12], yrange = [-3, 3],  
                fill_color = khaki,  
                filled_func = true,  
                explicit ( sin ( x ), x, -2, 12) )$
```



9 plot random points in 2d

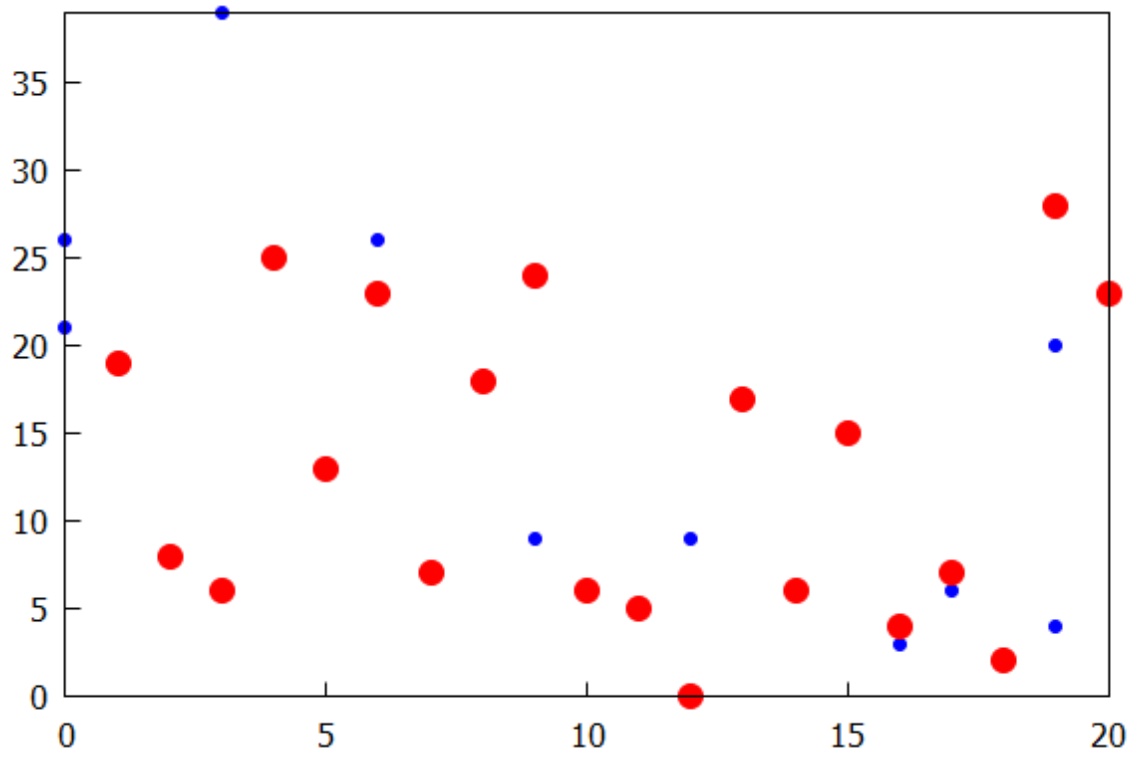
```
(%i22) random( 30);  
      random (30);  
      random (30);
```

```
(%o20) 2  
(%o21) 12  
(%o22) 14
```

```
(%i23) makelist ( [random (20), random (50) ], k, 1, 3);  
              (%o23) [[5,4],[11,29],[5,48]]
```

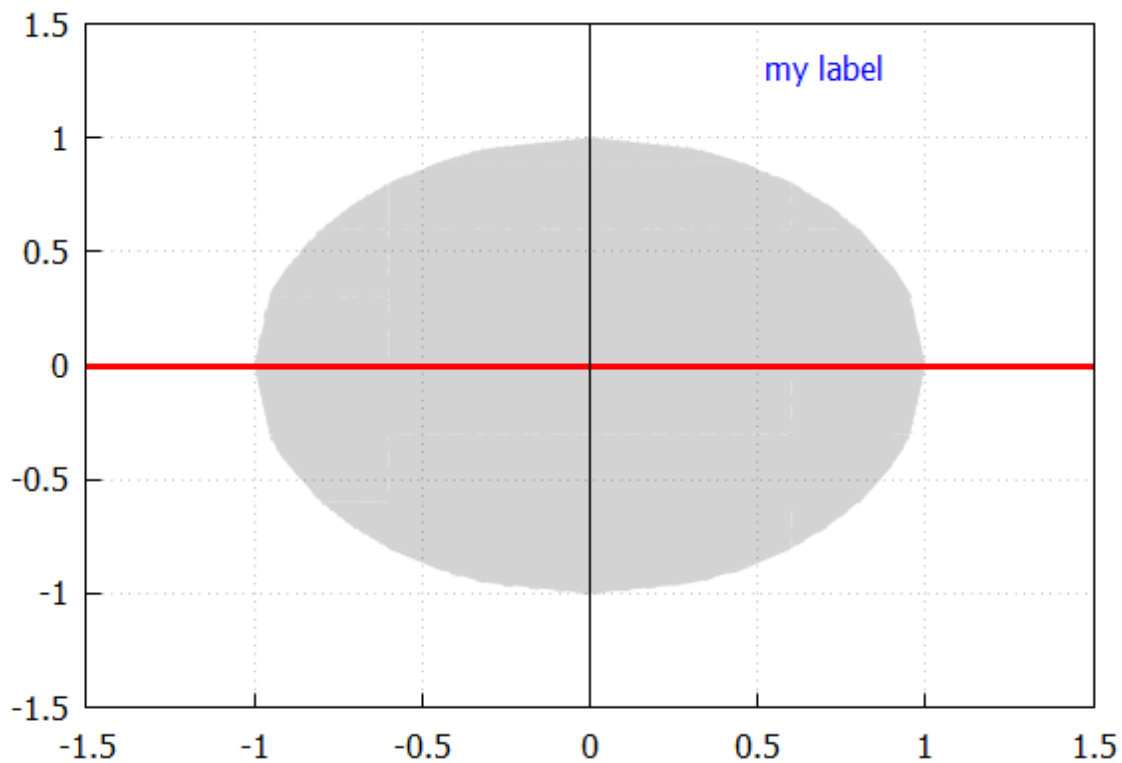
```
(%i25) makelist ( k, k, 1, 3);  
      makelist ( random (30), k, 1, 3);  
              (%o24) [1,2,3]  
              (%o25) [13,25,5]
```

```
(%i26 wxdraw2d ( point_type = filled_circle,  
)  
      points ( makelist ( [ random ( 20 ), random ( 50 )  
], k, 1, 10)),  
      point_size = 2,  
      color = red,  
      points ( makelist ( k, k, 1, 20 ), makelist ( random ( 30 ), k,  
1, 20 ) ) )$
```

10 filled ellipse with label, use of region

```
(%i27) wxdraw2d(  
    xaxis = true, xaxis_type = solid,  
    xaxis_color = red, xaxis_width = 3,  
    yaxis = true, yaxis_type = solid,  
    user_preamble = "set grid front",  
    fill_color = light_gray,  
    region (  $x^2 + y^2 < 1$  , x, -1.5, 1.5, y, -1.5, 1.5 ),  
    label ( [ "my label", 0.7, 1.3 ] ))$
```

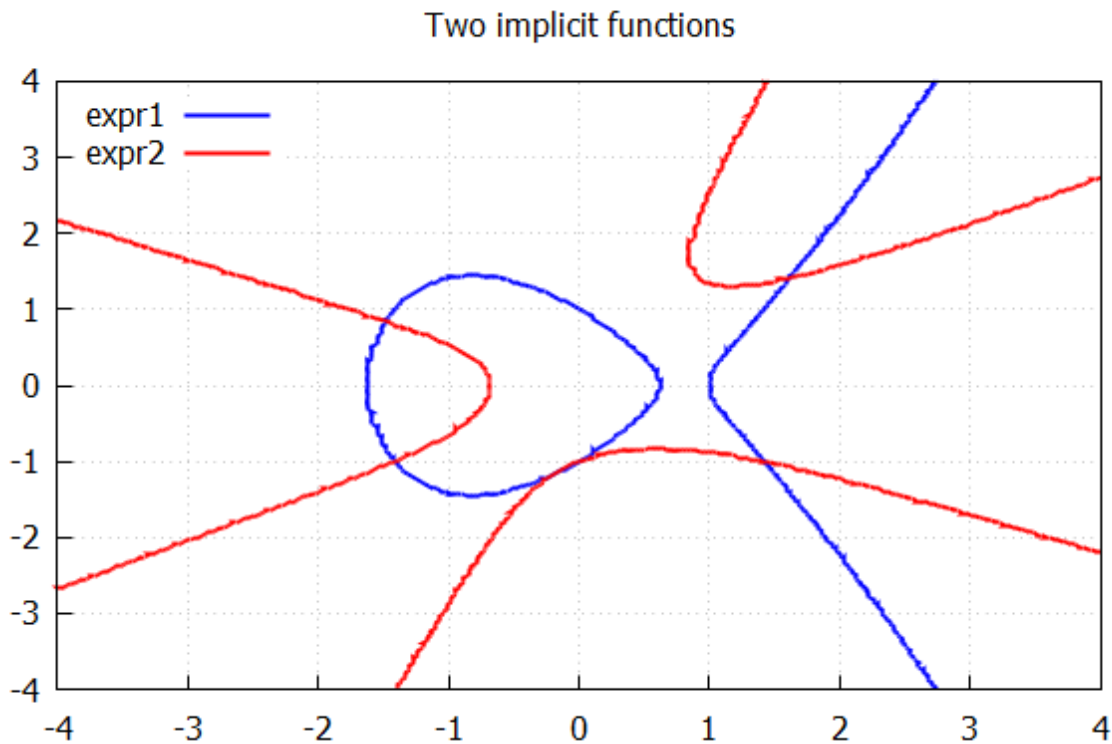


11 plot of two implicit functions together

```
(%i28) kill(all)$
```

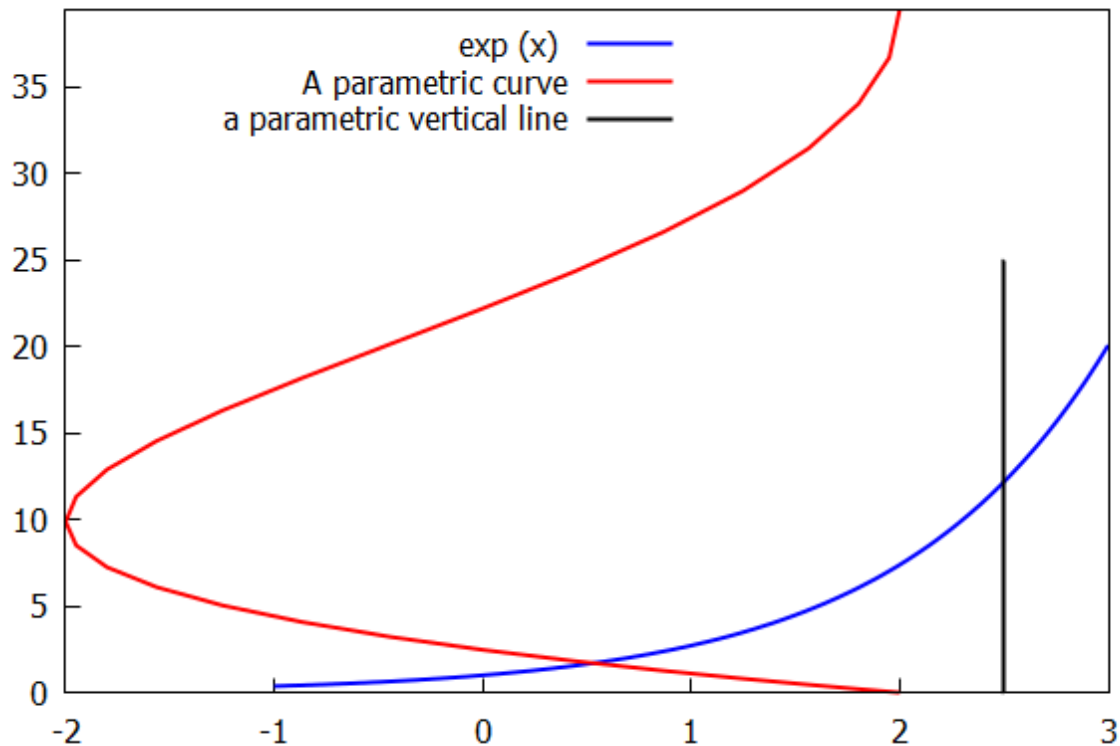
```
(%i2) expr1 : y^2 = x^3 - 2*x + 1$  
      expr2 : x^3 + y^3 = 3*x*y^2 - x - 1$
```

```
(%i3) wxdraw2d ( grid      = true,  
                line_type = solid, line_width = 2,  
                key_pos   = top_left,  
                key       = "expr1",  
                implicit ( expr1, x, -4,4, y, -4,4),  
                color     = red,  
                key       = "expr2",  
                implicit ( expr2, x, -4, 4, y, -4, 4 ),  
                title     = "Two implicit functions" )$
```



12 combine explicit and parametric curves

```
(%i4) wxdraw2d ( line_width = 2,  
                key_pos = top_left,  
                key = " exp ( x ) ",  
                explicit ( exp ( x ), x, -1, 3 ),  
                color = red,  
                key = "A parametric curve",  
                parametric ( 2*cos ( t ), t^2, t, 0, 2*%pi ),  
                color = black, key = " a parametric vertical line",  
                parametric ( 2.5, t, t, 0, 25 ) )$
```



13 create an impulse plot from set of random points

```
(%i5) fpprintprec : 4$
```

```
(%i6) mypoints : makelist ( [random (20.0), random (50.0) ], k, 1, 10 );
```

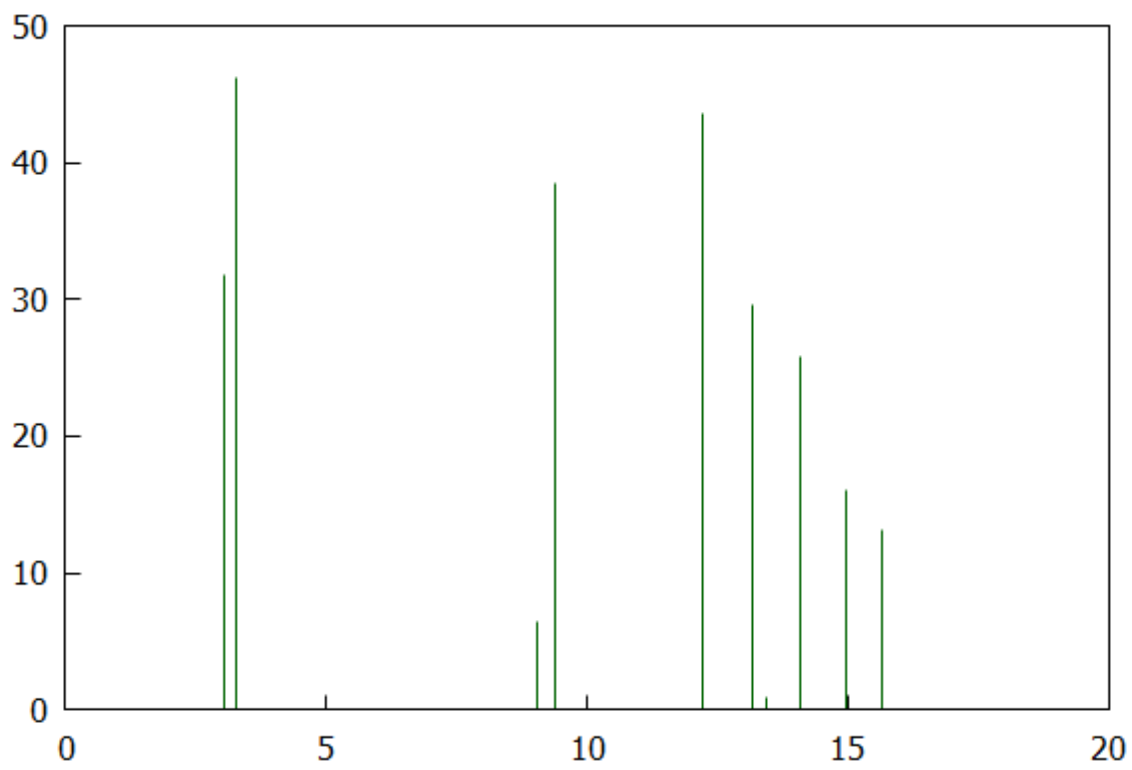
```
(mypoints)
```

```
[[12.23,43.56],[3.035,31.81],[13.17,29.53],[9.369,38.54],[14.09,25.79],[14.97,16.0],[9.054,6.411  
],[13.45,0.8185],[15.65,13.17],[3.257,46.18]]
```

```
(%i7) length (mypoints);
```

```
(%o7) 10
```

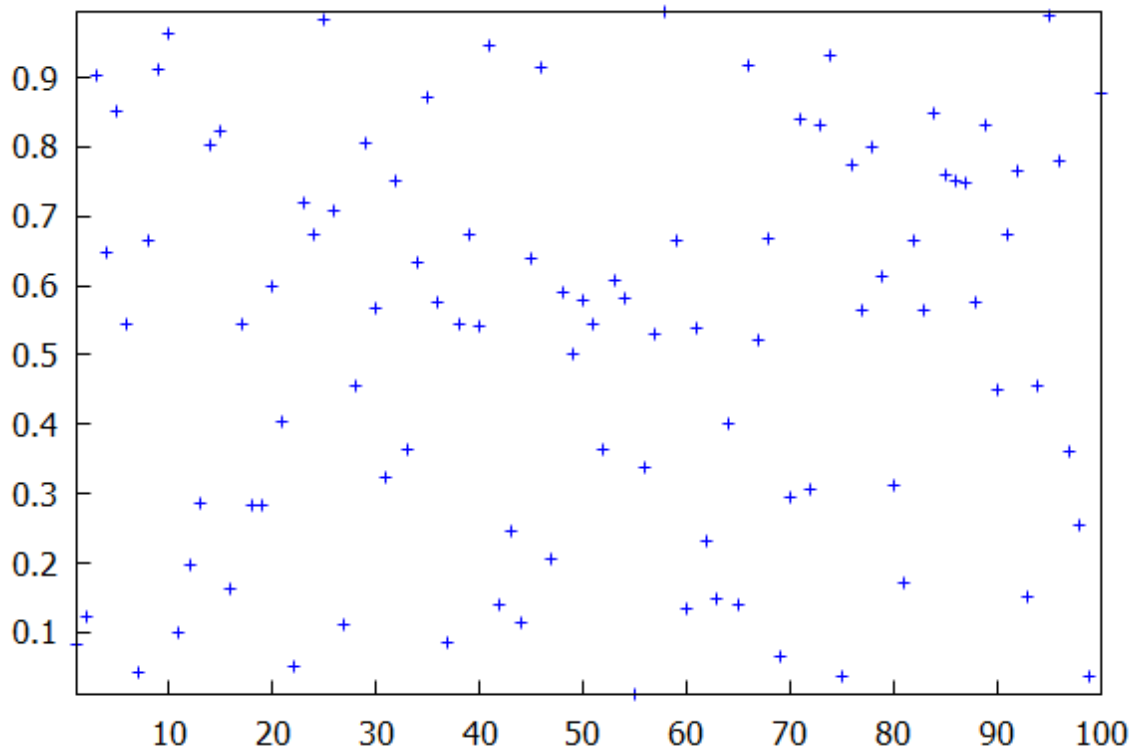
```
(%i8) wxdraw2d ( xrange = [0, 20], yrange = [0, 50],  
points_joined = impulses,  
line_width = 1,  
color = dark_green,  
points ( mypoints ) )$
```



14 create and plot an array of random points in two dimensions

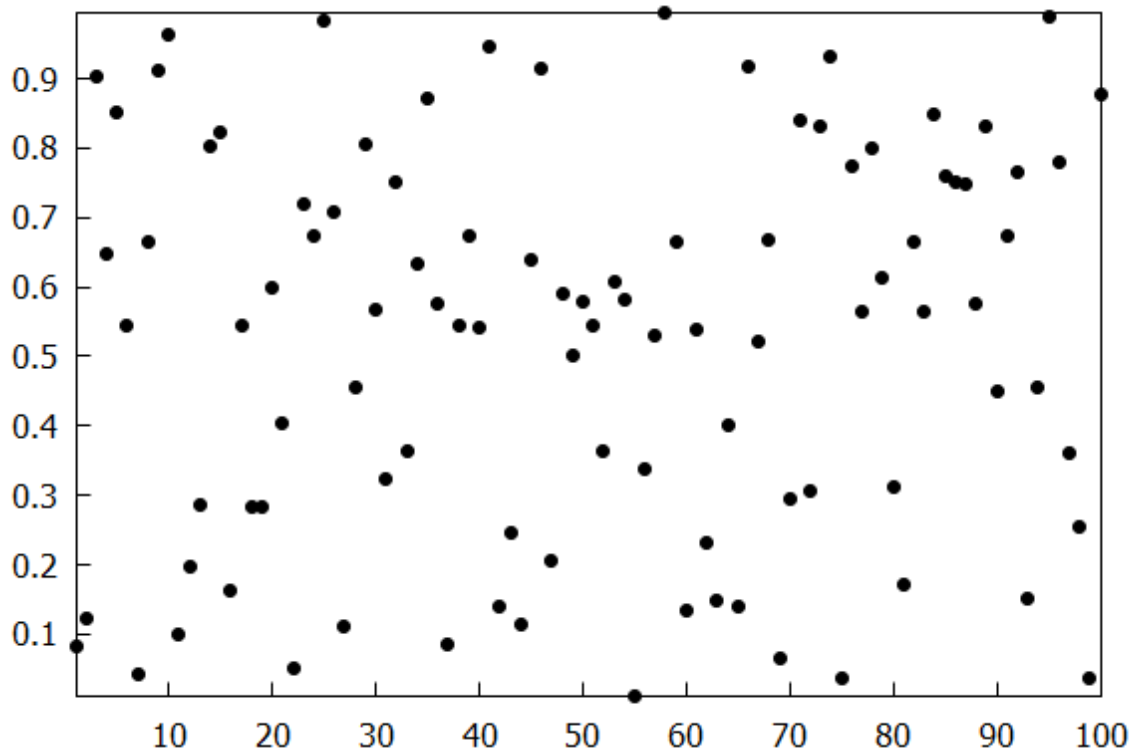
array a holds 100 random floating point numbers in the range from 0 to 1. Calling points(a) uses (1, a[0]) as point 1 to be plotted, (2, a[1]) as point 2 to be plotted, (100, a[99]) as the last point.

```
(%i11) a: make_array (flonum, 100) $  
      for i : 0 thru 99 do a[i] : random(1.0) $  
      wxdraw2d ( points ( a ) ) $
```



Change from the default cross to filled_circle and use the default size.

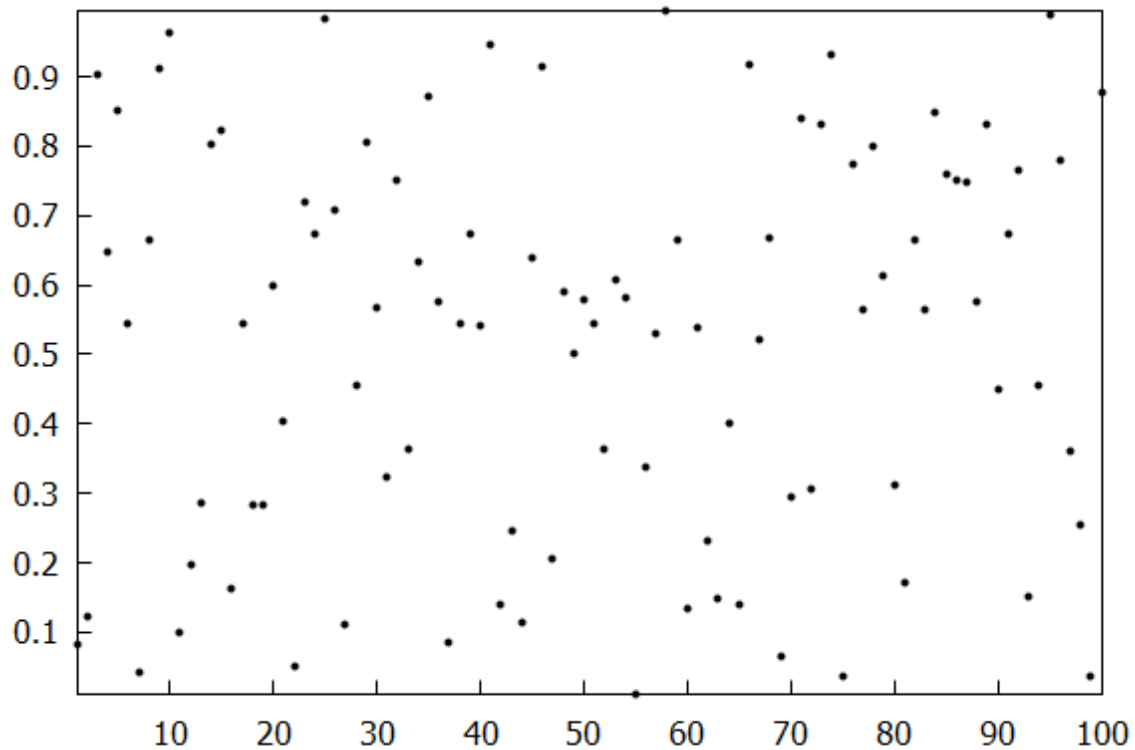
```
(%i12) wxdraw2d (color = black, point_type = filled_circle,  
                points ( a ) ) $
```



Decrease the point size.

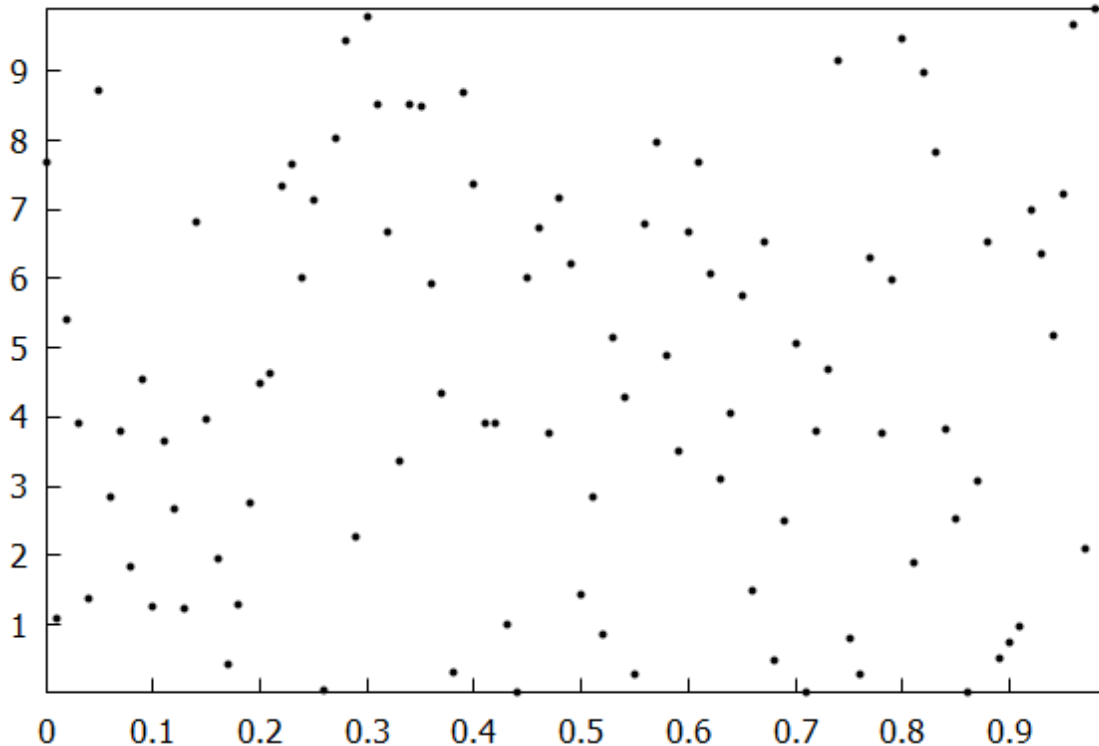
(%i13)

```
wxdraw2d (color = black, point_size = 0.5,  
          point_type = filled_circle, points ( a ) )$
```



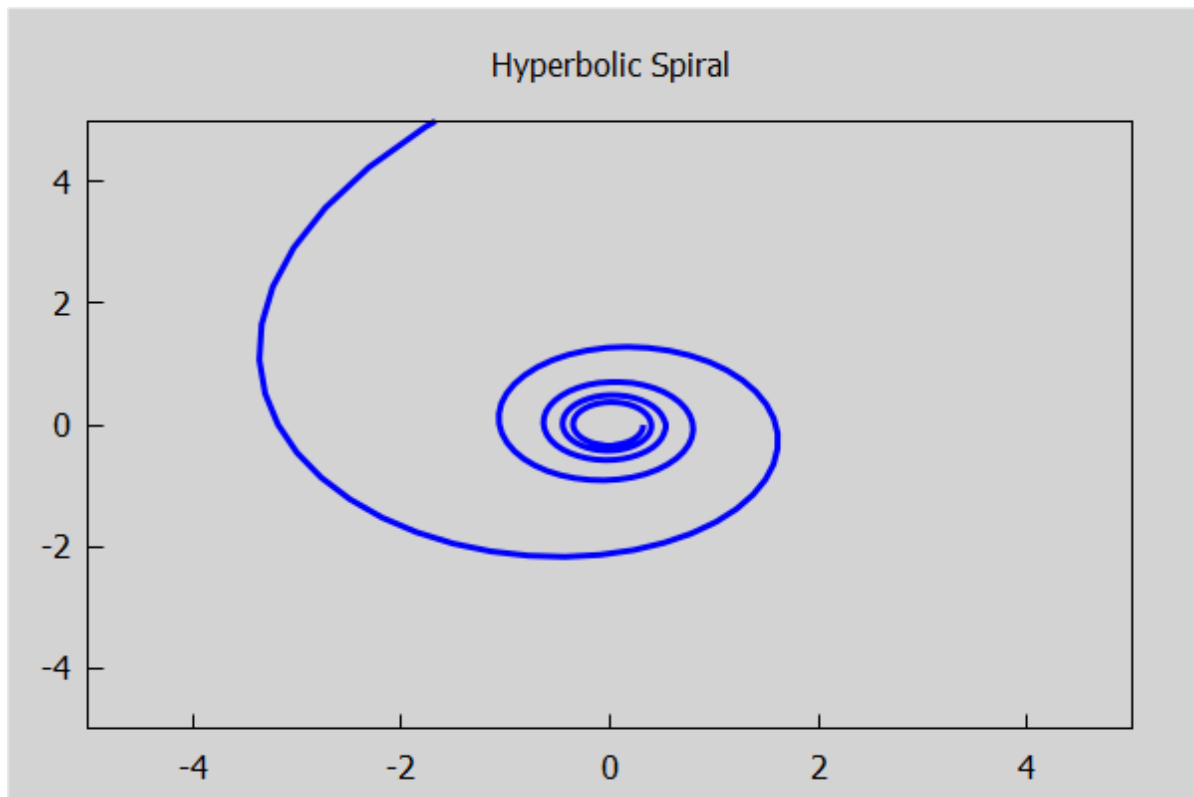
Two arrays, one holding the x coordinates, the other holding the y coordinates. $x[0] = 0$, $x[1] = 0.01$, ..., $x[99] = 0.99$

```
(%i17) x: make_array (flonum, 100) $  
      y: make_array (fixnum, 100) $  
      for i : 0 thru 99 do (  
        x[i] : float (i/100),  
        y[i] : random (10.0) ) $  
      wxdraw2d (color = black, point_type = filled_circle,  
               point_size = 0.5, points ( x, y ) ) $
```



15 hyperbolic spiral using polar

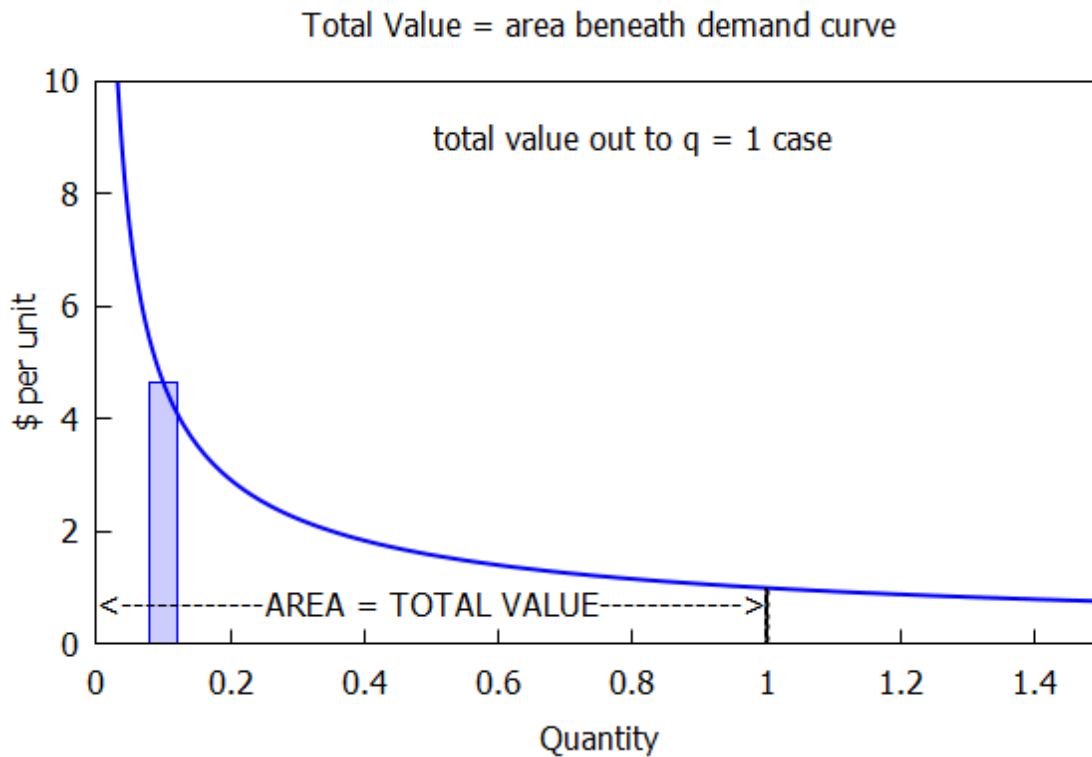
```
(%i18) wxdraw2d ( user_preamble = "set grid polar",  
                 nticks = 200, xrange = [-5, 5], yrange = [-5, 5],  
                 background_color = light_gray,  
                 color = blue, line_width = 3,  
                 title = "Hyperbolic Spiral",  
                 polar ( 10/theta, theta, 1, 10*%pi ) )$
```



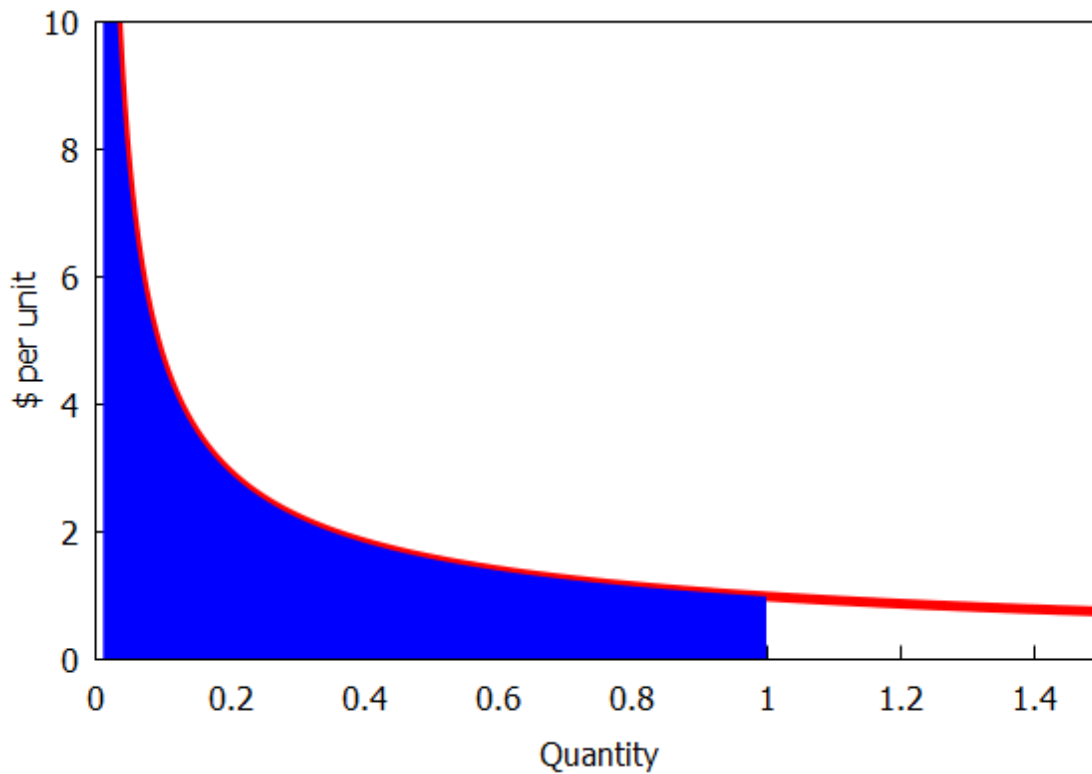
16 a demand curve

(%i19) $DP0 : 1/q^{(2/3)}$

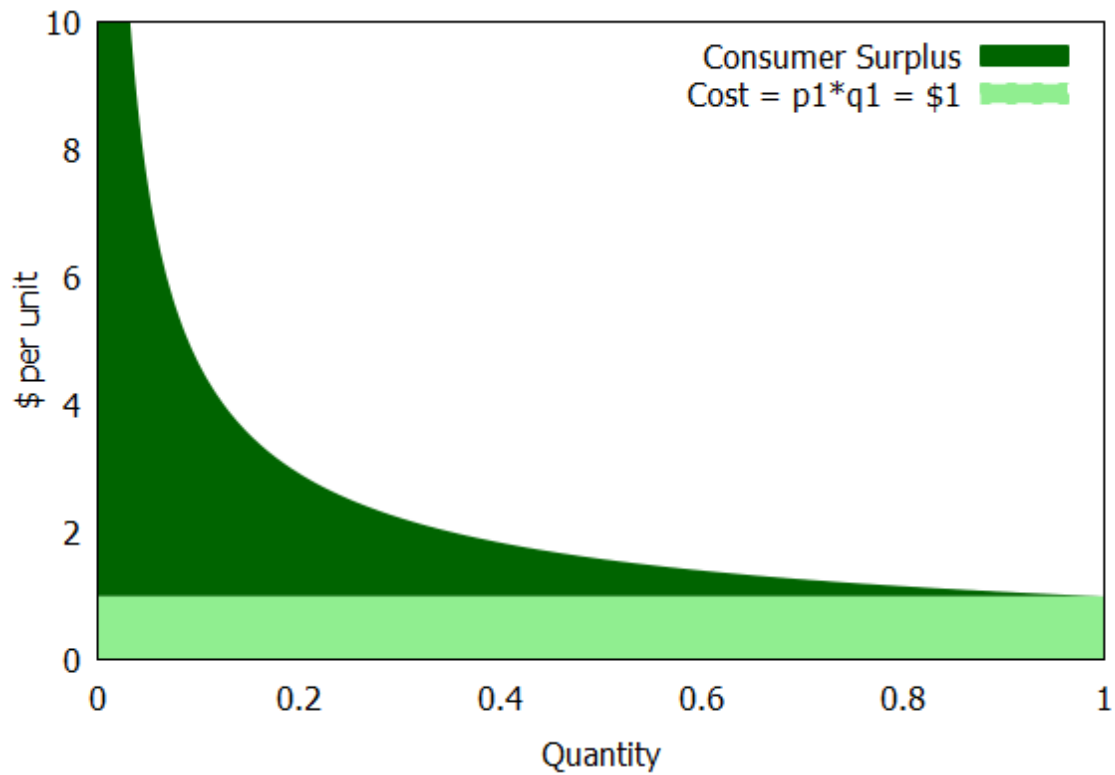
```
(%i20) wxdraw2d ( xlabel = "Quantity", ylabel = "$ per unit",
  title = "Total Value = area beneath demand curve",
  yrange = [0, 10],
  fill_color = blue,
  fill_density = 0.2,
  bars ( [0.1, subst (q = 0.1, DP0), 0.04]),
  line_width = 2,
  explicit ( DP0, q, 0, 1.5),
  color = black,
  parametric (1,t,t,0,1),
  label (["<-----AREA = TOTAL VALUE----->", 0.5, 0.7],
    ["total value out to q = 1 case", 0.8, 9] ) )$
```



```
(%i21) wxdraw2d ( xrange = [0, 1.5], yrange = [0, 10],  
                xlabel = "Quantity", ylabel = "$ per unit",  
                color = red,  
                line_width = 5,  
                explicit ( DP0, q, 0.01, 1.5),  
                fill_color = blue,  
                filled_func = true,  
                explicit ( DP0 , q, 0.01, 1) )$
```



```
(%i22) wxdraw2d ( xlabel = "Quantity", ylabel = "$ per unit",
  xrange = [0, 10],
  fill_color = dark_green,
  filled_func = 1,
  key = "Consumer Surplus",
  explicit ( DP0, q, 0.0001, 1),
  fill_color = light_green,
  filled_func = 0,
  key = " Cost = p1*q1 = $1",
  explicit (1, q, 0, 1) )$
```



17 supply curves

```
(%i24) SQ : b*p^ES$  
      SP : (q/b)^(1/ES)$
```

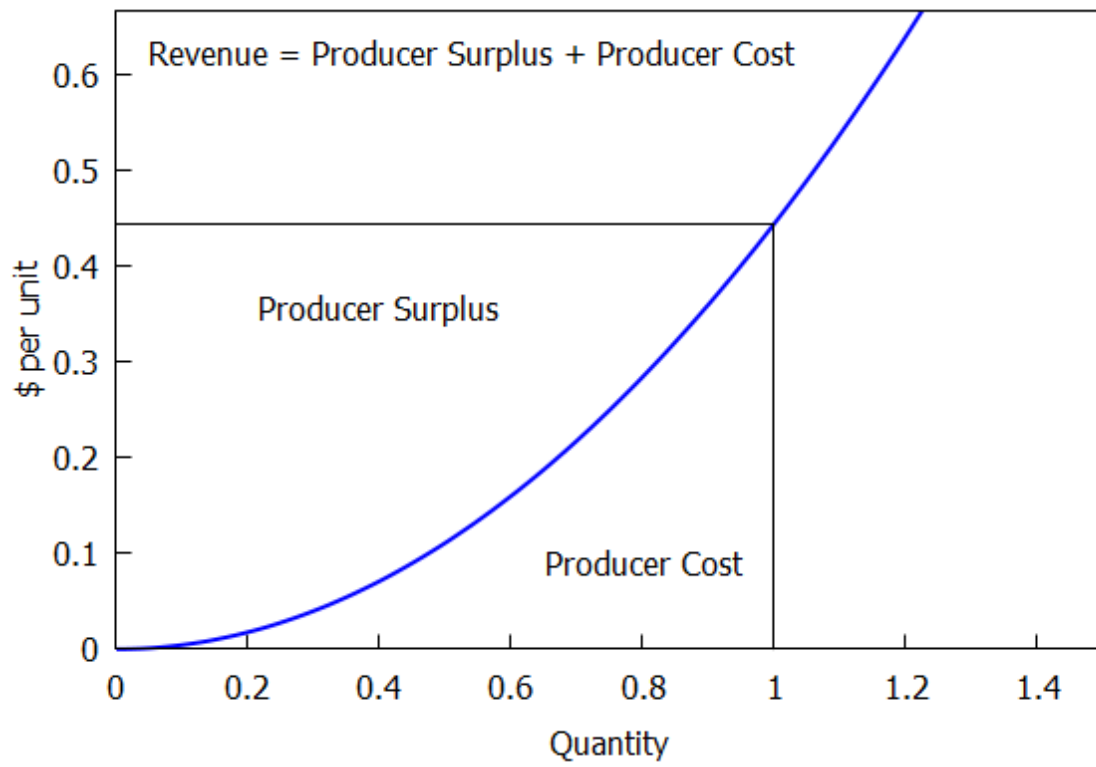
```
(%i25) [SQ0 : SQ, SP0 : SP], [b = 1.5, ES = 0.5]$
```

```
(%i26) ptemp : SP0, q = 1;
```

```
(ptemp) 0.4444
```

```
(%i27) q1 : 0.8$
```

```
(%i28) wxdraw2d ( xlabel = "Quantity", ylabel = "$ per unit",  
                xrange = [0, 1.5*ptemp],  
                line_width = 2,  
                explicit ( SP0, q, 0, 1.5*max (1, q1)),  
                color = black,  
                line_width = 1,  
                explicit (ptemp,q,0,1),  
                parametric (1,t,t,0,ptemp),  
                label ( ["Producer Cost", 0.8, 0.2*ptemp],  
                      ["Producer Surplus", 0.4, 0.8*ptemp] ),  
                label_alignment = left,  
                label ( ["Revenue = Producer Surplus + Producer  
Cost",0.05, 1.4*ptemp]) )$
```



plot of a supply curve (supply schedule): the greater the price, the greater the supply.

```
(%i30) Qlist : [ "Q, units per week :", 1000, 2000, 4000, 7000, 11000];  
Plist : ["P, $ per unit :", 5, 6, 7, 8, 9];  
(Qlist) [Q,unitsperweek:,1000,2000,4000,7000,11000]  
(Plist) [P,$perunit:,5,6,7,8,9]
```

```
(%i31) wxdraw2d (  
  xrange = [0, 12000],  
  yrange = [0, 10],  
  ylabel = "Price, $ per unit",  
  xlabel = "Quantity, units per week",  
  point_type = filled_circle,  
  points_joined = true,  
  points ( rest(Qlist), rest(Plist) ) )$
```

