

☐—Draw-Descriptions (Maxima)

Options

[Maxima Draw] `adapt_depth` (=10) maximum number of adaptive interval splits when drawing curves
`adapt_depth := adapt_depth`

[Maxima Draw] `axis_3d` (=true): draw 3D axes, false: don't draw axes
`axis_3d := axis_3d`

[Maxima Draw] `axis_bottom` (=true): draw bottom axis in 2D, false: don't draw axis
`axis_bottom := axis_bottom`

[Maxima Draw] `axis_top` (=true): draw top axis in 2D, false: don't draw axis
`axis_top := axis_top`

[Maxima Draw] `axis_left` (=true): draw left axis in 2D, false: don't draw axis
`axis_left := axis_left`

[Maxima Draw] `axis_right` (=true): draw right axis in 2D, false: don't draw axis
`axis_right := axis_right`

[Maxima Draw] `background_color` (=white): background color of the plot
`background_color := background_color`

[Maxima Draw] `border` (=true): draw borders of `polygon()`, `ellipse()` or `rectangle()` objects
`border := border`

[Maxima Draw] `capping` (=false): indicates if extremes of the tube object remain open (false) or are closed (true). If only one end is to be closed, provide a list with two entries.
`capping := capping`

[Maxima Draw] `cbrange` (=auto): color bar range (list min,max)
`cbrange := cbrange`

[Maxima Draw] `cbticks` (=auto): color bar tick marks
`cbticks := cbticks`

[Maxima Draw] `color` (=blue): color for lines, points, borders and text, given as name or as "#rrggb"
`color := color`

[Maxima Draw] `colorbox` (=true): draw color scale without label, false: no colorbar, string: draw colorbar with string as label.
`colorbox := colorbox`

[Maxima Draw] `contour` (=none): contour lines. none: no contours, base: on xy plane, surface: on 3D surface, both: on xy plane and surface, map: on xy plane plus vertical view.
`contour := contour`

[Maxima Draw] `contour_levels` (=5): contour levels. number oder list (min, Δ , max), or set(L1, L2,...) of values
`contour_levels := contour_levels`

[Maxima Draw] `data_file_name` ("data.gnuplot"): internal data file name for gnuplot
`data_file_name := data_file_name`

[Maxima Draw] `delay` (=5): time step in animated gif files
`delay := delay`

[Maxima Draw] `dimensions` (= 600 x 500) list [width, height]
`dimensions := dimensions`

[Maxima Draw] `draw_realpart` (=true) use real part for plotting complex values

`draw_realpart := draw_realpart`

[Maxima Draw] `enhanced3d` (=none) don't use color palette, draw 3D surfaces as grid (control using `color` und `surface_hide`), true: coloring of surfaces by z-value, or list `[f(x,y,z), x, y, z]`: coloring according to function `f` or list `[f(u,v),u,v]` or list `[f(k), k]` (`k` being the point counter in `points()` objects)

`enhanced3d := enhanced3d`

[Maxima Draw] `error_type` (=y) type of error bars in `errors()`: `x`, `y`, `xy` or `boxes`

`error_type := error_type`

[Maxima Draw] `file_name` ("maxima_out") output file, extension is set according to format

`file_name := file_name`

[Maxima Draw] `fill_color` (=red) fill color for polygons and explicit functions `y(x)` in 2D

`fill_color := fill_color`

[Maxima Draw] `fill_density` (=0) opacity of the fill color in `bars()`-objects, 0..1

`fill_density := fill_density`

[Maxima Draw] `filled_func` (=false): don't fill explicit functions `y(x)`, true: fill between the next `explicit()` object and the bottom border of the diagram, `g(x)`: fill between `g(x)` and the next `explicit()` object

`filled_func := filled_func`

[Maxima Draw] `gnuplot_file_name` ("maxout.gnuplot"): name of the temporary gnuplot command file

`gnuplot_file_name := gnuplot_file_name`

[Maxima Draw] `gnuplot_term` (): gnuplot terminal options (arguments of "set terminal")

`gnuplot_term := gnuplot_term`

[Maxima Draw] `gnuplot_out_file` (=maxplot.xxx) output file name (xxx depends on terminal)

`gnuplot_out_file := gnuplot_out_file`

[Maxima Draw] `grid` (=false): don't draw xy grid, true: draw grid

`grid := grid`

[Maxima Draw] `head_angle` (=45): head angle of `vector()`-Objekten (angle between border and vector direction in °)

`head_angle := head_angle`

[Maxima Draw] `head_both` (=false): don't draw a second head of `vector()` objects, true: draw a second head

`head_both := head_both`

[Maxima Draw] `head_length` (=2): head length of `vector()` objects in units of the x-axis

`head_length := head_length`

[Maxima Draw] `head_type` (=filled) head type of `vector()` objects: filled, empty or nonfilled

`head_type := head_type`

[Maxima Draw] `ip_grid` ({50, 50}): primary grid for sampling of implicit functions, two-element list

`ip_grid := ip_grid`

[Maxima Draw] `ip_grid_in` ({5, 5}): secondary grid for sampling of implicit functions

`ip_grid_in := ip_grid_in`

[Maxima Draw] `key` (""): legend key for subsequent graphics object.

`key := key`

[Maxima Draw] `key_pos` ("") equivalent to `top right`: position of the legend. Valid options: top

[Maxima Draw] *key_pos* (= , equivalent to *top_right*): position of the legend. valid options: *top_left*, *top_center*, *top_right*, *center_left*, *center*, *center_right*, *bottom_left*, *bottom_center*, *bottom_right*

key_pos := *key_pos*

[Maxima Draw] *label_alignment* (=center), left or right: alignment of text labels

label_alignment := *label_alignment*

[Maxima Draw] *label_orientation* (=horizontal) or vertical: orientation of text labels

label_orientation := *label_orientation*

[Maxima Draw] *line_type* (=solid) line type. Other options: *dots* (for all formats) or if supported by Gnuplot: *dashes*, *short_dashes*, *short_long_dashes*, *short_short_long_dashes*

line_type := *line_type*

[Maxima Draw] *line_width* (=1) , >0

line_width := *line_width*

[Maxima Draw] *logcb* (=false) or true: log scale of the colorbar

logcb := *logcb*

[Maxima Draw] *logx* (=false) or true: log scale of x-axis

logx := *logx*

[Maxima Draw] *logx_secondary* (=false) or true: log scale of secondary x-axis

logx_secondary := *logx_secondary*

[Maxima Draw] *logy* (=false) or true: log scale of y-axis

logy := *logy*

[Maxima Draw] *logy_secondary* (=false) or true: log scale of secondary y-axis

logy_secondary := *logy_secondary*

[Maxima Draw] *logz* (=false) or true: log scale of z-axis

logz := *logz*

[Maxima Draw] *nticks* (=29): number of points for curve sampling (explicit and parametric)

nticks := *nticks*

[Maxima Draw] *palette* (=color) default color palette, gray: gray levels, list { *color1*, *color2*... or false: don't color 3D objects

palette := *palette*

[Maxima Draw] *point_size* (=1) size of points()-symbols

point_size := *point_size*

[Maxima Draw] *point_type* (=1) type of points() objects, name or key number: none -1, dot 0, plus 1, multiply 2, asterisk 3, square 4, filled_square 5, circle 6, filled_circle 7, up_triangle 8, filled_up_triangle 9, down_triangle 10, filled_down_triangle 11, diamant 12, filled_diamant 13

point_type := *point_type*

[Maxima Draw] *points_joined* (=false), true or impulses: how to join points

points_joined := *points_joined*

[Maxima Draw] *proportional_axes* (=none), xy or xyz: set axes of equal scaling)

proportional_axes := *proportional_axes*

[Maxima Draw] *surface_hide* (=false) or true: handling of hidden lines in surface grids

surface_hide := *surface_hide*

[Maxima Draw] *terminal* (=screen) separate interactive terminal window, png, jpg, gif, eps, svg, pdf,

animated_gif: gnuplot output format

[Maxima Draw] title (="") headline for the plot

title := title

[Maxima Draw] transform (=none) co-ordinate transformation, for 2D as list {f1(x,y), f2(x,y), x, y, for 3D as list {f1(x,y), f2(x,y), f3(x, y, z), x, y, z

transform := transform

[Maxima Draw] transparent (=false) fill objects with border, true: don't fill (make them transparent)

transparent := transparent

[Maxima Draw] unit_vectors (=false) draw vectors in original length, true: draw unit vectors.

unit_vectors := unit_vectors

[Maxima Draw] user_preamble (="") string or list of strings, gnuplot commands placed immediately before the plot commands

user_preamble := user_preamble

[Maxima Draw] view (=(60,30)) view direction

view := view

[Maxima Draw] wired_surface (=false) no grid on colored surfaces, true: draw grid on surfaces

wired_surface := wired_surface

[Maxima Draw] x_voxel (=10) number of x steps for the Marching Cube algorithm (for implicit 3D functions)

x_voxel := x_voxel

[Maxima Draw] y_voxel (=10) number of y steps for the Marching Cube algorithm (for implicit 3D functions)

y_voxel := y_voxel

[Maxima Draw] z_voxel (=10) number of z steps for the Marching Cube algorithm (for implicit 3D functions)

z_voxel := z_voxel

[Maxima Draw] xaxis (=false) or true: draw x-axis

xaxis := xaxis

[Maxima Draw] yaxis (=false) or true: draw y-axis

yaxis := yaxis

[Maxima Draw] zaxis (=false) or true: draw z-axis

zaxis := zaxis

[Maxima Draw] xaxis_color (=black) color of the x-axis

xaxis_color := xaxis_color

[Maxima Draw] yaxis_color (=black) color of the y-axis

yaxis_color := yaxis_color

[Maxima Draw] zaxis_color (=black) color of the z-axis

zaxis_color := zaxis_color

[Maxima Draw] xaxis_secondary (=false) or true: use the secondary x-axis (top) for subsequent objects

xaxis_secondary := xaxis_secondary

[Maxima Draw] yaxis_secondary (=false) or true: use the secondary y-axis (right) for subsequent objects

yaxis_secondary := yaxis_secondary

[Maxima Draw] xaxis_type (=dots) or solid

xaxis_type := xaxis_type

[Maxima Draw] `yaxis_type` (=dots) or solid

`yaxis_type := yaxis_type`

[Maxima Draw] `zaxis_type` (=dots) or solid

`zaxis_type := zaxis_type`

[Maxima Draw] `xaxis_width` (=1)

`xaxis_width := xaxis_width`

[Maxima Draw] `yaxis_width` (=1)

`yaxis_width := yaxis_width`

[Maxima Draw] `zaxis_width` (=1)

`zaxis_width := zaxis_width`

[Maxima Draw] `xlabel` (="") label of the x-axis

`xlabel := xlabel`

[Maxima Draw] `ylabel` (="") label of the y-axis

`ylabel := ylabel`

[Maxima Draw] `zlabel` (="") label of the z-axis

`zlabel := zlabel`

[Maxima Draw] `xlabel_secondary` (="") label of the secondary x-axis

`xlabel_secondary := xlabel_secondary`

[Maxima Draw] `ylabel_secondary` (="") label of the secondary y-axis

`ylabel_secondary := ylabel_secondary`

[Maxima Draw] `xrange` (=auto) or list {min,max: range of the x-axis

`xrange := xrange`

[Maxima Draw] `yrange` (=auto) or list {min,max: range of the y-axis

`yrange := yrange`

[Maxima Draw] `zrange` (=auto) or list {min,max: range of the z-axis

`zrange := zrange`

[Maxima Draw] `xrange_secondary` (=auto) or list {min,max: range of the secondary (top) x-axis

`xrange_secondary := xrange_secondary`

[Maxima Draw] `yrange_secondary` (=auto) or list {min,max: range of the secondary (right) y-axis

`yrange_secondary := yrange_secondary`

[Maxima Draw] `xtics` (=auto): automatic x ticks, none: no x ticks, number: tick interval, list of {min, step, max, or set() of numbers: explicit tick values, or set() of lists ("text",pos) for spec of position and text.

`xtics := xtics`

[Maxima Draw] `ytics` (=auto): automatic y ticks, none: no y ticks, number: tick interval, list of {min, step, max, or set() of numbers: explicit tick values, or set() of lists ("text",pos) for spec of position and text.

`ytics := ytics`

[Maxima Draw] `ztics` (=auto): automatic z ticks, none: no x ticks, number: tick interval, list of {min, step, max, or set() of numbers: explicit tick values, or set() of lists ("text",pos) for spec of position and text.

`ztics := ztics`

[Maxima Draw] `xtics_secondary` (=auto): automatic ticks on the secondary (top) x-axis, none: no ticks, number: tick interval

ticks, number: tick interval,
list of {min, step, max, or set()} of numbers: explicit tick values, or set() of lists
("text",pos) for spec of position and text.

`xtics_secondary:=xtics_secondary`

[Maxima Draw] `ytics_secondary` (=auto): automatic ticks on the secondary (right) y-axis, none: no ticks, number: tick interval,
list of {min, step, max, or set()} of numbers: explicit tick values, or set() of lists
("text",pos) for spec of position and text.

`ytics_secondary:=ytics_secondary`

[Maxima Draw] `xtics_axis` (=false): ticks on the boundary, true: ticks on the axis

`xtics_axis:=xtics_axis`

[Maxima Draw] `ytics_axis` (=false): ticks on the boundary, true: ticks on the axis

`ytics_axis:=ytics_axis`

[Maxima Draw] `ztics_axis` (=false): ticks on the boundary, true: ticks on the axis

`ztics_axis:=ztics_axis`

[Maxima Draw] `xtics_secondary_axis` (=false): ticks on the secondary (top) boundary, true: ticks on the axis

`xtics_secondary_axis:=xtics_secondary_axis`

[Maxima Draw] `ytics_secondary_axis` (=false): ticks on the secondary (right) boundary, true: ticks on the axis

`ytics_secondary_axis:=ytics_secondary_axis`

[Maxima Draw] `xtics_rotate` (=false): horizontal tick labels, true: vertical tick labels

`xtics_rotate:=xtics_rotate`

[Maxima Draw] `ytics_rotate` (=false): horizontal tick labels, true: vertical tick labels

`ytics_rotate:=ytics_rotate`

[Maxima Draw] `ztics_rotate` (=false): horizontal tick labels, true: vertical tick labels

`ztics_rotate:=ztics_rotate`

[Maxima Draw] `xtics_rotate_secondary` (=false): don't rotate ticks of the secondary (top) x-axis, true: rotate about 90°

`xtics_rotate_secondary:=xtics_rotate_secondary`

[Maxima Draw] `ytics_rotate_secondary` (=false): don't rotate ticks of the secondary (right) y-axis, true: rotate about 90°

`ytics_rotate_secondary:=ytics_rotate_secondary`

[Maxima Draw] `xu_grid` (=30) sampling points in x- or u-direction

`xu_grid:=xu_grid`

[Maxima Draw] `yv_grid` (=30) sampling points in y- or v-direction

`yv_grid:=yv_grid`

[Maxima Draw3D] `xyplane` (=false): automatic allocation of the xy-plane, reax number: xy-plane intersects the z-axis at this level

`xyplane:=xyplane`

Objekte

[Maxima Draw2D] `bars(b1,b2...)` vertical bars, given as lists (x,h,w). Options: key, color, fill_color, fill_density and line_width

`bars ( ):=bars ( )`

[Maxima Draw3D] `elevation_grid(Matrix,x0,y0,w,h)` Draws a m x n-matrix in 3D. M₁₁ at x=x0,y=y0+h, M_{mn} at x=x0+w,y=y0. Options: xu_grid, yv_grid, line_type, key, wired_surface, enhanced3d and color

`elevation_grid( 1, 2, 3, 4, 5):=elevation_grid( 1, 2, 3, 4, 5)`

[Maxima Draw2D] `ellipse(xc,yc,a,b,φ1,φ2)` plots an ellipse centered at [xc, yc] with horizontal and vertical semi axis a and b, respectively, starting at angle ang1 with an amplitude equal to angle ang2. Options: nticks, transparent, fill_color, border, line_width, line_type, key and color.

[Maxima Draw2D] `express(a1,a2,...)` Draws points with error bars horizontally, vertically or both

[Maxima Draw2D] `errors(e1,e2,...)` draws points with error bars, horizontally, vertically or both, depending on the value of option `error_type`.

If `error_type = x`, arguments are lists `[x, y, xdelta]` or `[x, y, xlow, xhigh]`. If `error_type = y`, arguments are lists `[x, y, ydelta]` or `[x, y, ylow, yhigh]`.

If `error_type = xy` or `error_type = boxes`, arguments are lists `[x, y, xdelta, ydelta]` or `[x, y, xlow, xhigh, ylow, yhigh]`.

Options: `error_type`, `points_joined`, `line_width`, `key`, `line_type`, `color`, `fill_density`, `xaxis`

[Maxima Draw2D] `explicit(f,x,xmin,xmax)` plots explicit function `f`, with variable `x` taking values from `xmin` to `xmax`.

Options: `nticks`, `adapt_depth`, `draw_realpart`, `line_width`, `line_type`, `key`, `filled_func`, `fill_color` and `color`

[Maxima Draw3D] `explicit(f,x,xmin,xmax,y,ymin,ymax)` plots the explicit function `f`, with variable `x` taking values from `xmin` to `xmax` and variable `y`

taking values from `ymin` to `ymax`. Options: `draw_realpart`, `xu_grid`, `yv_grid`, `line_type`, `line_width`, `key`, `wired_surface`, `enhanced3d`, and `color`.

[Maxima Draw2D] `image(matrix,x0,y0,w,h)` plots matrix in the rectangular region from vertex `(x0,y0)` to `(x0+w,y0+h)` according to option `palette`.

`image(#1;#2;#3;#4;#5):=image(#1;#2;#3;#4;#5)`

[Maxima Draw2D] `implicit(f,x,xmin,xmax,y,ymin,ymax)` plots the implicit function defined by `f`, with variable `x` taking values from `xmin` to `xmax`, and variable `y` taking values from `ymin` to `ymax`.

Options: `ip_grid`, `ip_grid_in`, `line_width`, `line_type`, `key`, `color`

`implicit(#1;#2;#3;#4;#5;#6;#7):=implicit(#1;#2;#3;#4;#5;#6;#7)`

[Maxima Draw3D] `implicit(f,x,xmin,xmax,y,ymin,ymax,z,zmin,zmax)` plots the implicit surface defined by `f`,

with variable `x` taking values from `xmin` to `xmax`, variable `y` taking values from `ymin` to `ymax` and variable `z` taking values from `zmin` to `zmax`.

This object implements the marching cubes algorithm. Options: `x_voxel`, `y_voxel`, `z_voxel`, `line_width`, `line_type`, `key`, `wired_surface`, `enhanced3d` and `color`

[Maxima Draw2D and 3D] `label(list) text label` specified in 2D as list `[text,x,y]`, in 3D as list `[text,x,y,z]`,

Options: `color`, `label_alignment`, `label_orientation`

`label( ):=label( )`

[Maxima Draw2D] `parametric(fx,fy,p,pmin,pmax)` plots the parametric function `fx`, `fy` with parameter `p` taking values from `pmin` to `pmax`. Options

`nticks`, `line_width`, `line_type`, `key`, `color`, `enhanced3d`

`parametric(#1;#2;#3;#4;#5):=parametric(#1;#2;#3;#4;#5)`

[Maxima Draw3D] `parametric(fx,fy,fz,p,pmin,pmax)` parametrisch gegebene Kurve. Optionen: `nticks`, `line_width`, `line_type`, `key`, `color`, `enhanced3d`

`parametric(#1;#2;#3;#4;#5;#6):=parametric(#1;#2;#3;#4;#5;#6)`

[Maxima Draw2D and 3D] `points(matrix)` 2D: `n x 2`, 3D: `n x 3`, plots points, given as the rows in a matrix.

special options: `points_joined`, `point_type`, `point_size`

`points(#1):=points(#1)`

[Maxima Draw2D] `points(vx,vy)` points given by two vectors, special options: `points_joined`, `point_type`, `point_size`

`points(#1;#2):=points(#1;#2)`

[Maxima Draw3D] `points(vx,vy,vz)` points given by three vectors, special options: `points_joined`, `point_type`, `point_size`

`points(#1;#2;#3):=points(#1;#2;#3)`

[Maxima Draw2D] `polar(r,φ,φmin,φmax)` draws functions given in polar co-ordinates (radius `r` as function of angle `φ`). Special options: `key`, `nticks`, `line_width`, `line_type`, `color`

`polar(#1;#2;#3;#4):=polar(#1;#2;#3;#4)`

[Maxima Draw2D] `polygon(matrix)` polygon defined by matrix (each row one point). Special options: `transparent`, `fill_color`, `border`, `line_width`, `key`, `line_type` `color`

`polygon(#1):=polygon(#1)`

[Maxima Draw2D] `polygon(vx,vy)` polygon defined by two vectors. Special options: `transparent`, `fill_color`, `border`, `line_width`, `key`, `line_type` `color`

`polygon(#1;#2):=polygon(#1;#2)`

[Maxima Draw2D/3D] `quadrilateral(l1,l2,l3,l4)` quadrilateral given by lists of vertex coordinates: `[x,y]` in 2D or `[x,y,z]` in 3D.
Special options: `transparent`, `fill_color`, `border`, `line_width`, `key`, `line_type`, `color`, `enhanced3D` (3D only)

`quadrilateral(#1; #2; #3; #4) := quadrilateral(#1; #2; #3; #4)`

[Maxima Draw2D] `rectangle(l1,l2)` specified by lists `(x,y)` of opposite corners.
Special options: `transparent`, `fill_color`, `border`, `line_width`, `key`, `line_type`, `color`

`rectangle(#1; #2) := rectangle(#1; #2)`

[Maxima Draw2D] `region(expr,x,xmin,xmax,y,ymin,ymax)` Fills the region where `expr` is true in the domain given by `xmin`, `xmax`, `ymin`, `ymax`. Options: `fill_color`, `key`, `x_voxel`, `y_voxel`

`region(#1; #2; #3; #4; #5; #6; #7) := region(#1; #2; #3; #4; #5; #6; #7)`

`spherical()` funktioniert nicht

[Maxima Draw2D/3D] `triangle(l1,l2,l3)` given by co-ordinate lists `(x,y)` (2D) or `(x,y,z)` (3D) of the corners
Special options: `transparent`, `fill_color`, `border`, `line_width`, `key`, `line_type`, `color`

`triangle(#1; #2; #3) := triangle(#1; #2; #3)`

[Maxima Draw3D] `tube(fx,fy,fz,fr,p,pmin,pmax)` tube, given by functions `fx,fy,fz` of the centerline and radius function `fr`
`pf` parameters `p`.
Special options: `xu_grid`, `yv_grid`, `line_type`, `line_width`, `key`, `wired_surface`, `enhanced3d`, `color`, `capping`.

`tube(#1; #2; #3; #4; #5; #6; #7) := tube(#1; #2; #3; #4; #5; #6; #7)`

[Maxima Draw2D/3D] `vector(l1,l2)` given by co-ordinate lists `(x,y)` (2D) or `(x,y,z)` (3D) of the end points.
Spezielle Optionen: `head_both`, `head_length`, `head_angle`, `head_type`, `line_width`, `line_type`, `key`, `color`.

`vector(#1; #2) := vector(#1; #2)`

[Maxima] `set(e1, e2, ...)` Maxima set (in Maxima this is a list in curly braces)

`set(_):= set(_)`

[Maxima] `args(expr)` list of the top level arguments of `expr`

`args(_):= args(_)`

Sonstige

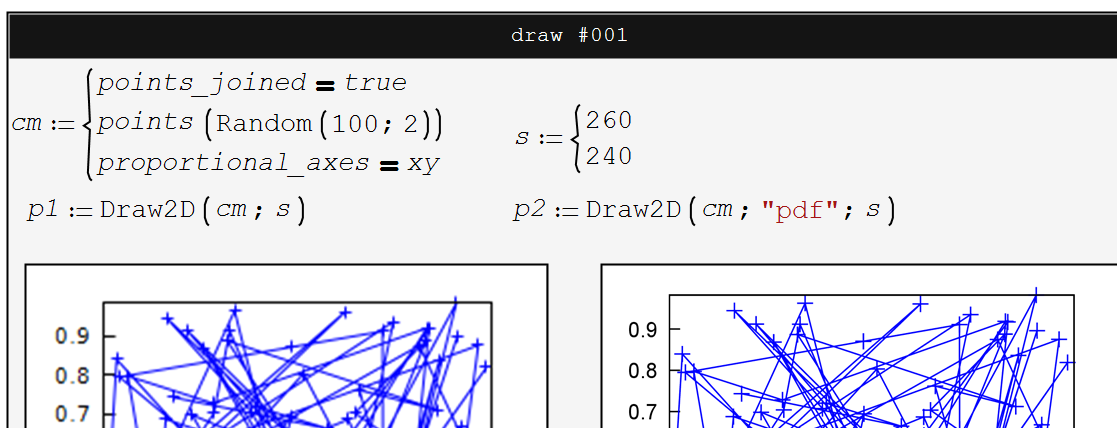
[Maxima]

[Maxima]

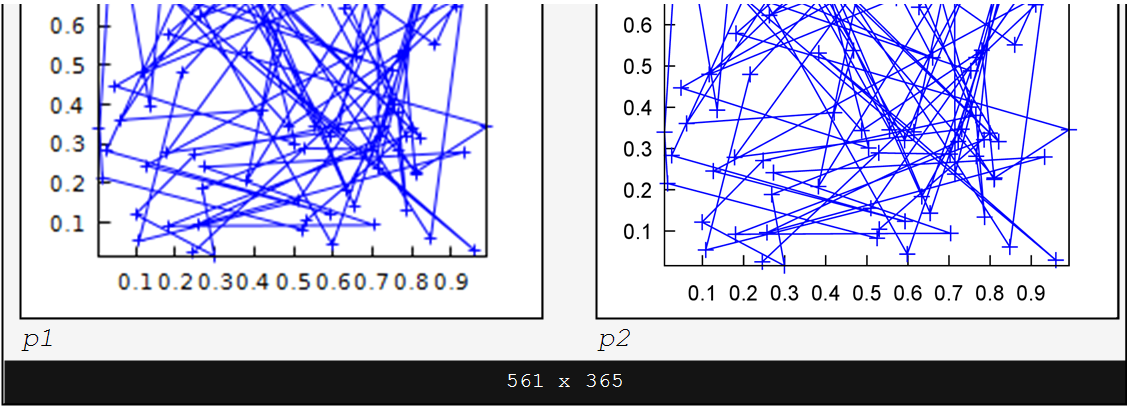
`true := true` `false := false`

`CurrentDirectory(DocumentDirectory("")) = "C:\FHB\Software\SMath\SMath Skript\Bilder\"`

`MaximaControl("restart") = "Restart complete."`



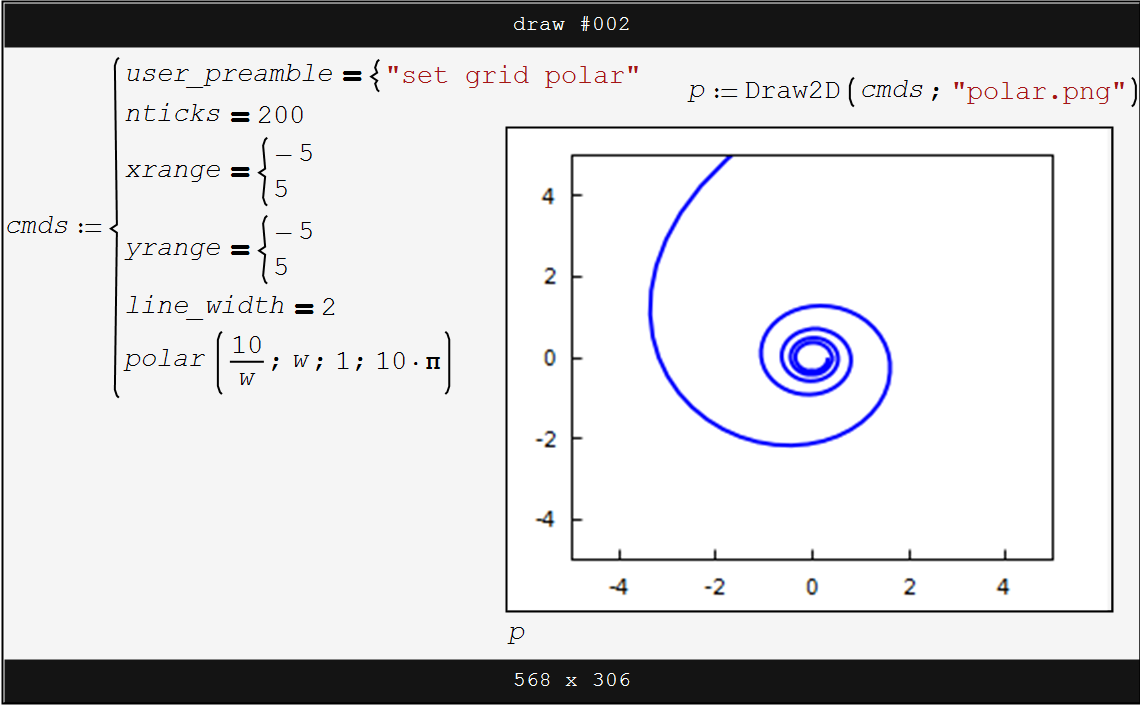
MaximaLog(■) = (%i
(%o



"draw"

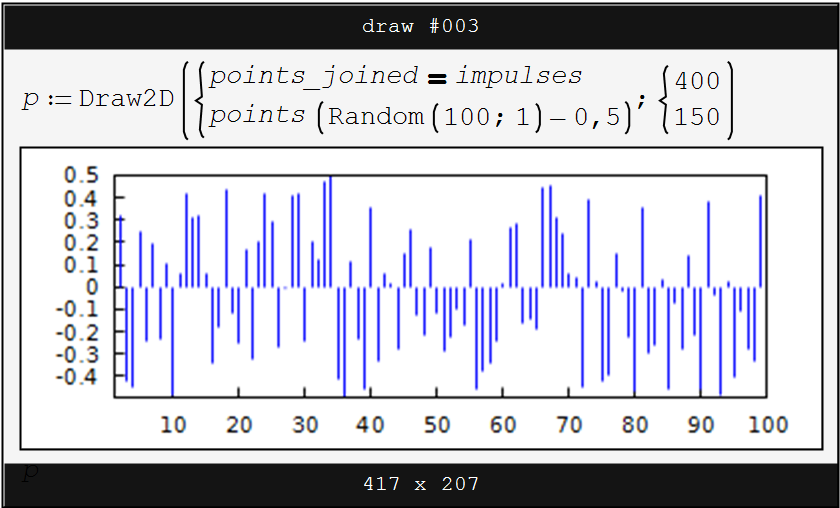
Draw-Descriptions (Maxima)

dra

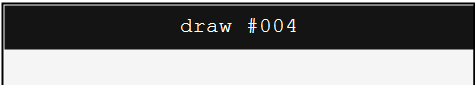


MaximaLog (■) = (%i24) c
(%o24)

"draw"

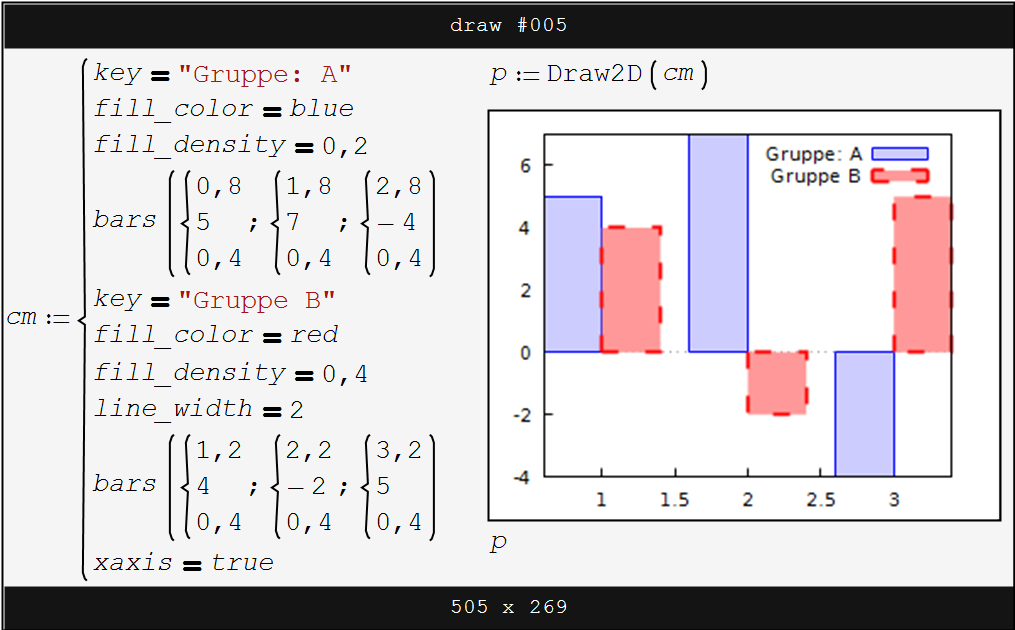


"draw"



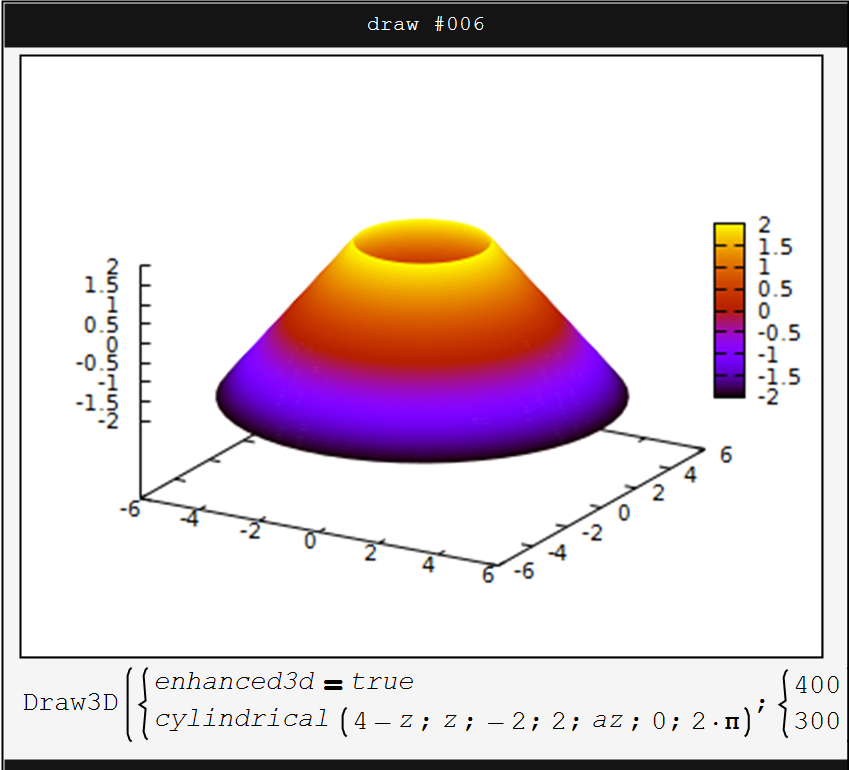


"draw"



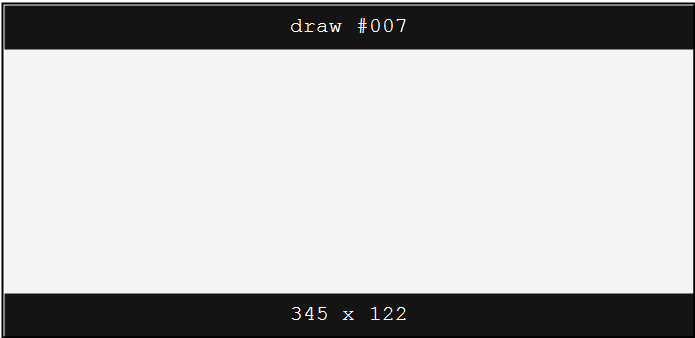
"draw"

"Received Bytes:
MaximaLog (■) = (%i26) draw2d([t
(%o26) [gr2d(bars

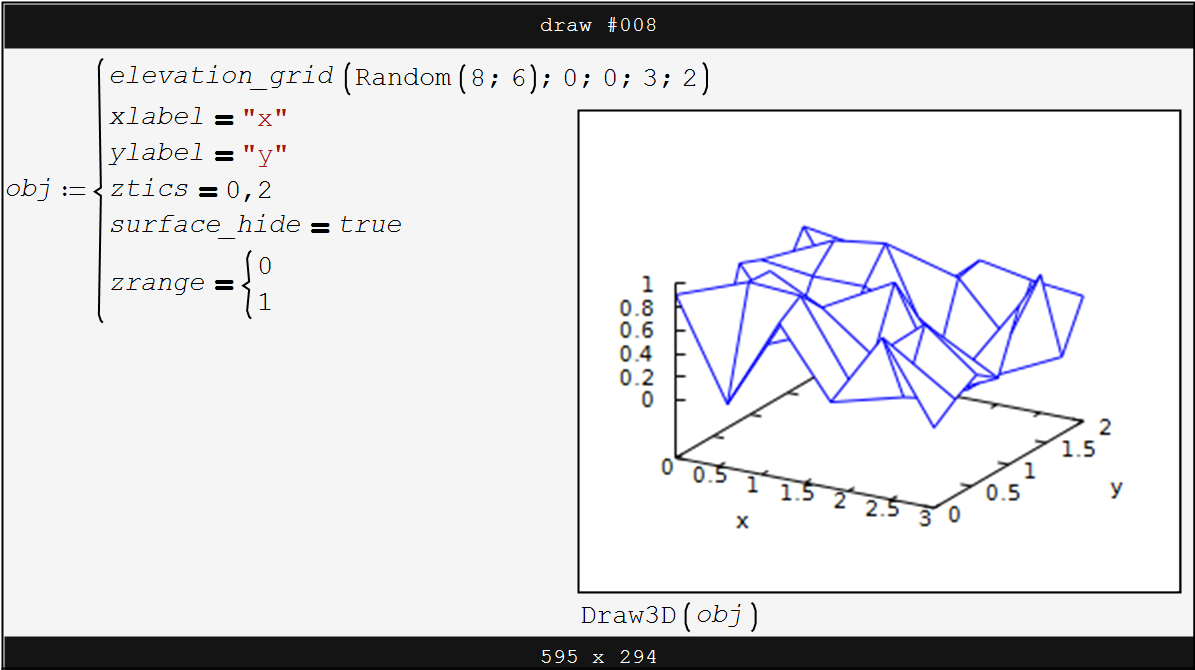


422 x 356

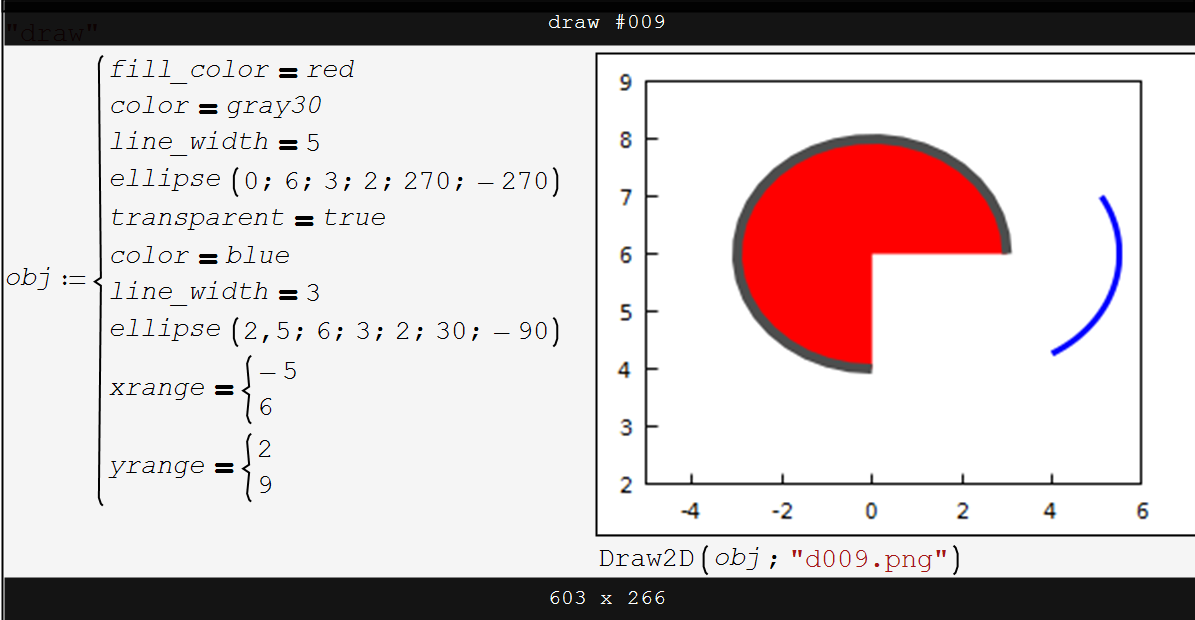
"draw"



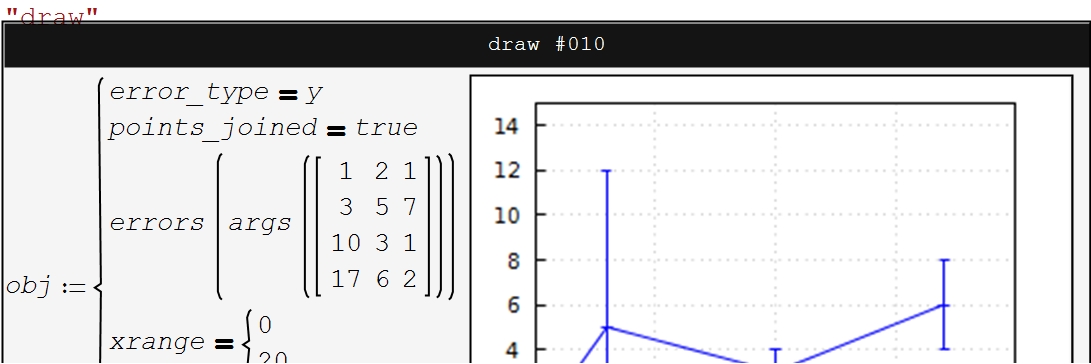
"draw"



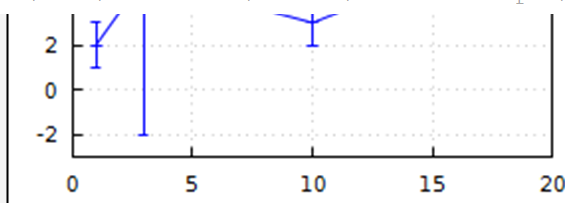
MaximaLog(■) = (%i28 (%o28



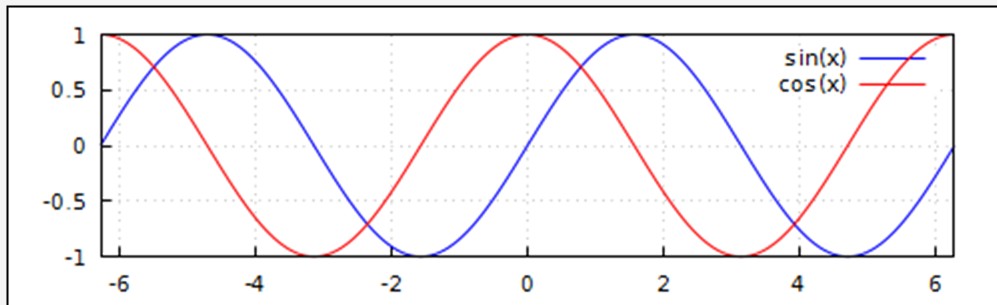
MaximaLo



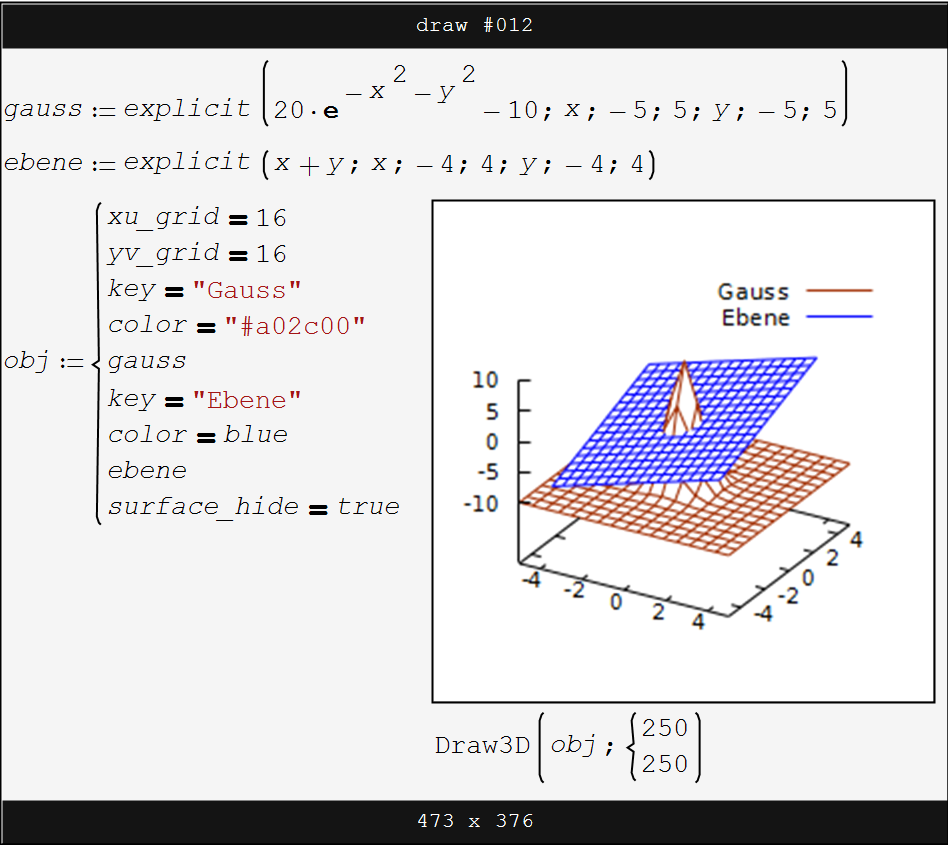
MaximaLog(■) = (%i30) draw (%o30) [gr2

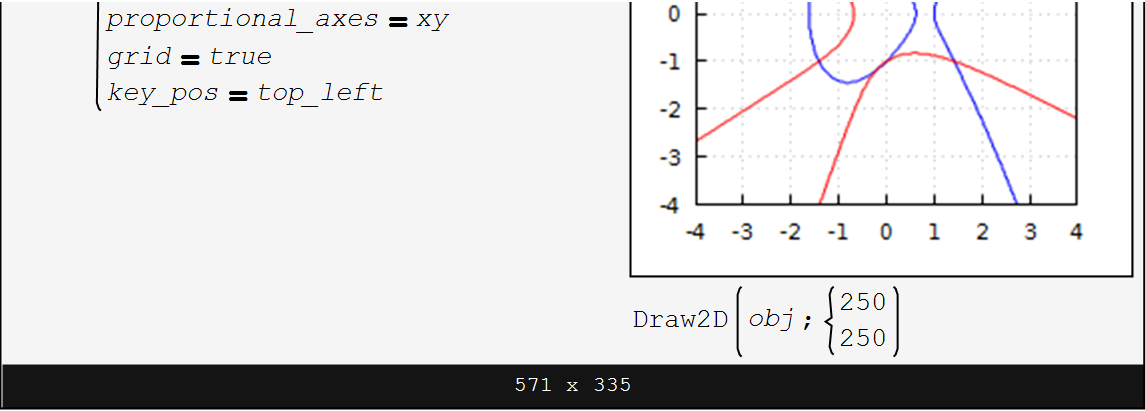

$$543 \times 267$$

```
draw #011
```

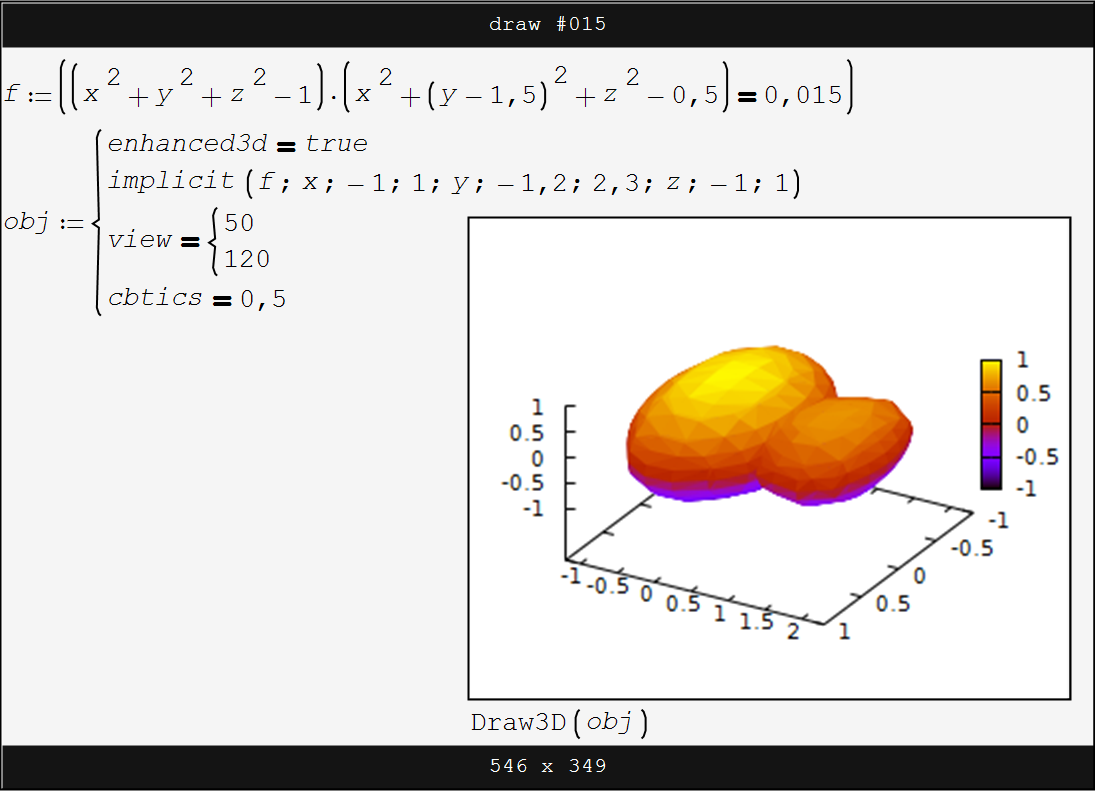
$$cmds := \begin{cases} key = "sin(x)" \\ explicit(\sin(x); x; -2 \cdot \pi; 2 \cdot \pi) \\ key = "cos(x)" \\ color = red \\ explicit(\cos(x); x; -2 \cdot \pi; 2 \cdot \pi) \\ grid = true \end{cases}$$

$$\text{Draw2D} \left(cmds ; \begin{Bmatrix} 500 \\ 150 \end{Bmatrix} \right)$$

```
"draw"
```

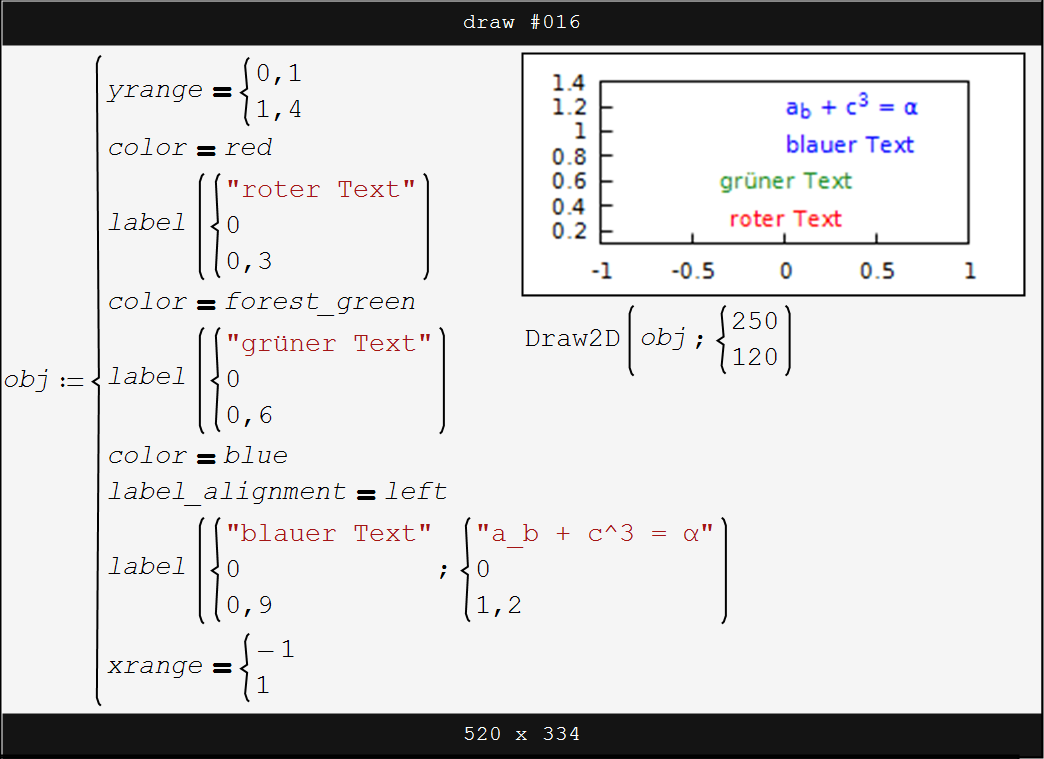




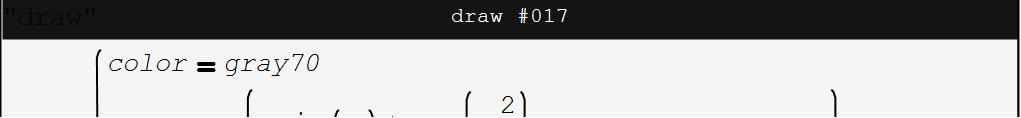
"draw"



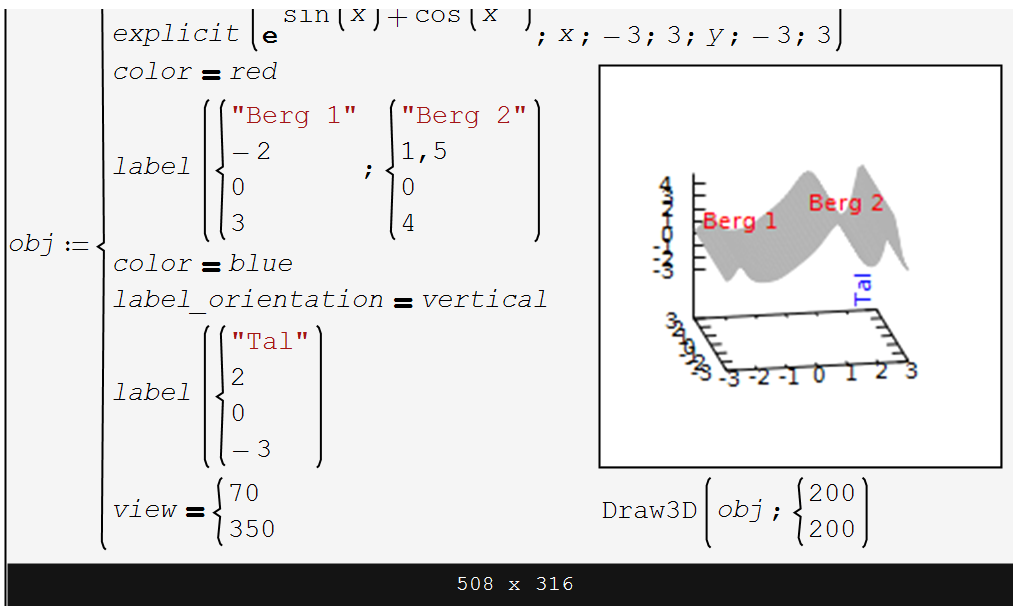
"draw"



"draw"

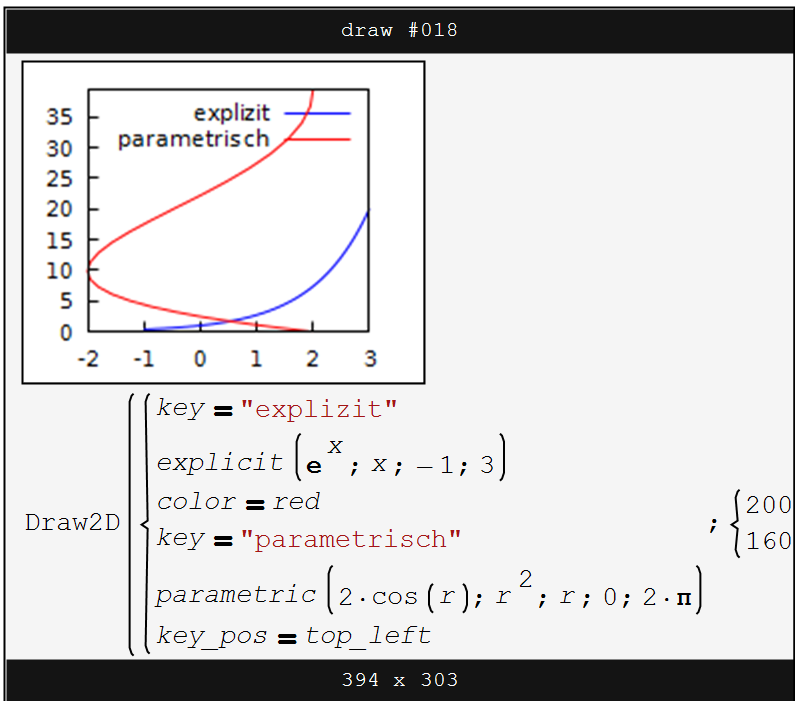


"Received B
MaximaLog(■) = (%i36) draw
(%o36) [gr2

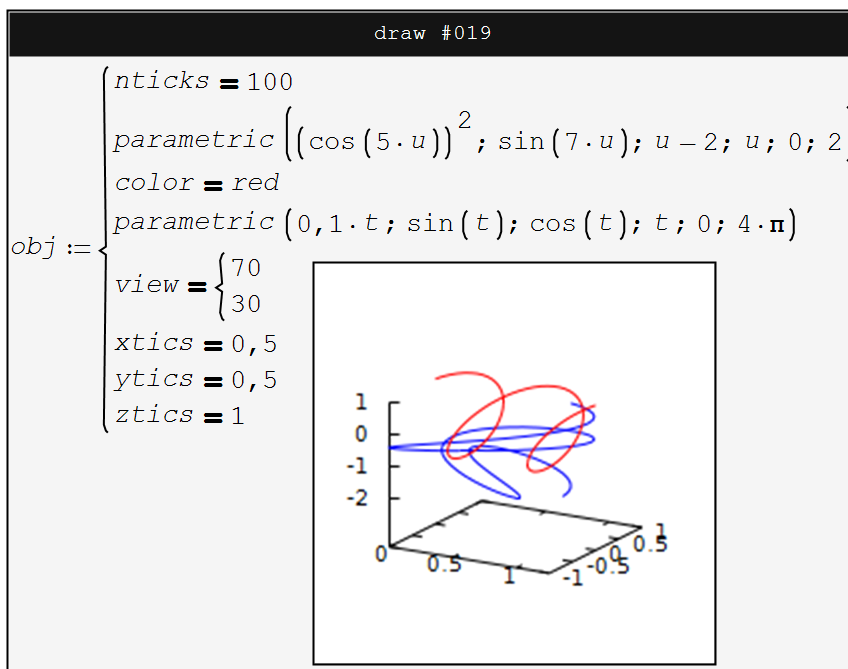


"draw"

☒ Draw-Descriptions (Maxima)



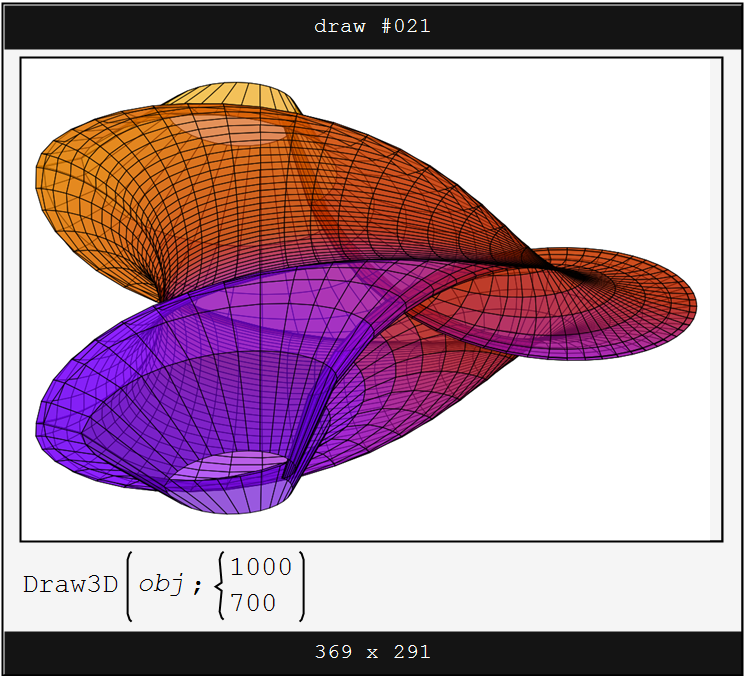
```
"draw"
```



```
Draw3D(obj; {200
424 x 351
```

```
"draw"
draw #020
{
  user_preamble = { "set style fill transparent solid 0.65 border"
                    "set view 120, 357, 1.75, 1.0"
  axis_3d = false
  colorbox = false
  enhanced3d = true
  xu_grid = 51
  yv_grid = 51
  xtics = none
  ytics = none
  ztics = none
  wired_surface = true
  color = black
  parametric_surface (fx; fy; fz; u; -4,5; 4,5; v; 0,05; pi-0,05)
}
a := 1
fx := (2*a*(cos(u)+u*sin(u))*sin(v))/(1+u^2*(sin(v))^2)
fy := (2*a*(sin(u)-u*cos(u))*sin(v))/(1+u^2*(sin(v))^2)
fz := a*ln(tan(v/2)) + (2*cos(v))/(1+u^2*(sin(v))^2)
560 x 248
```

"draw"



"draw"

```
draw #022
fx := u/2 * cos(u) * (cos(v) + 1)
fy := u/2 * sin(u) * (cos(v) + 1)
fz := u * sin(v) - ((u+3)/8 * pi)^2 - 20
```


draw #023

Draw3D(*obj*)

323 x 276

```

draw #024

{
  line_width = 2
  quadrilateral  $\left( \left\{ \begin{smallmatrix} 1 \\ 1 \end{smallmatrix} \right\}; \left\{ \begin{smallmatrix} 2 \\ 2 \end{smallmatrix} \right\}; \left\{ \begin{smallmatrix} 5 \\ 1 \end{smallmatrix} \right\}; \left\{ \begin{smallmatrix} 4 \\ 0 \end{smallmatrix} \right\} \right)$ 
  polygon  $\left( \left( \left( \begin{smallmatrix} 3 & 2 \\ 7 & 2 \\ 6 & 4 \\ 5 & 5 \\ 4 & 4 \end{smallmatrix} \right) \right) \right)$ 
  fill_color = yellow
  color = green
  rectangle  $\left( \left\{ \begin{smallmatrix} 2 \\ 3,5 \end{smallmatrix} \right\}; \left\{ \begin{smallmatrix} 5 \\ 2,5 \end{smallmatrix} \right\} \right)$ 
  color = black
  ellipse (3; 5; 2; 1; 0; 270)
  fill_color = light_blue
  ((1 (6 (7

```

```
triangle  $\left\{ \begin{pmatrix} 4 \\ 1 \end{pmatrix}; \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \begin{pmatrix} 1 \\ 4 \end{pmatrix} \right\}$ 
proportional_axes = xy
Draw2D  $\left( obj; \begin{pmatrix} 300 \\ 300 \end{pmatrix} \right)$ 
```

611 x 355

"draw"

draw #025

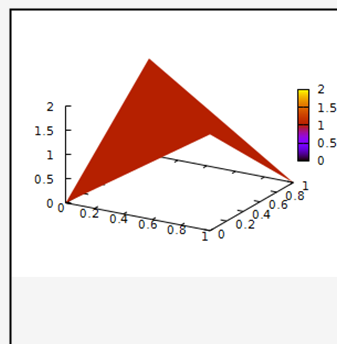
412 x 227

"draw"

draw #026

```
cm := { enhanced3d = true
        quadrilateral  $\left( \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}; \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}; \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right)$ 
        xyplane = 0
```

p := Draw3D (cm)



p

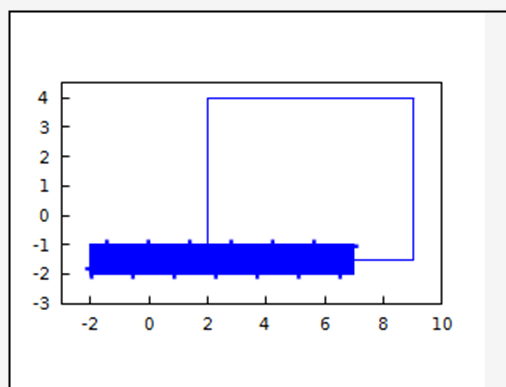
503 x 230

"draw"

draw #027

```
cm := { line_width = 6
        line_type = dots
        fill_color = blue
        rectangle  $\left( \begin{pmatrix} -2 \\ -2 \end{pmatrix}; \begin{pmatrix} 7 \\ -1 \end{pmatrix} \right)$ 
        transparent = true
        line_type = solid
        line_width = 1
        rectangle  $\left( \begin{pmatrix} 9 \\ 4 \end{pmatrix}; \begin{pmatrix} 2 \\ -1,5 \end{pmatrix} \right)$ 
        xrange =  $\begin{pmatrix} -3 \\ 10 \end{pmatrix}$ 
        yrange =  $\begin{pmatrix} -3 \\ 4,5 \end{pmatrix}$ 
        proportional_axes = xy
```

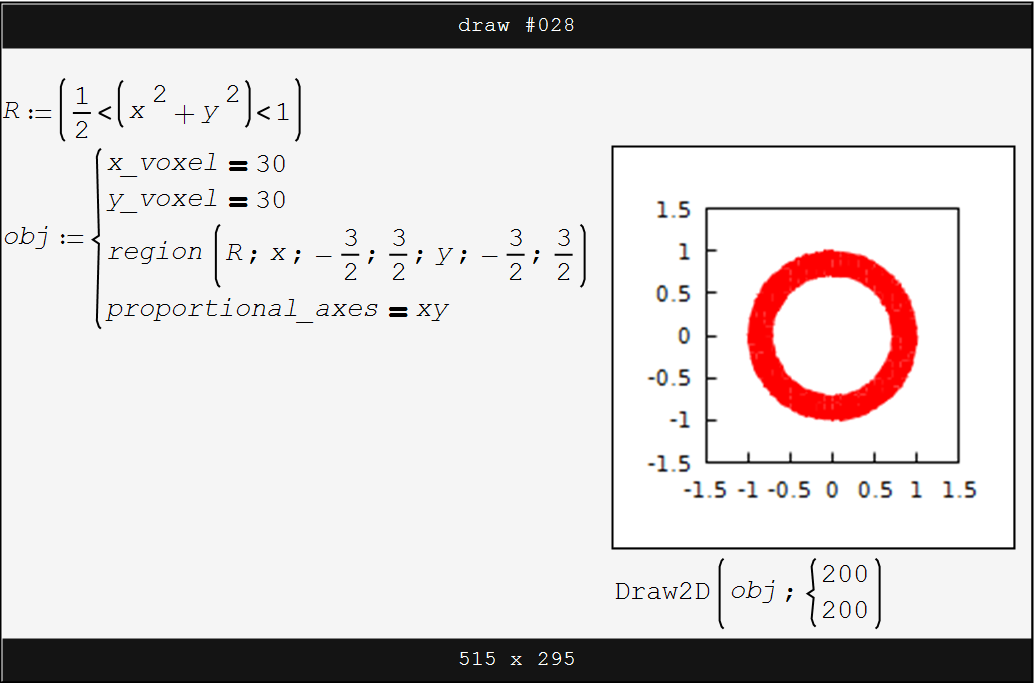
p := Draw2D (cm)



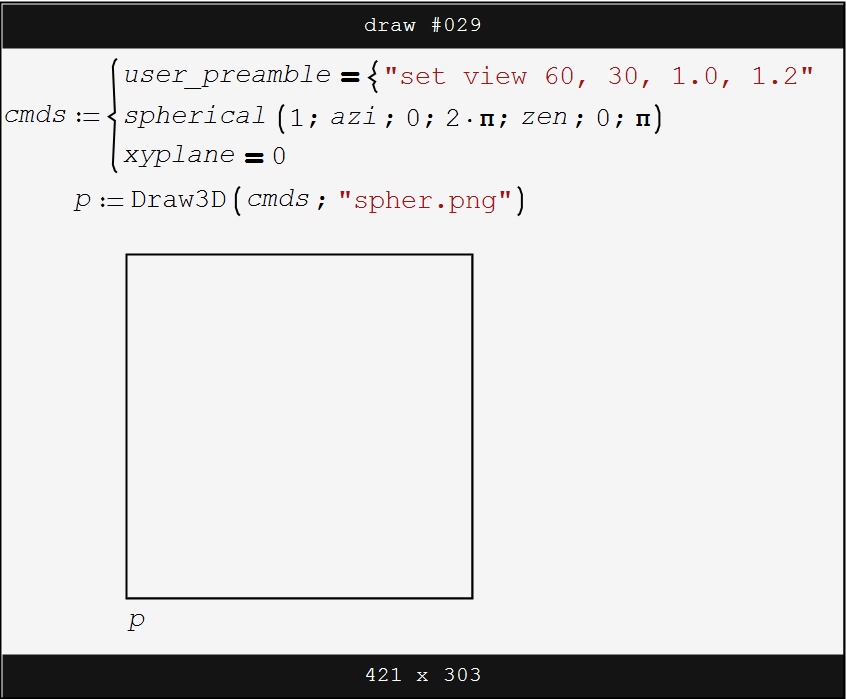
p

493 x 288

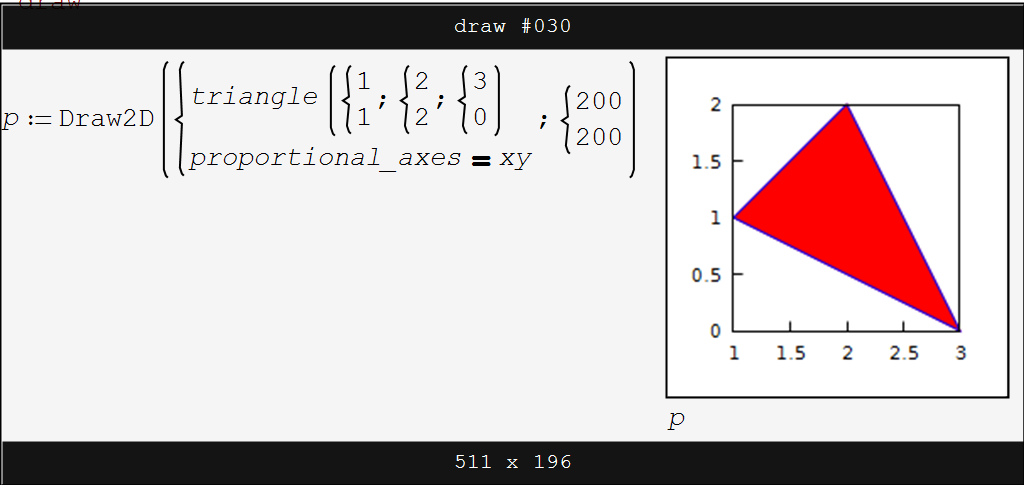
"draw"



"draw"



"draw"



"draw"

```

a := 2,5   b := 1,5   c := 2   d := 3

bottle := {
  color = blue
  tube (0; 0; (-a).sin(t); a + b.cos(t); t; 0;  $\frac{\pi}{2}$ )
  tube (0; 0; (-a).sin(t); a + b.cos(t); t;  $\frac{\pi}{2}$ ;  $\pi$ )
  color = red
  tube (0; 0; d.t; a + b.cos(t); t; 0;  $\pi$ )
  color = forest_green
  tube (c - c.cos(t); 0; d.t; a - b; t; 0;  $\pi$ )
  color = magenta
  tube (c - c.cos(t); 0; d. $\pi$  + c.sin(t); a - b; t; 0;  $\pi$ )
}

```

464 x 245

"draw"

