

Mathematical Computing

IMT2b2 β

Department of Mathematics
University of Ruhuna

A.W.L. Pubudu Thilan

Graphics in Maxima

Introduction

- ▶ **Gnuplot** is a command-line oriented plot program.
- ▶ Maxima's plotting is based on the **Gnuplot**, which allows for plotting of two and three dimensional functions, and datasets.
- ▶ Maxima's plotting commands collect the functions and parameters, and passes them all to **Gnuplot** for plotting.
- ▶ **Gnuplot** is bundled with Maxima (Windows version), so there is no need to install it separately.

Gnuplot Interfaces

- ▶ Two various **Gnuplot** interfaces are available.
- ▶ First one is the Maxima standard functions with the stem **plot** in the function names.
- ▶ Second one is the routines of the additional package **Draw** with the stem **draw** in the function names.

Gnuplot Interfaces

Graphic Interface Plot

- ▶ Displays a plot of one or more expressions as a function of one or more variables or parameters.
- ▶ **plot2d** displays one or several plots in two dimensions.
- ▶ **plot3d** displays a plot of one or more surfaces defined as functions of two variables or in parametric form.
- ▶ A plot can also be defined in the **discrete** or **parametric** forms.
- ▶ The abilities of Maximas **plot** command are a bit restricted and do not satisfy all needs.

Gnuplot Interfaces

Graphic Interface Draw

- ▶ Package **draw** provides a more powerful and flexible plotting and offer much more possibilities to adapt the graphics with the aid of options to particular requirements.
- ▶ Furthermore it is possible, to set output format (eps, png, jpg, etc.) and output target (i. e. the filename) in the **Gnuplot** console after the graphic has been produced.
- ▶ Before we can use the package it must be loaded using **load(draw)**.

Methods of Displaying the Graphics

Two various methods for displaying the graphics are possible.

1. Graph popping up in separate window.
2. Plotting into the wxMaxima working window.

Methods of Displaying the Graphics

Graph Popping Up in Separate Window

- ▶ When calling the standard plotting routines **plot2d**, **plot3d**, **draw2d**, **draw3d**, etc., a **Gnuplot** output window containing the graphic is popping up.
- ▶ The **Gnuplot** output window and consequently the graphic too can be resized using the mouse.
- ▶ Measurements can be performed in 2d-plots with the mouse, 3d-plots can be rotated in any arbitrary direction.

Methods of Displaying the Graphics

Plotting into the wxMaxima Working Window

- ▶ When preceding the letters **wx** to the names of the plotting routines (**wxplot2d**, **wxplot3d**, **wxdraw2d**, **wxdraw3d**,...), PNG-graphics are produced in screen resolution and placed directly into the wxMaxima working window.
- ▶ As the graphics remain visible during the entire Maxima session, that method is beneficial for interactive work.
- ▶ Right clicking the graphic enables copying into the clipboard or saving as a file.
- ▶ Nevertheless, due to its low resolution, further use of a graphic produced in that way is not reasonable.

Graphic Interface Plot

`plot2d (expr, xrange)`

- ▶ **expr** is an expression to be plotted on the vertical axis as a function of one variable.
- ▶ **xrange**, the range of the horizontal axis, is a list of the form **[variable, min, max]**, where variable is a variable which appears in **expr**.
- ▶ **plot2d (expr, xrange)** plots **expr** as a function of the variable named in **xrange**, over the range specified in **xrange**.
- ▶ Since the vertical range has not been specified, it will be chosen automatically.

Graphic Interface Plot

plot2d (expr, xrange) ⇒ Examples

(i) $y = x^2 + 4, -2 \leq x \leq 2$

(ii) $y = \sin(2\theta), -\pi \leq \theta \leq \pi$

(iii) $y = 2 \cos(2\theta), 0 \leq \theta \leq \pi$

(iv) $y = e^x, -2 \leq x \leq 2$

(v) $y = \ln x, -2 \leq x \leq 2$

(vi) $y = \cos(\theta) + \sin(4\theta), -\pi \leq \theta \leq \pi$

Graphic Interface Plot

`plot2d (expr, xrange,yrange)`

- ▶ **expr** is an expression to be plotted on the vertical axis as a function of one variable.
- ▶ **xrange**, the range of the horizontal axis, is a list of the form **[variable, min, max]**, where variable is a variable which appears in **expr**.
- ▶ **yrange**, the range of the vertical axis, is a list of the form **[y, min, max]**.
- ▶ **plot2d (expr, xrange,yrange)** plots **expr** as a function of the variable named in **xrange**, over the range specified in **xrange**.
- ▶ The vertical range is set to **yrange**.

Graphic Interface Plot

plot2d (expr, xrange,yrange)⇒Examples

(i) $y = x^2 + 4, -2 \leq x \leq 2, 0 \leq y \leq 10$

(ii) $y = \cos(2\theta), -\pi \leq \theta \leq \pi, 0 \leq y \leq 1$

(iii) $y = 2e^x, -2 \leq x \leq 2, 0 \leq y \leq 5$

(iv) $y = \frac{1}{\ln(2x)}, -2 \leq x \leq 2, -1 \leq y \leq 1$

Graphic Interface Plot

```
plot2d([expr_1, ..., expr_n], xrange)
```

- ▶ **plot2d([expr_1, ..., expr_n], xrange)** plots **expr_1, ..., expr_n** as a function of the variable named in **xrange**, over the range specified in **xrange**.
- ▶ Since the vertical range has not been specified, it will be chosen automatically.

Graphic Interface Plot

`plot2d([expr_1, ..., expr_n], xrange)⇒Examples`

(i) $y = \sin(x), y = \cos(x), 0 \leq x \leq 2\pi$

(ii) $y = \sin(x), y = \sin(2x), 0 \leq x \leq 2\pi$

(iii) $y = 8 \sin(x), y = 4 \cos(x), 0 \leq x \leq 2\pi$

(iv) $y = 2x^2, y = \cos(x), -\pi \leq x \leq \pi$

Graphic Interface Plot

```
plot2d([expr_1, ..., expr_n], xrange,yrange)
```

- ▶ `plot2d([expr_1, ..., expr_n], xrange,yrange)` plots `expr_1`, ..., `expr_n` as a function of the variable named in `xrange`, over the range specified in `xrange`.
- ▶ The vertical range is set to `yrange`.

Graphic Interface Plot

`plot2d([expr_1, ..., expr_n], xrange,yrange)⇒Examples`

(i) $y = \sin(x), y = \cos(x), 0 \leq x \leq 2\pi, 0 \leq y \leq 1$

(ii) $y = \sin(x), y = \sin(2x), 0 \leq x \leq 2\pi, 0 \leq y \leq 1$

(iii) $y = 8 \sin(x), y = 4 \cos(x), 0 \leq x \leq 2\pi, 0 \leq y \leq 8$

(iv) $y = 2x^2, y = \cos(x), -\pi \leq x \leq \pi, 0 \leq y \leq 5$

Graphic Interface Plot

```
plot2d([discrete,xvalues,yvalues])
```

- ▶ Plots the points with declaration of x-values and y-values in two distinct lists **xvalues** and **yvalues**.
- ▶ The points are connected by line segments by default.

Eg:

```
xvalues:[0,3,6,4,6,3,0,2,0];⇒ Declaration of x-values.
```

```
yvalues:[0,2,0,3,6,4,6,3,0];⇒ Declaration of y-values.
```

```
plot2d([discrete,xvalues,yvalues]);
```

Graphic Interface Plot

`plot2d([discrete,pairs])`

- ▶ Plots the points given as pairs in a nested list `[[x1,y1],[x2,y2],...]`.
- ▶ The points are connected by line segments by default.

Eg:

```
plot2d([discrete,[[3,1],[5,3],[3,5],[1,3],[3,1]]]);
```

Graphic Interface Plot

```
plot2d([parametric,x(t),y(t),trange],opts)
```

- ▶ In a parametric plot, you give both the **x** and **y** coordinates of each point as a function of a third parameter, say **t**.
- ▶ The command plots a parametric curve with the parameter **t** within the range **trange**.
- ▶ In order to achieve a smooth curve, the number of initial points has to be increased using the option **nticks**.

Eg:

```
wxplot2d([parametric,sin(t),sin(2*t),[t,0,2*%pi]], [nticks,100]);
```

Graphic Interface Plot

`plot2d([parametric,x(t),y(t),trange],opts)`

Plot the following parametric curves.

(i) $x = 3 \sin(t), y = 4 \cos(t), 0 \leq t \leq 2\pi$

(ii) $x = \sin(t), y = \sin(2t), 0 \leq t \leq 2\pi$

(iii) $x = 8 \sin(t), y = 4t \cos(t), 0 \leq t \leq 2\pi$

(iv) $x = 2t^2, y = \cos(t), -\pi \leq t \leq \pi$

Graphic Interface Plot

3d Plot

- ▶ The Graphic Interface **Plot** offers only very limited possibilities for creating 3d-plots.
- ▶ The use of the Graphic Interface **Draw** is recommended for the creation of 3d-graphics.
- ▶ The Graphic Interface **Plot** can produced two kinds of 3d-object.

Graphic Interface Plot

3d Plot \Rightarrow First Type \Rightarrow `plot3d(f(x,y), xrange, yrange)`

- ▶ $z = f(x, y)$; is the function that is to be plotted in 3d space.
- ▶ The ranges for x -values and y -values have to be declared as **xrange** and **yrange** respectively.
- ▶ However, the z -range cannot be declared.

Graphic Interface Plot

3d Plot \Rightarrow First Type \Rightarrow `plot3d(f(x,y), xrange, yrange)` \Rightarrow Examples

(i) $f(x, y) = 1/(1 + x^2 + y^2), -3 \leq x \leq 3, -3 \leq y \leq 3$

(ii) $f(x, y) = \sin(x^2 + y^2), -5 \leq x \leq 5, -5 \leq y \leq 5$

(iii) $f(x, y) = x + y + \sin(xy), -10 \leq x \leq 10, -10 \leq y \leq 10$

(iv) $f(x, y) = x^2 - 2y^2 + \cos(xy), -10 \leq x \leq 10, -10 \leq y \leq 10$

Graphic Interface Plot

3d Plot \Rightarrow Second Type \Rightarrow `plot3d([x(u, v),y(u, v),z(u, v)],urange,vrange)`

- ▶ 3d-plot of a surface in parametric form can be plotted.
- ▶ An object in parametric form $\mathbf{x}(\mathbf{u}, \mathbf{v})$, $\mathbf{y}(\mathbf{u}, \mathbf{v})$, $\mathbf{z}(\mathbf{u}, \mathbf{v})$ in dependence of two parameters \mathbf{u} and \mathbf{v} has to be considered.
- ▶ The ranges of the parameters \mathbf{u} and \mathbf{v} have to be declared as **urange** and **vrangle** respectively.
- ▶ The ranges of the coordinates \mathbf{x} , \mathbf{y} and \mathbf{z} cannot be declared.

Graphic Interface Plot

3d Plot \Rightarrow Second Type \Rightarrow `plot3d([x(u, v),y(u, v),z(u, v)],urange,vrange)` \Rightarrow Examples

Obtain the surfaces of the following parametric equations.

(i) $x = \cos(u), y = v, z = v^2 + \sin(u), -3 \leq u \leq 3, -3 \leq v \leq 3$

(ii) $x = t, y = u, z = \sin(tu), 0 \leq t \leq 3, 0 \leq u \leq 3$

(iii) $f = \cos(x), g = y, h = y^2 + \sin(x), 0 \leq x \leq 2\pi, 0 \leq y \leq 1$

Graphic Interface Plot

Contour plot \Rightarrow `contour_plot(f(x,y), xrange, yrange)`

- ▶ A **contour plot** is a graphical technique for representing a 3-dimensional surface by plotting constant z slices, called contours, on a 2-dimensional format.
- ▶ That is, given a value for z , lines are drawn for connecting the (x, y) coordinates where that z value occurs.
- ▶ **contour_plot** draws contour lines of a function in two variables within the ranges **xrange** and **yrange**.

Graphic Interface Plot

Contour plot \Rightarrow `contour_plot(f(x,y), xrange, yrange)` \Rightarrow Examples

Obtain the contour plot of the following parametric equations.

(i) $f(x, y) = 1/(1 + x^2 + y^2), -3 \leq x \leq 3, -3 \leq y \leq 3$

(ii) $f(x, y) = \sin(x^2 + y^2), -5 \leq x \leq 5, -5 \leq y \leq 5$

(iii) $x = \cos(u), y = v, z = v^2 + \sin(u), -3 \leq u \leq 3, -3 \leq v \leq 3$

(iv) $x = t, y = u, z = \sin(tu), 0 \leq t \leq 3, 0 \leq u \leq 3$

Graphic Interface Plot

Options

- ▶ Additional optional parameters allow adapting a graphic to particular requirements with respect to colors, line types, sizes, labels, output formats etc.
- ▶ The options are lists (mostly contains two elements).
- ▶ The first element is always the name of the option, additional elements are the associated values.

Graphic Interface Plot

Options⇒Cont...

- ▶ Options can be stated as additional parameters in every plot command.
- ▶ They can also be declared as default values using the command **set_plot_option** as shown below.

set_plot_option([**name**, **v**]) ⇒ Assigns the option **name** to the value **v**.

- ▶ In that case they are valid for all subsequent plot commands.
- ▶ The command **plot_options** shows all default values of the options.

Graphic Interface Plot

Options⇒Commands

[xlabel, "text"] ⇒ x-axis label in 2d-plots.

[ylabel, "text"] ⇒ y-axis label in 2d-plots.

[legend, "text1", "text2", ...] ⇒ Legends for particular curves in a 2d-Plot.

[gnuplot_term, terminal] ⇒ Output format; ignored by the wx-routines

[gnuplot_out_file, "filename"] ⇒ Output target; ignored by the wx-routines.

Graphic Interface Plot

Options⇒Commands⇒Example

Plot $y = x^2$ and $y = x^2 + 4$ in the range of $-2 \leq x \leq 2$,
 $0 \leq y \leq 10$.

- (i) Name x axis as **Independent variable**.
- (ii) Name y axis as **Dependent variable**.
- (iii) By using a legend name two curves as **First** and **Second**.
- (iv) Save the plot in gif format with the name **Graph**.

Graphic Interface Plot

Options⇒Style command

- ▶ The option **style** assigns the type of the curve, line width and line color (and the point type, if desired).
- ▶ In pixel graphics the line width is given in pixels.
- ▶ In vector graphics as multiples of 0.25pt (about 0.088mm).

Graphic Interface Plot

Options⇒Style command⇒Cont...

The curve type can obtain the following values:

lines ⇒ solid line

points ⇒ points

linespoints ⇒ solid line and points

impulses ⇒ bars (line widths and colors are ignored)

Graphic Interface Plot

Options⇒Style command⇒Cont...

Colors can have the following values:

1 ⇒ blue

2 ⇒ red

3 ⇒ magenta

4 ⇒ yellow

5 ⇒ brown

6 ⇒ green

7 ⇒ cyan

Graphic Interface Plot

Options⇒Style command⇒Example

Plot following data in a scatter plot.

x_i	y_i
23.1	10.5
32.8	16.7
31.8	18.2
32.0	17.0
30.4	16.3
24.0	10.5
39.5	23.1
24.2	12.4
52.5	24.9
37.9	22.8

Thank You