The **interactiveplot** Package

Roberto Bock, Javier Toro, Pedro Linares

Saturday 20$^{\text{th}}$ September, 2014

Abstract

A LaTeX package for creating multi-parametric explicit 2D and 3D plots that can be interactively controlled by buttons inside a PDF.

Keywords: build button controllable interactive plot parameter LaTeX pdfLaTeX JavaScript Adobe Reader

Contents

1 Introduction

This package provides an interface to create Portable Document Format (PDF) files with 2D and 3D elemental explicit functions, that allow variable parameters as arguments. These parameters can be modified interactively by buttons.

The final PDF can be viewed with Adobe Reader 11.0 or above.

2 Requirements

pdfTeX, version $\geq$ 1.20

Adobe Reader, version $\geq$ 11.0

3 Installation

Unzip the file 'interactiveplot.zip' and copy the 'interactiveplot.sty' in a place where LaTeX can find it.

MiKTeX users should run the package manager for installation.

4 Using the package

Invoke the package by putting the line

```
\usepackage{interactiveplot}
```

in the preamble of your document, i. e. somewhere between `\documentclass` and `\begin{document}`.

5 The user interface

Interactive plots are defined using three basic instructions provided by this package. There are two kind of graphs allowed by this package: 2D and 3D. Once you have defined the type of graph you wish to plot you have to set up the functions you would like to display.

- A 2D Graph is inserted using the environment `iplotdd`.
- A 3D Graph is inserted using the environment `iplotddd`.
- Functions are defined using the macro `iplot` .

5.1 Inserting a 2D Graph

```
\begin{iplotdd}   %  Environment that defines as 2D
                  %  the kind of function to be plotted
\iplot[options]   %  Macro that defines the first function
                  %  and its options
{<2d_function1>}  %  function in natural language
...
\iplot[optionsN]  %  Last function to be plotted
{<2d_functionN>}
\end{iplotdd}     %  End of environment
```

5.2 Inserting a 3D Graph

```
\begin{iplotddd}  %  Environment that defines as 3D
                  %  the kind of function to be plotted
\iplot[options]   %  Macro that defines the first function
                  %  and its options
{<3d_function1>}  %  function in natural language
...
```

```
\iplot[optionsN]  %  Last function to be plotted
{<3d_functionN>}
\end{iplotddd}     %  End of environment
```

5.3 The `\iplotdd` and `\iplotddd` options.

The environments `\iplotdd` and `\iplotddd` share the same options. These are `<width>` and `<height>`.

The `<width>` and `<height>` arguments must be given in valid TEX dimensions. Horizontal and vertical dimensions of the graph are scaled independently to fit `<width>` and `<height>`.

5.4 The `\iplot` options.

The macro `\iplot` is used to define the function to be graphed. This macro can be used inside an `\iplotdd` or `\iplotddd` environment. If you would like to produce a 2D graph (by using the macro inside the environment `\iplotdd`) you have to define only 1 <variable> option (explained below). If you would like to use inside a 3D graph (using the environment `\iplotddd`) you have to define 2 <variable> options for the `\iplot` macro.

The macro `\iplot` admits the following options:

- `<var={<variable_name>[,start[,end]]}>`: this option defines the property of a variable. It has 3 entries:
  - `<variable_name>`: it defines the name of the variable. This value is mandatory.
  - `[variable_start]`: it defines the inferior graphing limit; that is, it sets the starting value for plotting of the variable. This value is optional, if it is not defined a default value of -25 is set for the 2D case and of -1 for the 3D case.
  - `[variable_end]`: it defines the superior graphing limit; that is, it sets the ending value for plotting of the variable. This value is optional, if it is not defined a default value of 25 is set for the 2D case and of 1 for the 3D case.

  If you set a variable option as var={ X,-1,1 } you will get a graph plotted in the interval (-1,1).

- `[param={<parameter_name>,<value>,<step>,<start>,<end>}]`: this option defines a function parameter property. It has 4 entries values:

  - `<parameter_name>`: it defines the name or label of the parameter.
  - `<value>`: it defines the initial value of the parameter. It is used to make the first display of the graph.
  - `<step>`: it defines a positive constant value used to increase or decrease the value of the parameter.
  - `<start>`: it defines the minimum value that the parameter can reach.
  - `<end>`: it defines the maximum value that the parameter can reach.

  When this option is specified, it insert 2 buttons in the PDF which can be used to increase or decrease the `<value>` by the `<step>` value.

  If no parameters are given, the resulting plot will be static.

- `[frange={<frange>[,start[,end]]}]`: this option specifies the plottable range in the Y axis in 2D case or the Z axis in 3D case.
  - `<frange_name>`: it specifies the name of the function range. This value is mandatory.
  - `[start]`: it specifies the minimal plottable value.
  - `[end]`: it specifies the maximal plottable value.
- `[color=<color>]`: this option defines the color of the current function graph. The colors that you can select are:
  - `black`.
  - `white`.
  - `red`.
  - `green`.
  - `blue`.
  - `cyan`.
  - `magenta`.
  - `yellow`.
  - `dark-gray`.
  - `gray`.
  - `light-gray`.

If the color option is not specified, the default value is black.

- `[resolution=<resolution>]`: it defines the number of points used to build the graph of the function. If not specified a value of 500 or 30 will be assumed for 2D and 3D respectively.

When you use the `\iplot` macro inside a `\iplotdd` environment the `\iplot` macro builds a 2D graph. In this case, you have to define one <var> option. When you use the `\iplot` macro inside a `\iplotddd` environment the `\iplot` macro builds a 3D graph. In this case, you have to define two <var> options. The *var* option is not optional.

5.5 The `\iplot` arguments.

The `\iplot` macro has as argument the function itself. The entire function can be composed by another functions. In this section all the admissible functions will be described:

Function Name	**Normal**	**Inverse**	**Hyperbolic**	**Inverse H**
sine function	**sin**	**asin**	**sinh**	**asinh**
cosine function	**cos**	**acos**	**cosh**	**acosh**
tangent function	**tan**	**atan**	**tanh**	**atanh**
cotangent function	**cot**	**acot**	**coth**	**acoth**
secant function	**sec**	**asec**	**sech**	**asech**
cosecant function	**csc**	**acsc**	**csch**	**acsch**

Non trigonometric functions

Function Name	**Syntax**
exponential function	**exp**
square root function	**sqrt**
neperian logarithm function	**ln**
logarithm function	**log**
Power function	**^**

6 Examples

In this section, two examples will be shown. The first one will show how the package can be used to display a 2D graph. The second one will show how the package can be used to display a 3D graph.

6.1 Graphing the 2D function: `f(x)=(x^2+p)/q+t`

This example displays the 2D function f(x)=(xˆ2+p)/q+t.

```
\begin{iplotdd}[width=200,height=200]
      \iplot[var={x,-5,5},param={p,-1,1,-10,10}
            ,param={q,0,1,-10,10}
            ,param={t,0,1,-10,10}
            ,color=black]
            {(x^2+p)/q+t}
\end{iplotdd}
```

6.2 Graphing the 3D function: `f(x)=x^*2t+y^2/q+p`

This example displays the 2D function `f(x)=x^2*t+y^2/q+p`.

```
\begin{iplotddd}[width=200,height=200]
      \iplot[
            var={x,-5,5}
            ,var={y,-5,5}
            ,param={p,1,1,-10,10}
            ,param={q,1,1,-10,10}
```

```
          ,param={t,1,1,-3,7}
          ,frange={z,-5,5}
          ,color=black]
          {x^2*t+y^2/q+p}
\end{iplotddd}
```

7 Known issues

This section shows the issues you may experience during the use of this package.

7.1 2D Issues

When you graph a 2D function, if undefined segments between the specified graphing limits exist, the package will draw a connecting line over the undefined segment. For instance, the function `arcsec(x)` is undefined in interval (-1,1) but a line connecting the inferior limit value (-1) with the superior one (1) is drawn.

7.2 3D Issues

If the `<resolution>` for a 3D function is low the package could draw connecting lines between two points where no connecting lines should exist. For instance, with function `-4*x^2+y^2+1` if you define a resolution of 30 you will get the following result:

In this case, you can solve this problem by increasing the `<resolution>`, but the speed at which the plot is displayed could be affected.