

Graph Definition Syntax – Developer’s Reference

Updated 24 November 2012

Overview

- Graphs can be either dynamic or static
- Dynamic graphs can have one or more *functions* and up to six *sliders*. Functions can contain variables that are replaced by dynamic values from sliders.
- Display includes the function, in LaTeX form, rendered by MathJax. By default, the equation is displayed at the top left of the window. For graphs with more than one function, the function definition can specify the coordinates where each equation is displayed.
- The function definition is in standard infix: e.g. $y=3x^2$.
- A parser translates the infix into: a) an expression that can be evaluated by javascript to create the graph display; b) LaTeX for the equation that will be displayed.
- The function definition can include fields to override the parser.
- The graph scale can be in decimal or pi unites

Graph Definition

- Sequence of lines terminated by semicolons or newlines
- Each line begins with one of five keywords: **function**, **slider**, **window**, **points**, or **control bar**.
- An empty definition will display the default window with no graph
- Each line in a definition can contain several fields; fields are separated by colons (:)

Function Definition

- The function definition defines the function that is graphed.
- **function:** $y <op> <infix\ exp>[:**eval** = <eval\ exp>][:**latex** = <latex\ exp>] [:**dynamic**]$
[:**position**=###,###] [:**color** = <color\ def>][:**dashed**]
- $<op>$: usually the = sign; but can be =, <, <=, >, >=
- $<infix\ exp>$: standard math infix. Example: $y=\%<var>x^2+\%<var>$.
- $\%<var>$: variable replaced by a value from a slider
- $<eval\ exp>$: overrides the parser. This expression is passed to javascript eval. Example:
 $\%a*\text{Math.pow}(x, 2) + \%k$
- $<latex\ exp>$: overrides the parser. This expression is passed to MathJax.
- **dynamic**: dynamic equation. Variables in the equation are replaced by slider values.
- **position**: display position for the equation (x,y) relative to the top left corner of the window.
- **color**: color of the graphed function. Can be standard color, or any hex value. $<color\ def>$: [orange | blue | purple | green | pink | brown | #nnnnnn]
- **dashed**: the graphed line is dashes instead of a solid line

Slider Definitions

- A slider definition defines a slider control. The var field is required. Default values for other fields are: min=-10; max = 10; ival = 0; step = 1.
- **Slider:** $\text{var}=<variable>[:**min** = <num>][:**max** = <num>][:**ival** = <num>][:**step** = <num>] >][:**color** = <color\ def>]$

- `<variable>` Single letter defines the variable. The slider replaces this variable in all functions.
- `<num>` integer or decimal number. Must start with a number (e.g. 0.1)
- `<step>` This field defines the precision of the numeric display as well as the step increment for the slider.

Window definition

- The window definition defines the coordinate system for the graph. This is not required. All fields are optional. Defaults are: `xmin = -10; ymin = -10; xmax = 10; ymax = 10; scale = decimal; grid = yes; labels = xy`
- **Window**`[:xmin=<num>] [:xmax=<num>] [:ymin=<num>] [:ymax=<num>] [:grid = yes | no] [:scale = none | decimal | pi] [:labels = <var><var>]`

Points definition

- The points definition defines points that are plotted on the graph.
- **points**`:(x1,y1),(x2,y2),...`

Control Bar definition

- This allows the developer to change the height or layout of the control bar that is shown in gray at the bottom of the graph window
- **Control bar**`[:height=##] [:divider = ##] [:slider width = ##]`