

[4.9] Faster plots of slow functions

The built-in 89/92+ function plotter can be very slow to plot complicated or time-consuming user-functions when the ZoomFit menu item is used. It seems that the plotter evaluates the function twice at each plot point, perhaps to determine the window limit system variables. Plotting time can be substantially reduced by graphing the function as a data plot instead of a function graph. This program does the plotting:

```
plotxy(fname,xl,xh,pn,res)
Prgm
@("f(x)",xlow,xhigh,plot#,xres) plot function using data plot
@leaves global variables xx and yy
©1jul00/dburkett@infinet.com

local dxx,k

©Find x-step and make x-, y-lists
©Change all 239 to 159 for TI89
res*((xh-xl)/239)->dxx
seq((k-1)*dxx+xl,k,1,ceiling(239/res))->xx
seq(expr(fname)|x=xx[k],k,1,ceiling(239/res))->yy

©Set graph window limits
xl->xmin
xh->xmax
min(yy)->ymin
max(yy)->ymax

©Create plot; display graph
fnoff
plotsoff
newplot pn,2,xx,yy,,,5
dispg

EndPrgm
```

The function arguments are:

<i>fname</i>	Name of function to be plotted, as a string, with an independent variable of <i>x</i>
<i>xl</i>	Lower limit for independent variable <i>x</i>
<i>xh</i>	Upper limit for independent variable <i>x</i>
<i>pn</i>	Number of data plot to use (1 to 9)
<i>res</i>	Plot resolution. Set to 1 to plot every pixel, 2 for every other pixel, etc.

The call to plot the function $\sin(x)$, from 0 to 12, at every pixel, using plot number 2, is

```
plotxy("si(x)",0,12,2,1)
```

After executing this call, the graph is displayed. You may perform all the usual operations on data graphs.

This program has these limitations: You can only plot a single function. The program will over-write current data plot definitions; use the *pn* parameter to specify which data plot to use to avoid this. For simple functions, the built-in function plotter is faster. You must pass the function name as a string, and the independent variable must be *x*. Since this program writes to the graph screen, it must be a program, not a function. The arguments to the *newplot* command must be global variables, not local variables, so these will still exist in the folder from which *plotxy()* is run. You must delete them manually. Since *plotxy()* makes a data plot, not a function plot, you cannot perform Math menu operations such as Zero, Minimum, Maximum and so on.

For even faster plotting, set *res* to 2 or 3 or even larger.

These timing results (92+, HW2, AMS 2.04) show the advantages of using *plotxy()*. As an example I use a user-function called *si(x)*, which calculates the sine integral. I plotted over a range of $x=0$ to $x=12$, with a resolution of 1.

Built-in function plotter, using ZoomFit:	226 seconds
<i>plotxy()</i> :	144 seconds (saves 82 seconds or 36%)
Built-in function plotter, using manual limits:	114 seconds

The fastest method is to use the built-in plotter, and set the *ymin* and *ymax* limits manually.