free42: Programs For Functions

Mitch Richling

2021-03-19

Author: Mitch Richling Updated: 2021-05-18 21:09:45

Copyright 2021 Mitch Richling. All rights reserved.

Contents 1 Metadata

-		-
2	Introduction 2.1 Functionality & the Main Menu 2.2 Notes Regarding Some Individual Programs 2.2.1 Interactive vs Non-interactive 2.2.2 FVARM: Global Variables 2.2.3 FEVL: Evaluating the current equation 2.2.4 PICF: Selecting a MVAR program 2.2.5 SFWRP: Wrap a simple function 2.2.6 FSLV & FINT: Apply the built in SOLVE & INTG tools to the current EqP 2.2.7 FPRD & FSUM: Non-interactive Sums & Products 2.3 Organizing Principles	2 2 2 2 2 2
	FMENU: Main menu for all apps 3.1 Menu Code	3 3 3
	5.1.1 Main Menu	12 12
6	Test Functions	24
7	EOF	26

1 Metadata

The home for this HTML file is: https://richmit.github.io/hp42/pgmforfun.html A PDF version of this file may be found here: https://richmit.github.io/hp42/pgmforfun.pdf

Files related to this document may be found on github: https://github.com/richmit/hp42 Directory contents:

rc - The org-mode file that generated this HTML document Ready to convert source listings for 42s code in this document

docs - This html document and associated PDF

bin - Importable RAW program files

2 Introduction

Plots Sums
$$\sum_{n=1}^{15} \frac{1}{n^2}$$
 Products $\prod_{n=1}^{15} \frac{1}{n^2}$ Integrals $\int_1^{15} \frac{1}{x^2} dx$ Roots

This is a collection of programs that work with MVAR-type programs (equations). I was inspired to do this when the PGMENU function was added to free 42.

2.1 Functionality & the Main Menu

Please ignore the first two columns in the table – they are used to auto-generate the menu code for the application.

MSTR	LBL	Menu	Description
SPT	FSPT	SPT	Interactive application for computing Sums, Products, and Tables
LBL 99	FPLT	PLOT	Interactive application for plotting a function (DM42 Specific)
EVAL	LBL 91	EVAL	Evaluate EqV on X
FVARM	LBL 95	FVARM	MVAR program with all the global variables for these apps
PICF	LBL 96	PICF	Interactive application to select and describe an MVAR program
SOLVE	FSLV	SOLVE	Solve EqP for EqV in [LLIM, ULIM]
INTG	FINT	INTG	Integrate EqP for EqV over [LLIM, ULIM]
SUM	FSUM	SUM	Sum EqP for values of EqV from LLIM to ULIM stepping by STEP
PROD	FPRD	PROD	Product EqP for values of EqV from LLIM to ULIM stepping by STEP
FVARM	LBL 95	FVARM	MVAR program with all the global variables for these apps
PICF	LBL 96	PICF	Interactive application to select and describe an MVAR program

2.2 Notes Regarding Some Individual Programs

2.2.1 Interactive vs Non-interactive

SPT & FPLT are fully interactive, user friendly applications. The rest are stripped down programs designed to offer performance and good programmatic usability – they are decidedly unfriendly. Except for FEVL (menu key [EVAL]) these other programs provide no substantive error checking, they just do what they do as fast as they can do it. This can lead to rather cryptic error messages when they are incorrectly used. This is a feature.

2.2.2 FVARM: Global Variables

While FVARM is simply an MVAR program with all the global variables, the [FVARM] menu key actually shows the variable menu with VARMENU.

2.2.3 FEVL: Evaluating the current equation

I frequently use VARMENU to repeatedly evaluate a formula, with different input values. Functionally that means typing in the value, and then hitting the variable button twice. That's not super efficient when repeatedly changing the same variable. That's where this little function comes in!

Note the program FEVL and the menu key [EVAL] are different. The menu will do some error checking before it runs while the program just attempts to evaluate EqP on X.

2.2.4 PICF: Selecting a MVAR program

This little function is at the center of all the tools on this page. It queries for an MVAR program, lets you set constants, pick an independent variable, and describe the number of returns from the function. See the section Organizing Principles for more about the global variables used to house this data.

2.2.5 SFWRP: Wrap a simple function

This MVAR program allows one to wrap a simple function for use with tools that need an MVAR program. For example, if you have a program for SECH you can use this program to wrap SECH allowing you to use the built in solver & integrator or programs like FPLT.

To use this program simply put the name of the simple function into the variable WFUN. Note you can do that in the FUN button on the last page of the main menu.

2.2.6 FSLV & FINT: Apply the built in SOLVE & INTG tools to the current EqP

These are handy little functions that allow one to apply the built in solver and integration capabilities of the 42s to the program currently active function. I debated putting them in the menu because they are really more suited to programmatic use.

2.2.7 FPRD & FSUM: Non-interactive Sums & Products

These programs are well suited for use by other programs. They print no status, and are significantly faster than the interactive tool FSPT. Note that they do not support the LN transform capability of FSPT. I debated putting them in the menu because they are really more suited to programmatic use.

2.3 Organizing Principles

All the top level programs use PICF to collect information about the MVAR program to be used. The collected information is housed in three global variables:

Variable	Contents
EqP	Name of MVAR program
EqV	Name of independent variable
EqD	Number of returns from MVAR program

Depending on what we are doing with the MVAR program, we may need to know something about the independent variable too. For example when summing function values we need to know lower and upper ranges over which to sum, and a step size by which to increment the variable. We use a consistent set of global variables based upon names used by built in tools:

Variable	Contents
LLIM	Lower limit of independent variable
ULIM	Upper limit of independent variable
STEP	A step size

If a second variable is used, then similar variables are used; however, we have no precedent for what to name them. So I have decided to use the following:

Variable	Contents
YMIN	Lower limit of second or dependent variable
YMAX	Upper limit of second or dependent variable
YSTP	A step size for a second or dependent variable

If an accuracy is required, then we use the variable ACC.

0000 Page 1 of menu FMENU

FMENU: Main menu for all apps

3.1 Menu Code

```
(MJR-generate-42-menu-code "FMENU" 0 tbl 0 1 'stay 'up 'auto #'MJR-custom-gen-lab #'MJR-custom-gen-sub)
(FMENU)
0000 DSC: Auto-generated menu program
LBL "FMENU"
```

"SPT" KEY 1 XEQ "FSPT"

XEO 99

LBL 01

CLMENU

KEY 2 XEQ "FPLT" "EVAL"

KEY 4 XEQ 91

"FVARM" KEY 5 XEQ 95

"PICF"

KEY 6 XEQ 96

KEY 7 GTO 02

KEY 8 GTO 02 KEY 9 GTO 00

MENU

STOP GTO 01

LBL 02 0000 Page 2 of menu FMENU

CLMENU "SOLVE"

KEY 1 XEQ "FSLV"

"INTG"

KEY 2 XEQ "FINT"

KEY 3 XEQ "FSUM"

"PROD"

KEY 4 XEQ "FPRD"

"FVARM"

KEY 5 XEQ 95

"PICF"

KEY 6 XEQ 96

KEY 7 GTO 01

KEY 8 GTO 01

KEY 9 GTO 00

MENU

STOP GTO 02

LBL 00 @@@@ Application Exit

EXITALL RTN

0000 Free labels start at: 3

3.2 Functions

@@@@ LBL: 80-87, 91-87, 99

```
occorrections of the contraction of the contracti
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     (PICF)
```

@@@@ DSC: Pick an MVAR function and variable

0000 IN: X: integer

0000 FAQ: If X=0, then EqD is set to 1. Otherwise the user is prompted.

@@@@ UPD: 2021-02-24

0000 GBL: EqP -- Name of selected function @@@@ GBL: EqV -- Name of selected variable

0000 GBL: EqD -- Return count

@@@@ BUG: Variable menu works like solver, not like integrator. Can't fix that...

0000 REQ: REQ:free42>=3.0

```
LBL "PICF"
FUNC 10
L4STK
CLV "EqP"
CLV "EqV"
CLV "EqD"
X≠0?
GTO 87
1
STO "EqD"
LBL 87
EXITALL
PGMMENU
"Pick Function"
AVIEW
STOP
ASTO "EqP"
EXITALL
VARMENU IND "EqP"
"Set Var; "
⊢"Pick Free Var"
AVIEW
STOP
ASTO "EqV"
EXITALL
SF 25
RCL "EqD"
FS?C 25
GTO 86
"Select Return"
⊢" Count"
AVIEW
CLMENU
"1"
KEY 1 XEQ 82
"2"
KEY 2 XEQ 83
"3"
KEY 3 XEQ 84
"4"
KEY 4 XEQ 85
MENU
STOP
EXITALL
STO "EqD"
LBL 86
"F: "
ARCL "EqP"
⊢"("
ARCL "EqV"
⊢"):→"
ARCL "EqD"
AVIEW
RTN
LBL 82
1
RTN
LBL 83
RTN
LBL 84
3
RTN
LBL 85
4
RTN
0000 DSC: Run PICF with a 1 value for X -- so return count is queried
LBL 96 @NM@ IPICF
FUNC 00
1
XEQ "PICF"
RTN
```

```
LBL 95 @NM@ FVARMM
"Set Vars; "
⊢"R/S To End"
AVIEW
VARMENU "FVARM"
STOP
RTN
(FPRD)
0000 DSC: Product of a function evaluated at regular intervals over a range
@@@@ OUT: X: The product
0000 FAQ: No error checking is done on LLIM, ULIM, STEP, EqV, or EqP
0000 FAQ: Faster than product in FSPT
0000 FAQ: EqD is ignored -- only last function value is summed
@@@@ UPD: 2021-04-06
@@@@ REQ: REQ:free42>=3.0
LBL "FPRD"
FUNC 01
LSTO "_PROD"
0000 Step through independent variable values and do product
RCL "LLIM"
STO IND "EqV"
LBL 80
0000 Evaluate function
SF 25
XEQ IND "EqP"
FC?C 25
RTNERR 2
STO× "_PROD"
0000 Increment independent variable
RCL "STEP"
RCL+ IND "EqV"
STO IND "EqV"
RCL "ULIM"
X≥Y?
GTO 80
0000 All done, put Prd on stack
RCL "_PROD"
RTN
(FSUM)
0000 DSC: Sum a function evaluated at regular intervals over a range
0000 OUT: X: The sum
0000 FAQ: No error checking is done on LLIM, ULIM, STEP, EqV, or EqP
0000 FAQ: Faster than sum in FSPT
0000 FAQ: EqD is ignored -- only last function value is summed
@@@@ UPD: 2021-04-06
0000 REQ: REQ:free42>=3.0
LBL "FSUM"
FUNC 01
0
LSTO "_SUM"
0000 Step through independent variable values and do sum
RCL "LLIM"
STO IND "EqV"
LBL 81
0000 Evaluate function
SF 25
XEQ IND "EqP"
FC?C 25
RTNERR 2
STO+ "_SUM"
0000 Increment independent variable
RCL "STEP"
RCL+ IND "EqV"
STO IND "EqV"
RCL "ULIM"
X≥Y?
GTO 81
0000 All done, put Sum on stack
RCL "_SUM"
RTN
(FSLV)
```

@@@@ DSC: Wrap "FVARM"

5

```
0000 DSC: Use SOLVE to find a root for EqV in EqP on [LLIM, ULIM]
@@@@ OUT: T: Code indicating solver exit reason
                     0 Found a root
രരരര
                     1 Sign Reversal
0000
                      2 Extremum
0000
                       3 Bad Guess or Guesses
രരരര
                       3 Constant?
@@@@ OUT: Z: EqP evaluated at X
0000 OUT: Y: Previous best guess
0000 OUT: X: Final best guess -- a root if we are lucky
0000 FAQ: No error checking is done on LLIM, ULIM, STEP, EqV, or EqP
0000 FAQ: Faster than product in FSPT
0000 FAQ: EqD is ignored -- only last function value is summed
@@@@ GBL: EqP, EqV, LLIM, ULIM
@@@@ UPD: 2021-04-06
0000 REQ: REQ:free42>=3.0
LBL "FSLV"
FUNC 04
PGMSLV IND "EqP"
RCL "LLIM"
STO IND "EqV"
RCL "ULIM"
SOLVE IND "EqV"
R.TN
(FINT)
0000 DSC: Use INTEG to find a root for EqV in EqP on [LLIM, ULIM]
0000 OUT: X: The integral
@@@@ FAQ: No error checking is done on ACC, LLIM, ULIM, STEP, EqV, or EqP
0000 FAQ: EqD is ignored -- only last function value is summed
@@@@ GBL: EqP, EqV, LLIM, ULIM, ACC
@@@@ UPD: 2021-04-06
@@@@ REQ: REQ:free42>=3.0
LBL "FINT"
FUNC 03
PGMINT IND "EqP"
INTEG IND "EqV"
occorrections of the contraction of the contracti
                                                                                                                                                                   (FEVL)
0000 DSC: Evaluate EqP at X
@@@@ OUT: X: EqP(x)
0000 FAQ: No error checking is done on EqV or EqP
@@@@ GBL: EqP, EqV
@@@@ UPD: 2021-04-09
@@@@ REQ: REQ:free42>=3.0
LBL "FEVL"
STO IND "EqV"
XEQ IND "EqP"
RTN
0000 DSC: FEVL Menu Action
LBL 94 @NM@ FEVLM
XEQ 91 @NM@ CHKEqX
X≠0?
GTO 97 @NM@ AVRTN
XEQ "FEVL"
RTN
0000 DSC: GTO TARGET!! Display string and cause calling function to return
LBL 97 @NM@ AVRTN
AVIEW
RTN
0000 DSC: FEVLM menu action -- wrapper for FEVL
LBL 91 @NM@ FEVLM
"ERR: EqP"
⊢" is unset"
SF 25
RCL "EqP"
FC?C 25
GTO 92
"ERR: EqV"
```

```
⊢" is unset"
R↓
SF 25
RCL "EqV"
FC?C 25
GTO 92
"ERR: EqD"
⊢" is unset"
R↓
SF 25
RCL "EqD"
FC?C 25
GTO 92
XEQ "FEVL"
RTN
LBL 92
AVTEW
RTN
(FVARM)
@@@@ DSC: MVAR Program with all the global vartiables used by the F* programs
@@@@ GBL: EqP, EqV, EqD, ACC, WFUN, LLIM, ULIM, STEP, YMIN, YMAX, & YSTP
@@@@ UPD: 2021-04-06
LBL "FVARM"
MVAR "LLIM"
MVAR "ULIM"
MVAR "STEP"
MVAR "YMIN"
MVAR "YMAX"
MVAR "YSTP"
MVAR "EqP"
MVAR "EqV"
MVAR "EqD"
MVAR "ACC"
MVAR "WFUN"
RTN
(SFWRP)
{\tt QQQQQ} DSC: Wrap a simple function inside an =MVAR= program
LBL "SFWRP"
MVAR "WFUN"
MVAR "X"
RCL "X"
XEQ IND "WFUN"
RTN
0000 I.BI. for FPI.T menu
LBL 99 @NM@ FPLTM
CLA
SF 25
RCL "GrMod"
FC?C 25
RTNNO
"PLOT"
RTNYES
END
```

4 FSPT: Interactive Sums, Products, & Tables

0000 GBL: ULIM -- Upper limit for variable

Create function tables and computes sums & products. Works much like the built in integration application.

The LOG button transforms the function results by taking the natural logarithm – for example this allows us to compute very large products by summing the logarithms.

The ${\tt IND}$ button stores the independent variable value in the table.

When not provided, the values for LLIM, STEP, & EqD default to 1.

Note: Using [SHIFT] before [LLIM], [ULIM], and [STEP] menu keys will recall the current value to the stack instead of setting the value.

```
0000 GBL: STEP -- Size of steps to make
0000 FLG: 82: Set: log function returns
0000 FLG: 83: Set: Store independent variable in table
0000 FLG: 84: Set: doing table, Clear: Doing sum or product (see flag 85)
0000 FLG: 85: Set: doing sum, Clear; doing product
@@@@ LLB: 00-12,14-24,26-33
@@@@ GLB: FSPT
0000 USE: PICF
@@@@ REQ: REQ:free42>=3.0
LBL "FSPT"
CF 82
SF 83
1
XEQ "PICF"
R↓
                  0000 Page 1 of menu PROG_NAME
LBL 01
CLMENU
"LLIM"
KEY 1 XEQ 03
"ULIM"
KEY 2 XEQ 04
"STEP"
KEY 3 XEQ 05
πΣ.π
KEY 4 XEQ 06
"\pi"
KEY 5 XEQ 07
""
KEY 6 XEQ 08
KEY 7 GTO 02
KEY 8 GTO 02
KEY 9 GTO 00
MENU
STOP
GTO 01
LBL 02
                  0000 Page 2 of menu PROG_NAME
CLMENU
"EQ"
KEY 1 XEQ 09
FS? 82
"LN•"
FC? 82
"LN"
KEY 2 XEQ 10
FS? 83
"IND•"
FC? 83
"IND"
KEY 3 XEQ 11
KEY 7 GTO 01
KEY 8 GTO 01
KEY 9 GTO 00
MENU
STOP
GTO 02
LBL 00
EXITALL
RTN
LBL 03
                     0000 Action for menu key LLIM
FS? 64
RCL "LLIM"
STO "LLIM"
"LLIM: "
ARCL ST X
AVIEW
RTN
                     0000 Action for menu key ULIM
LBL 04
FS? 64
RCL "ULIM"
STO "ULIM"
"ULIM: "
ARCL ST X
AVIEW
RTN
LBL 05
                     0000 Action for menu key STEP
FS? 64
```

```
RCL "STEP"
STO "STEP"
"STEP: "
ARCL ST X
AVIEW
RTN
LBL 06
                    0000 Action for menu key \Sigma
CF 84
                    0000 84 clear -> not a table
SF 85
                     0000 85 set -> sum
GTO 20
RTN
                    0000 Action for menu key \pi
LBL 07
CF 84
                     @@@@ 84 clear -> not a table
                     @@@@ 85 clear -> product
CF 85
GTO 20
RTN
LBL 08
                     @@@@ Action for menu key □
SF 84
                     0000 84 set -> table
GTO 20
RTN
LBL 09
                     0000 Action for menu key EQ
1
XEQ "PICF"
R↓
RTN
LBL 10
                     0000 Action for menu key LN
FS?C 82
RTN
SF 82
RTN
LBL 11
                     @@@@ Action for menu key IND
FS?C 83
RTN
SF 83
RTN
LBL 20 \, QQQQ Guts of the sum, product, table code
0000 Default LLIM & STEP to 1 if unset
1
SF 25
RCL "LLIM"
FC?C 25
STO "LLIM"
SF 25
RCL "STEP"
FC?C 25
STO "STEP"
0000 Check for bad LLIM, ULIM, & STEP.
RCL "LLIM"
RCL "ULIM"
X>Y?
GTO 12
"ERR: LLIM ≥"
⊢" ULIM"
AVIEW
RTN
LBL 12
RCL "STEP"
X>0?
GTO 14
"ERR: STEP ≤ 0"
AVIEW
RTN
LBL 14
RCL "EqD"
X>0?
GTO 15
"ERR: EqD \leq 0"
AVIEW
RTN
LBL 15
4
X≽Y?
GTO 16
"ERR: EqD > 4"
AVIEW
RTN
```

```
LBL 16
0000 Init before loop
FS? 84
              0000 84 set -> table
GTO 21
                0000 84 clear -> (85 set -> sum , 85 clear -> product)
GTO 22
LBL 22
0000 Sum or Product
FS? 85
0
FC? 85
LSTO "_SOP"
GTO 23
LBL 21
0000 Table
1
ENTER
ENTER
RCL "EqD"
FS? 83
NEWMAT
LSTO "_TMAT"
R↓ @@@@ Drop matrix off stack to save RAM
INDEX "_TMAT"
GROW
LBL 23
0000 Step through independent variable values and do sum, product, or table
RCL "LLIM"
STO IND "EqV"
LBL 24
0000 Print progress
CLA
ARCL "EqV"
⊢"="
ARCL IND "EqV"
AVIEW
FC? 84
GTO 32
0000 Doing a table: Setup CTR for loop later
RCL "EqD"
LSTO "_CTR"
0000 Doing a table: Store independent variable if FS? 83
FC? 83
GTO 32
RCL IND "EqV"
ST0EL
J+
LBL 32
0000 Evaluate function
SF 25
XEQ IND "EqP"
FS?C 25
GTO 17
"ERR: Func Eval"
AVIEW
RTN
LBL 17
0000 Do thing for sum, product, or table
FS? 84
GTO 26
GTO 27
LBL 27
@@@@ Sum or Product
FC? 82
GTO 29
SF 25
LN
FS?C 25
GTO 29
"ERR: Bad Log"
AVIEW
LBL 29
FS? 85
STO+ "_SOP"
FC? 85
```

```
STO× "_SOP"
GTO 28
LBL 26
0000 Table
LBL 30
FC? 82
GTO 31
SF 25
LN
FS?C 25
GTO 31
"ERR: Bad Log"
AVIEW
LBL 31
STOEL
J+
R↓
DSE "_CTR"
GTO 30
LBL 28
0000 Increment independent variable
RCL "STEP"
RCL IND "EqV"
STO IND "EqV"
RCL "ULIM"
х≽ү?
GTO 24
0000 All done. Report Results
FS? 84
                0000 84 set -> table
                0000 84 clear -> (85 set -> sum , 85 clear -> product)
GTO 33
GTO 19
LBL 19
0000 Sum or Product
FS? 85
"SUM: "
FC? 85
"PROD: "
RCL "_SOP"
GTO 18
LBL 33
0000 Table
CLA
RCL "_TMAT"
LBL 18
ARCL ST X
AVIEW
R.TN
END
```

5 FPLT: Interactive Plotting

Features:

- Designed for high resolution mode of DM42
- $\bullet~$ Dots are connected
- Interactive zoom-box
- Optional autoscale of y-axis
- $\bullet\,$ Control over axis and grid line drawing
- · Quick access to solver and integration from plot
- Factor zoom
- Default window
- $\bullet\,$ Minimal & efficient interface
- $\bullet\,$ Integrated with other tools on this page solver & integration

5.1 User Interface

5.1.1 Main Menu

Menu	Description
EQ	Setup Equation
XRNG	Set X range (X: max Y: min). MAGIC
YRNG	Set Y range (X: max Y: min). MAGIC
GRID	Set grid width (X: Y-width Y: X-width). MAGIC
COMP	Computations (solve & integrate)
PLOT	Draw plot

Magic:

- XRNG set to [0, 0]: Sets range to default [-10, 10].
- XRNG reversed limits: Automatically detected and swapped
- XRNG set to [M, M]: Zooms range by a factor of M.
- YRNG set to [0, 0]: Sets range to default [-6, 6]
- YRNG reversed limits: Automatically detected and swapped
- YRNG set to [M, M]: Zooms range by a factor of M.
- GRID set to 0: If one of the grid widths is zero, then no grids are drawn for that axis
- GRID set to a negative value: No axis or grid marks will be drawn for that axis
- GRID: If grid lines are closer than 10 pixels, then grids are suppressed.

For a graph with default settings: [0] [ENTER] [XRNG] [YRNG] [GRID] [PLOT]

5.2 Computational Menu

This menu is for doing computations on the current equation based on the current plot range or last zoom box region.

Menu	Description
ROOT	Find a root in plot range
INTG	Integrate over plot range with ACC equal to the area of a pixel
EVAL	Evaluate the function
	-
	-
R <> B	Swap plot range with last zoom box range

5.2.1 Graphics Display

Key	Action
[2]	Adjust bottom side of zoom box
[4]	Adjust left side of zoom box
[8]	Adjust top side of zoom box
[6]	Adjust right side of zoom box
[ENTER]	Zoom to box & Redraw
[X<>X]	Auto-scale Y & Redraw. Zoombox is used for x-range
[+]	Set zoom box adjustment speed to fast
[-]	Set zoom box adjustment speed to slow
[*]	Zoom out 2x (multiply ranges by 2)
[/]	Zoom in 2x (divide ranges by 2)
[←]	Exit to main menu. Note: saves the zoom box range.
[.]	Toggle connected dots in graph

5.3 Code

```
@@@@ DSC: Plot a MVAR function
```

@@@@ GBL: EqP, EqV, LLIM, ULIM, STEP, YMIN, YMAX, YSTP

0000 GBL: ZLLIM, ZULIM -- zoom box limits.
0000 FAQ: STEP & YSTP are used for grids

0000 FLG: 03: PREF: Set: draw dots only; Clear: connect dots

0000 FLG: 04: PREF: Set: Use GrMod 2; Clear Use GrMod 3

@@@@ FLG: 05: PREF: Reserved

0000 FLG: 82: TEMP: Used for \max/\min computation

0000 FLG: 83: TEMP: Used for automatic autoscale redraw

0000 FLG: 84: TEMP: Reserved

@@@@ FRE: LBL: 79-99
@@@@ REQ: REQ:free42>=3.0

@@@@ UPD: 2021-04-09

LBL "FPLT"

(FPLT)

```
FC? 04
3
FS? 04
2
STO "GrMod"
R↓
131
STO "ResX"
R↓
16
STO "ResY"
R↓
XEQ 74
LBL 01
                     0000 Page 1 of menu FPLT
CF 83
                     0000 Turn off auto yscale redraw
CLMENU
"EQ"
KEY 1 XEQ 02
"XRNG"
KEY 2 XEQ 03
"YRNG"
KEY 3 XEQ 04
"GRID"
KEY 4 XEQ 05
"COMP"
KEY 5 XEQ 06
"PLOT"
KEY 6 XEQ 07
KEY 9 GTO 00
MENU
STOP
GTO 01
                     @@@@ Application Exit
LBL 00
EXITALL
RTN
LBL 02
                     0000 Action for menu key EQ \,
0
XEQ "PICF"
RTN
                     0000 Action for menu key XRNG
LBL 03
FUNC 00
FS? 64
                     @@@@ For shift we just display settings
GTO 10
X \neq Y?
GTO 24
X=0?
GTO 26
0000 Code for LLIM=ULIM\neq0
RCL "ULIM"
RCL- "LLIM"
2
RCL "ULIM"
RCL+ "LLIM"
2
RCL ST X
RCL ST Z
STO "LLIM"
R↓
STO "ULIM"
GTO 10
LBL 26
@@@@ Code for LLIM=ULIM=O
CLV "LLIM"
CLV "ULIM"
XEQ 74
GTO 10
LBL 24
0000 Code for LLIM\neqULIM
STO "ULIM"
X<>Y
STO "LLIM"
```

```
X<>Y
LBL 10
"XMIN: "
RCL "LLIM"
ARCL ST X
⊢"[LF]XMAX: "
RCL "ULIM"
ARCL ST X
AVIEW
RTN
LBL 04
                     0000 Action for menu key YRNG
FUNC 00
FS? 64
                     @@@@ For shift we just display settings
GTO 50
X \neq Y?
GTO 22
X=0?
GTO 23
0000 Code for YMIN=YMAX\neq0
RCL "YMAX"
RCL- "YMIN"
2
RCL "YMAX"
RCL+ "YMIN"
2
RCL ST X
RCL ST Z
STO "YMIN"
R↓
STO "YMAX"
GTO 50
LBL 23
@@@@ Code for YMIN=YMAX=0
CLV "YMIN"
CLV "YMAX"
XEQ 74
GTO 50
LBL 22
@@@@ Code for YMIN≠YMAX
STO "YMAX"
X<>Y
STO "YMIN"
X<>Y
LBL 50
"YMIN: "
RCL "YMIN"
ARCL ST X
⊢"[LF]YMAX: "
RCL "YMAX"
ARCL ST X
AVIEW
RTN
LBL 05
                     0000 Action for menu key GRID
FUNC 00
FS? 64
GTO 49
                     0000 For shift we just display settings
STO "YSTP"
X<>Y
STO "STEP"
LBL 49
"XSTP: "
RCL "STEP"
ARCL ST X
⊢"[LF]YSTP: "
RCL "YSTP"
ARCL ST X
AVIEW
RTN
LBL 06
                     0000 Action for menu key COMP
LBL 30
                     @@@@ menu COMP
CLMENU
```

```
"R00T"
KEY 1 XEQ 32
"INTG"
KEY 2 XEQ 33
"EVAL"
KEY 3 XEQ 34
@@@@ "RSV1"
0000 KEY 4 XEQ 35
0000 "RSV2"
0000 KEY 5 XEQ 36
"R<>B"
KEY 6 XEQ 37
KEY 9 GTO 31
MENU
STOP
GTO 30
LBL 31
                    0000 Menu Exit
RTN
LBL 32
                     0000 Action for menu key FSLV
FUNC 04
XEQ "FSLV"
RTN
LBL 33
                     0000 Action for menu key FINT
FUNC 01
RCL "ULIM"
RCL- "LLIM"
RCL÷ "ResX"
RCL "YMAX"
RCL- "YMIN"
RCL÷ "ResY"
XEQ "FINT"
RTN
LBL 34
                     0000 Action for menu key RSVO
XEQ "FEVL"
@@@@ RSVO
RTN
LBL 35
                     0000 Action for menu key RSV1
0000 RSV1
RTN
LBL 36
                     0000 Action for menu key RSV2
0000 RSV2
RTN
LBL 37
                     @@@@ Action for menu key RANGE <> BOX swap
XEQ 39
RTN
RCL "GrMod"
                     0000 Action for menu key PLOT
LBL 07
0000 Set AGRAPH flags to OR
CF 34
CF 35
0000 Just in case an important var is not set
XEQ 74
SF 25
RCL "EqV"
FC?C 25
GTO 76
R↓
SF 25
RCL "EqP"
FC?C 25
GTO 76
R↓
GTO 77 @@ All good
LBL 76
"ERR: Pick An"
⊢" EQ First!"
AVIEW
RTN
LBL 77
0000 Set YWID
RCL "YMAX"
RCL- "YMIN"
1
RCL "ResY"
```

15

```
STO "YWID"
0000 Set YWID
RCL "ULIM"
RCL- "LLIM"
RCL "ResX"
1
÷
STO "XWID"
0000 Draw stuff
CLLCD
CLMENU
                     @@@@ Only needed on DM42
EXITALL
                     0000 Only needed on DM42
FC? 83
GTO 17
0000 Doing an autoscale draw
RCL "ResX"
14
ΙP
1
LSTO "_I"
LBL 15
RCL "ResY"
2
                              0000 Yctr
9
RCL "_I"
14
                              0000 X coord
"@'px|~-~|xp'@"
                              0000 Draw up triangle
AGRAPH
"×∫πßffl?⊢?•→π∫×"
                               @@@@ Draw down triangle
X<>Y
12
X<>Y
AGRAPH
DSE "_I"
                               @@@@ Loopity doopity do
GTO 15
GTO 20
                     0000 Jump past axis and grid drawing for autoscale redraw
LBL 17
0000 Draw X Axis
RCL "STEP"
                     0000 No axis if negative
X<0?
GTO 11
XEQ 56
XEQ 58
1
XEQ 58
2
XEQ 58
LBL 11
0000 Draw X Grid
RCL "STEP"
Χ≤0?
GTO 08
                     0000 no grid when STEP<=0
RCL "XWID"
10
X>Y?
GTO 08
                    0000 no grid when too small
R↓
RCL "ULIM"
RCL÷ "STEP"
ΙP
RCL× "STEP"
RCL+ "STEP"
RCL "LLIM"
RCL÷ "STEP"
```

```
ΙP
RCL× "STEP"
RCL- "STEP"
LBL 09
ENTER
XEQ 46
XEQ 48
R↓
RCL+ "STEP"
X<Y?
GTO 09
LBL 08
0000 Draw Y Axis
RCL "YSTP"
                     0000 No axis if negative
X<0?
GTO 12
0
XEQ 46
XEQ 48
1
XEQ 48
2
XEQ 48
LBL 12
0000 Draw Y Grid
RCL "YSTP"
X≪0?
GTO 20
                     0000 no grid when STEP<=0
RCL "YWID"
-10
X>Y?
GTO 20
                     0000 no grid when too small
R↓
RCL "YMAX"
RCL÷ "YSTP"
ΙP
RCL× "YSTP"
RCL+ "YSTP"
RCL "YMIN"
RCL÷ "YSTP"
ΙP
RCL× "YSTP"
RCL- "YSTP"
LBL 21
ENTER
XEQ 56
XEQ 58
R↓
RCL+ "YSTP"
X<Y?
GTO 21
LBL 20
0000 Set Alpha to our box
"\pi\pi\pi"
0000 Plot curve
@@@@ Init YPMAX & YPMIN
SF 82
0000 Init LASTY
0
LSTO "_LASTY"
0000 Loop across screen
RCL "ResX"
1000
÷
1
+
LSTO "_XICUR"
LBL 41
RCL "_XICUR"
ΙP
XEQ 47
STO IND "EqV"
```

```
SF 25
XEQ IND "EqP"
FC?C 25
GTO 29
FC? 82
GTO 43
@@@@ Setup YPMIN & YPMAX
LSTO "_YPMAX"
LSTO "_YPMIN"
CF 82
LBL 43
@@@@ Update YPMIN & YPMAX if required.
RCL "_YPMAX"
X<>Y
X>Y?
LSTO "_YPMAX"
RCL "_YPMIN"
X<>Y
X<Y?
LSTO "_YPMIN"
FC? 83
GTO 78
0000 Draw autoscale progress bar
RCL "ResY"
1
RCL "_XICUR"
AGRAPH
R↓
R↓
GTO 19
LBL 78
0000 If y is on screen, then draw it
XEQ 56
XEQ 55
GTO 29
@@@@ DRAW FAT PIXEL
RCL "_XICUR"
                   @@@ XICUR YICUR
XEQ 51
0000 Do we draw lines?
FS? 03
GTO 19
0000 Don't do lines for an autoscale draw
FS? 83
GTO 19
0000 Do we have a last point?
RCL "_LASTY"
X=0?
GTO 19
0000 Yep & Yep. We draw a line!
RCL "_XICUR" @@@@ xi newy lasty
XEQ 40
LBL 19
0000 Save last Y
R↓
LSTO "_LASTY"
GTO 42
LBL 29 @@@@ Bad Y target
LSTO "_LASTY"
0000 Backend of loop
LBL 42
ISG "_XICUR"
GTO 41
@@@@ Done with graph.
0000 Setup zoom box
10
LSTO "_ZSPD"
1
LSTO "_ZBTOP"
LSTO "_ZBLFT"
RCL "ResY"
```

```
LSTO "_ZBBOT"
RCL "ResX"
LSTO "_ZBRGT"
0000 Do we redraw with autoscale?
FS?C 83
GTO 68
0000 graph UI
LBL 16
RCL "_ZSPD"
LSTO "_CTR"
LBL 52
RCL "_ZBRGT"
RCL "_CTR"
XEQ 48
DSE "_CTR"
GTO 52
RCL "_ZSPD"
LSTO "_CTR"
LBL 53
RCL "_ZBBOT"
RCL "_CTR"
XEQ 58
DSE "_CTR"
GTO 53
RCL " ZSPD"
LSTO "_CTR"
LBL 54
RCL "_ZBTOP"
RCL "_CTR"
XEQ 58
DSE "_CTR"
GTO 54
RCL "_ZSPD"
LSTO "_CTR"
LBL 59
RCL "_ZBLFT"
RCL "_CTR"
XEQ 48
DSE "_CTR"
GTO 59
LBL 44
GETKEY1
13
X=Y?
                     0000 13 ENTER key
GTO 73
R↓
17
X=Y?
                      0000 24 Backspace key
GTO 60
R↓
32
X=Y?
                      0000 - speed slow key
GTO 66
R↓
37
X=Y?
                      0000 - speed fast key
GTO 65
R↓
24
                      0000 24 LEFT key
X=Y?
GTO 61
R↓
26
X=Y?
                     0000 26 RIGHT key
GTO 62
R↓
20
                      0000 20 TOP key
X=Y?
GTO 63
R↓
30
X=Y?
                      0000 30 BOTTOM key
```

```
GTO 64
R↓
14
X=Y?
                     0000 14 autoscale key
GTO 68
R↓
22
X=Y?
                     @@@@ 22 zoom in key
GTO 67
R↓
35
                     0000 35 period. toggle dots/lines
X=Y?
GTO 75
R↓
27
X=Y?
                     0000 27 zoom out key
GTO 69
GTO 44
                     0000 Nothing matched get another key
LBL 67
                     0000 22 / zoom
0.5
ENTER
ENTER
XEQ 03
XEQ 04
GTO 07
LBL 69
                     0000 27 x zoom out
2.0
ENTER
ENTER
XEQ 03
XEQ 04
GTO 07
                     0000 Autoscale Y
LBL 68
1
RCL "_ZBLFT"
ABS
RCL "ResX"
RCL "_ZBRGT"
ABS
X=0?
GTO 14
0000 Zoom box has been set
RCL "_ZBLFT"
XEQ 47
RCL "_ZBRGT"
XEQ 47
STO "ULIM"
X<>Y
STO "LLIM"
SF 83
GTO 07
LBL 14
0000 No zoom box
RCL "_YPMAX"
-5
RCL× "YWID"
RCL "_YPMIN"
5
RCL× "YWID"
STO "YMIN"
X<>Y
STO "YMAX"
GTO 07
LBL 75
                     0000 35 toggle dots/lines
FS?C 03
GTO 07
SF 03
GTO 07
                     0000 37 + set speed fast
LBL 65
10
LSTO "_ZSPD"
```

```
GTO 16
LBL 66
                    0000 32 - set speed slow
1
LSTO "_ZSPD"
GTO 16
LBL 61
                     @@@@ LEFT REDO
RCL "_ZBRGT"
-2
RCL× "_ZSPD"
RCL "_ZSPD"
RCL+ "_ZBLFT"
X<Y?
LSTO "_ZBLFT"
XEQ 47
STO "ZLLIM"
GTO 16
LBL 62
                    0000 RIGHT REDO
RCL "_ZBLFT"
RCL× "_ZSPD"
RCL "_ZSPD"
+/-
RCL+ "_ZBRGT"
X>Y?
LSTO "_ZBRGT"
XEQ 47
STO "ZULIM"
GTO 16
                     @@@@ TOP REDO
LBL 63
RCL "_ZBBOT"
-2
RCL× "_ZSPD"
                     @@@@ Upper_limit
RCL "_ZSPD"
RCL+ "_ZBTOP"
                     @@@ New_ZBTOP Lower_limit
X<Y?
LSTO "_ZBTOP"
GTO 16
LBL 64
                     @@@@ BOT REDO
RCL "_ZBTOP"
2
RCL× "_ZSPD"
                     @@@@ Lower_limit
RCL "_ZSPD"
+/-
RCL+ "_ZBBOT"
                     @@@ New_ZBBOT Lower_limit
X>Y?
LSTO "_ZBBOT"
GTO 16
LBL 73
RCL "_ZBTOP"
XEQ 57
RCL "_ZBBOT"
XEQ 57
STO "YMIN"
R↓
STO "YMAX"
RCL "_ZBRGT"
XEQ 47
RCL "_ZBLFT"
XEQ 47
STO "LLIM"
X<>Y
STO "ULIM"
GTO 07
LBL 60
RTN
LBL 40 @@@@ Draw Line
FUNC 00
0000 xi newy lasty
LSTO "_XNUE"
               0000 yn yo
R↓
RCL ST Y
              @@@@ yo yn yo
RCL- ST Y
             @@@@ yo-yn yn yo
```

```
ABS
                @@@@ |yo-yn| yn yo
3
X>Y?
RTN @@@@ Pts too close, no line to draw
0000 We are drawing a line!
R↓
R↓
                0000 yn yo
LSTO "_YNUE"
X<>Y
LSTO "_YOLD"
                 0000 yo yn
2
ΙP
LSTO "_YCTR"
@@@@ CTR -> OLD Y
RCL "_XNUE"
RCL "_YOLD"
RCL "_YCTR"
XEQ 70
0000 CTR -> NEW Y
R↓
R↓
R↓
RCL "_XNUE"
RCL "_YNUE"
RCL "_YCTR"
XEQ 70
R↓
R↓
R↓
RTN
LBL 70
                @@@@ Draw Fat, Vertical Line Segment Ystart Yend X . Yend already has a fat point drawn.
FUNC 00
X<Y?
GTO 71
0000 Line going up
STO- ST T
R↓
                @@@@ Ystart Yend X-1
1
STO- ST Z
                @@@@ Ystart Yend-1 X-1
R↓
3
                @@@@ Ystart-3 Yend+1 X-1
LBL 72
                0000 ycur Yend-1 X-1
RCL ST Z
                @@@@ X-1 ycur Yend-1 X-1
AGRAPH
R↓
                @@@@ ycur Yend-1 X-1
3
                0000 ycur(updated) Yend-1 X-1
X>Y?
GTO 72
RTN
LBL 71
0000 Line going down
STO- ST T
                0000 Ystart Yend X-1
R↓
1
STO- ST Z
R↓
                0000 Ystart Yend-1 X-1
LBL 18
                @@@@ Ycur Yend-1 X-1
RCL ST Z
                @@@@ X-1 Ycur Yend-1 X-1
AGRAPH
R↓
                0000 yc yn+1 xn-1
3
                0000 yc yn+1 xn-1
+
X<Y?
GTO 18
RTN
                     0000 Draw BIG Pix (assumes Alpha is set)
LBL 51
FUNC 00
1
```

```
X<>Y
1
X<>Y
AGRAPH
RTN
LBL 58
                     @@@@ HLINE
FUNC 00
XEQ 55
GTO 27
+/-
1
PIXEL
LBL 27
RTN
LBL 48
                     @@@@ VLINE
FUNC 00
XEQ 45
GTO 28
+/-
1
X<>Y
PIXEL
LBL 28
RTN
LBL 45
                     0000 X Out Of Range
FUNC 00
X≪0?
RTNYES
RCL "ResX"
X<Y?
RTNYES
RTNNO
                     0000 Y Out Of Range
LBL 55
FUNC 00
X≤0?
RTNYES
RCL "ResY"
X<Y?
RTNYES
RTNNO
                     @@@@ XR->XI
LBL 46
FUNC 11
RCL- "LLIM"
RCL÷ "XWID"
1
RTN
LBL 47
                     @@@@ XI->XR
FUNC 11
RCL× "XWID"
RCL+ "LLIM"
RTN
LBL 56
                     @@@@ YR->YI
FUNC 11
RCL- "YMAX"
RCL: "YWID"
1
RTN
LBL 57
                     @@@@ YI->YR
FUNC 11
RCL× "YWID"
RCL+ "YMAX"
RTN
LBL 39
                     0000 Swap rng & box
FUNC 00
SF 25
RCL "ZLLIM"
FC?C 25
GTO 38
```

```
SF 25
RCL "ZULIM"
FC?C 25
GTO 38
SF 25
RCL "LLIM"
FC?C 25
GTO 38
SF 25
RCL "ULIM"
FC?C 25
GTO 38
STO "ZULIM"
R↓
STO "ZLLIM"
R↓
STO "ULIM"
X<>Y
STO "LLIM"
"XMIN: "
ARCL ST X
R↓
⊢"[LF]XMAX: "
ARCL ST X
R↓
AVIEW
RTN
LBL 38
"ERR: Range or "
" | unset"
AVIEW
RTN
        0000 Set any unset global variables to defaults
LBL 74
10
SF 25
RCL "ULIM"
FC?C 25
STO "ULIM"
-10
SF 25
RCL "LLIM"
FC?C 25
STO "LLIM"
6
SF 25
RCL "YMAX"
FC?C 25
STO "YMAX"
-6
SF 25
RCL "YMIN"
FC?C 25
STO "YMIN"
SF 25
RCL "YSTP"
FC?C 25
STO "YSTP"
SF 25
RCL "STEP"
FC?C 25
STO "STEP"
RTN
END
```

6 Test Functions

A few MVAR functions to test with

```
1
SIGN
5
RTN
END
(TFE)
@@@@ DSC: Exponential
LBL "TFE"
MVAR "X"
RCL "X"
E↑X
RTN
END
(TFL)
@@@@ DSC: Linear M*X+B
LBL "TFL"
MVAR "X"
MVAR "M"
MVAR "B"
RCL "X"
RCL× "M"
RCL+ "B"
RTN
END
(TFQ)
0000 DSC: A quadratic with zeros at 0 and 1
LBL "TFQ"
MVAR "X"
RCL "X"
1
RCL "X"
RTN
END
(TFC)
0000 DSC: A cubic with zeros at -1, 0.9, and 1
LBL "TFC"
MVAR "X"
RCL "X"
1
RCL "X"
1
+
RCL "X"
.9
RTN
END
(TFO)
{\tt @@@@} DSC: A function that with increasing frequency approaching zero
LBL "TFO"
MVAR "X"
RCL "X"
ABS
0.001
1/X
SIN
RTN
END
(TFI)
0000 DSC: A function that has many undefined regions
@@@@ FAQ: Good test to make sure we don't connect across known discontinuities
LBL "TFI"
```

```
MVAR "X"
RCL "X"
MOD
ΙP
1/X
5
RTN
END
(TFM)
\tt 0000\ DSC\colon A function that has many step discontinuities
LBL "TFM"
MVAR "X"
RCL "X"
2
MOD
ΙP
.5
10
RTN
END
(TFF)
0000 DSC: A function flips between 1 and -1 each call
{\tt QQQQ} FAQ: Worst case performance for connecting dots
LBL "TFF"
MVAR "X"
FS?C 85
RTN
SF 85
-5
RTN
END
```

7 EOF