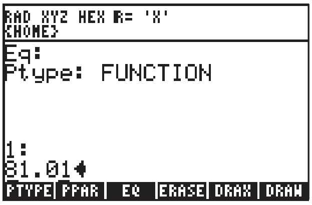
Chapter 22
Programs for graphics manipulation

This chapter includes a number of examples showing how to use the calculator's functions for manipulating graphics interactively or through the use of programs. As in Chapter 21 we recommend using RPN mode and setting system flag 117 to SOFT menu labels. « »

We introduce a variety of calculator graphic applications in Chapter 12. The examples of Chapter 12 represent interactive production of graphics using the calculator's pre-programmed input forms. It is also possible to use graphs in your programs, for example, to complement numerical results with graphics. To accomplish such tasks, we first introduce function in the PLOT menu.

The PLOT menu

Commands for setting up and producing plots are available through the PLOT menu. You can access the PLOT menu by using: 8 1 . 0 1 ← PRG NXT MODES MENU MENU.



```
RAD XYZ HEX R= 'X'
{HOME}
Eq:
Ptype: FUNCTION


1:
81.01◆
PTYPE PPAR  EQ  ERASE DRAX DRAW
```

The menu thus produced provides the user access to a variety of graphics functions. For application in subsequent examples, let's user-define the F3 (GRAPH) key to provide access to this menu as described below.

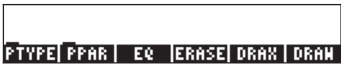
User-defined key for the PLOT menu

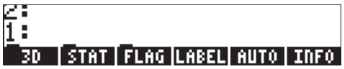
Enter the following keystrokes to determine whether you have any user-defined keys already stored in your calculator: ← PRG NXT MODES KEYS RCLKE.
Unless you have user-defined some keys, you should get in return a list containing an S, i.e., {S}. This indicates that the Standard keyboard is the only key definition stored in your calculator.

To user-define a key you need to add to this list a command or program followed by a reference to the key (see details in Chapter 20). Type the list `{ S << 81.01 MENU >> 13.0 }` in the stack and use function STOKEYS (← PRG NXT MODES KEYS STOK) to user-define key F3 as the access to the PLOT menu. Verify that such list was stored in the calculator by using ← PRG NXT MODES KEYS RCLK.

Note: We will not work any exercise while presenting the PLOT menu, its functions or sub-menus. This section will be more like a tour of the contents of PLOT as they relate to the different type of graphs available in the calculator.

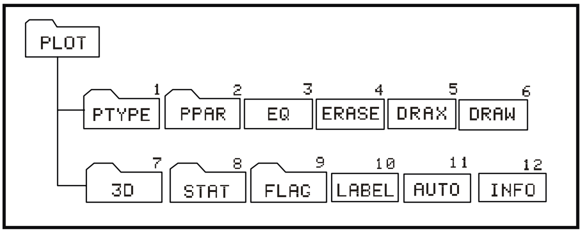
To activate a user defined key you need to press ← USER (same as the ALPHA key) before pressing the key or keystroke combination of interest. To activate the PLOT menu, with the key definition used above, press: ← USER F3 . You will get the following menu (press NXT to move to second menu)





Description of the PLOT menu

The following diagram shows the menus in PLOT. The number accompanying the different menus and functions in the diagram are used as reference in the subsequent description of those objects.



The soft menu key labeled 3D, STAT, FLAG, PTYPE, and PPAR, produce additional menus, which will be presented in more detail later. At this point we describe the functions directly accessible through soft menu keys for menu number 81.02. These are:

LABEL (10)
The function LABEL is used to label the axes in a plot including the variable names and minimum and maximum values of the axes. The variable names are selected from information contained in the variable PPAR.

AUTO (11)
The function AUTO (AUTOscale) calculates a display range for the y-axis or for both the x- and y-axes in two-dimensional plots according to the type of plot defined in PPAR. For any of the three-dimensional graphs the function AUTO produces no action. For two-dimensional plots, the following actions are performed by AUTO:

- FUNCTION: based on the plotting range of x, it samples the function in EQ and determines the minimum and maximum values of y.
- CONIC: sets the y-axis scale equal to the x-axis scale
- POLAR: based on the values of the independent variable (typically θ), it samples the function in EQ and determines minimum and maximum values of both x and y.
- PARAMETRIC: produces a similar result as POLAR based on the values of the parameter defining the equations for x and y.
- TRUTH: produces no action.
- BAR: the x-axis range is set from 0 to *n+1* where *n* is the number of elements in ΣDAT. The range of values of y is based on the contents of ΣDAT. The minimum and maximum values of y are determined so that the x-axis is always included in the graph.
- HISTOGRAM: similar to BAR.
- SCATTER: sets x- and y-axis range based on the contents of the independent and dependent variables from ΣDAT.

INFO (12)
The function INFO is interactive only (i.e., it cannot be programmed). When the corresponding soft menu key is pressed it provides information about the current plot parameters.

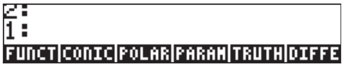
EQ (3)
The variable name EQ is reserved by the calculator to store the current equation in plots or solution to equations (see chapter …). The soft menu key labeled EQ in this menu can be used as it would be if you have your variable menu available, e.g., if you press [EQ] it will list the current contents of that variable.

ERASE (4)
The function ERASE erases the current contents of the graphics window. In programming, it can be used to ensure that the graphics window is cleared before plotting a new graph.

DRAX (5)
The function DRAX draws the axes in the current plot, if any is visible.

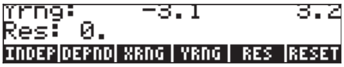
DRAW (6)
The function DRAW draws the plot defined in PPAR.

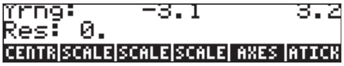
The PTYPE menu under PLOT (1)
The PTYPE menu lists the name of all two-dimensional plot types pre-programmed in the calculator. The menu contains the following menu keys:

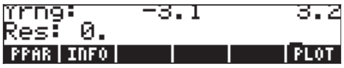


These keys correspond to the plot types *Function, Conic, Polar, Parametric, Truth*, and *Diff Eq*, presented earlier. Pressing one of these soft menu keys, while typing a program, will place the corresponding function call in the program. Press NXT PLOT to get back to the main PLOT menu.

The PPAR menu (2)
The PPAR menu lists the different options for the PPAR variable as given by the following soft menu key labels. Press NXT to move to next menus:

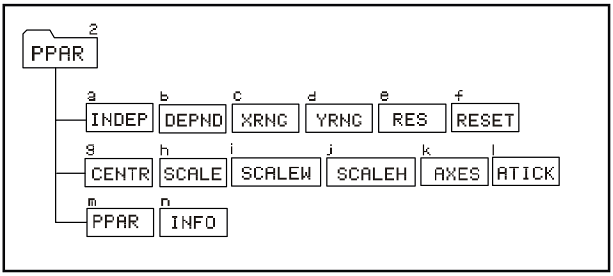

Yrng: -3.1 3.2
Res: 0.


Yrng: -3.1 3.2
Res: 0.


```
Yrng:     -3.1        3.2
Res: 0.
```

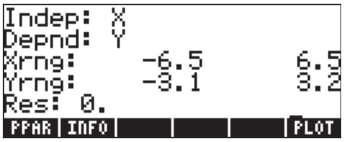
Note: the SCALE commands shown here actually represent SCALE, SCALEW, SCALEH, in that order.

The following diagram illustrates the functions available in the PPAR menu. The letters attached to each function in the diagram are used for reference purposes in the description of the functions shown below.



INFO (n) and PPAR (m)

If you press INFO, or enter ↱ PPAR, while in this menu, you will get a listing of the current PPAR settings, for example:


```
Indep: X
Depnd: Y
Xrng:     -6.5        6.5
Yrng:     -3.1        3.2
Res: 0.
```

This information indicates that X is the independent variable (Indep), Y is the dependent variable (Depnd), the x-axis range goes from –6.5 to 6.5 (Xrng), the y-axis range goes from –3.1 to 3.2 (Yrng). The last piece of information in the screen, the value of Res (resolution) determines the interval of the independent variable used for generating the plot.

The soft menu key labels included in the PPAR(2) menu represent commands that can be used in programs. These commands include:

INDEP (a)
The command INDEP specifies the independent variable and its plotting range. These specifications are stored as the third parameter in the variable PPAR. The default value is 'X'. The values that can be assigned to the independent variable specification are:

- A variable name, e.g., `'Vel'`
- A variable name in a list, e.g., { `Vel` }
- A variable name and a range in a list, e.g., { `Vel 0 20` }
- A range without a variable name, e.g., { `0 20` }
- Two values representing a range, e.g., `0 20`

In a program, any of these specifications will be followed by the command INDEP.

DEPND (b)
The command DEPND specifies the name of the dependent variable. For the case of TRUTH plots it also specifies the plotting range. The default value is Y. The type of specifications for the DEPND variable are the same as those for the INDEP variable.

XRNG (c) and YRNG (d)
The command XRNG specifies the plotting range for the x-axis, while the command YRNG specifies the plotting range for the y-axis. The input for any of these commands is two numbers representing the minimum and maximum values of x or y. The values of the x- and y-axis ranges are stored as the ordered pairs (x_{min}, y_{min}) and (x_{max}, y_{max}) in the two first elements of the variable PPAR. Default values for x_{min} and x_{max} are -6.5 and 6.5, respectively. Default values for x_{min} and x_{max} are –3.1 and 3.2, respectively.

RES (e)
The RES (RESolution) command specifies the interval between values of the independent variable when producing a specific plot. The resolution can be expressed in terms of user units as a real number, or in terms of pixels as a binary integer (numbers starting with #, e.g., #10). The resolution is stored as the fourth item in the PPAR variable.

<u>CENTR (g)</u>
The command CENTR takes as argument an ordered pair (x,y) or a value x, and adjusts the first two elements in the variable PPAR, i.e., (x_{min}, y_{min}) and (x_{max}, y_{max}), so that the center of the plot is (x,y) or (x,0), respectively.

<u>SCALE (h)</u>
The SCALE command determines the plotting scale represented by the number of user units per tick mark. The default scale is 1 user-unit per tick mark. When the command SCALE is used, it takes as arguments two numbers, x_{scale} and y_{scale}, representing the new horizontal and vertical scales. The effect of the SCALE command is to adjust the parameters (x_{min}, y_{min}) and (x_{max}, y_{max}) in PPAR to accommodate the desired scale. The center of the plot is preserved.

<u>SCALEW (i)</u>
Given a factor x_{factor}, the command SCALEW multiplies the horizontal scale by that factor. The W in SCALEW stands for 'width.' The execution of SCALEW changes the values of x_{min} and x_{max} in PPAR.

<u>SCALEH (j)</u>
Given a factor y_{factor}, the command SCALEH multiplies the vertical scale by that factor. The H in SCALEH stands for 'height.' The execution of SCALEW changes the values of y_{min} and y_{max} in PPAR.

Note: Changes introduced by using SCALE, SCALEW, or SCALEH, can be used to zoom in or zoom out in a plot.

<u>ATICK (l)</u>
The command ATICK (Axes TICK mark) is used to set the tick-mark annotations for the axes. The input value for the ATICK command can be one of the following:

- A real value x : sets both the x- and y-axis tick annotations to x units
- A list of two real values { x y }: sets the tick annotations in the x- and y-axes to x and y units, respectively.
- A binary integer #n: sets both the x- and y-axis tick annotations to #n pixels

A list of two binary integers {#n #m}: sets the tick annotations in the x- and y-axes to #n and #m pixels, respectively.

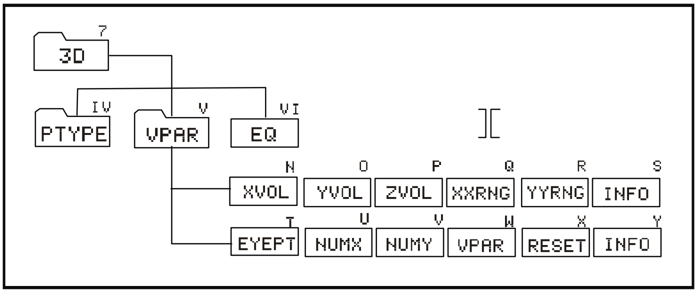
AXES (k)
The input value for the axes command consists of either an ordered pair (x,y) or a list {(x,y) *atick* "x-axis label" "y-axis label"}. The parameter *atick* stands for the specification of the tick marking annotations as described above for the command ATICK. The ordered pair represents the center of the plot. If only an ordered pair is given as input to AXES, only the axes origin is altered. The argument to the command AXES, whether an ordered pair or a list of values, is stored as the fifth parameter in PPAR.

To return to the PLOT menu, press PLOT.

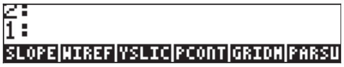
Press NXT to reach the second menu of the PLOT menu set.

RESET (f)
This button will reset the plot parameters to default values.

The 3D menu within PLOT (7)
The 3D menu contains two sub-menus, PTYPE and VPAR, and one variable, EQ. We are familiar already with the meaning of EQ, therefore, we will concentrate on the contents of the PTYPE and VPAR menus. The diagram below shows the branching of the 3D menu.

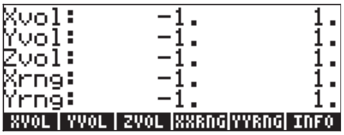


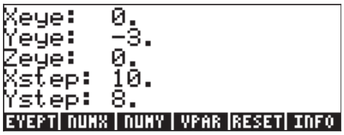
The PTYPE menu within 3D (IV)
The PTYPE menu under 3D contains the following functions:


```
2:
1:
SLOPE|WIREF|YSLIC|PCONT|GRIDM|PARSU
```

These functions correspond to the graphics options *Slopefield, Wireframe, Y-Slice, Ps-Contour, Gridmap* and *Pr-Surface* presented earlier in this chapter. Pressing one of these soft menu keys, while typing a program, will place the corresponding function call in the program. Press NXT 3D to get back to the main 3D menu.

The VPAR menu within 3D (V)
The variable VPAR stands for Volume PARameters, referring to a parallelepiped in space within which the three-dimensional graph of interest is constructed. When press [VPAR] in the 3D menu, you will get the following functions. Press NXT to move to the next menu:


```
Xvol:     -1.        1.
Yvol:     -1.        1.
Zvol:     -1.        1.
Xrng:     -1.        1.
Yrng:     -1.        1.
XVOL |YVOL |ZVOL |XXRNG|YYRNG|INFO
```


```
Xeye:  0.
Yeye:  -3.
Zeye:  0.
Xstep: 10.
Ystep: 8.
EYEPT|NUMX |NUMY |VPAR |RESET|INFO
```

Next, we describe the meaning of these functions:

<u>INFO (S) and VPAR (W)</u>
When you press INFO (S) you get the information shown in the left-hand side screen shot above. The ranges in *Xvol*, *Yvol*, and *Zvol* describe the extent of the parallelepiped in space where the graph will be generated. *Xrng* and *Yrng* describe the range of values of x and y, respectively, as independent variables in the x-y plane that will be used to generate functions of the form z = f(x,y).

Press NXT and INFO (Y) to obtain the information in the right-hand side screen shot above. These are the value of the location of the viewpoint for the three-dimensional graph (Xeye, Yeye, Zeye), and of the number of steps in x and y to generate a grid for surface plots.

XVOL (N), YVOL (O), and ZVOL (P)
These functions take as input a minimum and maximum value and are used to specify the extent of the parallelepiped where the graph will be generated (the viewing parallelepiped). These values are stored in the variable VPAR. The default values for the ranges XVOL, YVOL, and ZVOL are –1 to 1.

XXRNG (Q) and YYRNG (R)
These functions take as input a minimum and maximum value and are used to specify the ranges of the variables x and y to generate functions z = f(x,y). The default value of the ranges XXRNG and YYRNG will be the same as those of XVOL and YVOL.

EYEPT (T)
The function EYEPT takes as input real values x, y, and z representing the location of the viewpoint for a three-dimensional graph. The viewpoint is a point in space from which the three-dimensional graph is observed. Changing the viewpoint will produce different views of the graph. The figure below illustrates the idea of the viewpoint with respect to the actual graphic space and its projection in the plane of the screen.

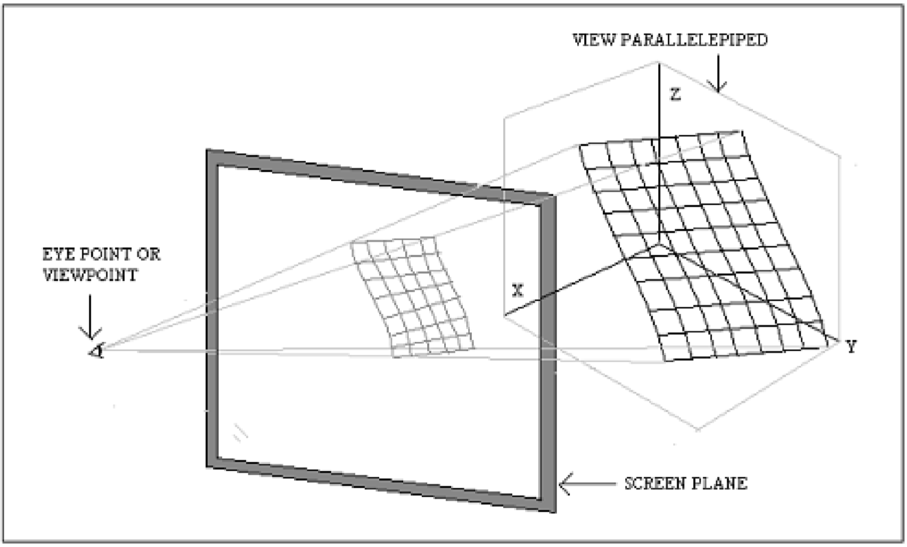
NUMX(U) and NUMY (V)
The functions NUMX and NUMY are used to specify the number of points or steps along each direction to be used in the generation of the base grid from which to obtain values of z = f(x,y).

VPAR (W)
This is just a reference to the variable VPAR.

RESET (X)
Resets parameters in screen to their default values.

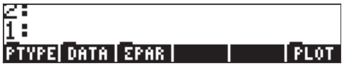
Press NXT 3D to return to the 3D menu.

Press PLOT to return to the PLOT menu.

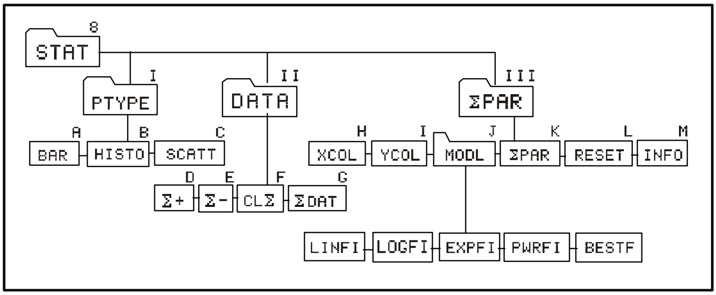


The STAT menu within PLOT

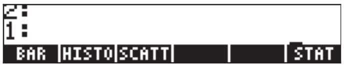
The STAT menu provides access to plots related to statistical analysis. Within this menu we find the following menus:



The diagram below shows the branching of the STAT menu within PLOT. The numbers and letters accompanying each function or menu are used for reference in the descriptions that follow the figure.

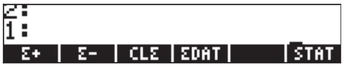


The PTYPE menu within STAT (I)
The PTYPE menu provides the following functions:


```
2:
1:
```

These keys correspond to the plot types *Bar (A), Histogram (B),* and *Scatter(C),* presented earlier. Pressing one of these soft menu keys, while typing a program, will place the corresponding function call in the program. Press STAT to get back to the STAT menu.

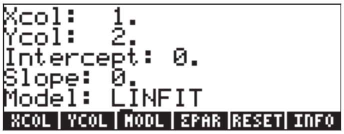
The DATA menu within STAT (II)
The DATA menu provides the following functions:


```
2:
1:
```

The functions listed in this menu are used to manipulate the ΣDAT statistical matrix. The functions Σ+ (D) and Σ- (E), add or remove data rows from the matrix ΣDAT. CLΣ (F) clears the ΣDAT (G) matrix, and the soft menu key labeled ΣDAT is just used as a reference for interactive applications. More details on the use of these functions are presented in a later chapter on statistical applications. Press STAT to return to the STAT menu.

The ΣPAR menu within STAT (III)

The ΣPAR menu provides the following functions:

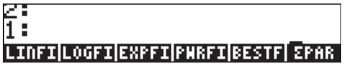

```
Xcol:   1.
Ycol:   2.
Intercept: 0.
Slope: 0.
Model: LINFIT
```

<u>INFO (M) and ΣPAR (K)</u>
The key INFO in ΣPAR provides the information shown in the screen shot above. The information listed in the screen is contained in the variable ΣPAR. The values shown are the default values for the x-column, y-column, intercept and slope of a data fitting model, and the type of model to be fit to the data in ΣDAT.

XCOL (H)
The command XCOL is used to indicate which of the columns of ΣDAT, if more than one, will be the x- column or independent variable column.

YCOL (I)
The command YCOL is used to indicate which of the columns of ΣDAT, if more than one, will be the y- column or dependent variable column.

MODL (J)
The command MODL refers to the model to be selected to fit the data in ΣDAT, if a data fitting is implemented. To see which options are available, press MODL. You will get the following menu:



These functions correspond to Linear Fit, Logarithmic Fit, Exponential Fit, Power Fit, or Best Fit. Data fitting is described in more detail in a later chapter. Press ΣPAR to return to the ΣPAR menu.

ΣPAR (K)
ΣPAR is just a reference to the variable ΣPAR for interactive use.

RESET (L)
This function resets the contents of ΣPAR to its default values.

Press NXT STAT to return to the STAT menu. Press [PLOT] to return to the main PLOT menu.

The FLAG menu within PLOT

The FLAG menu is actually interactive, so that you can select any of the following options:

- AXES: when selected, axes are shown if visible within the plot area or volume.
- CNCT: when selected the plot is produced so that individual points are connected.

- SIMU: when selected, and if more than one graph is to be plotted in the same set of axes, plots all the graphs simultaneously.

Press PLOT to return to the PLOT menu.

Generating plots with programs

Depending on whether we are dealing with a two-dimensional graph defined by a function, by data from ΣDAT, or by a three-dimensional function, you need to set up the variables PPAR, ΣPAR, and /or VPAR before generating a plot in a program. The commands shown in the previous section help you in setting up such variables.

Following we describe the general format for the variables necessary to produce the different types of plots available in the calculator.

Two-dimensional graphics

The two-dimensional graphics generated by functions, namely, *Function, Conic, Parametric, Polar, Truth and Differential Equation*, use PPAR with the format:

{ (x_{min}, y_{min}) (x_{max}, y_{max}) `indep res axes ptype depend` }

The two-dimensional graphics generated from data in the statistical matrix ΣDAT, namely, *Bar, Histogram,* and *Scatter*, use the ΣPAR variable with the following format:

```
{ x-column y-column slope intercept model }
```

while at the same time using PPAR with the format shown above.

The meaning of the different parameters in PPAR and ΣPAR were presented in the previous section.

Three-dimensional graphics

The three-dimensional graphics available, namely, options *Slopefield, Wireframe, Y-Slice, Ps-Contour, Gridmap* and *Pr-Surface*, use the VPAR variable with the following format:

{x_{left}, x_{right}, y_{near}, y_{far}, z_{low}, z_{high}, x_{min}, x_{max}, y_{min}, y_{max}, x_{eye}, y_{eye}, z_{eye}, x_{step}, y_{step}}

These pairs of values of x, y, and z, represent the following:

- Dimensions of the view parallelepiped (x_{left}, x_{right}, y_{near}, y_{far}, z_{low}, z_{high})
- Range of x and y independent variables (x_{min}, x_{max}, y_{min}, y_{max})
- Location of viewpoint (x_{eye}, y_{eye}, z_{eye})
- Number of steps in the x- and y-directions (x_{step}, y_{step})

Three-dimensional graphics also require the PPAR variable with the parameters shown above.

The variable EQ

All plots, except those based on ΣDAT, also require that you define the function or functions to be plotted by storing the expressions or references to those functions in the variable EQ.

In summary, to produce a plot in a program you need to load EQ, if required. Then load PPAR, PPAR and ΣPAR, or PPAR and VPAR. Finally, use the name of the proper plot type: FUNCTION, CONIC, POLAR, PARAMETRIC, TRUTH, DIFFEQ, BAR, HISTOGRAM, SCATTER, SLOPE, WIREFRAME, YSLICE, PCONTOUR, GRIDMAP, or PARSURFACE, to produce your plot.

Examples of interactive plots using the PLOT menu

To better understand the way a program works with the PLOT commands and variables, try the following examples of interactive plots using the PLOT menu.

Example 1 – <u>A function plot:</u>

↰ USER F3	Get PLOT menu (*)
PTYPE FUNCT	Select FUNCTION as the plot type
'√r' ENTER ↰ EQ	Store function '√r' into EQ

PPAR	Show plot parameters
ALPHA ← R ENTER INDEP	Define 'r' as the indep. variable
ALPHA ← S ENTER DEPND	Define 's' as the dependent variable
1 +/- SPC 10 XRNG	Define (-1, 10) as the x-range
1 +/- SPC 5 YRNG NXT	Define (-1, 5) as the y-range
{ (0,0) {.4 .2} "Rs" "Sr"} ENTER	Axes definition list
AXES	Define axes center, ticks, labels
NXT PLOT	Return to PLOT menu
ERASE DRAX NXT LABEL	Erase picture, draw axes, labels
NXT DRAW	Draw function and show picture
EDIT NXT MENU	Removes menu labels
NXT NXT PICT CANCL	Returns to normal calculator display

(*) PLOT menu available through user-defined key F3 as shown earlier in this Chapter.

Example 2 – <u>A parametric plot (Use RAD as angles)</u>:

← USER F3	Get PLOT menu
PTYPE PARAM	Select PARAMETRIC as the plot type
{ 'SIN(t)+i*SIN(2*t)' } ENTER	Define complex function X+iY
← EQ	Store complex function into EQ
PPAR	Show plot parameters
{t 0 6.29} ENTER INDEP	Define 't' as the indep.variable
ALPHA Y ENTER DEPND	Define 'Y' as the dependent variable
2.2 +/- SPC 2.2 XRNG	Define (-2.2,2.2) as the x-range
1.1 +/- SPC 1.1 YRNG NXT	Define (-1.1,1.1) as the y-range
{ (0,0) {.4 .2} "X(t)" "Y(t)"} ENTER	Axes definition list
AXES	Define axes center, ticks, labels
NXT PLOT	Return to PLOT menu
ERASE DRAX NXT LABEL	Erase picture, draw axes, labels
NXT DRAW	Draw function and show picture
EDIT NXT MENU NXT NXT PICT CANCL	Finish plot

Example 3 – <u>A polar plot</u>:

← USER F3	Get PLOT menu
PTYPE POLAR	Select POLAR as the plot type
'1+SIN(θ)' ENTER ← EQ	Store complex funct. r = f(θ) into EQ

PPAR	Show plot parameters
{ θ 0 6.29} ENTER INDEP	Define ‘θ’ as the indep. Variable
ALPHA Y ENTER DEPND	Define ‘Y’ as the dependent variable
3 +/- SPC 3 XRNG	Define (-3,3) as the x-range
0.5 +/- SPC 2.5 YRNG NXT	Define (-0.5,2.5) as the y-range
{ (0,0) {.5 .5} “x” “y”} ENTER	Axes definition list
AXES	Define axes center, ticks, labels
NXT PLOT	Return to PLOT menu
ERASE DRAX NXT LABEL	Erase picture, draw axes, labels
NXT DRAW	Draw function and show picture
EDIT NXT MENU	Remove menu labels
NXT NXT PICT CANCL	Return to normal calculator display

From these examples we see a pattern for the interactive generation of a two-dimensional graph through the PLOT menu:

1 – Select PTYPE.
2 – Store function to plot in variable EQ (using the proper format, e.g., ‘X(t)+iY(t)’ for PARAMETRIC).
3 – Enter name (and range, if necessary) of independent and dependent variables
4 – Enter axes specifications as a list { center atick x-label y-label }
5 – Use ERASE, DRAX, LABEL, DRAW to produce a fully labeled graph with axes

This same approach can be used to produce plots with a program, except that in a program you need to add the command PICTURE after the DRAW function is called to recall the graphics screen to the stack.

Examples of program-generated plots

In this section we show how to implement with programs the generation of the last three examples. Activate the PLOT menu before you start typing the program to facilitate entering graphing commands (← USER F3 , see above).

Example 1 – A function plot. Enter the following program:

```
«                                   Start program
{PPAR EQ} PURGE                     Purge current PPAR and EQ
'√r' STEQ                           Store '√r' into EQ
'r' INDEP                           Set independent variable to 'r'
's' DEPND                           Set dependent variable to 's'
FUNCTION                            Select FUNCTION as the plot type
{ (0.,0.) {.4 .2}
"Rs" "Sr" } AXES                    Set axes information
-1. 5. XRNG                         Set x range
-1. 5. YRNG                         Set y range
ERASE DRAW DRAX LABEL               Erase & draw plot, axes, and labels
PICTURE »                           Recall graphics screen to stack
```

Store the program in variable PLOT1. To run it, press VAR, if needed, then press PLOT1.

Example 2 – <u>A parametric plot</u>. Enter the following program:

```
«                                   Start program
RAD {PPAR EQ} PURGE                 Change to radians, purge vars.
'SIN(t)+i*SIN(2*t)' STEQ            Store 'X(t)+iY(t)' into EQ
{ t 0. 6.29} INDEP                  Set indep. variable to 'r', with range
'Y' DEPND                           Set dependent variable to 'Y'
PARAMETRIC                          Select PARAMETRIC as the plot type
{ (0.,0.) {.5 .5} "X(t)"
"Y(t)" } AXES                       Set axes information
-2.2 2.2 XRNG                       Set x range
-1.1 1.1 YRNG                       Set y range
ERASE DRAW DRAX LABEL               Erase & draw plot, axes, and labels
PICTURE                             Recall graphics screen to stack
»                                   End program
```

Store the program in variable PLOT2. To run it, press VAR, if needed, then press PLOT2.

Example 3 – A polar plot. Enter the following program:

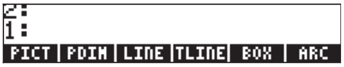
```
«                               Start program
RAD {PPAR EQ} PURGE             Change to radians, purge vars.
'1+SIN(θ)' STEQ                 Store 'f(θ)' into EQ
{ θ 0. 6.29} INDEP              Set indep. variable to 'θ', with range
'Y' DEPND                       Set dependent variable to 'Y'
POLAR                           Select POLAR as the plot type
{ (0.,0.) {.5 .5}
"x" "y"} AXES                   Set axes information
-3.  3. XRNG                    Set x range
-.5  2.5 YRNG                   Set y range
ERASE DRAW DRAX LABEL           Erase & draw plot, axes, and labels
PICTURE                         Recall graphics screen to stack
»                               End program
```

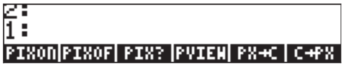
Store the program in variable PLOT3. To run it, press VAR, if needed, then press PLOT3.

These exercises illustrate the use of PLOT commands in programs. They just scratch the surface of programming applications of plots. I invite the reader to try their own exercises on programming plots.

Drawing commands for use in programming

You can draw figures in the graphics window directly from a program by using commands such as those contained in the PICT menu, accessible by ← PRG NXT PICT. The functions available in this menu are the following. Press NXT to move to next menu:





Obviously, the commands LINE, TLINE, and BOX, perform the same operations as their interactive counterpart, given the appropriate input. These and the other functions in the PICT menu refer to the graphics windows whose x- and y-ranges are determined in the variable PPAR, as demonstrated above for different graph types. The functions in the PICT command are described next:

PICT

This soft key refers to a variable called PICT that stores the current contents of the graphics window. This variable name, however, cannot be placed within quotes, and can only store graphics objects. In that sense, PICT is like no other calculator variables.

PDIM

The function PDIM takes as input either two ordered pairs (x_{min},y_{min}) (x_{max},y_{max}) or two binary integers #w and #h. The effect of PDIM is to replace the current contents of PICT with an empty screen. When the argument is (x_{min},y_{min}) (x_{max},y_{max}), these values become the range of the user-defined coordinates in PPAR. When the argument is #w and #h, the ranges of the user-defined coordinates in PPAR remain unchanged, but the size of the graph changes to #h $\times$ #v pixels.

PICT and the graphics screen

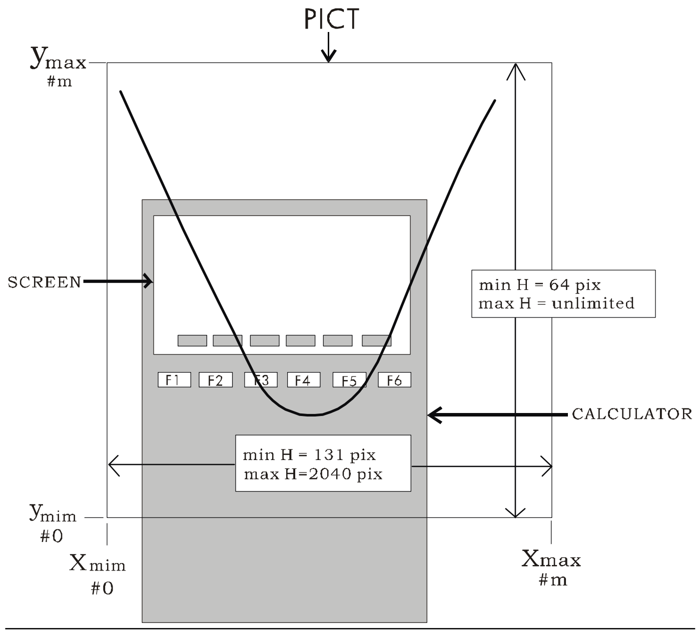
PICT, the storage area for the current graph, can be thought of as a two dimensional graph with a minimum size of 131 pixels wide by 64 pixels high. The maximum width of PICT is 2048 pixels, with no restriction on the maximum height. A pixel is each one of the dots in the calculator's screen that can be turned on (dark) or off (clear) to produce text or graphs. The calculator screen has 131 pixels by 64 pixels, i.e., the minimum size for PICT. If your PICT is larger than the screen, then the PICT graph can be thought of as a two dimensional domain that can be scrolled through the calculator's screen, as illustrated in the diagram shown next.

LINE

This command takes as input two ordered pairs (x_1,y_1) (x_2, y_2), or two pairs of pixel coordinates {#n_1 #m_1} {#n_2 #m_2}. It draws the line between those coordinates.

TLINE

This command (Toggle LINE) takes as input two ordered pairs (x_1,y_1) (x_2, y_2), or two pairs of pixel coordinates {#n_1 #m_1} {#n_2 #m_2}. It draws the line between those coordinates, turning off pixels that are on in the line path and vice versa.



BOX

This command takes as input two ordered pairs (x_1,y_1) (x_2, y_2), or two pairs of pixel coordinates {#n_1 #m_1} {#n_2 #m_2}. It draws the box whose diagonals are represented by the two pairs of coordinates in the input.

ARC

This command is used to draw an arc. ARC takes as input the following objects:

- Coordinates of the center of the arc as (x,y) in user coordinates or {#n, #m} in pixels.
- Radius of arc as r (user coordinates) or #k (pixels).
- Initial angle θ_1 and final angle θ_2.

PIX?, PIXON, and PIXOFF

These functions take as input the coordinates of point in user coordinates, (x,y), or in pixels {#n, #m}.

- PIX? Checks if pixel at location (x,y) or {#n, #m} is on.
- PIXOFF turns off pixel at location (x,y) or {#n, #m}.
- PIXON turns on pixel at location (x,y) or {#n, #m}.

PVIEW

This command takes as input the coordinates of a point as user coordinates (x,y) or pixels {#n, #m}, and places the contents of PICT with the upper left corner at the location of the point specified. You can also use an empty list as argument, in which case the picture is centered in the screen. PVIEW does not activate the graphics cursor or the picture menu. To activate any of those features use PICTURE.

PX→C

The function PX→C converts pixel coordinates {#n #m} to user-unit coordinates (x,y).

C→PX

The function C→PX converts user-unit coordinates (x,y) to pixel coordinates {#n #m}.

Programming examples using drawing functions

In this section we use the commands described above to produce graphics with programs. Program listing are provided in the attached diskette or CD ROM.

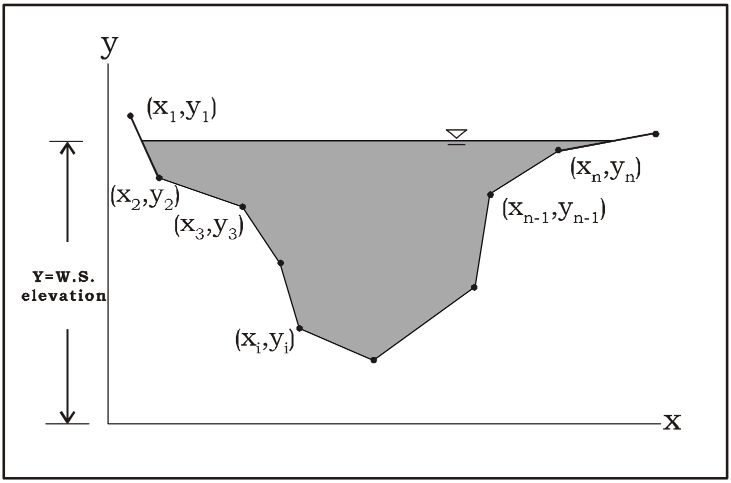
Example 1 - A program that uses drawing commands
The following program produces a drawing in the graphics screen. (This program has no other purpose than to show how to use calculator commands to produce drawings in the display.)

```
«                                 Start program
DEG                               Select degrees for angular measures
0. 100. XRNG                      Set x range
0. 50.  YRNG                      Set y range
ERASE                             Erase picture
(5., 2.5) (95., 47.5) BOX         Draw box from (5,5) to (95,95)
(50., 50.) 10. 0. 360. ARC        Draw a circle center (50,50), r =10.
```

```
(50., 50.) 12. –180. 180. ARC            Draw a circle center (50,50), r= 12.
1 8 FOR j                                Draw 8 lines within the circle
    (50., 50.) DUP                       Lines are centered as (50,50)
    '12*COS(45*(j-1))' →NUM              Calculate x, other end at 50 + x
    '12*SIN(45*(j-1))' →NUM              Calculates y, other end at 50 + y
     R → C                               Convert x y to (x,y), complex num.
     +                                   Add (50,50) to (x,y)
     LINE                                Draw the line
NEXT                                     End of FOR loop
{ } PVIEW                                Show picture
»
```

Example 2 - A program to plot a natural river cross-section
This application may be useful for determining area and wetted perimeters of natural river cross-sections. Typically, a natural river cross section is surveyed and a series of points, representing coordinates x and y with respect to an arbitrary set of coordinates axes. These points can be plotted and a sketch of the cross section produced for a given water surface elevation. The figure below illustrate the terms presented in this paragraph.

The program, available in the diskette or CD ROM that comes with your calculator, utilizes four sub-programs FRAME, DXBED, GTIFS, and INTRP. The main program, called XSECT, takes as input a matrix of values of x and y, and the elevation of the water surface Y (see figure above), in that order. The program produces a graph of the cross section indicating the input data with points in the graph, and shows the free surface in the cross-section.



It is suggested that you create a separate sub-directory to store the programs. You could call the sub-directory RIVER, since we are dealing with irregular open channel cross-sections, typical of rivers.

To see the program XSECT in action, use the following data sets. Enter them as matrices of two columns, the first column being x and the second one y. Store the matrices in variables with names such as XYD1 (X-Y Data set 1) and XYD2 (X-Y Data set 2). To run the program place one of the data sets in the stack, e.g., VAR XYD1, then type in a water surface elevation, say 4.0, and press XSECT. The calculator will show an sketch of the cross-section with the corresponding water surface. To exit the graph display, press ON.

Try the following examples:

XYD1 2 XSECT

XYD1 3 XSECT

XYD1 4 XSECT

XYD1 6 XSECT

Please be patient when running program XSECT. Due to the relatively large number of graphics functions used, not counting the numerical iterations, it may take some time to produce the graph (about 1 minute).

Data set 1

x	y
0.4	6.3
1.0	4.9
2.0	4.3
3.4	3.0
4.0	1.2
5.8	2.0
7.2	3.8
7.8	5.3
9.0	7.2

Data set 2

x	y
0.7	4.8
1.0	3.0
1.5	2.0
2.2	0.9
3.5	0.4
4.5	1.0
5.0	2.0
6.0	2.5
7.1	2.0
8.0	0.7
9.0	0.0
10.0	1.5
10.5	3.4
11.0	5.0

Note: The program FRAME, as originally programmed (see diskette or CD ROM), does not maintain the proper scaling of the graph. If you want to maintain proper scaling, replace FRAME with the following program:

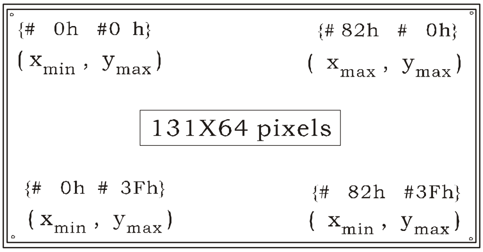
```
« STOΣ MINΣ MAXΣ 2 COL→ DUP →COL DROP – AXL ABS AXL 20 /
DUP NEG SWAP 2 COL→ + →ROW DROP SWAP → yR xR « 131 DUP
R→B SWAP yR OBJ→ DROP – xR OBJ→ DROP - / * FLOOR R→B
PDIM yR OBJ→ DROP YRNG xR OBJ→ DROP XRNG ERASE » »
```

This program keeps the width of the PICT variable at 131 pixels – the minimum pixel size for the horizontal axis – and adjusts the number of pixels in the vertical axes so that a 1:1 scale is maintained between the vertical and horizontal axes.

Pixel coordinates

The figure below shows the graphic coordinates for the typical (minimum) screen of 131×64 pixels. Pixels coordinates are measured from the top left corner of the screen {# 0h # 0h}, which corresponds to user-defined coordinates

(x_{min}, y_{max}). The maximum coordinates in terms of pixels correspond to the lower right corner of the screen {# 82h #3Fh}, which in user-coordinates is the point (x_{max}, y_{min}). The coordinates of the two other corners both in pixel as well as in user-defined coordinates are shown in the figure.

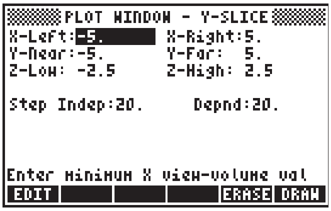


Animating graphics

Herein we present a way to produce animation by using the Y-Slice plot type. Suppose that you want to animate the traveling wave, f(X,Y) = 2.5 sin(X-Y). We can treat the X as time in the animation producing plots of f(X,Y) vs. Y for different values of X. To produce this graph use the following:

- ⬅ 2D/3D simultaneously. Select Y-Slice for TYPE. '2.5*SIN(X-Y)' for EQ. 'X' for INDEP. Press NXT OK.

- ⬅ WIN , simultaneously (in RPN mode). Use the following values:


PLOT WINDOW - Y-SLICE
X-Left:-5. X-Right:5.
Y-Near:-5. Y-Far: 5.
Z-Low: -2.5 Z-High: 2.5
Step Indep:20. Depnd:20.
Enter minimum X view-volume val

- Press ERASE DRAW. Allow some time for the calculator to generate all the needed graphics. When ready, it will show a traveling sinusoidal wave in your screen.

Animating a collection of graphics

The calculator provides the function ANIMATE to animate a number of graphics that have been placed in the stack. You can generate a graph in the graphics screen by using the commands in the PLOT and PICT menus. To place the generated graph in the stack, use PICT RCL. When you have *n* graphs in levels *n* through *1* of the stack, you can simply use the command *n* ANIMATE to produce an animation made of the graphs you placed in the stack.

Example 1 – Animating a ripple in a water surface
As an example, type in the following program that generates 11 graphics showing a circle centered in the middle of the graphics screen and whose radius increase by a constant value in each subsequent graph.

```
«                                  Begin program
RAD                                Set angle units to radians
131 R→B 64 R→B PDIM                Set PICT to 131×64 pixels
0 100 XRNG 0 100 YRNG              Set x- and y-ranges to 0-100
1 11 FOR j                         Start loop with j = 1 .. 11
  ERASE                            Erase current PICT
  (50., 50.) '5*(j-1)' →NUM        Centers of circles (50,50)
  0 '2*π' →NUM ARC                 Draw circle center r = 5(j-1)
  PICT RCL                         Place current PICT on stack
NEXT                               End FOR-NEXT loop
11 ANIMATE                         Animate
»                                  End program
```

Store this program in a variable called PANIM (Plot ANIMation). To run the program press VAR (if needed) PANIM. It takes the calculator more than one minute to generate the graphs and get the animation going. Therefore, be really patient here. You will see the hourglass symbol up in the screen for what seems a long time before the animation, resembling the ripples produced by a pebble dropped on the surface of a body of quiescent water, appears in the screen. To stop the animation, press ON.

The 11 graphics generated by the program are still available in the stack. If you want to re-start the animation, simply use: 11 ANIMATE. (Function

ANIMATE is available by using ← PRG NXT GROB NXT ANIMA). The animation will be re-started. Press ON to stop the animation once more. Notice that the number 11 will still be listed in stack level 1. Press ◀ to drop it from the stack.

Suppose that you want to keep the figures that compose this animation in a variable. You can create a list of these figures, let's call it WLIST, by using:

/ / ← PRG TYPE →LST ' ALPHA ALPHA W L I S T ALPHA STO▶

Press VAR to recover your list of variables. The variable WLIST should now be listed in your soft-menu keys. To re-animate this list of variables you could use the following program:

```
«                                     Start program
WLIST                                 Place list WLIST in stack
OBJ→                                  Decompose list, stack level 1 = 11
ANIMATE                               Start animation
»                                     End program
```

Save this program in a variable called RANIM (Re-ANIMate). To run it, press RANIM.

The following program will animate the graphics in WLIST forward and backwards:

```
«                                     Start program
WLIST  DUP                            Place list WLIST in stack, make extra copy
REVLIST  +                            Reverse order, concatenate 2 lists
OBJ→                                  Decompose list in elements, level 1 = 22
ANIMATE                               Start animation
»                                     End program
```

Save this program in a variable called RANI2 (Re-ANImate version 2). To run it, press RANI2. The animation now simulates a ripple in the surface of otherwise quiescent water that gets reflected from the walls of a circular tank back towards the center. Press ON to stop the animation.

Example 2 - Animating the plotting of different power functions

Suppose that you want to animate the plotting of the functions $f(x) = x^n$, $n = 0, 1, 2, 3, 4$, in the same set of axes. You could use the following program:

```
«                                    Begin program
RAD                                  Set angle units to radians
131 R→B 64 R→B PDIM                  Set PICT screen to 131×64 pixels
0 2 XRNG 0 20 YRNG                   Set x- and y-ranges
0 4 FOR j                            Start loop with j = 0,1,…,4
  'X^j'  STEQ                        Store 'X^j' in variable EQ
  ERASE                              Erase current PICT
  DRAX LABEL DRAW                    Draw axes, labels, function
  PICT RCL                           Place current PICT on stack
NEXT                                 End FOR-NEXT loop
5 ANIMATE                            Animate
»
```

Store this program in a variable called PWAN (PoWer function ANimation). To run the program press VAR (if needed) PWAN. You will see the calculator drawing each individual power function before starting the animation in which the five functions will be plotted quickly one after the other. To stop the animation, press ON.

More information on the ANIMATE function

The ANIMATE function as used in the two previous examples utilized as input the graphics to be animated and their number. You can use additional information to produce the animation, such as the time interval between graphics and the number of repetitions of the graphics. The general format of the ANIMATE function in such cases is the following:

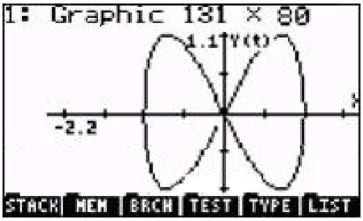
```
        n-graphs   { n {#X #Y} delay rep }  ANIMATE
```

`n` represents the number of graphics, `{#X #Y}` stand for the pixel coordinates of the lower right corner of the area to be plotted (see figure below), `delay` is the number of seconds allowed between consecutive graphics in the animation, and `rep` is the number of repetitions of the animation.

Graphic objects (GROBs)

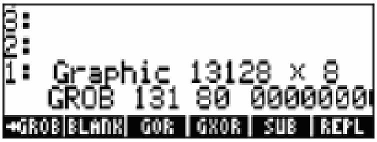
The word GROB stands for GRaphics OBjects and is used in the calculator's environment to represent a pixel-by-pixel description of an image that has been produced in the calculator's screen. Therefore, when an image is converted into a GROB, it becomes a sequence of binary digits (*b*inary dig*its* = *bits*), i.e., 0's and 1's. To illustrate GROBs and conversion of images to GROBS consider the following exercise.

When we produce a graph in the calculator, the graph become the contents of a special variable called PICT. Thus, to see the last contents of PICT, you could use: PICT RCL (← PRG NXT MEM DIR ← RCL).
The display shows in stack level 1 the `line Graphic 131×80` (if using the standard screen size) followed by a sketch of the top part of the graph. For example,



If you press ▽ then the graph contained in level 1 is shown in the calculator's graphics display. Press CANCL to return to normal calculator display.

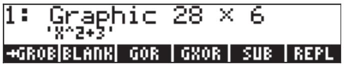
The graph in level 1 is still not in GROB format, although it is, by definition, a graphics object. To convert a graph in the stack into a GROB, use: 3 ENTER ← PRG NXT GROB →GROB . Now we have the following information in level 1:


```
3:
2:
1: Graphic 13128 × 8
   GROB 131 80 0000000
```

The first part of the description is similar to what we had originally, namely, `Graphic 131×64`, but now it is expressed as `Graphic 13128 × 8`. However, the graphic display is now replaced by a sequence of zeroes and ones representing the pixels of the original graph. Thus, the original graph as now been converted to its equivalent representation in bits.

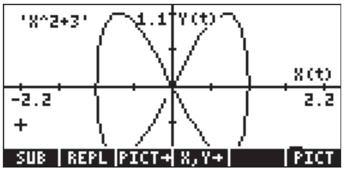
You can also convert equations into GROBs. For example, using the equation writer type in the equation 'X^2+3' into stack level 1, and then press

/ ENTER ← PRG NXT GROB →GROB . You will now have in level 1 the GROB described as:



```
1: Graphic 28 × 6
   'X^2+3'
→GROB BLANK GOR GXOR SUB REPL
```

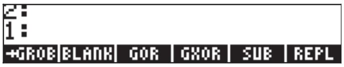
As a graphic object this equation can now be placed in the graphics display. To recover the graphics display press ◄. Then, move the cursor to an empty sector in the graph, and press EDIT NXT NXT REPL. The equation 'X^2-5' is placed in the graph, for example:



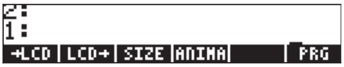
Thus, GROBs can be used to document graphics by placing equations, or text, in the graphics display.

The GROB menu

The GROB menu, accessible through ← PRG NXT GROB →GROB, contains the following functions. Press NXT to move to the next menu:



```
2:
1:
→GROB BLANK GOR GXOR SUB REPL
```



```
2:
1:
→LCD LCD→ SIZE ANIMA PRG
```

→GROB

Of these functions we have already used SUB, REPL, (from the graphics EDIT menu), ANIMATE [ANIMA], and →GROB. ([*PRG*] is simply a way to return to the programming menu.) While using →GROB in the two previous examples you may have noticed that I used a 3 while converting the graph into a GROB, while I used a 1 when I converted the equation into a GROB. This parameter of the function →GROB indicates the size of the object that is being converted into a GROB as 0 or 1 – for a small object, 2 – medium, and 3 – large. The other functions in the GROB menu are described following.

BLANK
The function BLANK, with arguments #n and #m, creates a blank graphics object of width and height specified by the values #n and #m, respectively. This is similar to the function PDIM in the GRAPH menu.

GOR
The function GOR (Graphics OR) takes as input *grob*$_2$ (a target GROB), a set of coordinates, and *grob*$_1$, and produces the superposition of *grob*$_1$ onto *grob*$_2$ (or PICT) starting at the specified coordinates. The coordinates can be specified as user-defined coordinates (x,y), or pixels {#n #m}. GOR uses the OR function to determine the status of each pixel (i.e., on or off) in the overlapping region between *grob*$_1$ and *grob*$_2$.

GXOR
The function GXOR (Graphics XOR) performs the same operation as GOR, but using XOR to determine the final status of pixels in the overlapping area between graphic objects *grob*$_1$ and *grob*$_2$.

Note: In both GOR and GXOR, when *grob2* is replaced by PICT, these functions produce no output. To see the output you need to recall PICT to the stack by using either PICT RCL or PICTURE.

→LCD
Takes a specified GROB and displays it in the calculator's display starting at the upper left corner.

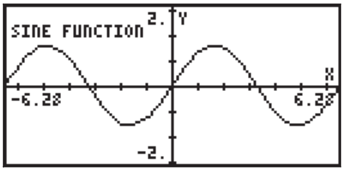
LCD→
Copies the contents of the stack and menu display into a 131 x 64 pixels GROB.

SIZE
The function SIZE, when applied to a GROB, shows the GROB's size in the form of two numbers. The first number, shown in stack level 2, represents the width of the graphics object, and the second one, in stack level 1, shows its height.

An example of a program using GROB
The following program produces the graph of the sine function including a frame – drawn with the function BOX – and a GROB to label the graph. Here is the listing of the program:

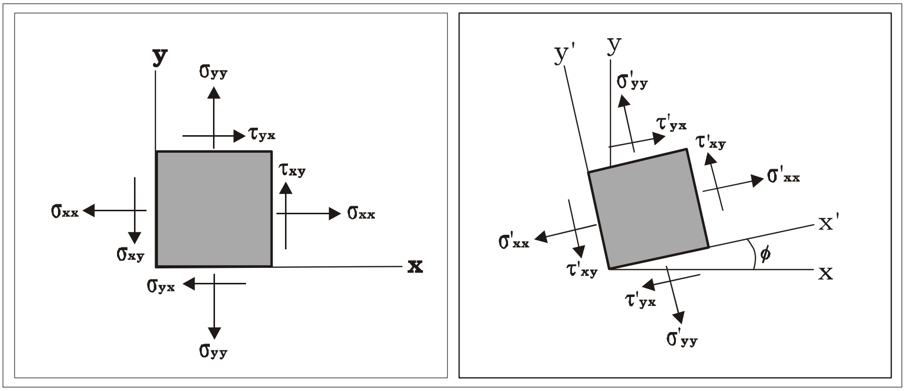
```
«                                   Begin program
RAD                                 Set angle units to radians
131 R→B 64 R→B PDIM                 Set PICT screen to 131×64 pixels
-6.28 6.28 XRNG –2. 2. YRNG         Set x- and y-ranges
FUNCTION                            Select FUNCTION type for graphs
'SIN(X)' STEQ                       Store the function sine into EQ
ERASE DRAX LABEL DRAW               Clear, draw axes, labels, graph
(-6.28,-2.) (6.28,2.) BOX           Draw a frame around the graph
PICT RCL                            Place contents of PICT on stack
"SINE FUNCTION"                     Place graph label string in stack
1 →GROB                             Convert string into a small GROB
(-6., 1.5) SWAP                     Coordinates to place label GROB
GOR                                 Combine PICT with the label GROB
PICT STO                            Save combined GROB into PICT
{ } PVIEW                           Bring PICT to the stack
»                                   End program
```

Save the program under the name GRPR (GROB PRogram). Press GRPR to run the program. The output will look like this:



A program with plotting and drawing functions

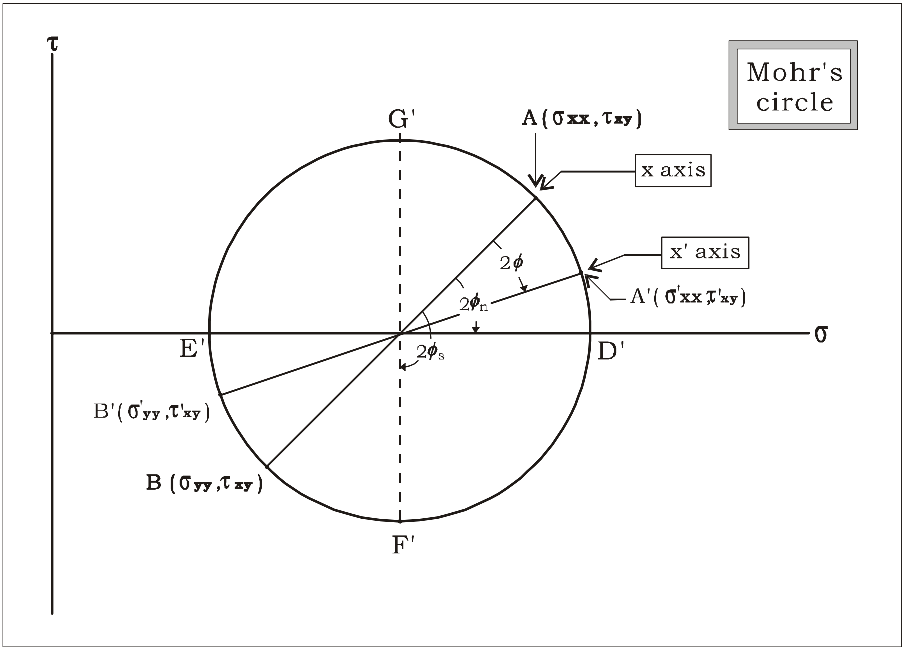
In this section we develop a program to produce, draw and label Mohr's circle for a given condition of two-dimensional stress. The left-hand side figure below shows the given state of stress in two-dimensions, with σ_{xx} and σ_{yy} being normal stresses, and $\tau_{xy} = \tau_{yx}$ being shear stresses. The right-hand side figure shows the state of stresses when the element is rotated by an angle ϕ. In this case, the normal stresses are σ'_{xx} and σ'_{yy}, while the shear stresses are τ'_{xy} and τ'_{yx}.



The relationship between the original state of stresses (σ_{xx}, σ_{yy}, τ_{xy}, τ_{yx}) and the state of stress when the axes are rotated counterclockwise by f (σ'_{xx}, σ'_{yy}, τ'_{xy}, τ'_{yx}), can be represented graphically by the construct shown in the figure below.

To construct Mohr's circle we use a Cartesian coordinate system with the x-axis corresponding to the normal stresses (σ), and the y-axis corresponding to the shear stresses (τ). Locate the points A(σ_{xx},τ_{xy}) and B (σ_{yy}, τ_{xy}), and draw the segment AB. The point C where the segment AB crosses the σ_n axis will be the center of the circle. Notice that the coordinates of point C are ($\frac{1}{2}\cdot(\sigma_{yy} + \sigma_{xy})$, 0). When constructing the circle by hand, you can use a compass to trace the circle since you know the location of the center C and of two points, A and B.

Let the segment AC represent the x-axis in the original state of stress. If you want to determine the state of stress for a set of axes x'-y', rotated counterclockwise by an angle ϕ with respect to the original set of axes x-y, draw segment A'B', centered at C and rotated clockwise by and angle 2ϕ with respect to segment AB. The coordinates of point A' will give the values (σ'_{xx},τ'_{xy}), while those of B' will give the values (σ'_{yy},τ'_{xy}).



The stress condition for which the shear stress, τ'_{xy}, is zero, indicated by segment D'E', produces the so-called *principal stresses*, σ^P_{xx} (at point D') and σ^P_{yy} (at point E'). To obtain the principal stresses you need to rotate the coordinate system x'-y' by an angle ϕ_n, counterclockwise, with respect to the system x-y. In Mohr's circle, the angle between segments AC and D'C measures $2\phi_n$.

The stress condition for which the shear stress, τ'_{xy}, is a maximum, is given by segment F'G'. Under such conditions both normal stresses, $\sigma'_{xx} = \sigma'_{yy}$, are equal. The angle corresponding to this rotation is ϕ_s. The angle between segment AC and segment F'C in the figure represents $2\phi_s$.

Modular programming

To develop the program that will plot Mohr's circle given a state of stress, we will use modular programming. Basically, this approach consists in decomposing the program into a number of sub-programs that are created as separate variables in the calculator. These sub-programs are then linked by a main program, that we will call *MOHRCIRCL*. We will first create a sub-directory called MOHRC within the HOME directory, and move into that directory to type the programs.

The next step is to create the main program and sub-programs within the sub-directory.

The main program MOHRCIRCL uses the following sub-programs:

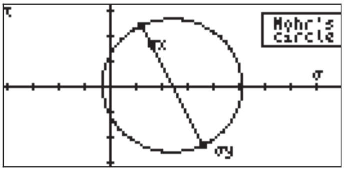
- INDAT: Requests input of σx, σy, τxy from user, produces a list σL = {σx, σy, τxy} as output.
- CC&r: Uses σL as input, produces σc = ½(σx+σy), r = radius of Mohr's circle, ϕn = angle for principal stresses, as output.
- DAXES: Uses σc and r as input, determines axes ranges and draws axes for the Mohr's circle construct
- PCIRC: Uses σc, r, and ϕn as input, draw's Mohr's circle by producing a PARAMETRIC plot
- DDIAM: Uses σL as input, draws the segment AB (see Mohr's circle figure above), joining the input data points in the Mohr's circle
- σLBL: Uses σL as input, places labels to identify points A and B with labels "σx" and "σy".
- σAXS: Places the labels "σ" and "τ" in the x- and y-axes, respectively.
- PTTL: Places the title "Mohr's circle" in the figure.

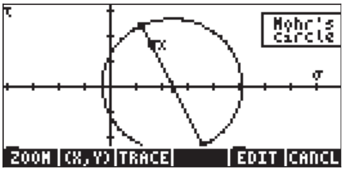
Running the program

If you typed the programs in the order shown above, you will have in your sub-directory MOHRC the following variables: PTTL, σAXS, PLPNT, σLBL, PPTS, DDIAM. Pressing NXT you find also: PCIRC, DAXES, ATN2, CC&r, INDAT, MOHRC. Before re-ordering the variables, run the program once by pressing the soft-key labeled MOHRC. Use the following:

MOHRC	Launches the main program MOHRCIRCL
2 5 ▼	Enter σx = 25
7 5 ▼	Enter σy = 75
5 0 ENTER	Enter τxy = 50, and finish data entry.

At this point the program MOHRCIRCL starts calling the sub-programs to produce the figure. Be patient. The resulting Mohr's circle will look as in the picture to the left.





Because this view of PICT is invoked through the function PVIEW, we cannot get any other information out of the plot besides the figure itself. To obtain additional information out of the Mohr's circle, end the program by pressing ON. Then, press ◀ to recover the contents of PICT in the graphics environment. The Mohr's circle now looks like the picture to the right (see above).

Press the soft-menu keys TRACE and (X,Y). At the bottom of the screen you will find the value of ϕ corresponding to the point A(σx, τxy), i.e., `φ = 0, (2.50E1, 5.00E1).`

Press the right-arrow key (▶) to increment the value of ϕ and see the corresponding value of (σ'_{xx}, τ'_{xy}). For example, for $\phi = 45°$, we have the values (σ'_{xx}, τ'_{xy}) = (1.00E2, 2.50E1) = (100, 25). The value of σ'_{yy} will be found at an angle 90° ahead, i.e., where ϕ = 45 + 90 = 135°. Press the ▶ key until reaching that value of ϕ, we find (σ'_{yy}, τ'_{xy}) = (-1.00E-10,-2.5E1) = (0, 25).

To find the principal normal values press ◀ until the cursor returns to the intersection of the circle with the positive section of the σ-axis. The values found at that point are ϕ = 59°, and (σ'_{xx}, τ'_{xy}) = (1.06E2,-1.40E0) = (106, -1.40). Now, we expected the value of $\tau'_{xy} = 0$ at the location of the principal axes. What happens is that, because we have limited the resolution on the independent variable to be $\Delta\phi$ = 1°, we miss the actual point where the shear stresses become zero. If you press ◀ once more, you find values of are ϕ = 58°, and (σ'_{xx}, τ'_{xy}) = (1.06E2,5.51E-1) = (106, 0.551). What this information tell us is that somewhere between $\phi = 58°$ and $\phi = 59°$, the shear stress, τ'_{xy}, becomes zero.

To find the actual value of ϕn, press ON. Then type the list corresponding to the values {σx σy τxy}, for this case, it will be `{ 25 75 50 } [ENTER]`

Then, press EXIT. The last result in the output, 58.2825255885°, is the actual value of ϕn.

A program to calculate principal stresses

The procedure followed above to calculate ϕn, can be programmed as follows:

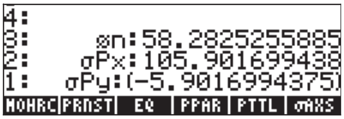
Program PRNST:

«	Start program PRNST (PRiNcipal STresses)
INDAT	Enter data as in program MOHRCIRC
CC&r	Calculate σc, r, and fn, as in MOHRCIRC
"ϕn" →TAG	Tag angle for principal stresses
3 ROLLD	Move tagged angle to level 3
R→C DUP	Convert σc and r to (σc, r), duplicate
C→R + "σPx" →TAG	Calculate principal stress σPx, tag it
SWAP C→R - "σPy" →TAG	Swap,calculate stress σPy, tag it.
»	End program PRNST

To run the program use:

VAR PRNST	Start program PRNST
2 5 ▼	Enter σx = 25
7 5 ▼	Enter σy = 75
5 0 ENTER	Enter τxy = 50, and finish data entry.

The result is:


```
4:
3:     øn:58.2825255885
2:   σPx:105.901699438
1:  σPy:(-5.9016994375)
MOHRC PRNST EQ PPAR PTTL σAXS
```

Ordering the variables in the sub-directory

Running the program MOHRCIRCL for the first time produced a couple of new variables, PPAR and EQ. These are the Plot PARameter and EQuation variables necessary to plot the circle. It is suggest that we re-order the variables in the sub-directory, so that the programs MOHRC and PRNST are the two first variables in the soft-menu key labels. This can be accomplished by creating the list { MOHRCIRCL PRNST } using: VAR ← {} MOHRC PRNST ENTER

And then, ordering the list by using: ← PRG MEM DIR ORDER.

After this call to the function ORDER is performed, press VAR. You will now see that we have the programs MOHRCIRCL and PRNST being the first two variables in the menu, as we expected.

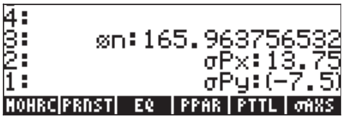
A second example of Mohr's circle calculations

Determine the principal stresses for the stress state defined by σ_{xx} = 12.5 kPa, σ_{yy} = -6.25 kPa, and τ_{xy} = - 5.0 kPa. Draw Mohr's circle, and determine from the figure the values of σ'_{xx}, σ'_{yy}, and τ'_{xy} if the angle ϕ = 35°.

To determine the principal stresses use the program PRNST, as follows:

VAR PRNST	Start program PRNST
1 2 . 5 ▼	Enter σx = 12.5
6 . 2 5 +/- ▼	Enter σy = -6.25
5 +/- ENTER	Enter τxy = -5, and finish data entry.

The result is:

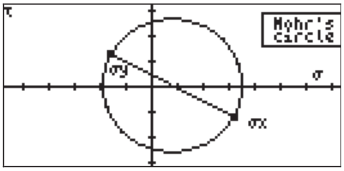


```
4:
3:     θn:165.963756532
2:            σPx:13.75
1:           σPy:(-7.5)
MOHRC PRNST EQ PPAR PTTL σAXS
```

To draw Mohr's circle, use the program MOHRC, as follows:

VAR MOHRC	Start program PRNST
1 2 . 5 ▼	Enter σx = 12.5
6 . 2 5 +/- ▼	Enter σy = -6.25
5 +/- ENTER	Enter τxy = -5, and finish data entry.

The result is:



To find the values of the stresses corresponding to a rotation of 35° in the angle of the stressed particle, we use:

ON ◀	Clear screen, show PICT in graphics screen
TRACE (X,Y).	To move cursor over the circle showing ϕ and (x,y)

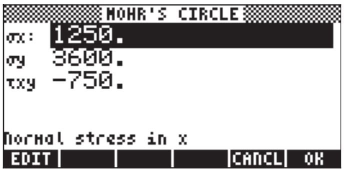
Next, press ▶ until you read ϕ = 35. The corresponding coordinates are (1.63E0, -1.05E1), i.e., at ϕ = 35°, σ'_{xx} = 1.63 kPa, and σ'_{yy} = -10.5kPa.

An input form for the Mohr's circle program

For a fancier way to input data, we can replace sub-program INDAT, with the following program that activates an input form:

```
« "MOHR'S CIRCLE" { { "σx:" "Normal stress in x" 0 } {
"σy:" "Normal stress in y" 0 } { "τxy:" "Shear stress" 0}
}   { } { 1 1 1 } { 1 1 1 }  INFORM DROP »
```

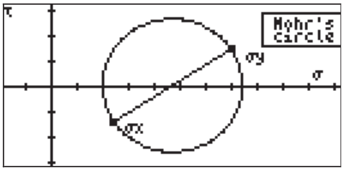
With this program substitution, running MOHRC will produce an input form as shown next:



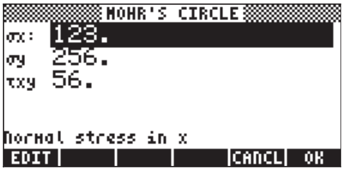
```
MOHR'S CIRCLE
σx: 1250.
σy  3600.
τxy -750.

Normal stress in x
EDIT |    |    |    |CANCL| OK
```

Press OK to continue program execution. The result is the following figure:



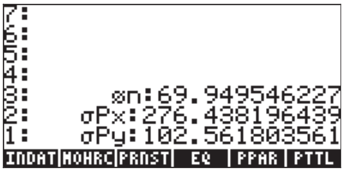
Since program INDAT is used also for program PRNST (PRiNcipal STresses), running that particular program will now use an input form, for example,



```
MOHR'S CIRCLE
σx: 123.
σy  256.
τxy 56.

Normal stress in x
EDIT |  |  |  |CANCL| OK
```

The result, after pressing OK, is the following:



```
7:
6:
5:
4:
3:      φn:69.949546227
2:    σPx:276.438196439
1:    σPy:102.561803561
INDAT|MOHRC|PRNST| EQ |PPAR|PTTL
```