

Polar1.bas

companion program to the article

[UNDERSTANDING AND PLOTTING POLAR COORDINATES](#) by *Tom Nally*

```
! *****
!
'Polar1.bas by Nally + April 2000
'Released as open source
!
'Polar1.bas plots the function defined
'by  $R = 2.5 + 2 \cdot \sin(12 \cdot \phi)$  using polar
'coordinates.
!
'To change the function plotted, locate
'the branch label [Plot.click]. The
'function is defined a few lines below
'that branch label.
!
! *****

pi = 3.141592
PlotType$ = "Markers"
MarkerValue$ = "set"
LineValue$ = "reset"
Scalefactor = 1
BiggestXY = 0
xmin = 0
xmax = 0
ymin = 0
ymax = 0
Dim R(360)
Dim x(360)
Dim y(360)
Dim xscaled(360)
Dim yscaled(360)
Dim xplot(360)
Dim yplot(360)
NoMainWin
WindowWidth = 440
WindowHeight = 265
Graphicbox #main.PlotArea, 214, 21, 200, 200
Button #main.Plot, "Plot !", [Plot.click], UL, 25, 126, 170, 30
Groupbox #main.groupbox4, "Plot Appearance", 26, 16, 168, 105
```

```
Radiobutton #main.radioMarkers, " Plot markers", [markers.Set], [
markers.Set], 38, 46, 112, 20
Radiobutton #main.radioLines, " Plot Lines", [Lines.Set], [Lines.
Set], 38, 71, 96, 20
Button #main.About, "About...", [About.click], UL, 30, 186, 74, 25
Button #main.Quit, "Quit", [Quit.click], UL, 118, 186, 72, 25
Open "Plotting Polar Coordinates" For Window As #main
Print #main.PlotArea, "fill white; flush"
Print #main, "font arial 10"
Print #main, "trapclose [Quit.click]"
GoSub [Print.Initial.Control.Values]
[main.inputLoop] 'wait here for input event
Wait
GoTo [main.inputLoop]
[Quit.click]
Close #main: End
[Print.Initial.Control.Values]
Print #main.radioMarkers, "Set"
Print #main.PlotArea, "fill white"
Return
[markers.Set]
Print #main.radioMarkers, "value? MarkerValue$"
If (MarkerValue$ = "set") Then
PlotType$ = "Markers"
End If
If (MarkerValue$ = "reset") Then
PlotType$ = "Lines"
End If
GoTo [main.inputLoop]
[Lines.Set]
Print #main.radioLines, "value? LineValue$"
If (LineValue$ = "set") Then
PlotType$ = "Lines"
End If
If (LineValue$ = "reset") Then
PlotType$ = "Markers"
End If
GoTo [main.inputLoop]
[Plot.click]
'for all angles between zero and 360 degrees,
'find the value of R as defined by
'R = 2.5 + 2*sin(12*phi)
For degree = 0 to 360
phi = (degree/360) *(2*pi)
R(degree) = 2.5 + 2*Sin(12*phi)
Next degree
```

```
'Convert polar coordinates to
'cartesian coordinates
For degree = 0 to 360
phi = (degree/360) * (2*pi)
x(degree) = R(degree) * Cos(phi)
y(degree) = R(degree) * Sin(phi)
Next degree
'Find the extreme values of all cartesian
'x and y values as a prelude to
'finding the scalefactor
xmax = 0
xmin = 0
ymax = 0
ymin = 0
For degree = 0 to 360
If (x(degree) > xmax) Then xmax = x(degree)
If (x(degree) < xmin) Then xmin = x(degree)
If (y(degree) > ymax) Then ymax = y(degree)
If (y(degree) < ymin) Then ymin = y(degree)
Next degree
BiggestXY = 0
If (abs(xmax) > BiggestXY) Then BiggestXY = abs(xmax)
If (abs(xmin) > BiggestXY) Then BiggestXY = abs(xmin)
If (abs(ymax) > BiggestXY) Then BiggestXY = abs(ymax)
If (abs(ymin) > BiggestXY) Then BiggestXY = abs(ymin)
'Establish the scalefactor
Scalefactor = 90 / BiggestXY
'Create a scaled set of the cartesian coordinates
'so that the data will plot nicely in the Graphicbox
For degree = 0 to 360
xscaled(degree) = Scalefactor * x(degree)
yscaled(degree) = Scalefactor * y(degree)
Next degree
'Shift the data set to the center of the
'Graphicbox, while also reversing the
'direction of y.
For degree = 0 to 360
xplot(degree) = 100 + xscaled(degree)
yplot(degree) = 100 - yscaled(degree)
Next degree
If (PlotType$ = "Lines") Then [Skip.To.Lines]
'If markers have been selected, plot markers
Print #main.PlotArea, "discard"
Print #main.PlotArea, "cls"
Print #main.PlotArea, "fill white"
Print #main.PlotArea, "down"
```

```
For degree = 0 to 360
Print #main.PlotArea, "place " + Str$(xplot(degree)) + " " + Str$(
yplot(degree))
Print #main.PlotArea, "circle 2"
Next degree
GoTo [Plotting.Accomplished]
[Skip.To.Lines]
'If lines have been selected, plot lines
Print #main.PlotArea, "discard"
Print #main.PlotArea, "cls"
Print #main.PlotArea, "fill white"
Print #main.PlotArea, "down"
For degree = 1 to 360
xplot1 = xplot(degree - 1)
yplot1 = yplot(degree - 1)
xplot2 = xplot(degree)
yplot2 = yplot(degree)
Print #main.PlotArea, "line " + Str$(xplot1) + " " + Str$(
yplot1) + " " + _
Str$(xplot2) + " " + Str$(yplot2)
Next degree
[Plotting.Accomplished]
Print #main.PlotArea, "flush"
GoTo [main.inputLoop]
[About.click]
Notice "About Plotting Polar Coordinates" + Chr$(13) + _
"By Nally + April 2000 " + Chr$(13) + _
"Made with Liberty Basic " + Chr$(13) + _
"Released as open source "
GoTo [main.inputLoop]
```