Two and Three Dimensional BASIC Graphics Programming and Its Uses

Umesh P. Nagarkatte and Shailaja U. Nagarkatte

<u>Introduction</u> Two dimensional BASIC graphics programming on IBM PC's or compatibles is extremely easy to use at any level starting from elementary algebra. It is flexible in the sense that it can be made as sophisticated as necessary depending on the student's mathematical background. The use of these programs requires no previous programming knowledge. It can be used to explain various difficult concepts in algebra, precalculus and calculus.

In elementary algebra, the concepts of consistency of systems of equations and in intermediate algebra, perpendicularity of lines, systems of linear inequalities in two variables can be explained using superimposing graphs. Students do not readily see the connection between graphs of functions and the real and imaginary roots of corresponding equations. These can be emphasized using graphics. Conic sections, and asymptotes to hyperbolas can also be easily handled. In precalculus, the concept of function - its domain, range and graph, the concepts of composite functions, inverse functions, graphs of rational functions, concepts of asymptotes, graphs of trigonometric functions and their inverses, effect of changing parameters on graphs of functions can be studied. In calculus, the concepts of limits, continuity and derivatives can be explained using graphing. The concept of Riemann Integral and areas between curves can be explained. Graphs of polar equations, which are time consuming to draw on the board, can be drawn very quickly. Taylor series approximation to functions can also be studied graphically. These activities can be handled effectively by using two dimensional graphics programming. In addition, derivatives can be applied as in shift operators to draw fast graphs of curves.

Students have a difficult time in visualizing three dimensional graphs. In this respect, a shell for three dimensional graphics based on easy BASIC instructions is helpful. Vectors, planes and surfaces, the tangent, gradient and normal to given surfaces can be explained using such a shell. But the ideas involved in 3D graphing are quite sophisticated.

<u>Implementation</u> We use 2D graphics programming in various classes as follows. Each student is provided a disk containing some DOS programs and a set of graphic programs. The instructor spends approximately 30 minutes in the micro computer lab explaining the procedure for using the graphics programs. Then lectures involving topics on graphing are conducted in the lab. As the instructor explains the concepts, the computers are used for reinforcement. Some exercises involving computer graphics are assigned in addition to regular exercises.

Quoted below are some of the actual assignments used in our classrooms. For more details see [8].

1. Given two lines with equations, $y = 2x - 3$ and $y = -x + 2$, complete a parallelogram with lines through (-5,-1). Students are provided a program containing two lines, which they modify to do this problem.

2. Given the program dealing with Riemann sums using the midpoint rule for $y = x^2 + 4$, write a program that draws proper rectangles and computes Riemann sums for $y = (\sin x)/x$.

Two Dimensional Graphics Programming These programs consist of simple instructions. They can be easily modified. They use built-in functions in IBM or GW BASICA. These programs can be made interactive. They can incorporate other ideas such as shift algorithms, which are applications of derivatives for fast graphing. Only five BASIC graphics instructions are required to draw a graph. These instructions are as follows:

1. SCREEN - for medium or high resolution, activating color if desired.
2. WINDOW - for bringing the origin of the screen which is located at the upper left corner to the center. The "third quadrant" is brought to the lower left part of screen. This function also takes care of "line clipping" to disallow overflow or wrapping that might occur. By varying the scale, one can vary the size of the window to "zoom and pan".
3. LINE - for drawing the axes and tick marks, or for joining two given points.
4. PSET - for plotting points.
5. PAINT - for painting regions which are solutions of inequalities or areas between curves. Choose an interior point of the region for this instruction.

A typical graphics program is given below.

```
10 REM Draw a graph of the equation x^2 + y^2 = 25
20 CLS:SCREEN 1,0         'Clear screen, set med.res.,color on
40 x = 20                 'Scale
50 WINDOW (-x,-x)-(x,x) 'Bring origin to center of screen
60 LINE (-x,0)-(x,0):LINE (0,-x)-(0,x)     'Draw x and y axes
70 LOCATE 1,20:PRINT "Y"                   'Name y-axis
80 LOCATE 14,20:PRINT "O"                  'Name the origin
90 LOCATE 14,35:PRINT "X"                  'Name x-axis
100 FOR x = -5 to 5 step .01
110      y = sqr(25 - x^2) 'Separate positive part
120      PSET (x,y)        'Plot the point
130      y = -sqr(25-x^2)  'Separate negative part
140      PSET (x,y)        'Plot the point
150 NEXT x
```

One can draw tick marks by inserting the instructions:

```
91  FOR t = -x+1 TO x: LINE (t,-.2)-LINE (t,.2):NEXT t
92  FOR t = -x+1 TO x: LINE (-.2,t)-LINE (.2,t):NEXT t
```

The above program draws an ellipse instead of a perfect circle. This distortion is due to the "aspect ratio" of the monitor screen. The aspect ratio is the ratio of the pixels along the horizontal axis to those along the vertical axis. Since this ratio is 4:3, the distortion could be corrected by using the instruction WINDOW (-4*x+1,-3*x+1)-(4*x,3*x).

Three Dimensional Graphing Three dimensional graphics programs use the same instructions as in two dimensions except that we replace WINDOW by WINDOW SCREEN. This instruction keeps the negative x and y coordinates in the upper left part of the screen. Three dimensional graphics consist of simulating the three dimensional objects on a two dimensional screen. Drawing a view of a surface depends on the location of the viewer. For this purpose it is convenient to take the origin at the viewing point of the viewer, the positive x-axis to the right, positive y-axis up and positive z-axis behind the viewer. The WINDOW SCREEN instruction facilitates this configuration.

The 3D graphics programming uses sophisticated ideas from projective geometry and linear algebra. The question which projective geometry directs itself is, "How do we see things?" or "What is the relation between a thing itself, and the thing seen?"[9] These problems are handled mathematically by using homogeneous coordinates. The regular 3D coordinates are transformed into view coordinates using rotations and translations and then into screen coordinates using projections. When we consider homogeneous coordinates, rotations and translations become linear transformations. Matrix multiplications play an important role in linear transformations. These transformations are discussed in [3,4,5,6,10].

Another difficulty is to remove the surfaces and lines which are hidden from view. Otherwise, the 3D effect is distorted. There are two types of methods for removing hidden surfaces and lines: Object-space methods based upon the object and image-space methods based upon the 2D image. Most books on graphics discuss the image-space methods. A failure proof method in hidden surface removal is the plane equation method, which is an object-space method. It uses calculation of the surface normal vector. The hidden surface routine expects the view coordinates to be provided in a consistent manner located counterclockwise around the outline of the surface. [2] When the surface normal points to the viewer, the surface is visible and when it points away from the viewer, the surface is hidden. In order to draw a visible surface, the area is cleared of all other graphics using "key matte approach". First, the required surface is drawn using an erasing color (cyan, for instance). Then the area is filled with the erasing color. Next, the outline is drawn in the correct drawing color (white, for instance). The area enclosed is filled with black. This is called the key matte apporach. Using this approach, the nearer surfaces will always cover surfaces which are further away, provided that the nearest surfaces are drawn last. [2] The authors are in the process of developing a shell of 3D graphing using the above ideas.

REFERENCES

1. Adams, J. Alan and Billow, Leon Descriptive Geometry and Geometric Modeling A Basis for Design: Holt, Reinhart and Winston, Inc., New York, NY (1988)

2. Adams, Lee High Performance Interactive Graphics: Tab Books, Blue Ridge Summit, PA 17214 (1987)

3. Artwick, Bruce Microcomputer Displays, Graphics, and Animation: Prentice-Hall, Inc., Englewood Cliffs, NJ 07632 (1985)

4. Mufti, Aftab A. Elementary Computer Graphics: Reston Publishing company, Reston, VA 22090 (1983)

5. Newman, William M. and Sproull, Robert F. Principles of Interactive Computer Graphics Second Edition: McGraw-Hill Book Company, New York, NY (1979)

6. Pavilidis, Theo Algorithms for Graphics and Image Processing: Computer Science Press, Rockville, MD 20850 (1982)

7. Rothenberg, Ronald I. Basic Computing for Calculus: McGraw-Hill Book Company, New York, NY (1985)

8. Sayrafiezadeh, Mahmoud Math Lab Manual for Graphics under preparation. Medgar Evers College, CUNY, Brooklyn, NY (1988)

9. Seidenberg, A. Lectures in Projective Geometry D. Van Nostrand Company, Inc., Princeton, NJ (Springer-Verlag, Inc., New York, NY) (1963)

10. Smith, David A. Three Dimensional Graphics: An Application of Linear Algebra An Apple Side Show Presentation Notes: Department of Mathematics, Duke University, Durham, NC (1984)

Medgar Evers College, CUNY
1650 Bedford Avenue, Brooklyn, NY 11225

Queensborough Community College, CUNY
56 Avenue and Springfield Blvd., Bayside, NY 11364