

# PHIGS PLUS For Scientific Graphics

**Roger A. Crawfis**

**Lawrence Livermore National Lab  
PO Box 808, L-301  
Livermore, CA 94550**

**This paper was prepared for  
the NCGA '91 Conference to  
be held April 22-25, 1991 at  
Chicago, Illinois**

**January 14, 1990**

# PHIGS PLUS For Scientific Graphics

**Roger A. Crawfis**

**Lawrence Livermore National Lab  
PO Box 808, L-301  
Livermore, CA 94550**

## **Abstract**

This paper gives a brief overview of the use of computer graphics standards in the scientific community. It particularly details how PHIGS PLUS meets the needs of users at the Lawrence Livermore National Laboratory. Although standards for computer graphics have improved substantially over the past decade, their acceptance in the scientific community has been slow. As the use and diversity of computers has increased, the scientific graphics libraries have not been able to keep pace with the additional capabilities these new machines offer. Therefore, several organizations have or are now working on converting their scientific libraries to rest upon a portable standard. This paper will address why this transition has been so slow and offer suggestions for future standards work to enhance scientific visualization. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

## **An Overview Of PHIGS PLUS**

PHIGS PLUS added several extensions to the standard PHIGS library that are of particular interest to the scientific community. This section will briefly go over what has been added to the PHIGS PLUS standard. Future sections will discuss how well these extensions can be applied to present scientific data.

## **Additional Output Primitives**

PHIGS PLUS adds a set of output primitives which contain as part of their specification nongeometric data: colors, normals, edge visibility flags, and application data. Rather than add an additional primitive that would parallel each PHIGS primitives to allow these data specifications, PHIGS PLUS chose the two most general PHIGS primitives for which this additional data applied and added the two new primitives POLYLINE SET 3 WITH DATA, and FILL AREA SET 3 WITH DATA. For area defining primitives (such as the FILL AREA SET 3 WITH DATA), data can be specified for the entire area (the facet), and at each individual vertex.

In addition to these new primitives, PHIGS PLUS adds five additional geometric primitives that can also specify this additional data. These allow the user to easily specify curves and surfaces. There are three surface mesh primitives that have been added: TRIANGLE STRIP 3 WITH DATA, QUADRILATERAL MESH 3 WITH DATA, and SET OF FILL AREA SET 3 WITH DATA. The triangle strip allows  $N-2$  triangles to be defined with only  $N$  points. The quadrilateral mesh primitive defines a curvilinear grid of quadrilaterals, hence  $(N-1)(M-1)$  quadrilaterals can be defined by  $NM$  points. The SET OF FILL AREA SET 3 WITH DATA (boy that's a mouthful) can be used to describe an arbitrary or polygonal mesh. Such a mesh would consist of a single list of vertices or points on the mesh, followed by a description of which points go together to form each zone within a mesh. The SET OF FILL AREA SET 3 WITH DATA can also be used to describe many other complex objects or shapes. Again, data can be specified for both the vertices of a mesh and for each zone or facet of a mesh. Two-dimensional interfaces for the triangle strip and the quadrilateral mesh have also been added into the proposed standard. This can result in a substantial memory savings for large finite element calculations.

PHIGS PLUS has also added a much needed curve primitive, NON-UNIFORM B\_SPLINE CURVE. However, B-Splines do not directly interpolate their data points (called knots in B-Spline terminology), and complex formulas for calculating a B-spline to interpolate a given set of points must be developed. A similar complex formulation can lead to approximating curves (in a least squares sense), which is not possible with normal interpolatory splines. These same statements can be said about the surface primitive, NON-UNIFORM B-SPLINE SURFACE. The surface also allows the user to define trimming curves that restrict the area of the surface to display.

## Direct Color Specifications

Another important enhancement that PHIGS PLUS adds to the PHIGS standard is the use of true or direct color. For most of us, this means 24-bit color, but the specification is general and not machine specific. Direct color is implemented by a general color specification that states both the color type or model and the corresponding color definition in that model. PHIGS PLUS uses the PHIGS color models (RGB, CIELUV, HSV, HLS) and the color type INDIRECT (for table lookups). Any colors specified by table indices are dereferenced (looked up) before the rendering pipeline (conceptually speaking of course). This implies that there is no direct relationship between the PHIGS PLUS color table and the hardware color table.

## The Rendering Pipeline

PHIGS PLUS adds a rendering pipeline that all primitives go through. However, all primitives are not affected by every stage of the pipeline. Lighting and shading only apply to the geometrically defined primitives. There are various stages of the pipeline, but the order of the stages is controlled by the user with a shading method. Each of these stages is described more fully below.

### Application Data Mapping Facilities

With PHIGS PLUS, users can specify any amount of scalar data values with a geometric primitive. One or two of these values can then be used to determine a color. Depending on the shading method, the data values will be mapped to colors at the vertices and the colors interpolated across the primitive, or the data values can be interpolated across the primitive and mapped to colors at each individual pixel. Data values can be associated with the zones or faces of the primitives as well as the vertices. The data mapping is specified by a list of (data value, general color) pairs. The data values of a primitive that lie in between the values in this mapping list are given the color associated with the first smaller data value. No interpolation is performed. Large tables (~256 entries) can be used to achieve smooth shading, whereas coarse tables will lead to color banded contours. Data values and data mapping can not be associated with the polyline primitive (as of this printing).

### Lighting

PHIGS PLUS correctly separates the calculation of the amount of light reflected back towards the user and the determination of where these complex calculations should be applied. This allows both Gouraud and Phong shading, as well as many other useful calculations. Lighting can be applied to all area defining primitives (for the purpose of this discussion, text is not an area defining primitive), including cell arrays. There are essentially two lighting components within PHIGS PLUS: lighting calculations and light sources. Several light sources can be defined. Ambient, directional (lights source at infinity), positional (point), and spot light sources can be defined. Several of each of these may be defined and the user has control over whether they are active or not (turned on or off). In addition to this, the user can control how lighting is applied to each primitive. Choices include no reflection calculations; ambient only; ambient and diffuse only; and ambient, diffuse and specular calculations.

### Shading

Where the lighting calculation and the data mapping are performed is controlled on a per primitive basis by selecting a shading method. Shading methods of NONE, COLOUR, DOT, and NORMAL are defined. NONE implies flat shading and the lighting and data mapping are performed for each facet. COLOUR implies that the lighting and data mapping are performed at each vertex of a facet and the resultant colors are then interpolated across the primitive. DOT is defined as a faster alternative to NORMAL. It performs part of the lighting calculation at each vertex (namely that of calculating the dot products in the lighting equation), and then interpolates these dot product values (as well as the data mapping values) across the primitive. Data mapping and the rest of the lighting calculation is then performed at each pixel. NORMAL is used to achieve Phong shading. The normals are interpolated across the primitive and the lighting calculation is performed at each pixel. While

PHIGS PLUS can not specify the implementation of the lighting calculation, it does offer a suggested lighting equation in the appendix.

### **Depth Cueing**

Depth Cueing is a technique to attenuate the amount of light emitted from objects as they move farther away from the eye. Hence, it provides another cue to the actual 3D nature of an object. It provides a relatively cheap alternative to hidden surface processing on some workstations, and provides a little more realistic view of objects in general.

### **Attributes for both Front and Back Faces of Area Primitives**

PHIGS PLUS distinguishes between the front face and the back face of a polygon (or mesh or surface). Thus, users can specify different colors or patterns for each surface, as well as lighting and shading methods. This mirrors well a piece of paper that may have two totally different appearances on each side.

## **PHIGS PLUS Applied to Scientific Graphics**

While PHIGS PLUS does not immediately offer the functionality desired for scientific graphics, it does offer a good base on which to build the functionality needed. The incorporation of quadrilateral meshes and data mapping helps solve some immediate needs. Currently, shaded color contours are created with user written scan conversion software and displayed as a raster image or using a cell array. Thus, not only did the user need to write the scan conversion, but if the user wanted to look at the mesh from different views (such as for a stack of 2D contours, or a contour shaded surface plot), all of the transformation code also had to be written by the user. PHIGS PLUS allows us to generate these types of images directly with data formats familiar to scientists and engineers.

The use of data mapping can allow us to set up a surface and associate all of our application data with it. The user can then change between different data mappings to see different application variables. One drawback is that for smooth shading, a shading method of DOT or NORMAL is required. Since these are closely coupled with more sophisticated lighting stages, an expensive lighting calculation may be required to achieve the data mapping. Also, the data mapping in PHIGS PLUS only maps data to colors. Standard techniques such as mapping an application variable to a surface height are not handled. The user needs to construct a surface (quadrilateral mesh 3) for each application variable that a surface plot is needed. Furthermore, other mappings are possible, such as the mapping of vector fields to normals, and other scalar variables to textures. The data mapping in PHIGS PLUS does however fill a very important void for scientific graphics.

Another area where PHIGS PLUS greatly simplifies the creation of scientific graphics is in the lighting and shading. Surfaces plots and renderings of three-dimensional objects which are generated using light sources and smooth shading can be far more informative than black and white wireframe drawings. The task of adding light sources and doing the 3D scan conversion and shading in existing scientific graphics libraries has been too overwhelming for most developers of such libraries. As three-dimensional simulations become more common, the need for these types of images has grown. The current use of 3D isocontour surfaces which can generate very nonintuitive and *busy* 3D objects simply can not succeed without lighting and shading. Currently however, one finds that most of the packages that generate these images exist as separate entities from the mainstream graphics library of the particular research lab. PHIGS PLUS promises to allow these to be integrated with the existing scientific graphics.

The SET OF FILL AREA SET 3 WITH DATA primitive can be used for many 2D finite element mesh problems. While the quadrilateral mesh handles many 2D finite element meshes resulting from fluid flow and electrostatics, there is a large class of problems that require very irregular mesh structures. This new primitive can be used for these irregular meshes. However, the current PHIGS PLUS specification lacks a 2D interface for this primitive so the user needs to create new data structures in order to use it with their existing 2D codes. For surface plots, this is needed anyway, but it may be more efficient or easier to simply pass a collection of disjoint polygons.

The curve primitive in PHIGS PLUS is more specific to computer-aided design than it is to general scientific applications. Curves are typically used in xy-plots and contour plots to show the data in a smoother fashion than a sequence of line segments. A curve or spline usually has a tension associated with it in scientific graphics that controls how closely it follows a linear spline to a set of points. The B-spline is very general and pow-

erful. In order to control it effectively for scientific problems, much work is needed to develop efficient methods of generating interpolating and approximating B-splines. It is not clear whether the typical user will choose to do this or to use an existing spline package and output a very high resolution polyline. There may also be severe computational disadvantages to using a B-spline for this purpose. Further work in this area is needed.

The NURB primitive in PHIGS PLUS has the same advantages/disadvantages as the B-spline, only the complexity of a surface makes them even harder to control. This, plus the fact that smoothed interpolatory surfaces are not used as much as curves, will render the NURB ineffective for many years in scientific computing.

Other advantages of PHIGS PLUS in scientific graphics include the ability to render large 3D objects or surfaces quickly with depth cueing, and the ability to use different colors for the backside of a surface. With depth cueing, a user can interactively manipulate a large model to a position where a better rendering is desired. The use of different backfacing attributes provides yet another clue as to the 3D nature of a surface. By changing the color or data mapping on the back side, the user can more readily determine whether the surface is viewed from underneath it or from above it.

## **Future Directions for Scientific Graphics and Standards**

While the above sections reviewed the functionality of PHIGS PLUS and how it can be applied to scientific graphics, there are still some areas of scientific graphics that are very complex and for which PHIGS PLUS offers little support. This is not due to an oversight on PHIGS PLUS, but due to some specialized scientific requirements for which PHIGS PLUS was not chartered to do, but still have a profound impact on its incorporation into the scientific community. The first of these is the use of axes in scientific graphics. High quality efficient contouring algorithms with nice labelled contours is another. Other components that provide a base for scientific graphics are vector plots, legends for contour plots, and logarithmic scales (or grids). Ribbons and streamlines are also heavily used in computational fluid dynamics.

The big thrust today in scientific visualization is for pretty color or 3D pictures. While this allows the scientist to get a qualitative or overall view of the data, the scientist typically needs to determine more quantitative information about a problem. Simple things like xy-plots can provide this. While these plots in and of themselves may not be difficult, the key element to making sense of such a plot is the display of axes or grids. While to the average user an axis may look like a trivial collection of lines and text, it typically comprises more coding than any other graphical element in both its implementation and its use. This is mainly due to the various controls or options for producing it. Typical scientific graphics libraries have attributes for a 2D axis such as: line style, width and color for both the main axis line and any tick marks, the style of the tick marks (none, in, out, cross), the length of the tick marks (in screen space), the number of major and minor tick marks, the angle of the tick marks, text attributes for the labels and for the legend or title, control over how the labels are generated, and attributes to describe how to format the numbers into text (scientific, integer, real, etc.). Add to this the fact that a 2D grid typically has four axes surrounding the plot, and additional attributes for a grid control major and minor grid lines, and one can see the complexity of generating axes.

The big problem with generating 3D scientific graphics is not so much the representation of the data, but the generation of a 3D grid (which is now six times as complicated as a 2D grid). Just as hidden surfaces depth cueing, and lighting and shading provide clues to help us understand the 3D nature of an object, axes and grids provide us with clues as to the extent and meaning of the data that is being represented. The generation of these 3D axes will be a critical step in making 3D scientific graphics more useful. The difficulty in developing these axes is also one of the reasons why most public domain scientific graphics libraries offer only limited 3D functionality.

While the generation of isocontour lines can be thought of as a purely mathematical operation disjoint from any graphics system, the rendering of these isocontour lines requires some specialized functionality. With several isocontour lines on a plot, some mechanism to apply the quantitative values associated with each contour line is needed (i.e., what is the contour value?). Using different line styles or colors for each contour line and plotting a legend off to the side is one method that is easily doable on top of PHIGS PLUS. Several scientists prefer to have labels plotted directly with the contour lines that state explicitly what the contour value is. The placement of these labels, their orientation and how they follow the path of the contour are all issues that need to be addressed when designing a scientific graphics system. Furthermore, since the label lies on the contour line, the intersection of the text (as it is plotted in screen space) with the contour line must be calculated and

the contour line divided up into pieces with holes for the plotting of the labels. Since we can place contour lines on the surface of a 3D object, these calculations need to be performed in three-space as well.

All of this functionality is possible on top of PHIGS in a portable standards based fashion. The big question now is whether the investment to write a new 3D library on top of PHIGS PLUS is worth the cost, or whether porting an existing scientific graphics library to each machine is sufficient. Perhaps a push for an ANSI scientific graphics standard is the next logical step for computer graphics standards. One must note however that the development of such a library only provides a base of functionality and not a true scientific visualization system. Other issues such as interactivity, ease-of-use, database management, and data filters must be taken into account. Systems such as AVS, apE, and PV~Wave take into account some of the more graphical issues, but still lie disjoint from the systems developed over the years to deal with a specific users database and data formats.

## **Concluding Remarks**

The need for more advanced color and 3D representations of scientific data has made the use of PHIGS PLUS attractive to the scientific community. The anticipated arrival of PEX and its relationship to PHIGS PLUS provide even more fuel to the fire. Distributed scientific graphics displayed in a window of a workstation across a network is a vision that will become commonplace over the next several years. Perhaps one stumbling block to this vision is the lack of transparency (or opacity) in the standard. The power and use of transparency is obvious from results of the scientific visualization community. However, the use and control over it is also just being determined, and hence not ready for formally placing in the PHIGS PLUS standard. Regardless, the potential use of PHIGS PLUS in scientific graphics is still very high.

Computer graphics standards have come a long way, and can now handle many of the scientific graphics representations needed for day to day use. The advent of PHIGS PLUS and the expected arrival of PEX will greatly enhance the distribution and movement of scientific graphics. But as scientific graphics tries to jump on the standard's bandwagon, scientific visualization is rocketing the sophistication and complexity of scientific graphics beyond the capabilities of the standards and relying on the cutting edge of computer graphics research. Will the long turn around time of graphics standards lose the scientific community again? Only time will tell.