3D VIEW

Sini Varghese [#1] , Dr. Jagathy Raj. V. P [#2]

#1 Anna University M.Tech By Research Student,Kerala,India

[#2] Professor,CUSAT,Kerala,India

Abstract

3D-view is planned to be a software tool to analyse, understand and extract information from the flow field data computed using computational fluid dynamics technique. It is a tool to view computational grid used to compute the flow field, to explore complex data sets, arrange multiple 2-D and 3-D plots and to display the results with a high resolution output. This software has a host of features like contour plots, lighting effects, color mapping etc .It allows 3 dimensional transformations such as rotation, mirror images and scaling.3D-view also has the provision for saving and restoring the objects created. This software is implemented in the Linux platform using the graphics library glib and other tools developed over 'glib'. In short it is an interactive graphics tool with good GUI. 3D-view has immense applications in the field of computational fluid dynamics. This software finds much use in various areas of technology such as aerodynamics, automobile, marine, education and hardware industry.

Keywords

Computational Fluid Dynamics**,** Requirement analysis for 3D View, Feasibility Study of 3D View ,Design, Implementation details, Testing, Working Environment.

1. Introduction.

In the recent past ,there has been a tremendous drop in the cost per unit of computing power and storage and this has led to a large increase in the size and computational complexity of CFD computations. This result in the production of a large volume of data that is difficult to process by traditional means. It is obvious that the graphical visualization such CFD data is a necessity for both the developer of CFD code and also for researcher to develop a good CFD visualization tool. This is because 3D data visualization is nontrival. Also the developing tool such that it works across different operating systems and hardware is not easy and takes experience and effort.CFD researchers are also better served if they can concentrate on their CFD related work and not to have to worry.CFD always starts with a grid structure. The process of dividing the flow domain into a series of cells is generally called mesh or grid structure. Grid generation is the most expensive part of CFD analysis. Also grid quality strongly influences solution accuracy and convergence. Most commercial CFD programs will have a built-in grid generator. However separate grid generator programs can also be purchased. CFD data are typically in terms of m,x,y,z and t where m can be pressure, density, temperature etc. The results are generally visualized in a 3D world. Visualization objects are generated to trace flows, reveal complex or intricate flow details and map computed values. Color maps and surfaces keyed to quantitative values present analytical information.CFD computes huge amount of data, in the range of hundreds of mega bytes, hence graphical display is an important part of computational fluid dynamics. It gives the user an idea about the shape of the body, its structure, the quality of output obtained from the computational processes etc.

2. Computational Fluid Dynamics

Computational fluid dynamics(CFD) once the preserve of the academic or the specialist within large organization is a tool which is becoming increasingly available and which is finding growth applications in many industries. The most fundamental consideration in CFD is how one treats a continuous fluid in a discretized fashion on a computer. One method is to discretize the spatial domain into small cells to form a volume mesh or grid, and then apply a suitable algorithm to

solve the equations of motion. With the advent of modern computers, computational fluid dynamics evolved from potential-flow and boundary-layer methods and is now used in many diverse fields including engineering, physics, chemistry, meteorology, and geology. The crucial elements of computational fluid dynamics are discretization, grid generation and coordinate transformation, solution of the coupled algebraic equations, turbulence modelling and visualization. The continuous nature of the equations in a discrete form represents the numerical solutions of partial differential equations. The equations are discretized by subdividing the domain into cells or elements and expressing the equations in a discrete form at each point in the grid by using finite difference, volume or element method. A structured grid arrangement is used in the finite difference method while the other 2 methods are more flexible and adjusts with both structured and unstructured grids. There are 3 common cell integration techniques used are Finite Volume, Finite Element and Finite Difference.

2.1.Grid generation and mesh generation.

Grid generation is the most expensive part of CFD analysis. Also grid quality strongly influences solution accuracy and convergence.CFD data are transformable into 3D workspace. The results are easily visualized in a 3D world. Visualization objects are generated to trace flows, reveal complex or intricate flow details and map computed values.CFD grid are classified into Structured Cartesian, Unstructured curvilinear and Unstructured. The 3 specialized grid generating strategies are :

2.1.1.Unstructured grid generating strategies with trimmed cells
2.1.2.Structured curvilinear/ unstructured.
2.1.3.Chimera

2.2. Quality.

Mesh features.
2.2.1.Density-generally for high accuracy.
2.2.2.Smoothness-to avoid diffusion and dispersion errors, adjacent cell size changes should be small.
2.2.3.Structure-various cell shapes can map the region.
2.2.4.Storage locations-variables can be stored at cell centers or corners.

2.3. Various processes involved in CFD analysis

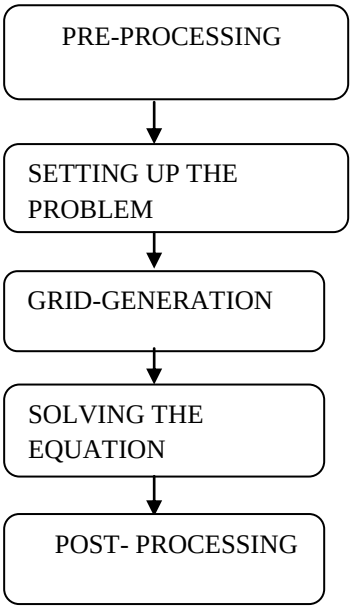


Figure 2.1 Various processes

CFD processing involves the following 5 steps.

2.3.1 Pre-processing.

All the task that takes place before the numerical solution process is started are called pre-processing. This is mainly include the analysis process. A mathematical analysis of fluid flow can be made and it leads to a series of partial differential equations that govern the flow. These partial differential equations can be discretized to produce numerical analogue of the equations.

2.3.2 Setting up the problem.

This stage involves the following operations.
2.3.2.1. Define a grid of points and perhaps volumes or elements.
2.3.2.2. Define the boundary of the computational domain.
2.3.2.3. Specify the boundary conditions.
2.3.2.4. Specify the initial conditions.
2.3.2.5. Set the fluid properties.
2.3.2.6. Set the numerical parameters.

2.3.3. Grid generation

The most difficult task in the processing phase is the generation of the grid points or mesh. this task can be simplified by using software that has been especially designed to carry out mesh generation. Such software is commonly available and can be interface with computer aided design systems.

2.3.4. Solving the equations

Each package has a program that solves the numerical equations for the problem under consideration. this program must be given all the relevant data that has been defined by the processor. To transfer the data between programs, the pre-processor writes out the data files that the solver program can read. These files can also be moved, if necessary between computers. This interactive is extremely useful as it means that the solver program can run on a machine specifically designed for a high speed numerical work such as super computer, while the interactive task are carried out on a small machine. Once the data files are in place , the solver program is activated and the required solution process is carried out. Although the solver program is the core of any CFD software system, the user sees little of its operation.

2.3.5.Post -processing

The post processing program is used to display the results, and as with the pre-processor ,this program is interactive and so usually run on the same machine as the pre-processor. By post processing we obtain both qualitative and quantitative results. Typical pictures obtained with the post processor might contain a section of the mesh together with vector plots of the velocity field or contour plots of scalar variables such as pressure, density tec. These pictures enable global trends in the data to be seen.

3. Requirement analysis for 3D View

The size of the flow field data generated is so huge that is impossible to go to each data point and study its values. In short , physically it is impossible to look at the digital data and grasp its meaning. That is why it is better to graphically represent the data. So we require a post processor which will show the results in a comprehensible manner .During my interaction with ISRO Engineers that what they required was a computer tool for graphical visualization of surface and flow field distribution of parameters such as pressure, temperature, density etc. Also there was the necessity to subject the 3D object to geometric transformations and thus view it from different angles.

4. Feasibility Study of 3D View.

Feasibility study involves the viability of the software to be developed. There are 3 Kinds of feasibility namely,

4.1.Technical Feasibilty-3D View is compatible with all versions of Linux and so technically feasible.

4.2.Economical Feasibility- To be economically feasible , the proposed system should be cost effective in nature.'3D View' runs on a free Operating system. The X window interface needed for it is also free. The number of computers , accessories as well as the software and hardware needed must be minimal and least expensive. Thus we can say that the benefit/cost ratio of 3D View is very high.

4.3. Operational Feasibility-Operational feasibility indicates the ease with which a user can interact with the software system.3D View features a very attractive graphical user interface(GUI). Easy to use buttons and pop up menus are provided for it using the IDE(Integrated Development Environment). Thus even a non technical user will not find it hard to operate with 3D View and to analyse its outputs. Hence it is operationally feasible

5. Design.

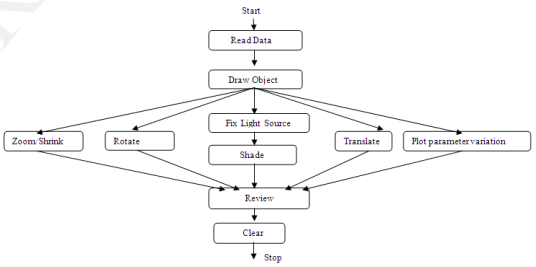


Figure 5.1.Flow Diagram.

The flow diagram indicates that the control flow between the different modules in the program. The modules are represented by rounded rectangles and the arrows indicate the direction of flow. In 3D View since the first step is reading the flow field data, the execution of the program must begin with it. Then after drawing the 3D image the user has several options of either subjecting the image to geometric transformations or to view the shading effect due to a light source. There are separate procedures for each of these features and also for representing the surface distribution of various parameters. The Review module enables the user to see the original image after saving modifications while Clear module wipes the view port and the screen clean

6. Platform, Language And Libraries

6.1. Linux.

Linux is an open source operating system and can be freely downloaded from internet sites. It is a fast growing operating system and it is inexpensive and flexible. Linux is also a major player in server field. It is a rich and powerful platform. It provides unsurpassed computing power, portability and stability. Linux provides connectivity to Microsoft, Novel and Apple proprietary networking .Thousands of free applets, tools and smaller programs are also available for it. Moreover Linux has a free X window graphical user interface which allows the most X based programs to run Linux without any modification. Linux supports all of the most common internet protocols including Electronic mail telnet, Web, FTP, POP,DNS, NIS and many more.

6.2. Language

The C programming language is chosen for implementing 3D View.

6.3. Graphics Library

The various powerful graphics libraries use in Linux are OpenGL, Motif, glade and Tk. The most commonly used library is OpenGL.OpenGL is the most widely adopted, cross-platform standard for 3D rendering and 3D hardware acceleration. GLIB is a powerful library , developed by VSSC.GLIB is a software interface to graphics hardware.CFD applications that involve massive amount of data can be manipulated using GLIB. Clipping, transformations and rendering in GLIB makes 3D effects possible. GLIB applications are portable and stable. It supports hard copying. GLIB is a visualization library written in C.

6.4. User Interface.

Creating complex user interface is fairly complex and a visual programming is very helpful in such situations. Makegui is a fully menu driven graphical tool to create complex dialog boxes with all types of controls. Makegui provides many items for each control and to reposition them.

7. Implementation Details.

7.1. Reading Flow Field Data

Format of the data file:
No. of blocks in the 3D object
No. of points in each block in X,Y and Z direction.
X, Y and Z coordinates of each point.
Temperature, pressure, density values at each point.

7.1.1. A structure is defined for storing the data read.(each block of data read).It has field for storing the no. of X, Y and Z coordinates as well as pointers for storing coordinate value as well as the pressure ,density and temperature of each point.

7.1.2. The data file is opened in read mode using the file pointer.

7.1.3. The no. of blocks of data in the data file is read and memory is accordingly allocated.

7.1.4. The no. of X, Y and Z coordinates as well as the values of these coordinates in each of these coordinates in each of the points in the block is read.

7.1.5. The differing values of pressure, density and temperature in each point are also read.

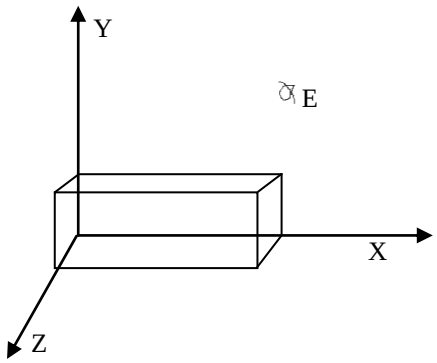
7.1.6. The maximum and minimum values in all 3 directions are calculated.

7.2. Drawing the 3d Object.

One of the main objectives of computer graphics is to represent and visualize objects and to analyse the characteristics of the objects and are defined with respect to some coordinate system. In computer graphics, objects are defined in either 2D(planar) or 3D Cartesian coordinate space. The 3D space is represented by [x,y,z].It can be right handed or left handed space. The drawing feature of the post processor involves the following steps.

7.2.1. Determining the co-ordinate system to use.

The right handed co-ordinate system is used in '3D View' since most scientific applications use the right handed system



7.2.2.Setting up the view port.
Using appropriate functions the extremities of the 3D object is calculated and then with these as limits the area for viewing the object is set up.

7.2.3.Determining the eye position and target position.
An eye position is set up as shown in above figure. The position for viewing the object decides its magnitude and resolution.

7.2.4. Plotting.

The data for X, Y and Z coordinates of each pointing the 3D object read from the flow field data file are plotted along the 3 axes inside the view port.

7.3. Transforming The 3d Object.

In many applications, there is a need for altering or manipulating the objects in a scene. Sometime we need to reduce the size of an object or we may want to rearrange the objects in the model. Geometric transformations are the procedures for carrying out these manipulations. The 3D transformations method include the z coordinate.

7.3.1. Translation

A point[x,y,z] in 3D is translated by adding a displacement vector[dx,dy,dz]

x' = x + dx
y' = y + dy
z' = z + dz

7.3.2.Scaling

A point [x,y,z] is scaled by multiplying the coordinates of point with the scaling parameters [Sx, Sy, Sz]

x' = x * Sx
y' = y * Sy
z' = z * Sz

7.3.3.Rotation

Rotation is specified by the axis of rotation about which the object is to be rotated

About X axis:

x' = x
y' = y *cosΘ - z * sinΘ
z' = y * sinΘ + z * cosΘ

About Y axis:

x' = z * sinΘ + x * cosΘ
y' = y
z' = z * cosΘ – x * sinΘ

About Z axis:

x' = x * cosΘ - y * sinΘ
y' = x * sinΘ + y * cosΘ
z' = z

7.4 Matrix Representation And Homogeneous Coordinates

Any picture or model typically requires a series of transformations to be applied to its shapes. The efficient approach is to represent the transformations in a matrix form and concatenate the sequence of transformations. Matrix representations are standard methods for implementing transformations in graphics system. Matrix representation for translation, scaling and rotation about X are given below.

$$[x' \; y' \; z'] = [x \; y \; z] + [dx \; dy \; dz]$$

$$[x' \; y' \; z'] = [x \; y \; z] * \begin{pmatrix} Sx & 0 & 1 \\ 0 & Sy & 0 \\ 0 & 0 & Sz \end{pmatrix}$$

$$[x' \; y' \; z'] = [x \; y \; z] * \begin{pmatrix} Cos\Theta & -\sin\Theta & 0 \\ Sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

And rotation involve matrix multiplication. Therefore inorder to avoid ambiguity and for a uniform representation, homogeneous coordinate system is used. By representing the points using homogeneous coordinates, all 3D transformations can be represented as 4x4 matrices.

Translation Matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{pmatrix}$$

Scaling Matrix:

$$\begin{pmatrix} Sx & 0 & 0 & 0 \\ 1 & Sy & 0 & 0 \\ 1 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation about X axis:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\Theta & \sin\Theta & 0 \\ 0 & -\sin\Theta & \cos\Theta & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

7.5. Shading

Gouraud shading is used to simulate the effects of light across the surface of a 3D object. Gouraud shading is used to achieve smooth lighting on the 3D object while avoiding the computational requirement of calculating lighting separately for each pixel.

7.5.1. Basic principle

Color intensities at the vertices of the polygon are determined based on the position of the light source. The pixel intensities of remaining points in the polygon are linearly interpolated from the intensities of vertices. i.e. a gradient is formed in the area. Gouraud shading's strength as well as its weakness is its interpolation. Gouraud shading works optimally for 3 or 4 sided polygons. Steps in Gouraud shading are:-

1. Determine the average unit normal vector at each polygon vertex.
2. Apply an illumination model at each vertex to calculate vertex intensity.
3. Linearly interpolate the vertex intensities over the surface of the polygon.

Consider the polygon

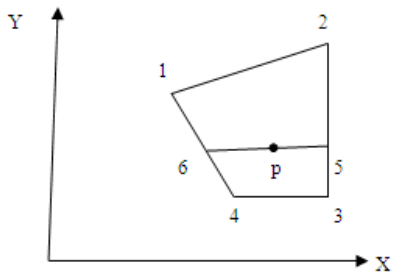


Figure 7.5.1-Polygon

In the scan line '6-5', the vertex intensity of the scan line at the point of intersection of the scan line and the polygonis obtained by

$$15 = \frac{(y5-y3)}{(y2-y3)} *12 + \frac{(y2-y5)}{(y2-y3)} * 13$$

Once the intensities of end points of a scan line are obtained, the intensities of remaining points along the line can be interpolated from them. For the point P,

$$lp = \frac{(xp - x5)}{(x6 - x5)} *16 + \frac{(x6 - xp)}{(x6 - x5)} * 15$$

Gouraud shading removes intensity discontinuities associated with flat shading. Highlights on the surface are displayed with anomalous shapes at times. Moreover linear interpolation may cause bright and dark intensity streaks called Mach bands. These effects can be reduced and improved shading can be obtained by dividing surface into larger no. of polygons.

8. Testing.

Software testing is the process of checking whether the developed system is working according to the original objectives and requirements. It commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Software testing is a critical element of software quality assurance and represent the ultimate review of specific design and coding. All test can be planned and designed before any code has been generated. To be most effective, we mean the testing that has the highest probability of finding errors. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. System testing makes a logical assumption that if all parts of the system are correct the goal will be successfully achieved. Thus a series of test were performed for the proposed system before the system is ready for implementation. In my work, 3D View, Testing was done by visualizing the CFD data on a graphical screen. Various data were given as input and the results were verified by visualizing the output. Complexity of the geometry is not a factor because , once it works for a simple geometry , it will obviously work for complex geometry. Format of the data will not change under geometric shapes, so testing in this case is simple. Tested 3D View for multiple spheres, single sphere, ring and V-plane data.

9. Working Environment

Aerodynamics Research and Dynamic of VSSC has initiated the Linux cluster concept for high speed in 1998.This was necessitated by the huge computing requirements for the numerical flow simulation. Costly supercomputers projects at national level started during this period failed to meet these requirements. Freely available Linux operating system and Message Passing Interface for parallel node communication were used. The present high availability second generation Linux cluster consist of off-the-shelf available 192 Athlon AMD XP 3200 processors as the fundamental nodes and these are connected through two 96 port 100Mbps switch. The processors boards are assembled in vertical racks with proper thermal management. The system has 5 file servers and these are provided with 1Gbps links. The Distributed Re-locatable Block Devices (DRBD) technology has been used for fail safe operation of the servers. This system has very attractive cost to speed ratio.

Conclusion.

A Software product for CFD post-processing has been discussed in my report. **'3D View'** makes it easy for a user to visualize the huge amount of calculations and the data involved with Computational Fluid Dynamics. It gives the user an idea about the shape of the body , its structure, the quality of output obtained from the computational processes. The importance of the tool from the perspective of the CFD visualization is been explained. Also the different features of this CFD post processor have been covered. It has got immense applications in the field of computational fluid dynamics. It provides an alternative way to visualize the complex CFD data. It finds much use in various areas of technology like aerodynamics, automobile, marine and educational purposes. The electronics and hardware industry also needs its functionalities. Due to lack of time, only a certain set of features have been implemented. Advanced features and utilities like particle traces, streamlines, flow animation, turbulent and laminar flows are to be implemented in its coming versions.

References.

[1]. Introduction to Grids and meshes by Paul Tucker and Antony Mosquera-NAFEEMS publications.
[2]. Graphics in C by Yashavant Kanetkar-BPB publications.
[3]. Development of Grid , portable graphics lib by Thomas C Babu, Aerodynamics division VSSC.
[4]. Journal of Aeronautical Society of India, volume 53, Issue January 2007 by Prabhu Ramachandran, M Ramakrishnan and S C Rajan.
[5]. Executing and observing CFD applications on the grid by Jan Wendler, Volume 21-issue1 January 2005 pages 11-18.
[6]. Computer Graphics by Donald Hearn and M.Pauline Baker,2nd edition, 1994-Prentice Hall of India Private Limited.
7. SIGGRAPH 1989,Volume 23-16th annual conference on Computer Graphics and Interactive Techniques,Boston Massachusetts Papers.
[8]. X Window inside and out by Levi Reiss & Joseph Radin,Paperback edition 1992-McGraw Hill inc publication.
9. Websites-www.opendx.com, www.flow3d.com, www.flowsimulations.com, www.opengl.com, www.mesa3d.org, www.linuxgazette.com