Virtual Environments – Improving the Understanding of Engineering Models

J Leng[1], L Margetts[2], Nigel W John[1], G W Leaver[1], P G Lever[1]
[1] *Manchester Visualization Centre, University of Manchester*
[2]*Manchester Computing, University of Manchester*
(currently at the Universidad Politecnica de Madrid, Spain)

j.leng@man.ac.uk, *lee.margetts@man.ac.uk*, *n.w.john@man.ac.uk*, *george.leaver@man.ac.uk*, *paul.lever@man.ac.uk*

Abstract

Research currently being carried out at the Manchester Visualization Centre (MVC) in collaboration with the Centre for Civil and Construction Engineering, UMIST, gives an insight into the limitations and problems engineers have understanding their models. The Engineers use finite element (FE) models in their design process. We show how analysis and validation of two models could not have been achieved without using the virtual environment developed at MVC.

The first model was of a standard CFD problem known as the lid-driven cavity. The second was of a magnetohydrodynamics code which has two completely different but coupled physical processes, a magnetic field and the flow of an electrically conductive fluid.

We are in the process of developing a number of tools that will improve analysis for engineers and more generally the virtual environment at Manchester. Tools are being developed in AVS/Express to aid engineers in the analysis of their data while at the same time Manchester is involved in several projects that improve the use of AVS/Express in virtual environments.

1. Introduction

MVC has supported users of graphics and visualization since the mid 70's. Users now tend to have larger 3D datasets that require more intensive analysis. These data often benefit from custom made tools for the visual analysis and are more efficiently handled in a virtual environment. These two requirements of users have driven research at MVC into two independent but related areas:

- Work with scientist in their domain area to produce appropriate toolkits.
- Develop interfaces with virtual environments so that toolkits can benefit from more intuitive user interfaces.

The Centre for Civil and Construction Engineering has used finite elements (FE) methods for many years. Recently the problems have become more complex in terms of shape, size and the physics that they model. This has opened up a number of areas of research outside the engineering domain where a collaborative research effort is needed:

- Parallel FEM codes are needed to run analyses in a reasonable length of time.
- Visualization/VR tools are needed to validate/'debug' such codes.
- Visualization/VR tools are needed to help explore and understand the physics in the FEM.
- Visualization/VR tools are needed in the engineering design process.

2. Virtual Environments

An immersive virtual environment traditionally has a visualization supercomputer at its centre. This has tended to be a multiprocessor machine with large amounts of memory and specialist graphics, typically an SGI Onyx system. MVC has a 32-processor (MIPS R14000) SGI Onyx 300 visualization system with 6 graphics pipes for its visualization supercomputer. This platform was used to develop the multipipe edition of AVS/Express.

In recent years graphics hardware has developed more rapidly and reduced cost more quickly than any other hardware component [4]. This has had two effects:

- The bottleneck in the visualization pipeline has moved back out of the rendering engine.
- Clusters of graphics PC's are starting to be used as an alternative to the traditional visualization supercomputer.

2.1. Application Toolkits

AVS/Express is a modular visualization system with a high level GUI with its own visual programming environment. Visualization systems generally have a high learning curve for new users. A new user will probably have to understand at least some components of all these subjects; the domain science; principals of visualization; specific visualization algorithms; graphic design; elements of cognition; and hardware configuration. The high level GUI is designed to reduce the learning curve. It allows users to experiment with many different settings and see the results instantly.

Advanced users of AVS/Express can use the available high quality libraries as well as developing their own libraries of modules. All modules whether native or not are accessed through the GUI. New libraries build up toolkits of functions required by specific users. When necessary these can be integrated into a stand alone application where the user is presented with a user interface especially developed for them.

These features of AVS/Express make it suitable for use by research groups from various domains including engineering. There is one basic tool but it can be customized into a number of tools which can then be used depending on the skills of the user and the work they are doing.

2.2. Interfaces to Virtual Environments

A limitation of AVS/Express is that it was not designed for use in a virtual environment. It does provide active stereo support but this was mainly used on the desktop. There is no support for the multi-projector display environments often found in virtual environments. The AVS/Express Multipipe Edition (MPE) [5] project has extended the rendering engine within AVS/Express so that it can be used in these virtual environments.

The AVS/Express renderer has been extended using the SGI Multipipe SDK software. This allows MPE to take advantage of the multiple graphics pipes found in high end visualization supercomputers. The output from each graphics pipe can then be projected in the display environment. Common environments include workbenches, such as ImmersaDesks, three-wall large screen environments such as RealityCenters and immersive environments such as CAVEs. The display environment to be used is described in a simple text file (containing projector location and orientation for example) and so applications can be used in a variety of environments without modification. With support for both active and passive stereo hardware, together with tracking and wand devices, the Multipipe Edition allows the user to be immersed in and interact with their visualization.

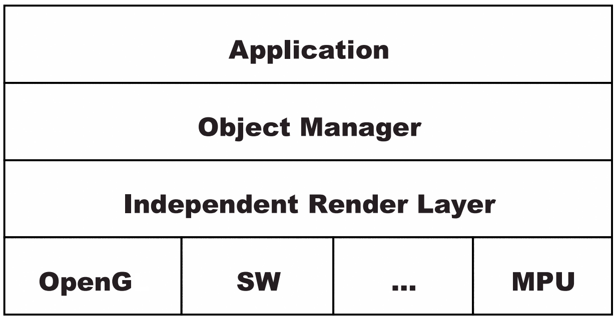

Figure 1. Architecture of AVS/Express the Multipipe Edition (MPE)

The diagram shows how the Multipipe renderer (MPU) is integrated with the AVS/Express architecture. At the top level the user's application communicates with the Object Manager. This controls data transport between modules within the application and when a module is executed, for example. A user's normally asks the Object Manager to render the visualization. The Object Manager does this by making calls to the independent graphics layer. This is independent of the renderer and hardware in use. It is this independent layer than communicates with the particular renderer in use within AVS/Express. For example the Software renderer may only be available on some platforms where as the OpenGL renderer can be used on others. By adding the MPU renderer in the bottom layer, multipipe support is added transparently to the rest of the AVS/Express application.

2.3. Computational Efficiency in the Renderer

The rendering resources provided by multiple graphics pipes can be exploited in several ways. The most common arrangement (or 'decomposition') is to connect each pipe to a separate projector thereby increasing the overall area available to the rendered visualization. This is the arrangement found in RealityCenters where three projectors each cover roughly a third of a large curved screen. However, multiple rendering pipes can also be used to increase rendering performance when only one screen or projector is to be viewed. 2D screen decomposition allows a single graphics window to be divided amongst the graphics pipe. For example, if two pipes are available then one pipe will render the left half of the window and the other pipe will render the right half of the window. This reduces the load on each graphics pipe, assuming the primitives to be rendered are distributed fairly evenly across the two image regions in the final window. Alternatively one pipe could render the left eye view in a stereo configuration and the other pipe can render the right eye view. A third decomposition is temporal decomposition. Here, each pipe is given a frame to render from a constantly changing visualization. Each time AVS/Express generates a new frame to be rendered it uses one of the unused graphics pipes to render the frame.

This can be rendered while another pipe is rendering and displaying the current frame. The frames are distributed to each pipe in a cyclical manner. By eliminating the need for the application to wait for one frame to render before it can render the next frame in the sequence the overall frame rate of the visualization presented to the user is increased.

2.4. Computational Efficiency in The Visualization Pipeline

The visualization pipeline is the computational processes through which the data is read and manipulated before an image can be produced. The image production step, rendering, is the last part of the pipeline normally implemented in OpenGL on local hardware. The visualization pipeline can be thought of as having 2 parts the rendering and the non-rendering part. The last section focused on the rendering; here we look at the data manipulation/non-rendering part.

Performance bottlenecks in the visualization pipeline have 2 main causes:

- Highly computational modules can cause bottlenecks.
- Large data sets can not fit into the real memory of a single computational node.

These problems are generally tackled by parallel implementations of some or all of the non-rendering part of the pipeline. Generally speaking parallel solutions tend to spread a computation across several processors, a step called decomposition. An implementation can spread the data or the function across processors and then after the computation the results are collected and composited back into one data block.

We are interested in solutions that improve user interactivity not techniques like data streaming where small blocks are passed through the entire visualization pipeline and so eventually one image is composited.

An initial project VIPAR (VIsualization in PARallel) [1], [2], [3], [4] produced a library of parallel functions that had a wrapper layer so that they were portable both between different MPI libraries and visualization systems. The portability was useful but meant that an implementation could not be tuned to a particular architecture and peak efficiency could not be gained. VIPAR had one parallel implementation for each module and complex visualization pipelines were slowed by repeated costly data decomposition and composition.

Recently the PST (Parallel Support Toolkit) project has commenced work on implementing a comprehensive suite of tools to support parallelism across the whole of AVS/Express. Its approach is to provide a suite of modules and libraries that interface with and are controlled by the AVS/Express object manager. No attempt is being made to parallelize the underlying structure of Express. Support will include parallelization on shared memory architecture, distributed heterogeneous clusters, integration with multipipe hardware and PC graphics clusters. The framework will allow control over what and how data decomposition is performed and will reduce the need to recomposite the data unnecessarily.

Rendering decomposition modes will also be utilized to improve large data set rendering capabilities. Additionally a suite of parallelized modules covering a range of visualization techniques will be provided. Finally the project will investigate level-of-detail and data streaming methods for use with asynchronous computation, rendering and computational steering.

3. Engineering Case Studies

Manchester Computing has developed a library of parallel routines for finite element analysis that are callable from FORTRAN90. These have been used to parallelize a suite of programs [6] covering 10 different algorithms taken from Smith and Griffiths [7]. With these programs, a range of engineering analyses may be performed covering static and dynamic material behavior in structural mechanics and geomechanics, as well as fluid and heat flow.

In the following section, two examples are presented. The first, the lid-driven cavity problem required visualization expertise to help validate the code through visualizing the results. The second involves a more complex physical problem, magnetohydrodynamics (MHD). In this case, visualization was used as an investigative tool to understand the complex physics being simulated.

Other problem specific tools have been written to help the engineers at UMIST validate new codes that are under development. These tools are not covered here. However the key point is that with the more complex 3D models that can be analyzed in a reasonable time through parallel computing, code validation by visualization is becoming increasingly important in the area of finite element analysis.

3.1. The Lid-driven Cavity

The lid-driven cavity problem is a well documented test case for computational fluid dynamics (CFD) algorithms. A cubic cavity contains a fluid that is initially at rest. The top surface of the cavity or 'lid' is driven at a constant velocity. A steady state solution is then sought for the motion of the fluid inside the cavity.

The domain was subdivided into a quarter of a million finite elements, giving rise to 1 million grid points where values for the pressure and velocity field were to be calculated. With approximately 4 unknowns at each grid point, the computational task was to solve a system of 4 million non-linear simultaneous equations.

There are various ways of solving CFD problems that typically involve numerical techniques such as finite differences or finite elements. Various formulations are in use that simplifies the Navier Stokes equations by taking mathematical short cuts. For simple test problems, these methods produce similar answers with reasonable computation times. However, for complicated geometries, these simplified methods may not give the correct answer or give any answer at all.

The closest we can get to the correct solution is to solve the full system of simultaneous equations with no simplifications. This is often referred to as direct numerical simulation or DNS. Being computationally very expensive, DNS is not typically used for everyday CFD problems. However, solving large complicated problems using DNS benefits the developers of simplified industrial algorithms by providing accurate solutions for validation work.

3.2. Program Performance

Table 1 shows the performance data for the problem described, which was analyzed using different values of Reynolds Number. The parallel time is for 256 processors, and the serial time is estimated from the time per iteration. It should be noted not only that the parallel version runs 256 times faster than the serial, but also that the actual performance relative to the peak performance of the processors is also very good.

Problem	Reynolds Number	Parallel Time Mins	Serial Time Days	%Peak Performance	Gflops
Half Cubic Cavity	10	20	2-3	29	59
	100	47	8-9	29	59
	1000	180	>1 month	29	59

Table 1 Performance data versus Reynolds number

The Reynolds Number is a dimensionless parameter that characterizes the flow. In simple terms, one can imagine a low Reynolds flow being laminar, like treacle and a very high Reynolds flow being turbulent, like smoke rising from a cigarette. As the Reynolds number increases, the computational cost increases and at a certain point a solution is unobtainable. Increasing the resolution of the analysis (number of grid points) enables the computation to proceed to a higher Reynolds number. To resolve fully turbulent flows using DNS, it has been suggested that around 1 billion computational grid points would be needed. We're a long way off at 1 million!

3.3. Visualization

As the Navier Stokes test problem shows, parallel computation can lead to significant improvements in simulation time. This not only increases the complexity of engineering model that can be investigated, it also means that more sophisticated tools are required to interpret the results.

All features of the data, both expected and unexpected need to be found and analyzed intuitively. Doing this not only requires the use of high-end interactive visualization equipment but also the application of appropriate visualization techniques.

The initial visualization application was developed using a low resolution data set (roughly 5000 grid points) on an SGI O2 machine. The application used a specialist data reader developed so that the engineering department could read their FEM data into AVS/Express. The reader is designed to read AVS/Express cell data using a format similar to the AVS/Express field file descriptor. A simple text file is used to describe the data. Information about the data is at the beginning of the file, its dimensions, number of nodes, data type, whether it is scalar or how many vector components it has and number and type of cells. After this at the end of the file pointers to the data files and position of the data are given.

The benefit of this type of reader is that the text descriptor file is easy to write and edit. The reader can read coordinates, cell information and node data from the files the engineers use. No new large data files need to be written or stored. Displacement data, important to this group of users, is a special case of vector data that can be marked and handled specially within AVS/Express. The format can be extended as the engineers need change.

The final visualization application used a texture based rendering method to show the complexity of the flow profile through the data. Cut planes were placed orthogonal to each of the principal axes of the data, although these could be moved by a user who wished to explore the flow profile. The magnitude of the data flow was assigned a texture coordinate across the diagonal of an image, in this case a blue shaded mandrill. The resulting image does not show the strength of the flow at any point but shows how it varies between points.

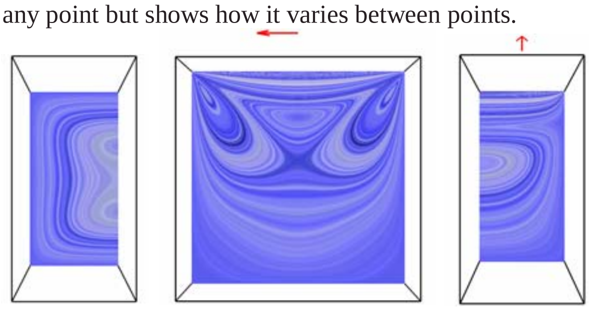

Figure 2. Flow profile for the central orthogonal slice for each of the three principal axes

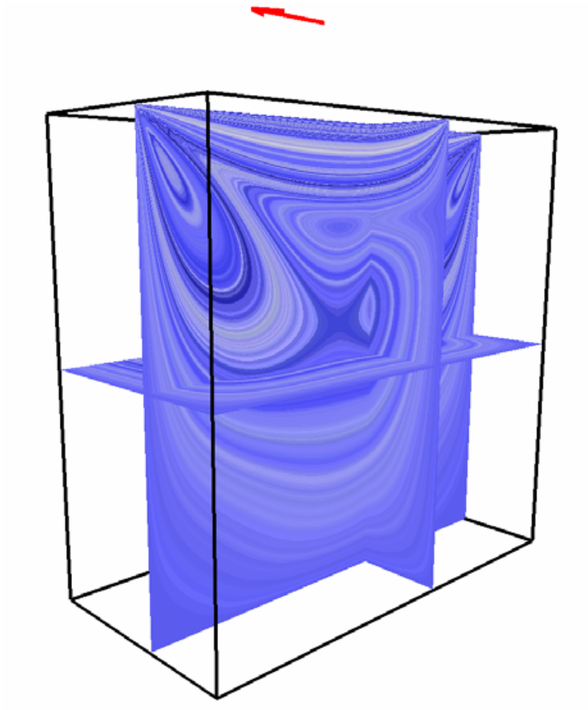

Figure 3. Flow profile of the three orthogonal slices in figure 2 but shown together.

The flow profile does not show where or how fast particles would move in this data. A probe is used to select a point or series of points within the data. These points are used as seed points which show what the movement of particles released from these points would be. The flow paths are shown as streamlines, the speed of the flow is shown as colour. The flow is fast where it is coloured red, against the lid. The flow is slow where the colour is blue, away from the lid. Flow is a time dependent phenomenon and as is nicely demonstrated by a well known animated technique called particle advection.

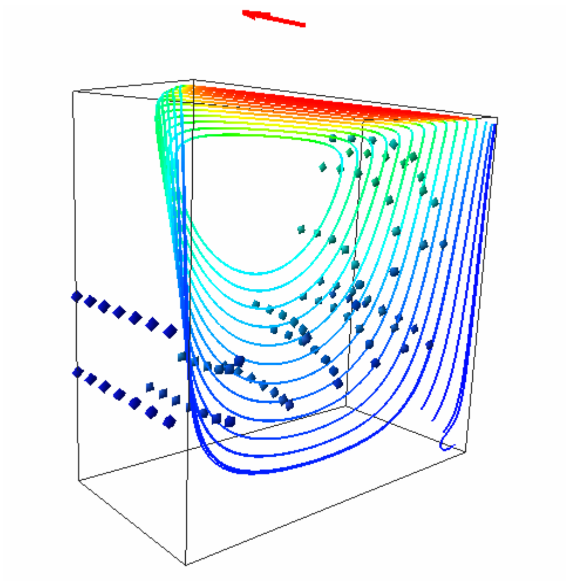

Figure 4. Streamlines and practical advection show the flow in 3D.

An arrow is used to show in which direction the lid of the box is moving.

When the visualization functionality was defined and the application developed the high resolution data could be examined. The high resolution data had over 200 times the number of grid points. The application would not run at all on the O2 but was moved to a 32-processor (MIPS R14000) SGI Onyx 300 visualization system with 6 graphics pipes. Initially the application was run across the network which meant the rendering was done locally on an O2. Some simple exploration could be done but it was difficult to explore the data for example it was difficult rotating the model, moving the orthoslices and probes. By moving into the immersive virtual reality environment it was possible to take full advantage of the graphics hardware and use the MPE version of AVS/Express. Another advantage of using this environment was that the engineer could see the results in stereo and could work easily with others. Data that had been difficult to validate was validated to the satisfaction of the engineer and associates within hours.

3.4. Magnetohydrodynamics

The parallel algorithms used to model the Navier Stokes equations were modified to simulate a more complicated flow problem [8] – the flow of an electrically conducting fluid in the presence of a magnetic field. In magneto-hydrodynamic (MHD) flows, the two distinct physical processes, electromagnetism and fluid flow, are coupled. The fluid flow affects the magnetic field and the magnetic field affects the fluid flow.

A simple test problem is described here where a fluid flows into a duct with a prescribed parabolic velocity profile and uniform pressure applied across the inlet face. A uniform pressure gradient is applied at the walls, along the length of the duct. The interaction of the two physical processes is then investigated by visualizing the results of a parallel FE analysis. There is no simple analytical solution to this problem and the validation of the model can only be undertaken by comparing the features of the simulation with physical experiments.

Interpreting the results of a simulation where two or more physical processes interact is particularly difficult. The engineers were originally using conventional visualization tools that output to a two-dimensional medium – a computer screen or hardcopy. This was very limited and the physical processes were virtually impossible to understand.

This simulation used a very similar visualization application to the one used for the lid-driven cavity simulation. The important difference here is that it was impossible to show both fields (magnetic and fluid flow) together. The previously described visualization application was extended so that the magnetic field profile could be seen in the duct with streamlines showing the particle flow of the conductive fluid and above a second instance of the duct showed the fluid flow profile with magnetic field lines. Colour was used to convey information about the physical processes. In both instances of the duct, the fluid flow was depicted using a blue colour scheme and the magnetic field was represented by red.

In the lower instance of the duct in figure 5, the position of the externally applied magnetic field is picked out by the distortion of the flow lines which twist both in the plane of the paper and perpendicular to it. This is impossible to make out visualizing in only two dimensions. To be able to identify and examine this feature, stereoscopic projection in an immersive environment proved an invaluable, if not necessary, analytical tool.

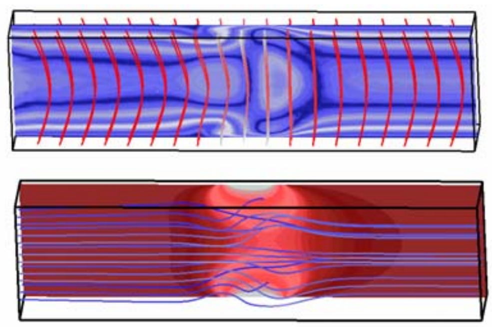

Figure 5. MHD duct flow

4. Conclusions and Future Work

4.1. Summary

This project has focussed on two engineering models - a CFD problem known as the lid-driven cavity; and a magnetohydrodynamics code that had two completely different but coupled physical processes, a magnetic field and the flow of an electrically conductive fluid. We have shown that immersive visualization has increased the understanding of these complex engineering processes. The comments received from end users have been very favourable and they are keen to exploit this approach further. Although we have just used MVC's immersive projection theatre, the application could also be used without modification in any other immersive environment such as a CAVE or workbench.

4.2. An Engineering Visualization Toolkit

Further visualization modules are being added to AVS/Express. In particular modules suitable for geotechnical engineering are being developed. The properties of each element of soil are governed by a stochastic field. Until recently all models run at Manchester were 2D. New 3D models are being developed that require validation and “debugging”. New tools will be required to deal with programming an adaptive mesh refinement algorithm. Engineers plan to use these models to show the dispersal of a pollutant through a geological domain. This too will require new visualization functionality.

4.3. Virtual Prototyping

Contemporary engineering design involves many stages. These may include drawing up specifications; producing a CAD model; evaluating the design, perhaps through computer simulation; physical prototyping; manufacturing and maintenance. The design process may require several iterations through this list until the final product emerges.

Fast parallel computation not only enables large problems to be run in a reasonable time, it also means that smaller analyses can be run very quickly - so fast that ‘virtual prototyping’ may be carried out. By integrating visualization with finite element analysis and virtual reality devices, an interactive ‘steerable’ environment can be created. The results of a simulation may be visualized in real-time as the model is being altered by the user.

In the future virtual prototyping will enable all the associated professionals to meet and collaborate together within a virtual environment, evaluating many aspects of the design reducing the number of iterations through the design cycle, thus reducing the overall design cost and time to market.

5. References

[1] S. Larkin, A. Grant, W.T. Hewitt, “Libraries to Support Distributed Processing of Visualization Data”, Future Generation Systems (Special Issue HPCN 96) Vol. 12, no. 5, North-Holland. Pp. 431-440, April 1997

[2] S. Larkin, A. Grant, W.T. Hewitt, “Vipar Libraries to Support Distributed Processing of Visualization Data”, High Performance Computing and Networking (proceedings of HPCN Europe ’96), Springer pp. 711-721, Brussels, Belgium, April 1996

[3] S. Larkin, A. Grant, W.T. Hewitt, “A Data Decomposition Tool for Writing Parallel Modules in Visualization Systems”, Proceedings of Eurographics UK Conference, London, pp.205-216 (Vol 1), March 1996

[4] J. Leng, K. Roy, J.M. Brooke, “VIPAR: A Case Study for UKHEC”, http://www.ukhec.ac.uk/publications, January 2001

[5] P.G. Lever, G.W. Leaver, I. Curington, J.S. Perrin, A. Dodd, N.W. John, W.T. Hewitt, "Design Issues in the AVS/Express Multi-Pipe Edition" IEEE Visualization 2000 Conference Works In Progress, Oct. 2000

[6] I.M. Smith and L. Margetts, “Parallel Processing for Non-Linear Problems”, to appear in VII International Conference on Computational Plasticity, Barcelona, 2003.

[7] I.M. Smith and D.V.Griffiths, “Programming the Finite Element Method”, Third Edition, Wiley, 1998

[8] I.M. Smith, L. Margetts, “Parallel Finite Element Analysis of Coupled Problems, Numerical Methods in Geomechanics”, Editors Pande G.N., Pietuszczak S., Balkema 2002