Visualization of Spherical Geometries Produced by Large Scale Simulation on the Cray T3E at Manchester, UK

J. Leng[1], J. M. Brooke[1], W. T. Hewitt[1] and J.H. Davies[2]

August 1, 2001

1, CSAR, Manchester Computing, University of Manchester
2, Jane Herdamn Laboratories, Dept. Earth Sciences, University of Liverpool

Abstract

We discuss the problems arising from the visualization of data from a geographical application running on 512 processors of the Cray T3E. The computational domain is a spherical shell and the user wishes to cut through the computational domain in a variety of ways and visualize the data, thus compounding the problems caused by the sheer size of the data set. We present solutions designed to allow the user to make informed decisions about data and information management and discuss how this can be extended to monitoring and steering the simulation on the T3E.

Keywords:
Visualization, spherical geometry, computational steering

1 Introduction

1.1 Motivation

In the UK it is rare for computational scientists to use 3D visualization. The UKHEC in its first year of operation has worked on two major case studies thus recognising the visualization issue as an important concern. The first case study worked with the data produced by the Terra group. The Terra group run simulations of the Earth's mantle currents and use "ray-tracing" of seismic waves to compare their simulations with structure revealed by seismological observations. They have identified two clear problems:

- Articles were being rejected by academic journals because the images the group were producing were not able to reveal key physical features.

- The group could not decide how to continue with their work, the visual interpretation was lacking and meant they could not fully utilize their computational time.

This paper shows through the first case study how 3D visualization enhanced this group's research, it details the solution applied and suggests how this work may be extended to data and information management and computational steering.

1.2 A Multi-Platform Service

This visualization case study was undertaken on the UK High End Computing facilities provided by CSAR at Manchester. To understand some of the specific data management and visualization problems it is helpful to understand the configuration of the CSAR cluster. The structure of the original machine configuration at CSAR is given in (Fig1), this had not significantly changed at the start of the case study. This configuration is multi-architecture and such configurations are increasingly seen in large HPC centres, e.g., CSC in Finland and IDRIS in France.

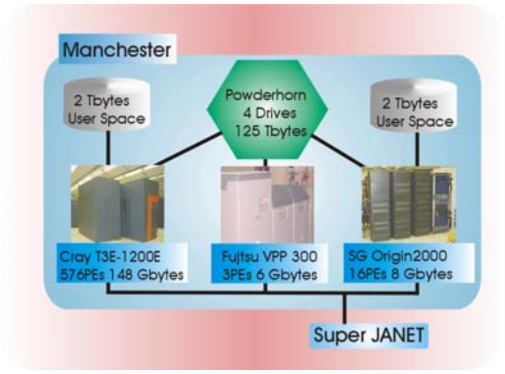
Manchester
2 Tbytes User Space
Powderhorn 4 Drives 125 Tbytes
2 Tbytes User Space
Cray T3E-1200E 576PEs 148 Gbytes
Fujitsu VPP 300 3PEs 6 Gbytes
SG Origin2000 16PEs 8 Gbytes
Super JANET

Figure 1: The original machine configuration at CSAR.

2 Case Study: Seismic Tomography and Convection Modelling of The Earth's Mantle by The Terra Group

The Terra group run computational simulations of the Earth's mantle (the layer between the crust and core) which run on 512 processors of the Cray T3E for up to 12 hours. The study of mantle currents are important for understanding both continental drift as well as volcanic and seismic activity.

The mantle is a relatively thick layer which extends nearly halfway to the centre of the Earth/sphere. The spherical shell being modelled is much thicker than the shells used for ocean or atmospheric modelling, it is 2900 km compared to the 100 km of the atmosphere (the radius of the Earth is 6370 km). The mantle is usually modelled by finite element analysis while the flows in the ocean and atmosphere on the other hand are usually modelled by finite difference methods. Finite element analysis can be thought of as the integral process of finite differences and is thus more computationally expensive. The shape and structure of this spherical shell makes it more similar to astrophysical simulations of stars or planets rather than ocean or atmosphere modelling.

Seismograms from just over 3000 seismic stations resulting from around 10,000 earthquakes have been collected. The travel times of these seismic observations are inverted by velocity tomography so the group can compare the lateral variations in the reconstructed seismic velocities with the lateral temperature perturbations of the mantle model. In this way they can verify the simulation, by comparing theory with observation.

2.1 Problems For The Terra Group

The problems can be boiled down to four main areas, modelling or inverting this spherical geometry, visualizing it in 3D, resource restrictions and their ethos.

Modelling This Spherical Shell
The solution to the inversion of seismic data dates back to the 1930's. Since the variations are dominantly radial the model is effectively reduced to 1D, depth is the only factor. Given this history it has been quite common to analyse the data by 2D projections (fig2).

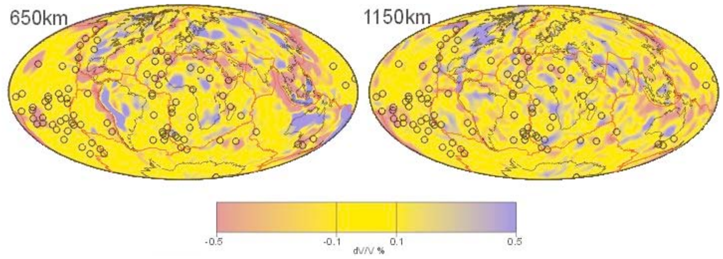

Figure 2: 2D projections for two shells of given depth.

However academic peers are now demanding 3D analysis of what are truly 3D problems (both the tomography and the simulation).

3D Visualization
We have concentrated on visualising the tomography data initially. The group can convert the simulation data to the tomography format, and hence can visualize both data from this format. We will look to visualize the unconverted simulation data in future work. The data mesh for the tomography is made

of cells. The cells are not defined by array data as they are for atmospheric modelling but by a data structure. The data cells are defined in a spherical coordinate system so implicitly the cells have curved surfaces. Visualization systems use straight lines for all surfaces and these are defined in a cartesian coordinate system. To transform the simulation cells into renderable objects they must be resampled in space defined by cartesian coordinates to fit the original curved shape enough to tessellate.

The complexity of the data and of visualization systems mean that this reader is best produced by a graphics expert rather than a geophysics researcher. The number of cells required to visualize the data mean the results will be slow to manipulate unless specialist graphics hardware is used.

Resource Restrictions
The consortium work remotely on the T3E. Their connection is by Net North West, a special high bandwidth (622Mbps) academic network. At their site they mainly have low end machines with limited graphics capabilities, no graphics hardware and the local bandwidth may be drastically reduced from 622Mbps, therefore large data sets may be slow to ftp. While they would like to keep their graphics capabilities local this is not currently possible without reduced functionality. It would mean a one off investment in hardware and a yearly expenditure for software, most commercial visualization software have yearly licenses. They need support to develop software but also need advise on software and hardware.

Each simulation is computationally expensive. Currently they can only analyse the results at the end of the simulation and if the simulation diverges away from the solution they have wasted their resources. We believe that they could improve their performance by combining their expertise with that of the program, this is called computational steering.

Their Ethos
They have no special graphics hardware or machines at their site. Their expertise is with computation not graphics and not surprisingly they are focused on improving their simulation, any new computational resources are used to increase the data resolution. Ironically this means their graphical problems are increased and it becomes more difficult to resolve.

3 3D Visualization

3.1 Special Challenges of Visualizing Thick Spherical Shells

There are several reasons why this type of data is difficult to visualize.

- Graphics renderers only handle straight lines, spheres have few straight lines, are defined by spherical coordinates and to get a well formed tessellated cell set extra data cells must be introduced.

- The number of cells needed to visualize this data means special data reduction/handling techniques are required.
- Placement of reference information is important since this allows the researcher to relate features inside the sphere to surfical features. The user needs "volcanic" hot spots (e.g. Hawaii, Iceland), tectonic plate boundaries and coastlines to be present in each image.
- The shell is thick and placed close to the centre of the sphere, the angular component of the data is more important than it would be for atmospheric data.
- Perceptual problems with understanding a spherical shell:
  - A sphere looks the same from every angle so it is difficult to orientate.
  - Perspective has little effect.
  - Colour, transparency and shading sometimes work to make the pseudocolour difficult to interpret. There is a stronger darker band of colour around the outer circumference of the sphere.
- Problems with depth:
  - A 2D projection of one layer gives no depth information, is distorted and does not link around on itself.
  - The original visualization only related to depth slices. When seen in 3D many depths are seen at the same time, the depth values need a new normalization function to make 3D images valid.
  - A cut plane through the sphere shows how the data varies with depth but a plane is 2D and loses 3D information i.e. data with an angular component is lost.
  - Internal data with an angular component may be self obscuring, difficult to position because of problems with perspective and is related to reference data ambiguously.

3.2 Visualization of The Earth's Mantle As A Spherical Shell

A C data structure defines the cells used by the simulation. The data lies in layers that wrap like onion skins around the sphere. Each layer is divided into latitude bands which ring the Earth in the same orientation as the equator. The smallest band lies at the pole while the largest is at the equator and each of the intervening bands vary in size. Each band is divided into cells of equal volume so adjacent bands have different numbers of cells and the edges between bands do not line up. Schematic diagram of the cell layout is given in fig 3, it also shows the cell arrangement of one layer for 5 bands from the pole.

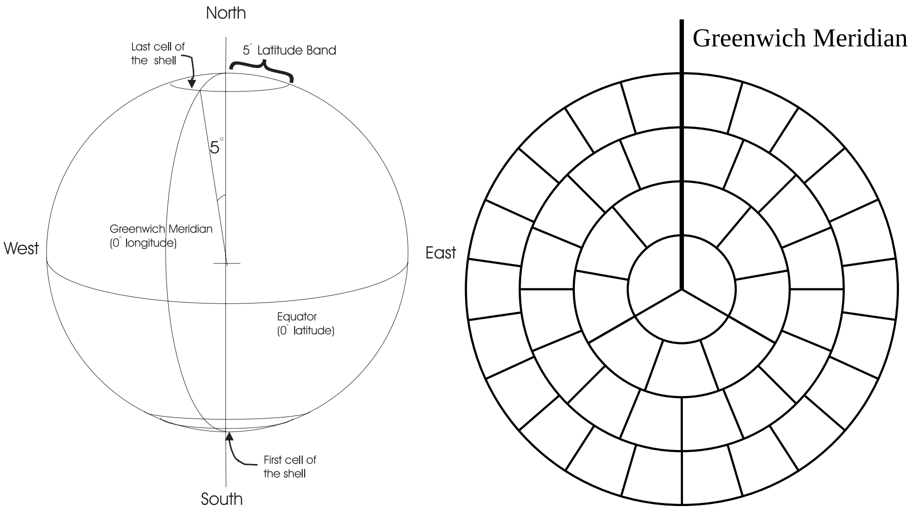

Figure 3: Diagrams to show the lay out of the data cells. On the left a schematic diagram to show the over all layout, while on the right a view of 4 latitude bands of cell's from the pole.

The structure is given in fig 4 which shows the number of cells and angle of longitude for each cell in each 5 degree latitude band where nlon is the number of cells so far in the shell, nbox is the number of cells in that latitude band and mesh is the angle of longitude for each cell in the band. It can be seen from this structure that each shell contains 1654 data cells and there are 29 shells used. There are three cells at the pole and these are in the shape of a curved triangular prisms while the rest of the cells are curved hexahedrons (rectangular prisms).

```
const struct {int nlon, nbox; float mesh;} pgrid[36] = {
3, 0, 120.000000,
9, 3, 40.000000,
16, 12, 22.500000,
22, 28, 16.363636,
28, 50, 12.857142,
33, 78, 10.909091,
39, 111, 9.230769,
44, 150, 8.181818,
49, 194, 7.346939,
53, 243, 6.792453,
57, 296, 6.315790,
61, 353, 5.901639,
64, 414, 5.625000,
67, 478, 5.373134,
69, 545, 5.217391,
70, 614, 5.142857,
71, 684, 5.070423,
72, 755, 5.000000,
72, 827, 5.000000,
71, 899, 5.070423,
70, 970, 5.142857,
69, 1040, 5.217391,
67, 1109, 5.373134,
64, 1176, 5.625000,
61, 1240, 5.901639,
57, 1301, 6.315790,
53, 1358, 6.792453,
49, 1411, 7.346939,
44, 1460, 8.181818,
39, 1504, 9.230769,
33, 1543, 10.909091,
28, 1576, 12.857142,
22, 1604, 16.363636,
16, 1626, 22.500000,
9, 1642, 40.000000,
3, 1651, 120.000000,
};
```

Figure 4: The C structure that defines the low resolution simulation cells in spherical coordinates.

The tomography cells are defined so that each cell of a shell is of equal volume. The structure given here is for the lowest resolution data set, there are higher resolution tomography data sets with 1 degree latitude bands and the convection simulations approach the equivalent of 0.4 degrees bands.

AVS/Express was used for this application. A number of modules were developed in C and C++ with the Developers Edition of the software. User interfaces were designed for these modules and application networks built so they could be used at the Terra group's site in Liverpool using the Visualization Edition of AVS/Express. Further reasons for our choice of visualization system are in section 5.1.

Cells used by a visualization system are defined as being of a particular type, e.g., prism or hexahedron. A coordinate is given for each apex of the cell, then the connections between each coordinate are specified and finally a data value is given for each cell (cell data) or for each apex (node data).

Initially the data was mapped to just the cells explicitly defined by the C structure (fig 3). The coordinates were calculated for the apex of each cell in turn even though neighbouring cells would have common apex locations. The coordinate of each apex would appear at least twice in the coordinate array, once for each cell with a shared apex. Not surprising the resulting cells did not tessellate, gaps appeared between most latitude bands. More interestingly there was a banding effect across the surface caused by light shading. The shading had two causes. Firstly between latitude bands not only were there gaps but there were cell overlaps which stopped the surface being smooth. Secondly there was a shading effect between cells of the same band, this was caused by two adjacent cells with shared apices referencing different coordinates on the co ordinates array although they should be the same point in 3D space. The reason for this seems to be the imprecise nature of floats and coordinates are stored as floats. The result was very unsatisfactory, the solution was to increase the number of cells but to keep the number of coordinates as low as possible.

3.3 Resampling The Cell Sets

Removing Duplicate 3D Locations From The Coordinates Array Coordinates were calculated for the cells of a whole latitude band so neighbouring cells with common apices would never reference difference points on the coordinate array for the same point in 3D space. This removed most of the shared apices from the coordinate array but some incidental ones remained. The first cell on each band begins at the same longitude and will share an apex coordinates with the bands above and below. The bands that lie on each side of the equator share all their adjoining apices. It is a default throughout that when there are two possible coordinates that the one from the upper band will be used.

Neighbouring Bands With A Harmonic Number Of Cells Occasionally adjacent bands would both have a number of component cells that would be divisible by the same small integer, e.g., in the low resolution data the second band has 3 cells and the third band has 12 cells, both are divisible by 3. This effect causes an incidentally shared apex with its known shading problems. In these cases the coordinate from the upper band is selected.

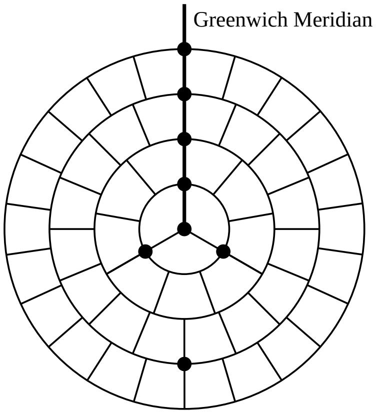

Figure 5: Shows the first four latitude bands, viewed from the South Pole, the dots show where there are duplicate 3D locations in the coordinates array.

Resampling The Poles

In all resolutions of the data the latitude band at the poles consists of 3 triangular prisms and the next band consists of 9 hexahedra. For each pole the prisms were resampled to give 9, 3 for each of the original prisms. The coordinates were used from the neighbouring band whether up per or lower.

Resampling The Hexahedrons In All Other Latitude Bands

Each cell from the other latitude bands should consist of one curved hexahedron. This does not tessellate because it is impossible to give it curved edges they must be straight. Each shell has the same number of cells, they tile up on each other so extra cells only need to be added within each shell. The solution must be scalable so it can be extended to higher resolution data.

The resampling solution was to map one hexahedron for all of the original simulated data cells but not to map it to the full extents of the cell and to add triangular prisms to fill in the gaps. There were gaps being caused by either cells in the band immediately above or below not having apices exactly in line with the cell being looked at. The number of cells increase in each band from the south pole to the equator and then decrease up to the north pole. A solution was found that was symmetrical through the equator so the geometry in the southern hemisphere is a mirror image to that of the northern hemisphere.

Each data cell consists of one hexahedron and a number of triangular prisms. The hexahedron is cut so it changes from its initial rectangular prism shape to that of a trapezoid prism and triangular prisms are added to give the cell a curvature that tessellates with the cells in the latitude bands above and below. All hexahedrons will keep four of their original apices but some may keep six.

Imagine you are looking at the surface of a hexahedron so it looks like a rectangle. Now imagine the cells in the band above, they are smaller and there may be one or two which lie across the top of our cell. Our hexahedron must curve at the apex of each of the cells above if it is to tessellate. To do this we cut our cell from the apex of the cell above to the bottom left apex of our cell giving us one triangular prism and a trapezoid prism. If there are two apices above we cut twice and get two triangular prisms and one trapezoid prism. The cells in the band below are larger and also do not align with our cell. We must make our trapezoid prism curve to tessellate with the cells below. This time we cut the cell from the last apex above to the first apex below, we keep a trapezoid prism but add another triangular prism.

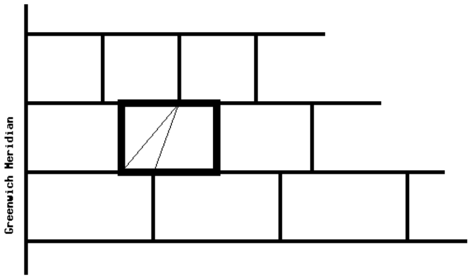

Figure 6: A diagram showing the resampling scheme. It is the surface of a group of cells in 3 latitude bands projected in 2D. The cells appear rectangular; they are below the equator so each band has progressively more cells. We are looking at the cell with the thickened border; the thin line shows how it is cut into two triangular prisms and a trapezoid prism to make it curve with the bands above and below it.

Reading Cell Sets For Better Display And Data Management

The cell sets produced by the method given above gave a smooth evenly shaded sphere. Unfortunately the whole data had been resampled so it contained about 3 times the number of cells. It stopped being interactive (interactivity is generally accepted as 10 frames per second or more) and problems with visualization methods become clearer. The data set used to produce this visualization was composed of a large number of data cells (roughly 144 000). Interactivity is probably best achieved on a stand alone system with fast CPU, a large amount of memory and with dedicated graphics hardware. This data was not interactive on a SGI O2 but was interactive on a SGI Octane and Onyx.

One of the most common techniques used in visualization is the bounding box, its shows the extents of the data. For a spherical shell the bounds are not easy to detect, the extents of the data are clear when it is seen as the sphere's surface but one of the extents is completely embedded within the other, i.e. the innermost shells is entirely within the outermost one.

All the shells are needed for visualization but removing cells as early in the pipeline as possible increases interactivity e.g. when the sphere is cut the data values across the plane must be calculated but, after cutting, any data cell that is completely internal can be removed. Separating the outermost shell and innermost shell from the rest is helpful as they can then be used to mark the data's bounds.

To achieve both of these objectives the innermost shell, the outermost shell and the rest of the shells were placed in separate groups of cell sets. Each of these three groups could then be separated or kept together and passed down different visualization pipelines to produce the most appropriate images.

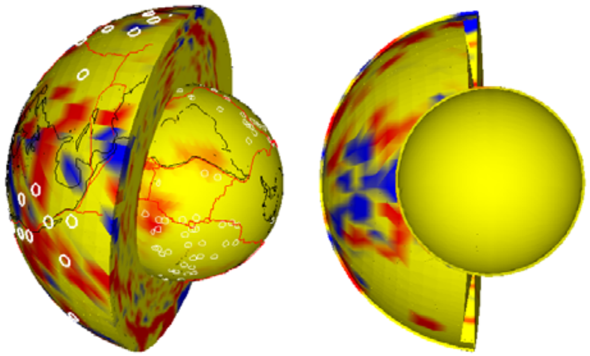

Figure 7: These figures use the three groups of cell sets one for the inner most shell, the outermost one and one for all the other shells. The picture on the left shows how these can be used to create a visualization while the picture on the right shows that the internal data was removed early in the visualization pipeline to make it more interactive.

3.4 Reference Data, Pseudo-Colour and Data Normalisation

The tomographic data of the Earth's mantle is strongly related to other information, volcanic "hot spots", tectonic plate boundaries and coastlines. It is vital that this reference data is always visible and unambiguously displayed.

For tectonic plate boundaries and coastlines a simple polyline could be used to join the appropriate coordinates into a line. The "volcanic" hot spots are traditionally viewed as a ring centring on the location of the hot spot on the Earth's surface. To achieve this an equation was used to calculate 12 points around the hot spot. The hot spot "ring" is defined as a circle on the surface of the sphere, i.e. the radius is a curved line. This defines a curved surface rather than a flat one when considered in the three dimensional space in which the

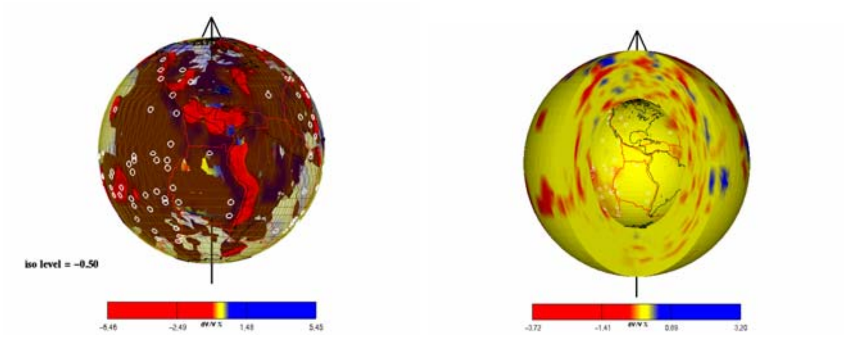

Figure 8: Two further visualizations. The picture on the left is of an isosurface, the outer most shell is semi-transparent while the inner most shell is opaque and shows the bounds of the data. The picture on the right shows how the inner most shell is used to display reference data that would otherwise be cut away.

sphere is embedded and to which the cartesian coordinates relate. Therefore it is necessary to translate from one definition to the other. We do this by calculating the latitude and longitude of a selected number of points on the ring, this is done using the formulae of spherical trigonometry (appendix 6.1). The latitude and longitude are then translated into the equivalent cartesian coordinates of the embedding space. All of the reference data is stored in spherical coordinates and then converted to cartesian coordinates.

The colour used for all the data, the simulated data, the reference data and the background adheres to the standard colour representations of the field of mantle studies. The tomography data is given as percentage velocity perturbation which means the data is tightly clustered around the zero value. Many cells have a zero; i.e. have a zero perturbation or are at the average value. The scientists interest is in the data close to zero either just faster or slower than average so a saturated colour scale is used at the extents of the data values that rapidly changes around zero.

A new normalisation function was added to give the root mean square of data values for each layer (appendix 6.2). The colour map has been designed to highlight data values close to but not at zero, the zero value is the average value of the distribution for each layer. The lower layers have data values more closely clustered around zero than the outer layers, the lower layers are far hotter and vary less from the average. The normalisation function makes the variation in the inner most layers as prominent as the outer most one while still using the same colour map for all layers.

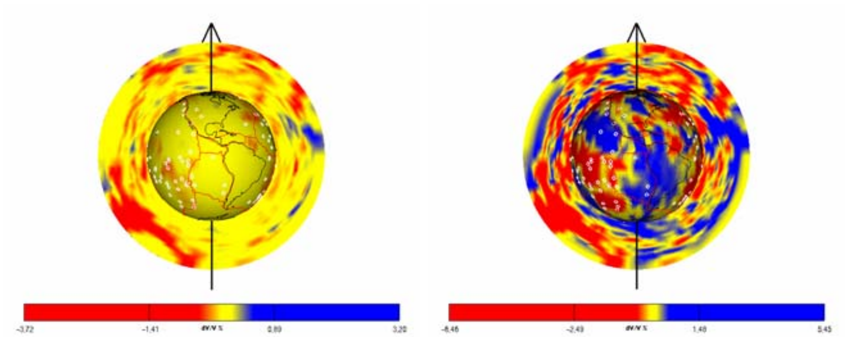

Figure 9: The image on the left uses data, which is not normalized while the one on the right uses normalized data.

4 Other areas where spherical geometry is of importance

It is our intention to generalise the techniques described here to other areas. Spherical geometry is clearly of vital importance in the study of the interiors of both planets and stars. To make this discussion concrete, we discuss the next application area where we intend to apply these techniques, namely the study of the behaviour of the convection zone in the sun.

The sun's interior can be divided into two different regions in terms of energy transport. The inner core is stable against convection and therefore energy transport is via radiation (conduction is negligible in stars such as the sun although it is important in compressed stars such as white dwarfs). The outer third of the sun is convectively unstable and complex patterns of flow arrive through the interaction of convection, rotation and magnetic fields (the solar gas is highly electrically conducting). Effects of this activity can be seen via surface phenomena such as sun spots, granulation, prominences, flares. This highly dynamic environment is responsible for spectacularly explosive events such as coronal mass ejections where material is ejected from the sun. On arrival at the earth this solar wind interacts with the earth's ionosphere and magnetic field to produce aurorae and magnetic storms. The latter can be highly destructive to electrical and electronic systems.

As a result of the enormous importance of such events, there is an intense programme of solar observation via ground based and satellite based systems. The accuracy of such techniques makes possible a new science of helioseismology, giving information about the rotation patterns inside the sun. The currently accepted theory for the origin of the sun's magnetic field is that it is created by dynamo action in the convection zone or in the overshoot layer immediately below it. The dynamo arises through the interaction of rotational shear and cyclonic convection (an effect of the Coriolis force, as are cyclones in the earth's atmosphere). Thus the rotation profile is of enormous importance in the theory of the sun's magnetic behaviour which in turn influences the solar wind.

An important new application of the visualization techniques presented here, is to utilise compare theoretical models of convection and dynamo action with observations. The problems have many similarities with the geophysical application. There is data about the distribution of magnetically active bands on the sun's surface which needs to be mapped onto the surface of our shell as has been done for the hot spots (see section 3.4). However an extension to the present techniques is visualize both scalar and vector fields in a spherical shell. Most previous work has presented data in two dimensional slices usually along lines of solar longitude and as contour plots at specified depths. Given the complexity and three dimensional nature of the fluid flow we hope to highlight new aspects of the data by visualizing it in its full dimensionality.

The data comes both from satellite observations of the sun, helioseismology and from numerical simulations that aim to model the solar convection zone. This is similar to the situation with geophysical data and a major challenge is comparing theory to observation.

5 General Issues Raised by The Case Study

There is increasing realisation that high-end computing brings special challenges with regard to the storage, movement and archiving of the very large data sets that simulations using hundreds of processors can produce. One can view such data sets as a challenge in several ways order of magnitude terms, volume of data, numbers of files, cataloguing issues etc... However the output of these simulations also raises problems in relation to the extraction of information and insight which cannot be so easily evaluated.

Visualization techniques can help with the process of the extraction of in formation from large-scale simulations. This in turn can help with decisions about what data needs to be stored and to make decisions as to how the simulations are set up and run. The simulation process can be optimised by providing tools to harness the physical insight of the scientists running the simulation. The geometry of a spherical shell presents novel challenges which are interesting both in terms of mathematics and also human perception. It is known that to envisage globally movement on a spherical surface is challenging, this becomes even more so when questions of depth and varying radius come into play. We have drawn attention in section 3.1 to questions of normalization. Many physical quantities (temperature, pressure, velocity of sound) are strongly radially dependent in spherical objects and this presents challenges to the visualization software. It is important to provide solutions which can be readily adapted by the end-user, questions about appropriate normalisations, zooming techniques, use of symbolic referencing information (e.g. colour, symbols, superimposed outlines) needs to be decided by the scientists for the detailed interpretation of their simulation results.

How the data from the simulations is then stored depends on many factors. If a group has sufficient storage access they may decide to store all data for analysis over an extended period. However, it will be more usual that some trade-off needs to take place between the resource implications of storing data and the desire to extract from it as much scientific benefit as possible. It is here that the ability to make informed decisions can potentially be advantageous. While some aspects of data reduction can be automated, e.g. by compression techniques, it is not currently possible to automate the intuition and judgement of expert scientists. For future work, these visualization tools may be combined with expert systems and artificial intelligence soft ware to assist the process of scientific value judgement. To assist such work, it will important to have clearly described case studies and to initiate collaboration between scientific groups and providers of visualization and virtual reality tools.

5.1 Tools For Visualization, VR and Computational Steering

Many of the current tools have very similar functionality. How they differ is the user interface and their market stability. Visualization has a steep learning curve and the use of a full graphical user interface can reduce the learning time and improve the complexity of visualizations produced. As an alternative there are visualization systems that are composed of C or C++ libraries, while many (particularly **vtk** (8)) offer good functionality it is more difficult to build applications especially for the novice. Often with this sort of software the user needs a developer to build an application and suitable user interface from scratch. An application of this type is much more difficult to adapt and allows the scientist less chance to experiment with new visualization techniques.

New visualization applications will be used by the scientists for several years and they need to be able to know that the software will still be available and run on their latest operating systems. The well established visualization systems (**AVS** (5,6), **Iris Explorer** (16,17), **IDL** (15)) have proven longevity and as commercial products they will be continually developed although the company who owns the software will decide in which way. There are many non-commercial products some of which have developed via open source methodology and are tested by the users on many of the operating systems, some of the larger systems use accepted QA systems (**vtk**). Support for such software is on a volunteer basis, that can work very well or very badly, this should be taken into account when deciding to invest in using open source products. Interestingly some non-commercial products may turn into commercial ones which the users then have to pay for, **COVISE** (9) started as a free product but recently became commercial, vtk is free for the libraries but for support or for the version with a graphical user interface you must pay. Another product IBM's **Data Explorer** used to be commercial but last autumn became free and is no longer supported by IBM.

There is extra functionality the user may wish to have apart from core visualization. These functions include collaborative environments, design for remote use, integral data reduction, output for VR equipment and computational steering. These are not always defined in the same way for all software but depends on the implementation and design of the system. For example **COVISE** was originally designed for collaborative visualization to use with high end VR equipment, **COVISE** was extended to computational steering by considering the computation to be another partner in its collaborative environment. **SCIRun** (10) acts as a kernel in an operating system and allows access to data within a computation to be accessed and visualized. It can be thought of as a kind of "debugging" tool allowing data arrays from the simulation to be interactively interrogated and examined its own visualization libraries. **SCIRun** was built and designed as a computational steering tool not as a visualization system.

Many of these extras are now beginning to be considered as integral to visualization systems. All graphics engines can output to VR equipment but not all can take advantage of the specialist graphics hardware. VR is rapidly becoming more affordable and offers some good techniques that may aid data visualization, e.g., stereo, large shared environments.

The product chosen for this UKHEC case study was **AVS/Express**. It is a product with a good GUI for beginners but that is easily extendable. It has two licenses a developers and a visualization one, while the developers license is probably too expansive for many academic users the visualization edition is reasonably priced and through the CHEST agreement many UK universiti es already have licenses. The Visualization Edition can do most of the things the Developers Edition can and it is all supported by a stable company. The output of **AVS/Express** can be passed into VR equipment although this is not the primary functionality currently needed since most UK academic sites do not have VR equipment. Now there is **AVS/Express Multi-Pipe Edition** that means **AVS/Express** can also make use of specialist high end VR equipment. However an increasing number of universities are acquiring such facilities and UKHEC can play a role in providing training and support.

In the last few years there have been new web based, distributed visualization systems developed as free ware (13,14). While these look promising none are commercial products with the associated QA assurance and none quite meet our requirements. These systems work from a totally different design concept and may prove the way forward. There is certainly more demand for systems that can run on PC and these systems work on distributed clusters of PC's. The visualization field seems to be at a point of expansion, more businesses are using it for information visualization, VR is more common place because games companies have pushed forward the development of graphics cards and e-science is proving itself profitable.

6 Appendix

6.1 Equation to Calculate Coordinates That Ring a Point On a Sphere

The calculation uses sphereical coordinates and all angles are measured in radians. The coordinates of the initial point on the sphere is given as R, LAT and LON. The coordinates of the point on the ring that circles the initial point is R, lat and lon while the radius of this ring is r and the angle between points on the ring is δ.

$$lat = \arccos(\cos r . \cos LAT) + \sin r . \sin LAT . \sin \delta$$

The function arccos can return two possible results but we always need a positive result.

$$lon = LON + \frac{r . \cos \delta}{\cos LAT}$$

6.2 Equation to Give The Root Mean Square Normalisation for Each Depth Layer of Data Values

The equation is given where there are n data points in one depth layer, a data point is refered to as x while a normalised data point is refered to as χ.

The mean is defined as:

$$\overline{x} = \frac{\Sigma x_n}{n}$$

The root mean squared deviation is given as:

$$\sigma' = \frac{\Sigma (x_n - \overline{x})^2}{n}$$

$$\sigma = \sqrt{\sigma'}$$

For each data point the normalised value is given as:

$$\chi_n = \frac{x_n - \overline{x}}{\sigma}$$

References

[1] Davies, JH; Bunge, HP, "Seismically ‘fast’ geodynamic mantle models", submitted to Geophys. Res. Lett.

[2] Bunge, HP; Davies, JH, ”Tomographic images of a mantle circulation model”, submitted to Geophys. Res. Lett.

[3] Rhodes, M; Davies, JH, ”Global tomographic detection of mantle plumes in the upper most lower mantle”, submitted to Geophys. J. Int.

[4] Davies, JH web pages:-
http://mantle.esc.liv.ac.uk/davies/AGU99/

[5] Advanced Visual Systems Inc., ”AVS/Express Developer's Reference, Release 3.0”

[6] International AVS Centre homepage:-
http://iavsc.org/

[7] Advanced Visual Systems (AVS) homepage:-
http://www.avs.com/index.htm

[8] Visualization tool kit (vtk) homepage:-
http://www.kitware.com/vtk.html

[9] COVISE homepage:-
http://www.hlrs.de/structure/organisation/vis/covise/

[10] SCIRun homepage:-
http://www.cs.utah.edu/sci/scirun/

[11] S.G. Parker and C.R. Johnson, ”SCIRun: Applying interactive computer graphics to scientific problems.”,SIGGRAPH 96,1996.

[12] S.G. Parker, D. Beazley, and C.R. Johnson ”Computational steering software systems and strategies.”, IEEE Computational Science and Engineering, 1996 (submitted).

[13] Visad homepage:-
http://www.ssec.wisc.edu/ billh/visad.html

[14] Vis5d homepage:-
http://www.ssec.wisc.edu/ billh/vis5d.html

[15] IDL homepage:-
http://www.floating.co.uk/idl

[16] Iris Explorer homepage:-
`http://www.nag.co.uk/Welcome_IEC.html`

[17] Iris Explorer Centre of Excellence homepage:-
http://www.comp.leeds.ac.uk/iecoe