

SAND 97-8249

Distribution
Category UC-405

Unlimited Release
Printed April, 1997

ASCI Visualization Tool Evaluation Version 2.0

by

The ASCI Visualization Common Tools Subgroup
Philip Kegelmeyer, Editor
Center for Computational Engineering
Sandia National Laboratories
Livermore, CA, 94551

February 27, 1997

Abstract

The charter of the ASCI Visualization Common Tools subgroup was to investigate and evaluate 3D scientific visualization tools. As part of that effort, a Tri-Lab evaluation effort was launched in February of 1996. The first step was to agree on a thoroughly documented list of 32 features against which all tool candidates would be evaluated. These evaluation criteria were both gleaned from a user survey and determined from informed extrapolation into the future, particularly as concerns the 3D nature and extremely large size of ASCI data sets. The second step was to winnow a field of 41 candidate tools down to 11. The selection principle was to be as inclusive as practical, retaining every tool that seemed to hold any promise of fulfilling all of ASCI's visualization needs. These 11 tools were then closely investigated by volunteer evaluators distributed across LANL, LLNL, and SNL.

This report contains the results of those evaluations, as well as a discussion of the evaluation philosophy and criteria.

Disclaimer: The recommendations in this document are made with specific consideration of ASCI requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Acknowledgments

This project was supported by the United States Department of Energy through the Accelerated Strategic Computing Initiative program. Contributors came from the Los Alamos, Livermore, and Sandia National Laboratories.

The evaluators who volunteered their time to install, test, and report on the various visualization tools include Stephany Bouchier, John Fowler, Bob Kares, Dave Modl, and Bob Shea of Los Alamos, Eric Brugger and Jeff Rowe of Livermore, and Steve Attaway, Verlan Gabrielson, Mike Glass and Meeko Oishi of Sandia.

Additional others who participated the ASCI Visualization Common Tools subgroup through e-mail or attending related meetings, and so who contributed to working out the evaluation criteria and philosophy, include John Ambrosiano, Chuck Hansen, Mike Krough, Andy Martinez and Pat McCormick of Los Alamos, Tom Kelleher of Livermore, Arthurine Breckenridge and Philip Heermann of Sandia, and Thuc Hoang of DOE Headquarters.

Contents

1	Feature/Tool Matrix Summary	6
2	Evaluation Methods and Philosophy	7
3	Descriptions of the Required Features	8
4	Tool Descriptions and Evaluations	12
4.1	AVS Express	12
4.2	Data Visualizer	18
4.3	EnSight	23
4.4	IBM Visualization Data Explorer (DX)	35
4.5	General Mesh Viewer (GMV)	39
4.6	Khoros	43
4.7	Mesh-TV — A Graphical Analysis Tool	46
4.8	EIGEN/VR	53
4.9	MUSTAFA	61
4.10	Phigure	72
4.11	Spectrum Visualizer	78
A	Tool Evaluation Resources	81
B	Related Tools	81

1 Feature/Tool Matrix Summary

	AVS	DV	EnSight	DX	GMV	Khoros	MeshTV	EIGEN	MUSTAFA	Phigure	SV
Support	1	-2	2/2	2	1/0	1	2/1	2/1	0/0	1	2
Documentation	2	2	2/2	2	2/1	0	2/1	1.5/1.5	1/1	2	1
Turnkey	-2	2	2/2	-1	2/2	1	2/2	1.5/-2	2/2	?	2
Simple	-1	2	0/0	1	2/2	-1	2/2	2/-2	2/2	1	2
Variable Types	2	2	2/1	2	2/2	-1	2/2	2/2	2/2	2	2
Axis	2	2	2/2	2	1/1	1	2/-2	2/2	-2/1	2	0
Query	0.5	1	2/2	2	2/2	-2	2/0	2/-1	1/1	0	2
Scale	0.5	2	2/0	2	1/1	1	2/0	2/-2	2/1	2	2
Annotation	1	2	2/1.5	2	0/0	1	1/-1	2/0	1/1	2	2
Algebra	1	2	2/2	2	2/1	2	2/2	1/-2	-2/1	-2	2
Viewing	2	2	0/2	2	1/1	2	2/1	2/2	1/2	2	2
Cut-planes	2	2	2/2	2	2/1	1	1/0	2/1	1/2	2	2
Isosurfaces	2	2	2/2	2	1/1	1	1/1	1.5/-1	1/2	2	2
Isovolume	2	-2	-2/-2	2	-2/-2	0	-2/-2	1/-2	1/2	1	?
Geometries	1	2	2/2	2	2/2	-1	1/1	2/2	1/1	2	2
Deformations	0	-	2/2	2	?/?	0	-2/-2	1/-2	2/2	-2	2
Animation, t	0.5	2	2/2	2	0/-2	2	2/1	2/0	1/-1	2	2
Animation, xyz	0.5	2	2/2	2	2/1	2	-1/-1	2/0	0/-1	2	2
Many problems	1	2	2/1	2	-2/-2	2	2/2	1.5/-2	2/2	2	2
Thresholds	2	1	2/1	2	0/0	2	1/1	1.5/-1	2/2	0	2
Erasure	0.5	2	0/1	2	0/-1	2	0/-2	2/2	1/1	2	2
Streamlines	1	2	2/2	2	-2/-2	1	1/1	1/-2	1/1	0	2
Pathlines	1	-2	2/2	2	-2/-2	-2	-2/-2	1/-2	1/-1	0	2
Test Data	1	2	2/0	2	2/1	1	1/1	1.5/1	-1/1	1	1
Line Data	-1	2	2/1	2	1/1	2	2/1	1.5/-2	2/1	0	2
Symmetry	1	1	2/2	2	2/1	1	2/1	1.5/1	1/1	0	2
Input	1	2	1/1	2	2/1	0	2/1	1.5/1.5	1/1	1	1
Output	2	2	2/2	2	0/0	1	2/2	2/1	1/1	1	2
Memory	2	1	2/2	2	2/0	2	1/-1	2/1	-2/-1	2	2
Large-Data	1	1	2/2	2	-1/-1	2	2/1	1/0	1/1	1	2
Remote	2	0	2/2	2	-2/0	1	0/0	1/0	0/0	2	1
MP	0	-2	-1/-2	1	?/-2	0	0/0	1/1	-2/0	0	0
Extras	N	N	N/N	N	N/N	N	N/N	N/N	N/N	N	N
Who	N	N	N/N	N	N/N	N	N/N	N/N	N/N	N	N
Context	N	N	N/N	N	N/N	N	N/N	N/N	N/N	N	N

2 Evaluation Methods and Philosophy

Summary

Each tool is to be evaluated against an explicit set of desired features, listed in Section 2. Each feature is named with a short phrase, and expanded upon by a series of questions. These questions should be asked of the tool under investigation, and addressed by the evaluator in their comments on that tool/feature pair. These questions will also guide the evaluator in assigning a numerical summary score for that feature, on the scale (-2,-1,0,1,2).

Numerical vs. Descriptive Evaluation

It might be asked why a numerical summary score is appropriate, given that a detailed description of what a given tool does toward satisfying a particular functional requirement can be a lot more useful than a numerical score. An excellent example of this principle is the SARA report (<http://www.sara.nl/Rik/REPORT.update>) by Sander Belien and Rik Leenders, which also compares visualization packages. There are no numerical scores or evaluation matrices in this document, but reading it nonetheless provides a huge amount of useful and directly relevant info.

However, for *summarizing* and comparing information, a numerical description is valuable. The SARA report itself didn't draw any final conclusions as to which tool best met their needs, and its summary and comparison section was (necessarily) a bit disorganized and difficult to absorb.

Further, that report was only looking at four tools. The current effort examines ten candidates (from an initial field of forty-one), making systemic comparisons all the more complicated. Accordingly, a summary numerical matrix, in which one can easily note trends and patterns, and which can be (admittedly crudely) summarized by a weighted average, will be a very useful adjunct to feature/tool descriptions.

Description of Numerical Evaluation Scales

One difficulty in assigning a numerical summary evaluation of how well a given feature is supported by a given tool is that there are different dimensions along which a tool can fail or succeed. The approach here will be to use a single fixed range of (-2,-1,0,1,2), but to permit that range to have a variety of meanings in different contexts. If it is unclear, or particularly pertinent, the particular context intended can be spelled out in the accompanying comments.

Some sample contexts (and more will likely be added as tools are evaluated) are as follows:

Implementation Difficulty: For application development tools and objective features such as “Line Data” (Section 3), the ratings would mean

- 2 This feature would be very difficult to implement with this tool
- 1 This feature would be difficult to implement with this tool
- 0 This feature would be possible to implement with this tool.
- 1 This feature would be easy to implement and/or is partially implemented with this tool
- 2 This feature is already completely supported by this tool.

Level of Support: For turnkey (non-extensible) tools and objective features, the ratings might be interpreted as

Descriptions of the Required Features

- 2 This tool does not at all support this feature.
- 1 This tool provides only crude support for this feature.
- 0 This tool provides basic support for this feature.
- 1 This tool provides extensive support for this feature.
- 2 This tool provides unlimited support for this feature.

Subjective Features: For more subjective features, such as “Simple” (Section 3) the ratings would mean

- 2 I strongly disagree that this tool possesses this feature
- 1 I disagree that this tool possesses this feature
- 0 I am ambivalent on whether this tool possesses this feature
- 1 I agree that this tool possesses this feature
- 2 I strongly agree that this tool possesses this feature

In all cases “?” will be a permitted value, where “?” means “I don’t know, I could not determine, this feature/tool combination is not meaningful, there was insufficient data, etc” depending on context. “—” will be used to indicate that the evaluation is still pending.

Other Evaluation Method Issues

The Evaluation Environment

We recognize that the evaluation hardware environment will make a difference to the apparent usefulness of various features, and that this will be difficult to control for, as the various evaluators will be working in different contexts. Accordingly, the evaluators will record the environment in which they do the evaluation, and are encouraged to make subjective comparisons to how other tools have fared in the same environment. In fact, in general comparisons to other tools are welcome.

Emphasize Noncompliance

It is generally best to organize the review of a feature/tool pair by listing the desired properties that the tool does *not* have. This will help address the ambiguity inherent in claiming “the tool has properties A and B”, when the feature description requires properties A, B, and C. The previous sentence probably means that property C is not supported, but it is somewhat ambiguous. “The tool lacks property C but other meets the requirements of this feature” is just as terse and not at all ambiguous.

3 Descriptions of the Required Features

Generic Features

Support

Is the tool fully supported? Is there someone dedicated to answering questions and fixing bugs? Are bugs given a reference number and tracked until final resolution?

Descriptions of the Required Features

Documentation

Is there a printed users manual? An on-line user manual? Are simple generic test datasets and configuration files (for users to analyze and become familiar with) provided?

Turnkey

Can the end-user access any of the features discussed in this document by invoking an executable, *without* writing any code? (Other than a data converter?) For tools which are innately application development environments, can they generate such stand alone executables?

Simple

Is the tool straightforward to learn and use? Is it easy to discover how to do something with the tool? Does basic functionality have a short learning curve? Are advanced or rarely used features hidden until needed, or otherwise organized so as to not hinder everyday use?

General Data Features

Variable Types

Can the tool visualize both scalar and vector data? Both cell-based and node-based data? Null data?

Axis

Can the tool place an (x,y,z) axis in the field of view? Can the user fix its size and position?

Query

Can the user interrogate any point in the viewing window, determining its (x,y,z) location as well as its mesh vertices, node numbers, and the values of any variable at that point?

Scale

Can the user “measure” the size of any object in the viewing window? Can the user bring up an axis or bounding box with tick marks to determine the extent of the field being viewed?

Annotation

Can the user add a title and simple annotation to the viewing window? Can the annotation be ephemeral, attached to only the viewing window and possibly some output graphic? Can the annotation be attached to and stored with the data itself, so that it will reassert itself when the data is revisited?

Algebra

Can the user perform simple algebraic manipulation of the data (add, subtract, multiply, divide, div, curl, etc, etc)?

General Visualization Features

Viewing

Can the user view the data from any angle and distance? Can the user set up multiple views? Multiple light sources? Orthographic projections on any view?

Cut planes

Can the user place a plane oriented in any fashion which slices through the data field? Can any scalar or vector field can be plotted on this plane? Can its color map be adjusted? Can the cut planes provide color shaded mappings, contours and isolines? Are an arbitrary number of cut planes allowed? Can a cut plane be animated along an arbitrary direction in space? Is a legend provided?

Isosurfaces

Can the user can make an isosurface of any scalar variable? Can any scalar or vector field can be plotted on this surface? Can surfaces be arbitrarily colored, or colored by any other variable? Can the surface be animated though a range of values? Is an arbitrary number allowed? Is a legend provided?

Isovolume

Can isovolumes be constructed from any scalar? Can any scalar or vector field can be plotted on this plane? Can the user select the range of values and adjust the color map? Can the volume be animated though a range of values? Is an arbitrary number allowed? Is a legend provided?

Geometries

Can predefined geometrical objects can be plotted and colored as desired? (The geometrical objects might be, e.g., material interfaces or meshes.) Can the geometries can be copied, moved, and reflected independently? Is an arbitrary number allowed?

Geometry Deformation

Can the geometry deformation vector be visualized relative to the initial geometric state, scaled, and rendered with arbitrary scalar or vector functions? Are attributes retained for each new time state?

Animation (time)

Can the user do time animations of anything that can be plotted at a specific point in time, to include geometries, cut planes, isovolumes, and isosurfaces? Is the range and number of frames user selectable?

Animation (space)

Can the user do spatial animations (rotations, zooms and general motions through the data field) of cut planes, geometries, isovolumes, and isosurfaces? Is the rotation point user selectable? Is the range and number of frames user selectable? Can space and time animations be combined?

Descriptions of the Required Features

Multiple problems

Can the user read in as many different calculations as desired on as many different meshes as desired?

Thresholds

Can the user threshold the rendered data on a range of values? Is this feature present for each variable type supported by the tool?

Erasure

Can the user erase or remove any part of the data volume wished? Can the erasure be animated? Does it have 2D and 3D capability?

Streamlines

Can the user construct and display streamlines and streamtubes for vector variables? Does the tool support particle animation?

Pathlines

Can the user construct and display pathlines (time dependent streamlines) for vector variables? Does the tool support particle animation?

Point Data

Can the user display data defined at specific points in space and representing a particular data variable? Can the user display point sets (such as experimental data)? Does the tool support scatter plots, clouds, shaded glyphs?

Line Data

Can the user select an arbitrary line through a data field and make an x-y plot of any particular variable along this line? Can the user select any point and make a time history plot of any variable? Can this data be separately stored to disk?

Symmetry

Is symmetry supported by the tool? Is it efficiently implemented?

Interfaces

Input

Can the tool accept a wide variety of meshes, specifically including structured, unstructured, rectilinear, curvilinear, multi-block, AMR? Does the unstructured mesh representation support, at the least, hexes,

tetrahedrons, prisms and sets of same? Can it accept point data? Can it read simple ascii files that can easily be constructed by the user?

Output

Is it possible to write out any object being viewed? Is it possible to write out each frame of an animation as it is displayed? Are a wide variety of standard output formats available? Can the tool be upgraded to eventually support the “ASCII Visualization” format under construction by the ASCII Visualization Data Formats Group?

Memory

Can the tool “remember” settings? That is, does it have a resource file that can be recorded automatically, or a scripting feature, or journaling capability, or some equivalent functionality?

Computer Features

Large Data

Does the tool gracefully handle large data sets? Can the user select and deselect portions of the data to load? To view? Can the user load in variables and when finished, free up the memory? Can it run on multi-processor platforms?

Remote

Does the tool support having the visualization number crunching on one machine and the viewing/rendering on another? Is this capacity cross-platform? Is it X compatible?

MP

Does the tool run on an SMP box and take advantage of the multiple processors? What portions of the code are parallelized in this fashion?

4 Tool Descriptions and Evaluations

4.1 AVS Express

Code Description:

- From David Andaleon: “We have created modules within AVS to satisfy our viz needs, such as an Exodus II reader, so we prefer referring to it as AVS-based tools”
- From Mike Glass: “The next generation AVS-based viz tool is in “beta” mode right now and has been getting good use in the engineering sciences center. It’s built with AVS/Express. Express is a totally different environment than the AVS that most people are familiar with. It has its own graphics programming language so saying an application is built with Express is

4.1 AVS Express

like saying an application is built with C++. It's a complete Motif GUI driven application. Also, AVS has a person there that they hired from Thinking Machines and has a couple (or plans to) of graduate students working with him on an MPI layer to Express targeted towards accessing data on MP machines and eventual visualization on MP machines."

- "AVS consists of five interactive applications, viz. Geometry Viewer, Image Viewer, Graph Viewer, Data Viewer, and Network Editor. Each viewer can be run as a stand alone turnkey application."

Hardware Platforms: Data General, DEC, Evans & Sutherland, HP, IBM, Kubota Pacific Computer, Set Technologies, Silicon Graphics, and Sun Microsystems, Convex, CRAY Research, and Intel.

Licenses: \$6,500/user for AVS "Lite", \$25,000 for first license and \$6,000 for additional licenses of the full AVS.

Contacts:

- Stephany Bouchier, scb@lanl.gov
- Mike Glass, mwglass@sandia.gov

E-Contact:

- <http://www.avs.com/>
- http://sslabor.colorado.edu:2222/projects/AVS_toc.html
- For a non-ASCI evaluation of AVS, see http://sgrabber.cern.ch/~lecointe/AVS_evaluation.html

Disclaimer: The recommendations in this document are made with specific consideration of ASCI requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Stephany Bouchier (main contact), with Pat McCormick, Chuck Hansen, and John Fowler (on the development of the application prototype described below under "comments about evaluation")

Extra Features and Comments: AVS/Express is not a turnkey application, but instead provides tools and an environment for the development of end-user applications. In addition to examining specific visualization capabilities, our evaluation of Express needed to assess the difficulty and flexibility of:

- building applications using modules provided with the packages,
- writing and integrating custom modules,
- developing and integrating graphical user interfaces for applications, and
- modifying applications which have been developed using the packages.

To evaluate these aspects, we are developing a prototype application using AVS/Express. At the time that this evaluation report is due, the prototype has not been completed, but things we have learned during the development work are an integral part of our evaluation and are discussed here.

Ed Bagdonas, a consultant for AVS, spent a week with us developing the initial prototype. Prior to this week, we had designed the prototype, modeled loosely after the turnkey product Data Visualizer

(DV), from WaveFront. The prototype was designed to give a user access to separate tools for different functions, such as applying an isosurface or a cut plane to data. As in DV, we wanted to allow the application of an arbitrary number of tools and also to provide the ability to delete or temporarily disconnect any chosen tool.

Our design was from the user interface point of view. As the week with Ed progressed, it was obvious that design of an Express application requires more of an understanding of the software's internals than we initially possessed. However, it was to learn such things that we hired a consultant, and we gained valuable experience for any future development.

By the end of the week, we had put together the top-level user interface, much of the internal infrastructure for control of the various subcomponents, and a subset of the individual tools. The following week, we realized that we didn't fully understand all of the controls that Ed had implemented, and we struggled to continue the development. After another week and hours of long-distance consultation, it was discovered that Express 3.0 has a bug in one of the main components we are using for the control infrastructure. We've been given a temporary way to work around this bug and are waiting for the actual fix. In the meantime, we are continuing to develop the individual tools, hopeful that it can all be put together readily when the true fix is applied. We also have plans for Ed to return for another week to help us wrap up the development. This second week with his help will take place between the time that this report is due and the time of the face-to-face meeting planned for discussing all the evaluations, so further impressions of Express can be conveyed verbally during the upcoming meeting.

The existence of a bug in a key module adds to the difficulty of assessing our experiences in developing the prototype. It clearly hampered our development, as well as our confidence that we were gaining an understanding of how to use Express effectively. It's hard to say how much this should affect the subjective evaluation results, especially now that we're expecting the fix "any day now." Our experiences after the fix is applied will be disclosed in the face-to-face meeting.

Because a major purpose in developing the prototype was to evaluate application development and not just visualization capabilities, only a small subset of the specific visualization features addressed in this document were included in the prototype. Features which were not included in the prototype were tested in simple configurations.

Evaluation Hardware Environment: Evaluations were conducted using a mixture of SGI hardware configurations, with all machines running IRIX 6.2. Most of the development work (described in the following section) was done with a local installation of Express 3.0 on SGI Indy workstations, with a single processor (100 MHZ IP22), main memory of 32 Mbytes, and 8-bit Indy graphics boards.

Most of the stand-alone evaluations of specific visualization features were done remotely using Express 3.0 running on SGI Reality2 Engine, with 4 processors (200 MHZ IP19), 4 processors (150 MH IP19), and main memory of 704 Mbytes, 1-way interleaved from the local machine, which was a SGI Onyx Reality2 Engine, with 4 processors (150 MHZ IP19) and main memory of 128 Mbytes, 2-way interleaved.

Support (Rating: 1) This is difficult to rate because we believe we received special treatment due to our association with a consultant for our prototype work. However, Ed Bagdonas and Peter Stark, another consultant, have been fairly quick to respond to questions and diligent at tracking down the bug. They also gave us a temporary way to work around the identified bug until it is fixed.

We understand that the regular technical support through AVS is swamped and that service can be slow, but we don't have first-hand experience with it.

4.1 AVS Express

Documentation (Rating: 2) Printed manuals include:

Getting Started	User's Guide
Developer's Reference	User Interface Kit
Data Visualization Kit	Annotation and Graphics Kit
Graphics Display Kit	Database Kit

In general, this documentation is thorough and well-written. The full document set is also on-line, with an interface which provides four ways to access the information: a hierarchical approach through the table of contents, an index, a find option, and a "help on selected item" option.

AVS/Express comes with many useful examples.

The developer of any end-user application would need to provide user documentation for the specific application. Using the GUI tools of Express and perhaps tying into the documentation provided with Express, this should be fairly simple to implement.

Turnkey (Rating: -2) An end-user would not find it easy to generate a stand-alone executable, except for a very simple application (ie, something with only a single function, such as to produce an isosurface). To develop a full-function, flexible application requires in-depth understanding of AVS/Express and lots of time.

Simple (Rating: -1) For an end-user who does not want to learn the details, AVS/Express is practically impossible to use, except for very basic functionality. For an applications builder, it is also very difficult. Even with experience and a good understanding of the software, a full-functioning and efficient application would require a lot of time to develop. However, it would be possible to create a flexible end-user application which is straightforward to learn and use.

To their credit, AVS makes no claims that Express is simple to use. They encourage new users to take their training classes. The week-long class I attended provided very good basic instruction in Express, but it wasn't until we later started on the prototype development that I felt I had a good understanding of how to use it to create a comprehensive application.

The more I work with Express, of course, I am finding it easier to use. I believe it could be a satisfying package to work with once I've gotten over the learning hump.

Variable Types (Rating: 2) Express can handle scalar and vector data, either cell-based or node-based. It also handles null data.

Axis (Rating: 2) This feature is easy to add to a user application. We used it while developing the symmetry tool of our prototype, although without providing an interface to give a user control of size and position. These controls should be relatively easy to add.

Query (Rating: 0.5) Express has a probe tool to provide this functionality, but I found it difficult to use. With some effort, I'm sure this could be included in an end-user application.

Scale (Rating: 0.5) Express provides a way to place a grid plane orthogonal to the x, y, and/or z axes. This feature includes controls for details like the number of tick marks or grid lines upon the planes or the visibility of any of the planes. These planes could be incorporated in a tool for measuring objects in the viewing window.

Annotation (Rating: 1) Flexible annotation capabilities already exist in Express.

4.1 AVS Express

Algebra (Rating: 1) The add and subtract functionalities already exist. Others would be quite easy to implement.

Viewing (Rating: 2) Express provides great flexibility for viewing, allowing easy rotation, translation, or scaling, as well as multiple views or light sources.

Cut planes (Rating: 2)

Isosurfaces (Rating: 2)

Isovolume (Rating: 2) Express has a wide variety of capabilities for all three of the above features, so any options should be easy to provide in an end-user application.

Geometries (Rating: 1) Express allows an arbitrary number of predefined objects to be plotted and colored. Adding the ability to copy, move, and reflect these objects independently will take some development effort, but should not be too difficult to implement.

Geometry Deformations (Rating: 0) Any individual geometry can of course be visualized with Express, so the ability to deform geometries could be implemented. However, I can find no such capability already available.

Animation (time) (Rating: 0.5)

Animation (space) (Rating: 0.5) Either type of animation could be implemented in an end-user application.

Multiple problems (Rating: 1) This should be fairly easy to implement in an end-user application.

Thresholds (Rating: 2) Express provides two ways to threshold - one sets all values outside the range to a specified null value, the other sets all values outside the range equal to the minimum or maximum threshold level.

Erasure (Rating: 0.5) Express has a cut plane capability which erases data on either side of an arbitrarily-placed plane. Multiple cut planes can be combined, so I'm confident that a volume eraser could be rather easily created.

Streamlines (Rating: 1)

Pathlines (Rating: 1)

Point Data (Rating: 1) Express supports some functionality for each of these. Additional capabilities could be added.

Line Data (Rating: -1) Express has some capabilities which should help in implementing this feature, but I couldn't get them to work. Although I believe it would be possible to implement this, I'm giving it a low rating based on my failure.

Symmetry (Rating: 1) This feature is not already implemented in Express, but we were able to implement it in our prototype in a reasonably efficient manner.

4.1 AVS Express

Input (Rating: 1) Express already has implementations for accepting structured, unstructured, rectilinear, and multi-block meshes. Curvilinear and AMR do not seem to be specifically addressed. However, by creating custom data readers, these could probably be implemented. We are currently working on a reader for AMR data and should be able to report on it at the face-to-face meeting.

Mesh representation supports all of point, line, triangle, quad, tetrahedron, pyramid, prism, and hex. Express can easily accept simple ascii files constructed by the user.

The AVS/Express API provides functions that ease the writing of custom data readers and writers. These include functionality such as: open, close, get/set width/height/depth, and set file-type/colortype.

Output (Rating: 2) Any image rendered in Express can be written to a file. The following output formats are currently implemented:

- AVS .x format
- Microsoft BMP format
- GIF (Graphics Interchange Format)
- JFIF (JPEG File Interchange Format)
- Portable Bitmap Utilities
- SGI Image format
- Sun Rasterfile format
- TIFF (Tag Image File Format)

Custom writers for additional formats could be created. See the comments on the API under the input section above.

Memory (Rating: 2) Some effort might be required to provide journaling in an end-user application, but because the Express development environment has a good capability, it should be relatively simple to provide.

Large Data (Rating: 1) I have not experimented with very large data sets, and of course the performance with large data will depend heavily on memory size of the machine. However, an Express application can be built to allow flexibility in loading, viewing, or freeing portions of the data.

Remote (Rating: 2) Express does not run in a client-server mode, but it does allow execution on one machine and viewing on another. It is X compatible, allowing cross-platform usage. It attempts to use the best rendering capability, but reverts to something common when not available. When I ran from an SGI to a Sun, I got the message “no GLX extension on this X server” but it worked fine without it.

MP (Rating: 0) A development effort for MP/Express is currently underway at AVS. Email to the developer, Gary Oberbrunner, yielded this response:

MP/Express is going well; I have some very preliminary demos of parallel modules working, and right now I'm working on the parallel array distribution stuff. I expect to have something decently demoable by Supercomputing . . .

The folks at Manchester are in the process of integrating their parallel array library into MP/Express; this will be a big milestone that we hope to have working before

4.2 Data Visualizer

Supercomputing. At that point we should be able to start doing some useful work with MP/Express, at least with regular grid data sets.

At this point we're looking for an initial release date in Q2 of 97, depending on the AVS/Express release schedule. You folks could probably get it a quarter earlier, if you don't mind a rough edge or two.

The rating given for this feature of course depends on the outcome of Gary's development work. I've given it a zero out of an optimistic hope that he will provide a good solution.

4.2 Data Visualizer

Code Description: "Data Visualizer is a so-called turnkey application. This kind of application allows the user to supply data and select from a fixed set of operations, hence functional extension is not supported."

Hardware Platforms: Silicon Graphics, DEC, Sun, IBM, and HP.

Licenses: Commercial product from Wavefront Technologies, which was recently purchased by SGI. SGI has discontinued support for Data Visualizer, and are currently trying to spin the product off to a third party. There is no news that any deal has been concluded, though.

Contacts:

- Gary Dilts (505-665-0190, gad@lanl.gov) — Gary brought the product into LANL and interfaced it with their hydro code. He has used it and is familiar with its inner workings.
- Bob Kares (505-667-7789, rjk@lanl.gov) — he uses the product.
- Bob Shea (505-665-3540, shea@lanl.gov) — also a user.

E-Contact: None.

Disclaimer: The recommendations in this document are made with specific consideration of ASCI requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Bob Shea.

Evaluation Hardware Environment: ???

Support (Rating: -2) This product is not supported.

Documentation (Rating: 2) Documentation consists of a User's Manual, a Programmer's Guide and a set of lessons for new users. The Lessons are basically a self teaching guide. The Programmer's Manual explains the format of data files, how to interface your own C-code to DV, etc. There is no on-line help.

Turnkey (Rating: 2) This is a turnkey application. It does have a widget builder and an interface to manipulate the data with your own C-code. I played with the widget builder one afternoon. As far as I know no one has exercised the C-interface.

Simple (Rating: 2) The great glory of DV is that it is so simple. The product is organized around a set of tools. For DV an isosurface is a tool, a cut plane is a tool, an axis is a tool, etc. The user can create as many of these tools as he desires. Tools can also be copied or deleted. Whenever you bring up one of these tools there is a menu where you can move, rotate, scale and control the color map of the tool. What is so nice about DV is that this interface and the options available are exactly the same for each tool. There is also a submenu for each tool which is specific to the tool: do you want color shaded cut planes or contours, etc. A lot of thought has gone into the user interface. It's simple, clean and powerful.

Variable Types (Rating: 2) The code supports both scalars and vectors. Node and element data can also be written out. DV does not allow the user any ability to color this information arbitrarily. Changing colors is very helpful for keeping track of things or in even being able to see the information if there is some background in the view.

Axis (Rating: 2) You can create as many axes as desired. They can be sized and positioned arbitrarily. There is no provision for putting tic marks on an axis but you can control the length.

Query (Rating: 1) One can use a probe tool to query the dataset. The position of the probe can be controlled in the window or by typing in the position. The probe can return information on the xyz location or the value of any scalar or vector variable at that point. It can also be used to determine node and element data. You can choose to display the element in which the probe is located. Interactively moving the probe causes the element display to jump to the next element as the probe moves. A series of points can be interrogated using the line tool as described below. The data appears only in the viewing window next to the point selected.

DV cannot probe in time. You could probe each file in a time sequence to get the same information. This certainly awkward. Also if you want values for several variables, you need change to variable associated with the probe. The probe data writes to the screen next to the position of the probe. It is often difficult to read the numbers against the colored background. Of course you can turn off the background but you have no control over the color of the numbers printed out. In all I think EnSight does a better job here.

Scale (Rating: 2) With DV you can bring up a bounding box that contains the data field and is essentially the outline of the mesh. You can ask for sizes to be marked on the edges. Depending on the orientation they may or may not be readable. You can also put a grid on the bottom of this bounding box.

DV also has a ruler tool. This tool can be positioned arbitrarily in 3D space and its length can be fixed. You can put tick marks on the ruler as well as its full length. It can be used as any ruler to measure the size of a feature.

Annotation (Rating: 2) DV has a simple annotation capability. One can put several lines of text on the viewing window. Text can be positioned by typing positions or dragging with the mouse. It can be colored as desired. Text can be 2D or 3D. Superscripts and subscripts are supported. EnSight does not support the subscripts and superscripts but it does allow arrows to point to specific features. EnSight also allows the user to have a variable in the text string that gets updated when the view or position changes. EnSight also allows an arbitrary title for the color bar while DV uses the name associated with the variable on the data file. This latter name is usually obscure and not readily understandable by others. EnSight also allows an arbitrary placement and orientation of the color bar. DV has something like this in the last release but I could never get it to work.

Algebra (Rating: 2) DV supports the usual algebraic operations (+,-,/,*,**) as found in Fortran. They also support trigonometric and hyperbolic functions, logs, divergence, curl, min, max and vector operations. They do not have things like the area of an isosurface that are in EnSight.

Viewing (Rating: 2) Control of the view is again a feature that is superbly done with DV. The view can be controlled with the mouse: the left button is for x and y, the right button is for z. The user has the same choices as in other menus: select whether you want to translate or rotate. In addition to control with the mouse, it is also possible to type in numbers directly. This allows precise orientation of the view. Once the view is oriented properly, you still have the mouse buttons controlling the xyz translations and rotations in the original axis system. At this point the user has lost control: the mouse buttons are referring to a coordinate system different from what he sees on the screen. Never fear. A click on “define view” changes the meaning of the buttons so that the left button again controls up/down and left/right on the view. The right mouse button controls the direction perpendicular to the screen. So simple, so useful.

The pivot point (or point you zoom in to or rotate around) can be set with a probe tool. Bring up the probe tool, position it where desired and click on “pivot point”.

Positioning objects in 3D space is tricky and by just looking at the screen it is impossible to get it right. Both DV and EnSight have a multiple view option where the main viewing window can be split into 2 or 4 subwindows giving different views of the same scene. This greatly facilitates the placement of things like a probe exactly where you want them. By the way both products allow the user to display different things in these windows (say a cut plane in one, an isosurface in another, etc.).

Cut planes (Rating: 2) DV can have an arbitrary number of cut planes, oriented in any way. In contrast to EnSight controlling the position and orientation is very simple. Though it can be done with the mouse I prefer to type in directly the rotation angles and position along the axis perpendicular to the view.

Since one can make a copy of any tool by clicking on “copy”, I usually get one cut plane oriented and positioned correctly and make copies. It is a simple matter to then click on the variable being plotted and change it to something else. With a few seconds I can generate several cut planes all representing different variables.

Cut planes can show data for each cell in the cut plane or on a sampling grid. In wireframe mode each individual cell is outlined. In shaded mode colors are smoothly interpolated from one cell to the next. The data can also be plotted as contours. The number, color and value for each contour is easily controlled. In general I found that DV did a better job than EnSight in the way it did a color shaded cut plane. Part of the problem lies with the lousy choice of default color map in EnSight; part lies with the way the thresholding is done. DV also has a simpler way to adjust the color map. A color bar can be switched on. With EnSight it is always on and must be turned off if you don't want it.

Isosurfaces (Rating: 2) DV supports isosurfaces. The isosurface can sample any scalar variable, and the number of isosurfaces is unlimited. The value of the scalar can be animated through a range of values. The isosurface can be given a fixed color, a color corresponding to the scalar it was created from, or colored according to a second scalar's value. You can apply a vector grid to an isosurface. The eraser option works with isosurfaces. A color bar can be turned on.

Isovolume (Rating: 2) Isovolumes have all the same features as the isosurfaces described above. In addition one can select the range of values to be displayed in the isovolume. The user can also

select to create an isovolume where all the values are above a certain level or all the values are below a certain level. I have found it useful to create an isovolume with values in a certain range and then animate it as the lower value approaches the upper value. Again a vector grid can be applied to an isovolume. One could create an isovolume of density, colored by pressure and with vectors representing velocity attached to it.

Geometries (Rating: 2) An arbitrary number of geometries can be read into DV, colored as desired. They can be in wireframe mode or shaded. The user has the same functionality as with other tools: copy, reflect, rotate, translate.

DV treats the computational grid differently than other geometries. It cannot be reflected or moved arbitrarily.

Geometry Deformations (Rating: —)

Animation (time) (Rating: 2) A sequence of files can be read in by clicking on the first file in the sequence. One can then form a cut plane or isosurface, etc., and step through the sequence to watch the representation change. One can annotate the frame with a sequence number. The user can select the number of frames. The user can select one frame for each time step or decide to interpolate between time times (for smoother animation).

Animation (space) (Rating: 2) DV has a very simple way to do this. One selects the view and tools wanted, clicks on a begin sequence button, then go to the final view and click on an end sequence button. Clicking on the animate button, shows the full sequence. The default is 30 frames but this can be easily changed. Several keyframes can saved to build sequences of animations. I haven't tried this multiple keyframe option.

On nice thing about DV is that pretty much everything can be animated, and animated at the same time. It would be very simple to have multiple cut planes moving in different directions and have the view animated at the same time. The only thing I found that could not be animated is the color map.

The pivot point can be controlled by the probe tool. Move it where you want, click on pivot point. The probe location becomes the point that you into and rotate around. This is very convenient.

Multiple problems (Rating: 2) Despite what is in the SARA report, DV can read in many problems. I think the guys at SARA just aren't as smart as Gary Dilts! The only catch is that you need to give the additional variables different names. For example, if you read in "density_material_1" from one problem and then read in "density_material_1" from another problem, the first data will be lost. If, however, you read in "prob_a.density_material_1" and "prob_b.density_material_1" you will have two data fields to plot and compare. Memory is the only limit on the number of problems you can read in. I use this feature a lot.

Thresholds (Rating: 1) The user can select the limits of the color bar. Values outside this range will be pegged at the color at the limits. Each tool, however, has a color slider separate from the color bar which can be adjusted such that values off the scale assume the color of the background. Some of the same functionality as EnSight but a bit more awkward.

Erasure (Rating: 2) DV supports a 2D eraser (a plane) and a 3D eraser (a box); the eraser can be sized and positioned arbitrarily. You can erase inside or outside of the eraser. Geometries, meshes, isosurfaces, and isovolumes can all be erased. The user can control this separately for each geometry, isosurface, etc. At one time I thought that this was a just a nice feature. Then I was trying to understand a code

problem and had to focus in on a single cell. The only way to do this and see anything is to erase everything outside that cell. The eraser is also useful with a solid body such as an isovolume. By moving the eraser plane down through the isovolume, you can see what the inside of the isovolume looks like.

Streamlines (Rating: 1) DV has an option to launch particles. You can follow vector arrows or spheres moving in space. You can opt to leave the particle track on the screen, essentially creating a streamline. There is an option that makes the streamline look like a ribbon. The particles can be launched from an arbitrary box, line or point. I don't use streamlines often and am not adept at it. I could never get the DV display to look as good as what I remember from EnSight. From DV it seemed that the particles were only willing to go so far and then they stopped. Also the display was very slow. Of course EnSight was doing its particle traces on more than one processor with highly optimized code running on the Power Challenge instead of my workstation. In addition I could make the graphics look a lot better with EnSight.

Pathlines (Rating: -2) As best I can tell here is no support for this in DV.

Point Data (Rating: 2) Data defined at points can be read in and viewed. The points can be displayed as tetrahedrons, cylinders or spheres. The size of these objects can be uniform or scaled with respect to some scalar field. The objects can be colored with respect to a scalar field.

Data defined on a mesh can also be displayed as points. This provides a convenient way to sample your data and get a quick idea of what the data field looks like.

Line Data (Rating: 2) The ruler tool controls probing data along a line. The ruler is placed interactively or by typing in coordinates. An interactive feature allows the user to pop up a small window where a plot of any variable along the length of the ruler can be displayed. As the ruler is moved this small plot updates.

One can also bring up a gnuplot window to display the curves and write the data to a file. One can generate several curves on the same graph. As with nplot in EnSight, gnuplot is a little clumsy. Your best bet is to use things like gnuplot to check the data that is being sent to the file, then use the data file with something like xmgr to get a polished graph.

Symmetry (Rating: 1) In DV you can use the scale operation to reflect objects. It's a little indirect. DV can make a copy of something easily. Then, for example, scaling this copy by -1 in x will reflect the copy through the yz plane. This only works with geometric objects. If you do this on a cut plane and there is no data out there for $x < 0$, you will have a cut plane with nothing on it. EnSight actually will give you the reflected cut plane with the reflected data.

Input (Rating: 2) DV allows meshes to be composed of prisms, hexahedrons, or tetrahedrons. These elements are defined by giving points in xyz space and a list of how the points are connected. One nice feature is that for regular grids you only need to specify the node spacing in x, y, and z. For rectilinear meshes you only need to specify the x,y, and z components of the grid separately.

Geometrical entities are points, lines and polygons with from three to eight sides.

EnSight can describe the same types of objects but not as conveniently.

Output (Rating: 2) Images in the viewing window can be saved. Animation sequences can be saved. When an animation sequence is saved, a separate file is written for each frame in the sequence. DV supports Quantel digital video format, two separate Wavefront formats, Targa, Vista, PostScript(all types) and SGI's RGB format.

4.3 EnSight

Memory (Rating: 1) The user can turn on a journal which records user input into an ascii file. As with EnSight the language in the file is a readily understood shorthand. The files can be written from scratch, edited, etc. You can also save your whole session. This will write a file with your data, and all the tools created. This large file can be read in at a later time to get you back to exactly the point you were at. EnSight does a little better job at this kind of thing.

Large Data (Rating: 1) Again this is hard to deal with. DV does not have any capability to load and then deactivate variables. We are clearly going to need this feature. Using the same data set I used for the numbers discussed in the EnSight discussion I found that DV took about 94 MBytes. I didn't see any difference when I had 2 isovolumes as opposed to just one. Of course DV had all the variables loaded into memory: 5 scalars and 1 vector. It also had two meshes (one for the cell centered variables and one for the vertex centered variable).

Remote (Rating: 0) DV can operate in a server client mode. Running in this mode brings up a bounding box describing the data field and a volume of interest tool (a different colored box). One can then select the region of interest in the data field you want. The server ships this data over to the client. So it does run in a server client mode, but its almost trivial: the server really doesn't do very much.

MP (Rating: -2) No support for MP platforms.

Extra Features and Comments: Gary Dilts was responsible for bringing DV into Los Alamos about one and a half years ago. His interest was in having some tool available to use with a code he was developing. He asked if I would be willing to use DV in my own work. Up to this point in time I had done very little visualization. I tried once or twice to use EV (a locally developed product) on my SUN workstation. The experience was so painfully slow that I no desire to repeat it. In December of 94 I found that one of the SGI Indigo 2's that had come into the group had not been assigned to anyone. On Gary's advice, I moved it into my office and claimed possession. I spent roughly an afternoon working through the lessons provided with DV. By mid-December Gary had DV reading files from my code. With Gary's assistance I looked at one of my recent calculations. Within 20 minutes I found some things in the calculation that I had not expected and which seemed to explain some anomalous results. I became a true believer.

I have been using DV extensively ever since.

DV allows control of the background color but not to the same level as EnSight. No logo is allowed. The manipulation of the color map is easier with DV than with EnSight. DV allows quick and easy introduction of new levels on the color bar. Levels can be selected with the mouse and moved to a new position or deleted. The same same things can be done with EnSight but not as conveniently.

One significant advantage that DV has in manipulating tools interactively is that you have a choice as to which coordinates you wish to change. For example if you wish to move a tool in x, you can click off the y and z coordinates. Thus motion will only take place in x and the other coordinates are fixed. Similar features apply to rotations and scaling operations. For accutately positioning tools in the viewing window this is an extremely useful feature.

4.3 EnSight

Code Description: EnSight is a visualization system applicable to a wide range of analysis including computational fluid dynamics, structural analysis, combustion modeling, thermodynamics, etc. It

4.3 EnSight

assumes geometries are unstructured and can be used for finite elements and finite difference elements. It is designed to be used easily with time dependent, vector and scalar quantities. It utilizes network resources. The code is licensed for a server machine where the data sets reside and compute bound tasks are performed. The server transmits graphical data to a client where all interface interaction and graphics rendering is done using the clients built-in graphics hardware.

EnSight was first developed by Cray Research and released under the name MPGS in early 1989. CEI was formed in early 1994 and is currently 30% owned by Cray Research.

Hardware Platforms: Works very well with R3000, R4000, and R8000 SGI clients using GL protocols. Have had success with an HP client using PHIGS software. SUN clients require Solaris 2.4 with PHIGS and X11 software and has been installed on a client with mixed success. Company literature implies licenses available for Crays, DEC, and IBM servers.

Licenses: Commercial product, license available from CEI, Computational Engineering International, Inc, 919-481-4301. 1/2/4 user licenses for \$8000/\$9500/\$12,500, respectively. This price includes the first year of support and maintenance. Thereafter, support and maintenance is approximately 15% of the license fee.

A temporary license exists at LANL and SNL.

Contacts:

- Bob Shea, 505-665-3540, shea@lanl.gov
- Verlan Gabrielson, (510) 294-2184, gabe@ca.sandia.gov

E-Contact: <http://www.ceintl.com>

Disclaimer: The recommendations in this document are made with specific consideration of ASCII requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Bob Shea.

Evaluation Hardware Environment: I received a temporary license for EnSight this Spring. The product runs in a client/server mode. The server was installed on an SGI Power Challenge L with 512 MB of memory. The client was an SGI Indigo2 with 192 MB of memory and Extreme graphics. As I understand the workings of EnSight, reading the data and number crunching are done on the server; rendering is done on the client.

Support (Rating: 2) The tool appears to be very well supported. The company produces a quarterly newsletter giving information on new features and a little PR. During the trial period I had occasion to call their 800 number hot line. The calls were not answered by some operator but in fact by the programmers working on the code. In one case the manual was unclear and my problem was resolved after the person on the other end asked a few questions in order to define my problem. In another case they led me through a few steps in order to better define what was happening. This problem was more serious and they took my name and number. I received a call back within 10 minutes. They had a bug report which they thought might fix my problem and offered to send a new tape. I declined since my license was due to expire soon and the feature was not essential to my needs.

4.3 EnSight

In one of phone calls to the company, I was told they have a new release roughly every 6 months. One is a major release and one is a minor release.

For bug reports they would like the user to send in “command files”. These are discussed below in some detail. In general they are short ascii files generated automatically by EnSight. These files contain the commands entered by the user. One nice feature for us is that there is no classification issue with regard to these files.

Documentation (Rating: 2) In addition to the Learning Guide and Users Manual there is on line help. The tutorials and demos include a number of data files to work with. The 800 number will offer help and suggestions.

Turnkey (Rating: 2) This is a completely turnkey product. There are also no hooks for users to add on features. DV does have a hook for users to link their own C-code into DV. I have never found a need to exploit this feature.

Simple (Rating: 0) Not quite. This is a fairly complicated product. One problem I had is that it speaks a language which is very much like that of a finite element code or a CAD package. Users with this kind of background will probably be more comfortable with the product. I remember back to learning DV. Those tutorials took about half a day (for a sloppy run through). At that point one is functionally literate with product. EnSight takes a fair amount longer.

Variable Types (Rating: 2) No problem here. Scalars and vectors seem to be standard fare. It can write out cell and node data. One nice feature is that you can color the cell and node IDs differently. This is a big help in keeping things straight.

Axis (Rating: 2) In EnSight an axis is called a frame. You can have an independent axis of arbitrary location and size. Most frames, however, are attached to parts. These axis are then used to move and manipulate parts. For example one could use these multiple frames to create an exploded view.

Query (Rating: 2) The query feature is quite extensive and can be done in a multitude of ways; the basic mode is to bring up a cursor, move it some point and select the correct menu option. In the dialog window you receive xyz location, values of all active variables at that point, element number and nearest node number. A query can be done interactively. Move the cursor to the point in question, hit the keyboard and receive in the viewing window the value of any selected variable as well as a marker at the cursor point. When data is written on the view, it may not be visible due to the colors behind the point queried. EnSight allows the user to select the color for number written out as well as for the marker. As for all color selection in EnSight, you can choose values by typing in RGB values or by bringing up a color palette and clicking on color you want. For us dummies who don't think in RGB values this can be very useful. Queries can also be done by element number or node number. In this case information is limited to coordinate and node information. A superior feature of DV is that when moving a tool you have the option of turning off motion in the other directions so as not to inadvertently change these settings. Sounds simple but it is extremely useful.

Scale (Rating: 2) EnSight doesn't have a ruler like DV. You cannot bring up an object on the screen which looks like a yardstick. You can, however, bring up a line tool, position it arbitrarily, choose its length and accomplish the same thing. There is no bounding box with mesh extents. You can bring up a window which gives information on the datasets: mins and max, range of coordinates, number of nodes. You can also create an axis, position it arbitrarily, fix its length and indicate how you want the axis subdivided. An axis (EnSight calls them frames) can be associated with a part (such as a

4.3 EnSight

geometry that is read in) and periodicity can be associated with the frame. You can use this feature to make multiple copies of an object.

Annotation (Rating: 2) Annotation is very good. It's more complex than DV but can also do more. Sizes, colors and angles are user selectable via typing or via manipulation with the mouse. One can add arrows to point out various features of interest on the graphic. Users can also read in a bitmap to display a logo on the graphic.

Algebra (Rating: 2) There is extensive support for this feature within EnSight. Built in math function include the trigonometric functions, dot products and cross products as well as square roots. What are termed "general functions" include not only divergence and curl but flow variables such as Mach number, fluid shear stress etc. In total there are 30 or so "general functions". In one of the tutorials I created an isosurface and calculated the area of this surface. I tried to calculate kinetic energy which for my code involves variables defined on different meshes (cell centered and vertex centered). I would have been completely blown over had it been smart enough to complete the calculation. It flagged an error instead. No matter what, the computational features of EnSight are still pretty amazing.

Viewing (Rating: 0) The product offers full control over the view. The catch is that it's a bit difficult. Problems come up with the same orientation as a piece of graph paper: the x axis is horizontal, the y axis vertical and z is pointing at the viewer. You can rotate the view with the mouse. You also have the option of clicking an axis identifier and then pulling a slider to rotate about that axis. When you release the slider it snaps back to 0 and you are never sure if you got the angle you wanted. Rotations can also be accomplished with the keys F1, F2, F3. They perform rotations in 45 degree increments about x, y, and z. This is a pretty nice idea and I hadn't seen it before. It only works for what is called the global axis (basically the original xyz orientation). I much prefer the system in DV where you just punch in numbers to rotate about xyz. When you are at a point where it's natural to look at your data, you can click to reset the view. This resets the mouse functions to be relative to the new view and not the original one. Angles then snap back to zero since everything is now measured with respect to this new orientation. This system is much simpler and easier. In fact the way EnSight controls the view is the feature I like least about the tool. The control of viewing is so cumbersome, for example, that I found it impossible to zoom way in on a cell sized feature.

Cut planes (Rating: 2) Excellent support for cut planes. Compared to DV they are more complicated to move and orient to the exact position desired. One can have any number of cut planes. They can be reflected so that if you run a 1/4 or 1/8 problem you can view the data on the other side of the symmetry plane. (DV cannot do this.) Color bar can have whatever title desired, arbitrary number of tic marks and labels, arbitrary position and orientation, even more than one color bar. You can click to make the mapping quadratic or logarithmic, or you can adjust the map in some arbitrary fashion. Color maps can be saved. Several options are provided for how to treat the data out of the range of the color map. All of this also holds for contours. Animation support is also excellent. Flipbook and keyframing are supported. EnSight refers to cut planes as a clip part. Clip parts can be planes (the most common form), or clip lines, spheres, cylinders, cones, a general quartic surface or some user defined surface of revolution. I did create a clip sphere and then plotted cell average pressure on the sphere. Position and radius are user selectable. You can animate the position of the sphere and have it move through your data like a clip plane. Features such as a changing radius cannot be animated. There is a lot of flexibility here and I haven't yet figured out how to take advantage of it.

4.3 EnSight

Isosurfaces (Rating: 2) Pretty much the same as for cut planes. An arbitrary number of isosurfaces colored any which way you want. Easily animated through a range of values.

Isovolume (Rating: -2) A big gap here; isovolumes are not supported.

Geometries (Rating: 2) Yes, though the way my code was set up I did the easy thing and wrote out geometries as separate files rather than in one file. Thus different geometries became separate problems and the number of separate problems one can view is limited to 8. It should not be an issue for someone less lazy.

Geometry Deformations (Rating: 2) Any part (geometry, isosurface, cut plane, etc.) can be displaced by a vector variable. The displacement can scaled arbitrarily. The displacement will remain in effect over a time sequence.

Animation (time) (Rating: 2) The first thing I tried with my transient data was to do flipbook animation using what EnSight calls “graphical objects”. These objects contain 3D information. Thus while the sequence is running, they can be translated or rotated. The bad news is that all this information is loaded into memory on the client. I crashed the system with a relatively modest data set. Next I tried flipbook animation using “graphical images”. This is just a screen image and cannot be rotated. The images are much smaller and can all be loaded into memory. Of course during playback the sequence is much too fast and must be slowed down by at least a factor of 10 in order to see anything. After I had created the animation, I played with the color map just to see if I could make it more informative. When I went back to the flipbook sequence and tried to replay it, it got totally confused. Next I tried keyframe animation on the transient data. This is in fact the recommended approach. With the keyframe data you have access to the output features which include putting data directly on video tape as well as making RGB files. I wanted to check and confirm that it would write out the whole sequence. (It did.) I didn’t try writing to video tape.

There is an option within EnSight for writing dynamical data on the frames. One of the variables EnSight reads in is the solution time and EnSight knows which time goes with each frame. This is better than DV which can basically only sequence frames. This is not usually a problem for as I try to write out frames from the hydro every microsecond. In this case a sequence number is equivalent to a time stamp.

Animation (space) (Rating: 2) EnSight has extensive capabilities for keyframe animation. The user has the ability to identify keyframes and then interpolate between them with a selected number of frames. The user can build several such sequences by identifying a number of keyframes. The user can select the number of subframes for each sequence. You can also specify a number of commands (or even a command file) to be executed at each keyframe. This is much more elaborate than anything in DV.

The pivot point can be selected. Cut planes, geometries, and isosurfaces can be animated. The code does not have isovolumes.

Multiple problems (Rating: 2) Different problems on different meshes are called cases by EnSight. Limited to 8 cases. I certainly wouldn’t down rate it for limiting the number to 8. Each new case opens up a new socket connection between client and server.

Thresholds (Rating: 2) EnSight offers several options for thresholding. The limits on the color bar can be easily selected by the user. At that point he can have off scale values pegged at the color at the end of the color bar. He can also select to not display elements with off scale values. A third choice is color the off scale values by some arbitrary color.

4.3 EnSight

Erasure (Rating: 0) I had a hard time finding this and then it was just by chance. EnSight calls this feature Auxiliary Clipping. You bring up a plane which can be oriented arbitrarily and it removes that part of the figure on the +z side. Any “part” can be toggled on and off with regard to auxiliary clipping. The position of the clipping plane can be arbitrarily animated. Unfortunately this feature is only implemented with the plane. I have had occasion to look at my Eulerian mesh with material interfaces drawn through the cells. Having an eraser in the form of a box is essential for being able to look at that one cell where the problem is located. Yes, it could be done in EnSight by combining 6 planes with the proper orientation and position to remove everything but the single cell. I consider the operation too complex to even try.

Streamlines (Rating: 2) I was able to construct streamlines or particle paths for a velocity field. Streamlines can originate at a point, along an arbitrary line in space or on a plane. In the latter two cases you need to specify how many points on the line or in the plane will originate the streamlines. There are a large number of complex features connected with streamlines, due, I am sure, to a large number of users in the engineering world.

EnSight also has a lot of capability with respect to particle advection. I tried this feature on some of my data. There are no streamtubes in EnSight but users can select streamribbons rather than streamlines. The streamribbon feature is useful in identifying complex flow patterns.

Pathlines (Rating: 2) The manual details how to construct pathlines. This was a feature I did not explore.

Point Data (Rating: 2) The product has the ability to read in a set of predefined points and an associated data value or values. Particles can be sized by a scalar and/or colored by a scalar. I tried this but the code failed to read in my data file correctly. The points appeared on the screen at the anticipated locations but the scalar variable was set to zero everywhere. This appears similar to a problem I encountered while running through the feature demos. When working on the demo I called the hot line they informed me of a bug in the code. The bug has been fixed in version 5.5.3 and I have increased the rating from a 1 to a 2.

Line Data (Rating: 2) The user can bring up a line tool positioned and oriented as desired. It can be used to query any variable along the line. The data can be written to the viewport, to an ascii file or to a plot file where it can be displayed using a utility called nplot. This is very much the same as the gnuplot option in DV. The only advantage to DV is that a small xy plot can be brought on the main display to show values along the line. Then as the line is moved interactively the small xy plot displays the updated values. EnSight lacks this interactive feature for line data.

EnSight allows the user to select a point or node in his data and query that point over time. The user can ask for max and min over the whole data set or write out the values at a specified point as a function of time.

Symmetry (Rating: 2) EnSight has the ability to reflect any part in any plane or combination of planes.

Input (Rating: 1) The product assumes that 3D meshes are described by tetrahedrons, hexahedrons or pentahedrons (or some combination). These objects are defined by giving xyz locations in space and a list of how these points are connected. For simple equally spaced meshes or rectilinear meshes you still have to specify the nodes and connectivities. Geometries are described by polygons. The only polygons allowed are 2, 3 and 4 sided. For my code, which describes geometries as 3, 4, 5, or 6 sided polygons, I had to break up the higher order polygons into simpler ones. A bit more work but no limitation. Support is provided for some popular formats: ABAQUS, ANSYS, ESTET, FIDAP, FLUENT, Movie.BYU, MPGS, N3S, PLOT3D, and STAR-CD. These can be read in directly.

4.3 EnSight

An addendum from the vendor states that “EnSight describes Geometries in terms of elements. The following linear and quadratic element types are supported: 2 node bar, 3 node bar, 3 node triangle, 6 node triangle, 4 node quadrangle, 8 node quadrangle, 4 node tetrahedron, 10 node tetrahedron, 8 node hexahedron, 20 node hexahedron and 6 node pentahedron.

Output (Rating: 2) EnSight doesn’t support the semi-infinite number of options that one has with DV. They do give options for Apple PICT, PCL, PostScript, SGI RGB and TARGA. The user can choose to print images in black and white.

EnSight has the feature of dumping sequences directly to video. Bringing up the output options reveals Lyon Lamb Mini VAS, Laser Disc, VLAN and others. The recent EnSight newsletter describes how to hook your PC and home VCR to dump animation sequences directly to tape. They costed out the system at \$9200 (including \$4850 for the PC, VCR and video monitor). I haven’t tried these features but I was impressed by the write up they gave it and the fact that all this could be done so cheaply. (Their recommended source for the TV monitor and VCR was Circuit City.) They even gave hints on how to get the best quality and where you could pick up free software for scaling images.

Memory (Rating: 2) This is done extremely well in EnSight. The code automatically produces a file in /tmp containing all the commands entered (called a command file). The idea is that you have a backup in case of one of those rare system crashes. Reading in the file gets you back to where you were. The ascii file is easily understood. For example “view_transf: action translate” switches on the translation mode for manipulating the view.

You can also save the current state of the client and server. These full backups are going to be large files.

There is also a save view feature which allows the user to save the setup for the viewing window and then restoring it. This feature only operates on angle and position in the viewing window.

An unexplored feature links a command file to a keyboard key. This would be useful for frequently repeated tasks or for someone like me who runs a number of problems with parameter variations. This Macro facility would allow a standard way of analyzing a number of files.

Another facility allows the user to have some control over the way windows are positioned. Turning on this option will allow information on window positions to be saved for use on subsequent runs.

Large Data (Rating: 2) This is hard to answer at this point. I only have a rough idea what is going on with memory usage. I have monitored both server and client for my test problems. My cell centered mesh had 25 million nodes and my vertex centered mesh had 27 million. The meshes are hexahedral. One mesh had 211344 hexahedrons and the other had 223125. These are very modest (small) compared to what we might expect in the future. At almost all times my workstation was using more memory than the server. The two meshes were treated by me as different problems. That gave rise to two processes running on the server. With the meshes, four variables and an isosurface I was using 30 MBytes on the server and 34 MBytes on the client. Note that the client’s usage went up by almost 5 MBytes when I made the isosurface a solid surface rather than a wireframe. Thus memory usage on the client is very display dependent. At one point I asked to display the full mesh rather than the border. The client finally crashed after running up over 200 MBytes of memory usage. If our server had graphics hardware, it would have been interesting to try running both the client and server programs on the same machine.

4.3 EnSight

Remote (Rating: 2) Basic operation of EnSight is across platform in the client server mode discussed above. Naturally you can run both on the same machine. Servers can be Cray, Convex, SGI, IBM, HP, SUN or DEC. Supported clients are SGI, IBM, HP and SUN. HP and SUN clients require GL or PHIGS graphics libraries.

MP (Rating: -1) The only portion of EnSight which has been parallelized in this fashion is the streamline algorithm.

Extra Features and Comments: The product comes with a Users Manual and Learning Guide. I worked through the eight tutorials (each several pages) in about a day and a half. I also worked through most of the 43 feature demos which are more advanced and designed to emphasize specific features. My previous experience with this product was limited to watching Jennie Fox demo the product last summer using unclassified data. My initial reaction at the time was that, compared to DV, I found the product awkward.

I modified our 3D code to produce dumps that EnSight could read and used these dumps to investigate EnSight. I received valuable help from Jennie Fox with regard to debugging these files. Unless otherwise specified all the discussion above refers to features actually investigated with real data from a 3D safety code. The problems I looked at with EnSight had all been previously analyzed with DV.

One thing I have found out about commercial software is that there are often features in the code that appear to be fairly useless. I have looked at some features of these products with the thought: “why would anyone want to do that”. Then some months later while trying to understand something new or a new problem, I discover that the “useless feature” is exactly what I need. What I think this means is that the people writing the codes have thought about the problem a lot and they have had the advantage of extensive feedback and suggestions from a number of users. The reader should bear this in mind when browsing through this document. EnSight has a lot of features I am not sure how to utilize.

The “language” of EnSight is tied to engineering. EnSight talks about “parts”. For them the mesh I read in was a part; the geometrical definition (in terms of Young’s polygons) of a material surface is also a part. In addition objects created from these are also parts. An isosurface or cut plane created from the mesh is also a part. All these parts are pretty much on equal footing: what you can do to one you can do to another.

In addition to parts, EnSight also talks about frames. Frames are just coordinate frames or axis. Frames are associated with parts. You can have a separate frame for each part. That way you can rotate and translate each part separately. For example, by associating different frames with a cut plane and an isosurface, they could be moved separately and displayed side by side in your viewing window.

EnSight recently optimized and parallelized their streamline algorithm. The parallelization was exclusively for SGI platforms. This work was funded by the Army Research Laboratory. I sent a mail message to ARL in order to pulse them and determine how the interaction with CEI went. The response was very favorable. I plan to follow up with a phone call in order to get more information.

Background color is user selectable. The background can also be colored in a graded sort of way. For example you could design the background to gradually go from light blue to sunset red just in case you were showing air flow around a jet fighter and wanted something that looked realistic. With the logo feature mentioned above I guess you could read in any bitmap you wanted and use that for the background.

4.3 EnSight

EnSight is a product of CEI located in Research Triangle Park, North Carolina. The product was first released in 1987 when CEI was a part Cray Research. The company itself began operations in March 1994 as a spin off from Cray Research. Cray retained 25% of the stock. The remainder of the company is owned by the employees. I don't know if this will change now that Cray has been absorbed by SGI. I hope they don't destroy EnSight the way they destroyed DV.

There are a few convenience features that would be helpful. The product needs an abort button. This is especially useful if the user gets into some slow process and wants to abort it so he can make some changes. For a long time I thought the product didn't have a "new session" button. Finally I found the equivalent.

I think it's appropriate to finish up with a few completely subjective reactions. As I mentioned above, when I first saw this product, I thought it messy in the way it would pop up windows all over the place. I had been used to DV and liked the larger viewing window and the fact that nothing obscured the main view. There is some logic to EnSight: Parts and Frames are always on the left; variables are always on the right, etc. There is a way within EnSight to rearrange windows more in the of the way DV does it. I tried this out and thought it a reasonable compromise. The window arrangement can be saved and if the file exists EnSight will always start up with this configuration you want. Perhaps this has something to do with old dogs and new tricks.

I found it difficult to produce vugraphs as clean and as crisp as I could with DV. Part of the problem is that the default range of colors that EnSight uses for palettes is not very good. By changing to the same colors as DV uses I got a much better picture. There were still some features that looked different and believe that this is due to the way that the codes threshold the data. The color control with DV is easier and more informative. EnSight has more options but it just doesn't work quite as well.

EnSight has some part operations I really didn't explore since they were not called out in the requirements. The operations are cut, extract and merge. The cut option might be useful. A user could cut his mesh, form a new mesh and only have to deal with new part. Similarly parts could be merged to form a new part. This is reminiscent of the grouping capability in DV. What is nice in DV is that you could group say a cut plane plane and a probe together and then work with them as single entity. EnSight only only allows merges on parts and the cursor is not a part.

On the down side I feel that when compared to DV, EnSight is difficult to learn and somewhat clumsy. It does, however, have an awful lot of nice features. There are a lot of things that one can do to produce fancier graphics that can be achieved with DV. We have used DV to make animation sequences and then used an Alias/Wavefront's Composer product to put them on videotape. This software can be used (and is used) to product professional quality videos. EnSight is nowhere as fancy as Composer but it still has some nice features for producing videos directly which would be better than just copying DV files to tape.

Evaluator: Verlan Gabrielson

Evaluation Hardware Environment:

Hardware environment: Server: SGI-Challenger — R8000, 4-Processor, MOS-6.1. Client — SGI, R3000, MOS 5.3.

Support (Rating: 2) My experience has been excellent. Inquiries result in communication with code developers or experienced analysts. Reported problems have been handled in a timely manner.

Documentation (Rating: 2) Manual and Learning Guide - good index. User document available using online help option. Numerous examples of data sets and command files.

4.3 EnSight

Turnkey (Rating: 2) This is a licensed code. All functions exist as modules available to user. Any function not available would require negotiation with vender.

Simple (Rating: 0) Learning curve dependent on visualization experience of user. Inexperienced users would need some hands on training. Quite dependent on clients window manager since code can spawn a large number of windows for complex applications.

Variable Types (Rating: 1) All scaler and vector functions defined on unstructured grids. No cell data functions available. Particle data sets can be processed. Structured data sets are processed as unstructured data.

Axis (Rating: 2) Axis glyphs for local and global axis can be made visible. Scaling, sizing, and labeling can be modified by user if new axis are created.

Query (Rating: 2) Any item in the data base can be queried using cursor in viewing window. Time history or functional data can be obtained for any node or element and along an arbitrary line segment. 2-D plots can be initiated of functions on a line segment and time values.

Scale (Rating: 0) Axis annotation available. Bounding box can be made visible and used to optimize interactive 3-D rotations. An auxiliary axis can be created for defining field of view.

Annotation (Rating: 1.5) Annotation of nodes, elements, and parts is available. Text can be assigned easily to any location, size, and color. Dynamic labels such as time can be assigned as processed. Text labels cannot be assigned to data. Function legends can be designed and located in the viewport by the user. There are only a limited number of fonts.

Algebra (Rating: 2) Algebra can be performed with any function. A large number of predefined functions available. A calculator is included. This is a very powerful attribute, easy to use, applicable to all input functions, and for time dependent functions, applies to entire time domain.

Viewing (Rating: 2) Rotation, translation, scaling, for any axis and global and rubber-band zooming available using mouse or specified inputs. Above options can be applied independently to data objects and viewports. Orthographic viewing, fixed and moving single light source plus diffusion, texture, and intensity attributes for surface properties are available.

Cut planes (Rating: 2) Multiple cut planes can be built with surfaces rendered for any defined function. Color maps can be adjusted and contours can be rendered for any function. Cuts can also be made with cylinders, spheres and arbitrary quadric surfaces. Legends can be assigned.

Isosurfaces (Rating: 2) Complete isosurface capability exists for all scaler and vector functions. Isosurfaces can be applied to specified objects(parts). Function editor permits color assignments, ranges, and visibility. Line contours have the same properties.

Isovolume (Rating: -2) No tools available for isovolumes.

Geometries (Rating: 2) All geometries are treated as unstructured elements with part specification. Multiple geometries can be processed for a given visualization. For example, a CTH structured data set can be processed with a finite element unstructured data set. The input module reads a sizable number of user data bases. The geometries can be copied, moved, reflected, as independent objects.

4.3 EnSight

Geometry Deformations (Rating: 2) A deformation vector can be applied to any object as a function of initial node location. This vector can be arbitrarily scaled and all tools use the current deformed location as geometry reference.

Animation (time) (Rating: 2) Flipbook and Keyframe animation available. Flipbook animation is executed on line in which any viewing attribute can be modified during animation. Flipbook animation success dependent on memory size and speed of client. Animation of eigenvalue mode shapes can be done if eigenvector function is available. Numerous options available for animation of streamlines and pathlines which can be synced with time dependent functions. Keyframe animations stores frames in several formats. Time sequences very easy to activate and can be applied to isosurfaces, cutplanes, and isosurfaces. User has control over range and number of frames.

Animation (space) (Rating: 2) Spatial animations use Keyframe option with a large set of options. Range and number of frames selectable. Animation of isovolumes and cutplanes can be processed. Keyframe animation can sync geometric and time features.

Multiple problems (Rating: 1) Multiple problems can be processed using the case option. Up to four cases can be displayed and independently processed on given viewports. Viewing and function selection are permitted for each case.

Thresholds (Rating: 1) This function is available by adjusting the legend function. Values outside the function range can be made invisible. Is node based, not element based.

Erasure (Rating: 1) Visibility of any object can be turned on and off. The global Z-clipping plane can be used to visualize interiors of models. Arbitrary model clipping would have to be done using clipping function attributes.

Streamlines (Rating: 2) Available for structured and unstructured vector data bases with excellent animation features. Streamlines can be colored as function of speed or arbitrary scalar function. Numerous options available for spawning lines. Advection options available for following particle paths.

Pathlines (Rating: 2) Pathlines of vectors in time dependent data bases are computed very efficiently. Can be processed with the same features as streamlines. Provides feature of spawning pathlines as function of given time increment. Excellent animations features for on line tracers. Animated traces can be synced to transient function animations.

Point Data (Rating: 0) Test or point data, termed measured data in EnSight can be processed and visualized with options. Have no data to exercise this option.

Line Data (Rating: 1) Functions along arbitrary selected lines can be output to x-y plots and stored as specified files. Plots of time histories of any variable can be output to files. A 2D plotter module can be used interactively with the above two options.

Symmetry (Rating: 2) Objects can be mirrored to any quadrant at display time, thus not requiring additional memory. This attribute can be turned on or off during flipbook animation. Features are also available for creating symmetry objects.

Input (Rating: 1) EnSight's native data base is unstructured and point data sets. The ascii version of this data base is quite simple with good flexibility for defining objects and multiple type elements. A large number of vander data bases such as Plot3d, Abacus, Movie-BYU, Patran, etc. Can be accessed by the code. The code reads structured data bases but treats them as unstructured during

4.3 EnSight

processing. Unstructured data types of bars, triangles, quadrangles, tetrahedrons, and hexahedrons with or without midside nodes are permitted. Support for the Ascii Viz input format would have to be negotiated with vender.

Output (Rating: 2) The main viewing window can be output as RGB, Pict, and Postscript interactively. Animation frames are stored as specified files with various output options. No mpeg available and a limited set of output formats available.

Memory (Rating: 2) All interactive tasks are recorded in an ascii journal file which can used or edited for following processing. Viewing settings and color table settings can be output to specified files for latter processing.

Large Data (Rating: 2) Have had no problems with problems < 100000 elements. Max Recommended CM size for client not less than 64 Meg. 32 Meg is minimum. EnSight is currently single threaded. Next years version will be designed for MPP machines. User can be selective for some types of data. Has lots of flexibility to view portions of the data set. Quality of dynamic memory memory management unknown. Vender implied million cell data sets are routinely processed with success.

Remote (Rating: 2) A primary feature: Data sets reside on server(licensed machine) and compute bound tasks such as streamlines computed on server. All graphics done on client with clients graphics software and hardware. All X-clients use Clients X11 software.

MP (Rating: -2) The code currently is not programmed for MPP processing.

Extra Features and Comments: This code is designed for unstructured grid applications created by Finite Element codes. With experience an analyst could use it at his desktop for most of his post-processing and animation needs. In addition it has excellent capabilities for CFD applications, specifically its steamline and pathline features. Its ability to identify geometry features as objects permits independent processing of the various tools, applying viewing attributes, and axis transformation to given objects. The ability to define contours, function rendering, cut planes, isosurfaes, provides the analyst an excellent tool to visualize many features of the data.

Its animation features are very robust, and have had success creating mpeg and quicktime animations from stored keyframes.

If user has access to machines with large memories (.25 to 1 gig) the Flipbook animation provides features which can emulate VR features. As in many VR applications the load time of the animation may be long but when loaded, all viewing attributes can be modified interactively, a very powererful feature.

A weak feature of the system is an inadequate default color table, multiple light sources, and surface attributes, I did not see any significant improvement of color quality in invoking RGB color option over the default pseudo color table. Improving the picture quality may be feasible but default options should be better, esp on SGI.

Its strongest features are in my estimation;

- Unstructured grid applications.
- All tools can operate on independent objects.
- Time dependent data bases.
- Ability to process pathlines

4.4 IBM Visualization Data Explorer (DX)

- Flipbook animation.
- Its server-client interaction.
- Useful journal, transformation, color table, etc. files created during processing.
- The ease to obtain symmetry objects and the ability to turn on or off visibility of each object.

4.4 IBM Visualization Data Explorer (DX)

Code Description: “A general-purpose software package for scientific data visualization and analysis. It employs a data-flow-driven client-server execution model.”

“DX supports a number of realization techniques for generating renderable geometry from data. These include color and opacity mapping (e.g., for surface and volume rendering), contours and isosurfaces, histograms, two-dimensional and three-dimensional plotting, surface deformation, etc. for scalar data. For vector data, arrow plots, streamlines, streaklines, etc. are provided. Realizations may be annotated with ribbons, tubes, axes, glyphs, text and display of data locations, meshes and boundaries. Data probing, picking, arbitrary surface and volume sampling, and arbitrary cutting/mapping planes are supported.”

“DX supports a number of non-graphical functions such as point-wise mathematical expressions (e.g., arithmetic, transcendental, boolean, type conversion, etc.), univariate statistics and image processing (e.g., transformation, filter, warp, edge detection, convolution, equalization, blending, morphological operations, etc.). Field/vector operations such as divergence, gradient and curl, dot and cross products, etc. are provided. Non-gridded or scattered data may be interpolated to an arbitrary grid or triangulated, depending on the analysis requirements. The length, area or volume of various geometries may also be computed. Tools for data manipulation such as removal of data points, subsetting by position, sub/supersampling, grid construction, mapping, interpolation, regridding, transposition, etc. are available.”

Hardware Platforms:

- IBM RISC System/6000 uni- and multi-processor workstations running AIX 3.2.5 and 4.1
- Sun Sparcstation 2 uni- and multi-processor workstations running SunOS 4.1.3 (and compatible) or Solaris 2.3 (and compatible)
- HP 700 Series workstations running HP-UX 9.05 (and compatible)
- SGI uni- and multi-processor workstations running IRIX 5.2 (and compatible)
- Data General AViiON workstations running DG/UX 5.4.2 (and compatible)
- DEC Alpha workstations running OSF/1 2 (and compatible)
- The IBM Power Parallel Systems.

Licenses: Commercial product, license available from IBM. Floating License (\$7400 in the US, other locales differ) One user per license may run DX on any supported workstation. Node locked License (\$5900 in the US, other locales differ) Any number of users can use DX on a single licensed machine. All licenses include “unlimited” technical support

Contacts: Dave Modl, digem@deictic.lanl.gov

4.4 IBM Visualization Data Explorer (DX)

E-Contact: <http://www-1.almaden.ibm.com/dx/>

Disclaimer: The recommendations in this document are made with specific consideration of ASCI requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Dave Modl, LANL.

Evaluation Hardware Environment: I used a trial license from IBM for the latest version of DX (3.1.2). I installed this software on my desk-top workstation, an SGI Indigo 2 High Impact with the following attributes (excerpts from `hinv`):

- 1 250 MHZ IP22 Processor
- FPU: MIPS R4010 Floating Point Chip Revision: 0.0
- CPU: MIPS R4400 Processor Chip Revision: 6.0
- Data cache size: 16 Kbytes
- Instruction cache size: 16 Kbytes
- Secondary unified instruction/data cache size: 2 Mbytes
- Main memory size: 128 Mbytes
- Graphics board: High Impact
- Running Irix 5.3.

Support (Rating: 2) There are hot line help facilities, e-mail help facilities, and Web info/help available. Eric Nakata of IBM DX Development team has been very helpful to me in getting information about DX. This all provided even though we have not yet purchased the software, but are working with a fully-functioning evaluation license copy of the software.

Documentation (Rating: 2) It has context sensitive Help built into the Visual Programmers Interface. A printed manual with extensive comments is available. Included in the system is a sample directory that includes sample data sets to be used with sample visual 'program' network files. These sample programs demonstrate the visualization techniques and usability examples needed to start to build your own visual program networks.

Turnkey (Rating: -1) The system is not turnkey by nature, but provides a visual programmers interface that allows users to build and save programs which can be turned over to other 'turnkey-users' for direct use.

Simple (Rating: 1) Computer users (in general) can learn to use the software to build pedagogical programs quickly. It will take someone very familiar with the vast array of tools and methods available to build solutions that we are interested in using for ASCI. The hope is that a new interface can be developed with the help of an IBM DX consultant to better meet the needs of the scientists. The underlying system has many of the processing features we desire to begin with (see the Visualization Features discussed below).

Variable Types (Rating: 2) Data types:

- Scalar – Yes, well supported and examples are provided;

4.4 IBM Visualization Data Explorer (DX)

- Vector – Yes, well supported and examples are provided;
- Cell – Tools are available to create and display cell data from arbitrary data sets;
- Node – Similarly, Node data is supported in the tools;
- Null – It is supported as ‘invalid’ data attributes where data elements or complete segments can be marked as invalid and are not used in the rendering or any other manipulation of the data set.

Axis (Rating: 2) DX supports this function fully.

Query (Rating: 2) Using the Pick tool, one can obtain data associated with points in space of the image currently being viewed.

Scale (Rating: 2) A bounding box with dimensions displayed is available as the ShowBox tool found in the Realization Category.

Annotation (Rating: 2) Full featured in this respect, the system allows for values, filenames and attributes to be displayed on the image window. A ColorBar tool allows for the display of a legend to which is mapped the extremes of the data set(s).

Algebra (Rating: 2) A full set of tools is available for this purpose which can be applied to individual segments of a data set or the complete set.

Viewing (Rating: 2) The Image tool has many controls setup to help the user position the view angle arbitrarily. Light sources can be added as needed, and there is a mechanism available that allows the user to switch the view from perspective to orthographic. Multiple simultaneous views could be displayed with another instance of the Image tool for the alternate viewing needed.

Cut planes (Rating: 2) There are at least two forms of this function called ‘Slice’ and ‘Slab’.

Isosurfaces (Rating: 2) A very straight forward Tool is available for rendering isosurfaces of volume data with all the requisite features.

Isovolume (Rating: 2) DX meets this function fully. It does this in an interesting manner. A group of modules (in the visual programming parlance) are assembled to compute the isosurfaces of the extreme values selected, then ‘caps’ (computed by a slicing plane) are mapped into the spaces that would be holes in the new data set. This saves from having to compute the integral isosurfaces between the min and max values selected. It also saves in storage space, but other tools that might want to use the results of this computation must know that the full volume of the space is not stored directly for the result of the isovolume — this is completely ‘tunable’ in DX — we can implement the processing method to solve the more data rich result or do the ‘capping’ method, based on the users’ needs.

Geometries (Rating: 2) Various forms of data can be viewed with the appropriate data import tool for the form the geometry data takes. Geometry data in DX is no different than any other data that may be manipulated once it is incorporated in the system.

Geometry Deformations (Rating: 2) Deformation information can be added as a dataset to the system and can be manipulated individually like any other data set. However, I don’t know if it is intended that the deformation data is a separate dataset or not as constituted by this requirement.

4.4 IBM Visualization Data Explorer (DX)

Animation (time) (Rating: 2) Frames of data can be rendered and presented as a time sequence which should be able to fit the required functionality of this item.

Animation (space) (Rating: 2) DX has the ability to meet these requirements for which the product already supports the feature (i.e. isovolume is not directly supported so DX can't animate an isovolume spatial animation sequence directly).

Multiple problems (Rating: 2) There is no limit to the number of data sets and processing on these data sets - this seems to only be limited by CPU 'horsepower.'

Thresholds (Rating: 2) You can specify mins and maxs of a data set at the time the data is retrieved from the file. You can specify frame ranges of multipart data also.

Erasure (Rating: 2) This functionality is completely programmable in the visual programming language to support an end-user application.

Streamlines (Rating: 2) DX fully supports the streamlines functions required.

Pathlines (Rating: 2) DX calls this a 'streakline' and fully supports this function.

Point Data (Rating: 2) DX supports the rendering of glyphs and cells in the manner needed to meet the requirements of this function.

Line Data (Rating: 2) These functions are supported, including the ability to store results to a file.

Symmetry (Rating: 2) It has all the tools available to meet this functional requirement.

Input (Rating: 2) It can read HDF files of various structures, including ASCII text data files. A import function can be created to read ASCII specific data through the extensible nature of the system.

Output (Rating: 2) It can export images and the data associated with the images to external files. It can be extended to include any format that is desired for the ASCII project.

Memory (Rating: 2) Configuration information is stored for each visual program and can be regenerated quite simply.

Large Data (Rating: 2) This is supported and configurable.

Remote (Rating: 2) DX is capable of computing on one machine and sending a DGL image over to another machine, but not able to perform this function to the letter. Note: it does not support GL-R at this time.

MP (Rating: 1) Eric Nakata has told me that DX will run on the IBM SP2 system. I have not seen this and I don't know how easy it would be to port to another type of architecture.

Extra Features and Comments: Note that this is an initial cut of an evaluation of IBM's DX viz package. It is a naive attempt on my part to ascribe values based on a system of measurement that I was not initially involved with creating, for a software system that I only understand through a pedagogical use via the four day seminar on DX back in June. I have not tried to use DX to solve any real scientific viz problems to date. I have tried to ask the experts at IBM for understanding about things I could not possibly answer on my own.

4.5 General Mesh Viewer (GMV)

The IBM response to the original evaluation was to point out that I may have inappropriately down-graded the capability of DX for the specific features: Axis, Isovolume, Geometries, Erasure, Remote.

IBM Consultants were able to demonstrate how to use DX to meet the particular feature with examples of DX programming which is being used in the on-going development of the prototype visualization tool.

4.5 General Mesh Viewer (GMV)

Code Description: ???

Hardware Platforms: SGIs — support for other platforms unknown.

Licenses: Developed by LANL, believed to be generally available for distribution.

Contact: Frank Ortega, LANL, 505-665-0688, Mail Stop F645, fao@lanl.gov

E-Contact: ???

Disclaimer: The recommendations in this document are made with specific consideration of ASCI requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: John Fowler, LANL.

Evaluation Hardware Environment: Not specified.

Support (Rating: 1) GMV is supported by the programmer who works on it. User requests and bug fixes are added subject to programmer time limits. Recently a second programmer has been added to the team.

Documentation (Rating: 2) There is a hard copy (PostScript) and also an on-line (WWW) manual. This manual is kept up-to-date and describes all features of the code.

Turnkey (Rating: 2) This is a totally turnkey application.

Simple (Rating: 2) The code works on pulldown menus and fill-in Motif-based forms for supplying data. Although you do need to know what pulldown to click on in order to get the isosurfaces form to come up, for instance, this can be quickly learned.

Variable Types (Rating: 2) MV can visualize scalar and vector, cell-based and/or node-based, data as well as independent sets of polygons and flag data for each cell or node.

Axis (Rating: 1) Axes can be turned on or off and are labeled and color-coded, but there is no axis scale. Axes are situated at the origin and cannot be moved. There is a small, icon-sized drawing area in the corner of the main window that also shows axis orientation. This is useful for observing the axis orientation when the axes are hidden by the graphics in the main window.

4.5 General Mesh Viewer (GMV)

Query (Rating: 2) The Query Data button brings up a panel that permits selection of a cell or node by clicking on the viewing area. The location of mesh vertices, node numbers of the cell, and/or field variable values are then displayed. One can also select a field value and get number of the node where the field most closely matches that value.

Scale (Rating: 1) The user can get the distance between any two selected points in user units, in the projection plane. Thus it is possible to measure the length of an object by rotating it so that it is parallel to the projection plane, and clicking the mouse on each end of the projection. The result is printed on the screen in units corresponding to the data's coordinate system.

Annotation (Rating: 0) Problem time and cycle number can be added to the viewing window. These data must exist in the input file. In addition, one can enter a one-line title at any time. This will be displayed at the top center of the viewing area.

Algebra (Rating: 2) Pairs of node or cell based fields can be combined to create a new field, using any of 18 algebraic operations. The newly created field can then similarly be combined with others.

Viewing (Rating: 1) Translation, rotation, and zooming can be set and controlled several ways: By pressing the appropriate mouse button and moving the mouse, by manipulating on-screen sliders, or by selecting forms with the appropriate sliders to come up.

Cut planes (Rating: 2) All of the listed criteria for cut planes are provided for except one. Only one cut plane can be created and viewed at a time. In addition to these criteria, one can show cell, node, and vector information within a specified distance perpendicular to the cut plane. Also, the cut plane can be drawn in a manner where its height normal to the plane is proportional to value of the field being rendered.

Isosurfaces (Rating: 1) One can create up to 20 concurrent isosurfaces of any scalar field in the data set. These can be clipped on a subset box or can be shown only for a selected subset of cells. Isosurfaces can also be created on material number with a user-adjustable material fraction. It is not possible, however, to arbitrarily color an isosurface by another variable.

Isovolume (Rating: -2) This feature is not available.

Geometries (Rating: 2) Any polygonal data can be included in a GMV input file and viewed as any number of polygonal regions. Typically these correspond to material interfaces. They cannot be copied and moved independently, but may be reflected. Also, they can be colored and rendered semitransparent by the built-in color editor. And they can be clipped by a user-selected plot box or polygon subset typein. Also there is an "explode" feature that allows one to separate different polygon regions spatially.

Geometry Deformations (Rating: ?) This feature was not rated.

Animation (time) (Rating: 0) This is not supported in GMV, but there is a postprocessor that can make time animations from sequences of GMV dumps. The output is a set of image files that can be viewed with software that is provided.

Animation (space) (Rating: 2) There is an extensive facility for spatial animation sequences that can be viewed directly or written out as a sequence of image files.

Multiple problems (Rating: -2) GMV supports only one input file at a time.

4.5 General Mesh Viewer (GMV)

Thresholds (Rating: 0) Thresholding is not directly supported, but isosurfaces, cutplanes, and polygon data can be clipped by adjusting subsetting or clipping planes.

Erasure (Rating: 0) There is partial erasure support through subsetting, which is well implemented, and through the cell and node data widgets.

Streamlines (Rating: -2) There is no support for streamlines.

Pathlines (Rating: -2) There is no support for time dependent streamlines. Vector data can be displayed as arrows with user-controlled scaling.

Point Data (Rating: 2) GMV provides for visualization of point data. Tracer particles are also supported. These are points that are not necessarily related to node or element locations.

Line Data (Rating: 1) All of the listed features are provided for except for time histories.

Symmetry (Rating: 2) One can create reflections about the X, Y, or Z axes or any combination of them. The implementation is efficient.

Input (Rating: 2) GMV accepts a wide variety of structured and unstructured grids, point sets, field data, and polygons. It can read either a binary or an ASCII file. Tools exist for creating these files and for converting from binary to ASCII.

Output (Rating: 0) GMV will output individual or sets of image frames in a standard format. Object data itself cannot be output. GMV could be upgraded to eventually support the "ASCII Visualization" format.

Memory (Rating: 2) GMV can write an attributes file that can be used to subsequently bring the user back to that point.

Large Data (Rating: -1) Not well supported.

Remote (Rating: -2) Not supported, except for what is provided by the X Window System.

MP (Rating: ?) This feature was not rated.

Extra Features and Comments: None.

Evaluator: Bob Kares. (Please note that this review, by Bob Kares, is a *companion* review, written in view of the original review, by John Fowler. Accordingly, there are comments for each item here only when Bob wishes to suggest additions or changes to John's original notes.)

Evaluation Hardware Environment: Not specified.

Support (Rating: 0) GMV is supported by a single programmer. There is no formal bug tracking system or hotline support. Until very recently this has been a single person effort.

Documentation (Rating: 1) There is a very useful User Manual which reviews in detail the various menus and features. There is also an html version of this document. However, there is no tutorial, no examples and no context sensitive on-line help as EnSight provides so this doesn't quite rate a 2.

Turnkey (Rating: 2) Ok.

Simple (Rating: 2) Ok.

4.5 General Mesh Viewer (GMV)

Variable Types (Rating: 2) Ok.

Axis (Rating: 1) Ok.

Query (Rating: 2) Ok.

Scale (Rating: 1) Ok.

Annotation (Rating: 0) Ok.

Algebra (Rating: 1) Does not support differential operations like div or curl nor does it support surface or volume intergrals.

Viewing (Rating: 1) Ok.

Cut planes (Rating: 1) Like John Fowler states, only one cut-plane can be viewed at a time which rates a bit of a score reduction.

Isosurfaces (Rating: 1) Ok.

Isovolume (Rating: -2) Ok.

Geometries (Rating: 2) Ok.

Geometry Deformations (Rating: ?) This feature was not rated.

Animation (time) (Rating: -2) Time animation is not supported interactively in GMV although GMV can be run in a batch mode to render a sequence of images in time. This is not what I mean by time animation.

Animation (space) (Rating: 1) GMV can animate the spacial motion of cutplanes, isosurfaces and vector flows combined with viewing transformations and output the result as a sequence of image files. It can also animate flythrough of the data by picking key points along the path of the virtual helicopter, a simple type of keyframe animation. However, even though this capability is nicely implemented in GMV, it is still far less powerful than that provided by EnSight so I feel a score reduction is in order here.

Multiple problems (Rating: -2) Ok.

Thresholds (Rating: 0) Ok.

Erasure (Rating: -1) There are no simple erasure tools like an erasure plane as in Data Visualizer. This gets a score reduction to -1.

Streamlines (Rating: -2) Ok.

Pathlines (Rating: -2) Ok.

Point Data (Rating: 1) GMV can visualize point data but there is no point cloud sampling of a 3D data volume.

Line Data (Rating: 1) Ok.

Symmetry (Rating: 1) Only simple reflections are implemented.

4.6 Khoros

Input (Rating: 1) GMV implements a very general polygonal cell element type but allows input only in its own proprietary file format.

Output (Rating: 0) Ok.

Memory (Rating: 0) GMV can put out an attributes file which remembers tool settings like view perspective and color map but this is all it can do. There is no journalling, no command language and no session or archive files as both EnSight and Data Visualizer provide.

Large Data (Rating: -1) Ok.

Remote (Rating: 0) GMV can support a remote OpenGL connection but has no client-server model. For consistency with MUSTAFA I gave it a 0.

MP (Rating: -2) John didn't rate this feature but GMV does not use multiple CPUs in a box.

Extra Features and Comments: None.

4.6 Khoros

Code Description: "Khoros is a software integration and development environment which puts an amazingly broad range of tools at your fingertips. Khoros includes a visual programming language, a suite of software development tools that extend the visual language and help you create new applications, an interactive user interface editor, an interactive image display package, 2D/3D plotting, and an extensive suite of image processing, data manipulation, scientific visualization, geometry and matrix operators."

Hardware Platforms: SPARCStations (SunOS and Solaris), Dec Alpha, SGI Irix, RS6000 AIX, Data General DG/UX, Cray EL92 UNICOS, 486 (Linux and BSDI), HP PA-RISC HP-UX, Sequent SysVR, Paragon OSF.

Licenses: The Free Access License allows you to use Khoros and distribute your Khoros-based applications at no cost, and comes into effect upon acquisition via ftp or CD-ROM. The System Integrator License allows you to distribute Khoros bundled with your own software or service, costs \$5K/year per institution. The Developer License gives you the power to distribute Khoros derivative works, and costs \$50K/year per institution.

Contacts: Philip Kegelmeyer, (510) 294-3016, wpk@ananda.ran.sandia.gov. User and developer.

E-Contact: <http://www.khoros.unm.edu/>

Disclaimer: The recommendations in this document are made with specific consideration of ASCII requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Verlan Gabrielson, SNL, gabe@ca.sandia.gov

Evaluation Hardware Environment: SGI Indigo R4000, 64 Mbyte RAM, MOS 5.3.

4.6 Khoros

Support (Rating: 1) Khoros Research Ins (KRI) does not provide support to users who have accessed the code as open software. It will provide support to customers who have the Developer's licence.

Documentation (Rating: 0) Excellent documentation exists for image processing applications and software developers. I felt it was very inadequate for new users and geometry applications.

Turnkey (Rating: 1) The user has access to any of the open software available with the code. Numerous tools would have to be developed for ASCII applications. The system permits user written tools.

Simple (Rating: -1) It cannot be considered simple since many tools would have to be developed.

Variable Types (Rating: -1) Data functions for structured grids are node based. Data functions for unstructured grids (unknown)

Axis (Rating: 1) Axis features are available.

Query (Rating: -2) Query functions would have to be built.

Scale (Rating: 1) Object viewing uses bounding box — measure function unknown.

Annotation (Rating: 1) Text annotation available (robustness-unknown).

Algebra (Rating: 2) Probably the strongest feature of the code. User has access to extensive algebra and mathematical tools.

Viewing (Rating: 2) The RenderMonster tool provides extensive viewing options.

Cut planes (Rating: 1) Available for structured data sets. Unknown for unstructured.

Isosurfaces (Rating: 1) Available for structured data sets. Unknown for unstructured.

Isovolume (Rating: 0) Tool exists, unable to determine its capabilities.

Geometries (Rating: -1) I feel most of the attributes under this heading would require user designed tools.

Geometry Deformations (Rating: 0) Would require user designed tools, May be able to use Khoros Algebra features.

Animation (time) (Rating: 2) Animation features available. Have no experience in designing animation.

Animation (space) (Rating: 2) Animation features available. Have no experience in designing animation.

Multiple problems (Rating: 2)

Cantata provides two unique networks for which user can switch easily. There does not seem to be any restrictions for multiple tasks and viewing windows for a given network.

Thresholds (Rating: 2) Have no knowledge of the robustness of color tables.

Erasure (Rating: 2) Has the required features.

Streamlines (Rating: 1) Tool exists, have no experience with use.

Pathlines (Rating: -2) No indication feature exists.

Point Data (Rating: 1) No experience but Khoros tools should apply to this feature.

Line Data (Rating: 2) Has the required features.

Symmetry (Rating: 1) No tool known which specifically does this, but Khoros algebra and math tools could be used.

Input (Rating: 0) The native Khoros input file is not very practical with ASCII data bases. User would create their own using Khoros tools.

Output (Rating: 1) Have little knowledge of these needs.

Memory (Rating: 2) Code is used with large data sets with special memory allocation. The cantata data flow method may not handle very large data sets.

Large Data (Rating: 2)

Remote (Rating: 1) Khoros seems to work well with the X-window protocols.

MP (Rating: 0) From my knowledge it has not been designed for MPP machines.

Extra Features and Comments: This evaluation reports my knowledge of this system which at present is quite limited. I had no previous experience with the code before July 96. Numerous problems occurred which limited the evaluation. We obtained a new version Khoros2.1 in July and found it would not make on a SGI 6.2 operating system, our system of choice. I do not know at this time if these installation problems have been resolved. We installed the system on an SGI Indigo using MOS 5.3. This machine is quite limited in size and speed making the evaluation tedious. Khoros provides both a application environment and a software development environment for multidimensional data processing, analysis, exploration, and visualization. I attempted to use it as an application tool on ascii type data bases similar to how I evaluated Ensign and other visualization tools. I was not successful with this approach since most application tools available are for image processing and software development tasks. A number of scientific visualization tools are available in the geometry directory but are quite problem specific and do not have the robustness I feel we need for the ascii vis tools. This does not imply that a software team could not design an excellent geometry visualization tools from huge set software development tools available with the system. The second difficulty was data structure. At present I have not been successful in finding sufficient documentation on how I would implement ascii type data structures as data streams for Khoros functions. A somewhat awkward format exists for structured grids, and could not find any support for complete unstructured data base, except a simplistic input for geometry of triangular facets. To use Khoros for our ascii needs would require the development of a robust reader, which should be feasible with available tools. Geometry tools currently available are isosurfaces, cut planes, and streamlines. These tools apply to structured data, but I found no examples if they apply to unstructured data. An interesting isovolume tool exists but I was unable to have it execute properly on our workstation which had limited memory, and disk space. To use the code for ascii problems would require numerous tools to be developed for the unstructured data bases.

I like the visual programming language cantata which works quite well. It is designed much like the AVS dataflow networks. This may prove to be incompatible with large data sets. Having experience with AVS, cantata was quite easy to use. The help widgets available for each tool were quite useful but at times confusing since much information was not defined and I could not locate adequate explanation. It was also somewhat difficult to identify proper data files for input output ports. The

4.7 Mesh-TV — A Graphical Analysis Tool

RenderMonster tool is probably the best feature of the Khoros system. This tool which creates the visual displays from multiple input ports has all the options for creating the desired viewing of your data. Numerous options for orientation, viewpoints, surface attributes, light sources, backgrounds, etc. are available.

4.7 Mesh-TV — A Graphical Analysis Tool

Code Description: This is an interactive graphical analysis tool for querying, visualizing and analyzing data on 2D and 3D meshes. It is a general-purpose tool in that it handles many different mesh types and provides many different ways of viewing the data. For most vendor platforms it provides graphics at the speed of the native graphics hardware. Mesh-TV handles 2D and 3D rectilinear, curvilinear and unstructured mesh data. It also handles multi-block meshes, mixed material zones and mixed species materials.

Hardware Platforms: SGI, HP, IBM, Sun, Cray, Meiko, DEC and Linux

Licenses: None needed

Contacts:

- Linnea Cook 510-422-1686 cook13@llnl.gov
- Eric Brugger 510-423-1293 brugger@llnl.gov

E-Contact:

- anonymous ftp: [coral.llnl.gov:/pub/meshtv/meshtv2_7_2](ftp://coral.llnl.gov/pub/meshtv/meshtv2_7_2)
- http://www.llnl.gov/liv_comp/mesh.html
- The Silo manual (the library for writing files for Mesh-TV) has a URL:
<http://sphere.llnl.gov/silo/silo.html>

Disclaimer: The recommendations in this document are made with specific consideration of ASCII requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Eric Brugger.

Evaluation Hardware Environment: SGI 4D/35 TG with Elan graphics (36 Mhz R3000 w/128 Mbytes of memory).

Support (Rating: 2) MeshTV is fully supported. There are developers actively correcting bugs and adding new features to the code. Major release are every two to three months, with minor releases occurring between those intervals as necessary to correct bugs. Questions can be answered by sending e-mail to or calling one of the developers. Bugs are given a reference number and tracked. The list of current bugs is listed in the on-line release notes as well as a list of the corrected bugs.

Documentation (Rating: 2) There is a printed users manual for MeshTV and the Silo library (The Silo library is used to create data files that can be read by MeshTV). There is on-line documentation available for the Silo library in HTML format. There is currently no on-line documentation for MeshTV, although it will be available shortly in HTML format. The documentation consists of a reference manual for the MeshTV command line interface (CLI) and a users guide for the MeshTV graphical user interface (GUI). The users guide contains a tutorial that leads the user through the basic features of MeshTV. There is a reference manual for the Silo library as well as some programs which generate sample data files readable by MeshTV. The sample programs allow users to modify existing programs to create Silo files instead of writing new programs from scratch. Simple generic data sets are provided with the distribution. They consist of both 2d and 3d data sets for each of the supported mesh types.

Turnkey (Rating: 2) MeshTV is a turnkey application. Since the source code is available it can be modified by the user.

Simple (Rating: 2) MeshTV is simple to use. It has been my experience that new users can learn the basic concepts in a couple of hours. The tool is organized around plots and operators. The user opens a file or time sequence of files. Once a file is open the user selects plots (e.g. mesh, material interface, iso-contour, etc.). The user can then perform operations on the data (e.g. reflection, slicing operations, etc.). Each plot and operator has a properties window associated with it. The user can go to the previous, next or arbitrary time point by clicking a button or moving a slider.

Variable Types (Rating: 2) MeshTV supports both scalars and vectors. It also supports node and element data.

Axis (Rating: 2) MeshTV supports a 3d bounding box with labels and tick marks.

Query (Rating: 2) The user can query any variable in the data set as well as node and element number. Variables are queried by first selecting a slice plane and then querying points on the slice plane. When a point is queried, the x, y and z coordinate of the point selected, the value of a single variable, and the node and element number are output to a separate window. For structured meshes it outputs the i, j, and k index of the element instead of the node and element number. A letter is displayed on the slice of the queried point.

Scale (Rating: 2) MeshTV supports a 3d bounding box with labels and tick marks.

Annotation (Rating: 1) MeshTV provides some annotation information on plots that is present in the data file. Legends are provided for all the plot types (e.g. color bars for pseudo color plots, contour levels for contour plots, etc.). Miscellaneous annotation is also provided by MeshTV such as the name of the data file, the time and cycle associated with an image, etc. The legends and annotation can be turned on or off. It is also possible to add banners to postscript output.

Algebra (Rating: 2) MeshTV provides the basic arithmetic operations ($+$, $-$, $/$, $*$, $^$), as well as absolute value, square root, log, natural log, gradient, divergence, curl, magnitude, dot product and potential. It is possible to perform arithmetic operations between data from different time states as well as different problems so long as the meshes on which the data is defined have the “size”.

Viewing (Rating: 2) Control of the view can be done either with the mouse or by typing in precise viewing manipulations. The left mouse button is used for doing rotations, the center button for zooming in and out, and the shift key in conjunction with the left mouse button for panning. All these operations

4.7 Mesh-TV — A Graphical Analysis Tool

are also available by typing in commands. The user can specify the center of rotation by entering the x, y, and z coordinate of a point. The user can easily return to the default view. The user can set up multiple views in multiple windows. There is no automatic support for setting up the orthographic projections but the user could do it manually. The user can define up to eight different light sources.

Cut planes (Rating: 1) Cutting planes are fully supported except that they can not be animated. The user can specify multiple cutting planes that can be oriented in any arbitrary fashion in the data field. Scalars and vectors can be plotted on this plane. The color map can be adjusted. The cut planes can be pseudo colored, and can have iso lines displayed on them. Legends are provided.

Isosurfaces (Rating: 1) Isosurfaces are fully supported except that vectors can not be displayed on their surfaces and they can not be animated. The user can select an arbitrary number of isosurfaces. Scalars can be plotted on the surface. The surface can be pseudo colored with another variable. Legends are provided.

Isovolume (Rating: -2) There is no support for isovolumes. Since isovolumes are a variation on isosurfaces it would be fairly easy to add this functionality.

Geometries (Rating: 1) Geometries are fully supported except that they can not be copied and moved independently. The geometries can be reflected independently. An arbitrary number of them are allowed.

Geometry Deformations (Rating: -2) Geometry deformations are not supported. Geometry deformation could easily be added as an algebraic manipulation.

Animation (time) (Rating: 2) The user can animate anything that is displayed in time. The image can be panned, zoomed and rotated while the animation is playing. MeshTV also supports the caching of the geometric data. When geometry caching is on, the geometry information for each frame in the animation is stored in memory and results in faster animations. If the user is looking at a very large problem or at many time states and all the geometry would not fit into memory the user can turn off the geometry caching. The animations can be played forwards or backwards at any desired frame rate. The user can also advance or go backward one frame at a time. If the user selects a frame rate that is faster than the computer can maintain, it will drop frames. The user can plot every other time state or only a subset of the frames by manually removing frames from the list of frames. There is no way to automatically select every other frame or every third frame.

Animation (space) (Rating: -1) The only spacial animations that MeshTV supports is rotations. The rotational animation can be combined with time animation. The center of rotation is user selectable.

Multiple problems (Rating: 2) MeshTV can read in as many different calculations on as many different meshes as desired.

Thresholds (Rating: 1) The limits on the color bar can be set by the user. Values below the minimum get mapped to the color at the low end of the color bar and values above the maximum get mapped to the color at the high end of the color bar.

Erasure (Rating: 0) The user can select up to six erasure planes with arbitrary orientation.

Streamlines (Rating: 1) MeshTV supports streamlines, stream ribbons, and stream tubes. The user selects a density of stream lines and MeshTV automatically seeds the volume with streamlines. There is no support for particle animation.

Pathlines (Rating: -2) MeshTV has no support for pathlines. Pathlines are a nice feature for small problems but would be totally useless for large problems since the amount of disk, memory and cpu time required to generate the pathlines would be prohibitive.

Point Data (Rating: 1) MeshTV can read a set of points with associated data values. The points are displayed as cubes. The points can be colored by one of the variables.

Line Data (Rating: 2) The user can select an arbitrary line through space. The user first selects a cutting plane and then selects a line on the cutting plane. Once the line is selected, a curve that consists of a variable as a function of distance along that line is displayed in another output window. An extensive number of manipulations may be done on the curves once they have been created.

Symmetry (Rating: 2) The user can reflect objects about their minimum and maximum extents in the x, y, and z directions of the object. When data is reflected it is done by re-rendering the same graphics primitives with a different view transformation so that they appear reflected. This means that there is no impact on the amount of memory used by MeshTV when reflecting objects.

Input (Rating: 2) MeshTV supports structured, unstructured, rectilinear, curvilinear, multi-block and AMR meshes. For unstructured meshes it supports tetrahedrons, pyramids, prisms and hexahedrons. MeshTV treats structured meshes as structured meshes and unstructured meshes as unstructured meshes. MeshTV supports AMR meshes in the sense that it supports unstructured meshes which the rest of the committee believes can be used to support any type of mesh. MeshTV also supports point data. MeshTV reads its data from Silo files which can be written using the Silo library. Since it is fairly easy to write a simple Silo file, I am not down grading MeshTV for not reading simple ASCII files. ASCII files are fine for small problems but there are numerous performance gains to be realized for large problem sizes. The tool will be upgraded to the “ASCII Visualization” format under construction by the ASCII Visualization Data Formats Group.

Output (Rating: 2) MeshTV supports Postscript, XWD, SGI’s RGB format, and TIFF files.

Memory (Rating: 1) The default settings of all the controls and the placement of all the windows can be saved in a configuration file that is read on start up by MeshTV. There is no way to return the MeshTV graphical user interface to a particular state. It is possible to return the command line driven portion of MeshTV to a particular state by having it read the log that is automatically generated as it is run.

Large Data (Rating: 2) MeshTV handles large data sets well. It does not do caching of variables. It reads in the variables it needs to generate a plot and then frees the memory. The only thing that MeshTV caches in memory is the geometric information needed to render the output images. This does not result in a noticeable performance penalty since MeshTV reads binary data files which can be read very quickly. The following memory usage information was obtained by using the sample data files.

TABLE 1. 3d rectilinear mesh memory usage with various plots

Size (number of zones)	No plots	Mesh	Material Interfaces (10 materials)	Iso-surface (10 levels)
36 K	6.4 Mbytes	7.5 Mbytes	11 Mbytes	9.3 Mbytes
288 K		7.5 Mbytes	25 Mbytes	16 Mbytes
2.3 M		7.5 Mbytes	88 Mbytes	45 Mbytes

4.7 Mesh-TV — A Graphical Analysis Tool

TABLE 2. 3d curvilinear mesh memory usage with various plots

Size (number of zones)	No plots	Mesh	Material Interfaces (3 materials)	Iso-surface (10 levels)
36 K	6.4 Mbytes	8.2 Mbytes	9.7 Mbytes	8.8 Mbytes
288 K		12 Mbytes	19 Mbytes	15 Mbytes
1.7 M			60 Mbytes	47 Mbytes

TABLE 3. 3d unstructured mesh memory usage with various plots

Size (number of zones)	No plots	Mesh	Material Interfaces (3 materials)	Iso-surface (10 levels)
4 K	6.4 Mbytes	8.2 Mbytes	8.6 Mbytes	8.2 Mbytes
32 K		11 Mbytes	19 Mbytes	11 Mbytes
256 K		28 Mbytes	34 Mbytes	32 Mbytes

MeshTV does not exploit multi-processor platforms.

Remote (Rating: 0) MeshTV can do number crunching on one machine and rendering on another if the display device is GL or OpenGL capable. MeshTV is fully X compatible and can display to a generic X Terminal with no loss of functionality.

MP (Rating: 0) MeshTV does not take advantage of SMPs. MeshTV has been written to run in parallel using MPI. The data needs to be written out in a domain decomposed fashion. When running in parallel, MeshTV is started on n processors with one of the processes controlling the output display, and the rest acting as slaves. Each processor reads in some number of domains and generates an image for those domains. Then the images are composited using the frame buffer in conjunction with the z buffer. The final composited image is then displayed in the output window. The parallel version is still in the experimental stage and is not ready for “production” use.

Extra Features and Comments: For the purpose of this evaluation, MeshTV was treated as a turnkey application. Had it been treated as an application development tool some of the scores would have gone up since some of the features could have been easily added since the underlying functionality needed to implement them were already present.

MeshTV has extensive 2d plotting capabilities. It treats 2d data as 2d data rather than 3d data with a constant z value. This results in much nicer plots with nice 2d coordinate axes. It also allows the panning and zooming to be tailored to 2d images and the automatic update of axes as an image is panned or zoomed. It also allows zooming by rubber banding a region with the mouse.

MeshTV has extensive 1d plotting capabilities. The user can perform numerous operations on the curves including the standard arithmetic operations, all the trigonometric functions, logarithms, Bessel functions, smoothing, averaging, fast Fourier transforms, derivatives and integrals. The 1d plotting capabilities are all driven by the command line interface.

MeshTV can be run from the GUI or with a CLI. The CLI is always available when using the GUI.

MeshTV can be run in a batch mode with no display device connected to it. The end result is a sequence of images stored in Postscript files.

4.7 Mesh-TV — A Graphical Analysis Tool

When the user saves or prints an image, s/he can either select to capture the image displayed on the screen or render a 24 bit image at any resolution up to 2048 by 2048. This is a very useful feature when your display device is an 8 bit X terminal.

Evaluator: Bob Kares. (Please note that this review, by Bob Kares, is a *companion* review, written in view of the original review, by Eric Brugger. Accordingly, there are comments for each item here only when Bob wishes to suggestion has additions or changes to Eric's original notes.)

Evaluation Hardware Environment: Unknown

Support (Rating: 1) Some developer support and a formal bug tracking system but still not supported at the level of a commercial product. No hotline, no dedicated support personnel, etc.

Documentation (Rating: 1) User Manual is useful and some example datasets are provided as SILO files but documentation is not anything like as extensive as EnSight. The only type of on-line documentation available are brief notes on enhancements and bug fixes for recent releases. There is no on-line access to manuals and context sensitive help as EnSight provides.

Turnkey (Rating: 2) Ok.

Simple (Rating: 2) Ok.

Variable Types (Rating: 2) I'm ok with a rating of 2 here but it should be noted that in comparison EnSight supports a larger class of elements like the 20 node hexa in 3D.

Axis (Rating: -2) MeshTV does NOT provide a 3D triad of axes as required. Eric claims that a 3D bounding box with labels and ticks is supported but I cannot find it. Even if this is the case, it is not really what I want here, nor does it sound like these could be moved or sized as required.

Query (Rating: 0) I am ambivalent about whether or not this feature is really supported. Query in my mind refers to a real 3D probe which both Data Visualizer and EnSight have. MeshTV has a probe but it works only on a 2D planar representation of the data. To change the 3D location of the probe you have to move the slice plane operator and then probe on the newly selected slice, a procedure which is awkward in comparison to a true 3D data probe. This is one of many examples which suggest to me that MeshTV is pitched more toward 2D rather than 3D visualization.

Scale (Rating: 0) Again I am ambivalent about whether or not MeshTV really possesses this feature. MeshTV has a distance plot mode which produces value vs distance plots on a line drawn across a 2D planar representation of the data. To obtain data at a different orientation again requires moving the slice plane operator. This is really awkward and does not compare well with the operation of the ruler tool in Data Visualizer.

Annotation (Rating: -1) The annotation features of MeshTV are rather limited compared to EnSight. Labels appear in fixed positions and are in fonts that are virtually unreadable. Color bars appear in fixed positions and cannot be moved.

Algebra (Rating: 2) Ok.

Viewing (Rating: 1) View control is more convenient in EnSight although the functionality is comparable. There is no orthographic projection toggle which is inconvenient for precise positioning of tools.

Cut planes (Rating: 0) Again I am ambivalent as to whether or not MeshTV has a cut-plane in the sense in which I mean the term. In MeshTV the user applies a slice plane operator to a pseudo-color representation of the 3D data to obtain a 2D planar pseudo-color slice. Eric calls this a cut-plane. I disagree. By cut-plane I mean a 3D planar slice through the data volume whose orientation is readily apparent and which can be animated to view the 3D volume as a sequence of slices. That's not what you get in MeshTV.

Isosurfaces (Rating: 1) Ok.

Isovolume (Rating: -2) Ok.

Geometries (Rating: 1) Ok.

Geometry Deformations (Rating: -2) Ok

Animation (time) (Rating: 1) MeshTV's time animation is comparable to MUSTAFA's which is considerably short of EnSight's.

Animation (space) (Rating: -1) Ok.

Multiple problems (Rating: 2) Ok.

Thresholds (Rating: 1) Ok.

Erasure (Rating: -2) Eric says that the user can select up to 6 erasure planes. I just can't find this capability anywhere. Perhaps we each mean something different by this feature. Certainly Data Visualizer, EnSight and MUSTAFA all have what I mean by an erasure plane. I can't find the corresponding capability in MeshTV which says something about the documentation.

Streamlines (Rating: 1) Ok.

Pathlines (Rating: -2) Ok.

Point Data (Rating: 1) You can display point data but you cannot do a point cloud sampling of a 3D data volume. That should be mentioned in the writeup but was not.

Line Data (Rating: 1) Again you can plot line data, but only on a 2D planar representation obtained by applying a slice plane operator to the 3D data volume. What I mean by this feature is something like the ruler tool in Data Visualizer and that is not what is implemented here.

Symmetry (Rating: 1) Simple reflections about the min and max extents are supported. This is minimal functionality, not a 2.

Input (Rating: 1) MeshTV supports both structured and unstructured meshes and a variety of unstructured elements. Note, however, that EnSight supports a more general class of unstructured elements including both 8 and 20 node hexas in 3D. Further, MeshTV can only read data in SILO format.

Output (Rating: 2) Ok.

Memory (Rating: -1) As Eric says, there is no way to return the GUI portion of MeshTV to a particular state although the command line portion's state can be restored from a log file. In general the lack of synchronization between the GUI and the command line portion of the program leads to problems like this one. There is no session capability like in Data Visualizer or EnSight.

4.8 EIGEN/VR

Large Data (Rating: 1) MeshTV does handle rectilinear data in a native manner which means significant memory savings for this type of data. However, this misses several very significant points for ASCII. EnSight has an enormous advantage in its Client-Server model which allows memory use to be spread across multiple platforms. MeshTV has no such model in production use at present although I gather that one is being experimented with. Also MeshTV does not take advantage of multiple CPUs on a single platform. A rating of 2 is too high for this category.

Remote (Rating: 0) Ok.

MP (Rating: 0) Ok.

Extra Features and Comments: None.

4.8 EIGEN/VR

Code Description: The EIGEN/VR (Exponentially Improving Graphics, Engineering, and Networking with Virtual Reality) project is located at Sandia National Laboratories in Albuquerque, New Mexico.

EIGEN/VR software is a result of Sandia National Laboratories' effort to develop an open, multi-purpose software interface between general classes of scientific information and a highly interactive, multi-dimensional visualization system (including the incorporation of immersive systems often referred to as virtual reality systems).

The focus of EIGEN/VR software is a human/computer interface for large scientific data visualization. The new feature which VR brings is the removal of the computer as the center object of perception allowing the user to interact directly with the information. "Adapting the computer to the human" and not the other way around. This definition of VR means input methods can range from the mouse to evolutionary tracking devices. At SNL, the information is the focus— not the expensive hardware needed tracking and voice recognition that is often associated with virtual reality.

The code is divided into three main sections:

- Applications – functionality unique to a specific problem. The user has full control of adding functionality via C programming. Rating Note: If the feature has been used in at least one EIGEN/VR application, the rating will be 1.0.
- Utilities – Some unique function in an application is considered of general use and has been made available in a set of utility libraries. User can simply call subroutine for additional functions. Additional features generally remain in utility status until numerous applications use this functionality. Rating Note: If in the utility libraries, the rating will be 1.5.
- Daemon – a standard set of functionality that the application gets by default. Functionality that is used over and over. Rating Note: If in the daemon, the rating will be 2.0. In some cases, basic functionality is provided by the daemon but advanced features are being tested in applications.

Hardware Platforms: MUSE currently runs on all SGI platforms, Evans and Sutherland, and SUN Microsystems. In all cases, OpenGL support is required. The system also contains utilities for data processing that run on the Paragon. The code is based C.

4.8 EIGEN/VR

Licenses: The product evaluated is EIGEN/VR not MUSE. The difference can be described as follows: Multi-dimensional, User-oriented Synthetic Environment (MUSE) was based on Laboratory Directed Research at Sandia National Labs. It has been commercialized under a technology transfer initiative to Muse Technology, Inc. The licenses agreement information can be obtained by contacting aandrews@musetech.com. The price quoted to Oak Ridge in March 1996 was \$60,000.

EIGEN/VR is a current project at Sandia National Laboratories with the focus on continuing the initial MUSE research initiative. The license for Sandia is limited to non-profit government institutions. The license is granted based on joint project proposals with the Sandia National Labs.

Contacts:

- George Davidson, gsdavid@sandia.gov
- Arthurine Breckenridge, arbreck@sandia.gov
- Phil Heerman, pdheerm@sandia.gov

E-Contact: <http://www.cs.sandia.gov/SEL>

Disclaimer: The recommendations in this document are made with specific consideration of ASCI requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Meeko Oishi (summer student) with additions by Arthurine Breckenridge (project contact with general focus) and Phil Heerman (project contact with ASCI focus).

Evaluation Hardware Environment: Evaluation were conducted using a mixture of SGI hardware configurations, with a mixture of IRIX 5.3 and 6.2. The system has been tested on a SUN Microsystems machine with no graphics acceleration. It works but is not recommended since you will lose the VR-sense of immersion (defined as response time less than 300 ms). Immediacy was obtained on the Evans & Sutherland graphics acceleration on a Sparc 20 processor.

EIGEN/VR functionality ranges from a basic keyboard, mouse, and monitor – to – what we consider a showcase system. The showcase system would include voice recognition, speech synthesis, sound generation, advanced 6DOF device, head tracking, and stereo projection. The VR immersion is worth the extra hardware to gain full understanding of the data.

Support (Rating: 2) Currently, the support is provided by the EIGEN/VR project team. There are developers actively adding new features to the code. Questions can be answered by sending e-mail to or calling one of the developers. There is no dedicated person for providing public support in general. However, there are two staff members dedicated to ASCI project needs. SNL has the source to make the product meet the needs of ASCI scientists. However, the EIGEN/VR version is not a commercial product. The commercial support is available if purchased from the startup company.

Documentation (Rating: 1.5) There is a web page under construction which references almost any type of documentation related to EIGEN and MUSE, including articles, reviews, a video library, the user's guide, the reference manual, an application developer's guide, a quickstart document, and a FAQ. Although most of these items are not yet available on the web, the User's Guide and Reference Manual are available in hardcopy. The User's Guide and Reference Manual contain simple example applications for ease in learning.

The key to a successful EIGEN/VR application is based on OpenGL which is a fully documented language. Data formats generally supported by EIGEN/VR are also documented such as Exodus II and CTH visualization files.

Turnkey (Rating: 1.5) For a developed application, the user can access EIGEN with simple executable commands: one for the EIGEN program daemon and one for the application. Someone with computer code knowledge must generate the application, but the user of the application needs no knowledge of code. For basic applications at Sandia based on CTH and Exodus II, turnkey applications have been developed. Other applications, for example Brookhaven Molecular Database information, have been developed within hours.

Therefore, for developed applications, EIGEN is highly turnkey (Rating: 1.8). The reason it is so simple is that the functionality provided by the daemon is available for all applications. The user must only concentrate on dealing with their unique data requirements.

For developing new applications, the turnkey rating is about 0 because it does require knowledge of OpenGL.

Simple (Rating: 2) The design principles behind EIGEN/VR is the human interface should be intuitive; thus the learning curve is minimal. However, the user is unfamiliar with this new 3-D tool and can easily get lost in 3-D space. Navigation tools are built in (e.g., floorgrid and wall map) to aid the user in this new experience.

The following is the evaluation of a customer using the tool for the first time. EIGEN involves the use of two types of vehicles: a plane and a hovercraft. The differences between these two modes are not easily discernable and easily confused. When controlled by a mouse, the same button has different effects, depending on its mode. This makes intuitive movement of the vehicle difficult. However, when using a full 6DOF device navigation was simple and lightyears better than the mouse.

The buttons on the console are easy to control and easily accessible. Their functionality is clear. All of the features displayed are used, and organized in a non-hindering way. Existing applications are relatively easy to operate. A basic set of functionality based on human interface principles has been provided. An application developer can add to or override this interface. Thus, a poorly written application could be developed and very difficult to master.

Variable Types (Rating: 2) EIGEN can handle scalar and vector data. Both cell-based and node-based data are supported. For example, current applications can support limited reading of Exodus, CTH viz, and Patran 2.4 files. There are two steps to the variable type: one reading the information and two converting it to a graphical representation. Readers from the Internet, etc are generally available. When we have worked with new data file format, generally some effort has been required to work with the data representation. Work is underway to improve/extend the existing readers in our utility library. For example, basic work has begun to add the ability to read SILO datasets. Applications can be customized using 'C', therefore if a 'C' reader exists or can be readily written it can be used by EIGEN in a 2D format and data fusion to place it in the 3D world. Often, the variable types are extendable to 3D representations.

Axis (Rating: 2) The axis functionality is present in all EIGEN/VR components. The daemon has a "wall map" drawing the universe position relative to objects and the operator's perspective. The axis is not in the field of view by design. It is displayed in a separate window since it is needed for orientation in 3D space.

4.8 EIGEN/VR

A utility is available to display the standard XYZ axis in the display scene. This functionality has been built into numerous applications to display XYZ axis relative to each object since an object can be controlled individually and will have its own orientation. Size and position are standard OpenGL commands.

Query (Rating: 2) Generally, this is a 2-D interaction and not used in 3D. 3D query function has been mapped to correlate more information at a time. The customer is viewing the 3-D representation and is show a complete data fusion of XY plot information or values are mapped to sonification. Other representations have been used. For example, data is represented on the isosurface based on color coding that the customer can change interactively, contour, etc. Applications do exist (e.g., Dino's dynamic meshing in 3D space) that do carry complete information with query techniques for point, vertex, etc.

Scale (Rating: 2) In the 3D world, the environment has a default scale based on setting of the LEAST and GREAT world coordinates (e.g., bounding box respresented by a grid on the bottom of the universe floor). The graphical representation is scaled to this world. Parameters exist to scale this world to the needs of the customer.

EIGEN/VR has the ability to magnify the objects as well as dynamically scale the values. The scale can also be changed by square root and logarithmic.

The application must provide the scale representation (e.g., ruler tool) since the world coordinates are not based on units.

Annotation (Rating: 2) The daemon has a simple annotation capabilty. Eight titles are available by default. Annotations can easily follow changes in the data. Under OpenGL version, the system uses the full power of X11 font processing. Also, stroke support has been added to allow text to displayed as 3D objects.

Algebra (Rating: 1) Eigen provides no support for algebraic manipulation of individual data points. The geometric objects do have full algebraic manipulation with OpenGL commands and matrix operations. See scale.

Viewing (Rating: 2) This is the main feature of the product. It's veiw functionality is way above other products used.

The user can fly around and through the data, pausing at any point to record the position of the craft and the position and time of the object. This can be saved as an overlay and returned to at any time. A headlight is available on the craft to increase and decrease lighting at the user's command. Full power of OpenGL lighting is also available.

Cut planes (Rating: 2) Standard in the daemon is one cutplane that can be toggled on/off, orientation changed with attaching to object (animated), and surface displayed as a rectangle on which information can be mapped. Full power of OpenGL cut planes. The user can create cut planes in any orientation based on $Ax + By + Cz = 0$. Mappings on the cut planes are demonstrated in a medical application.

Isosurfaces (Rating: 1.5) Isosurfaces are unique to data representation and not part of core deamon. So according to the EIGEN/VR self-defined rating system we could not get a rating of 2.0. However we have a working prototype application that will change this shortly. Using a comparision with other products, we have very advanced isosurface generation code.

Isosurfaces are generated in preprocessing utilities. Surfaces can be generated from any variable and colored with any variable. Legend of surfaces and coloring is provided set by a slider widget or a typein. The utilities run on the Paragon platform as well as the visualization systems.

The working prototype application is experimenting with dynamic generation of isosurfaces based on pre-processing min/max in a k-d tree. If the teraflop machine produces output in the correct form, the application can dynamically let the operator set thresholds and view different isosurface representations. The goal is to build a standard isosurface interface for the daemon.

Isovolume (Rating: 1.0) Not currently supported. Isovolume algorithm is available under virtual toolkit. One of the strengths of using OpenGL, tcl, etc. is that work on the Internet can be used. I used the rating of 1.0 based on the MUSTAFA evaluation. This is how easy it would be to add this feature in an application.

Geometries (Rating: 2) Full hierarchical control of geometries is support. An object can be colored, textured and moved as desired. No limit (besides hardware power needed) is placed on the number allowed.

Geometry Deformations (Rating: 1) No daemon functionality. Not sure of definition to clear state we have utilities.

For example, in an advanced manufacturing application. if one sees an error in a PRO/E dataset. The customer can change the information and re-save the file into a format needed directly by the tooling machinery. Or in the ship application, a wireframe is drawn to show the original shape so the deformation after collision can be compared. Is this what you mean by deformation?

Animation (time) (Rating: 2) Core strength of the product. Animations are functional with respect to both time and space. EIGEN makes full use of the SGI pipe rendering pipes. The system is optimized for maximum rendering speed. The operator has the ability to mark locations and return to time and space.

Animation (space) (Rating: 2) Core strength of the product. Animations are functional with respect to both time and space. EIGEN makes full use of the SGI pipe rendering pipes. The system is optimized for maximum rendering speed. The operator has the ability to mark locations and return to time and space.

Multiple problems (Rating: 1.5) The daemon is in tight drawing loop. It has no concept of multiple problems. Code is optimized at this point to be a “drawing” frame.

Another key feature of EIGEN/VR is data fusion. Multiple datasets from multiple sources. There are existing applications that read in multiple data sets. At present, the system has no limit to the number of data sets and processing on these data sets – there is however it is currently limited to the what will fit into memory so until pre-fetching is implemented in the daemon it deserves a lower rating than 2.0.

Thresholds (Rating: 1.5) In general isosurface thresholds are prespecified and calculated before display. Isosurface thresholds is an area of active tool development.

Two other utilities for calculating threshold values are supported:

- Instead of an independent isosurface, the values (e.g., pressure values) can be mapped onto the surface of another isosurface. These values can be color coded based on minimum and maximum threshold level.

4.8 EIGEN/VR

- Isosurfaces can be generated dynamically based on min and max values stored in k-d tree. The k-d tree must be pre-processed.

Erasure (Rating: 2) The user can selectively remove and add parts of the object. All functionality still holds for the remaining sections visible. Objects can be made transparent.

Streamlines (Rating: 1) Not currently supported in the daemon. One application, jupiter impact, uses streamlines but the data had this information based on pre-processing algorithms. I used the rating of 1.0 based on the MUSTAFA evaluation. This is how easy it would be to add this feature in an application.

Pathlines (Rating: 1) Not currently supported in the daemon. One application, jupiter impact, uses streamlines but the data had this information based on pre-processing algorithms. Pathlines were supported in the sense that streamlines were animated in time. I used the rating of 1.0 based on the MUSTAFA evaluation. This is how easy it would be to add this feature in an application.

Point Data (Rating: 1.5) Utilities are available for easy inclusion into an application. Point data is fully supported. One application involves the use of thermal data. The thermal temperature is represented by color. The user can also display data sets in form of graphs, plots, and glyphs.

Line Data (Rating: 1.5) Utilities are available for easy inclusion into an application. X-Y plots of history files are supported. Arbitrary plotting of data requires modification of the application and is not automatic. Line data can easily be mapped to sonification.

Symmetry (Rating: 1.5) Utilities are available for easy inclusion into an application. All objects can be mirrored about any plane. For some planes modification to the application is required.

Input (Rating: 1.5) Input to the daemon is eventually OpenGL based.

EIGEN can support a wide variety of data sets. Currently, only a limited number of readers have been developed for existing applications based on request. We are not building a generic product. We work on readers only when we have real work to accomplish. To date, we have not had a data format that we could not process. However, this is based on C programming or the strength of other products (e.g., media conversion).

Output (Rating: 2) EIGEN does contain advanced builtin features to support output. The BOOM has been programmed to push a button to take a snapshot of the screen so you do not have to move your viewpoint (e.g., like a camera). The daemon has a record session feature that produces an animation or can be journaled and playback in auto pilot mode.

The utilities available are basic unix functionality such as screen capture, etc. or advance data media (e.g. SIRRUS video) of SGI. One strength is we do not re-invent the tools but use existing products with a broad spectrum of options, when available.

Memory (Rating: 2) The daemon has built in return to time and space.

The journaling is based on capturing device actions. See output.

Large Data (Rating: 1) (Note: rating originally supplied as “0/1.0/2.0??”).

Relative to other products we desire a 2.0 rating. Relative to ASCI we have a lot of work to do. This is a hard problem. The basic design of EIGEN separates device control (slow) from rendering process (fast) to get the highest performance possible. Data in memory is the geometric information

4.8 EIGEN/VR

needed to render the output images. The largest machine tested had 2G (based on limitation of IRIX 5.3) and the interactive rates of 120HZ for stereo were still maintained.

Currently, to keep the interactive speeds needed for the VR immersion the data is limited to what is in memory or swap space. EIGEN/VR relies on data preprocessing tools. Therefore, large data sets can be reduced to meeting the memory limitation of hardware. This is focus area we are working on to support paging of large data sets into EIGEN using level of detail and hierarchical techniques. Working prototypes exist.

Remote (Rating: 1) (Note: rating originally supplied as “0/1.0/2.0??”).

Relative to other products we desire a 2.0 rating. Relative to ASCII we have a lot of work to do. This is a hard problem.

Several remote projects are in progress:

1. EIGEN/VR is fully X compatible and can display to a generic X Terminal with no loss of data. Functionality goes to “sleep” in the VR immersion sense.
2. We have separate input (via X11) from output (via RGB or video) to remote offices. Gives you an Infinite Reality on your desktop if you have the network for non-TCP/IP connection to IR.
3. Experimenting with GLR and works but takes over the IR pipeline so it is not much better than remote consoles except the transport mechanism is TCP/IP.
4. Experimenting with virtual collaborative environment where protocol used is X11 but when arrives at local station it uses full power of local hardware. Powerful software demonstration. This feature alone desires a 2.0 rating.
5. Experimenting with Germany’s COVISE remote software.
6. Experimenting with VRML 2.0

MP (Rating: 1) (Note: rating originally supplied as “0/1.0/2.0??”).

Relative to other products we desire a 2.0 rating. Relative to ASCII we have a lot of work to do. This is a hard problem.

EIGEN presently uses 2 processors. Work is underway to support more processors and tightly/loosely couple to interaction with the teraflop machine.

Extra Features and Comments:

Evaluator: Bob Kares. (Please note that this review, by Bob Kares, is a *companion* review, written in view of the original review, by Meelo Oishi, et. al.. Accordingly, there are comments for each item here only when Bob wishes to suggestion has additions or changes to the original notes.)

As I mentioned before I have not used this tool and the only direct knowledge I have about it comes from the video that Phil Heerman showed at the meeting. However, it is very apparent from reading Authurine’s comments in her evaluation that there is something seriously amiss with the scores she assigned to the tool. What I have tried to do below is to assign scores which seem to me to be more consistent with her description of the features of the tool. Please note that my scoring is based upon classifying ENGEN/VR as an applications builder, so that most of the numerical scores should be interpreted according to the “Implementation Difficulty” scale on page 3 of the evaluation document.

Evaluation Hardware Environment: Unknown.

Support (Rating: 1) It sounds like there is some level of support available from the developers but this is not at the level of a commercial product. Authurine says as much in her comments. She also says

4.8 EIGEN/VR

that there are two staff members dedicated to ASCI project needs. I'd like to know what this means in terms of support for the user.

Documentation (Rating: 1.5) Ok.

Turnkey (Rating: -2) This is clearly not a turnkey tool. It is an application development environment somewhat like KHOROS.

Simple (Rating: -2) It doesn't sound very simple from the description. I'd say that if you have to write C code and know about OpenGL library calls, then it isn't very simple.

Variable Types (Rating: 2) Ok. (As far as I can tell from the description.)

Axis (Rating: 2) Ok.

Query (Rating: -1) Sounds like it does not have a 3D probe as I mean the term but that one could be implemented with some work.

Scale (Rating: -2) Again according to the comments EIGEN/VR does not provide a scale representation and I can't tell how difficult it would be to implement something like a ruler tool.

Annotation (Rating: 0) It doesn't sound like there is very much annotation capability beyond simple title and color scales.

Algebra (Rating: -2) Aurthurine says directly that EIGEN/VR provides no support for algebraic manipulation of data. This would be alot of work to implement.

Viewing (Rating: 2) Ok.

Cut planes (Rating: 1) Comes standard with one cutplane. That does not rate a 2.

Isosurfaces (Rating: -1) Description says that isosurfaces are not part of the core daemon. Apparently they are generated off-line. This means no interactive control of the isosurfaces. You have to know what you want beforehand. This is alot of work to implement so that it is possible to pick isosurfaces interactively.

Isovolume (Rating: -2) The description says not currently supported. How could this get a rating of 1 under any interpretation of the rating scale? This would be alot of work to implement.

Geometries (Rating: 2) Ok.

Geometry Deformations (Rating: -2) Again there is no daemon functionality. How could this rate a 1 ?

Animation (time) (Rating: 0) Obviously EIGEN/VR has alot of interactive capability. But can I make time animations of multiple viz tools and save the results to a sequence of image files which is the functionality I want. Or do I have to get out my tractor trailer and haul the EIGEN/VR hardware somewhere.

Animation (space) (Rating: 0) Ditto for space animations.

Multiple problems (Rating: 1.5) It says here that "(the daemon) has no concept of multiple problems". Need I say more?

4.9 MUSTAFA

Thresholds (Rating: -1) Isosurface thresholds are preset since the isosurfaces are computed off-line. That means no interactivity and hence my score.

Erasure (Rating: 2) Ok.

Streamlines (Rating: -2) Not supported. That is what it says. And a lot of work to add.

Pathlines (Rating: -2) Again not supported.

Point Data (Rating: 1) If it supports reading and drawing of point data I give it a 1.

Line Data (Rating: -2) Doesn't sound like this feature is directly supported but probably could be added with alot of work.

Symmetry (Rating: 1) Does simple mirroring and that sounds like all.

Input (Rating: 1.5) Ok.

Output (Rating: 1) It's very unclear to me whether or not it is possible to obtain a simple sequence of image files in a standard format using EIGEN/VR.

Memory (Rating: 1) Again it sounds like there is some limited journaling based upon saving device actions. However, there does not appear to be a command language or builtin features for saving and restoring the full state.

Large Data (Rating: 0) Obviously EIGEN/VR is setup to do most of its preprocessing off-line on a big machine like the Paragon. This is not what I mean by handling big datasets. I'm talking about how big a dataset you can visualize interactively changing things like isosurface value on the fly. I don't think EIGEN/VR can do this kind of thing at all. Hence, I have assigned a score here which reflects my ambivalence on this issue.

Remote (Rating: 0) Can utilize OpenGL over a network but has no client-server model.

MP (Rating: 1) Ok.

Extra Features and Comments: None.

4.9 MUSTAFA

Code Description: This is an end-user application built with AVS/Express. It is a Motif GUI driven application, primarily aimed at Sandia's EXODUSII unstructured grid FEA databases, though it also has readers for AVS's geometry, unstructured, structured, and NetCDF file formats. The user interface options dynamically reconfigure themselves depending upon what type of data has been read. Other data format readers can easily be dropped in as a module. MUSTAFA is a general purpose tool that provides a wide variety of viz methods such as mesh plots, side and node set extraction, mesh displacements, element paint, multimaterial mapping, external edges, external faces, slice planes, isolines, isosurfaces, isovolumes, vector plots, streamlines, and particle advection. Additionally, models can be subsetted by thresholding on node or element data, element block, and cutting planes. Data can be clamped to specific ranges and datamaps are under user control. Miscellaneous features include Data query with an interactive probe, user annotations, 2d history variable plotting, 2d

4.9 MUSTAFA

variable vs distance plotting, time animation, rasterfile and postscript file output, interface to Sony laser video disc system. Multiple datasets and viewers can be active at any given time and multiple instances of each viz method can be also be active at any given time. MUSTAFA supports 8bit and 24bit displays and supports local hardware frame buffers (XGL, PEX, OpenGL) and has a software renderer for running over a network. Because MUSTAFA is built with AVS/Express, other features available in Express can easily be incorporated into MUSTAFA as user demand merits and all the other merits of Express can be leveraged.

Hardware Platforms: Those supported by AVS/Express: SUN SunOS and Solaris, Dec Alpha, SGI IRIX, IBM AIX, HP HPUX, Data General DGUX, Windows 95, Windows NT. Currently only actively developing on SunOS, Solaris, SGI, IBM, DEC, and HP.

Licenses: Requires a runtime AVS/Express license. A single runtime license is about \$2500 but AVS offers very aggressive volume discounting that can drop the price/license to about \$1200.

Contacts: Mike Glass, (505)844-8451 mwglass@sandia.gov Developer and User

E-Contact: None.

Disclaimer: The recommendations in this document are made with specific consideration of ASCII requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Mike Glass

Evaluation Hardware Environment: SparcStation 20 with dual 125 MHz Hypersparc processors, 256 MB RAM, 450 MB swap space, ZX graphics accelerator.

Support (Rating: 0) MUSTAFA was originally not a formally supported application but was developed in my spare time to meet my visualization requirements. However, it is now being used in a collaborative research project with an external customer and will receive funding for further enhancements, support, and documentsion.

Documentation (Rating: 1) A draft of a User's Guide is available in postscript format. It is supplemented by the AVS/Express Data Visualization Manual which describes the various visualation methods in more detail.

Turnkey (Rating: 2) This is a completely turnkey application. However, since it was developed with AVS/Express, it is very easy for a developer with an Express developer's license to added new or customized features.

Simple (Rating: 2) Hard for me to answer this since I developed it so am very familiar with it. From other user's comments, it is pretty easy to use with not much of a learning curve. Most have been able to perform their visualizations without ever having to open the User's Guide.

Variable Types (Rating: 2) MUSTAFA has viz methods for scalar and vector data. Everything is treated as scalar fields and the user builds vectors by defining which scalar fields make up the vector components. It reads nodal and element variables. The only viz methods available for element data is are a "paint cells" and "multimaterial map" method. Element data is also interpolated to the nodes so that all the other viz methods can operate on them. Both NULL node and element data is supported.

Axis (Rating: 1) Currently this is not available. It would be fairly easy to add since Express has an Axis module available that could be integrated into MUSTAFA. The rating is based on how easy it would be to add this feature.

Query (Rating: 1) Data vaules can be queried at a (x, y, z) location via a mouse driven probe or user typein. For node data, the interpolated value is returned. For element data, the interpolated value obtained from the element-to-node interpolation and the actual element value is returned. A popup window displays the element number the probe is in, the nodes that make up the connectivity of that element, the data values at those node locations, and the element value if an element data variable is being queried. It does not highlight the element. You can only query by spatial location, not by element or node number.

Scale (Rating: 1) MUSTAFA doesn't really have a viz method to do quantitative measuring. Once again, this would be fairly easy to add since Express has a module for displaying axis and grids with tick marks, etc. It would also be easy to add a method similar to the data query method which would accept two mouse clicks and calculate the distance between them. The rating is based on how easy it would be to add this feature.

Annotation (Rating: 1) Supports simple 2D text annotations. Text position, font, size, color, and other miscellaneous attributes can be set by the user. For more extensive annotions, the image must be imported into an application such as FrameMaker where further annotations can be added.

Algebra (Rating: 1) Currently this is not available. Once again, this would be fairly easy to add since Express has a module for performing math operations on field data. The rating is based on how easy it would be to add this feature.

Viewing (Rating: 2) The user has full control over the camera (position, projection method, etc), the object (transformation, transparency, material properties, etc), the lighting (number turned on, position, type, intensity, etc), the renderer (software or hardware), and the background color.

Cut planes (Rating: 2) Cut planes are available for solid contouring, isolines, and vector plots. The visibility of the cut plane itself can be toggled on/off and its color and transparency can also be set by the user. The cut plane can be position by the mouse or by a popup interface to the plane's transformation matrix. A colormap editor is available along with a color legend.

Isosurfaces (Rating: 2) For each instance of the isosurface viz method, up to 5 isosurfaces can be set. Any component can be color mapped onto the isosurface. The isosurface value is set by a slider widget or a typein. A colormap editor is available along with a color legend.

Isovolume (Rating: 2) The isovolume viz method creates a volume between 2 bounding isosurfaces. Any component can be color mapped to the isovolume. A colormap editor is available along with a color legend.

Geometries (Rating: 1) Geometries that can be defined as an unstructured mesh can be viewed in wire-frame mode (with or without hidden line removal) or in shaded surface mode (with or without the mesh lines). The color of the lines and surfaces can be adjusted by the user. Geometries such as circles, spheres, cylinders, etc cannot be input as a primitive.

Geometry Deformations (Rating: 2) The displacement vector is specified by the user and then applied to the mesh coordinates. All the components of the displacement vector can be applied or just single components, this is useful for code debugging purposes.

Animation (time) (Rating: -1) Animation in space and viz method parameters are not currently supported by Express and hence, are not supported in MUSTAFA. I'm lumping all the animation capabilities for the various viz methods into this category since a keyframe animator for spatial position should also be able to animate viz method parameters as well. A beta version of an animation module is available for Express v3 and AVS says a keyframe animator is under development.

Animation (space) (Rating: -1) Animation in space and viz method parameters are not currently supported by Express and hence, are not supported in MUSTAFA. I'm lumping all the animation capabilities for the various viz methods into this category since a keyframe animator for spatial position should also be able to animate viz method parameters as well. A beta version of an animation module is available for Express v3 and AVS says a keyframe animator is under development.

Multiple problems (Rating: 2) MUSTAFA supports multiple data files, viewers, and viz methods. The output from the various data files / viz method combinations can be sent to the same viewer or separate viewers.

Thresholds (Rating: 2) The range of the data values can be clamped globally or locally for each viz method. Data outside of a specified range can also be set as NULL data which will cull that portion of the mesh.

Erasure (Rating: 1) Erasure is available as a local filter for each viz method. It works for both node and element data. The only erasure probe available at this time is a cut plane which can be arbitrarily oriented and the mesh on either side is erased.

Streamlines (Rating: 1) Streamlines can be launched from the mesh nodes, a line probe, or a plane probe. A list of user specified points cannot be specified. The streamlines can be colormapped to the vector magnitude or not. As a separate viz method, massless particles can be animated (advected) along the streamlines. One can choose from a selection of glyphs to represent the particles. The glyph size and color is adjustable by the user.

Pathlines (Rating: -1) This method is currently not supported and is not available as an Express module but user written modules can easily be integrated into the Express environment which would allow for development of a module to perform this function.

Point Data (Rating: 1) Currently this is not available. Once again, this would be fairly easy to add since Express has a module for displaying point data. The rating is based on how easy it would be to add this feature.

Line Data (Rating: 1) A line probe is available which samples the data at a user selectable number of sample points. The data is then displayed as a small x-y plot in a separate window. The data can also be written to a file as x-y pairs so they can be read into a user's favorite x-y plotting tool for further processing. The position of the line probe can be manipulated with the mouse, typeins for the endpoints, or a popup interface to the line's transformation matrix. Support for data values as a function of time is not supported.

Symmetry (Rating: 1) Symmetry is only available by mirroring in x, y, or z. It is pretty efficient since only the field's transformation matrix is adjusted, not it's actual coordinate values. Reflection about an arbitrary plane would be fairly easy to implement.

Input (Rating: 1) MUSTAFA currently supports the ExodusII format for unstructured meshes. Hooks exist in MUSTAFA to build it with multiple readers for other formats. Readers for AVS5's native field and geometry formats have been added this way as a proof in concept though they are not available in the general release version of MUSTAFA. The ExodusII reader supports the standard finite element type such as hexes, tets, quads, etc. In addition, it supports generalized 2D polygons and 3D polyhedron surfaces.

Output (Rating: 1) MUSTAFA supports rasterfile dumps of the viewers in AVS image, TIFF, PPM, and YUV formats. It should be noted that the resolution of the rasterdumps can be set independently of the resolution of the viewer (or the screen) so fairly high resolution rasterfiles can be created. In addition, postscript and CGM formats are also available. An autoincremented sequence number can be appended to each file name. This is useful when doing a time animation. The sequenced files can then be assembled into a MPEG bitstream by a MPEG encoder.

Memory (Rating: -1) Currently this is not available. Journaling may be difficult to implement due to Express's V language concept. However, a commandline scripting support would be fairly easy to implement along with resource files. Support for commandline scripting and resource files will be added as part of the project with an external customer.

Large Data (Rating: 1) Express uses a data reference paradigm which is much more efficient than a data flow paradigm. I've used it routinely on 250K node datasets with 36 nodal variables and 18 element variable on my desktop Sparcstation. It has been used on a 1 million node problem on our large SGI visualization server.

Remote (Rating: 0) MUSTAFA (Express for that matter) does not support remote client/server type execution where the various portions of the application (such as viz methods or data readers) can execute on different machines. It does support a software renderer so it can execute on a remote machine and display back to your desktop. AVS is working on adding remote modules (much like in AVS5) to Express.

MP (Rating: 0) MUSTAFA (Express for that matter) does not support multiple cpus though AVS is actively developing a MPI based version of Express for parallel visualization.

Extra Features and Comments: Though MUSTAFA does not support time history plot of element or nodal data, it does support time history plots of global variables. The x-y plotting package xmgr has been integrated into MUSTAFA for this purpose. This also serves to show how easily other tools can be integrated into MUSTAFA.

In addition to isolines and isovolumes, MUSTAFA also supports isolines. The isolines can be displayed on the exterior surface of 3d models or on an arbitrary cut plane passed through the model.

MUSTAFA can also display the side sets, node sets, and element blocks associated with a finite element analysis.

An interface to Sony's laser video disc is also supported. This allows time animations to be directly recorded to video disc.

MUSTAFA is based on three main objects, Readers, Viz Methods, and Viewers. Any number of these objects can be present at any given time, subject to system resources of course. Connections can be made between any Reader to any Viz Method and connections can be made between any Reader to any Viz Method to any Viewer. This allows for a pretty flexible visualization system. Each Reader consists of a datafile reader and a set of global filters. The output from the global

filters is what is available to the various viz methods. Each Viz Method consists of a set of local filters, which only operates on the data for that particular Viz Method, and the actual viz tool itself. This combination of global and local filters provides a great deal of flexibility without becoming too complicated.

During the course of development of MUSTAFA, I was very impressed by how easy it was to build a fairly complete application with Express. Because of Express's object oriented approach, it is very straight forward and easy to extend the capabilities of MUSTAFA.

Evaluator: Bob Kares

Evaluation Hardware Environment: This evaluation examined MUSTAFA Version 1.15, and was performed on an SGI Indigo2 High Impact workstation with a 250 MHz R4400 CPU and 128 MB of memory under IRIX 6.2.

Support (Rating: 0) This product is not formally supported, although Mike Glass is willing to answer questions and fix problems as his time permits. No formal bug tracking and problem resolution are available, however.

Documentation (Rating: 1) The MUSTAFA documentation consists of a User's Manual written by Mike supplemented by the documentation for AVS/Express. The User's Manual is brief but adequate to get the user started with MUSTAFA and Mike provides some EXODUS II databases to use while learning the code. There is a menu selection for on-line help but it is not yet implemented. No tutorial is available.

Turnkey (Rating: 2) MUSTAFA is an end-user application written in AVS/Express and is run as a standalone turnkey application. Note, however, that an AVS/Express development or runtime application license is required to execute the MUSTAFA binary.

Simple (Rating: 2) MUSTAFA is very straightforward to learn and use. All of MUSTAFA's basic functionality is accessed from the menu bar on a small window in the upper left hand corner of the screen called the main menu window. The pulldown menus allow the user to access all the major functions. For example, the Viz Methods pulldown menu allows the user to create as many different visualizations in the current data viewer as one would like, to delete visualizations, or to toggle their visibility on and off. (By visualization here I mean something like a cutplane or an isosurface, a particular representation of the data. In Data Visualizer this would be called a tool.) A visualization may also be "selected" from this pulldown menu causing the controls for it to appear in the main control window which runs along the far left hand border of the screen, just below the main menu window. Buttons in the main control window for a visualization often popup additional windows containing control widgets like the window for the color map editor. MUSTAFA is particularly convenient in the way in which it handles time varying data. The main control window has a slider and associated typein for moving through time and the corresponding visualizations are automatically updated as the user changes time even if the view is rotating. In fact if the user selects a simulation time value between two time dumps in the data file then interpolation of data in time between the two dumps is automatically performed. All in all the screen organization is very similar to EnSight, not as clean or convenient as Data Visualizer but still rather easy to use. It is quite easy to discover how to do something with this tool because virtually everything one can do is accessed via the main menus. The learning curve is short and the less frequently used advanced features are properly hidden in the control windows for the visualizations.

Variable Types (Rating: 2) The code has viz methods for both scalar and vector data. The only viz methods that directly support cell based as opposed to node based data are the paint cell and multimaterial map methods. However, the code can interpolate cell based data to the nodes so that the other viz methods are available for the interpolated cell data as well. MUSTAFA also supports the concept of NULL data. Elements and nodes with NULL data are culled from the model. Thus if you have element data available on only one part of the mesh, the rest of the mesh is discarded.

Axis (Rating: -2) There is no way to place a set of axes in the field of view. This is quite annoying since it is difficult to understand your orientation without them.

Query (Rating: 1) MUSTAFA has a useful 3D probe. By pointing with the mouse and clicking while holding down the Control button the user can read the value of a scalar at a point along with the (x,y,z) coordinates of the point. Clicking the “More Info” button at the bottom of the probe’s control panel pops up a window which displays the element number that the probe is in and what nodes are associated with it and their nodal values. For element based data it will show both the element value and the interpolated value obtained from doing the element-to-node interpolation. It would be useful if the probe would draw the outline of the element which it is in as Data Visualizer does.

Scale (Rating: 2) It is possible to measure the size of an object with MUSTAFA’s data vs distance viz method which is rather similar to the ruler tool in Data Visualizer. What happens when this viz method is selected is that MUSTAFA brings up a line probe in the view window whose orientation and position can be adjusted from a popup window reached from the data vs distance control window. The value of a scalar is plotted vs distance along the line of the probe in a separate small X-Y plot window which pops up when the visualization is created. The probe position and orientation can also be directly manipulated via the mouse by selecting the probe to be the “selected” object. This is done with the “object selector” available from the “ViewEditors” menu in the data viewer window. There is a bounding box mode which you can toggle off and on to cause large visualizations to be replaced by a bounding box representation during view transformations like a rotation in order to avoid slow response in adjusting the view of the data. However, this bounding box has no scales attached to it and is thus of no use in measuring the size of objects.

Annotation (Rating: 1) MUSTAFA supports some simple annotation of the data visualization. 2D text labels and titles with underlining or a bounding box can be placed anywhere in the view. A choice of X window fonts for text is also available. Simple time dynamic labels are also possible for displaying simulation time on animations. However, these labels cannot be attached and stored with that data. 3D text is not supported, nor are arrow glyphs and other aids to labelling features of interest in the scene.

Algebra (Rating: -2) No algebraic or vector analysis operations on the data are supported.

Viewing (Rating: 1) The user may view the data from any angle or distance with relative ease. Multiple data viewers are supported, each in a separate window, although I think that this is done more conveniently in EnSight. There are 4 light sources available for each viewer. Only one light source is on by default. The others can be turned on from the “Lights . . .” option under the “ViewEditors” menu in the data viewer. Properties such as color and so forth can be adjusted by the user. Each viewer can be set for either orthographic or perspective projection.

Cut planes (Rating: 1) Cut planes are pretty well supported. You can have as many as you like, oriented at any position and angle to the data. Orientation and position of the plane in MUSTAFA is controlled either with a popup and sliders or by direct positioning with the mouse in the viewer. The later

method is achieved by selecting the cut plane to be the “selected” object using the “object selector” menu item available under the “ViewEditors” menu. Only scalars may be plotted on the plane. The color map is user editable to some extent but not as fully as in EnSight or Data Visualizer. Vectors, contours and isolines are not provided on the cut plane; they are actually separate 3D viz methods in MUSTAFA. Vector fields and isolines can be plotted on planar probes in the corresponding vector hedgehog and isoline viz methods. However, these planes are defined separately within those viz methods and are not related to a particular cut plane. No animation of cut planes is supported except time animation. That is to say, you cannot animate the motion of a cut plane being dragged through the data although you can animate the variation in time of the cutplane at a fixed location in space. A color bar is provided and its position and labelling is user selectable.

Isosurfaces (Rating: 1) The user can make an isosurface of any scalar variable with MUSTAFA. The way the isosurface functionality is organized in MUSTAFA is that isosurface is one of the selectable viz methods and each creation of an isosurface visualization allows the user to display up to 5 isosurfaces. However, since an arbitrary number of visualizations of the same type is allowed, this means that effectively there is no limit on the total number of isosurfaces. Actually, the way MUSTAFA is organized is quite convenient here because if the user wants to display 5 or fewer isosurfaces of the same scalar variable only one isosurface visualization needs to be created to do that. An isosurface can be colored by any other scalar variable but you cannot plot vectors on it. Again no animation of the isosurface is allowed in the sense that you cannot animate the appearance of the isosurface as the isovalue is varied at a fixed time although the time variation of an isosurface of fixed isovalue can be animated. A color bar is provided and its position and labelling are user selectable.

Isovolume (Rating: 1) MUSTAFA supports isovolumes rendered as a volume region between two isosurfaces of a given value with the surface of this volume colored either by the isovolume scalar or any other scalar value. No transparency of the volume is supported; only the surface of the volume is visible. No vector field can be plotted on the volume surface. The user can select the range of values which bound the volume with sliders or a corresponding typein and the color map editor allows the data-to-color assignment to be controlled by the user although again EnSight and Data Visualizer both have a better color map editor. The volume cannot be animated through a range of values at a fixed time. An arbitrary number of isovolumes may be created and a color legend is provided.

Geometries (Rating: 1) Mesh objects may be visualized in wireframe or a variety of shading methods with the color of the shaded surface or mesh lines independently selectable. MUSTAFA applies what it calls global filters to this mesh data to transform or cull particular groups of elements before passing them over to a visualization method. A variety of these filters are available, including mesh displacement, element block selection, mirroring, data thresholding, and data clamping. The mesh displacement filter, for example, creates a new mesh by displacing the nodes of the original mesh by an amount proportional to the value of a scalar on the mesh. Mirroring reflects the mesh about the x, y or z axes or some combination of these. An arbitrary number of these objects is allowed.

Geometry Deformations (Rating: 2) MUSTAFA does most of what is required in this category. You can use vector glyphs at the nodes of the original mesh to represent mesh displacements colored according to magnitude or normalized with fixed user-selected color. You can also apply a global filter called “Mesh Displacement” to automatically create a new deformed mesh based on the vector displacements stored in the Exodus II file. You can also represent scalar quantities in shaded color representation on the displaced mesh. All this may be animated in time and all representations are automatically updated so that, for example, it is straightforward to create a time animation of the mesh deformation.

Animation (time) (Rating: 1) Time animation is pretty well supported in MUSTAFA. The time animation controls allow the user to create flipbook animations in time with user selectable start and stop times and time increment. Data at simulation time values between two data dumps in the database are automatically computed by interpolation. All visualizations active at the time the animation is performed are automatically updated as time progresses. One somewhat inconvenient aspect of MUSTAFA's method of doing time animation is that the frames are generated from scratch by redrawing the visualizations every time the animation is run rather than caching the frames so that they can be played back at speed. Keyframe animation is not supported.

Animation (space) (Rating: 0) The MUSTAFA user can do some very limited spatial animation in the following sense. When the user starts a spatial motion in the viewer using the mouse by releasing the left mouse button when the mouse is still in motion, that motion will continue until the user again clicks the mouse in the viewer to stop it. This represents a very crude sort of spatial animation which is limited only to motion of the view point of the user. It cannot be recorded to a file, nor can animations of the visualization parameters like an isosurface value be performed. Some combination of spatial and time animation is possible in the sense that if the user starts a spatial motion of the view point with the mouse, and then a time animation, the spatial motion is captured in the time animation, and this can be save to disk. Generally speaking, animation is one of MUSTAFA's significant and obvious weaknesses, and this is a result of the essentially non-existent support for animation in the current release of AVS/Express.

Multiple problems (Rating: 2) MUSTAFA can support the simultaneous visualization of an arbitrary number of multiple problems. It does this in an interesting way. MUSTAFA has the notion of a Data Group which consists of an input dataset read from an external database, together with global filters that are applied to the data. The result is then fed to one or more viz methods. Visualizations from different Data Groups can be sent to one viewer or multiple viewers so there is a lot of flexibility in displaying multiple problems on different meshes simultaneously. This is an improvement over EnSight which can handle a maximum of 8 cases at once. Note, however, that although EnSight supports a limited number of simultaneous cases, the user's ability to view them with EnSight is rather more convenient than with MUSTAFA. With EnSight one can split the main view window several different ways and put a case in each portion of the window, making side-by-side comparison particularly convenient.

Thresholds (Rating: 2) MUSTAFA's global filters provide thresholding and data range clamping for all the scalar variables.

Erasure (Rating: 1) MUSTAFA has a plane erasure feature which is implemented as what it terms a "local filter" associated with every viz method. The plane can be moved but not animated. No 3D erasure is available.

Streamlines (Rating: 1) The tool does support streamlines launched from an arbitrarily oriented line or plane or from each mesh node. Streamlines may be colored by the magnitude of the vector field or not at the user's discretion. Various controls are present to orient the launch probe, control the order of the integration and the number of steps. Particle advection along the streamlines is supported with a user choice of particle glyph and glyph normalization. Glyphs can be scaled according to the magnitude of the vector field. I should note that while these capabilities are available, they do not seem to be completely robust. I crashed the code several times playing with these tools. Stream ribbons are available but they do not seem to display when turned on. I would also comment that EnSight does a much better job on streamlines.

Pathlines (Rating: 1) This tool does support pathlines in the sense that streamlines may be animated in time like all the other viz methods.

Point Data (Rating: -1) There is no point cloud sampling of a 3D data volume as in Data Visualizer. There is also no method for displaying experimental data at points in space using shaded glyphs or glyphs of varying size. There is a viz method called Show Node Sets which allows the user to display defined sets of nodes as points or sphere glyphs.

Line Data (Rating: 2) MUSTAFA has a feature very much like Data Visualizer in which a line probe can be oriented in the data volume and an X-Y plot of scalar variable vs distance can be displayed. The line orientation can be controlled via a popup and the tool creates a small X-Y plot window when this viz method is initiated. This X-Y plot is updated automatically as the probe is moved around. Again I would remark that the ruler tool in DV is more convenient because it can be dragged around in the viewer with the mouse whereas MUSTAFA uses a popup window with sliders to orient the probe. The X-Y data can be output to an ASCII file but no method for directly printing the X-Y plot is supported nor is there a method for turning the contents of the X-Y plot window into a raster file.

Symmetry (Rating: 1) MUSTAFA implements symmetry operations via the global filter for mirroring. Mirroring in x, y and z or combinations of these is implemented.

Input (Rating: 1) The version of MUSTAFA which I was using accepts fully unstructured mesh data in the EXODUS II database format. I gather, however, that implementing a reader for other input formats is straightforward. The element types supported derive from AVS/Express and include hexas, prisms and tets and their 2D counterparts. All three of these element types can have nodes on both the corners and edges of elements so that, for example, both 8 and 20 node hexas are supported. MUSTAFA also supports 2D n-sided polygon elements. MUSTAFA does not have a simple ASCII native file format which can be constructed by the user.

Output (Rating: 1) The only outputs which MUSTAFA can generate are raster files or print files of the viewer contents. The raster file formats supported are AVS image, TIFF, PPM and YUV. The print file formats supported are color PostScript and CGM. It would be nice to see native support for the SGI RGB format as well but the PPM format could be converted to that if desired. The time animation feature can be used to generate frames in raster file format to disk with automatic sequence numbers and a file prefix. The tool could probably add support for any ASCII Visualization format adopted with modest effort.

Memory (Rating: -2) The tool has no capability to save a configuration and reload it to restore all the viz methods and data. It also has no journaling capability meaning it cannot record user actions for later playback. It has no scripting capability.

Large Data (Rating: 1) MUSTAFA has the concept of a Data Group which is a dataset which is read from a file and has viz methods associated with it. Data Groups can be loaded and then deleted, presumably freeing up memory when they are removed. The largest dataset I had available had only 240 K nodes, but the multimaterial map I displayed in 3D was reasonably responsive under interactive rotation. The tool does not use multiple CPUs on a multi-CPU platform.

Remote (Rating: 0) MUSTAFA does not support distributed visualization in the sense of EnSight. There is, for example, no server which may run remotely and perform the number crunching on a big machine and a remote desktop client which renders the visualization. However, MUSTAFA can utilize OpenGL across a network. I tried remote logging into a second Indigo2 Impact machine

from the Indigo2 Impact machine on my desktop and then set the DISPLAY variable on the remote machine to display back to my desktop Indigo2. The resulting OpenGL performance across the network showed only modest degradation relative to running the tool on my desktop machine. I also tried this same procedure by logging into the second Indigo2 machine from an older SUN IPX running SunOS 4.1.2. Again I was able to open MUSTAFA windows back to the older SUN but performance was very poor as one might expect since the software renderer is utilized in this case.

MP (Rating: -2) MUSTAFA is based upon Express and currently Express does not utilize multiple CPUs on SMP type machines. This is in contrast to IBM's Data Explorer which automatically partitions work for graphics rendering and other operations among multiple CPUs on the SGI Onyx platforms and other vendor machines as well. Some work is currently being done by Gary Oberbrenner on creating an MPI based version of Express for doing some types of parallel rendering but this is NASA supported work and is currently not a part of the mainline of Express development. Although there is some prototype parallel code available for the SP2, there is currently no clear plan to support this development in the production release of Express. LANL has been told that this could change if we were willing to spend some money to make it happen.

Extra Features and Comments: The datasets used were in the EXODUS II format and were provided by Mike Glass. All were relatively small compared to our typical 3D datasets, the largest database used, rsc.exo, being about 240 K nodes.

There are a couple of extra MUSTAFA features worth mentioning. One is the time history plot feature which allows you to plot global variables in an EXODUS II file over time. In addition to time history plots you can also make scatter plots of one global variable against another. The X-Y plotting package xmgr is used for this purpose and variables are selected from scrollable X and Y lists. X-Y data may be written to a file and plots may be printed directly from the package. Another feature worth mentioning is the support for a video disk video production system added in Version 1.15. It looks like either a socket connection over a network or a direct serial interface hookup to the video disk recorder is supported although I was not able to test this feature.

The process of looking at MUSTAFA is been very interesting to me for several reasons. First, it is of considerable interest to see how much can be accomplished in a period of the order of a year by a single developer starting with a high level applications builder like AVS Express and some consulting help (4 to 6 weeks worth) from AVS. MUSTAFA has many if not most of the basic features I would require in a 3D viz tool. In fact, it is quite comparable in functionality to the LANL Generalized Mesh Viewer (GMV) which has been under development for several years and is written from scratch in C. It seems clear to me that a high level applications builder like AVS/Express can provide substantial benefit when used intelligently. One of the particularly nice features of MUSTAFA is how easily it can be modified because it was built with Express. According to Mike Glass adding new features and viz methods is straightforward. It is also clear, however, that MUSTAFA suffers from the shortcomings of Express as well. One sees that particularly in the area of animation where Express has still not subsumed the animation functionality of AVS Version 5 into the Express product.

Another point which comes through clearly in evaluating MUSTAFA is that while MUSTAFA has many if not most of the basic functionality needed in a 3D viz tool, it does not have the great richness of features and capabilities of commercial products like EnSight and Data Visualizer. EnSight and Data Visualizer have a great deal of manpower and development effort put into it, as well as the benefit of user input from a large number of customers and all this really shows in the richness of the product when compared to MUSTAFA.

4.10 Phigure

I think that this comparison identifies a critical point which should be of great interest to the ASCII viz effort. To achieve richness and flexibility in a viz tool, you need not only developer manpower and a powerful development environment, but also input from a sufficiently large community of users whose requests and ideas you can incorporate into the tool to grow its capabilities. That is something we need to think carefully about before we go off and try to develop our own viz tools for ASCII, even if we start from a high level applications builder like AVS/Express.

4.10 Phigure

Code Description: PHIGURE is a high-level library for building visualization utilities and applications. For ASCII visualization needs, it should be well suited for applications requiring efficient, multi media graphics output- e.g. steerable and direct output applications where graphics data file construction would be too expensive in I/O and size.

PHIGURE enables the construction, visualization and animation of simple or very complex 2-D/3-D graphic objects. Due to its functions for creating graphic objects including axes, 2-D/3-D isosurfaces on Cartesian, polar or spherical coordinate planes, it simplifies the job of the engineer designing pre- or post-processors for scientific applications. Statisticians can use it to represent data quickly and easily in the form of curves, histograms, pie charts, etc.

The PHIGURE library is based on the PHIGS ISO standard. All graphic objects generated by PHIGURE are made up of primitives defined by this standard. PHIGURE may therefore be considered a PHIGS superset, taking full advantage of all the capabilities provided by the standard. PHIGURE was developed using the GPHIGS library produced by G5G. This concept enables maximum portability of any application developed with PHIGURE, as well as optimum use of the graphic capacities of the terminal, the workstation or the printer used.

PHIGURE has many mesh objects, iso-contours, vector fields, etc. It has many X/Motif widget tools for rapid GUI development. It handles 2D and 3D rectilinear, curvilinear and unstructured mesh data.

PHIGURE is also now available for Open Inventor from TGS.

A parallel z buffer extension has been developed for PHIGURE (and GPHIGS) for use on MPP applications. Contact Jeff Rowe for availability.

Hardware Platforms: SGI, HP, IBM, Sun, DEC, Cray, Meiko, Macintosh, IBM PC

Licenses: Phigure must be purchased from G5G (mainframes about \$25K, workstations about \$5K)

Contacts:

- Jeff Rowe 510-422-4042 jrowe@llnl.gov
- G5G, A TGS Company, Avenue Henri Becquerel, Parc d'Activites Kennedy I, BP 227, 33708 Merignac cedex (FRANCE). Phone : (33) 56 13 37 77. Fax : (33) 56 13 02 10. Contact: Jean-Pascal Butte, Marketing & Sales Manager. Email: jpbutte@g5g.fr.
- TGS, Template Graphics Software, Inc., 9920 Pacific Heights Boulevard. Suite 200. San Diego 92121-4331. Phone : 619/457-5359. Fax : 619/452-2547.
- Local Contact: Steve Stuckey, Regional Sales Manager, TGS, 1190 Saratoga Ave, Suite 210. San Jose, CA 95129-3456. Phone : (408) 236-3296. Fax: (408) 236-3235. Email : steve@tgs.com.

4.10 Phigure

E-Contact:

- http://www.llnl.gov/ascii/ads/G5G_mosaic/G5G.html
- http://www.llnl.gov/ascii/ads/OPGS-DOCS/6.0/doc/OPGS_DOC.htm(Version 6.0)

Disclaimer: The recommendations in this document are made with specific consideration of ASCII requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Jeff Rowe.

Evaluation Hardware Environment: PHIGURE is installed at LLNL Livermore Computing Center on the MEIKO CS/2 MPP (Solarius 2.3), a Cray YMP and a Cray J90 cluster (Unicos 7.0), and on the former NERSC Cray T3D.

Support (Rating: 1) The G5G products are well supported. G5G has provided rapid response to the few problems I have had. The main problem I see currently is that the staff for support in the US is very limited. Complex problems may have to be referred to Europe which can take considerable longer to resolve.

Documentation (Rating: 2) The manuals and on-line documentation are excellent. The HTML documentation is very complete with detailed examples and graphics.

Turnkey (Rating: ?) PHIGURE and GPHIGS are graphics libraries to be used in building visualization applications. These libraries provide many widgets for building Motif and X interactive applications. In addition, GPHIGS_GUI provides a graphics user interface toolkit used to manage expose and resize, and simplifies the management of actions in the graphic window, particularly locator and pick with the Motif callback mechanism. The use of this widget enables high-level manipulation functions to be accessed.

Simple (Rating: 1) In the context of application programming, PHIGURE offers high level and powerful modules. For example, a single module can plot an entire 3D mesh object or a 3D vector field, or an iso surface or contour. Using the many sample programs to cut and paste into your application can greatly reduce coding time. On the other hand, applications which require special effects or viewing control, or which need to utilize PHIGS, can get quite involved.

Variable Types (Rating: 2) PHIGURE has primitives for scalar and vector data . Mesh data can be colored based on cell or average of node values. Cell edges can be colored on average or cell value. User can specify min or max undefined values for cells not to be processed for isoline or isosurface calculations.

Axis (Rating: 2) PHIGURE provides an extensive set of and axis drawing routines for linear, log, polar, and radial grids in both 2D and 3D. Time , calendar, and format axis are also available. All axis properties such as size, number and orientation of tick marks, size orientation and font for text are settable. Application has complete control over grid or axis location.

Query (Rating: 0) PHIGURE and PHIGS have input functions for locator, stroke, string, valuator, pick and choice devices. These devices can be configured to operate as Motif or X window widgets or actual physical input devices. They can operate in sample, event or request input mode. Locator,

4.10 Phigure

stroke and pick devices provide provide information which could be mapped to application mesh data. However, the pickability of 3D mesh cells is dependent on the primitives used to represent the mesh. Single cell pickability would require mesh decomposition into low level 3D elements-and thus a greater amount of data and processing.

Scale (Rating: 2) Providing an interactive grid or scale for measuring could be easily added as a separate object under the control of a menu and input device or widget.

Annotation (Rating: 2) PHIGURE has functionality for full 3D text. Text and text attributes are handled just like any other object in the picture. Text attributes include color, font, size, orientation and justification. Text is subject to all the normal viewing and picture transformation. Annotation text can be tagged (linked via a line) to picture elements and is not affected by viewing transformations.

Algebra (Rating: -2) Not available.

Viewing (Rating: 2) PHIGURE provides complete 3D viewing setup and control via interactive Motif/X widgets. Applications can of course modify or design their own viewing control interfaces. Multiple light sources, depth, and shading options are available.

Cut planes (Rating: 2) Functionality for cut planes is excellent. Any number of cut planes can be used. Orientation is settable and interactive widgets can be used to adjust these and other cut plane parameters. Contours and iso lines with appropriate color settings can be mapped to the cut planes. Again, interactive widgets can adjust the number and attributes these elements.

Isosurfaces (Rating: 2) A full array of isoline and isosurface mesh elements are available. Isolines can be defined for main and secondary contour lines. Non overlapping annotations can be applied. Isoline colors can be assigned based on cell values. Color bar and full text legend functions are available. In 3D, isolines for a cut plane in a 3D mesh can be generated. Isosurfaces on a 3D mesh for any scalar value can be generated. Light sources for shading the surface and the color of the surface and the number of graduations of color can be defined. Animations of the surface though a range of values can be done by repeated calls to these functions (or by settings in dialog or animation widgets).

Isovolume (Rating: 1) PHIGURE has voxel visualization functions for voxel volume representation, generalized color tables for rendering voxel volumes, and transparency control.

Geometries (Rating: 2) PHIGURE places all graphical objects and elements into “structures”. Each structure has an identifier. Structures can be stored in centralized structure storage and archived for later use; edited on an element by element basis and plotted as part of an image. Structures can invoke other structures. Transformations can be applied to structures and individual elements. Structures can be copied, rotated, scaled, and translated. Structure visibility and pickability can be defined. Structures provide a great deal of flexibility in picture generation and viewing- at the expense of the implied storage and access. GPHIGS provides options to by pass structures and plot directly on the output workstations-with the disadvantage that picture geomtries may have to be regenerated during interactive sessions.

Geometry Deformations (Rating: -2) Not available.

Animation (time) (Rating: 2) Full animation capability is implied by the nature of PHIGURE-one simply gathers animation control parameters from interactive widgets and loops over the picture generation process if the geometry is changing, or over the viewing parameters for structures or workstations otherwise. Output workstations can be defined for X, PostScript, CGM or raster file.

4.10 Phigure

Animation (space) (Rating: 2) See above.

Multiple problems (Rating: 2) The ability to read one or more than one data set is basically under application control. There are no constraints in PHIGURE/GPHIGS other than screen space and geometry storage.

Thresholds (Rating: 0) PHIGURE provides a function for setting the min and max values for data to be processed for mesh, iso line and iso volume calculations.

Erasure (Rating: 2) Erasure of image parts can be controlled by structure visibility (user “picks” an image part, the application determines the structure id and sets the structure’s visibility attribute), and by controlling the various clipping planes and regions.

Streamlines (Rating: 0) There is no direct streamline function. PHIGURE and PHIGS basic geometry elements could be used to construct streamlines or pathlines. Cubic spline function is available for path calculations.

Pathlines (Rating: 0) See above.

Point Data (Rating: 1) PHIGURE supports 2D and 3D point and vector fields. 2D and 3D point fields can be converted to regular or irregular rectangular mesh, regular or irregular parallelepiped mesh. Appearance and color of vectors can be set. Also, selection and elimination criteria for vectors.

Line Data (Rating: 0) No direct support for line data through a data field. Basic PHIGURE or GPHIGS primitives and widgets could be used to support this capability.

Symmetry (Rating: 0) No direct support. However refer to Geometries section above for basic capabilities.

Input (Rating: 1) PHIGURE supports the following 2D mesh types: rectangular, quadrangular, polar, sector, triangular, non-structured triangular, non-structured quadrangular, and heterogeneous non-structured.

PHIGURE supports the following 3D mesh types: parallelepiped, hexahedral, cylindrical, spherical, non-structured mesh of tetrahedrons, and heterogeneous non-structured.

Heterogeneous non-structured meshes are restricted to a set of 4,5,6 or 8 vertices polygons.

Multi-block is not directly supported but could be done with multiple calls to appropriate mesh routines above. AMR could likewise be supported.

No ASCII file input is supported.

Output (Rating: 1) PHIGURE/GPHIGS support X, MOTIF, PostScript, CGM, SunRaster, OpenGL, PEX output. Any image can be plotted to one or more active workstations by simply sending the structure list to the workstation and requesting a workstation update.

Memory (Rating: 2) PHIGURE has resource file capability for workstation settings. Structure storage and archival can be used to maintain images and picture definitions.

Large Data (Rating: 1) Again, large data sets can be handled by the structure storage facilities. Pictures and picture parts can be saved, retrieved and modified. GPHIGS_GUI is a separate package which can be dynamically linked to provide a GUI for structure manipulation and editing.

4.10 Phigure

Remote (Rating: 2) A PHIGURE/GPHIGS application can run on one platform and display on another via X or Motif. Input can likewise be handled. PHIGURE utilizes double buffering and Z-Buffering to enhance X through put. X and Motif widgets are available to integrate to aid 2D and 3D graphics input and viewing.

MP (Rating: 0) A MPP z buffer extension has been developed to support parallel PHIGURE/GPHIGS applications at LLNL Livermore Computing. Contact Jeff Rowe at LC for information.

Extra Features and Comments: The following information is taken from:

http://www.llnl.gov/asci/ads/G5G_mosaic/Html/Overview_phigure.html

Designed for the development of high-performance, interactive graphics applications, PHIGURE supports the X-Window and PEX graphics systems, workstation graphics accelerators and output devices driven by PostScript, HPGL. . . Based on standard PHIGS calls, PHIGURE simplifies the implementation of graphics applications for anything from the display of 2D curves to the display of complex 3D volumes. PHIGURE packages together the most frequently used combinations of graphics functions for the creation, visualization and manipulation of graphics data. With its standard widgets, PHIGURE is integrated with Motif as well as with OPEN LOOK to assure the ergonomic quality of the application. Applications evolve easily since PHIGURE gives you access to the GPHIGS library, furnished as a standard feature.

- PHIGURE reduces development time for graphics applications Its advanced capabilities allow visualization of numerical data in 2 or 3 dimensions, with only few lines of code.
- PHIGURE possesses both a C and a FORTRAN interface
- PHIGURE accesses a full set of local and networked peripherals The library uses numerous drivers to control local and at a network, terminals and workstations running the X Window System, PostScript printers, HPGL plotters. . .
- PHIGURE improves performance by taking advantage of 3D graphics accelerators Many workstations have graphics accelerator cards and libraries (GL on Silicon Graphics and IBM, XGL on Sun, Starbase on HP). PHIGURE profits by using such accelerators to obtain optimum performance.
- PHIGURE is integrated with the X Window System All the power and ease of use derived from the X-Window multi-window environment is also available through PHIGURE. Visual comfort is also maximized with PHIGURE's use of its double buffer and Z-buffer.
- PHIGURE uses the resources of Motif The integration of PHIGURE and Motif makes a graphics development tool that is powerful and easy to use.
- PHIGURE takes advantage of all the power of GPHIGS GPHIGS is the most powerful and the most open standardized API on the market because it allows access to the basic graphics of the X-Window System, the graphics capabilities of PEX and OpenGL, the printing and plotting power of PostScript, HPGL, CGM. . .
- PHIGURE assures long software life and application portability Applications are free of hardware and development environment constraints thanks to PHIGURE's support of:
 - UNIX, VMS, VM/CMS, UNIX SV, SCO and Windows
 - OSF/Motif, the X Window System, PEX, OpenGL
 - PostScript, HPGL, CGM, Raster RLE. . .

4.10 Phigure

- PHIGURE is delivered with a directory of instructive program examples The time to learn PHIGURE is greatly reduced by providing many C and FORTRAN example applications.
- PHIGURE possesses a utility for animation Subtle changes in data can be brought out with the animation utility, and no extra programming is required.
- PHIGURE is furnished with GPHIGS as a standard feature Every 2D and 3D functionality possessed by PHIGURE is possible through the use of PHIGS primitives, assuring long software life and application portability.

PHIGURE is designed for users who needs to quickly visualize their data and develop graphics rapidly applications.

- Axis system Each axis is characterized by a set of parameterized attributes. PHIGURE also creates legends to clarify data representations.
 - Parameterized axes in 2D or 3D: linear, logarithmic, polar, calendar
 - Any system of 2D or 3D axes
 - Display of curves, annotation, legends. . .
 - Interpolation: linear, step, spline
 - Symbols and character fonts vectorized for improved visual quality
- Generalized 2D and 3D Primitives: Circle, circular arc, ellipse, elliptical arc, sphere, cylinder, parallelogram, cube. . .
- Representation of Statistical Data: 2D and 3D pie charts, histograms, stock market (high-low-close) graphs
- Treatment of Surfaces and Volumes
 - PHIGURE handles data organized as:
 - * 2D rectangular, quadrangular, triangular, polar meshes
 - * 3D hexahedral, cylindrical, spherical, parallelepipedal meshes
 - * 2D fields and 3D clusters of points
 - PHIGURE integrates highly-evolved functions for:
 - * Isolines on a 2D mesh (wire-frame or continuous surface)
 - * Isolines on 3D mesh cut-planes (wire-frame or continuous surface)
 - * Annotation of isovalues on 2D and 3D isolines
 - * Smoothing of 2D and 3D isolines
 - * Wire-frame isosurfaces for 3D volumes
 - * Shaded isosurfaces (flat and Gouraud shading) for 3D volumes
 - * Calculation of isolines on 2D or 3D mesh elements
 - * Representation of 2D and 3D vector fields
 - * Display of functions of the form $z(t) = f(x,y)$
 - * Surface coloring as a function of scalar values (3D, 4D)
 - * Isovalue legends
- Utilities: The utilities enrich the quality of graphics data and manipulate graphic representations by providing:
 - Legends, text boxes, vector arrows

4.11 Spectrum Visualizer

- Matrix images (n*m pixels)
 - Graphic object clipping along one or more planes
 - Dynamic invisibility and detectability filters
 - File generation (PostScript, HPGL, CGM, Raster RLE)
 - Scene visualization and manipulation (simplified definition of 2D and 3D views)
 - Dynamic color table definition (monochrome, RGB, HLS. . .)
 - Animation sequences in Raster RLE format under X Windows
- **Application Directory** In addition to the PHIGURE library, PHIGURE is furnished with a set of documented demonstration programs. Each demonstration program illustrates the principle PHIGURE functionalities and can even serve as a template for numerous graphics applications.
 - **Systems** PHIGURE is available on a great number of platforms currently available on the market: Sun Microsystems, Hewlett-Packard, Silicon Graphics, IBM, DEC, CRAY, CONVEX, 88 Open, MIPS, APPLE. . . and on PCs (UNIX SVR4.2, SCO, WINDOWS).
 - **Standard Features** PHIGURE is furnished with an object code library including C and FORTRAN 77 interfaces, the GPHIGS library, the GPHIGS Widget and the X11, PEX, 3D-specific (GL, XGL, STARBASE, depending on hardware), PostScript, HPGL, CGM and Raster RLE drivers.
 - **Support and Assistance** G5G assures a high-quality customer assistance service, notably through our Telephone Support Department and e-mail access.
 - **Open Software:** Delivered with GPHIGS, PHIGURE is an open software product — for adapting and creating functionalities specific to an application or a field of activity.
 - **Productivity:** Conceived as a productivity tool, PHIGURE is an easy to use library for implementing and working with local and network resources such as X terminals, printers, plotters, 3D workstations . . . A 3D display program of only a few lines of code written with PHIGURE calls represents one hundred lines of code using PHIGS calls, or a thousand or more lines of programming with X-Window calls.
 - **Integration under Motif:** To develop applications that are modern and ergonomic, Motif is well adapted to the management of user-machine interfaces, as PHIGURE is to graphic visualization. PHIGURE uses a widget to easily manage 2D and 3D graphics in Motif windows. With this widget, PHIGURE integrates itself easily with the principle Interface Builders on the market.

4.11 Spectrum Visualizer

Code Description: A very powerful, flexible, and friendly tool for post-processing. The Visualizer displayed a 112,000-element data set (the largest David Gardner had access to at the time) very quickly, including rotations and translations and manipulations for slicing, contouring, etc.

The vendor reports that the largest he has handled as been a 2,000,000-brick-element mesh, and that some people at Ford are doing about 1,600,000 tet meshes)

Hardware Platforms: Sun, HP, SGI, IBM, Xterms

Licenses: Commercial product, from Centric.

Contacts:

4.11 Spectrum Visualizer

- Steve Attaway, swattaw@sandia.gov.
- David Parker, Centric's manager of development for scientific visualization.

E-Contact: <http://www.centric.com>

Disclaimer: The recommendations in this document are made with specific consideration of ASCII requirements and are not to be construed as a general endorsement of any specific company or product by the University of California, the U.S. government, the Department of Energy, nor any of their employees.

Evaluator: Steve Attaway, SNL, swattaw@sandia.gov

Evaluation Hardware Environment:

Support (Rating: 2) Fully supported. There is a bug tracking system.

Documentation (Rating: 1) There is a printed users manual. An html version is in the works. On-line help is not implemented in version 2.0 but will be in version 2.1. There are simple example files provided for the tutorials.

Turnkey (Rating: 2) Can access any of the features as end-user system. Can also build custom end-user applications (e.g. BLOT look-alike). David Parker adds that although it has never been done, new visualization techniques (and readers) could be added to the visualizer by users.

Simple (Rating: 2) Easy to learn and use. Well organized. Very flexible.

Variable Types (Rating: 2) Can visualize scalar, vector, tensor and general array data, both cell-based and node-based. Can visualize components of vector and tensor data. Null data was not yet evaluated.

Axis (Rating: 0) Axis is fixed to upper-right window corner.

Query (Rating: 2) Have both node query and element query functions. Can show element geometry at the query point. Can also query for max/min of any data values.

Scale (Rating: 2) Have a ruler visualization technique which can do such measurements.

Annotation (Rating: 2) Can add title and simple annotation. Can't be attached to data.

Algebra (Rating: 2) Have symbolic calculator which can compute on given data. Equations can be pre-defined or constructed interactively.

Viewing (Rating: 2) Have 3d rotations/translations and scale. Scale/rotate about a point. Zoom widow. Perspective and parallel projection. Can move eye point. Multiple viewports. Manipulate light position and color; multiple light sources.

Cut planes (Rating: 2) Cut planes for both vector and scalar fields in arbitrary orientation. Map scalar onto vector cut plane. Can adjust color mapping. Any number of cut planes can be defined. A legend is provided both on the vis panel and in the graphics window. Scalar cut plane can be displayed as contours, isolines or color shaded. Cut plane be animated. Vector cut planes can project vectors to plane, color by vector magnitude, color by direction (if vector points away from plane) and show the surface created by the vectors on the slice.

Can also extract a mesh slice or slice the surface mesh.

Support for the definition of graphics clip planes which can selectively remove geometry.

4.11 Spectrum Visualizer

Isosurfaces (Rating: 2) Can isosurface any nodal value. Can map scalar or vector field. A legend is provided both on the vis panel and in the graphics window. Can be colored by isosurface value or changed interactively. Animation is supported. Any number of isosurfaces can be displayed. Can display variation of second scalar as smooth shaded, contour or isolines.

Isovolume (Rating: ?) Not yet evaluated.

Geometries (Rating: 2) Attributes for geometrical objects (color, transparency, rendering style, visibility, etc.) can be modified interactively. Geometrical objects can be copied and reflected about any angle any number of times.

Geometrical objects such as surfaces (side sets) or mesh interfaces can be contoured.

Geometry Deformations (Rating: 2) All visualization techniques can operate on deformed meshes. The original (undeformed) mesh can be displayed simultaneously. Attributes are retained for each new time state.

There is also support for mode shape animation.

Animation (time) (Rating: 2) This code can animate any geometrical object that changes in time. Frame number is selectable. Can manipulate orientation of geometry during animation.

Animation (space) (Rating: 2) Can animation any geometrical object that changes in orientation. Can select center of rotation and scale. Space and time animations can be combined.

Multiple problems (Rating: 2) Unlimited number of problems can be read in. Problems can be displayed in the same or different graphics window.

Thresholds (Rating: 2) Thresholding is done on a cell basis. A subset of the mesh is Computed using a range of values. Any data type can be used to threshold. Thresholding is not implemented on a visualization technique basis.

Erasure (Rating: ?) Not yet evaluated.

Streamlines (Rating: 2) Can construct and display streamlines but not streamtubes. Supports steady and unsteady particle animation. Can manipulate geometry while particles are animating.

Pathlines (Rating: 2) Does not display pathlines now but will be implemented soon. This is just part of unsteady particle animation. Supports steady and unsteady particle animation.

Point Data (Rating: 1) Can graph a set of selected points. Experimental data can be read in and displayed as spheres or points. It will soon be possible to directly compare experimental data with computed results by interpolated the mesh at the experimental data points and then displaying the results as a graph. Only x-y and histogram plots are supported now. It would not be hard to add others.

Line Data (Rating: 2) All of this is supported.

Symmetry (Rating: 2) Can reflect any geometry about an axis.

Input (Rating: 1) Currently unstructured and structured meshes are supported. New mesh types can be added in a straight forward manner. Unstructured mesh supports all element types including higher-order elements and spheres.

Can accept point data.

Tool Evaluation Resources

New readers can be added.

Has a reader for a simple ascii file format. Can also read unix compressed ascii files.

Output (Rating: 2) Animation frames can be written out in several formats. Can be upgraded for new formats.

Memory (Rating: 2) A scripting feature is available. Commands can also be specified in a start-up config file. Scripts can be read in at any time.

Large Data (Rating: 2) Can handle large data sets. Largest mesh visualized to date was a 2 million element brick mesh. Can select regions or element sets to visualize. Can subset the mesh using thresholding or a geometric constraint. Data is only read-in when it is needed. User preferences are currently being expanded to include memory.

Remote (Rating: 1) Can do visualization number crunching on one machine and display using X on another. The vendor adds that true distributed processing is not here yet but will be in the future, as it is waiting for funding.

MP (Rating: 0) The vendor comments that currently a multi-processor version is not available. It is in the works for a version which runs on distributed and shared memory multi-processors, and is also waiting for funding. He thinks it would be it would be straight-forward to do this.

Extra Features and Comments:

A Tool Evaluation Resources

A collection of documents useful for comparing evaluation tools:

- *Comparison of Visualization Techniques and Packages*, Sander Belein and Rik Leenders, SARA Visualization Center, the Netherlands. This is version 2.0, updated from an earlier release last year. It compares Data Visualizer 3.0, IRIS Explorer 3.0, Application Visualization System 5.0, and Visualization Data Explorer 2.1. It is available at <http://www.sara.nl/Rik/REPORT.update>
- *Review of Visualization Systems*, K.W. Brodlie et. al, the Advisory Group on Computer Graphics, UK. This report compares Application Visualization System, IBM Data Explorer, IRIS Explorer, Khoros, and PV-WAVE. It is available at http://www.agocg.ac.uk:8080/agocg/PrevPj/VisSyst/dogbook_1.html
- *From Imagery to Interactivity — Evaluation Guidelines and Concepts*, Arthurine Breckenridge, Sandia National Laboratories, New Mexico. This document provides a comprehensive list of dimensions and issues against which to evaluate any given visualization tool. For a copy, contact Arthurine at arbreck@sandia.gov.

B Related Tools

This section consists of codes that were considered early in the evaluation process, but did not receive a full evaluation. This was either because the code is a useful adjunct to a viz tool without being one itself

Related Tools

(DDI and PDBView), or because the tool clearly supported too few features and/or architectures to be a candidate for ASCI.

AMRVIS — Adaptive Mesh Refinement Visualization

Code Description: This is a viewing tool for adaptive data sets. It provides raster and numerical views of the data, animations in time, interactive subregions, planar slicing and animation (in 3D), and interactive volumetric rendering. It handles 2D and 3D adaptive mesh data. It also handles mixed material zones and mixed species materials.

Hardware Platforms: SGI, IBM, Sun, and Cray (in the future: HP, DEC)

Licenses: Easily available to LLNL personnel. Requires Memorandum of Understanding for release to other government agencies (this should be easy). For volumetric rendering AMRVIS uses an external library — if someone wants to try this code we will check out whether it is OK for us to release it.

Contact: Vince Beckner 510-423-1377 beckner2@llnl.gov

E-Contact: None

Bob — AHPCRC Graphics Software

Code Description: Bob is an interactive volume renderer. It provides very good performance when run on a system that has high polygon performance such as a Silicon Graphics VGX workstation. The input to Bob is a 3D array (or brick) of bytes, thus the name “Bob”.

Bob is a tool for browsing three dimensional data sets. The program was designed to quickly render portions of potentially large time dependent volumes. Bob can also create a series of images that can be later compiled into a movie.

Hardware Platforms: Most anything that supports OpenGL, but “Machines without z-buffers are limited to alpha rendering. Machines without blending hardware, such as the Personal Iris, are limited to maximum value rendering.”

Licenses: Publically and freely available.

Contact: Ken Chin-Purcell, Minnesota Supercomputer Center Inc., gvlware@ahpcrc.umn.edu

E-Contact: <http://www.arc.umn.edu/html/gvl-software/bob.html>

DDI — Data Dimensions Interface

Code Description: This tool transfers rectilinear mesh data between different files, different formats (DRS, HDF, NetCDF, Dyna/Taurus, AVS, IRIS Explorer) and different visualization systems (AVS, Collage, Ximage, IDL, IRIS Explorer, and PV-Wave). It handles multidimensional uniform, rectilinear and curvilinear mesh data.

Hardware Platforms: SGI, HP, IBM, Sun, DEC OSF, Cray

Related Tools

Licenses: None needed (Public Domain)

Contact: Chris Anderson 510-423-1915 andersoncl@nersc.gov

E-Contact:

- anonymous ftp: [ftp.nersc.gov](ftp://ftp.nersc.gov) in /pub/DDI.
- <http://www.nersc.gov/doc/Services/Applications/Graphics/DDI/DDI.html>

NAG Explorer

Code Description: Supplemented by several algorithms developed at LLNL for visualizing 3D vector fields and volume rendering for unstructured grids. These include flow volumes, textured splats and line bundles. The goal of this research is to explore methods of representing 3D vector fields and multiple data fields. These tools handle 3D rectilinear and unstructured mesh data and data stored in trees. It handles multi-block data.

Hardware Platforms: SGI (in the future: HP, IBM, Sun, DEC)

Licenses: Explorer must be purchased from NAG

Contacts: None.

E-Contact:

- <http://www.nag.co.uk:80/>
- <http://www.llnl.gov/graphics/sciviz.html>
- For a non-ASCII evaluation of NAG Explorer, see http://sgrabber.cern.ch/~lecointe/IE_evaluation.html

FAST

Code Description: FAST is a visualization system designed for structured data bases created from Computational Fluid Dynamics analysis. The tool uses GL graphics features and is designed for SGI platforms. This system was designed for the Plot3d program and requires a Plot3d data base.

Hardware Platforms: SGI, IBM, DEC, Kubota, HP, Sun.

Licenses: Commercial product from Sterling Software, (415) 964-9900.

Contacts: Verlan Gabrielson, (510) 294-2184, gabe@ca.sandia.gov

E-Contact: ???

GRIZ

Code Description: This tool supports visualization of transient finite element analyses on 2D and 3D unstructured grids. It reads output files from the heavily used DYNA and NIKE codes.

Hardware Platforms: SGI, IBM, Sun

Licenses: Users must sign a license agreement with LLNL's Methods Development Group.

Contact: Doug Speck 510-423-4208 despeck@llnl.gov

E-Contact: None.

PDBView — Portable DataBase Viewer

Code Description: This is a binary data access tool which supports printing, plotting, editing and querying of data in a variety of formats. It is user extensible by virtue of having an embedded SCHEME interpreter. It is part of the PACT tool set. PDBView handles 1D, 2D and 3D rectilinear, curvilinear, adaptive and unstructured mesh data, tables, general arrays, and arrays of rank greater than 3. It also handles multi-block meshes.

Hardware Platforms: SGI, HP, IBM, Sun, DEC, Cray, Meiko, Macintosh

Licenses: None needed

Contact:

- Stewart Brown 510-423-4889 sabrown@llnl.gov
- Dennis Braddy 510-423-4129 braddy@llnl.gov

E-Contact:

- anonymous ftp: [coral.llnl.gov:/pub/pact](ftp://coral.llnl.gov/pub/pact)
- http://www.llnl.gov/def_sci/pact/pact_homepage.html

VCS — Visualization and Computation System

Code Description: This tool provides basic capabilities for validating, comparing and diagnosing climate modeling behavior. It handles 1D, 2D, and 3D rectilinear mesh data.

Hardware Platforms: SGI, HP, IBM, Sun, DEC (in the future: IBM, DEC, Cray)

Licenses: Standard LLNL collaboration agreement

Contact: Dean Williams 510-423-0145 williams13@llnl.gov

E-Contact: None

Distribution

1	9003	D.L. Crawford, 8900
1	9004	M. E. John, 8100
1	9214	L.M. Napolitano, 8117
5	9214	W.P. Kegelmeyer, Jr., 8117
10	9021	Technical Communications Department, 8815, for OSTI
1	9021	Technical Communications Department, 8815/Technical Library, MS 0899, 4414
4	0899	Technical Library, 4414
3	9018	Central Technical Files, 8940-2