MAVIS: an interactive visualization tool for Computational Chemistry calculations in a distributed networked environment.

C.I. Parkinson[a]
Department of Chemistry, University of Manchester, Brunswick Street, Manchester, M13 9PL, UK

M.D. Cooper*, W.T. Hewitt
Manchester Visualization Centre, Manchester Computing, Oxford Road, University of Manchester, Manchester, M13 9PL, UK

I.H. Hillier
Department of Chemistry, University of Manchester, Brunswick Street, Manchester, M13 9PL, UK

This paper describes the software package, MAVIS (Molecular Animation and VIsualization System), which was developed as a collaborative project between the University of Manchester Department of Chemistry and the Manchester Visualization Centre as a tool for the visualization of chemical systems and their properties and for the generation, submission, monitoring and study of results from calculations in a networked computing environment.
MAVIS has been developed using AVS, a general visualization environment, as a suite of modules which can be extended with relative ease. Using the AVS system gives the application the power and flexibility of a general system with no requirement for the programmer to understand many of the complexities involved in modern image rendering systems and allowing them to concentrate upon the features of the functionality and the user interface. It also allows the code to be as portable between platforms as the AVS package itself and so it is available on a wide range of systems.

1 Introduction.

MAVIS was designed with two specific goals in mind which were not already served by the available molecular visualization packages. The primary purpose was to develop a package which could interface quickly and easily with the set of application packages available and in use in Manchester's Computational Chemistry groups to allow users to build molecular structures, create job decks, run calculations and view the calculation results with ease. The set of applications included initially was MM2[12], Amber 4.0[9], MOPAC[8] and Gaussian 92[6].

The secondary goal, which is becoming increasingly important with the wide diversity of computer hardware available to the users across the univer-

[a] Current: Battelle Pacific Northwest National Laboratories, Richland, WA 99352, USA

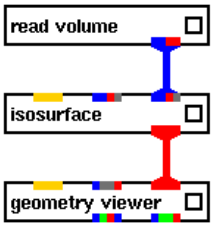

Figure 1: A small AVS network rendering an isosurface from volume data.

sity's campus network, was to make a range of computers available to the user from their own desktop and enable them to use those machines to perform calculations without the need for the user to manually transfer files between machines and remotely login to run each calculation. The environment in Manchester is that of a large number of workstations across the University Campus, mostly Hewlett-Packard 9000-7xx but some Silicon Graphics machines, as well as number of larger compute servers such as Silicon Graphics 'Power Challenge' and 'Origin' systems.

1.1 Advantages of AVS.

AVS, the Application Visualization System[2], is a scientific visualization package developed to provide a general, modular and highly extensible visualization resource. Applications can be assembled from a large number of standard modules by selecting the desired set and defining the interconnections using a graphical interface called the 'Network Editor'. The interconnections can be viewed as conduits through which data flows from one module to the next with each module executing each time it receives new data, processing the input data and, perhaps, writing output which will flow to the next module. This 'dataflow' model is central to the operation of AVS and controls the execution of modules. A simple example of a small network is shown in figure 1. In this example network data is read from a file by the *Read Volume* module and flows to the *Isosurface* module which then executes to compute an isosurface. *Isosurface* produces a sequence of AVS geometry primitives which are passed to *Geometry Viewer* which then executes to process these data and render the final image.

Where modules suitable to a particular purpose are not available, a well-

defined API is provided to allow the user to add new modules of their own. AVS is used to define the new module's interconnection ports and then user-supplied C, C++ or Fortran code is added to process the data. In this way new applications can be constructed with user code providing application-specific features and AVS providing image rendering and user interface support as well as providing modules to implement many standard visualization methods.

The advantage of using AVS over one of the commonly-used graphics support libraries is the removal of the need for a user to write programs in order to be able to create custom applications. The user is presented with the suite of modules which make up the MAVIS module library and a standard network which defines the MAVIS application. The MAVIS application can then be used as given or the user can modify the network, removing modules or adding modules from the MAVIS library, the standard AVS module set or modules obtained from other sources, in order to construct an application uniquely suited to the work which they wish to undertake.

1.2 MAVIS features.

The set of modules, some 26 in total, included in the MAVIS library provide a range of facilities to the user. Molecules can be constructed from single atoms, fragments or residues via a graphical user interface. Structures can also be loaded from files with translations from a number of standard formats including PDB, XMol's 'XYZ' format and MacroModel files. Other modules allow the loaded structures to be displayed in a range of styles from simple wire-frame or 'ball-and-stick' for bonds and atoms to 'ribbon' displays showing the 'backbone' of residue-based structures. Still more modules allow the displayed structure to be manipulated, edited, measured and for these structures to be used in the performance of calculations, either locally or on one of a set of remote machines, interactively or in batch mode, making use of a set of application codes of which MAVIS is aware.

Remote access is managed through the use of a user configuration file, *$HOME/.mavis_startup*, which includes definitions of a set of computers which are available to the MAVIS package. The definitions for each machine include the machine name, an account to be used on that machine and paths for the user's home directory and to the application packages available on that machine. Access to each machine is made possible through the facility for remote logins from a trusted account without authentication made possible by the '.rhosts' and 'hosts.equiv' facility of the remote shell daemon on standard Unix systems.

1.3 Comparison with other Molecular visualization systems.

Over the last few years a number of software packages have become available which provide facilities comparable with those of MAVIS. One of the most well-known is the Cerius^2 package from Molecular Simulations Inc. which can be taken as a good example of the state of the art of commercial molecular visualization software. Cerius^2 provides a wide range of facilities including the ability to build structures and submit calculations. In combination with the Cerius^2 Software Developers Kit, the user can incorporate their own code within the Cerius^2 visualization environment.

The main advantages which MAVIS has over such packages, apart simply from the low cost, is that the use of AVS makes the package available on a large range of machines with little extra effort on the part of the programmer and that the programmer has access to all of the facilities of a fully-fledged general visualization package.

2 Implementation details.

This section describes, in general terms, features of MAVIS which are fundamental to its operation. An application of this size will require far greater space to describe in detail so for further information the reader is referred to the MAVIS programmer's guide[4] and module guide[5].

2.1 User Interface.

Experienced users of the AVS package will be familiar with its layout of control and network editor panels but the MAVIS application, when run in stand-alone mode, ensures that these features of AVS are not visible. The AVS controls can still be accessed by selecting certain options to raise the AVS control panel which has been deliberately hidden but, in stand-alone mode, the MAVIS user interface consists of a three-window layout as shown in figure 2. The menu bar along the top of the display allows the user to select functions to perform such as to make the application load and display a molecular structure or to select the functions of a particular module such as *Build Molecule* or *Measure Molecule*.

The large panel on the left is used to present options associated with the currently selected function with each module using this portion of the interface when it is invoked from the menu bar. In the case shown it contains options for the creation of a Gaussian calculation as presented by the *Calculate G92* module but it is also used, for example, to present the file browser when the *Open Molecule* function is selected and to display the atom and bond types

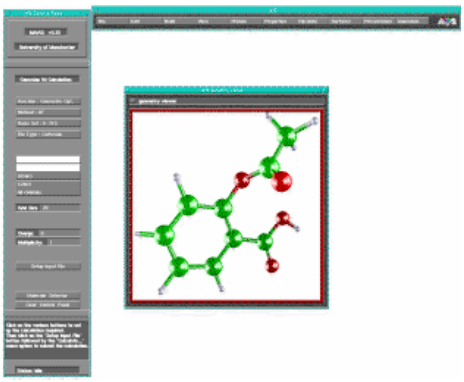

Figure 2: The MAVIS application user interface.

which may be selected when the *Build Molecule* or *Edit Molecule* functions have been invoked.

The central window is the AVS geometry viewer window and is created and controlled by the *Geometry Viewer* module which is present in the MAVIS network and performs all of the final rendering of images for display.

The MAVIS application can, of course, be simply loaded into the AVS network editor and manipulated in the same way as all AVS networks. When loaded into AVS in this way the network is as shown in figure 3.

2.2 Remote file access.

Access to files on remote systems has been made possible by the provision of a remote file browser which closely resembles the standard AVS file browser except that it supports an additional control to allow the user to select from the predefined list of machines read from the user's configuration file. Selecting a particular machine causes the browser to list the files appropriate to the current directory on that machine and the user can navigate the directory structure in the same manner for remote systems as for the local system.

When a remote file is selected to be read then it is copied locally and it is the local filepath which is returned to the requesting module which can then read this file. When used to create a remote file the remote browser takes a local pathname from the requesting module which has already written the file locally and copies the file to the selected remote location.

2.3 Internal file structure for inter-module data passing.

It was determined quite early in the project[3] that the AVS 'dataflow' model, described in section 1.1 above, would not provide sufficient flexibility for the project in hand and so an internal data representation was derived which would provide the required features. This file structure resides in some user-defined location, usually in temporary space, and each module accesses the file structure through a well-defined set of routines. Any modules which the user adds to the system can make use of the same routines to read and modify the MAVIS data as appropriate. The data files hold all global information which must be available to all the modules and holds geometries and associated information for the current set of molecules loaded into the system. While this temporary storage is usually cleared at the end of a session the user can request that MAVIS stores the current set of data in a 'scene' file. These data can then be restored later to restart a session.

2.4 Inter-module messaging.

AVS 'dataflow' is used in MAVIS to control the execution of modules through the transmission of messages through the AVS network. These messages, consisting of a command and qualifier pair, inform modules, according to the sequence defined by the MAVIS network's interconnections, of the actions which have been performed by their predecessors. This allows each module to execute as and when required and to ignore messages, simply passing them on to the next module, which do not affect them directly. The messaging protocol is, again, supported by a well-defined set of routines allowing the user to create their own modules using these features.

2.5 Display of properties data.

Beyond the simple rendering of molecular structure data as ball-and-stick diagrams, a molecular visualization package must be able to present other data associated with the structure. Many forms of data can be evaluated by the calculation packages with which MAVIS interfaces directly. Within the AVS model it is simply necessary to provide a module to extract these data from the output of the calculation package and transform them to a suitable format in order to process and display them using standard AVS modules. Examples implemented in the MAVIS application include the visualization of electron density, electrostatic potential and molecular orbital data associated with a system, as calculated by the Gaussian[6,7] packages. These volume data can be rendered in the form of contour plots on a user-specified cutting plane through

the volume or as surfaces evaluated by use of the standard AVS module *Isosurface* which draws a surface passing through points in the volume of constant value. The evaluated contour plots and surfaces can be displayed alone or superimposed over the molecular structure using opaque or transparent display of the surface data.

It is also possible to add more information to the display by the use of a further volume data set to provide colour information to the display. In this way, for example, the electron density isosurface can be coloured according to the electrostatic potential to include further information in the surface and structure image. This is also implemented in the MAVIS application.

2.6 Rendering of images.

AVS5 includes a module, known as the *Geometry Viewer*, which creates the display window within the MAVIS user interface and which renders all of the images produced by the application. This module accepts, as input, AVS geometry data which is a description of the image as a set of 'primitives' such as meshes, polyhedra, polytriangle and sphere objects. This geometry data can be generated by any of a number of AVS-supplied modules or can be produced by the user's own modules. These data, when passed to the geometry viewer, cause it to render an image in the display window which the user can then manipulate in three dimensions to rotate and translate the view. The geometry viewer also permits the user to interact with the displayed image by returning 'pick' information when the user selects the display with the mouse buttons. This feature is exploited by the MAVIS modules to permit the user to select specific atoms in the display. This is used by such modules as those associated with the building and editing of molecular systems and the performance of measurements of bond-lengths and angles.

The current transformation applied to a displayed object can also be read from the geometry viewer allowing the transformed view in the display to be used when generating output graphics files to ensure that the output images match those currently shown in the display window.

A module, *Draw_Molecule*, is provided in the MAVIS library to produce the geometry data associated with the molecular structures. Other modules which may be usefully included may also generate geometry data which must, again, be passed to the *Geometry Viewer*. As described in section 2.5, one such module included in the standard MAVIS network is the *Isosurface* module which is used to view volume data produced by some of the application codes. Molecular visualization of these properties presents an unusual problem in AVS in that the geometry data produced to describe the surface are not connected

with those describing the molecular structure. This has the unfortunate side-effect of allowing the user to select and manipulate the separate parts of the display and so rotate or translate the surface independently from the molecular structure. Obviously this surface data should be anchored closely to the atoms and bonds. This problem has been overcome through the inclusion of a module, *Geometry Combine*[10], which accepts multiple streams of geometry data and combines them to form a single geometry object which is then rendered. This combined object, including both surface and structure, cannot be separated and so can be freely moved on the screen without confusion.

2.7 Animated molecular displays.

Animated displays have a number of uses within molecular visualization and can be achieved within AVS with relative ease. In MAVIS this is implemented through the inclusion of a module called *Animate Molecule*. This module is able to read generic animation files or the sequences of geometries present in the output deck from certain types of MOPAC and Gaussian calculations such as geometry optimizations. *Animate Molecule* is also capable of reading vibrational frequency description data as produced by Gaussian and can produce animated displays for the vibrational modes of a molecular structure.

The data are read by *Animate Molecule* and an AVS geometry for the current frame is produced. By manipulation of the module's controls the user can specify which frame in the sequence will be displayed or allow the sequence to be played continuously. The smoothness of the animated display is dependent upon many factors such as the speed of the user's graphics hardware but, obviously, is also strongly affected by the size of steps taken during the calculation procedure producing the structure data.

2.8 Implementation of application codes interface.

There are a number of molecular visualization tools available for use with application packages. Some calculation packages are supplied with a graphical user interface which provides an extremely easy way to manipulate the calculation package by allowing the user to construct molecular systems, set options, submit calculations and view their results. Most of these packages are, however, closely coupled to the single calculation package in question and few provide any support for batch job submission in a general networked computing environment.

MAVIS was designed to be usable with a number of packages and to enable the addition of new packages with relative ease. The ability to submit calculations both interactively and through the use of a batch queueing system

was also of importance. It was also felt that tying the system to any single batch system or attempting to closely couple the code to the application packages would seriously limit it's usability. To this end MAVIS was developed to interact with the calculation packages and with the batch queueing system through as simple an interface as possible.

Execution of external calculations.

The execution of all external calculation packages from MAVIS is assumed to be performed using a text input deck. The procedure by which a calculation is initiated is through the generation of this data/control deck by a module. The MAVIS user, having generated or imported the structure which they wish to work with, defines the calculation to be performed by the selection of a particular package from the range of which MAVIS is aware. A package-specific options panel is then displayed and the user defines the calculation by checking a series of options in the set presented. The final stage in this procedure involves the writing of the calculation deck to temporary space.

It should be made clear that, since the number of options and their combinations offered by computational chemistry packages is generally extremely large, the MAVIS interface does not attempt to include all options for all packages since this would render the interface so complex and unwieldy that the user would find it less profitable than hand-editing the deck. To this end the user has the option of writing the deck to user-space for later editing prior to manual submission.

The addition of a new calculation package is achieved by the creation of a new module to produce its input deck. A suitable options panel is developed and a program to write the deck appropriate to the options selected must then be added to the module. Reconfiguration of the user interface then permits the user to select this new module in order to raise its options panel prior to creation of the calculation deck.

Running calculations.

Having created a suitable calculation deck the user is able to submit the calculation to any of the number of computer systems which are specified in the user's configuration file. Where interactive, remote submission is used, MAVIS copies the data decks to the remote system and executes the appropriate command on the remote system to perform the calculation. A window is opened on the user's display to monitor the execution. Upon completion the output data decks are retrieved and the results are loaded into the MAVIS display. Thus

simple minimisations, for example, using the MM2 package can be carried out interactively prior to submission of a longer calculation to a batching system.

Where batch submission is used, MAVIS will copy the data decks to the remote machine and submit the calculation according to the selected queue information. In this instance, of course, it does not wait for completion but immediately returns control to the user. The user can monitor execution of the batched calculation from within MAVIS by simply using the 'queue list' option provided and can monitor the progress, for example of a running geometry optimization calculation, by loading the output deck of the partially completed job and examining the sequence of geometries it contains.

3 Conclusions and planned future work.

MAVIS is a useful tool for Computational Chemistry work and has been taken up by a number of users both within the United Kingdom and in continental Europe. It allows the novice user to construct and perform simple calculations with little difficulty and provides the experienced user with a comprehensive interface to calculation packages.

The AVS package, within which MAVIS is constructed, provides a flexible and usable environment for development of interactive applications which run across many platforms. MAVIS makes use of these features by allowing the user to develop new modules and add them to the standard MAVIS application network to make new features and new external software packages available.

The user interface provided in AVS5 has some drawbacks in that coding with it can be an exacting procedure with little development support provided but, with care, excellent results can be obtained. It is hoped that the port to AVS/Express will allow a more flexible user interface to be developed which will allow the user to add controls and modify or even replace the interface to suit their own tasks and style of work using only the AVS/Express GUI and without recourse to modification of the actual source code.

Current work on the MAVIS project includes the rewriting of the application for the AVS/Express system which is currently replacing AVS5 and the addition of new features for graphics output including VRML support and support for improved protein displays through the provision of an interface to the package Molscript[11]

Acknowledgements.

Initial funding for the development of the AVS5-based version of MAVIS was provided by Fujitsu Systems (Europe) Ltd.

Ongoing support is being provided by the U.K. funding body, the Joint Information Systems Committee, under the Technology Applications Programme funded project, 'VISUAL'[1].

Appendix

1. http://www.man.ac.uk/MVC/research/val.html
2. "AVS", Advanced Visual Systems Inc., 300 Fifth Avenue, Waltham, MA. 02154, U.S.A. http://www.avs.com/
3. "The Application of Visualization Techniques to Molecular Modelling", C. Parkinson, PhD thesis submitted to the Victoria University of Manchester, Oxford Road, Manchester, UK.
4. "MAVIS Module Guide", http://www.man.ac.uk/MVC/MAVIS/doc/Module_Guide.ps.gz
5. "MAVIS programmers guide", http://www.man.ac.uk/MVC/MAVIS/doc/Programmers_Guide.ps.gz
6. "Gaussian 92", M.J. Frisch, G.W. Trucks, M. Head-Gordon, P.M.W. Gill, M.W. Wong, J.B. Foresman, B.G. Johnson, H.B. Schlegel, M.A. Robb, E.S. Replogle, R. Gomperts, J.L. Andrés, K. Raghavachari, J.S. Binkley, C. Gonzalez, R.L. Martin, D.J. Fox, D.J. Defrees, J. Baker, J.J.P. Stewart, J.A. Pople, Gaussian Inc., Pittsburgh., PA, 1992.
7. "Gaussian 94", M.J. Frisch, G.W. Trucks, H.B. Schlegel, P.M.W. Gill, B.G. Johnson, M.A. Robb, J.R. Cheeseman, T.A. Keith, G.A. Peterson, J.A. Montgomery, K. Raghavachari, M.A. Al-Laham, V.G. Zakrzewski, J.V. Ortiz, J.B. Foresman, J. Cioslowski, B.B. Stefanov, A. Nanayakkara, M. Challacombe, C.Y. Peng, P.Y. Ayala, W. Chen, M.W. Wong, J.L. Andrés, E.S. Replogle, R. Gomperts, R.L. Martin, D.J. Fox, J.S. Binkley, D.J. Defrees, J. Baker, J.J.P. Stewart, M. Head-Gordon, C. Gonzalez, J.A. Pople, Gaussian Inc., Pittsburgh., PA, 1995.
8. "MOPAC 93", J.J.P. Stewart, Fujitsu Ltd, Japan.
9. "Amber 4", P. Kollmann et al, University of California at San Francisco. D. Case, Scripps Institute.
10. The *Geometry Combine* module was based upon work described in: "AVS Technote: Interpreting GEOM Information", J. Vroom, AVS 93 Conference proceedings, 1993.
11. "MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures", P.J. Kraulis, J. App. Crys. **24**, 946-950 (1991)
12. N.L. Allinger J. Amer. Chem. Soc. **99**, 8127 (1977).

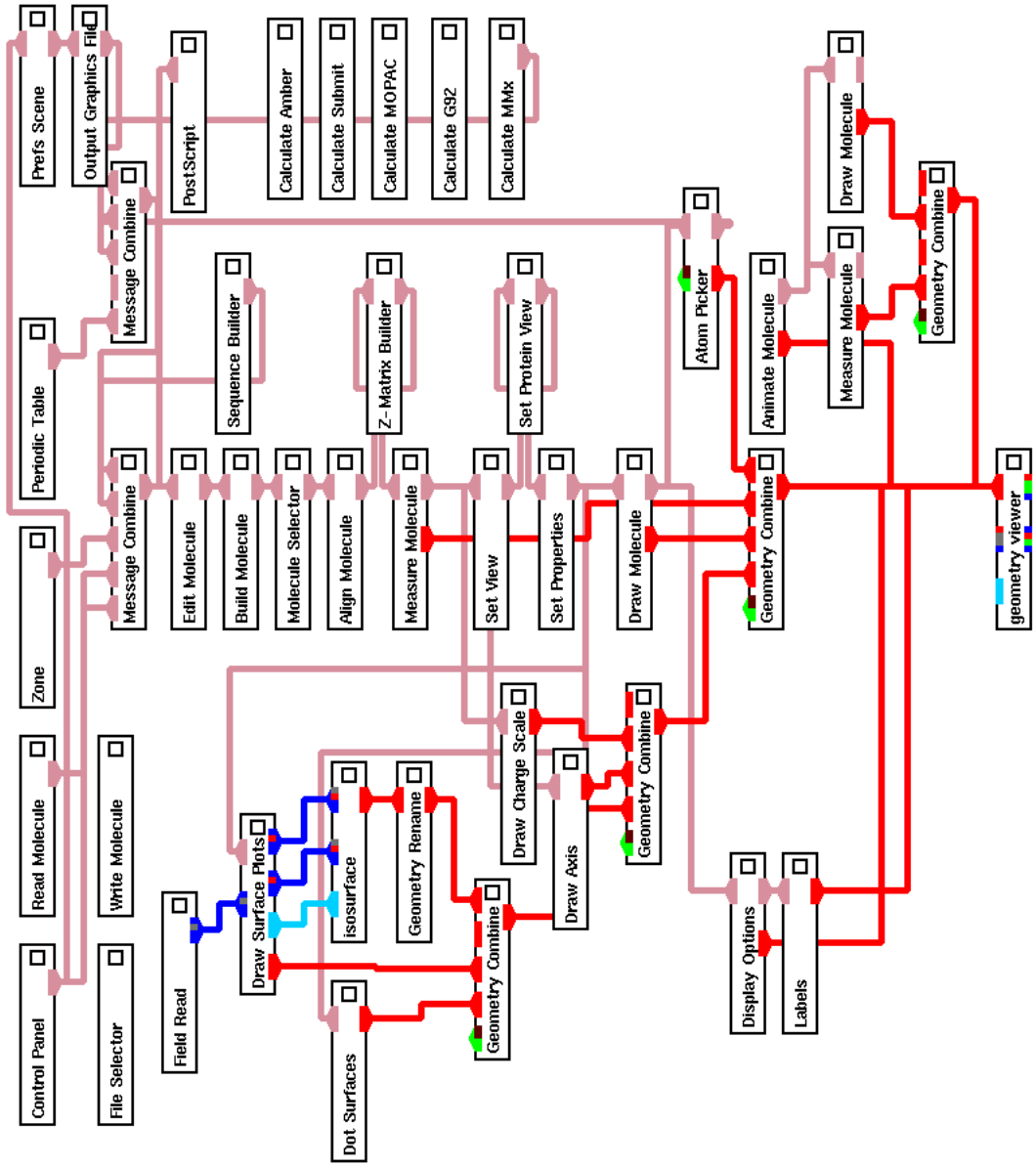

Figure 3: The full MAVIS network.