MacMolPlt: A graphical user interface for GAMESS

Brett M. Bode and Mark S. Gordon

Department of Chemistry, Iowa State University, Ames, Iowa, USA

A description of MacMolPlt, a graphical user interface for the General Atomic and Molecular Electronic Structure System, GAMESS, is presented. Major features include an input builder for GAMESS; and display and animation of molecular structure, normal modes of vibration, reaction paths, orbitals, total electron densities, molecular electrostatic potentials, and density differences. The strategy for direct computation of orbital, total electron density, and molecular electrostatic potential surfaces is discussed. © 1999 by Elsevier Science Inc.

INTRODUCTION

As the speed of computers has increased and computational methods have improved, the complexity of molecular systems studied has steadily increased. Hence, the need exists for better tools to analyze the results produced by general electronic structure programs such as GAMESS[1] or Gaussian.[2] Indeed, several commercial programs, Chem3D,[3] HyperChem,[4] Spartan,[5] and Xmol,[6] for example, now have an interface to Gaussian.

Since GAMESS is one of the most widely used *ab initio* quantum chemistry programs, MacMolPlt has been developed to provide a user-friendly graphical interface to view the results from GAMESS calculations. MacMolPlt provides a range of visualization options that are useful to any user. The simple, intuitive interface is particularly useful for introducing novice users to GAMESS. Thus, MacMolPlt is useful in a variety of classroom settings, from teaching students the basic orbital shapes and orbital interactions in their first chemistry course, to introducing the techniques of quantum chemistry to advanced students without requiring an in depth knowledge of GAMESS. MacMolPlt is also very useful to advanced GAMESS users, providing insights into complex problems through the use of animations and the visualization of complex 2-D and 3-D surfaces such as orbitals, total electron densities, or molecular electrostatic potentials.

The remainder of this article is organized as follows: a description of the general capabilities of MacMolPlt is followed by more in-depth discussions of the most significant features—supported file formats, the GAMESS input generator, direct computation of surfaces, and output visualization. Special emphasis is placed on features that are unique to MacMolPlt.

PROGRAM OVERVIEW

The primary goal in the development of MacMolPlt was to provide a program to help analyze GAMESS output that is useful for both novices and advanced users. To aid the novice user, MacMolPlt also includes an optional input generator for GAMESS that allows the user to set up most of the common GAMESS input options. The advanced user may create input files by hand, perform the calculation, and then use MacMolPlt to analyze the results. Thus, MacMolPlt is capable of analyzing results from calculations that are too complex to be set up with the built-in input generator.

For flexibility, MacMolPlt can visualize information from a variety of file formats for input and output. These include all of the files used by GAMESS as well as several other formats, as discussed in the section on input file formats. In addition to files, MacMolPlt also allows the user to copy and paste molecular coordinates in several formats, providing a quick way to visualize molecular structures. Input can contain as little as a single atom in one geometry, to many hundreds of atoms in a series of hundreds of geometries, each potentially containing normal modes of vibration and several forms of molecular orbitals. The current focus of development has been to provide an interface to GAMESS. However, the core of the program is completely general. Thus, it can be easily extended to other programs, such as Gaussian, once the code to interpret the proper file formats is added.

Memory is allocated dynamically in MacMolPlt. Thus, there are no compile time limits on any significant parameter, including the number of atoms, the number of geometries, and the number of basis functions. This flexibility is made possible by an object-oriented code design. The design allows each geometry point, or frame, to be independent of the other frames. Thus, frames can be easily added or removed and each frame may contain different types and amounts of data. For

Color Plates for this article are on page 164.

Address reprint requests to: M.S. Gordon, Department of Chemistry, Iowa State University, Ames, Iowa 50011, USA.

Journal of Molecular Graphics and Modelling 16, 133–138, 1998

655 Avenue of the Americas, New York, NY 10010

1093-3263/98/$–see front matter
PII S1093-3263(99)00002-9

instance, only the first and last frames in an optimization normally contain molecular orbital vectors (MOs) and only the last frame will contain localized orbitals.

The most powerful features of MacMolPlt come from its output visualization. Not only can the visualization of output be crucial to a correct interpretation of the results, but it can also provide a valuable educational tool. MacMolPlt is tuned to providing real-time visualization with modest hardware requirements. This makes it possible to load onto a standard laptop computer to present results at scientific meetings or in classrooms for educational demonstrations. In addition to direct onscreen visualization, MacMolPlt can produce output in a variety of formats suitable for both hardcopy and Web-based publication. This is discussed in detail in the section on output file formats.

The output files from GAMESS are read directly and presented for the user to choose which parts to visualize. Visualization options include simple structures, series of structures, normal modes of vibration, molecular orbitals, total electron densities, density differences, and molecular electrostatic potentials. All surface visualization options are implemented directly in MacMolPlt, so that no further batch processing is required. Since some visualization options require enough CPU time to prevent their being considered truly real time, MacMolPlt allows the user to easily specify the display quality. Thus the user can compute surfaces and images to a low resolution for daily work, and then increase the resolution for publication quality output. This also allows users to tune the graphics level to the performance of their individual computers, from a 10-year-old computer to today's high-end computer with 3-D hardware graphics acceleration.

INPUT FILE FORMATS

MacMolPlt supports several text file formats for input. These include the three primary GAMESS output files (log file, dat file and irc file, described below), GAMESS input files, MolPlt mol files, XMol XYZ files, and Protein Data Bank (PDB) files. The last three formats provide compatibility with other programs, but typically include only basic structural information and possibly normal modes. Additional information such as energetics and molecular orbitals currently must come from GAMESS output files. The file parsing code is designed to automatically determine the type of the file selected for reading. Thus the user is not required to do any hand editing of the input file or perform any intermediate steps in order to read in the results of a GAMESS calculation.

The GAMESS output file containing the most information (the "usual" output for an electronic structure calculation) is the log file. Therefore, it is from the log file that MacMolPlt reads most of the results of the GAMESS calculation: initial geometry, basis set, GAMESS control parameters, molecular wavefunction, normal modes of vibration, localized molecular orbitals, plus all geometries computed as the result of a molecular optimization, intrinsic reaction coordinate (IRC), or dynamic reaction path (DRP) calculation. Results from IRC and DRP calculations, which may take several computational runs to complete, can be easily merged into one file.

In addition to the GAMESS log file, MacMolPlt also reads information from the GAMESS irc and dat files. The irc file contains the basic structure and energy information for IRC and DRP calculations. Since these calculations usually result in large numbers of geometries it is advantageous to store only the irc file since it is much smaller than the log file for the same run. The dat file is used by GAMESS to store formatted information that would be useful to restart a calculation, such as molecular orbitals, gradient, and hessian. MacMolPlt uses the dat file as an alternative source for the molecular orbitals.

MacMolPlt also reads in the input files for the MolPlt program, a 2-D X-Windows program included with GAMESS. These files contain the molecular structure and optionally one or more vibrational normal modes. For compatibility with other programs the XMol style XYZ format can be used. The XYZ format is very compact, including only the cartesian coordinates and possibly a single vibrational normal mode. Multiple geometries may be concatenated together, so it is a useful format for transferring a reaction animation from one program to another. Finally, files in the Protein Data Bank can also be used for input, though only the basic atomic coordinates are used by MacMolPlt.

OUTPUT FILE FORMATS

There are two purposes to the output file formats provided by MacMolPlt. The first is to create input files for GAMESS. This feature of MacMolPlt will be discussed in detail in the section on GAMESS input generation. The second purpose is to provide high-quality graphics for publication either on the Web or as hardcopy.

Hardcopy may be produced by two options. First, the desired image may be copied and pasted into another application for annotation before printing. This method works well for 2-D vector graphics, but can be problematic for bitmaps, since many applications cannot handle the very large bitmaps (possibly several megabytes in size) needed to produce acceptable quality when printed. The user may also print directly to any printer, or to a PostScript file suitable for later printing on any postscript printer.

Publication on the Web has somewhat different requirements than hardcopy output. The graphics resolution on the Web is usually limited to 72 dots per inch (dpi), and the size of the images must be kept to a minimum to ensure a reasonable download time. To achieve small file size MacMolPlt uses three different file formats. The first is the XYZ format. As described in the previous section, the XYZ format is quite compact and very useful for transferring one or more structures between programs or for Web display. The downside of this format is that the publisher has little control over what the viewer sees since the image is rendered upon download. The viewer must also have installed a browser plug-in capable of viewing XYZ files.

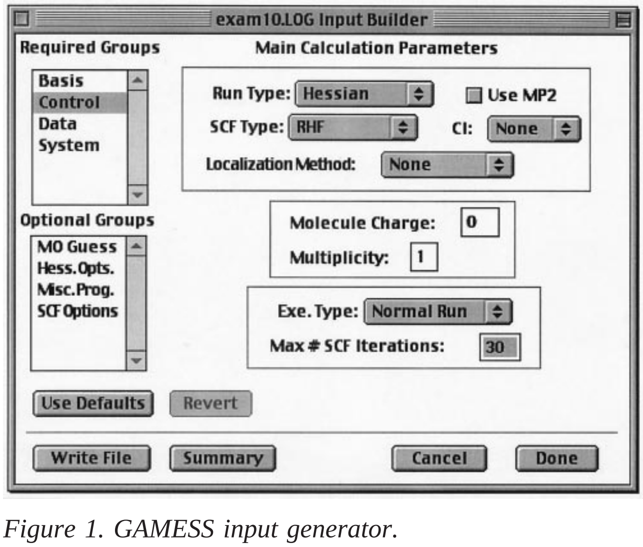
To provide images viewable on a wider variety of computers, MacMolPlt makes use of the QuickTime software libraries.[7] Using QuickTime, MacMolPlt can produce still images in either the QuickTime movie format or in the jpeg still image format. MacMolPlt also exports animations, including normal mode animations and reaction path animations, into QuickTime movie format. Movies of reaction path animations can include a simple X-Y plot of the energy or other property such as bond lengths or bond angles. The graph gives the viewer a better understanding of how the currently viewed structure relates to the other points in the animation. To help minimize the size of the movie file, MacMolPlt makes use of both spatial (per frame) and temporal (based on a series of frames) compression.

INPUT GENERATION

MacMolPlt incorporates a basic input generator to aid users, particularly novice users, to produce input files for GAMESS. However, the use of the input builder is completely optional. The ability of MacMolPlt to parse GAMESS output files correctly is not dependent on the use of the input builder. Figure 1 illustrates the basic interface of the input builder. Currently many, but not all, GAMESS input groups have been programmed into the input builder. The input groups that are present allow the user to specify the type of wavefunction, basis set, type of run (optimization, IRC, saddle point search, etc), molecular parameters (charge, multiplicity, symmetry, etc), and various other options that affect how GAMESS runs the calculations (amount of memory, maximum CPU time, etc). In addition, the input builder can provide assistance even to advanced users by making it easy to include groups such as optimized MO vectors from a previous calculation, or a list of Z-matrix variables, which tend to be tedious to input by hand.

Currently, MacMolPlt does not include a graphical point-and-click molecule builder. However, molecules can be built from scratch or modified from other sources by editing the cartesian or internal coordinates (in Z-matrix form). To make this process easier the molecule display updates as the user types in the coordinates, providing immediate visual feedback. In addition, the user can easily take coordinates from a variety of other sources including various file formats as well as by copying and pasting cartesian coordinates from a graphical builder such as Chem3D. Once a complex molecule has been created it is easy to modify its structure to create other isomers.

As the user makes choices such as the wavefunction type, MacMolPlt adjusts the available options to be consistent with the capabilities of GAMESS. For example, if the run type is set to optimize and the SCF type is set to restricted open shell Hartree-Fock (ROHF), the option to use Møller-Plesset second order perturbation theory (MP2) will be disabled since GAMESS cannot currently perform gradients with open shell MP2 wavefunctions. However, if the SCF type is set to RHF and MP2 is also selected, then an additional group will appear in the optional groups list for setting MP2 specific options. These optional groups, when selected, provide a pane with a list of options that normally default to an appropriate setting but still might be of interest. For example, if the user wants to reduce disk space use he or she can choose the SCF Options pane and turn on the Direct SCF option.



Main Calculation Parameters

Run Type: Hessian

□ Use MP2

SCF Type: RHF

CI: None

Localization Method: None

Molecule Charge: 0

Multiplicity: 1

Exe. Type: Normal Run

Max # SCF Iterations: 30

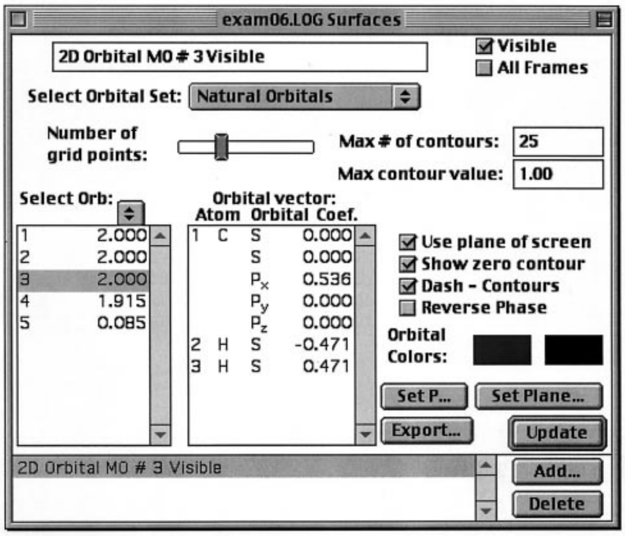
Figure 1. GAMESS input generator.

As a further aid to the new GAMESS user, the input builder also provides online help by means of the balloon help system. When activated via a menu item, helpful messages are displayed next to the item the cursor is currently over. The help message includes a brief description of the option along with the GAMESS group and keyword corresponding to this option. For example, the help text for the "Use MP2" option shown in Figure 1 is: "$CONTRL:MPLEVL - Click to use 2nd order Møller-Plesset perturbation theory. Implemented for RHF energies and gradients and open shell energies." Thus the user can easily associate keywords in the resulting input file with selections in the interface. This also provides information to the user regarding where to find more detailed information in the GAMESS manual.

SURFACES

The four supported surface types in MacMolPlt are Orbitals, Total Electron Densities (TED), Molecular Electrostatic Potentials (MEP), and any general grid of data read in from a file. Each of the supported surface types can be viewed either as a 2-D contour map in a user-defined plane, or as a 3-D isosurface. Each surface is independent of all other surfaces both within the same frame and for different frames, although it is quite easy to apply the same surface parameters across all frames, creating an animation of that surface. In addition, a single frame can have multiple surfaces of the same or different types. While any number of surfaces can be visible at the same time, it is unusual for more than two or three surfaces to be visually useful at once. A more useful advantage to having multiple surfaces is to set up surfaces with the optimum settings, and then make them invisible. Later they can be shown quickly, one at a time, for presentation purposes. The surface display is independent of the surface type and is discussed in more detail in the next section. One key feature of MacMolPlt is that all three major surface types are computed directly by MacMolPlt on the local CPU(s). This means that the user can visualize any desired surface without waiting on a batch process or other intermediate step.

The orbital surface supports the following types of orbitals: atomic orbitals (AOs), molecular eigenvectors including UHF alpha and beta sets and MCSCF or GVB optimized and natural orbitals, and localized molecular orbitals. Figure 2 illustrates the options available for 2-D orbital surfaces. Color Plate 1 illustrates a corresponding contour map in the plane of the CH_2 molecule. Color Plate 1 also includes a 3-D isosurface of the same orbital. The 3-D isosurface corresponds directly to one of the 2-D contours and helps illustrate the relative merits of 2-D and 3-D surfaces. The user is free to choose any orbital from any of the orbital sets that have been read in from GAMESS output. The example in Figure 2, which is an MCSCF calculation on CH_2, includes the MCSCF natural and optimized orbitals and the atomic orbitals as specified in the basis set. For MOs the user may list the orbital occupation numbers, as shown in Figure 2, or the orbital energies. The orbital symme-


exam06.LOG Surfaces

2D Orbital MO # 3 Visible

☑ Visible
□ All Frames

Select Orbital Set: Natural Orbitals

Number of grid points:

Max # of contours: 25

Max contour value: 1.00

Select Orb:	
1	2.000
2	2.000
3	2.000
4	1.915
5	0.085

Orbital vector:

Atom		Orbital	Coef.
1	C	S	0.000
		S	0.000
		P_x	0.536
		P_y	0.000
		P_z	0.000
2	H	S	-0.471
3	H	S	0.471

☑ Use plane of screen
☑ Show zero contour
☑ Dash - Contours
□ Reverse Phase

Orbital Colors: ■ ■

2D Orbital MO # 3 Visible

Figure 2. Orbital surface options.

tries are also shown if they are known. One particularly nice feature is that when the user selects a molecular orbital, the MO vector is listed. This allows users to see the orbital visually and observe how the orbital is numerically constructed from the individual atomic basis functions.

The total electron density surface type is available for any calculation that included the natural orbitals (the orbitals for which the density matrix is diagonal) in the output, such that the orbital occupations are known. Since the goal is to achieve real-time performance, the order of computation is very important. The method illustrated in Figure 3 works quite well. The key to this method is that AO amplitudes are calculated the minimum possible number of times. To accomplish this, the computation of the AO amplitudes is done before the loop over MOs and is stored in a one-dimensional array of the same dimension as the basis set. Then, within the loop over the occupied MOs, the MO coefficients are multiplied by the AO amplitudes to produce an MO amplitude. The MO amplitude is squared and multiplied by the MO occupation number to obtain the electron density for that orbital. The individual MO densities are summed to produce the total electron density. In addition to plain TED plots, MacMolPlt can also color map the MEP to the surface of 3-D contours. To reduce the time required to compute the MEP the calculation is performed only on the actual surface points. Thus, the MEP calculation is done after the TED grid is computed and contoured and the contour is reduced to the unique 3-D points.

```
Loop over grid points {
    Loop over basis functions (or AOs) {
        Compute and store AO amplitudes
    }
    DensitySum = 0.0;
    Loop over occupied MOs {
        MODensity = 0.0;
        Loop over basis functions {
            MODensity += MOCoefficient[I] * AOAmplitude[I];
        }
        DensitySum += MODensity * MODensity * MOOccupation;
    }
    Grid[n] = DensitySum;
}
```

Figure 3. Pseudocode for fast generation of a TED grid.

MacMolPlt also allows computation of MEP surfaces. An MEP is defined by Eq. (1):

$$V(r) = \sum_A \frac{Z_A}{|r - R_a|} - \sum_{\mu,\nu} P_{\mu\nu} \int \frac{\varphi_\mu \varphi_\nu}{|r - r'|} dr' \qquad (1)$$

where Z_A is the nuclear charge on atom A centered at R_a, and $P_{\mu\nu}$ is the density element corresponding to the basis functions ϕ_μ and ϕ_ν. While this integral is not difficult or expensive to compute, it does depend on r and thus must be computed separately at each grid point. This can result in substantial calculation times for large molecules and 3-D grids. However, the performance is quite acceptable for 2-D grids and will only get better as the speed of CPUs increases.

The fourth surface type supported by MacMolPlt is a simple grid read in from a file. Thus, it can be used to display any arbitrary property. However, because grids can be squared, added, and subtracted as they are read from file, the main use is for density differences. All surface types incorporate the ability to export grids and to synchronize grid parameters. Thus, it is possible to create density differences for individual orbitals (of any orbital type) as well as for the total electron density.

GRAPHICS

The goal of the graphics in MacMolPlt is to provide visual results with sufficient quality to be useful, while keeping the speed high enough to provide smooth real-time model rotation. This goal is accomplished using two separate drawing engines. The first uses simple 2-D vector graphics to visualize molecular structures, normal modes, and 2-D contour maps. The second uses a true 3-D display to render everything available in the 2-D mode plus complex 3-D surfaces. Real-time rotation is provided in either drawing mode via a virtual 3-D-trackball scheme.[8]

Color Plate 2 illustrates the basic 2-D drawing mode, which utilizes a rendering engine based on 2-D vector graphics, which are z-buffered on the basis of the atomic coordinates in the screen orientation. This engine provides high-speed graphics on virtually any computer made in the last ten years, while providing sufficient detail for normal modes with ball and stick model display. Also, owing to the small size required to save vector-based graphics, it is quite effortless to copy and paste images into other applications for annotation or inclusion into larger documents.

Color Plate 3 illustrates the true 3-D rendering mode, which uses the QuickDraw 3D[9] rendering engine to provide true-3-D display with lighting and shading effects. While this graphics engine does require considerably more CPU power than the vector graphics mode, it does run on most computers built in the last 4 years with no 3-D hardware accelerator required. MacMolPlt will also automatically take advantage of any 3-D hardware graphics accelerators available on a particular computer to provide enhanced speed and additional effects such as transparent surfaces.

Two-dimensional contour maps can be viewed using either display mode. The maps consist of a 2-D grid spanning a user-defined plane. Specification of the 2-D plane can be cumbersome for the user; to avoid this pitfall MacMolPlt uses the plane of the screen to define the plotting plane. Once the desired plane has been found, the plane's orientation can be fixed such that it rotates with the molecule. One advantage of this scheme is that with the plane set to the plane of the screen and with the ability of MacMolPlt to compute the surfaces in real time, the user can scan the plane through the molecule, allowing the user to locate areas of special interest very quickly. The plane of the screen may also be easily set to a plane defined by any three atoms or defined by the cartesian axes. One time- and memory-saving feature is that the contours are displayed as the grid is contoured. Thus the individual contours are not stored.

A 3-D grid spanning a volume of user-customizable size and resolution defines all 3-D surfaces, viewable only in the true 3-D mode of display. Once the grid has been computed or read in from a file, it is contoured into a 3-D surface of constant value (an isosurface), using an improved marching cube algorithm.[10] Vectors perpendicular to the surface are computed to smooth out solid surface display. The surface is then submitted to the 3-D-drawing engine for display as either a solid (opaque or transparent) or wire frame surface. The colors of the positive and negative contours of each surface can be set independently such that multiple surfaces can be distinguished from each other if displayed simultaneously. Since the grid is preserved, the user can easily and quickly scan the possible isosurface values to find the value that the user deems best.

ANIMATIONS

One important feature of MacMolPlt is its ability to animate computations. This includes the animation of normal modes as well as any multiframe computation from GAMESS. However, the geometry alone does not provide enough information to truly understand a reaction. Other parameters are also important, such as the energy (the total energy or the kinetic energy for DRPs), gradient, or specific bond lengths or angles. To provide information on these parameters MacMolPlt includes a simple graphing feature that can provide an indication of the relative value of the current frame to the other frames in the animation. Because the simple graphing code is not intended for the creation of publication quality output, the graphed values may be exported as tab-delimited text suitable for most graphing software. Since an animation is difficult to illustrate on paper, a sample figure is not presented here. However, sample animations are available on this group's online Web site.[11]

The two primary examples of the utility of animation of a multiple geometry calculation are IRCs and DRPs. The IRC path is a minimum energy path connecting a transition state to its nearest minima. Because each point is significant, it can be important to view the IRC path to gain a better understanding of how a reaction proceeds. MacMolPlt makes viewing the entire reaction path easy by providing built in animation capabilities as well as the ability to splice together multiple computations (e.g, several IRCs for several steps in a complex mechanism) into one smooth animation. Thus an entire reaction sequence can be shown instead of just the stationary points, leading to a better understanding of the entire reaction.

Animations are even more important for viewing the results of a dynamic reaction path calculation, since there are no stationary points computed. By their nature, dynamics calculations require the computation of a large number of geometries. However, since MacMolPlt imposes no limits on the number of geometries even large runs can be accommodated. Also, since the DRP is often required to compute more geometries for numerical stability than are necessary for a smooth animation, the code can also be configured to skip points as the file is read in from disk. Finally, the speed of the animation can be adjusted to suit the number of points in the animation and the user's personal taste.

CONCLUSION

MacMolPlt provides an easy-to-use interface for the GAMESS package, offering a range of features appealing to the novice as well as the veteran user. While the input is aimed primarily at new GAMESS users, the powerful output visualization capabilities should appeal to any user. MacMolPlt also bridges the need for a program that is fast enough to use for every computation, yet capable of producing output of sufficient quality for publication. MacMolPlt also supports the most popular formats for publication of results on the Web, such as jpeg and quicktime movies. In the future, as the speed of computers and computational methods improves, the need for the powerful visualization techniques provided by programs such as MacMolPlt will only increase as the complexity of molecular systems studied increases.

Many enhancements are planned for future versions of MacMolPlt. Among these are plans to extend the file parsing code to other programs, such as Gaussian, and to other cross-platform formats, such as the MDL MolFile. In addition, direct support for symmetry is planned to aid students in learning molecular group theory and molecular point group symmetry. Visualization enhancements, such as a line graph of infrared intensity versus frequency, are planned. Finally, versions of MacMolPlt for other operating systems, such as Windows, are planned.

AVAILABILITY

MacMolPlt is available free of charge to all users. Currently, MacMolPlt is available on the Macintosh platform (PPC and 68K native versions available). A Windows version is also planned but not yet available. To obtain MacMolPlt please refer to the Web address: http://www.msg.ameslab.gov/GAMESS/Graphics/MacMolPlt.shtml. Complete system requirements, feature lists, as well as the program itself can be found at that site.

ACKNOWLEDGMENTS

The development of MacMolPlt has been supported by grants from the National Science Foundation (CHE-9633480), the

Department of Defense CHSSI program, Iowa State University in the form of a Department of Education GAANN fellowship awarded to B.M.B., and Apple Computer, Inc., which provided a computer for code development.

REFERENCES

1 Schmidt, M.W., Baldridge, K.K., Boatz, J.A., Elbert, S.T., Gordon, M.S., Jensen, J.H., Koseki, S., Matsunaga, N., Nguyen, K.A., Su, S., Windus, T.L., Dupuis, M., and Montgomery, J.A., Jr. The general atomic and molecular electronic structure system. *J. Comput. Chem.*, 1993, **14,** 1347

2 Frisch, M.J., Trucks, G.W., Schlegel, H.B., Scuseria, G.E., Robb, M.A., Cheeseman, J.R., Zakrzewski, V.G., Montgomery, J.A., Stratmann, R.E., Burant, J.C., Dapprich, S., Millam, J.M., Daniels, A.D., Kudin, K.N., Strain, M.C., Farkas, O., Tomasi, J., Barone, V., Cossi, M., Cammi, R., Mennucci, B., Pomelli, C., Adamo, C., Clifford, S., Ochterski, J., Petersson, G.A., Ayala, P.Y., Cui, Q., Morokuma, K., Malick, D.K., Rabuck, A.D., Raghavachari, K., Foresman, J.B., Cioslowski, J., Ortiz, J.V., Stefanov, B.B., Liu, G., Liashenko, A., Piskorz, P., Komaromi, I., Gomperts, R., Martin, R.L., Fox, D.J., Keith, T., Al-Laham, M.A., Peng, C.Y., Nanayakkara, A., Gonzalez, C., Challacombe, C.Y., Gill, P.M.W., Johnson, B.G., Chen, W., Wong, M.W., Andres, J.L., Head-Gordon, M., Replogle, E.S., and Pople, J.A. *Gaussian 98.* Gaussian, Inc., Pittsburgh, Pennsylvania, 1998

3 CambridgeSoft Corporation, Cambridge, Massachusetts

4 HyperCube, Inc., Gainsville, Florida

5 Wavefunction, Inc., Irvine, California

6 Network Computing Services, Inc., Minneapolis, Minnesota

7 Apple Computer, Inc., Cupertino, California; see http://www.apple.com/quicktime/

8 Chen, M. 3-D rotation using a 2-D device. *Develop: The Apple Technical Journal*, 1993, **14,** 40; Chen, M., Mountford, S.J., and Sellen, A. A Study in interactive 3-D rotation using 2-D control devices. *ACM Siggraph Comput. Graphics*, 1988, **22,** 121

9 Apple Computer, Inc., Cupertino, California; see http://www.apple.com/quicktime/qd3d/

10 Lorensen, W.E., and Cline, H.E. Marching cubes: A high resolution 3-D surface construction algorithm. *ACM Siggraph Comput. Graphics*, 1987, **21,** 163

11 The animations and other supplementary material are available at http://www.msg.ameslab.gov/Group/Supplementary_Material/MacMolPlt/