

# Computational Quantum Chemistry Experiments via the Web

Brian F. Yates<sup>1\*</sup> and Brian J. Duke<sup>2</sup>

<sup>1</sup> School of Chemistry, University of Tasmania, Private Bag 75,  
Hobart TAS 7001, Australia  
Brian.Yates@utas.edu.au

<sup>2</sup> School of Biological, Environmental and Chemical Sciences,  
Northern Territory University, Darwin, NT 0909, Australia  
b\_duke@octa4.net.au

**Abstract.** The use of web technology for the provision of efficient and user-friendly interfaces to a variety of computational chemistry programs used in undergraduate experiments is described. Together with pedagogical descriptions of the theory, these interfaces are used to enhance the accessibility of experiments in computational quantum chemistry for senior students at the undergraduate level. The advantages and disadvantages of this approach are discussed and recommendations for future incorporation into the curriculum are made.

## 1 Introduction

For some time we have been involved in teaching computational quantum chemistry courses to students at our respective institutions. For many students this has been their first experience of computers beyond simple word processing and, prior to the work described here, they had to contend with a bewildering variety of personal computers, operating systems, communication programs, and analysis tools. In 1995 we were successful in obtaining funding from the Committee for Australian Undergraduate Teaching for a project entitled "Learning Modules for Computational Chemistry Over AARNET". This enabled us to develop the skeleton of a set of web-based modules for teaching quantum chemistry. Other modules in molecular modelling were developed by team members from Monash University and Swinburne University of Technology.[1] In 1997 one of us was successful in obtaining a Teaching Development Grant from the University of Tasmania which enabled us to develop the quantum chemistry modules further to the point where they could be used as part of our fourth-year undergraduate course.[2] Although direct funding for this work has now stopped, the team members continue to collaborate and we have continued to improve and develop the material under the banner of the Australian Computational Chemistry Via the Internet Project (ACCVIP).[3]

The quantum chemistry modules consist of some 200 web pages of material that comprehensively covers a large amount of quantum chemistry, several on-line com

---

\* To whom correspondence should be addressed.

prehension tests which give students immediate feedback and guidance when the answer is incorrect, a large number of on-line tutorial problems and exercises, and web-based interfaces to several computational chemistry programs. Thus the modules consist of a large amount of pedagogical material plus the interfaces.

The advantage of having so much of the pedagogical information on the web is that it provides more flexible access to the material. It is a resource which is available to the students 24 hours a day and it can be used in a distance education environment. However in our experience there is still a need for the students to print out the material and for the instructor to lead them through it in a classroom situation in order to gain the best understanding. Clearly different things work for different people. Part of the skill of being a good teacher is to assess where people are up to and present the material accordingly. For all of its use in flexible delivery, a computer cannot respond in a genuinely personal way to an individual's situation. Thus the real strength of this project and the one that has been of most benefit to the students is the development of the web-based program interfaces.

We have used material from the quantum chemistry modules in teaching a variety of courses, both remotely to a small number of MSc students based overseas and locally to students at our institutions in third and fourth year. This article describes some of our recent work in adapting the material to provide a set of computational quantum chemistry experiments via the web.

## 2 Web Interfaces

We have written the web interfaces as simple html documents which obtain information from the user via the use of forms and then process the information using various CGI scripts. Students are not required to log in to a specific computer or to learn any Unix commands. It is of course possible to restrict access to the web pages (either by use of passwords or by IP address) and this is used in some situations. We have developed web-based interfaces for a number of programs including the James-Coolidge, Hückel, Extended Hückel, MOPAC,[4] Gaussian,[5] GAMESS(US)[6] and DeFT[7] computer programs. Our CGI scripts incorporate a variety of features which are listed below:

*a) The ability to run the job interactively or via a queue.* We use the interactive method for the Hückel and Extended Hückel programs and the queuing method for the quantum chemistry programs. One of the advantages of the use of a queue is that because we have designed the experiments to involve fairly small calculations we can set a fairly short time limit on the length of each job. This prevents a student's calculation from going on endlessly when a mistake has been made.

*b) The ability to check the status of the queue.* This is a simple web interface to the utility in whatever queuing system is being used that displays all the entries in the queue.

*c) The ability to execute jobs remotely.* The web server machine does not need to be the same computer that the actual calculations are run on. For example, this allows an instructor at the Northern Territory University in Darwin to design and implement their own web interface as an html file served from an NTU computer, but for the actual calculations to be carried out on a computer at the University of Tasmania in Hobart. We have tested this in the case of the MOPAC program. Subject to licensing

agreements and permissions from the appropriate software companies, this allows computing resources to be shared for educational purposes. It also gives the instructor flexibility while reducing duplication of system administration.

*d) The ability to produce a Chime[8] or Rasmol[9] picture of the input structure.* This is achieved by calling Babel[10] from within the CGI script to convert the input data for the quantum chemistry programs into pdb format. This is then displayed for the user.

*e) The ability to view the output as it is running.* This is an advantage for short calculations in that it gives the student immediate feedback and allows them to see the results of the calculations as they are produced. For interactive jobs (for the Hückel and Extended Hückel programs) this is the only way the output is available.

*f) The ability to email the output back to the user.* Students are asked to specify their email address in the web interface and then the output from the calculation is emailed back to them. The success of this feature relies on the student typing their address correctly, so to help them the instructor can set the default for that part of their address relating to the particular institution (for example at the University of Tasmania it is set to "@postoffice.utas.edu.au").

*g) The ability to log all usage.* When difficulties occur due to jobs not running correctly or when students request extensions for work it is useful to be able to look at the logs to determine what went wrong. Our CGI scripts record the date, the calling page, the email address of the user, the address of the remote host, the input data file, and any error result upon submission of the job.

We have used three different styles of web-based interfaces for the computational chemistry programs:

*a) Forms with drop down menus.* These interfaces have spaces where the user is required to fill in certain information (for example, the title and Cartesian coordinates), and other areas where options for the calculation can be easily selected through the use of drop down menus (for example, the level of theory, the basis set, the type of calculation (single point, optimisation or frequency), and the charge and multiplicity). This style of interface is well suited for the beginning student where most of the information can be hidden away or presented as the default menu item so as to simplify the input. It may be that all the student has to do is fill in their email address and press 'submit'. This style of interface is also well suited to fairly straight forward experiments where only a few theoretical methods are to be used. This has the advantage that only the options specified by the instructor for that experiment need be presented in the menu. Finally this style of interface is also well suited for enhancing with extra commands the basic input data provided by the student. For example, the choice of 'single point' in the type-of-calculation menu may be translated to mean 'scf=tight' in the Gaussian program input data.

In practice we have developed several different general interfaces of this type to the quantum chemistry programs, each with an increasing number of options.

*b) Plain form.* With this style of interface the user is presented with a single box on the page into which they may enter a complete data file, including pre- and post-processing commands (for example, the '%' commands in Gaussian).

*c) Upload a file.* With this style of interface the user selects an input file on their local computer and it is then uploaded to the web server and submitted to the queue.



```

Z-matrix: Make sure you hit "Enter" after the last
line, and only hit it once.<BR>
<TEXTAREA NAME="matrix" ROWS=4 COLS=40></TEXTAREA><P>
<INPUT TYPE="submit" VALUE="Run Gaussian data">
<INPUT TYPE="reset" VALUE="Reset"><P>
</FORM>
</BODY>
</HTML>

```

In this case we have restricted the user to the choice of two levels of theory, four basis sets, and two types of calculation. This produces the interface shown in Fig. 1.

Select the appropriate menu or fill in the appropriate box. The output will be emailed to you.

Your e-mail address:  
 @

Title:

Select the method in the menu below.  
Method:  ▾

Select the basis set in the menu below.  
Basis set:  ▾

Charge  Multiplicity

Select geometry optimisation or single point:  ▾

Z-matrix: Make sure you hit "Enter" after the last line, and only hit it once.

**Fig. 1.** Example interface to the Gaussian program

While the html code is fairly straight forward, the crown jewels of this project are of course the CGI scripts that make it all work.

### 3 Experiments

The interfaces described above have been used extensively in a series of exercises and problems incorporated into our modules on semi-empirical theory and basic *ab initio*

theory which were developed as part of the ACCVIP collaboration. More recently we have written a series of third-year undergraduate experiments in computational quantum chemistry which have been used at the Northern Territory University (NTU) over a number of years and will be used for the first time at the University of Tasmania in late 2003.

These experiments include one entitled "Calculating simple force constants with computational quantum chemistry", developed as part of the Australian Physical Chemistry Enhanced Laboratory Learning (APCELL) initiative.[11] The experiment was first used in the current form with second-year undergraduate students at NTU in 2002. It grew out of a similar experiment on the bending force constant of CO<sub>2</sub> which had been used for many years at NTU. That experiment in turn owed its origins to a description by Mathews.[12] Full details of the current web-based experiment will be published elsewhere.[13]

This experiment investigates the calculation of the force constants of some simple diatomic molecules: N<sub>2</sub>, O<sub>2</sub> and F<sub>2</sub>. For each molecule, students are asked to calculate the minimum energy and the single point energies at slightly distorted geometries. Then by assuming a quadratic parabola the force constant is determined. Students are asked to investigate the results using different levels of semi-empirical, *ab initio*, and density functional theory, and compare their findings to their expectations and to experiment.

In this experiment we have used web interfaces with drop down menus to limit the number of choices that the students have. For example, in the semi-empirical calculations with MOPAC the methods are restricted to just the choice of AM1 or PM3. We have also used these web pages to provide a combination of the explanation and the interface. This allows us to provide the information (about the program or about the experiment) at the time it is needed to perform the calculation. The MOPAC interface used in this experiment is shown in Figures 2 and 3. This shows how the MOPAC keywords are described where necessary, and how extra information is provided about the O<sub>2</sub> triplet molecule. This interface is available at <http://www.chem.utas.edu.au/teaching/KRA301/exp8/mopac.html>.

**Method 1 - Semi-empirical methods using Mopac**  **Mopac**

MOPAC is a well-established and extremely widely used program for calculating the energy of a molecule using a semi-empirical method. These methods are based on the formalism of molecular orbital theory but select many of the parameters from experimental data, such as ionisation potentials, electronic transitions, etc., or fit parameters so the predicted enthalpies of formation agree with experiment for a test set of molecules. They can be readily used for quite large molecules.

**Fig. 2.** Interface to MOPAC for the force constant experiment

There are several different methods in the MOPAC code. We will use just two - AM1 and PM3. The method is selected from a drop down menu. First, determine the optimised bond distance for your diatomic molecule. Then calculate the energy difference when the bond length is increased by 1.0 pm and then decreased by 1.0 pm. Note that you actually have to give the bond length in Ångstrom units. Select just one of the two methods, but if you are in a group, organise different members of the group to select different methods and later pool your results.

Since this method uses significant resources on the server computer, although only for molecules much larger than the ones you are using, the calculations are queued and the results are returned to you by e-mail. Make sure you get your e-mail address correct below.

Here is a simple interface to **MOPAC**. All you need to do to run this calculation yourself is fill in your email address (change the default that the page gives you), select one of the methods, PM3 or AM1 from the menu, modify the title window, type in a Z-matrix, check you have the keywords correct and then press "Run MOPAC data".

Your e-mail address:

@

Title:

Method:  ▼

Other keywords:

The default action in MOPAC is to optimise the geometry - the 'precise' keyword tightens up the convergence criteria to ensure that sufficient accuracy is obtained in the final geometry. Do this calculation first. Subsequently, to do what is called a single point calculation, i.e. just at the geometry you have given it, for the shortened and lengthened bonds, you have to replace 'precise' with the '**Iscf**' keyword.

No other keywords need be added for  $F_2$  and  $N_2$ . However, since  $O_2$  is a triplet ground state, you have to add, after either 'precise' or 'Iscf', '**uhf triplet**'.

Z-matrix or cartesian co-ordinates:

Extract the energy from either the enthalpy of formation value in kJ (actually kJ/mol) or the total energy in eV [ 1 electron volt (eV) = 96.485 kJ/mol]. Why do the enthalpy and energy values in kJ/mol give the same value for the energy difference caused by expanding or contracting the bond?

Fig. 3. Continuation of interface to MOPAC for the force constant experiment

## 4 Discussion

The major advantage of the web-based approach described here is that it provides a consistent cost-effective interface to a variety of computational chemistry programs. Rather than confronting students with a disconcerting array of login procedures, Unix commands, and job submission processes and perhaps restricting them to certain computers in the undergraduate lab where the software is available, this approach enables the student to use just one program, a web browser, and to perform the calculations from wherever internet access is available (their lab, dorm or internet café). It does not matter which type of personal computer or operating system the students are using; each student will see the same thing. This uniformity encourages a more ready acceptance by the students. In addition the material is available 24 hours a day and the students are not restricted to just certain times when the computer lab is open. These reasons all make it easier for the student to focus on the underlying computational chemistry which is being taught. It also makes it much easier for the instructor since they do not need to worry about setting up communication programs (telnet, x-windows, etc.) and login accounts, and teaching the students about the operating system. Instead they can focus on teaching students about computational chemistry.

There are some disadvantages however. In our implementation we have deliberately removed nearly all reference to and use of files. In the basic web interface the student does not deal with any input files, output files, scratch files, or checkpoint files. Our experiments are designed to use small molecules (so there is no need to type a lot of information) and there is no need to use a checkpoint file. This simplifies things for the student, but it would certainly constrain the serious user. We do print the input data at the top of the output so that the student begins to get the idea of what a complete input file looks like. Our more advanced experiments also make use of the plain form and the file upload interfaces in which a complete set of input data is required. These move the student towards an understanding of how to run real research jobs. We have also developed some advanced interfaces which have ways to keep and handle the checkpoint file in Gaussian, although this has not yet been trialled with students.

Another possible disadvantage is that although this method of presentation provides a good mode of instruction for undergraduate students, are we really teaching them how to use the computational quantum chemistry programs in real life? After completing the experiments could they say they know how to use the MOPAC program in a research situation for example? The answer is probably 'no'.

It is worth comparing our approach to that of others. In recent years a number of other workers have developed web interfaces to computational quantum chemistry programs (see for example references [14]-[17]). The most sophisticated of these is WebMO developed in the group of Professor William Polik.[14,18] This not only provides some of the capabilities described in this article but also includes a tool for visualising the input data and output results on the web via a Java applet. This is a very nice package and is highly recommended. We have continued to use our system because of the flexibility it offers us as instructors, and because we are able to readily integrate the delivery of information within the interfaces. As described above, this

allows us to provide the instructions for an experiment on the same page as the interface and to very easily tailor the interface for the particular set of options required for that experiment.

The ubiquitous use of the web means that without doubt it is the most used and most accessible interface in the world today. Nearly all the traditional computer information systems in a university (library catalogues, financial management, student records, timetables, booking systems, email, lecture notes, and so on) are now available via the web. Students have embraced this and become used to the convenience that it offers. It is only natural that they expect the same ease of use from our computational chemistry programs and the interfaces described here go some way to meet their needs. We expect this demand from students to increase and our computational quantum chemistry experiments will be adapted and expanded to meet this demand.

**Acknowledgements.** It is a pleasure to acknowledge our collaborators who contributed so much to this project: Dr Graeme Cross, Dr Margaret Wong, Dr Ted Lloyd, Dr Kai Ling Ng, Dr Yos Ginting, Trent Wale and Vinutha Ramakrishna.

## References

1. Salter-Duke, B.J.; Wong, M.G.; Lloyd, E.J.; Yates, B.F.; Cross, G.J.: Learning Computational Chemistry via the Internet. CAL-laborate (October 1998) 31–33
2. Yates, B.F.: KRA400 Chemistry Honours – Topics in computational chemistry. <http://www.chem.utas.edu.au/staff/yatesb/honours>, accessed on 21 February 2003
3. Australian Computational Chemistry Via the Internet Project (ACCVIP). <http://www.chem.swin.edu.au/CAUT.html>, accessed on 21 February 2003
4. Stewart, J.J.P.: MOPAC93. Fujitsu Limited, Tokyo, Japan (1993)
5. Frisch, M.J., Trucks, G.W., Schlegel, H.B., Gill, P.M.W., Johnson, B.G., Robb, M.A., Cheeseman, J.R., Keith, T., Petersson, G.A., Montgomery, J.A., Raghavachari, K., Al-Laham, M.A., Zakrzewski, V.G., Ortiz, J.V., Foresman, J.B., Peng, C.Y., Ayala, P.Y., Chen, W., Wong, M.W., Andres, J.L., Replogle, E.S., Gomperts, R., Martin, R.L., Fox, D.J., Binkley, J.S., Defrees, D.J., Baker, J., Stewart, J.P., Head-Gordon, M., Gonzalez, C., Pople, J.A.: Gaussian 94, Revision B.3. Gaussian, Inc., Pittsburgh PA (1995)
6. 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.J., Windus, T.L., Dupuis, M., Montgomery, J.A.: GAMESS(US). *J. Comput. Chem.* **14** (1993) 1347–1363
7. St-Amant, A.: DeFT. University of Ottawa, Ottawa (1998)
8. Chime. <http://www.mdli.com/chime/>, accessed on 21 February 2003
9. Rasmol. <http://www.umass.edu/microbio/rasmol/>, accessed on 21 February 2003
10. Walters, P., Stahl, M.: Babel. (1992–1996)
11. Kable, S., Buntine, M., Jamie, I.: APCELL. <http://www.apcell.org>, accessed on 21 February 2003
12. Mathews, G.P.: Experimental Physical Chemistry. Clarendon Press, Oxford (1985) section 5.13, pages 280–287
13. Duke, B.J., Ramakrishna, V.: Calculating Simple Force Constants With Computational Quantum Chemistry. *Aust. J. Educ. Chem.* (2003) to be submitted
14. Polik, W.: WebMO. <http://www.webmo.net>, accessed on 21 February 2003

15. Rösch, N.: Web interface to Gaussian. [http://theo1.theochem.tu-muenchen.de/qc1/help/interface\\_e.html](http://theo1.theochem.tu-muenchen.de/qc1/help/interface_e.html), accessed on 21 February 2003
16. Portal interface to Gaussian. <http://www.osc.edu/research/bioinformatics/portal/gaussian.shtml>, accessed on 21 February 2003
17. WebSubmit. <http://math.nist.gov/mcsd/savg/websubmit/>, accessed on 21 February 2003
18. A review of WebMO. [http://www.gaussian.com/nl001\\_up.htm](http://www.gaussian.com/nl001_up.htm), accessed on 21 February 2003