Learning Computational Chemistry via the Internet

[B. J. Salter-Duke]
School of Mathematical and Physical Sciences, Northern Territory University

[M. G. Wong]
School of Engineering and Science, Swinburne University of Technology

E. J. Lloyd, Victorian College of Pharmacy, Monash University

B. F. Yates, Department of Chemistry, University of Tasmania

G. J. Cross, Water Studies Centre, Monash University

Chemistry, and in particular computational chemistry and molecular modeling, is well suited for learning over the Internet using the World Wide Web augmented by other Internet tools. We have developed a series of learning modules which are now used for teaching a Masters degree in computational chemistry offered by three Australian universities in distance education mode over the Internet as well as being used in modified form for final year and Honours year courses. Two techniques play a key role in allowing the student a hands-on flexible approach. Web forms allow access to a variety of computational codes allowing students to run their own data without having to have the code on their own machine and in a more user-friendly fashion than standard batch mode. Web helper applications and plug-ins such as *Rasmol* and *Chime* allow students to investigate molecular structures in ways that they, not the instructor, determine. These developments will be discussed along with pointers to extending these approaches to other areas of chemistry.

The Internet, and particularly the World Wide Web (the web), has been hyped-up as the ideal way for offering flexible education. We have been using web-based materials for three years and some of the advantages and disadvantages are beginning to become clear. We start with a cautionary note as a preliminary to our thesis that the web should be used largely for the things it does well and that it should not be expected to do everything. We then describe some approaches that are particularly appropriate for teaching computational chemistry but have some validity in the teaching of other areas of chemistry. These areas include those such as stereochemistry where visualization is a great aid to understanding the relationship between structure and reactivity and theoretical chemistry where interaction with both local computer programs and remote computer programs with the web can be integrated.

Even though the majority of students, when polled, claim to enjoy learning via the web they still try to print out most web pages. This may be due to the sheer impossibility of keeping large amounts of material in the mind. They then go away and study the printed material, returning to the computer to use the web interactively. When asked about this, they respond that too much time in front of a computer screen may damage their eye sight or give them a stiff neck. These are responses to any intensive computer-based instruction. If students print everything off, then a more cost effective way of delivering flexible learning might be the traditional approach of sending out study booklets. We prefer to conclude that web materials should contain a lot of activities for which the web is particularly appropriate. One further advantage of the web-based materials over printed manuals and CD-ROMs is that web-based materials can easily be modified, updated with current information or corrected in response to student feedback. In this way the materials are never static. Our considered opinion is that web materials should be used by instructors who are capable of modifying HTML (HyperText Markup Language - the standard method for writing web pages) and even CGI (Common Gateway Interface) scripts. CGI scripts are usually written in *Perl*; a scripting language, although any computer language can be used.

We try to keep our web pages as flexible as possible and include interactive activities which are particularly appropriate for web use whenever these are possible and useful. It is not appropriate to just transfer lecture notes to the web. Our computational chemistry materials were initially prepared with the support of a 1995 CAUT (Committee for Advancement of University Teaching, Australia) Teaching Grant, then trialled with various final year and Honours students, and finally cast in a form which is used to teach a Masters and Graduate Diploma[1] course offered in distance education mode. The web materials here are the main technique for delivering the content, but they are augmented by some written materials and much email contact between the instructors and students. Students are expected to write essays and reports based on the web materials and the interactive exercises where they used the web to carry out computational chemistry calculations. Two techniques stand out as being particularly appropriate for web-based chemistry and it is these we will concentrate on here.

The first tackles the problem of how to get students to do calculations. Computational chemistry involves calculations which require highly intensive use of computer resources and use code that is often extremely large and complex, having been developed over a long period of time by many people. While some of these codes have been adopted - often with a loss of function - to run on the current range of fast PCs, it is impossible to expose students to the full range of computational chemistry techniques if all they have at home is a PC while they study by distance education. There are even problems for internal students: indeed our initial grant application, written in 1994 before the web had become really popular, addressed the problem of students from a group of universities accessing computer codes which for cost and other reasons were only available in one or a few of the universities involved. Cost is also a factor for the home based student since many adaptations of computational chemistry codes for PCs are very expensive, while codes for Unix servers are cheaper or even free.

The current web technology for providing programs is Java, where a program is delivered by the server to the user's browser and run there. Java is a machine independent language so it can in theory be used on all computers. However this requires converting a wide variety of programs; many very large and written in a variety of languages into Java. This is just not feasible. Even if it were, download times for set up of the programs would be prohibitive. A further problem is that these programs often require large amounts of memory and disk storage space which we cannot be sure the users have available.

The solution is to provide the programs and run them on the server machine. The user enters data into a web page with forms, boxes, buttons and pull-down menus. The data are returned to the server where processing occurs via a CGI script. The script essentially reformulates the data into the form expected by the appropriate program and then calls the executable code to run the program. The output is then collected by the CGI script and returned to the user as a web page. In many cases the executable program takes too long for this to be practical, so the output is collected later by the server and sent to the user by email. The job is also queued to ensure that only one job at a given time is using the server processor and the server disks which have to store large amounts of temporary data in scratch files. Users are able to check the queue to see where their job lies and, if it is running, can inspect the output as it happens.

While users are initially given carefully restricted tasks and view annotated output from the programs to learn how to interpret the output, they can quickly move on to use the programs for any task they themselves devise. These will of course include attempts to complete assignment problems, but is not restricted to them. Even with assignments the students can choose their own way to do them and will often make mistakes in preparing the data. They each learn in their own way.

The only limitations of this technique lie with the server. Will it get overburdened with jobs so the user has to wait hours or days before receiving results? Does the server have to do other things? So far we have not had major problems. Jobs are restricted to a maximum time and memory usage as determined by the CGI scripts. Only users who have authenticated their use with a user name and password can access the web pages that lead to running these large programs. If pressure on one machine gets too high, a simple change in a configuration file can redirect the jobs to a different machine from the main web server. This technique is quite general in that it is relatively simple to change a CGI script to run different data for a different program. Writing the first CGI scripts was difficult for us but now adapting them is fairly simple.

Clearly this technique is not going to be useful in all web education, or even in all areas of chemistry education over the Internet. However, there are areas outside computational chemistry where it is applicable. One such area is environmental modeling. In many other areas quite simple calculations would enhance the learning experience. These could be done either by CGI scripts on the server or by Java applets on the browser.

The second specifically chemical application arises from the visual nature of much good chemistry learning. Chemists have to appreciate the three-dimensional nature of molecules and explore them from many different viewpoints. Pictures of molecules can be downloaded from the server in the same image formats as used for other pictures on the web, but they have two disadvantages. Firstly too many large image files slow the web experience particularly if downloaded via a slow modem line. Secondly they are static pictures with only one viewpoint. The student cannot explore alternative viewpoints. Animated movie pictures of molecules would partially lessen the second problem but would increase the first problem. The student can still not control the viewpoints and how they investigate the molecule. The current solutions to this problem of visualizing molecules are fairly widely known and are not original to our work. Common solutions are the use of *Chime* (or *Rasmol*) and VRML. Both involve the downloading of a relatively small data (not image) file and the use of extra code on the browser machine.

Rasmol[2] is a molecular modeling program which takes as data the Cartesian coordinates of each atom in the molecule. The molecule is then displayed and the user can, using the mouse and pull-down menus, rotate the molecule in any desired direction or alter the ways the atoms and bonds are displayed. *Rasmol* can be used as a helper application by the browser. The most common format for the atomic data used by *Rasmol* is the PDB (Brookhaven Protein Data Base) format. Such a file has the extension .pdb to its name. The browser is configured so that when a file with the .pdb extension is received from the server, *Rasmol* is opened up as a new window and the molecule displayed.

Chime[3] takes this process further. The *Chime* code, developed from *Rasmol*, is incorporated into the browser such as *Netscape Navigator* or *Internet Explorer* as a plug-in. When the file with the .pdb extension is received it can be handled in two ways. In the simplest approach a new browser screen displays the molecule which can then be manipulated with the mouse. The left mouse button allows rotation of the molecule. The right mouse button leads to a pull-down menu. Most of the flexibility of *Rasmol* is present. In a more elaborate use of *Chime*, the web page defines an embedded window and initial conditions for the display. In this way a page can display several molecules each from their own PDB files on the screen at the same time. Each molecule can be manipulated in the same way as described above. *Chime* with embedded windows for several molecules is now the preferred technique, but *Rasmol* is still useful for general interactive use and if the user has a browser that can not support *Chime*.

VRML (Virtual Reality Modeling Language) is in many ways more powerful and more general as it is not restricted to molecular models. Like *Chime* it uses a plug-in. A common one for PCs is *Cosmos Player*[4]. Some chemistry applications have recently been discussed by Casher, Page and Rzepa[5]. The great advantages of VRML over *Chime* are the ability to zoom-in on part of the image or manipulate just part of the image. We anticipate a great future for this technique in chemistry web pages, but we have not as yet included any major use of it in our own materials. Preparation of VRML materials is more difficult than preparation of *Chime* materials and the required plug-ins are not yet as widely available as *Chime*, although this is changing fast.

These visualization techniques allow most of the usage of physical ball and stick kits, along with adding new features. One important new feature is the ability by a click on a menu to change the type of model used. For example a ball and stick display can be replaced by a space filling model based on Van der Waal's radii. For proteins a ribbon or a strand model can replace the atom based models. Each display has its own virtues. The main disadvantage is that the student cannot design their own model but only get what is in the PDB file sent by the server. They can, however, view the model in a large number of ways and their viewpoints are theirs, not the instructors. We are investigating web pages that assist the student in designing their own molecules for display via *Chime* or *Rasmol*.

These techniques are particularly pertinent in computational chemistry where the shape and structure of a molecule is crucial for predicting its properties. Structure, along with energetics and kinetics, are central themes in chemistry and biochemistry so web techniques must be exploited to emphasise this. All molecular diagrams in web pages should preferably be presented as embedded *Chime* images. Students using the material are not forced to move the molecule around, but they might get greater insight by doing so.

A number of other plug-ins or helper applications can greatly improve use of the web for learning. Examples include spreadsheets to access and manipulate data from the web allowing a wide range of exercises and manipulations. Plug-ins are also required for video and audio applications which can enliven the web experience.

Web materials, like books or CD-ROMs are arranged and ordered by the instructor. The techniques described here allow students more freedom. Within some constraints students can define their own tasks and their own learning. The disadvantages of web materials will only be overcome if they contain many examples of these more interactive techniques.

References

1. For more details see:- http://www.smps.ntu.edu.au/msc/
2. *Rasmol* 2.6 Beta-2a, Roger Sayle, Glaxo Wellcome, UK, ftp://ftp.dcs.ed.ac.uk/pub/rasmol/
3. *Chime* 2.0 Beta 4, Chemscape, http://www.mdli.com/download/chime/
4. *Cosmo Player* 2.1. http://www.cosmosoftware.com/download/index_player.html
5. Casher, O., Page, C. and Rzepa, H., *Chemistry in Britain*, Volume 34, Number 9, September 1998, page 26 and http://www.chemsoc.org/gateway/molmodel.htm

B. J. Salter-Duke
School of Mathematical and Physical Sciences
Northern Territory University
Darwin, NT 0909
Australia
b_duke@lacebark.ntu.edu.au

M. G. Wong
School of Engineering and Science
Swinburne University of Technology
VIC 3122
Australia
marg@freon.chem.swin.edu.au

E. J. Lloyd
Victorian College of Pharmacy
Monash University
VIC 3052
Australia

B. F. Yates
Department of Chemistry
University of Tasmania
Hobart, TAS 7001
Australia

G. J. Cross
Water Studies Centre
Monash University
Caulfield Campus, VIC 3145
Australia