

## Use of JavaScript in simple quantum-chemical applications

David H. Mosley\*, Jean-Marie André

*Laboratoire de Chimie Théorique Appliquée, Facultés Universitaires Notre-Dame de la Paix, Rue de Bruxelles 61, B-5000 Namur, Belgium*

Received 1 November 1996; accepted 28 January 1997

---

### Abstract

The JavaScript scripting language is a simple way of adding interactive functionality to HTML (Hypertext Mark-up Language) pages. In this paper we present the use of JavaScript in two sample quantum-chemical applications, destined primarily for educational purposes. Examples are given in which students are able to enter data and invoke calculations using JavaScript scripts via forms-based interfaces, and follow links to outlines of the theory involved, key equations, and to other pages on the World Wide Web. The performance and potential of interactive scripts in these and wider quantum-chemical applications is discussed. © 1997 Elsevier Science B.V.

*Keywords:* World Wide Web; Internet; JavaScript; Ab initio calculations; Helium atom; FSGO

---

### 1. Introduction

The amount of chemistry-related information on the World Wide Web (WWW) continues to grow at a remarkable rate [1]. Hypertext documents, with embedded static and manipulable images, animations, sounds, and links to other documents, have the potential to communicate information in a highly attractive and original way. The interested reader may like to consult selected chemical examples given in [2,3]. Interactive functionality through which hypertext pages respond to user input can be added into HTML (Hypertext Mark-up Language) documents by using languages such as Java [4] and JavaScript [5]. In the last year, many examples of the use of Java have appeared on the WWW for molecular structure editors, and front-end interfaces for calculations and database searches [6].

A significant amount of educational material can also be found on the WWW. This may be presented in the form of virtual textbooks, as support for regular lecture and practical courses, or as standalone remote learning material [7]. Aspects of applied quantum and computational chemistry play an increasingly important role in undergraduate and graduate chemistry courses as molecular modelling tools are used more and more widely in several domains of chemical research. Our aim here is to investigate the potential of JavaScript scripts embedded within HTML pages for model, interactive, quantum-chemical calculations. The two examples we give are:

- the self-consistent field–linear combination of atomic orbitals (SCF-LCAO) calculation of the  $1s^2$  ground state of the helium atom in a double-zeta basis of Slater-type orbitals; and
- calculation on the  $H_2$  molecule using a sub-minimal FSGO basis set.

In both cases, a simple interface for user input

---

\* Corresponding author. E-mail: David.Mosley@fundp.ac.be

is provided using HTML forms. The JavaScript calculations are supported with hyper-text links to relevant equations and background theory to help the user understand the calculation. The examples are inspired by exercises in a book, *Exploring Aspects of Computational Chemistry: Concepts and Exercises* [8]. The examples remain close to the one of the aims of the book in seeking to provide insights into the actual mechanics of quantum-chemical calculations, through performing model calculations on simple systems by hand or with the help of short programs. This is to help bridge the gap, apparent to many students, between formal theory and practical implementation.

In the next section we give a brief introduction to the JavaScript language, its functionality, and its embedding within HTML documents. This is followed by a more detailed discussion of the examples, before closing the paper with comments on the performance of the scripts and on scope for further development of this type of teaching material on the WWW.

## 2. A few words about JavaScript

JavaScript is an object-based scripting language through which dynamic content can be added to HTML documents, or HTML pages can be made to respond to user input. It is important to note the relationship between the Java and JavaScript languages. While some of the syntax of JavaScript resembles the Java language, the difference between JavaScript and Java should be stressed. Java is a fully-fledged programming language, both for standalone applications and applications accessed from within HTML pages (called *applets*). Java programs are compiled into byte-codes before being made available for execution on clients. JavaScript scripts are embedded within HTML documents, and are interpreted (rather than being transferred in a compiled byte-code form and executed) by the client. This is also distinct from CGI (Common Gateway Interface) programs, in which execution is carried out on the server. For Java and JavaScript applications, the computation is performed on the client, sparing the server from increased load, and decreasing network traffic. Many examples exist of JavaScript scripts being used to verify form

```
<SCRIPT Language="JavaScript">
function dist(form) {
    xx=Math.pow((form.x1.value-form.x2.value),2)
    yy=Math.pow((form.y1.value-form.y2.value),2)
    zz=Math.pow((form.z1.value-form.z2.value),2)
    form.distance.value=Math.sqrt(xx+yy+zz)
}
</SCRIPT>
<FORM>
x1: <INPUT TYPE="TEXT" NAME="x1" SIZE=10>
y1: <INPUT TYPE="TEXT" NAME="y1" SIZE=10>
z1: <INPUT TYPE="TEXT" NAME="z1" SIZE=10><BR>
x2: <INPUT TYPE="TEXT" NAME="x2" SIZE=10>
y2: <INPUT TYPE="TEXT" NAME="y2" SIZE=10>
z2: <INPUT TYPE="TEXT" NAME="z2" SIZE=10><BR>
<INPUT TYPE="button" VALUE="Calculate distance"
OnClick="dist(this.form)">
<INPUT TYPE="RESET" VALUE="Clear input">
<BR>
Result: <INPUT TYPE="TEXT" NAME="distance" SIZE=25> <BR>
</FORM>
```

Fig. 1. JavaScript script and HTML source for a simple example of using JavaScript script to find the distance between two points, given by their respective  $x$ ,  $y$ , and  $z$  coordinates, introduced by the user via an HTML form.

input to CGI programs to reduce unnecessary data transfer.

JavaScript supports run-time operations on a small number of data types (numeric, Boolean and string values), objects and methods. Functions are also supported within JavaScript, which enhances both the power of the language and the simplicity of the relevant scripts. Indeed, the appeal of JavaScript lies in its relative simplicity as a tool for developing client and server Internet applications.

A simple example of a JavaScript script to find the distance between two points, given by their respective  $x$ ,  $y$ , and  $z$  coordinates introduced by the user via a HTML form is given in Fig. 1. The resulting screen output is then given in Fig. 2, which the reader may then try.

The example given in Figs. 1 and 2 is a very basic example of the use of JavaScript, but does illustrate the simple interface between the HTML form-based input and the numerical calculation carried out by the script. The two sets of  $x,y,z$  coordinates entered in the data fields of the form are passed to the function `dist`, defined between the `<SCRIPT>...</SCRIPT>` tags, using the `OnClick` event handler, which, as its name suggests, responds to a mouse click on the Calculate distance button. Values of individual elements of the form ( $x1,y1,\dots$ ), identified by the `NAME` attribute in the form, are referred to in the function `dist` as `form.name.value`. In the example, the result of the calculation is written to the empty `Result` box in the form, via the `NAME` attribute of the input data field. It is also possible, using the JavaScript `write` method, to write directly to an HTML document, to a separate frame in the browser window, or to a different browser window, this giving the possibility to create HTML pages *on the fly*.

The screenshot shows a web form with a dotted border. It contains two rows of input fields for coordinates: the first row has fields for x1, y1, and z1; the second row has fields for x2, y2, and z2. Below the input fields are two buttons: 'Calculate distance' and 'Clear input'. At the bottom of the form is a label 'Result:' followed by an empty input field.

Fig. 2. Screen output for the JavaScript script and HTML source listed in Fig. 1.

JavaScript scripts can be included at any point in an HTML document, as is the case with the example in Fig. 1. It is common practice to place the scripts in the `<HEAD>...</HEAD>` section of an HTML document. The JavaScript itself can be hidden from old browsers which are unable to run scripts through enclosing the commands between comment tags, `<!--...-->`, which must be included immediately after (and immediately before) the `<SCRIPT>` (`</SCRIPT>`) tags.

In the applications presented here, we are particularly interested in exploiting the numerical capabilities of JavaScript in illustrative quantum-chemical calculations. It is perhaps worth noting that several methods for simple mathematical functions are built in to the `Math` object which make JavaScript attractive for these applications and for scientific usage. These methods include basic trigonometric functions (`sin`, `cos`, `tan`, `asin`, `acos`, `atan`), square root and power functions (`sqrt`, `pow`) (as used in Figs. 1 and 2), a random number generator (`random`), logarithmic and exponential functions (`log`, `exp`), and miscellaneous numerical functions (`floor`, `ceil`, `max`, `min`, `round`).

Another important issue for the development of client/server Internet applications is that of portability. Unfortunately, such has been the rapid pace of development of JavaScript that compatibility across platforms and versions is not assured. Examples of bugs and differences in behaviour on different platforms do exist. Scripts written in JavaScript, which originated at Netscape [5], should work with versions of Netscape 2.0x and higher, and also Microsoft Internet Explorer 3.0. However, full and equivalent functionality of JavaScript cannot be assumed across platforms and versions. Support for JavaScript across future versions of all browsers is not assured. This is an important question to be faced by the Internet community. It would be a great pity if moves away from platform-dependency of codes and applications were only to be replaced by equivalent problems with browser-dependency. The scripts presented here have been developed and tested with Netscape Versions 2.02 and 3.0 on a Macintosh Quadra 610 and under AIX on IBM RS6000 machines. They do not make use of any of the more particular version- and platform-sensitive aspects of JavaScript, and so stand a good chance of being portable.

### 3. Sample applications

#### 3.1. SCF-LCAO calculation of the $1s^2$ ground state of the helium atom in a double-zeta basis of Slater-type orbitals [9]

This calculation uses the single determinant singlet ground-state wavefunction of the helium atom based on a doubly occupied atomic orbital. Using a double-zeta basis in the LCAO-SCF method, the atomic orbital is expanded in terms of two basis functions,  $1s$  and  $1s'$ . The solution of the Hartree–Fock equation in a basis of two functions  $1s$  and  $1s'$  is equivalent to the solution of the LCAO-SCF matrix equation of order 2. In the version of this example available on the WWW since May 1996 [9], a link to a page giving a more detailed background to the theory and form of the matrix elements is given.

This calculation, despite its simplicity, does illustrate the full conceptual complexity of an SCF calculation. In the form given in Fig. 3, the reader is able to select orbital exponents, the convergence threshold, and the maximum number of iterations. Good values to choose for the exponents are in the region of 1.45 and 2.90. These values are close to the values for the *best* double-zeta results for the helium atom, exponents of 1.45363 and 2.91093, obtained by Roetti and Clementi [10], which give an SCF total energy of  $-2.861672598$  a.u. On running the calculation, a new HTML document is created in a second window

Fig. 3. Input form for SCF-LCAO calculation of the  $1s^2$  ground state of the helium atom in a double-zeta basis of Slater-type orbitals.

which contains the results of the calculation, together with hypertext links to pages giving the formulae used to calculate the one- and two-electron integrals, and the total energy. In an educational environment, the student may use the program to find the optimum orbital exponents and to investigate the variation of the total energy as a function of the orbital exponents in region of the minimum, or to carry out some integral evaluations and SCF iterations by hand using the formulae available at the click of a mouse button.

The trial solution (or initial guess) of the SCF equations is  $c_{1s} = 1$  and  $c_{1s'} = 0$ . This is an arbitrary choice, and superior choices for the initial guess do exist. In most cases, 20 iterations is more than sufficient for the change in the value of the coefficients between successive iterations to be less than  $10^{-8}$ . An error is flagged if the convergence conditions are not met within the number of iterations specified.

#### 3.2. Calculation on the $H_2$ molecule using a subminimal FSGO basis set [11]

FSGOs (Floating Spherical Gaussian Orbitals), introduced by Frost in the mid 1960s [12], correspond to s-type Gaussian orbitals which are allowed to *float* in space so as to optimally represent each localized pair of electrons. The basis set used is often referred to as *subminimal*, since only one function is required to represent each electron pair. In this example, the user is able to study the  $H_2$  molecule by using a single FSGO. As in the previous example, input to the script is via an HTML form (Fig. 4), and the user is able to specify the internuclear distance, an initial guess for the orbital exponent, and the displacement of the orbital from the centre of the H–H bond. In the course of the calculation, the orbital exponent is optimized according to a simple optimization scheme.

The utility of this calculation is to enable the student to explore the symmetric and asymmetric FSGO solutions for the hydrogen molecule [13]. Using symmetry arguments, the location of the FSGO at the centre of the H–H bond leads to a physically nonsensical situation at the dissociation limit, i.e., two protons ( $2H^+$ ) and an isolated pair of electrons ( $2e^-$ ). At distances greater than a certain critical distance, the optimal (lowest energy) solution will no longer be the situation in which the orbital is located at the centre of the H–H bond, but in which

Enter starting exponent:

Enter H–H distance: (in bohr)

Enter displacement of FSGO from centre of H–H bond: (in bohr)

Start calculation    Reset input

Fig. 4. Input form for the calculation on the  $H_2$  molecule using a subminimal FSGO basis set.

the orbital has *jumped* towards one of the atoms. Thus, dissociation in the FSGO scheme corresponds to



Plotting the total energy as a function of the displacement of the centre of the FSGO from the centre of the H–H bond ( $R_{H-H} > 5.5$  bohr) in small steps will give access to these symmetry broken solutions, and provide an insight into the symmetry dilemma in the conventional Hartree–Fock LCAO-SCF scheme and the question of Hartree–Fock instability.

#### 4. Discussion

The two examples presented here demonstrate part of the potential of interactive pages on the WWW. The calculations, while being simple, do illustrate important concepts in quantum and computational chemistry. An attractive feature is the ease and transparency of the interface provided by the HTML form. With this in place, the reader is not obliged to deal with unfamiliar operating environments, or with an abundance of program options, few of which are of any significance for the task in hand. The presence of the theoretical background and relevant equations at the click of the button helps to make the computational process somewhat less abstract.

As a choice of language, JavaScript seems to be well suited for the cases chosen. However, a more complete language, such as Java, would be necessary

for more complex examples, able to execute across different platforms on the WWW. For the simple applications given, computational efficiency and performance is not an issue. Work is in progress on the extension and enhancement in the form of a Java *applet* of the FSGO application illustrated here, which will include a graphical representation of the results and the possibility to optimize the position of the orbital. The double-zeta LCAO-SCF calculation on the helium atom could be easily be extended to, for example, a configuration interaction calculation. On a wider scale, the possibility now exists and is being investigated for the development of object-oriented computational chemistry codes across the WWW using Java.

#### 5. Notes on the References

The dynamic and vast nature of the WWW means that it is impossible to create up-to-date and exhaustive lists of contents and examples. The links given in many of the references below are simply suggested starting points and are personal choices of examples. All links were last checked to be active on 5th August, 1997.

#### Acknowledgements

This work has benefited from the financial support

of the Belgian National Interuniversity Research Program on “Sciences of Interfacial and Mesoscopic Structures” (PAI/IUAP No. P3-049). The authors gratefully acknowledge the financial support of the FNRS-FRFC, the ‘Loterie Nationale’ for the convention No. 9.4593.92, and the FNRS within the framework of the “Action d’impulsion à la recherche fondamentale” of the Belgian Ministry of Science under the convention D.4511.93.

## References

- [1] Several comprehensive lists of chemistry resources on the Internet are available. Three good examples are: the Sheffield ChemDex at <http://www.shef.ac.uk/academic/A-C/chem/chemdex/>; UCLA ChemPointers at <http://www.chem.ucla.edu/chempointers.html>; and Chemistry Information via the Internet, US FDA/CFSAN at <http://www.cfsan.fda.gov/~dms/chemist.html>.
- [2] H.S. Rzepa, B. Whitaker and M.J. Winter, Chemical applications of the World-Wide-Web system, *J. Chem. Soc. Chem. Comm.*, (1994) 1907 (also see <http://chem.leeds.ac.uk/papers/html/chem-com/chem-com.html>); O. Casher, G.K. Chandramohan, M.J. Hargreaves, C.J. Leach, P. Murray-Rust, H.S. Rzepa, R. Sayle and B.J. Whitaker, Hyperactive molecules and the World-Wide-Web information system, *J. Chem. Soc. Perkin Trans. 2*, (1995) 7 (also see <http://chem.leeds.ac.uk/papers/html/Perkin2/tartrazine.html>).
- [3] WWW Chemical Structures Database, <http://schiele.organik.uni-erlangen.de/services/webmol.html>; NIST Chemistry WebBook, <http://webbook.nist.gov/chemistry/>; WebElements, <http://www.shef.ac.uk/~chem/web-elements/>; Gaussian Basis Set Order Form, <http://www.emsl.pnl.gov:2080/forms/basisform.html>; Molecule of the Month, <http://www.bris.ac.uk/Depts/Chemistry/MOTM/motm.htm>.
- [4] Java, Sun Microsystems, Inc., 2550 Garcia Ave., Mountain View, CA 94043-1100, USA, 1996. See <http://java.sun.com/index.html>.
- [5] JavaScript. See <http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/index.html>.
- [6] Molecule Editor 1.0, D.M. Bayada, <http://www.chem.leeds.ac.uk/ICAMS/people/denis/moledit.html>; Sketch and Fetch, TRIPOS, Inc., <http://www.webcom.com/tripos2/SandF.html>; Java<sup>TM</sup>-powered Quantum Chemistry, F. Friedman-Hill, <http://herzberg.ca.sandia.gov/MolEdit/>; JavaScript Pocket Computer for Physics, J.P. Vigneron, <http://www.physique.fundp.ac.be/Comp/PhysComp.html>.
- [7] For example: Global Instructional Chemistry at Imperial College, UK (<http://www.ch.ic.ac.uk/GIC/>); Practical Exercises in Quantum Chemistry at ETH, Switzerland (<http://www.scsc.ethz.ch/chem/gcii.html>). Also see H.P. Lüthi, G. Vacek, A. Hilger and W. Klopper, Practical exercises in ab initio quantum chemistry — the World Wide Web as a teaching environment, to appear in: J. Leszczynski (Ed.), *Computational Chemistry: Reviews of Current Trends*, Vol. 2, World Scientific, Singapore (1997); Frank Potter’s Science Gems — Physical Science I and II, at University of California, Irvine (<http://www-sci.lib.uci.edu/SEP/physical.html> and <http://www-sci.lib.uci.edu/SEP/physical2.html>); Theory of Atoms in Molecules, R.F.W. Bader, at McMaster University, Canada (<http://www.chemistry.mcmaster.ca/faculty/bader/aim/>).
- [8] J.M. André, D.H. Mosley, M.C. André, B. Champagne, E. Clementi, J.G. Fripiat, L. Leherte, L. Pisani, D. Vercauteren and M. Vračko, *Exploring Aspects of Computational Chemistry: Concepts and Exercises*, Presses Universitaires de Namur (Namur) (1997).
- [9] This example can be found permanently at [http://www.chimie.fundp.ac.be/javas/he\\_dz\\_calc.html](http://www.chimie.fundp.ac.be/javas/he_dz_calc.html).
- [10] C. Roetti, E. Clementi, *J. Chem. Phys.* 60 (1974) 4725.
- [11] This example can be found permanently at [http://www.chimie.fundp.ac.be/javas/fsgo\\_calc.html](http://www.chimie.fundp.ac.be/javas/fsgo_calc.html).
- [12] A.A. Frost, *J. Chem. Phys.* 47 (1967) 3707.
- [13] J.M. André, G. Hardy, D.H. Mosley and L. Piela, FSGO Hartree-Fock instabilities of hydrogen in external electric fields, in: Y. Ellinger and M. Defranceschi (Eds.), *Strategies and Applications in Quantum Chemistry*, Kluwer, Dordrecht, 1996.