

WebMO: Web-based computational chemistry calculations in education and research

William F. Polik¹  | J. R. Schmidt² 

¹Department of Chemistry, Hope College, Holland, Michigan, USA

²Theoretical Chemistry Institute and Department of Chemistry, University of Wisconsin—Madison, Madison, Wisconsin, USA

Correspondence

J. R. Schmidt, Theoretical Chemistry Institute and Department of Chemistry, University of Wisconsin—Madison, Madison, WI 53706, USA.
 Email: schmidt@chem.wisc.edu

Edited by: Anna Krylov, Associate Editor

Abstract

WebMO is a web-based interface for all major quantum chemistry programs. WebMO uses a server-client architecture that installs on a single server or cluster computer and provides access to state-of-the-art computational chemistry programs from a standard web browser. The web interface provides a 3-D molecular editor, pre-defined calculations types, job submission and monitoring, visualization of results, and user management tools. Barriers to using state-of-the-art computational chemistry in teaching and research are minimized through WebMO's universal accessibility, its intuitive and uniform interface to all programs, no software to install on client computers, and support for multiple users with a single instance. Applications of WebMO throughout the undergraduate curriculum are provided. The extensible open-architecture design allows for collaboration among educators, researchers, quantum chemistry program developers, and the WebMO interface developers.

This article is categorized under:

Computer and Information Science > Visualization

Electronic Structure Theory > Ab Initio Electronic Structure Methods

Software > Quantum Chemistry

KEY WORDS

chemistry education, electronic structure, visualization

1 | BACKGROUND

1.1 | Computational modeling

Computation has been identified as the third pillar of scientific inquiry, alongside experiment and theory.¹ Computation allows scientists to model realistic problems that do not have analytical solutions and to answer questions which are difficult to address by experimental means. In chemistry, computational modeling is used extensively in organic, physical, biochemical, material science, and pharmaceutical research. Computational chemistry provides insight into chemical structure, interactions, and reactivity, which in turn permits prediction of chemical behavior.^{2,3} A substantial fraction of the published literature in the chemical sciences include some computation to analyze and interpret results,⁴ and modeling can be used to predict promising leads for subsequent experimental investigation.

The utilization of computation has had many challenges associated with it, including access to sufficient computing power, accuracy and efficiency of algorithms, user comprehension of complex theories, and usability of the computational software. Many of the technical issues associated with scientific computing have been recently overcome. For example, typical student laptop computers have more computing power than supercomputers of 25 years ago⁵ and high-end research computers of a decade ago. Current algorithms can now accurately compute reaction energies to within a kcal/mole and vibrational frequencies to within several cm^{-1} for small molecules in the gas phase, and modern codes can qualitatively model more complex molecules and condensed phase systems. Graphical user interfaces make computers and computational software far more accessible to non-technical users than older text-based interfaces.

Over the last two decades, use of computational chemistry has moved from the realm of specialized theoretical chemistry developers, through many graduate research groups, and more recently into some undergraduate and even high school courses (Figure 1). Because of its utility and importance, the American Chemical Society has identified access to and use of computational chemistry as essential for undergraduate chemistry programs.⁶ While progress has been made to introduce computation into the undergraduate chemistry curriculum, it is often as an isolated laboratory experience or in a specialized upper-level course. Computational chemistry has not yet become infused throughout undergraduate chemistry education, in contrast to chemical synthesis or spectroscopic characterization.

1.2 | Barriers to usage

Despite the widespread availability of modern computing resources, three significant barriers remain that are limiting the use of computational chemistry in education and by non-experts:

- *Usability*: State-of-the-art computer codes are difficult for non-specialists to use due to differing and sometimes cryptic input syntax and voluminous text output. This issue can be mitigated by installing a specialized graphical user interface on the user's computer for the computational program, for example, Gaussview for Gaussian,⁷ IQmol for Q-Chem,⁸ gmolpro for MolPro,⁹ or a combined package, for example, Spartan.¹⁰ However, each computational program requires its unique user interface with different terminology, different workflows, and different ways of communicating between the interface and program. Although some general-purpose interfaces exist, they typically do not expose the varying specialized capabilities of each computational program, for example, Avogadro,¹¹ Gabedit,¹² Chemcraft.¹³ Other interfaces offer visualization capabilities but cannot create input files or submit jobs, for example, Jmol,¹⁴ Molden,¹⁵ Molekel.¹⁶ In all cases the interface must be installed separately on each user's computer, requiring that non-technical users install, maintain, and troubleshoot highly specialized software. Finally, these user interfaces are typically not available for all platforms (Windows, MacOS, Chrome OS, and Linux), which poses a significant practical problem in an educational environment.
- *Resources (money, expertise, time)*: Supporting computational chemistry requires significant financial resources for computers powerful enough to run complex calculations, expertise to configure the computing environment and install specialized software, and time to maintain systems and train users. Different implementation models exist, but they carry various trade-offs. Users can run basic calculations on their existing laptop computers without

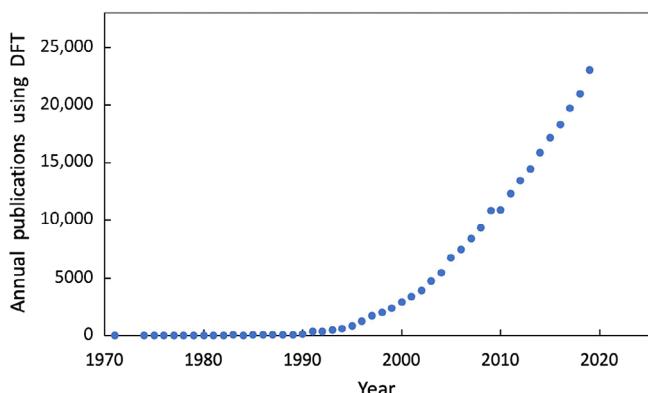


FIGURE 1 Computational chemistry usage in science. The number of publications each year that include the topic of density functional theory was searched with web of science, illustrating continued growth of computation in chemical modeling in science literature

purchasing any additional hardware, but this is limited to small molecules at low theory levels and typically requires substantial user-support for software installation and trouble-shooting on widely varying computers. Computer labs can be configured with more powerful workstations, computational codes, and graphical user interfaces, but this can be costly in terms of hardware, multiple software licenses, classroom space, and significant ongoing maintenance. Centralized computer clusters are often used to run high-end calculations, but this requires a very large investment in computing and networking hardware as well as technical expertise to build, configure, and maintain such a system.

- **Accessibility:** The accessibility of computational programs and user interfaces depends on the deployment model. Being able to run calculations on one's own laptop or office computer is most convenient, but is limited in capabilities and requires software installation for each user. Computer labs require physical access to a limited set of resources. A centralized computational server typically requires installation of a local user interface on the user's computer, configuration of network access and firewall rules, and creation of individual Linux accounts on the server, none of which may be transparent for non-expert users.

1.3 | WebMO approach

WebMO¹⁷ is a web-based interface for all major computational chemistry packages. WebMO installs on a single server or computer cluster and provides access to computational chemistry programs, allowing users to run state-of-the-art computational chemistry calculations using only a web browser. WebMO's approach greatly lowers usability, resource, and accessibility barriers, and thereby allows non-specialists to easily use modern computational chemistry in their teaching and research.

The WebMO interface is designed for both ease-of-use and flexibility. Input files are created using a 3-D editor, job options are available through intuitive menus, and output is displayed in tables, plots, and interactive images. Reasonable defaults are provided for novice users, and full access to input and output files is available for advanced users. The interface is uniform across all computational programs, allowing WebMO to be used with different programs and throughout the entire curriculum.

WebMO reduces financial and administrative demands because a single instance serves many users, thereby minimizing computer cost, license fees, and maintenance effort. Installation is automated and administration is done through a web-based interface, allowing non-specialists to install and maintain the software.

A key feature of WebMO is that since users interact with WebMO via their web browser, there is no software to install on end user computers, and WebMO is independent of the end user's computing platform. WebMO supports all modern operating systems (Windows, MacOS, Chrome OS, Linux), all popular web browsers (IE, Edge, Safari, Chrome, Firefox), and even smartphones and tablets (iOS, Android). In short, WebMO makes computational chemistry universally accessible from almost any computer or smart device.

1.4 | WebMO history

WebMO was initially developed in 2000 to support use of computational chemistry by local non-experts and then released for free public use. In 2001 WebMO Pro added various computational, visualization, and file management enhancements and was released as a commercial add-on to the freeware version. In its early incarnation, WebMO's 3-D editor and visualization features were implemented as Java applets, which allowed interactive graphical visualization to be implemented by client computers rather than by the server, thereby distributing this computationally intensive task. WebMO's predominant uses were for teaching at the undergraduate level and for computational research in both academia and industry. Subsequent releases added support for more calculation types and compatibility with additional computational programs. In 2007 WebMO Enterprise was released which includes features to support large numbers of users, SMP and MPI parallel computing, and cluster-based computing architectures. As web technology evolved, internal architectural changes were made in 2017 to move 3-D interactivity from Java to CSS/HTML5 technology, making WebMO even more independent of the client computer. There are currently three feature levels of WebMO¹⁸:

- WebMO Basic includes the integrated 3-D editor, interfaces to computational chemistry programs, and graphical visualization of results.

- WebMO Pro provides support for additional calculations, visualization, and job management features.
- WebMO Enterprise has enhancements for managing large numbers of users and interfacing with computer clusters.

In order to provide broader access to WebMO, WebMO apps for both iOS and Android were developed that can either function as a self-contained molecular editor/viewer or connect to a WebMO server.

WebMO usage has continued to grow, and it is now widely used by industry, supercomputer centers, research centers, universities and colleges, and even high schools. WebMO has been recognized as the quantum chemistry interface “most commonly used by undergraduate students.”¹⁹

2 | OVERVIEW OF CAPABILITIES

2.1 | Supported programs

WebMO encapsulates the capabilities of popular computational chemistry programs into a uniform web-based interface. The programs currently supported include: GAMESS,^{20,21} Gaussian,^{22,23} MolPro,^{24–26} Mopac,^{27,28} NWChem,^{29–31} Orca,^{32–35} PQS,^{36,37} PSI,^{38–40} Q-Chem,^{41,42} TeraChem,^{43–45} Tinker,^{46,47} Quantum Expresso,^{48,49} and VASP.^{50,51}

Support for each computational program (“engine”) is done via an open-architecture set of files that define the engine configuration, display of HTML pages, job templates, running jobs, and parsing job output. To add support for additional programs, a script is provided that generates support files for a new engine based on a similar existing engine. These support files can then be edited to reflect the capabilities and syntax of the new engine. Adding the set of new files to a WebMO installation enables the new computational engine without making any modifications to the main WebMO source code.

2.2 | Workflow

Regardless of the particular computational engine, WebMO uses a uniform workflow to run and visualize a new computational chemistry job: (1) create the input structure, (2) choose a computational engine and optionally a queue, (3) select a calculation type and job options, (4) submit and monitor the job, and (5) visualize results.

Input structures are created by using the built-in 3-D editor, from the result of a previous WebMO calculation, or by importing a structure from one of several external databases. The desired engine is selected from the list of installed engines. Common calculation types and job options are available through drop-down menus. The job is submitted to the job queue and automatically executed when compute resources become available. Job status is continuously displayed and intermediate results are available as the job runs. When the job completes, the output file is automatically parsed and results can be viewed.

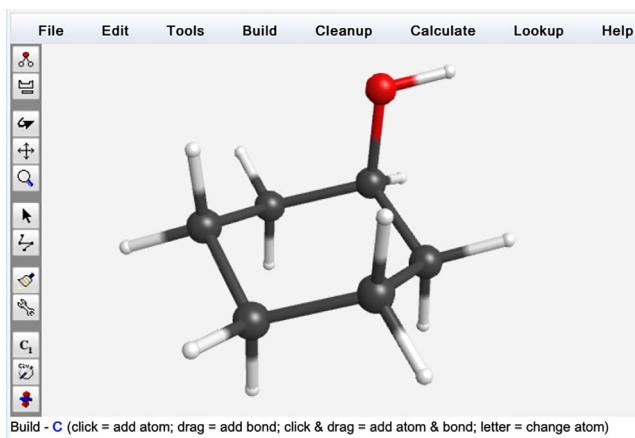
2.3 | Editor

The WebMO 3-D molecular editor allows users to create arbitrary molecules by clicking to insert atoms and dragging to insert bonds (Figure 2). A perspective image is rendered with rotate, translate, and zoom capabilities. The clean-up command automatically adds missing hydrogens, defines atom hybridization, applies VSEPR rules, and idealizes bond lengths, angles, and dihedrals, which can subsequently be manually adjusted. Use of previous jobs, a local fragment library, and importing structures by name from online databases^{52,53} are available for creating more complex structures.

The 3-D editor has some built-in computational capabilities to generate accurate starting geometries, including pre-optimization with molecular mechanics^{54,55} and symmetrization⁵⁶ of nearly symmetric structures. For additional control over the creation of geometry definitions with input files, WebMO offers a Z-matrix editor for adjusting the order and definition of internal coordinates and a Cartesian coordinate editor for facilitating construction of unit cells in periodic systems.

WebMO's editor can calculate and minimize molecular mechanics energies, calculate and display extended Huckel molecular orbitals, determine point groups and display symmetry elements directly in the browser. It can also build

FIGURE 2 WebMO 3-D editor. The WebMO editor allows users to easily create or import molecular structures. Atoms and bonds are added by clicking and dragging. Structures are cleaned-up using Lewis valences, VSEPR structures, and idealized or mechanics-based geometries. Bond lengths, bond angles, and dihedral angles may all be individually specified. Mechanics energy, symmetry, and extended Huckel molecular orbital calculations are available within the editor



surfaces and solids by replicating unit cells and creating slabs along Miller planes to support periodic boundary condition calculations.

Although not the primary purpose of WebMO, the Editor can also look up reference data for the drawn molecule from a variety of external databases including PubChem, ChemSpider, NIST WebBook, Sigma-Aldrich, NMRShiftDB, and SDBS Spectral Database. Thus WebMO can serve as a portal for accessing and displaying safety information, physical properties, thermodynamic data, chemical reactivity, and spectral information (NMR, IR, UV-VIS, MS), and literature references.

2.4 | Calculations types and job options

WebMO supports a wide variety of common calculation types including: single point energy, geometry optimization, vibrational frequencies, coordinate scans, electronically excited states, molecular spectra (infrared, Raman, UV-VIS, NMR), molecular orbitals, natural bond orbitals, transition states, and other calculations specific to each computational engine. These calculation types can be extended by adding to the underlying calculation templates that define each calculation type. WebMO also provides edit access to the input file prior to submission, allowing for inspection and customization of each job.

It is important to recognize that WebMO does not perform these calculations itself. Rather, it prepares the input file for an installed engine based on the selected calculation type and options, runs the job, and parses the output file. Thus the accuracy and performance of the calculation is fully dependent on the underlying program and selected options.

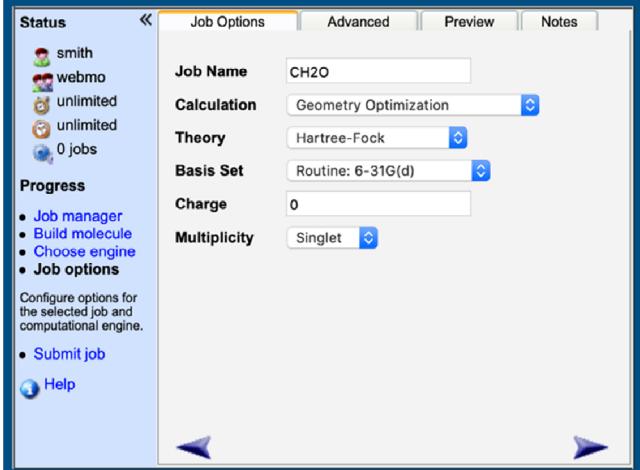
For each calculation type, the user can select various job options such as theory, basis, solvation model, and so forth (Figure 3). The available options vary by choice of computational engine and calculation type. There are also options that control how the job is to be run, such as number of cores and selected queue. Option values are typically specified using drop-down menus, though user-specified values can also be provided. In order to not overwhelm the user with a multitude of non-relevant job options, only those options that are pertinent to a particular job type are displayed.

Input files are created using a template system. For each engine, WebMO includes a pre-defined template for each supported calculation type. The templates can be relatively simple, for example, substitute job option variable values into the calculation type template, or they can include a variety of programming structures such as conditional statements, loops, and so forth. Additional templates can be defined by the user for additional calculation types or customized options, thereby extending the capabilities of WebMO. New template variables can be defined along with their allowed values. Template creation and variable definitions are implemented using the web-based template manager provided within WebMO.

2.5 | Visualization

Beyond simplifying the creation of input files, WebMO parses the output file and displays calculated information in tables, plots, and interactive images.

Configure Gaussian Job Options



Configure Gaussian Job Options

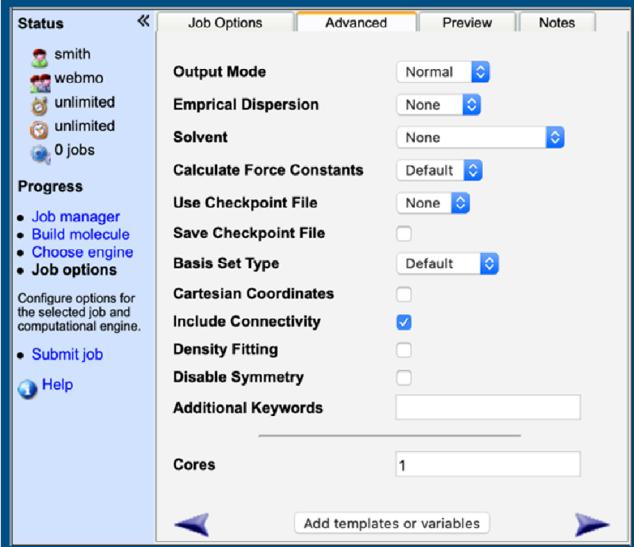


FIGURE 3 Calculation job options. Basic job options are specified similarly for all calculations. Advanced job options are optional and depend on calculation type and engine

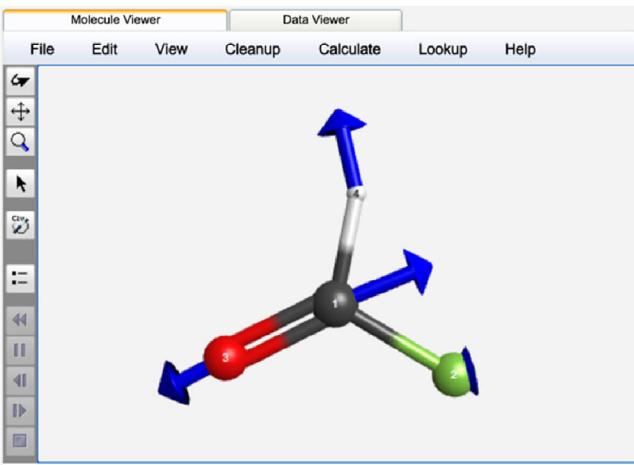
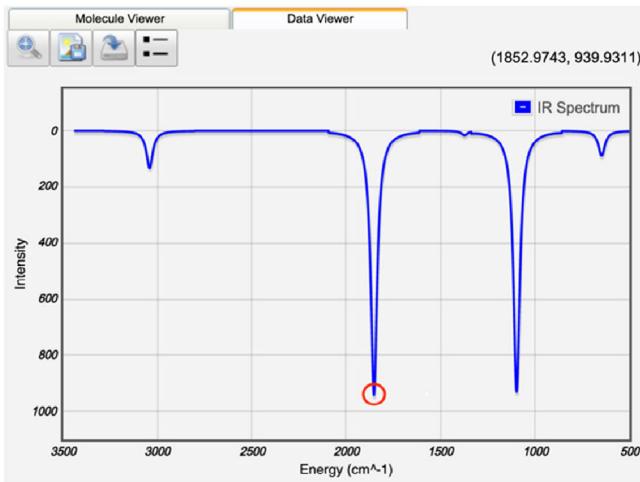


FIGURE 4 Interactive spectra. Clicking on a peak in an infrared (or NMR) spectrum will display the corresponding molecular vibration (or nuclei)

The optimized geometry is displayed in a 3-D viewer, which functions similar to the 3-D editor. Individual bond lengths, angles, and dihedrals can be measured. Calculated numeric values (energy, partial charges, dipole moment, thermodynamic quantities, vibrational frequencies, orbital energies, etc.) are displayed in tables. They can often also be superimposed on the molecular structure (partial charges, dipole moment, normal vibrational modes, symmetry elements), or they may be displayed as separate plots (coordinate scan, molecular spectra), as illustrated in Figure 4.

WebMO plots are interactive. Clicking on a plot location displays the corresponding feature on the molecule, for example, clicking on a vibrational peak displays the corresponding normal mode, or clicking on a coordinate scan value displays the corresponding molecular geometry. This feature is extremely useful for teaching, as it visually indicates the link between computed values and molecular structure.

Molecular orbitals can be displayed on top of the molecular structure as isosurfaces or as color-mapped values on a slice plane (Figure 5). Additional quantities can be calculated and displayed as color-mapped values on electron density surfaces, including electrostatic potential surfaces and electrophilic/nucleophilic frontier surfaces.

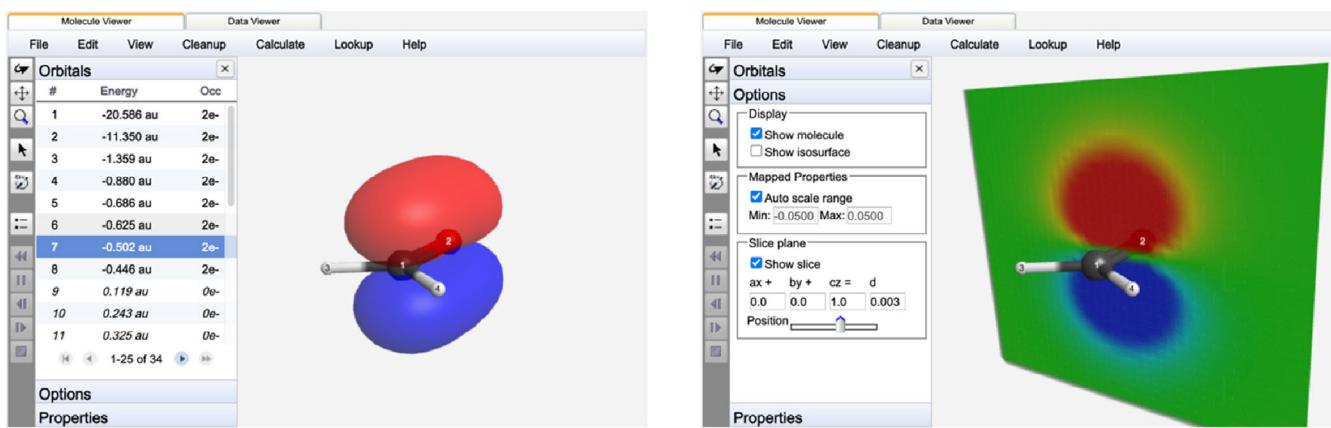


FIGURE 5 Visualizing molecular orbitals. WebMO can display molecular orbitals as phase-indicated isosurfaces, or as color-coded values on a scrollable cross-section plane

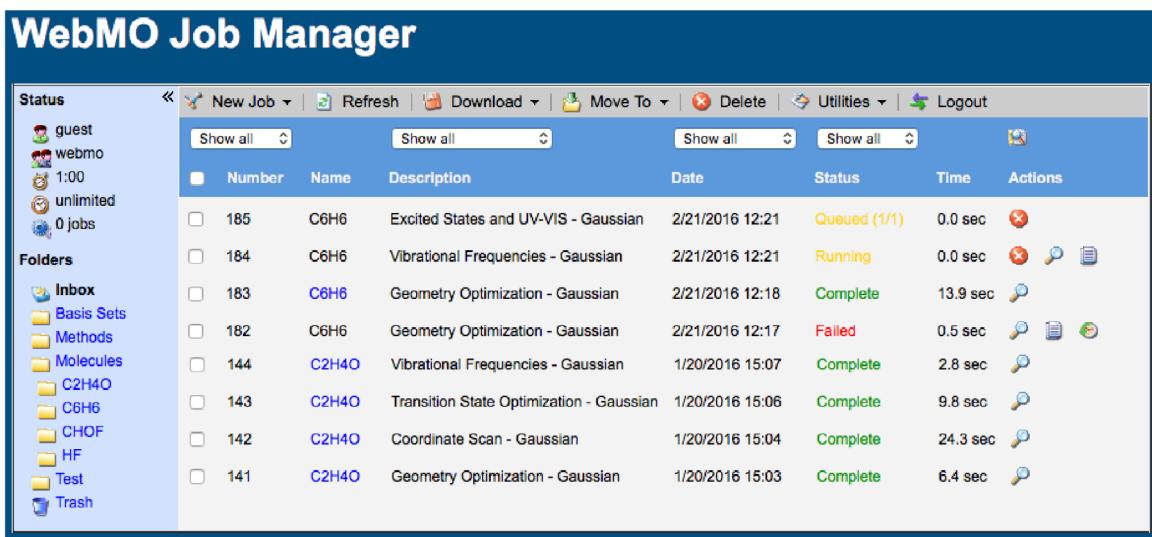


FIGURE 6 Job manager. The WebMO job manager allows users to create, monitor, and view jobs. One can also organize jobs into folders, import legacy calculations, and export results

Beyond visualization in a web browser, information from WebMO can be downloaded for visualization or analysis by other programs. Molecular geometries can be exported in a wide variety of formats, orbitals as cube files, and periodic solids as cif files. The results of coordinate scan and spectra calculations can be downloaded as text files, and all parsed values from the output file can be downloaded as an Excel spreadsheet. Output files can be downloaded for specialized analysis by programs like AIM,^{57–59} MultiWfn,⁶⁰ and NBO.^{61–63} WebMO molecular images (molecules, orbitals, surfaces, slices) can be exported as png image files for web purposes or as POV-Ray files from which publication quality images can be rendered.

2.6 | Job management

WebMO contains a job manager that functions like an email inbox for computational chemistry jobs (Figure 6). The job manager lists queued, running, and completed jobs, along with an indicator of whether or not the job successfully completed. Jobs may be viewed, searched, downloaded, or deleted. Notes can be added to individual jobs, and groups of jobs can be organized into user-defined folders.

Computational chemistry calculations are frequently “stacked” so that the output geometry of one job becomes the input geometry for the next job for example, a geometry optimization may be followed by a vibrational frequency calculation. WebMO keeps track of the history of a sequence of calculations. WebMO can also create spreadsheet summaries of multiple jobs to tabulate compare their results.

2.7 | Administration

All WebMO administration is performed within its menu-based web interface.

For system configuration, the WebMO administrator specifies the location of installed computational programs and system settings for each program (scratch directory, number of available cores, etc.) While WebMO includes a first-in first-out queueing system, WebMO can also be configured to use an external batch queuing system (Torque/PBS, SLURM, etc.).

For user management, a WebMO administrator can add, edit, disable, or delete WebMO users. WebMO users are internal to WebMO, which means that Linux user accounts do not need to be created for each user, though Linux users can be included as WebMO users if desired. Authentication of WebMO users is done either through a built-in password system, or via a variety of external authentication schemes (passwd, LDAP, Shibboleth, etc.). WebMO administrators can view, download, or delete jobs by WebMO users, as well as set CPU time limits on a per job or cumulative basis.

WebMO allows for user groups, each with a group subadministrator. These are particularly useful for having a single instance of WebMO support multiple courses or research groups, as an instructor (subadministrator) can add students and view jobs within their course (group) only. This organizational feature offloads the user and job management to subadministrators while keeping all system configuration settings with the WebMO administrator.

2.8 | WebMO app

WebMO apps⁶⁴ for iOS and Android mobile devices have been developed with similar capabilities as the web-based application, including a 3-D editor and structure importing from external databases. The WebMO app functions both as a standalone application and also as a client for a WebMO server. One can locally build molecules and optimize geometries, as well as calculate and display molecular mechanics energies, vibrational modes, Huckel MO's, and symmetry elements. The app can search a variety of external databases for chemical properties and spectral information. Jobs can also be submitted to a WebMO server, and previously run jobs can be viewed. Thus the WebMO app extends computational chemistry access from laptop computers to mobile smart devices. During 2020, there were 18,000 installations of the app on iOS and Android devices.

3 | TECHNOLOGY

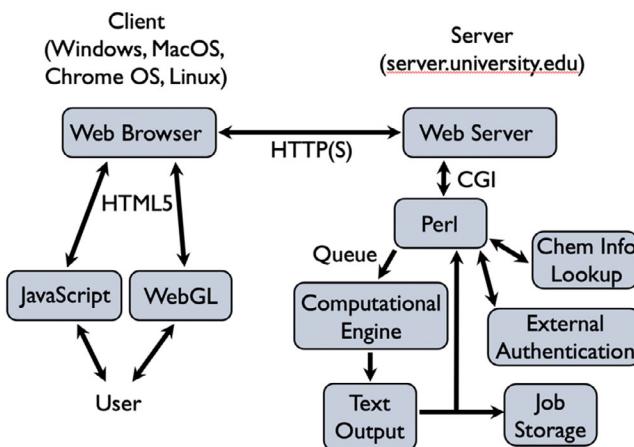
3.1 | Server-client model

WebMO uses a variety of web technologies split between the WebMO server on which the software is installed and the web browser clients on user computers (Figure 7). Communication between the WebMO server and web browser clients is done using standard or encrypted HTTP(S).

3.2 | Client-side

The client-side of WebMO displays web pages written in HTML5 and JavaScript. User interaction is implemented via HTML forms and JavaScript. The 3-D molecular editor uses a combination of JavaScript and WebGL (a web implementation of OpenGL) to build, render, and interact with molecular structures. Computationally inexpensive calculations such as molecular mechanics optimizations, Huckel calculations, and isosurface maps are performed directly in the browser using JavaScript. These technologies are supported by all modern web browsers and computing platforms, and

FIGURE 7 WebMO technology. WebMO uses a web-based client–server model. Perl scripts are installed on a server. Web pages are delivered to the user and provide interactivity with JavaScript and 3-D visualization with WebGL. Jobs are submitted to engines via an internal or external queue, their progress is monitored, and output is stored for future viewing. WebMO can interact with external services for user authentication and chemical information lookup



they allow interactive visualization and modest calculations without the need for Java applets, helper applications, plug-ins, or interaction with the WebMO server.

3.3 | Server-side

After preparing a molecule and configuring job options, the client submits the job to the WebMO server. The server-side of WebMO consists of Perl-based CGI scripts that are executed via the web server. These scripts handle the vast majority of WebMO operations, including generation of dynamic HTML content for the client, creating input files for computational programs, starting computational chemistry jobs, monitoring running jobs, parsing job output files, and job/user/engine management. WebMO Enterprise has additional scripts that facilitate the interaction of WebMO with external authentication options (LDAP, Active Directory, Shibboleth, etc.) to integrate WebMO with pre-existing departmental or campus infrastructure. It is important to note that WebMO user accounts are separate from Linux system user accounts, thereby allowing access to computational resources in a controlled fashion without having to create Linux user accounts for each WebMO user. The Enterprise version also supports existing HPC clusters via integration with existing batch queue systems (Torque/PBS, SLURM, SGE, etc.). The WebMO server stores configuration information in its home directory on the server computer and user job files in its directory, system user home directories, or on a network drive.

WebMO is distributed as a complete package that is installed on a Linux, MacOS, or Windows computer that has a web server and Perl. This server computer frequently also has the computational programs being used, although WebMO can access computational programs on remote servers via SSH or on computer clusters via their batch queue system. Installation is performed via a single script that copies all files to their correct locations and performs various configuration and compatibility checks. Upgrades are installed via a single command that updates the codebase while preserving all existing configuration settings, user accounts, and previously run jobs. The web-based architecture means that once the server is upgraded, all clients are also automatically upgraded.

4 | EDUCATION ACROSS THE UNDERGRADUATE CURRICULUM

4.1 | Accessibility and versatility

Since a user can access their WebMO account from a laptop computer or mobile device without installing additional software, WebMO is an excellent platform for deploying computational chemistry within educational environments. Support for groups (courses) and subadministrators (instructors), single sign-on external authentication, and automated WebMO account creation simplify the administration of WebMO for large numbers of students. WebMO's uniform interface for all computational chemistry programs simplifies user training. As a result, WebMO enables computational chemistry to be a versatile tool used across the curriculum for modeling chemical properties and reactivity, similar to spectroscopy for materials characterization or spreadsheets for data analysis.

WebMO has been successfully used throughout the undergraduate curriculum to introduce students to modern computational chemistry. Beyond providing a convenient and accessible platform for students to run computational chemistry calculations, WebMO has the ability to produce high quality images in png or POV-Ray format, which can be used to produce instructional material. WebMO can also produce dynamic and interactive materials using its HTML export feature, which creates a fully interactive View Job page including the 3-D molecular viewer and interactive spectra. The HTML exported files can be placed on any web server, making it straightforward for students investigate pre-computed results without having run the calculations themselves.

Several collections of computational chemistry exercises exist, which contain a wide variety of computational chemistry exercises suitable for the undergraduate level.^{65–68} Specific applications for various courses and literature references for use with WebMO follow.

4.2 | General chemistry

WebMO can be used for computational chemistry activities in General Chemistry lecture and laboratory courses. Examples of applications (and corresponding WebMO calculation type) include^{69–72}:

- Ideal (3-D editor) versus actual (geometry optimization) VSEPR molecular shapes (Figure 8).
- 3-D molecular structures (import by name into 3-D editor).
- Molecular properties (external database portal).
- Dipole moments (energy calculation, Figure 9).
- Atomic orbital energies and shapes (atomic orbitals).

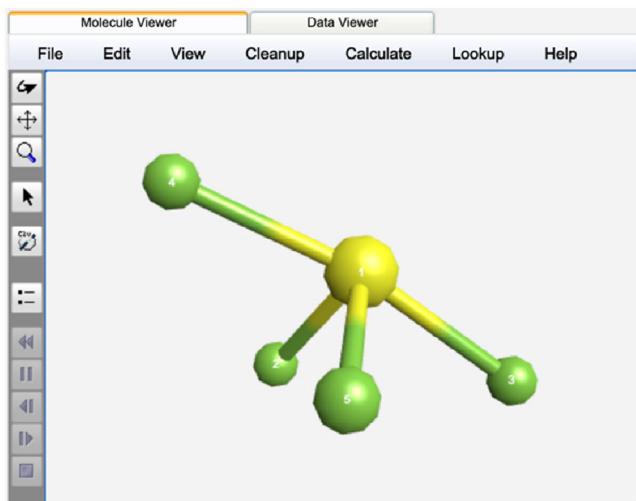


FIGURE 8 VSEPR structures. “Seesaw” VSEPR molecular shape of SF₄

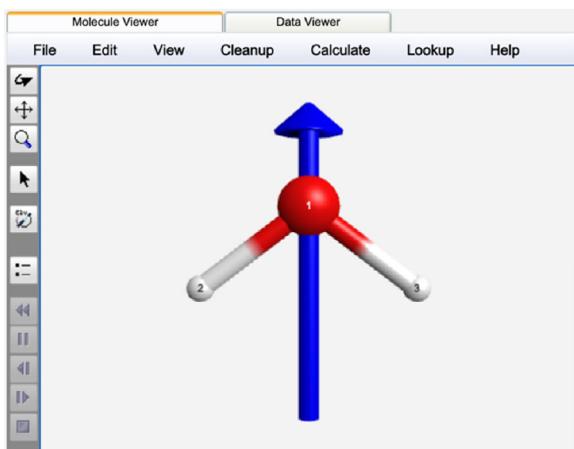


FIGURE 9 Dipole moments. Dipole moment of H₂O

- Periodic Trends (atomic orbitals).
- Formal charge and partial charges (partial charges).
- Diatomic bonding (homonuclear molecular orbitals).
- Molecular orbitals (extended molecular orbitals).

4.3 | Organic chemistry

WebMO can be used to provide computational and visual explanations of many concepts in organic chemistry courses and laboratories^{72–83}:

- Conformational energies (coordinate scans, Figure 10).
- Steric interactions (molecular mechanics strain energy).
- Overlap of atomic orbitals and valence bonding (natural bonding orbitals).
- Molecular reactivity (natural bonding orbitals, transition state calculations, IRC calculations).
 - Nucleophilic reaction pathways.
 - Endo versus exo cycloaddition.
 - Effect of electron donating and withdrawing groups.
 - Hybridization and basicity of lone pairs.
- Characteristic vibrational frequencies (interactive IR spectrum).
- Simulation of NMR spectra (NMR spectrum, Figure 11).
- Visualization of chirality (3-D editor, VCD spectra).
- Site of attack (nucleophilic/electrophilic density maps).

FIGURE 10 Conformational energy scan. Coordinate scan about the C1–C2–C3–C4 dihedral angle of butane at PM3 model chemistry, illustrating anti, gauche, and eclipsed energetics. Coordinate scan plots are interactive, so that clicking on an energy point displays the corresponding molecular geometry

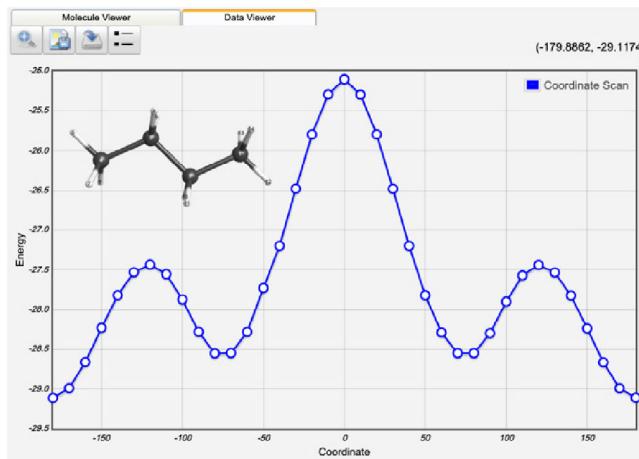
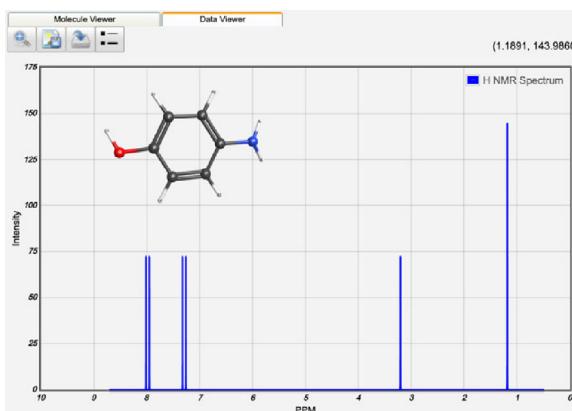


FIGURE 11 Computed NMR spectrum of 4-aminophenol. TMS offset and first-order splitting are implemented by WebMO. NMR spectra are interactive, as clicking on a peak highlights the corresponding nuclei



4.4 | Physical chemistry

WebMO can be used to compute a wide variety of thermodynamic and spectroscopic quantities, thereby enhancing the connection between experiment and theory^{84–95}:

- Reaction enthalpies (frequency and thermodynamic calculations).
- Reaction rates (geometry optimization, transition state optimization).
- Vibrational normal modes and spectra (IR and Raman spectra).
- Molecular orbital energies (molecular orbital calculation, Figure 12).
- Molecular bonding (molecular orbital calculation).
- Electronic transitions and color (UV–VIS spectra).
- Natural optical activity (optical rotation spectra).
- Potential energy curve calculations (coordinate scan, Figure 13).
- Basis sets and theoretical methods (spreadsheet export).

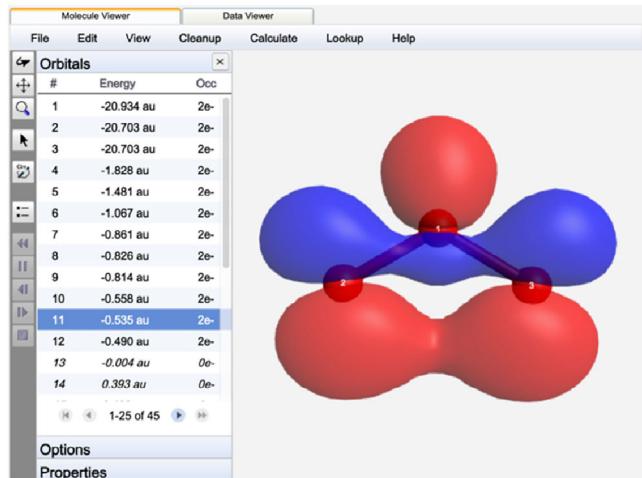


FIGURE 12 Molecular orbital of ozone. Geometry optimization, molecular orbitals, and NBO bonding calculation illustrate in-plane π -bonding results in a partial bond between the outer oxygen atoms and reduces the bond angle to less than 120°

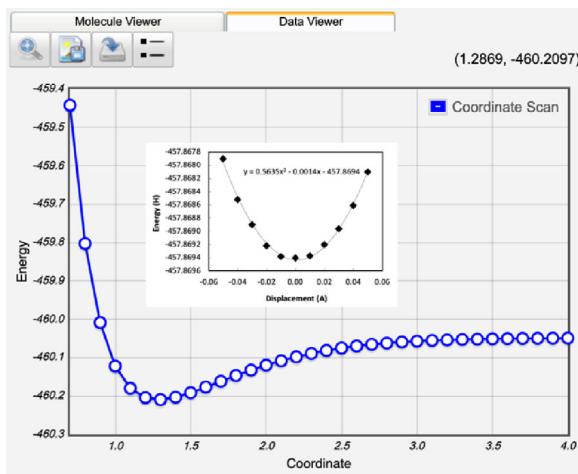


FIGURE 13 Potential energy curve of HCl. Coordinate scan of bond length at CCSD/6-31G(d) model chemistry. Additional points at the bottom of the potential can be exported to Excel and fit to a parabola, from which the spring constant ($k = 2 \cdot 0.5937 \text{ Ha/Å}^2 = 517.7 \text{ Nm}^{-1}$) and vibrational frequency ($\tilde{\nu} = 2995 \text{ cm}^{-1}$) can be calculated

4.5 | Inorganic chemistry

WebMO can be used to illustrate concepts related to molecular bonding and symmetry^{96–99}:

- Determine point groups and display symmetry elements (symmetry calculation, Figure 14).
- Compute and visualize molecular orbitals (molecular orbital calculation, Figure 15).
- Molecular spectra (NMR, IR, and UV–Vis spectra).
- Visualize overlap between central atom atomic orbitals and ligand molecular orbitals (molecular orbitals).
- Unit cells (PBC editor).

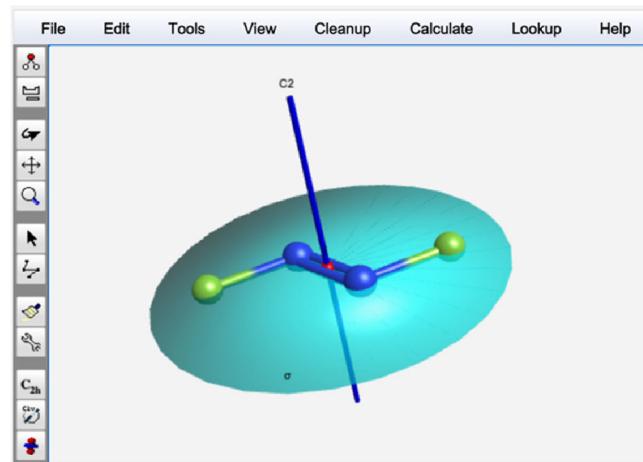


FIGURE 14 Symmetry elements. WebMO can calculate the point group of molecular structures and display their symmetry elements. WebMO can also symmetrize nearly symmetric structures

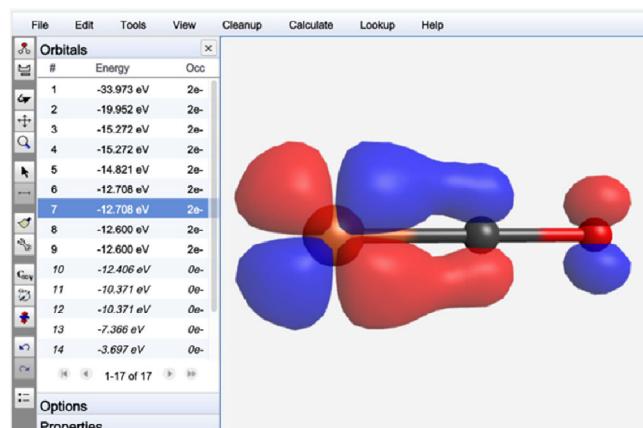
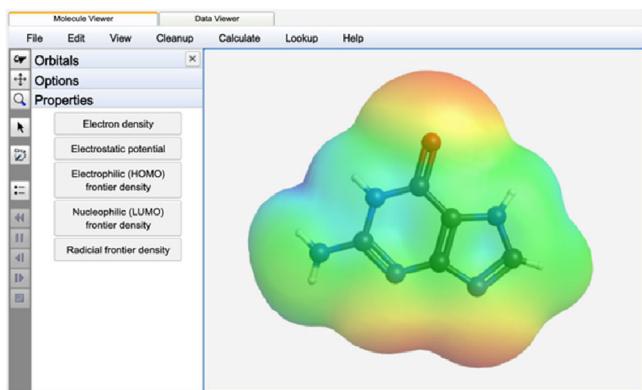


FIGURE 15 Backbonding. Overlap of a metal d-orbital and CO π^* -bond is easily computed and visualized using WebMO

4.6 | Biochemistry

While WebMO is an interface to quantum chemistry calculations, it can be used to visualize and study small amino acid sequences and ligand-binding site interactions:

- Polypeptide geometry (geometry optimization).
- Ligand conformation (geometry optimization).
- Ramachandran plots (2-D coordinate scan).
- Electrostatic interactions (electrostatic potential map, Figure 16).
- Zwitterion stability (solvation energy).

**FIGURE 16** Electrostatic potential surface of guanine.

Electrostatic potential surface is computed from occupied molecular orbitals and displayed with a color-code map on an electron density isosurface. Sites for hydrogen bonding are readily visible

4.7 | Analytical chemistry

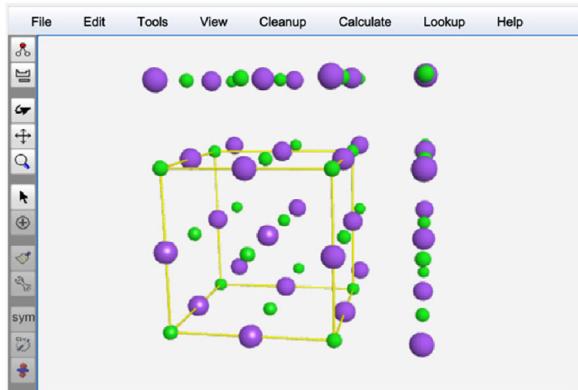
WebMO can be used to study binding interactions and spectroscopy used to separate and identify chemical species¹⁰⁰:

- Dipole moments (energy calculation).
- Electrostatic interactions (electrostatic potential map).
- Compound identification (IR and UV–VIS spectra).

4.8 | Materials chemistry

WebMO can be used to model nanotubes, 2-D materials, and organic semi-conductors (Figure 17)^{101–103}:

- Band gap (energy calculation).
- Strain energy (geometry optimization).
- Vibrational motions (IR spectra).

**FIGURE 17** Modeling solids and materials. The WebMO Cartesian coordinate editor supports definition of unit cells, creation of supercells, and creation of slabs along Miller planes. The crystal structure of NaCl is illustrated here

4.9 | Computational chemistry

WebMO provides an ideal hands-on platform for an electronic structure theory course.^{68,92} Students have immediate access to available electronic structure programs, can inspect and customize their input files, and have full access to the output files.

5 | ADDITIONAL USE CASES

5.1 | Research

Although WebMO was originally conceived to support use of computational chemistry in education, WebMO is also widely used by experimentalists in academia and industry to perform theoretical calculations in support their research. Hundreds of published research papers reference WebMO as the user interface for performing computational chemistry calculations.¹⁰⁴

The ease-of-use in carrying out computational chemistry calculations afforded by WebMO allows non-theoretical chemists to effectively use computation to understand and interpret their research. Beyond simplifying input file creation, calculation type specification, job monitoring and management, and result interpretation, WebMO offers features and flexibility to support research-level work. Full edit access to the input file, customizable job templates for specialized applications, the ability to import legacy calculations, automatic job history tracking, and powerful visualization capabilities allow non-theoretical researchers to readily use computation to supplement and inform their research.

The client–server architecture of WebMO allows a single implementation of WebMO to be deployed for an entire research group, department, institution, or company. This permits a computational chemistry investment to be leveraged across a large number of users, thereby maximizing the benefit of the implementation.

5.2 | Pedagogical support

Technology is increasingly being used to support student learning. Because WebMO allows chemical computation to be used over the internet and with a wide variety of student hardware, it has been regularly employed in collaborative and distance learning environments.^{84,105} The exploratory nature of computational exercises has resulted in WebMO being used in Process Oriented Guided Inquiry Learning (POGIL) exercises.^{89,94,106} Most recently, WebMO was used by a number of institutions in response to the COVID-19 outbreak to teach chemical concepts online.^{107–109}

5.3 | Computational chemistry consortia

WebMO is used by various academic consortia of computational chemists to share computational resources. Its uniform and accessible framework enables groups of computational chemistry users from different institutions to benefit from collective resources and interaction.

In the early 2000's, the Computers in Chemistry at Cabrillo College¹¹⁰ (C4) consortium supported seven community colleges in using computational chemistry in their organic chemistry curriculum. Cabrillo College hosted a WebMO instance in a self-described “hub-and-spoke” model that students and faculty from other community colleges could access. More recently, the University of Wisconsin at Madison organized an outreach program whereby regional institutions can run calculations to supplement experimental laboratory exercises⁷⁷ or access pre-computed results created with the WebMO HTML export feature.^{72,78}

To support research, the Midwestern Undergraduate Computational Chemistry Consortium¹¹¹ (MU3C) acquired three NSF grants for computing infrastructure that offer computing resources to primarily undergraduate institutions for research purposes using WebMO as the common interface. iOpenShell¹¹² is a consortium of experimental spectroscopists and theoretical chemists who use WebMO to access state-of-the-art computer codes that model experimental observations involving electronically excited states. ACENET¹¹³ and WestGrid¹¹⁴ are consortia of Canadian universities providing high-performance computing resources, visualization, and collaboration tools to participating research institutions, and they use WebMO as the common platform by which their research community can access computational chemistry programs.

5.4 | Curriculum development consortia

A wide variety of high school, undergraduate, and graduate groups are developing educational materials that use computation to teach chemistry concepts.

At the high school level, the Shodor Education Foundation¹¹⁵ has been a national leader in introducing computation into the curriculum. They have developed pedagogical materials to support teaching of computational chemistry on many platforms including WebMO.¹¹⁶

At the undergraduate level, several consortia of educators organize regular meetings and have websites that distribute educational materials for broader use. The Molecular Computation and Visualization in Undergraduate Education¹¹⁷ (MoleCVUE) consortium is an association of predominantly undergraduate institution faculty that develops and employs modern computational and visualization tools, including educational exercises that use WebMO. Psi4Education¹¹⁸ is an education and outreach user group that develops tutorials and computational laboratories using the freely-available Psi4 program together with WebMO as the user interface. The Virtual Inorganic Pedagogical Electronic Resource⁹⁹ (VIPER) focuses on developing and distributing vetted teaching resources, including many computational chemistry activities and laboratories that utilize WebMO.

5.5 | Computational chemistry workshops

WebMO is a convenient platform for workshops to train high school and college faculty on the use of computational chemistry. Participants do not need any specific software or hardware beyond a web browser on their laptop.

The National Computational Science Institute¹¹⁹ (NCSI) annually sponsors week-long workshops that train chemical educators in the theory and practice of computational chemistry. WebMO LLC regularly runs half-day “hands-on” WebMO workshops at the Biennial Conference on Chemical Education (BCCE) and regional chemistry meetings.^{120–123}

6 | OUTLOOK

6.1 | Usage

WebMO has widespread usage with over 4300 unique registered implementations, each of which can support thousands of individual users and hundreds of thousands of jobs (depending on local memory and disk resources). For example, the WebMO Demo Server, which is open to the world for running jobs up to 30 s of CPU time, has run over 800,000 jobs since its inception.¹²⁴

While installation on a local server and access via a web browser is the predominant usage model, there are several directions in which WebMO is evolving to utilize new computing platforms and integrate with new access methods.

6.2 | Cloud implementation

Cloud computing provides computer hardware, application software, and storage at a remote location that is accessible via the internet. Currently the major providers of cloud computing are Amazon Web Services,¹²⁵ Google Cloud Platform,¹²⁶ and Microsoft Azure.¹²⁷ Many organizations are finding it cost-beneficial to move from locally-hosted to cloud computing, and the global cloud computing market is expected to grow from USD 371 billion in 2020 to 832 billion in 2025, a compound annual growth rate of 17.5%.¹²⁸

While cloud computing is a commercial service, the cost can be much less than purchasing hardware when the instance is not running full-time, for example, only during the offering of a computational chemistry experiment in a laboratory course. The cloud also has other advantages over purchasing local computing resources, such as the ability to adjust computer type (CPU's, RAM, disk), zero initial capital outlay, no hardware maintenance, and no space requirements.

WebMO offers an automated installation of WebMO on all three popular cloud computing platforms through its Server In The Cloud (SITC) package.¹²⁹ SITC enables WebMO usage on flexible, low-cost cloud platforms by users with minimal system administration expertise. One creates a virtual machine in the cloud with the desired computer type and Linux OS distribution, downloads the latest version of SITC from WebMO, and runs the install script. By default SITC will then update the operating system, configure the web server, download and install WebMO, create a WebMO user, and install mopac7 as an engine. The SITC script can also perform updates and install other engines.

6.3 | WebMO as a service

The WebMO Demo Server¹²⁴ currently offers free access to WebMO Enterprise and most major computational engines so that users can experience WebMO and verify its capabilities. Since the Demo Server is an openly accessible resource, compute time is limited to 30 CPU seconds per job, and jobs are automatically deleted in 30 days.

Many course instructors have requested the ability to use a pre-configured and remotely-hosted instance of WebMO, with no software to install and no hardware to maintain. To accommodate this need, WebMO is developing WebMO as a Service (WaaS). WaaS provides WebMO access, job control, and data storage to users with no hardware, system configuration, software installation, or administrative expertise. Students and faculty simply login and immediately begin using a fully-functional instance of WebMO.

WaaS consists of individual WebMO instances that are pre-installed on a dynamic cluster running in the cloud, which allows for expansion and contraction of the compute nodes in response to usage.^{130–132} WebMO sub-administrator accounts are created for each faculty instructor, who can then create user accounts for their students. Compute engines are installed on the cluster and enabled as licensed by the instructor. Student jobs are handled by the SLURM batch queue system running on the cluster, which also serves to resize the cluster according to demand.

6.4 | Programmatic access and control

While administrative and user interaction with WebMO is typically done through a web browser, the client–server model allows for other interaction mechanisms that are compatible with HTTP communication. WebMO recently added a Representational State Transfer (REST) API that allows users to interact with WebMO from outside of a standard browser environment.¹³³ This API can be accessed either on the command line or more commonly via a Python program. The API currently allows programmatic access to information and results from previously run jobs and the ability to carry out basic user or job management functions. WebMO returns results from prior calculations in JavaScript Object Notation (JSON) format using an extension of the MolSSI QCSchema¹³⁴ standards. Future plans for the API include mechanisms to submit calculations to WebMO engines and return the results.

A particularly convenient way to implement programmatic access to WebMO is to use Jupyter notebooks, which create a virtual Python programming environment within a web-browser (Figure 18).¹³⁵ The details of accessing WebMO jobs using Jupyter, along with an annotated example, are provided on the WebMO website.¹³⁶ We anticipate that this programmatic interface, in conjunction with common Jupyter notebooks, will open up a variety of interesting additional applications of WebMO in upper level undergraduate and graduate courses.

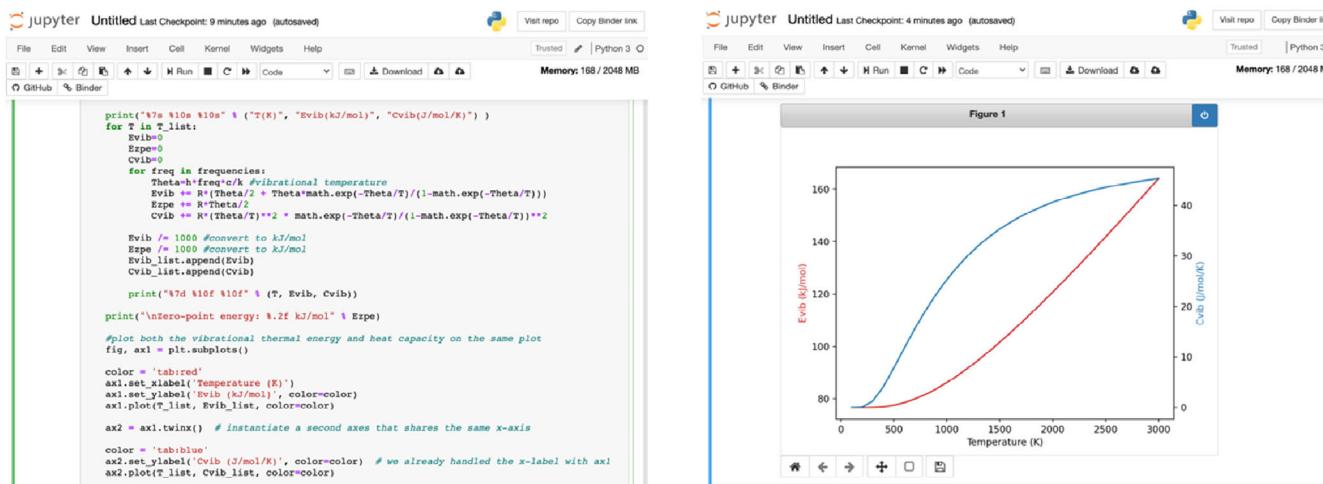


FIGURE 18 Using Jupyter notebooks to access and process computational results. Python code in a Jupyter notebook retrieves results from a WebMO vibrational frequency calculation, calculates the partition function, and plots the internal energy and heat capacity as a function of temperature

6.5 | Open-architecture, collaborations, and feedback

WebMO has been developed with an open-architecture, extensible approach. The server scripts are written and distributed in Perl, and the design allows new capabilities and engines to be added without modifying the main WebMO code.

The WebMO developers welcome feedback from WebMO users, ranging from bug reports, to feature suggestions, to sharing of code customizations. The WebMO Forum promotes discussion among users to share ideas and experience with each other and with the WebMO developers.¹³⁷ The WebMO developers also regularly interact with the developers of computational chemistry programs, so as to incorporate unique or recent features of their programs into the WebMO interface. The ability to extend support to new features and additional computational programs by simply editing the distributed scripts allows program developers to contribute to the WebMO codebase. Finally, the WebMO developers regularly participate in training workshops and interact with educators who develop computational chemistry exercises for use in high school, undergraduate, and graduate level courses.

CONFLICT OF INTEREST

William F. Polik and J. R. Schmidt are cofounders of WebMO, LLC.

AUTHOR CONTRIBUTIONS

William Polik: Conceptualization; methodology; writing-original draft. **J.R. Schmidt:** Conceptualization; methodology; writing-review & editing.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

William F. Polik  <https://orcid.org/0000-0003-0419-5135>

J. R. Schmidt  <https://orcid.org/0000-0002-1438-117X>

RELATED WIREs ARTICLES

[Q-Chem: an engine for innovation](#)

[NWChem: scalable parallel computational chemistry](#)

[The ORCA program system](#)

[Molpro: a general purpose quantum chemistry program package](#)

[Psi4: an open-source ab initio electronic structure program](#)

REFERENCES

1. U.S. Presidential Information Technology Advisory Committee. Computational science: ensuring America's competitiveness. Arlington, VA: National Coordination Office for Information Technology Research and Development; 2005.p. 1.
2. Cramer CJ. Essentials of computational chemistry: theories and models. 2nd ed. Chichester, West Sussex, England; Hoboken, NJ: Wiley; 2004.
3. Jensen F. Introduction to computational chemistry. 3rd ed. Chichester, UK; Hoboken, NJ: Wiley; 2017.
4. Computational Chemistry. <https://www.chem.pitt.edu/newsletter/FY2016/Computational-Chemistry>. Accessed Feb 2021.
5. Top 500 Supercomputer Sites. <https://top500.org/>. Accessed Feb 2021.
6. Committee on Professional Training. ACS guidelines and evaluation procedures for Bachelor's degree programs. Washington, DC: American Chemical Society; 2015.p. 8 (section 4.3).
7. GaussView 6. <https://gaussian.com/gaussview6/>. Accessed Feb 2021.
8. IQmol Molecular Viewer. <http://iqmol.org/>. Accessed Feb 2021.
9. GMOLPRO: a graphical user interface. <https://www.molpro.net/info/release/doc/manual/node918.html>. Accessed Feb 2021.
10. Spartan. <https://www.wavefun.com/spartan>. Accessed Feb 2021.
11. Avogadro. <https://avogadro.cc/>. Accessed Feb 2021.
12. Gabedit. <http://gabedit.sourceforge.net/>. Accessed Feb 2021.
13. Chemcraft. <https://www.chemcraftprog.com/>. Accessed Feb 2021.
14. Jmol. <http://jmol.sourceforge.net/>. Accessed Feb 2021.
15. Molden. <https://www3.cmbi.umcn.nl/molden/>. Accessed Feb 2021.
16. Molekel. <https://ugovaretto.github.io/molekel/>. Accessed Feb 2021.
17. WebMO, LLC. <https://www.webmo.net/>. Accessed Feb 2021.

18. WebMO Feature Comparison. <https://www.webmo.net/features/compare.html>. Accessed Feb 2021.
19. McQuarrie DA. Quantum chemistry. 2nd ed. Sausalito, CA: University Science Books; 2008.
20. GAMESS. <https://www.msg.chem.iastate.edu/gamess/>. Accessed Feb 2021.
21. Barca GMJ, Bertoni C, Carrington L, Datta D, De Silva N, Deustua JE, et al. Recent developments in the general atomic and molecular electronic structure system. *J Chem Phys.* 2020;152:154102. <https://doi.org/10.1063/5.0005188>.
22. Gaussian, Inc. <https://gaussian.com/>. Accessed Feb 2021.
23. Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, et al. *Gaussian 16 Revision C.01*; 2016.
24. Molpro. <https://www.molpro.net/>. Accessed Feb 2021.
25. Werner H-J, Knowles PJ, Knizia G, Manby FR, Schütz M. Molpro: a general-purpose quantum chemistry program package. *WIREs Comput Mol Sci.* 2012;2:242–53. <https://doi.org/10.1002/wcms.82>.
26. Werner H-J, Knowles PJ, Manby FR, Black JA, Doll K, Heßelmann A, et al. The Molpro quantum chemistry package. *J Chem Phys.* 2020;152:144107. <https://doi.org/10.1063/5.0005081>.
27. MOPAC. <http://openmopac.net/>. Accessed Feb 2021.
28. Stewart JJP. MOPAC: A Semiempirical Molecular Orbital Program. *J Comput Aided Mol Des.* 1990;4:1–103. <https://doi.org/10.1007/BF00128336>.
29. NWChem. <https://nwchemgit.github.io/>. Accessed Feb 2021.
30. van Dam HJJ, de Jong WA, Bylaska E, Govind N, Kowalski K, Straatsma TP, et al. NWChem: scalable parallel computational chemistry. *WIREs Comput Mol Sci.* 2011;1:888–94. <https://doi.org/10.1002/wcms.62>.
31. Aprà E, Bylaska EJ, de Jong WA, Govind N, Kowalski K, Straatsma TP, et al. NWChem: past, present, and future. *J Chem Phys.* 2020;152:184102. <https://doi.org/10.1063/5.0004997>.
32. ORCA. <https://orcaforum.kofo.mpg.de/app.php/portal>. Accessed Feb 2021.
33. Neese F. The ORCA program system. *WIREs Comput Mol Sci.* 2012;2:73–8. <https://doi.org/10.1002/wcms.81>.
34. Neese F. Software update: the ORCA program system, version 4.0. *WIREs Comput Mol Sci.* 2018;8:e1327. <https://doi.org/10.1002/wcms.1327>.
35. Neese F, Wennmohs F, Becker U, Riplinger C. The ORCA quantum chemistry program package. *J Chem Phys.* 2020;152:224108. <https://doi.org/10.1063/5.0004608>.
36. PQS. <http://www.pqs-chem.com/>. Accessed Feb 2021.
37. Baker J, Janowski T, Wolinski K, Pulay P. Recent developments in the PQS program. *WIREs Comput Mol Sci.* 2012;2:63–72. <https://doi.org/10.1002/wcms.80>.
38. PSI4. <https://psicode.org/>. Accessed Feb 2021.
39. Turney JM, Simmonett AC, Parrish RM, Hohenstein EG, Evangelista FA, Fermann JT, et al. Psi4: an open-source ab initio electronic structure program. *WIREs Comput Mol Sci.* 2012;2:556–65. <https://doi.org/10.1002/wcms.93>.
40. Smith DGA, Burns LA, Simmonett AC, Parrish RM, Schieber MC, Galvelis R, et al. Psi4 1.4: open-source software for high-throughput quantum chemistry. *J Chem Phys.* 2020;152:184108. <https://doi.org/10.1063/5.0006002>.
41. Q-Chem, Inc. <https://www.q-chem.com/>. Accessed Feb 2021.
42. Krylov AI, Gill PMW. Q-Chem: an engine for innovation. *WIREs Comput Mol Sci.* 2013;3:317–26. <https://doi.org/10.1002/wcms.1122>.
43. TeraChem. <http://www.petachem.com/products.html>. Accessed Feb 2021.
44. Seritan S, Bannwarth C, Fales BS, Hohenstein EG, Kokkila-Schumacher SIL, Luehr N, et al. TeraChem: accelerating electronic structure and ab initio molecular dynamics with graphical processing units. *J Chem Phys.* 2020;152:224110. <https://doi.org/10.1063/5.0007615>.
45. Seritan S, Bannwarth C, Fales BS, Hohenstein EG, Isborn CM, Kokkila-Schumacher SIL, et al. TeraChem: a graphical processing unit-accelerated electronic structure package for large-scale ab initio molecular dynamics. *WIREs Comput Mol Sci.* 2020;11:e1494. <https://doi.org/10.1002/wcms.1494>.
46. Tinker. <https://dasher.wustl.edu/tinker/>. Accessed Feb 2021.
47. Rackers JA, Wang Z, Lu C, Laury ML, Lagardère L, Schnieders MJ, et al. Tinker 8: software tools for molecular design. *J Chem Theory Comput.* 2018;14:5273–89. <https://doi.org/10.1021/acs.jctc.8b00529>.
48. Quantum Espresso. <https://www.quantum-espresso.org/>. Accessed Feb 2021.
49. Giannozzi P, Baseggio O, Bonfà P, Brunato D, Car R, Carnimeo I, et al. Quantum ESPRESSO toward the Exascale. *J Chem Phys.* 2020;152:154105. <https://doi.org/10.1063/5.0005082>.
50. VASP. <https://www.vasp.at/>. Accessed Feb 2021.
51. Hafner J. Ab-initio simulations of materials using VASP: density-functional theory and beyond. *J Comput Chem.* 2008;29:2044–78. <https://doi.org/10.1002/jcc.21057>.
52. American Mineralogist Crystal Structure Database. Chemical identifier resolver. <https://cactus.nci.nih.gov/chemical/structure>. Accessed Feb 2021.
53. American Mineralogist Crystal Structure Database. <http://rruff.geo.arizona.edu/AMS/amcsd.php>. Accessed Feb 2021.
54. Allinger NL. Conformational analysis. 130. MM2. A hydrocarbon force field utilizing V1 and V2 torsional terms. *J Am Chem Soc.* 1977;99:8127–34. <https://doi.org/10.1021/ja00467a001>.
55. Burkert U, Allinger NL. Molecular mechanics. Washington, D.C: American Chemical Society; 1982.

56. Largent RJ, Polik WF, Schmidt JR. Symmetrizer: algorithmic determination of point groups in nearly symmetric molecules. *J Comput Chem.* 2012;33:1637–42. <https://doi.org/10.1002/jcc.22995>.
57. AIMPAC. <https://chemistry.mcmaster.ca/aimpac/imagemap/imagemap.htm>. Accessed Feb 2021.
58. AIMAll. <http://aim.tkgristmill.com/index.html>. Accessed Feb 2021.
59. AIM2000. <http://www.aim2000.de/>. Accessed Feb 2021.
60. Multiwfn. <http://sobereva.com/multiwfn/>. Accessed Feb 2021.
61. NBO. <https://nbo6.chem.wisc.edu/>. Accessed Feb 2021.
62. Glendening ED, Landis CR, Weinhold F. Natural bond orbital methods. *WIREs Comput Mol Sci.* 2012;2:1–42. <https://doi.org/10.1002/wcms.51>.
63. Weinhold F. Discovering chemistry with natural bond orbitals. 1st ed. Hoboken, NJ: Wiley; 2012.
64. WebMO App. <https://www.webmo.net/features/app.html>. Accessed Feb 2021.
65. Hehre WJ, Burke LD, Shusterman AJ, Pietro WJ. Experiments in computational organic chemistry. Irvine, CA: Wavefunction, Inc; 1993.
66. Caffery ML, Dobosh PA, Richardson DM. Laboratory exercises using Hyperchem. Florida: Hypercube, Inc.; 1998.
67. Polik WF, Schmidt JR. WebMO user's guide. Holland, MI: WebMO LLC; 2003.
68. Foresman JB, Frisch A. Exploring chemistry with electronic structure methods. 3rd ed. Wallingford, CT: Gaussian; 2015.
69. Feller SE, Dallinger RF, McKinney PC. A program of computational chemistry exercises for the first-semester general chemistry course. *J Chem Educ.* 2004;81:283. <https://doi.org/10.1021/ed081p283>.
70. McNaught IJ. Testing and extending VSEPR with WebMO and MOPAC or GAMESS. *J Chem Educ.* 2011;88:421–5. <https://doi.org/10.1021/ed900038a>.
71. Martin CB, Vandehoef C, Cook A. The use of molecular modeling as “pseudoexperimental” data for teaching VSEPR as a hands-on general chemistry activity. *J Chem Educ.* 2015;92:1364–8. <https://doi.org/10.1021/ed500806h>.
72. Esselman BJ, Block SB. VSEPR-plus: correct molecular and electronic structures can Lead to better student conceptual models. *J Chem Educ.* 2019;96:75–81. <https://doi.org/10.1021/acs.jchemed.8b00316>.
73. Rowley CN, Woo TK, Mosey NJ. A computational experiment of the endo versus exo preference in a Diels–Alder reaction. *J Chem Educ.* 2009;86:199. <https://doi.org/10.1021/ed086p199>.
74. Foresman JB, Clarke DD. Substituent interactions in aromatic rings: student exercises using FT-NMR and electronic structure calculations. NMR spectroscopy in the undergraduate curriculum. Volume 1128. ACS Symposium Series; American Chemical Society; 2013. p. 189–204. <https://doi.org/10.1021/bk-2013-1128.ch012>.
75. Hill NJ, Bowman MD, Esselman BJ, Byron SD, Kreitinger J, Leadbeater NE. Ligand-free Suzuki–Miyaura coupling reactions using an inexpensive aqueous palladium source: a synthetic and computational exercise for the undergraduate organic chemistry laboratory. *J Chem Educ.* 2014;91:1054–7. <https://doi.org/10.1021/ed400531w>.
76. Hein SM, Kopitzke RW, Nalli TW, Esselman BJ, Hill NJ. Use of ^1H , ^{13}C , and ^{19}F -NMR spectroscopy and computational modeling to explore chemoselectivity in the formation of a Grignard reagent. *J Chem Educ.* 2015;92:548–52. <https://doi.org/10.1021/ed500700c>.
77. Zdanovskiaia MA, Schwarz CE, Habib AD, Hill NJ, Esselman BJ. Access to computational chemistry for community colleges via WebMO. *J Chem Educ.* 2018;95:1960–5. <https://doi.org/10.1021/acs.jchemed.8b00310>.
78. Esselman BJ, Ellison AJ, Zhou J. Computational chemistry in organic chemistry lecture using WebMO. In: 2019 CCCE Newsletter; 2019.
79. Esselman BJ, Hill NJ. Integrating computational chemistry into an organic chemistry laboratory curriculum using WebMO. Using computational methods to teach chemical principles. Volume 1312. Washington, D.C: ACS Symposium Series; American Chemical Society; 2019. p. 139–62. <https://doi.org/10.1021/bk-2019-1312.ch011>.
80. Schafer AGL, Yezierski EJ, Hartley CS. Understanding thermodynamic control in covalent self-assembly: a mixed synthetic-computational experiment for the undergraduate organic-chemistry laboratory. *J Chem Educ.* 2019;96:1230–5. <https://doi.org/10.1021/acs.jchemed.8b01046>.
81. Dood AJ, Dood JC, Cruz-Ramírez de Arellano D, Fields KB, Raker JR. Using the research literature to develop an adaptive intervention to improve student explanations of an SN1 reaction mechanism. *J Chem Educ.* 2020;97:3551–62. <https://doi.org/10.1021/acs.jchemed.0c00569>.
82. Esselman BJ, Hofstetter H, Ellison AJ, Fry CG, Hill NJ. SN1, E1, and E2 reactions of Tert-amyl compounds: improved analysis using computational chemistry and ASAP-HSQC NMR spectroscopy. *J Chem Educ.* 2020;97:2280–5. <https://doi.org/10.1021/acs.jchemed.0c00071>.
83. Pelter MW, Howell MT, Anderson C, Sayeed A. Computational activity to visualize stereoisomers in molecules with an Axis of chirality. *J Chem Educ.* 2020;97:754–9. <https://doi.org/10.1021/acs.jchemed.9b00934>.
84. Paniagua JC, Mota F, Solé A, Vilaseca E. Quantum chemistry laboratory at home. *J Chem Educ.* 2008;85:1288. <https://doi.org/10.1021/ed085p1288>.
85. Williams DL, Flaherty TJ, Alnasleh BK. Beyond Amax part 2: predicting molecular color. *J Chem Educ.* 2009;86:333. <https://doi.org/10.1021/ed086p333>.
86. Simpson S, Autschbach J, Zurek E. Computational modeling of the optical rotation of amino acids: an ‘in silico’ experiment for physical chemistry. *J Chem Educ.* 2013;90:656–60. <https://doi.org/10.1021/ed300680g>.

87. Simpson S, Van Fleet A, Zurek E. A computational investigation of a molecular switch. *J Chem Educ.* 2013;90:1528–32. <https://doi.org/10.1021/ed400278x>.
88. Nassabeh N, Tran M, Fleming PE. Dissociation of the ethyl radical: an exercise in computational chemistry. *J Chem Educ.* 2014;91:1248–53. <https://doi.org/10.1021/ed4007748>.
89. Reeves MS, Whitnell RM. New computational physical chemistry experiments: using POGIL techniques with ab initio and molecular dynamics calculations. Addressing the Millennial Student in Undergraduate Chemistry. Volume 1180. ACS Symposium Series; American Chemical Society; 2014. p. 71–90. <https://doi.org/10.1021/bk-2014-1180.ch006>.
90. Adams W, Sonntag MD. Development of a physical chemistry laboratory experiment to explore vibrational spectroscopy selection rules. Raman spectroscopy in the undergraduate curriculum. Volume 1305. Washington, D.C: ACS Symposium Series; American Chemical Society; 2018. p. 115–28. <https://doi.org/10.1021/bk-2018-1305.ch007>.
91. Adams W, Sonntag MD. Vibrational spectroscopy of Hexynes: a combined experimental and computational laboratory experiment. *J Chem Educ.* 2018;95:1205–10. <https://doi.org/10.1021/acs.jchemed.8b00135>.
92. Grushow A. A lab course in computational chemistry is not about computers. Using Computational Methods to Teach Chemical Principles. Volume 1312. Washington, D.C: ACS Symposium Series; American Chemical Society; 2019. p. 211–25. <https://doi.org/10.1021/bk-2019-1312.ch015>.
93. Morales C, Chen F. Exploration of substituent and isotope effects on reaction rates by a computational modeling experiment. *J Chem Educ.* 2019;96:792–6. <https://doi.org/10.1021/acs.jchemed.8b00651>.
94. Reeves MS, Berghout HL, Perri MJ, Singleton SM, Whitnell RM. How can you measure a reaction enthalpy without going into the lab? Using computational methods to teach chemical principles. Volume 1312. Washington, D.C: ACS Symposium Series; American Chemical Society; 2019. p. 51–63. <https://doi.org/10.1021/bk-2019-1312.ch005>.
95. Stocker KM. Using electronic structure calculations to investigate the kinetics of gas-phase ammonia synthesis. Using computational methods to teach chemical principles. Volume 1312. Washington, D.C: ACS Symposium Series; American Chemical Society; 2019. p. 21–32. <https://doi.org/10.1021/bk-2019-1312.ch003>.
96. Pritchard BP, Simpson S, Zurek E, Autschbach J. Computation of chemical shifts for paramagnetic molecules: a laboratory experiment for the undergraduate curriculum. *J Chem Educ.* 2014;91:1058–63. <https://doi.org/10.1021/ed400902c>.
97. Cleaver RJ, Raymond CC. Carboxyborate: a main group inorganic laboratory experience. Advances in teaching inorganic chemistry. Volume 2: Laboratory enrichment and faculty community. Volume 1371. Washington, D.C: ACS Symposium Series; American Chemical Society; 2020. p. 13–20. <https://doi.org/10.1021/bk-2020-1371.ch002>.
98. Johnson AR, Teaching NC. Molecular orbital theory better. Advances in teaching inorganic chemistry. Volume 1: Classroom innovations and faculty development. Volume 1370. Washington, D.C: ACS Symposium Series; American Chemical Society; 2020. p. 47–63. <https://doi.org/10.1021/bk-2020-1370.ch005>.
99. VIPER: virtual inorganic pedagogical electronic resource. <https://www.ionicviper.org/teaching-resources>. Accessed Feb 2021.
100. Pemberton AT, Magers DB, King DA. Integrated TGA, FTIR, and computational laboratory experiment. *J Chem Educ.* 2019;96:132–6. <https://doi.org/10.1021/acs.jchemed.8b00607>.
101. Simpson S, Lonie DC, Chen J, Zurek E. A computational experiment on single-walled carbon nanotubes. *J Chem Educ.* 2013;90:651–5. <https://doi.org/10.1021/ed3006067>.
102. Miller DP, Phillips A, Ludowieg H, Swihart S, Autschbach J, Zurek E. The computational design of two-dimensional materials. *J Chem Educ.* 2019;96:2308–14. <https://doi.org/10.1021/acs.jchemed.9b00485>.
103. Hoover GC, Dicks AP, Seferos DS. Upper-year materials chemistry computational modeling module for organic display technologies. *J Chem Educ.* 2021;98:305–811. <https://doi.org/10.1021/acs.jchemed.0c01325>.
104. webmo—Google Scholar. <https://scholar.google.com/scholar?q=webmo>. Accessed Feb 2021.
105. Evans MJ, Moore JS. A collaborative, wiki-based organic chemistry project incorporating free chemistry software on the web. *J Chem Educ.* 2011;88:764–8. <https://doi.org/10.1021/ed100517g>.
106. Whitnell RM, Reeves MS. Process oriented guided inquiry learning computational chemistry experiments: revisions and extensions based on lessons learned from implementation. Using computational methods to teach chemical principles. Volume 1312. Washington, D.C: ACS Symposium Series; American Chemical Society; 2019. p. 65–77. <https://doi.org/10.1021/bk-2019-1312.ch006>.
107. Chan BC, Baker JL, Bunagan MR, Ekanger LA, Gazley JL, Hunter RA, et al. Theory of change to practice: how experimentalist Teaching enabled faculty to navigate the COVID-19 disruption. *J Chem Educ.* 2020;97:2788–92. <https://doi.org/10.1021/acs.jchemed.0c00731>.
108. Fulfer KD, Wachter E, Muzyka JL, Demoranville LT, Fieberg JE, Haile JD, et al. #StayCentred: maintaining personal education at centre college during COVID-19. *J Chem Educ.* 2020;97:2783–7. <https://doi.org/10.1021/acs.jchemed.0c00726>.
109. Giordano AN, Christopher CR. Repurposing best teaching practices for remote learning environments: chemistry in the news and Oral examinations during COVID-19. *J Chem Educ.* 2020;97:2815–8. <https://doi.org/10.1021/acs.jchemed.0c00753>.
110. Computers in Chemistry at Cabrillo College. <http://c4.cabrillo.edu/index.shtml>. Accessed Feb 2021.
111. Midwest Undergraduate Computational Chemistry Consortium. <https://mu3c.chem.sites.carleton.edu/>. Accessed Feb 2021.
112. iOpenShell. <http://iopenshell.usc.edu/>. Accessed Feb 2021.
113. ACENET. <https://www.ace-net.ca/wiki/WebMO>. Accessed Feb 2021.
114. WestGrid. <https://www.westgrid.ca/support/software/webmo>. Accessed Feb 2021.
115. Shodor: A National Resource for Computational Science Education. <http://www.shodor.org/>. Accessed Feb 2021.
116. CCCE Labs Using WebMO. <http://www.computationalscience.org/ccce/about/labs/labs.php>. Accessed Feb 2021.

117. MoleCVUE. <https://sites.google.com/view/molecvue>. Accessed Feb 2021.
118. Psi4Education. <https://psi4code.org/posts/psi4education/>. Accessed Feb 2021.
119. National Computational Science Website. <http://www.computationalscience.org/>. Accessed Feb 2021.
120. Polik WF. WebMO hands-on workshop. Grand Valley State University; 2014.
121. Polik WF. Integrating computational chemistry throughout the curriculum using WebMO. Hillsdale College; 2018.
122. Polik WF. WebMO hands-on workshop. Wesleyan University; 2019.
123. Polik WF, Schmidt JR. Using WebMO throughout the chemistry curriculum. University of Notre Dame; 2019.
124. WebMO Demo Server. <https://www.webmo.net/demo/>. Accessed Feb 2021.
125. Amazon Web Services. <https://aws.amazon.com/>. Accessed Feb 2021.
126. Google Cloud Computing Services. <https://cloud.google.com/>. Accessed Feb 2021.
127. Microsoft Azure. <https://azure.microsoft.com/en-us/>. Accessed Feb 2021.
128. Research and Markets. Cloud Computing Market by Service Model (Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)), Deployment Model (Public and Private), Organization Size, Vertical, and Region—Global Forecast to 2025; Report ID 5136796; 2020.
129. Server In The Cloud (SITC). https://www.webmo.net/support/sitc_commands.html. Accessed Feb 2021.
130. Basilyan M, Gorman W, Binder K, Ma-Weaver A. Easy HPC clusters on GCP with Slurm <https://cloud.google.com/blog/products/gcp/easy-hpc-clusters-on-gcp-with-slurm>. Accessed Feb 2021.
131. Ma-Weaver A, Blasius A. HPC made easy: Announcing new features for Slurm on GCP. <https://cloud.google.com/blog/products/compute/hpc-made-easy-announcing-new-features-for-slurm-on-gcp>. Accessed Feb 2021.
132. Binder K, Gorman W. Deploy an auto-scaling HPC cluster with Slurm <https://codelabs.developers.google.com/codelabs/hpc-slurm-on-gcp#0>. Accessed Feb 2021.
133. WebMO Help—REST-JSON Interface. <https://www.webmo.net/link/help/RESTJSONInterface.html>. Accessed Feb 2021.
134. MolSSI—QCSchema. <https://github.com/MolSSI/QCSchema>. Accessed Feb 2021.
135. WebMO Help—Jupyter Notebooks. <https://www.webmo.net/link/help/JupyterNotebooks.html>. Accessed Feb 2021.
136. WebMO and Jupyter Notebooks. <https://www.webmo.net/support/jupyter.html>. Accessed Feb 2021.
137. WebMO Support Forum. <https://www.webmo.net/forum/index.php>. Accessed Feb 2021.

How to cite this article: Polik WF, Schmidt JR. WebMO: Web-based computational chemistry calculations in education and research. *WIREs Comput Mol Sci*. 2021;e1554. <https://doi.org/10.1002/wcms.1554>