VALENCE: A Massively Parallel Implementation of the Variational Subspace Valence Bond Method

Graham D. Fletcher,*[a] Colleen Bertoni,[b] Murat Keçeli,[c] and Michael D'Mello[d]

This work describes the software package, VALENCE, for the calculation of molecular energies using the variational subspace valence bond (VSVB) method. VSVB is an *ab initio* electronic structure method based on nonorthogonal orbitals. Important features of practical value include high parallel scalability, wave functions that can be constructed automatically by combining orbitals from previous calculations, and ground and excited states that can be modeled with a single configuration or determinant. The open-source software package includes tools to generate wave functions, a database of generic orbitals, example input files, and a library build intended for integration with other packages. We also describe the interface to an external software package, enabling the computation of optimized molecular geometries and vibrational frequencies. © 2019 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.25818

Introduction

Valence bond (VB) theory provides a rigorous mathematical basis for intuitive ideas in chemistry concerning bonding, hybridization, the octet rule, resonance, and so on.[1] Beginning with the seminal work of Heitler and London,[2] and Coulson and Fischer,[3] the concepts of Pauling,[4] and Lewis were translated into an *ab initio* methodology for dihydrogen. However, beyond dihydrogen, the presence of core orbitals posed a major obstacle and, subsequently, the methods of generalized VB (GVB)[5] and spin-coupled VB (SCVB)[6] used a basis of molecular orbitals (MOs) to gain access to larger molecules. As then there has been steady progress in the efficiency of VB implementations.[7] Among these, the MOLPRO[8] and MOLCAS[9] programs implement CASVB,[10,11] which is related to SCVB; GAMESS-US[12,13] implements GVB; the TURTLE[14] module of the GAMESS-UK software package[15] implements the VB self-consistent field method.[16] The variety of methods has broadened to include breathing orbital VB,[17] the block-localized wave function method,[18,19] and MOVB,[20] among others. VB has been combined with molecular mechanics (MM/VB)[21] and effective fragment potentials (VBEFP).[22] Many methods are incorporated into the VB2000[23,24] and XMVB packages.[25,26] Recently, it has become possible to refer to the closely related GVB and SCVB methodologies under the combined heading of spin-coupled generalized valence bond, or SCGVB.[27]

This article describes VALENCE, a quantum chemistry program designed for performing *ab initio* electronic-structure calculations based on the variational subspace valence bond (VSVB) method.[28] The key idea that sets VSVB apart is the use of multicenter orbitals that are required to be linearly independent but not orthogonal. The purpose of linear independence is to prevent collapse during optimization of the orbital expansion coefficients, playing a role analogous to that of orthogonality in MO theory. Orthogonality is itself a form of linear independence, but the ability to remove the requirement of orthogonality results in orbitals that tend to be highly local. In addition, multicenter expansions of the VSVB orbitals minimize the number of determinants required in the VB representation. Both these features lead to high scalability, whereas the VB representation yields a highly versatile form of wave function capable of modeling diverse chemical situations. Unlike many of the VB methods described above, VSVB allows wave functions to be constructed without needing MOs at any stage and, thus, VSVB permits the direct application of VB ideas to arbitrary molecules. With VSVB, it is possible for a VB wave function to achieve Hartree–Fock (HF) quality with a single determinant, and model excited states with a single configuration or determinant.

The purpose of the present work is to describe the major features of the software implementation in conjunction with the qualitative aspects of how VSVB can be used. To this end, in the sections to follow we describe the VSVB theory, discuss the implementation of the method, the tools developed for input generation, and its integration with other codes. We also use simple example cases to illustrate the major features of VSVB, such as spin coupling, excited states, bond dissociation, and static correlations and demonstrate the parallel scalability.

[a] G. D. Fletcher
Computational Science Division, Argonne National Laboratory, 9700 S. Cass Ave., Lemont, Illinois 60439
E-mail: fletcher@alcf.anl.gov
[b] C. Bertoni
Argonne Leadership Computing Facility, Argonne National Laboratory, 9700 S. Cass Ave., Lemont, Illinois 60439
[c] M. Keçeli
Computational Science Division, Argonne National Laboratory, 9700 S. Cass Ave., Lemont, Illinois 60439
[d] M. D'Mello
Intel® Corporation, 425 N. Martingale Road, Suite 1500, Schaumburg, Illinois

Contract Grant sponsor: Office of Science; Contract Grant sponsor: U.S. Department of Energy; Contract Grant number: DE-AC02-06CH11357; Contract Grant sponsor: Argonne National Laboratory

Theory

The general form of the VSVB wave function, Ψ, is multi-structural (or multiconfigurational),

$$\Psi = \sum_i C_i \psi_i \tag{1}$$

where each ψ_i is a single-structure wave function with expansion coefficient C_i. For simplicity, the single-structure (or single configuration) form,

$$\psi = \hat{A}\varphi_1\varphi_2\ldots\varphi_N\Theta_S^N \tag{2}$$

is used throughout this work, where $\hat{A}$ is the antisymmetrizer, φ_i are the one-electron orbitals, and Θ_S^N is a spin function coupling the N electrons to the total spin, S. For more details see Ref. [28]. In VSVB, each orbital is expanded in a subset of the basis functions on the molecule,

$$\varphi_i = \sum_\mu^{M_i} C_{\mu i}\zeta_{\mu i} \tag{3}$$

where $C_{\mu i}$, $\zeta_{\mu i}$, and M_i are the expansion coefficients, basis functions, and number of basis functions, for ith orbital. This contrasts with standard MO theory, where each orbital is expanded in the set of all basis functions on the molecule,

$$\varphi_i^{\mathrm{MO}} = \sum_\mu^M C_{\mu i}\zeta_\mu, \tag{4}$$

where M is the total number of basis functions. In VSVB, the orbitals are required to be linearly independent rather than orthogonal, and eq. (3) facilitates this by allowing the summation to run over any subset of the molecular basis. The subsets are chosen at input, not by the program itself, but either directly by the user or through the use of an automated tool for generating the input, which is described in the Implementation section.

Linear independence is achieved for arbitrary values of the orbital expansion coefficients when no orbital expansion is a subset of any other (provided, of course, that the underlying basis set is itself free of linear dependence problems). For example, if each orbital expansion contains a basis function that is unique to it and not present in any other orbital expansion, then the orbitals are linearly independent. In this case, the number of basis functions M_i needed to expand each orbital φ_i is $M_i = M - N + 1$. In the case of $M_i = M - N + 1$, the optimized single-determinant VSVB wave function with all orbitals double-occupied has the Hartree–Fock energy. To distinguish this method from Hartree–Fock, it is called 'variational subspace Hartree–Fock" (VSHF) and can be regarded as a special case of VSVB.

In addition to double-occupied (and single-occupied orbitals) orbitals, the general VSVB wave function can incorporate "spin-coupled" (SC) orbitals. Currently, VALENCE supports pairs of spatially distinct orbitals occupied by electrons coupled to a singlet using a spin function of the form.[29]

$$\Theta_0^2 = \frac{\alpha(i)\beta(j) - \beta(i)\alpha(j)}{\sqrt{2}} \tag{5}$$

where i and j are electrons/orbitals. Such spin functions give rise to 2^{N_S} determinants in the wave function, where N_S is the number of SC pairs. The corresponding number of determinants in the energy expression is 2^{2N_S}, that is, exponential in the number of SC electrons. VSHF wave functions can be improved by replacing double-occupied orbitals with SC pairs to yield more general VSVB wave functions with greatly enhanced applicability, akin to SCGVB. SC pairs reflect charge polarizations yielding static correlation corrections beyond Hartree–Fock quality, and are spatially separable in order to model bond dissociation accurately, as shown in the "Static Correlations and Bond Dissociation" section.

In VSVB, the total electronic energy is obtained by inserting eq. (2) into the pseudo-eigenvalue form of the Schrödinger equation,

$$E = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} \tag{6}$$

where $\hat{H}$ is the non-relativistic N-electron Hamiltonian and rearranging to give,

$$E = \sum_{i,j} h_{ij}D_{ij} + \sum_{ijkl} g_{ijkl}D_{ijkl} \tag{7}$$

where h_{ij} and g_{ijkl} are the standard one- and two-electron integrals over the orbitals, and D_{ij} and D_{ijkl} are the corresponding density cofactors, respectively. The analogous equation to eq. (7) in standard MO theory reduces the one-electron term to a sum over one index and the two-electron term to a sum over two indexes due to orthogonality [instead of two and four indexes, respectively, in eq. (7)]. However, with the orthogonality constraint, each MO must still be expanded in all basis functions necessitating an integral transformation whose cost scales as the fifth power of the problem size. Relaxing the orthogonality constraint allows the orbital expansions to be shorter, and results in high parallel efficiency, as demonstrated in the Test Calculations section.

For the optimization of the linear parameters (e.g., the orbital expansion coefficients) of the wave function [eq. (2)], there are many possible methods. A common approach is to differentiate eq. (7) with respect to the expansion coefficients of a given orbital (e.g., $C_{\mu i}$ in eq. (3)) and rearrange to yield a generalized eigenproblem,

$$\mathbf{H}_i\mathbf{C}_i = \mathbf{S}_i\mathbf{C}_i\mathbf{E}_i \tag{8}$$

where $\mathbf{H}_i$, $\mathbf{S}_i$, $\mathbf{C}_i$, and $\mathbf{E}_i$ are the matrix representations of the Hamiltonian operator, overlap operator, the orbital expansion coefficients, and eigenvalues, respectively. See Ref. [28] for more details. eq. (8) is then solved by standard methods to obtain improved expansion coefficients. This problem is solved without constraints because the issue of collapse is handled by the linear independence of the orbitals and their normalization is carried out in a separate step afterward. The order of the

eigenproblem is M_i that does not generally increase with the molecule size, due to locality. Where there is more than one orbital to consider, this process must be repeated for each orbital until self-consistency is achieved. However, because the cost of computing $\mathbf{H}_i$ is approximately that of $M_i(M_i + 1)/2$ energy calculations, wave function optimization in VSVB is much more expensive than in, say, Hartree–Fock. On the other hand, the N-electron Hamiltonian used in eq. (6) generates a manifold of N-electron states including the ground state, where higher roots correspond to excited states. Any state can be selected and optimized to self-consistency. This is true regardless of the form of the N-electron wave function, and so it can be true for a single-reference or even a single-determinant wave function. Table 1 summarizes the difference between HF, VSHF, and SC VSVB, which is VSVB using spin-coupled orbitals.

Implementation

VALENCE is written in Fortran 90. It is open-source, distributed under the 3-clause BSD license, and can be downloaded from GitHub.[30] We use software engineering tools to make VALENCE more maintainable and usable, as well as to improve the resiliency and stability. To do this, we use continuous integration and automated testing with Travis CI, version control with git, and automated documentation with Doxygen. We also include a Python tutorial in a Jupyter notebook[31] and Docker[32] and Singularity[33] recipe files for users that would like to run the code using a container.

In this section, we discuss automating input generation with VTOOLS, the implementations of the VSHF method, the spin-coupled VSVB method, excited states in the VSVB method, and discuss integrations with other codes and the memory requirements of VALENCE.

Automated input generation

Nowadays, quantum chemistry researchers are accustomed to high levels of support when it comes to performing at least entry-level calculations. The ease of performing MO calculations is due to the wealth of technology (such as methods for guess wave functions, convergence enhancers, and so on) developed over the years by many researchers. In a similar spirit, we are developing analogous levels of support for VSVB calculations. Here, we describe VTOOLS, a Python code that depends on various modules: py3Dmol,[34] EBSEL,[35,36] Open Babel,[37] obtools, and iotools.[38,39] With the following minimum information given by the user, VTOOLS can currently generate VALENCE input files for any type of closed-shell molecule in which the orbitals are doubly occupied:

- Chemical identifier: A SMILES or an InChI string, or a path to any file that can be recognized by Open Babel.[37]
- Basis-set name as defined in "Basis set exchange" community database.[40,41]

Given this information, VTOOLS generates an input file with a minimal wave function. In this basic form, the wave function has only one-center orbitals (core and lone-pair) and two-center bonding orbitals. Such a wave function will not provide a realistic representation of the molecule; however, it can be used in general for any system.

A more sophisticated wave function can be generated with VTOOLS using a database of orbitals (currently, one- or two-center orbitals are supported). Orbitals may be copied from the database and transferred to a "target" molecule using bonding information from Open Babel[37] to place the bonding and lone-pair orbitals correctly. The orbitals from the database are reoriented to match the geometry of the target molecule (details in the Supporting Information).

At the time of writing, the database in our repository[30] currently covers the major orbital types associated with H, C, N, and O atoms for the 6-31G basis set, with on-going work to incorporate more atom types, orbital types, and basis sets. The reorientation of an orbital is depicted in Figure 1.

To make generating an input file simple and help users visualize and understand VALENCE input files, we include a tutorial written in Python in a Jupyter notebook.[31] The user can input a molecule, basis set, and method for generating orbitals, and an input file for VALENCE will be generated. In addition, the tutorial can be run in a web browser using Binder,[42,43] allowing the tutorial to be run interactively and immediately, without the need for downloading additional dependencies or configuring to a specific computing environment. Using Binder, the tutorial can be launched directly from the GitHub repository. Figure 2 shows an example of part of the interactive interface of the VSVB tutorial.

VSHF

A VSHF calculation begins by reading a file which contains the initial wave function specification. The wave function specification is

Table 1. Summary of differences between HF, VSHF, and SC VSVB.

	HF	VSHF	SC VSVB
Orthogonality requirement?	Yes	No	No
Ability to model bond breaking?	No	No	Yes
How orbitals are expanded	Over all M basis functions in the system	Over subset M_i of all M basis functions in the system	Over subset M_i of all M basis functions in the system
How orbitals are optimized	By solving the HF equations self-consistently	By solving secular equations for each orbital self-consistently until convergence	By solving secular equations for each orbital self-consistently until convergence
Orbital optimization equation	$\mathbf{FC} = \mathbf{SC}\epsilon$ (all matrices are M x M)	$\mathbf{H}_i\mathbf{C}_i = \mathbf{S}_i\mathbf{C}_i\mathbf{E}_i$ (all matrices are M_i x M_i)	$\mathbf{H}_i\mathbf{C}_i = \mathbf{S}_i\mathbf{C}_i\mathbf{E}_i$ (all matrices are M_i x M_i)
Computational complexity (with screening)	M^4 (M^{2-3})	N^7 (N^{2-3})	$2^{2N_s}N^7$ ($2^{2N_s}N^{2-3}$)

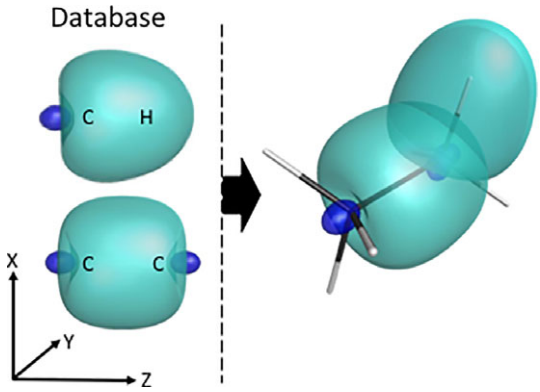

Figure 1. The reorientation of CC and CH bond orbitals from the database to a target molecule. The left side of the figure represents the VTOOLS database, with CC and CH bond orbitals in the "standard" orientation. The right side of the figure represents rotating the CC and CH bond orbitals from the database to the target molecule (ethane). Note that, for clarity, only one CH bond (out of six) is shown in the target molecule. [Color figure can be viewed at wileyonlinelibrary.com]

composed of orbitals, which are defined using an "orbital basis set" (OBS).

The orbital definitions, eq. (3), provide a unique opportunity to construct the entire wave function on the basis of chemical intuition, conceptually simplified by the absence of orthogonality. The orbital choices can be entered by hand using the OBS format, but they are also readily automated and we exploit this feature to augment VALENCE with a database of predefined orbitals that can be used at least as initial guesses, together with tools for combining orbitals to make wave functions (as described in the "Automated input generation" section).

The first step of a run is to compute the energy of the input wave function using eq. (7) (the "guess" energy). It is possible for this to be the only task of the run. However, if the user requested the energy to be optimized with respect to the orbitals, the guess energy step is followed by a self-consistent procedure which optimizes the energy with respect to one orbital at a time [in the presence of all other orbitals, as shown in eq. (8)]. The procedure loops, optimizing each orbital one at a time until the energy no longer changes. Figure 3 outlines the energy optimization procedure.

The energy calculations themselves are split into one- and two-electron terms [the first and second term in eq. 7, respectively]. Due to the much larger order of the two-electron terms (a loop over N^4 terms), the two-electron term typically dominates the calculation time. The one- and two-electron terms are parallelized using MPI, where each process computes a block of integrals and density cofactors in a round-robin manner. VALENCE makes use of any available memory to store integrals and avoids their recomputation in optimization runs.

Spin-coupled VSVB

As mentioned in the theory section, VSVB wave functions can incorporate spin-coupled orbitals for enhanced chemical versatility, at the expense of increasing the number of determinants in the wave function. A common issue with multideterminental methods is the need to annotate the *N*-electron wave function (determinants), requiring exponential or even combinatorial storage. VALENCE avoids this storage by generating the determinant information corresponding to the choice of spin coupling on-the-fly. Consequently, there is no limit in terms of storage to the value of N_S, compute time being the only penalty. In general, there are multiple possible spin couplings for a given value of N_S. VALENCE currently allows multiple couplings of the kind shown in eq. (5) to be entered, yielding a variational spin space. The flow diagram for optimizing the spin-coupling expansion coefficients is the same as the energy calculation in Figure 3 but with an additional step analogous to eq. (8) for the spin-coupling expansion coefficients. Online documentation[30] contains detailed examples and advice on how to perform SC calculations.

Excited states in the VSVB method

As mentioned in the theory section, VALENCE can optimize single-reference excited-state wave functions. Excited states correspond to higher roots of the generalized eigenproblems obtained by taking the first derivative of the energy with respect to the orbital expansion coefficients. Thus, selecting a higher root for a given orbital at input causes VALENCE to self-consistently optimize the corresponding excited-state wave function. This is true for any type of orbital such that selecting a higher root for a double-occupied orbital results in a double excitation. Selecting higher roots for multiple orbitals results in multiple excitations and so on. An example of an excited-state calculation is given in the "Demonstration of different methods in VALENCE" section.

Integrations with other codes

It is our goal to make it easy for other software packages to use VALENCE and for it to be easy for VALENCE to use optimized libraries in other packages. To this end, there is a library build of VALENCE with a Fortran API, and community standards are used in VALENCE when possible. Currently, VALENCE uses the Simint integral library[44] to take advantage of an integral package that is highly vectorized. To make it easy to call additional integral libraries in the future, VALENCE uses the standard ordering of Cartesian basis functions set by the Common Component Architecture.[45]

The API for the library version, *libvalence*, is defined in our online documentation.[30] The decision to include a library build and API was made so that we can more easily integrate VALENCE with other codes. As an example of the usefulness of creating a library version of VALENCE, VALENCE was interfaced with NITROGEN,[46] a code which can compute rovibrational energies for a given potential energy surface library. Becuase VALENCE does not currently have geometry optimizations or rovibrational calculations implemented, being able to use NITROGEN with VALENCE immediately expands the properties we can compute using the methods in VALENCE. The VALENCE/NITROGEN integration was used to calculate optimized geometries and harmonic vibrational frequencies of small molecules, as described in the "Test Calculations" section.

The API contains functions to initialize a VSVB calculation, finalize a VSVB calculation, and run a calculation based on the inputs from the initialize step and coordinates provided to the

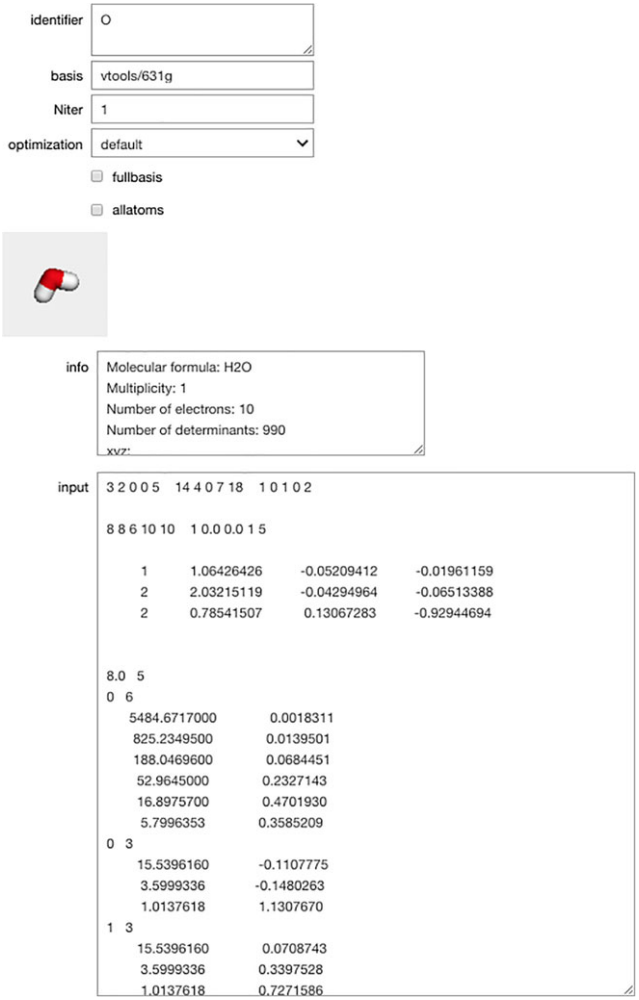

Figure 2. Interface for part of the VSVB Jupyter tutorial for a water molecule. The "O" is the SMILES string for water, and the 6-31G basis set is chosen. For simplicity, not all of the output is shown. The "Number of determinants" refers to the upper bound for the number of density cofactors in the computation of the energy. [Color figure can be viewed at wileyonlinelibrary.com]

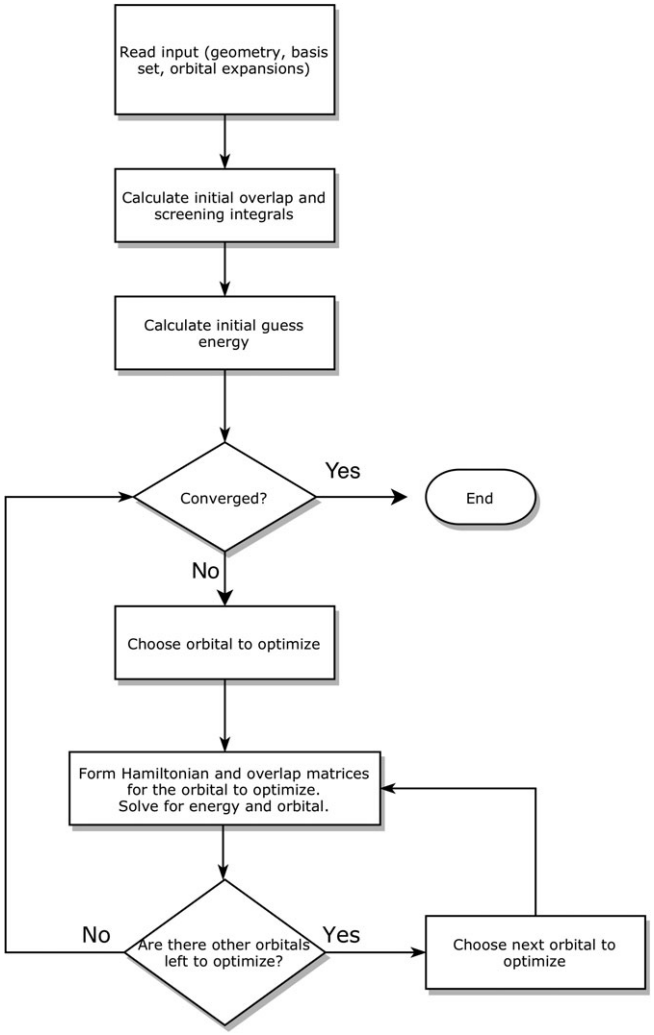

Figure 3. Diagram showing optimized energy calculation.

API. The initialization step reads in input (coordinates, basis set, expansion of orbitals, and various run parameters), allocates storage for the inputs, and optionally calls *MPI_Init*. The finalize step deallocates memory allocated in initialize and optionally calls *MPI_Finalize*. The optional calls to MPI were added in with the intention of easing integration with other codes. They allow the programmer to initialize MPI themselves before calling VALENCE, and pass a user-defined MPI communicator or sub-communicator to VALENCE to use in the calculations.

Memory requirements

Quantum chemistry methods often have high memory demands compared to those of other fields. In VSVB, the removal of orthogonality constraints decouples the orbitals from one another resulting in memory requirements that are relatively low compared to other methods in typical calculations. We choose to compare VSVB with closed-shell, or restricted, Hartree–Fock (RHF), as the latter typically has the smallest memory requirements among *ab initio* quantum chemistry methods based on MO theory. However, the exact memory requirements of RHF (and VSVB) are often a complex function of the specific implementation and the hardware resources. To simplify matters, we model the highest-order storage required for RHF as 2.5 M^2, where M is the number of basis functions, while that for VSVB is $2n^2$, where n is the number of spatial orbitals. As mentioned earlier, storage of the VSVB wave function scales linearly with n, due to orbital locality. Table 2 compares the model memory requirements of RHF and VSVB for a series of simple alkanes using the 6-31G* basis set.

As the number of carbon atoms increases, the ratio for simple alkanes with 6-31G* tends to 3125/64 or $\approx$ 48.8.

Test Calculations

This section describes test calculations that demonstrate the key features of VALENCE. We first discuss and compare the results of different methods available in VALENCE for a LiH system and then demonstrate the parallel scalability, accuracy of molecular geometries, vibrational frequencies, total energies, and static correlations. Finally, we discuss orbital transferability.

Table 2. Comparison of memory requirements for RHF and VSVB (units of 8-bytes).

Molecule	Memory required		RHF/VSVB
	RHF	VSVB	
C_4H_{10}	30,250	578	52.3
C_8H_{18}	110,250	2178	50.6
$C_{16}H_{34}$	420,250	8450	49.7
$C_{32}H_{66}$	1,640,250	33,283	49.3
$C_{64}H_{130}$	6,480,250	132,098	49.1

Demonstration of different methods in Valence

In this section, some of the features of VSVB are demonstrated using a progression of simple calculations. We begin with a VSHF calculation on the ground state of the diatomic molecule, lithium hydride, with the 6-31G** basis set[47] of spherical harmonic functions, at the RHF/6-31G** geometry (the interatomic separation is 1.6296 Å). Next, we convert the double-occupied valence orbital into a spin-coupled pair and re-optimize. Following this, we optimize an excited-state wave function corresponding to the next-to-lowest root of the equations for a single valence orbital. The results are summarized in Table 3.

In Table 3, the wave function output from the previous calculation was used as the guess wave function for the next calculation. The wave function with spin-coupled valence orbitals (SC) offers a qualitative improvement over VSHF (i.e., superior to Hartree-Fock quality) together with the ability to dissociate correctly into atoms as the internuclear separation is increased. The excited-state wave function, similarly, can be dissociated. A more complex example of bond dissociation is described in the "Static Correlations and Bond Dissociation" section, for dinitrogen.

Computational performance and scalability

The computation of the total energy, eq. (7), is dominated by the two-electron interactions (second term on the right hand side) where the number of terms grows nominally as the fourth power of the problem size. However, with integral screening to avoid small terms the overall complexity is typically much closer to cubic. Furthermore, the low memory requirements described in the Memory Requirements section are easily replicated on all nodes in modern computer architectures allowing all terms to be computed independently with low communication overheads, resulting in high scalability.

To demonstrate the computational performance of VSVB calculations, we chose the ground state of a lithium fluoride (LiF) cluster. Such calculations are part of a broader study of excitons, where valence bond methods are highly relevant (to be reported elsewhere). While the spin-coupled calculations are more accurate, they are impractical as a basis for parallel scalability measurement where smaller processor partitions might be required. On the other hand, the ground-state calculation has nearly identical scaling while being significantly shorter. The LiF cluster has 64 lithium cations (Li^+) and 64 fluoride anions (F^-), for a total electron count of 768. For the input file see the Supporting Information. The performance of the energy calculation is measured on the Blue Gene/Q system, *Mira*, hosted by the Argonne Leadership Computing Facility. The speedup and parallel efficiency are shown in Figure 4 where it can be seen that Valence achieves 92% efficiency to 1,048,576 processes. The high parallel efficiency of VSVB greatly mitigates its cost relative to MO-based methods.

Table 3. LiH calculations with different methods.

Wave function	Total E/AU	ΔE[a] (kcal/mol)	Notes
VSHF	−7.981275	0.0	Hartree–Fock energy
SC[b]	−7.997674	−10.28	spin coupled, valence
SC*[c]	−7.880466	73.55[d]	excited, $1\sigma^2 2\sigma 3\sigma\ ^1\Sigma^+$

[a] ΔE refers to the difference of the total energy with that in the previous row.
[b] SC refers to use of a spin coupled orbital pair for the valence, or bonding, electrons
[c] SC* indicates an excited state calculation using the SC form.
[d] Corresponds to a "vertical" excitation energy.

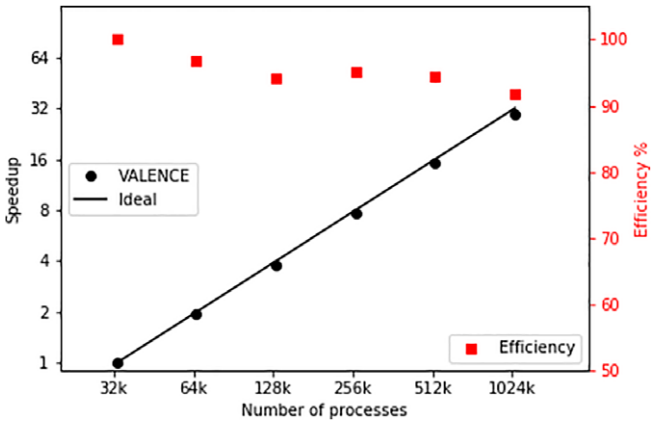

Figure 4. Speedup (black circles, left axis) and efficiency percentage (red squares, right axis) plot of Valence energy calculation on the *Mira* supercomputer using up to 1,048,576 processes. [Color figure can be viewed at wileyonlinelibrary.com]

Geometries and frequencies

To validate the accuracy of VSHF in Valence, the ground-state energies, optimized geometries, and harmonic vibrational frequencies for three small systems—CH_4, NH_3, H_2O— are compared to HF values in Tables 4–6. The geometries and frequencies for VSHF were generated by calling Valence from Nitrogen.

As shown in Tables 4–6, the VSHF energy values produced agree with the HF energy values within 1.0×10^{-6} Hartrees. The harmonic vibrational frequencies calculated by VSHF agree with the results from HF within 1.2 cm^{-1} for all systems, and on average only differs by 0.3 cm^{-1}. In Tables 4 and 5, the VSHF frequencies are the average of the frequencies of several degenerate modes. For the frequencies in Table 5, the range of frequencies averaged over was <0.09 cm^{-1}, and for the frequencies in Table 4, the range of frequencies averaged over was <0.06 cm^{-1}.

The optimized geometries are nearly identical between VSHF and HF, with the bond lengths agreeing within 0.001 Å$ $ and the angles agreeing within 0.02 °. In Tables 4 and 5, the angle and the bond lengths are average values. For the bond lengths and angles in Table 4, the range of values averaged over was <9.0×10^{-7} Å and 2.0×10^{-7}, respectively. For the bond lengths

Table 4. Comparison of properties between VSHF and HF for CH_4.

	E (AU)	ω_1 (bend) (cm^{-1})	ω_2 (bend) (cm^{-1})	ω_3 (stretch) (cm^{-1})	ω_4 (stretch) (cm^{-1})	CH angle (°)	CH bond length (Å)
HF/6-31G[a]	−40.180554	1517	1709	3182	3296	109.471	1.082
VSHF/6-31G	−40.180554	1517.0	1708.8	3181.9	3296.5	109.471	1.082

[a] Values from Ref. [49]

Table 5. Comparison of properties between VSHF and HF for NH_3.

	E (AU)	ω_1 (bend) (cm^{-1})	ω_2 (bend) (cm^{-1})	ω_3 (stretch) (cm^{-1})	ω_4 (stretch) (cm^{-1})	NH angle (°)	NH bond length (Å)
HF/6-31G[a]	−56.165521	599	1815	3780	3984	116.116	0.991
VSHF/6-31G	−56.165521	597.8	1814.6	3779.8	3984.0	116.131	0.991

[a] Values from Ref. [49]

Table 6. Comparison of properties between VSHF and HF for H_2O.

	E (AU)	ω_1 (bend) (cm^{-1})	ω_2 (stretch) (cm^{-1})	ω_3 (stretch) (cm^{-1})	HOH angle (°)	OH bond length (Å)
HF/6-31G[a]	−75.985359	1737	3988	4145	111.545	0.950
VSHF/6-31G	−75.985359	1737.0	3988.5	4145.4	111.545	0.950

[a] Values from Ref. [49].

and angles in Table 5, the range of values averaged over was < 1.1×10^{-6} Å and 3.2×10^{-4}, respectively.

Overall, the differences between VSHF and HF properties are small, validating that VSHF produces the same properties as HF. See the Supporting Information for all frequencies and geometries.

Static correlations and Bond dissociation

Table 7 compares total energies computed with VSHF, VSVB with multiple different orbitals being spin coupled, and complete active space self-consistent field (CASSCF)[48] for H_2O and N_2. All calculations were performed with the 6-31G* basis set, and the geometries are the RHF/6-31G* geometries from Ref. [49]. The CASSCF energies were calculated using GAMESS.[12] Table 7 demonstrates how spin-coupling different orbitals in VSVB recovers different amounts of pair-wise static correlation.

Table 7. Comparison of VSHF, SC VSVB, and CASSCF.

Molecule	Wave function, spin-coupled pair	Total E /AU	$E - E_{VSHF}$ (kcal/mol)
H_2O	VSHF	−76.023574	0.00
	VSVB, OH-bond	−76.043563	−12.54
	VSVB, σ-lone pair	−76.030532	−4.37
	VSVB, π-lone pair	−76.034037	−6.57
	VSVB, all pairs	−76.081129	−36.12
	VSVB additivity error: −0.10 kcal/mol		
	CASSCF	−76.106919	−52.30
N_2	VSHF	−108.943950	0.00
	VSVB, σ-bond	−108.952915	−5.63
	VSVB, π-bond	−108.974339	−19.07
	VSVB, lone pair	−108.952397	−5.30
	VSVB, all pairs	−109.028290	−52.92
	VSVB additivity error: 1.44 kcal/mol		
	CASSCF	−109.115247	−107.49

CASSCF and VSVB both incorporate "static" correlations that arise from the charge polarizations between individual electrons. CASSCF wave functions involve all possible configurations arising from eight electrons occupying eight (valence) orbitals for H_2O, and 10 electrons occupying 10 (valence) orbitals for N_2. Including multiple configurations in the wave function accounts for static correlation in CASSCF. In VSVB, the static correlation is accounted for by spin-coupling pairs of orbitals, as mentioned in the Theory section.

A unique aspect of VSVB is that the total static correlation energy can often be estimated accurately by computing the static correlation energy for each chemically important pair of orbitals, and then summing the contributions. As shown in Table 7, for water, we note that the difference of the "all pairs" VSVB energy and the sum of the individual pair VSVB energies computed separately (the additivity error) is just 0.1 kcal/mol. (Note that the pair contributions are doubled in the case of the two OH-bonds for water and two π-bonds for N_2.) This means that a good estimate of the total static correlation energy can be computed by summing the contributions of the pairs. For N_2, the additivity error is 1.44 kcal/mol. The relatively larger additivity error in the case of N_2 is likely due to the strongly interacting σ-orbitals (two lone-pair and one bonding) along the bond axis. For both test cases, the total correlation contribution for "VSVB, all pairs" is less than that of CASSCF because the latter incorporates additional "N-particle static" correlations arising from the full configuration interaction calculation over the active orbitals, whereas the particular use of SC orbitals in VSVB here recovers only the combined effect of the static correlations involving the valence electron pairs selected with a single-spin coupling. Overall, VSVB provides chemically meaningful components of the

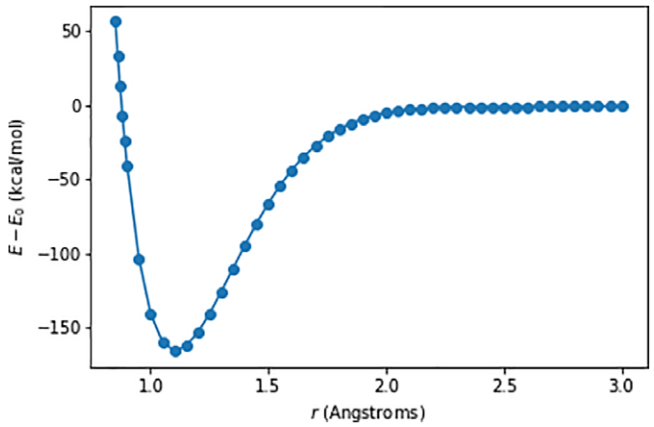

Figure 5. N_2 dissociation plot. $E - E_0$ corresponds to the energy difference between the ground state of the N_2 molecule and the sum of energies of two N atoms at the dissociation limit and r is the distance between N atoms. [Color figure can be viewed at wileyonlinelibrary.com]

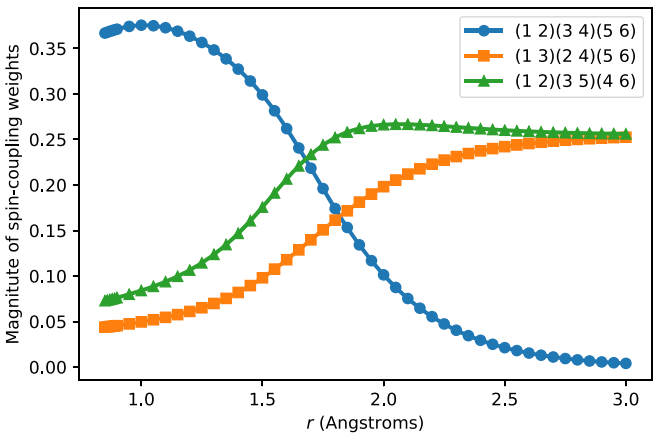

Figure 6. Variation in spin-coupling expansion coefficients with the dissociation of dinitrogen. r is the distance between N atoms. The spin-coupling modes are illustrated in Figure 7. [Color figure can be viewed at wileyonlinelibrary.com]

static correlation, while at the same time being more scalable (exponential cost) than CASSCF (combinatorial cost), especially if additivity is employed (linear cost).

Although individual pairwise static correlations yield modest improvements to the total energy, a larger benefit of SC orbitals lies in the extension of the methodology to model broader regions of the potential energy surface, including bond breaking. We provide a simple demonstration of the use of SC orbitals in VSVB for computing potential energy surfaces (PESs). Figure 5 shows the PES for the dissociation of dinitrogen, N_2, breaking the triple bond. The cc–pVDZ basis set is used.[50] All three bonds (one σ and two π types) and two (σ) lone pairs are spin coupled for a total of five SC pairs. The PES was obtained by optimizing all wave function parameters at each point, including the spin-coupling expansion coefficients which are plotted separately in Figure 6. The spin-coupling modes are indicated using the notation of "(1,2)" for the electron pair of the σ-bond, "(3,4)" for that of the π_x-bond, and "(5,6)" for the π_y-bond. Thus, the "perfect pairing" mode is denoted "(1,2)(3,4)(5,6)". In addition, a mode that cross-couples the two π-bonds, "(1,2)(3,5)(4,6)" is used, along with two equivalent modes that cross-coupled the σ-bond with one π-bond, denoted "(1,3)(2,4)(5,6)" and "(1,5)(3,4)(2,6)." As the latter two modes are degenerate, only three curves are plotted in Figure 6. The spin-coupling modes are illustrated in Figure 7. Note that the current model is limited to a total of four SC modes primarily to obtain a simple chart showing the dependence of the spin-coupling expansion coefficients on the inter-atomic distance. More accurate models can use up to 42 SC modes for the coupling of the 10 valence electrons to an overall singlet.

Orbital transferability

The "Automated input generation" section described how VSVB wave functions can be constructed by transferring orbitals between different molecules. Here, we demonstrate orbital transferability in four simple alkanes by showing how orbitals optimized for C_2H_6 can be used in larger alkane calculations with reasonable accuracy. All calculations use the 6-31G basis set[47] and the RHF/6-31G geometries obtainable from the NIST Computational Chemistry Comparison and Benchmark Database.[49]

The VSVB wave functions are constructed as follows. The contracted 1 *s* functions of the 6-31G basis set are used exclusively for the carbon atom cores. Two-center expansions use all available basis functions (i.e., all except the carbon 1 *s* functions) on two atomic centers. Such two-center expansions offer a simple, although limited, case of linear independence. All orbitals are double occupied.

First the wave function and orbitals of an ethane molecule were variationally optimized. Then the optimized CH and CC σ bond orbitals were used to construct the guess wave function for C_3H_8, C_4H_{10}, C_5H_{12}, and C_6H_{14}. (The orbitals from ethane were duplicated, rotated, and translated as necessary.) For the CH-bond of ethane, the representative guess orbital is obtained by averaging over all six instances. Table 8 shows the difference (ΔE) between the guess energy and optimized energy for each alkane when the initial wave function was constructed from the ethane orbitals.

As might be expected with such chemically similar species, the energy difference between the guess energy and

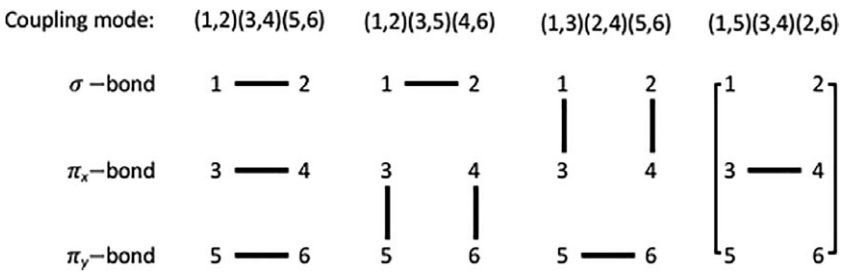

Figure 7. Spin-coupling modes for the bonding electrons of dinitrogen. Each line represents a singlet coupling. The rightmost two modes are degenerate.

Table 8. Transferability for alkane calculations (kcal/mol), where $\tilde{N}$ is the number of atoms in the molecule.

Molecule	ΔE	ΔE/$\tilde{N}$
C_3H_8	0.99	0.090
C_4H_{10}	1.19	0.085
C_5H_{12}	1.82	0.107
C_6H_{14}	2.05	0.103

optimized energy associated with the transfer of ethane orbitals to larger alkanes is low, around 2 kcal/mol, or less. Because the energy difference stands to grow with the molecule size, we also consider the energy difference per atom, ΔE/$\tilde{N}$, where $\tilde{N}$ is the number of atoms in the molecule. In Table 8 it is notable that ΔE/$\tilde{N}$ does not strictly increase with the target molecule size. This shows that orbitals optimized for ethane can be used to create a reasonable starting guess wave function for larger alkanes.

In principle, automatic guess wave functions can be improved to the point where further optimization is, for many practical purposes, obviated. The availability of high-quality guess wave functions, together with the high scalability described earlier, stands to greatly mitigate the higher cost of VSVB compared to MO-based methods.

Conclusions

In this article, we described VALENCE, an implementation of the VSVB method. The open-source software package includes tools to generate wave functions, a database of orbitals to provide a starting point, example input files, and a library build intended for integration with other packages. For example, we described the interface with the NITROGEN software package, that enables molecular geometries and vibrational frequencies to be computed. We showed how VTOOLS automates input generation to simplify running valence bond calculations. In turn, this capability facilitates the study of the responses of generic orbitals to different molecular environments, which lends itself to machine-learning techniques. In addition, we showed how VALENCE can strong scale to more than 1 million processes and offered the functionality to model diverse chemical problems such as excited states and bond-breaking.

We are planning to present a more detailed study of the strong scaling and compute efficiency. We are also in the process of developing further methodological improvements including efficient models for the atomic cores of heavy atoms, dynamic correlation functionals for high accuracy results, and periodic boundary conditions for the study of bulk materials. We also plan to explore using machine learning techniques for wave function generation and continue to incorporate more atom types, orbital types, and basis sets into the database of generic orbitals.

Acknowledgments

At Argonne, the authors thank Susan Coghlan, Mike Papka, Rick Stevens, and Tim Williams for a generous allocation on Mira*, the IBM Blue Gene/Q system. MJD would also like to thank Intel Corporation for its support during the course of this work. An award of computer time was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357. Argonne National Laboratory's work was supported by the U.S. Department of Energy, Office of Science under contract DE-AC02-06CH11357. The authors would also like to thank Thom Dunning, Jr. for helpful suggestions.*

Keywords: electronic structure · valence bond · localized orbitals · strong scaling

How to cite this article: G. D. Fletcher, C. Bertoni, M. Keçeli, M. D'Mello. *J. Comput. Chem.* **2019**, 00, 1–10. DOI: 10.1002/jcc.25818

Additional Supporting Information may be found in the online version of this article.

[1] D. L. Cooper, Valence Bond Theory, Elsevier, Amsterdam, **2002**.
[2] W. Heitler, F. London, *Z. Phys.* **1927**, *44*, 455.
[3] C. A. Coulson, I. Fischer, *Philos. Mag.* **1949**, *40*, 386.
[4] L. Pauling, The Nature of the Chemical Bond and the Structure of Molecules and Crystals: An Introduction to Modern Structural Chemistry, Cornell University Press, Ithaca, NY, **1960**.
[5] W. A. Goddard, III., T. H. Dunning, Jr., W. J. Hunt, P. J. Hay, *Acc. Chem. Res.* **1973**, *6*, 368.
[6] D. L. Cooper, J. Gerratt, M. Raimondi, *Chem. Rev.* **1991**, *91*, 929.
[7] W. Wu, P. Su, S. Shaik, P. C. Hiberty, *Chem. Rev.* **2011**, *111*, 7557.
[8] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schutz, *WIREs Comput. Mol. Sci.* **2012**, *2*, 242.
[9] F. Aquilante, L. De Vico, N. Ferre, G. Ghigo, P.-K. Malmqvist, P. Neograd, T. B. Pedersen, M. Pitonak, M. Reiher, B. O. Roos, L. Serrano-Andrés, M. Urban, V. Veryazov, R. Lindh, *J. Comput. Chem.* **2010**, *31*, 224.
[10] D. L. Cooper, T. Thorsteinsson, J. Gerratt, Advances in Quantum Chemistry, Vol. *32*, Academic Press, San Diego, CA, **1998**, p. 51.
[11] T. Thorsteinsson, D. L. Cooper, J. Gerratt, P. B. Karadakov, M. Raimondi, *Theor. Chim. Acta* **1996**, *93*, 343.
[12] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis, J. A. Montgomery, *J. Comput. Chem.* **1993**, *14*, 1347.
[13] M. S. Gordon, M. W. Schmidt, In Theory and Applications of Computational Chemistry; C. E. Dykstra, G. Frenking, K. S. Kim, G. E. Scuseria, Eds., Elsevier, Amsterdam, **2005**, p. 1167.
[14] F. Dijkstra, J. H. van Lenthe, *J. Comput. Chem.* **2001**, *22*, 665.
[15] M. F. Guest, I. J. Bush, H. J. J. V. Dam, P. Sherwood, J. M. H. Thomas, J. H. V. Lenthe, R. W. A. Havenith, J. Kendrick, *Mol. Phys.* **2005**, *103*, 719.
[16] J. H. van Lenthe, G. G. Balint-Kurti, *Chem. Phys. Lett.* **1980**, *76*, 138.
[17] P. C. Hiberty, S. Shaik, *Theor. Chem. Acc.* **2002**, *108*, 255.
[18] Y. Mo, S. D. Peyerimhoff, *J. Chem. Phys.* **1998**, *109*, 1687.
[19] Y. Mo, J. Gao, S. D. Peyerimhoff, *J. Chem. Phys.* **2000**, *112*, 5530.
[20] Y. Mo, J. Gao, *J. Phys. Chem. A* **2000**, *104*, 3012.
[21] F. Bernardi, M. Olivucci, M. A. Robb, *J. Am. Chem. Soc.* **1992**, *114*, 1606.
[22] F. Ying, X. Chang, P. Su, W. Wu, *J. Phys. Chem. A* **2012**, *116*, 1846.
[23] J. Li, R. McWeeny, *Int. J. Quantum Chem.* **2002**, *89*, 208.
[24] J. Li, B. J. Duke, T. M. Klapötke, R. McWeeny, *J. Theor. Comput. Chem.* **2008**, *7*, 853.
[25] L. Song, Y. Mo, Q. Zhang, W. Wu, *J. Comput. Chem.* **2005**, *26*, 514.
[26] C. Zhenhua, Y. Fuming, C. Xun, S. Jinshuai, S. Peifeng, S. Lingchun, M. Yirong, Z. Qianer, W. Wei, *Int. J. Quantum Chem.* **2015**, *115*, 731.
[27] P. B. Karadakov, D. L. Cooper, *Chemphyschem* **2018**, *19*, 3186.
[28] G. D. Fletcher, *J. Chem. Phys.* **2015**, *142*, 134112.
[29] R. Pauncz, Spin Eigenfunctions, Plenum, New York, NY, **1979**.

[30] VALENCE, Quantum chemistry software implementing the Variational Subspace Valence bond method, v1.0, https://github.com/VALENCE-software/VALENCE; doi:10.5281/zenodo.1493601 (accessed March 19, 2019).
[31] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, Jupyter Development Team, In Positioning and Power in Academic Publishing: Players, Agents and Agendas; F. Loizides, B. Schmidt, Eds., IOS Press: Amsterdam, The Netherlands, **2016**; p. 87.
[32] D. Merkel, J. Linux **2014** (2014), 2014, 76.
[33] G. M. Kurtzer, V. Sochat, M. W. Bauer, *PLoS One* **2017**, *12*, 1.
[34] A. Virshup. py3Dmol. https://github.com/avirshup/py3dmol (accessed Nov 21, 2018).
[35] J. Rodríguez-Guerra. Ebsel, https://github.com/jaimergp/ebsel (accessed Nov 21, 2018).
[36] T. Applencourt. EMSL basis set exchange local, http://tapplencourt.github.io/EMSL_Basis_Set_Exchange_Local/ (accessed November 21, 2018).
[37] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, G. R. Hutchison, *J. Chem.* **2011**, *3*, 33.
[38] M. Keçeli, S. N. Elliott, S. J. Klippenstein. Quantum Thermochemistry Calculator, https://github.com/keceli/QTC/ (accessed Nov 21, 2018
[39] M. Keçeli, S. N. Elliott, Y.-P. Li, M. S. Johnson, C. Cavallotti, Y. Georgievskii, W. H. Green, M. Pelucchi, J. M. Wozniak, A. W. Jasper, S. J. Klippenstein, *Proc. Combust. Inst.* **2018**, *37*, 363.
[40] D. Feller, *J. Comput. Chem.* **1996**, *17*, 1571.
[41] K. Schuchardt, B. Didier, T. Elsethagen, L. Sun, V. Gurumoorthi, J. Chase, J. Li, T. Windus, *J. Chem. Inf. Model.* **2007**, *47*, 1045.
[42] Binder (beta), https://mybinder.org/ (accessed November 21, 2018.
[43] J. Freeman, A. Osheroff, Toward publishing reproducible computation with Binder, https://elifesciences.org/labs/a7d53a88/toward-publishing-reproducible-computation-with-binder (accessed November 21, 2018).
[44] B. P. Pritchard, E. Chow, *J. Comput. Chem.* **2016**, *37*, 2537.
[45] J. P. Kenny, C. L. Janssen, E. F. Valeev, T. L. Windus, *J. Comput. Chem.* **2008**, *29*, 562.
[46] P. B. Changala. NITROGEN, Numerical and Iterative Techniques for Rovibronic Energies with General Internal Coordinates, http://www.colorado.edu/nitrogen (accessed Nov 21, 2018).
[47] P. C. Hariharan, J. A. Pople, *Theor. Chim. Acta* **1973**, *28*, 213.
[48] B. O. Roos, *Adv. Chem. Phys.* **1987**, *69*, 339.
[49] R. D. Johnson, III. NIST Computational Chemistry Comparison and Benchmark Database, NIST Standard Reference Database Number 101. Release 19, April 2018, http://cccbdb.nist.gov/ (accessed Sept 10, 2018).
[50] T. H. Dunning, Jr., *J. Chem. Phys.* **1989**, *90*, 1007.

Received: 30 November 2018
Revised: 1 February 2019
Accepted: 19 February 2019