New Algorithms for Iterative Matrix-Free Eigensolvers in Quantum Chemistry

Dmitry Zuev,[a] Eugene Vecharynski,[b] Chao Yang,[b] Natalie Orms,[a] and Anna I. Krylov*[a]

New algorithms for iterative diagonalization procedures that solve for a small set of eigen-states of a large matrix are described. The performance of the algorithms is illustrated by calculations of low and high-lying ionized and electronically excited states using equation-of-motion coupled-cluster methods with single and double substitutions (EOM-IP-CCSD and EOM-EE-CCSD). We present two algorithms suitable for calculating excited states that are close to a specified energy shift (interior eigenvalues). One solver is based on the Davidson algorithm, a diagonalization procedure commonly used in quantum-chemical calculations. The second is a recently developed solver, called the "Generalized Preconditioned Locally Harmonic Residual (GPLHR) method." We also present a modification of the Davidson procedure that allows one to solve for a specific transition. The details of the algorithms, their computational scaling, and memory requirements are described. The new algorithms are implemented within the EOM-CC suite of methods in the Q-Chem electronic structure program. © 2014 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23800

Introduction

The task of finding several eigenpairs of large matrices is ubiquitous in science and engineering applications. It is encountered in structural dynamics, electrical networks, magnetohydrodynamics, control theory, and so forth.[1] In quantum chemistry, it appears in the context of computing ground- and excited-state wave functions using configuration interaction (CI),[2,3] equation-of-motion coupled-cluster (EOM-CC),[3–6] and algebraic diagrammatic construction (ADC)[7] methods as well as in Kohn–Sham time-dependent density functional theory.[8]

Even though computational power has improved dramatically in the last few decades, standard algorithms that solve for the entire eigen-spectrum (like the QR algorithm [9–11]) cannot handle large matrices (e.g., $N > 10^5$) that appear routinely in quantum chemistry calculations. For example, in an EOM-CC or CI calculation with single and double substitutions (EOM-CCSD or CISD, respectively) of a moderate-size system (around 300 atomic basis functions), the dimension of the full matrix exceeds 10^9. Constructing and storing such a matrix would be unwise; thus, in practical codes, the matrix is accessed only through a procedure that multiplies the matrix with a vector or a block of vectors (this also allows one to exploit the sparcity of the matrix). Using standard numerical methods of finding the entire set of eigenpairs for matrices of such a size is impractical, especially, since only a few eigenstates are of interest.

Several methods that compute only a few eigenpairs of large matrices have been developed. These methods do not require the construction of the entire matrix (thus, they are called matrix free), but rather involve a projection of the original matrix onto a so-called search subspace. The cost of these methods is determined by the projection step, which scales as nN^2 (where n is the number of vectors in the search subspace and N is the dimension of the matrix); for large matrices, this is much less than N^3 (the cost of the full diagonalization). Some examples are the Arnoldi[1,12] and Lancsoz[1,13] algorithms; for a detailed description of the algorithms for solving large eigenvalue problems, we refer the reader to a specialized book.[1]

In the quantum chemistry community, Davidson's method[14] has been predominantly used for solving both Hermitian and non-Hermitian eigen-problems. Davidson's method can be viewed as a generalization of the Lancsoz algorithm that uses the diagonal of the matrix as a preconditioner (Jacobi preconditioner) for new vectors that are added to the subspace.[1] The original (i.e., full) matrix is projected onto a search subspace of an increasing dimension and diagonalized within this subspace, yielding approximate eigenpairs of the full matrix. The matrices in quantum-chemical calculations are extremely sparse (only about 1–5% elements are nonzero) and are strongly diagonally dominant, which explains the success and effectiveness of Davidson's method. The detailed description of Davidson's algorithm is given below.

The original Davidson method was designed for finding a few lowest eigenpairs of a matrix, as in CI calculations of the ground and several lowest excited states. However, one might also be interested in high-lying states. For example, in core

[a] D. Zuev, N. Orms, A. I. Krylov
Department of Chemistry, University of Southern California, Los Angeles, California 90089-0482
E-mail: krylov@usc.edu

[b] E. Vecharynski, C. Yang
Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720

Contract grant sponsor: U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Sciences (Scientific Discovery through Advanced Computing (SciDAC) program).

ionization processes[15–17] the electron is ejected from a low-lying orbital with energy as high as several hundred electron-volts. Another example is autoionizing resonances, that is, excited states above the ionization threshold. When using the conventional Davidson method, one would need to request a large number of roots (often, more than 100) to find the desired root (see, for example, Ref. [18]). In most cases, it is not feasible to compute that many states since, in practice, the convergence of Davidson's method for more than 10 roots is poor and the storage and cost requirements of the algorithm grow proportionally to the number of states requested.

Motivated by these issues, we present two modifications of the Davidson algorithm that enable calculations of user-specified states. The first algorithm finds roots around an energy shift specified by the user. This algorithm is intended for a user who, after consulting experimental or other data, is interested in transitions that occur within a certain energy range. The second algorithm targets the solution dominated by a specific transition, for example, from/to user-defined orbitals (e.g., an excitation from HOMO-3 to LUMO+5 or an ionization from the 1s orbital of carbon [15,17]). This solver should be useful in calculations of low and high-lying states of a specific character. A similar algorithm is available in CFOUR[19] (J.F. Stanton, private communication).

We also present a production-level implementation of a new solver introduced in[20]—the generalized preconditioned locally harmonic residual (GPLHR) method. GPLHR involves construction of a subspace using approximate eigenvectors, preconditioned residuals, and a set of additional preconditioned residual-like vectors. The full matrix is then projected onto the subspace and a low-dimensional eigen-problem is solved, just as in the Davidson method. Similar to the modified Davidson method, the GPLHR algorithm allows the user to find roots around a specified energy shift.

The solvers are implemented within the EOM-CCSD suite of methods in the Q-Chem electronic structure package[21–23] and benchmarked using the EOM-CCSD methods for calculating ionization potentials (EOM-IP-CCSD) and excitation energies (EOM-EE-CCSD). Based on the structure of the underlying matrices, we expect a similar performance on the algorithms in the analogous CI or ADC schemes. The structure of the article is as follows: the next section presents a brief summary of the EOM-CC methods and describes the algorithms associated with each solver. We then illustrate the solvers' performance using several moderate-size systems (Benchmarks section), followed by concluding remarks.

Theory and Algorithms

EOM-CC family of methods

The EOM-CC family of methods[3–6,24] offers a toolset for describing electronically excited, ionized, and attached states in molecular systems. EOM-CC provides a balanced treatment of the reference and selected target states, accurate recovery of correlation energy, and is size intensive. The wave function of the target state is written as:

$$\Psi_{\mathrm{EOM}} = \hat{R} e^{\hat{T}} \Phi_0 \tag{1}$$

where $\hat{R}$ is a general excitation operator, $\hat{T}$ denotes the coupled-cluster excitation operator, and Φ_0 is the reference determinant. Depending on the type of physical process (excitation, ionization, electron attachment) or the character of target states, the operator $\hat{R}$ assumes different forms:

$$\hat{R}^{\mathrm{EE}} = r_0 + \sum_{ia} r_i^a a^\dagger i + \frac{1}{4}\sum_{ijab} r_{ij}^{ab} a^\dagger b^\dagger j i + \ldots \tag{2}$$

$$\hat{R}^{\mathrm{IP}} = \sum_{i} r_i i + \frac{1}{2}\sum_{ija} r_{ij}^{a} a^\dagger j i + \ldots \tag{3}$$

$$\hat{R}^{\mathrm{EA}} = \sum_{a} r^a a^\dagger + \frac{1}{2}\sum_{iab} r_{i}^{ab} a^\dagger b^\dagger i + \ldots, \tag{4}$$

where i, j, k and a, b, c denote occupied and virtual (in the reference Φ_0) orbitals, respectively. In EOM-CCSD, the $\hat{R}$ (and $\hat{T}$) series are truncated at the double excitation level, that is, 2-holes-2-particle (2h2p) in EOM-EE, 2h1p in EOM-IP, and 1h2p in EOM-EA. Similar to CI, the problem of finding the wave function amplitudes (coefficients r in the equations above) is reduced to diagonalizing the similarity transformed Hamiltonian:

$$\bar{H} = e^{-\hat{T}} H e^{\hat{T}} \tag{5}$$

where $\bar{H}$ is a real, non-Hermitian, and (usually) diagonally dominant matrix. By solving the non-Hermitian eigenvalue problem for the right and left (EOM bra states are $<\Phi_0|e^{-\hat{T}}\hat{L}^{+}$, where the operator $\hat{L}$ is an excitation operator similar to $\hat{R}$) eigenvectors, one obtains the energies and wave function coefficients of the target states:

$$(\bar{H} - E_{\mathrm{CC}} I) R = R\Omega \tag{6}$$

$$L(\bar{H} - E_{\mathrm{CC}} I) = \Omega L \tag{7}$$

$$<L|R> = I \tag{8}$$

where Ω is a diagonal matrix containing eigenenergies, I is the identity matrix, and matrices R and L contain the right and left eigenvectors (wave function coefficients). Note that only right eigenvectors are needed for energy calculations.

To clarify the notations, below we discuss algorithms for solving an eigenproblem for matrix A, which in the context of EOM-CCSD is the matrix of $\bar{H}$ in the basis of the reference, and all singly and doubly excited determinants (in CISD, A is the matrix of the bare Hamiltonian in the same basis). In CIS, A is the matrix of the bare Hamiltonian in the basis of singly excited determinants. Note that the structure of A in EOM-CCSD, ADC, and CISD is similar, as it is dominated by the bare (untransformed) Hamiltonian. Consequently, in EOM-CCSD, A is nearly Hermitian. Importantly, in all these methods (as well as in CIS) A is diagonally dominant when canonical or semicanonical orbitals are used. That is why the performance of the original Davidson solver is very similar in these methods. Therefore, we expect that the performance of the new algorithms should also be similar and that the main conclusions derived from the EOM-CCSD benchmarks should hold in the case of CISD, ADC, and CIS/TD-DFT.

Davidson's method

The development of the original Davidson algorithm was driven by large-scale CI calculations[14] in which matrices are real and symmetric. As discussed above, Davidson's method entails the projection of the full matrix onto an orthogonal search subspace and solving the eigenvalue problem for that projected matrix. These solutions (X_i and ω_i) can be used to obtain approximate eigenpairs (VX_i and ω_i, where V is the search subspace) of the original (i.e., full) matrix A; the error can be quantified by computing the residual vectors, $AVX_i - \omega_i VX_i$. At each step, the search subspace is expanded by using preconditioned, unconverged residuals as new vectors which, after performing an orthogonalization, are added to the search subspace. The details of the algorithm for finding n lowest roots of a diagonally dominant matrix A are given below:

Original Davidson's method.

1. Generate guess vectors based on diagonal D of matrix A. Sort the diagonal elements in ascending order and generate the corresponding unit vectors. Choose the first n (or more) unit vectors to form the initial search subspace, $V = \{v_1, v_2, \ldots, v_n\}$. In the case of degenerate values, adjust the number of guess vectors and target roots accordingly. We note that using a larger number of guess vectors can improve the convergence, especially in cases when the diagonal (so-called Koopmans) guess is a poor approximation to the target states.
2. Compute σ-vectors corresponding to the new vectors added to the search subspace, $\sigma_i = Av_i$.
3. Compute the subspace matrix: $A_V = V^T \sigma$.
4. Solve the eigenvalue problem for subspace matrix: $A_V X = X\Lambda$. Sort the eigenvalues in ascending order ($\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_k$) and choose the n lowest. Discard the rest of the eigenpairs.
5. Compute the residuals of the eigenpairs corresponding to matrix A: $r_i = \sigma X_i - \lambda_i VX_i$, where λ_i are eigenvalues from step (4) and X_i are the corresponding eigenvectors.
6. If $||r_i|| < \varepsilon$ for all residuals where ε is the convergence threshold, terminate and return eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ and their corresponding eigenvectors $\{VX_1, VX_2, \ldots, VX_n\} = \{R_1, R_2, \ldots, R_n\}$.
7. If the subspace reached a specified maximum size, collapse the subspace using the current approximations of eigenvectors as a new guess space and go to step (2).
8. Apply the preconditioner to the residuals with $||r|| \geq \delta$: $s_i = (D - \lambda_i)^{-1} r_i$, where D is the diagonal of matrix A.
9. Orthogonalize preconditioned residuals s_i against all other vectors in the search subspace, then normalize and add them to the search subspace if $||z_i|| \geq \delta$, where z_i is the orthogonalized preconditioned residual. Go to step (2).

The original Davidson method for symmetric matrices has been generalized for nonsymmetric matrices.[25] In the nonsymmetric case, the right and left search subspace vectors are constructed separately and the resulting left and right eigenvectors are biorthogonalized. When only eigenvalues and right eigenvectors are needed, the method is equivalent to the symmetric Davidson method; the only difference being that the subspace matrix (step 3 and 4) is nonsymmetric. Davidson's method has also been successfully combined with Jacobi's approach for eigenvalue approximation leading to the Jacobi–Davidson method.[1,26] It can be viewed as an instance of Newton's method with subspace acceleration for eigenvalue problems.

For non-Hermitian Hamiltonians, the subspace matrix is nonsymmetric and, in principle, complex eigenvalues may appear in subspace diagonalization. However, in practice, complex eigenvalues are rarely encountered in standard EOM-CC methods formulated for bound states with real energies because $\bar{H}$ is dominated by the bare Hamiltonian and is, therefore, nearly Hermitian. Thus, complex eigenvalues can be discarded as unphysical solutions (the handling of complex roots by the Davidson and GPLHR algorithms is described in the next section).

The extension of the EOM-CC methods for resonance states[27,28] involves modifying the Hamiltonian by performing a complex-scaling[27,29,30] transformation or by adding a complex absorbing potential[31–33] leading to complex, non-Hermitian matrices. The resulting eigenvalues and eigenfunctions are complex with the real part of the eigenvalue representing the position of the resonance state and the imaginary part being inversely proportional to its lifetime. The implementation of the Davidson solver within complex-scaled EOM-CC[34] and complex absorbing potential EOM-CC[35,36] methods entails working with complex values at all steps of the Davidson algorithm and using the c-product metric[37–39] instead of the conventional scalar product for the normalization and orthogonalization of the complex wave function [steps (8) and (9) above]; the rest of the algorithm is the same.

The most expensive step of the algorithm is step (2), the projection of the original matrix onto the search subspace which involves matrix-vector multiplications; it is called the "σ-vector calculation" in the electronic structure community. The number of matrix-vector evaluations at each step is equal to the number of new vectors added to the subspace. For EOM-EE-CCSD, ADC(3), or CISD, each matrix-vector multiplication scales as $O(N^6)$, where N is proportional to the number of electrons, and for EOM-EA-CCSD and EOM-IP-CCSD—as $O(N^5)$. For CIS/TD-DFT, σ-vector calculations scale as $O(N^4)$. In terms of memory, the most expensive part [$O(N^4)$ for EOM-EE-CCSD/CISD, $O(N^3)$ for EOM-IP/EA-CCSD, and $O(N^2)$ for CIS/TD-DFT] is storing the subspace vectors v_i and the associated σ-vectors, $\sigma_i = Av_i$. Thus, the maximum number of stored vectors at each iteration is twice the size of the subspace.

The original method proposed by Davidson[14] is designed for finding a few (1–10) lowest eigenpairs of the original matrix. However, one might require eigenpairs that lie high in the spectrum (the so-called "interior eigenpairs"). We present a modification of the original Davidson algorithm that enables computations of several eigenpairs around user-specified shift

η. The changes to the algorithm are minor and only concern steps (1) and (4):

Davidson's method with shift (only steps that are different from the original algorithm are described).

1. Generate guess vectors based on the absolute value of diagonal D shifted by η, sort the values in the ascending order ($|D_1-\eta| \le |D_2-\eta| \le \ldots \le |D_n-\eta|$) and generate the corresponding unit vectors. Choose the first n (or more) unit vectors to form the initial search subspace, $V=\{v_1, v_2, \ldots, v_n\}$. Using larger number of guess vectors may improve the convergence.
2. Solve the eigenvalue problem for subspace matrix $A_V X = X\Lambda$. Sort the absolute values of the eigenvalues shifted by η in ascending order ($|\lambda_1-\eta| \le |\lambda_2-\eta| \le \ldots \le |\lambda_k-\eta|$). Choose n lowest and discard the rest of the eigenpairs.

This algorithm targets the eigenvalues close to shift η (e.g., transitions around η = 300 eV), which is specified by the user based on the problem at hand.

We also implemented a harmonic Ritz modification of the Davidson solver. Instead of the regular diagonalization in step (4) of the Davidson procedure, a harmonic eigen-problem is solved, as in step (3) of the GPLHR algorithm described below.

The algorithm that allows the user to find a single root dominated by a particular transition (for example, ionization from HOMO-10 or core-LUMO excitation) also entails only minor deviations from the original Davidson procedure. Mathematically, it means that the algorithm looks for an eigenvector with the leading amplitude corresponding to a transition specified by the user. The algorithm returns only one root (number of roots n = 1) corresponding to the eigenvector of the requested character.

Davidson's method with user-defined guess (only steps that are different from the original algorithm are described).

1. Generate the guess vector as a unit vector corresponding to the transition specified by the user. Form the initial search subspace, $V=\{v_1\}$.
2. Solve the eigenvalue problem for the subspace matrix $A_V X = X\Lambda$. Sort eigenpairs in descending order of overlap with the current eigenvector approximation (i.e., the user guess vector $<v_1|VX_1> \ge <v_1|VX_2> \ldots \ge <v_1|VX_k>$) and choose the largest one. Since the approximate eigenvector is constructed as a linear combination of the vectors in the search subspace (which are orthonormal), the values of the overlap are given by the first row of matrix X: $<v_1|VX_i> = <v_1|\sum_j v_j X_{ji}> = \sum_j <v_1|v_j> x_{ji} = x_{1i}$.

GPLHR method

The GPLHR method[20] is a new eigensolver for computing a subset of eigenpairs of Hermitian and non-Hermitian matrices around user-specified shift η. Similarly to the modified Davidson method, GPLHR is a matrix-free solver that can make use of a preconditioner if available.

At every iteration, an orthonormal search subspace is constructed based on approximations from the previous iterations, preconditioned residuals, and a set of additional preconditioned residual-like vectors introduced to maintain a proper size and quality of the subspace. The approximate eigenpairs are then calculated using harmonic projection.[40,41] The details of the algorithm for finding n roots around shift η of a non-Hermitian matrix A are given below.

GPLHR method

1. Generate guess vectors based on the absolute value of diagonal D shifted by η. Sort the values in ascending order ($|D_1-\eta| \le |D_2-\eta| \le \ldots \le |D_n-\eta|$) and generate the corresponding unit vectors $V=\{v_1, v_2, \ldots, v_n\}$ as guess vectors.
2. Form the search subspace spanned by the columns of $\{V, W, S^{(1)}, \ldots, S^{(m)}, P\}$, where

$V=\{v_1, v_2, \ldots, v_n\}$ is a block of current approximate eigenvectors;

$W=\{w_1, w_2, \ldots, w_k\}$ is a block of preconditioned unconverged residuals $w_i = T(Av_i - \rho_i v_i)$, T is a preconditioner, and $\rho_i = v_i^* A v_i$;

$S^{(j)}=\{s_1^{(j)}, s_2^{(j)}, \ldots, s_k^{(j)}\}$ is a block of preconditioned residual-like vectors $s_i^{(j)} = T(As_i^{(j-1)} - \rho_i s_i^{(j-1)})$ corresponding to the unconverged roots computed recurrently from $S^{(j-1)}$ ($S^{(0)}=W$);

$P=\{p_1, p_2, \ldots, p_n\}$ is a block of vectors from the span of approximate eigenvectors at the current and previous iterations (not available at the first iteration);

m is the number of residual-like vectors; it is a user-defined parameter that controls the size of the search subspace.

Orthogonalize all columns of $\{V, W, S^{(1)}, \ldots, S^{(m)}, P\}$ (discard vectors with a norm less than threshold δ) and form a matrix Z that contains an orthonormal basis of the search subspace. It is assumed that n initial columns of Z (denoted by Z_V) span the approximate eigenspace given by V.

3. Solve the projected eigenvalue problem for the shifted matrix, $A-\eta$:

$$Z^*(A-\eta)^*(A-\eta)ZX = Z^*(A-\eta)^*ZX\Theta. \qquad (9)$$

Choose the n lowest eigenvalues (with respect to absolute values) θ_i of (9) and define X_n to be the matrix of the corresponding (normalized) eigenvectors. Discard the rest of the eigenpairs.

4. Compute new approximate eigenvectors using the solution of the projected problem at step (3) as $V=ZX_n$. Set $P=V-Z_V X_{n,V}$, where $X_{n,V}$ is a submatrix of X_n formed from its n initial rows. Normalize v_i and compute the corresponding approximate eigenvalues: $\rho_i = v_i^* A v_i$. Discard $\theta_i{}'s$.
5. Compute the residuals, $r_i = Av_i - \rho_i v_i$. If $||r_i|| < \varepsilon$ for all residuals (ε is the convergence threshold), then return the set of eigenvalues $\lambda_1=\rho_1, \lambda_2=\rho_2, \ldots, \lambda_n=\rho_n$ and their corresponding eigenvectors $v_1, v_2, \ldots, v_n$. Otherwise, go to step (2).

As one can see, the GPLHR algorithm is defined by two key components—generation of the search subspace and the eigenvector extraction procedure. Both components are different from the original Davidson method. GPLHR differs from the original harmonic projection schemes of Morgan[40,41] in the way the search subspaces are constructed. Specifically, Morgan considers the harmonic Rayleigh–Ritz procedure within the framework of single-vector Arnoldi and generalized Davidson algorithms, whereas GPLHR expands the subspace with preconditioned residuals and residual-like blocks.

The GPLHR algorithm allows for different choices of preconditioner *T*. In general, *T* should approximate the shifted and inverted operator, $(A-\eta)^{-1}$. In the present implementation, however, Davidson's preconditioner is used, that is, separate preconditioning operations $T_i=(D-\rho_i)^{-1}$ are applied to the respective columns when constructing *W* and $S^{(j)}$. This choice of preconditioning provides robust convergence in quantum-chemical calculations when diagonally dominant matrices are diagonalized.

Another feature of the algorithm is the use of harmonic projection, which results in a reduced eigenvalue problem (9). The eigenvalues θ_i of this problem are called the harmonic Ritz values. The corresponding vectors *ZX* are referred to as the harmonic Ritz vectors. Thus, at each iteration, GPLHR defines new approximate eigenvectors as the harmonic Ritz vectors associated with the *n* smallest (by absolute value) harmonic Ritz eigenvalues.

The harmonic Ritz pairs should be distinguished from the standard Ritz pairs that are used in the Davidson algorithm. The former are given by the solution of the reduced problem, eq. (9), and are known to yield better approximations to eigenpairs located deeper in the interior of the spectrum,[42] whereas the latter are obtained from the standard eigenvalue problem, $A_V X=X\Lambda$ [step (4) of Davidson's algorithm] and typically favor eigenpairs closer to the spectrum's exterior. To test whether using harmonic Ritz projection may improve the convergence of the Davidson method for interior eigen-states, we implemented a shifted harmonic Davidson eigensolver in which step (4) of the original Davidson procedure is replaced by steps (3) and (4) of GPLHR.

Although the left-hand side of eq. (9) is symmetric and positive (semi-) definite, the matrix on the right-hand side is nonsymmetric. Therefore, in general, eigenvalues of eq. (9) may appear in complex conjugate pairs. In this case, the associated harmonic Ritz vectors are also complex valued, that is, the solver should be capable of performing computations in complex arithmetic. However, in conventional EOM-CC methods, only real values have physical meaning and complex eigenvalues almost never occur. Similar to the Davidson procedure, if complex eigenpairs appear we assume that their imaginary parts are small (and they are, in practice). We consider only the real parts of the first complex conjugate pair, ignoring the second eigenvalue and eigenvector. Thus, the entire algorithm is performed in real arithmetic.

We note that GPLHR can be applied to the solution of complex-scaled and augmented by complex absorbing potentials EOM-CC[34–36] problems, where the eigenvalues and eigenvectors are complex and correspond to nonstationary resonance states.[27,28] Our GPLHR implementation can easily be extended to this situation, as will be pursued in future work.

Finally, note that the maximum size of the GPLHR search subspace is fixed and is at most $n(m+3)$ vectors (if no vectors are omitted after the orthogonalization and convergence check). This is different from the Davidson algorithm, where the subspace is expanded at each iteration until it has reached a user-specified maximum size. When the maximum is reached the space is collapsed and the algorithm is restarted.

The size of the GPLHR search subspace is controlled by the user when specifying the number of residual-like vectors, *m*. The amount of memory required by the GPLHR algorithm depends on the parameter *m* and, in our implementation, should be sufficient to store $3n(m+3)$ vectors, *Z*, *AZ*, and $(A-\eta)Z$, as well as *n* approximate eigenvectors, *V*. Note that the method can also be implemented without storing $(A-\eta)Z$, which will be pursued in future work.

Similar to the Davidson method, the most expensive steps in the GPLHR algorithm are matrix-vector multiplications, and their number depends on the size of the subspace. Since we store the additional subspaces, *AZ* and $(A-\eta)Z$, each GPLHR iteration can be implemented with at most $n(m+1)$ matrix-vector multiplications. This number is lower if vectors are dropped during orthogonalization or if some roots have already converged. Note that since vectors $w_i, s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(m)}$ are generated only for the unconverged roots, the size of the subspace, $k(m+3)+n$, shrinks and can become insufficient. For this reason, in our implementation, we automatically augment the parameter *m* by an integer part of $(n-k)/k$ if *n* – *k* (number of converged eigenpairs) is larger than *k* (number of unconverged roots).

Benchmarks

Computational details

All benchmarks were performed using the Q-Chem electronic structure package,[21–23] version 4.2. The algorithms presented above are implemented for the EOM-EE/EA/IP-CCSD methods using our general C++ libtensor library.[43] Details for how to specify the parameters for the eigensolvers within the EOM-CC family of methods in Q-Chem can be found in the user manual.[23]

The performance of the new algorithms was benchmarked using the following systems (see Fig. 1):

1. A hydrated photoactive yellow protein chromophore (PYPa-W$_p$ from Ref. [44], C_1 symmetry). We used the 6-31+G(d,p) basis set (292 basis functions).
2. Dihydrated 1,3-dimethyluracil (mU-$(H_2O)_2$ from Ref. [45], C_1 symmetry); the 6-311+G(d,p) basis set was used (336 basis functions).
3. Water dimer from Ref. [46] at two different interfragment distances (R_{OO} = 2.91 and 5.82 Å), C_s symmetry; the cc-pVTZ basis (116 basis functions) was used.
4. Dihydrated thymine T-$(H_2O)_2$ from Ref. [47], C_1 symmetry. We used the cc-pVDZ basis (215 basis functions).

All structures are from Refs. [44–47]; for convenience, we provide all Cartesian geometries in Supplemental Information.

We investigated the convergence behavior of the GPLHR and Davidson solvers within the EOM-IP-CCSD (systems 1–3) and EOM-EE-CCSD (system 4) methods.

The following convergence parameters were used in benchmarks: $||r||^2 < 10^{-5}$ as the convergence criterion for both the

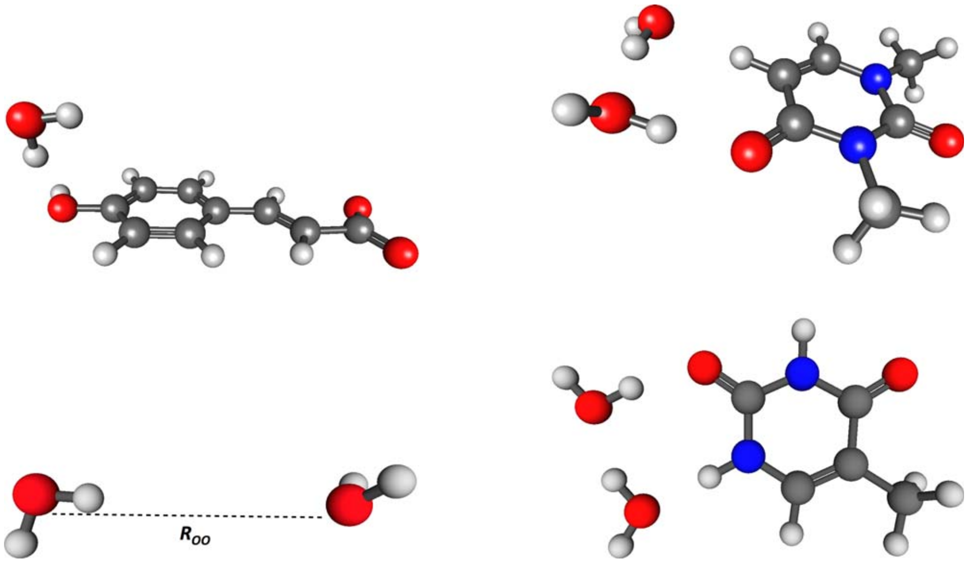

Figure 1. Benchmark systems: Hydrated photoactive yellow protein chromophore PYPa-W$_p$ (top left), dihydrated 1,3-dimethyluracil mU-$(H_2O)_2$ (top right), water dimer $(H_2O)_2$ (bottom left), and microhydrated thymine T-$(H_2O)_2$ (bottom right). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Davidson and GPLHR solvers, $||r||^2 < 5 \cdot 10^{-6}$ as the criterion to ignore insignificant vectors for the Davidson solver and $||r||^2 < 10^{-14}$ for GPLHR. The maximum number of iterations for both methods was set at 60; the maximum subspace size was set at 60 for the Davidson procedure. The residuals and residual-like vectors $w_i, s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(m)}$ were constructed only for unconverged roots. For converged roots, only the current approximation of the eigenvector and vector p_i were used to expand the GPLHR subspace. If the number of converged roots exceeded the number of unconverged roots in GPLHR, we increased the parameter m for the unconverged roots by an integer factor $\frac{\#\text{converged}}{\#\text{unconverged}}$, but kept it under 10. In the following benchmarks, we evaluate the convergence behavior of the solvers (the number of iterations to converge) as well as their computational (the number of matrix-vector operations) and memory (the number of stored vectors) costs.

Results and discussion

Table 1 compares the performance of the conventional Davidson and GPLHR solvers (no shift) when solving for several lowest roots. The GPLHR solver was executed with parameters m = 1, 2, 3, and 5 and the results with the smallest number of matrix-vector multiplications for each number of the requested roots are compiled in Table 1. One can see that, in all cases, the GPLHR solver converges to the requested number of roots in fewer iterations than Davidson's method. In some cases, the GPLHR solver converges with up to 3 times fewer iterations. However, since each iteration of the GPLHR solver is more expensive than an iteration of Davidson's solver, the overall number of matrix-vector multiplications for GPLHR is usually slightly (by 5–10%) higher than for Davidson's method. We note that for the chosen solver parameters, GPLHR requires significantly less memory to store subspace vectors Z and corresponding vectors AZ, as compared to Davidson's procedure. Note that memory requirements for both solvers can be controlled by changing the maximum subspace for the Davidson procedure and varying the number of residual-like vectors (m) for the GPLHR procedure.

Table 1. Comparison of the conventional Davidson and GPLHR (η = 0) solvers in converging various number of lowest EOM-IP-CCSD roots.

nroots[a]	niters[b]	m	Max. # of stored vectors[c]	# matvec[d]
PYPa-W$_p$/6-31+G(d,p) Davidson's method				
1	19		38	19
3[e]	12		60	30
5	9		74	37
10	20		120	101
GPLHR				
1	11	1	8	23
3[e]	4	1	24	25
5	4	1	40	41
10	8	1	80	117
(mU)$_2$-$(H_2O)_2$/6-311+G(d,p) Davidson's method				
1	6		14	7
3	8		48	24
5	8		76	38
10	15		120	99
12	13		120	118
GPLHR				
1	4	1	8	9
3	4	1	24	27
5	4	1	40	43
10	7	1	80	118
12	5	2	120	151

[a] The number of requested roots. [b] The number of iterations to converge all roots. [c] Only the subspace vectors Z and the corresponding σ-vectors AZ are taken into account. [d] The total number of matrix-vector multiplications (σ-vector updates). [e] In this case, Davidson's solver converged to the roots with energies 3.93, 4.11, and 4.20 eV, whereas GPLHR (m = 1) converged to 4.11, 4.20, and 4.87 eV. GPLHR with m = 3, 5, and 7 found the same roots as Davidson's solver.

Table 2. Comparison of the Davidson and GPLHR solvers in converging various number of roots around a specified energy shift (**η**= 11 a.u.) in EOM-IP-CCSD/6-31+G(d,p) calculations of PYPa-W_p.

nroots[a]	niters[b]	m	Max. # of stored vectors[c]	# matvec[d]
		Shifted Davidson's method		
1	DNC[e]		–	–
2	DNC[e]		–	–
3	DNC[e]		–	–
5	DNC[e]		–	–
		Harmonic shifted Davidson's method		
1	8		18	9
2	7		32	16
3	DNC[e]		–	–
5	DNC[e]		–	–
		GPLHR		
1	4	1	8	9
2	4	1	16	18
3	4	1	24	27
5	8	1	40	63

[a] The number of requested roots. [b] The number of iterations to converge all roots. [c] Only the subspace vectors Z and the corresponding σ-vectors AZ are taken into account. [d] The total number of matrix-vector multiplications (σ-vector updates). [e] Did not converge all roots in 60 iterations.

Table 3. Comparison of the Davidson and GPLHR solvers in converging two roots for various shifts around 10.6 a.u. in EOM-IP-CCSD/6-31+G(d,p) calculations of PYPa-W_p.

η[a]	niters[b]	m	Max. # of stored vectors[c]	# matvec[d]
		Davidson's method		
5.6	DNC[e]	–	–	–
6.6	6		28	14
9.6	6		28	14
10.1	6		28	14
10.6	6		28	14
11.1	DNC[e]	–	–	–
11.6	DNC[e]	–	–	–
		GPLHR		
5.6	DNC[e]	3	–	–
6.6	4	1	16	18
9.6	4	1	16	18
10.1	4	1	16	18
10.6	4	1	16	18
11.1	4	1	16	18
11.6	DNC[e]	3	–	–

[a] Energy shift in a.u. [b] The number of iterations to converge all roots. [c] Only the subspace vectors Z and the corresponding σ-vectors AZ are taken into account. [d] The total number of matrix-vector multiplications (σ-vector updates). [e] Did not converge all roots in 60 iterations.

Next, we compare the solvers' ability to converge interior eigenvalues when a large energy shift is specified (Table 2). Calculations are performed for PYPa-W_p with an energy shift of 11 a.u. (299.32 eV). The GPLHR solver easily converged a block of 1, 2, 3, and 5 roots around this shift. The energies for the block of 5 roots are 289.13, 290.20, 290.36, 290.58, and 291.12 eV; each root corresponds to a one-electron ionization dominated by a transition from one of the low-lying orbitals clustered around −(11.1–11.2) a.u. At the same time, Davidson's solver was unable to converge even a single root for a chosen energy shift within 60 iterations. Using the harmonic version of the shifted Davidson method improves convergence, but the block of 2 roots returned by this method contains a root with an energy of 288.66 eV, slightly lower than the lowest energy root returned by the GPLHR solver. The harmonic version of the shifted Davidson procedure still fails to converge when 3 and 5 roots are requested. We can conclude that the GPLHR solver provides a higher quality subspace with regard to the harmonic projection technique than the Davidson solver.

To assess the sensitivity of the GPLHR and Davidson solvers when locating eigenvalues near the specified energy shift, we perform a scan in which we start with a shift very close to the previously found solution, η = 11 a.u. (around 10.6 a.u.) and vary it. Results for solving for 2 roots using either the Davidson or GPLHR solver with various shifts around 10.6 a.u. are summarized in Table 3. We see that both the Davidson and GPLHR solvers easily converge two roots with a shift of 10.6 a.u. (in 6 and 4 iterations, respectively). Interestingly, when we decrease the shift, both the Davidson and GPLHR algorithms are able to converge to solutions with energies of 288.16 and 289.13 eV for shifts as low as 6.6 a.u. (around 179.60 eV). This result can be explained by the spacing of the molecular orbitals; there are 9 low-lying orbitals with energies around −(11.1–11.2) a.u., whereas the next orbital has an energy of −1.34 a.u. Because of this energy gap, one should anticipate a cluster of target states dominated by the ejection of an electron from an orbital with an energy around 300 eV (which we recovered with a shift of 11 a.u.), whereas other transitions will have much lower energy. Interestingly, by increasing the shift by just 0.5 a.u. (to 11.1 a.u.), Davidson's solver is unable to

Table 4. Convergence of the GPLHR solver for the different numbers of residual-like vectors.

nroots[a,e]	niters[b]	m	Max. # of stored vectors[c]	# matvec[d]
		PYPa-W_p/EOM-IP-CCSD/6-31+G(d,p)		
3	4	1	24	25
3	3	2	30	30
3	4	3	36	43
3	4	5	48	62
3	3	7	60	67
		mU-$(H_2O)_2$/EOM-IP-CCSD/6-311+G(d,p)		
3	4	1	24	27
3	3	2	30	30
3	3	3	36	39
3	3	5	48	50
		T-$(H_2O)_2$ /EOM-EE-CCSD/cc-pVDZ		
10	10	1	74	169
10	7	2	94	176
10	7	3	106	209
10	5	5	146	260
10	4	7	192	291

[a] The number of requested roots. [b] The number of iterations to converge all roots. [c] Only the subspace vectors Z and the corresponding σ-vectors AZ are taken into account. [d] The total number of matrix-vector multiplications (σ-vector updates). [e] m = 1,2 converged to the roots with energies 4.11, 4.20, and 4.87 eV; m = 3,5,7 converged to the roots with energies 3.93, 4.11, and 4.20 eV.

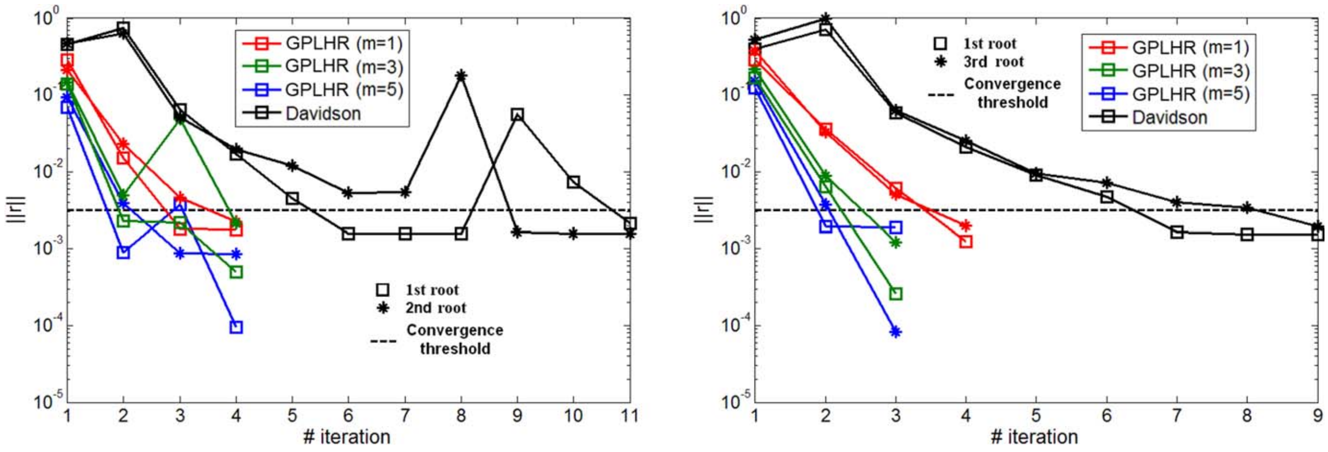

Figure 2. Norm of the residual for the two roots for the GPLHR and Davidson solvers at each iteration. Left: PYPa-W$_p$/6-31+G(d,p) for the roots with converged energies of 4.11 and 4.20 eV; right: mU-$(H_2O)_2$/6-311+G(d,p) for the roots with converged energies of 8.89 and 10.04 eV.

converge, whereas GPLHR easily finds the requested roots. We conclude that GPLHR is less sensitive to the choice of the energy shift than Davidson's algorithm.

As mentioned above, the size of the subspace in GPLHR can be controlled by the user by changing the number of residual-like vectors generated for unconverged roots. In Table 4, we compare how this parameter affects the convergence and the total number of matrix-vector multiplications in the EOM-IP-CCSD calculations of the three lowest states in PYPa-W$_p$ and mU-$(H_2O)_2$ and in the EOM-EE-CCSD calculations of the 10 lowest states of T-$(H_2O)_2$. We see that increasing the number of residual-like vectors may reduce the number of iterations. When only three states are requested (as in EOM-IP-CCSD calculations), varying m has virtually no effect on the number of iterations—all roots converge in 3–4 iterations. However, we observe that using smaller values on m may result in converging to higher-energy roots. For example, the GPLHR solver with $m = 1$, 2 yields roots with energies of 4.11, 4.20, and 4.87 eV for PYPa-W$_p$. When m is increased to 3, 5, or 7, the lower root is recovered with an energy of 3.93 eV (roots are 3.93, 4.11, and 4.20 eV for both the Davidson and GPLHR $m =$ 3, 5, 7 procedures). This explains why going from $m = 2$ to $m = 3$ slightly increases the number of iterations. In the former case, the solver was able to capture the lower lying root and needed more iterations to refine the solution. In the case of EOM-EE-CCSD/cc-pVDZ calculations of the 10 lowest states of T-$(H_2O)_2$, the dependence on m is more pronounced—increasing m from 1 to 5 reduces the number of iterations by a factor of 2. However, executions with small m are the most efficient in terms of matrix-vector multiplications and storage requirements, for example, in the above example, the number of matrix-vector multiplications increases by 50%, and the number of stored vectors by 97%.

Table 5. Convergence of Davidson's solver with user-defined guess in EOM-IP-CCSD calculations.

norb[a]	Orb. coeff. [b]	Energy[c]	niters[d]	Max. # of stored vectors[e]	# matvec[f]
		PYPa-W$_p$ /6-31+G(d,p)			
4	0.908	536.10	6	14	7
13	0.892	288.66	6	14	7
30	0.885	12.40	58	118	59
		mU-$(H_2O)_2$/6-311+G(d,p)			
4	–	–	DNC^g	–	–
10	0.901	293.86	5	12	6
28	0.920	16.77	35	72	36

[a] The number of user-specified guess orbital. [b] The r_1 coefficient of the user-specified orbital in the final solution. [c] Energy of the converged state (eV). [d] The number of iterations to converge all roots. [e] Only the subspace vectors Z and the corresponding σ-vectors AZ are taken into account. [f] The total number of matrix-vector multiplications (σ-vector updates). [g] Did not converge all roots in 60 iterations.

Figure 2 shows the norms of the residuals of the two EOM-IP roots at each iteration for the Davidson solver and for the GPLHR solver when various values of parameter m are used. We see that in all cases GPLHR gives much smoother convergence than Davidson's method. For example, even for the smallest subspace size ($m = 1$) the requested roots converged in fewer iterations and more monotonically than with Davidson's solver. As noted above, increasing the size of the subspace may improve convergence, but makes each iteration more computationally expensive.

As one can see, GPLHR with $m = 1$ is comparable in overall computational cost and wall time to Davidson's method, whereas increasing m makes the computations more expensive and also leads to higher storage requirements. Thus, although increasing the number of residual-like vectors may lead to a faster (and more robust, in terms of finding the correct roots) convergence, using $m > 1$ is only recommended for cases with problematic convergence.

Table 5 demonstrates how the Davidson procedure with a user-specified guess converges on roots of a requested transition character in the EOM-IP-CCSD calculations of PYPa-W$_p$ and mU-$(H_2O)_2$. We see that, in most cases, the modified Davidson method successfully converges the root dominated by a transition from the orbital specified by the user. In all cases, the respective amplitude in the EOM wave function (r_1 coefficient) exceeds 0.88 (the respective weight is 0.77). We also note that problematic convergence (e.g. transition from 30th orbital in

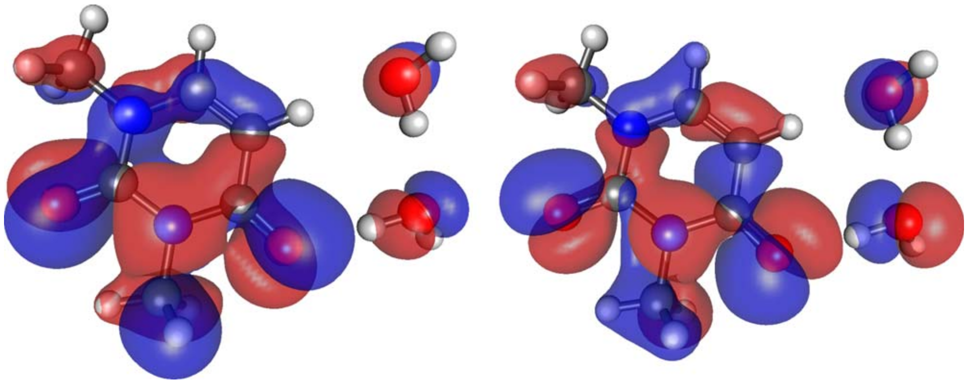

Figure 3. The third (left) and second (right) highest occupied molecular orbitals of mU-(H_2O)$_2$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

PYPa-W$_p$) arises when another orbital has considerable contributions to the wave function (r_1 larger than 0.1).

The utility of the modified Davidson procedure with the option of a user-specified guess vector is not limited to calculations of high-lying states; by eliminating the need to converge lower roots, it leads for computational savings even when the desired state is relatively low-lying. To illustrate this, let us consider two low-lying ionized states of mU-(H_2O)$_2$ corresponding to ionizations from the second (MO #46) and third (MO #45) HOMOs (Fig. 3). MO #46 can be described as a lone pair-type orbital, whereas MO #45 is a π-type orbital; both MOs are localized on the dimethyluracyl moiety. Ionization from MO #45 gives rise to the third lowest state in the EOM-IP-CCSD calculation with the r_1 coefficient of 0.946 and energy of 10.04 eV. The second lowest root is dominated by ionization from MO #46 with the r_1 coefficient of 0.950 and energy of 9.96 eV. The lowest ionized state arises from ionization of MO #47 with energy of 8.90 eV.

One can compute these states by using the standard Davidson procedure and requesting three roots, however, this requires more computational resources, as illustrated in Table 6. Although the total number of iterations required to converge all roots is less for the canonical Davidson procedure than for the modified Davidson procedure, the total cost of the calculation (i.e., the number of matrix-vector multiplications and the maximum number of stored vectors) is less for the modified algorithm. For example, the number of matrix-vector multiplications is reduced by 21% when solving for the state corresponding to ionization from MO #46. The difference in cost between calculating the root dominated by MO #45 using the modified Davidson algorithm and the canonical procedure that solves for all three lowest roots is 50%.

To investigate whether the performance of the new algorithms may be affected by (near)-degeneracies of the target states, we consider ionized states of the water dimer. Figure 4 shows energies and the respective MOs of the four lowest ionized states of the water dimer at its equilibrium geometry (OO distance, R_{OO}, of 2.91 Å) and at a stretched geometry (R_{OO} = 5.82 Å). As the interfragment separation increases, the overlap between the fragments' MOs decreases, asymptotically leading to the two-fold degeneracy of each ionized state. Table 7 compares the performance of the canonical Davidson and GPLHR procedures. We observe that both methods are robust and their performance is not affected by the increased degeneracy between the target states.

As a more challenging example, we consider core-ionized states of the dimer corresponding to ionization from O(1s). Because core orbitals are more compact than valence orbitals, their overlap is small leading to small splittings between the

Table 6. Cost of calculating two and three EOM-IP-CCSD roots (canonical Davidson procedure) versus cost of calculating the second or third roots only (modified Davidson procedure with a user-specified guess) for mU-(H_2O)$_2$.

nroots[a]	niters[b]	Max. # of stored vectors[c]	# matvec[d]
		Canonical Davidson procedure	
2	6	28	14
3	8	48	24
		Modified Davidson procedure	
46[e]	10	22	11
45[e]	11	24	12

[a] The number of requested roots. [b] The number of iterations to converge all roots. [c] Only the subspace vectors *Z* and the corresponding σ-vectors *AZ* are taken into account. [d] The total number of matrix-vector multiplications (σ-vector updates). [e] The user-specified guess orbital.

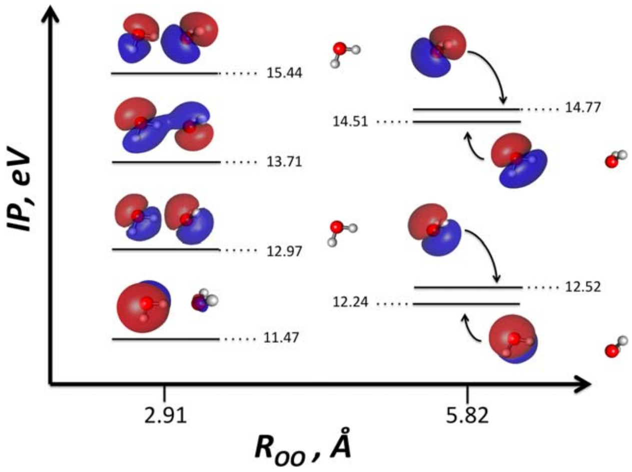

Figure 4. Four lowest valence ionized states of water dimer at the equilibrium (R_{OO} = 2.91 Å) and stretched (R_{OO} = 5.82 Å) geometries.

Table 7. Performance of Davidson and GPLHR solvers for EOM-IP calculations of the four lowest ionized states of water dimer at different geometries.

R_{OO}, Å	niters[a]	Max. # of stored vectors[b]	# matvec[c]
	Canonical Davidson procedure		
2.91	6	50	25
5.82	5	48	24
	GPLHR		
2.91	3	42	44
5.82	3	42	48

[a] The number of iterations to converge all roots. [b] Only the subspace vectors *Z* and the corresponding σ-vectors *AZ* are taken into account. [c] The total number of matrix-vector multiplications (σ-vector updates).

Table 9. Energies and dominant amplitudes for 10 lowest EOM-EE-CCSD transitions of T-$(H_2O)_2$ using the cc-pVDZ basis.

Root	E_{ex}, eV	Occ. orb.[a]	Virt. orb[b]	Amplitude/weight[c]
1	5.31	41	1	0.5895/0.69
2	5.69	43	1	0.6374/0.81
3	6.93	40	3	0.4209/0.35
4	7.06	42	1	0.4793/0.46
5	7.17	43	3	0.5130/0.53
6	7.84	41	3	0.4413/0.39
7	7.96	43	2	0.5487/0.60
8	8.15	42	3	0.6285/0.79
9	8.15	40	1	0.5165/0.53
10	8.68	39	2	0.3401/0.23

[a] MO # of the occupied orbital that gives rise to the dominant amplitude. [b] MO # of the virtual orbital that gives rise to the dominant amplitude. [c] Absolute value of the dominant r_1 amplitude; the weight is $2\times r_1^2$.

respective target states. The core IEs at the two dimer geometries are 541.5 and 539.7 eV (at R_{OO} = 2.91 Å) and 540.9 and 540.6 eV (at R_{OO} = 5.82 Å). The results are summarized in Table 8. Again, the performance of both variants of the Davidson method and of GPLHR is not affected by the degeneracies. Although it is not surprising that both methods converge well when two nearly-degenerate roots are requested, we note that both algorithms successfully tackle a potentially problematic situation in which only one root (out of the doubly degenerate manifold) is requested. The Davidson solver converges at the same number of iterations regardless of whether a user-specified guess or energy shift is used, and the convergence pattern is the same at both geometries. The GPLHR solver performs similarly.

The performance of the solvers within EOM-EE-CCSD is investigated using the T-$(H_2O)_2$ example. Table 9 summarizes the energies and the leading r_1 amplitudes of the 10 lowest singlet EOM-EE states. We note that the weights of the leading configurations are smaller than in the case of EOM-IP calculations (excited states show more configuration mixing than ionized states).

Table 10 compares the convergence of the conventional Davidson and GPLHR solvers when solving for several low-lying roots. The convergence pattern of the two solvers in EOM-EE-CCSD is similar to that observed in EOM-IP-CCSD. GPLHR converges in fewer iterations and requires less storage, but the overall cost is slightly higher. The dependence of GPLHR on the number of residual-like vectors was analyzed above (see Table 4). As in the EOM-IP calculations, the GPLHR solver may occasionally miss a lower energy root converging to slightly higher states.

The results for shifted Davidson and GPLHR procedures (η = 7 eV) are summarized in Table 11. Due to a high density of excited states, this is a challenging example. We note that when using the default thresholds, both methods may miss roots. Tightening the threshold in the shifted Davidson procedure recovers the correct roots, however, leads to convergence

Table 8. Convergence of Davidson and GPLHR solvers for core-ionized states of water dimer.

R_{OO}[a]	nroots[b]	norb[c]	η[d]	niters[e]	Max. # of stored vectors[f]	# matvec[g]
			Davidson with guess or user-defined shift			
2.91	1	1	–	5	12	6
2.91	1	2	–	5	12	6
2.91	1	–	19.9	5	12	6
2.91	2	–	19.9	5	24	12
5.82	1	1	–	5	12	6
5.82	1	2	–	5	12	6
5.82	1	–	19.9	5	12	6
5.82	2	–	19.9	5	24	12
			GPLHR (η = 19.9 a.u.)			
2.91	1			3	12	13
2.91	2			3	24	26
5.82	1			3	12	13
5.82	2			3	22	26

[a] The internuclear separation of the two oxygen atoms in angstroms. [b] The number of requested roots. [c] The number of the user-specified guess orbital, if applicable. [d] Energy shift in a.u. [e] The number of iterations to converge all roots. [f] Only the subspace vectors *Z* and the corresponding σ-vectors *AZ* are taken into account. [g] The total number of matrix-vector multiplications (σ-vector updates).

Table 10. Comparison of the conventional Davidson and GPLHR (η = 0) solvers in converging various numbers of lowest EOM-EE-CCSD roots for T-$(H_2O)_2$ using the cc-pVDZ basis.

nroots[a]	niters[b]	*m*	Max. # of stored vectors[c]	# matvec[d]
		Davidson's method		
1	8		20	10
3^e	24		108	54
5	24		120	97
10	21		120	178
		GPLHR		
1	3	3	12	14
3[e]	5,5	3,5	32,42	55,80
5	8	3	52	128
10	7	3	106	209

[a] The number of requested roots. [b] The number of iterations to converge all roots. [c] Only the subspace vectors *Z* and the corresponding σ-vectors *AZ* are taken into account. [d] The total number of matrix-vector multiplications (σ-vector updates). [e] When three roots are requested, the Davidson solver returns EOM-EE transitions with energies equal to 5.31, 5.69, and 6.93 eV, while the GPLHR solver returns roots with energies of 5.31, 5.69, and 7.96 eV when *m* = 3. With *m* = 5, the GPLHR solver recovers the same three roots as the Davidson solver.

Table 11. Convergence of modified Davidson and GPLHR solvers for computing two EOM-EE-CCSD roots around $\eta = 7$ eV for T-$(H_2O)_2$ using the cc-pVDZ basis set.

Thresh[a]	m	E_{ex},eV	niters[a]	Max. # of stored vectors[b]	# matvec[c]
			Davidson's method		
1e-05		6.93, 7.84	38	120	68
1e-09		6.93, 7.06	57	120	111
			GPLHR		
1e-05	3	5.69, 7.96	4	22	31
1e-09	3	–	DNC^e	–	–
1e-09	5	6.93, 7.84	27	32	267

[a] Threshold for EOM-EE convergence. [b] The number of iterations to converge all roots. [c] Only the subspace vectors Z and the corresponding σ-vectors AZ are taken into account. [d] The total number of matrix-vector multiplications (σ-vector updates). [e] Solver did not converge within 60 iterations.

problems in GPLHR. Increasing m from 3 to 5 (in addition to tightening the convergence threshold) solves the convergence problems and allows one to recover correct roots.

Finally, we test the behavior of the Davidson procedure with a user-specified guess. As one can see from Table 9, the weights of the dominant configurations are smaller for the EOM-EE states, as compared to EOM-IP. For example, for roots 3, 6, and 10, the weight of the dominant configuration is below 50%—obviously, these states cannot be targeted by this algorithm. We note that even for the states in which the weight of the dominant configuration exceeds 50%, one cannot guarantee that the procedure will converge to the lowest state (because there might be a higher state that is also dominated by the same transition, but the weight of the leading configuration is higher).

We performed calculations with user-specified guess for roots 2, 4, 5, 7, and 8. In all cases, the Davidson procedure converged to the desired state with the same r_1 amplitudes as in the reference calculation of 10 states. The number of iterations varies from 8 (for state 2) to 32 (state 5); for other states, the procedure converged in 21–22 iterations. As in the mU-$(H_2O)_2$ example above, the ability to solve for a single root (instead of a larger block) leads to significant reduction of storage requirements and computational costs.

Conclusions

Two modifications of the canonical Davidson method are presented. The first modification enables the user to find multiple roots around a specified energy shift (Davidson method with shift); the second allows the user to solve for a single root of a specific character (Davidson method with a user-defined guess). The first version (shifted Davidson) has been also implemented using harmonic Ritz projection. This new functionality enables calculations of high energy roots that would be inaccessible by the standard method, which is designed to find only a few lowest energy transitions. For example, if an experimental or theoretical estimate of the energy of a particular transition is available one can use Davidson's method with energy shift to calculate the transitions around that chosen energy. Additionally, if one is interested in finding a root dominated by a particular transition (e.g., ionization of or excitation from core orbitals), the Davidson method with a user-defined guess can be used. Moreover, this feature leads to computational savings for low-lying roots.

We also present an implementation of a new solver for non-Hermitian matrices—the GPLHR method.[20] This solver is also capable of finding roots around a specified energy shift. The subspace size of the GPLHR solver can be controlled by the user by choosing the number of residual-like vectors generated for each unconverged root. A larger subspace size usually leads to more rapid convergence (fewer number of iterations) and more reliable recovery of closest eigenvalues, but requires more memory and more matrix-vector multiplications per iteration. The methods are implemented for the EOM-CCSD family of methods (EA, EE, and IP) in the Q-Chem[21–23] electronic structure package.

New solvers were tested on several benchmark systems using the EOM-IP-CCSD and EOM-EE-CCSD methods and, in most cases, successfully find the roots targeted by the user. We observe that, in general, GPLHR with zero shift converges in fewer number of iterations than the conventional Davidson solver, however, the cost of each iteration of GPLHR is higher than that of Davidson's method. Thus, the overall number of time-limiting operations (matrix-vector multiplications) is slightly higher for GPLHR, depending on the number of residual-like vectors m. When interior eigenvalues around a specified energy shift are requested, GPLHR yields better and more robust convergence than the shifted Davidson method (even when the latter employs harmonic Ritz eigenproblem), which sometimes failed to find the requested solution. Therefore, while GPLHR displayed faster convergence (in terms of number of iterations) and was comparable to Davidson's method with regard to the cost of finding several lowest roots, it was found to be superior in terms of robustness when used to find interior eigenvalues corresponding to higher excitations.

Keywords: diagonalization algorithms · interior eigenstates · eigensolvers · equation-of-motion · coupled-cluster · excited states · harmonic Ritz problem

How to cite this article: D. Zuev, E. Vecharynski, C. Yang, N. Orms, A. I. Krylov. *J. Comput. Chem.* **2015**, *36*, 273–284. DOI: 10.1002/jcc.23800

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

[1] Y. Saad, Numerical Methods for Large Eigenvalue Problems, 2nd ed.; SIAM, Philadelphia, PA, **2001**.

[2] A. Szabo, N. S. Ostlund, Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory; McGraw-Hill: New York, **1989**.

[3] T. Helgaker, P. Jørgensen, J. Olsen, Molecular Electronic Structure Theory; Wiley: Hoboken, NJ, **2000**.

[4] H. Sekino, R. J. Bartlett, *Int. J. Quantum Chem.* **1984**, *26*, 255.

[5] J. F. Stanton, R. J. Bartlett, *J. Chem. Phys.* **1993**, *98*, 7029.

[6] A. I. Krylov, *Annu. Rev. Phys. Chem.* **2008**, *59*, 433.

[7] A. Dreuw, M. Wormit, *WIREs Comput. Mol. Sci.* (in press), doi:10.1002/wcms.1206.

[8] S. Hirata, M. Head-Gordon, *Chem. Phys. Lett.* **1999**, *314*, 291.

[9] J. G. F. Francis, *Comput. J.* **1961**, *4*, 265.

[10] J. G. F. Francis, *Comput. J.* **1962**, *4*, 332.

[11] G. H. Golub, C. F. Van Loan, Matrix Computations; Johns Hopkins University Press: Baltimore, MD, **1996**.

[12] W. E. Arnoldi, *Quart. Appl. Math*. **1951**, *9*, 17.

[13] C. Lanczos, *J. Res. Natl. Bur. Stand*. **1950**, *45*, 255.

[14] E. R. Davidson, *J. Comput. Phys*. **1975**, *17*, 87.

[15] R. Franchy, D. Menzel, *Phys. Rev. Lett*. **1979**, *43*, 865.

[16] J. F. Morar, F. J. Himpsel, G. Hollinger, J. L. Jordan, G. Hughes, F. R. McFeely, *Phys. Rev. B* **1986**, *33*, 1340.

[17] P.-F. Loos, X. Assfeld, *Int. J. Quantum Chem*. **2007**, *107*, 2243.

[18] E. Epifanovsky, I. Polyakov, B. L. Grigorenko, A. V. Nemukhin, A. I. Krylov, *J. Chem. Theory Comput*. **2009**, *5*, 1895.

[19] J. F. Stanton, J. Gauss, M.E. Harding, P.G. Szalay, In CFOUR, Coupled Cluster techniques for Computational Chemistry, a quantum-chemical program package, Available at: www.cfour.de.

[20] E. Vecharynski, F. Xue, C. Yang, Computing interior eigenpairs of nonhermitian matrices, In: Presentation on the 1st Annual "SciDAC Scientific Discovery through Advanced Computing" investigator meeting at UC, Bekeley, LBNL, Berkeley, **2013**.

[21] Y. Shao, L. Fusti-Molnar, Y. Jung, J. Kussmann, C. Ochsenfeld, S. Brown, A. T. B. Gilbert, L. V. Slipchenko, S. V. Levchenko, D. P. O'Neill, R. A. Distasio, Jr, R. C. Lochan, T. Wang, G. J. O. Beran, N. A. Besley, J. M. Herbert, C. Y. Lin, T. Van Voorhis, S. H. Chien, A. Sodt, R. P. Steele, V. A. Rassolov, P. Maslen, P. P. Korambath, R. D. Adamson, B. Austin, J. Baker, E. F. C. Byrd, H. Daschel, R. J. Doerksen, A. Dreuw, B. D. Dunietz, A. D. Dutoi, T. R. Furlani, S. R. Gwaltney, A. Heyden, S. Hirata, C.-P. Hsu, G. S. Kedziora, R. Z. Khalliulin, P. Klunziger, A. M. Lee, W. Z. Liang, I. Lotan, N. Nair, B. Peters, E. I. Proynov, P. A. Pieniazek, Y. M. Rhee, J. Ritchie, E. Rosta, C. D. Sherrill, A. C. Simmonett, J. E. Subotnik, H. L. Woodcock, III, W. Zhang, A. T. Bell, A. K. Chakraborty, D. M. Chipman, F. J. Keil, A. Warshel, W. J. Hehre, H. F. Schaefer, III, J. Kong, A. I. Krylov, P. M. W. Gill, M. Head-Gordon, *Phys. Chem. Chem. Phys*. **2006**, *8*, 3172.

[22] A. I. Krylov, P. M. W. Gill, *WIREs Comput. Mol. Sci*. **2013**, *3*, 317.

[23] Y. Shao, Z. Gan, E. Epifanovsky, A. T. B. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Kus, A. Landau, J. Liu, E. I. Proynov, Y. M. Rhee, R. M. Richard, M. A. Rohrdanz, R. P. Steele, E. J. Sundstrom, H. L. Woodcock, III, P. M. Zimmerman, D. Zuev, B. Albrecht, E. Alguires, B. Austin, G. J. O. Beran, Y. A. Bernard, E. Berquist, K. Brandhorst, K. B. Bravaya, S. T. Brown, D. Casanova, C.-M. Chang, Y. Chen, S. H. Chien, K. D. Closser, D. L. Crittenden, M. Diedenhofen, R. A. DiStasio, Jr., H. Do, A. D. Dutoi, R. G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M. W. D. Hanson-Heine, P. H. P. Harbach, A. W. Hauser, E. G. Hohenstein, Z. C. Holden, T.-C. Jagau, H. Ji, B. Kaduk, K. Khistyaev, J. Kim, J. Kim, R. A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C. M. Krauter, K. U. Laog, A. Laurent, K. V. Lawler, S. V. Levchenko, C. Y. Lin, F. Liu, E. Livshits, R. C. Lochan, A. Luenser, P. Manohar, S. F. Manzer, S.-P. Mao, N. Mardirossian, A. V. Marenich, S. A. Maurer, N. J. Mayhall, C. M. Oana, R. Olivares-Amaya, D. P. O'Neill, J. A. Parkhill, T. M. Perrine, R. Peverati, P. A. Pieniazek, A. Prociuk, D. R. Rehn, E. Rosta, N. J. Russ, N. Sergueev, S. M. Sharada, S. Sharmaa, D. W. Small, A. Sodt, T. Stein, D. Stuck, Y.-C. Su, A. J. W. Thom, T. Tsuchimochi, L. Vogt, O. Vydrov, T. Wang, M. A. Watson, J. Wenzel, A. White, C. F. Williams, V. Vanovschi, S. Yeganeh, S. R. Yost, Z.-Q. You, I. Y. Zhang, X. Zhang, Y. Zhou, B. R. Brooks, G. K. L. Chan, D. M. Chipman, C. J. Cramer, W. A. Goddard, III, M. S. Gordon, W. J. Hehre, A. Klamt, H. F. Schaefer, III, M. W. Schmidt, C. D. Sherrill, D. G. Truhlar, A. Warshel, X. Xu, A. Aspuru-Guzik, R. Baer, A. T. Bell, N. A. Besley, J.-D. Chai, A. Dreuw, B. D. Dunietz, T. R. Furlani, S. R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D. S. Lambrecht, W. Z. Liang, C. Ochsenfeld, V. A. Rassolov, L. V. Slipchenko, J. E. Subotnik, T. Van Voorhis, J. M. Herbert, A. I. Krylov, P. M. W. Gill, M. Head-Gordon, *Mol. Phys.* (in press), doi:10.1080/00268976.2014.952696.

[24] M. Head-Gordon, T. J. Lee, In Modern Ideas in Coupled Cluster Theory, R. J. Bartlett, Ed.; World Scientific: Singapore, **1997**.

[25] K. Hirao, H. Nakatsuji, *J. Comput. Phys*. **1982**, *45*, 246.

[26] G. L. G. Sleijpen, H. A. Van der Vorst, *SIAM Rev* **2000**, *42*, 267.

[27] W. P. Reinhardt, *Annu. Rev. Phys. Chem*. **1982**, *33*, 223.

[28] S. Klaiman, I. Gilary, *Adv. Quantum Chem*. **2012**, *63*, 1.

[29] E. Balslev, J. M. Combes, *Commun. Math. Phys*. **1971**, *22*, 280.

[30] B. Simon, *Commun. Math. Phys*. **1972**, *27*, 1.

[31] R. Kosloff, D. Kosloff, *J. Comput. Phys*. **1986**, *63*, 363.

[32] U. V. Riss, H.-D. Meyer, *J. Phys. B* **1993**, *26*, 4503.

[33] U.V. Riss, H.-D. Meyer, *J. Phys. B* **1995**, *28*, 1475.

[34] K. B. Bravaya, D. Zuev, E. Epifanovsky, A. I. Krylov, *J. Chem. Phys*. **2013**, *138*, 124106.

[35] T.-C. Jagau, D. Zuev, K. B. Bravaya, E. Epifanovsky, A. I. Krylov, *J. Phys. Chem. Lett*. **2014**, *5*, 310.

[36] D. Zuev, T.-C. Jagau, K. B. Bravaya, E. Epifanovsky, Y. Shao, E. Sundstrom, M. Head-Gordon, A. I. Krylov, *J. Chem. Phys*. **2014**, *141*, 024102.

[37] N. Moiseyev, P. R. Certain, F. Weinhold, *Mol. Phys*. **1978**, *36*, 1613.

[38] N. Moiseyev, *Phys. Rep*. **1998**, *302*, 212.

[39] C. W. McCurdy, ACS Symposium Series, Vol. *263*; American Chemical Society: Washington, DC, **1984**; pp. 17–34.

[40] R. B. Morgan, *Linear Algebra Appl*. **1991**, *154*, 289.

[41] R. B. Morgan, M. Zeng, *Numer. Linear Algebra Appl*. **1998**, *5*, 33.

[42] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, Eds. Templates for the Solution of Algebraic Eigenvalue Problems. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, **2000**.

[43] E. Epifanovsky, M. Wormit, T. Kuś, A. Landau, D. Zuev, K. Khistyaev, P. Manohar, I. Kaliman, A. Dreuw, A. I. Krylov, *J. Comput. Chem*. **2013**, *34*, 2293.

[44] D. Zuev, K. Bravaya, M. Makarova, A. I. Krylov, *J. Chem. Phys*. **2011**, *135*, 194304.

[45] K. Khistyaev, A. Golan, K. B. Bravaya, N. Orms, A. I. Krylov, M. Ahmed, *J. Phys. Chem. A* **2013**, *117*, 6789.

[46] P. A. Pieniazek, J. VandeVondele, P. Jungwirth, A. I. Krylov, S. E. Bradforth, *J. Phys. Chem. A* **2008**, *112*, 6159.

[47] K. Khistyaev, K. B. Bravaya, E. Kamarchik, O. Kostko, M. Ahmed, A. I. Krylov, *Faraday Discuss*. **2011**, *150*, 313.

Received: 8 July 2014
Revised: 25 September 2014
Accepted: 2 November 2014
Published online on 2 December 2014