The calculation of molecular double ionization spectra by Green’s functions

Francesco Tarantelli *

Dipartimento di Chimica, Università di Perugia, Via Elce di Sotto 8, 06123 Perugia, Italy
I.S.T.M.-C.N.R., Via Elce di Sotto 8, 06123 Perugia, Italy

Received 29 April 2006; accepted 3 July 2006
Available online 6 July 2006

Abstract

The Green’s function ADC(2) has long proved to be a successful method for the study of dense double ionization spectra of molecules and clusters. This paper focuses on the computational aspects of the method, illustrating in particular the algorithms it entails in the context of subspace iteration techniques to compute eigenvalues and eigenvectors, and how these can be efficiently implemented. It is shown that, as in direct-CI procedures, the key matrix-multiply operation is effectively reduced to a number of smaller cache-size operations with matrix blocks built on-the-fly from the two-electron integrals, which can be implemented through fast basic linear algebra routines. Some parallelization aspects and specific diagonalization procedures are briefly discussed.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Green’s function; Algebraic diagrammatic construction; Computational chemistry; Algorithms; Double ionization

1. Introduction

The accurate calculation of even moderately excited wavefunctions and energies by ab initio methods is a notoriously awkward and time consuming task. Therefore, the theoretical study of high energy electronic spectra, in spite of its importance for the advancement of knowledge of phenomena, structure and chemistry of molecules and materials, is difficult by accurate wavefunction techniques and becomes near-impossible for dense spectra such as molecular inner valence ionization or multiple ionization ones, as revealed for example by photoelectron and decay spectroscopies. In these experiments, typically no strict selection rules apply and the number of observable states is large for all but the smallest molecules. In this context, the adoption of theoretical formalisms based on the so-called Green’s functions (GF), or propagators [1], is very useful because these methods target directly the spectroscopic observables (transition energies and spectroscopic factors) rather than stationary wavefunctions and energies. Just to mention two familiar examples, for computing excitation spectra at the simplest level of theoretical approximation, random phase approximation (RPA) [2] calculations are a standard tool of most computational packages while, in recent years, time-dependent density functional theory (TDDFT) (see Ref. [3] and references therein) has become a popular method for large molecular systems (see, e.g., Refs. [4,5]).

In the development and application of accurate Green’s function-based methods for the calculation of excitation and, especially, ionization spectra, important and innovative contributions, assiduous over the last three decades, have come from the work of Cederbaum, Schirmer and coworkers, of which an incomplete overview may be pieced together from, e.g., Refs. [6–15] and the papers cited therein. In particular, the algebraic diagrammatic construction (ADC) formalism [7,8,10,11,13] has proved to be a powerful tool in the study of ionization spectra, paving the road
towards important insights into phenomena such as, to cite but few examples, the correlation-induced breakdown of main lines in inner-valence photoionization spectra [16], the stability of gas-phase molecular negative ions [17], foreign-imaging in Auger spectroscopy [18] and the intermolecular coulombic decay (ICD) [19]. Recent important developments include the introduction of a complex absorbing potential for the one-particle GF ADC [14], enabling the study of decaying-state lifetimes, and the derivation of a “non-Dyson” ADC scheme, again for the one-particle GF, which, by avoiding the explicit use of the Dyson equation, achieves the enormous simplification of decoupling the $N - 1$ and $N + 1$ electron problems [20].

Among ionization spectroscopies, those producing dicationic species are especially useful tools for chemical analysis, witness the long-standing importance of Auger spectroscopy [21], as well as sensitive probes for a deeper understanding of electronic structure, energy and charge transfer processes. Experimental techniques to produce and detect dications, including double charge transfer (DCT) (see, e.g., [22]) and direct double photoionization (see, e.g., Ref. [23]) have continuously been developed and improved, in particular in combination with the coincidence detection of electrons, and/or molecular fragments (see, e.g., Ref. [24]). The theoretical simulation and interpretation of molecular double ionization spectra afforded by the ADC scheme for the particle–particle propagator [10] again proves very valuable here (for recent examples, see Refs. [25–27]), especially in conjunction with a simple computational device for the analysis of the double-hole density [28]. In this paper, we shall document in some detail the implementation of the second order ADC scheme, ADC(2), used for the routine calculation of double ionization spectra, focusing on the main algorithmic aspects and strategies adopted. Parts of these concepts and procedures apply equally well to the implementation of ADC schemes for other propagators, as used for the theoretical description of single [29] and triple ionization spectra [30], and are analogous to those used in other well-known methods of quantum chemistry such as direct Configuration Interaction (CI) (see Refs. [31,32] and references therein).

2. Theoretical overview

The ADC theory leads in general to the formulation of an hermitian eigenvalue equation, the solutions of which yield the approximate poles and residues of the target propagator. In the case discussed here, this is the $N - 2$-particle part of the particle–particle propagator, which, in energy space (and except of an infinitesimal imaginary shift of the energy variable $\omega$), reads

$$\Pi^{(\omega)} = X^{\dagger}(\omega I + A)^{-1}X$$

(1)

It is a matrix whose rows and columns run over pairs of one-particle indices (spin–orbitals). $I$ is the identity matrix and $A$ is the diagonal matrix of double ionization energies $\Delta_n$ from a reference $N$-electron state $\left| \psi_0^{(N)} \right>$ (the ground state) to a dicationic state $\left| \psi_n^{(N-2)} \right>$, while $X$ is the matrix of spectroscopic factors (residue amplitudes)

$$x_{nij} = \left< \psi_0^{(N)} \right| a_i^\dagger a_j^\dagger \left| \psi_n^{(N-2)} \right>$$

(2)

and $a_i^\dagger$ ($a_i$) are fermion creation (annihilation) operators in the given one-particle basis. As is evident, the propagator Eq. (1) has the double ionization energies of the system as infinities. The ADC formulation stems from the ansatz [10]

$$\Pi^{(\omega)} = F^{\dagger}(\omega I - \Gamma)^{-1}F$$

(3)

where $\Gamma$ is a (non-diagonal) hermitian matrix constructed over the space of $N - 2$ electron configurations (or, more generally, over suitable intermediate states [33]), rather than exact dication states, and proceeds to derive an appropriate perturbation expansion for the matrices $\Gamma$ and $F$, based on the unperturbed system represented by the ground state Hartree–Fock hamiltonian, by comparison with the corresponding diagrammatic perturbation expansion of the propagator. Because the double ionization energies are then obtained as eigenvalues of the matrix $\Gamma$ (and the spectroscopic factors by transforming the matrix $F$ over the eigenvectors), the method ensures that, at a chosen order $n$ of perturbation, the corresponding ADC($n$) scheme incorporates an infinite partial sum of the perturbation series for $\Pi^{(\omega)}$ which is, however, complete through $n$th order. The configuration space of the matrix $\Gamma$ at each order of perturbation is finite and specified as follows. The configurations are classified with respect to the $N$-electron Hartree–Fock ground state $|\phi_0\rangle$ according to the number of creation/destruction operators applied. If $i,j,k,\ldots$ denote occupied spin–orbital indices (holes) and $r,s,t,\ldots$ virtual ones (particles), then configurations of the type $|\phi_{ijkl}\rangle = a_i^\dagger a_j^\dagger a_k a_l |\phi_0\rangle$ are “two-hole” (2h) configurations, $|\phi_{ijk}\rangle = a_i^\dagger a_j^\dagger a_k |\phi_0\rangle$ are “three-hole-one-particle” (3h1p), and so on. In zeroth and first order, the $\Gamma$ matrix elements involve just the 2h configurations. For a non-correlated ground state, these are the sole configurations yielding non-zero spectroscopic factor contributions, and they are called main configurations. At higher orders of perturbation theory the $\Gamma$ matrix extends over progressively higher-excited (satellite) configurations, and it turns out that the configuration space increases by one configuration class at each even order [10]. The matrix elements of the 2h/2h block of $\Gamma$ are of $n$th order in the perturbation (i.e., in the coulomb two-electron integrals), while the order of the other matrix blocks decreases progressively by one for each level of particle-hole excitation involved (either row-wise or column-wise). In the second-order scheme discussed here, just the two lowest order configuration classes, 2h and 3h1p, are involved. The 3h1p/3h1p block is accordingly of zeroth order (diagonal) but it is promoted to first order for increased accuracy, especially in order to have a sensible description of the lowest excited (satellite) dicationic states.

For completeness, we report here the $\Gamma$ matrix elements to second order of ADC extracted from Ref. [10], with a
change of sign to obtain positive eigenvalues. The coulomb two-electron integrals over the spin-orbitals \{\phi_i\} are written as

\[ V_{ijkl} = \left\langle \phi_i(1)\phi_j(2) \right| \frac{1}{|r_i - r_j|} \left| \phi_k(1)\phi_l(2) \right\rangle \]  

(4)

and we use \( V_{ij|kl} = V_{ijkl} - V_{ijk}\), while \( \epsilon_j \) denote the orbital energies and \( \epsilon_{ijk} = \epsilon_i + \epsilon_j + \epsilon_k + \cdots \). For the 2h/2h block, the matrix elements read

\[ \Gamma_{ij|kl} = V_{ij|kl} + \left\{ \delta_{ji} \frac{1}{2} \sum_{rs} \left[ V_{rs|kl} V_{lm|jr} \right] (\epsilon_m - \epsilon_l) (\epsilon_m - \epsilon_m - \epsilon_{ij}/2) \right\} - \left\{ i \leftrightarrow j \right\} - \left\{ k \leftrightarrow l \right\} + \left\{ i \leftrightarrow j, k \leftrightarrow l \right\} \]

\[- \frac{1}{2} \sum_{rs} \left( V_{rs|kl} V_{ij|r} \right) \left[ (\epsilon_m - \epsilon_{ij}) (\epsilon_m - \epsilon_{ij}) (\epsilon_{ij}/2) - \delta_{ik} \delta_{jl} \right] \]

(5)

The 3h1p/2h block reads (correcting a misprint in Eq. (C8) of Ref. [10]):

\[ \Gamma_{rkm|ij} = \left\{ \delta_{ik} V_{j|m|rl} + \delta_{il} V_{m|kr} + \delta_{lm} V_{i|kl} \right\} - \left\{ i \leftrightarrow j \right\} \]  

(6)

and finally the 3h1p/3h1p block is:

\[ \Gamma_{rjklmn} = \left\{ \delta_{ij} \left[ \delta_{kl} V_{j|km} + \delta_{lm} V_{j|kl} + \delta_{mn} V_{j|lm} \right] \right\} + \left\{ (i, j, k) \rightarrow (j, k, i) \right\} + \left\{ (i, j, k) \rightarrow (k, i, j) \right\} + \left\{ \delta_{lm} \delta_{kj} V_{ij|rl} + \delta_{lm} \delta_{ij} V_{ij|kl} - \delta_{ij} \delta_{lu} V_{ij|lm} \right\} + \left\{ (i, j, k) \rightarrow (k, j, i) \right\} - \left\{ i \leftrightarrow j \right\} + \delta_{kl} \delta_{ij} \delta_{mn} \left( \epsilon_r - \epsilon_{ij} \right) \]  

(7)

It is precisely the choice of having a first-order non-diagonal 3h1p/3h1p block that makes the number of non-zero \( \Gamma \) matrix elements large and requires that the matrix is not stored but recomputed directly from the integrals and the orbital energies whenever it is needed, as is routinely done for the hamiltonian matrix in large-scale configuration interaction based methods [31]. How this is best done will be described in the following section.

3. Implementation and computational aspects

3.1. Notation and preliminaries

The first thing to note in the ADC(2) expressions for the \( \Gamma \) matrix equations (5)-(7) is that only two-electron integrals having at most two-particle indices are required. We shall here assume that these integrals are readily available in memory, but the procedure that will be described may be easily adapted to allow for non-memory-resident integrals or integral dispatching from an upstream direct integral evaluation procedure. We shall now allow for real spatial orbitals \( \phi_i \) and coulumb integrals, which we write in density notation

\[ (ab|cd) = \int \phi_a(r)\phi_b(r)\phi_c(r')\phi_d(r') \frac{dr dr'}{|r - r'|} \]  

(8)

\[ (ab|cd)^\pm = (ab|cd) \pm (ad|bc) \]  

(9)

whereby the \( \Gamma \) matrix becomes real symmetric. It is useful to arrange the integrals having just one particle index \((rk|ij)\) as vectors \( V_{ij,k} \) identified by the index triplet made up of the hole pair index \( ij \) \((i \geq j)\) and the further hole index \( k \). Similarly, the integrals having two-particle and two-hole indices (we shall generically call them 2h integrals) are arranged in matrices \( A_{ij} \) and \( B_{ij} \), labeled by the hole pair, of elements \( A_{ijrs} = \langle r|ij|s\rangle \) and \( B_{ijrs} = \langle rs|ij\rangle \). Sorting and addressing the integrals in this way, as well as the integrals with four hole indices (4h integrals), possibly exploiting molecular symmetry to block the integral matrices and skip identically vanishing elements, is relatively straightforward and will not be discussed here. Similarly, the possible use of symmetry in configuration generation and \( \Gamma \) construction will be addressed only cursorily in the following discussion.

3.2. Spin adaption and \( \Gamma \) matrix structure

The ADC matrix element formulas reported in Eqs. (5)-(7) refer to a non-degenerate neutral ground state system assumed to have a closed-shell Hartree-Fock representation. The dicaticion configurations, over which the \( \Gamma \) matrix is built, are therefore spin-adapted to give singlet and triplet functions. (The satellite space would also give rise to a quintet case, which we neglect.) The eigenvalue problem for the \( \Gamma \) matrix in the spin-adapted configuration basis factorizes of course into a singlet and a triplet problem, treated separately. The spin-adapted functions used, together with the resulting spin-adapted \( \Gamma \) matrix elements are detailed in Appendix A.

As is generally done for open-shell configurations, in 3h1p space we distinguish between two classes of functions, those with two unpaired electrons (type I) and those with four (type II). The latter give rise to each configuration (orbital label) to two singlet and three triplet functions. In generating the final spin-adapted configuration list, we establish that the 2h configurations precede the 3h1p ones (with the closed shells coming first in the singlet case) and the 2h configuration indices \( ij \) are always restricted to \( i \geq j \) \((i > j \text{ for triplet})\). Among the 3h1p configurations, all the configurations of type I precede those of type II. In the latter set, the spin-adapted functions are segregated by spin label, all those with spin label 1 preceding those with spin label 2 and so on. Of course, type II functions of different spin label are scanned and handled by a unique orbital index loop, with a fixed offset.

It is important that, in the orbital pattern loop within each 3h1p type, the particle index runs fastest, whereby the functions are naturally further subdivided in sets (vectors) labeled by 3h (and spin) indices. For type II, a 3h index \( kmn \) is generated with \( k > l > m \) and, as mentioned above, it clearly identifies multiple vectors, one for each available spin case. If molecular symmetry is present and exploited, the \( \Gamma \) matrix factorizes further by symmetry and each irreducible representation case is treated separately. This means in particular that, in each given
symmetry, the particle space associated to a 3h configuration index will be specifically restricted by symmetry, so that different 3h configuration vectors will in general have different lengths.

According to the above, the spin-adapted matrix elements involving 3h1p configurations are best analysed in terms of 3h configuration sets, by treating the interaction of a 3h set with other configurations as a whole. This reveals a particularly convenient structure. A matrix portion involving a particular 3h index triplet $k ln$ will be written as $\Gamma_{k ln, \mu}$, where $\mu$ stands for any other configuration (or configuration set) index. Thus, for example, the matrix elements of the 3h1p/2h block are naturally subdivided in vectors $\Gamma_{k l m, \mu}$ (or collection of vectors, for multiple 3h spin cases). It turns out that these vectors just involve combinations of corresponding vectors of integrals $V_{\mu, k}$ having one particle index (see Appendix A).

The 3h1p/3h1p portion of the matrix is naturally subdivided in 3h/3h coupling blocks, most of which vanish because no index equality (Kronecker’s delta) is satisfied. The contributions to a given non-zero 3h/3h block $\Gamma_{\mu, \nu}$ for any particular spin combination, have the general structure

$$c_A A_\mu + c_B B_\nu + x_c I$$

where $A_\mu$ and $B_\nu$ are the matrices of 2h integrals introduced earlier, $c_A$ and $c_B$ are numerical coefficients, while $x_c$ is a constant diagonal scalar contribution resulting as a combination of 4h integrals. Note that if symmetry is exploited, the particle spaces associated to the 3h indices $\mu$ and $\nu$ may be different (and $\Gamma_{\mu, \nu}$ is in general a rectangular block), in which case $x_c = 0$ identically. Fig. 1 illustrates the structure of the ADC matrix by way of an example obtained in a real calculation on a small symmetric molecule (formaldehyde). It evidences clearly the sparse block structure of the 3h1p/3h1p block, where rectangular blocks and diagonal blocks (when only the $x_c$ contributions are non-zero) can be seen.

### 3.3. Implementation considerations

In large-scale diagonalization problems, one usually adopts subspace iteration methods [34,35] to find selected eigenvalues and eigenvectors. In these methods, the problem matrix is only required at each iteration to multiply one or more new basis vectors and this matrix-multiply step invariably represents a computational bottleneck. In direct algorithms, it is therefore essential to devise a matrix evaluation procedure which makes this matrix-multiply step as efficient as possible, preferably fractioning it in a series of smaller matrix operations [31]. In the direct calculation of dense spectra such as double ionization ones, this task is particularly crucial and delicate, not only because of the large matrix, but because we are interested (in sharp contrast with typical wavefunction calculations) to a large number of eigensolutions, as this is necessary to describe wide regions of a spectrum. Clearly, this often implies a very large number of iterations (large subspaces). Now, the structure of the $\Gamma$ matrix presented above suggests immediately a natural and efficient way of carrying out the matrix multiplication step. The following discussion will illustrate some specific aspects in greater detail.

A significant simplification of our computational procedure can be preliminarily achieved by exploiting in combination a physical feature of the desired double ionization spectrum and some simple general features of subspace iteration. In principle, we are mainly interested in elements of the spectrum which have a non-zero component in 2h space, since these correspond, from a perturbation theory viewpoint and in experimental observations, to the most important states. This consideration clearly steers the choice of the initial subspace kicking off the iterations towards an easy and appropriate solution, namely the set of $N_0$ 2h cartesian vectors (let us denote the 2h space size by $N_0$). With this choice, it follows that: (i) the result of the multiplication of the initial basis vectors by $\Gamma$ is precisely the set of vectors made up of the 2h/2h and 3h1p/2h blocks of $\Gamma$ (i.e., its $N_0$ leftmost columns); and (ii) since subsequent basis vectors are usually required (either explicitly or implicitly) to be orthogonal to previous ones, one may simply drop their 2h components and, when applying $\Gamma$ to them, perform only the multiplication of the 3h1p/3h1p block of the matrix with the 3h1p vector components. This design has several advantages, the two principal ones being that: (a) the 2h/2h and 3h1p/2h blocks of $\Gamma$ are only required once at the beginning of the iterative diagonalization procedure. Furthermore, they are actually not used in a multiplication but are simply evaluated as the first explicit set

![Fig. 1. Outline of a 645 x 645 ADC(2) matrix (lower triangle), obtained for the $^1A_1$ doubly ionized states of the H$_2$CO molecule with 42 basis functions. In black are the non-zero elements of the matrix. The size of the (valence) main space is 10.](image-url)
of basis vectors (pre-orthonormalization). Subsequent iterations only require the 3h1p/3h1p block of $\Gamma$; and (b) the basis vectors (and thus the subspace eigenvectors) always span exactly the complete 2h space, with no missing portions and – more important – without redundancies introduced by possible loss of orthogonality among the basis vectors during the iterations [34–36]. This is especially useful in Lanczos-type procedures [35,36] to avoid duplicate and spurious vectors and for generally obtaining a set of solutions with a clean and accurate main projection.

Thus, the 2h/2h and 3h1p/2h blocks of $\Gamma$ are only required once in our procedure. The 2h/2h block is very small, even if non-sparse, compared with the rest, and is computed and stored using Eqs. (A.1)–(A.5). In the 3h1p/2h interaction, as we observed above, only integral vectors $V_{ijk}$ with one particle index contribute. Therefore, the construction of the 3h1p/2h block consists of filling vector segments corresponding to 3h configuration sets which satisfy the required index equalities (Kronecker’s deltas), by linear combinations of appropriate integral vectors. These superpositions can easily be carried out as successive vector updates (triad or, in the terminology of the level-1 basic linear algebra subprograms (BLAS) [37], daxpy operations).

Turning now to the evaluation of the 3h1p/3h1p block of the $\Gamma$ matrix, this constitutes essentially, as outlined above, the crucial matrix-multiply step at the core of subspace iteration. The natural partitioning of the non-zero matrix elements in small blocks labeled by their 3h/3h coupling indices (and spin combination) lends itself ideally to this task. Each such block can easily be computed as required by combining available integral matrices $A_{ij}$ and $B_{ij}$ (through daxpy operations) and it clearly selects a respective segment of each target vector by which it must be multiplied and added to a corresponding result vector. By running over just the lower triangular blocks, furthermore, each of them (except the diagonal ones) is used in two matrix multiplications as it also identifies an identical transposed block in the upper triangle of $\Gamma$. The procedure is outlined in Fig. 2. The $x_i$ constant terms in each block, arising from 4h integrals (see Eq. (10)) are processed separately, as scalar multipliers of the target vector segment. Fig. 2 also schematically evidences a specific feature of our problem, already hinted at above, namely that we typically have not one but a number (possibly large) of new basis vectors to which the matrix must be applied at each iteration. Ideally this number is equal to the 2h space size, but it may be smaller for a single matrix evaluation step if not enough memory is available or to optimize efficiency. The need of having multiple vectors has been mentioned earlier and is also dictated by the fact that such “block-diagonalization” procedures (see, e.g., Ref. [38]) have better convergence properties in general when many roots are sought and, in particular in our case, to converge the lowest moments of the 2h distribution of the spectrum, which is useful to give a good description of dense spectra even when individual eigenvectors are not converged [39]. Using multiple vectors also has the added advantage of lending efficiency to the matrix multiply, in that individual level-2 BLAS matrix-vector operations ($dgemv$) are naturally replaced by their level-3 matrix–matrix analogue ($dgemm$) with improved cache reuse [37]. Note that the elementary matrix blocks have linear size at most equal to the number of particles and so, not only can they be handled by individual $dgemm$ operations but, for not too big cases, fit comfortably in typical processor caches.

However, as Fig. 2 also suggests, when a collection of basis vectors is used, the sections of it required for multiplication by a $\Gamma$ block do not constitute contiguous memory arrays in FORTRAN memory ordering: the elements involved of each vector are separated from those of the next by a number of elements equal to the vector length. This spoils the efficiency gained by having actual matrix-matrix multiply operations. It also makes particularly inefficient processing the constant diagonal terms arising from the 4h integrals, which must multiply the whole selected target sections of the basis vectors. These drawbacks may be circumvented by using transpose (C-like memory ordered) basis vectors, thought to multiply on the left the $\Gamma$ matrix. In this way the target vector sections do constitute contiguous arrays in memory and it is immediately seen that a $x_i$ term may be processed through a daxpy operation: the target vector is a unit-stride array of length equal to the number of basis vectors at hand times the row-size of the vector section involved. The maximum efficiency of the BLAS operations is thus restored. Since it is usually inconvenient to handle transpose vectors during the remaining subspace iteration steps (orthogonalization, etc.), this procedure has the price of transposing the basis matrix used (by swapping target and result) upon entry into, and exit from, the matrix-multiply step.

We would finally like to observe an inherent, efficient, and easily exploited parallelism in the matrix-multiply step. As we know from the previous analysis and Tables A.4–A.7 show, most 3h/3h couplings trigger more than one elementary $dgemm$ operation, in fact up to 10 for each delta case in the triplet interaction between type II
configurations (this is considering block transposition, and it is less than the theoretical limit of 18 because, as Table A.7 shows, a number of spin coupling cases always vanishes). Simple inspection reveals that these operations can be divided in groups targeting different non-overlapping sections of the result vector matrices. These groups are therefore totally independent operations that lend themselves ideally to being executed in parallel on multi-processor machines.

3.4. Computing eigenvalues and eigenvectors

Double ionization spectra computable by the present method are characterized, even for systems of moderate size and except near threshold, by being very dense, with an essentially complete breakdown of the main 2h character and numbers of relevant eigenstates running easily into the tens per eV (see, e.g., Ref. [25]). Extracting these eigensolutions is a time-consuming task, requiring many iterations and large subspaces. The routine subspace iteration approach to extract these spectra in our implementation is the block-Lanczos method [35], essentially as described in Ref. [38]. An important advantage of this method is that it is computationally very undemanding because, at the price of retaining only the eigenvector projection on the initial subspace (i.e., as specified earlier, the main 2h subspace) and thus discarding all details on the satellite structure of the double ionization transitions, only three sets of \( N_0 \) basis vectors must be retained at each iteration and these are easily kept in memory or spilled to disk as the situation requires. With our choice of initial subspace, the main components of the Ritz vectors [34] are directly the top \( N_0 \) components of the subspace eigenvectors (while the bottom \( N_0 \) components may be used to compute the residuals [34]). This feature, together with the banded structure of the \( \Gamma \) matrix in Lanczos space, permits the execution, with minimal storage requirements, of the usually very large number of iterations (running easily into the several hundreds) required to approximate a satisfactory portion of the spectrum. In this connection, we already mentioned how our choice of the 2h initial space can straightforwardly be exploited to guarantee – very conveniently – that the computed Ritz subspace is automatically rid of duplicate or spurious vectors with any significant 2h component. Equally extremely convenient is the property [39] of the Lanczos iterations that the moments of the spectral distribution of the eigenvector projections onto the initial space converge faster than the individual eigenvectors, in fact much faster if, as is actually almost invariably the case, the discrete distribution lines are somewhat broadened by non-stationarity or unresolved roto-vibrational structure. With our choice, this means that qualitatively satisfactory 2h spectral profiles may be obtained even in (dense) unconverged regions of the spectra.

There are cases, however, when one would like to converge accurately the eigenvectors in some specific inner energy region of the spectrum of particular experimental interest. The most effective approach in these cases is usually filter diagonalization [40–43], but we would like to briefly recall here a variant of the Davidson method [44] (or block-Davidson [38] in our case) proposed by Morgan [45], which we found particularly useful in our context. In contrast to filter diagonalization, this method is more specifically suited to converge on a relatively small selected number of eigensolutions of the inner spectrum, such as, in our case, those with large 2h weight. Its main idea is a modification of the basic Rayleigh–Ritz (RR) step [34] of the Davidson method, where one normally diagonalizes, at each iteration, the problem matrix projection onto the current subspace, i.e. the matrix \( \mathbf{A} = \mathbf{Q}^T \mathbf{Q} \), where \( \mathbf{Q} \) is the matrix collecting (as columns) the current set of (orthonormal) basis vectors. In this way approximate eigenvectors (Ritz vectors) of \( \mathbf{A} \) are obtained, of the form \( \mathbf{c} = \mathbf{Q} \mathbf{v} \), where \( \mathbf{v} \) is a suitable eigenvector of \( \mathbf{A} \). The mean value (Ritz value or approximate eigenvalue) associated to \( \mathbf{c} \) is of course the eigenvalue of \( \mathbf{A} \) associated to \( \mathbf{v} \).

When one is interested in specific inner eigenvectors with eigenvalue close to \( \sigma \), say, then better Ritz vectors would be obtained from the eigenvectors of \( \mathbf{Q}^T (\mathbf{\Gamma} - \sigma \mathbf{1})^{-1} \mathbf{Q} \), as this would bring the desired roots at the edge of the spectrum and effectively space them apart, markedly improving convergence. But this is extremely impractical because, for large matrices, matrix inversion (or equivalently the solution of the linear system yielding \( (\mathbf{\Gamma} - \sigma \mathbf{1})^{-1} \mathbf{Q} \)) is a lengthy iterative process in its own right. To circumvent this hurdle, Morgan proposed a formal change of basis, at the outset of each RR step, from \( \mathbf{Q} \) to \( \mathbf{P} = (\mathbf{\Gamma} - \sigma \mathbf{1}) \mathbf{Q} \). Over this, the inverted matrix projection is simply

\[
\tilde{\mathbf{A}} = \mathbf{P}^T (\mathbf{\Gamma} - \sigma \mathbf{1})^{-1} \mathbf{P} = \mathbf{A} - \sigma \mathbf{1}
\]

After solving the (now non-orthogonal) subspace problem

\[
\tilde{\mathbf{A}} \mathbf{v} = \delta \mathbf{s} \mathbf{v}
\]

with \( \mathbf{S} = \mathbf{P}^T \mathbf{P} \), a Ritz vector for the \( \Gamma \) problem would be \( \tilde{\mathbf{c}} = \mathbf{P} \mathbf{v} \) and its Ritz value \( \tilde{\epsilon} = 1/\delta + \sigma \) (note that the latter is not the expectation value of \( \mathbf{c} \)). But now, one can easily improve on this solution by applying one step of inverse iteration [34], which – very conveniently – leads us back to \( \mathbf{c} = \mathbf{Q} \mathbf{v} / \| \mathbf{v} \| \) (normalized), with expectation value \( \epsilon = 1/ \| \mathbf{v} \|^2 + \sigma \). Except of the above, the (block)-Davidson procedure proceeds exactly as usual, by computing one new basis vector specifically aimed at each interesting approximate eigenvector produced in the subspace and reiterating until appropriate convergence criteria are met. Having better approximations in the modified RR procedure often markedly improves convergence and this combines particularly well with the implementation of restrictive root search criteria such as, in our case, a threshold on 2h weight and/or an energy window.

As the above description clearly shows, the modifications to the usual Davidson method are marginal. Because
the two separate sets of vectors $Q$ and $Z = f^T Q$ are kept and updated as part of the standard procedure, along with $A = Q^T Z$, the additional overlap matrix $S$ is easily obtained as $S = Z^T Z - \sigma(A + A^T)$, and $Z^T Z$ may be kept and updated as well, with little additional storage. One particular benefit of the procedure as we have implemented it is that the shift $\sigma$ may be freely changed from iteration to iteration, thus enabling dynamic adjustments to improve convergence and the automatic scanning of a given energy window, by moving $\sigma$ close to successive Ritz values as the desired roots converge.

The major drawbacks of the above method arise from the fact that the subspace RR problem is non-orthogonal. This leads to non-orthogonal Ritz vectors and a clear tendency, in cases of difficult convergence and many iterations, to generate multiple subspace solutions approximating the same true root. This, of course, adds to the most significant shortcoming of the Davidson procedure as such, namely the necessity of storing the whole accumulated set of $Q$ and $Z$ vectors, which severely limits the maximum subspace size reachable during the iterations. It is always possible to transform the basis set to the current set of (reorthonormalized) Ritz vectors with consequent space truncation, but this is expensive and often unproductive. Also, RR non-orthogonality impairs the effective monitoring of the actual $2h$ space spanned by the Ritz vectors, which is normally a very important element to evaluate the progress of the procedure.

RR non-orthogonality may be avoided in a second variant of Morgan’s method [45], which essentially leads to diagonalizing the variance matrix of the basis $Q$ with respect to $\sigma$. The idea here is that of using not the projection of the inverse operator $(\Gamma - \sigma I)^{-1}$ but rather of $(\Gamma - \sigma I)^{-2}$. Switching again to the transformed basis

<table>
<thead>
<tr>
<th>Table A.2</th>
<th>$\Gamma$ matrix elements for the 3h1p/2h block over spin adapted dicatonic configurations. For the notation, see text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singlet</td>
<td>$\Gamma_{ijkl} = \sqrt{2} {\delta_{ik}(2V_{il,j} - V_{ij,l}) - \delta_{il}(V_{jl,k} + V_{il,k})$</td>
</tr>
<tr>
<td></td>
<td>$\Gamma_{ijkl} = \delta_{ik}(2V_{il,j} - V_{ij,l}) - \delta_{il}(V_{jl,k} + V_{il,k}) + \delta_{jk}(2V_{ij,l} - V_{il,j}) - \delta_{jl}(V_{lk,i} + V_{lj,i})$</td>
</tr>
<tr>
<td></td>
<td>$\Gamma_{ijkl} = \delta_{ik}(2V_{il,j} - V_{ij,l}) - \delta_{il}(V_{jl,k} + V_{il,k}) + \delta_{jk}(2V_{ij,l} - V_{il,j}) - \delta_{jl}(V_{lk,i} + V_{lj,i}) + (i \leftrightarrow j)$</td>
</tr>
<tr>
<td>Triplet</td>
<td>$\Gamma_{ijkl} = \delta_{ik}V_{jl,i} - \delta_{il}V_{jk,i} - \delta_{jk}V_{il,j} + \delta_{jl}V_{ik,i}$</td>
</tr>
<tr>
<td></td>
<td>$\Gamma_{ijkl} = -\delta_{ik}V_{jm,i} + \delta_{il}V_{jm,i} + \delta_{jm}V_{jk,i} - \delta_{jm}V_{jk,i}$</td>
</tr>
</tbody>
</table>

$P = (\Gamma - \sigma I)Q$ straightforwardly leads to the orthogonal RR problem $Hv = \lambda v$, where $H = Q^T (\Gamma - \sigma I)^2 Q$. Again, the application of an inverse iteration step to the Ritz vector leads to $c = Qv$, with $c = v^T Av$. Here too, the sole mod-

<table>
<thead>
<tr>
<th>Table A.1</th>
<th>Singlet and triplet spin eigenfunctions used in the ADC(2) implementation. The subscript indices here label the spatial orbitals and a bar over a creation or annihilation operator distinguishes spin $\beta$ from spin $\alpha$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singlet</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>Triplet</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$\frac{1}{\sqrt{2}}(\frac{1}{2}</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
</tbody>
</table>

**Auxiliary functions**

$|f_{\alpha i}\rangle = \frac{1}{\sqrt{2}}(a_\alpha a_\beta a_\gamma \pm a_\beta a_\gamma a_\alpha)|\Phi_\alpha \rangle$

$|f_{\beta i}\rangle = \frac{1}{\sqrt{2}}(a_\alpha a_\beta a_\gamma \pm a_\beta a_\gamma a_\alpha)|\Phi_\alpha \rangle$

$|f_{\gamma i}\rangle = \frac{1}{\sqrt{2}}(a_\alpha a_\beta a_\gamma \pm a_\beta a_\gamma a_\alpha)|\Phi_\alpha \rangle$
fication to the standard Davidson procedure essentially consist in the additional computation and storage of $Z^JZ$. The advantage of an orthogonal RR problem is fully regained but, on the flip side, the method clearly collapses onto one both sides of the spectrum above and below $\sigma$, with a substantial increase in density of states, and we often found that this leads to exceedingly slow convergence.

4. Conclusions

The second-order ADC method for computing double ionization spectra of large molecular systems and aggregates can be implemented very efficiently. The implicit large matrix construction step required in subspace iteration techniques to extract (many) eigenvalues and eigenvectors is reduced to a sequence of small matrix-multiply operations which exploit modern computer architectures to their full potential, and many of which can be executed in parallel. A whole range of partial diagonalization methods can thus be brought to bear on the problem, the routinely most effective of which proves to be the block-Lanczos procedure. Morgan’s variants of the Davidson method to converge some inner roots of the spectrum have also been implemented and analysed in detail, proving helpful for some specific purposes.

Acknowledgements

The author is grateful to L.S. Cederbaum, J. Schirmer and H.-D. Meyer for many enlightening discussions, and to G. Handke for having brought Morgan’s method to his attention.
Appendix A. Spin-adapted formulation of the ADC(2) equations

In this Appendix, we report the complete spin-adapted working formulas implemented for ADC(2), using the notation introduced in Section 3.1. The spin eigenfunctions used in our implementation are displayed in Table A.1. Transforming the $\Gamma$ matrix expressions Equations (5)–(7) over the spin-adapted functions yields the following expression for the lower triangle of the $2h/2h$ block of the $\Gamma$ matrix (when $i > j$ and $k > l$):

$$\Gamma_{ijkl} = (ik|jl)^\pm - U_{ijkl} + \delta_{jk} W_{jl} \pm \delta_{jl} W_{il} - \delta_{ik} \delta_{jl} \epsilon_{ij}$$  \hspace{1cm} \text{(A.1)}$$

where the upper sign refers to the singlet case and the lower sign to the triplet case. $U_{ijkl}$ and $W_{ik}$ collect the second order terms as follows:

$$U_{ijkl} = \sum_{r \geq s} (1 - \delta_{rs}/2)(ri|sj)(rk|sl)^\pm \frac{\epsilon_{rs} - \epsilon_{ij}/2}{(\epsilon_{rs} - \epsilon_{il})(\epsilon_{rs} - \epsilon_{kl})}$$ \hspace{1cm} \text{(A.2)}$$

where $W_{ik}$ is given by:

$$W_{ik} = \sum_{m} (1 - \delta_{mk}/2)(rm|sm)(rk|sm) + (rm|st)(rm|sk)$$

$$+ (ri|sm)(rk|sm) \frac{\epsilon_{sm} - \epsilon_{im} - \epsilon_{ik}/2}{(\epsilon_{rs} - \epsilon_{im})(\epsilon_{rs} - \epsilon_{km})}$$ \hspace{1cm} \text{(A.3)}$$

For the cases in which closed shells are involved, Eq. (A.1) reduces to

$$\Gamma_{ijkl} = \sqrt{2}[(ik|jl)^\pm - U_{ijkl} + (\delta_{jk} + \delta_{kl}) W_{il}]$$ \hspace{1cm} \text{(A.4)}$$

$$\Gamma_{ii,jj} = (ij|ij)^\pm - U_{ii,jj}/2 + 2\delta_{ij}(W_{ii} - \epsilon_i)$$ \hspace{1cm} \text{(A.5)}$$

The $3h1p/2h$ matrix elements over spin-adapted configurations are shown in Table A.2. Here a matrix block such as $\Gamma_{klm,ij}$ is a set of matrix elements relative to all the $3h1p$ configurations identified by the $3h$ index $kln$ ($k > l > m$). Multiple spin cases for the same $3h$ index are surrounded by brackets.

Tables A.3–A.7 collect all the contributions to the $3h1p/3h1p$ matrix (omitting the zeroth order double ionization energies) divided by configuration types and blocked after the $3h$ indices. Again, the $kln$ $3h$ indices of type II are restricted to $k > l > m$ and only the terms contributing to the lower triangle of the matrix are shown.

<table>
<thead>
<tr>
<th>$\delta_{ij}\delta_{km}$</th>
<th>$\frac{1}{2}(A_{ln} - B_{ln})$</th>
<th>$-\frac{\sqrt{2}}{2}A_{ln}$</th>
<th>$-\frac{\sqrt{2}}{2}A_{ln}$</th>
<th>$-\frac{1}{2}B_{ln}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_{ij}\delta_{lm}$</td>
<td>$2A_{jm} - B_{jm}$</td>
<td>0</td>
<td>$-B_{jm}$</td>
<td></td>
</tr>
<tr>
<td>$\delta_{ij}\delta_{mn}$</td>
<td>$\frac{1}{2}(B_{rn} - 2A_{rn})$</td>
<td>$\frac{\sqrt{2}}{2}B_{rn}$</td>
<td>$-\frac{\sqrt{2}}{2}B_{rn}$</td>
<td></td>
</tr>
<tr>
<td>$\delta_{ij}\delta_{nl}$</td>
<td>$\frac{1}{2}(B_{rn} - 2A_{rn})$</td>
<td>$-\frac{\sqrt{2}}{2}B_{rn}$</td>
<td>$\frac{\sqrt{2}}{2}B_{rn}$</td>
<td>$B_{rn}$</td>
</tr>
</tbody>
</table>

Diagonal $x_i$ terms

| $\delta_{ii}$ | $(jm|kn) - \frac{1}{2}(jn|km)$ | $\frac{\sqrt{2}}{2}(jn|km)$ | $(jm|kn) + \frac{1}{2}(jn|km)$ |
|----------------|--------------------------------|-------------------------|-------------------------|
| $\delta_{im}$ | $(ij|km)^+$ | 0 | $(ij|km)-$ |
| $\delta_{il}$ | $(im|jn) - \frac{1}{2}(im|jm)$ | $\frac{\sqrt{2}}{2}(im|jm)$ | $(im|jn) + \frac{1}{2}(im|jm)$ |
| $\delta_{in}$ | $(il|jm) - \frac{1}{2}(im|jl)$ | $-\frac{\sqrt{2}}{2}(im|jl)$ | $(il|jm) + \frac{1}{2}(im|jl)$ |
Table A.7
Contributions to a $\Gamma_{ijkl}$, block of the triplet ADC(2) matrix. The $3 \times 3$ arrays for each delta case map the coupling of spin cases among types II configurations. The diagonal orbital energy terms are omitted. For the notation see text

$$
\begin{align*}
\delta_{ij}\delta_{kn} & \left( \begin{array}{ccc}
-B_{mn} & 0 & 0 \\
0 & A_{kn} - B_{kn} & A_{kn} \\
0 & A_{kn} & A_{kn} - B_{kn}
\end{array} \right) \\
\delta_{ij}\delta_{lm} & \left( \begin{array}{ccc}
(A_{lm} - B_{lm}) & 0 & A_{lm} \\
0 & -B_{lm} & 0 \\
0 & 0 & A_{lm} - B_{lm}
\end{array} \right) \\
\delta_{ij}\delta_{kn} & \left( \begin{array}{ccc}
B_{mn} - A_{mn} & 0 & -A_{mn} \\
0 & A_{mn} & -A_{mn} \\
0 & 0 & B_{mn} - A_{mn}
\end{array} \right)
\end{align*}
$$

### Diagonal $x$, terms

$$
\begin{align*}
\delta_{ij} & \left( \begin{array}{ccc}
(j|m|n) & -(j|m|n) & 0 \\
-(j|m|n) & (j|m|n) & 0 \\
0 & 0 & (j|m|n)
\end{array} \right) \\
\delta_{jm} & \left( \begin{array}{ccc}
(i|l|k)n & 0 & -(i|l|k)n \\
0 & -(i|l|k)n & 0 \\
-(i|l|k)n & 0 & (i|l|k)n
\end{array} \right) \\
\delta_{kl} & \left( \begin{array}{ccc}
0 & 0 & -(i|l|j)m \\
-(i|l|j)m & 0 & 0 \\
-(i|l|j)m & 0 & (i|l|j)m
\end{array} \right) \\
\delta_{in} & \left( \begin{array}{ccc}
0 & 0 & -(i|m|j)n \\
-(i|m|j)n & 0 & 0 \\
0 & 0 & (i|m|j)n
\end{array} \right)
\end{align*}
$$

### References