Journal of Mathematical Chemistry 21 (1997) 395–412

An iterative method to solve the algebraic eigenvalue problem

Emili Besalú and Ramon Carbó-Dorca

Institute of Computational Chemistry, University of Girona, 17071 Girona, Catalunya, Spain

Received 28 April 1997

An iterative method based on perturbation theory in matrix form is described as a procedure to obtain the eigenvalues and eigenvectors of square matrices. Practical vector notation and elementary schematic algorithm codes are given. The particular programming characteristics of the present computational scheme are based upon eigenvector corrections, obtained through a simple Rayleigh–Schrödinger perturbation theory algorithm. The proposed methodological processes can be used to evaluate the eigensystem of large matrices.

1. Introduction

Recently, the authors have published various papers dealing with several aspects of perturbation theory (PT). In ref. [1], the foundation of the Rayleigh–Schrödinger PT (RSPT) framework in matrix form was described, possessing a sufficiently general structure as to deal with several perturbation matrices or operators at the same time. Other recent and old papers of our Laboratory have dealt with general aspects of the formal problem associated with RSPT [2,5–12].

The present work is also related to the RSPT formalism and intends to describe an iterative algorithm, which may be useful to achieve the diagonalization of a *general* square matrix. The theoretical formalism will be developed first, and the practical algorithms will be given afterwards.

From the previously described RSPT matrix formalism [1], it is straightforward to build up a connected vector notation, useful for the perturbative resolution of a given eigenvalue–eigenvector pair. Within the quantum mechanical framework, this is equivalent as to find a particular system's state.

2. General theory

The goal of the present paper is to discuss a new procedure useful to numerically solve the algebraic eigenvalue problem. The method which will be developed here can be applied to any kind of general square matrices, including hermitian or symmetrical ones.

Diagonalization of a general square matrix $\boldsymbol{H}$, possessing real or complex eigenparameters may be considered as the way of solving the secular equation

$$\boldsymbol{HZ} = \boldsymbol{ZE}, \tag{1}$$

where the $\boldsymbol{Z}$ and $\boldsymbol{E}$ matrices contain in their columns and diagonal elements, respectively, the eigenvectors and eigenvalues of matrix $\boldsymbol{H}$.

Diagonalization of matrix $\boldsymbol{H}$ can always be formulated in an intermediate form such as to find the eigenparameters of the matrix defined as

$$\boldsymbol{H} = \boldsymbol{Z}_0\boldsymbol{R}_0(\boldsymbol{Z}_0)^{-1} \tag{2}$$

and the matrix $\boldsymbol{R}_0$ can be expressed by the sum

$$\boldsymbol{R}_0 = \boldsymbol{E}_0 + \boldsymbol{P}_0, \tag{3}$$

where $\boldsymbol{E}_0$ is a diagonal matrix and $\boldsymbol{P}_0$ acts as a perturbative term. A related point of view may also be found in the Collar and Jahn's diagonalization method [15].

Some examples, where the previous framework applies, can be given as follows:

(1) When *starting up* a diagonalization problem. The trivial approach considers the following identities:

$$\boldsymbol{R}_0 = \boldsymbol{H} \quad \wedge \quad \boldsymbol{Z}_0 = (\boldsymbol{Z}_0)^{-1} = \boldsymbol{I}. \tag{4}$$

According to equation (3), the matrix $\boldsymbol{R}_0$ can be organized using the pattern

$$\boldsymbol{E}_0 = \mathrm{diag}\{\boldsymbol{H}\} \quad \wedge \quad \boldsymbol{P}_0 = \mathrm{outdiag}\{\boldsymbol{H}\}, \tag{5}$$

where the following definitions are used:

$$\boldsymbol{A} = \mathrm{diag}\{\boldsymbol{H}\} = \left\{a_{ii} = h_{ii}\ \forall i\ \wedge\ a_{ij} = 0\ \forall i \neq j\right\} \tag{6}$$

and

$$\boldsymbol{B} = \mathrm{outdiag}\{\boldsymbol{H}\} = \left\{b_{ii} = 0\ \forall i\ \wedge\ b_{ij} = h_{ij}\ \forall i \neq j\right\}. \tag{7}$$

(2) When looking at an *intermediate* form of several diagonalization procedures. For instance, the well-known iterative Jacobi [16] method, at every cycle of the process the transformation (2), uses the matrix $\boldsymbol{Z}_0$ chosen to be orthogonal. Such a method drives the matrix $\boldsymbol{P}_0$ to become the null one at the iteration limit. At the end of the procedure the eigenvectors of matrix $\boldsymbol{H}$ are stored as columns in $\boldsymbol{Z}_0$ and the eigenvalues in the diagonal of $\boldsymbol{R}_0 = \boldsymbol{E}_0$.

(3) When *analyzing* the RSPT scheme. This is the case which will be considered here.

3. Theoretical basis of the proposed algorithm

RSPT is formulated in such a way as to achieve the diagonalization of the so called perturbed matrix, $\boldsymbol{H}$, which is expressed in terms of a unperturbed matrix, $\boldsymbol{H}_0$, plus a perturbation matrix, $\boldsymbol{V}$:

$$\boldsymbol{H} = \boldsymbol{H}_0 + \boldsymbol{V}. \tag{8}$$

The diagonalization of $\boldsymbol{H}$ supposes the previous knowledge of the unperturbed system solution, which can be attached to the secular equation eigenparameters:

$$\boldsymbol{H}_0\boldsymbol{Z}_0 = \boldsymbol{Z}_0\boldsymbol{E}_0, \tag{9}$$

and this entirely gives the information defining a typical PT problem. Thus, in this study, the set of involved input data, constituting a given PT problem, will be expressed by means of the following ordered row hypervector:

$$(\boldsymbol{H}, \boldsymbol{H}_0, \boldsymbol{V}, \boldsymbol{Z}_0, \boldsymbol{E}_0), \tag{10}$$

which can be used as a shorthand representation of equations (8) and (9). It will be called from now on a *PT problem vector*.

Within the PT scheme, the unperturbed eigenvectors, collected in the matrix $\boldsymbol{Z}_0$, are used as the working basis set, the reference space, which expands a vector space of the same dimension as the one attached to all the involved matrices. The unperturbed vectors are linearly independent and, thus, can be considered orthonormalized without loss of generality. It is easy to realize that they can act as a canonical basis set, if the proper definitions are taken into account. From equation (9), expressing $\boldsymbol{H}_0$ as

$$\boldsymbol{H}_0 = \boldsymbol{Z}_0\boldsymbol{E}_0(\boldsymbol{Z}_0)^{-1} \tag{11}$$

and defining the following similarity transformation over the perturbation matrix $\boldsymbol{V}$:

$$\boldsymbol{P}_0 = (\boldsymbol{Z}_0)^{-1}\boldsymbol{V}\boldsymbol{Z}_0, \tag{12}$$

it is straightforward to express the perturbed matrix to be diagonalized in the same notation as the one used in equations (2) and (3). That is, diagonalization of matrix $\boldsymbol{H}$ has been reduced to solve the diagonalization of matrix $\boldsymbol{R}_0$, appearing in equation (2). In other words, the PT problem vector has been brought to solving the following transformed one, defined by the new PT problem vector

$$(\boldsymbol{R}_0, \boldsymbol{E}_0, \boldsymbol{P}_0, \boldsymbol{I}, \boldsymbol{E}_0). \tag{13}$$

The presence of the unit matrix in equation (13) connects the original unperturbed basis set to the canonical one.

At this stage, the diagonal values of the matrix $\boldsymbol{R}_0$ can be freely manipulated, for instance, expressing the whole matrix as the sum

$$\boldsymbol{R}_0 = \boldsymbol{R}_d + \boldsymbol{R}_u, \tag{14}$$

where $\boldsymbol{R}_d$ is a diagonal matrix, and $\boldsymbol{R}_u$ acts as a perturbational term. Obviously, as an example of the whole theory, expression (3) could be considered as a particular case of the sum (14).

There are various choices to perform such an arbitrary splitting as the one depicted in equation (14), and some of them are commented in ref. [2]. This extra degree of freedom can be particularly useful to override intermediate degeneracy problems or, in general, to enhance the method's convergence [2]. Moreover, the presence of non-zero diagonal elements in the $\boldsymbol{R}_u$ matrix indicates that new corrections are to be added to the eigenvalues. This explains why, in some cases, the method converges faster if, at every iterative step, the principal diagonal of the $\boldsymbol{R}_u$ matrix is set to zero and their elements are added to the corresponding $\boldsymbol{R}_d$ matrix ones. This option implies, in the RSPT scheme, that first order eigenvalue corrections become null [2].

Now, diagonalization of matrix $\boldsymbol{R}_0$ by means of the PT problem vector can be handled using the decomposition (14), and may be transformed into the new PT problem vector

$$(\boldsymbol{R}_0, \boldsymbol{R}_d, \boldsymbol{R}_u, \boldsymbol{I}, \boldsymbol{R}_d). \tag{15}$$

From the numerical point of view, PT problem vectors (13) and (15) are not equivalent. However, from the mathematical point of view, they define the *same* diagonalization problem. Therefore, and from now on, we will refer to equation (15) as the representative of both choices.

At the present situation of the procedure, it can be supposed that a suitable method is available to provide a correction matrix, Θ_1, associated to the canonical eigenvectors $\boldsymbol{I}$, appearing in the PT problem vector (15). *Optionally*, another diagonal matrix correction, Δ_1, attached to the related eigenvalues contained in $\boldsymbol{R}_d$ could be also made at hand. Both corrections are to be seen as an approximation to the $\boldsymbol{R}_0$ eigenparameters. As matrix $\boldsymbol{R}_0$ will not be, in general, symmetric along all the iterative process, it is important to note here that a method, as the one presently described, must be suitable to compute the corresponding corrections, even when applied over a general matrix diagonalization problem. For example, these corrections can come from the RSTP framework. In this case, the correction matrix Θ_1, can be the first eigenvector correction or a sum involving the lower order ones. The same can be said with respect of the eigenvalues.

Once the correction matrices Θ_1 and Δ_1, are known, a new set of approximate unnormalized eigenvectors and eigenvalues can be defined for the perturbed system matrix $\boldsymbol{R}_0$, appearing in the PT problem vector:

$$\boldsymbol{Z}_1 = \boldsymbol{I} + \Theta_1 \quad \wedge \quad \boldsymbol{E}_1 = \boldsymbol{E}_0 + \Delta_1. \tag{16}$$

In general, these approximations will not diagonalize the perturbed system matrix $\boldsymbol{R}_0$, but the new matrix pair $\{\boldsymbol{Z}_1, \boldsymbol{E}_1\}$ can be considered the eigenparameters of some matrix $\boldsymbol{H}_1$, which is defined by the following similarity transformation:

$$\boldsymbol{H}_1 = \boldsymbol{Z}_1 \boldsymbol{E}_1 (\boldsymbol{Z}_1)^{-1}. \tag{17}$$

The basic idea of the method, which is being described here, uses the assumption that the new matrix $\boldsymbol{H}_1$ can act as a new unperturbed system and, then, the matrix $\boldsymbol{R}_0$ diagonalization, associated to the PT problem vector (15), can be rewritten as a new PT problem vector:

$$(\boldsymbol{R}_0, \boldsymbol{H}_1, \boldsymbol{V}_1, \boldsymbol{Z}_1, \boldsymbol{E}_1), \tag{18}$$

which uses the information supplied by the new matrix pair, defined in equation (16). It is only necessary to furnish an explicit form for the new perturbative term $\boldsymbol{V}_1$, which will be defined as

$$\boldsymbol{V}_1 = \boldsymbol{R}_0 - \boldsymbol{H}_1. \tag{19}$$

In order to construct an iterative procedure, the PT vector problem (18) must be rewritten again in terms of the notation present within the PT vector problem (15). This can be performed defining a new transformed perturbation matrix $\boldsymbol{P}_1$ as

$$\boldsymbol{P}_1 = (\boldsymbol{Z}_1)^{-1}\boldsymbol{V}_1\boldsymbol{Z}_1, \tag{20}$$

and a new form for the matrix $\boldsymbol{R}_0$ can be deduced from expression (17):

$$\boldsymbol{R}_0 = \boldsymbol{Z}_1(\boldsymbol{E}_1 + \boldsymbol{P}_1)(\boldsymbol{Z}_1)^{-1}. \tag{21}$$

This allows writing the matrix to diagonalize $\boldsymbol{H}$ using equations (2) and (21) as

$$\boldsymbol{H} = \boldsymbol{Z}_0\boldsymbol{Z}_1(\boldsymbol{E}_1 + \boldsymbol{P}_1)(\boldsymbol{Z}_1)^{-1}(\boldsymbol{Z}_0)^{-1}. \tag{22}$$

The previous expression shows how an effective step has been done, in order to diagonalize $\boldsymbol{H}$.

A correction has been made involving the matrix $\boldsymbol{Z}_0$, and the central bracketed term has to be expected more diagonally dominant than $\boldsymbol{R}_0$ or $\boldsymbol{H}$. The formulation described until now is related to the so-called refinement method [14].

The required information is now prepared to start a new iteration. As equation (22) shows, the diagonalization of matrix $\boldsymbol{H}$ will be the same as to solve the diagonalization of the central bracketed term, taken as the following PT problem vector:

$$(\boldsymbol{R}_1, \boldsymbol{E}_1, \boldsymbol{P}_1, \boldsymbol{I}, \boldsymbol{E}_1), \tag{23}$$

which can be identified, in turn, with the former PT problem vector (15). This signals the possible way to restart the process, just from the PT problem vector (15), and construct the full computational procedure in an iterative manner.

The previous process can be repeated until convergence. At the kth iteration step, the procedure described above will lead to a generalization of the expression (22), which constructs, in turn, the original matrix to be diagonalized. Defining the matrix product: $\boldsymbol{Z}^{(k)} = \boldsymbol{Z}_0\boldsymbol{Z}_1 \cdots \boldsymbol{Z}_k$; then

$$\boldsymbol{H} = \boldsymbol{Z}^{(k)}(\boldsymbol{E}_k + \boldsymbol{P}_k)\left(\boldsymbol{Z}^{(k)}\right)^{-1}. \tag{24}$$

Due to the successive insertion of eigenvalue and eigenvector corrections of the kind described in equation (16), the method can converge to the diagonalization of the matrix $\boldsymbol{H}$ when $k \to \infty$. The best test for convergence achievement could be the computation of some appropriate norm of the matrix outdiag$\{\boldsymbol{E}_k + \boldsymbol{P}_k\}$, appearing in equation (24). In the case that this matrix part becomes negligible, then this will mean that no longer exists a relevant perturbation. So, the diagonalization of matrix $\boldsymbol{H}$ can be considered complete.

Along the iterative process, several sets of matrices are generated:

(1) The sequence of eigenvector corrections, $\{\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_k, \ldots\}$, which will produce, upon applying to them a sequential multiplication, a set of *unnormalized* eigenvectors for matrix $\boldsymbol{H}$:

$$\boldsymbol{Z} = \boldsymbol{Z}_0 \boldsymbol{Z}_1 \cdots \boldsymbol{Z}_k \cdots . \tag{25}$$

(2) A sequence of successive approximations to the eigenvalues of matrix $\boldsymbol{H}$: $\{\boldsymbol{E}_0, \boldsymbol{E}_1, \ldots, \boldsymbol{E}_k, \ldots\}$, which will converge towards the eigenvalues of matrix $\boldsymbol{H}$, producing the diagonal matrix

$$\boldsymbol{E} = \lim_{k \to \infty} \boldsymbol{E}_k . \tag{26}$$

In the same manner as occurs in the usual PT schemes, the set of eigenvalue approximations allows to define the sequence of successive eigenvalue corrections:

$$\left\{\boldsymbol{E}^{(1)} = \boldsymbol{E}_1 - \boldsymbol{E}_0,\ \boldsymbol{E}^{(2)} = \boldsymbol{E}_2 - \boldsymbol{E}_1, \ldots,\ \boldsymbol{E}^{(k)} = \boldsymbol{E}_k - \boldsymbol{E}_{k-1}, \ldots\right\}.$$

(3) A sequence of perturbation matrices $\{\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_k, \ldots\}$, which will converge towards the zero matrix

$$\boldsymbol{0} = \lim_{k \to \infty} \boldsymbol{P}_k . \tag{27}$$

(4) A sequence of matrices $\{\boldsymbol{R}_0, \boldsymbol{R}_1, \ldots, \boldsymbol{R}_k, \ldots\}$, which due to the previous conditions, will also tend to be diagonal and identical to the eigenvalue matrix $\boldsymbol{E}$

$$\boldsymbol{E} = \lim_{k \to \infty} \boldsymbol{R}_k = \lim_{k \to \infty} \boldsymbol{E}_k . \tag{28}$$

The basic descriptive ideas, which are the principal features of the proposed algorithm, can be resumed as follows:

1. A general PT or diagonalization problem is defined by means of the equations (10), (13) or (15).
2. Some eigenvalue and eigenvector corrections of the unperturbed system are obtained in order to approximate those of the perturbed one in equation (16).
3. The previous eigenparameter set corrections define a new matrix, which will be considered as a new unperturbed system in equation (17). This feature requires redefining the perturbation term as shown in equations (19) and (20).

4. The original PT problem is reformulated using the new unperturbed system defined in the previous step in equation (23).
5. Iterate to step 2 until convergence.

Due that, in step 3 above, a new unperturbed system matrix is defined, their eigenvectors generate, in turn, a *new* active space. Therefore, the working basis set is evolving at every step throughout the procedure. This constitutes an important difference with respect to the well-known RSPT framework, where the working basis set, $\boldsymbol{Z}_0$, is kept fixed from the beginning. Due to this characteristic, the proposed present procedure may be called *autoadjusting perturbation theory* (APT). As it will be shown below, such an algorithm converges faster than the usual RSPT scheme. Also, due to its iterative nature, the cost of the method is a linear function of the number of needed corrections.

In order to schematize the procedure outlined in the above discussion, there are described three possible algorithms, allowing the easy implementation of the theoretical framework.

4. General matrix algorithm

Algorithm 1 presents the computational matrix implementation of the described procedure. The algorithm is optimized in the sense that a minimal number of matrices must be kept within the computer RAM. Apart from the PT problem vector input data: $(\boldsymbol{H}, \boldsymbol{H}_0, \boldsymbol{V}, \boldsymbol{Z}, \boldsymbol{R})$, two additional process parameters are also needed, the procedure tolerance, ε, and the allowed maximal number of iterations, $k_{\max}$.

The eigenparameters of matrix $\boldsymbol{H}$ are returned within the same input matrices: $\boldsymbol{Z}$ will contain the eigenvectors and $\boldsymbol{R}$ the eigenvalues. Then, algorithm 1 constitutes a *destructive* procedure: a diagonalization procedure destroying the original eigenparameters. It is straightforward to modify it and transform the algorithm into a *non-destructive* one. In algorithm 1, the method, which supplies the eigenvector corrections, is the RSPT. Either first or second order eigenvector RSPT corrections can be used. The algorithm becomes slightly simplified if it is applied to solve a matrix diagonalization. For this case the unperturbed system can be taken to be $\boldsymbol{H}_0 = \mathrm{diag}\{\boldsymbol{H}\} = \boldsymbol{R}$, with the canonical eigenvectors collected in the unit matrix, the eigenvalues contained in the matrix $\boldsymbol{R}$ and the perturbation matrix is defined as $\boldsymbol{V} = \mathrm{outdiag}\{\boldsymbol{H}\}$. For this case, step 2 of the algorithm is irrelevant.

5. Vector algorithm

The bottleneck of algorithm 1 is found in the inversion of matrix $\boldsymbol{Z}_c$ at step 8a. Also, the similarity transformations of the kind: $(\boldsymbol{Z}_c)^{-1}\boldsymbol{R}\boldsymbol{Z}_c$, found in step 8b and the product $\boldsymbol{Z}\boldsymbol{Z}_c$, found in step 7b, are time consuming procedures. These characteristics lead to design a practical algorithm, which performs the search of one eigenvalue with the corresponding eigenvector at a time.

If only the pth vector of the active space is allowed to be transformed and the remaining unperturbed eigenvectors of matrix $\boldsymbol{H}_0$ are kept fixed, the correction matrix $\boldsymbol{Z}_c$ generated at step 7a of algorithm 1 takes the form

$$\boldsymbol{Z}_c = \boldsymbol{I} + \boldsymbol{W}, \tag{29}$$

where the $(n \times n)$ matrix $\boldsymbol{W}$ contains the eigenvector correction. When such a correction is supplied by the RSPT, it is well known that matrix $\boldsymbol{W}$ can be expressed as the null matrix, where the pth column has been replaced by a column vector $\boldsymbol{w}$, whose elements are defined by

$$\boldsymbol{w} = \left\{ w_{kp} \;\; \forall k \neq p \;\; \wedge \;\; w_{pp} = 0 \right\}. \tag{30}$$

The zero value attached to the element w_{pp} is particularly useful due to the following reasons:

(a) The product between the matrices $\boldsymbol{Z}$ and $\boldsymbol{Z}_c$, found at step 7b of algorithm 1, only adds to the pth column of matrix $\boldsymbol{Z}$ a column vector $\boldsymbol{s}$, defined by means of the following product:

$$\boldsymbol{s} = \boldsymbol{Z}\boldsymbol{w}. \tag{31}$$

(b) Owing to the characteristic structure of the matrix $\boldsymbol{Z}_c$, as previously defined in equation (29), posses an inverse, which must be computed at step 8a of algorithm 1, may be simply written as

$$(\boldsymbol{Z}_c)^{-1} = \boldsymbol{I} - \boldsymbol{W}. \tag{32}$$

Thus, the explicit expression for the transformation performed over the matrix $\boldsymbol{R} = \{r_{ij}\}$, found in step 8b of algorithm 1, is computed now by the following set of transformations:

$$\begin{aligned}
&\text{a) } i = p,\; j = p\text{: } r_{pp} \leftarrow r_{pp} + \boldsymbol{R}_{p:}\boldsymbol{w},\\
&\text{b) } i \neq p,\; j = p\text{: } r_{ip} \leftarrow r_{ip} + \boldsymbol{R}_{i:}\boldsymbol{w} - w_{ip}(r_{pp} + \boldsymbol{R}_{p:}\boldsymbol{w}),\\
&\text{c) } i = p,\; j \neq p\text{: } r_{pj} \leftarrow r_{pj},\\
&\text{d) } i \neq p,\; j \neq p\text{: } r_{ij} \leftarrow r_{ij} - w_{ip}r_{pj},
\end{aligned} \tag{33}$$

where the notation $\boldsymbol{R}_{i:}$ stands for the ith row vector of matrix $\boldsymbol{R}$.

All the submatrices of the involved matrices, remaining constant along the process, do not need to be stored in fast memory. The relevant vectors, which should be present, are $\boldsymbol{w}$ and $\boldsymbol{s}$. Expression (33) founds its most effective implementation when applied to the diagonalization of large matrices.

6. Application to the diagonalization of large matrices

If the APT method will be applied when the matrix $\boldsymbol{R}$ to be diagonalized is very large, it is well known that, usually, their elements are not stored in memory but can be

recalculated every time they are needed. The vector ***s***, defined in equation (31) plays the same role as the one found in the Davidson [13] or Nesbet [17] diagonalization algorithms. In this manner, the implementation of the APT algorithm does not require any extra programming effort.

One of the advantages of the present method lies in the fact that successive transformations over the eigenvectors, expressed in step 7b of algorithm 1, can be conceived as a unique transformation. In other words, as such corrections are additions over a column, it is only necessary to store all of them into a unique vector, which acts as an algorithm reservoir. Such is the reason why the APT method does not need a large amount of computer memory or extraordinary hard disk requirements.

Algorithm 2 describes the vector procedure. The successive eigenvector corrections are obtained from the first order RSPT scheme. When using algorithm 2, the convergence of the method decreases slightly with respect to the algorithm 1 performance, due to the fact that $n-1$ of the active space basis set vectors are kept constant. Despite of this characteristic, the entire procedure becomes *faster* that the equivalent PT procedure and a minimal amount of computer memory is needed. This permits to implement the algorithm in order to perform the diagonalization of large matrices.

Algorithm 3 is the partner of algorithm 2, but the eigenvector corrections are computed by means of a RSPT scheme going up to second order. The convergence of the method is ameliorated in a noticeable manner; however, this has its drawback: the bottleneck of the process, consisting in the operations equivalent to the computation of the ***s*** vector, have to be performed in algorithm 3 *twice* at every cycle in steps 6a and 6e. Also, the implementation of second order eigenvector corrections needs an additional auxiliary vector, $\boldsymbol{z}^{(a)}$.

In algorithms 2 and 3 a particular kind of residual vector is computed, the δ parameter used in steps 3d and 6d, respectively. Other choices can be considered.

7. Numerical results

Table 1 presents some numerical results obtained after application of the APT algorithm. Some Hilbert-like matrices were diagonalized using both, the APT and the RSPT vector algorithms. These matrices are defined as the Hilbert ones, but scaling the non-diagonal terms by a factory γ, in order to make them more diagonally dominant, if needed. Their elements are defined by means of

$$\boldsymbol{H} = \left\{ H_{ii} = \frac{1}{2i-1} \;\; \forall i \;\; \wedge \;\; H_{ij} = \frac{\gamma}{i+j-1} \;\; \forall i \neq j \right\}. \tag{34}$$

The eigenvector attached to the largest eigenvalue has been found for several cases. APT algorithm obtains the vector corrections using first or second order RSPT at every iterative step. The tolerance for the APT routine was set to 10^{-10} in order to achieve a very precise result. The quadratic error between the obtained and the exact

Table 1
Numerical results obtained using the APT algorithm. RSPT algorithm diverges in most cases and when convergence is achieved it requires much more iterations. See also figures and text for meaning.

Calculation number	Matrix order	γ coefficient	Order of the PT corrections	Number of iterations	Computed eigenvalue	Quadratic error in eigenvector
1	10	$1/2$	1	15	1.225121522	5.843×10^{-11}
2	10	$1/2$	2	10	1.225121522	3.993×10^{-11}
3	100	$1/2$	1	26	1.340201069	2.965×10^{-10}
4	100	$1/2$	2	13	1.340201069	7.362×10^{-11}
5	100	$2/5$	1	25	1.209680361	3.814×10^{-10}
6	100	$2/5$	2	12	1.209680360	2.575×10^{-10}
7	100	$1/3$	1	23	1.138587683	4.595×10^{-10}
8	100	$1/3$	2	12	1.138587683	6.041×10^{-11}
9	1000	$1/2$	1	36	1.404512233	1.694×10^{-9}
10	1000	$1/2$	2	18	1.404512233	7.381×10^{-10}

eigenvector is also shown. For small matrices the exact eigenparameters are obtained from a Jacobi routine, while for the (1000×1000) matrix a standard Davidson [13] algorithm was used.

APT obtained data were compared against the results coming from a RSPT vector program [2]. In many cases the RSPT procedure diverges. Only for calculations 7 and 8, RSPT convergence is achieved but more eigenparameter corrections are needed in order to reproduce the APT algorithm results. This shows how, in general, APT algorithm converges faster than the RSPT one.

In figures 1–7 it is visually analyzed the convergence behavior of the algorithms studied in this work. Figures 1, 3 and 7 show the convergence capabilities for calculations of the three algorithms (APT using first and second order eigenvector corrections and RSPT procedure) with respect to the number of iterations in APT routines. These figures correspond to the calculations 1 and 2, 3 and 4, and 9 and 10 of table 1, respectively. The number of iterations of the APT method, when using first order PT corrections, coincide with the correction order found in the usual RSPT. Figures 2, 4, 5, and 6 show the logarithm of the absolute difference between the exact eigenvalue and the actual one, obtained at every iteration for the algorithms used in calculations 1 and 2, 3 and 4, 5 and 6, and 7 and 8 of table 1, respectively. In *all* cases the first and second order APT methods give better results than the RSPT one.

8. Conclusions

Matrix and vector forms of a new and very general diagonalization procedure based on PT have been described. It has been shown how the computer implementation of the APT related algorithms is straightforward.

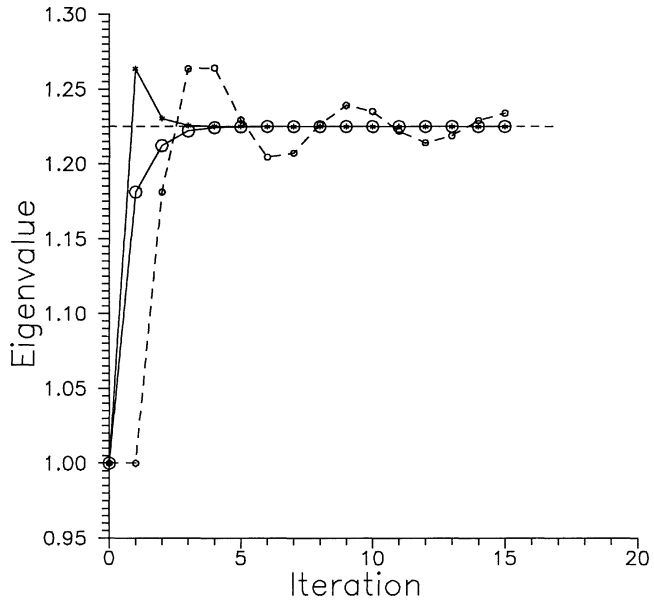

Figure 1. Convergence behavior of calculations 1 and 2 of table 1 for the three studied algorithms with respect to the number of iterations. Methods are: first order PT corrections in APT algorithm (big circles), second order eigenvector corrections in APT algorithm (stars) and RSPT results (small circles).

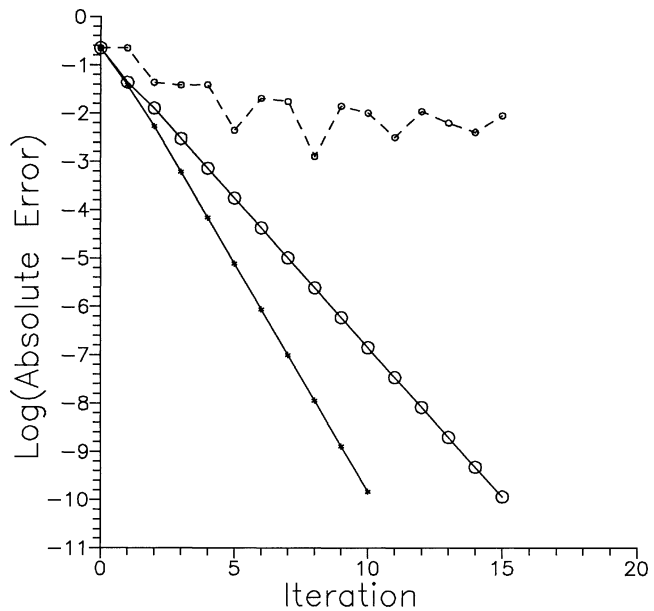

Figure 2. Logarithm of the absolute difference between the exact eigenvalue and the one obtained at every iteration and for each method in calculations 1 and 2 of table 1. Methods are: first order PT corrections in APT algorithm (big circles), second order eigenvector corrections in APT algorithm (stars) and RSPT results (small circles).

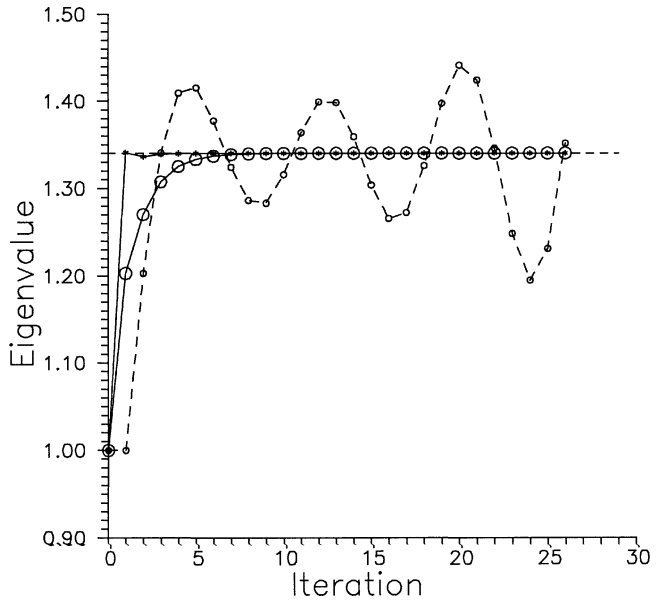

Figure 3. Convergence behavior of calculations 3 and 4 of table 1 for the three algorithms with respect to the number of iterations. Methods are: first order PT corrections in APT algorithm (big circles), second order eigenvector corrections in APT algorithm (stars) and RSPT results (small circles).

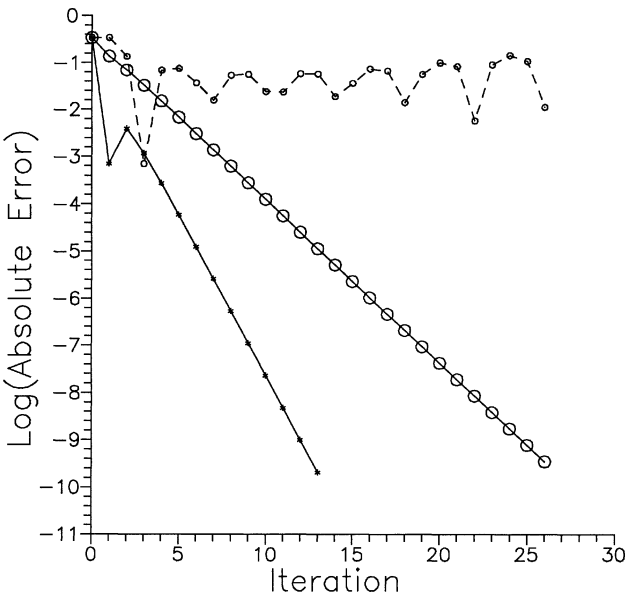

Figure 4. Logarithm of the absolute difference between the exact eigenvalue and the one obtained at every iteration for each method in calculations 3 and 4 of table 1. Methods are: first order PT corrections in APT algorithm (big circles), second order eigenvector corrections in APT algorithm (stars) and RSPT results (small circles).

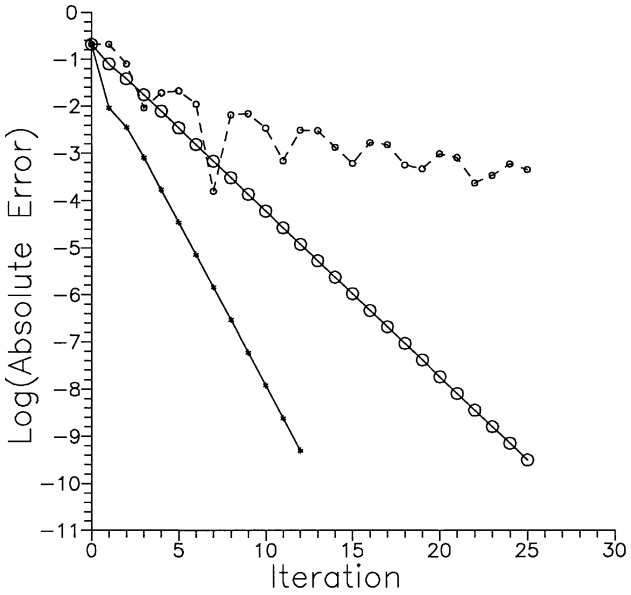

Figure 5. Logarithm of the absolute difference between the exact eigenvalue and the one obtained at every iteration for each method in calculations 5 and 6 of table 1. Methods are: first order PT corrections in APT algorithm (big circles), second order eigenvector corrections in APT algorithm (stars) and RSPT results (small circles).

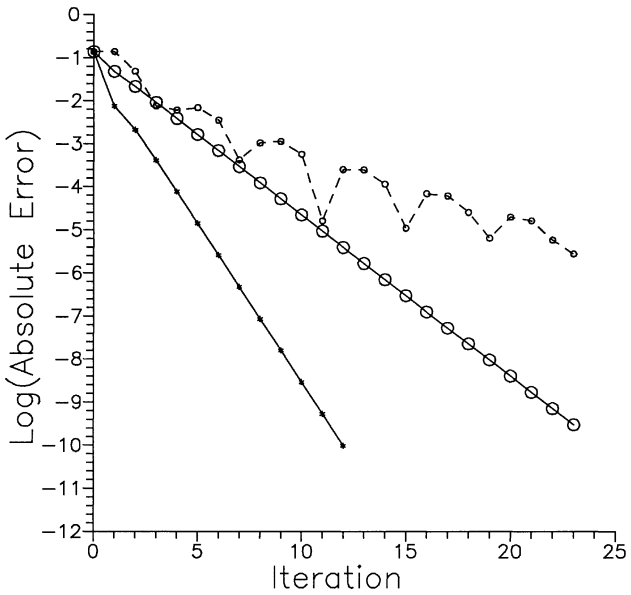

Figure 6. Logarithm of the absolute difference between the exact eigenvalue and the one obtained at every iteration for each method in calculations 7 and 8 of table 1. Methods are: first order PT corrections in APT algorithm (big circles), second order eigenvector corrections in APT algorithm (stars) and RSPT results (small circles).

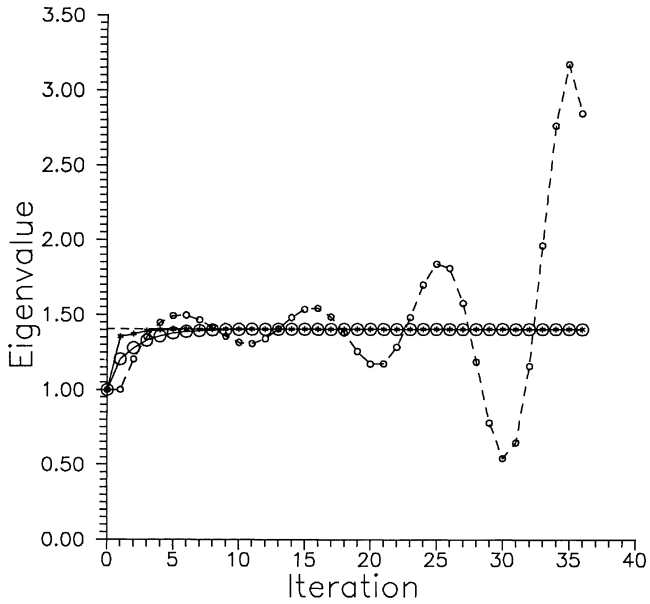

Figure 7. Convergence behavior of calculations 9 and 10 of table 1 for the three algorithms with respect to the number of iterations. Methods are: first order PT corrections in APT algorithm (big circles), second order eigenvector corrections in APT algorithm (stars) and RSPT results (small circles).

As an overview of the mathematical and practical characteristics of the APT method, some advantages can be stressed again:

- As APT algorithm is iterative, no error accumulation is found. The final error is given by the last iteration.
- Only a few vectors must be kept in memory and this feature allows the treatment of huge matrices.
- The method can be applied to the diagonalization of *any matrix column* if the matrix is properly conditioned. The convergence is ensured for strongly diagonally dominant matrices as those found in typical CI calculations and in other scientific fields [3,4].
- The involved algebra is general and the method can be used over *any kind* of square matrices.
- Finally, APT converges in all studied cases, while RSPT clearly shows a divergent behavior.

Acknowledgements

This work has been financed by Spanish CICYT under the Programa Nacional de Salud y Farmacia through the grant SAF96-6158.

Appendix

Algorithm 1. APT full matrix diagonalization procedure.

Solves the perturbation problem: $\boldsymbol{H} = \boldsymbol{H}_0 + \boldsymbol{V}$ where the eigenparameters $\{\boldsymbol{Z}, \boldsymbol{R}\}$ of $\boldsymbol{H}_0$ are known. The algorithm returns the solution in the same $\boldsymbol{Z}$ and $\boldsymbol{R}$ matrices: at the end of the procedure the equality: $\boldsymbol{HZ} = \boldsymbol{ZR}$ is obtained if convergence is achieved.

1. Initializations. Set: $k = 0$. Iterations counter.
   $k_{\max} > 0$. Maximal number of iterations.
   $\varepsilon > 0$. Process tolerance.
2. $\boldsymbol{R} = \boldsymbol{E} + \boldsymbol{Z}^{-1}\boldsymbol{VZ}$.
3. Compute a norm for the non diagonal part of $\boldsymbol{R}$: $N_0 = |\,\text{outdiag}\{\boldsymbol{R}\}|$.
4. If $N_0 < \varepsilon$, then: Stop procedure. Convergence achieved.
   Else if $k = k_{\max}$ then: Stop procedure. Convergence not achieved.
5. Create a $\boldsymbol{R}$ matrix perturbation problem: Split $\boldsymbol{R}$ into the sum $\boldsymbol{R} = \boldsymbol{R}_d + \boldsymbol{R}_u$, where $\boldsymbol{R}_d = \text{diag}\{\boldsymbol{R}\}$ and $\boldsymbol{R}_u = \text{outdiag}\{\boldsymbol{R}\}$.
6. Obtain a first order correction matrix, $\boldsymbol{Z}_c$, for the $\boldsymbol{R}_d$ eigenvectors (the canonical basis set $\boldsymbol{I}$), in order to obtained a suitable approximation for the $\boldsymbol{R}$ matrix eigenspace:
$$\boldsymbol{Z}_c = \left\{ z_{ij} = \frac{R_{ij}}{R_{ii} - R_{jj}} \;\; \forall i \neq j; z_{ii} = 0 \right\}.$$
   If second order corrections are to be used, then the following substitution holds:
$$z_{ij} \leftarrow z_{ij} - \frac{[\boldsymbol{R}_u \boldsymbol{Z}_c]_{ij}}{R_{ii} - R_{jj}} \quad \forall i \neq j.$$
7. Update (a) $\boldsymbol{Z}_c \leftarrow \boldsymbol{I} + \boldsymbol{Z}_c$;
   (b) $\boldsymbol{Z}_c \leftarrow \boldsymbol{ZZ}_c$.
8. (a) Compute $(\boldsymbol{Z}_c)^{-1}$;
   (b) Update $\boldsymbol{R} \leftarrow (\boldsymbol{Z}_c)^{-1}\boldsymbol{RZ}_c$.
9. $k \leftarrow k + 1$. Go to step 3.

Algorithm 2. First order vector APT diagonalization procedure algorithm.

Finds one eigenvalue $\{e\}$ and the related unnormalized eigenvector $\{\boldsymbol{z}\}$ of a $(n \times n)$ matrix $\boldsymbol{R}$. The pth column of the matrix is diagonalized. Uses first order RSPT corrections.

1. Initializations. Set: $k = 0$. Iteration counter.
   $k_{\max} > 0$. Maximal number of iterations.
   $1 \leqslant p \leqslant n$. Column to diagonalize.
   $\varepsilon > 0$. Process tolerance.
   $\delta = \infty$. Convergence test.
2. $\boldsymbol{z} = \{z_i = \frac{R_{ip}}{R_{pp} - R_{ii}} \ \forall i \neq p;\ z_p = 1\}$: first order eigenvector corrections.
3. Do while $\delta > \varepsilon$ and $k < k_{\max}$:
   (a) $\boldsymbol{s} = \boldsymbol{R}\boldsymbol{z}$.
   (b) $e = s_p$: actual eigenvalue approximation.
   (c) Add new first order eigenvector correction.
   Here, the new eigenvector approximation is obtained:

$$z_i \leftarrow z_i + \frac{s_i - z_i e}{e - R_{ii} + z_i R_{pi}} \quad \forall i \neq p.$$

   (d) $\delta = \frac{1}{n} \sum_{i \neq p} |s_i - z_i e|$: new column residual.
   (e) $k \leftarrow k + 1$: increment iteration counter
   end do
4. If $\delta > \varepsilon$ then the procedure has not converged. Stop process.
5. Stop procedure: convergence achieved.

Algorithm 3. Second order vector APT diagonalization procedure.

Finds one eigenvalue $\{e\}$ and the related unnormalized eigenvector $\{\boldsymbol{z}\}$ of a $(n \times n)$ matrix $\boldsymbol{R}$. The pth column of the matrix is diagonalized. Uses second order RSPT corrections.

1. Initializations. Set: $k = 0$. Iterations counter.
   $k_{\max} > 0$. Maximal number of iterations.
   $1 \leqslant p \leqslant n$. Column to diagonalize.
   $\varepsilon > 0$. Process tolerance.
   $\delta = \infty$. Convergence test.
2. $\boldsymbol{z} = \{z_i = \frac{R_{ip}}{R_{pp} - R_{ii}} \ \forall i \neq p;\ z_p = 1\}$: first order eigenvector corrections.
3. $\boldsymbol{z}^{(a)} = \boldsymbol{z}$: auxiliary vector.
4. $\boldsymbol{s} = \text{outdiag}\{\boldsymbol{R}\}\boldsymbol{z}^{(a)}$: second order eigenvector corrections.
5. $z_i \leftarrow z_i + \frac{s_i}{R_{pp} - R_{ii}} \ \forall i \neq p$: first and second order eigenvector corrections.
6. Do while $\delta > \varepsilon$ and $k < k_{\max}$:
   (a) $\boldsymbol{s} = \boldsymbol{R}\boldsymbol{z}$: updates the global $\boldsymbol{R}\boldsymbol{z}$ transformation.
   (b) $e = s_p$: actual eigenvalue approximation.
   (c) Keeps new first order eigenvector corrections.

$$z_i^{(a)} = \frac{s_i - z_i e}{e - R_{ii} + z_i R_{pi}} \quad \forall i \neq p.$$

   (d) $\delta = \frac{1}{n} \sum_{i \neq p} |s_i - z_i e|$: new column residual.
   (e) $\boldsymbol{s} = \text{outdiag}\{\boldsymbol{R}\}\boldsymbol{z}^{(a)}$.
   (f) Computation of the second order eigenvector corrections.
   The new eigenvector approximation is also obtained:

$$s_i \leftarrow s_i - z_i \sum_{j \neq p} R_{pj} z_j^{(a)} \quad \forall i \neq p;$$

$$z_i \leftarrow z_i + z_i^{(a)} + \frac{s_i}{e - R_{ii} + z_i R_{pi}} \quad \forall i \neq p.$$

   (g) $k \leftarrow k + 1$: increment iteration counter
   end do
7. If $\delta > \varepsilon$ then procedure has not converged. Stop process.
8. Stop procedure: convergence achieved.

References

[1] E. Besalú and R. Carbó, J. Math. Chem. 15 (1994) 397–406.
[2] E. Besalú and R. Carbó, J. Math. Chem. (submitted).
[3] A. Booten and H. van der Vorst, Comput. Phys. 10(3) (1996) 239–242.
[4] A. Booten and H. van der Vorst, Comput. Phys. 10(4) (1996) 331–334.
[5] R. Carbó, Theoret. Chimica Acta 17 (1970) 74.
[6] R. Carbó, Rev. Roumaine de Chimie 16 (1971) 1155.
[7] R. Carbó, Int. J. Quantum Chem. 6 (1972) 609.
[8] R. Carbó and E. Besalú, J. Math. Chem. 13 (1993) 331–342.
[9] R. Carbó and R. Gallifa, Nuovo Cimento 10 (1972) 576.
[10] R. Carbó and R. Gallifa, Anales de Física 68 (1972) 197.
[11] R. Carbó and R. Gallifa, Nuovo Cimento 17 (1973) 46.
[12] R. Carbó and R. Gallifa, Anales de Física 69 (1973) 331.
[13] E.R. Davidson, J. Comput. Phys. 17 (1975) 87–94.
[14] E. Durand, *Solutions Numériques des Équations Algébriques*, Vol. 2 (Masson and Cie, Paris, 1961).
[15] A.S. Householder, *The Theory of Matrices in Numerical Analysis* (Blaisdell Publ. Co., New York, 1964).
[16] C.G.J. Jacobi, J. Reine Angew. Math. 30 (1846) 51.
[17] R.K. Nesbet, J. Chem. Phys. 43 (1965) 311.