Theor Chem Acc (1999) 103:163–166
DOI 10.1007/s002149900057

Theoretical Chemistry Accounts

Regular article

Remarks on large-scale matrix diagonalization using a Lagrange–Newton–Raphson minimization in a subspace

Josep Maria Anglada[1], **Emili Besalú**[2], **Josep Maria Bofill**[3]

[1] C.I.D. – C.S.I.C., Jordi Girona Salgado 18–26, E-08034 Barcelona, Catalunya, Spain e-mail: anglada@qteor.cid.csic.es
[2] Institut de Química Computacional i Departament de Química, Universitat de Girona, Campus de Montilivi, E-17071 Girona, Catalunya, Spain e-mail: emili@stark.udg.es
[3] Departament de Química Orgànica i Centre Especial de Recerca en Química Teòrica, Universitat de Barcelona, Martí i Franquès 1, E-08028 Barcelona, Catalunya, Spain e-mail: jmbofill@qo.ub.es

Received: 16 February 1999 / Accepted: 10 May 1999 / Published online: 9 September 1999

Abstract. We present a matrix diagonalization method where the diagonalization is carried out through a normal Lagrange–Newton–Raphson method solved in a subspace. The subspace is generated using the correction vector that predicts the standard Lagrange–Newton–Raphson formula in the full space. Some numerical examples and the performance of the algorithm are given.

Key words: Large-scale matrix diagonalization – Lagrange–Newton–Raphson minimization – Ritz–Galerkin method

1 Introduction

Many physical and chemical problems require the solution of the large-scale eigenvalue equation

$$\mathbf{HC} = \lambda\mathbf{C} \ , \tag{1}$$

where $\mathbf{H}$ is a symmetric real matrix, $\mathbf{C}$ is the eigenvector with the requirement that $\mathbf{C}^{\mathrm{T}}\mathbf{C} = 1$, and λ is the corresponding eigenvalue. Due to the large dimension of the eigenvalue problem (Eq. 1), only iterative methods have been developed which require a matrix–vector product. All these iterative methods are based on the Ritz–Galerkin algorithm [1], which solves the equation under study in a subspace. The efficiency of the methods based on the Ritz–Galerkin technique depends on the efficiency of the construction of the subspace basis vector. Within these types of methods the most popular is the Davidson algorithm [2], though others methods have also been developed [3–8]. These methods can be seen as derivations of the Lanczos-type algorithm [9–11] with a precondition [12–13].

Correspondence to: J.M. Bofill

Here we propose a large-scale diagonalization method based on the direct solution of the Lagrangian function associated with the eigenvalue problem [14]. The algorithm can be applied to the evaluation of the lower and higher roots.

2 Theory and computational details

The eigenvalue problem can be seen as the solution of the stationary condition of the Lagrangian function [14]

$$\mathrm{L}(\mathbf{C},\lambda) = \mathbf{C}^{\mathrm{T}}\mathbf{HC} - \lambda\left(\mathbf{C}^{\mathrm{T}}\mathbf{C} - 1\right) \ , \tag{2}$$

where the Lagrangian multiplier λ, also called the Raigleigh–Ritz quotient, is the corresponding eigenvalue at the solution. Expanding Eq. (2) with respect to $\mathbf{C}$ and λ to second order and applying the stationary conditions, we get

$$\begin{pmatrix} 2\mathbf{HC} - 2\lambda\mathbf{C} \\ -\left(\mathbf{C}^{\mathrm{T}}\mathbf{C} - 1\right) \end{pmatrix} = -\begin{pmatrix} 2\mathbf{H} - 2\lambda\mathbf{I} & -2\mathbf{C} \\ -2\mathbf{C}^{\mathrm{T}} & 0 \end{pmatrix}\begin{pmatrix} \delta\mathbf{C} \\ \delta\lambda \end{pmatrix} \ , \tag{3}$$

where $\mathbf{I}$ is the unit matrix. Equation (3) is the so-called Lagrange–Newton–Raphson equation of Eq. (2) and can be written in the following way:

$$\mathbf{HC} - (\lambda + \delta\lambda)\mathbf{C} = -(\mathbf{H} - \lambda\mathbf{I})\delta\mathbf{C} \tag{4a}$$

$$\mathbf{C}^{\mathrm{T}}\mathbf{C} - 1 = -2\mathbf{C}^{\mathrm{T}}\delta\mathbf{C} \ . \tag{4b}$$

If the $\mathbf{C}$ vector is normalized and its Raigleigh–Ritz quotient is $\lambda = \mathbf{C}^{\mathrm{T}}\mathbf{HC}$, from Eq. (4b) we get $\mathbf{C}^{\mathrm{T}}\delta\mathbf{C} = 0$ and from Eq. (4a) after some rearrangement we obtain the improvement of both the Lagrangian multiplier, λ, and the $\mathbf{C}$ vector, i.e.,

$$\delta\lambda = \frac{\mathbf{C}^{\mathrm{T}}(\mathbf{H} - \lambda\mathbf{I})^{-1}(\mathbf{HC} - \lambda\mathbf{C})}{\mathbf{C}^{\mathrm{T}}(\mathbf{H} - \lambda\mathbf{I})^{-1}\mathbf{C}} \tag{5}$$

$$\delta\mathbf{C} = -(\mathbf{H} - \lambda\mathbf{I})^{-1}(\mathbf{HC} - (\lambda + \delta\lambda)\mathbf{C}) \ . \tag{6}$$

Equations (5) and (6) were derived by Olsen et al. [4] and Bofill and Anglada [7]. From these equations a new Lagrangian multiplier, $\lambda \leftarrow \lambda + \delta\lambda$, and $\mathbf{C}$ vector, $\mathbf{C} \leftarrow \mathbf{C} + \delta\mathbf{C}$, are obtained. Equations (4)–(6) are the bases of the Lagrange–Newton–Raphson algorithm to solve the eigenvalue equation (Eq. 1).

In order to introduce stability into the preceding algorithm, we use the Ritz–Galerkin technique [1], i.e., Eq. (4a) is solved in a subspace of an orthonormal basis generated by the set of vectors $\delta\mathbf{C}$ that are evaluated at each iteration using Eq. (6). The outline of the algorithm is as follows:

1. Begin with a normalized vector $\mathbf{C}_0$ and the corresponding $\mathbf{HC}_0$. Store both vectors. Set $i = 1$.
2. Compute the Raigleigh–Ritz quotient $\lambda_i = \mathbf{C}_i^{\mathrm{T}}\mathbf{HC}_i$ and the residuum vector $\mathbf{r}_i = \mathbf{HC}_i - \lambda_i\mathbf{C}_i$.
3. Compute an improvement to both the Raigleigh–Ritz quotient and the $\mathbf{C}_i$ vector using slightly modified versions of Eqs. (5) and (6):

$$\delta\lambda_i = \frac{\mathbf{C}_i^{\mathrm{T}}(\mathbf{H}_{\mathrm{D}} - \lambda_i\mathbf{I})^{-1}\mathbf{r}_i}{\mathbf{C}_i^{\mathrm{T}}(\mathbf{H}_{\mathrm{D}} - \lambda_i\mathbf{I})^{-1}\mathbf{C}_i} \tag{7}$$

$$\delta\mathbf{C}_i = -(\mathbf{H}_{\mathrm{D}} - \lambda_i\mathbf{I})^{-1}(\mathbf{r}_i - \delta\lambda_i\mathbf{C}_i) \ , \tag{8}$$

where $\mathbf{H}_{\mathrm{D}}$ is the diagonal part of the $\mathbf{H}$ matrix.

4. Perform a Gram–Schmidt orthonormalization of the $\delta\mathbf{C}_i$ vector. With the corresponding orthonormalized vector, $\mathbf{q}_i$, compute $\mathbf{Hq}_i$. Store both vectors in the matrices $\mathbf{Q}_i = (\mathbf{Q}_{i-1}|\mathbf{q}_i)$ and $\mathbf{HQ}_i = (\mathbf{HQ}_{i-1}|\mathbf{Hq}_i)$.
5. Solve the Lagrange–Newton–Raphson equation (Eq. 4a) in the subspace defined by the set of orthonormal vectors $\{\mathbf{q}_j\}_{j=1}^{i}$.

$$\mathbf{Q}_i^{\mathrm{T}}[\mathbf{HC}_i - (\lambda_i + \delta\lambda_i)\mathbf{C}_i] = -\mathbf{Q}_i^{\mathrm{T}}[\mathbf{HC}_i - \lambda_i\mathbf{I}]\mathbf{Q}_i\mathbf{a}_i \ , \tag{9}$$

where the $\mathbf{a}_i$ vector is trivially computed in the following way

$$\mathbf{a}_i = -\left\{\mathbf{Q}_i^{\mathrm{T}}[\mathbf{HC}_i - \lambda_i\mathbf{I}]\mathbf{Q}_i\right\}^{-1}\mathbf{Q}_i^{\mathrm{T}}[\mathbf{HC}_i - (\lambda_i + \delta\lambda_i)\mathbf{C}_i] \ . \tag{10}$$

6. If the norm of the vector $\mathbf{g}_i = \mathbf{Q}_i^{\mathrm{T}}[\mathbf{HC}_i - (\lambda_i + \delta\lambda_i)\mathbf{C}_i]$, i.e., the gradient vector in the subspace, is smaller than a given threshold, $(\mathbf{g}_i^{\mathrm{T}}\mathbf{g}_i)^{1/2} \leq \varepsilon$, convergence has been reached, otherwise compute the improved vector $\mathbf{C}_i' = \mathbf{C}_i + \mathbf{Q}_i\mathbf{a}_i$. Normalize $\mathbf{C}_i'$, i.e., $\mathbf{C}_{i+1} = N\mathbf{C}_i'$, where N is the normalization factor. Compute $\mathbf{HC}_{i+1} = N(\mathbf{HC}_i + \mathbf{HQ}_i\mathbf{a}_i)$ and store both $\mathbf{C}_{i+1}$ and $\mathbf{HC}_{i+1}$. Set $i = i + 1$ and go to step 2.

Note that if the convergence is reached after i iterations the number of the $\mathbf{HC}$ vector evaluations is $i + 1$. The algorithm only needs to store three vectors in the high-speed memory, $\mathbf{C}_i$, $\mathbf{HC}_i$ and $\mathbf{H}_{\mathrm{D}}$. Finally we say that Eq. (9) results from the projection of Eq. (4a) into the subspace spanned by the set of the vectors $\{\mathbf{q}_j\}_{j=1}^{i}$, i.e., the set of column vectors that define the matrix $\mathbf{Q}_i$. In this subspace $\delta\mathbf{C}_i = \mathbf{Q}_i\mathbf{a}_i$. We call this algorithm LNR.

It is possible to introduce the variance minimization technique [3, 8, 15] into the above algorithm. The variance minimization technique forces the convergence to the eigenpair with the eigenvalue nearest the initial Raigleigh–Ritz quotient [16]. In this case step 6 is modified in the following way. If the convergence criterion is not satisfied then minimize

$$\left(\mathrm{b}_1^i\ \mathrm{b}_2^i\right)(\mathbf{C}_i|\mathbf{C}_i + \mathbf{Q}_i\mathbf{a}_i)^{\mathrm{T}}[\mathbf{H} - (\lambda_i + \delta\lambda_i)\mathbf{I}] \times [\mathbf{H} - (\lambda_i + \delta\lambda_i)\mathbf{I}](\mathbf{C}_i|\mathbf{C}_i + \mathbf{Q}_i\mathbf{a}_i)\begin{pmatrix}\mathrm{b}_1^i\\ \mathrm{b}_2^i\end{pmatrix} \tag{11}$$

subject to the restriction

$$\left(\mathrm{b}_1^i\ \mathrm{b}_2^i\right)(\mathbf{C}_i|\mathbf{C}_i + \mathbf{Q}_i\mathbf{a}_i)^{\mathrm{T}}(\mathbf{C}_i|\mathbf{C}_i + \mathbf{Q}_i\mathbf{a}_i)\begin{pmatrix}\mathrm{b}_1^i\\ \mathrm{b}_2^i\end{pmatrix} = 1 \ . \tag{12}$$

Compute the improved vector $\mathbf{C}_i' = \mathrm{b}_1^i\mathbf{C}_i + \mathrm{b}_2^i(\mathbf{C}_i + \mathbf{Q}_i\mathbf{a}_i)$ and proceeding as before normalize it obtaining the new $\mathbf{C}_{i+1}$. In the same way compute the vector $\mathbf{HC}_{i+1} = N\left[\mathrm{b}_1^i\mathbf{HC}_i + \mathrm{b}_2^i\mathbf{H}\ (\mathbf{C}_i + \mathbf{Q}_i\mathbf{a}_i)\right]$ and continue as described above. This LNR algorithm coupled with the variance minimization technique is labeled LNRV.

3 Performance and numerical examples

We have performed three different numerical tests to illustrate the algorithms presented above. First, we have considered the negative scaled Hilbert matrix

$$\mathbf{H}_{ij}(\gamma) = \begin{cases} \frac{-1}{2i-1} & \text{if } i = j \\ \frac{-1}{\gamma(i+j-1)} & \text{if } i \neq j \end{cases} \tag{13}$$

taking $\gamma = 10$ and a dimension of 100 000, second the 50×50 Raffenetti matrix [17], which despite its small dimension is very difficult to diagonalize, and finally we have studied the $1^2\mathrm{B}_2$ and $3^2\mathrm{B}_2$ electronic states of CH_2^+ using a full configuration interaction (FCI) wave function correlating the five valence electrons. The molecular parameters are the bond distance C—H = 1.120 Å and the bond angle H—C—H = 101.8°. The basis set is the same as used in Ref. [18], namely $(9s5p1d)/[4s2p1d]$ for C and $(4s1p)/[2s1p]$ [19] for H. The dimension of the FCI is 176264 Slater determinants. The FCI calculations were carried out using the program described in Ref. [20].

The first four eigenvalues of the Hilbert matrix, the first six eigenvalues of the Raffenetti matrix, and the first

Table 1. The first lower eigenvalues of the Hilbert, Raffenetti and full configuration interaction (*FCI*) matrices

Root	Hilbert[a]	Raffenetti[b]	FCI[c]
1	−1.009610	0.033608	−38.483502
2	−0.353981	0.143251	
3	−0.200494	0.251975	−38.147909
4	−0.154196	0.362343	
5		2.349421	
6		10.349958	

[a] The dimension of the Hilbert matrix is 100 000
[b] The dimension of the Raffenetti matrix is 50
[c] The dimension of the FCI matrix is 176264. This FCI corresponds to the electronic structure calculation of CH_2^+ in C_{2V} symmetry. The values are given in atomic units. The first eigenvalue corresponds to the $1^2\mathrm{B}_2$ electronic state and the third eigenvalue is the $3^2\mathrm{B}_2$ electronic state

and third eigenvalues of the FCI matrix are given in Table 1. All the roots presented in Table 1 were optimized using the LNR and LNRV methods. With the aim of making comparisons, another method was used, the so-called crude LNR (cLNR), i.e. the LNR method without the construction of the orthonormal subspace. In this algorithm the new **C** vector is built by adding the correction vector obtained from Eq. (8) to the current **C** vector. The square root of the $\mathbf{HC} - \lambda\mathbf{C}$ vector norm was taken as the convergence criterion.

The number of iterations needed by the above algorithms to converge to roots considered in Table 1 is shown in Table 2. The convergence criterion was taken as $\varepsilon = 10^{-4}$. Except for some roots, as discussed later, the initial vector was taken as the unit vector with the 1 placed in the more important component of the desired converged eigenvector.

The convergence performance of the cLNR method is very poor since it converges only for the first and second roots of the Hilbert matrix. In other cases it tends to converge to other roots or does not converge. This manifestly reveals the very well know fact that the Ritz–Galerkin technique [1] introduces stability into the diagonalization process.

The other two methods, LNR and LNRV, converge very well and show similar behavior except for the optimization of roots 3, 4, and 5 of the Raffenetti matrix and root 4 of the Hilbert matrix. For these four roots the LNR method does not converge to the desired eigenpairs. This result can be explained by looking at Table 3, where the first six components of the eigenvectors corresponding to the six lowest eigenvalues of the Raffenetti matrix are shown. Starting with an initial vector whose components 3 and 4 are −0.707 and 0.707, respectively, the LNR method converges to the third root within 11 iterations. Proceeding in a similar way, if the composition of the initial vector is −0.53 and 0.88 at positions 4 and 5 the LNR method converges to the fourth root using the same number of iterations. Finally, selecting the initial vector such that components 2–5 are equal to 0.5 then the LNR method converges to root 5 within 11 iterations. On the other hand, the LNRV method converges to roots 3 and 4 starting with the corresponding initial unit vector and employing only ten iterations. For root 5 the starting vector was 0.707 and 0.707 at components 4 and 5, respectively. With this initial vector the LNRV method converges to this root after nine iterations. For root 5, the selection of this initial vector rather than the unit vector is obvious by looking at column 6 of Table 3. In this case is not possible to start with the unit vector to converge to this root.

Finally, for the FCI case the starting vector has six components different to zero, which correspond to the four configuration state functions describing the four lowest electronic states of $^2B_2$ symmetry. For the lowest state, Table 2 shows that convergence is reached with ten iterations using the LNR method and 11 iterations using the LNRV method. On the other hand, the third state has two main components in its wave function with coefficients 0.84 and 0.25. In this case both methods converge after the same number of iterations, 21. In order to check the convergence behavior of these methods we also performed a further diagonalization for this third state using as the initial vector a unit vector where the 1 is placed in the component of the highest weight in the final FCI eigenvector. With this starting vector, both LNR and LNRV methods converge to the desired root with 20 and 30 iterations, respectively.

Table 2. Number of iterations needed to optimize the different eigenvalues of the Hilbert, Raffenetti and FCI matrices given in Table 1 using the cLNR, LNR and LNRV methods. The normalized initial vector was the unit vector with the 1 placed in the most important component of the desired eigenvector. The convergence criterion was $\varepsilon = 10^{-4}$

Root	Hilbert				Raffenetti						FCI	
	1	2	3	4	1	2	3	4	5	6	1	3
cLNR	7	25	[a]	[a]	[b]	[b]	[b]	[b]	[b]	[b]	[b]	[b]
LNR	4	7	10	11[c]	8	9	11[d]	11[d]	11[d]	5	10[d]	20 (21)[e]
LNRV	4	7	9	10	8	11	10	10	9[d]	8	11[d]	30 (21)[e]

[a] It converges to the second root
[b] No convergence is reached within 30 iterations
[c] It converges to higher root
[d] The initial vector is not the unit vector. See text for more details
[e] The number of iterations needed to converge using an initial vector different to the unit vector is given in parentheses. See text for more details

4 Conclusions

We have presented a large-scale diagonalization method based on the direct solution of the Lagrange–Newton–Raphson equation in the iterative subspace. This subspace is built using the correction vector obtained from the Lagrange–Newton–Raphson formula in the full space. This algorithm can even be used to optimize high roots starting with only one vector. The coupling of this algorithm with the variance minimization technique does not improve the number of iterations needed to reach convergence. The real improvement of the variance minimization technique merely consists of converging to the eigenpair with the eigenvalue closer to the

Table 3. The first six components of the eigenvectors associated with the first six eigenvalues of the Raffenetti matrix. The dimension of the Raffenetti matrix is 50

Root	0.033608	0.143251	0.251975	0.362343	2.349421	10.349958
1	0.869918	−0.226516	−0.131669	−0.086303	0.349186	0.032651
2	−0.440358	−0.750236	−0.218307	−0.119200	0.411710	0.032970
3	−0.175707	0.571800	−0.638333	−0.192624	0.430865	0.033294
4	−0.109749	0.207012	0.690829	−0.501601	0.451889	0.033626
5	−0.079795	0.126384	0.224132	0.830412	0.475069	0.033964
6	−0.002933	0.003292	0.003403	0.003245	−0.121051	0.965650

initial Raigleigh–Ritz quotient even if one starts with a very poor initial vector.

Acknowledgements. We are indebted to Professor S. Olivella for his valuable suggestions. This research was supported by the Spanish DGICYT (grants PB95-0278-C02-01 and PB95-0278-C02-02).

References

1. (a) Ritz W (1909) J Rein Angew Math 135: 1; (b) Galerkin BG (1915) Vestn Ingenerov 1: 897
2. (a) Davidson ER (1975) J Comput Phys 17: 87; (b) Davidson ER (1980) J Phys A Math Gen 13: L179; (c) Davidson ER (1989) Comput Phys Comm 53: 49; (d) Murray CW, Racine SC, Davidson ER (1992) J Comput Phys 103: 382
3. Wood DM, Zunger A (1985) J Phys A Math Gen 18: 1343
4. Olsen J, Jørgensen P, Simons J (1990) Chem Phys Lett 169: 463
5. van Lenthe JH, Pulay P (1990) J Comput Chem 11: 1164
6. Mitin AV (1994) J Comput Chem 15: 747
7. Bofill JM, Anglada JM (1994) Chem Phys 183: 19
8. Besalú E, Bofill JM (1998) J Comput Chem 19: 1777
9. (a) Lanczos C (1950) J Res Natl Bur Stand (US) 45: 255; (b) Lanczos C (1952) J Res Natl Bur Stand (US) 49: 33
10. Karush W (1951) Pac J Math 1: 233
11. Hestenes MR, Stiefel E (1952) J Res Natl Bur Stand (US) 49: 498
12. Parlett BN (1980) The symmetryc eigenvalue problem. Prentice Hall, Englewood Cliffs
13. Saad Y (1992) Numerical methods for large eigenvalue problems. Manchester University Press, Wiley
14. Carbó R, Hernández JA (1976) Introducción a la teoría de matrices. Editorial Alhambra, Madrid
15. Feler MG (1974) J Comput Phys 14: 341
16. Weinstein DH (1934) Proc Natl Acad Sci USA 20: 529
17. Raffenetti RC (1979) J Comput Phys 32: 403
18. Anglada JM, Bofill JM (1995) Theor Chim Acta 92: 369
19. Dunning TH (1970) J Chem Phys 53: 2823
20. Anglada JM, Bofill JM (1994) FCI program copyright of CID-CSIC and Universitat de Barcelona