A simple iterative algorithm for generating selected eigenspaces of large matrices

2002 J. Phys. A: Math. Gen. 35 L61

(http://iopscience.iop.org/0305-4470/35/5/101)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **35** (2002) L61–L66 PII: S0305-4470(02)28830-7

LETTER TO THE EDITOR

A simple iterative algorithm for generating selected eigenspaces of large matrices

F Andreozzi, A Porrino and N Lo Iudice

Dipartimento di Scienze Fisiche, Università di Napoli Federico II, and Istituto Nazionale di Fisica Nucleare, Complesso Universitario di Monte S. Angelo, Via Cintia, 80126 Napoli, Italy

E-mail: loiudice@na.infn.it

Received 17 September 2001
Published 25 January 2002
Online at stacks.iop.org/JPhysA/35/L61

Abstract

We propose a new iterative algorithm for generating a subset of eigenvalues and eigenvectors of large matrices which generalizes the method of optimal relaxations. We also give convergence criteria for the iterative process, and investigate its efficiency by evaluating computer storage and time requirements and by a few numerical tests.

PACS numbers: 02.70.-c 21.60.-n 71.10.-w

The increase in computational power has stimulated a growing interest toward developing and refining methods which allow one to determine selected eigenvalues of a complex quantum system with extreme accuracy. Widely adopted, especially for computing ground state properties, are the quantum Monte Carlo methods [1], where a properly defined function of the Hamiltonian is used as a stochastic matrix which guides a Markov process to sample the basis.

Alternatively, one may resort to direct diagonalization methods, like the Lanczos [2] and Davidson [3] algorithms, widely used in several branches of physics. The critical points of direct diagonalization methods are the amount of memory needed and the time spent in the diagonalization process. Because of these limitations, several systems are still out of reach even with the computing power now available.

In this letter we present an iterative method, which is extremely easy to implement, for generating a subset of eigenvectors of a large matrix, give convergence criteria and show that it represents a generalization of the method of optimal relaxations [4].

We assume first that the matrix A represents a self-adjoint operator $\hat{A}$ in an orthonormal basis $\{|1\rangle, |2\rangle, \ldots, |N\rangle\}$ and is symmetric ($a_{ij} = \langle i|\hat{A}|j\rangle = a_{ji}$). For the sake of simplicity, we illustrate the procedure for a one-dimensional eigenspace. The algorithm consists of a first approximation loop and subsequent iterations of refinement loops. The first loop goes through the following steps:

(1a) Start with the first two basis vectors and diagonalize the matrix $\begin{pmatrix} \lambda_1^{(1)} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}$ where $\lambda_1^{(1)} = a_{11}$.

(1b) Select the eigenvalue $\lambda_2^{(1)}$ and corresponding eigenvector $|\phi_2^{(1)}\rangle = K_{2,1}^{(1)}|\phi_1^{(1)}\rangle + K_{2,2}^{(1)}|2\rangle$ where $|\phi_1^{(1)}\rangle \equiv |1\rangle$.

for $j = 3, \ldots, N$

(1c) compute $b_j^{(1)} = \langle\phi_{j-1}^{(1)}|\hat{A}|j\rangle$.

(1d) Diagonalize the matrix $\begin{pmatrix} \lambda_{j-1}^{(1)} & b_j^{(1)} \\ b_j^{(1)} & a_{jj} \end{pmatrix}$.

(1e) Select the eigenvalue $\lambda_j^{(1)}$ and the corresponding eigenvector $|\phi_j^{(1)}\rangle$.

end j

The first loop yields an approximate eigenvalue $\lambda_N^{(1)} \equiv E^{(1)} \equiv \lambda_0^{(2)}$ and an approximate eigenvector $|\psi^{(1)}\rangle \equiv |\phi_N^{(1)}\rangle \equiv |\phi_0^{(2)}\rangle = \sum_{i=1}^{N} K_{N,i}^{(1)}|i\rangle$. With these new entries we start an iterative procedure which goes through the following refinement loops:

for $n = 2, 3, \ldots,$ till convergence

for $j = 1, 2, \ldots, N$

(2a) Compute $b_j^{(n)} = \langle\phi_{j-1}^{(n)}|\hat{A}|j\rangle$.

(2b) Solve the eigenvalue problem in the general form

$$\det\left[\begin{pmatrix} \lambda_{j-1}^{(n)} & b_j^{(n)} \\ b_j^{(n)} & a_{jj} \end{pmatrix} - \lambda \begin{pmatrix} 1 & K_{j-1,j}^{(n)} \\ K_{j-1,j}^{(n)} & 1 \end{pmatrix}\right] = 0$$

where the appearance of the metric matrix follows from the non-orthogonality of the redefined basis $|\phi_{j-1}^{n}\rangle$ and $|j\rangle$.

(2c) Select the eigenvalue $\lambda_j^{(n)}$ and the corresponding eigenvector $|\phi_j^{(n)}\rangle$.

end j

end n.

The nth loop yields an approximate eigenvalue $\lambda_N^{(n)} \equiv E^{(n)} \equiv \lambda_0^{(n+1)}$. As for the eigenvector, at any step of the j-loop, we have

$$|\phi_j^{(n)}\rangle = p_j^{(n)}\,|\phi_{j-1}^{(n)}\rangle + q_j^{(n)}\,|j\rangle \tag{1}$$

with the appropriate normalization condition $[p_j^{(n)}]^2 + [q_j^{(n)}]^2 + 2\,p_j^{(n)}q_j^{(n)}K_{j-1,j}^{(n)} = 1$. The iteration of equation (1) yields the nth eigenvector

$$|\psi^{(n)}\rangle \equiv |\phi_N^{(n)}\rangle = P_0^{(n)}\,|\psi^{(n-1)}\rangle + \sum_{i=1}^{N} P_i^{(n)}\,q_i^{(n)}\,|i\rangle \tag{2}$$

where the numbers $P_i^{(n)}$ are

$$\begin{aligned} P_i^{(n)} &= \prod_{k=i+1}^{N} p_k^{(n)} \quad (i = 0, 1, \ldots, N-1) \\ P_N^{(n)} &= 1. \end{aligned} \tag{3}$$

The algorithm defines therefore the sequence of vectors (2), whose convergence properties we

can now examine. The $q_j^{(n)}$ and $p_j^{(n)}$ coefficients can be expressed as

$$q_j^{(n)} = \frac{|B_j^{(n)}|}{\left[(a_{jj}K_{j-1,j}^{(n)} - b_j^{(n)})^2 \; + \; 2K_{j-1,j}^{(n)} \; (a_{jj}K_{j-1,j}^{(n)} - b_j^{(n)})B_j^{(n)} \; + \; (B_j^{(n)})^2\right]^{1/2}}$$
$$p_j^{(n)} = (a_{jj}K_{j-1,j}^{(n)} - b_j^{(n)}) \; \frac{q_j^{(n)}}{B_j^{(n)}} \tag{4}$$

where

$$B_j^{(n)} = \left[\lambda_{j-1}^{(n)} - \lambda_j^{(n)}\right] \; - \; K_{j-1,j}^{(n)}\left[(a_{jj} - \lambda_j^{(n)})(\lambda_{j-1}^{(n)} - \lambda_j^{(n)})\right]^{1/2}. \tag{5}$$

It is apparent from these relations that, if

$$|\lambda_{j-1}^{(n)} - \lambda_j^{(n)}| \; \to \; 0 \qquad \forall j \tag{6}$$

the sequence $|\psi^{(n)}\rangle$ has a limit $|\psi\rangle$, which is an eigenvector of the matrix A. In fact, defining the residual vectors

$$|r^{(n)}\rangle = (\hat{A} - E^{(n)}) \; |\psi^{(n)}\rangle \tag{7}$$

a direct computation gives for their components

$$r_l^{(n)} = p_N^{(n)}\left[(a_{ll} - \lambda_l^{(n)})(\lambda_{l-1}^{(n)} - \lambda_l^{(n)})\right]^{1/2} + q_N^{(n)} \; \left\{a_{lN} - \lambda_N^{(n)}\delta_{lN}\right\}$$
$$- p_N^{(n)}\left\{\left(\lambda_{l-1}^{(n)} - \lambda_l^{(n)}\right) \; K_{l,l-1}^{(n)} + \left(\lambda_{N-1}^{(n)} - \lambda_N^{(n)}\right) \; K_{l,N-1}^{(n)}\right\}. \tag{8}$$

By virtue of (6), the norm of the nth residual vector converges to zero, namely $||r^{(n)}|| \to 0$. Equation (6) gives therefore a necessary condition for the convergence of $|\psi^{(n)}\rangle$ to an eigenvector $|\psi\rangle$ of A, with a corresponding eigenvalue $E = \lim E^{(n)}$. This condition holds independently of the prescription adopted for selecting the eigensolution. Indeed, we never had to specify the selection rule in steps (1b), (1e) and (2c). Equation (6) is not only a necessary but also a sufficient condition for the convergence to the lowest or the highest eigenvalue of A. In fact, the sequence $\lambda_j^{(n)}$ is monotonic (decreasing or increasing, respectively), bounded from below or from above by the trace and therefore convergent.

The algorithm we have just outlined has a variational foundation. Its variational counterpart is just the method of optimal relaxation [4]. Indeed, for the $p_j^{(n)}$ and $q_j^{(n)}$ given by equations (4), the $\alpha_j^{(n)}(= q_j^{(n)}/p_j^{(n)})$ derivative of the Rayleigh quotient

$$\rho(\phi_j^{(n)}) \; = \; \frac{\langle\phi_j^{(n)}|\hat{A}|\phi_j^{(n)}\rangle}{\langle\phi_j^{(n)}|\phi_j^{(n)}\rangle} \tag{9}$$

vanishes identically.

On the other hand, the present matrix formulation allows in a straightforward way for the optimal relaxation of an arbitrary number t of coordinates, thereby improving the convergence rate of the procedure. We only need to turn the two-dimensional into a $(t+1)$-dimensional eigenvalue problem in steps (1d) and (2b), compute t elements b_j in steps (1c) and (2a), and accordingly redefine the j-loops. The current eigenvector is still defined by the iterative relation ($\alpha_{kj}^{(n)} = q_k^{(n)}/p_j^{(n)}$)

$$|\phi_{j+t}^{(n)}\rangle \; = \; p_j^{(n)}\left(\; |\phi_j^{(n)}\rangle \; + \sum_{k=j+1}^{j+t} \; \alpha_{kj}^{(n)} \; |k\rangle\right) \tag{10}$$

which automatically fulfils the extremal conditions

$$\frac{\partial}{\partial \alpha_{kj}^{(n)}} \rho(\phi_{j+t}^{(n)}) = 0 \qquad k = j+1, \ldots, j+t. \tag{11}$$

Moreover, the algorithm can be naturally extended to generate at once an arbitrary number m of lowest eigenstates. We have simply to replace the two-dimensional matrices with multidimensional ones having the following block structure: A $m \times m$ submatrix diagonal in the selected m eigenvalues, which replaces $\lambda_{j-1}^{(n)}$, a $m' \times m'$ submatrix corresponding to a_{jj} and two $m \times m'$ off-diagonal blocks replacing $b_j^{(n)}$ or $K_{j-1,j}^{(n)}$. This new formulation amounts to an optimal relaxation method of several coordinates into a multidimensional subspace. It avoids therefore the use of deflation or shift techniques for the computation of higher eigenvalues and eigenvectors.

It remains now to investigate the practical feasibility of the method. The main issues to be faced are the storage and time requirements. In the one-dimensional case, we need to store a single N-dimensional vector (the eigenvector). The time is mainly determined by the j loop. This requires N operations for implementing point (2a) plus $k \simeq 15$ remaining operations. Since $n = 1, 2, \ldots, n_c$ and $j = 1, 2, \ldots, N$, the algorithm requires altogether $n_c(N^2 + kN)$ operations. It follows that, for large dimensional matrices, the number of operations grows like N^2. For sparse matrices with an average number L of non-zero matrix elements, the required number of operations is $n_c(L + k)N$ and therefore grows linearly with N. In the multidimensional case we need to store m N-dimensional vectors. If necessary, however, we can keep only one at a time and store the remaining $m - 1$ vectors in a secondary storage. This latter feature clearly shows that the algorithm lends itself to a straightforward parallelization. Also in the multidimensional case, the number of operations grows as $n_c m N^2$.

The algorithm has other remarkable properties: (i) It works perfectly even in the case of degeneracy of the eigenvalues. (ii) The diagonalization of the submatrices of order $m + m'$ ensures the orthogonalization of the full N-dimensional eigenvectors at each step. Therefore, no *ghost* eigenvalues occur. (iii) The range of validity of the algorithm can be easily enlarged if we remove some of the initial assumptions. Clearly, the iterative procedure applies to a non-orthogonal basis. We simply need to substitute steps (1a) and (1d) of the first loop with the appropriate generalized eigenvalue problem. It applies also to non-symmetric matrices. We have only to update both right and left eigenvectors and perform steps (1c) and (2a) for both non-diagonal matrix elements.

In order to test the efficiency and the convergence rate of the iterative procedure, we have applied the method to several examples. The first is a five-point finite difference matrix arising from the two-dimensional Laplace equation [5]. This is a block-tridiagonal matrix of n_b b-dimensional blocks, whose eigenvalues are

$$\lambda_{ij} = 4\left(\sin^2 \frac{i\pi}{2(n_b+1)} + \sin^2 \frac{j\pi}{2(b+1)}\right) \tag{12}$$

where $i = 1, 2, \ldots, n_b$ and $j = 1, 2, \ldots, b$. As in [5], we considered a block matrix with $n_b = 15$ and $b = 20$. We have tested the one-dimensional as well as the multidimensional version of the algorithm. As shown in figure 1, the iterative procedure converges much faster in the multidimensional case. In fact, the convergence rate increases with the number ν of generated eigenvalues and is considerably faster than in Lanczos. It is also to be stressed that our algorithm allows for an arbitrarily high accuracy, up to the machine precision limit. The method, especially in its multidimensional extension, is quite effective even if applied to the same matrix with $n_b = b$ so as to allow for degeneracy. For $n_b = b = 80$, it yields the lowest seven roots, including two couples of degenerate eigenvalues, with an accuracy of 10^{-12}.

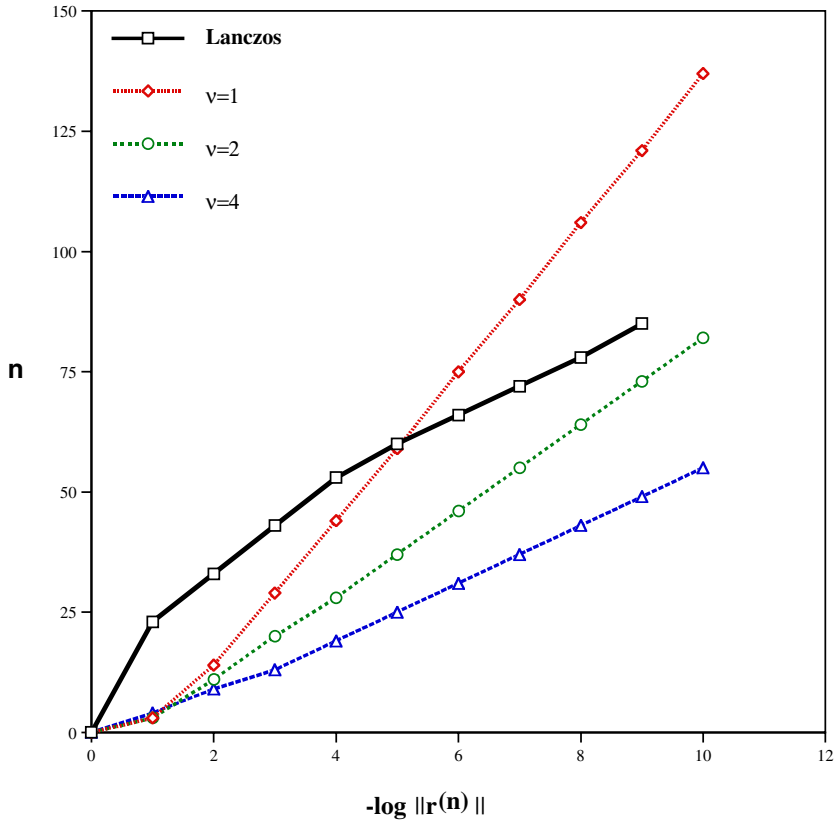

Figure 1. Convergence rate of the present algorithm applied to a finite difference matrix deduced from Laplace equation [5] for different numbers ν of generated eigenvalues. The data referring to the Lanczos convergence rate are taken from [5].

A second example, still taken from [5], is a one-dimensional biharmonic band matrix whose eigenvalues

$$\lambda_k = 16 \sin^4 \frac{k\pi}{2(N+1)} \qquad k = 1, \ldots, N \tag{13}$$

are small and poorly separated from each other. A similarly high density of levels occurs in the Anderson model of localization [6]. Because of this peculiarity, the limit of the machine precision is reached for a modest increase of the dimension N of the matrix. Our method reproduces perfectly any number of eigenvalues with the required accuracy after a small number of iterations. In the specific example discussed in [5] ($N = 20$) we attained the highest accuracy after eight iterations, much less than all methods discussed there. We have checked that, unlike others, our method works without any modification even if we increase the dimension N up to the limit compatible with the machine precision. In this case the number of iterations needed increase by an order of magnitude, in any case, below 100.

A third example is provided by a matrix with diagonal matrix elements $a_{ii} = 2\sqrt{i} - a$ and off-diagonal ones $a_{ij} = -a$ or $a_{ij} = 0$ according to whether they fall within or outside a band of width $2L$. Such a matrix simulates a pairing Hamiltonian relevant to many branches of physics. We have considered a matrix of dimension $N = 10^8$ and half-band width $L = 400$. We found convergence after 28 iterations, reaching an accuracy of 10^{-8} for the eigenvalues.

The time required to compute the lowest eigenvalue through the one-dimensional algorithm is $t = 18\,697$ s for a workstation of 500 MHz and 512 Mb of RAM.

Finally, we generalize the latter example by considering a full matrix of dimension $N = 10^5$ with matrix elements $a_{ij} = 2\sqrt{i}\delta_{ij} + (-1)^{i+j} a \frac{i+j}{\sqrt{i^2+j^2}}$. Their alternating signs are also to be noticed, since they decrease somewhat the rate of convergence of the process. We reproduce the lowest eigenvalue with an accuracy of 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8} after $n_c = 42, 70, 155, 330$ iterations, respectively.

In conclusion, the present diagonalization algorithm is a generalization of the variational optimal relaxation method and, on the basis of the examples discussed, appears to be more competitive than the methods currently adopted. It seems to be faster and to require a minimal amount of computer storage. It is extremely simple to implement and is *robust*, yielding always stable numerical solutions. Moreover, it is free of *ghost* eigenvalues. Because of these features, we are confident that it can be applied quite effectively to physical systems, like medium-light nuclei or quantum dots with few electrons.

The work was partly supported by the Prin 99 of the Italian MURST

References

[1] A comprehensive account of the existing techniques can be found in
Nightingale M P and Umrigar C J 1999 *Quantum Monte Carlo Methods in Physics and Chemistry* (Dordrecht: Kluwer Academic)
[2] See, for instance,
Golub G H and Van Loan C F 1996 *Matrix Computations* (Baltimore, MD: Johns Hopkins University Press)
[3] Davidson E R 1975 *J. Comput. Phys.* **17** 87
[4] Shavitt I, Bender C F, Pipano A, and Hosteny R P 1973 *J. Comput. Phys.* **11** 90
[5] Ruhe A 1977 *Lecture Notes in Mathematics* **527** ed A Dold and B Eckmann (Berlin: Springer) p 130
[6] Elsner U, Mehrmann V, Milde F, Römer R A and Schreiber M 1999 *SIAM J. Sci. Comput.* **20** 2089