Journal of Sound and Vibration (2001) **243**(5), 899–908
doi:10.1006/jsvi.2000.3446, available online at http://www.idealibrary.com on IDEAL®

A CLUSTER SERIES EXPANSION TECHNIQUE FOR SOLVING VIBRATION PROBLEMS

P. Žitňan

Department of Applied Mathematics, Faculty of Sciences, University of Žilina, Hurbanova 15, 01026 Žilina, Slovakia and
Computing Centre, Slovak Academy of Sciences, Dúbravská cesta 9, 842 35 Bratislava, Slovakia

(*Received* 26 *May* 1999, *and in final form* 11 *September* 2000)

A cluster series expansion technique for spectral solution of vibration problems is presented. In this technique the eigenfunction approximations are composed of clusters of single basis functions instead of the classical Fourier series of single basis functions. To realize the cluster expansion an alternating fixed-free subspace strategy is used to improve the coefficients of the single basis functions inside the clusters. This strategy generates a family of m-step iterative algorithms solving large dense matrix eigenproblem of order $m.n$ as a few dense matrix eigenproblems of order $n + (m-1)\ell$, where ℓ denotes the number of computed eigenvalues. The computational behaviour of the proposed m-step algorithms is investigated by solving matrix eigenproblems up to the order 3600 by using the m-step algorithms for $m = 2, 3, 4, 5$.

© 2001 Academic Press

1. INTRODUCTION

The methods for solving matrix eigenproblems are very important tools of research in mechanical engineering, quantum chemistry and many other scientific disciplines [1]. Starting in 1846, when Jacobi wrote his famous paper on solving the symmetric eigenvalue problem $Au = \lambda u$, the number of new methods and various improvements of the known methods has grown rapidly [1–4]. At present the scientific community has a great choice of methods for solving the majority of their matrix eigenproblems together with effective and reliable software [5–7] freely available on the Internet (http://www.netlib.org). The solution of the remaining eigenvalue problems which are very large ($>10^7$) and sparse, large ($>10^5$) and dense or non-symmetric may be very difficult or impossible. This is the reason why new methods and improvements of the known ones are of permanent interest for the computational scientists.

The aim of this article is to present a new iterative algorithm for solving large dense matrix eigenproblems resulting from the variational solution of the vibration problems using spectral basis functions. As presented in our computational experiments, the proposed algorithm is able to save great amount of main computer memory and CPU time as well.

2. OUTLINE OF THE METHOD

In the variational solution of a vibration problem

$$\mathscr{A}\phi = \lambda\phi, \tag{1}$$

one seeks minima of the corresponding Rayleigh quotient

$$(\mathscr{A}\phi, \phi)/(\phi, \phi) \tag{2}$$

over a finite-dimensional subspace of the space of square-integrable functions $L_2(\Omega)$ [8]. Denoting φ_i, $i = 1, 2, \dots,$ a basis system of $L_2(\Omega)$, the desired functions $\phi_{k,n}$ minimizing expression (2) over the n-dimensional subspace generated by the basis functions $\varphi_1, \varphi_2, \dots, \varphi_n$ can be written in the form

$$\phi_k \approx \phi_{k,n} = \sum_{i=1}^{n} c_i^k \varphi_i. \tag{3}$$

Here the eigenvectors $\mathbf{c}^k$ and the corresponding eigenvalues $\lambda_{k,n}$, which are the upper bounds of the exact eigenvalues λ_k, solve the generalized matrix eigenproblem

$$\mathbf{Ac} = \lambda \mathbf{Bc}, \tag{4}$$

where $a_{ij} = (\mathscr{A}\varphi_i, \varphi_j)$, and $b_{ij} = (\varphi_i, \varphi_j)$.

The usual procedure for improving the approximations $\phi_{k,n}$ and $\lambda_{k,n}$ is to use more basis functions and, consequently, to solve larger matrix eigenproblems (4). The important question now is—*is it possible to use more basis functions without solving larger matrix eigenvalue problems?* To answer this question one can try to approximate the eigenfunctions ϕ_k of equation (1) by clusters of basis functions instead of the single basis functions; i.e., instead of the truncated Fourier series (3) one can try to expand ϕ_k in the series of clusters $\mathbf{\Phi}_i^k$ as

$$\phi_k \approx \sum_{i=1}^{m} \mathbf{C}_i^k \mathbf{\Phi}_i^k, \quad k = 1, 2, \dots, \ell, \tag{5}$$

where each cluster $\mathbf{\Phi}_i^k$ is linear combination of n basis functions $\varphi_{(i-1)*n+j}$ for $j = 1, 2, \dots, n$.

To realize this idea it is necessary to improve the coefficients of the single basis functions φ_j inside the clusters $\mathbf{\Phi}_i^k$ alternately. For this reason one can use, say, an alternating fixed-free subspace strategy, which is certainly used in practice in different forms and in different situations as for example in reference [9], where separable nonlinear least squares problems are solved by alternating improvement of partitioned variables.

2.1. TWO-STEP ALGORITHM

Assume that $\varphi_i, i = 1, 2, \dots,$ are orthonormal basis functions and the clusters $\mathbf{\Phi}_1^k$ are the variational approximations of the first ℓ eigenfunctions of equation (1) achieved by minimizing expression (2), by using n basis functions φ_i: i.e.,

$$\mathbf{\Phi}_1^k = \sum_{i=1}^{n} \alpha_{1,i}^k \varphi_i, \quad k = 1, 2, \dots, \ell. \tag{6}$$

Starting with these initial approximations the following two-step algorithm describes a simple way of how to improve the single clusters $\mathbf{\Phi}_1^k$ and $\mathbf{\Phi}_2^k$ with essential saving of main computer memory.

First step. Consider the set of trial functions

$$\mathbf{\Phi}_1^1,\ \mathbf{\Phi}_1^2, \ldots, \mathbf{\Phi}_1^\ell, \varphi_{n+1}, \varphi_{n+2}, \ldots, \varphi_{2n}. \tag{7}$$

The minimization of expression (2) by using these trial functions leads to the solution of the partial matrix eigenproblem

$$\begin{pmatrix} (\mathscr{A}\mathbf{\Phi}_1^i,\ \mathbf{\Phi}_1^j) & (\mathscr{A}\mathbf{\Phi}_1^i, \varphi_{n+s}) \\ (\mathscr{A}\varphi_{n+r},\ \mathbf{\Phi}_1^j) & (\mathscr{A}\varphi_{n+r},\ \varphi_{n+s}) \end{pmatrix} \mathbf{c} = \lambda \begin{pmatrix} (\mathbf{\Phi}_1^i,\ \mathbf{\Phi}_1^j) & 0 \\ 0 & \mathrm{E} \end{pmatrix} \mathbf{c}, \tag{8}$$

where E is an $n \times n$ identity matrix. In this step as well as in the second one the indices i, j, r, s range as follows: $1 \leqslant i, j \leqslant \ell$, $1 \leqslant r, s \leqslant n$; and the whole matrices are of order $\ell + n$. The eigenfunction approximations $\phi_{k,2n}$ resulting from equation (8) are of the form

$$\phi_{k,2n} = \sum_{i=1}^{\ell} c_i^k \mathbf{\Phi}_1^i + \sum_{i=1}^{n} c_{\ell+i}^k \varphi_{n+i}, \qquad k = 1, 2, \ldots, \ell \tag{9}$$

and the improved clusters $\mathbf{\Phi}_2^k$ may be defined as $\mathbf{\Phi}_2^k = \sum_{i=1}^n c_{\ell+i}^k \varphi_{n+i}$. If the algorithm iterates (one iteration is composed by the solution of the eigenvalue problems (8) and (11)), then $c_i^k \to 0$ for $i \neq k$ and $i \leqslant \ell$ and, consequently, $\phi_{k,2n} \to c_k^k \mathbf{\Phi}_1^k + \mathbf{\Phi}_2^k$. This is expected behaviour, because the clusters $\mathbf{\Phi}_1^i$ are composed of the same basis functions $\varphi_1, \varphi_2, \ldots, \varphi_n$ and only one of them, $\mathbf{\Phi}_1^k$, may participate in the final cluster expansion (5) of ϕ_k. Consequently, the cluster coefficients $\mathbf{C}_i^k$ in equation (5) may be determined as $\mathbf{C}_1^1 = c_1^1$, $\mathbf{C}_1^2 = c_2^2, \ldots, \mathbf{C}_1^\ell = c_\ell^\ell$, and $\mathbf{C}_2^k = 1$ for $k = 1, 2, \ldots, \ell$. The same situation happens also in equations (12) to follow and will not be commented on there.

Second step. Consider the set of trial functions

$$\mathbf{\Phi}_2^1, \mathbf{\Phi}_2^2, \ldots, \mathbf{\Phi}_2^\ell, \varphi_1,\ \varphi_2, \ldots, \varphi_n, \tag{10}$$

where $\mathbf{\Phi}_2^k$ are defined in equation (9). The minimization of expression (2) using these trial functions leads to the solution of the partial matrix eigenproblem

$$\begin{pmatrix} (\mathscr{A}\mathbf{\Phi}_2^i,\ \mathbf{\Phi}_2^j) & (\mathscr{A}\mathbf{\Phi}_2^i, \varphi_s) \\ (\mathscr{A}\varphi_r,\ \mathbf{\Phi}_2^j) & (\mathscr{A}\varphi_r,\ \varphi_s) \end{pmatrix} \mathbf{c} = \lambda \begin{pmatrix} (\mathbf{\Phi}_2^i,\ \mathbf{\Phi}_2^j) & 0 \\ 0 & \mathrm{E} \end{pmatrix} \mathbf{c}. \tag{11}$$

The eigenfunction approximations $\phi_{k,2n}$ resulting from equation (11) are of the form

$$\phi_{k,2n} = \sum_{i=1}^{\ell} c_i^k \mathbf{\Phi}_2^i + \sum_{i=1}^{n} c_{\ell+i}^k \varphi_i \qquad k = 1, 2, \ldots, \ell \tag{12}$$

and the improved clusters $\mathbf{\Phi}_1^k$ may be defined as $\mathbf{\Phi}_1^k = \sum_{i=1}^n c_{\ell+i}^k \varphi_i$. If the accuracy of $\lambda_{k,2n}$ is not sufficient, go back to the first step.

While the usual procedure for computing $\phi_{k,2n}$ consists of the minimization of expression (2) *at once* over the whole $2n$-dimensional subspace generated by $\varphi_1, \varphi_2, \ldots, \varphi_{2n}$, the presented alternating fixed-free subspace strategy enables one to improve $\mathbf{\Phi}_1^k$ and $\mathbf{\Phi}_2^k$ *alternately*. Firstly, $\mathbf{\Phi}_2^k$ are improved over the n-dimensional subspace generated by $\varphi_{n+1}, \varphi_{n+2}, \ldots, \varphi_{2n}$ with respect to $\mathbf{\Phi}_1^k$ (in this step $\varphi_1, \varphi_2, \ldots, \varphi_n$ are fixed in $\mathbf{\Phi}_1^k$) improved in the second step of the previous iteration. Secondly, $\mathbf{\Phi}_1^k$ are improved over the n-dimensional subspace generated by $\varphi_1, \varphi_2, \ldots, \varphi_n$, with respect to $\mathbf{\Phi}_2^k$ (in this step $\varphi_{n+1}, \varphi_{n+2}, \ldots, \varphi_{2n}$, are fixed in $\mathbf{\Phi}_2^k$) improved in the first step of the previous iteration. This algorithm alternately improving $\mathbf{\Phi}_1^k$ and $\mathbf{\Phi}_2^k$ is then repeated until convergence.

In principle the algorithm can be generalized also for the case when the used basis system is split into any "reasonable" number of subsystems.

2.2. THREE-STEP ALGORITHM

If the basis system is split into three subsystems we can derive the three-step algorithm. As the initial approximations the clusters $\mathbf{\Phi}_1^k$ and $\mathbf{\Phi}_2^k$ computed by the two-step algorithm using $2n$ basis functions $\varphi_1, \varphi_2, \dots, \varphi_{2n}$ can be used.

First step. Consider the set of trial functions

$$\mathbf{\Phi}_1^1, \mathbf{\Phi}_1^2, \dots, \mathbf{\Phi}_1^\ell, \mathbf{\Phi}_2^1, \mathbf{\Phi}_2^2, \dots, \mathbf{\Phi}_2^\ell, \varphi_{2n+1}, \varphi_{2n+2}, \dots, \varphi_{3n}, \tag{13}$$

where the clusters $\mathbf{\Phi}_1^k$ and $\mathbf{\Phi}_2^k$ are improved in the second and the third step of the previous iteration, respectively. Similarly as in the two-step algorithm the minimization of expression (2) using the trial functions (13) leads to the solution of the partial matrix eigenproblem

$$\begin{pmatrix} (\mathscr{A}\mathbf{\Phi}_1^i, \mathbf{\Phi}_1^j) & (\mathscr{A}\mathbf{\Phi}_1^i, \mathbf{\Phi}_2^j) & (\mathscr{A}\mathbf{\Phi}_1^i, \varphi_{2n+s}) \\ (\mathscr{A}\mathbf{\Phi}_2^i, \mathbf{\Phi}_1^j) & (\mathscr{A}\mathbf{\Phi}_2^i, \mathbf{\Phi}_2^j) & (\mathscr{A}\mathbf{\Phi}_2^i, \varphi_{2n+s}) \\ (\mathscr{A}\varphi_{2n+r}, \mathbf{\Phi}_1^j) & (\mathscr{A}\varphi_{2n+r}, \mathbf{\Phi}_2^j) & (\mathscr{A}\varphi_{2n+r}, \varphi_{2n+s}) \end{pmatrix} \mathbf{c} = \lambda \begin{pmatrix} (\mathbf{\Phi}_1^i, \mathbf{\Phi}_1^j) & 0 & 0 \\ 0 & (\mathbf{\Phi}_2^i, \mathbf{\Phi}_2^j) & 0 \\ 0 & 0 & \mathrm{E} \end{pmatrix} \mathbf{c}. \tag{14}$$

In this as well as in the following two steps the submatrix indices i, j, r, s range as follows: $1 \leqslant i, j \leqslant \ell$, $1 \leqslant r, s \leqslant n$; and the whole matrices are of order $2\ell + n$. The eigenfunction approximations $\phi_{k,3n}$ resulting from equation (14) are of the form

$$\phi_{k,3n} = \sum_{i=1}^{\ell} c_i^k \mathbf{\Phi}_1^i + \sum_{i=1}^{\ell} c_{\ell+i}^k \mathbf{\Phi}_2^i + \sum_{i=1}^{n} c_{2\ell+i}^k \varphi_{2n+i}, \qquad k = 1, 2, \dots, \ell \tag{15}$$

and the improved clusters $\mathbf{\Phi}_3^k$ may be defined as $\mathbf{\Phi}_3^k = \sum_{i=1}^n c_{2\ell+i}^k \varphi_{2n+i}$. If the algorithm iterates (one iteration is composed by the solution of the eigenvalue problems (14), (17), and (20)), similarly as in equations (9) and (12), $c_i^k \to 0$ and $c_{\ell+i}^k \to 0$ for $i \neq k$ and $i \leqslant \ell$. This convergence immediately provides the cluster expansion coefficients $\mathbf{C}_i^k$ in equation (5) of the form $\mathbf{C}_1^k = c_k^k$, $\mathbf{C}_2^k = c_{\ell+k}^k$, and $\mathbf{C}_3^k = 1$ for $k = 1, 2, \dots, \ell$. The same situation happens also in equations (18) and (21) and will not be commented on there.

Second step. Consider the set of trial functions

$$\mathbf{\Phi}_2^1, \mathbf{\Phi}_2^2, \dots, \mathbf{\Phi}_2^\ell, \mathbf{\Phi}_3^1, \mathbf{\Phi}_3^2, \dots, \mathbf{\Phi}_3^\ell, \varphi_1, \varphi_2, \dots, \varphi_n, \tag{16}$$

where $\mathbf{\Phi}_3^k$ are defined in equation (15). The minimization of expression (2) using these trial functions leads to the solution of the partial matrix eigenproblem

$$\begin{pmatrix} (\mathscr{A}\mathbf{\Phi}_2^i, \mathbf{\Phi}_2^j) & (\mathscr{A}\mathbf{\Phi}_2^i, \mathbf{\Phi}_3^j) & (\mathscr{A}\mathbf{\Phi}_2^i, \varphi_s) \\ (\mathscr{A}\mathbf{\Phi}_3^i, \mathbf{\Phi}_2^j) & (\mathscr{A}\mathbf{\Phi}_3^i, \mathbf{\Phi}_3^j) & (\mathscr{A}\mathbf{\Phi}_3^i, \varphi_s) \\ (\mathscr{A}\varphi_r, \mathbf{\Phi}_2^j) & (\mathscr{A}\varphi_r, \mathbf{\Phi}_3^j) & (\mathscr{A}\varphi_r, \varphi_s) \end{pmatrix} \mathbf{c} = \lambda \begin{pmatrix} (\mathbf{\Phi}_2^i, \mathbf{\Phi}_2^j) & 0 & 0 \\ 0 & (\mathbf{\Phi}_3^i, \mathbf{\Phi}_3^j) & 0 \\ 0 & 0 & \mathrm{E} \end{pmatrix} \mathbf{c}. \tag{17}$$

The eigenfunction approximations $\phi_{k,3n}$ resulting from equation (17) are of the form

$$\phi_{k,3n} = \sum_{i=1}^{\ell} c_i^k \mathbf{\Phi}_2^i + \sum_{i=1}^{\ell} c_{\ell+i}^k \mathbf{\Phi}_3^i + \sum_{i=1}^{n} c_{2\ell+i}^k \varphi_i, \qquad k = 1,2,\dots,\ell \tag{18}$$

and the improved clusters $\mathbf{\Phi}_1^k$ may be defined as $\mathbf{\Phi}_1^k = \sum_{i=1}^n c_{2\ell+i}^k \varphi_i$.

Third step. Consider the set of trial functions

$$\mathbf{\Phi}_1^1, \mathbf{\Phi}_1^2, \dots, \mathbf{\Phi}_1^\ell, \mathbf{\Phi}_3^1, \mathbf{\Phi}_3^2, \dots, \mathbf{\Phi}_3^\ell, \varphi_{n+1}, \varphi_{n+2}, \dots, \varphi_{2n}, \tag{19}$$

where $\mathbf{\Phi}_1^k$ are defined in equation (18). The minimization of expression (2) using these trial functions leads to the solution of the partial matrix eigenproblem

$$\begin{pmatrix} (\mathscr{A}\mathbf{\Phi}_1^i, \mathbf{\Phi}_1^j) & (\mathscr{A}\mathbf{\Phi}_1^i, \mathbf{\Phi}_3^j) & (\mathscr{A}\mathbf{\Phi}_1^i, \varphi_{n+s}) \\ (\mathscr{A}\mathbf{\Phi}_3^i, \mathbf{\Phi}_1^j) & (\mathscr{A}\mathbf{\Phi}_3^i, \mathbf{\Phi}_3^j) & (\mathscr{A}\mathbf{\Phi}_3^i, \varphi_{n+s}) \\ (\mathscr{A}\varphi_{n+r}, \mathbf{\Phi}_1^j) & (\mathscr{A}\varphi_{n+r}, \mathbf{\Phi}_3^j) & (\mathscr{A}\varphi_{n+r}, \varphi_{n+s}) \end{pmatrix} \mathbf{c} = \lambda \begin{pmatrix} (\mathbf{\Phi}_1^i, \mathbf{\Phi}_1^j) & 0 & 0 \\ 0 & (\mathbf{\Phi}_3^i, \mathbf{\Phi}_3^j) & 0 \\ 0 & 0 & \mathrm{E} \end{pmatrix} \mathbf{c}. \tag{20}$$

The eigenfunction approximations $\phi_{k,3n}$ resulting from equation (20) are of the form

$$\phi_{k,3n} = \sum_{i=1}^{\ell} c_i^k \mathbf{\Phi}_1^i + \sum_{i=1}^{\ell} c_{\ell+i}^k \mathbf{\Phi}_3^i + \sum_{i=1}^{n} c_{2\ell+i}^k \varphi_{n+i}, \qquad k = 1,2,\dots,\ell \tag{21}$$

and the improved clusters $\mathbf{\Phi}_2^k$ may be defined as $\mathbf{\Phi}_2^k = \sum_{i=1}^n c_{2\ell+i}^k \varphi_{n+i}$. If the accuracy of $\lambda_{k,3n}$ is not sufficient go back to the first step.

The four-step and five-step algorithms, also considered in our computational experiments, will start with the trial functions

$$\mathbf{\Phi}_1^1, \mathbf{\Phi}_1^2, \dots, \mathbf{\Phi}_1^\ell, \mathbf{\Phi}_2^1, \mathbf{\Phi}_2^2, \dots, \mathbf{\Phi}_2^\ell, \mathbf{\Phi}_3^1, \mathbf{\Phi}_3^2, \dots, \mathbf{\Phi}_3^\ell, \varphi_{3n+1}, \varphi_{3n+2}, \dots, \varphi_{4n} \tag{22}$$

and

$$\mathbf{\Phi}_1^1, \mathbf{\Phi}_1^2, \dots, \mathbf{\Phi}_1^\ell, \mathbf{\Phi}_2^1, \mathbf{\Phi}_2^2, \dots, \mathbf{\Phi}_2^\ell, \mathbf{\Phi}_3^1, \mathbf{\Phi}_3^2, \dots, \mathbf{\Phi}_3^\ell, \mathbf{\Phi}_4^1, \mathbf{\Phi}_4^2, \dots, \mathbf{\Phi}_4^\ell,$$
$$\varphi_{4n+1}, \varphi_{4n+2}, \dots, \varphi_{5n}, \tag{23}$$

respectively.

3. COMPUTATIONAL EXPERIMENTS

The behaviour of the presented algorithms has been investigated in the variational solution of the eigenvalue problem

$$-\Delta u + e^x e^y u = \lambda u \quad \text{on } \Omega = (0,\pi)\times(0,\pi), \tag{24}$$

subject to the homogeneous Dirichlet boundary condition $u = 0$ on $\partial\Omega$. While the methods and software for solving large sparse matrix eigenproblems are well developed, it is the intention of this study to concentrate upon the solution of dense problems. When using the sine basis functions and artificial coefficient $e^x e^y$ in equation (24), the resulting matrices generated by the scalar product $(\mathscr{A}\varphi_i, \varphi_j)$ are dense.

In the first four tables the experiments with the two-, three-, four-, and five-step algorithms are reported. These tables show how many iterations are needed in order to compute the first three eigenvalues of the corresponding matrices with the accuracy of at least 13 significant figures. In these tables only the number of iterations in the m-step algorithm (without computing initial cluster approximations) are shown. The initial cluster approximations $\mathbf{\Phi}_s^k(k = 1, 2, \dots, \ell;\ s = 1, 2, \dots, m-1)$ needed for the m-step algorithm have been computed iteratively starting from the two-step algorithm giving $\mathbf{\Phi}_s^k(k = 1, 2, 3;$ $s = 1, 2)$, then followed by the three-step algorithm giving $\mathbf{\Phi}_s^k(k = 1, 2, 3;\ s = 1, 2, 3)$, and finally by the four-step algorithm giving $\mathbf{\Phi}_s^k(k = 1, 2, 3;\ s = 1, 2, 3, 4)$. In these computations one iteration in each step was sufficient in order to give good initial cluster approximations for the m-step algorithm. There is also a possibility to obtain the initial cluster approximations directly by solving the first $m-1$ neighbouring main submatrix eigenproblems of order n resulting from the original matrix eigenproblem (4) of order $N = m.n$. However, as seen in Table 5 for the five-step algorithm, the convergence is slower than in the previous case.

Standard matrix eigenproblems have been solved by the subroutine NSHOUD (Householder-bisection-QR-inverse iteration) and generalized partial matrix eigenproblems

TABLE 1

Number of iterations of the two-step algorithm in computing the first three eigenvalues of equation (24) *using N basis functions*

$N = 2n$	120	240	420	900	1800	2400	3600
λ_1	7	5	4	3	3	2	2
λ_2	7	6	4	3	3	2	2
λ_3	8	6	5	3	3	3	2

TABLE 2

Number of iterations of the three-step algorithm in computing the first three eigenvalues of equation (24) *using N basis functions*

$N = 3n$	120	240	420	900	1800	2400	3600
λ_1	6	6	4	3	2	2	2
λ_2	10	8	5	4	3	3	3
λ_3	10	9	7	4	3	3	3

TABLE 3

Number of iterations of the four-step algorithm in computing the first three eigenvalues of equation (24) *using N basis functions*

$N = 4n$	120	240	420	900	1800	2400	3600
λ_1	7	7	5	3	2	2	2
λ_2	11	7	6	4	3	2	2
λ_3	12	7	7	5	3	2	2

TABLE 4

Number of iterations of the five-step algorithm in computing the first three eigenvalues of equation (24) *using N basis functions*

$N = 5n$	120	240	420	900	1800	2400	3600
λ_1	7	6	6	3	2	2	2
λ_2	10	9	7	4	2	2	2
λ_3	11	10	8	4	2	2	2

TABLE 5

Number of iterations of the five-step algorithm in computing the first three eigenvalues of equation (24) *using N basis functions* (*initial approximations are computed directly from diagonal submatrices*)

$N = 5n$	120	240	420	900	1800	2400	3600
λ_1	11	9	8	6	5	4	4
λ_2	13	11	10	7	5	5	4
λ_3	15	13	11	8	5	5	4

by the subroutine NGHOUD (simultaneous triangular decomposition + NSHOUD) from the package NICER [10]. To speed up the computations it would be probably suitable to transform the generalized partial matrix eigenproblems into the standard form. This is advantageous owing to block-diagonal structure of the right-hand side matrices B.

4. PERFORMANCE DISCUSSION

In this section a discussion is presented of the two most important practical aspects of the proposed cluster series expansion technique (CSET) as amount of the floating point operations and size of the used main computer memory. While the cost of the QR solution ($QRsol$) of one dense matrix eigenproblem of order $N = m.n$ scales as N^3, i.e., $QRsol = m^3.n^3$, the cost of a few iterations ($ITsol$) of the m-step algorithm, solving m partial matrix eigenproblems of order $n + (m-1)\ell$ in each iteration, scales as $ITsol = iter.m(n + (m-1)\ell)^3$. Here $iter$ is number of iterations of the used m-step algorithm and ℓ is number of the computed eigenvalues. Therefore, the theoretical performance acceleration parameter ACC_t of the m-step algorithm in comparison with an $O(n^3)$ QR solver may be estimated as

$$ACC_t = \frac{QRsol}{ITsol} = \frac{m^3.n^3}{iter.m(n + (m-1)\ell)^3} \approx \frac{m^2}{iter}, \tag{25}$$

where the assumption n is large enough implies $n + (m-1)\ell \approx n$. The dependence of ACC_t with respect to m and N, based on the values $iter$ reported in Tables 1–4, is shown in Table 6. As seen in this table ACC_t increases if both parameters m and N increases and, consequently, one can expect more efficient behaviour of the proposed algorithm for $m > 5$ and $N > 3600$. Because the values in Table 6 are only theoretical and based on high-quality initial cluster approximations computed iteratively using a few iterations in each previous m-step algorithm, it is of interest to investigate a more practical and simpler case, when the initial cluster approximations are computed directly. In Table 7 the actual CPU time (t_5) for the five-step algorithm is shown, for the NICER QR solution (t_{QR}) of the original problem (24) using N basis functions, the actual performance acceleration parameter $ACC_a = t_{QR}/t_5$ and the theoretical values ACC_t based on expression (25) and the number of iterations reported in Table 5 for λ_3. Note that the values of iterations reported in Table 5 correspond to the initial cluster approximations computed directly from four matrix eigenproblems (4) of order $n = N/5$ corresponding to the following four sets of basis functions—$\{\varphi_1, \varphi_2, \dots, \varphi_n\}$, $\{\varphi_{n+1}, \varphi_{n+2}, \dots, \varphi_{2n}\}$, $\{\varphi_{2n+1}, \varphi_{2n+2}, \dots, \varphi_{3n}\}$, and $\{\varphi_{3n+1}, \varphi_{3n+2}, \dots, \varphi_{4n}\}$ giving $\mathbf{\Phi}_1^i$, $\mathbf{\Phi}_2^i$, $\mathbf{\Phi}_3^i$ and $\mathbf{\Phi}_4^i$ ($i = 1, 2, 3$), respectively, for each N under consideration. These initial cluster approximations are more advantageous from a practical point of view than the ones computed iteratively, although, as seen in Tables 1–4, the initial cluster approximations computed iteratively are more accurate. The visible discrepancy between ACC_a and ACC_t in Table 7 is caused mainly by neglecting the CPU time needed for creating the partial matrix eigenproblems (e.g., equations (8) and (11) in the two-step algorithm). However, as shown in Table 8 the CPU time (t_c) needed for creating the corresponding matrix eigenproblems is not negligible in comparison with the CPU time (t_s) of their solution.

To save the CPU time in the solution of large matrix eigenproblems it is necessary to keep the whole matrix array in the main computer memory in order to avoid expensive data transfer between the main and secondary storage. Also the comparison of the main memory requirement between a QR solution ($QRmem$) and the iterative m-step algorithm ($ITmem$) is

TABLE 6

Theoretical performance acceleration of the m-step algorithms using iteratively computed initial cluster approximations in comparison with the QR ($O(n^3)$ operations) solution of matrix eigenproblems of order N

N	Two-step	Three-step	Four-step	Five-step
120	0·50	0·90	1·33	2·27
240	0·67	1·00	2·29	2·50
420	0·80	1·29	2·29	3·13
900	1·00	2·25	3·20	6·25
1800	1·33	3·00	5·33	12·50
3600	2·00	4·50	8·00	12·50

TABLE 7

The CPU time (t_{QR}) needed for the QR solution of equation (24), *the CPU time (t_5) needed for the five-step CSET solution of equation* (24) *using N basis functions, the actual performance acceleration $ACC_a = t_{QR}/t_5$ and the corresponding theoretical values $ACC_t = m^2/iter$ based on iter from Table* 5; *timings are given in seconds*

N	120	240	420	900	1800	2400	3600
t_{QR}	0·22	1·41	9·57	105·25	866·17	2068·71	7799·38
t_5	2·97	11·48	32·14	90·12	281·78	671·03	2202·41
ACC$_a$	0·07	0·12	0·30	1·17	3·07	3·08	3·54
ACC$_t$	1·67	1·92	2·27	3·13	5·00	5·00	6·25

TABLE 8

The CPU time (t_c) *needed for creating the partial eigenproblems of order* $N/5 + 12$ *corresponding to the five-step CSET solution of equation* (24) *using N basis functions and the CPU time* (t_s) *used for their solution*; *timings are given in seconds*

N	120	240	420	900	1800	2400	3600
t_c	0·06	0·17	0·55	2·63	10·77	19·12	43·58
t_s	—	0·05	0·22	1·87	17·41	48·02	176·71

optimistic. If the assumption $(m-1)\ell < n$ is considered, then

$$\frac{QRmem}{ITmem} = \frac{m^2n^2}{(n+(m-1)\ell)^2} > \frac{m^2\,n^2}{4n^2} = \frac{m^2}{4}. \tag{26}$$

Similarly, the assumption $(m-1)\ell < n/3$ implies the estimation $QRmem/Itmem > 9m^2/16$. In this case, using the five-step algorithm one needs 14 times less memory than using a QR algorithm working with the whole matrix.

5. CONCLUDING REMARKS

The computational experiments reported in the first four tables warrant one to expect that the algorithm could work effectively also for $m > 5$. The expectation is supported by computational experiments with the solution of systems of linear equations presented in references [11] and [12], where higher step algorithms and larger matrices are considered. Moreover, the partial matrix eigenproblems solved inside the iterations could be as large as one is able to solve effectively and with sufficient accuracy, i.e., essentially larger than 732 in the solution of equation (4) by the five-step CSET using $N = 3600$ basis functions. Finally, the optimization of these two features of the proposed algorithm, supported by the newest achievments in parallel computing, may reveal a new horizon in the solution of very large dense matrix eigenproblems resulting from the solution of the vibration problems by using spectral methods.

Unfortunately, the alternating fixed-free subspace strategy is not suitable for the use with local approximation methods as the finite element method and spline approximation. It this case it is not clear how to find good initial approximations. The problem originates from the fact that subsets of finite element trial functions (representing clusters of FEM trial functions) localized on a part of the whole domain Ω are identically zero on the remaining part of the domain and cannot give neither good nor reasonable approximation of an eigenfunction of equation (1) localized on the whole domain Ω. Although the algorithm works also in this case, the convergence of the resulting eigenvalue approximations is extremely slow and the use of the multigrid methods [13, 14] is recommended.

The computations presented in this article have been carried out on a personal computer with 150 MHz Pentium processor, 128 MB RAM, and 32-bit Microsoft Fortran PowerStation Professional Development System working under Windows 95.

REFERENCES

1. Y. SAAD 1992 *Numerical Methods for Large Eigenvalue Problems*. Manchester: Manchester University Press.

2. J. H. WILKINSON and C. REINSCH (editors) 1971 *Handbook for Automatic Computation, Vol. II: Linear Algebra.* Heidelberg–Berlin–New York: Springer-Verlag.
3. B. N. PARLETT 1980 *The Symmetric Eigenvalue Problem.* Englewood Cliffs, NJ: Prentice-Hall.
4. G. GOLUB and H. A. VAN DER VORST 2000 *Journal of Computational and Applied Mathematics* **123**, 35–65. Eigenvalue computation in the 20th century.
5. J. K. CULLUM and R. A. WILLOUGHBY 1985 *Lanczos Algorithms for Large Symmetric Eigenvalue Computations.* Boston: Birkhäuser.
6. E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV and D. SORENSEN 1995 *LAPACK—User's Guide.* Philadelphia: SIAM, second edition.
7. R. B. LEHOUCQ, D. C. SORENSEN and C. YANG 1998 *ARPACK User's Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* Philadelphia: SIAM.
8. M. A. KRASNOSEL'SKII, G. M. VAINIKO, P. P. ZEBREICO, Y. B. RUTITSKII and V. Y. STETSENKO 1972 *Approximate Solution of Operator Equations.* Groningen: Wolters-Noordhoff.
9. A. RUHE and P. Å. WEDIN 1980 *Society for Industrial and Applied Mathematics Review* **22**, 318–337. Algorithms for separable nonlinear least squares problems.
10. Y. BEPPU and I. NINOMIYA 1981 *Computers Physics Communications* **23**, 123–126. NICER—fast eigenvalue routines.
11. P. ŽITŇAN 2000 *Computational Mechanics* **25**, 28–32. A cluster series expansion technique for the spectral solution of differential equations.
12. P. ŽITŇAN 2000 *Parallel Numerics* 2000 *Conference. Bratislava, Slovakia*, September 18–20. Spectral solution of differential equations with great potential for parallelisation.
13. A. BRANDT, S. MCCORMICK and J. RUGE 1983 *Society for Industrial and Applied Mathematics Journal on Scientific and Statistical Computing* **4**, 244–260. Multigrid methods for differential eigenproblems.
14. C. BRAND and S. PETROVA 1994 *Preliminary Proceedings of the Colorado Conference on Iterative Methods. Breckenridge, CO*, April 1994. Preconditioned iterations to calculate extreme eigenvalues.