Journal of Sound and Vibration (2002) **254**(3), 599–612
doi:10.1006/jsvi.2001.4102, available online at http://www.idealibrary.com on IDEAL®

COMPUTING THE LOWEST EIGENVALUE WITH RAYLEIGH QUOTIENT ITERATION

S. RAJENDRAN

School of Mechanical and Production Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. *E-mail: msrajendran@ntu.edu.sg*

(*Received* 7 *August* 2000, *and in final form* 2 *August* 2001)

1. INTRODUCTION

Free vibration analysis of structures involves solution of generalized eigenproblem given by

$$\mathbf{Kp} = \lambda \mathbf{Mp}, \tag{1}$$

where $\mathbf{K}$ and $\mathbf{M}$ are $n \times n$ matrices, $\mathbf{p}$ is an eigenvector, and λ is the corresponding eigenvalue. In this work, it is assumed that the matrices $\mathbf{K}$ and $\mathbf{M}$ are real, symmetric and positive definite. The eigenvalues of such eigenproblems are all real and positive. Several eigensolution methods exist in the literature [1–7]. Inverse iteration is a vector iterative method primarily used for the computation of the smallest eigenvalue and the corresponding eigenvector. The inverse iteration also forms an integral part of hybrid eigensolution methods (e.g., see reference [1]) such as Lanczos methods, simultaneous/subspace iteration method, determinant search method, Rayleigh quotient iteration and Householder QR inverse iteration (HQRI) method.

The convergence of inverse iteration can be very slow if the eigenvalues are closely spaced or the starting iteration vector is dificient in the first eigenvector. Rayleigh quotient iteration [1] is a shifted inverse iteration method where the iterations are performed as an alternate sequence between the following pair of equations:

$$[\mathbf{K} - \rho(\mathbf{x}_{i-1})\mathbf{M}]\bar{\mathbf{x}}_i = \mathbf{M}\mathbf{x}_{i-1}, \qquad \mathbf{x}_i = \frac{\bar{\mathbf{x}}_i}{\sqrt{\bar{\mathbf{x}}_i^{\mathrm{T}}\mathbf{M}\bar{\mathbf{x}}_i}}, \tag{2, 3}$$

where Rayleigh's quotient is computed as

$$\rho(\bar{\mathbf{x}}_{i-1}) = \frac{\bar{\mathbf{x}}_{i-1}^{\mathrm{T}}\mathbf{K}\bar{\mathbf{x}}_{i-1}}{\bar{\mathbf{x}}_{i-1}^{\mathrm{T}}\mathbf{M}\bar{\mathbf{x}}_{i-1}} \equiv \frac{\mathbf{x}_{i-1}^{\mathrm{T}}\mathbf{K}\mathbf{x}_{i-1}}{\mathbf{x}_{i-1}^{\mathrm{T}}\mathbf{M}\mathbf{x}_{i-1}} \equiv \mathbf{x}_{i-1}^{\mathrm{T}}\mathbf{K}\mathbf{x}_{i-1} \equiv \rho(\mathbf{x}_{i-1}). \tag{4}$$

The iteration is terminated whenever the relative change in Rayleigh's quotient between successive iterations is less than the allowable tolerance. The convergence of this method is reported to be cubic [1, 8]. In spite of its excellent convergence characteristics, the method may, in principle, converge arbitrarily to any eigenvalue depending on the magnitude of the shift value, and hence assuring convergence to any particular eigenvalue is difficult. This is in contrast with the inverse iteration method, where the convergence is always towards the first eigenvalue, provided the starting vector is not deficient in the first eigenvector.

0022-460X/02/$35.00

If the first eigenpair, viz., the first eigenvalue and the corresponding eigenvector, is required to be computed using Rayleigh quotient iteration, one possibility is to initially use the inverse iteration until Rayleigh's quotient becomes sufficiently close to the first eigenvalue, and then switch over to Rayleigh quotient iteration. However, it is often difficult [1] to decide how many iterations are required before switching so as to ensure convergence to the first eigenpair. In this work, a modification of Rayleigh quotient iteration algorithm is proposed for forcing the iterations converge towards the first eigenvector.

At every iteration of the proposed algorithm, a linear combination of the latest and the preceding iteration vectors is formed involving an undetermined scalar, and the scalar is determined by minimizing the Rayleigh quotient. The concept of Rayleigh quotient minimization for obtaining an optimum linear combination of two vectors has been used in several different contexts, as, for example, the conjugate gradient methods [9–12] and co-ordinate relaxation methods [12–14]. In this work, Rayleigh quotient minimization is used as a tool to force the iterations converge towards the first eigenvector. Recently, the technique has been used for improving the convergence of inverse iteration by Rajendran *et al.* [15] and for accelerating the subspace iteration by Rajendran and Narasimhan [16].

2. MODIFIED RAYLEIGH QUOTIENT ITERATION ALGORITHM

The modified Rayleigh quotient iteration is carried out as follows:

Step 1: Set up a randomly generated vector, $\mathbf{x}_0$, of order n. Set the iteration no. $i = 0$. Let the shift value, s, be initially set to zero.

Step 2: Increment the iteration number by unity. Solve for $\bar{\mathbf{x}}_i$ from the equation

$$[\mathbf{K} - s\mathbf{M}]\bar{\mathbf{x}}_i = \mathbf{M}\mathbf{x}_{i-1}. \tag{5}$$

Step 3: Form a linear combination of vectors $\bar{\mathbf{x}}_i$ and $\mathbf{x}_{i-1}$ as

$$\hat{\mathbf{x}}_i = \mathbf{x}_{i-1} + \alpha\bar{\mathbf{x}}_i, \tag{6}$$

where α is an undetermined scalar. Choose the value of α such that Rayleigh quotient, $\rho(\hat{\mathbf{x}}_i)$, reaches a minimum value, $\rho(\hat{\mathbf{x}}_i)_{min}$. Set $s = \rho(\hat{\mathbf{x}}_i)_{min}$.

Step 4: Normalize the vector $\hat{\mathbf{x}}_i$ to obtain

$$\mathbf{x}_i = \frac{\hat{\mathbf{x}}_i}{\sqrt{\hat{\mathbf{x}}_i^T\mathbf{M}\hat{\mathbf{x}}_i}}. \tag{7}$$

Step 5: Check for convergence. If convergence has not been reached go to Step 2.

The difference between the proposed algorithm and the classical algorithm is Step 3. The choice of α such that the Rayleigh's quotient is minimum tends to improve the strength of first eigenvector component in the iteration vectors. This enhances the chances of convergence towards the first eigenvector. Of course, for the iterations to proceed towards first eigenvector, the vectors $\bar{\mathbf{x}}_i$ and $\mathbf{x}_{i-1}$ should not be completely void of the first eigenvector.

As apparent from the algorithm described above, the proposed algorithm is still a single-iteration-vector algorithm like the original Rayleigh quotient algorithm. However, at every iteration, the subspace spanned by the previous iteration vector, $\mathbf{x}_{i-1}$, and the

current iteration vector, $\bar{\mathbf{x}}_i$, is scanned to locate the "best" iteration vector, $\bar{\mathbf{x}}_i$, which is then used on the right-hand side of equation (5) during the next iteration after a normalization process shown by equation (7). The "best" vector is obtained by Rayleigh quotient minimization with respect to the parameter, α, defined in equation (6). Nevertheless, the present algorithm, cannot be viewed as a subspace iteration algorithm which is basically a multi-iteration-vector algorithm. Furthermore, at every iteration of subspace iteration method, the "best" input vectors for the next iteration are located from the subspace spanned by the current iteration vectors only, and the previous iteration vectors do not enter the process of locating the "best" vectors. While multi-vector algorithms like the subspace iteration are more efficient for the extraction of a set of lowest eigenvalues, the single-vector algorithms are generally more efficient when only the first eigenvalue is sought which is often the case in many practical engineering problems.

2.1. COMPUTATING THE SCALAR, α

Rayleigh's quotient corresponding to the vector $\hat{\mathbf{x}}_i$ is given by

$$\rho(\hat{\mathbf{x}}_i) = \frac{\hat{\mathbf{x}}_i^{\mathrm{T}} \mathbf{K} \hat{\mathbf{x}}_i}{\hat{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \hat{\mathbf{x}}_i}. \tag{8}$$

Using equations (5)–(7), this can be rewritten as

$$\rho(\hat{\mathbf{x}}_i) = \frac{A + 2\alpha B + \alpha^2 C}{1 + 2\alpha D + \alpha^2 E}, \tag{9}$$

where

$$A = \rho(\hat{\mathbf{x}}_{i-1})_{min} = \rho(\mathbf{x}_{i-1}) = s, \tag{10}$$

$$B = 1 + s\,\bar{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \mathbf{x}_{i-1}, \qquad C = \bar{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \mathbf{x}_{i-1} + s\,\bar{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \bar{\mathbf{x}}_i, \tag{11, 12}$$

$$D = \bar{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \mathbf{x}_{i-1}, \qquad E = \bar{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \bar{\mathbf{x}}_i. \tag{13, 14}$$

Equating the first derivative of $\rho(\hat{\mathbf{x}}_i)$ with respect to α to zero, the condition for the stationarity of $\rho(\hat{\mathbf{x}}_i)$ is obtained as

$$F\alpha^2 + G\alpha + H = 0, \tag{15}$$

where

$$F = 2(DC - BE) = 2\left[(\bar{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \mathbf{x}_{i-1})^2 - \bar{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \bar{\mathbf{x}}_i\right], \tag{16}$$

$$G = 2(C - AE) = 2\bar{\mathbf{x}}_i^{\mathrm{T}} \mathbf{M} \mathbf{x}_{i-1}, \tag{17}$$

$$H = 2(B - AD) = 2. \tag{18}$$

It can be shown that equation (15) has two real roots, one negative and the other positive, and the negative root corresponds to the minimum and is used in equation (6). This can be

proved by establishing that $(G^2 - 4FH) > G^2 \Rightarrow FH < 0 \Rightarrow F < 0$. This is done by first rewriting the right side of equation (16) as $2[(\bar{\mathbf{x}}_i^{\mathrm{T}}\mathbf{M}\mathbf{x}_{i-1})^2 - (\bar{\mathbf{x}}_i^{\mathrm{T}}\mathbf{M}\bar{\mathbf{x}}_i)(\mathbf{x}_{i-1}^{\mathrm{T}}\mathbf{M}\mathbf{x}_{i-1})]$ wherein the result $(\mathbf{x}_{i-1}^{\mathrm{T}}\mathbf{M}\mathbf{x}_{i-1}) = 1$ arising out of normalization in equation (3) is used, and then applying Cauchy–Schwarz inequality to equation (16).

2.2. NUMERICAL ASPECTS

The numerical stability of the method of computing the optimum value of α, particularly under the condition when the vector $\mathbf{x}_{i-1}$ has already converged close to the first eigenvector $\mathbf{p}_1$, is investigated here.

Let

$$\mathbf{x}_{i-1} = \mathbf{p}_1 + \varepsilon\mathbf{z}, \tag{19}$$

where ε is a scalar of small value and $\mathbf{z}$ is the polluting vector:

$$\mathbf{x}_{i-1}^{\mathrm{T}}\mathbf{M}\mathbf{x}_{i-1} = 1 \quad \text{(by equation (7))}. \tag{20}$$

Using equation (19) in equation (20), and assuming that the eigenvector $\mathbf{p}_1$ is normalized such that $\mathbf{p}_1^{\mathrm{T}}\mathbf{M}\mathbf{p}_1 = 1$, it can be shown that

$$\mathbf{p}_1^{\mathrm{T}}\mathbf{M}\mathbf{z} = -\tfrac{1}{2}\varepsilon\mathbf{z}^{\mathrm{T}}\mathbf{M}\mathbf{z}. \tag{21}$$

The expression for Rayleigh's quotient $\rho(\mathbf{x}_{i-1})$ can be shown (e.g., see reference [1]) to be

$$\rho(\mathbf{x}_{i-1}) = \lambda_1 + o(\varepsilon^2), \tag{22}$$

where λ_1 is the first eigenvalue and the notation $o(\varepsilon^2)$ means "of the order of ε^2".

Using equation (19) for eliminating $\mathbf{x}_{i-1}$ in equation (5) and noting that $s = \rho(\mathbf{x}_{i-1})$, equation (5) can be shown (e.g., see p. 624 of reference [1]) to yield

$$\bar{\mathbf{x}}_i \approx \frac{\mathbf{p}_1}{o(\varepsilon^2)}. \tag{23}$$

Using equations (19) and (20) in equations (16)–(18), it can be shown that

$$F = o(\varepsilon^{-2}) \quad \text{and} \quad G = o(\varepsilon^{-2}). \tag{24}$$

As the iterations approach convergence, F and G assume larger and larger values. In the case of 16-digit precision arithmetic, unless the convergence tolerance is equal to the precision of computation itself, viz., 10^{-16}, ε is non-zero, and therefore F and G take only finite values. For engineering computations, a tolerance of 10^{-6} is usually sufficient and hence solution of the quadratic equation (15) does not pose any numerical difficulties. The method of computation of α is therefore numerically stable.

During computation, it is quite possible that the iteration suddenly converges to the first eigenpair, i.e., $\rho(\bar{\mathbf{x}}_i)$ becomes very close to λ_1 although $|(\rho(\mathbf{x}_{i-1}) - \rho(\bar{\mathbf{x}}_i))/\rho(\bar{\mathbf{x}}_i)|$ is greater than the convergence tolerance. This happens when the Rayleigh quotient computed in the previous iteration is close to λ_1. Since the Rayleigh quotient is used as the shift value, the

matrix $[\mathbf{K} - s\mathbf{M}]$ may become singular to working precision. In such a case, the iteration is stopped and the current Rayleigh's quotient is taken as the first eigenvalue.

In the case of inverse iteration, the magnitude of Rayleigh quotient decreases monotonically with the iteration number. However, in Rayleigh quotient iteration and hence in the proposed algorithm such a monotonic decrease cannot be guaranteed. During iterations, it is sometimes possible that $\rho(\bar{\mathbf{x}}_i) > \rho(\mathbf{x}_{i-1})$. Such a condition will induce slow convergence. Whenever such condition occurs during iterations, the current iteration is discarded and another inverse iteration is performed before proceeding to the next iteration.

With these precautions, it has been possible to compute the eigenvalue upto an accuracy 10^{-14} for the numerical results reported in this work.

3. DEMONSTRATIVE EXAMPLES

The effectiveness of the proposed modification is demonstrated here for four typical test eigenproblems. The first three problems represent three broad type eigenproblems encountered in practice. Problem no. 1 has all *distinct eigenvalues*, problem no. 2 has a pair of *close eigenvalues*, and problem no. 3 has three pairs of *coincident eigenvalues*, each pair well separated from others. Problem no. 4 is used to demonstrate a case of slow convergence and the ways to overcome it. For simplicity, the matrix, $\mathbf{M}$, has been chosen to be unit matrix for all the problems.

Test problem no. 1:

$$\mathbf{K} = \begin{bmatrix} 10 & -2 & 0 & 0 & 0 & 0 \\ -2 & 10 & -2 & 0 & 0 & 0 \\ 0 & -2 & 10 & -2 & 0 & 0 \\ 0 & 0 & -2 & 10 & -2 & 0 \\ 0 & 0 & 0 & -2 & 10 & -2 \\ 0 & 0 & 0 & 0 & -2 & 10 \end{bmatrix}.$$

Eigenvalues: 6·396124528390, 7·506040792565, 9·109916264174, 10·890083735825, 12·493959207434, 13·603875471609.

Test problem no. 2:

$$\mathbf{K} = \begin{bmatrix} 1{\cdot}2649 & 1{\cdot}0297 & -0{\cdot}6953 & 0{\cdot}0711 & -0{\cdot}2032 \\ 1{\cdot}0297 & 2{\cdot}0528 & -1{\cdot}5026 & 0{\cdot}3291 & -0{\cdot}5689 \\ -0{\cdot}6953 & -1{\cdot}5026 & 2{\cdot}4776 & -1{\cdot}3202 & 0{\cdot}6515 \\ 0{\cdot}0711 & 0{\cdot}3291 & -1{\cdot}3202 & 11{\cdot}5785 & 2{\cdot}9143 \\ -0{\cdot}2032 & -0{\cdot}5689 & 0{\cdot}6515 & 2{\cdot}9143 & 1{\cdot}7983 \end{bmatrix}.$$

Eigenvalues: 0·460397657502, 0·460510328556, 1·191540364370, 4·544285979171, 12·515365670401.

Test problem no. 3:

$$\mathbf{K} = \begin{bmatrix} 300 & 47 & -17 & 0 & -25 & 77 \\ 47 & 400 & -54 & 25 & 0 & 85 \\ -17 & -54 & 500 & -77 & -85 & 0 \\ 0 & 25 & -77 & 300 & 47 & -17 \\ -25 & 0 & -85 & 47 & 400 & -54 \\ 77 & 85 & 0 & -17 & -54 & 500 \end{bmatrix}.$$

Eigenvalues: 268·6291653522, 268·6291653522, 338·2127620091, 338·2127620091, 593·1580726385, 593·1580726385.

Test problem no 4: The **K** matrix for this problem is similar to problem no. 1 (with 10 as diagonal elements and -2 as the super- and sub-diagonal elements, all other elements being zero), but the size is 20×20.
Eigenvalues: 6·04467669509949, 6·17770877685544, 6·39612452839032, 6·69504490273602,

7·06779251268069, 7·50604079256507, 8·00000000000000, 8·53863590253442,

9·10991626417474, 9·70107962565430, 10·29892037434570, 10·89008373582526,

11·46136409746558, 12·00000000000000, 12·49395920743493, 12·93220748731931,

13·30495509726398, 13·60387547160968, 13·82229122314456, 13·95532330490051.

The test problems were solved for the smallest eigenvalue using the classical Rayleigh quotient method as well as the proposed method. The computations were carried out in double precision arithmetic (16 digits) on a personal computer using MATLAB under Windows environment.

For each test problem, the starting vector (which was generated randomly) and Rayleigh's quotients computed at every iteration are listed in Tables 1–4. Tables 1 and 2 show that for test problem nos. 1 and 2, the unassisted Rayleigh quotient iteration converges to the third eigenvalue. Table 3 shows that for test problem no. 3, it converges to the second pair of repeated eigenvalues. For test problem no. 4, it converges to the sixth eigenvalue (Table 4). However, the modified Rayleigh quotient iteration converges invariably to the first eigenvalue for all the four test problems.

Although not apparent from Tables 1–4, under certain conditions the proposed algorithm has a tendency to "home in" towards each eigenvalue in a descending sequence, and thereby increase the number of iterations. This characteristics is exhibited particularly for larger eigenproblems, and starting vectors rich in eigenvectors corresponding to large eigenvalues, and has been investigated for test problem no. 4. The results are summarized in Table 5. The starting vector was "doped" to different levels using a few steps of power method so that it becomes rich in eigenvectors corresponding to the large eigenvalues. The doping factor in Table 5 indicates the number of power iterations employed. It is seen from Table 5 that for a doping factor of 5, the algorithm has not converged to the first eigenvalue even after 20 iterations.

In order to overcome the problem discussed above, two schemes of implementation of the proposed algorithm were tried. In scheme 1, five inverse iterations were carried out before starting the proposed algorithm. In scheme 2, the inverse iteration and the modified Rayleigh quotient algorithm were used in an alternate sequence. Table 6 shows the iteration

TABLE 1

Test results for problem no. 1

Starting vector:
[0·8673034878447652, 0·2032898580655595, −0·5859922412927910, 0·654497117681329, −0·8613606283392751, −0·7142207277469658]^T

Itrn. no.	Rayleigh's quotient
Classical Rayleigh quotient itrn.	
1	9·351328902712025
2	9·215555235890722
3	9·111379533590579
4	9·109916263068502
5	9·109916264174743
6	9·109916264174743
Modified Rayleigh quotient itrn.	
1	9·351328902712025
2	9·215555235890722
3	7·513273388370029
4	7·506040754073311
5	6·398799284847999
6	6·396124528391378
7	6·396124528390323

TABLE 2

Test results for problem no. 2

Starting vector:
[0·6605069125071168, 0·9625388039276095, −0·1693293884017829, 0·3212135355498786, 0·01390640349600392]^T

Itrn. no.	Rayleigh's quotient
Classical Rayleigh quotient itrn.	
1	1·082662339604678
2	1·172597059940892
3	1·191526325238709
4	1·191540364370301
5	1·191540364370307
Modified Rayleigh quotient itrn.	
1	1·082662339604678
2	1·172597059940892
3	0·4844642201007558
4	0·4604712352736948
5	0·4604912064345630
6	0·4603976575020788
7	0·4603976575020757

results for a value of dope factor equal to 5. It is seen that in both the schemes, the number of modified Rayleigh quotient iterations required are considerably reduced as compared to Table 5. The reduction is more significant for scheme 1. The additional inverse

TABLE 3

Test results for problem no. 3

Starting vector:
[0·6605069125071168, 0·9625388039276095, −0·1693293884017829, 0·3212135355498786, 0·01390640349600392]^{T}

Itrn. no.	Rayleigh's quotient
Classical Rayleigh quotient itrn.	
1	330·8221490264374
2	337·7638518050452
3	338·2127422808746
4	338·2127620091811
5	338·2127620091811
Modified Rayleigh quotient itrn.	
1	330·8221490264374
2	337·7638518050452
3	270·9079758588956
4	268·6293006299139
5	268·6291653522920
6	268·6291653522920

TABLE 4

Test results for problem no. 4

Starting vector:
[0·20040303902474, 0·14084898383871, 0·06828870872872, −0·04982408234872, −0·12279682670017, −0·09409858914350, 0·01603567451475, 0·12581172770249, 0·15344333567763, 0·07871936937592, −0·04179011661593, −0·11733902140054, −0·09133886262867, 0·01603567451475, 0·12305200118766, 0·14798553037800, 0·07068540364313, −0·05222077726314, −0·12993337323945, −0·10581556865075]

Itrn. no.	Rayleigh's quotient
Classical Rayleigh quotient itrn.	
1	7·5027115550871
2	7·060406199968
3	7·5060407925650
Modified Rayleigh quotient itrn.	
1	7·5027115590871
2	7·4285573446753
3	6·4930807293976
4	6·1838851162124
5	6·151337825768
6	6·089373331722
7	6·0450965853649
8	6·044676659537
9	6·0446769509949

TABLE 5

Convergence of the modified Rayleigh quotient iteration with poor starting vectors

	Dope factor			
Itrn. no.	0	1	3	5
1	7·11275158423816	8·46186084741981	11·91212178140190	13·02664781169423
2	6·90402948968495	7·15395777402287	11·59859793730027	12·77146578263790
3	6·52632562428630	6·84702897039340	11·25098389014921	12·02230188692784
4	6·34910417830313	6·47104428167842	10·59055425932153	11·99296733422946
5	6·19078524984364	6·36086679869961	10·04629858964318	11·89017957701190
6	6·11352442101270	6·16908851821433	9·61209183528322	11·24064332441345
7	6·05108061371926	6·11465189743524	9·21615859573107	10·69200627770691
8	6·04467777763758	6·05462362604674	8·50581457415805	10·12992301027776
9	6·04467669509949	6·04468576755762	8·16599739192099	9·77243513371510
10		6·04467669509949	7·52267708838232	9·69956708541769
11			7·49925722205958	9·15477395085166
12			7·41892764898962	9·01917323735062
13			6·80133395926676	8·64130949211539
14			6·53422804614823	8·23479023407235
15			6·35475689202736	7·64779961186355
16			6·18449960604182	7·47974822060612
17			6·14149657750936	7·13120032981022
18			6·07502368511566	7·03768854603411
19			6·04480501377325	6·79884178630988
20			6·04467669510920	6·61275242383886

TABLE 6

Convergence of schemes 1 *and* 2 *of modified Rayleigh quotient iterations for dope factor* = 5

Itrn. no.	Scheme 1	Scheme 2
1	12·6674226364862 I[†]	12·66742263648623 I
2	11·9121217814019 I	11·81324230549916 M[‡]
3	10·4123650573217 I	11·23426585451453 I
4	8·46186084741981 I	10·51700855671701 M
5	7·11275158423816 I	9·63912983399942 I
6	6·90402948968494 M	9·39690564358033 M
7	6·52632562428629 M	8·29664826791887 I
8	6·34910417830313 M	6·88059785749399 M
9	6·19078524984363 M	6·67543664443282 I
10	6·11352442101274 M	6·57098888749898 M
11	6·05108061371928 M	6·48667628888171 I
12	6·04467777763758 M	6·33083556240958 M
13	6·04467669509949 M	6·31224789992960 I
14		6·08794738514923 M
15		6·07275547061112 I
16		6·04479400090013 M
17		6·04478400585309 I
18		6·04467669510401 M
19		6·04467669510245 I
20		6·04467669509949 M

[†]The letter "I" refers to inverse iteration.
[‡]The letter "M" refers to modified Rayleigh quotient iteration.

iterations required in these schemes increase the total computational time marginally as the inverse iteration is computationally less intensive compared to Rayleigh quotient iteration.

The convergence of the proposed method towards the first eigenvalue is enabled by the Rayleigh quotient minimization employed. Although the proposed algorithm has invariably converged to the first eigenvalue for all the four test problems, theoretically, there exists a possibility that the algorithm could converge to eigenvalues other than the first, whenever the starting vector is deficient in the first eigenvector. Such possibility also exists whenever the shifts value (i.e., Rayleigh quotient computed in the previous iteration) is accidentally very close to an eigenvalue other than the first. After reaching convergence, a Sturm sequence would be useful to ensure if the computed eigenvalue is indeed the first eigenvalue.

4. COMPARISON OF PERFORMANCE WITH THE CONJUGATE GRADIENT ALGORITHMS

The proposed algorithm has some similarities with the conjugate gradient methods. Vast amount of literature exists on conjugate gradient methods, and for the present comparison, we refer, in particular, to the work of Yang *et al.* [17], and Perdon and Gambolati [18]. Typically, in conjugate gradient methods, the $(k+1)$th iteration vector, $\mathbf{x}_{k+1}$, is obtained from the kth iteration vector, $\mathbf{x}_k$, as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + t_k \mathbf{q}_k, \tag{25}$$

where t_k is a scalar parameter and $\mathbf{q}_k$ is the search direction. Equation (25) appears similar to equation (6) in form. Further, as in the proposed algorithm, the scalar parameter, t_k, is computed by minimizing the Rayleigh quotient, $\rho(\mathbf{x}_{k+1})$. In the conjugate gradient method, the successive search directions are chosen so as to satisfy the orthogonality condition

$$\mathbf{q}_{k+1}\, \mathbf{H} \mathbf{q}_k = 0, \tag{26}$$

where $\mathbf{H}$ is an $n \times n$ matrix, the different choice of which yields conjugate gradient algorithms of different rates of convergence [17]. In the proposed algorithm, however, the search direction is not explicitly defined as in the conjugate gradient methods although $\bar{\mathbf{x}}_i$ in equation (6) may perhaps be looked upon as the search direction; the search for the "best" vector is carried out in the two-dimensional subspace spanned by the current and the previous vectors (viz., $\bar{\mathbf{x}}_i$ and $\mathbf{x}_{i-1}$) of Rayleigh quotient iteration, as implied by equation (6).

In the rest of this section, the performance of the modified Rayleigh quotient algorithm is compared with that of typical conjugate gradient algorithms. First, the example problem solved by Yang *et al.* [17] is considered. This problem which is referred to as problem no. 5 hereinafter involves computing the smallest eigenvalue of the eigenproblem, $\mathbf{A}\mathbf{p} = \lambda \mathbf{p}$ where $\mathbf{A}$ is defined as follows.

Test problem no. 5:

$$\mathbf{A} = \begin{bmatrix} r_0 & r_1 & \cdots & r_{15} \\ r_1 & r_0 & \cdots & r_{14} \\ r_2 & r_1 & \cdots & r_{13} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ r_{15} & r_{14} & \cdots & r_0 \end{bmatrix},$$

where

$$r_0 = 1{\cdot}00000000,\ r_1 = 0{\cdot}91189350,\ r_2 = 0{\cdot}75982820,\ r_3 = 0{\cdot}59792770,$$

$$r_4 = 0{\cdot}41953610,\ r_5 = 0{\cdot}27267350,\ r_6 = 0{\cdot}13446390,\ r_7 = 0{\cdot}00821722,$$

$$r_8 = -0{\cdot}09794101,\ r_9 = -0{\cdot}21197350,\ r_{10} = -0{\cdot}30446960,\ r_{11} = -0{\cdot}34471370,$$

$$r_{12} = -0{\cdot}34736840,\ r_{13} = -0{\cdot}32881280,\ r_{14} = -0{\cdot}29269750,\ r_{15} = -0{\cdot}24512650.$$

The smallest eigenvalue of this problem is 0·00325850037049.

The four versions of conjugate gradient algorithms, viz., "CA", "TJ", "FR" and "HE" versions studied by Yang *et al.* [17], and the Rayleigh quotient modified conjugate gradient iteration (RQ-MCG) of Perdon and Gambolati [18] are used here for comparison. Perdon and Gambolati [18] consider various implementation of the RQ-MCG algorithm. For the present comparison, we use that particular implementation in which $\mathbf{A}^{-1}$ is taken as the pre-condition matrix as this particular implementation has been shown to be better than others. It is important to note that the various versions discussed by Yang *et al.* [17] do not involve inverse of **A** matrix whereas the RQ-MCG algorithm [18] does. The eigenvalues were computed for a convergence tolerence of 10^{-6} as well as 10^{-12}. The starting vector, $\mathbf{x}_0$, for all the algorithms is the same, and is obtained by perturbing the eigenvector corresponding to the largest eigenvalue; the first element of this eigenvector is multiplied by a factor of $(1 + 10^{-6})$. This process gives a starting vector which is very rich in the eigenvector corresponding to the largest eigenvalue and hence very poor in the first as well as other eigenvectors. Such a starting vector is useful in assessing how well the various algorithms converge to the first eigenvalue with a poor starting vector. The results are summarized in Table 7 wherein MRQ refers to a case where the proposed modified Rayleigh quotient iteration is used all through the iterations. INV-MRQ refers to a case where the first few iterations are inverse iterations followed by the modified Rayleigh quotient iterations thereafter.

Table 7 shows that INV-MRQ requires the least number of iterations and also the lowest computational time for convergence both for the convergence tolerance of 10^{-6} as well as

TABLE 7

Comparison of effectiveness with the conjugate gradient algorithms for test problem no. 5

	Conv. tol. = 10^{-6}			Conv. tol. = 10^{-12}		
Alg.	No. of Itrns.	CPU time†	Computed eigenvalue	No. of Itrns.	CPU time†	Computed eigenvalue
CA [17]	153	129	0·00325851622410	443	410	0·00325850037080
TJ [17]	106	84	0·00325853197061	172	137	0·00325850037055
FR [17]	119	92	0·00325852053118	196	150	0·00325850037053
HE [17]	83	77	0·00325851284326	139	128	0·00325850037050
RQ-MCG [18]	9	6·9	0·00325850039639	11	91	0·00325850037050
MRQ‡	7	9·7	0·00325850037049	7	9·7	0·00325850037049
INV-MRQ§	6	4·6	0·00325850037049	6	4·6	0·00325850037049

†In arbitrary units.

‡Modified Rayleigh quotient algorithm at every iteration.

§k number of inverse iterations first, followed by modified Rayleigh quotient iteration for the rest of the solution process. $k = 3$ and $k = 2$ for convergence tolerance 10^{-6} and 10^{-12} respectively.

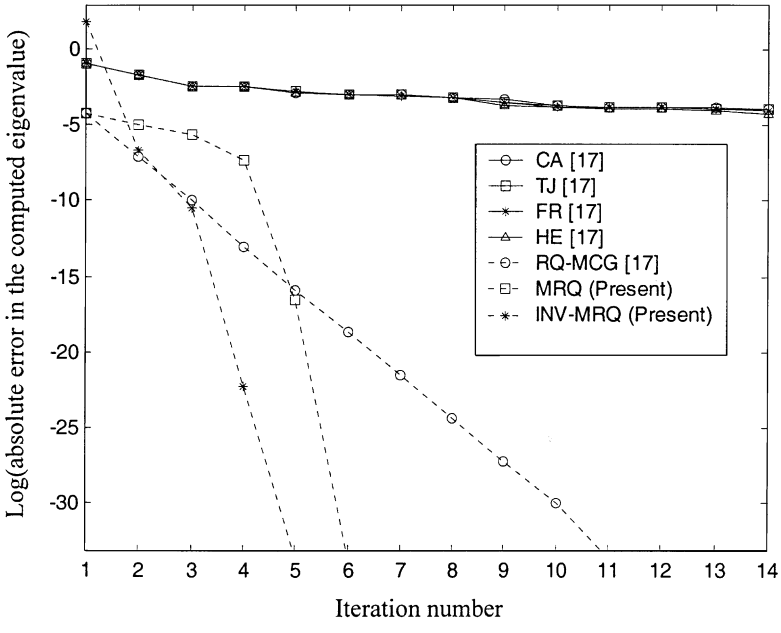

Figure 1. Convergence of first eigenvalue for test problem no. 5.

10^{-12}. The computational times reported have been calculated using the standard MATLAB commands, and are to be taken as a rough estimate as the computations have been carried out under the Windows environment, wherein the effect of the system-related background processes on the time estimates is difficult to assess. The values reported here are the average of 1000 repetitive time estimates. An interesting observation from Table 7 is that the eigenvalue computed with MRQ or INV-MRQ algorithm is accurate to 14 decimal places even with a specified convergence tolerance of 10^{-6}. This feature of the proposed algorithm is due to the inheritance of the cubic convergence characteristics from the parent algorithm, viz., Rayleigh quotient iteration, and suggests that MRQ and INV-MRQ are particularly effective when the eigenvalue needs to be computed to a higher precision. Figure 1 shows the plot of convergence of various algorithms wherein the higher order convergence of MRQ and INV-MRQ is clearly seen.

The various algorithms compared above were also applied to test problem no. 4. In this case, the starting vector used is of the form $[1, 0, 0, \ldots, 0]$ where the first entry is unity and the rest are all zero. The results are summarized in Table 8. A study of this table shows that, here again, INV-MRQ algorithm requires the least number of iterations as well as the lowest computational time. Also, the eigenvalue computed with MRQ or INV-MRQ algorithm is accurate to 14 decimal places even with a convergence tolerance of 10^{-6} as has already been observed for test problem no. 5. Figure 2 shows the convergence plot for test problem no. 4, which again confirms the higher order convergence characteristics of MRQ and INV-MRQ.

5. CONCLUDING REMARKS

The classical Rayleigh quotient iteration may converge arbitrarily to any eigenvalue, and hence assuring convergence to the first eigenvalue is difficult. In this work, a modification of

TABLE 8

Comparison of effectiveness with the conjugate gradient algorithms for test problem no. 4

Alg.	Conv. tol. = 10^{-6}			Conv. tol. = 10^{-12}		
	No. of Itrns.	CPU time†	Computed eigenvalue	No. of Itrns.	CPU time†	Computed eigenvalue
CA [17]	70	60	6·04471185521904	161	156	6·04467669513271
TJ [17]	31	31	6·04468149804272	60	50	6·04467669511078
FR [17]	31	29	6·04468305754015	53	41	6·04467669511614
HE [17]	30	28	6·04468116207983	54	50	6·04467669512065
RQ-MCG [18]	66	49	6·04470849686265	156	112	6·04467669513494
MRQ‡	18	24	6·04467669509949	18	24	6·04467669509949
INV-MRQ§	14	16	6·04467669509949	14	16	6·04467669509949

†In arbitrary units.
‡Modified Rayleigh quotient algorithm at every iteration.
§5 inverse iterations first, followed by modified Rayleigh quotient iteration for the rest of the solution process.

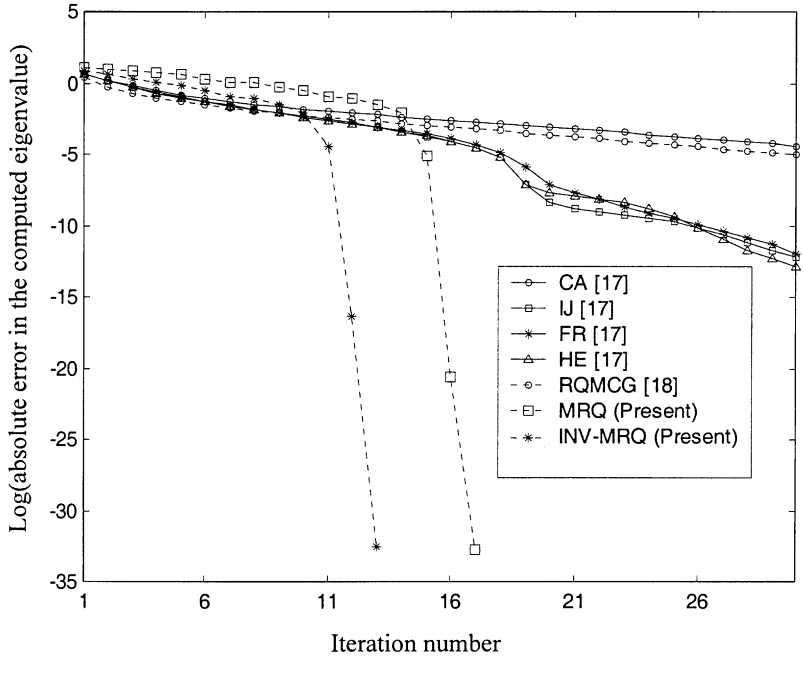

Figure 2. Convergence of first eigenvalue for test problem no. 4.

the Rayleigh quotient iteration is proposed to enable the iterations to converge towards the first eigenvector. Numerical examples presented demonstrate that the proposed algorithm does converge to the first eigenvalue. However, for larger eigenproblems and poor starting vectors, the proposed algorithm has a tendency to "home in" towards each eigenvalue in a decending sequence thereby increasing the number of iterations. This problem has been overcome by employing a few inverse iterations before starting the modified Rayleigh quotient algorithm. A comparison of performance with typical conjugate gradient

algorithms shows that the modified Rayleigh quotient algorithm requires less number of iterations and less computational time in most cases. The higher order convergence characteristics of MRQ and INV-MRQ algorithms renders their use particularly attractive when the eigenvalue needs to be computed with a greater precision. These attractive features suggest that the proposed algorithm is worth further investigations by other researchers.

REFERENCES

1. K. J. BATHE 1982 *Finite Element Procedures in Engineering Analysis*. Englewood Cliffs, NJ: Prentice-Hall.
2. L. MEIROVITCH 1980 *Computational Methods in Structural Dynamics*. Rockville, MD, U.S.A.: Sijthoff & Noordhoff.
3. A. JENNINGS 1980 *Shock and Vibrations Digest* **12**, 3–15. Eigenvalue methods for vibration analysis.
4. A. JENNINGS 1984 *Shock and Vibrations Digest* **16**, 25–33. Eigenvalue methods for vibration analysis II.
5. J. H. WILKINSON 1965 *The Algebraic Eigenvalue Problem*. Oxford: Clarendon Press.
6. K. K. GUPTA 1973 *International Journal for Numerical Methods in Engineering* **7**, 17–42. Eigenproblem solution by a combined Sturm sequence and inverse iteration technique.
7. K. K. GUPTA 1973 *International Journal for Numerical Methods in Engineering* **7**, 509–518. On a combined Sturm sequence and inverse iteration technique for eigenproblem solution of spinning structures.
8. B. N. PARLETT and W. KAHAN 1968 *Proceedings, I.F.I.P. Congress, Edinburgh*, 5–10 *August*. On the convergence of a practical QR algorithm.
9. F. SARTORETTO, G. PINI and G. GAMBOLATI 1989 *Journal of Computational Physics* **81**, 53–69. Accelerated simulataneous iterations for large finite element eigenproblems.
10. G. GAMBOLATI, G. PINI and F. SARTORETTO 1988 *Journal of Computational Physics* **74**, 41–60. An improved iterative optimization technique for the left most eigenpairs of large symmetric matrices.
11. M. GERADIN 1971 *Journal of Sound and Vibration* **19**, 319–331. The computational efficiency of a new minimization algorithm for eigenvalue analysis.
12. G. W. STEWART 1976 in *Sparse Matrix Computations* (J. R. Bunch and D. J. Rose, editors). New York: Academic Press Inc. A bibliographical tour of the large, sparse generalized eigenvalue problem.
13. A. PELLEW and R. U. SOUTHWELL 1940 *Proceedings of Royal Society*, **A 175**, 262–290. Relaxation methods applied to engineering problems VI. The natural frequencies of systems having restricted freedom.
14. H. R. SCHWARZ 1974 *Numerische Mathematik* **23**, 135–151. The method of coordinate overrelaxation for $(\mathbf{A} - \lambda\mathbf{B})\mathbf{x} = \mathbf{0}$.
15. S. RAJENDRAN, M. R. RAGHAVAN and M. V. NARASIMHAN 1995 *Computers & Structures* **55**, 709–715. Improving the convergence rate of inverse iteration method.
16. S. RAJENDRAN and M. V. NARASIMHAN 1994 *International Journal for Numerical Methods in Engineering* **37**, 141–153. An accelerated subspace iteration method.
17. X. YANG, T. K. SARKAR and E. ARVAS 1989 *IEEE Transactions on Acoustics, Speech and Signal Processing* **37**, 1550–1556. A survey of conjugate gradient algorithms for solution of extreme eigen-problems of a symmetric matrix.
18. A. PERDON and G. GAMBOLATI 1986 *Computer Methods in Applied Mechanics and Engineering* **56**, 251–264. Extreme eigenvalues of large sparse matrices by Rayleigh quotient and modified conjugate gradients.