COMPUTATION OF MINIMAL EIGENPAIR IN THE LARGE SPARSE GENERALIZED EIGEN-PROBLEM USING VECTOR COMPUTERS

Emanuele GALLIGANI', Valeria RUGGIERO'

' Department of Mathematics, University of Modena
' Department of Mathematics, University of Ferrara

Abstract

This paper is concerned with the computation of the minimal eigenpair of the generalized eigen-problem. An iterative method of first degree with the arithmetic mean preconditioner and the SSOR preconditioner has been analyzed. These preconditioners have been evaluated also for the method of Rayleigh quotient minimization.

1. THE BASIC ITERATIVE METHOD

In many areas of science and technology it is required to solve the generalized eigen-problem

\[ \lambda x = \lambda_B x \]  \hspace{1cm} (1)

where \( A \) and \( B \) are \( n \times n \) symmetric positive definite matrices. We are treating the case for which \( A \) and \( B \) are large and sparse. Usually, in this case, we are interested in the computation of the smallest eigenvalue of (1) and its corresponding eigenvector.

It is a well-known fact that, if \( A \) and \( B \) are \( n \times n \) symmetric matrices with \( B \) positive definite, problem (1) has \( n \) real eigenvalues \( \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \) and \( n \) corresponding linearly independent eigenvectors \( u_1, u_2, \ldots, u_n \) that can be chosen to be orthogonal in the inner product \( \langle u, v \rangle_B = u^T B v \). Moreover, if \( A \) is positive definite, then the eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_n \) are all positive.

Let \( A \) and \( B \) be \( n \times n \) symmetric matrices with \( B \) positive definite; then, the smallest eigenvalue \( \lambda_1 \) and the largest eigenvalue \( \lambda_n \) are the minimum and the maximum, respectively, of the Rayleigh quotient

\[ \rho(x) = \frac{x^T A x}{x^T B x} . \]  \hspace{1cm} (2)
An effective means for solving (1) is the iterative method of first degree \( k = 0, 1, \ldots \):
\[
M \frac{x^{(k+1)} - x^{(k)}}{\tau} + (A - \rho \, B) \, x^{(k)} = 0
\]  
(3)
where \( x^{(0)} \) is a given starting vector, \( M \) is a non-singular matrix, \( \tau \) is an iterative parameter \( \tau > 0 \) and \( \rho = \rho(x^{(k)}) \). The matrix \( M \) is taken to be an easily invertible matrix.

In literature (see, e.g. [9], [3]) widely used forms of \( M \) in (3) are obtained by a splitting of the symmetric matrix \( A - \rho \, B \).

An alternative strategy for speeding up the process (3) is to use a splitting of the matrix \( A \) instead of the matrix \( A - \rho \, B \) [2]. Now, since \( M \) is independent of \( k \), it is not necessary to compute again the elements of \( M \) at each iteration \( k \).

If the symmetric matrix \( A \) has the splitting \( A = L + D + L^T \), where \( L \) and \( L^T \) are the strictly lower and upper parts of \( A \) and \( D \) is the diagonal matrix with the same diagonal elements of \( A \), we consider in (3) the following forms of \( M \):
\[
M^{-1} = (D + \omega L^T)^{-1} D (D + \omega L)^{-1}
\]  
(4)
and
\[
M^{-1} = \frac{1}{2} \left[ (\frac{1}{\omega} D + L)^{-1} + (\frac{1}{\omega} D + L^T)^{-1} \right] \quad 0 < \omega < 2
\]  
(5)
We observe that the matrix \( M \) in (5) is the arithmetic mean preconditioner considered in [4] and the method (3)-(5) is ideally suited for implementation on a multiprocessor system with two or more vector processors.

Now, we follow [10] to prove the convergence of methods (3)-(4) and (3)-(5).

**Theorem** — Let \( x^{(k)} \) be the vectors determined by (3) and \( \rho \) be the corresponding sequence computed by (2), then the sequence \( \rho \) is a decreasing sequence if \( \omega \) and \( \tau \) are chosen to be in the intervals
\[
\frac{1 - \sqrt{\tau / 2}}{2 / \tau - 1} \leq \omega \leq \frac{1 + \sqrt{\tau / 2}}{2 / \tau - 1} \quad 0 < \tau \leq 1
\]  
(6)
when \( M \) has the form (4) and
\[
0 < \omega < 2 \quad 0 < \tau \leq 1
\]  
(7)
when \( M \) has the form (5).

**Proof** — An alternative form of (3) is given by
\[
x^{(k+1)} = x^{(k)} - \tau M^{-1} (A - \rho \, B) \, x^{(k)} = x^{(k)} - p^{(k)}
\]
where
\[
p^{(k)} = \tau M^{-1} (A - \rho \, B) \, x^{(k)} = \tau M^{-1} r^{(k)}
\]
For any vector \( x^{(k)} \) determined by (3) and the corresponding \( \rho \) obtained by (2) we have
\[(\rho_{k+1} - \rho_k) x^{(k+1)}^T B x^{(k+1)} = (x^{(k)} - p^{(k)})^T A x^{(k+1)} - \rho_k x^{(k+1)} B (x^{(k)} - p^{(k)}) =
\]
\[x^{(k+1)}^T (A - \rho_k B) x^{(k)} - p^{(k)}^T (A - \rho_k B) x^{(k+1)} = -p^{(k)}^T \left( \frac{2}{\tau} M - A + \rho_k B \right) p^{(k)} \]

(8)

Note that \(x^{(k)}^T r^{(k)} = 0\) and \(r^{(k)} = \frac{1}{\tau} M p^{(k)}\).

When we consider the matrix of formula (4), we have

\[
\frac{2}{\tau} M - A + \rho_k B = \left( \frac{2}{\tau} - 1 \right) D + \left( \frac{2 \omega^2}{\tau} - 1 \right) (L + L^T) + \frac{2 \omega^2}{\tau} L D^{-1} L^T + \rho_k B
\]

Since \(\omega\) and \(\tau\) are in the given intervals (6), for any \(p^{(k)} \neq 0\), it is easy to prove that

\[
\left( \frac{2 \omega^2}{\tau} - 1 \right)^2 p^{(k)^T} L D^{-1} L^T p^{(k)} \leq \frac{2 \omega^2}{\tau} p^{(k)^T} L D^{-1} L^T p^{(k)}
\]

Then

\[
p^{(k)^T} \left( \frac{2}{\tau} M - A + \rho_k B \right) p^{(k)} \geq p^{(k)^T} \left[ \frac{2}{\tau} - 2 \right] D + \\
+ \left[ D + \left( \frac{2 \omega}{\tau} - 1 \right) L \right] D^{-1} \left[ D + \left( \frac{2 \omega}{\tau} - 1 \right) L^T \right] p^{(k)} + \rho_k B p^{(k)} > 0
\]

Note that \(\rho_k > 0\) for all \(k = 0, 1, \ldots\)

Thus, by (8), the sequence \(\rho_k\) will form a decreasing sequence.

If we set \(R = \frac{1}{2} (L - L^T)\), the matrix \(M\) of formula (7) has the expression (4)

\[M = \frac{1}{2} \left( A + \frac{2 - \omega}{\omega} D \right) + R \left( \frac{1}{2} \left( A + \frac{2 - \omega}{\omega} D \right) \right)^{-1} R^T \quad 0 < \omega < 2\]

The symmetric matrix

\[\frac{2}{\tau} M - A + \rho_k B = \left( \frac{1}{\tau} - 1 \right) A + \frac{2 - \omega}{\tau \omega} D + \frac{2}{\tau} R \left( \frac{1}{2} \left( A + \frac{2 - \omega}{\omega} D \right) \right)^{-1} R^T + \rho_k B
\]

is positive definite for \(0 < \tau < 1\) and \(0 < \omega < 2\) since \(\rho_k > 0\) for all \(k=0,1,\ldots\). Thus, by (8), the sequence \(\rho_k\) will form a decreasing sequence.

To analyse the rate of convergence of methods (3)-(4) and (3)-(5), we remember that, using the Rayleigh quotient in an iterative method for the determination of the eigenvalues, the accuracy of the eigenvalue obtained is the square of that of the corresponding eigenvector (6). Hence, it seems reasonable to say that the rate of convergence of the method (3) is determined by the convergence rate of the limiting iteration (\(\tau=1\))

\[x^{(k+1)} = (I - M^{-1} (A - \lambda_k B)) x^{(k)} = Q x^{(k)}
\]

(9)

The matrix \(A - \lambda_k B\) is a positive semidefinite matrix.

When \(M\) has the form (5), the iteration matrix \(Q\) in (9) is convergent. For this, it suffices to prove that the symmetric
matrix \( M + M^r - (A-\lambda_1I) \) is positive definite; thus, the result follows from a theorem proved in [5]. We have

\[
P = M + M^r - (A-\lambda_1I) = (A + \frac{2-\omega}{\omega} D) + 4R \left( A + \frac{2-\omega}{\omega} D \right)^{-1} R^r - A + \lambda_1I =
\]

\[
= \frac{2-\omega}{\omega} D + 4R \left( A + \frac{2-\omega}{\omega} D \right)^{-1} R^r + \lambda_1I
\]

Obviously, \( P \) is symmetric positive definite for all \( 0 < \omega < 2 \).

Since \( Qu_1 = u_1 - M^r (A-\lambda_1I) u_1 = u_1 \), the vector \( u_1 \) is an eigenvector of \( Q \) corresponding to the eigenvalue \( \eta_1 = 1 \) of \( Q \). Besides, the iteration matrix \( Q \) has no principal vectors corresponding to the eigenvalue \( \eta_1 = 1 \). This follows immediately from the general proof given in [5].

Then, for \( 0 < \omega < 2 \), the spectral radius of \( Q \) is one and the eigenvalues \( \eta_j \neq 1 \) of \( Q \) have moduli strictly less than one. Therefore, the rate of convergence of (3)-(5) (with \( \tau = 1 \)) is determined by the number \( R = \max \frac{1}{\eta_j} \), where \( \eta_j \) is an eigenvalue of \( Q \).

When \( A \) has the 2-cyclic form

\[
A = \begin{pmatrix}
aI & -X \\
-X^T & aI
\end{pmatrix}
\]

and \( B = I \) (identity matrix), it is possible to determine an optimal value of \( \omega \) for the limiting iteration (9) with \( M \) given by formula (5) and \( \tau = 1 \). We assume that the constant \( a > \lambda_1 \).

We consider the Jacobi iteration matrix

\[
J = \frac{1}{a-\lambda_1} (aI - A) \quad \quad (a > \lambda_1 > 0)
\]

associated with the matrix \( A - \lambda_1I \) where \( A \) has the form (10). Since \( A - \lambda_1I \) is a 2-cyclic, symmetric positive semidefinite matrix with positive diagonal entries, we can use the theory of the successive overrelaxation method for symmetric and semidefinite linear systems [5]. Hence the eigenvalues \( \mu_j \) of the Jacobi matrix \( J \) are real, have pairwise opposite signs and \( |\mu_j| \leq 1 \). Moreover, the eigenvalue \( \mu_1 = 1 \) has the same multiplicity \( r_1 \) as \( \lambda_1 \) has, and therefore \(-1\) is also an eigenvalue of \( J \) of multiplicity \( r_1 \).

Then, \( \mu_j = \max \frac{1}{\eta_j} < 1; \mu_1 \) is the subdominant eigenvalue of \( J \).

Now, using a standard proof given in [1], it is possible to determine the best choice of the parameter \( \omega \) in (5). We have the following result.
Theorem. If $A$ has the form (10) with $a > \lambda_1$ and $B=I$, the optimal parameter $\omega_0$ in method (3)–(5) (with $\tau=1$) is given by

$$
\omega_0 = \frac{\mu_2 - \frac{3}{2} \sqrt{3-2\mu_1^2}}{(1-a^{-1}\lambda_1)(\frac{1}{4}+\mu_1-\mu_2)}
$$

(12)

and the corresponding optimal convergence rate is

$$
R = \frac{1}{8} \left(\omega_0^2 (1-a^{-1}\lambda_1)^2 + 4\omega_0 (1-a^{-1}\lambda_1 - 4) \right)
$$

2. COMPUTATIONAL EXPERIMENTS

In [10] and [3] several numerical tests of the iterative method (3) with $M$ obtained by the splitting of $A-\rho_x B$ have been performed on problems arising from the discretization of elliptic partial differential equations using finite difference or finite element methods.

In this section we evaluate the effectiveness of (3) when the matrix $M$ has the forms (4) and (5).

Sixteen different test-problems have been considered. Test-problem # 1 arises in solving the diffusion equation with piecewise continuous coefficients on a rectangular domain by the finite difference method with a non uniform net; $n = 4096$. Test-problems # 2–4 are those considered in [3]; the order $n$ of matrices $A$ and $B$ in these problems is varying from 1000 to 4096. Storing of such matrices is only by their nonvanishing diagonals (diagonal storing) [11]. Test-problems # 5–16 arise from the finite integration of flow and structural mechanics equations [8]. The matrices $A$ and $B$ of these problems are very sparse and within a banded structure; their order $n$ is varying from 456 to 4560. Each matrix is stored in compressed matrix representation (see ITPACKV and ELLPACK). In test-problems # 5–10, $B$ is the identity matrix; in test-problems # 11–16, $B$ is the matrix $SS^T$ where $S$, a lower triangular matrix, is determined using the incomplete Cholesky factorization of $A$.

In table 1 we report some results of computational experiments carried out on these test-problems using methods (3)-(5) and (3)-(4). We report also the results of the method described in [3] using the splitting of $A-\rho_x B$ (Preconditioned Method). $k^*$ indicates the number of iterations to make the relative error of the smallest eigenvalue of (1) less than $\varepsilon$. The initial vector $x^{(0)}$ is $(1 \ 1 \ldots 1)^T$ in test-problems # 1–4 and $x^{(0)} = (.1 \ .1 \ldots \ .1)^T$ in test-problems # 5–16. All calculations are performed using the
Table 1

<table>
<thead>
<tr>
<th>test-problem</th>
<th>method (3)-(5)</th>
<th>method (3)-(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ε</td>
<td>ω</td>
</tr>
<tr>
<td># 1 10^-2</td>
<td>1.95</td>
<td>1.7</td>
</tr>
<tr>
<td>n=4096</td>
<td></td>
<td></td>
</tr>
<tr>
<td># 2 10^-2</td>
<td>1.99</td>
<td>0.07</td>
</tr>
<tr>
<td>n=1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td># 3 10^-4</td>
<td>1.86</td>
<td>0.12</td>
</tr>
<tr>
<td>n=4096</td>
<td></td>
<td></td>
</tr>
<tr>
<td># 4 10^-2</td>
<td>1.99</td>
<td>0.07</td>
</tr>
<tr>
<td>n=1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td># 6 10^-2</td>
<td>1.9</td>
<td>14.15</td>
</tr>
<tr>
<td>n=1338</td>
<td></td>
<td></td>
</tr>
<tr>
<td># 8 10^-4</td>
<td>1.9</td>
<td>9.81</td>
</tr>
<tr>
<td>n=1802</td>
<td></td>
<td></td>
</tr>
<tr>
<td># 10 10^-4</td>
<td>1.9</td>
<td>31.75</td>
</tr>
<tr>
<td>n=4560</td>
<td></td>
<td></td>
</tr>
<tr>
<td># 13 10^-3</td>
<td>1.9</td>
<td>12.88</td>
</tr>
<tr>
<td>n=1338</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Computer CRAY Y-MP/432. Time indicates the computer time expressed in seconds for determining the minimal eigenpair of (1). Method (3)-(5) is implemented in parallel on two vector processors; for method (3)-(4) (and for the Preconditioned Method) only one vector processor is used.

For test-problems # 1-4, the triangular systems which appear in (4) and (5) are solved with the block forward-elimination algorithm or the block back-substitution algorithm combined with the cyclic reduction algorithm. For test-problems # 5-16, the triangular systems of (4) and (5) are solved with the Gaussian algorithm by using gather-scatter type operations to collect the nonzero elements. In table 1 the iterative parameter τ is equal to 1 and ω is the optimum value determined experimentally. For the test-problem # 4 the exact optimal value of ω given by
formula (12) is \( \omega_0 = 1.999952 \).

In table 2 we report the results of a numerical experiment on test-problem \# 2 in order to determine the number of iterations \( k' \) needed to satisfy, for different values of \( \omega \), the criterion

\[
\| A x^{(k')} - \rho \_k B x^{(k')} \| < 10^{-2} \left( x^{(k')} B x^{(k')} \right)^{1/2}
\]

Here: \( \text{err}_x = | \rho \_k - \lambda_1 | / | \lambda_1 |, \text{err}_x = \| x^{(k')} - u_1 \| / \| u_1 \| \) with \( u_1^T B u_1 = 1 \).

All these numerical experiments show a definite advantage for the method (3)-(5). However, it is well known [7], [8] that, when \( A \) and \( B \) in (1) arise from the finite element discretization of flow and structural mechanics equations, the method of Rayleigh quotient minimization by modified conjugate gradients (RQ-MCG) is very effective for solving (1). In this method the choice of the preconditioner is of paramount importance. In this paper, a comparison between the arithmetic mean preconditioner (5) and the SSOR preconditioner (4) for the RQ-MCG method has been carried out on the above test-problems \# 1-16. Results of a few computational experiments are reported in table 3. The initial vector \( y^{(0)} \) of the iterative method RQ-MCG has unitary elements; this method stops when

\[
\frac{\| A y^{(j)} \|}{\| B y^{(j)} \|} - \rho ( y^{(j)} ) < \sigma
\]

where \( \| \cdot \| \) is the euclidean norm and \( \sigma = 10^{-9} \). In table 3, \( j \) indicates the number of iterations for the convergence and time the computer-time on CRAY Y-MP/432 expressed in seconds for determining the eigenvalue \( \lambda_1 \) with the relative error

\[
\text{err} = | \rho ( y^{(j)} ) - \lambda_1 | / | \lambda_1 |
\]

For the preconditioner (4) only one vector processor is used; for the preconditioner (5) two vector processors are used. \( \omega \) is the optimum value determined experimentally.

From these results, conclusions that are similar to those drawn for the solution of linear systems of equations [4] apply: the arithmetic mean preconditioner appears attractive for determining, with the RQ-MCG method, the minimal eigenpair of (1) on a multivector computer; it is accurate and economical. For some test-problems (i.e. \# 10, \# 13 and \# 16) this preconditioner is also more efficient than the preconditioners that have been analyzed in [8].
<table>
<thead>
<tr>
<th>Method (3)-(5)</th>
<th>Method (3)-(4) [Preconditioned method]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega )</td>
<td>( \text{time (2CPU)} )</td>
</tr>
<tr>
<td>1.99</td>
<td>.25</td>
</tr>
<tr>
<td>1.9</td>
<td>.60</td>
</tr>
<tr>
<td>1.8</td>
<td>1.24</td>
</tr>
<tr>
<td>1.6</td>
<td>2.78</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Table 3</td>
<td>Preconditioner (5)</td>
</tr>
<tr>
<td>Test problem</td>
<td>( \omega )</td>
</tr>
<tr>
<td>1 #</td>
<td>1.8</td>
</tr>
<tr>
<td>2 #</td>
<td>1.86</td>
</tr>
<tr>
<td>6 #</td>
<td>1.5</td>
</tr>
<tr>
<td>8 #</td>
<td>1.9</td>
</tr>
<tr>
<td>9 #</td>
<td>1.3</td>
</tr>
<tr>
<td>10 #</td>
<td>1.8</td>
</tr>
<tr>
<td>12 #</td>
<td>1.9</td>
</tr>
<tr>
<td>13 #</td>
<td>1.5</td>
</tr>
<tr>
<td>16 #</td>
<td>1.3</td>
</tr>
</tbody>
</table>
REFERENCES