'95 대한전기학회 하계학술대회 논문집(1995. 7. 20~22)

Hermitian 행렬의 고유쌍을 계산하는 효율적인 알고리즘

°전창완*, 김형중*, 이장규*
*서울대학교 제어계측공학과 *강원대학교 제어계측공학과
자동제어특화연구센터

Efficient Algorithms for Computing Eigenpairs of Hermitian Matrices

°Chang Wan Jeon*, Hyung Jung Kim*, and Jang Gyu Lee*

*Department of Control and Instrumentation Eng., Seoul Nat'l University. Automatic Control Research Center

*Department of Control and Instrumentation Eng., Kwang-won Nat'l University.

Abstract- This paper presents a Generalized Iteration (GI) which includes power method, inverse power method, shifted inverse power method, and Rayleigh quotient iteration (RQI), and modified RQI (MRQI). Furthermore, we propose a GI-based algorithm to find arbitrary eigenpairs for Hermitian matrices. The proposed algorithm appears to be much faster and more accurate than the valuable generalized MRQI of Hu (GMRQI-Hu). The idea of GI is also employed to speed up the GMRQI-Hu and we propose a modified version of Hu's GMRQI (GMRQI-Hu-mod) which is improved in the convergence rate. Some numerical simulation results are presented to confirm our contributions

Key words : eigenvalue, eigenvector, GI, RQI, MRQI, GMRQI, Hermitian matrix

I. INTRODUCTION

Computation of the eigenvalues and eigenvectors of a matrix has received much attention for a long time. It has found important applications in signal processing, control theory, and total least squares problem.

In the past, the focus of research has been on finding the minimum or maximum eigenpair. Power method and inverse power method are regarded as representative methods. If one knows an estimated value of an eigenvalue, the shifted inverse power method is very popular. These methods, however, may be very slow in the convergence rate, since their convergenc rate is linear. Although Rayleigh quotient iteration achieves cubic convergence, it has a critical disadvantage, that is, it cannot give any desired eigenpair unless the starting vector is not properly selected.

Many efficient methods have been reported in the last ten years. They are focused on computing arbitrary eigenpair as well as extreme eigenpair. The method of Cybenko and Van Loan [1], MRQI of Hu and Kung [2], and new MRQI based algorithms of Jeon, Kim, and Lee [3][4] are notable for computing extreme eigenpair. Cybenko and Van Loan proposed a method for computing the smallest eigenvalue of a symmetric positive definite Toeplitz matrix, which achieves a quadratic convergence rate. Hu and Kung presented a modified RQI (MRQI) procedure for computing the smallest eigenvalue of a symmetric positive definite Toeplitz matrix, which achieves asymptotically cubic convergence rate. Recently, Jeon, Kim, and Lee presented a new MRQI procedure for computing the extreme eigenpair for Hermitian mareices, which also achieves an asymptotically cubic convergence rate. For any desired eigenpair, Hu [5] generalized his MRQI procedure as GMRQI. Trench [6] proposed a differrent method as a generalization of Cybenko and Van Loan. Noor and Morgera [7] also presented a modified method based on the works of Hu and Trench. In this paper, we will propose a generalized iteration (GI) which includes the MRQI of Jeon, Kim, and Lee as well as the traditional methods such as power method, inverse power method, shifted inverse power method, and Rayleigh quotient iteration (RQI). As a next step, we will also propose a more efficient method featuring an asymptotically cubic convergence rate, which computes any desired eigenpairs for Hermitian matrices. This should be considered to be a continuation of the previous work [4]. Furthermore, we will propose a modified version of Hu's GMRQI which is more efficient than the convensional GMRQI. The modification is based on the idea of GI.

II. RELATED WORK

Among various related works mentioned in the introduction, the GMRQI-Hu is notable for computing any subset of eigenpairs for Hermitian matrices by the following reasons.

1) His method does not require any rational function. Root finding from the rational function may be time-consuming and numerically unstable.

2) His method achieves asymptotically cubic convergence rate.

3) His method is applicable to matrices which are Hermitian as well as Toeplitz.

Thus, we review his method briefly. The review is necessary for comparison with the methods in this paper. Consider an $n \times n$ Hermitian positive definite and nondefective matrix A. Assume that the eigenvalues of the matrix are distinct. The GMRQI-Hu procedure can be decomposed into two phases: a course search phase and a zoom-in phase.

In the coarse search phase, the objective is to find an inclusion interval for each desired eigenvalue, denoted by $\left[b_{l_p}, b_{u_p}\right]$ for p-th eigenvalue. Such inclusion intervals are obtained from the LDU factorization of the matrix $A-\mu_k I$

$$A-\mu_k I = LDL^* \tag{1}$$

where L is a lower triangular matrix with 1's along their diagonals, $D=diag[d_1 \quad d_2 \quad \cdots \quad d_n]$ is a diagonal matrix, and L^* is a complex conjugate transpose of L. Since the number of negative $d_i(\lambda)$'s is defined by an eigenvalue distribution function, $m(\lambda)$, the A matrix has $m(\lambda)$ eigenvalues smaller than λ by Cauchy's eigenvalue interlacing theorem. For p-th eigenvalue, the lower bound and the upper bound of a valid inclusion interval must satisfy $m\left(b_{l_p}\right)=p-1$ and $m\left(b_{u_p}\right)=p$. The coarse search begins with an initial estimate of the inclusion interval. Bisection method is often employed to find the next "origin shift" μ'. That is,

$$x'=\frac{b_{l_p}+b_{u_p}}{2} \tag{2}$$

for p-th eigenpair. Refer to [5] for details. The algorithms proposed in this paper also employed this procedure for coarse search.

In the zoom-in phase, the eigenpair within the inclusion interval is computed using mixed of RQI and bisection method. This is implemented using a modified Rayleigh quotient iteration (MRQI) method. The algorithm requires the solution of (Toeplitz) linear system of equation

$$(A-\mu I)x=u, \quad |u|=1 \tag{3}$$

in each iteration. To solve the equation, $O\left(n^3\right)$ operations are required for general Hermitian matrices, while the $O\left(n^2\right)$ operations are required

for Toeplitz matrices. If the Rayleigh quotient shifts are used through the zoom-in phase, asymptotically, a very effective cubic convergence rate can be achieved. However, it is rare for the computed Rayleigh quotient to fall into the inclusion interval in practice. In this case, bisection method will replace the RQI method in the current iteration. For details of Hu's GMRQI algorithm, see [5]. Note that the norm of the vector resulted from the solution of the linear system in the GMRQI-Hu is used as a termination criterion. In his algorithm, 1000 is used as the limit. In our experience, this value have to be chosen differently according to the A matrix to obtain more accurate results. Unfortunately, there is no way to determine the limit optimally. Therefore, this point becomes an important reason for which GMRQI-Hu cannot give an accurate result more often than not.

III. DERIVATION OF THE GENERALIZATION ITERATION

The ordinary power method operates exactly on the principle of a difference equation [8], [9]. The method converges to the largest eigenvalue. The inverse power method is just the power method applied with A^{-1} instead of A. The theory of the inverse power method guarantees convergence to the smallest eigenvalue. The most serious limitation of these methods is slow convergence rate. In order to get around this limitation, 'shift' is employed to accelerate the convergence rate. The shifted inverse power method is it. However, to determine the optimal shift is still a problem. In the symmetric matrix case, the most accurate choice seems to be the Rayleigh quotient. Thus, the Rayleigh quotient iteration becomes a promising method. Although the RQI achieves a cubic convergence rate, it does not guarant convergence to a desired eigenpair. Several modified RQI methods (MRQI) are proposed to overcome the intrinsic problem of the RQI [2]-[4].

In this section, we propose a Generalized Iteration (GI), as a unified approach, which covers all the methods mentioned previously. The following procedure shows the proposed GI. The GI can be regarded as a function with p and α as a parameter.

Generalized Iteration : GI(p,α).

Solve $(A-\alpha\mu I)^p x = u$ for x (x=*orth*(u))

$$\mu(x) = \frac{x^* Ax}{x^* x}$$

It is easy to see that how the GI works as various methods. The relation between GI and various methods are summarized as follows.

$GI(-1,0)$: Power method.

$GI(1,0)$: Inverse power method.

$GI\left(1,\frac{\hat{\mu}}{\mu}\right)$: Shifted inverse power method with $\hat{\mu}$ is an estimate of the eigenvalue.

$GI(1,1)$: Rayleigh quotient iteration.

$GI(1,\alpha)$: Modified Rayleigh quotient iteration [3][4].

Although the idea and structure of the GI may seem simple, it appears to have important consequences for developing more efficient algorithms, in that it becomes more easy to combine various methods. In the subsequent sections, we will show that how the GI can be used to develope more efficient algorithms.

IV. DERIVATION OF A GI-BASED HERMITIAM EIGENSYSTEM SOLVER

In the GMRQI-Hu, *LDU* factorization is called at every iteration. Conventionally, *LDU* factorization of an $n \times n$ matrix requires $O(2n^3/3)$ operations. In this section, we are particularly interested in the reduction of computational burden by proposing a new algorithm (GMRQI-JKL) which reduces the number of calls to the *LDU* factorization drastically. Moreover, we try to calculate more accurately. The algorithm is based on the generalized iteration (GI) described previously.

The algorithm is decomposed into three phases, coarse search phase, move-in phase, and zoom-in phase. The coarse search phase is the same as that of GMRQI.

A. Move-in Phase

In the move-in phase, the starting vector is transferred to the neighborhood of the desired eigenvalue. Ostrowski described well the neighborhood of an eigenpair [10]. In order to perform in $O(n^2)$ operations, GI with $p \geq 2$ and $\alpha = b_l + b_u/2\mu$ is used. This process is the same as shifted inverse power method with $\hat{\mu} = b_l + b_u/2$ except p is not equal to one. The following equation shows convergence to the desired eigenpair. Assume that a Hermitian and nondefective matrix A has a basis of eigenvectors $\{v_1, \cdots, v_n\}$ and that $Av_i = \lambda_i v_i$ for $i = 1, 2, \cdots, n$. If $u_0 = \sum_{i=1}^{n} \beta_i v_i$, then

$$x_k = \gamma_1\gamma_2\cdots\gamma_{k-1}\left(\frac{1}{\prod_{i=0}^{k-1}(\lambda_1-\alpha_i\mu_i)}\beta_1 v_1 + \frac{1}{\prod_{i=0}^{k-1}(\lambda_2-\alpha_i\mu_i)}\beta_2 v_2 + \cdots + \frac{1}{\prod_{i=0}^{k-1}(\lambda_n-\alpha_i\mu_i)}\beta_n v_n\right) \quad (4)$$

If α_i is chosen as above, $\alpha_i\mu_i$ becomes a constant for all k and keeps closest distance to the desired eigenvalue λ_d. Then, x_k becomes increasingly rich in the direction of x_d, the desired eigenvector, as the iterations proceed. Consequently, x_k will converge to x_d. In this process, we have only to solve the linear system

$$(A - \alpha_i\mu_i I)x = u \quad (5)$$

at first and use it in the subsequent iterations. Like this, the reduction to $O(n^2)$ operations can be attained. Note that the system have to be solved by way of *LDU* factorization, which requires $O(2n^3/3)$ operations, at every iteration in Hu's GMRQI. In the move-in phase, p is chosen greater than one. The role of p is to speed up the convergence rate at the cheap expense. To solve the linear system

$$(A - \alpha_i\mu_i I)^p x = u \quad ,(6)$$

$(A-\alpha_i\mu_i I)^{-1}$ is multiplied to u at p times, where only $O(pn^2)$ operations are required additionally.

The additional burden of computations may be neglected compared to the amount of computations required in the iterations which will be proceeded if the convergence rate is not accelerated. This point becomes more effective in the modified method dealt in section IV.

B. Criteria for Successful Movement

It is difficult to define the criterion for a successful movement by a simple measure. However, it is possible to develop an adaptive scheme to check the successful movement. The residual norm

$$\|r\| = \|(A-\mu I)x\|_2 \quad (7)$$

has good properties called minimal residual and stationarity [11]. We have exploited the properties to check the successful movement. Although it is a certain way to utilize the residual, it is also a better choice to use the criterion of Szyld [12] as a supplementary criterion for better performance. The criterion of Szyld, of course, can not be applied directly in our algorithm. His assumption that

$$\eta \leq \min_{\lambda_i \neq \lambda_j} |\lambda_i - \lambda_j|/4 \quad (8)$$

where $(\gamma-\eta, \gamma+\eta)$ is a given inclusion interval, is not satisfied in our case. If we, however, modify the criterion, it is helpful to improve the performance, since it works adaptively. We utilized $k(b_u - b_l)/2$ ($0<k<1$) instead of η in the criterion, that is,

$$\|q_x\| < \frac{k(b_u - b_l)}{2} \quad (9)$$

where $q_x = Ax - \gamma x$ and x is the estimated eigenvector. In our simulation, k is chosen as 0.8. The modified criterion may sometimes fail, since $\|q_x\|$ converges to a larger value than the value on the right. In this case, the degree of convergence of $\|q_x\|$ or residual is used as a back-up criterion. The initial bound of the criterion is chosen by the user. Then, the bound becomes tighter by an adaptive way as the iterations are proceeded. Note that the peak of the residual is monitored to speed up the convergence. This technique is based on the property of MRQI, that is, the MRQI has only one maximum residual norm between two adjacent eigenvalues [4].

C. Zoom-in Phase

In the zoom-in phase, the RQI method is employed to locate the desired eigenpair. The convergence rate of this phase is cubic. Since the movement has been finished, this phase can make the utmost use of the advantage of RQI.

D. Check the Validity of the Result

Finishing the zoom-in phase, we must check the validity of the result. The validity can be checked easily by computing the eigenvalue distribution function $m(\hat{\mu}-\varepsilon), \varepsilon << 1$ and $\hat{\mu}$ is the estimated eigenvalue. In this paper, $\varepsilon = 10^{-10}$ was used. Of course, the result will be valid except some cases such that the desired eigenvalue is very close to either of the adjacent eigenvalues. In this case, we must reduce the bound, which the criteria must satisfy in the move-in phase, by the user defined quantity and then go back to the move-in phase. This process is repeated until valid results are obtained.

Based on the above discussions, we shall summarize below the GI-based algorithm for computing the desired eigenpairs for positive definite Hermitian matrices.

GI-Based Algorithm for computing eigenpairs for Hermitian matrices (GMRQI-JKL).

Input : $n, P_0, P_f \left(1 \le P_0, P_f \le n\right), p$

Initiation : $b_{l_i}, b_{u_i}, C_1, C_2, C_4, x, k$

Step 1- Coarse search phase : Compute the inclusion interval $\left[b_{l_i}, b_{u_i}\right]$ of the eigenvalue λ_i such that $m\left(b_{l_i}\right) = i-1$ and $m\left(b_{u_i}\right) = i$ for $p \le i \le q$ and $1 \le p < q \le n$ using bisection.

For $i = P_0$ To P_f.

$f = \dfrac{b_{l_i} + b_{u_i}}{2}$, $A_{inv} = (A - fI)^{-1}$

Step 2- Move-in phase : Move the starting vector to the neighborhood of the desired eigenpair.

Repeat until *move(i)*=true

(a) $GI(p, f/\mu)$

(b) $\|r\| = \|(A-\mu I)x\|_2$, $\|q_s\| = \|(A-fI)x\|_2$

(c) If one of the following conditions is satisfied, then *move(i)*=true.

i) $\|r\| < C_1$

ii) $\left|\|q_s\|_{past} - \|q_s\|\right| < C_2$

iii) $\|q_s\| < C_3$

iv) flag=1, $\|r\|_{past} > \|r\|$, and $u < u_{past}$

v) flag=0, $\|r\|_{past} > \|r\|$, and $u > u_{past}$

(d) $\|q_s\|_{past} = \|q_s\|, \|r\|_{past} = \|r\|, u_{past} = u$

End Repeat Loop

$x_{past} = x$

Step 3- Zoom-in phase : Compute the eigenpairs within the inclusion intervals.

Repeat until *zoom(i)*=true

(a) $GI(1,1)$

(b) $\|r\| = \|(A-\mu I)x\|_2$

(c) If $\|r\| < C_4$, then *zoom(i)*=true

End Repeat Loop

Step 4. Check the validity of the results.

(a) Find $m(\mu-\varepsilon)$, $\varepsilon << 1 (\varepsilon = 10^{-10}$ in this paper)

(b) Reduce C_1 and C_3. $r_{past} = 0, x = x_{past}$

(c) If m=i, then flag=1 and goto move-in phase.

If m=i-2, then flag=0, set $C_2 = a, (a < 0)$, and goto move-in phase.

Next *i*

V. MODIFIED GMRQI ALGORITHM

The zoom-in phase of GMRQI-Hu is composed of bisection and RQI step. If Hu's algorithm is carefully compared with our algorithm, some similarities may be discovered. The bisection step of Hu's algorithm plays the similar role of move-in phase in our algorithm, while the RQI step plays a role of zoom-in phase. Therefore the idea of GI can be applied to the bisection step to improve the convergence. The idea is to replace the linear system equation to be solved $(A-\mu I)x = u$ with $(A-\mu I)^p x = u$. The constant p is chosen as p=1 in the RQI step and as $p \ge 2$ in the bisection step. The additional burden of computations is $O(n^2)$ if $(A-\mu I)^{-1}$ is multiplied to the vector on the right recursively. The additional operations are much less than the operations, $O(n^3)$, required in the iterations to be continued if the algorithm is not modified. The modified algorithm is summarized as below. The underlined parts of the algorithm are modified.

Modified GMRQI Algorithm (GMRQI-Hu-mod)

Step 1- Coarse search phase : Compute the inclusion interval $\left[b_{l_i}, b_{u_i}\right]$ of the eigenvalue λ_i such that $m\left(b_{l_i}\right) = i-1$ and $m\left(b_{u_i}\right) = i$ for $p \le i \le q$ and $1 \le p < q \le n$ using bisection.

Step 2- Zoom-in phase : Compute the eigenpairs within the inclusion intervals, respectively.

For *i=p* To *q*.

Repeat until *convergent(i)*=true

a) Solve $(A-\mu I)^p x = u$ for x

if $a = |x|^2 > 1000$ then *convergent(i)*=true.

u=orth(x)

b) Compute the new shift.

$Ray = \dfrac{u^T A u}{u^T u}$.

If $b_{l_i} < Ray < b_{u_i}$

Then $\mu = Ray$ and p=1

Else $\mu = \dfrac{b_{l_i} + b_{u_i}}{2}$, $\underline{p = p_{con}, p_{con} \ge 2}$

c) Compute the new inclusion interval.

A-μI=LDU, Find $m(\mu)$.

If $m \le i-1$, then $b_{l_i} = \mu$

If $m \ge i$, then $b_{u_i} = \mu$

d) Check the termination criterion. If satisfy, *convergent(i)*=true.

Next *i*

End

VI. SIMULATION RESULTS

This numerical simulation is intended to show the performance of three algorithms, the GMRQI-Hu, GMRQI-Hu-mod, and GMRQI-JKL. A numerical simulation is performed using MATLAB. We construct 50 positive-definite Hermitian matrices of order 10×10, 20×20, 30×30, and 50×50, respectively. We distribute eigenvalues over [0, 10000] randomly. Starting vectors for GMRQIs are given randomly. For 50 matrices of respective order, we compute the average number of iterations, average number of flops (floating-point operations), and average absolute error. Iterations are terminated when $\|(A-\mu I)x\|_2 < 10^{-8}$. We choose the initial lower bound of each inclusion interval by 0 because the matrix is assumed positive-definite. The initial upper bound is given by the maximum Gershgorin radius due to the Gershgorin's circle theorem. A variable in GMRQI-Hu-mod, p_{con} by 3. Other parameters appeared in GMRQI-Hu and GMRQI-Hu-mod are the same in Hu [5]. In GMRQI-JKL, four variables are specified as follows: $C_1 = C_2 = 0.1$, $k = 0.8$, $p = 2$. Variables around the above nominal values slightly affect the performance.

Table I shows the computational complexity of each algorithm.

The three algorithms are all computing the inclusion interval at the initial phase. "No.itf" corresponds to the total number of iterations required to obtain all the inclusion intervals. Number of "No.itf" phases are needed until the satisfactory inclusion interval is identified. "Ray.it" and "Bi.it" correspond to the number of iterations required in the RQI step and bisection step, respectively, in GMRQI-Hu and modified GMRQI-Hu-mod method. "Move-in" and "Zoom-in" correspond to the number of iterations required in the move-in phase and zoom-in phase in GMRQI-JKL, respectively. Note that the move-in phase in GMRQI-JKL is far less computation-intensive than other operations. Since the GMRQI-JKL adopts the form of shifted inverse iteration, an inverse matrix needed in the move-in phase is computed in the initial stage computing the inclusion interval. Thus, simply matrix-vector multiplication rather than solving linear equation is required in the move-in phase. Zoom-in phase also less computation-intensive by a half than operations in GMRQI-Hu and GMRQI-Hu-mod. In Table II, we can observe that the required iterations in the GMRQI-Hu-mod are much less than those of GMRQI-Hu. Especially, improvements in the bisection step is eye-opening. The modification might seem minor, it appeared to have important consequences for efficiency. It is natural that the overall number of operations of GMRQI-Hu-mod is far less than that of GMRQI-Hu. It is due to the modification. Moreover, the two methods are basically akin. However, those figures of GMRQI-JKL are far higher than those of GMRQI-Hu. These figures may mislead the readers. In this sense, Table III is provided. Even though the number of zoom-in operations in GMRQI-JKL are larger than that of bisection operations in GMRQI-Hu, the former operation is far less computation-intensive as illustrated in Table I. Thus, compare the number of flops. Then, it is clear why we claim the GMRQI-JKL to be fast. The GMRQI-Hu takes 314 Mflops while the GMRQI-JKL takes only a fifth part. Of course, the required number of flops in computing the inclusion interval is excluded in the table, since it is common in the three algorithms. In Figure 1, the error norm of the algorithms is illustrated. From (a)-(d), we see that the GMRQI-JKL results in much better accuracy than the other methods. The GMRQI-Hu-mod is slightly better than GMRQI-Hu.

VII. CONCLUSIONS

In this paper, the Generalized Iteration has been proposed. Furthermore, based on the GI, an efficient algorithm for solving the Hermitian eigensystem, called the GMRQI-JKL, has been described. Through the fair simulation, we showed a better performance of the proposed algorithm than existing algorithms. Locating the eigenpair is to be studied when a matrix has an eigenvalue(s) having a multiplicity of one or more.

REFERENCES

[1] G. Cybenko and C. Van. Loan, "Computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix," *SIAM J. Sci. Stat. Comput.*, vol. 7, 1986, pp. 123-131.

[2] Y. H. Hu and S. Y. Kung, "Toeplitz eigensystem solver," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 4, Oct. 1985, pp. 1264-1271.

[3] C. W. Jeon, H. J. Kim, and J. G. Lee, "Improvement of the Rayleigh quotient iteration method," *Proc. of the Korean Institute of Electrical Engineers Conference*, Seoul, Korea, Nov. 18, 1994, pp. 319-321

[4] C. W. Jeon, H. J. Kim, and J. G. Lee, "A New Extreme Eigensystem Solver," *Proc. of the Control and Instrumentation Engineering Conference*, Chuncheon, Korea, April 29, 1995, pp. 131-135

[5] Y. H. Hu, "Parallel eigenvalue decomposition for Toeplitz and related matrices," in *Proc. ICASSP '89*, Glasgow, Scotland, pp. 1107-1110.

[6] W. F. Trench, " Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices, " *SIAM J. Matrix Anal. Appl.*, vol. 10, no. 2, Apr. 1989, pp. 135-146.

[7] F. Noor and S. D. Morgera, " Recursive and iterative algorithms for computing eigenvalues of Hermitian Toeplitz matrices," *IEEE Trans. Signal Processing*, vol. 41, no.3, March 1993, pp. 1272-1280.

[8] G. Strang, *Linear Algebra and Its Applications*, Harcourt Brace Jovanovich, Inc., 3rd Edition, 1988.

[9] E. H. Golub and C. Van Loan, *Matrix Computations*, Baltimore, MD: John Hopkins, University Press, 1983, pp. 305-312.

[10] Ostrowski, " On the convergence of the Rayleigh quotient iteration for the computation of characteristic roots and vectors. I-IV," *Arch. Rational Mech. Anal.*, vol. 1-4, 1958-1959, pp. 233-241, 423-428, 325-340, 341-347, 472-481, 153-165.

[11] N. Parlett, " The Rayleigh quotient iteration and some generalizations for nonnormal matrices," *Math. Comp.* 28, pp.679-693, 1974.

[12] D. B. Szyld, "Criteria for Combing Inverse and Rayleigh Quotient Iteration," *SIAM J. Numer. Anal.*, vol. 25, no. 6, Dec. 1988, pp. 1369-1375.

Table I
Comparison of the Complexity of the algorithms, GMRQI-Hu, GMRQI-Hu-mod, and GMRQI-JKL

	GMRQI-Hu			GMRQI-Hu-mod			GMRQI-JKL		
	No.itf	Ray.it	Bi.it	No.itf	Ray.it	Bi.it	No.itf	Move-in	Zoom-in
Complexity	$O\left(\frac{2}{3}n^3\right)$	$O\left(\frac{2}{3}n^3\right)$	$O\left(\frac{2}{3}n^3\right)$	$O\left(\frac{2}{3}n^3\right)$	$O\left(\frac{2}{3}n^3\right)$	$O\left(\frac{2}{3}n^3\right)$	$O\left(\frac{2}{3}n^3\right)$	$O(n^3)$	$O\left(\frac{1}{3}n^3\right)$

Table II
Comparison of the Required Average Number of Iterations for Matrices with Various Order.
The values are averaged during 50 trials for each matrix order

	GMRQI-Hu				GMRQI-Hu-mod				GMRQI-JKL			
Order	No.itf	Ray.it	Bi.it	Overall	No.itf	Ray.it	Bi.it	Overall	No.itf	Move-in	Zoom-in	Overall
10	51.80	23.52	27.52	102.84	51.80	19.26	12.00	83.06	51.80	34.42	32.24	118.46
20	133.82	45.06	57.00	235.88	133.82	37.36	24.30	195.48	133.92	70.38	64.54	268.74
30	230.52	65.98	91.86	388.36	230.52	56.66	38.08	325.20	230.52	113.90	100.10	444.52
50	440.86	106.18	154.04	701.08	440.86	91.60	62.58	595.04	440.86	215.74	172.76	829.36

Table III
Comparison of the Required Average Number of Flops for Matrices with Various Order.
The values are averaged during 50 trials for each matrix order

Order	GMRQI-Hu	GMRQI-Hu-mod	GMRQI-JKL
10	0.6 M	0.4 M	0.2 M
20	8.6 M	6.0 M	2.0 M
30	42.8 M	30.0 M	8.6 M
50	313.5 M	218.4 M	57.8 M

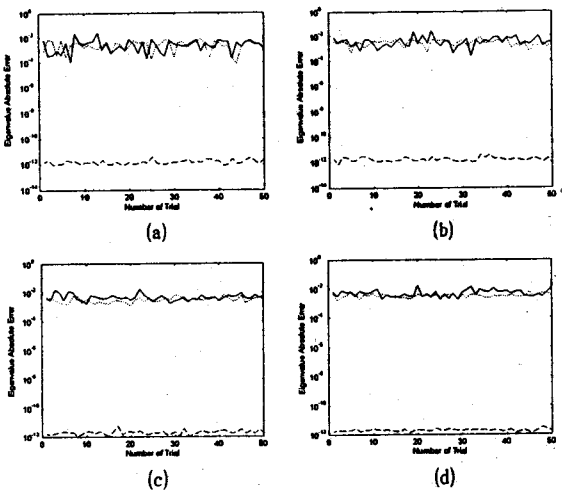

Figure 1. Error performance of the methods for several matrix order. (a) Error performance : matrices of order n=10. (b) Error performance : matrices of order n=20. (c) Error performance : matrices of order n=30. (d) Error performance : matrices of order n=50.GMRQI-Hu(solid), GMRQI-Hu-mod(dotted), GMRQI-JKL(dashed)