BIT 0006-3835/00/4004-0001 $15.00
2000, Vol. 40, No. 4, pp. 001–004

THE INEXACT NEWTON-LIKE METHOD FOR INVERSE EIGENVALUE PROBLEM*

R. H. CHAN[1], H. L. CHUNG[1], and S.-F. XU[2] †

[1] *Department of Mathematics, Chinese University of Hong Kong, Shatin, Hong Kong. email: rchan@math.cuhk.edu.hk*

[2] *School of Mathematical Sciences, Peking University, Beijing, 100871, People's Republic of China. email: xsf@math.pku.edu.cn*

Abstract.

In this paper, we consider using the inexact Newton-like method for solving inverse eigenvalue problem. This method can minimize the oversolving problem of Newton-like methods and hence improve the efficiency. We give the convergence analysis of the method, and provide numerical tests to illustrate the improvement over Newton-like methods.

AMS subject classification: 65F18, 65F10, 65F15.

Key words: Nonlinear equations, Newton-like method, inverse eigenvalue problem.

1 Introduction

Let $\mathbf{c} = (c_1, c_2, \cdots, c_n)^T \in \mathbb{R}^n$ and $\{A_i\}_{i=1}^n$ be a sequence of real symmetric $n \times n$ matrices. Define

$$A(\mathbf{c}) \equiv \sum_{i=1}^{n} c_i A_i \tag{1.1}$$

and denote its eigenvalues by $\lambda_i(\mathbf{c})$ for $i = 1, 2, \ldots, n$ with the ordering $\lambda_1(\mathbf{c}) \leq \lambda_2(\mathbf{c}) \leq \cdots \leq \lambda_n(\mathbf{c})$. The inverse eigenvalue problem (IEP) is defined as follows: For n given real numbers $\{\lambda_i^*\}_{i=1}^n$ where $\lambda_1^* \leq \cdots \leq \lambda_n^*$, find a vector $\mathbf{c}^* \in \mathbb{R}^n$ such that $\lambda_i(\mathbf{c}^*) = \lambda_i^*$ for $i = 1, \ldots, n$. Our goal in this paper is to derive an efficient algorithm for solving the IEP especially when n is large. In Friedland, Nocedal, and Overton [4], there are two examples where we may need to solve IEP with large n: the inverse Sturm-Liouville problem where n is the number of grid points and in nuclear spectroscopy where n is the number of measurements.

In [4], the IEP is solved by applying a Newton-like method, where in each Newton iteration (the outer iteration), we need to solve two linear systems: (i) the inverse power method to find the approximate eigenvectors of the current iterate,

*Received July 1997. Revised September 2000. Communicated by Lars Eldén.

†The work of the first author was partially supported by the HK Research Grant Council grant CUHK4212/99P and CUHK DAG 2060183, and the work of the third author was partially supported by the Natural Science Foundation of China under Grant 19971007.

and (ii) to solve the approximate Jacobian equation. When n is large, the inversions are costly, and one may employ iterative methods to solve both systems (the inner iterations). Although iterative methods can reduce the complexity, it may *oversolve* the systems in the sense that the last few inner iterations before convergence may not improve the convergence of the outer Newton iteration, see [3]. The inexact Newton-like method is a method that stops the inner iteration before convergence. By choosing suitable stopping criteria, we can minimize the total cost of the whole inner-outer iteration.

In this paper, we give an inexact Newton-like method for solving the IEP. We show that our method converges superlinearly. In effect, we have shown that of the two inner iterations, the inverse power method to find the approximate eigenvectors can be solved very roughly, and it will not affect the convergence rate of the outer iteration. However, the accuracy of the second inner iteration, i.e. the solution to the approximate Jacobian equation, is the crucial one in governing the convergence of the outer iteration.

This paper is organized as follows. In §2, we recall the Newton-like methods for solving the IEP. In §3, we introduce our inexact Newton-like method. The convergence analysis is given in §4 and we present our numerical results in §5.

2 The Newton-Like Method

In this section, we briefly recall the Newton and Newton-like methods for solving the IEP. For details, see [4]. For any $\mathbf{c} = (c_1, \ldots, c_n)^T \in \mathbb{R}^n$, define $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\mathbf{f}(\mathbf{c}) = (\lambda_1(\mathbf{c}) - \lambda_1^*, \cdots, \lambda_n(\mathbf{c}) - \lambda_n^*)^T, \tag{2.1}$$

where $\lambda_i(\mathbf{c})$ are the eigenvalues of $A(\mathbf{c})$ defined in (1.1) and λ_i^* are the given eigenvalues. Clearly, $\mathbf{c}^*$ is a solution to the IEP if and only if $\mathbf{f}(\mathbf{c}^*) = \mathbf{0}$. Therefore, we can formulate the IEP as a system of nonlinear equations $\mathbf{f}(\mathbf{c}) = \mathbf{0}$.

As in [4], we assume that the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ are distinct. Then the eigenvalues of $A(\mathbf{c})$ are distinct too in some neighborhood of $\mathbf{c}^*$. It follows that the function $\mathbf{f}(\mathbf{c})$ is analytic in the same neighborhood and that the Jacobian of $\mathbf{f}$ is given by

$$\Big[J(\mathbf{c})\Big]_{ij} = \frac{\partial[\mathbf{f}(\mathbf{c})]_i}{\partial c_j} = \frac{\partial \lambda_i(\mathbf{c})}{\partial c_j} = \mathbf{q}_i(\mathbf{c})^T \frac{\partial A(\mathbf{c})}{\partial c_j}\mathbf{q}_i(\mathbf{c}), \quad 1 \le i, j \le n,$$

where $\mathbf{q}_i(\mathbf{c})$ are the normalized eigenvectors of $A(\mathbf{c})$ corresponding to the eigenvalues $\lambda_i(\mathbf{c})$, see [7, Eq. (4.6.2)] or [2]. Hence by (1.1),

$$\Big[J(\mathbf{c})\Big]_{ij} = \mathbf{q}_i(\mathbf{c})^T A_j \mathbf{q}_i(\mathbf{c}), \quad 1 \le i, j \le n. \tag{2.2}$$

Thus by (1.1) again, we have

$$[J(\mathbf{c})\mathbf{c}]_i = \sum_{j=1}^{n} c_j \mathbf{q}_i(\mathbf{c})^T A_j \mathbf{q}_i(\mathbf{c}) = \mathbf{q}_i(\mathbf{c})^T A(\mathbf{c}) \mathbf{q}_i(\mathbf{c}) = \lambda_i(\mathbf{c}), \quad 1 \le i, j \le n,$$

i.e. $J(\mathbf{c})\mathbf{c} = (\lambda_1(\mathbf{c}), \cdots, \lambda_n(\mathbf{c}))^T$. By (2.1), this becomes

$$J(\mathbf{c})\mathbf{c} = \mathbf{f}(\mathbf{c}) + \boldsymbol{\lambda}^*, \tag{2.3}$$

where $\boldsymbol{\lambda}^* \equiv (\lambda_1^*, \cdots, \lambda_n^*)^T$.

Recall that the Newton method for $\mathbf{f}(\mathbf{c}) = \mathbf{0}$ is given by $J(\mathbf{c}^k)(\mathbf{c}^{k+1} - \mathbf{c}^k) = -\mathbf{f}(\mathbf{c}^k)$. By (2.3), this can be rewritten as

$$J(\mathbf{c}^k)\mathbf{c}^{k+1} = J(\mathbf{c}^k)\mathbf{c}^k - \mathbf{f}(\mathbf{c}^k) = \boldsymbol{\lambda}^*.$$

We emphasize that $J(\mathbf{c}^k)$ is in general a non-symmetric matrix even if all $\{A_j\}_{j=1}^n$ are symmetric. To summarize, we have the following Newton method for solving the IEP, see [4].

Algorithm 1: The Newton Method

For $k = 0$ until convergence, do:

1. *Compute the eigen-decomposition of $A(\mathbf{c}^k)$:*
$$Q(\mathbf{c}^k)^T A(\mathbf{c}^k) Q(\mathbf{c}^k) = \mathrm{diag}(\lambda_1(\mathbf{c}^k), \cdots, \lambda_n(\mathbf{c}^k)),$$
where $Q(\mathbf{c}^k) = [\mathbf{q}_1(\mathbf{c}^k), \cdots, \mathbf{q}_n(\mathbf{c}^k)]$ is orthogonal.
2. *Form the Jacobian matrix: $[J(\mathbf{c}^k)]_{ij} = \mathbf{q}_i(\mathbf{c}^k)^T A_j \mathbf{q}_i(\mathbf{c}^k)$.*
3. *Solve $\mathbf{c}^{k+1}$ from the Jacobian equation: $J(\mathbf{c}^k)\mathbf{c}^{k+1} = \boldsymbol{\lambda}^*$.*

This method converges quadratically, see for instance [7, Theorem 4.6.1]. Notice that in Step 1, we have to compute all the eigenvalues and eigenvectors of $A(\mathbf{c}^k)$ exactly. In [4, 2], it was proven that if we only compute them approximately, we still have the quadratic convergence. This results in the following Newton-like method.

Algorithm 2: The Newton-Like Method

1. *Given $\mathbf{c}^0$, iterate Algorithm 1 once to obtain $\mathbf{c}^1$. In particular, we have $Q(\mathbf{c}^0) = [\mathbf{q}_1^0, \cdots, \mathbf{q}_n^0]$.*
2. *For $k = 1$ until convergence, do:*
   (a) *Compute $\mathbf{v}_i^k$ by the one-step inverse power method:*
   $$\big(A(\mathbf{c}^k) - \lambda_i^* I\big)\mathbf{v}_i^k = \mathbf{q}_i^{k-1}, \quad 1 \le i \le n. \tag{2.4}$$
   (b) *Normalize $\mathbf{v}_i^k$ to obtain an approximate eigenvector $\mathbf{q}_i^k$ of $A(\mathbf{c}^k)$:*
   $$\mathbf{q}_i^k = \frac{\mathbf{v}_i^k}{\|\mathbf{v}_i^k\|}, \quad 1 \le i \le n. \tag{2.5}$$
   (c) *Form the approximate Jacobian matrix: $[J_k]_{ij} = (\mathbf{q}_i^k)^T A_j \mathbf{q}_i^k$.*
   (d) *Solve $\mathbf{c}^{k+1}$ from the approximate Jacobian equation:*
   $$J_k \mathbf{c}^{k+1} = \boldsymbol{\lambda}^*. \tag{2.6}$$

In (2.5) and also in the following, we use $\|\cdot\|$ to denote the 2-norm.

3 The Inexact Newton-Like Method

In deriving the quadratic convergence of Algorithm 2, it was assumed that the systems (2.4) and (2.6) are solved exactly, see [4, 2]. However, if n is large, one may want to solve these systems by iterative methods. In that case, one may even be tempted to solve the systems only approximately to reduce the cost of the inner iterations. It is interesting to know how accurately one has to solve these systems in order to retain the superlinear convergence rate of the whole algorithm, and this is the main thesis of this paper.

For a general nonlinear equation $\mathbf{g}(\mathbf{x}) = \mathbf{0}$, it was shown in [3] that the Jacobian equation

$$J(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = -\mathbf{g}(\mathbf{x}^k) \tag{3.1}$$

need not be solved exactly. In fact, if (3.1) is to be solved by an iterative method, then the last few iterations before convergence are usually insignificant as far as the convergence of the (outer) Newton iteration is concerned. This *oversolving* of the (inner) Jacobian equation will cause a waste of time and does not improve the efficiency of the whole method.

The inexact Newton-like method is derived precisely to avoid the oversolving problem in the inner iterations. Instead of solving (3.1) exactly, one solves it iteratively until a reasonable tolerance is reached. More precisely, one solves for a vector $\tilde{\mathbf{x}}^{k+1}$ such that the residual

$$\mathbf{r}^{k+1} \equiv J(\mathbf{x}^k)(\tilde{\mathbf{x}}^{k+1} - \mathbf{x}^k) + \mathbf{g}(\mathbf{x}^k)$$

satisfies $\|\mathbf{r}^{k+1}\| \le \tau$ for some prescribed tolerance τ. The tolerance is chosen carefully such that it is small enough to guarantee the convergence of the outer iterations, but large enough to reduce the oversolving problem of the inner iterations. For more details, see [3].

Returning to Algorithm 2, there are two inner iterations: (2.4) and (2.6). In order to apply the idea in [3], we have to find suitable tolerances for both equations. We find that the tolerance for (2.4) can be set to any number less than $1/2$ (see (4.26) below); whereas the tolerance for (2.6) has to be of $O(\|\mathbf{c}^k - \mathbf{c}^*\|^\beta)$ in order to have a convergence rate of β for the outer iteration. Finding a computable tolerance for (2.6) is one of the main tasks of the paper. Below we give our algorithm. We set the tolerance for (2.4) to $1/4$ and the tolerance for (2.6) is given in (3.7). We will prove in §4 that the convergence rate of the method is equal to β.

Algorithm 3: The Inexact Newton-Like Method

1. *Given* $\mathbf{c}^0$*, iterate Algorithm 1 once to obtain* $\mathbf{c}^1$*. In particular, we have* $Q(\mathbf{c}^0) = [\mathbf{p}_1^0, \cdots, \mathbf{p}_n^0]$.
2. *For* $k = 1$ *until convergence, do:*

(a) *Solve* $\mathbf{v}_i^k$ *inexactly in the one-step inverse power method*

$$\big(A(\mathbf{c}^k) - \lambda_i^* I\big)\mathbf{v}_i^k = \mathbf{p}_i^{k-1} + \mathbf{t}_i^k, \quad 1 \le i \le n, \tag{3.2}$$

until the residual $\mathbf{t}_i^k$ *satisfies*

$$\|\mathbf{t}_i^k\| \le \frac{1}{4}. \tag{3.3}$$

(b) *Normalize* $\mathbf{v}_i^k$ *to obtain an approximate eigenvector* $\mathbf{p}_i^k$ *of* $A(\mathbf{c}^k)$*:*

$$\mathbf{p}_i^k = \frac{\mathbf{v}_i^k}{\|\mathbf{v}_i^k\|}, \quad 1 \le i \le n. \tag{3.4}$$

(c) *Form the approximate Jacobian matrix:*

$$[J_k]_{ij} = (\mathbf{p}_i^k)^T A_j \mathbf{p}_i^k, \quad 1 \le i, j \le n. \tag{3.5}$$

(d) *Solve* $\mathbf{c}^{k+1}$ *inexactly from the approximate Jacobian equation*

$$J_k \mathbf{c}^{k+1} = \boldsymbol{\lambda}^* + \mathbf{r}^k, \tag{3.6}$$

until the residual $\mathbf{r}^k$ *satisfies*

$$\|\mathbf{r}^k\| \le \Big(\max_{1 \le i \le n} \frac{1}{\|\mathbf{v}_i^k\|}\Big)^{\beta}, \qquad 1 < \beta \le 2. \tag{3.7}$$

Note that the main difference between Algorithm 2 and Algorithm 3 is that we solve (3.2) and (3.6) approximately rather than exactly as in (2.4) and (2.6).

4 Convergence Analysis

In the remaining of this paper, we will use $\mathbf{c}^k$ to denote the kth iterate produced by Algorithm 3, and let $\{\lambda_i(\mathbf{c}^k)\}_{i=1}^n$ and $\{\mathbf{q}_i(\mathbf{c}^k)\}_{i=1}^n$ be the eigenvalues and normalized eigenvectors of $A(\mathbf{c}^k)$, i.e.

(4.1)
$$A(\mathbf{c}^k)\mathbf{q}_i(\mathbf{c}^k) = \lambda_i(\mathbf{c}^k)\mathbf{q}_i(\mathbf{c}^k) \quad \text{and} \quad \mathbf{q}_i(\mathbf{c}^k)^T\mathbf{q}_j(\mathbf{c}^k) = \begin{cases} 0, & 1 \le i \ne j \le n, \\ 1, & 1 \le i = j \le n. \end{cases}$$

As in [4], we assume that the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ are distinct and that the Jacobian $J(\mathbf{c}^*)$ defined in (2.2) is nonsigular at the solution $\mathbf{c}^*$. Under these two assumptions, we prove in this section that if the initial guess $\mathbf{c}^0$ is close to the solution $\mathbf{c}^*$, then the sequence $\{\mathbf{c}^k\}$ converges to $\mathbf{c}^*$ with

$$\|\mathbf{c}^{k+1} - \mathbf{c}^*\| \le \alpha \|\mathbf{c}^k - \mathbf{c}^*\|^{\beta}$$

for a constant α independent of k. Here β is the parameter given in (3.7). We begin by estimating the distance between $\mathbf{p}_i^k$ in (3.2) and $\mathbf{q}_i(\mathbf{c}^k)$.

LEMMA 4.1. *Let* $\mathbf{t}_i^k$ *and* $\mathbf{p}_i^k$ *be as in (3.2). Assume that*

$$|\lambda_j(\mathbf{c}^k) - \lambda_i^*| \geq \gamma > 0, \qquad 1 \leq i \neq j \leq n, \tag{4.2}$$

$$|\mathbf{q}_i(\mathbf{c}^k)^T(\mathbf{p}_i^{k-1} + \mathbf{t}_i^k)| \geq \xi > 0, \qquad 1 \leq i \leq n. \tag{4.3}$$

Then we have

$$\frac{1}{\|\mathbf{v}_i^k\|} \leq \frac{1}{\xi}|\lambda_i(\mathbf{c}^k) - \lambda_i^*|, \quad 1 \leq i \leq n \tag{4.4}$$

and

$$\|\mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^k)\| \leq \frac{2}{\xi\gamma}|\lambda_i(\mathbf{c}^k) - \lambda_i^*|, \quad 1 \leq i \leq n. \tag{4.5}$$

PROOF. Let us prove (4.4) first. Since $\{\mathbf{q}_i(\mathbf{c}^k)\}_{i=1}^n$ is an orthonormal basis, we can write $\mathbf{p}_i^{k-1} + \mathbf{t}_i^k$ as

$$\mathbf{p}_i^{k-1} + \mathbf{t}_i^k = \sum_{j=1}^n \xi_j \mathbf{q}_j(\mathbf{c}^k) \tag{4.6}$$

for some $\xi_j \in \mathbb{R}$, $j = 1, \cdots, n$. By (3.3) and (3.4), we have

$$\sum_{j=1}^n \xi_j^2 = \|\mathbf{p}_i^{k-1} + \mathbf{t}_i^k\|^2 \leq \big(\|\mathbf{p}_i^{k-1}\| + \|\mathbf{t}_i^k\|\big)^2 \leq \big(1 + \frac{1}{4}\big)^2 \leq 2. \tag{4.7}$$

Combining (3.2) with (4.6), we have

$$\mathbf{v}_i^k = (A(\mathbf{c}^k) - \lambda_i^* I)^{-1}(\mathbf{p}_i^{k-1} + \mathbf{t}_i^k) = \sum_{j=1}^n \xi_j \big(A(\mathbf{c}^k) - \lambda_i^* I\big)^{-1}\mathbf{q}_j(\mathbf{c}^k), \quad 1 \leq i \leq n.$$

Clearly $\mathbf{q}_j(\mathbf{c}^k)$ are eigenvectors for $(A(\mathbf{c}^k) - \lambda_i^* I)^{-1}$ with eigenvalues $(\lambda_j(\mathbf{c}^k) - \lambda_i^*)^{-1}$. Hence we have

$$\mathbf{v}_i^k = \sum_{j=1}^n \frac{\xi_j}{\lambda_j(\mathbf{c}^k) - \lambda_i^*}\mathbf{q}_j(\mathbf{c}^k), \quad 1 \leq i \leq n. \tag{4.8}$$

Therefore for $i = 1, \ldots, n$,

$$\begin{aligned}
\frac{1}{\|\mathbf{v}_i^k\|} &= \left(\sum_{j=1}^n \frac{\xi_j^2}{[\lambda_j(\mathbf{c}^k) - \lambda_i^*]^2}\right)^{-\frac{1}{2}} \\
&= \frac{|\lambda_i(\mathbf{c}^k) - \lambda_i^*|}{|\xi_i|}\left(1 + \sum_{j\neq i} \frac{\xi_j^2[\lambda_i(\mathbf{c}^k) - \lambda_i^*]^2}{\xi_i^2[\lambda_j(\mathbf{c}^k) - \lambda_i^*]^2}\right)^{-\frac{1}{2}},
\end{aligned}$$

i.e.

$$\frac{1}{\|\mathbf{v}_i^k\|} \leq \frac{|\lambda_i(\mathbf{c}^k) - \lambda_i^*|}{|\xi_i|}. \tag{4.9}$$

Note that by (4.6), (4.3), and the fact that $\{\mathbf{q}_i(\mathbf{c}^k)\}_{i=1}^n$ are orthonormal, we have

$$|\xi_i| = |\mathbf{q}_i(\mathbf{c}^k)^T(\mathbf{p}_i^{k-1} + \mathbf{t}_i^k)| \geq \xi > 0, \qquad i = 1, \dots, n. \tag{4.10}$$

By putting this into (4.9), we have proved (4.4).

Next we establish (4.5). For $i = 1, \dots, n$, we can write $\mathbf{p}_i^k$ in (3.4) as

$$\mathbf{p}_i^k = \frac{\mathbf{v}_i^k}{\|\mathbf{v}_i^k\|} = \sum_{j=1}^n \beta_j \mathbf{q}_j(\mathbf{c}^k)$$

for some $\beta_j \in \mathbb{R}$, $j = 1, \dots, n$. Using (4.1) and (4.8), we have

$$\beta_i = \mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^k = \frac{1}{\|\mathbf{v}_i^k\|} \mathbf{q}_i(\mathbf{c}^k)^T \sum_{j=1}^n \frac{\xi_j}{\lambda_j(\mathbf{c}^k) - \lambda_i^*} \mathbf{q}_j(\mathbf{c}^k).$$

By (4.9), this becomes

$$\beta_i = \frac{|\lambda_i(\mathbf{c}^k) - \lambda_i^*|}{|\xi_i|} \left(1 + \sum_{j \neq i} \frac{\xi_j^2[\lambda_i(\mathbf{c}^k) - \lambda_i^*]^2}{\xi_i^2[\lambda_j(\mathbf{c}^k) - \lambda_i^*]^2}\right)^{-\frac{1}{2}} \mathbf{q}_i(\mathbf{c}^k)^T \sum_{j=1}^n \frac{\xi_j}{\lambda_j(\mathbf{c}^k) - \lambda_i^*} \mathbf{q}_j(\mathbf{c}^k).$$

Since $\mathbf{q}_i(\mathbf{c}^k)$ are orthonormal, we have

$$|\beta_i| = \left(1 + \sum_{j \neq i} \frac{\xi_j^2[\lambda_i(\mathbf{c}^k) - \lambda_i^*]^2}{\xi_i^2[\lambda_j(\mathbf{c}^k) - \lambda_i^*]^2}\right)^{-\frac{1}{2}} \leq 1.$$

Notice that by (3.5), the entries of the approximate Jacobian matrix J_k are independent of the signs of $\mathbf{p}_i^k$. Therefore, without loss of generality, we may assume that the sign of $\mathbf{p}_i^k$ is such that $\beta_i = \mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^k \geq 0$, i.e.

$$0 \leq \beta_i = \left(1 + \sum_{j \neq i} \frac{\xi_j^2[\lambda_i(\mathbf{c}^k) - \lambda_i^*]^2}{\xi_i^2[\lambda_j(\mathbf{c}^k) - \lambda_i^*]^2}\right)^{-\frac{1}{2}} \leq 1.$$

For any $t \geq 0$, since

$$1 - (1+t)^{-\frac{1}{2}} = \frac{t}{\sqrt{1+t}(1 + \sqrt{1+t})} \leq t,$$

we have

$$1 - \beta_i \leq \sum_{j \neq i} \frac{\xi_j^2[\lambda_i(\mathbf{c}^k) - \lambda_i^*]^2}{\xi_i^2[\lambda_j(\mathbf{c}^k) - \lambda_i^*]^2}.$$

Or by (4.10), we have

$$1-\beta_i \le \frac{[\lambda_i(\mathbf{c}^k)-\lambda_i^*]^2}{\xi_i^2}\sum_{j\neq i}\frac{\xi_j^2}{[\lambda_j(\mathbf{c}^k)-\lambda_i^*]^2} \le \frac{[\lambda_i(\mathbf{c}^k)-\lambda_i^*]^2}{\xi^2}\sum_{j\neq i}\frac{\xi_j^2}{[\lambda_j(\mathbf{c}^k)-\lambda_i^*]^2}.$$

It follows from (4.2) and (4.7) that

$$1-\beta_i \le \frac{[\lambda_i(\mathbf{c}^k)-\lambda_i^*]^2}{\xi^2\gamma^2}\sum_{j\neq i}\xi_j^2 \le \frac{2[\lambda_i(\mathbf{c}^k)-\lambda_i^*]^2}{\xi^2\gamma^2}.$$

Since $\|\mathbf{p}_i^k\| = \|\mathbf{q}_i(\mathbf{c}^k)\| = 1$, we have

$$\begin{aligned} \|\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^k)\|^2 &= \left(\mathbf{p}_i^k\right)^T\mathbf{p}_i^k - 2\mathbf{q}_i(\mathbf{c}^k)^T\mathbf{p}_i^k + \mathbf{q}_i(\mathbf{c}^k)^T\mathbf{q}_i(\mathbf{c}^k) \\ &= 2(1-\beta_i) \le \frac{4}{\xi^2\gamma^2}[\lambda_i(\mathbf{c}^k)-\lambda_i^*]^2. \end{aligned}$$

Hence (4.5) is proved. □

Next we estimate the errors in $\lambda_i(\mathbf{c}^k)$ and $\mathbf{q}_i(\mathbf{c}^k)$. In particular, we show that assumption (4.2) in Lemma 4.1 holds.

LEMMA 4.2. *Let the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ be distinct and $\{\mathbf{q}_i(\mathbf{c}^*)\}_{i=1}^n$ be the normalized eigenvectors of $A(\mathbf{c}^*)$ corresponding to λ_i^*. Then there exist positive numbers δ_0, ρ_0, and γ, such that if $\|\mathbf{c}^k-\mathbf{c}^*\| \le \delta_0$, then*

$$|\lambda_i(\mathbf{c}^k)-\lambda_i^*| \le \rho_0\|\mathbf{c}^k-\mathbf{c}^*\|, \quad 1\le i\le n, \tag{4.11}$$

$$\|\mathbf{q}_i(\mathbf{c}^k)-\mathbf{q}_i(\mathbf{c}^*)\| \le \rho_0\|\mathbf{c}^k-\mathbf{c}^*\|, \quad 1\le i\le n, \tag{4.12}$$

$$|\lambda_i(\mathbf{c}^k)-\lambda_j^*| \ge \gamma > 0, \quad 1\le i\neq j\le n. \tag{4.13}$$

PROOF. This follows from the analyticity of simple eigenvalues and their corresponding eigenvectors, see for instance [7, p.249]. □

By using the continuity of matrix inverses, we can show that the approximate Jacobian matrix J_k of Algorithm 3 (see (3.5)) is nonsingular, provided that the approximate eigenvector $\mathbf{p}_i^k$ is close to $\mathbf{q}_i(\mathbf{c}^*)$.

LEMMA 4.3. *Let the Jacobian $J(\mathbf{c}^*)$ be nonsingular. Then there exist positive numbers δ_1 and ρ_1, such that if $\max_{1\le i\le n}\|\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^*)\| \le \delta_1$, then J_k is nonsingular and*

$$\|J_k^{-1}\| \le \rho_1. \tag{4.14}$$

PROOF. By (3.5) and (2.2), we have

$$|[J_k]_{ij} - [J(\mathbf{c}^*)]_{ij}| = |(\mathbf{p}_i^k)^T A_j\mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)^T A_j\mathbf{q}_i(\mathbf{c}^*)|, \quad 1\le i,j\le n.$$

Hence by the Cauchy-Schwarz inequality

$$\begin{aligned} |[J_k]_{ij} - [J(\mathbf{c}^*)]_{ij}| &= |(\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^*))^T A_j\mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)^T A_j(\mathbf{q}_i(\mathbf{c}^*)-\mathbf{p}_i^k)| \\ &\le \|\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^*)\|\|A_j\|\|\mathbf{p}_i^k\| + \|\mathbf{q}_i(\mathbf{c}^*)\|\|A_j\|\|\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^*)\| \\ &= 2\|A_j\|\|\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^*)\|, \quad 1\le i,j\le n. \end{aligned}$$

In particular, using the Frobenius norm $\|\cdot\|_F$, we have

$$\|J_k - J(\mathbf{c}^*)\| \le \|J_k - J(\mathbf{c}^*)\|_F \le 2n \max_{1\le j\le n} \|A_j\| \max_{1\le i\le n} \|\mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)\|.$$

Since $J(\mathbf{c}^*)$ is nonsingular, by the continuity of matrix inverses, we see that J_k^{-1} exist and $\|J_k^{-1}\|$ are uniformly bounded. □

Next we give an estimate on the error of J_k. For this, we need the following lemma. Recall by (2.3) that $J(\mathbf{c}^*)\mathbf{c}^* = \boldsymbol{\lambda}^*$.

LEMMA 4.4. *Let $\mathbf{w}_i$ be vectors approximating $\mathbf{q}_i(\mathbf{c}^*)$ for $i = 1, \ldots, n$. Define the approximate Jacobian matrix $[J_{\mathbf{w}}]_{ij} = \mathbf{w}_i^T A_j \mathbf{w}_i$ for $1 \le i, j \le n$. Then*

$$\|J_{\mathbf{w}}\mathbf{c}^* - \boldsymbol{\lambda}^*\| \le 2n\|\boldsymbol{\lambda}^*\|_\infty \max_{1\le i\le n} \|\mathbf{w}_i - \mathbf{q}_i(\mathbf{c}^*)\|^2. \tag{4.15}$$

PROOF. Let $W = [\mathbf{w}_1, \cdots, \mathbf{w}_n]$ and $Q(\mathbf{c}^*) = [\mathbf{q}_1(\mathbf{c}^*), \ldots, \mathbf{q}_n(\mathbf{c}^*)]$. Define $\Lambda^* = \mathrm{diag}[\lambda_1^*, \ldots, \lambda_n^*]$ and $E = Q(\mathbf{c}^*)^T W - I$. Then we have

$$\begin{aligned} W^T A(\mathbf{c}^*) W &= W^T Q(\mathbf{c}^*)\Lambda^* Q(\mathbf{c}^*)^T W = (I+E)^T \Lambda^* (I+E) \\ &= \Lambda^* + E^T\Lambda^* + \Lambda^* E + E^T \Lambda^* E. \end{aligned} \tag{4.16}$$

By (1.1), the diagonal entries of $W^T A(\mathbf{c}^*) W$ are given by

$$\begin{aligned} [W^T A(\mathbf{c}^*) W]_{ii} &= \sum_{j=1}^n c_j^* [W^T A_j W]_{ii} \\ &= \sum_{j=1}^n (\mathbf{w}_i^T A_j \mathbf{w}_i) c_j^* = [J_{\mathbf{w}}\mathbf{c}^*]_{ii}, \qquad 1 \le i \le n. \end{aligned}$$

By comparing it with the diagonal entries of (4.16), we see that $J_{\mathbf{w}}\mathbf{c}^* - \boldsymbol{\lambda}^*$ is the vector consisting of diagonal entries of $E_k^T\Lambda^* + \Lambda^* E_k + E_k^T \Lambda^* E_k$. The bound (4.15) on these diagonal entries has already been established for instance in [7, pp. 249–251], see also [2]. □

By applying the lemma to J_k and $\{\mathbf{p}_i^k\}_{i=1}^n$ in (3.5), we have the following corollary.

COROLLARY 4.5. *Let*

$$\mathbf{s}^k \equiv J_k \mathbf{c}^* - \boldsymbol{\lambda}^*. \tag{4.17}$$

Then

$$\|\mathbf{s}^k\| \le 2n\|\boldsymbol{\lambda}^*\|_\infty \max_{1\le i\le n} \|\mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)\|^2. \tag{4.18}$$

According to Lemmas 4.2–4.4, we define

$$\rho = \Big(\frac{8}{\gamma} + 1\Big)\rho_0, \tag{4.19}$$

$$\alpha = \rho_1\Big(2n\|\boldsymbol{\lambda}^*\|_\infty \rho^2 + 4^\beta \rho_0^\beta\Big), \tag{4.20}$$

$$\delta = \min\Big\{1, \delta_0, \delta_1, \frac{1}{\alpha}, \frac{1}{\alpha^{1/(\beta-1)}}, \frac{1}{\rho_0+\rho}\Big\}. \tag{4.21}$$

Now we come to the main lemma of the paper, which gives the convergence rate in terms of $\max_{1\le i\le n}\|\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^*)\|$.

LEMMA 4.6. *Let the given eigenvalues* $\{\lambda_i^*\}_{i=1}^n$ *be distinct and the Jacobian* $J(\mathbf{c}^*)$ *be nonsigular. Then the conditions*

$$\|\mathbf{c}^k-\mathbf{c}^*\|\le\|\mathbf{c}^{k-1}-\mathbf{c}^*\|\le\delta \tag{4.22}$$

and

$$\max_{1\le i\le n}\|\mathbf{p}_i^{k-1}-\mathbf{q}_i(\mathbf{c}^*)\|\le\rho\|\mathbf{c}^{k-1}-\mathbf{c}^*\| \tag{4.23}$$

imply

$$\max_{1\le i\le n}\|\mathbf{p}_i^{k}-\mathbf{q}_i(\mathbf{c}^*)\|\le\rho\|\mathbf{c}^{k}-\mathbf{c}^*\| \tag{4.24}$$

and

$$\|\mathbf{c}^{k+1}-\mathbf{c}^*\|\le\alpha\|\mathbf{c}^k-\mathbf{c}^*\|^\beta. \tag{4.25}$$

PROOF. We begin by proving that (4.3) holds with $\xi=1/4$. Note that

$$\|\mathbf{p}_i^{k-1}-\mathbf{q}_i(\mathbf{c}^k)\|\le\|\mathbf{p}_i^{k-1}-\mathbf{q}_i(\mathbf{c}^*)\|+\|\mathbf{q}_i(\mathbf{c}^*)-\mathbf{q}_i(\mathbf{c}^k)\|.$$

Using (4.23) and (4.12), this becomes

$$\|\mathbf{p}_i^{k-1}-\mathbf{q}_i(\mathbf{c}^k)\|\le\rho\|\mathbf{c}^{k-1}-\mathbf{c}^*\|+\rho_0\|\mathbf{c}^k-\mathbf{c}^*\|.$$

Hence it follows from (4.22) and (4.21) that

$$\|\mathbf{p}_i^{k-1}-\mathbf{q}_i(\mathbf{c}^k)\|\le(\rho+\rho_0)\|\mathbf{c}^{k-1}-\mathbf{c}^*\|\le\frac{\|\mathbf{c}^{k-1}-\mathbf{c}^*\|}{\delta}\le 1.$$

Since $\mathbf{p}_i^k$ and $\mathbf{q}_i(\mathbf{c}^k)$ are normalized vectors, we have

$$\begin{aligned}1\ge\|\mathbf{p}_i^{k-1}-\mathbf{q}_i(\mathbf{c}^k)\|^2 &= \left(\mathbf{p}_i^{k-1}\right)^T\mathbf{p}_i^{k-1}-2\mathbf{q}_i(\mathbf{c}^k)^T\mathbf{p}_i^{k-1}+\mathbf{q}_i(\mathbf{c}^k)^T\mathbf{q}_i(\mathbf{c}^k)\\ &= 2-2\mathbf{q}_i(\mathbf{c}^k)^T\mathbf{p}_i^{k-1}.\end{aligned}$$

Therefore $\mathbf{q}_i(\mathbf{c}^k)^T\mathbf{p}_i^{k-1}\ge 1/2$. Hence by (3.3), we have

$$\mathbf{q}_i(\mathbf{c}^k)^T\left(\mathbf{p}_i^{k-1}+\mathbf{t}_i^k\right)\ge\frac{1}{2}+\mathbf{q}_i(\mathbf{c}^k)^T\mathbf{t}_i^k\ge\frac{1}{2}-\|\mathbf{q}_i(\mathbf{c}^k)\|\|\mathbf{t}_i^k\|\ge\frac{1}{4}. \tag{4.26}$$

Thus (4.3) holds with $\xi=1/4$. Recall that by (4.13), assumption (4.2) also holds. Therefore, by Lemma 4.1, (4.4) and (4.5) are valid with $\xi=1/4$.

Let us now prove (4.24). Note that

$$\|\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^*)\|\le\|\mathbf{p}_i^k-\mathbf{q}_i(\mathbf{c}^k)\|+\|\mathbf{q}_i(\mathbf{c}^k)-\mathbf{q}_i(\mathbf{c}^*)\|,\qquad 1\le i\le n. \tag{4.27}$$

Putting $\xi = 1/4$ in (4.5) and using (4.11), we have

$$\|\mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^k)\| \le \frac{8}{\gamma}|\lambda_i(\mathbf{c}^k) - \lambda_i^*| \le \frac{8}{\gamma}\rho_0\|\mathbf{c}^k - \mathbf{c}^*\|, \quad 1 \le i \le n.$$

By (4.12), we have

$$\|\mathbf{q}_i(\mathbf{c}^k) - \mathbf{q}_i(\mathbf{c}^*)\| \le \rho_0\|\mathbf{c}^k - \mathbf{c}^*\|.$$

Putting the last two inequalities into (4.27) and using (4.19), we have

$$\|\mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)\| \le \frac{8}{\gamma}\rho_0\|\mathbf{c}^k - \mathbf{c}^*\| + \rho_0\|\mathbf{c}^k - \mathbf{c}^*\| = \rho\|\mathbf{c}^k - \mathbf{c}^*\|, \qquad 1 \le i \le n.$$

Thus (4.24) is proved.

Next we prove (4.25). Combining (3.6) with (4.17), we have $J_k(\mathbf{c}^* - \mathbf{c}^{k+1}) = \mathbf{s}^k - \mathbf{r}^k$. Therefore by (4.14), we have

$$\|\mathbf{c}^{k+1} - \mathbf{c}^*\| \le \|J_k^{-1}\|\big(\|\mathbf{s}^k\| + \|\mathbf{r}^k\|\big) \le \rho_1(\|\mathbf{s}^k\| + \|\mathbf{r}^k\|). \tag{4.28}$$

By (4.18) and (4.24), we have $\|\mathbf{s}^k\| \le 2n\|\boldsymbol{\lambda}^*\|_\infty\rho^2\|\mathbf{c}^k - \mathbf{c}^*\|^2$. Putting $\xi = 1/4$ in (4.4) and using (4.11), we get

$$\frac{1}{\|\mathbf{v}_i^k\|} \le 4|\lambda_i(\mathbf{c}^k) - \lambda_i^*| \le 4\rho_0\|\mathbf{c}^k - \mathbf{c}^*\|, \quad 1 \le i \le n.$$

Hence by (3.7), $\|\mathbf{r}^k\| \le 4^\beta\rho_0^\beta\|\mathbf{c}^k - \mathbf{c}^*\|^\beta$. Thus (4.28) becomes

$$\|\mathbf{c}^{k+1} - \mathbf{c}^*\| \le \rho_1\big(2n\|\boldsymbol{\lambda}^*\|_\infty\rho^2\delta^{2-\beta} + 4^\beta\rho_0^\beta\big)\|\mathbf{c}^k - \mathbf{c}^*\|^\beta \le \alpha\|\mathbf{c}^k - \mathbf{c}^*\|^\beta,$$

where the last inequality follows from (4.20) and the fact that $\delta \le 1$. □

With Lemma 4.6, we are in the position to prove the local convergence of Algorithm 3.

THEOREM 4.7. *Let the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ be distinct and the Jacobian $J(\mathbf{c}^*)$ be nonsingular. Then Algorithm 3 is locally convergent with convergence rate β. More precisely, if $\|\mathbf{c}^0 - \mathbf{c}^*\| \le \delta$, then $\mathbf{c}^k$ converges to $\mathbf{c}^*$ with*

$$\|\mathbf{c}^{k+1} - \mathbf{c}^*\| \le \alpha\|\mathbf{c}^k - \mathbf{c}^*\|^\beta, \qquad k = 1, 2, \ldots.$$

Here α and δ are defined in (4.20) and (4.21) respectively.

PROOF. We first prove by mathematical induction that (4.22) and (4.23) are true for all k.

For $k = 1$, we recall that the first step of Algorithm 3 is the same as the first iteration of Algorithm 1. In particular, $\mathbf{p}_i^0 = \mathbf{q}_i(\mathbf{c}^0)$ are the exact eigenvectors of $A(\mathbf{c}^0)$. It follows from (4.12) that

$$\|\mathbf{p}_i^0 - \mathbf{q}_i(\mathbf{c}^*)\| \le \rho_0\|\mathbf{c}^0 - \mathbf{c}^*\| \le \rho\|\mathbf{c}^0 - \mathbf{c}^*\|. \tag{4.29}$$

Hence (4.23) is true for $k = 1$. Also since the Jacobian equation is solved exactly in the first iteration, we have $\|\mathbf{r}^0\| = 0$. From (4.18) and (4.29), $\|\mathbf{s}^0\| \le 2n\|\boldsymbol{\lambda}^*\|_\infty \rho^2 \|\mathbf{c}^0 - \mathbf{c}^*\|^2$. Put these two estimates into (4.28) and we have

$$\|\mathbf{c}^1 - \mathbf{c}^*\| \le \rho_1(2n\|\boldsymbol{\lambda}^*\|_\infty \rho^2)\|\mathbf{c}^0 - \mathbf{c}^*\|^2.$$

It follows from (4.20) and (4.21) that

$$\|\mathbf{c}^1 - \mathbf{c}^*\| \le \alpha\|\mathbf{c}^0 - \mathbf{c}^*\|^2 \le \alpha\delta\|\mathbf{c}^0 - \mathbf{c}^*\| \le \|\mathbf{c}^0 - \mathbf{c}^*\| \le \delta.$$

Thus (4.22) is also valid for $k = 1$.

Next we assume that (4.22) and (4.23) are true for all positive integers less than or equal to k. Then by Lemma 4.6, (4.24) and (4.25) are valid for the same k's. By (4.24), we see that (4.23) is true for $k + 1$. Moreover by (4.25), (4.22) and (4.21), we have

$$\|\mathbf{c}^{k+1} - \mathbf{c}^*\| \le \alpha\|\mathbf{c}^k - \mathbf{c}^*\|^\beta \le \alpha\delta^{\beta-1}\|\mathbf{c}^k - \mathbf{c}^*\| \le \|\mathbf{c}^k - \mathbf{c}^*\| \le \delta,$$

i.e. (4.22) is also true for $k+1$. Thus by mathematical induction, we have proved that (4.22) and (4.23) are true for all positive integers k. Therefore by Lemma 4.6, (4.25) also holds for all positive k, and the theorem is established. □

5 Numerical Experiments

In this section, we compare the convergence rate of Algorithm 2 (the Newton-like method) with that of Algorithm 3 (the inexact Newton-like method). We use Toeplitz matrices as our A_i in (1.1):

$$A_1 = I, A_2 = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}, \cdots, A_n = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & \ddots & \ddots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \ddots & \ddots & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

In particular, $A(\mathbf{c})$ is a symmetric Toeplitz matrix with the first column equals to $\mathbf{c}$. All our tests were done in Matlab.

In the tests, we tried Algorithms 2 and 3 on ten 60-by-60 matrices. For each matrix, we first generate $\mathbf{c}^*$ with entries randomly chosen between 0 and 10. Then we compute the eigenvalues $\{\lambda_i^*\}_{i=1}^n$ of $A(\mathbf{c}^*)$. The initial guess $\mathbf{c}^0$ is formed by chopping the components of $\mathbf{c}^*$ to two decimal places. For both algorithms, the stopping tolerance for the outer (Newton) iterations is 10^{-10}. The inner systems (2.4), (2.6), (3.2), and (3.6) are all solved by the Matlab-provided QMR method [6]. The stopping tolerances for (2.4) and (2.6) is 10^{-13}, and for (3.2) and (3.6), they will be as given in the equations. Besides the tolerances, we also set the maximum number of iterations allowed to 400 for all inner iterations. For the inverse power equations (2.4) and (3.2), we use $\mathbf{v}_i^{k-1}$

β	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	Algo. 2
Iterations	8.3	6.5	5.1	4.8	4.4	4.3	4.3	4.3	4.3	4.3	4.3

Table 5.1: Average numbers of outer iterations for Algorithms 2 and 3.

β	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	Algo. 2
I	36.1	26.5	17.0	15.9	12.8	12.5	12.7	12.9	13.1	13.1	21.8
J	1.02	.873	.746	.735	.701	.690	.717	.729	.737	.741	0.930

Table 5.2: Averaged total numbers of inner iterations in thousands for the inverse power method (I) and for the approximate Jacobian equation (J).

as the initial guess, and for the Jacobian equations (2.6) and (3.6), we use $\mathbf{c}^k$ as the initial guess.

Table 5.1 gives the average numbers of outer iterations for the ten test matrices with respect to different choices of β. For comparison, we also give the average number of outer iterations for Algorithm 2. We see that for β between 1.6 and 2, our method converges at the same rate as Algorithm 2. In Table 5.2, we give the total numbers of inner iterations (averaged over the ten test matrices) required by the inverse power method I (i.e. (2.4) in Algorithm 2 and (3.2) in Algoritm 3) and by the Jacobian equation J (i.e. (2.6) in Algorithm 2 and (3.6) in Algorithm 3). We see that if $\beta \geq 1.3$, then one requires less inner iterations in Algorithm 3 than that in Algorithm 2. The most effective β is around 1.6.

From Tables 5.1–5.2, we can conclude that Algorithm 3 with β between 1.6 and 2 is much better than Algorithm 2. To further illustrate this, we plot in Figure 5.1 the convergence history for one of the test matrices. To illustrate the oversolving problem, we plot the error $\|\mathbf{c}_k - \mathbf{c}^*\|_2$ of the iterates both in the outer iterations and also in the inner iterations with the marks denote the errors at the outer iterations. We can see that our method converges faster than Algorithm 2. Moreover, for Algorithm 2, oversolving problem is very significant (see the horizontal lines between iteration numbers 150 to 300, 430 to 520 and 690 to 800) whereas it is not quite significant for Algorithm 3 with $\beta = 2$ (see the horizontal lines near iteration numbers 430, 600, and 770). However, for $\beta = 1.6$, we only see oversolving at the second outer iteration between iteration numbers 150 to 300 (the same as Algorithm 2 or Algorithm 3 with $\beta = 2$) and there are no oversolving at all for the subsequent iterations.

Finally, we remark that when Algorithm 3 is applied to symmetric Toeplitz matrices $\{A_j\}_{j=1}^n$, (3.2) become Toeplitz systems. For iterative methods, the main cost in each iteration is the matrix-vector multiplications. By embedding the Toeplitz matrices into circulant matrices, the cost per iteration can be reduced from $O(n^2)$ to $O(n \log n)$, see for instance [1]. As for the linear system in (3.6), the cost per iteration can also be reduced. In fact, for any vector

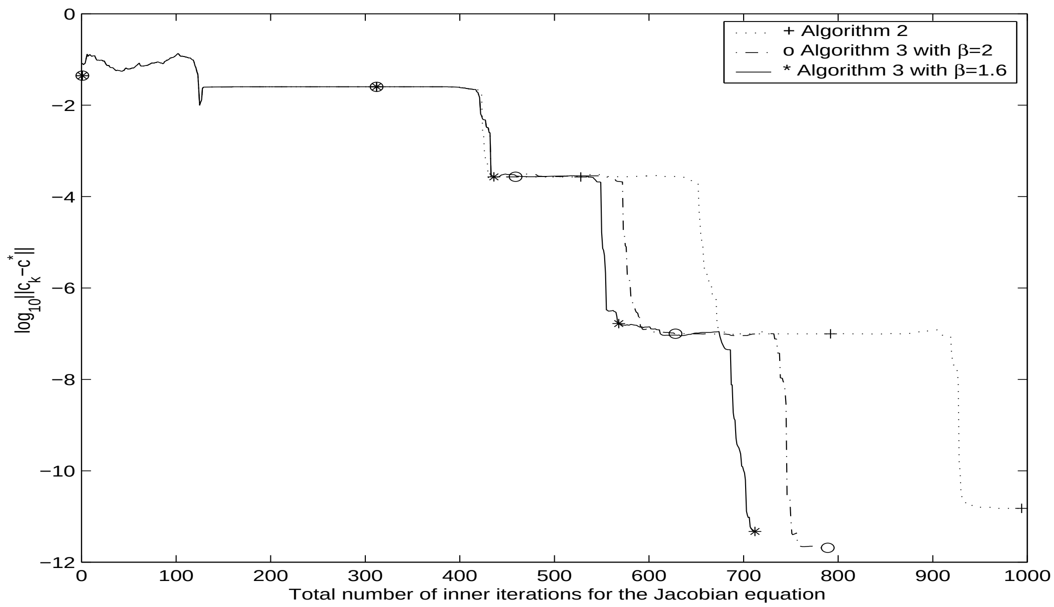

Figure 5.1: Convergence history of one of the test matrices.

$\mathbf{x} = (x_1, \dots, x_n)^T$,

$$[J_k \mathbf{x}]_i = \sum_{j=1}^{n} x_j (\mathbf{p}_i^k)^T A_j \mathbf{p}_i^k = (\mathbf{p}_i^k)^T A(\mathbf{x}) \mathbf{p}_i^k, \quad 1 \le i \le n.$$

Thus it is not necessary to form J_k explicitly and $J_k \mathbf{x}$ can be found via Toeplitz matrix-vector multiplications in $O(n^2 \log n)$ operations instead of $O(n^3)$ operations. As a comparison, the recent algorithm in [8] is essentially a Newton method and only partly utilizes the special structure of Toeplitz matrix; and so in each iteration, its computational cost is $O(n^3)$ operations.

REFERENCES

1. R. Chan and M. Ng, *Conjugate Gradient Method for Toeplitz Systems*, SIAM Review, 38 (1996), 427–482.
2. R. Chan, S. Xu, and H. Zhou, *On the Convergence Rate of a Quasi-Newton Method for Inverse Eigenvalue Problem*, SIAM J. Numer. Anal., 36 (1999), 436–441.
3. S. Eisenstat and H. Walker, *Choosing the Forcing Terms in an Inexact Newton Method*, SIAM J. Sci. Comput., 17 (1996), 16–32.
4. S. Friedland, J. Nocedal, and M. Overton. *The Formulation and Analysis of Numerical Methods for Inverse Eigenvalue Problems*, SIAM J. Numer. Anal., 24 (1987), 634–667.

5. G. Golub and C. van Loan, *Matrix Computations*, 2nd Ed., The John Hopkins University Press, Baltimore, 1989.

6. R. Freund and N. Nachtigal, *QMR: a Quasi-Minimal Residual Method for Non-Hermitian Linear Systems*, Numer. Math., 60 (1991), 315–339.

7. S. Xu, *An Introduction to Inverse Algebraic Eigenvalue Problems*, Peking University Press, 1998.

8. W. Trench, *Numerical solution of the inverse eigenvalue problem for real symmetric Toeplitz matrices*, SIAM J. Sci. Comput., 18 (1997), 1722–1736.