BIT Numerical Mathematics (2007) 47: 27–44
DOI: 10.1007/s10543-006-0100-1
Published online: 15 November 2006 – © Springer 2006

CONVERGENCE OF INEXACT INVERSE ITERATION WITH APPLICATION TO PRECONDITIONED ITERATIVE SOLVES*

M. A. FREITAG[1] and A. SPENCE[1]

[1] *Department of Mathematical Sciences, University of Bath, Claverton Down, BA2 7AY, United Kingdom. email: {m.freitag, a.spence}@maths.bath.ac.uk*

Abstract.

In this paper we study inexact inverse iteration for solving the generalised eigenvalue problem $\mathbf{Ax} = \lambda\mathbf{Mx}$. We show that inexact inverse iteration is a modified Newton method and hence obtain convergence rates for various versions of inexact inverse iteration for the calculation of an algebraically simple eigenvalue. In particular, if the inexact solves are carried out with a tolerance chosen proportional to the eigenvalue residual then quadratic convergence is achieved. We also show how modifying the right hand side in inverse iteration still provides a convergent method, but the rate of convergence will be quadratic only under certain conditions on the right hand side. We discuss the implications of this for the preconditioned iterative solution of the linear systems. Finally we introduce a new ILU preconditioner which is a simple modification to the usual preconditioner, but which has advantages both for the standard form of inverse iteration and for the version with a modified right hand side. Numerical examples are given to illustrate the theoretical results.

AMS subject classification (2000): 65F15, 65F10.

Key words: inverse iteration, Newton's method, preconditioned iterative methods.

1 Introduction.

This paper is about inexact inverse iteration applied to the generalised nonsymmetric eigenvalue problem $\mathbf{Ax} = \lambda\mathbf{Mx}$, where $\mathbf{A}$ and $\mathbf{M}$ are large, sparse (possibly complex) matrices.

Many methods for the iterative calculation of eigenvalues of $\mathbf{Ax} = \lambda\mathbf{Mx}$ involve the repeated solution of the shifted linear system

$$(\mathbf{A} - \sigma\mathbf{M})\mathbf{y} = \mathbf{Mx}, \tag{1.1}$$

for some $\sigma \in \mathbb{C}$, with the simplest iterative method being inverse iteration. We assume that it is impracticable to factorise the matrices, due to excessive memory

* Received February 8, 2005. Accepted in revised form May 28, 2006. Communicated by Axel Ruhe.

requirements, and so to solve the large, sparse, linear shifted systems iterative methods are used. These require only matrix-vector products and can therefore exploit the sparse structure of the matrices.

Previous studies on inexact inverse iteration applied to the standard symmetric eigenvalue problem include the very early paper [16] and the more recent [20, 19] and [2]. In these papers Rayleigh quotient shifts were used in the inverse iteration algorithm and various aspects are discussed, for example, the convergence theory and the implementation and performance of the iterative solver. For incomplete Cholesky preconditioned iterative solves, [19] introduced the idea of modifying the right hand side to improve the performance of the iterative solver. Also in [19] it was noted that Rayleigh quotient iteration can be related to Newton's method on a Grassmann manifold (see [8]). Other related papers on the topic of inexact eigenvalue solvers include [12] and [14]. For inexact inverse iteration applied to the nonsymmetric eigenvalue problem we refer to [13] and [9] for a fixed shift, and [4] for a variable shift strategy. In many of these papers the analysis is based on eigenvector expansions and convergence is determined by looking at a (generalised) tangent of the error in the desired eigendirection. Often, in such accounts the norm of a matrix of all the eigenvectors arises in the convergence analysis and in error bounds, which is a drawback to the approach.

In this paper a completely different and novel approach to the analysis for variable shifts is used which provides a much simpler analysis, and also suggests a way of analysing preconditioned iterative solves when the right hand side is modified as in [19]. We show that inexact inverse iteration is a modified Newton method and hence obtain a convergence analysis for inexact inverse iteration applied to the calculation of an algebraically simple eigenvalue. In addition, the approach here suggests a "tuning" strategy for the preconditioner that works well in numerical examples.

The plan of the paper is as follows. Section 2 contains a review of some known results about Newton's method and inverse iteration. The main theory of the paper is in Section 3 where the convergence results for inexact inverse iteration are obtained. In Section 4 we discuss how to maintain quadratic convergence for a version of inexact inverse iteration where the right-hand side is modified to improve the performance of a preconditioned iterative solver. We illustrate this theory by introducing a "tuned" ILU preconditioner which is a simple rank one modification of the standard preconditioner. This tuned preconditioner turns out to have a significantly improved performance over the standard ILU preconditioner in two different numerical examples. Section 5 provides a summary of the main results in the paper.

Throughout this paper we use $\|\mathbf{z}\| = \|\mathbf{z}\|_\infty$ unless otherwise stated.

2 Inverse iteration.

It is well-known that inverse iteration can be formulated as a Newton method. This was first done in [22] but was then rediscovered in [15] and [21]. Here we revise the convergence theory briefly for a generalised eigenvalue problem.

Let $\mathbf{A}$ and $\mathbf{M}$ be real or complex $n \times n$ matrices, and consider the generalised eigenvalue problem

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{M}\mathbf{x}, \quad \lambda \in \mathbb{C},\ \mathbf{x} \in \mathbb{C}^n. \tag{2.1}$$

Assume that $(\mathbf{x}_1, \lambda_1)$ is an algebraically simple eigenpair of (2.1) with $\mathbf{u}_1^H$ the corresponding left eigenvector, so that,

$$\mathbf{u}_1^H \mathbf{M}\mathbf{x}_1 \neq 0. \tag{2.2}$$

Also, for some non-zero constant vector $\mathbf{c} \in \mathbb{C}^n$ assume the normalisation

$$\mathbf{c}^H \mathbf{x}_1 = 1. \tag{2.3}$$

One version of inverse iteration is given by Algorithm 2.1.

Algorithm 2.1 (Inverse iteration). *Given* $\lambda^{(0)}$ *and* $\mathbf{x}^{(0)}$ *with* $\mathbf{c}^H\mathbf{x}^{(0)} = 1$. *For* $i = 0, 1, 2, \dots$

```
(1) Solve $(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \mathbf{M}\,\mathbf{x}^{(i)}$,
(2) Set $\Delta\lambda^{(i)} = \frac{1}{\mathbf{c}^H\mathbf{y}^{(i)}}$;   $\lambda^{(i+1)} = \lambda^{(i)} + \Delta\lambda^{(i)}$,
(3) Update $\mathbf{x}^{(i+1)} = \Delta\lambda^{(i)}\mathbf{y}^{(i)}$,
(4) Evaluate $\mathbf{r}^{(i+1)} = (\mathbf{A} - \lambda^{(i+1)}\mathbf{M})\mathbf{x}^{(i+1)}$,
(5) Test for convergence.
```

Note that from steps (2) and (3) of Algorithm 2.1, $\mathbf{c}^H\mathbf{x}^{(i+1)} = 1$ and hence $\mathbf{c}^H\Delta\mathbf{x}^{(i)} = 0$, where $\Delta\mathbf{x}^{(i)} = \mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}$.

Now, let us introduce the nonlinear system $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ where, for $\mathbf{z} := (\mathbf{x}, \lambda)^T$,

$$\mathbf{F}(\mathbf{z}) = \begin{bmatrix} (\mathbf{A} - \lambda\mathbf{M})\mathbf{x} \\ \mathbf{c}^H\mathbf{x} - 1 \end{bmatrix}. \tag{2.4}$$

Then the steps in Algorithm 2.1 may be rewritten in the following block matrix form

$$\begin{bmatrix} (\mathbf{A} - \lambda^{(i)}\mathbf{M}) & -\mathbf{M}\mathbf{x}^{(i)} \\ \mathbf{c}^H & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^{(i)} \\ \Delta\lambda^{(i)} \end{bmatrix} = \begin{bmatrix} -(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{x}^{(i)} \\ 0 \end{bmatrix}, \tag{2.5}$$

with

$$\begin{bmatrix} \mathbf{x}^{(i+1)} \\ \lambda^{(i+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(i)} + \Delta\mathbf{x}^{(i)} \\ \lambda^{(i)} + \Delta\lambda^{(i)} \end{bmatrix}. \tag{2.6}$$

Equations (2.5) and (2.6) are merely Newton's method applied to (2.4), namely,

$$\mathbf{J}(\mathbf{z}^{(i)})\Delta\mathbf{z}^{(i)} = -\mathbf{F}(\mathbf{z}^{(i)}), \quad \mathbf{z}^{(i+1)} = \mathbf{z}^{(i)} + \Delta\mathbf{z}^{(i)}, \tag{2.7}$$

where $\mathbf{J}(\mathbf{z}^{(i)})$ denotes the Jacobian

$$\mathbf{J}(\mathbf{z}^{(i)}) = \begin{bmatrix} (\mathbf{A} - \lambda^{(i)}\mathbf{M}) & -\mathbf{M}\mathbf{x}^{(i)} \\ \mathbf{c}^H & 0 \end{bmatrix}. \tag{2.8}$$

LEMMA 2.1. *If* $\mathbf{z}_1 = (\mathbf{x}_1, \lambda_1)^T$ *is an algebraically simple eigenpair of* (2.1) *then under* (2.3), $\mathbf{J}(\mathbf{z}_1)$ *is nonsingular.*

PROOF. Lemma 2.8 in [10] shows that if $\operatorname{Rank}(\mathbf{A} - \lambda_1\mathbf{M}) = n - 1$, and if (2.2), (2.3) hold, then $\mathbf{J}(\mathbf{z}_1)$ is nonsingular. (Note that one can obtain explicit bounds on the norm of $\mathbf{J}(\mathbf{z}_1)^{-1}$ as discussed in [1] and [18], where the bounds depend on $|\mathbf{u}_1^H\mathbf{M}\mathbf{x}_1|^{-1}$.) □

Standard convergence theory for Newton's method (see, for example, [6]) applied to (2.7) provides the following well-known convergence result.

COROLLARY 2.2. *If* $\mathbf{z}_1 = (\mathbf{x}_1, \lambda_1)$ *is an algebraically simple eigenpair of* (2.1) *and if* (2.3) *holds, then Algorithm 2.1 converges quadratically for a close enough starting guess.*

This quadratic rate of convergence is observed in practice (see, for example, the numerical results given by the solid line in Figure 3.1).

For $(\mathbf{x}^{(i)}, \lambda^{(i)})^T$ the eigenvalue residual

$$\mathbf{r}^{(i)} = (\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{x}^{(i)}, \tag{2.9}$$

is calculated in step (4) of Algorithm 2.1. Since $\mathbf{c}^H\mathbf{x}^{(i)} = 1$, $\forall i$, we have $\|\mathbf{r}^{(i)}\| = \|\mathbf{F}(\mathbf{z}^{(i)})\|$. Now with $\mathbf{z}_1 = (\mathbf{x}_1, \lambda_1)$ denoting the root of $\mathbf{F}(\mathbf{z}) = \mathbf{0}$,

$$\|\mathbf{F}(\mathbf{z}^{(i)})\| = \|\mathbf{F}(\mathbf{z}^{(i)}) - \mathbf{F}(\mathbf{z}_1)\| \leq C_1\|\mathbf{z}^{(i)} - \mathbf{z}_1\|,$$

holds, where C_1 is a bound on the norm of $\mathbf{J}(\mathbf{z})$ in some ball centered on $\mathbf{z}_1$. Hence

$$\|\mathbf{r}^{(i)}\| \leq C_1\|\mathbf{z}^{(i)} - \mathbf{z}_1\|. \tag{2.10}$$

It is important to note that, in practice, to stop a Newton iteration one would typically use a relative stopping condition, see, for example, [11, Section 5.2] or [7, Chapter 2] where a clear scaling-invariant account of Newton's method is given.

In this section we have shown how exact inverse iteration can be regarded as a Newton method. In the next section we show how inexact inverse iteration can be regarded as a modified Newton method and hence derive corresponding convergence results.

3 Inexact inverse iteration & modified Newton's method.

Let us now consider a version of inexact inverse iteration that introduces two changes from Algorithm 2.1. First, as the name implies, we solve the linear systems iteratively to a given residual tolerance (and hence the linear systems are solved 'inexactly'). Second, instead of (1.1) we consider the linear system

$$(\mathbf{A} - \sigma\mathbf{M})\mathbf{y} = \mathbf{Z}(\lambda)\mathbf{x}, \tag{3.1}$$

where $\mathbf{Z}(\lambda)$ is a complex $n \times n$ matrix depending on λ. If $\mathbf{Z}(\lambda) = \mathbf{M}$, then (3.1) reduces to (1.1). However, in Section 4 we consider the system $(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \frac{1}{\lambda^{(i)}}\mathbf{P}\mathbf{x}^{(i)}$, where $\mathbf{P}$ is a preconditioner for $\mathbf{A} - \lambda^{(i)}\mathbf{M}$, and so we consider the convergence theory for the more general form given by (3.1). Thus we discuss the following algorithm.

Algorithm 3.1 (Inexact inverse iteration). *Given* $\lambda^{(0)}$ *and* $\mathbf{x}^{(0)}$ *with* $\mathbf{c}^H\mathbf{x}^{(0)} = 1$. *For* $i = 0, 1, 2, \dots$

```
(1) Choose $\tau^{(i)}$,
(2) Solve $(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)}$ inexactly, that is,
        $\|(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} - \mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)}\| \le \tau^{(i)}\|\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)}\|$,
(3) Set $\Delta\lambda^{(i)} = \frac{1}{\mathbf{c}^H\mathbf{y}^{(i)}}$;   $\lambda^{(i+1)} = \lambda^{(i)} + \Delta\lambda^{(i)}$,
(4) Update $\mathbf{x}^{(i+1)} = \Delta\lambda^{(i)}\mathbf{y}^{(i)}$,
(5) Evaluate $\mathbf{r}^{(i+1)} = (\mathbf{A} - \lambda^{(i+1)}\mathbf{M})\mathbf{x}^{(i+1)}$,
(6) Test for convergence.
```

To analyse this algorithm let us introduce the linear system residual

$$\mathbf{res}^{(i)} := (\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} - \mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)}, \tag{3.2}$$

which should not be confused with the eigenvalue residual $\mathbf{r}^{(i)}$ defined by (2.9). From step (2) in Algorithm 3.1 we know

$$\|\mathbf{res}^{(i)}\| \le \tau^{(i)}\|\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)}\|. \tag{3.3}$$

Now, using $\mathbf{x}^{(i+1)} = \Delta\lambda^{(i)}\mathbf{y}^{(i)}$ from step (4) of Algorithm 3.1, we may write (3.2) as

$$(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{x}^{(i+1)} = \Delta\lambda^{(i)}(\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)} + \mathbf{res}^{(i)}),$$

or, equivalently,

$$(\mathbf{A} - \lambda^{(i)}\mathbf{M})\Delta\mathbf{x}^{(i)} - \Delta\lambda^{(i)}(\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)} + \mathbf{res}^{(i)}) = -(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{x}^{(i)}.$$

This equation along with $\mathbf{c}^H\Delta\mathbf{x}^{(i)} = 0$ gives

$$\begin{bmatrix} \mathbf{A} - \lambda^{(i)}\mathbf{M} & -(\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)} + \mathbf{res}^{(i)}) \\ \mathbf{c}^H & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^{(i)} \\ \Delta\lambda^{(i)} \end{bmatrix} = \begin{bmatrix} -(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{x}^{(i)} \\ 0 \end{bmatrix}, \tag{3.4}$$

which, with (2.6), we can write as

$$\widetilde{\mathbf{J}}(\mathbf{z}^{(i)})\Delta\mathbf{z}^{(i)} = -\mathbf{F}(\mathbf{z}^{(i)}), \quad \mathbf{z}^{(i+1)} = \mathbf{z}^{(i)} + \Delta\mathbf{z}^{(i)}, \tag{3.5}$$

where $\mathbf{z}^{(i)} = (\mathbf{x}^{(i)}, \lambda^{(i)})^T$ and $\widetilde{\mathbf{J}}$ is defined by

$$\widetilde{\mathbf{J}}(\mathbf{z}^{(i)}) = \begin{bmatrix} \mathbf{A} - \lambda^{(i)}\mathbf{M} & -(\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)} + \mathbf{res}^{(i)}) \\ \mathbf{c}^H & 0 \end{bmatrix}. \tag{3.6}$$

Clearly (3.5) is a modified Newton method for $\mathbf{F}(\mathbf{z}) = \mathbf{0}$ with $\widetilde{\mathbf{J}}(\mathbf{z}^{(i)})$ being an approximation to the exact Jacobian $\mathbf{J}(\mathbf{z}^{(i)})$ given by (2.8). Hence the convergence of the inexact inverse iteration method given by Algorithm 3.1 can be proved using the convergence theory of modified Newton's method.

Theorem 3.1 (Convergence of inexact inverse iteration). *Let $\mathbf{z}_1 = (\mathbf{x}_1, \lambda_1)^T$ be an algebraically simple eigenpair of* (2.1) *satisfying* (2.3). *Since* $\mathbf{J}(\mathbf{z}_1)$ *defined by* (2.8) *is nonsingular we assume $\|\mathbf{J}(\mathbf{z}_1)^{-1}\| \leq \beta$ (see Lemma 2.1). For some $\tau_{max}, r > 0$, consider the use of Algorithm 3.1 with $\tau^{(i)} \leq \tau_{max}$, $\forall i$, with starting value $\mathbf{z}^{(0)} = (\mathbf{x}^{(0)}, \lambda^{(0)})^T \in \mathcal{B} = \mathcal{B}(\mathbf{z}_1, r)$. If r, τ_{max} and $\mathbf{Z}(\lambda)$ are such that*

$$\beta\{|\lambda_1 - \lambda|\|\mathbf{M}\| + \|\mathbf{Z}(\lambda)\mathbf{x} - \mathbf{M}\mathbf{x}_1\| + \tau_{max}\|\mathbf{Z}(\lambda)\mathbf{x}\|\} =: \delta < 1 \tag{3.7}$$

for $\mathbf{z} = (\mathbf{x}, \lambda) \in \mathcal{B}$, and if

$$\left\{\frac{\beta\|\mathbf{M}\|r}{1-\delta} + \delta\right\} =: \alpha < 1, \tag{3.8}$$

then with $\mathbf{e}^{(i)} := \mathbf{z}^{(i)} - \mathbf{z}_1$,

a) Algorithm 3.1 converges linearly to $\mathbf{z}_1 = (\mathbf{x}_1, \lambda_1)^T$ with

$$\|\mathbf{z}^{(i+1)} - \mathbf{z}_1\| \leq \alpha\|\mathbf{z}^{(i)} - \mathbf{z}_1\|,$$

b) $\mathbf{e}^{(i+1)}$ satisfies

$$\|\mathbf{e}^{(i+1)}\| \leq \frac{\beta}{1-\delta}K^{(i)}\|\mathbf{e}^{(i)}\|, \tag{3.9}$$

where $K^{(i)} = \|\mathbf{M}\|\|\mathbf{e}^{(i)}\| + \|\mathbf{M}\mathbf{x}^{(i)} - \mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)}\| + \|\mathbf{res}^{(i)}\|$, and,

c) if, in addition, $\tau^{(i)}$ in Algorithm 3.1 satisfies

$$\tau^{(i)} \leq C_2\|\mathbf{r}^{(i)}\|, \tag{3.10}$$

for some constant C_2 independent of i with $\mathbf{r}^{(i)}$ given by (2.9), *and*

$$\|\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)} - \mathbf{M}\mathbf{x}^{(i)}\| \leq C_3\|\mathbf{e}^{(i)}\|, \tag{3.11}$$

for some constant C_3 independent of i, then Algorithm 3.1 converges quadratically.

Proof. The proof consists of verifying the conditions of Theorem A.1 on the convergence of modified Newton's method (given in the Appendix) for $\mathbf{F}$ defined by (2.4). First note that γ, the Lipschitz constant of $\mathbf{J}$, can be taken as $2\|\mathbf{M}\|$. Next, by reducing r and τ_{max} and taking $\mathbf{Z}(\lambda)$ close enough to $\mathbf{M}$, conditions (3.7) and (3.8) can always be made to hold. Thus conditions (A.1) and (A.4) of Theorem A.1 hold and so the linear convergence of Algorithm 3.1

(part (a)) is proved. Part (b) of Theorem 3.1 follows immediately from part (b) in Theorem A.1. Under (3.10) and (3.11) (and recalling (2.10)),

$$\|\mathbf{J}(\mathbf{z}^{(i)}) - \widetilde{\mathbf{J}}(\mathbf{z}^{(i)})\| \leq \|\mathbf{M}\mathbf{x}^{(i)} - \mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)}\| + \|\mathbf{res}^{(i)}\| \leq C_4\|\mathbf{e}^{(i)}\|, \tag{3.12}$$

for some constant C_4 independent of i. The quadratic convergence follows from case (c) in Theorem A.1. □

We see from (3.9) that the possibility of achieving quadratic convergence in Algorithm 3.1 is determined by the size of $\|\mathbf{M}\mathbf{x}^{(i)} - \mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)}\|$ and how $\|\mathbf{res}^{(i)}\|$ is controlled. However if $\tau^{(i)}$ is held fixed, or if (3.11) does not hold, then linear convergence is all that can be expected. We now discuss the natural case $\mathbf{Z}(\lambda^{(i)}) = \mathbf{M}$.

3.1 Standard inexact inverse iteration.

In this subsection we consider the standard form of inexact inverse iteration by making the choice $\mathbf{Z}(\lambda^{(i)}) = \mathbf{M}$. In this context we discuss two choices of $\tau^{(i)}$ in Algorithm 3.1, namely, either $\tau^{(i)}$ is chosen to decrease or $\tau^{(i)}$ is held fixed (cases (a) and (b) respectively in the following corollary).

COROLLARY 3.2. *Assume* $\mathbf{Z}(\lambda^{(i)}) = \mathbf{M}$ *in Algorithm 3.1 and let the conditions of Theorem 3.1 hold. Then we obtain the following rates of convergence, depending on the tolerance* $\tau^{(i)}$.

a) Decreasing tolerance. *If* $\tau^{(i)} \leq C_2\|\mathbf{r}^{(i)}\|$ *in step (1) of Algorithm 3.1 then, for a close enough starting guess, Algorithm 3.1 achieves quadratic convergence, which is equal to the rate achieved by Algorithm 2.1 for exact inverse iteration.*

b) Fixed tolerance. *If* $\tau^{(i)} = \tau$ *in step (1) of Algorithm 3.1, where* τ *is fixed but small enough, then, for a close enough starting guess, Algorithm 3.1 converges linearly.*

PROOF. For $\mathbf{Z}(\lambda^{(i)}) = \mathbf{M}$, condition (3.11) in Theorem 3.1 is obviously satisfied with $C_3 = 0$. In the case of a decreasing tolerance (a), condition (3.10) of Theorem 3.1 is assumed and therefore quadratic convergence follows immediately from Theorem 3.1. In the case of a fixed tolerance (b), the bound in (3.3) becomes

$$\|\mathbf{res}^{(i)}\| \leq \tau\|\mathbf{M}\mathbf{x}^{(i)}\|, \tag{3.13}$$

where τ is fixed. Then the $K^{(i)}$ term in (3.9) does not decrease with i and hence only linear convergence can be proved. □

We now present some numerical results to illustrate the theoretical results from Corollary 3.2, and also provide a comparison with Algorithm 2.1 (exact solves).

3.2 Numerical example.

Here we present numerical results to illustrate the convergence behaviour of inexact inverse iteration for two different choices of the solve tolerance in step (2) of Algorithm 3.1.

EXAMPLE 3.1. Consider the standard eigenvalue problem $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ where $\mathbf{A}$ is the finite difference discretisation (central differences) on a 32×32 grid of the following eigenvalue problem of the convection-diffusion operator

$$-\Delta u + 5u_x + 5u_y = \lambda u \quad \text{on} \quad (0,1)^2, \tag{3.14}$$

with homogeneous Dirichlet boundary conditions. Here $\mathbf{Z}(\lambda^{(i)}) = \mathbf{M} = \mathbf{I}$. This eigenvalue problem is also discussed in [9]. Consider finding the smallest eigenvalue ($\lambda_1 \approx 32.18560954$) by Algorithm 2.1 and by Algorithm 3.1 with both decreasing and fixed tolerances. We take an initial vector $\mathbf{x}^{(0)}$ with $\cos(\mathbf{x}_1, \mathbf{x}^{(0)}) \approx 0.84$ and $\lambda^{(0)} = 20$. We use GMRES as the inexact solver: for the inexact solves with fixed tolerance we take $\tau^{(i)} = \tau = 0.3$ (case (b) in Corollary 3.2) and for the inexact solves with decreasing tolerance (case (a) in Corollary 3.2) we take

$$\tau^{(i)} = \min\{\tau, \|\mathbf{r}^{(i)}\|\}, \quad \text{with } \tau = 0.3, \tag{3.15}$$

where the eigenvalue residual $\|\mathbf{r}^{(i)}\|$ is given by (2.9). As an overall stopping condition we use the norm of the relative eigenvalue residual, so that, once

$$\left\| \frac{\mathbf{r}^{(i)}}{\lambda^{(i)}} \right\| < 10^{-10}$$

is satisfied, the computation stops. Note that the computations use $\|\cdot\| = \|\cdot\|_2$, since this is the standard norm used in GMRES.

The results are illustrated in Figure 3.1 which gives logarithmic plots for the norm of the error at step $i+1$ against the norm of the error at step i. The dotted outer lines indicate the slopes expected for linear and quadratic convergence. Clearly, exact inverse iteration specified by the solid line yields quadratic convergence as expected, since it corresponds to Newton's method (see Section 2). Also inexact inverse iteration with decreasing tolerance indicated by the dash-dotted line gives quadratic convergence as expected from Corollary 3.2, part (a). For inexact inverse iteration with fixed tolerance shown by the dashed line we get only linear convergence as predicted in Corollary 3.2, part (b).

If the convection term in the problem is increased in (3.14), a closer starting guess is required since the spectrum of the convection-diffusion operator becomes more bunched, with complex eigenvalues moving close to the desired eigenvalue λ_1.

4 Preconditioned iterative solvers.

In this section we consider the preconditioned iterative solution of the shifted linear systems in inverse iteration. First we show how a modified right hand

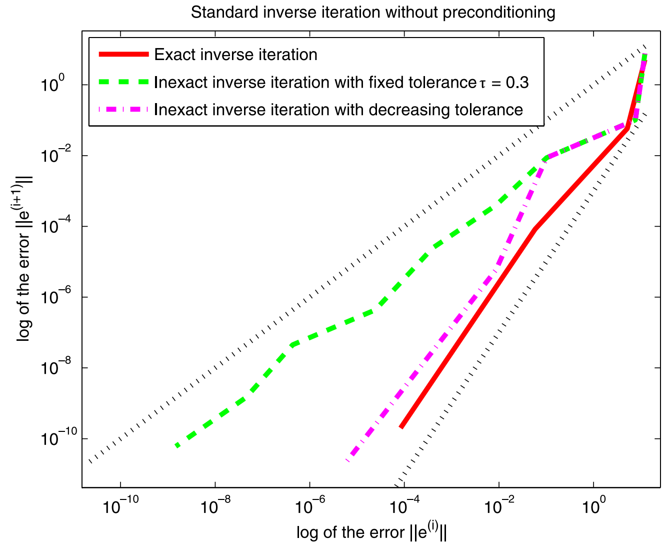

Figure 3.1: Numerical results for Example 3.1. The slopes of the solid, dashed and dashed-dotted lines indicate the rates of convergence achieved. The dotted lines indicate the slopes expected for linear and quadratic convergence.

side as in (3.1) can arise, and then we show how quadratic convergence in Algorithm 3.1 can be maintained by a simple rank one update to the standard preconditioner. Our motivation for choosing a different right hand side arises in the consideration of the performance of the iterative solver used in inexact inverse iteration. It was noted in [19] for the standard symmetric eigenvalue problem that it was advantageous to alter the right hand side in inverse iteration to reduce the number of iterations used by a Krylov solver. We consider this idea applied to the nonsymmetric, generalised case.

The obvious way to implement the (left) preconditioned solution of the shifted system $(\mathbf{A}-\lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \mathbf{M}\mathbf{x}^{(i)}$, with $\mathbf{P}$ a suitable preconditioner, is

$$\mathbf{P}^{-1}(\mathbf{A}-\lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \mathbf{P}^{-1}\mathbf{M}\mathbf{x}^{(i)}. \tag{4.1}$$

However, the idea in [19] is to alter the right hand side of (4.1) to produce a linear system whose solution requires fewer steps of GMRES. For the nonsymmetric eigenvalue problem we argue heuristically as follows. If $(\mathbf{x}^{(i)}, \lambda^{(i)})$ is close enough to $(\mathbf{x}_1, \lambda_1)$, then $\mathbf{A}\mathbf{x}^{(i)} \approx \lambda^{(i)}\mathbf{M}\mathbf{x}^{(i)}$ and so the right hand side of (4.1), namely $\mathbf{P}^{-1}\mathbf{M}\mathbf{x}^{(i)}$, can be approximated by (assuming $\lambda^{(i)} \neq 0$) $\mathbf{P}^{-1}\mathbf{M}\mathbf{x}^{(i)} \approx \frac{1}{\lambda^{(i)}}\mathbf{P}^{-1}\mathbf{A}\mathbf{x}^{(i)}$, and if, in addition, $\mathbf{P}^{-1}\mathbf{A} \approx \mathbf{I}$ then $\mathbf{P}^{-1}\mathbf{M}\mathbf{x}^{(i)} \approx \frac{1}{\lambda^{(i)}}\mathbf{x}^{(i)}$.

Hence, if the preconditioner for $\mathbf{A}-\lambda^{(i)}\mathbf{M}$ is chosen to approximate $\mathbf{A}$, then it is reasonable to replace (4.1) by

$$\mathbf{P}^{-1}(\mathbf{A}-\lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \frac{1}{\lambda^{(i)}}\mathbf{x}^{(i)}. \tag{4.2}$$

Hence, the right hand side vector is roughly in the direction of the approximate null vector of the iteration matrix $\mathbf{P}^{-1}(\mathbf{A}-\lambda^{(i)}\mathbf{M})$ and this helps to reduce costs in GMRES.

A detailed account of the costs of Krylov solvers applied to shifted linear systems in inverse iteration is discussed in [2] (for symmetric problems) and in [3] (for nonsymmetric problems). From the viewpoint of outer convergence theory (4.2) reduces to the equation

$$(\mathbf{A}-\lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \frac{1}{\lambda^{(i)}}\mathbf{P}\mathbf{x}^{(i)}, \tag{4.3}$$

that is, the equation analysed in Section 3 with $\mathbf{Z}(\lambda^{(i)}) = \frac{1}{\lambda^{(i)}}\mathbf{P}$. Due to the factor $\frac{1}{\lambda^{(i)}}$ we introduce an additional assumption that for some C_L, with $C_L > 0$ and independent of i,

$$|\lambda^{(i)}| \geq C_L, \quad \forall i. \tag{4.4}$$

The following corollary provides the key theoretical result, where we allow the preconditioner to depend on i.

Corollary 4.1. *Let $\mathbf{P}^{(i)}$ be a preconditioner for $\mathbf{A}-\lambda^{(i)}\mathbf{M}$, where $\mathbf{P}^{(i)}$ depends on i. Assume that the conditions of Theorem 3.1 are satisfied and that* (4.4) *holds. Let $\tau^{(i)}$ be chosen as in* (3.10). *If $\mathbf{Z}(\lambda^{(i)})$ is chosen as*

$$\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)} = \frac{1}{\lambda^{(i)}}\mathbf{P}^{(i)}\mathbf{x}^{(i)}, \tag{4.5}$$

and $\mathbf{P}^{(i)}$ satisfies

$$\mathbf{P}^{(i)}\mathbf{x}^{(i)} = \mathbf{A}\mathbf{x}^{(i)} \tag{4.6}$$

then Algorithm 3.1 exhibits quadratic convergence.

Proof. Using (4.6) we can write (4.3) as

$$(\mathbf{A}-\lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \frac{1}{\lambda^{(i)}}\mathbf{A}\mathbf{x}^{(i)}.$$

Theorem 3.1 can now be applied with $\mathbf{Z}(\lambda^{(i)}) = \dfrac{1}{\lambda^{(i)}}\mathbf{A}$. We have that

$$\|\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)} - \mathbf{M}\mathbf{x}^{(i)}\| = \left\|\frac{1}{\lambda^{(i)}}(\mathbf{A}\mathbf{x}^{(i)} - \lambda^{(i)}\mathbf{M}\mathbf{x}^{(i)})\right\|, \tag{4.7}$$

and so, using (2.9), (2.10) and (4.4),

$$\|\mathbf{Z}(\lambda^{(i)})\mathbf{x}^{(i)} - \mathbf{M}\mathbf{x}^{(i)}\| \leq \frac{1}{|\lambda^{(i)}|}C_1\|\mathbf{e}^{(i)}\| \leq C_3\|\mathbf{e}^{(i)}\|,$$

where $C_3 := \frac{C_1}{C_L}$. Hence (3.10) and (3.11) in Theorem 3.1 hold, proving that the convergence is quadratic. □

Thus we see from (4.5) and (4.6) that if the right hand side of (4.3) can be made to approximate $\frac{1}{\lambda^{(i)}}\mathbf{A}\mathbf{x}^{(i)}$ then there is the possibility of quadratic outer convergence, with the added advantage of an efficient solution procedure for the shifted linear systems. In the following section we explain how it is possible to satisfy (4.6) and hence to achieve this quadratic convergence rate using an ILU preconditioner.

4.1 Incomplete LU preconditioning and tuning.

In this subsection we consider the use of an incomplete LU factorisation of $\mathbf{A}$ as a preconditioner for $\mathbf{A}-\lambda^{(i)}\mathbf{M}$. This is common in applications involving discretised PDEs, where there is a well-established technology for obtaining a good preconditioner for $\mathbf{A}$ and where $\mathbf{M}$ usually represents a discretised lower order operator. We shall explain how condition (4.6) may be achieved and implemented and then present two numerical examples.

Assume we have the following incomplete factorisation

$$\mathbf{A} = \mathbf{L}\mathbf{U} + \mathbf{E}, \tag{4.8}$$

where $\mathbf{L}$ is a lower triangular matrix and $\mathbf{U}$ is an upper triangular matrix. The matrix $\mathbf{E}$ is the error matrix. We take

$$\mathbf{P}_S := \mathbf{L}\mathbf{U}, \tag{4.9}$$

as the preconditioner for $\mathbf{A}-\lambda^{(i)}\mathbf{M}$. We shall call this the "standard" preconditioner. However, there is no reason why $\mathbf{P}_S$ should satisfy (4.6), but we now show how a simple modification of $\mathbf{P}_S$ can ensure that (4.6) is achieved. We define

$$\mathbf{f}^{(i)} := \mathbf{A}\mathbf{x}^{(i)} - \mathbf{P}_S\mathbf{x}^{(i)} \tag{4.10}$$

for a given $\mathbf{x}^{(i)}$ and introduce the preconditioner

$$\mathbb{P}^{(i)} := \mathbf{P}_S + \mathbf{f}^{(i)}\mathbf{c}^H, \tag{4.11}$$

where $\mathbf{c}^H\mathbf{x}^{(i)} = 1$, with $\mathbf{c}$ being the normalising vector in Algorithm 3.1. Clearly, by construction,

$$\mathbb{P}^{(i)}\mathbf{x}^{(i)} = \mathbf{A}\mathbf{x}^{(i)} \tag{4.12}$$

and (4.6) holds for $\mathbf{P}^{(i)} = \mathbb{P}^{(i)}$. We say that $\mathbb{P}^{(i)}$ is "tuned" in the sense that, as well being a preconditioner in the usual sense, $\mathbb{P}^{(i)}$ agrees with $\mathbf{A}$ in the direction $\mathbf{x}^{(i)}$, the current estimate for $\mathbf{x}_1$. Note that $\mathbb{P}^{(i)}$ is a rank-one change of $\mathbf{P}_S$, and so, using the Sherman-Morrison formula (see, for example, [5, p. 95]), the additional cost of calculating the action of ${\mathbb{P}^{(i)}}^{-1}$ compared with the action of $\mathbf{P}_S^{-1}$ for a given i, is merely one forward and one back substitution. Therefore, in Algorithm 3.1, step (2), we solve

$$(\mathbf{A}-\lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \frac{1}{\lambda^{(i)}}\mathbb{P}^{(i)}\mathbf{x}^{(i)} \tag{4.13}$$

so that $\mathbf{Z}(\lambda^{(i)}) = \frac{1}{\lambda^{(i)}}\mathbb{P}^{(i)}$, with $\mathbb{P}^{(i)}$ given by (4.11), and Algorithm 3.1 should achieve quadratic convergence. Note that (4.13) is implemented as

$$\mathbb{P}^{(i)^{-1}}(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \frac{1}{\lambda^{(i)}}\mathbf{x}^{(i)}. \tag{4.14}$$

We now give two numerical examples to illustrate Corollary 4.1 and also to compare the performance of $\mathbb{P}^{(i)}$ and $\mathbf{P}_S$ as preconditioners for the standard shifted system $(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \mathbf{M}\mathbf{x}^{(i)}$.

Example 4.1. We consider the same convection-diffusion operator with Dirichlet boundary conditions as in Example 3.1, but a generalised eigenvalue problem $\mathbf{A}\mathbf{x} = \lambda\mathbf{M}\mathbf{x}$ is derived by discretising (3.14) using a Galerkin-FEM on regular triangular elements with piecewise linear functions. We use a 32×32 grid leading to 961 degrees of freedom. Again, we seek the smallest eigenvalue, which in this case is given by $\lambda_1 \approx 32.15825765$. As an initial guess we take a vector $\mathbf{x}^{(0)}$ with $\cos(\mathbf{x}_1, \mathbf{x}^{(0)}) \approx 0.79$ and $\lambda^{(0)} = 20$. As solver we take preconditioned GMRES with either the usual ILU preconditioner, $\mathbf{P}_S$ given by (4.9), or the tuned preconditioner, $\mathbb{P}^{(i)}$ given by (4.11). We compare the costs of the following three methods.

a) "$\mathbb{P}^{(i)}$/modified-rhs": the tuned preconditioner is applied to the inverse iteration system with a modified right hand side, namely,

$$\mathbb{P}^{(i)^{-1}}(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \frac{1}{\lambda^{(i)}}\mathbf{x}^{(i)}, \quad \mathbb{P}^{(i)} = \mathbf{L}\mathbf{U} + \mathbf{f}^{(i)}\mathbf{c}^H, \tag{4.15}$$

where $\mathbf{U}$ and $\mathbf{L}$ are given by the ILU decomposition, and with $\mathbf{f}^{(i)}$ defined by (4.10).

b) "$\mathbb{P}^{(i)}$/standard-rhs": the tuned preconditioner is applied to the standard inverse iteration system, namely,

$$\mathbb{P}^{(i)^{-1}}(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \mathbb{P}^{(i)^{-1}}\mathbf{M}\mathbf{x}^{(i)}, \quad \mathbb{P}^{(i)} = \mathbf{L}\mathbf{U} + \mathbf{f}^{(i)}\mathbf{c}^H. \tag{4.16}$$

c) "$\mathbf{P}_S$/standard-rhs": the usual ILU preconditioner is applied to the standard inverse iteration system, namely,

$$\mathbf{P}_S^{-1}(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \mathbf{P}_S^{-1}\mathbf{M}\mathbf{x}^{(i)}, \quad \mathbf{P}_S = \mathbf{L}\mathbf{U}. \tag{4.17}$$

In each case we use the decreasing tolerance $\tau^{(i)} = \min\{\tau, \|\mathbf{r}^{(i)}\|\}$ with $\tau = 0.5$. So all three methods have quadratic convergence using Corollaries 3.2 and 4.1. The iteration stops once the relative residual satisfies $\|\frac{\mathbf{r}^{(i)}}{\lambda^{(i)}}\| < 10^{-14}$. As in Example 3.1, $\|\cdot\| = \|\cdot\|_2$.

In Figure 4.1 we give logarithmic plots of the errors obtained from method "$\mathbb{P}^{(i)}$/modified-rhs" for two different drop tolerances. The dotted lines indicate the slopes expected for linear and quadratic convergence. As predicted in Corollary 4.1 we achieve quadratic convergence. Table 4.1 shows the number of inner iterations for the inexact solves of the three methods. We see that for both drop

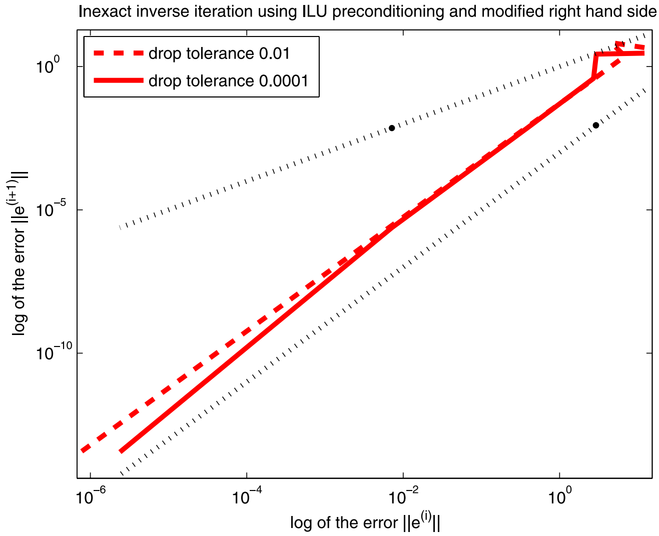

Figure 4.1: Numerical results for Example 4.1. The quadratic outer convergence rate for method "$\mathbb{P}^{(i)}$/modified-rhs" with different drop tolerances is readily observed.

Table 4.1: Iteration numbers for Example 4.1. Total number of iterations and number of inner iterations for the three methods using (4.15), (4.16) or (4.17) with decreasing tolerance. In each method the drop tolerances were 10^{-2} and 10^{-4}.

	"$\mathbb{P}^{(i)}$/modified-rhs"		"$\mathbb{P}^{(i)}$/standard-rhs"		"$\mathbf{P}_S$/standard-rhs"	
OUTER IT.	10^{-2}	10^{-4}	10^{-2}	10^{-4}	10^{-2}	10^{-4}
1	1	1	1	1	1	1
2	3	3	7	5	9	5
3	3	3	11	7	13	8
4	6	5	13	8	18	13
5	8	7	16	8	28	18
6	13	13				
7	18					
total	52	32	48	29	69	45

tolerances the tuned preconditioner applied to the standard inverse iteration formulation, method "$\mathbb{P}^{(i)}$/standard-rhs", requires fewer inner iterations than the other two methods. In particular, comparing the results for "$\mathbb{P}^{(i)}$/standard-rhs" with "$\mathbf{P}_S$/standard-rhs" we see that the tuned preconditioner is significantly better than the usual ILU preconditioner. Method "$\mathbb{P}^{(i)}$/standard-rhs" requires fewer outer iterations than "$\mathbb{P}^{(i)}$/modified-rhs", which may be explained by considering the constants in the convergence theory. In particular, method (a) is sensitive to the starting guess whereas methods (b) and (c) are more robust with respect to the starting vector. For example if we choose a starting vectors with $\cos(\mathbf{x}_1, \mathbf{x}^{(0)}) \approx 0.47$ method (a) fails to work, whereas methods (b) and (c) converge as expected, with (b) again proving superior to (c).

Hence, if the preconditioner is tuned as explained in this section, it appears to be best to apply it on the standard system $(\mathbf{A} - \lambda^{(i)}\mathbf{M})\mathbf{y}^{(i)} = \mathbf{M}\mathbf{x}^{(i)}$ rather than consider modifying the right hand side. This gives the best result in terms of the total number of iterations and the convergence rate.

Next we present an example arising in reactor design (see [17] for details).

Example 4.2. The standard model to describe the neutron balance in a 2D model of a nuclear reactor is given by the two-group neutron equations

$$\begin{aligned} -\mathrm{div}(K_1 \nabla u_1) + (\Sigma_{a,1} + \Sigma_s) u_1 &= \frac{1}{\mu_1}(\Sigma_{f,1} u_1 + \Sigma_{f,2} u_2) \\ -\mathrm{div}(K_2 \nabla u_2) + \Sigma_{a,2} u_1 - \Sigma_s u_2 &= 0, \end{aligned}$$

where u_1 and u_2 are defined on $[0,1] \times [0,1]$ and represent the density distributions of fast and thermic neutrons respectively. K_1 and K_2 are diffusion coefficients and $\Sigma_{a,1}, \Sigma_{a,2}, \Sigma_s, \Sigma_{f,1}$ and $\Sigma_{f,2}$ measure interaction probabilities and take different piecewise constant values in different regions of the reactor, which for this example are given in Figure 4.2 and Table 4.2. The largest μ_1 such that $1/\mu_1$ is an eigenvalue of the system equation is a measure for the criticality of a reactor with $\mu_1 < 1$ representing subcriticality and $\mu_1 > 1$ representing supercriticality. The aim is to maintain the reactor in the critical phase with $\mu_1 = 1$.

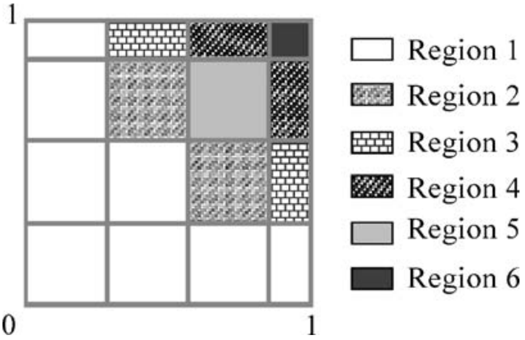

Figure 4.2: Nuclear reactor problem geometry.

Table 4.2: Data for the nuclear reactor problem.

	K_1	K_2	$\Sigma_{a,1}$	$\Sigma_{a,12}$	Σ_s	$\Sigma_{f,1}$	$\Sigma_{f,2}$
Region 1	2.939e-5	1.306e-5	0.0089	0.109	0.0	0.0	0.0079
Region 2	4.245e-5	1.306e-5	0.0105	0.025	0.0	0.0	0.0222
Region 3	4.359e-5	1.394e-5	0.0092	0.093	0.0066	0.140	0.0156
Region 4	4.395e-5	1.355e-5	0.0091	0.083	0.0057	0.109	0.0159
Region 5	4.398e-5	1.355e-5	0.0097	0.098	0.0066	0.124	0.0151
Region 6	4.415e-5	1.345e-5	0.0093	0.085	0.0057	0.107	0.0157

The boundary conditions for $g = 1, 2$ are

$$\begin{aligned} u_g &= 0 \quad \text{if} \quad x_1 = 0 \quad \text{or} \quad x_2 = 0, \\ K_g \frac{\partial u_g}{\partial x_i} &= 0 \quad \text{if} \quad x_i = 1, \quad \text{for} \quad i = 1, 2. \end{aligned}$$

Discretising the problem using a finite difference approximation on a $h \times h$ grid, where $h = 1/m$ we obtain a $2m^2 \times 2m^2$ discrete eigenproblem $\mathbf{Au} = \lambda \mathbf{Mu}$, where $\mathbf{A}$ and $\mathbf{M}$ are both nonsymmetric and $\mathbf{M}$ is singular. We seek the smallest eigenvalue $\lambda_1 (= 1/\mu_1)$, which determines the criticality of the reactor. We choose $m = 32$, which leads to a system of size $n = 2048$. For initial conditions, we take $\lambda^{(0)} = 1$, from physical considerations, and $\mathbf{u}^{(0)} = [1, \ldots, 1]^T/\sqrt{n}$. In fact, the exact eigenvalue is given by $\lambda_1 = 0.9707$ and $\cos(\mathbf{u}_1, \mathbf{u}^{(0)}) \approx 0.44$. We compare methods (a), (b) and (c) as in Example 4.1. In each case the iteration stops once the relative residual satisfies $\|\frac{\mathbf{r}^{(\mathbf{i})}}{\lambda^{(\mathbf{i})}}\| < 10^{-11}$.

Table 4.3 shows the results obtained by methods (a), (b) and (c). Again, we use an ILU preconditioner with, in this case, drop tolerances of 0.1 and 0.01. We observe that the use of the tuned preconditioner applied to the standard formulation (see the middle columns in Table 4.3) provides the best results with respect to overall costs. Also, the standard preconditioner applied to the standard formulation (see the right hand columns) performs least well. These numerical results are consistent with those obtained in the previous example and confirm the usefulness and applicability of the tuned preconditioner.

Table 4.3: Iteration numbers for Example 4.2. Total number of iterations and number of inner iterations for the three methods using (4.15), (4.16) or (4.17) with decreasing tolerance. In each method the drop tolerances were 10^{-1} and 10^{-2}.

	"$\mathbb{P}^{(i)}$/modified-rhs"		"$\mathbb{P}^{(i)}$/standard-rhs"		"$\mathbf{P}_S$/standard-rhs"	
OUTER IT.	10^{-1}	10^{-2}	10^{-1}	10^{-2}	10^{-1}	10^{-2}
1	6	7	4	4	1	4
2	4	10	2	11	8	11
3	12	21	5	22	3	27
4	22	29	27	27	5	37
5	43		38		26	45
6	65		59		38	
7					55	
8					76	
total	152	67	135	64	212	124

5 Conclusions.

We have analysed inexact inverse iteration for a generalised eigenvalue problem and have shown that for an algebraically simple eigenvalue it is a modified

Newton method. Using the convergence theory of the modified Newton method we obtained convergence rates for inexact solves with either fixed or decreasing tolerances. This approach is much simpler than previous approaches involving eigenvector expansions. Using the same tool, we also analysed preconditioned iterative solves. In situations where the right hand side in inexact inverse iteration is modified we have shown how an ILU preconditioner may be tuned to recover quadratic convergence. Additionally, we have given two examples which indicate that the application of the tuned preconditioner may be advantageous when applied to the standard inverse iteration formulation.

Acknowledgements.

The research of Melina Freitag is supported by EPSRC, UK, and the University of Bath. The authors thank Dr Jörg Berns-Müller for providing the matrices in Example 4.1.

A Modified Newton theory.

Here we state a convergence theorem for modified Newton's method (see for example [6]) that is used to prove Theorem 3.1.

THEOREM A.1. *Assume* $\mathbf{F} : \mathbb{C}^n \longrightarrow \mathbb{C}^n$ *and let* $\mathbf{F}(\mathbf{z}^*) = \mathbf{0}$*. For some* $r > 0$*, define* $\mathcal{B} := \mathcal{B}(\mathbf{z}^*, r)$ *and assume* $\mathbf{J}(\mathbf{z}) \in Lip_\gamma \mathcal{B}$*, where* $\mathbf{J}(\mathbf{z})$ *is the Jacobian of* $\mathbf{F}(\mathbf{z})$*. Further, assume* $\|\mathbf{J}(\mathbf{z}^*)^{-1}\| \leq \beta$*. For each* $\mathbf{z}$ *let* $\widetilde{\mathbf{J}}(\mathbf{z})$ *be a complex* $n \times n$ *matrix satisfying, for some* δ*,* $0 \leq \delta < 1$*,*

$$\|\mathbf{J}(\mathbf{z}^*)^{-1}(\widetilde{\mathbf{J}}(\mathbf{z}) - \mathbf{J}(\mathbf{z}^*))\| \leq \delta. \tag{A.1}$$

Then $\widetilde{\mathbf{J}}(\mathbf{z})^{-1}$ *exists in* $\mathcal{B}$ *and* $\|\widetilde{\mathbf{J}}(\mathbf{z})^{-1}\| \leq \dfrac{\beta}{1-\delta}$*. Next consider the solution of*

$$\mathbf{F}(\mathbf{z}) = \mathbf{0}, \quad \mathbf{z} \in \mathbb{C}^n \tag{A.2}$$

using modified *Newton's method:*

$$\mathbf{z}^{(i+1)} = \mathbf{z}^{(i)} - \widetilde{\mathbf{J}}(\mathbf{z}^{(i)})^{-1}\mathbf{F}(\mathbf{z}^{(i)}), \quad \mathbf{z}^{(0)} \in \mathcal{B}. \tag{A.3}$$

If

$$\left\{\frac{\beta\gamma r}{2(1-\delta)} + \delta\right\} =: \alpha < 1, \quad \forall \mathbf{z} \in \mathcal{B} \tag{A.4}$$

then, with $\mathbf{e}^{(i)} := \mathbf{z}^{(i)} - \mathbf{z}^*$*,*

a) modified Newton's method converges linearly *to* $\mathbf{z}^*$*,*
b) we have

$$\|\mathbf{e}^{(i+1)}\| \leq \frac{\beta}{(1-\delta)}\left\{\frac{\gamma}{2}\|\mathbf{e}^{(i)}\| + \|\mathbf{J}(\mathbf{z}^{(i)}) - \widetilde{\mathbf{J}}(\mathbf{z}^{(i)})\|\right\}\|\mathbf{e}^{(i)}\|,$$

and

c) if

$$\|(\mathbf{J}(\mathbf{z}^{(i)}) - \widetilde{\mathbf{J}}(\mathbf{z}^{(i)}))\mathbf{e}^{(i)}\| \leq C\|\mathbf{e}^{(i)}\|^2$$

for some constant C, then modified Newton's method converges quadratically.

REFERENCES

1. E. L. Allgower and H. Schwetlick, *A general view of minimally extended systems for simple bifurcation points*, ZAMM Z. Angew. Math. Mech., 77 (1997), pp. 83–98.
2. J. Berns-Müller, I. G. Graham, and A. Spence, *Inexact inverse iteration for symmetric matrices*, Linear Algebra Appl., 416 (2006), pp. 389–413.
3. J. Berns-Müller and A. Spence, *Inexact inverse iteration and GMRES*, 2006. Submitted.
4. J. Berns-Müller and A. Spence, *Inexact inverse iteration and the nonsymmetric generalized eigenvalue problem*, 2006. To appear in SIAM J. Matrix Anal. Appl.
5. J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
6. J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics in Applied Mathematics, vol. 16, SIAM, Philadelphia, PA, 1996. Unabridged, corrected reprint of the 1983 original.
7. P. Deuflhard, *Newton Methods for Nonlinear Problems*, Springer Series in Computational Mathematics, vol. 35, Springer, Berlin, Heidelberg, 2004.
8. A. Edelman, T. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 303–353.
9. G. H. Golub and Q. Ye, *Inexact inverse iteration for generalized eigenvalue problems*, BIT, 40 (2000), pp. 671–684.
10. H. B. Keller, *Numerical solution of bifurcation and nonlinear eigenvalue problems*, in Applications of Bifurcation Theory, P. H. Rabinowitz, ed., Academic Press, New York, 1977, pp. 359–384.
11. C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in Applied Mathematics, vol. 16, SIAM, Philadelphia, PA, 1995.
12. A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
13. Y.-L. Lai, K.-Y. Lin, and L. Wen-Wei, *An inexact inverse iteration for large sparse eigenvalue problems*, Numer. Linear Algebra Appl., 1 (1997), pp. 1–13.
14. K. Neymeyr, *A geometric theory for preconditioned inverse iteration I: Extrema of the Rayleigh quotient*, Linear Algebra Appl., 322 (2001), pp. 61–85.
15. G. Peters and J. Wilkinson, *Inverse iteration, ill-conditioned equations and Newton's method*, SIAM Rev., 21 (1979), pp. 339–360.
16. A. Ruhe and T. Wiberg, *The method of conjugate gradients used in inverse iteration*, BIT, 12 (1972), pp. 543–554.
17. R. Scheichl, *Parallel Solution of the Transient Multigroup Neutron Diffusion Equations with Multi-Grid and Preconditioned Krylov-Subspace Methods*, Master's thesis, Johannes Kepler Universität Linz, Austria, 1997.
18. H. Schwetlick and R. Lösche, *A generalized Rayleigh quotient iteration for computing simple eigenvalues of nonnormal matrices*, ZAMM Z. Angew. Math. Mech., 80 (2000), pp. 9–25.
19. V. Simoncini and L. Eldén, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, BIT, 42 (2002), pp. 159–182.

20. P. Smit and M. H. C. Paardekooper, *The effects of inexact solvers in algorithms for symmetric eigenvalue problems*, Linear Algebra Appl., 287 (1999), pp. 337–357.

21. H. Symm and J. Wilkinson, *Realistic error bounds for a simple eigenvalue and its associated eigenvector*, Numer. Math., 35 (1980), pp. 113–126.

22. H. Unger, *Nichtlineare Behandlung von Eigenwertaufgaben*, ZAMM Z. Angew. Math. Mech., 30 (1950), pp. 281–282.