Nonlinear CG-like iterative methods *

A.T. Chronopoulos

Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, United States

Received 8 November 1990
Revised 27 March 1991

Abstract


A nonlinear conjugate gradient method has been introduced and analyzed by J.W. Daniel. This method applies to nonlinear operators with symmetric Jacobians. Orthomin(1) is an iterative method which applies to nonsymmetric and definite linear systems. In this article we generalize Orthomin(1) to a method which applies directly to nonlinear operator equations. Each iteration of the new method requires the solution of a scalar nonlinear equation. Under conditions that the Hessian is uniformly bounded away from zero and the Jacobian is uniformly positive definite the new method is proved to converge to a globally unique solution. Error bounds and local convergence results are also obtained. Numerical experiments on solving nonlinear operator equations arising in the discretization of nonlinear elliptic partial differential equations are presented.

Keywords: Nonlinear algebraic systems, iterative methods, Orthomin.

1. Introduction

Nonlinear systems of equations often arise when solving initial- or boundary value problems in ordinary or partial differential equations. We consider the nonlinear system of equations

\[ F(x) = 0, \]  

(1.1)

where \( F(x) \) is a nonlinear operator from a real Euclidean space of dimension \( N \) or Hilbert space into itself. The Newton method coupled with direct linear system solvers is an efficient way to solve such nonlinear systems when the dimension of the Jacobian is small. When the Jacobian is large and sparse, some kind of iterative method may be used. This can be a nonlinear iteration (for example, functional iteration for contractive operators), or an inexact Newton method. In an inexact Newton method the solution of the resulting linear systems is

Correspondence to: Prof. A.T. Chronopoulos, Department of Computer Science, University of Minnesota, 200 Union Street S.E., Minneapolis, MN 55455, United States.

* The research was partially supported by NSF under grant CCR-8722260. The research was also partially supported by the U.S. Department of Energy while the author was at the University of Illinois. The Minnesota Supercomputing Institute provided time on the CRAY-2.
approximated by a linear iterative method. The following are typical steps in an inexact Newton method for solving the nonlinear system \((1.1)\).

1. Choose \(x_0\).

For \(j = 0\) Until Convergence do

2. Solve iteratively: \(F'(x_n) \Delta_n = -F(x_n)\);
3. \(x_{n+1} = x_n + \Delta_n\).

EndFor.

Step (2) often consists of an inner linear solver iteration. If the linear iterative method is a Krylov subspace method (e.g., conjugate gradient, Chebyshev), then the Jacobian is only required for performing Jacobian times vector operations. The explicit computation of the Jacobian requires additional sparse storage and computation time. Efficient methods to compute directly sparse Jacobians have been proposed \([19]\). Alternatively, the Jacobian times vector operation can be approximated \([4,16]\) using the following divided difference:

\[
F'(x_0)v = \frac{F(x_0 + \epsilon v) - F(x_0)}{\epsilon}.
\]  

(1.2)

A very important question is how to terminate the inner and outer iterations in an inexact Newton algorithm. Let the outer iteration terminate if

\[
\|F(x_n)\| \leq \epsilon,
\]  

(1.3)

where \(\epsilon\) is a given error tolerance. The following two possibilities for terminating the inner iteration and retaining convergence for an inexact Newton algorithm have been studied. Chan and Jackson \([5]\) proved that the inexact Newton method converges if the Jacobian is a uniformly definite operator in the neighborhood of the solution and the inner iteration stopping criterion is

\[
\|F(x_n) + F'(x_n) \Delta_n\| \leq \epsilon.
\]  

(1.4)

Dembo et al. \([10]\) proved that if we choose \(0 < \eta_n \leq t < 1\), for all \(n\), and if the inner iteration stopping criterion is

\[
\|F(x_n) + F'(x_n) \Delta_n\| \leq \eta_n \|F(x_n)\|,
\]  

(1.5)

then the convergence of the inexact Newton algorithm is locally at least linear. Thus, we can choose \(\eta_n = \eta < 1\) and obtain linear convergence rate in an inexact Newton algorithm. Superlinear or quadratic convergence for the outer Newton iteration can be obtained if the linear residual norm is \(O(\|F(x_n)\|)\) or \(O(\|F(x_n)\|^2)\), respectively. For quadratic convergence the Jacobian \(F'(x)\) needs to be locally Lipschitz continuous. We note that several inner iterations may still be required to obtain convergence (linear or higher rate) with this stopping criterion.

Nonlinear steepest descent methods for the minimal residual and normal equations have been studied by many authors (cf. \([22,23]\)). Fletcher and Reeves \([15]\), and Daniel \([7]\) have obtained a nonlinear conjugate gradient method which converges if the Jacobian is symmetric and uniformly positive definite. These nonlinear methods reduce to the standard conjugate gradient methods for linear systems. These methods are based on exact line search at each iteration and thus must solve a scalar nonlinear minimization problem in order to determine the steplengths. Several authors have suggested inexact line search and have given conditions.
under which these methods would still converge [1,12,14]. This is done to avoid solving exactly
the scalar minimization problem whose derivative evaluation involves evaluation of the nonlinear
operator.

In the last two decades many Krylov subspace iterative methods have been derived for
solving nonsymmetric linear systems of equations. These methods are generalizations of the
conjugate gradients methods for symmetric and positive definite linear systems. Some outstanding
examples are the generalized conjugate residual method (GCR), Orthomin(k) [13,25] and
the generalized conjugate gradient method (GCG) [2].

In this article we undertake the task of deriving a nonlinear generalization of Orthomin(1).
This new method is called Nonlinear Orthomin(1). This method consists of an iteration which
requires computation of a nonlinear steplength. It coincides with the linear Orthomin(1) if the
operator equation is linear. We prove global and local convergence results for this new method.
We also provide asymptotic residual error bounds. We compare the Nonlinear Orthomin(1) in
terms of performance to the inexact Newton-Orthomin(1) algorithms with stopping criteria
(1.3), (1.4) and (1.3)–(1.5). We present two test problems. These are operator equations arising
in the discretization of nonlinear elliptic partial differential equations. The Nonlinear Or-
thomin(1) demonstrated superior performance to the inexact Newton-Orthomin(1).

In Section 2, we review the Orthomin method. In Section 3, we derive a nonlinear extension
to Orthomin(1). Under assumptions on the Jacobian and Hessian of the nonlinear systems we
show that this method converges to a globally unique solution. In Section 4, we prove local
convergence results and give asymptotic residual error bounds. In Section 5, we describe
practical details in implementing the Newton-Orthomin(1) and Nonlinear Orthomin(1) meth-
ods. We also describe the preconditioned Nonlinear Orthomin(1) method with right constant
operator preconditioning. In Section 6, we show some numerical experiments for nonlinear
systems arising from the discretization of linear boundary value problems in PDEs. In Section
7, we draw conclusions and describe future work on this subject.

2. The Orthomin method

In this section, we review the Orthomin(k) method [13,25]. Let us consider the system of
linear equations

$$Ax = f,$$

(2.1)

where $A$ is a large and sparse matrix of order $N$. Direct methods may be inefficient for solving
this problem because of the large amount of work and storage involved. Iterative methods can
be used to obtain an approximate solution.

Assume that $A$ is Symmetric Positive Definite (SPD). Then the Conjugate Gradient method
(CG) applies. Solving a SPD linear system by use of the CG method is equivalent to minimizing
a quadratic error functional

$$E(x) = (x - h)^T A(x - h),$$

where $h = A^{-1}f$ is the solution of the system. In infinite precision arithmetic the exact solution
is reached in at most $N$ iterations. The conjugate residual (CR) method is a variant of the
conjugate gradient method in which the residual norm $E(x) = \| Ax - f \|^2$ is minimized at every iteration.

The generalized conjugate residual (GCR) [13] is an extension of CR which applies to nonsymmetric systems provided that the symmetric part of the matrix $\frac{1}{2}(A^T + A)$ is positive definite. This method also terminates (in infinite arithmetic) in at most $N$ steps. However, the storage requirements increase at every step. Vinsome [25] proposed the Orthomin($k$), as a practical version of GCR, where the latter has the drawback of keeping all the previous direction vectors. Let $Pr$ denote the approximate inverse operator of $A$ in a right preconditioning for the system (2.1).

**Algorithm 2.1. Orthomin($k$).**

(0) Choose $x_0$.

(1) Compute $r_0 = f - Ax_0$.

(2) $p_0 = Pr r_0$.

For $n = 0$ Step 1 Until Convergence Do

(3) $c_n = \frac{(r_n, Ap_n)}{(Ap_n, Ap_n)}$;

(4) $x_{n+1} = x_n + c_n p_n$;

(5) $r_{n+1} = r_n - c_n Ap_n$;

(6) $p_{n+1} = Pr r_{n+1} + \sum_{j=n}^{n} b^n_j p_j$, where $b^n_j = -\frac{(A Pr r_{n+1}, Ap_j)}{(Ap_j, Ap_j)}$, $j \leq n$;

(7) $Ap_{n+1} = A Pr r_{n+1} + \sum_{j=n}^{n} b^n_j Ap_j$, where $j = \min(0, n - k + 1)$;

Endfor.

Eisenstat et al. [13] proved that Orthomin($k$), $k > 0$, converges. Note that if the matrix is symmetric, Orthomin(1) is the CR method. Orthomin applied with right preconditioning minimizes the residual norm of the unpreconditioned system, while left preconditioning minimizes a preconditioned residual norm.

### 3. The Nonlinear Orthomin(1) method

In this section, we generalize the Orthomin(1) iteration to a nonlinear iteration which requires the solution of a scalar equation to determine the steplength. We then prove a global convergence result under assumptions that the Hessian and the Jacobian are uniformly bounded and the Jacobian is uniformly definite.

Let $F(x)$ be an operator mapping of the Euclidean space $\mathbb{R}^n$ (or, even more generally, a real Hilbert space) into itself. The notation $F'(x)$ and $F''(x)$ will be used to denote the Fréchet and Gâteaux derivatives, respectively. Also, for simplicity $F'_n$ and $F''_n$ will denote $F'(x_n)$ and
\[ F'(x_n), \text{ respectively. We seek to solve iteratively the nonlinear system of equations } F(x) = 0. \text{ In the linear case, } F(x) = Ax - b \text{ and } F'(x) = A. \]

Assume that \( F'(x) \) and \( F''(x) \) exist at all \( x \) and that there exist scalars \( 0 < m \leq M, 0 < B, \) independent of \( x, \) so that the following conditions are satisfied for any vectors \( x \) and \( v: \)

\[
m \| v \|^2 \leq (F'(x)v, v) \leq M \| v \|^2, \tag{3.1a}
\]

\[
\| F''(x) \| \leq B, \tag{3.1b}
\]

\[
m^2 \| v \|^2 \leq \left( (F'(x)^TF'(x))v, v \right) \leq M^2 \| v \|^2. \tag{3.1c}
\]

**Remark 3.0.** The rightmost inequality in (3.1a) and the leftmost inequality in (3.1c) are can be derived from the remaining inequalities. To see this we use the Schwarz inequality and the rightmost inequality in (3.1c) to obtain the rightmost inequality in (3.1a). We also use the Schwarz inequality and the leftmost inequality in (3.1a) to obtain the leftmost inequality in (3.1c).

Condition (3.1a) states that the symmetric part of the Jacobian is uniformly positive definite. This stems from the identity \((F'(x)v, v) = \frac{1}{2} [F'(x) + F'(x)^T]v, v).\) Note that (3.1c) is satisfied if for example the Jacobian and its inverse are bounded: \( \| F'(x) \| < M \) and \( \| F'(x)^{-1} \| < 1/m \) imply \( m^2 \| v \|^2 \leq (F'(x)^TF'(x)v, v) \leq M^2 \| v \|^2. \)

Under assumptions (3.1) we consider the following nonlinear generalization of Orthomin(1).

**Algorithm 3.1.** Nonlinear Orthomin(1).

0. Choose \( x_0. \)

1. \( p_0 = r_0 = -F(x_0). \)

For \( n = 0 \) Until Convergence Do

2. Select the smallest \( c_n > 0 \) to solve \( \min_{c > 0} \| F(x_n + cp_n) \| \);

3. \( x_{n+1} = x_n + c_n p_n; \)

4. \( r_{n+1} = -F(x_{n+1}); \)

5. \( p_{n+1} = r_{n+1} + b_n p_n, \) where \( b_n = -\frac{(F'_{n+1}r_{n+1}, F'_{n+1}p_n)}{\| F'_{n+1}p_n \|^2}; \)

EndFor.

The selection of scalars \( c_n \) and \( b_n \) guarantees that the two orthogonality conditions

\[
( r_n, F'_{n}p_{n-1} ) = 0 \tag{3.2}
\]

and

\[
( F'_{n}p_n, F'_{n}p_{n-1} ) = 0 \tag{3.3}
\]

hold. Under the assumptions (3.1) the following lemma holds true for Algorithm 3.1.

**Lemma 3.2.** Let \( \{ r_n \} \) be the nonlinear residuals and \( \{ p_n \} \) be the direction vectors in Algorithm 3.1; then the following identities hold true:

(i) \( (r_n, F'_{n}p_n) = (r_n, F'_{n}r_n); \)
(ii) \((F'_n p_n, F'_n r_n) = \|F'_n p_n\|^2;\)

(iii) \((F'_n p_n, p_n) = (F'_n r_n, r_n) + b_{n-1}(F'_n p_{n-1}, p_{n-1}) ;\)

(iv) \(\|F'_n r_n\|^2 = \|F'_n p_n\|^2 + b_{n-1}^2 \|F'_n p_{n-1}\|^2 ;\)

(v) \(\|r_n\| \leq \left(\frac{M}{m}\right)^{1/2} \|p_n\|;\)

(vi) \(\|p_n\| \leq \frac{M}{m} \|r_n\|;\)

(vii) \(\|r_{n+1}\| \leq \|r_n\| .\)

**Proof.** The orthogonality relations (3.2) and (3.3) combined with step (5) of Algorithm 3.1 imply (i)–(iv). Equality (iii) is used in proving inequality (v) as follows:

\[ m\|r_n\|^2 \leq |(r_n, F'_n r_n)| \leq |(p_n, F'_n p_n)| \leq \|p_n\| \|F'_n p_n\| \leq M\|p_n\|^2. \]

Equality (iv) is used in proving inequality (vi) as follows:

\[ m^2\|p_n\|^2 \leq \|F'_n p_n\|^2 \leq \|F'_n r_n\|^2 \leq M^2\|r_n\|^2. \]

Inequality (vii) follows from the definition of \(c_n\). \(\square\)

**Remark 3.3.** Let \(f_n(c)\) denote the scalar function \(\frac{1}{2}\|F(x_n + cp_n)\|^2\). Its first and second derivatives are given by

\[
f_n'(c) = (F(x_n + cp_n), F'(x_n + cp_n)p_n),
\]

\[
f_n''(c) = \left((F''(x_n + cp_n)p_n, p_n), F(x_n + cp_n)\right) + \|F'(x + cp_n)p_n\|^2.
\]

The following upper and lower bounds on \(f_n''(c)\) can be computed from (3.5), the assumptions (3.1) and Lemma 3.2:

\[
\|p_n\|^2 (m^2 - B \|r_0\|) \leq m^2\|p_n\|^2 - B\|p_n\|^2 \|r_n\| \leq f_n''(c),
\]

\[
f_n''(c) \leq \|p_n\|^2 (M^2 + B \|r_0\|) \leq \|p_n\|^2 (M^2 + B \|r_0\|).
\]

We next prove that if assumptions (3.1) hold, then the nonlinear Orthomin(1) iteration converges to a globally unique solution.

**Theorem 3.4.** Under the assumptions (3.1) on the nonlinear operator \(F(x)\) the sequence \(x_n\) generated by Algorithm 3.1 is well-defined for any \(x_0\), it converges to a unique solution \(x^*\) of the nonlinear system \(F(x) = 0\) and

\[
\|x_n - x^*\| < \frac{1}{m}\|F(x_n)\|.
\]

**Proof.** The proof is divided in four parts.

Firstly, we prove the existence of \(c_n\) in step (2) of Algorithm 3.1. The derivative of the real function \(f_n\) at zero

\[f_n'(0) = -(r_n, F'_n r_n)\]
is negative because of assumptions (3.1). So there exists \( c > 0 \) such that \( \| F(x_n + cp_n) \| < \| r_n \| \). We must prove that there is a \( c > 0 \) such that \( f_n(0) \leq f_n(c) \). This would imply that there exists a \( c_n > 0 \) where \( f_n(c) \) assumes a local minimum. We use the value theorem for the operator \( F(x) \) to obtain the following equation:

\[
(F(y) - F(x), (y - x)) = (F'(z)(y - x), (y - x)).
\]

Combining this with condition (3.1a) we obtain the inequality

\[
m \| y - x \|^2 \leq \| F(y) - F(x) \| \| y - x \|.
\]

This inequality implies that

\[
m \| y - x \| \leq \| F(y) - F(x) \|. \tag{3.8}
\]

For \( x = x_n \) and \( y = x_n + cp_n \) we conclude that \( F(y) \) grows unbounded for \( c \to \infty \). This proves that there is a \( c > 0 \) such that \( f_n(0) \leq f_n(c) \).

Secondly, we obtain a lower bound on the steplength \( c_n \). Taylor’s expansion gives \( f_n'(c_n) = 0 = f_n'(0) + c_n f_n''(\tilde{c}_n) \), where \( \tilde{c}_n = t_n c_n \) for some \( t \) in \((0, 1)\). We solve for \( c_n \). We then use the upper bound in equality (3.7) and Lemma 3.2(vi) to obtain

\[
\frac{m^3}{M^2(M^2 + B \| r_0 \|)} \leq \frac{m \| r_n \|^2}{\| p_n \|^2(M^2 + B \| r_0 \|)} \leq \frac{(r_n, F' r_n)}{\| p_n \|^2(M^2 + B \| r_n \|)} \leq c_n. \tag{3.9}
\]

Thirdly, we prove that the sequence of residual norms decreases to zero. For \( \tilde{c} = t c \) for some \( t \) in \([0, 1]\) we have

\[
f_n(c) = \frac{1}{2} \| r_n \|^2 - c(r_n, F' r_n) + \frac{1}{2} c^2 f_n''(\tilde{c}).
\]

To obtain the upper bound on \( f_n(c) \) we use (3.1a), (3.7) and Lemma 3.2(vi):

\[
f_n(c) \leq \left[ \frac{1}{2} - m c + \frac{1}{2} c^2 (B \| r_n \| + M^2 \| r_n \|^2) \right] \| r_n \|^2.
\]

Now by inserting

\[
c = \frac{m^3}{M^2(M^2 + B \| r_0 \|)}
\]

we obtain the following bound on the residual error:

\[
\frac{1}{2} \| r_{n+1} \|^2 = f_n(c_n) \leq f_n(c) \leq \frac{1}{2} \left[ 1 - \frac{m^3}{M^2(M^2 + B \| r_0 \|)} \right] \| r_n \|^2.
\]

Since \( M \) is the dominant term in the bound expression it follows that \( \| r_n \| \to 0 \).

Finally, we prove that the sequence of iterates converges to a unique solution of the nonlinear operator equation. By use of (3.8) with \( x = x_n \) and \( y = x_{n+k} \) we obtain that the sequence \( x_n \) is a Cauchy sequence. Thus it converges to \( x^* \) and \( F(x^*) = 0 \). The uniqueness and the error bound inequality in the theorem statement follow from (3.8) with \( x = x_n \) and \( y = x^* \). □
4. Asymptotic steplength estimates and error bounds

In this section, we obtain asymptotic estimates of the steplengths $c_n$ near the solution. We then obtain residual error bounds and prove a local convergence result.

Firstly, we prove a lemma giving bounds on the nonlinear steplengths.

**Lemma 4.1.** We consider assumptions (3.1) and the additional condition on the initial residual $\| r_0 \| < m^2/(2B)$. Let $c_n$ and $p_n$ be as in Theorem 3.4; then the inequalities

$$\frac{m^3}{M^2 + m^2} \leq c_n \leq \frac{2M^{3/2}}{m}$$  \hspace{1cm} (4.1)

and

$$\| c_n p_n \| \leq 2M \| r_n \|$$  \hspace{1cm} (4.2)

hold.

**Proof.** To prove the rightmost inequality in (4.1) we follow the proof of (3.9). We solve $f_n'(c_n) = 0 = f_n'(0) + c_n f_n''(\bar{c}_n)$ for $c_n$. Then we use the lower bound in (3.6) and Lemma 3.2(i) to obtain

$$c_n \leq \frac{(r_n, F_n'r_n)}{\| p_n \|^2 (m^2 - B \| r_0 \|)} \leq \frac{\| r_n \| \| F_n' p_n \|}{\| p_n \|^2 (m^2 - B \| r_0 \|)}. \hspace{1cm} (4.3)$$

Now, using (3.1c), Lemma 3.2(v) and the condition on the initial residual $\| r_0 \| < m^2/(2B)$, we obtain

$$c_n \leq \frac{M \| r_n \|}{\| p_n \|(m^2 - B \| r_0 \|)} \leq \frac{2M^{3/2}}{m}. \hspace{1cm}$$

The leftmost inequality in (4.1) is proved from (3.9) and the condition on the initial residual. The inequality (4.2) follows from (4.3) and (3.1c). \qed

We next obtain asymptotic estimates of the steplengths $c_n$.

**Proposition 4.2.**

$$\frac{(F_n'r_n, r_n)}{\| F_n p_n \|^2} \frac{1}{1 + \epsilon_n} \leq c_n \leq \frac{(F_n'r_n, r_n)}{\| F_n' p_n \|^2} \frac{1}{1 - \epsilon_n}, \text{ where } \epsilon_n = O(\| r_n \|).$$

**Proof.** We will prove only the rightmost inequality. The leftmost inequality is proved similarly. The derivative of $f_n$ by use of (3.4) and expansion into second-order derivative terms becomes

$$f_n'(c) = \left( [F_n p_n + c(F_n, p_n, p_n)] , [F(x_n) + cF_n' p_n + \frac{1}{2} c^2 (F_n'' p_n, p_n)] \right),$$

where $(F_n', p_n, p_n) = (F_n'(x_i + c t_i p_n) p_n, p_n)$ for some $t_i$ with $0 < t_i < 1$ and $i = 1, 2$. Now developing the inner product and using our assumptions (3.1) and Lemma 3.2 we obtain

$$f_n'(c) \leq -(F_n'r_n, r_n) + c \| F_n' p_n \|^2 + cB \| p_n \|^2 \left( \frac{3}{2} c \| F_n' p_n \| + \| r_n \| + \frac{1}{2} c^2 B \| p_n \|^2 \right).$$
Thus, for \( c = c_n \) we obtain

\[
(F_n' r_n, r_n) \leq c_n \| F_n' p_n \|^2 + c_n \| F_n' p_n \|^2 \varepsilon_n,
\]

where

\[
\varepsilon_n = \frac{\| p_n \|^2 B}{\| F_n' p_n \|^2} \left[ \frac{3}{2} c_n \| F_n' p_n \| + \| r_n \| + \frac{1}{2} c_n^2 B \| p_n \|^2 \right].
\]

Using condition (3.1c) and inequality (4.2) in Lemma 4.1 we obtain

\[
\varepsilon_n \leq \varepsilon \| r_n \|,
\]

where

\[
\varepsilon = \frac{B}{m^2} \left[ 3M^2 + 1 + 2M^2 B \| r_0 \| \right].
\]

We next prove a lemma relating the norms of two successive residuals in Algorithm 3.1.

**Lemma 4.3.** Under the assumptions of Theorem 3.4 we obtain the following asymptotic difference of the residual error \( E(x_n) = \| r_n \|^2 \) of two successive iterates:

\[
E(x_{n+1}) - E(x_n) = -c_n (r_n, F_n' r_n) + O(R_n).\]

**Proof.** From the identity \( E(x_{n+1}) - E(x_n) = (r_{n+1} - r_n, r_{n+1}) - (r_n, r_n - r_{n+1}) \) by expanding in Taylor's series around \( x_{n+1} \) and \( x_n \), respectively the first and second term we obtain

\[
E(x_{n+1}) - E(x_n) = \left[ (F_{n+1}'(-c_n p_n) + \frac{1}{2} c_n^2 (F_{n+1}'' p_n, p_n)), r_{n+1} \right]

- \left[ (r_n, c_n F_n' p_n + \frac{1}{2} c_n^2 (F_n'' p_n, p_n)) \right],
\]

where \( n+1 = x_{n+1} + t_c p_n \) and \( \bar{n} = x_n + s_c p_n \) for some \( t, s \) in \((0, 1)\). Since \( r_{n+1} \) is orthogonal to \( F_{n+1}' p_n \) and by using Lemma 3.2(i) and Lemma 4.1 the above expression gets reduced to

\[
-c_n (r_n, F_n' r_n) + \frac{1}{2} c_n^2 \left[ (F_{n+1}'' p_n, p_n), r_{n+1} \right] - \left[ (r_n, (F_n'' p_n, p_n)) \right].
\]

Now the difference in the square brackets is easily seen to be bounded by \( 2BM^2 \| r_n \|^3 \) using assumptions (3.1b) and Lemma 4.1. \( \square \)

Now, we use the previous lemma to obtain an asymptotic residual error bound for iterates in Algorithm 3.1.

**Proposition 4.4.** Under the assumptions of Theorem 3.4 we obtain the following inequality on the residual errors:

\[
E(x_{n+1}) \leq E(x_n) d_n,
\]

where \( d_n = [1 - m^2/M^2 + \sigma_n] \) and \( \sigma_n = 2M^2 \| r_n \|. \)
Proof. By Lemma 4.3 we need to have an estimate of \( c_n(r_n, F_n' r_n) \). Using conditions (3.1), Lemma 3.2 and Proposition 4.2 we prove the following inequality:

\[
c_n(r_n, F_n' r_n) \geq \frac{(r_n, F_n' r_n)^2}{\| F_n' p_n \|^2} \geq \frac{1}{(1 + \epsilon_n)} \geq \frac{(r_n, F_n' r_n)^2}{\| F_n' r_n \|^2} \geq \frac{1}{(1 + \epsilon_n)} \geq \frac{m^2}{M^2} \frac{1}{(1 + \epsilon_n)} \| r_n \|^2.
\]

Now using Lemma 4.3 we obtain

\[
E(x_{n+1}) \leq \| r_n \|^2 \left[ 1 - \frac{m^2}{M^2} \frac{1}{(1 + \epsilon_n)} \right] + O(\| r_n \|^3) \leq E(x_n) \left[ 1 - \frac{m^2}{M^2} \right] + O(\| r_n \|^3).
\]

The last term in this inequality is

\[
E(x_n) \left[ 1 - \frac{m^2}{M^2} + \sigma_n \right], \quad \text{where } \sigma_n = 2BM^2 \| r_n \|.
\]

In the following theorem we prove local convergence for the Nonlinear Orthomin(1) iteration and give an error bound estimate. Under assumptions (3.1) and an additional assumption on the initial residual norm it is proved that the iteration remains inside a ball centered at \( x_0 \) with the appropriate choice of a radius \( \delta_0 \).

**Theorem 4.5.** Let \( x_0 \) be selected such that

\[
\| r_0 \| < \frac{1}{2BM^2} \quad \text{and} \quad d_0^2 = \left( 1 - \frac{m^2}{M^2} \right) + \sigma_0 < 1,
\]

where \( \sigma_2 = 2BM^2 \| r_n \| \). Let \( \delta_0 \) denote \( 2M^2/(1 - d_0) \| r_0 \| \); the sequence \( x_n \) generated by Algorithm 3.1 remains in the ball \( B(x_0, \delta_0) \) and it converges to \( x^* \), which is the unique solution of \( F(x) = 0 \). Furthermore, \( d_n^2 = (1 - m^2/M^2) + \sigma_n \) decreases to \( 1 - m^2/M^2 \) and \( \| x_n - x^* \| \leq \delta_0 d_0 d_1 \cdots d_{n-1} \).

Proof. From Lemma 4.1 we obtain \( \| c_0 p_0 \| \leq 2M^2 \| r_0 \| \leq \delta_0 \). So, \( x_1 = x_0 + c_0 p_0 \) is in \( B(x_0, \delta_0) \).

Let \( \delta_1 \) denote \( 2M^2 \| r_1 \|/(1 - d_1) \). We will prove that \( B(x_1, \delta_1) \subset B(x_0, \delta_0) \). This follows from the inequality of Proposition 4.4 if we prove that the rightmost term is bounded by \( \delta_0 \):

\[
\| x - x_0 \| \leq \| x - x_1 \| + \| x_1 - x_0 \| \leq \delta_0 + 2M^2 \| r_0 \|. \quad (4.4)
\]

Proposition 4.4 implies that \( \| r_1 \| \leq d_0 \| r_0 \| \). This inequality and \( d_1 < d_0 \) imply that

\[
\| r_1 \| \leq \frac{d_0}{1 - d_1} \| r_0 \| \leq \frac{d_0}{1 - d_0} \| r_0 \| = \frac{1}{1 - d_0 \| r_0 \|}.
\]

This proves that the last term in (4.4) is less than \( \delta_0 \).

Since \( \| r_{n-1} \| \leq \| r_n \| \) and \( d_{n+1} \leq d_n \), we can prove by induction that the iterate \( x_n \) generated in Algorithm 3.1 satisfies the hypothesis of this theorem.

Now \( \| r_n \|^2 \leq d_{n-1}^2 \cdots d_0^2 \| r_0 \|^2 \leq d_0^{2n} \| r_0 \|^2 \) implies that the sequence of residuals \( r_n \) converges to 0. Also,

\[
\| x_{n+k} - x_n \| \leq 2M^2 \| r_{n+k-1} \| \leq \frac{2M^2}{m^2} \| r_0 \| \sum_{j=n}^{n+k-1} d_j d_{j+1} \cdots d_{n-1}.
\]
proves the sequence of iterates \( x_n \) converges to \( x^* \in B(x_0, \delta_0) \) and \( F(x^*) = 0 \). When \( k \) approaches infinity, the inequality becomes
\[
\| x^* - x_n \| \leq C_0 d_0 \cdots d_{n-1},
\]
for some constant \( C_0 \). \( \square \)

5. Implementation details

We next describe four algorithms based on Newton-Orthomin(1) and Nonlinear Orthomin(1) with different choices of stopping criteria. We then present the right preconditioning of the nonlinear system (1.1) with constant preconditioning matrix.

1. **Newton-Orthomin(1)**. This is the inexact Newton method with stopping criteria (1.3), (1.4) (described in the Introduction) for the outer and inner iterations, respectively.

2. **Nonlinear Orthomin(1)**. This is Algorithm 3.1. The algorithm halts when the norm of the nonlinear residual \( F(x_n) \) is less than a tolerance \( \epsilon \).

3. **Restarted Newton-Orthomin(1)**. This is the inexact Newton method with stopping criteria (1.3)–(1.5) (described in the Introduction) for the outer and inner iterations, respectively.

4. **Restarted Nonlinear Orthomin(1)**. The algorithm halts when the norm of the nonlinear residual \( F(x_n) \) is less than a tolerance \( \epsilon \). However, it restarts setting \( x_0 = x_n \) when
\[
\| F(x_n) \| \leq \eta_n \| F(x_0) \|.
\]

Algorithms (1) and (2) are nonlinear extensions of the linear Orthomin(1) Algorithm 2.1. This means that if the problem is linear, algorithm (1) will perform only one outer iteration. It also will perform the same number of inner iterations as algorithm (2). Algorithms (3) and (4) are restarted algorithms. If the problem is linear, then the restarted algorithms (3) and (4) generate the same iterates.

In the implementation, algorithms (2) and (4) are based on modification of Algorithm 3.1 in order to avoid exact line searches. The steplength to Proposition 4.2 can be approximated by the formula
\[
c_n \approx \frac{(r_n, F_n' r_n)}{\| F_n' p_n \|^2}.
\]

Also, we approximate
\[
b_n \approx -\frac{(F_n' r_n, F_n' p_n)}{\| F_n' p_n \|^2}
\]
in order to save computational work. The only vector product computed is \( F_n' r_n \) and this is used to approximate \( F_n' p_n = F_n' r_n + b_{n-1} F_{n-1}' p_{n-1} \). This is generally expected to lead to a stable method because the Jacobian does not vary much in a single iteration.

We briefly discuss here the right preconditioning of Algorithm 3.1 with a constant preconditioner \( Pr \). The matrix \( Pr \) is assumed to be an approximation to the inverse of the Jacobian \( F'(x) \) (i.e., \( F'(x) Pr \approx I \)). The problem \( F(x) = 0 \) is transformed to \( G(y) = F(Pr y) = 0 \), where \( y = Pr^{-1} x \). Note that the Jacobian of the transformed problem is \( G'(y) = F'(x) Pr \). Now, it is
easy to check that Algorithm 3.1 applied to \( G(y) = 0 \) yielding approximants \( y \) can be transformed to yield approximants \( x_n \). The only changes required are in steps (1) and (5). These steps become (1') and (5'), respectively:

\[
(1') \quad p_0 = \Pr r_0 = -F(x_0);
\]

\[
(5') \quad p_{n+1} = \Pr r_{n+1} + b_n p_n, \quad \text{where } b_n = -\frac{(F'_{n+1} \Pr r_{n+1}, F'_{n+1} p_n)}{\|F'_{n+1} p_n\|^2}.
\]

Most of the computational work is attributed to Jacobian \( (F'(x)) \) evaluations, Jacobian times vectors and functions \( (F(x)) \) evaluations. Algorithms (1) and (3) require one Jacobian, one function evaluation per outer iteration and one Jacobian times vector operation per inner iteration. Algorithms (2) and (4) require one Jacobian, one function evaluation and one matrix-vector product per iteration. If the Jacobian times vector operation is approximated as in (1,2), then no Jacobian evaluation is required. By using (1,2) algorithms (1) and (3) require one function evaluation per outer and one function evaluation per inner iteration. However, algorithms (2) and (4) require two function evaluations per iteration. For the preconditioned methods one matrix times vector product per iteration (by the matrix \( \Pr \)) must be added to the work for all algorithms.

6. Numerical tests

In this section, we present two nonlinear elliptic partial differential equation problems whose discretization gives rise to nonlinear algebraic systems with nonsymmetric Jacobians. We use the four algorithms described in the previous section. We compare their execution times on a CRAY-2 vector computer.

Problem 6.1. Let us consider the nonlinear differential equation

\[
- \Delta u + \beta u_x + \gamma u^2 = f(x, y),
\]

(6.1)

where \( u \) and \( f \) are defined on the unit square \([0, 1] \times [0, 1]\). We determine \( f(x, y) \) so that \( u(x, y) = e^{x^2+y^2} \) is the solution of (6.1) with inhomogeneous Dirichlet conditions imposed on the boundary. We discretize the problem using a central difference approximation for \( \Delta \) and upwind first-order approximation for \( u_x \). Thus we obtain a system of nonlinear equations \( F(x) = 0 \) of order \( N = n_x^2 \) with \( n_x \) being the number of interior grid points in each direction. The same problem with \( \beta = 0 \) and \( \gamma = 1 \) was used in [18].

Problem 6.2. Let us consider the nonlinear differential equation

\[
- \Delta u + \beta u_x + \gamma e^u = f(x, y),
\]

(6.2)

where \( u \) and \( f \) are defined on the unit square \([0, 1] \times [0, 1]\). Again we determine \( f(x, y) \) so that \( u(x, y) = e^{x^2+y^2} \) is the solution of (6.2) with inhomogeneous Dirichlet conditions imposed on the boundary. This is a simplified form of the Bratu problem [17]. It is known that for \( \gamma \geq 0 \) there exists a unique solution to the problem. We discretize this problem in the same way as Problem 6.1.
Table 1
Problem 6.1. Nonlinear Orthomin(1) and Newton-Orthomin(1) ($\gamma = 1$, $\beta = 10$)

<table>
<thead>
<tr>
<th>$\sqrt{N}$</th>
<th>NIT</th>
<th>Ntimes</th>
<th>OIT</th>
<th>IIT</th>
<th>MAX-IIT</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>27</td>
<td>0.0243</td>
<td>4</td>
<td>58</td>
<td>16</td>
<td>0.0427</td>
</tr>
<tr>
<td>32</td>
<td>44</td>
<td>0.0662</td>
<td>4</td>
<td>109</td>
<td>32</td>
<td>0.1372</td>
</tr>
<tr>
<td>64</td>
<td>77</td>
<td>0.2505</td>
<td>4</td>
<td>215</td>
<td>64</td>
<td>0.6200</td>
</tr>
<tr>
<td>128</td>
<td>151</td>
<td>1.6035</td>
<td>5</td>
<td>525</td>
<td>128</td>
<td>4.8985</td>
</tr>
<tr>
<td>160</td>
<td>183</td>
<td>2.9454</td>
<td>5</td>
<td>661</td>
<td>160</td>
<td>9.1898</td>
</tr>
<tr>
<td>200</td>
<td>220</td>
<td>5.4893</td>
<td>5</td>
<td>852</td>
<td>200</td>
<td>22.4173</td>
</tr>
</tbody>
</table>

Table 2
Problem 6.1. Restarted Nonlinear Orthomin(1) and Newton-Orthomin(1) ($\gamma = 1$, $\beta = 10$)

<table>
<thead>
<tr>
<th>$\sqrt{N}$</th>
<th>NIT</th>
<th>Ntimes</th>
<th>OIT</th>
<th>IIT</th>
<th>MAX-IIT</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>25</td>
<td>0.0222</td>
<td>15</td>
<td>28</td>
<td>2</td>
<td>0.0259</td>
</tr>
<tr>
<td>32</td>
<td>43</td>
<td>0.0622</td>
<td>16</td>
<td>40</td>
<td>4</td>
<td>0.0595</td>
</tr>
<tr>
<td>64</td>
<td>84</td>
<td>0.2804</td>
<td>19</td>
<td>86</td>
<td>8</td>
<td>0.2776</td>
</tr>
<tr>
<td>128</td>
<td>171</td>
<td>1.9352</td>
<td>21</td>
<td>172</td>
<td>16</td>
<td>1.7147</td>
</tr>
<tr>
<td>160</td>
<td>202</td>
<td>3.3169</td>
<td>21</td>
<td>205</td>
<td>20</td>
<td>3.1165</td>
</tr>
<tr>
<td>200</td>
<td>235</td>
<td>6.0235</td>
<td>22</td>
<td>263</td>
<td>24</td>
<td>5.8935</td>
</tr>
</tbody>
</table>

Table 3
Problem 6.1. Nonlinear Orthomin(1) and Newton-Orthomin(1) ($\gamma = 1$, $\beta = 30$)

<table>
<thead>
<tr>
<th>$\sqrt{N}$</th>
<th>NIT</th>
<th>Ntimes</th>
<th>OIT</th>
<th>IIT</th>
<th>MAX-IIT</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>26</td>
<td>0.0251</td>
<td>4</td>
<td>50</td>
<td>16</td>
<td>0.0461</td>
</tr>
<tr>
<td>32</td>
<td>52</td>
<td>0.0786</td>
<td>4</td>
<td>98</td>
<td>32</td>
<td>0.1323</td>
</tr>
<tr>
<td>64</td>
<td>113</td>
<td>0.3911</td>
<td>4</td>
<td>197</td>
<td>64</td>
<td>0.5515</td>
</tr>
<tr>
<td>128</td>
<td>280</td>
<td>3.0074</td>
<td>4</td>
<td>392</td>
<td>128</td>
<td>3.6663</td>
</tr>
<tr>
<td>160</td>
<td>379</td>
<td>6.1082</td>
<td>4</td>
<td>500</td>
<td>160</td>
<td>6.9018</td>
</tr>
<tr>
<td>200</td>
<td>535</td>
<td>16.3794</td>
<td>4</td>
<td>644</td>
<td>200</td>
<td>16.7861</td>
</tr>
</tbody>
</table>

The constants $\gamma$ and $\beta$ are used to control the nonlinearity of the problems and the nonsymmetry of the Jacobians. We considered $\gamma = 1$, $\beta = 10$ or 30. The Jacobian of this problem for small enough mesh size can be proven to satisfy conditions (3.1) locally in $\mathbb{R}^N$. Thus nonlinear Orthomin(1) can be used to solve the nonlinear discretized problems. The Jacobian is closer to symmetry in the case of $\beta = 10$. In the case of $\beta = 30$, although the Jacobian has a more significant skew symmetric part, it is also closer to linearity.

For the test problems described here the linear part of the Jacobian was computed once initially and it was stored in five diagonals. The nonlinear part consisting of a single diagonal was computed explicitly whenever needed. It was computed once in each outer iteration of algorithms (1) and (3). However it was computed once in every iteration of algorithms (2) and (4).

In Tables 1–8, we show numerical results on solving Problems 6.1 and 6.2 using the algorithms (1)–(4) with ILU(0) vectorizable preconditioning [6,21,24]. Right preconditioning
Table 4
Problem 6.1. Restarted Nonlinear Orthomin(1) and Newton-Orthomin(1) ($\gamma = 1$, $\beta = 30$)

<table>
<thead>
<tr>
<th>$\sqrt{N}$</th>
<th>NIT</th>
<th>Ntimes</th>
<th>OIT</th>
<th>IIT</th>
<th>MAX-IIT</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>23</td>
<td>0.0215</td>
<td>15</td>
<td>20</td>
<td>3</td>
<td>0.01809</td>
</tr>
<tr>
<td>32</td>
<td>41</td>
<td>0.0626</td>
<td>15</td>
<td>36</td>
<td>5</td>
<td>0.0511</td>
</tr>
<tr>
<td>64</td>
<td>78</td>
<td>0.2827</td>
<td>17</td>
<td>68</td>
<td>11</td>
<td>0.2175</td>
</tr>
<tr>
<td>128</td>
<td>109</td>
<td>1.2906</td>
<td>21</td>
<td>173</td>
<td>25</td>
<td>1.7402</td>
</tr>
<tr>
<td>160</td>
<td>141</td>
<td>2.3649</td>
<td>21</td>
<td>168</td>
<td>27</td>
<td>2.5152</td>
</tr>
<tr>
<td>200</td>
<td>197</td>
<td>5.066</td>
<td>21</td>
<td>217</td>
<td>42</td>
<td>5.2995</td>
</tr>
</tbody>
</table>

Table 5
Problem 6.2. Nonlinear Orthomin(1) and Newton-Orthomin(1) ($\gamma = 1$, $\beta = 10$)

<table>
<thead>
<tr>
<th>$\sqrt{N}$</th>
<th>NIT</th>
<th>Ntimes</th>
<th>OIT</th>
<th>IIT</th>
<th>MAX-IIT</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>23</td>
<td>0.0194</td>
<td>4</td>
<td>54</td>
<td>16</td>
<td>0.0443</td>
</tr>
<tr>
<td>32</td>
<td>38</td>
<td>0.0494</td>
<td>4</td>
<td>108</td>
<td>32</td>
<td>0.1299</td>
</tr>
<tr>
<td>64</td>
<td>73</td>
<td>0.2072</td>
<td>4</td>
<td>209</td>
<td>64</td>
<td>0.5091</td>
</tr>
<tr>
<td>128</td>
<td>135</td>
<td>1.2548</td>
<td>4</td>
<td>400</td>
<td>128</td>
<td>3.0540</td>
</tr>
<tr>
<td>160</td>
<td>158</td>
<td>2.1547</td>
<td>4</td>
<td>418</td>
<td>160</td>
<td>5.8677</td>
</tr>
<tr>
<td>200</td>
<td>233</td>
<td>5.2345</td>
<td>4</td>
<td>682</td>
<td>200</td>
<td>15.9824</td>
</tr>
</tbody>
</table>

Table 6
Problem 6.2. Restarted Nonlinear Orthomin(1) and Newton-Orthomin(1) ($\gamma = 1$, $\beta = 10$)

<table>
<thead>
<tr>
<th>$\sqrt{N}$</th>
<th>NIT</th>
<th>Ntimes</th>
<th>OIT</th>
<th>IIT</th>
<th>MAX-IIT</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>24</td>
<td>0.0210</td>
<td>14</td>
<td>26</td>
<td>2</td>
<td>0.0225</td>
</tr>
<tr>
<td>32</td>
<td>41</td>
<td>0.0555</td>
<td>17</td>
<td>42</td>
<td>4</td>
<td>0.0543</td>
</tr>
<tr>
<td>64</td>
<td>84</td>
<td>0.2535</td>
<td>19</td>
<td>88</td>
<td>7</td>
<td>0.2401</td>
</tr>
<tr>
<td>128</td>
<td>164</td>
<td>1.5348</td>
<td>20</td>
<td>167</td>
<td>14</td>
<td>1.3579</td>
</tr>
<tr>
<td>160</td>
<td>189</td>
<td>2.6093</td>
<td>21</td>
<td>197</td>
<td>18</td>
<td>2.3018</td>
</tr>
<tr>
<td>200</td>
<td>233</td>
<td>6.1126</td>
<td>21</td>
<td>261</td>
<td>22</td>
<td>7.0066</td>
</tr>
</tbody>
</table>

Table 7
Problem 6.2. Nonlinear Orthomin(1) and Newton-Orthomin(1) ($\gamma = 1$, $\beta = 30$)

<table>
<thead>
<tr>
<th>$\sqrt{N}$</th>
<th>NIT</th>
<th>Ntimes</th>
<th>OIT</th>
<th>IIT</th>
<th>MAX-IIT</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>26</td>
<td>0.0242</td>
<td>4</td>
<td>47</td>
<td>16</td>
<td>0.0432</td>
</tr>
<tr>
<td>32</td>
<td>50</td>
<td>0.0705</td>
<td>4</td>
<td>97</td>
<td>32</td>
<td>0.1197</td>
</tr>
<tr>
<td>64</td>
<td>109</td>
<td>0.3297</td>
<td>4</td>
<td>186</td>
<td>64</td>
<td>0.4676</td>
</tr>
<tr>
<td>128</td>
<td>264</td>
<td>2.4841</td>
<td>4</td>
<td>371</td>
<td>128</td>
<td>2.8146</td>
</tr>
<tr>
<td>160</td>
<td>367</td>
<td>5.2559</td>
<td>4</td>
<td>454</td>
<td>160</td>
<td>5.1785</td>
</tr>
<tr>
<td>200</td>
<td>509</td>
<td>15.4821</td>
<td>4</td>
<td>551</td>
<td>21</td>
<td>12.3918</td>
</tr>
</tbody>
</table>
Table 8
Problem 6.2. Restarted Nonlinear Orthomin(1) and Newton-Orthomin(1) ($\gamma = 1, \beta = 30$)

<table>
<thead>
<tr>
<th>$\sqrt{N}$</th>
<th>NIT</th>
<th>Ntimes</th>
<th>OIT</th>
<th>IIT</th>
<th>MAX-IIT</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>20</td>
<td>0.0190</td>
<td>15</td>
<td>21</td>
<td>3</td>
<td>0.0183</td>
</tr>
<tr>
<td>32</td>
<td>35</td>
<td>0.0474</td>
<td>15</td>
<td>35</td>
<td>5</td>
<td>0.0445</td>
</tr>
<tr>
<td>64</td>
<td>66</td>
<td>0.1968</td>
<td>18</td>
<td>82</td>
<td>11</td>
<td>0.2135</td>
</tr>
<tr>
<td>128</td>
<td>134</td>
<td>1.2832</td>
<td>19</td>
<td>135</td>
<td>23</td>
<td>1.0743</td>
</tr>
<tr>
<td>160</td>
<td>161</td>
<td>2.2292</td>
<td>20</td>
<td>137</td>
<td>28</td>
<td>1.5894</td>
</tr>
<tr>
<td>200</td>
<td>157</td>
<td>3.3931</td>
<td>20</td>
<td>200</td>
<td>40</td>
<td>3.5133</td>
</tr>
</tbody>
</table>

was used with the ILU(0) preconditioning obtained from the discrete linear part of the
differential operators. The error tolerance was $\epsilon = 10^{-6}$. The initial vector was chosen to be the
average of the solution in the four vertices of the square domain $x_0 = \frac{1}{4}[1 + 2e^1 + e^2]$. The
stopping criterion parameter $\eta$ was chosen $\eta = \frac{1}{2}$. This choice was observed to be best
compared to $\eta = 0.1, \ldots, 0.9$. Varying $\eta$ between iterations was not considered.

The flags used in the tables are defined as follows. Firstly, for algorithms (1) and (3):
OIT = number of outer (Newton) iterations, IIT = total number of inner iterations, MAX-IIT
= maximum number of inner linear iterations in a single Newton step. The maximum number
of linear iterations allowed was set at $nx$. Secondly, for algorithms (2) and (4): NIT = number of
nonlinear iterations. The execution times (Ntimes for algorithms (2), (4) and Times for
algorithms (1), (3)) are given in seconds on a single vector processor of CRAY-2 at the
University of Minnesota. Although the computations were not performed in a single-user
mode, the load of the machine was low. Several runs were performed and timings with
variations 0.001 seconds were observed.

7. Conclusions and future work

We have presented and analyzed a nonlinear iterative method for solving nonlinear algebraic
systems of equations. This method is a generalization of existing methods for linear systems of
equations. We show that under certain uniform assumptions on the Jacobians and Hessians the
method is guaranteed to converge globally to a unique solution. We also prove local conver-
gence results and give asymptotic error bound estimates. These results extend the work of other
authors [8,15] to deriving nonlinear methods for nonsymmetric Jacobians.

It is well known that Krylov subspace iterative methods for linear systems require the
formation of a small-dimensional subspace on which they project to obtain an approximation to
the solution. What we suggest here is the inversion of the Newton linear iterative method to a
nonlinear iteration Newton method.

The test results show that the Nonlinear Orthomin(1) algorithms are in most cases more
efficient than the Newton-Orthomin(1) algorithms. The number of (inner) iterations can be
used as a rough indicator to explain the difference in efficiency of the methods. However, each
iteration of the Nonlinear Orthomin(1) is more costly than the corresponding inner iterations
of Newton-Orthomin(1). For $\beta = 10$ the Jacobians are close to symmetric. This explains the
superiority of the Nonlinear Orthomin(1) algorithm. For $\beta = 30$ the Jacobians are sufficiently
nonsymmetric but closer to linear than for $\beta = 10$. For $\beta = 10$, Nonlinear Orthomin(1) is slower than the restarted Newton-Orthomin(1) or the restarted Nonlinear Orthomin(1). In this case a method based on GCR [13] or GCG [2] with more vectors in storage must be used for the linear problems. We did not consider nonlinear extensions of these methods here. In general the restarted methods gave unexpected number of iterations as a function of $nx$. The Newton-Orthomin(1) showed the worst performance in all cases.

The method presented here requires the solution of a scalar nonlinear equation. In a future publication we will investigate inexact line search approaches similar to [1,8,12] for determining the parameters in these nonlinear methods.

Acknowledgements

The author thanks the anonymous referees whose comments helped enhance significantly the quality of presentation of this article. He also thanks Dr. C.W. Gear for his help and advice.

References