Electronic Transactions on Numerical Analysis.
Volume 35, pp. 104-117, 2009.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

ACCELERATION OF IMPLICIT SCHEMES FOR LARGE SYSTEMS OF NONLINEAR ODEs*

MOUHAMAD AL SAYED ALI† AND MILOUD SADKANE†

Abstract. Implicit integration schemes for large systems of nonlinear ODEs require, at each integration step, the solution of a large nonlinear system. Typically, the nonlinear systems are solved by an inexact Newton method that leads to a set of linear systems involving the Jacobian matrix of the ODE which are solved by Krylov subspace methods. The convergence of the whole process relies on the quality of initial solutions for both the inexact Newton iteration and the linear systems. To improve global convergence, line search and trust region algorithms are used to find effective initial solutions. The purpose of this paper is to construct subspaces of small dimension where descent directions for line search and trust region algorithms and initial solutions for each linear system are found. Only one subspace is required for each integration step. This approach can be seen as an improved predictor, leading to a significant saving in the total number of integration steps. Estimates are provided that relate the quality of the computed initial solutions to the step size of the discretization, the order of the implicit scheme and the dimension of the constructed subspaces. Numerical results are reported.

Key words. nonlinear equations, nonlinear ODE systems, inexact Newton, GMRES, line search, trust region

AMS subject classifications. 65H10, 65L05

1. Introduction.

Consider the system of ODEs:

$$\dot{y}(t) = f(t, y(t)), \quad t_0 \leq t \leq T, \quad y(t_0) = y^{(0)}, \tag{1.1}$$

where $y(t) \in \mathbb{R}^n$ and n is large.

A class of implicit schemes for solving (1.1) is given by

$$G_i(y_i) = y_i - a_i - \beta h f(t_i, y_i) = 0, \quad i = q+1, \ldots, N, \tag{1.2}$$

where $y_0 = y^{(0)}$ and $y_1, \ldots, y_q$ are assumed to be known, $q \ll N$, y_i is an approximation to $y(t_i)$ with $t_i = t_0 + ih,\ h = \frac{T-t_0}{N}$, β is a scalar and a_i is a vector depending on y_{i-k} and $f(t_{i-k}, y_{i-k})$, $1 \leq k \leq q+1$. Most standard implicit schemes such as implicit Euler, Crank-Nicolson, Adams-Moulton and BDF methods can be written as in (1.2); see, for example, [15, 16].

These schemes have good stability properties, but necessitate the solution of the large system (1.2) at each integration step. The case when f is an affine function of the form $f(t, y(t)) = Ay(t) + b(t)$ leads to linear systems. This case has been considered in [1]. In the present paper we assume that f is a nonlinear function. The Newton method can be used to solve (1.2), but to keep the storage and computational cost low, the inexact Newton method is preferred [9, 11]. It is given by the following process:

$$\begin{aligned}
&\text{Choose } \ y_i^{(0)} \ \text{ as initial guess for } y_i\\
&\text{For } \ k = 0, 1, \ldots, \text{until convergence do :}\\
&\quad \text{Solve inexactly } \ G_i'(y_i^{(k)}) s_i^{(k)} = -G_i(y_i^{(k)}) &(1.3)\\
&\quad \text{Set } \ y_i^{(k+1)} = y_i^{(k)} + \tilde{s}_i^{(k)} &(1.4)
\end{aligned}$$

where $G_i'(y_i^{(k)})$ is the Jacobian matrix of G_i at $y_i^{(k)}$ and $\tilde{s}_i^{(k)}$ is an approximate solution to (1.3). Krylov subspace methods [20], for example GMRES [21], can be used to find $\tilde{s}_i^{(k)}$

*Received May 27, 2008. Accepted for publication January 21, 2009. Published online June 24, 2009. Recommended by R. Varga.

†Université de Bretagne Occidentale, Département de Mathématiques 6, avenue V. Le Gorgeu, CS 93837, 29238 Brest Cedex 3, France ({alsayed, sadkane}@univ-brest.fr).

in the affine subspace $\hat{s}_i^{(k)} + \mathcal{K}_i^{(k)}$, where $\hat{s}_i^{(k)}$ is an initial guess for the exact solution and $\mathcal{K}_i^{(k)}$ is a Krylov subspace constructed with the matrix $G_i'(y_i^{(k)})$ and the initial residual vector $G_i'(y_i^{(k)})\hat{s}_i^{(k)} + G_i(y_i^{(k)})$. One generally asks that $\tilde{s}_i^{(k)}$ satifies

$$\|G_i'(y_i^{(k)})\tilde{s}_i^{(k)} + G_i(y_i^{(k)})\| \leq \epsilon_i^{(k)}\|G_i(y_i^{(k)})\| \tag{1.5}$$

with some tolerance threshold $\epsilon_i^{(k)}$. The results in [9] indicate how $\epsilon_i^{(k)}$ must be chosen to ensure local convergence of the inexact Newton method.

Krylov subspace methods have been used, for example, in [9, 2, 5, 6] in the context of Newton's method and in [13, 3, 4, 12, 7, 17, 19] in the context of ODEs. They have the advantage of requiring only the multiplication of the Jacobian matrix $G_i'(y_i^{(k)})$ by a vector, the Jacobian matrix need not be formed explicitly. Also, good convergence can be expected provided an efficient preconditioner and/or a good initial guess $\hat{s}_i^{(k)}$ is available. In this paper we concentrate our effort on the latter possibility.

Like Newton's method, the inexact Newton method is locally convergent: it converges rapidly provided the initial guess $y_i^{(0)}$ is close enough to y_i. Traditionally, $y_i^{(0)}$ is found with the aid of a predictor, usually an explicit scheme applied to (1.1). The drawback is that often the Newton method does not converge with such a predictor; see Section 4.2. For this reason, the line search backtracking and trust region techniques have been used to improve global convergence [5, 6, 11] from any initial guess $y_i^{(0)}$. These techniques generate a sequence of vectors that converge under some conditions to a local minimum or a saddle point of the function $h_i(y) = \frac{1}{2}\|G_i(y)\|^2$. Here and throughout the paper, the symbol $\|\ \|$ denotes the Euclidean norm. Each vector is obtained from the preceding one by adding a descent direction, on which the effectiveness of these techniques strongly depends. Of particular importance from an algorithmic point of view is the paper [5], where the solution of (1.3) and line search and trust region algorithms all use Krylov subspaces. This results in a method that is effective for nonlinear equations, but rather expensive for ODEs since the nonlinear equation (1.2) must be solved $N - q$ times.

Our aim is to show that the subspace

$$\mathcal{V}_i = \text{span}\{f(t_{i-1}, y_{i-1}), \ldots, f(t_{i-r}, y_{i-r})\}, \quad r \ll n \tag{1.6}$$

and some modifications thereof (namely when $y_{i-1}, \ldots, y_{i-r}$ are replaced by approximations) contain a good approximation to $y_i - a_i$; see Theorems 2.1 and 2.2. As a consequence, the subspace $\mathcal{V}_i$ can be used to find $y_i^{(0)}$ and $\hat{s}_i^{(k)}$, $k \geq 0$.

More precisely, we will see that good descent directions for line search and trust region algorithms and a good initial guess $\hat{s}_i^{(k)}$ of the exact solution $s_i^{(k)}$ can be found from $\mathcal{V}_i$ and the Petrov-Galerkin process. When applied to (1.3), this process finds $\hat{s}_i^{(k)}$ in the subspace $-y_i^{(k)} + a_i + \mathcal{V}_i$, such that

$$G_i'(y_i^{(k)})\hat{s}_i^{(k)} + G_i(y_i^{(k)}) \ \perp\ G_i'(y_i^{(k)})\mathcal{V}_i.$$

This means that $\hat{s}_i^{(k)}$ is the best least squares solution to (1.3) in the subspace $-y_i^{(k)} + a_i + \mathcal{V}_i$. Once $\hat{s}_i^{(k)}$ is obtained as above, very few iterations by a Krylov subspace method are needed to satisfy (1.5). The subspace $\mathcal{V}_i$ does not depend on k and the main advantage here is that only one subspace of small dimension is required for each iteration i of the implicit scheme. The approach thus obtained leads to a significant saving in the total number of integration steps.

The paper is organized as follows. In Section 2 we describe the subspace of approximation $\mathcal{V}_i$ and provide estimates for the error between exact and approximate solutions of (1.2) in terms of the step size h, the order of the implicit scheme and the dimension r of $\mathcal{V}_i$. In Section 3 we briefly review the line search and trust region algorithms and explain how these algorithms can be used, in conjunction with $\mathcal{V}_i$ and the Petrov-Galerkin process, to compute good descent directions and good initial guesses for each linear system of the form (1.3). In Section 4 we discuss the algorithmic aspect of the proposed approach and show numerically its effectiveness on a very stiff problem. The Krylov subspace method used throughout the paper is GMRES.

2. Approximation of the initial guess. In practice we can only hope for an approximation $\tilde{y}_i$ to the exact solution y_i of (1.2). Therefore we replace (1.2) by

$$\|\tilde{G}_i(\tilde{y}_i)\| \leq \varepsilon, \; i = q+1, , \dots, N, \tag{2.1}$$

where $\tilde{G}_i(x) = x - \tilde{a}_i - \beta h f(t_i, x)$, the vector $\tilde{a}_i$ is obtained by replacing the y_j's in the expression of a_i by the $\tilde{y}_j$'s and ε is some tolerance threshold.

Throughout this section we assume that f is Lipschitz with respect to the second variable. We also assume that the scheme (1.2) is stable and of order p and that for $i = 1, \dots, q$, $\tilde{y}_i$ is computed with an i-step scheme such that

$$\max_{0 \leq i \leq q} \|\tilde{y}_i - y_i\| = \mathcal{O}(\varepsilon / h). \tag{2.2}$$

Then, there exists a constant C_0 such that (see, e.g., [18, Chap. 8])

$$\begin{aligned}\max_{q+1 \leq i \leq N} \|\tilde{y}_i - y_i\| &\leq C_0 \left(\max_{0 \leq i \leq q} \|\tilde{y}_i - y_i\| + \sum_{i=q+1}^{N} \|\tilde{G}_i(\tilde{y}_i)\| \right) \\ &\leq C_0 \left(\max_{0 \leq i \leq q} \|\tilde{y}_i - y_i\| + (N-q)\, \varepsilon \right) = \mathcal{O}(\varepsilon / h).\end{aligned}$$

The following theorem shows that the subspace

$$\mathcal{V}_i = \text{span}\{ f(t_{i-k}, \tilde{y}_{i-k}), \; 1 \leq k \leq r \} \tag{2.3}$$

contains a good approximation to $\tilde{y}_i - \tilde{a}_i$.

THEOREM 2.1. *Assume that $g(t) = f(t, y(t)) \in C^r([t_0, T])$ and f is continuously differentiable on $[t_0, T] \times \mathbb{R}^n$. Let $\mathcal{V}_i$ be the subspace defined in (2.3). Then, there exists a $\tilde{y}$ in $\tilde{a}_i + \mathcal{V}_i$ such that for $i = q+1, \dots, N$,*

$$\begin{aligned}\|\tilde{y}_i - \tilde{y}\| &= \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\varepsilon), \\ \|\tilde{G}_i(\tilde{y})\| &= \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\varepsilon).\end{aligned}$$

Proof. Since g is r-times continuously differentiable, from the Lagrange interpolation formula (see, e.g., [8, Chap. 3]), there exist constants α_l, $1 \leq l \leq r$, such that

$$\|g(t_i) - \sum_{l=1}^{r} \alpha_l g(t_{i-l})\| = \mathcal{O}(h^r). \tag{2.4}$$

From (2.4) and the fact that the scheme (1.2) is stable and of order p and f is Lipschitz, we have

$$\|f(t_i, \tilde{y}_i) - \sum_{l=1}^{r} \alpha_l f(t_{i-l}, \tilde{y}_{i-l})\| = \mathcal{O}(h^p) + \mathcal{O}(h^r) + \mathcal{O}(\varepsilon / h). \tag{2.5}$$

Now let $\tilde{y} = \tilde{a}_i + h\beta \sum_{l=1}^{r} \alpha_l f(t_{i-l}, \tilde{y}_{i-l}) \in \tilde{a}_i + \mathcal{V}_i$. Then

$$\begin{aligned}\tilde{y}_i - \tilde{y} &= \tilde{y}_i - \tilde{a}_i - h\beta \sum_{l=1}^{r} \alpha_l f(t_{i-l}, \tilde{y}_{i-l}) \\ &= \tilde{G}_i(\tilde{y}_i) + h\beta(f(t_i, \tilde{y}_i) - \sum_{l=1}^{r} \alpha_l f(t_{i-l}, \tilde{y}_{i-l}))\end{aligned}$$

which, with (2.1) and (2.5), gives

$$\|\tilde{y}_i - \tilde{y}\| = \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\varepsilon).$$

Now we have

$$\begin{aligned}\|\tilde{G}_i(\tilde{y})\| &\le \|\tilde{G}_i(\tilde{y}) - \tilde{G}_i(\tilde{y}_i)\| + \|\tilde{G}_i(\tilde{y}_i)\| \\ &= \|\tilde{y} - \tilde{y}_i + h\beta(f(t_i, \tilde{y}_i) - f(t_i, \tilde{y}))\| + \|\tilde{G}_i(\tilde{y}_i)\| \\ &\le \|\tilde{y} - \tilde{y}_i\| + h|\beta|\|f(t_i, \tilde{y}) - f(t_i, \tilde{y}_i)\| + \varepsilon.\end{aligned}$$

Hence,

$$\|\tilde{G}_i(\tilde{y})\| = \mathcal{O}(h^{r+1}) + \mathcal{O}(h^{p+1}) + \mathcal{O}(\varepsilon). \qquad \square$$

To reduce the cost in the Petrov-Galerkin process and in GMRES, we modify the subspace $\mathcal{V}_i$ by keeping only the last vectors whose computations necessitate the use of GMRES to satisfy (1.5). For example, suppose that at iteration i the Petrov-Galerkin process was enough to find $\hat{s}_i^{(k)}$ such that $y_i^{(k+1)} = y_i^{(k)} + \hat{s}_i^{(k)}$ with $\|\tilde{G}_i(y_i^{k+1})\| \le \varepsilon$. In this case we set $\tilde{s}_i^{(k)} = \hat{s}_i^{(k)}$, $\tilde{y}_i = y_i^{(k+1)}$ and use the same subspace $\mathcal{V}_i$ for the next iteration.

Let us denote, again, by

$$\mathcal{V}_i = \text{span}\{f(t_{i-j}, \tilde{y}_{i-j}),\ j = i_1, i_2, \ldots, i_s\} \tag{2.6}$$

the subspace associated with the last s vectors $\tilde{y}_{i-i_1}, \tilde{y}_{i-i_2}, \ldots, \tilde{y}_{i-i_s}$ with $i_1 < \ldots < i_s$, whose computations necessitate the use of GMRES to satisfy (1.5), and by

$$\mathcal{W}_i = \text{span}\{f(t_{i-j}, \tilde{y}_{i-j}),\ j = 1, 2, \ldots, i_1 - 1\}, \tag{2.7}$$

the set associated with the other vectors such that

$$\|\tilde{G}_{i-j}(\tilde{y}_{i-j})\| \le \varepsilon, \ \ \tilde{y}_{i-j} \in \tilde{a}_{i-j} + \mathcal{V}_i.$$

Such a situation is encountered in practice; see steps (TR3) and (N1.1) of Algorithm 4.1. Then we have the following theorem.

THEOREM 2.2. *Assume that $g(t) = f(t, y(t)) \in C^r([t_0, T])$ with $r = s + i_1 - 1$ and f continuously differentiable on $[t_0, T] \times \mathbb{R}^n$. Let $\mathcal{V}_i$ be the subspace defined in (2.6). Then there exists $\tilde{y}$ in $\tilde{a}_i + \mathcal{V}_i$ such that for $i = q + 1, \ldots, N$,*

$$\begin{aligned}\|\tilde{y}_i - \tilde{y}\| &= \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\varepsilon), \\ \|\tilde{G}_i(\tilde{y})\| &= \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\varepsilon).\end{aligned}$$

Proof. From Theorem 2.1 we know that there exist $z_1 \in \tilde{a}_i + \mathcal{V}_i$, $z_2 \in \mathcal{W}_i$, such that

$$\|\tilde{y}_i - (z_1 + z_2)\| = \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\varepsilon). \tag{2.8}$$

Let us write $z_2 = \beta h \sum_{j=1}^{i_1-1} \gamma_j f(t_{i-j}, \tilde{y}_{i-j})$ and define $\tilde{y} = z_1 + \sum_{j=1}^{i_1-1} \gamma_j (\tilde{y}_{i-j} - \tilde{a}_{i-j})$. Note that $\tilde{y} \in \tilde{a}_i + \mathcal{V}_i$ and

$$\tilde{y}_i - \tilde{y} = (\tilde{y}_i - (z_1 + z_2)) + \sum_{j=1}^{i_1-1} \gamma_j \left(h\beta f(t_{i-j}, \tilde{y}_{i-j}) - (\tilde{y}_{i-j} - \tilde{a}_{i-j}) \right),$$

$$\begin{aligned} \|\tilde{y}_i - \tilde{y}\| &\leq \|(\tilde{y}_i - (z_1 + z_2))\| + \sum_{j=1}^{i_1-1} |\gamma_j| \|\tilde{G}_{i-j}(\tilde{y}_{i-j})\| \\ &= \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\varepsilon). \end{aligned}$$

As in the proof of the previous theorem, we obtain

$$\|\tilde{G}_i(\tilde{y})\| = \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\varepsilon). \qquad \square$$

REMARK 2.3.
1. In the sequel we will essentially work with the subspace $\mathcal{V}_i$ defined in (2.6).
2. In practice, $r > p$; for example, the numerical tests are carried out with $r = 20$ and $p = 1$ or 2. Thus, $\|\tilde{y}_i - \tilde{y}\|$ and $\|\tilde{G}_i(\tilde{y})\|$ behave like $\mathcal{O}(h^{p+1}) + \mathcal{O}(\varepsilon)$.

3. Line search and trust region algorithms. Line search and trust region algorithms are two simple iterative methods for finding a local minimum of a function. They are based on the notion of descent directions that move the iterates towards a local minimum. More precisely, let

$$h_i(u) = \frac{1}{2}\|G_i(u)\|^2, \; u \in \mathbb{R}^n. \tag{3.1}$$

A vector $p \in \mathbb{R}^n$ is a descent direction of G_i at u if

$$\nabla h_i(u)^T p < 0. \tag{3.2}$$

This inequality guarantees that, for small positive δ,

$$h_i(u + \delta p) < h_i(u). \tag{3.3}$$

In fact, inequality (3.2) allows that h_i decreases with a rate proportional to $\nabla h_i(u)^T p$.

In the next two subsections we briefly explain how these algorithms can be used to find a good initial guess for the nonlinear system (1.2). More details on these algorithms can found, for example, in [10].

3.1. Backtracking line search algorithm. The line search algorithm computes a sequence of vectors $u_i^{(k)}$ such that

$$u_i^{(k+1)} = u_i^{(k)} + \lambda_k p_i^{(k)}, \tag{3.4}$$

where $p_i^{(k)}$ is a descent direction of G_i at $u_i^{(k)}$, and λ_k is a scalar chosen to satisfy the Goldstein-Armijo condition:

$$h_i(u_i^{(k+1)}) \leq h_i(u_i^{(k)}) + \alpha \lambda_k \nabla h_i(u_i^{(k)})^T p_i^{(k)}, \tag{3.5}$$

where $\alpha \in (0, 1/2)$ is a parameter typically set to 10^{-4}.

ETNA
Kent State University
http://etna.math.kent.edu

Under some conditions, the sequence $(u_i^{(k)})_{k\geq 0}$ converges to a local minimum of h_i or the sequence $\nabla h_i(u_i^{(k)})$ converges to 0. Moreover, for each descent direction $p_i^{(k)}$, Theorem 6.3.2 in [10] shows the existence of λ_k, such that (3.5) is satisfied.

The scalar λ_k is computed by the backtracking method. This method starts with $\lambda_k = 1$ and repeatedly reduces λ_k until an acceptable iterate $u_i^{(k+1)}$, satisfying (3.5), is found. When $\lambda_k = 1$, the natural question is which descent direction $p_i^{(k)}$ leads approximately to (3.5) and $u_i^{(k+1)} \approx y_i$. These conditions translate to

$$0 \lessapprox \frac{1}{2}\|G_i(u_i^{(k)})\|^2 + \alpha G_i(u_i^{(k)})^T G_i'(u_i^{(k)}) p_i^{(k)}. \tag{3.6}$$

It turns out that the Newton direction

$$p_i^{(k)} = -(G_i'(u_i^{(k)}))^{-1} G_i(u_i^{(k)}) \tag{3.7}$$

guarantees (3.6). However, this direction is expensive to compute. Therefore, using GMRES, we look for an approximation $\tilde{p}_i^{(k)}$ to $p_i^{(k)}$ that satisfies

$$\|G_i'(u_i^{(k)})\tilde{p}_i^{(k)} + G_i(u_i^{(k)})\| \leq \eta_i^{(k)} \|G_i(u_i^{(k)})\|, \tag{3.8}$$

with some tolerance threshold $\eta_i^{(k)} > 0$. Then we have

$$\begin{aligned}\nabla h_i(u_i^{(k)})^T \tilde{p}_i^{(k)} &= -\|G_i(u_i^{(k)})\|^2 + G_i(u_i^{(k)})^T \left(G_i'(u_i^{(k)})\tilde{p}_i^{(k)} + G_i(u_i^{(k)}) \right) \\ &\leq (\eta_i^{(k)} - 1)\|G_i(u_i^{(k)})\|^2.\end{aligned}$$

In particular, if $\eta_i^{(k)} \ll 1$, then $\tilde{p}_i^{(k)}$ is a good descent direction.

In conclusion, when $\tilde{p}_i^{(k)}$ satisfies (3.8) with $0 < \eta_i^{(k)} \ll 1$, then $\tilde{p}_i^{(k)}$ is a descent direction that satisfies, or almost satisfies, (3.5) with $\lambda_k = 1$. In the latter case, the backtracking method will rapidly find a scalar λ_k that satisfies (3.5) and the sequence $u_i^{(k)}$ converges to y_i. As we have mentioned, $\tilde{p}_i^{(k)}$ is obtained with GMRES. To accelerate the computation, we start GMRES with $\hat{p}_i^{(k)}$, obtained by the Petrov-Galerkin process applied to the subspace $-u_i^{(k)} + \tilde{a}_i + \mathcal{V}_i$. With this initial solution, only a few iterations, and sometimes no iteration, of GMRES are needed to satisfy (3.8); see Tables 4.1 and 4.2.

One drawback of the line search algorithm is that even when the condition (3.5) is not satisfied, the descent direction is still kept unchanged. Only the scalar λ_k is updated using a one-dimensional quadratic or cubic model. The trust region algorithm overcomes this drawback, at the price of increasing the computational cost.

3.2. Trust region algorithm. The trust region algorithm generates a sequence of vectors $u_i^{(k)}$, such that $u_i^{(k+1)} = u_i^{(k)} + d_i^{(k)}$, where $d_i^{(k)}$ is the solution of

$$\min_{\|d\| \leq \delta_i^{(k)}} \Psi_i^{(k)}(d), \tag{3.9}$$

and where $\delta_i^{(k)} > 0$ and $\Psi_i^{(k)}$ is given by the quadratic model

$$\begin{aligned}\Psi_i^{(k)}(d) &= \frac{1}{2}\|G_i(u_i^{(k)}) + G_i'(u_i^{(k)})d\|^2 \\ &= h_i(u_i^{(k)}) + \nabla h_i(u_i^{(k)})^T d + \frac{1}{2} d^T B_i^{(k)} d\end{aligned} \tag{3.10}$$

ETNA
Kent State University
http://etna.math.kent.edu

with $B_i^{(k)} = G_i'(u_i^{(k)})^T G_i'(u_i^{(k)})$.

Since $B_i^{(k)}$ is symmetric and positive definite, the solution of (3.9) is given by

$$d_i^{(k)} = \begin{cases} -(B_i^{(k)})^{-1}\nabla h_i(u_i^{(k)}) & \text{if } \|(B_i^{(k)})^{-1}\nabla h_i(u_i^{(k)})\| \le \delta_i^{(k)}, \\ -(B_i^{(k)} + \mu_i^{(k)} I_n)^{-1}\nabla h_i(u_i^{(k)}) & \text{if } \|(B_i^{(k)})^{-1}\nabla h_i(u_i^{(k)})\| > \delta_i^{(k)}, \end{cases}$$

where $\mu_i^{(k)} \ge 0$ solves

$$\|(B_i^{(k)} + \mu_i^{(k)} I_n)^{-1}\nabla h_i(u_i^{(k)})\| = \delta_i^{(k)}. \tag{3.11}$$

Note that $d_i^{(k)}$ is a descent direction of G_i at $u_i^{(k)}$ and coincides with the Newton direction (3.7) when $\|(B_i^{(k)})^{-1}\nabla h_i(u_i^{(k)})\| \le \delta_i^{(k)}$.

To compute the scalar $\mu_i^{(k)}$ that approximately satisfies (3.11), we will use two methods: the locally constrained optimal hook step method, and the dogleg step method. The former finds $\mu_i^{(k)}$, such that $\|(B_i^{(k)} + \mu_i^{(k)} I_n)^{-1}\nabla h_i(u_i^{(k)})\| \cong \delta_i{}^{(k)}$, and takes $u_i^{(k+1)} = u_i^{(k)} - (B_i^{(k)} + \mu_i^{(k)} I_n)^{-1}\nabla h_i(u_i^{(k)})$. The latter uses a piecewise linear approximation of the curve $\mu \to u_i^{(k)} - (B_i^{(k)} + \mu I_n)^{-1}\nabla h_i(u_i^{(k)})$ and takes $u_i^{(k+1)}$ as the point on this approximation, such that $\|u_i^{(k+1)} - u_i^{(k)}\| = \delta_i^{(k)}$. Note that both methods, as well as the computation of $d_i^{(k)}$, necessitate the solution of large linear systems. To reduce the cost, we proceed as in [5] and replace the full quadratic model (3.10) by the following lower-dimensional one,

$$\Phi_i^{(k)}(c) = h_i(u_i^{(k)}) + \nabla h_i(u_i^{(k)})^T V_i c + \frac{1}{2} c^T (V_i^T B_i^{(k)} V_i) c, \tag{3.12}$$

where V_i is an $n \times p$ matrix whose columns form an orthonormal basis of $\mathcal{V}_i$. Since $V_i^T B_i^{(k)} V_i$ is symmetric positive definite, the next iterate $u_i^{(k+1)}$ is given by $u_i^{(k)} + V_i c_i^{(k)}$, where $c_i^{(k)}$ is the solution of

$$\min_{\|c\| \le \delta_i^{(k)}} \Phi_i^{(k)}(c). \tag{3.13}$$

Note that $c_i^{(k)}$ is not expensive to compute and that $V_i c_i^{(k)}$ is a descent direction of G_i at $u_i^{(k)}$. The condition for accepting $u_i^{(k+1)}$ is the one written in (3.5), namely,

$$h_i(u_i^{(k+1)}) \le h_i(u_i^{(k)}) + \alpha \nabla h_i(u_i^{(k)})^T (u_i^{(k+1)} - u_i^{(k)}), \tag{3.14}$$

where $\alpha \in (0, \frac{1}{2})$. If $u_i^{(k+1)}$ does not satisfy (3.14), then we reduce $\delta_i^{(k)}$ and return to (3.13) to compute $c_i^{(k)}$ by the hook step or the dogleg step method. If $u_i^{(k+1)}$ satisfies (3.14), then one must decide whether $\delta_i^{(k+1)}$ should be increased, decreased, or kept the same for the next step $k+1$; see [10].

4. Algorithmic aspect. In this section we write, in an informal way, an algorithm that computes the sequence $(\tilde{y}_i)$ defined in (2.1). The initial guess for each nonlinear system is obtained with a backtracking line search (part 3.1) or trust region (part 3.2), and the approximate solution $\tilde{y}_i$ is obtained with the inexact Newton method (part 3.3). The parameters r, $\varepsilon_{LS}^{(1)}$, $\varepsilon_{LS}^{(2)}$, ε, $kmax_{LS}$ and $kmax_{IN}$ should be fixed by the user.

ETNA
Kent State University
http://etna.math.kent.edu

4.1. The algorithm.

ALGORITHM 4.1.

1. Assume that for $k = 1, \ldots, q$, the approximation $\tilde{y}_k$ is either given or is computed with an k-step scheme such that $\max_{1\le k\le q} \|\tilde{y}_k - y_k\| = \mathcal{O}(\varepsilon/h)$.
   Set $i = q + 1$.
2. Let R_i be the matrix formed by the last k_0 vectors $f(t_{i-k}, \tilde{y}_{i-k})$, $1 \le k \le k_0$, where $k_0 = \min(q + 1, r)$. Orthonormalize the columns of R_i in V_i.
3. Repeat until $i = N$
   - 3.0 Compute an initial solution $u_i^{(0)}$ (e.g., by an explicit method applied to (1.1))
   - 3.1 *Line Search:* Repeat until
     $\|\tilde{G}_i(u_i^{(k)})\| \le \varepsilon_{LS}^{(1)}$ or $\left| \|\tilde{G}_i(u_i^{(k+1)})\| - \|\tilde{G}_i(u_i^{(k)})\| \right| \le \varepsilon_{LS}^{(2)}$ or $k > kmax_{LS}$
     - (LS1) Compute an initial solution $\hat{p}_i^{(k)}$ of (3.7) by applying the Petrov-Galerkin process to $-u_i^{(k)} + \tilde{a}_i + \mathrm{range}(V_i)$.
       - If $\hat{p}_i^{(k)}$ satisfies (3.8), then set $\tilde{p}_i^{(k)} = \hat{p}_i^{(k)}$.
       - Otherwise, compute an approximate solution $\tilde{p}_i^{(k)}$ of (3.7) by GMRES, so that (3.8) holds, starting with $\hat{p}_i^{(k)}$.
     - (LS2) Compute the scalar λ_k by the backtracking method, so that (3.5) holds.
     - (LS3) Set $u_i^{(k+1)} = u_i^{(k)} + \lambda_k \tilde{p}_i^{(k)}$, $k := k + 1$, $y_i^{(0)} = u_i^{(k)}$.
   - 3.2 *Trust Region:* replace the steps (LS1), (LS2), and (LS3) above by
     - (TR1) Compute an approximation $c_i^{(k)}$ to (3.13) by the hook step or the dogleg step.
     - (TR2) Compute $u_i^{(k+1)} = u_i^{(k)} + V_i c_i^{(k)}$ satisfying (3.14).
       Set $k := k + 1$, $y_i^{(0)} = u_i^{(k)}$.
     - (TR3) If $\|\tilde{G}_i(y_i^{(0)})\| \le \varepsilon$, then set $\tilde{y}_i = y_i^{(0)}$, $i := i + 1$ and go to 3.
   - 3.3 *Inexact Newton:* Repeat until
     $$\|\tilde{G}_i(y_i^{(k)})\| \le \varepsilon \text{ or } k > kmax_{IN}$$
     - (N1) Compute an approximation $\hat{s}_i^{(k)}$ of the linear system (1.3) by applying the Petrov-Galerkin process to $-y_i^{(k)} + \tilde{a}_i + \mathrm{range}(V_i)$.
       - (N1.1) If $\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\| \le \varepsilon$, set $\tilde{y}_i = y_i^{(k)} + \hat{s}_i^{(k)}$, $i := i + 1$ and go to 3.
       - (N1.2) Else, if $\hat{s}_i^{(k)}$ satisfies (1.5), set $y_i^{(k+1)} = y_i^{(k)} + \hat{s}_i^{(k)}$, $k := k + 1$ and go to 3.3.
       - (N1.3) Else, compute an approximation $\tilde{s}_i^{(k)}$ to (1.3) by GMRES, starting with $\hat{s}_i^{(k)}$. Set $\hat{s}_i^{(k)} = \tilde{s}_i^{(k)}$, and go to (N1.1).
   - 3.4 Set $\tilde{y}_i = y_i^{(k)}$.
   - 3.5 Let k_0 be the number of columns of R_i. If $k_0 < r$, then set $R_{i+1} = [R_i, f(t_i, \tilde{y}_i)]$, else $R_i = [S_i, f(t_i, \tilde{y}_i)]$, where S_i is the matrix formed by the last $r - 1$ columns of R_i. Orthonormalize the columns of R_{i+1} in V_{i+1}
   - 3.6 $i := i + 1$.

In this algorithm, the subspace $\mathcal{V}_i = \mathrm{range}(V_i)$ corresponds to the one of Theorem 2.2. If we want to use the subspace $\mathcal{V}_i$ of Theorem 2.1, then step (TR3) should be replaced by:

$$\text{If } \|\tilde{G}_i(y_i^{(0)})\| \le \varepsilon, \text{ set } \tilde{y}_i = y_i^{(0)}, \text{ and go to 3.5,}$$

and step (N1.1) should be replaced by

$$\text{If } \|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\| \le \varepsilon, \text{ set } \tilde{y}_i = y_i^{(k)} + \hat{s}_i^{(k)}, \text{ and go to 3.5.}$$

Note that if step (TR3) or step (N1.1) is satisfied, then $\tilde{y}_i \in \tilde{a}_i + \mathcal{V}_i$. For this reason we do not change the subspace $\mathcal{V}_i$ for the next iteration $i+1$.

In step 3.4 the orthonormalization of R_i and R_{i+1} uses an updating QR factorization; see, e.g., [14, p. 594].

4.2. Numerical results. We now show the behavior of Algorithm 4.1 on the Robertson chemical reaction with one-dimensional diffusion [17]:

$$\begin{aligned}\frac{\partial u}{\partial t} &= -0.04\,u + 10^4 vw + \alpha\frac{\partial^2 u}{\partial x^2},\\ \frac{\partial v}{\partial t} &= 0.04\,u - 10^4 vw - 3\times 10^7 v^2 + \alpha\frac{\partial^2 v}{\partial x^2},\\ \frac{\partial w}{\partial t} &= 3\times 10^7 v^2 + \alpha\frac{\partial^2 w}{\partial x^2},\end{aligned}$$

with $0 \le x \le 1, 0 \le t \le 1$, $\alpha = 2\times 10^{-2}$ and Neumann boundary conditions $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} = 0$ at $x = 0, 1$, and initial values

$$u(x,0) = 1 + \sin(2\pi\, x), \quad v(x,0) = w(x,0) = 0.$$

The second partial derivative is discretized on a uniform grid of 5×10^3 points and the boundary conditions at $x = 0$ (resp. $x = 1$) are discretized by forward (resp. backward) differences. We thus obtain an ODE system of the form (1.1) of size $n = 15\times 10^3$. This system is known to be very stiff.

Recall that our aim is to show that the subspace $\mathcal{V}_i$ helps to find good initial solutions $y_i^{(0)}$ and $\hat{s}_i^{(k)}$. The implicit schemes used in (1.2) are implicit Euler and Crank-Nicolson. The numerical tests are carried out with the following parameters: $q = 0$, $r \equiv \dim(\mathcal{V}_i) = 20$, $h = 1/100$. The parameter ε in the inexact Newton method is given by 10^{-5}. The threshold parameters $\epsilon_i^{(k)}$ and $\eta_i^{(k)}$, used in the computation of $\tilde{s}_i^{(k)}$ in (1.5) and $\tilde{p}_i^{(k)}$ in (3.8), are fixed at 10^{-2}. Note that the choice of a constant $\epsilon_i^{(k)}$ implies that the convergence of the sequence $(y_i^{(k)})_{k\ge 0}$ is linear [9]. In steps (LS1) and (N.1.3), the computations of $\tilde{p}_i^{(k)}$ and $\tilde{s}_i^{(k)}$ are carried out by GMRES(20), i.e., restarted GMRES with restart value 20. We say that GMRES fails to converge when the stopping criterion is not satisfied within a total number of iterations (i.e., number of matrix-vector multiplies) fixed at 10^4. The parameters $kmax_{_{IN}}$, $kmax_{_{LS}}$, $\varepsilon_{_{LS}}^{(1)}$ and $\varepsilon_{_{LS}}^{(2)}$ are given by 15, 15, 10^{-2} and 10^{-6}, respectively. The initial solution $u_i^{(0)}$ in step 3.0 is given by explicit Euler, i.e., $u_i^{(0)} = \tilde{y}_{i-1} + hf(t_{i-1}, \tilde{y}_{i-1})$.

We also compare the proposed approach with some "standard predictor schemes" where the initial solutions to (1.3) and to line search are obtained with explicit Euler, explicit Runge-Kutta of order 4, or Adams-Bashforth of order r. Specifically, we remove steps 2 and 3.5 from Algorithm 4.1 and replace $u_i^{(0)}$ in step 3.0, $\hat{p}_i^{(k)}$ in step (LS1) and $\hat{s}_i^{(k)}$ in step (N1), respectively, by

$$u_i^{(0)} = y_i^{(e)},\ \hat{p}_i^{(k)} = -u_i^{(k)} + y_i^{(e)} \text{ and } \hat{s}_i^{(k)} = -y_i^{(k)} + y_i^{(e)},$$

where

$$y_{i+1}^{(e)} = \tilde{y}_i + hf(t_i, \tilde{y}_i)$$

for explicit Euler,

$$y_{i+1}^{(e)} = \tilde{y}_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

ETNA
Kent State University
http://etna.math.kent.edu

for Runge-Kutta 4 with $k_1 = f(t_i, \tilde{y}_i)$, $k_2 = f(t_i + h/2, \tilde{y}_i + hk_1/2)$,
$k_3 = f(t_i + h/2, \tilde{y}_i + hk_2/2)$, $k_4 = f(t_{i+1}, \tilde{y}_i + hk_3)$, and

$$y_{i+1}^{(e)} = \tilde{y}_i + h \sum_{k=0}^{r-1} \beta_k \nabla^k f(t_i, \tilde{y}_i)$$

for Adams-Bashforth with $\beta_k = \int_0^1 \prod_{j=0}^{k-1}(j+s)ds/k!$, $\nabla^0 f(t_i, \tilde{y}_i) = f(t_i, \tilde{y}_i)$ and $\nabla^{k+1} f(t_i, \tilde{y}_i) = \nabla^k f(t_i, \tilde{y}_i) - \nabla^k f(t_{i-1}, \tilde{y}_{i-1})$.

Figures 4.1-4.3 show the residual norm $\|\tilde{G}_i(y_i^{(0)})\|$, where $y_i^{(0)}$ is the initial solution computed by line search or trust region algorithm.

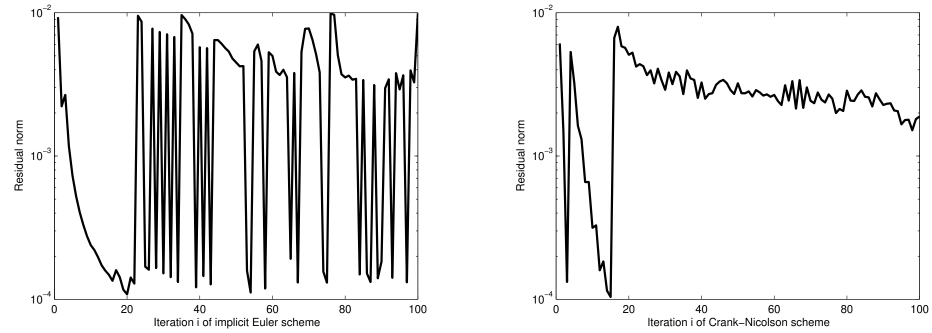

FIG. 4.1. *Line search.*

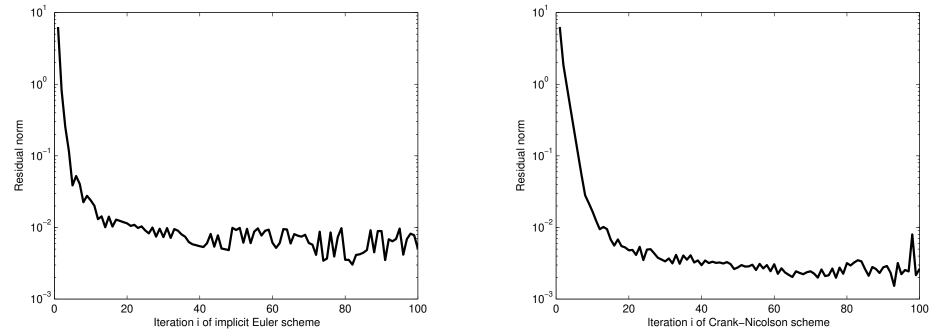

FIG. 4.2. *Trust region (dogleg step).*

Tables 4.1 and 4.2 show some computational details when the inexact Newton method is used along with the line search algorithm. The second column of these tables shows $\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\|$, where $y_i^{(k)}$ is the k-th Newton iterate for computing y_i and $\hat{s}_i^{(k)}$ is the initial guess of $s_i^{(k)}$, obtained from $\mathcal{V}_i$ and the Petrov-Galerkin process. The third column shows the number of iterations required by GMRES to compute $\tilde{s}_i^{(k)}$, starting with $\hat{s}_i^{(k)}$, such that (1.5) holds. For example, when $i = 1$, Table 4.1 shows that $4329 + 5440 = 9769$ iterations of GMRES have been used to satisfy the condition in step 3.3 of Algorithm 4.1, while 2320 iterations were sufficient when $i = 10$. Columns 4, 5, and 6 of these tables show analogous information for the line search (step 3.1 of Algorithm 4.1).

ETNA
Kent State University
http://etna.math.kent.edu

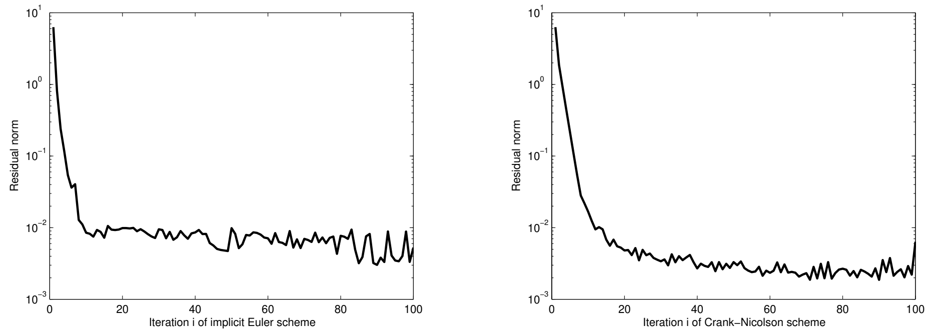

FIG. 4.3. *Trust region (hook step).*

From these tables, we see that at iteration $i = 1$, the number of iterations required by GMRES is large and that the initial solutions $\hat{s}_i^{(k)}$ and $\hat{p}_i^{(k)}$ are not good enough. This is because the subspace $\mathcal{V}_i$ contains only one vector. As $\dim(\mathcal{V}_i)$ increases, the relative residual associated with $\hat{p}_i^{(k)}$ decreases, especially when the Crank-Nicolson scheme is used. As a consequence, fewer iterations, and sometimes no iteration, of GMRES, are needed to compute $\tilde{p}_i^{(k)}$. This allows the line search algorithm to compute a good initial solution $y_i^{(0)}$; see also Figure 4.1. Similar comments apply when inexact Newton is used with trust region; see Tables 4.3-4.4 and Figures 4.2-4.3.

Table 4.5 shows the behavior of Algorithm 4.1 when the explicit Euler scheme is used as explained above. In this test GMRES did not converge. The norm $\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\|$ stagnated around 10^4. The algorithm did not perform better when explicit Runge-Kutta 4 and Adams-Bashforth were used. With Runge-Kutta, $\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\|$ stagnated around 10^{24} for $i = 1$, $k = 0, \ldots, 12$, and at iteration $(i, k) = (1, 13)$, GMRES failed to compute $\tilde{s}_i^{(k)}$. With Adams-Bashforth, $\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\|$ stagnated around 10^4 for $i = 1$, $k = 0, 1$ and $i = 2$, $k = 0, \ldots, 5$. At iteration $(i, k) = (2, 6)$, GMRES failed to compute $\tilde{s}_i^{(k)}$.

TABLE 4.1
Algorithm 4.1 with line search - Implicit Euler is used in (1.2).

Inexact Newton			Line search		
iteration (i,k)	$\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\|$	# iter. GMRES $\tilde{s}_i^{(k)}$	iteration (i,k)	$\frac{\|\tilde{G}_i'(u_i^{(k)})\hat{p}_i^{(k)} + \tilde{G}_i(u_i^{(k)})\|}{\|\tilde{G}_i(u_i^{(k)})\|}$	# iter. GMRES $\tilde{p}_i^{(k)}$
(1,0)	6,2976	4329	(1,0)	$1,8881 \times 10^{-4}$	0
(1,1)	6,2976	5440	(1,1)	$9,9999 \times 10^{-1}$	1192
(10,0)	$4,9076 \times 10^{-3}$	2320	(1,2)	1,6321	557
(20,0)	$2,6774 \times 10^{-3}$	1760	(1,3)	$2,1002 \times 10^{1}$	2641
(30,0)	$2,1495 \times 10^{-3}$	1620	(1,4)	$9,8262 \times 10^{1}$	3393
(40,0)	$1,9993 \times 10^{-3}$	893	(10,0)	$2,0496 \times 10^{-1}$	910
(40,1)	$1,9993 \times 10^{-3}$	1600	(20,0)	$2,4655 \times 10^{-1}$	793
(50,0)	$1,6472 \times 10^{-3}$	536	(30,0)	$1,4118 \times 10^{-1}$	520
(50,1)	$1,6472 \times 10^{-3}$	940	(90,0)	$6,8481 \times 10^{-2}$	130
(90,0)	$1,2606 \times 10^{-3}$	900			
(100,0)	$1,3811 \times 10^{-3}$	201			
(100,1)	$1,3811 \times 10^{-3}$	1000			

5. Conclusion. The purpose of this paper was to show one possibility for improving convergence of the linear and nonlinear systems that arise when solving large nonlinear sys-

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 4.2
Algorithm 4.1 with line search - Crank-Nicolson is used in (1.2).

Inexact Newton			Line search		
iteration (i,k)	$\lVert\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\rVert$	# iter. GMRES $\tilde{s}_i^{(k)}$	iteration (i,k)	$\frac{\lVert\tilde{G}_i'(u_i^{(k)})\tilde{p}_i^{(k)} + \tilde{G}_i(u_i^{(k)})\rVert}{\lVert\tilde{G}_i(u_i^{(k)})\rVert}$	# iter. GMRES $\tilde{p}_i^{(k)}$
(1,0)	6,2969	2347	(1,0)	$2,5662 \times 10^{-4}$	0
(1,1)	6,2969	2800	(1,1)	$9,9999 \times 10^{-1}$	642
(10,0)	$1,6742 \times 10^{-2}$	1540	(1,2)	1,8962	760
(20,0)	$4,9327 \times 10^{-3}$	817	(1,3)	6,5812	890
(20,1)	$4,9327 \times 10^{-3}$	1220	(1,4)	$2,7293 \times 10^{1}$	1405
(30,0)	$2,8972 \times 10^{-3}$	676	(1,5)	$1,3141 \times 10^{2}$	1837
(30,1)	$2,8972 \times 10^{-3}$	940	(10,0)	$9,6104 \times 10^{-6}$	0
(50,0)	$3,2204 \times 10^{-3}$	815	(10,1)	$5,2984 \times 10^{-1}$	728
(50,1)	$3,2204 \times 10^{-3}$	1100	(20,0)	$6,3566 \times 10^{-7}$	0
(80,0)	$2,8631 \times 10^{-3}$	686	(60,0)	$8,6954 \times 10^{-8}$	0
(80,1)	$2,8631 \times 10^{-3}$	960	(80,0)	$9,3942 \times 10^{-8}$	0
(100,0)	$1,8893 \times 10^{-3}$	547	(90,0)	$7,4865 \times 10^{-8}$	0
(100,1)	$1,8893 \times 10^{-3}$	700	(100,0)	$6,2467 \times 10^{-8}$	0

TABLE 4.3
Algorithm 4.1 with trust region - Implicit Euler is used in (1.2).

iteration	Inexact Newton & dogleg step		Inexact Newton & hook step	
(i,k)	$\lVert\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\rVert$	# iter. GMRES $\tilde{s}_i^{(k)}$	$\lVert\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\rVert$	# iter. GMRES $\tilde{s}_i^{(k)}$
(1,0)	6,2976	1192	6,2976	1192
(1,1)	6,2976	1057	6,2976	1057
(1,2)	6,2976	1318	6,2976	1317
(1,3)	6,2976	2393	6,2976	2393
(1,4)	6,2976	3157	6,2976	3157
(1,5)	6,2976	4007	6,2976	4007
(1,6)	6,2976	5292	6,2976	5292
(1,7)	6,2976	5440	6,2976	5440
(10,0)	$4,9086 \times 10^{-3}$	910	$4,9072 \times 10^{-3}$	1361
(10,0)	$4,9086 \times 10^{-3}$	2320	$4,9072 \times 10^{-3}$	2320
(20,0)	$3,3623 \times 10^{-3}$	1022	$2,4509 \times 10^{-3}$	762
(20,1)	$3,3623 \times 10^{-3}$	2040	$2,4509 \times 10^{-3}$	1640
(30,0)	$2,1081 \times 10^{-3}$	1016	$1,9199 \times 10^{-3}$	766
(30,1)	$2,1081 \times 10^{-3}$	1960	$1,9199 \times 10^{-3}$	1720
(40,0)	$1,3871 \times 10^{-3}$	435	$2,6895 \times 10^{-3}$	1100
(40,1)	$1,3871 \times 10^{-3}$	900	$2,6895 \times 10^{-3}$	2080
(50,0)	$1,8442 \times 10^{-3}$	1007	$1,4765 \times 10^{-3}$	640
(50,1)	$1,8442 \times 10^{-3}$	2080	$1,4765 \times 10^{-3}$	1600
(60,0)	$2,1950 \times 10^{-3}$	481	$2,1745 \times 10^{-3}$	836
(60,1)	$2,1950 \times 10^{-3}$	1320	$2,1745 \times 10^{-3}$	1560
(70,0)	$2,5777 \times 10^{-3}$	1282	$1,9232 \times 10^{-3}$	780
(70,1)	$2,5777 \times 10^{-3}$	2140	$1,9232 \times 10^{-3}$	1440
(80,0)	$1,7480 \times 10^{-3}$	633	$2,4491 \times 10^{-3}$	769
(80,1)	$1,7480 \times 10^{-3}$	980	$2,4491 \times 10^{-3}$	1640
(90,0)	$1,4131 \times 10^{-3}$	433	$1,3438 \times 10^{-3}$	502
(90,1)	$1,4131 \times 10^{-3}$	1020	$1,3438 \times 10^{-3}$	980
(100,0)	$1,7451 \times 10^{-3}$	791	$1,7411 \times 10^{-3}$	381
(100,1)	$1,7451 \times 10^{-3}$	1520	$1,7411 \times 10^{-3}$	1060

tems of ODEs by implicit schemes.

The nonlinear systems are solved by the Inexact Newton (IN) method and the linear systems that arise in IN are solved by GMRES. The convergence of both IN and GMRES can greatly be improved if good initial solutions are available. To this end, we have developed a strategy that allows the extraction of good initial solutions for IN and for the linear systems in IN. The strategy uses, at iteration i of the implicit scheme, a subspace $\mathcal{V}_i$ of small dimension that contains information on the last r iterates, the line search LS or trust region TR algorithm, and the Petrov-Galerkin process.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 4.4
Algorithm 4.1 with trust region - Crank-Nicolson is used in (1.2).

iteration	Inexact Newton & dogleg step		Inexact Newton & hook step	
(i,k)	$\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\|$	# iter. GMRES $\tilde{s}_i^{(k)}$	$\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\|$	# iter. GMRES $\tilde{s}_i^{(k)}$
(1,0)	6,2969	642	6,2969	642
(1,1)	6,2969	760	6,2969	760
(1,2)	6,2969	890	6,2969	890
(1,3)	6,2969	1405	6,2969	1405
(1,4)	6,2969	1837	6,2969	1837
(1,5)	6,2969	2347	6,2969	2347
(1,6)	6,2969	2800	6,2969	2800
(10,0)	$1,6748 \times 10^{-2}$	877	$1,6748 \times 10^{-2}$	877
(10,1)	$1,6748 \times 10^{-2}$	1540	$1,6748 \times 10^{-2}$	1540
(20,0)	$4,8107 \times 10^{-3}$	839	$4,8091 \times 10^{-3}$	839
(20,1)	$4,8107 \times 10^{-3}$	1220	$4,8091 \times 10^{-3}$	1220
(30,0)	$3,3681 \times 10^{-3}$	707	$3,3973 \times 10^{-3}$	701
(30,1)	$3,3681 \times 10^{-3}$	1020	$3,3973 \times 10^{-3}$	1020
(40,0)	$2,9851 \times 10^{-3}$	807	$2,6970 \times 10^{-3}$	825
(40,1)	$2,9851 \times 10^{-3}$	1100	$2,6970 \times 10^{-3}$	1080
(50,0)	$2,7778 \times 10^{-3}$	846	$3,3073 \times 10^{-3}$	832
(50,1)	$2,7778 \times 10^{-3}$	1100	$3,3073 \times 10^{-3}$	1120
(60,0)	$3,0731 \times 10^{-3}$	829	$2,3479 \times 10^{-3}$	846
(60,1)	$3,0731 \times 10^{-3}$	1060	$2,3479 \times 10^{-3}$	1040
(70,0)	$2,4460 \times 10^{-3}$	634	$2,3183 \times 10^{-3}$	613
(70,1)	$2,4460 \times 10^{-3}$	820	$2,3183 \times 10^{-3}$	760
(80,0)	$3,1872 \times 10^{-3}$	552	$2,6765 \times 10^{-3}$	482
(80,1)	$3,1872 \times 10^{-3}$	760	$2,6765 \times 10^{-3}$	660
(90,0)	$2,7831 \times 10^{-3}$	667	$1,8703 \times 10^{-3}$	471
(90,1)	$2,7831 \times 10^{-3}$	940	$1,8703 \times 10^{-3}$	600
(100,0)	$2,6554 \times 10^{-3}$	559	$2,2800 \times 10^{-3}$	262
(100,1)	$2,6554 \times 10^{-3}$	780	$2,2800 \times 10^{-3}$	720

TABLE 4.5
Explicit Euler with line search - Crank-Nicolson is used in (1.2).

Inexact Newton			Line search		
(i,k)	$\|\tilde{G}_i(y_i^{(k)} + \hat{s}_i^{(k)})\|$	# iter. GMRES $\tilde{s}_i^{(k)}$	(i,k)	$\frac{\|\tilde{G}_i'(u_i^{(k)})\tilde{p}_i^{(k)} + \tilde{G}_i(u_i^{(k)})\|}{\|\tilde{G}_i(u_i^{(k)})\|}$	# iter. GMRES $\tilde{p}_i^{(k)}$
(1,0)	$3,1434 \times 10^{4}$	1956	(1,0)	1	10
(1,1)	$3,1434 \times 10^{4}$	2140	(1,1)	$1,0511 \times 10^{2}$	80
			(1,2)	$1,0465 \times 10^{4}$	524
			(1,2)	$1,0465 \times 10^{4}$	1076
			(1,4)	$1,2660 \times 10^{6}$	1425
			(2,0)	1	10
			(2,1)	$1,0621 \times 10^{2}$	424
			(2,2)	$7,2056 \times 10^{3}$	13508
			(2,3)	$3,5319 \times 10^{4}$	*

$*$ The number of matrix-vector multiplies in GMRES largely exceeds 10^4.

The efficiency of LS depends mainly on the quality of the descent directions. The Newton direction (3.7) is known to be a good descent direction but necessitates the solution of linear systems. The fact that the proposed strategy allows the construction of a good initial solution $\hat{p}_i^{(k)}$ to (3.7), facilitates the task of GMRES. The resulting method is cheap and efficient. The TR algorithm uses a large quadratic model, which, after projection onto $\mathcal{V}_i$, leads to several small linear systems. This algorithm is also efficient but can be more expensive than LS if the latter uses a good descent direction.

Numerical tests with the approach, where the initial solutions (to (1.3) and to LS) are obtained with explicit methods, have been carried out. This approach lead to a stagnation of GMRES at early iterations of the implicit scheme.

REFERENCES

[1] M. AL SAYED ALI AND M. SADKANE, *Acceleration of implicit schemes for solving large linear systems of ordinary differential equations*, in preparation.

[2] P. N. BROWN, *A local convergence theory for combined inexact-Newton/finite-difference projection methods*, SIAM J. Numer. Anal., 24 (1987), pp. 407–434.

[3] P. N. BROWN AND A. C. HINDMARSH, *Matrix-free methods for stiff of ODE's*, SIAM J. Numer. Anal., 23 (1986), pp. 610–638.

[4] ———, *Reduced storage methods in stiff ODE systems*, J. Appl. Math. Comput., 31 (1989), pp. 40–91.

[5] P. N. BROWN AND Y. SAAD, *Hybrid Krylov methods for nonlinear systems of equations,* SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[6] ———, *Convergence theory of nonlinear Newton-Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330.

[7] M. BÜNTTNER, B. A. SCHMIDT, AND R. WEINER, *W-methods with automatic partitioning by Krylov techniques for large stiff systems,* SIAM J. Numer. Anal., 32 (1995), pp. 260–284.

[8] P. J. DAVIS, *Interpolation and Approximation*, Blaisdell, New York, 1963.

[9] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[10] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Prentice-Hall, Englewood Cliffs, NJ, 1983.

[11] S. C. EISENSTAT AND H. F. WALKER, *Globally convergent inexact Newton methods,* SIAM J. Optim., 4 (1994), pp. 393–422.

[12] E. GALLOPOULOS AND Y. SAAD, *Efficient solution of parabolic equations by Krylov approximation methods,* SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1236–1264.

[13] C. W. GEAR AND Y. SAAD, *Iterative solution of linear equations in ODE codes*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 583–601.

[14] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations,* Second ed., The John Hopkins University Press, Baltimore, MD, 1989.

[15] E. HAIRER, S. P. NORSETT, AND G. WANNER, *Solving Ordinary Differential Equations I. Nonstiff problems,* Second ed., Springer, Berlin, 1993.

[16] E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Second ed., Springer, Berlin, 1996.

[17] M. HOCHBRUCK, C. LUBICH, AND H. SELHOFER, *Exponential integrators for large systems of differential equations,* SIAM J. Sci. Comput., 19 (1998), pp. 1552–1574.

[18] E. ISAACSON AND H. B. KELLER, *Analysis of Numerical Methods,* Wiley, New York, 1966.

[19] L. O. JAY, *Inexact simplified Newton iterations for implicit Runge-Kutta methods,* SIAM J. Numer. Anal., 38 (2001), pp. 1369–1388.

[20] Y. SAAD, *Iterative Methods for Sparse Linear Systems,* Second ed., SIAM, Philadelphia, PA, 2003.

[21] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.