Electronic Transactions on Numerical Analysis.
Volume 39, pp. 253-270, 2012.
Copyright © 2012, Kent State University.
ISSN 1068-9613.

IMPROVED PREDICTOR SCHEMES FOR LARGE SYSTEMS OF LINEAR ODES[*]

MOUHAMAD AL SAYED ALI[†] AND MILOUD SADKANE[‡]

Abstract. When solving linear systems of ordinary differential equations (ODEs) with constant coefficients by implicit schemes such as implicit Euler, Crank-Nicolson, or implicit Runge-Kutta, one is faced with the difficulty of correctly solving the repeated linear systems that arise in the implicit scheme. These systems often have the same matrix but different right-hand sides. When the size of the matrix is large, iterative methods based on Krylov subspaces can be used. However, the effectiveness of these methods strongly depends on the initial guesses. The closer the initial guesses are to the exact solutions, the faster the convergence. This paper presents an approach that computes good initial guesses to these linear systems. It can be viewed as an improved predictor method. It is based on a Petrov-Galerkin process and multistep schemes and consists of building, throughout the iterations, an approximation subspace using the previous computations, where good initial guesses to the next linear systems can be found. It is shown that the quality of the computed initial guess depends only on the stepsize of the discretization and the dimension of the approximation subspace. The approach can be applied to most of the common implicit schemes. It is tested on several examples.

Key words. Convergence acceleration, implicit scheme, predictor, Petrov-Galerkin, GMRES

AMS subject classifications. 65L20, 65F10

1. Introduction.

Consider the linear system of ODEs

$$\begin{aligned} \dot{y}(t) &= Ay(t) + f(t), \quad \forall t \in [t_0, T], \\ y(t_0) &= y^{(0)}, \end{aligned} \tag{1.1}$$

which results, for example, from the method of lines applied to a linear time-dependent partial differential equation [15]. We assume that A is a real, large $n \times n$ matrix and $f : [t_0, T] \to \mathbb{R}^n$ is a sufficiently smooth function.

The system (1.1) is subsequently discretized in time. Using an implicit scheme over a uniform mesh, $t_i = t_0 + ih, i = 0, \ldots, N$, with $h = \frac{T-t_0}{N}$ and denoting by y_i an approximation of $y(t_i)$, the system (1.1) becomes

$$\begin{aligned} y_{i+1} &= y_i + hz_i, \quad i = q, q+1, \ldots, N-1, \\ y_0 &= y^{(0)}, \end{aligned} \tag{1.2}$$

where z_i is the solution of the linear system

$$Cz_i = b_i \tag{1.3}$$

with $C = I_n - \beta_{-1}hA$ and b_i is given by

$$b_i = \beta_{-1}Ay_i + \gamma_{-1}f(t_{i+1}) + \sum_{k=0}^{q} \left(\beta_k Ay_{i-k} + \gamma_k f(t_{i-k})\right), \tag{1.4}$$

where β_k and γ_k are given constants. The vectors $y_1, \ldots, y_q$ are either given or computed by another scheme.

[*] Received June 30, 2009. Accepted May 30, 2012. Published online on July 9, 2012. Recommended by M. Gander.

[†] Laboratoire de Biogeosciences - Université de Bourgogne, 6 Boulevard Gabriel, 21000 Dijon, France (moh.sayedal@gmail.com).

[‡] Université de Brest, CNRS - UMR 6205, Laboratoire de Mathématiques de Bretagne Atlantique, 6 Av. Le Gorgeu, 29238 Brest Cedex 3, France (sadkane@univ-brest.fr).

Most standard implicit methods can be formulated as (1.2)–(1.4) and satisfy these assumptions. For example, the implicit Euler method corresponds to $\beta_{-1} = \gamma_{-1} = 1$ and $\beta_k = \gamma_k = 0,\ k \geq 0$. The Crank-Nicolson method corresponds to the choice of parameters $\beta_{-1} = \beta_0 = \gamma_{-1} = \gamma_0 = 1/2$ and $\beta_k = \gamma_k = 0,\ k \geq 1$. The Adams-Moulton method corresponds to $\beta_k = \gamma_k,\ k \geq -1$. For more details on the properties of these methods see [12, 13, 14].

These methods have good stability properties, but the main difficulty and computational bottleneck is the numerical solution of the system (1.3), which must be solved at each stepsize. Since n is large, it is common to use iterative solvers based on Krylov subspaces [22]. However, unless a very good preconditioner is available, the effectiveness of these methods strongly depends on the initial guesses. The closer the initial guess is to the exact solution, the faster the convergence. Classically, to calculate an initial guess, we use a predictor that is an explicit scheme applied to the ODE. In this paper, we propose to go beyond this approach and employ a projection method of Petrov-Galerkin type to extract a better initial guess $\hat{z}_i$ to (1.3) from the preceding solutions $z_{i-1}, z_{i-2}, \ldots$. In other words, we find $\hat{z}_i$ in the subspace

$$\mathcal{V}_i = \text{Span}\{z_{i-1}, z_{i-2}, \ldots, z_{i-r}\},\ \ r \ll n. \tag{1.5}$$

We will also consider another subspace which stems naturally from the fact that a general class of explicit schemes applied to (1.1), such as Adams-Bashforth, is of the form

$$y_{i+1} = y_i + h \sum_{k=0}^{r-1} \beta_{i,k,r}(Ay_{i-k} + f(t_{i-k})),$$

where the coefficients $\beta_{i,k,r} = \frac{1}{h}\int_{t_i}^{t_{i+1}} \Pi_{j=0,j\neq k}^{r-1} \frac{t-t_{i-j}}{t_{i-k}-t_{i-j}} dt,\ k = 0, \ldots, r-1$, are chosen so that y_i is an approximation of $y(t_i)$. This suggests that $\hat{z}_i$ may be found in

$$\mathcal{V}_i = \text{Span}\{Ay_i + f(t_i), Ay_{i-1} + f(t_{i-1}), \ldots, Ay_{i-r+1} + f(t_{i-r+1})\},\ \ r \ll n. \tag{1.6}$$

We will show that these subspaces contain a good initial guess. In practice however, we can only hope for an approximation of z_i and y_i and hence only for an approximation of $\mathcal{V}_i$. We show that a good initial guess can still be found in the approximation subspaces. A preconditioned Krylov subspace method, started with this initial guess, is then used to solve (1.3). In general, only a few iterations are needed to obtain an accurate solution.

There are of course several alternatives to solving (1.3): a natural one is to factorize C once by a sparse direct method and use the same factorization for each iteration. However, for large n, such a factorization may not be feasible. Moreover, this approach does not exploit the fact that good initial guesses are available (see the numerical tests in Section 4) and clearly should not be used in case of a non-uniform mesh. The method proposed in [5] and the block approaches proposed, for example, in [20, 22, 24] necessitate the simultaneous availability of all right-hand sides, which is not the case in the present work. The idea of using a subspace spanned by the previously computed solutions to generate a good initial guess for the next linear system is not new but has been used, for example, in [10]. In [17], the subspace of previously computed solutions augmented with Ritz vectors or an approximate invariant subspace corresponding to the smallest eigenvalues for the solution of the first right-hand side is used to compute initial guesses for the remaining right-hand sides. In [9], the Krylov subspace generated from the first linear system with the conjugate gradient method (CG) is recycled to accelerate the convergence of the subsequent systems. In [23], the approximate eigenvectors corresponding to eigenvalues close to zero computed by CG for solving the first linear systems are used in a deflation procedure to solve the subsequent systems. Analogous strategies are developed, for example, in [4, 8, 6, 19, 21] for multiple and single right-hand sides with the main idea that retaining selected approximate invariant subspaces or Ritz/harmonic vectors during restarts helps to eliminate those eigenvalues that slow down the convergence of the corresponding linear system.

The acceleration of implicit schemes for solving linear ODEs is treated, for example, in [3, 16, 18]. In [3, 16], the boundary value method is used to approximate the solution of (1.1). This method leads to a very large system of a “Toeplitz plus a small perturbation” structure, solved by GMRES with a circulant-block preconditioner. Here the effort is put on the preconditioner rather than on the initial guess. In [18], an algorithm based on the implicit Runge-Kutta scheme is used to compute an accurate approximation of y_N by reducing the number of linear systems in this scheme. The algorithm is highly efficient for computing y_N but is not so for all the y_i.

The approach taken in the present paper is close to the one in [10] in the sense that the initial guess in [10] is formed by a linear combination of previous approximate solutions. However, the approximation subspace is not the same (see the comparisons in Section 4). We use the Petrov-Galerkin method to extract the best initial guess $\hat{z}_i$ from (a modification of) $\mathcal{V}_i$ and we prove that $\|b_i - C\hat{z}_i\| = \mathcal{O}(h^r)$ where r is the number of vectors in the subspace $\mathcal{V}_i$ and $\|\ \|$ denotes the 2-norm. An extension of this work to nonlinear ODEs can be found in [2].

This paper is organized as follows. In Section 2, we briefly review the projection method of Petrov-Galerkin type and describe the proposed approach for computing a good initial guess to the linear system (1.3). The estimates thus obtained show that the accuracy of the initial guess $\hat{z}_i$ depends on the stepsize h and on the number of vectors r in the subspace $\mathcal{V}_i$, but not on the order of the implicit scheme used. The application of an implicit Runge-Kutta scheme deserves a special treatment and is considered in Section 3. The algorithmic aspect of the proposed approach and comparisons with standard predictor schemes are discussed and illustrated numerically in Section 4. Comparisons with the approach in [10] and when the systems in (1.3) are solved exactly are also presented in Section 4. A conclusion is given in Section 5.

2. Acceleration of implicit schemes.

The subspaces (1.5) and (1.6) require the exact calculation of z_{i-k} and y_{i-k}, which we want to avoid since the size n of C is supposed to be large. In what follows, we will use an approximation $\tilde{y}_{i-k}$ of y_{i-k}, which leads to the following analogue of (1.2)

$$\begin{aligned} \tilde{y}_{i+1} &= \tilde{y}_i + h\tilde{z}_i, \qquad \text{for } i = q, q+1, \ldots, N-1, \\ \tilde{y}_0 &= y^{(0)}, \end{aligned} \tag{2.1}$$

where $\tilde{z}_i$ is an approximation to z_i such that

$$\left\| \tilde{b}_i - C\tilde{z}_i \right\| \leq \varepsilon \tag{2.2}$$

with some tolerance threshold ε and where $\tilde{b}_i$ is obtained by replacing y_j in (1.4) by the corresponding values $\tilde{y}_j$. We assume that the sequence $(\tilde{y}_i)$ is bounded.

The subspaces (1.5) and (1.6) will therefore be replaced by

$$\mathcal{V}_i = \mathrm{Span}\{\tilde{z}_{i-1}, \tilde{z}_{i-2}, \ldots, \tilde{z}_{i-r}\}, \quad r \ll n \tag{2.3}$$

and

$$\mathcal{V}_i = \mathrm{Span}\{A\tilde{y}_i + f(t_i), A\tilde{y}_{i-1} + f(t_{i-1}), \ldots, A\tilde{y}_{i-r+1} + f(t_{i-r+1})\}, \quad r \ll n. \tag{2.4}$$

We will use these subspaces along with the Petrov-Galerkin process to extract a good initial guess $\hat{z}_i$, which can be viewed as an improved predictor for the scheme (2.1). This initial solution will, in turn, be used as an initial guess for a Krylov solver for computing $\tilde{z}_i$. As we will see, the use of $\hat{z}_i$ results in considerable savings both in the total number of iterations and CPU time, especially when the Krylov solver is combined with preconditioning.

The Petrov-Galerkin process applied to (1.3) allows us to find $\hat{z}_i$ such that

$$\hat{z}_i \in \mathcal{V}_i, \quad \text{and} \quad \tilde{b}_i - C\hat{z}_i \perp C\mathcal{V}_i.$$

An important reason for choosing this process is that $\hat{z}_i$ satisfies the minimization property

$$\left\|\tilde{b}_i - C\hat{z}_i\right\| = \min_{z \in \mathcal{V}_i} \left\|\tilde{b}_i - Cz\right\|. \tag{2.5}$$

In other words, the Petrov-Galerkin approximation $\hat{z}_i$ is the best least squares solution in the subspace $\mathcal{V}_i$. Here and throughout this paper $\| \ \|$ denotes the 2-norm for vectors, matrices and functions. To compute $\hat{z}_i$, let V_i be an $n \times m$ matrix, with $m \leq r$, whose columns form a basis of $\mathcal{V}_i$. Then $\hat{z}_i$ is given by $V_i x_i$ where x_i is the solution of the low order linear system

$$\left((CV_i)^T CV_i\right) x_i = (CV_i)^T \, \tilde{b}_i.$$

An algorithm for computing V_i, $\hat{z}_i$ and hence $\tilde{z}_i$ and $\tilde{y}_i$ is given in Section 4.

2.1. Use of the subspace $\mathcal{V}_i$ in (2.3). We begin with the subspace $\mathcal{V}_i$ defined in (2.3). In the next theorems, we show that this subspace and some modifications of it contain a good initial guess $\hat{z}_i$. But first we need the following lemmas. The first one simply results from Lagrange's interpolation formula; see, e.g., [7].

LEMMA 2.1. *We have*

$$\left\| f(t_i) - \sum_{k=1}^{r} \alpha_{k,r} f(t_{i-k}) \right\| \leq \max_{t \in [t_0,T]} \left\| f^{(r)}(t) \right\| h^r \tag{2.6}$$

with $\alpha_{k,r} = (-1)^{k-1} \frac{r!}{k!(r-k)!}$.

LEMMA 2.2. *The sequence* $(\tilde{y}_i)$, $i \geq q$, *satisfies*

$$\left\| \tilde{y}_i - \sum_{k=1}^{r} \alpha_{k,r} \tilde{y}_{i-k} \right\| = \mathcal{O}(h^r) + \mathcal{O}(h\varepsilon). \tag{2.7}$$

Proof. We prove (2.7) by induction on r. Note that (2.2) can be written as

$$\left\| \tilde{y}_i - \tilde{y}_{i-1} - h(\tilde{X}_{i-1} + \beta_{-1} A\tilde{y}_i) \right\| \leq h\varepsilon,$$

where $\tilde{X}_i = \sum_{k=0}^{q} \beta_k A\tilde{y}_{i-k} + \sum_{k=-1}^{q} \gamma_k f(t_{i-k})$. Hence, (2.7) is satisfied with $r = 1$. Assume that it holds for $r-1$. Using $1 + \alpha_{1,r-1} = r = \alpha_{1,r}$, $\alpha_{k,r-1} - \alpha_{k-1,r-1} = \alpha_{k,r}$, and $\alpha_{r-1,r-1} = (-1)^r = -\alpha_{r,r}$, we have

$$\begin{aligned} \tilde{y}_i - \sum_{k=1}^{r} \alpha_{k,r} \tilde{y}_{i-k} &= \tilde{y}_i - \tilde{y}_{i-1} - \sum_{k=1}^{r-1} \alpha_{k,r-1} (\tilde{y}_{i-k} - \tilde{y}_{i-k-1}) \\ &= h(\tilde{X}_{i-1} - \sum_{k=1}^{r-1} \alpha_{k,r-1} \tilde{X}_{i-1-k}) + h\beta_{-1} A(\tilde{y}_i - \sum_{k=1}^{r-1} \alpha_{k,r-1} \tilde{y}_{i-k}) + \mathcal{O}(h\varepsilon). \end{aligned} \tag{2.8}$$

Finally, the induction hypothesis and (2.6) yield

$$\begin{aligned}\tilde{X}_{i-1} - \sum_{k=1}^{r-1} \alpha_{k,r-1}\tilde{X}_{i-1-k} &= \sum_{l=0}^{q} \beta_l A \left(\tilde{y}_{i-1-l} - \sum_{k=1}^{r-1} \alpha_{k,r-1}\tilde{y}_{i-1-l-k} \right) \\ &\quad + \sum_{l=-1}^{q} \gamma_l \left(f(t_{i-1-l}) - \sum_{k=1}^{r-1} \alpha_{k,r-1} f(t_{i-1-l-k}) \right) \\ &= \mathcal{O}(h^{r-1}) + \mathcal{O}(h\varepsilon). \qquad \square\end{aligned}$$

THEOREM 2.3. *Let* $\mathcal{V}_i = \operatorname{Span}\{\tilde{z}_{i-k}, 1 \leq k \leq r\}$ *be the subspace obtained by the scheme (2.1). Then there exists a* z *in* $\mathcal{V}_i$ *such that for* $i = q, \ldots, N-1$

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

Proof. The vector $z = \sum_{k=1}^{r} \alpha_{k,r}\tilde{z}_{i-k}$ is in $\mathcal{V}_i$ and satisfies

$$\tilde{z}_i - z = \frac{1}{h}\left(\tilde{y}_{i+1} - \tilde{y}_i - \sum_{k=1}^{r} \alpha_{k,r}(\tilde{y}_{i+1-k} - \tilde{y}_{i-k}) \right).$$

Then, by (2.8), we have

$$\tilde{z}_i - z = (\tilde{X}_i - \sum_{k=1}^{r} \alpha_{k,r}\tilde{X}_{i-k}) + \beta_{-1}A(\tilde{y}_{i+1} - \sum_{k=1}^{r} \alpha_{k,r}\tilde{y}_{i+1-k}) + \mathcal{O}(\varepsilon),$$

which, as in the proof of Lemma 2.2, gives

$$\|\tilde{z}_i - z\| = \mathcal{O}(h^r) + \mathcal{O}(\varepsilon). \qquad \square$$

REMARK 2.4. From (2.5) and Theorem 2.3 it follows that the Petrov-Galerkin approximation $\hat{z}_i$ satisfies

$$\left\|\tilde{b}_i - C\hat{z}_i\right\| \leq \left\|\tilde{b}_i - Cz\right\| \leq \left\|\tilde{b}_i - C\tilde{z}_i\right\| + \|C(\tilde{z}_i - z)\|$$

and hence

$$\left\|\tilde{b}_i - C\hat{z}_i\right\| = \mathcal{O}(h^r) + \mathcal{O}(\varepsilon).$$

All the estimates in this paper are actually stated in a way similar to Theorem 2.3; only the subspace $\mathcal{V}_i$ and its dimension r in $\mathcal{O}(h^r)$ will change.

In practice, it may happen that the Petrov-Galerkin approximation $\hat{z}_i$ directly satisfies (2.2). In such a case, we take $\tilde{z}_i = \hat{z}_i$ and use the same subspace $\mathcal{V}_i$ for the next iteration. This reduces the computational cost in the proposed approach and leads us to redefine

$$\mathcal{V}_i = \operatorname{Span}\{\tilde{z}_{i-(k+m)},\ 1 \leq k \leq r\}, \tag{2.9}$$

where m is the number of the last consecutive vectors whose computation do not necessitate the use of an iterative method to satisfy (2.2). Such a subspace contains a good initial guess $\hat{z}_i$, as it its shown by the following theorem.

THEOREM 2.5. *Let* $\mathcal{V}_i$ *be the subspace defined in (2.9). Then there exists a* z *in* $\mathcal{V}_i$ *such that for* $i = q, \ldots, N-1$

$$\|\tilde{z}_i - z\| = \mathcal{O}(h^{r+m}) + \mathcal{O}(\varepsilon).$$

Proof. Since $\tilde{z}_{i-l} = \hat{z}_{i-l} \in \mathcal{V}_i$ for $1 \leq l \leq m$, the space $\mathcal{V}_i$ is the subspace spanned by $\tilde{z}_{i-k}$, $1 \leq k \leq r+m$, and the result follows from Theorem 2.3. $\square$

2.2. Use of the subspace $\mathcal{V}_i$ in (2.4). Throughout this subsection we assume $\beta_{-1} = \gamma_{-1}$ and write the scheme (1.2) in the equivalent form

$$\begin{aligned} y_{i+1} &= a_i + hz_i, \qquad \text{for } i = q, q+1, \dots, N-1, \\ y_0 &= y^{(0)}, \end{aligned} \tag{2.10}$$

where

$$a_i = y_i + h\sum_{k=0}^{q}(\beta_k A y_{i-k} + \gamma_k f(t_{i-k})) \tag{2.11}$$

and

$$z_i = \beta_{-1}(Ay_{i+1} + f(t_{i+1}))$$

is the solution of the linear system

$$Cz_i = b_i$$

with $C = I_n - \beta_{-1}hA$ and

$$b_i = \beta_{-1}(Aa_i + f(t_{i+1})). \tag{2.12}$$

Such a scheme is sufficiently general to include, for example, the implicit Euler, the Crank-Nicolson and the Adams-Moulton methods. However, the implicit Runge-Kutta method is not of this form, and for this reason we treat it separately in Section 3.

The approximation scheme corresponding to (2.10) is given by

$$\begin{aligned} \tilde{y}_{i+1} &= \tilde{a}_i + h\tilde{z}_i, \qquad \text{for } i = q, q+1, \dots, N-1, \\ \tilde{y}_0 &= y^{(0)}, \end{aligned} \tag{2.13}$$

where $\tilde{z}_i$ satisfies

$$\left\|\tilde{b}_i - C\tilde{z}_i\right\| \le \varepsilon \tag{2.14}$$

and $\tilde{a}_i$ and $\tilde{b}_i$ are obtained from (2.11) and (2.12) by replacing y_j with $\tilde{y}_j$. Then we have the following result.

THEOREM 2.6. *Let $\mathcal{V}_i = \operatorname{Span}\{A\tilde{y}_{i+1-k} + f(t_{i+1-k}), 1 \le k \le r\}$ be the subspace obtained by the scheme (2.13). Then there exists a z in $\mathcal{V}_i$ such that for $i = q, \dots, N-1$*

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

Proof. The vector $z = \beta_{-1}\sum_{k=1}^{r}\alpha_{k,r}(A\tilde{y}_{i+1-k} + f(t_{i+1-k}))$ is in $\mathcal{V}_i$ and satisfies

$$\|\tilde{z}_i - z\| = \|(\tilde{z}_i - \beta_{-1}(A\tilde{y}_{i+1} + f(t_{i+1}))) + (\beta_{-1}(A\tilde{y}_{i+1} + f(t_{i+1})) - z)\|.$$

From (2.14) we have

$$\|\tilde{z}_i - \beta_{-1}(A\tilde{y}_{i+1} + f(t_{i+1}))\| \le \varepsilon \tag{2.15}$$

and from Lemma 2.1 we obtain as in the proof of Theorem 2.3

$$\|\beta_{-1}(A\tilde{y}_{i+1} + f(t_{i+1})) - z\| = \mathcal{O}(h^r) + \mathcal{O}(h\varepsilon). \qquad \square$$

A situation analogous to the one mentioned in Theorem 2.5 may occur with the scheme (2.13), namely that the Petrov-Galerkin approximation $\hat{z}_i$ satisfies (2.14). Then we take $\tilde{z}_i = \hat{z}_i$ and use the same subspace $\mathcal{V}_i$ for the next iteration. Such a favorable situation leads to a decrease in the computational cost, provided we redefine $\mathcal{V}_i$ as the subspace spanned only by the last vectors that necessitate the use of an iterative method to satisfy (2.14):

$$\mathcal{V}_i = \mathrm{Span}\{A\tilde{y}_{i+1-(k+m)} + f(t_{i+1-(k+m)}),\ 1 \le k \le r\} \tag{2.16}$$

and $A\tilde{y}_{i-l} + f(t_{i-l}),\ 0 \le l \le m-1$ are the last vectors whose computations do not necessitate the use of an iterative method because $\tilde{z}_{i-l-1} = \hat{z}_{i-l-1} \in \mathcal{V}_i$, $0 \le l \le m-1$. Then we have the following theorem.

THEOREM 2.7. *Let $\mathcal{V}_i$ be the subspace defined in (2.16). Then there exists a z in $\mathcal{V}_i$ such that for $i = q, \ldots, N-1$*

$$\|\tilde{z}_i - z\| = \mathcal{O}(h^{r+m}) + \mathcal{O}(\varepsilon).$$

Proof. Let $\mathcal{R}_i$ be the subspace spanned by $A\tilde{y}_{i+1-k} + f(t_{i+1-k}),\ 1 \le k \le r+m$. From Theorem 2.6, we can find $w \in \mathcal{R}_i$ such that

$$\|\tilde{z}_i - w\| = \mathcal{O}(h^{r+m}) + \mathcal{O}(\varepsilon). \tag{2.17}$$

Let us decompose w as $w_1 + w_2$, where $w_1 \in \mathcal{V}_i$ and $w_2 = \sum_{l=0}^{m-1} \alpha_l (A\tilde{y}_{i-l} + f(t_{i-l}))$ with some scalars α_l. Then the vector $z = w_1 + \frac{1}{\beta_{-1}} \sum_{l=0}^{m-1} \alpha_l \tilde{z}_{i-1-l}$ is in $\mathcal{V}_i$ and satisfies

$$\|\tilde{z}_i - z\| = \left\| (\tilde{z}_i - w) - \left(\frac{1}{\beta_{-1}} \sum_{l=0}^{m-1} \alpha_l \tilde{z}_{i-1-l} - w_2 \right) \right\|.$$

Using (2.17) and (2.15) we obtain

$$\|\tilde{z}_i - z\| = \mathcal{O}(h^{r+m}) + \mathcal{O}(\varepsilon). \qquad \square$$

3. Use of implicit Runge-Kutta scheme. Recall that an s-stage implicit Runge-Kutta (IRK) scheme applied to the system (1.1) is given by (see, e.g., [13])

$$\begin{aligned} y_{i+1} &= y_i + h(d \otimes A)z_i + h(d \otimes I_n)F_i, \qquad \text{for } i = 0, 1, \ldots, N-1, \\ y_0 &= y^{(0)}, \end{aligned}$$

where z_i is the solution of the $sn \times sn$ system

$$Cz_i = b_i \tag{3.1}$$

with

$$\begin{array}{lll} C = I_{sn} - h(A_0 \otimes A), & b_i = (\mathbf{1}_s \otimes y_i) + h\,(A_0 \otimes I_n)F_i, & A_0 = (a_{ij})_{1 \le i,j \le s}, \\ d = (d_1, \ldots, d_s), & F_i = \left(f(t_i + c_1 h)^T, \ldots, f(t_i + c_s h)^T\right)^T. & \end{array}$$

The Runge-Kutta coefficients are given by the vectors d^T, $c = (c_1, \ldots, c_s)^T$ and the matrix A_0. The symbol $\otimes$ denotes the Kronecker product and $\mathbf{1}_s = (1, \ldots, 1)^T \in \mathbb{R}^s$.

Since (3.1) is of large size, we compute an approximation $\tilde{z}_i$ of z_i such that

$$\left\| \tilde{b}_i - C\tilde{z}_i \right\| \le \varepsilon, \tag{3.2}$$

where $\tilde{b}_i = (\mathbf{1}_s \otimes \tilde{y}_i) + h\,(A_0 \otimes I_n)F_i$ and the sequence $(\tilde{y}_i)$ is given by

$$\begin{aligned} \tilde{y}_{i+1} &= \tilde{y}_i + h(d \otimes A)\tilde{z}_i + h(d \otimes I_n)F_i, \qquad \text{for } i = 0, 1, \dots, N-1, \\ \tilde{y}_0 &= y^{(0)} \end{aligned} \tag{3.3}$$

and is assumed to be bounded.

Our aim is to show that the subspace spanned by $\tilde{z}_{i-k},\ 1 \le k \le r$, contains a vector z such that

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

We begin with the following lemma, which is the analogue of Lemma 2.2.

LEMMA 3.1. *The sequence $(\tilde{y}_i)$ defined in (3.3) satisfies*

$$\left\| \tilde{y}_i - \sum_{k=1}^{r} \alpha_{k,r} \tilde{y}_{i-k} \right\| = \mathcal{O}(h^r) + \mathcal{O}(h\varepsilon). \tag{3.4}$$

Proof. We prove (3.4) by induction on r. From (3.3) we see that (3.4) holds with $r = 1$. Assume it holds with $r - 1$. Then, from (3.2), we have

$$\tilde{z}_i = \mathbf{1}_s \otimes \tilde{y}_i + h(A_0 \otimes I_n)F_i + h(A_0 \otimes A)\tilde{z}_i + \mathcal{O}(\varepsilon),$$

and by inserting this expression into (3.3) we obtain

$$\tilde{y}_{i+1} - \tilde{y}_i = h(d \otimes I_n)F_i + h(d\mathbf{1}_s \otimes A\tilde{y}_i) + h^2(dA_0 \otimes A)F_i + h^2(dA_0 \otimes A^2)\tilde{z}_i + \mathcal{O}(h\varepsilon).$$

Applying the same process to $\tilde{z}_i$, we easily obtain

$$\begin{aligned} \tilde{y}_{i+1} - \tilde{y}_i = &\sum_{k=1}^{r-1} h^k \left((dA_0^{k-1} \otimes A^{k-1})F_i + (dA_0^{k-1}\mathbf{1}_s \otimes A^k \tilde{y}_i)\right) \\ &+ h^r(dA_0^{r-1} \otimes A^{r-1})F_i + h^r(dA_0^{r-1} \otimes A^r)\tilde{z}_i + \mathcal{O}(h\varepsilon) \end{aligned}$$

and

$$\begin{aligned} &\tilde{y}_{i+1} - \tilde{y}_i - \sum_{l=1}^{r-1} \alpha_{l,r-1}(\tilde{y}_{i+1-l} - \tilde{y}_{i-l}) \\ &\quad = \sum_{k=1}^{r-1} h^k \left((dA_0^{k-1} \otimes A^{k-1})(F_i - \sum_{l=1}^{r-1} \alpha_{l,r-1} F_{i-l}) \right) \\ &\qquad + \sum_{k=1}^{r-1} h^k \left(dA_0^{k-1}\mathbf{1}_s \otimes A^k(\tilde{y}_i - \sum_{l=1}^{r-1} \alpha_{l,r-1}\tilde{y}_{i-l}) \right) \\ &\qquad + \mathcal{O}(h^r) + \mathcal{O}(h\varepsilon). \end{aligned}$$

From this, the induction hypothesis, and Lemma 2.1 we obtain

$$\left\| \tilde{y}_{i+1} - \tilde{y}_i - \sum_{l=1}^{r-1} \alpha_{l,r-1}(\tilde{y}_{i+1-l} - \tilde{y}_{i-l}) \right\| = \mathcal{O}(h^r) + \mathcal{O}(h\varepsilon),$$

which is exactly (3.4). □

THEOREM 3.2. *Let $\mathcal{V}_i = \operatorname{Span}\{\tilde{z}_{i-k}, 1 \le k \le r\}$ be the subspace obtained by the vectors satisfying (3.2)–(3.3). Then there exists a z in $\mathcal{V}_i$ such that for $i = 0, \dots, N-1$*

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

Proof. From (3.2) we have

$$\begin{aligned}\tilde{z}_i &= \tilde{b}_i + h(A_0 \otimes A)\tilde{z}_i + \mathcal{O}(\varepsilon)\\ &= \tilde{b}_i + h(A_0 \otimes A)\tilde{b}_i + h^2(A_0^2 \otimes A^2)\tilde{z}_i + \mathcal{O}(\varepsilon)\end{aligned}$$

and more generally

$$\tilde{z}_i = \sum_{k=0}^{r-1} h^k (A_0^k \otimes A^k)\tilde{b}_i + \mathcal{O}(h^r) + \mathcal{O}(\varepsilon).$$

The vector $z = \sum_{k=1}^{r} \alpha_{k,r}\tilde{z}_{i-k}$ belongs to $\mathcal{V}_i$ and satisfies the requirement of the theorem since

$$\tilde{z}_i - z = \sum_{k=0}^{r-1} h^k (A_0^k \otimes A^k)\left(\tilde{b}_i - \sum_{l=1}^{r} \alpha_{l,r}\tilde{b}_{i-l}\right) + \mathcal{O}(h^r) + \mathcal{O}(\varepsilon),$$

and from Lemmas 2.1 and 3.1 we have

$$\begin{aligned}\tilde{b}_i - \sum_{l=1}^{r}\alpha_{l,r}\tilde{b}_{i-l} &= \mathbf{1}_s \otimes \left(\tilde{y}_i - \sum_{l=1}^{r}\alpha_{l,r}\tilde{y}_{i-l}\right) + h(A_0 \otimes I_n)\left(F_i - \sum_{l=1}^{r}\alpha_{l,r}F_{i-l}\right)\\ &= \mathcal{O}(h^r) + \mathcal{O}(h\varepsilon). \qquad \square\end{aligned}$$

As in Theorems 2.5 and 2.7, we can reduce the amount of computations in the Petrov-Galerkin process with the subspace defined in Theorem 3.2 in the case when some Petrov-Galerkin approximations satisfy (3.2). Let

$$\mathcal{V}_i = \text{Span}\{\tilde{z}_{i-(k+m)}, 1 \le k \le r\}, \tag{3.5}$$

where $\tilde{z}_{i-l}$, $1 \le l \le m$, are the last vectors whose computations do not require the use of an iterative method because $\tilde{z}_{i-l} = \hat{z}_{i-l} \in \mathcal{V}_i$, $1 \le l \le m$. Then we have the following theorem whose proof is similar to the one of Theorem 2.5.

THEOREM 3.3. *Let $\mathcal{V}_i$ be the subspace defined in (3.5). Then there exists a z in $\mathcal{V}_i$ such that for $i = 0, \ldots, N-1$*

$$\|\tilde{z}_i - z\| = \mathcal{O}(h^{r+m}) + \mathcal{O}(\varepsilon).$$

In the numerical experiments, we will consider the 3-stage IRK scheme of order 6 (IRK6), defined by

$$A_0 = \begin{bmatrix} \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30}\\ \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24}\\ \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36}\end{bmatrix}, \; d^T = \begin{bmatrix}\frac{5}{18}\\ \frac{4}{9}\\ \frac{5}{18}\end{bmatrix}, \text{ and } c = \begin{bmatrix}\frac{1}{2} - \frac{\sqrt{15}}{10}\\ \frac{1}{2}\\ \frac{1}{2} + \frac{\sqrt{15}}{10}\end{bmatrix}.$$

4. Computational considerations. We begin with an algorithm written in a formal way that summarizes the computational aspect of the sequence $(\tilde{y}_i)$ defined in (2.1), (2.13) or (3.3).

Algorithm AIS. [Accelerated Implicit Scheme]

```
1. Assume that for $k = 1, \dots, q$, $\tilde{y}_k$ is either given or computed with an $k-$step scheme.
   Set $i = q$.
2. Let $R_i$ be the matrix formed by the last $k_0$ vectors $\tilde{z}_{i-k}$, $1 \le k \le k_0$, where $k_0 = \min(q, r)$. Orthonormalize the columns of $R_i$ into $V_i$. Compute $L_i = CV_i$, $C_i = L_i^T L_i$ and $\tilde{b}_i$.
3. Repeat until $i = N - 1$
   (a) Compute the initial guess $\hat{z}_i = V_i {C_i}^{-1} L_i^T \tilde{b}_i$.
   (b) If $\|\tilde{b}_i - C\hat{z}_i\| \le \varepsilon \|\tilde{b}_i\|$, then set $\tilde{y}_{i+1} = \tilde{y}_i + h\hat{z}_i$, $R_{i+1} = R_i$, $V_{i+1} = V_i$, $C_{i+1} = C_i$, $L_{i+1} = L_i$, compute $\tilde{b}_{i+1}$, set $i = i + 1$ and go to $(a)$.
   (c) Otherwise, compute an approximation $\tilde{z}_i$ to $C^{-1}\tilde{b}_i$ with $\|\tilde{b}_i - C\tilde{z}_i\| \le \varepsilon\|\tilde{b}_i\|$ by an iterative method starting with $\hat{z}_i$.
   (d) Set $\tilde{y}_{i+1} = \tilde{y}_i + h\tilde{z}_i$ and compute $\tilde{b}_{i+1}$.
   (e) Let $R_{i+1}$ be the matrix formed by the last $r$ vectors $\tilde{z}_{i-k}$, $0 \le k \le r - 1$. Orthonormalize the columns of $R_{i+1}$ into $V_{i+1}$.
   (f) Compute $L_{i+1} = CV_{i+1}$ and $C_{i+1} = L_{i+1}^T L_{i+1}$.
   (g) $i = i + 1$.
```

We have used the scheme (2.1) in steps 3(b) and 3(d). It is very easy to modify the algorithm if the scheme (2.13) is used instead. Also, in this algorithm, the subspace $\mathcal{V}_i$ corresponds to the one of Theorem 2.5. If the subspace $\mathcal{V}_i$ of Theorem 2.3 is used, then step 3(b) should be replaced by

$$3(\mathrm{b}_1) \qquad \text{If } \left\|\tilde{b}_i - C\hat{z}_i\right\| \le \varepsilon \left\|\tilde{b}_i\right\|, \text{ set } \tilde{z}_i = \hat{z}_i, \text{ and go to } 3(\mathrm{d}).$$

If the subspace $\mathcal{V}_i$ of Theorem 2.7 (or Theorem 2.6) is used, then in step 3(e) the vectors $\tilde{z}_{i-k}$ should be replaced by $A\tilde{y}_{i+1-k} + f(t_{i+1-k})$ (and step 3(b) should be replaced by 3(b_1)). Finally, if the subspace $\mathcal{V}_i$ of Theorem 3.3 (or Theorem 3.2) is used, then step 3(d) should be replaced by

$$3(\mathrm{d}_1) \qquad \tilde{y}_{i+1} = \tilde{y}_i + h(d \otimes A)\tilde{z}_i + h(d \otimes I_n)F_i, \text{ and compute } \tilde{b}_{i+1}$$

(and in step 3(b), $\tilde{y}_{i+1}$ is given by $\tilde{y}_{i+1} = \tilde{y}_i + h(d \otimes A)\hat{z}_i + h(d \otimes I_n)F_i$).

In steps 2 and 3(e), the orthonormalization of R_i and R_{i+1} uses an updated QR factorization (see, e.g., [11, p. 594]), and in step 3(f), the matrices L_{i+1} and C_{i+1} are updated from L_i, C_i and the QR factorization of R_{i+1}.

The cost of each iteration i is essentially dominated by the cost of steps 3(c), 3(e), and 3(f). For example, if GMRES is used in step 3(c) and if we denote by k_{gmres} the number of GMRES iterations needed in this step and by n_z the number of nonzero elements in C, then each iteration i requires $\mathcal{O}(k_{gmres}n_z + (k_{gmres}^2 + r)n)$ operations.

4.1. Numerical tests. All the numerical tests are run on an Intel processor c6100 with 2.66 Ghz. We use $r = 20$, a constant time step $h = \frac{1}{100}$, and $\varepsilon = 10^{-8}$.

We refer to AIS1 for the variant of the algorithm which uses the subspace $\mathcal{V}_i$ of Theorem 2.5 or Theorem 3.3 for IRK, and to AIS2 for the one which uses the subspace $\mathcal{V}_i$ of Theorem 2.7. We compare these variants with other approaches, where the initial guess to (1.3) is obtained with some classical predictors such as explicit Euler, Adams-Bashforth of order r, the second-order and fourth-order Runge-Kutta methods, and the approach [10] mentioned in Section 1. More precisely, for these approaches, we replace step 2 of algorithm AIS

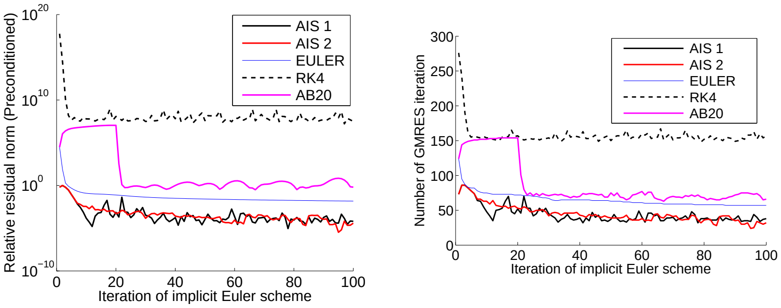

FIG. 4.1. *Test* 1. *Use of implicit Euler. Left: initial residual norms. Right: number of GMRES iterations.*

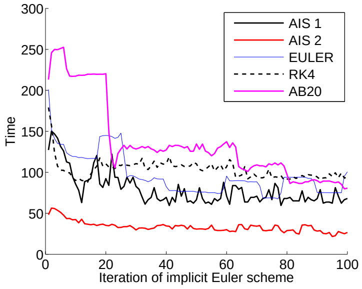

FIG. 4.2. *Test* 1. *Use of implicit Euler. CPU time in seconds.*

TABLE 4.1
Test 1. *Use of implicit Euler. Total CPU time in minutes and total number of GMRES iterations.*

	Time (min.)	GMRES iter.
AIS1	131.9	4409
AIS2	55.9	4507
Euler	161.1	6520
AB20	227.5	8700
RK4	172.1	15744

by: ”Compute $\tilde{b}_i$” and remove steps 3(e) and 3(f) and replace step 3(a) by

$$z_i^{(e)} = (y_{i+1}^{(e)} - \tilde{y}_i)/h,$$

where

$$y_{i+1}^{(e)} = \tilde{y}_i + h(A\tilde{y}_i + f(t_i))$$

for explicit Euler,

$$y_{i+1}^{(e)} = \tilde{y}_i + h\sum_{k=0}^{r-1} \beta_{i,k,r}(A\tilde{y}_{i-k} + f(t_{i-k}))$$

for Adams-Bashforth of order r,

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

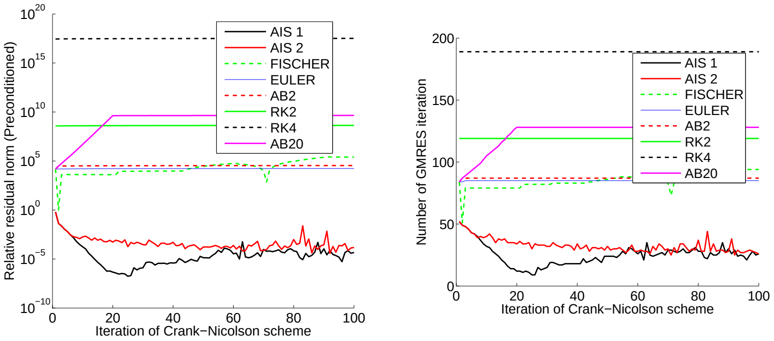

FIG. 4.3. *Test* 1. *Use of Crank-Nicolson. Left: initial residual norms. Right: number of GMRES iterations.*

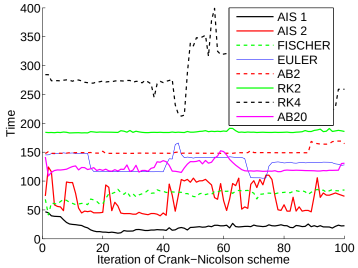

FIG. 4.4. *Test* 1. *Use of Crank-Nicolson. CPU time in seconds.*

TABLE 4.2
Test 1. *Use of Crank-Nicolson. Total CPU time in minutes and total number of GMRES iterations.*

	Time (min.)	GMRES iter.
AIS1	34.5	2476
AIS2	116.6	3254
FISCHER	127.4	8521
Euler	218.8	8498
RK2	308.7	11900
AB2	251.3	8697
RK4	483.0	18900
AB20	205.2	12354

for Runge-Kutta 2, with

$$k_1 = A\tilde{y}_i + f(t_i), \quad k_2 = A(\tilde{y}_i + hk_1) + f(t_{i+1}),$$

$$y_{i+1}^{(e)} = \tilde{y}_i + \frac{h}{6}(l_1 + 2l_2 + 2l_3 + l_4)$$

for Runge-Kutta 4, with

$$l_1 = A\tilde{y}_i + f(t_i), \qquad l_2 = A(\tilde{y}_i + hl_1/2) + f(t_i + h/2),$$
$$l_3 = A(\tilde{y}_i + hl_2/2) + f(t_i + h/2), \qquad l_4 = A(\tilde{y}_i + l_3) + f(t_{i+1}),$$

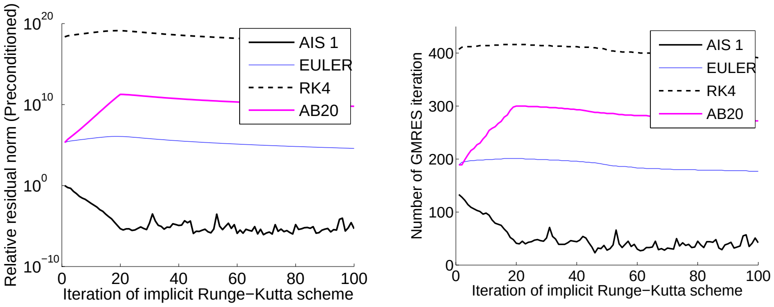

FIG. 4.5. *Test* 1. *Use of implicit Runge-Kutta. Left: initial residual norms. Right: number of GMRES iterations.*

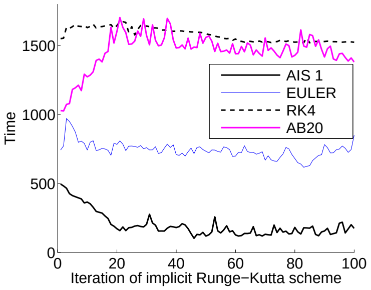

FIG. 4.6. *Test* 1. *Use of implicit Runge-Kutta. CPU time in seconds.*

TABLE 4.3
Test 1. *Use of implicit Runge-Kutta. Total CPU time in minutes and total number of GMRES iterations.*

	Time (min.)	GMRES iter.
AIS1	331.4	4995
Euler	1238.7	18899
AB20	2434.8	27722
RK4	2629.2	40492

and finally $y_{i+1}^{(e)}$ is the approximate solution computed by [10, Method 1].

These approaches will be referred to as Euler, ABr, RK2, RK4, and FISCHER, respectively.

4.2. Test 1. We consider equation (1.1) where A and f are obtained by a spatial discretization of the two-dimensional heat equation $\partial_t u - \Delta u = 0$ in $(-1,1)^2$, $0 < t \leq 1$, using the finite volume method with Dirichlet boundary conditions $u(x,y,t) = t\,(t+1)$, $0 \leq t \leq 1$. The k-th component of $y^{(0)}$ is given by $\left(y^{(0)}\right)_k = \sin\left(2\pi k/(n+1)\right)$. The matrix A is of order $n = 517396$, with $\|A\|_\infty \approx 2.66 \times 10^7$ and $\|f\|_\infty \approx 1.07 \times 10^5$.

In step 3(c) of algorithm AIS, we use the restarted preconditioned GMRES. The restarted value is 20 and the preconditioner is obtained from an incomplete LU factorization with a drop tolerance fixed at 10^{-3}.

Figures 4.1 and 4.2 show the results when the test problem is solved using the implicit Euler scheme. Figure 4.1 (left) shows the relative preconditioned residual norm corresponding to the initial guess computed by some of the approaches defined above. The horizontal axis shows the number of iterations i at which the residuals are evaluated. For each i, Figure 4.1 (right) shows the number of iterations required by GMRES for computing $\tilde{z}_i$ starting

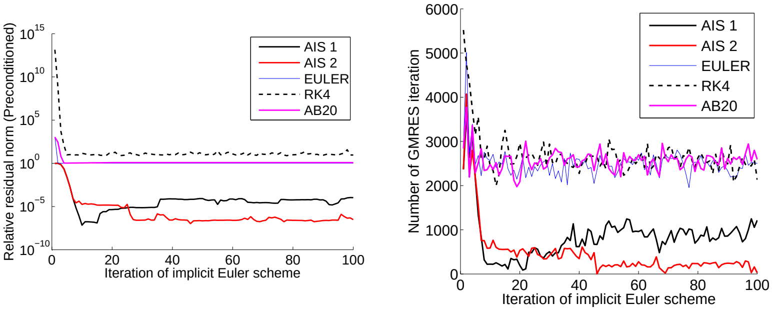

FIG. 4.7. *Test* 2. *Use of implicit Euler. Left: initial residual norms. Right: number of GMRES iterations.*

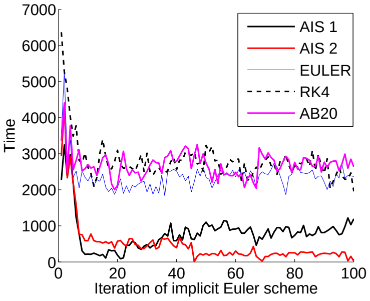

FIG. 4.8. *Test* 2. *Use of implicit Euler. CPU time in seconds.*

TABLE 4.4
Test 2. *Use of implicit Euler. Total CPU time in minutes and total number of GMRES iterations.*

	Time (min.)	GMRES iter.
AIS1	1337.5	84903
AIS2	809.4	45735
Euler	3948.2	250034
AB20	4492.1	268032
RK4	4655.5	256536

with $\hat{z}_i$ for AIS1 and AIS2 and with $z_i^{(e)}$ for Euler, AB20 and RK4. Figure 4.2 shows the running time required for each approach. Table 4.1 shows the total CPU time and total number of GMRES iterations for the five approaches. The figures and the table clearly show that AIS1 and AIS2 always provide the best results. Note that at the beginning AIS1 and AIS2 require more iterations and time since the corresponding approximation subspaces $\mathcal{V}_i$ do not contain enough vectors.

The results given by the Euler, AB20, and RK4 approaches are not satisfactory, which is due to the fact that the initial guess $z_i^{(e)}$ itself is not satisfactory. One reason is that the expressions for $z_i^{(e)}$ contain, for Euler and ABr, a small and large linear combination of $A\tilde{y}_{i-k} + f(t_{i-k})$ and for RK4 a linear combination of $f(t_{i-k})$ and powers of A times y_{i-k}. The large values of $\|A\|$ and $\|f\|$ necessarily introduce errors in $z_i^{(e)}$, leading to a significant increase in the number of GMRES iterations and CPU time. Analogous results are reported in Figures 4.3 and 4.4 and Table 4.2 when the Crank-Nicolson scheme is used instead of the

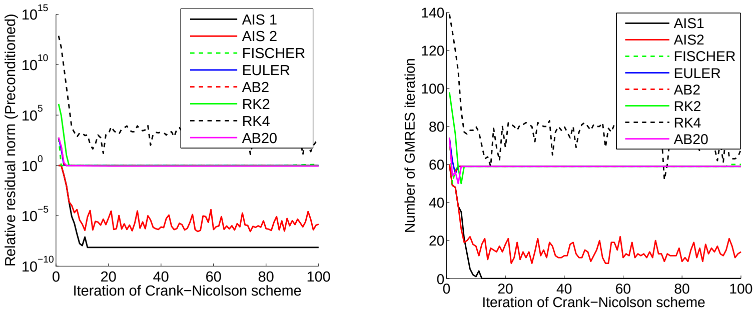

FIG. 4.9. *Test* 2. *Use of Crank-Nicolson. Left: initial residual norms. Right: number of GMRES iterations.*

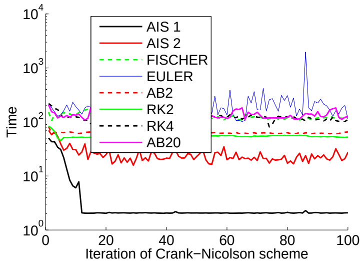

FIG. 4.10. *Test* 2. *Use of Crank-Nicolson. CPU time in seconds.*

TABLE 4.5
Test 2. *Use of Crank-Nicolson. Total CPU time in minutes and total number of GMRES iterations.*

	Time (min.)	GMRES iter.
AIS1	7.4	276
AIS2	41.7	1567
FISCHER	201.6	5909
Euler	343.2	5914
RK2	92.9	5974
AB2	106.1	5907
RK4	204.4	7733
AB20	235.1	5898

implicit Euler together with other predictors and in Figures 4.5 and 4.6 and Table 4.3 when the implicit Runge Kutta (IRK6) is used.

4.3. Test 2. We consider the advection diffusion system

$$\begin{aligned} \partial_t u - \tfrac{1}{\text{Pe}}\Delta u + a.\nabla u &= 0 && \text{in } \Omega = (-1,1)\times(0,1),\\ u &= (1-\tanh(\text{Pe}))\,t(t+1) && \text{on } \Gamma_0,\\ u &= 1+\tanh\left((2x+1)\text{Pe}\right)t(t+1) && \text{on } \Gamma_{in},\\ \frac{\partial u}{\partial n} &= 0 && \text{on } \Gamma_{out}, \end{aligned}$$

where the convective term is given by $a(x,y) = \begin{bmatrix} 2y(1-x^2) \\ -2x(1-y^2) \end{bmatrix}$ and the boundary is split into $\partial\Omega = \Gamma_{in}\cup\Gamma_{out}\cup\Gamma_0$ with $\Gamma_{in} := [-1,0]\times\{0\}$, $\Gamma_{out} := [0,1]\times\{0\}$ and Γ_0 is the

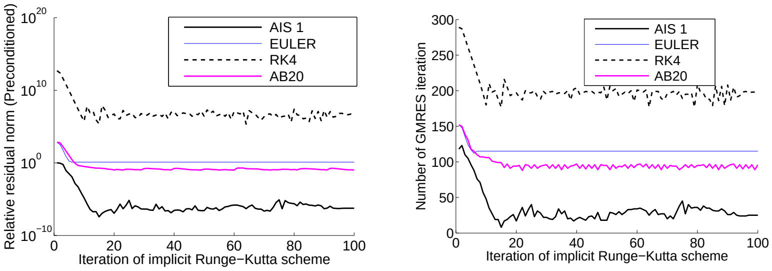

FIG. 4.11. *Test* 2*. Use of implicit Runge-Kutta. Left: initial residual norms. Right: number of GMRES iterations.*

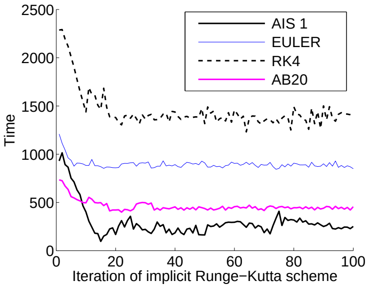

FIG. 4.12. *Test* 2*. Use of implicit Runge-Kutta. CPU time in seconds.*

TABLE 4.6
Test 2*. Use of implicit Runge-Kutta. Total CPU time in minutes and total number of GMRES iterations.*

	Time (min.)	GMRES iter.
AIS1	494.2	3326
Euler	1493.0	11599
RK4	2416.6	20045
AB20	770.4	9697

remaining part. A description of this system is given in [1, 25]

The Péclet number Pe is fixed at 10. We discretize the system in space with the finite volume method and obtain a problem of the form (1.1) of size $n = 775790$. The norms of the matrix A and function f in (1.1) are estimated as 1.59×10^6 and 1.78×10^7. The initial solution of the differential equation is the same as in Test 1.

In step 3(c) of algorithm AIS, we use the preconditioned restarted GMRES with the same parameter as in Test 1. The comparisons are shown in Figures 4.7–4.12 and Tables 4.4–4.6. Again similar comments apply as in Test 1.

Figure 4.13 and Table 4.7 show how the performance of AIS1 depends on the dimension r of the approximation subspace. The table indicates that a moderate value of r should be used, so $r = 20$ seems a plausible choice for our experiments.

Now, we discretize the advection-diffusion system using a triangulation mesh of smaller size $n = 62277$ so that an exact LU decomposition of C is feasible. The decomposition is done once and used for each iteration of the Crank-Nicolson scheme. Figure 4.14 shows

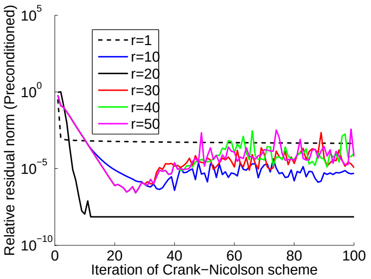

FIG. 4.13. *Test* 2. *Use of Crank-Nicolson. Performance of AIS1 for different values of* r.

TABLE 4.7
Test 2. *Use of Crank-Nicolson. Performance of AIS1 for different values of* r.

r	Time(min.)	GMRES iter.
1	109.4	1662
10	37.8	768
20	7.4	276
30	30.4	900
40	49.4	935
50	41.0	935

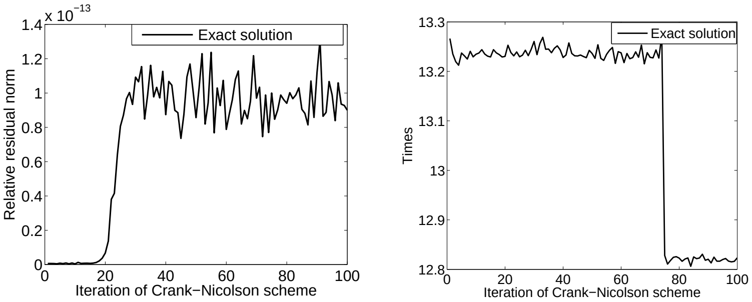

FIG. 4.14. *Test* 2. *Use of Crank-Nicolson. Left: initial residual norms. Right: CPU time in seconds.*

the relative residual norm of the exact solution z_i of the linear systems to be solved in the Crank-Nicolson scheme and the corresponding running time. The total time can be estimated from the figure on the right as approximately $100 \times 13 = 1300$ s ≈ 21.6 min. For this case, the approaches AIS1 and AIS2 require 1.01 min and 1.35 min, respectively.

5. Conclusion.

The purpose of this paper is to improve the initial guesses for the iterative solutions of large linear systems that arise in implicit schemes for large systems of linear ODEs. The improvement, summarized in algorithm AIS, is based on the Petrov-Galerkin process applied at iteration i to a subspace $\mathcal{V}_i$ of small dimension r built from previous computations. We have proposed two subspaces. The first one is spanned by the solutions $\tilde{z}_{i-k},\ 1 \leq k \leq r$, computed at previous iterations. The second one is spanned by the vectors $A\tilde{y}_{i-k} + f(t_{i-k})$, $0 \leq k \leq r-1$, also computed at previous iterations. The theory shows that these subspaces contain good approximation to $\tilde{z}_i$ at iteration i and that the quality of the approximation depends only on the stepsize h, the dimension r, and the tolerance ε at which the linear systems are solved. It is possible to use such approximation directly in the implicit scheme or continue to improve it via a preconditioned Krylov subspace method before adding it to the implicit scheme. We recommend the second option. Both subspaces lead to good numerical results with some advantage for the first one. This advantage is also observed with other numerical test examples not reported here. Future work includes devising a theory and an implementation when the time step sizes are not uniform.

Acknowledgment. We would like to thank the referees for their comments and constructive criticisms which helped us to improve the paper.

REFERENCES

[1] M. Afanasjew, M. Eiermann, O.G. Ernst, and S. Güttel, *Implementation of a restarted Krylov subspace method for the evaluation of matrix functions*, Linear Algebra Appl., 429 (2008), pp. 2293–2314.

[2] M. Al Sayed. Ali and M. Sadkane, *Acceleration of implicit schemes for large systems of nonlinear ODEs*, Electron. Trans. Numer. Anal., 35 (2009), pp. 104–117. http://etna.mcs.kent.edu/vol.35.2009/pp104-117.dir

[3] R. H. Chan, M. K. Ng, and X.-Q. Jin, *Strang-type preconditioners for systems of LMF-based ODE codes*, IMA J. Numer. Anal., 21 (2001), pp. 451–462.

[4] A. Chapman and Y. Saad, *Deflated and augmented Krylov subspace techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.

[5] T. F. Chan and M. K. Ng, *Galerkin projection methods for solving multilple linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 836–850.

[6] D. Darnell, R. B. Morgan, and W. Wilcox, *Deflated GMRES for sytems with multilple shifts and multilple right-hand sides*, Liner Algebra Appl., 429 (2008), pp. 2415–2434.

[7] P. J. Davis and P. Rabinowitz, *Numerical Integration*, Blaisdell, Waltham, 1967.

[8] E. de Sturler, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.

[9] J. Erhel and F. Guyomarc'h, *An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1279–1299.

[10] P. F. Fischer, *Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides*, Comput. Methods Appl. Mech. Engrg., 163 (1998), pp. 193–204.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, 1989.

[12] E. Hairer, S. P. Nörsett, and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, 2nd ed., Springer, Berlin, 1993.

[13] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, 2nd ed., Springer, Berlin, 1996.

[14] P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations*, Wiley, New York, 1962.

[15] W. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer, Berlin, 2003.

[16] X.-Q. Jin, V.-K. Sin, and L.-L. Song, *Circulant-block preconditioners for solving ordinary differential equations*, Appl. Math. Comput., 140 (2003), pp. 409–418.

[17] M. E. Kilmer and E. de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.

[18] M. Lopez-Fernandez, C. Lubich, C. Palencia, and A. Schädle, *Fast Runge-Kutta approximation of inhomogeneous parabolic equations*, Numer. Math., 102 (2005), pp. 277–291.

[19] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.

[20] M. Robbé and M. Sadkane, *Exact and inexact breakdowns in the block GMRES method*, Linear Algebra Appl., 419 (2006), pp. 265–285.

[21] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix. Anal. Appl., 18 (1997), pp. 435–449.

[22] ———, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[23] Y. Saad, M. Yeung, J. Erhel, F. Guyomarc'h, *A deflated version of the conjugate gradient algorithm*, SIAM J. Sci. Comput., 21 (2000), pp. 1909–1926.

[24] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right hand-sides*, SIAM J. Sci. Comput., 16 (1995), pp. 917–933.

[25] R. M. Smith and A. G. Hutton, *The numerical treatemt of advection: a performance comparison of current methods*, Numer. Heat Transfer, 5 (1982), pp. 439–461.