Acceleration of implicit schemes for large linear systems of differential–algebraic equations

Mouhamad Al Sayed Ali, Miloud Sadkane

Acceleration of implicit schemes for large linear systems of differential-algebraic equations

Mouhamad Al Sayed Ali , Miloud Sadkane *

Université de Brest. CNRS - UMR 6205, Laboratoire de Mathématiques de Bretagne Atlantique. 6 avenue Victor Le Gorgeu, CS 93837, 29285 Brest Cedex 3. France.

Abstract

Implicit schemes for solving large-scale linear differential-algebraic systems with constant coefficients necessitate at each integration step the solution of a linear system, typically obtained by a Krylov subspace method such as GMRES. To accelerate the convergence, an approach is proposed that computes good initial guesses for each linear system to be solved in the implicit scheme. This approach requires, at each integration step, a small dimensional subspace where a good initial guess is found using the Petrov-Galerkin process. It is shown that the residual associated with the computed initial guess depends on the dimension of the subspace, the order of the implicit scheme, and the discretization stepsize. Several numerical illustrations are reported.

Key words: Petrov-Galerkin, GMRES, initial guess, linear DAE

1. Introduction and motivation

In [2], the authors proposed a way to improve predictor schemes for large systems of ordinary differential equations with constant coefficients. Since the integration schemes involved are implicit, they require the solution of a large linear system at each integration step. These systems are usually solved by a Krylov subspace iterative method that requires good initial guesses and/or good preconditioners to accelerate the convergence. The improvement mentioned above consisted in constructing subspaces of small dimension in which good initial guesses are found. The accuracy of such initial guesses depends on the dimension of the constructed subspaces and the discretization stepsize

* Corresponding author.
Email addresses: moh.sayedal@gmail.com (Mouhamad Al Sayed Ali), miloud.sadkane@univ-brest.fr (Miloud Sadkane).

Preprint submitted to Elsevier *31 October 2019*

but is independent of the order of the implicit scheme. In the present work, we propose to extend the idea to large linear systems of differential algebraic equations with constant coefficients. Since these equations are of algebraic type, the subspaces defined in [2] may no longer be applicable. For this reason, new subspaces containing good initial guesses will be the focus of this work.

Consider the linear system of differential-algebraic equations (DAEs)

$$\begin{cases} B\dot{y}(t) = Ay(t) + f(t), \quad \forall \ \ t \in [t_0, T], \\ y(t_0) = y^{(0)}, \end{cases} \tag{1}$$

which results, for example, from the method of lines applied to a linear partial differential algebraic equation, see, e.g., [16], and arises in a wide variety of applications, see, e.g., [8]. Using the Weierstrass canonical form of $\lambda B - A$ (see [11]), equation (1) generally can be separated into an ordinary differential system and an algebraic differential system. A concept that plays an important role in the solution of (1) is the index which can be defined as the number of times the algebraic part has to be differentiated to obtain an ordinary differential system. A higher index causes numerical difficulties. For example, the papers [9,13,22] propose index reduction techniques for linear (time-varying or constant coefficients) DAEs with the aim of using appropriate numerical methods. The survey [3] presents the solution theory of DAEs of type (1) and discusses, among other subjects, numerical methods to check the solvability properties of DAEs as well as index reduction. A family of multistep difference schemes is proposed in [7] for the numerical solution of time-varying DAEs of index at most 2, see also [5,6] for a similar approach. Most of the proposed techniques are limited to small size problems.

However, as we will see below, our goal is not the numerical resolution of (1) and, therefore, the concept of index will not be relevant in this work.
We assume throughout this paper that A and B are real large $n \times n$ matrices and that the pencil $\lambda B - A$ is regular (i.e., there exists λ such that $\det(\lambda B - A) \neq 0$). We make no particular assumption on B.
Note that if $f \in C^{r+\nu-1}([t_0, T])$ (i.e., $(r + \nu - 1)$-times continuously differentiable in $[t_0, T]$), where r is a nonnegative integer and ν is the index of the pencil $\lambda B - A$ (i.e., the degree of nilpotency in the Weierstrass canonical form of $\lambda B - A$) and $\big(y^{(0)}, f(t_0)\big)$ satisfies the consistency condition (see, e.g., [17, chap. 2]), then (1) has a unique solution $y \in C^r([t_0, T])$. In order to avoid repetition, the function f will be assumed to be sufficiently smooth and the consistency condition is satisfied to guarantee the existence and uniqueness of the solution of (1) in the set $C^r([t_0, T])$, where r is a given integer.

Suppose we wish to solve (1) by an implicit scheme. Let $t_i = t_0 + ih$, where $h = (T - t_0)/N$ is the discretization stepsize, and let y_i be an approximation of $y(t_i)$. Most standard implicit schemes applied to (1) can be written as

$$\begin{cases} y_{i+1} = a_i + hz_i, \qquad q \leq i \leq N - 1, \\ y_0 = y^{(0)}, \end{cases} \tag{2}$$

where $y_0 = y^{(0)}, y_1, \ldots, y_q$ are given, $q \ll N$, a_i is a vector generally depending on f and $y_{i-k}, \ 0 \leq k \leq q$. Since this scheme is implicit, at each time step, one must solve a linear system of the form

$$Cz_i = b_i, \tag{3}$$

where C is a nonsingular matrix depending on A, B and h; and b_i is a vector depending on A, B, $y_{i-l},\ l = 0, \ldots, q$ and $f(t_{i-k}),\ k = -1, \ldots, q$.
Without loss of generality we suppose that b_i is of the form

$$b_i = \sum_{l=0}^{q} (\xi_l A + \psi_l B) y_{i-l} + \sum_{k=-1}^{q} \phi_k f(t_{i-k}), \tag{4}$$

where ξ_l, ψ_l and ϕ_k are real parameters.
As we will see in Section 2, the classical implicit schemes satisfy (2), (3) and (4).

Note that if B is invertible, then using for example the BDF scheme (see (9), (11)), the system (3) is mathematically equivalent to

$$(I - \beta h B^{-1} A) z_i = B^{-1} b_i.$$

Therefore this scheme is associated with the ordinary differential system

$$\begin{cases} \dot{y}(t) = B^{-1} A y(t) + B^{-1} f(t), \quad \forall \ \ t \in [t_0, T], \\ y(t_0) = y^{(0)}, \end{cases}$$

on which the results of [2] can be applied (i.e., estimates similar to those of Theorems 3.2, 3.3 can be obtained, where the $\mathcal{O}(h^p)$ term is absent).

Throughout this paper we assume that the scheme (2) is stable (see, e.g., [4, p.72]) and that, for $i = 1, \ldots, q$, y_i is obtained with an i-step stable scheme and of the same order as (2).

Since the linear system (3) is large and sparse and the right-hand sides, i.e., the $b_i' s$, will tend to get closer as the iterations unfold (see (17)), it is natural to use an iterative Krylov subspace method rather than a direct method even if the matrix C is constant for all iterations. However, unless a very good preconditioner is available, the effectiveness of such an iterative method strongly depends on the initial guess.

Our goal is to propose a cheap Petrov-Galerkin based approach that allows the computation of a good initial guess for each linear system to be solved in (3). We will illustrate this approach with the classical schemes mentioned above.

We emphasize that our objective is not the numerical solution of (1), but rather the computation of an approximate solution of the linear system (3) which will be used as initial guess in an iterative method in order to quickly arrive at a good approximate solution of (3). This will naturally lead to an acceleration, whatever the implicit scheme used. For this reason, issues specific to numerical methods for DAEs are not discussed. Such issues are well addressed in the literature (see, e.g., [4,17,18]).

This paper is organized as follows. In Section 2 we explain that the relations (2), (3) and (4) are shared by the classical implicit schemes. In Section 3 we briefly review the projection method of Petrov-Galerkin type, describe the proposed approach for computing a good initial guess to the linear systems (3), and provide estimate on the quality of the computed initial guess. The algorithmic aspect of the proposed approach is discussed and numerically illustrated in Section 4 and a conclusion is given in Section 5.

2. Examples of implicit schemes satisfying (2), (3) and (4)

As already mentioned, most standard implicit schemes can be expressed as in (2), (3) and (4). For example

- the implicit Euler scheme applied to (1) writes

$$B(y_{i+1} - y_i) = h(Ay_{i+1} + f(t_{i+1})). \tag{5}$$

It can be transformed to (2)-(3)-(4) with

$$a_i = y_i,\ C = B - hA,\ b_i = Ay_i + f(t_{i+1}). \tag{6}$$

- The Crank-Nicolson scheme applied to (1) writes

$$B(y_{i+1} - y_i) = \frac{h}{2}\left(A(y_i + y_{i+1}) + f(t_i) + f(t_{i+1})\right). \tag{7}$$

It can be transformed to (2)-(3)-(4) with

$$a_i = y_i,\ C = B - \frac{h}{2}A,\ b_i = Ay_i + \frac{f(t_i) + f(t_{i+1})}{2}. \tag{8}$$

- The backward differentiation formula (q-step BDF) applied to (1) writes

$$\sum_{j=0}^{q} \alpha_j B y_{i+j-q+1} = h\beta(Ay_{i+1} + f(t_{i+1})), \tag{9}$$

where the vectors $(y_j)_{0 \le j \le q-1}$ are given, $\alpha_q = 1$, and the coefficients $\alpha_0, \dots, \alpha_q, \beta$ are obtained by requiring that the order of accuracy of the method is as high as possible. In particular, the case $q = 1$ yields the implicit Euler scheme (also referred to as BDF1). For the intermediate values, $q = 2, 3, 4$ we have the following q-step BDF methods:

$$By_{i+1} - \frac{4}{3}By_i + \frac{1}{3}By_{i-1} = \frac{2}{3}h(Ay_{i+1} + f(t_{i+1})), \tag{10a}$$

$$By_{i+1} - \frac{18}{11}By_i + \frac{9}{11}By_{i-1} - \frac{2}{11}By_{i-2} = \frac{6}{11}h(Ay_{i+1} + f(t_{i+1})), \tag{10b}$$

$$By_{i+1} - \frac{48}{25}By_i + \frac{36}{25}By_{i-1} - \frac{16}{25}By_{i-2} + \frac{3}{25}By_{i-3} = \frac{12}{25}h(Ay_{i+1} + f(t_{i+1})), \tag{10c}$$

referred, respectively, to as BDF2, BDF3 and BDF4.
The scheme (9) can be rearranged as (2)-(3)-(4) with

$$a_i = -\sum_{j=0}^{q} \alpha_j y_{i+j-q},\ C = B - \beta hA,\ b_i = \beta(f(t_{i+1}) + Aa_i). \tag{11}$$

- The s-stage implicit Runge-Kutta scheme applied to (1) writes

$$By_{i,k} = A(y_i + h\sum_{j=1}^{s} a_{kj}y_{i,j}) + f(t_i + c_k h), \quad 1 \le k \le s, \tag{12a}$$

$$y_{i+1} = y_i + h\sum_{k=1}^{s} d_k y_{i,k}, \quad 0 \le i \le N-1, \tag{12b}$$

where the scalars c_k and d_k are given and the (unknown) quantities $y_{i,k}$ are estimates for $\dot{y}(t_i + c_k h)$. The equation (12a) defines a linear system of dimension $s.n$ of the form (3) where

$$z_i = \left(y_{i,1}^T, y_{i,2}^T, \ldots, y_{i,s}^T\right)^T, \; C = (I_s \otimes B) - h(A_0 \otimes A), \; b_i = (\mathbf{1}_s \otimes Ay_i) + F_i, \quad (13)$$

and where $\mathbf{1}_s = (1, \ldots, 1)^T \in \mathbb{R}^s$, $A_0 = (a_{kj})_{1 \le k,j \le s}$, $F_i = \left(f(t_i + c_1 h)^T, \ldots, f(t_i + c_s h)^T\right)^T$ and the symbol $\otimes$ denotes the Kronecker product.
The relation (12b) can then be cast in the form of (2) : $y_{i+1} = a_i + hZ_i$ with $a_i = y_i$, $Z_i = (d \otimes I_n) z_i$, where $d = (d_1, \ldots, d_s)$.

A detailed presentation of these schemes can be found, for example, in [1,4,15,17]. These schemes will be tested in Section 4.

3. Acceleration of implicit scheme

In this section we propose some subspaces of small dimension where a good initial guess for each linear system (3) is found using the Petrov-Galerkin process. Recall that the Petrov-Galerkin process for solving (3) requires a subspace $\mathcal{V}_i$ of small dimension in which an approximate solution $\hat{z}_i$ to z_i is found such that

$$\|b_i - C\hat{z}_i\| = \min_{z \in \mathcal{V}_i} \|b_i - Cz\|. \quad (14)$$

Here and throughout this paper, the symbol $\| \ \|$ denotes the 2-norm for vectors and matrices. The approximate solution $\hat{z}_i$ is given by $V_i x_i$, where V_i is a matrix whose columns form a basis of $\mathcal{V}_i$ and x_i is the solution of the projected linear system

$$\left((CV_i)^T CV_i\right) x_i = (CV_i)^T b_i. \quad (15)$$

Since the matrix $(CV_i)^T CV_i$ is of small size, the computation of x_i is not expensive. In the following theorem we define a subspace of small dimension that contains a good initial guess for the linear system (3).
Theorem 3.1 *Let $\mathcal{V}_i = span\{z_{i-r}, z_{i-(r-1)}, \ldots, z_{i-1}\}$. Then there exists a z in $\mathcal{V}_i$ such that for $i = q, \ldots, N-1$,*

$$\|b_i - Cz\| = \mathcal{O}(h^p) + \mathcal{O}(h^r), \quad (16)$$

where p is the order of the implicit scheme (2).
Proof Since $f, y \in C^r([t_0, T])$ (see Introduction), from Lagrange interpolation formula (see, e.g., [10]) we have, for $q \le i \le N-1$, $0 \le l \le q$ and $-1 \le k \le q$

$$\|y(t_{i-l}) - \sum_{m=1}^{r} \alpha_{m,r} y(t_{i-l-m})\| = \mathcal{O}(h^r), \quad (17a)$$

$$\|f(t_{i-k}) - \sum_{m=1}^{r} \alpha_{m,r} f(t_{i-k-m})\| = \mathcal{O}(h^r), \quad (17b)$$

where $\alpha_{m,r} = (-1)^{m-1} \frac{r!}{m!(r-m)!}$.
Since the scheme (2) is stable and of order p, we have (see [4, p. 72])

$$\begin{aligned}\|y_{i-l} - \sum_{m=1}^{r} \alpha_{m,r} y_{i-l-m}\| &= \|(y_{i-l} - y(t_{i-l})) + (y(t_{i-l}) - \sum_{m=1}^{r} \alpha_{m,r} y(t_{i-l-m})) \\ &\quad + \sum_{m=1}^{r} \alpha_{m,r}(y(t_{i-l-m}) - y_{i-l-m})\| \\ &= \mathcal{O}(h^p) + \mathcal{O}(h^r) \end{aligned} \tag{18}$$

Let $z = \sum_{m=1}^{r} \alpha_{m,r} z_{i-m}$. Then $z \in \mathcal{V}_i$ and from (3) and then (4) we obtain

$$\begin{aligned}\|b_i - Cz\| &= \|b_i - \sum_{m=1}^{r} \alpha_{m,r} b_{i-m}\| \\ &\leq \sum_{l=0}^{q} \left\| (\xi_l A + \psi_l B) \left(y_{i-l} - \sum_{m=1}^{r} \alpha_{m,r} y_{i-l-m} \right) \right\| \\ &+ \sum_{k=-1}^{q} |\phi_k| \left\| f(t_{i-k}) - \sum_{m=1}^{r} \alpha_{m,r} f(t_{i-k-m}) \right\|. \end{aligned}$$

Now from (17b) and (18), we obtain $\|b_i - Cz\| = \mathcal{O}(h^p) + \mathcal{O}(h^r)$. □

Actually, the motivation of the present work stems from the drawback of the subspace defined in Theorem 3.1 where the linear systems (3) are solved exactly for each integration step. To avoid this drawback, we use an approximation $\tilde{z}_{i-k}$ of z_{i-k}, $k = 1, \ldots r$. We therefore consider the following approximation of the scheme (2)

$$\begin{cases} \tilde{y}_{i+1} = \tilde{a}_i + h\tilde{z}_i, \text{ for } i = q, \ldots, N-1, \\ \tilde{y}_0 = y^{(0)}, \end{cases} \tag{19}$$

where, for $i = 1, \ldots, q$, $\tilde{y}_i$ is computed with an i-step scheme such that

$$\max_{1 \leq i \leq q} \|\tilde{y}_i - y_i\| = \mathcal{O}(\varepsilon), \tag{20}$$

with some tolerance threshold ε and where, for $i = q, \ldots, N-1$, $\tilde{z}_i$ is an approximation of z_i, computed by an iterative method such that

$$\|\tilde{b}_i - C\tilde{z}_i\| \leq \varepsilon \|\tilde{b}_i\| \tag{21}$$

and where $\tilde{a}_i$ and $\tilde{b}_i$ are obtained by replacing the $y'_j s$ in the expression of a_i in (2) and b_i in (4) by the $\tilde{y}'_j s$.
Since (2) is assumed to be stable, we deduce from (20) and (21) that (see [4, p.72]):

$$\max_{0 \leq i \leq N} \|\tilde{y}_i - y_i\| = \mathcal{O}(\varepsilon). \tag{22}$$

In the next theorem, we redefine $\mathcal{V}_i$ as the subspace spanned by the last r vectors $\tilde{z}_{i-k}$, and we show that it contains a good initial guess of the linear system (3).
Theorem 3.2 *Let $\mathcal{V}_i = span\{\tilde{z}_{i-r}, \tilde{z}_{i-(r-1)}, \ldots, \tilde{z}_{i-1}\}$ where $\tilde{z}_{i-m}$, $1 \leq m \leq r$, satisfy (21). Then there exists a z in $\mathcal{V}_i$ such that for $i = q, \ldots, N-1$,*

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

Proof Let $z = \sum_{m=1}^{r} \alpha_{m,r} \tilde{z}_{i-m}$. Then, $z \in \mathcal{V}_i$ and we can write

$$
\begin{aligned}
\|\tilde{b}_i - Cz\| &= \|\tilde{b}_i - \sum_{m=1}^{r} \alpha_{m,r}\tilde{b}_{i-m} + \sum_{m=1}^{r} \alpha_{m,r}\tilde{b}_{i-m} - \sum_{m=1}^{r} \alpha_{m,r} C\tilde{z}_{i-m}\| \\
&\leq \|\tilde{b}_i - \sum_{m=1}^{r} \alpha_{m,r}\tilde{b}_{i-m}\| + \sum_{m=1}^{r} |\alpha_{m,r}|\ \|\tilde{b}_{i-m} - C\tilde{z}_{i-m}\|.
\end{aligned}
$$

From (21), we have $\sum_{m=1}^{r} |\alpha_{m,r}|\ \|\tilde{b}_{i-m} - C\tilde{z}_{i-m}\| = \mathcal{O}(\varepsilon)$.
On the other hand we have

$$
\begin{aligned}
\|\tilde{b}_i - \sum_{m=1}^{r} \alpha_{m,r}\tilde{b}_{i-m}\| \leq & \sum_{l=0}^{q} \left\| (\xi_l A + \psi_l B) \left(\tilde{y}_{i-l} - \sum_{m=1}^{r} \alpha_{m,r}\tilde{y}_{i-l-m} \right) \right\| \\
& + \sum_{k=-1}^{q} |\phi_k| \left\| f(t_{i-k}) - \sum_{m=1}^{r} \alpha_{m,r} f(t_{i-k-m}) \right\|
\end{aligned}
$$

and

$$
\begin{aligned}
\|\tilde{y}_{i-l} - \sum_{m=1}^{r} \alpha_{m,r}\tilde{y}_{i-l-m}\| \leq & \|\tilde{y}_{i-l} - y_{i-l}\| + \|y_{i-l} - \sum_{m=1}^{r} \alpha_{m,r} y_{i-l-m}\| \\
& + \sum_{m=1}^{r} |\alpha_{m,r}|\ \|y_{i-l-m} - \tilde{y}_{i-l-m}\|.
\end{aligned}
$$

The proof follows by using (22) and proceeding as in the proof of Theorem 3.1. □

Suppose now that at some certain integration steps, the Petrov-Galerkin approximation $\hat{z}_i$ satisfies (21). This dispenses with the use of an iterative method to compute $\tilde{z}_i$. We set $\tilde{z}_i = \hat{z}_i \in \mathcal{V}_i$ and use the same subspace $\mathcal{V}_i$ to compute the next initial guess $\hat{z}_{i+1}$. This reduces the amount of computations in the proposed approach and leads us to re-define the subspace $\mathcal{V}_i$. A clarification is given in the following theorem.

Theorem 3.3 *Let $\mathcal{V}_i = span\{\tilde{z}_{i-l_k},\ 1 \leq k \leq m\}$, $l_k < l_{k+1}$ be the subspace spanned by the last m vectors $\tilde{z}_{i-l_k}$, $k = 1, \ldots, m$, whose computation necessitate the use of an iterative method to satisfy (21) and let $r = m + l_1 - 1$. Then there exists a z in $\mathcal{V}_i$ such that for $i = q, \ldots, N-1$,*

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

Proof Since $\tilde{z}_{i-l_1}$ is the last vector whose computation necessitates the use of an iterative method, the computation of $\tilde{z}_{i-j}$ for $j = 1, \ldots, l_1 - 1$ does not necessitate the use of an iterative method and therefore $\tilde{z}_{i-j} = \hat{z}_{i-j} \in \mathcal{V}_i$ for $j = 1, \ldots, l_1 - 1$. This means that $\mathcal{V}_i$ is spanned by the vectors $\{\tilde{z}_{i-l_k},\ k = 1, \ldots, m\}$ and the vectors $\{\tilde{z}_{i-j},\ j = 1, \ldots, l_1 - 1\}$. Then, as in the proof of Theorems 3.1 and 3.2, we show the existence of a vector $z \in \mathcal{V}_i$ such that

$$\|\tilde{b}_i - Cz\| = \mathcal{O}(h^p) + \mathcal{O}(h^{m+l_1-1}) + \mathcal{O}(\varepsilon).$$

□

4. Algorithm and numerical results

The computation of the sequence $(\tilde{y}_i)$ is formally summarized in the following algorithm.

Algorithm 1 [Computation of the sequence $\tilde{y}_i$ defined in (19)]

Input: $\tilde{y}_1, \ldots, \tilde{y}_q$

We assume that $\tilde{y}_1, \ldots, \tilde{y}_q$ are either given or computed with an k-step scheme such that $\max_{1 \le k \le q} \|\tilde{y}_k - y_k\| = \mathcal{O}(\varepsilon)$.

Output: $(\tilde{y}_i)_{q+1 \le i \le N}$.

1: Set $i = q$. 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 in V_i. Compute $L_i = CV_i$, $C_i = L_i^T L_i$, $\tilde{a}_i$ and $\tilde{b}_i$.
2: **while** $i \le N - 1$ **do**
3:   compute the initial guess $\hat{z}_i = V_i C_i{}^{-1} L_i^T \tilde{b}_i$
4:   **if** $\|\tilde{b}_i - C\hat{z}_i\| \le \varepsilon \|\tilde{b}_i\|$ **then**
5:     set $\tilde{y}_{i+1} = \tilde{a}_i + h\hat{z}_i, R_{i+1} = R_i$, $V_{i+1} = V_i$, $C_{i+1} = C_i$, $L_{i+1} = L_i$, compute $\tilde{b}_{i+1}$, set $i = i + 1$ and go to (3)
6:   **else**
7:     starting with $\hat{z}_i$, compute by a Krylov subspace method $\tilde{z}_i$ that satisfies (21)
8:   **end if**
9:   set $\tilde{y}_{i+1} = \tilde{a}_i + h\tilde{z}_i$ and compute $\tilde{b}_{i+1}$
10:   let k_0 be the number of columns of R_i. If $k_0 < r$ then $R_{i+1} = [R_i, \tilde{z}_i]$, else $R_{i+1} = [S_i, \tilde{z}_i]$, 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}
11:   Compute $L_{i+1} = CV_{i+1}$, $C_{i+1} = L_{i+1}^T L_{i+1}$, and set $i = i + 1$
12: **end while**

In Algorithm 1 and in our numerical tests, the subspace $\mathcal{V}_i$ is the one given by Theorem 3.3. If we want to use the subspace $\mathcal{V}_i$ of Theorem 3.2, then step 5 should be replaced by:

$$\text{set } \tilde{z}_i = \hat{z}_i, \text{ and go to 9.}$$

To illustrate the numerical behavior of Algorithm 1, we consider equation (1) with the following data:

- $[t_0, T] = [0, 1]$,
- the matrices A and B result from the discretization of a hydrodynamic problem (see [14])
$$A = \begin{pmatrix} F & -G \\ G^T & 0 \end{pmatrix}, \quad B = \begin{pmatrix} I_{n_f} & 0 \\ 0 & 0 \end{pmatrix}, \tag{23}$$
where $G \in \mathbb{R}^{n_f \times n_g}$ has full rank, $F \in \mathbb{R}^{n_f \times n_f}$, and I_{n_f} is the identity matrix of size n_f, $n_f = 11200$, $n_g = 3999$, the size of A and B is $n = n_f + n_g = 15199$,
- the right hand side $f(t)$ is given by $f(t) = (f_1(t), \ldots, f_n(t))^T$ where $f_k(t) = e^{-tk\delta x} \sin(k\delta x)$ with $\delta x = 1/(n+1)$,
- the initial value $y^{(0)}$ is given by $y^{(0)} = (y_1^{(0)}, \ldots, y_n^{(0)})^T$ where $y_k^{(0)} = \cos(k\delta x)$.

Remark 4.1 *(i) The assumption that G has full rank is important since, otherwise*

$$\ker A \cap \ker B = \left\{ \begin{pmatrix} 0 \\ x \end{pmatrix} : x \in \ker G \right\},$$

which implies that the pencil $\lambda B - A$ is singular.

(ii) With the notation (23), the system (1) can be transformed to the ordinary differential system

$$\begin{cases} \dot{y}(t) = \mathbb{A}y(t) + \mathbb{f}(t), \quad \forall \;\; t \in [t_0, T], \\ y(t_0) = y^{(0)}, \end{cases}$$

where

$$\mathbb{A} = \begin{pmatrix} F & -G \\ G^\dagger F^2 & -G^\dagger F G \end{pmatrix}, \; \mathbb{f}(t) = \begin{pmatrix} f_1(t) \\ G^\dagger(F f_1(t) + \dot{f}_1(t)) + (G^T G)^{-1} \ddot{f}_2(t)) \end{pmatrix}, \; f(t) = \begin{pmatrix} f_1(t) \\ f_2(t) \end{pmatrix},$$

and $G^\dagger = (G^T G)^{-1} G^T$ is the pseudo-inverse of G.

Therefore, in theory, we could use the subspaces defined in [2]. However, in practice, doing so does not allow to take the structure of the coefficient matrices of (23).

Algorithm 1 is implemented and executed in MATLAB 7.13 on an Intel Core 2 Duo 3.16 Ghz processor. The following parameters are used in the algorithm: $r = 20$, $h = 1/100$, $\varepsilon = 10^{-8}$, the computation of $\tilde{z}_i$ (step 7 of Algorithm 1) is done by restarted GMRES [20] with restart value 20.

The scheme (19) is obtained with implicit Euler, Crank Nicolson, BDF4 (which requires a prior implementation of BDF1, BDF2 and BDF3 for computing $\tilde{y}_1$, $\tilde{y}_2$, and $\tilde{y}_3$, see (10)), and implicit Runge-Kutta. For the latter, we consider the 3-stage scheme (see (12)) given by (see, e.g., [17, p. 226])

$$A_0 = \begin{pmatrix} \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\ \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} \\ \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} \end{pmatrix}, \quad \begin{array}{l} d_1 = \frac{5}{18}, \quad d_2 = \frac{4}{9}, \quad d_3 = \frac{5}{18} \\ c_1 = \frac{1}{2} - \frac{\sqrt{15}}{10}, \; c_2 = \frac{1}{2}, \; c_3 = \frac{1}{2} + \frac{\sqrt{15}}{10}. \end{array}$$

For each scheme, Figure 1 shows the residual norm $\|\tilde{b}_i - C\hat{z}_i\|/\|\tilde{b}_i\|$ associated with the initial guess $\hat{z}_i$ computed by the proposed approach, versus the number of iterations of the scheme. Figures 2 shows, for each scheme, the number of GMRES iterations for computing $\tilde{z}_i$ satisfying (21) and starting with $\hat{z}_i$. Figures 3 shows the run time required for each iteration of the used schemes.

The figures show that BDF provides the best results, followed by Crank Nicolson and implicit Euler. The relatively less good performance of implicit Runge-Kutta can be explained by the need to solve, at each iteration of the scheme, a linear system of size (see (13)): $s \times n = 3 \times 15199 = 45597$.

Overall, these results show that the subspace $\mathcal{V}_i$ has led to a significant acceleration of the used schemes. Table 1 shows the total number of GMRES iterations and the total time required for computing all the iterates (y_i). Table 2 shows, for some iterations i of the scheme, the number of iterations required by GMRES to compute $\tilde{z}_i$ starting with the initial solution $\hat{z}_i$ such that (21) holds. We see from this table that, at some iterations i, GMRES is not used, which means that $\hat{z}_i$ already satisfies (21). This can also be seen in Figures 1 and 2. Figures 1, 2, 3 and Table 2 show that during the first iterations of

Algorithm 1, the subspace $\mathcal{V}_i$ does not contain enough information, and this explains the large relative residual norm, the number of GMRES iterations and the run time. These numbers decrease as the subspace $\mathcal{V}_i$ contains more information on the last iterates of the scheme.

For comparison purposes, we computed the total number of GMRES iterations and the total time required for computing all the iterates (y_i) when $\hat{z}_i = 0$ is taken as initial guess in step 3 of Algorithm 1. The results for implicit Euler, Crank-Nicolson and BDF schemes are given in Table 3. For implicit Runge-Kutta, we had to interrupt the computation since after 24 hours, only the first two iterations i = 1, 2 were executed. This table has to be compared with Table 1. It shows that the proposed initial guess $\hat{z}_i$ strongly accelerates the computation of the sequence (y_i).

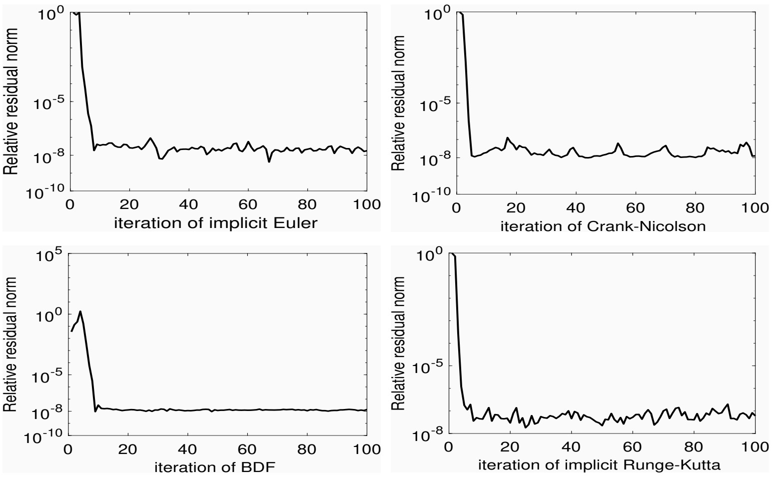

Fig. 1. Relative residual norm of the initial guess versus the number of iterations of the scheme

	Time (min.)	# GMRES iter.
IE	9.20	167408
CN	5.16	92640
BDF	7.76	140996
IRK	2773	69296

Table 1
Total CPU time in minutes and total number of GMRES iterations starting with $\hat{z}_i$ to satisfy (21)

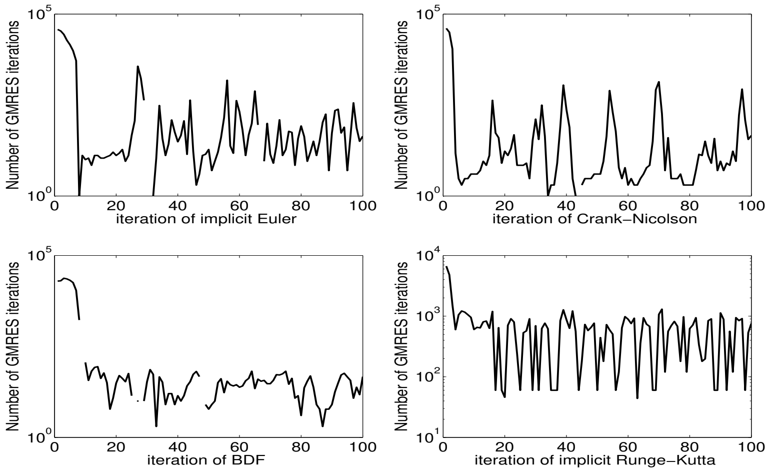

Fig. 2. Number of GMRES iterations required to satisfy (21) starting with $\hat{z}_i$ versus the number of iterations of the scheme

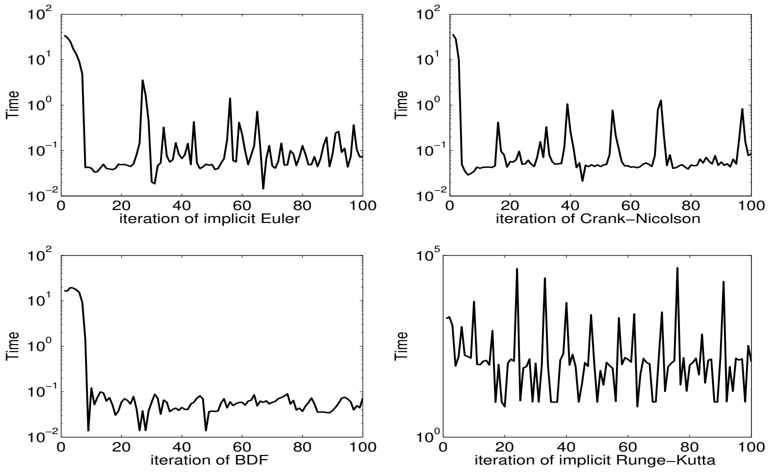

Fig. 3. Run time (in seconds) versus the number of iterations of the scheme

iteration i	1	20	30,31	40	60	80	100
IE	38931	13	0	14	40	28	280
iteration i	1	20	34,35	40	60	80	100
CN	40552	24	0	828	2	7	7
iteration i	1	20	9,26,28,48,79	40	60	80	100
BDF	19822	31	0	14	26	6	27
iteration i	1	20	40		60	80	100
IRK	6696	46	877		890	624	762

Table 2
Number of iterations required by GMRES when implicit Euler (IE), Crank Nicolson (CN), BDF and implicit Runge-Kutta (IRK) are used

	Time (min.)	# GMRES iter.
IE	70.49	2406702
CN	301.06	3596560
BDF	149.83	2470509

Table 3
Total CPU time in minutes and total number of GMRES iterations starting with 0 initial guess ($\hat{z}_i = 0$)

5. Conclusion

This work was concerned with the choice of the initial guess in the iterative solution of the linear systems that arise in implicit schemes for large linear differential-algebraic systems. A good choice leads to an acceleration of the scheme. To this end, a Petrov-Galerkin based approach has been developed to extract, at each iteration of the scheme, a good initial guess from a subspace of small dimension that contains information on the preceding iterates of the schemes. The estimates obtained on the norm of the residuals associated with the linear systems show that the accuracy depends on the stepsize (of the discretized time), the order of the schemes, and the dimension of the subspace $\mathcal{V}_i$. The effectiveness of this approach has been illustrated in the case where the linear systems are solved by GMRES and the schemes used are implicit Euler, Crank Nicolson, BDF and implicit Runge-Kutta. The user can easily adapt this strategy to other implicit schemes and possibly combine it with other strategies such as preconditioning [20, chap. 9-14], recycling [19] or deflation and augmentation [12].

Acknowledgements. The authors would like to thank the reviewers for useful remarks and suggestions.

References

[1] U.M. Ascher, L.R. Petzold, Computer methods for ordinary differential equations and differential-algebraic equations, SIAM, Philadelphia, PA, 1998.

[2] M. Al Sayed Ali, M. Sadkane, Improved predictor schemes for large systems of linear ODEs, Electron. Trans. Numer. Anal., 39 (2012), pp. 253–270.

[3] P. Benner, ; P. Losse, V. Mehrmann, M. Voigt, Numerical linear algebra methods for linear differential-algebraic equations. Surveys in differential-algebraic equations. III, 117175, Differ.-Algebr. Equ. Forum, Springer, Cham, 2015.

[4] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical solution of initial-value problems in differential-algebraic equations, SIAM, Philadelphia, PA, 1989.

[5] M.V. Bulatov, L. Ming-Gong, L.S. Solovarova, On first- and second-order difference schemes for differential-algebraic equations of index at most two, Computational Mathematics and Mathematical Physics, 50 (2010), pp. 1808–1817.

[6] M.V. Bulatov, V.H. Linh, L.S. Solovarova, On BDF-based multistep schemes for some class of linear differential-algebraic equations of index at most two, Acta Math. Vietnam, 41 (2016), pp. 715–730.

[7] M.V. Bulatov, V.H. Linh, L.S. Solovarova, Block difference schemes of high order for stiff linear differential-algebraic equations, Computational Mathematics and Mathematical Physics, 59 (2019), pp. 1049–1057.

[8] S.L. Campbell, Singular systems of differential equations, Research Notes in Mathemat-ics, Pitman, 1980.

[9] S. L. Campbell, P. Kunkel, On the numerical treatment of linear-quadratic optimal control problems for general linear time-varying differential-algebraic equations, J. Comput. Appl. Math., 242 (2013), pp. 213–231.

[10] P.J. Davis, Interpolation and approximation, Blaisdell, New York, 1963.

[11] F. R. Gantmacher, The Theory of Matrices II. Chelsea Publishing Company, New York, 1959.

[12] A. Gaul, M.H. Gutknecht, J. Liesen, R. Nabben, A Framework for deflated and augmented Krylov subspace methods, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 495–518.

[13] M. Hanke, R. März, C. Tischendorf, E. Weinmller, and S. Wurm, Least-squares collocation for linear higher-index differential-algebraic equations, J. Comput. Appl. Math., 317 (2017), pp. 403–431.

[14] G. Hechme, Y. M. Nechepurenko, and M. Sadkane, Efficient methods for computing spectral projectors for linearized hydrodynamic equations, SIAM J. Sci. Comput., 31 (2008), pp. 667–686,

[15] E. Hairer, G. Wanner, Solving ordinary differential equations II : Stiff and differential-algebraic problems, 2nd ed., Springer-Verlag, Berlin, 1996.

[16] W. Hundsdorfer, J.G. Verwer, Numerical solution of time-dependent advection-diffusion-reaction equation, Springer-Verlag, Berlin, 2003.

[17] P. Kunkel, V. Mehrmann, Differential-Algebraic Equations Analysis and Numerical Solution, EMS Publishing House, Zrich, Switzerland, 2006.

[18] R. März, Numerical methods for differential-algebraic equations, Acta Numerica, (1992), pp. 141–198.

[19] M.L. Parks, E. de Sturler, G. Mackey, D.D. Johnson, S. Maiti, Recycling Krylov subspaces for sequences of linear systems, SIAM J. Sci. Comput., 28(2006), pp. 1651–1674.

[20] Y. Saad, Iterative methods for sparse linear systems, 2nd ed., SIAM, Philadelphia, PA, 2003.

[21] M. Shirvani, J.W.-H. So, Solutions of linear differential algebraic equations, SIAM Rev. 40, (1998), pp. 344–346.

[22] M. Takamatsu, S. Iwata, Index reduction for differential-algebraic equations by substitution method, Linear Algebra Appl., 429 (2008), pp. 2268–2277.