Numerical solution of differential algebraic equations and computation of consistent initial/boundary conditions

Pierluigi Amodio, Francesca Mazzia*
Dipartimento di Matematica, Via E. Orabona 4, 70125 Bari, Italy
Received 3 September 1996; received in revised form 2 September 1997

Abstract

Boundary value methods for the solution of differential–algebraic equations are described. We consider both initial and boundary value problems and derive an algorithm that does not require additional information from the user, but only the initial or boundary conditions needed in theory to obtain a unique solution.

Keywords: Differential–algebraic equations; Boundary value methods

1. Introduction

We are concerned with the numerical solution of differential–algebraic equations of the form

\[ M(t)y' = f(t, y), \quad t \in [a, b], \]  

where the matrix \( M(t) \) may be singular for all values of the time variable \( t \). Such kind of differential equations arises from many mathematical models as, e.g., simulation of electric circuits, chemical reactions subject to invariants, etc. [6, 12, 13].

In order to obtain a unique solution, Eq. (1) must be provided with a set of initial or boundary conditions

\[ B(y(a), y(b)) = 0. \]  

The number of these conditions depends on the number of differential equations in (1), and hence is smaller than the size of the unknown \( y \).

The theory of numerical methods for DAEs has always been strictly connected to that of differential equation solvers. In general the idea is that of generalizing numerical methods for ODEs, in particular

* Corresponding author. E-mail: labor@alphamath.dm.uniba.it.
1 Work supported by M.U.R.S.T. (40% project).
Runge–Kutta methods and backward differentiation formulae (BDFs). A problem may derive from the fact that these methods require a complete set of initial/boundary conditions (a number of conditions equal to the size of the system) while, as said previously, the DAE provides only a subset of them. This problem has been studied in several papers [19, 20, 25], but till now every solution seems to be expensive or anyway is obtained by means of difficult procedures.

The most known packages for the solution of DAEs are DASPK, DASSL [6] and RADAU5 [13] for solving initial value problems, and COLDAE [5] that solves BVPs. Some of them require in input a complete set of consistent boundary conditions.

In this paper we are dealt with Boundary Value Methods (BVMs), a class of new methods that have been successfully used for the solution of initial and boundary value ODE problems [2, 7–9]. The basic idea of these methods is that the differential problem is approximated by means of a discrete boundary value one, and hence the solution is computed simultaneously at several grid points. A BVM may be applied to all the time interval, or, if \([a,b]\) is divided in smaller subintervals, sequentially to each of them. The theoretical properties of this new approach, as well as the numerical implementation on parallel computers, has been largely analyzed [1, 15, 21–23]. Moreover, complete families of methods have been determined and studied, obtaining stable methods of any high order.

Two steps BVMs have been applied to linear DAEs in [3], while convergence properties of some \(k\)-step BVMs are analyzed in [14, 24] for linear and nonlinear semi-explicit DAE BVPs and IVPs of index one. In this paper we consider the solution of nonlinear DAEs of different indices by means of a class of BVMs called generalized backward differentiation formulae (GBDFs). We derive a theorem about the convergence of the methods and a very interesting strategy to avoid the computation of a complete set of consistent boundary conditions.

These main results are the object of Sections 3 and 4. In the previous section we present the basic concepts related to BVMs, along with the family of GBDFs. The last section is devoted to some numerical tests on classical linear and nonlinear DAEs.

2. Boundary value methods

Suppose, for simplicity, that the matrix \(M(t)\) in (1) is constant, that condition (2) gives a complete set of consistent boundary conditions (the problem of avoiding their calculation will be the matter of the next section), and that a uniform mesh (with \(N + 1\) points) is defined on the integration interval. The simplest way to define BVMs is to consider the application of a \(k\)-step Linear Multistep Formula (LMF)

\[
\sum_{i=0}^{k} z_i M_{\alpha+i} = h \sum_{i=0}^{k} \beta_i f_{\alpha+i}, \quad n = 0, \ldots, N - k,
\]

over the mesh, relaxing the usual request of assigning all the \(k\) conditions needed by (3) at the initial points \(t_0, t_1, \ldots, t_{k-1}\). On the other hand, let \(k_1\) and \(k_2\) be two positive integers such that \(k_1 + k_2 = k\), we fix the first \(k_1\) values, \(y_0, y_1, \ldots, y_{k_1-1}\), of the discrete solution, and the final \(k_2\) ones, \(y_{N-k_2+1}, \ldots, y_N\). In this way problem (1)–(2) is approximated by means of a discrete BVP with \((k_1, k_2)\)-boundary conditions.

Even if earlier references exist, the methods originating from this approach have been systematically examined only in the last years, starting from [21]. In particular, the usual notions of 0-stability
and \( A \)-stability have been generalized, leading to the definitions of \( 0_{k, k_2} \)-stability and \( A_{k, k_2} \)-stability [2, 8]. It has also been shown that no more order barrier exists for such methods. In fact, several families of methods have been found, each containing \( 0_{k, k_2} \)-stable, \( A_{k, k_2} \)-stable formulae of arbitrary high order [2, 8, 9], including formulae of order \( p = 2k \), which is the highest possible order for a \( k \)-step LMF.

In practice, in its actual implementation, (3) is used together with a set of \( k - 1 \) additional equations that involve the same unknowns \( y_0, \ldots, y_N \). Since we suppose to have a complete set of consistent boundary conditions, we obtain \( N + 1 \) different equations that give a unique numerical solution. The resulting method is called \textit{boundary value method} or, shortly, \textit{BVM}.

Generalized BDFs (GBDFs) are a family of BVMs introduced in [8] as a generalization of the Backward Differentiation Formulae. When a constant grid-size \( h \) is used, the \( k \)-step GBDF with \( k_1 = \lceil k/2 \rceil + 1 \) and \( k_2 = k - k_1 \) associated to (1) is defined as

\[
\sum_{i=0}^{k} \xi_i\zeta_n M_{y_i} = h f_n, \quad n = 1, \ldots, k_1 - 1,
\]

\[
\sum_{i=0}^{k} \xi_i M_{y_{n-k+1+i}} = h f_n, \quad n = k_1, \ldots, N - k_2,
\]

\[
\sum_{i=0}^{k} \xi_i^{(N+k)} M_{y_{N-k+i}} = h f_n, \quad n = N - k_2 + 1, \ldots, N,
\]

where the first \( k_1 - 1 \) equations are defined by the \textit{initial additional methods}, the intermediate set of equations are defined by the \textit{main method}, and the last \( k_2 \) equations are obtained through the \textit{final additional methods}. If we solve an initial value problem, (4) corresponds to the following nonlinear equation,

\[
(A_N \otimes M)\mathbf{y} = h \mathbf{f} - (a_N \otimes M)\mathbf{y}_0,
\]

where \( \mathbf{y} = (y_1^T, \ldots, y_N^T)^T, \mathbf{f} = (f_1^T, \ldots, f_N^T)^T \) and \( [a_N \ A_N] \) is a \( N \times (N + 1) \) matrix containing the coefficients \( \xi_i \) of the methods. All those coefficients are such that the corresponding formulae are of order \( k \). In this way, the whole method (5) has order \( k \). The stability properties essentially depend on the main method. For example, it is proved that any \( k \)-step GBDF is \( A_{k, k_2} \)-stable [8].

A different way to approximate problem (1) is to consider a block BVM. In few words, such block version amounts to discretize the time interval by using a variable coarser mesh defined by the \( p + 1 \) points

\[ \tau_i = \tau_{i-1} + \hat{h}_i, \quad i = 1, \ldots, p, \]

where \( \tau_0 = t_0 \) and \( \tau_p = T \), and, on each subinterval \( [\tau_{i-1}, \tau_i] \), a discretization with \( s \geq k \) equally spaced points

\[ t_{(i-1)s+j} = t_{(i-1)s+j-1} + \hat{h}_j, \quad j = 1, \ldots, s, \]

where \( \hat{h}_j = \hat{h}_{i}/s \) and \( t_{(i-1)s} = \tau_{i-1} \). Then a BVM in the form (5)

\[
(A_s \otimes M)\mathbf{y}^{(i)} = \hat{h}_i \mathbf{f}^{(i)} - (a_s \otimes M)\mathbf{y}_{(i-1)s}, \quad i = 1, \ldots, p
\]
is applied, where \( \mathbf{f}^{(i)} = (\mathbf{f}_{i-1}^{T} y_{i-1}, \ldots, \mathbf{f}_{i}^{T} y_{i}^{T})^{T} \) and \( \mathbf{y}^{(i)} = (y_{i-1}^{T} y_{i-1}, \ldots, y_{i}^{T} y_{i})^{T} \) contains the numerical solutions inside the subinterval \( (\tau_{i-1}, \tau_{i}] \).

The resulting discrete problem is of size \( N + 1 \), where now \( N = s \cdot p \). For initial value ODE problems its stability and convergence properties have been stated in [16], where a complete analysis of block BVMs has been carried out starting from the theory on general linear methods. The numerical aspects related to this kind of problem are studied in [1, 10, 17], where a sequential and a parallel code based on block BVMs are presented.

### 3. The initial/boundary conditions calculation

If we simply apply a method for ODEs to a system of differential–algebraic equations, we do not have, in general, any initial information about the algebraic equations of the problem, and hence the procedure cannot start. For this reason, the system of nonlinear equations (6) should require a complete set of consistent initial/boundary conditions to determine the numerical solution.

If the problem is purely algebraic (no boundary conditions are provided), it is natural to add to (4) another equation involving \( y_0, \ldots, y_s \) or, in case of block BVMs, \( y_0, \ldots, y_s \). This equation must be such that the set of \( N + 1 \) equations has a unique numerical solution. For GBDFs we can choose the following equation:

\[
\sum_{i=0}^{k} \lambda_i^{(0)} M y_i = h f_0,
\]

where the coefficients \( \lambda_i^{(0)} \) are again so that the formula has order \( k \).

Let us analyze this kind of problem in more detail; for simplicity first consider the solution of the following linear algebraic equation

\[
E y' + y = g(t),
\]

where \( E \) is a nilpotent matrix with only the first lower offdiagonal elements different from zero. The discrete problem associated to (8) differs from (6) for the first step and can be represented as

\[
(A_s \otimes E + h_1 I_{s+1} \otimes I) \mathbf{y}^{(1)} = h_1 \mathbf{g}^{(1)},
\]

\[
(A_s \otimes E + h_1 I_s \otimes I) \mathbf{y}^{(i)} = h_i \mathbf{g}^{(i)} - (a_s \otimes E) y^{(i-1)}_{(i-1)s}, \quad i = 2, \ldots, p,
\]

where \( \mathbf{y}^{(1)} = (y_0^{T} y_0^{(1)} y_0^{(1)})^{T} \), \( \mathbf{g}^{(i)} = (g(t_{i-1}s+1)^{T}, \ldots, g(t_0)^{T})^{T} \), for \( i = 1, \ldots, p \), \( \mathbf{g}^{(i)} = (g(t_0)^{T}, g^{(i)}{y})^{T} \) and \( A_s \) is a \((s + 1) \times (s + 1)\) matrix containing the coefficients \( \lambda_i^{(0)} \) in the first row and \( [a_s A_s] \) as the remaining rows.

In (9) the coefficient matrix that gives \( \mathbf{y}^{(1)} \) is similar to a block bidiagonal matrix with diagonal elements equal to \( h_1 I_{s+1} \), and is therefore nonsingular for each stepsize \( h_1 \neq 0 \). The same hypothesis on all the used stepsizes \( h_i \neq 0 \) is also sufficient to guarantee that a unique numerical solution exists on the entire time interval under consideration. In [24] it is then proved that the numerical solution converges to the exact solution with order

\[
p_k = \min(k, k - v + 2),
\]

where \( v \) is the index of nilpotency of \( E \) and \( k \) is the order of the method.
A linear problem more general than (8) may be decoupled in the form

\[ y_1' + G y_1 = g_1(t), \quad E y_2' + y_2 = g_2(t). \]  

The second equation is purely algebraic and may be solved with the above described approach without requiring boundary conditions. The first one is a differential equation, and may be solved by means of (6) with the addition of the initial/boundary conditions.

If the equations are not decoupled and we want to handle in the same way the set of differential and algebraic equations, we cannot use the previous approach since we have not sufficient information to start the procedure. Therefore, we adopt the strategy of requiring that the boundary conditions must not be exactly satisfied, but they may have an error of the same order \( p_k \) of the method used. Moreover, the numerical solution must converge to the exact one on all the components \( y_0, \ldots, y_N \). This can be achieved by substituting Eq. (7) with the following

\[ \sum_{i=0}^{k} z_i^{(0)} M y_i + \gamma (B_a y_0 + B_b y_N) = h f_0 + \gamma \eta, \]  

where \( B_a y_0 + B_b y_N = \eta \) are the boundary conditions of the continuous problem and \( \gamma \neq 0 \) is a constant. That is, Eq. (12) is a linear combination between the boundary conditions and the first point method. The key point is that the boundary conditions are only those given by the continuous problem. Namely, they do not contain an additional set of consistent conditions.

Using this strategy we obtain a discrete problem that in the linear constant coefficient case can be decoupled in two different subsystems. For the algebraic variables we obtain the system (9), which has been proved to be convergent to the exact solution.

Even though the above procedure works for general boundary conditions (see Section 5), we here only study the case of initial conditions. In such a case, to analyze what happens to the differential variables we consider the solution of the initial value problem

\[ y' = f(t,y), \quad y(t_0) = \eta, \]  

where \( f \) is a sufficiently smooth function. The discretization by using a \( k \)-step block GBDF, with (12) as first equation, leads to the following nonlinear problem:

\[ (A_s \otimes I + \gamma e_1 e_1^T \otimes I) y^{(1)} = h_1 f^{(1)} + \gamma e_1 \otimes \eta, \]

\[ (A_s \otimes I) y^{(i)} = h_i f^{(i)} - (a_s \otimes I) y_{(i-1)}, \quad i = 2, \ldots, p, \]  

where \( e_1 \) is the first unit vector in \( \mathbb{R}^{s+1} \). It is clear that in this case \( y_0 \) is completely defined and hence this strategy is not useful, but effectively we obtain this kind of discrete system after the decoupling of the algebraic and the differential equations.

**Theorem 3.1.** The numerical solution of the initial value problem (13) by using the \( k \)-step block GBDF (14) is convergent of order \( k \).

**Proof.** In [16] it has been proved that the \( k \)-step block GBDF applied to an IVP is convergent of order \( k \). Moreover, it is shown that, to state the convergence, it is sufficient that each linear multistep formula has order \( k \) and the coefficient matrix \( A_s \) is nonsingular.
This means that we only need to state that the matrix $\tilde{A}_s + \gamma e_1 e_1^T$ is nonsingular, since the coefficients of the first formula,

$$\sum_{i=0}^{k} \tilde{a}_i(0) y_i + \gamma y_0 = h f_0 + \gamma \eta,$$

are such that its order is $k$. This result derives by observing that the matrix $A$ is nonsingular while $\tilde{A}$ is singular. In fact, each row of this matrix contains the coefficient of the polynomial $p$ of a linear multistep formula and the singularity derives by the order conditions. The matrix $A$ may be written as

$$\tilde{A}_s = \begin{pmatrix} \tilde{a}_0(0) & w^T \\ a_s & \tilde{A}_s \end{pmatrix},$$

and $\det(\tilde{A}_s) = \tilde{a}_0(0) \det(A_s) + \tilde{e} = 0$ with $\det(A_s) \neq 0$. Then one has that $\det(\tilde{A}_s + \gamma e_1 e_1^T) = (\tilde{a}_0(0) + \gamma) \det(A_s) + \tilde{e} = \gamma \det(A_s) \neq 0$ since $\gamma \neq 0$.

### 4. Nonlinear semi-explicit index one DAE

The class of semi-explicit index-one DAEs is characterized by the dependent variable vector $y$ splitted into a vector $x$ called the differential variable, and a vector $z$ called the algebraic variable, such that the equations have the form

$$x' = f(t, x, z), \quad 0 = g(t, x, z)$$

where $\frac{\partial g}{\partial z}$ is a nonsingular square matrix.

Semi-explicit index one DAEs have been largely studied since they arise frequently in problems in science and engineering.

We now extend the convergence properties of the $k$-step block GBDFs to semi-explicit index one DAEs of the form (15). The discrete problem resulting from the application of the GBDFs with the first point method (12) is

$$(A_s \otimes I + \gamma e_1 e_1^T \otimes I) \tilde{x}^{(i)} = h f^{(i)} + \gamma e_1 \otimes \eta,$$

$$\tilde{g}^{(i)} = 0,$$

$$(A_s \otimes I) x^{(i)} = h f^{(i)} - (a_s \otimes I) x_{(i-1)}, \quad i = 2, \ldots, p,$$

$$g^{(i)} = 0,$$

where all the vectors $\tilde{x}^{(i)}, x^{(i)}, \ldots$, are defined accordingly to those of the previous section.

**Theorem 4.1.** Suppose that the system (15) has the initial conditions $x_0 = \eta$, $f$ and $g$ are sufficiently smooth and $(\partial/\partial z) g(t, x, z)$ is invertible and bounded in a neighbourhood of the solution. Then the solution of (16) is convergent with order $k$. 
Proof. From the invertibility of \((\partial/\partial z)g(t,x,z)\), the second equation of (15) possesses a locally unique solution \(z = G(t,x)\), which inserted into the first gives an ordinary differential system
\[
x' = f(t, x, G(t,x)). \tag{17}
\]
From the equations in (16) involving the algebraic variables we deduce that the numerical solution of the differential equation in (15) is equivalent to that obtained by applying the \(k\)-step block GBDF directly to (17). Then, from the previous Theorem 3.1 we have
\[
\|x_i - x(t_i)\| = O(h^k)
\]
and from a Lipschitz condition for \(G\),
\[
\|z_i - z(t_i)\| = \|G(t_i, x_i) - G(t_i, x(t_i))\| = O(h^k). \quad \square
\]

5. Numerical examples

In this section we present the results of some numerical experiments on linear and nonlinear DAE BVPs and IVPs. The numerical experiments are performed using block GBDFs of order 4-6-8, with the size of each block equal to the order of the considered method. This means that a larger number of blocks corresponds to use a smaller stepsize. In all the examples we only give the conditions needed in theory to obtain a unique solution, and use (12) with \(\gamma = 2\|x_0^{(0)}\|\). The nonlinear equations are solved by means of the Newton scheme. The iteration is terminated when the norm of the difference between two successive iterates is smaller than a specified tolerance. All the computations are performed in MATLAB. In all the examples, a uniform stepsize \(h\) is used and the relative error is computed as \(\max_{0 \leq i < N} \|y_i - y(t_i)/(1 + |y(t_i)|)\|_{\infty}\), where \(y_i\) and \(y(t_i)\) denote the numerical and the exact solution at time \(t_i\), \(N + 1\) being the number of discretization points used, and \(\div\) is the MATLAB componentwise division.

The first test problem is the following linear variable-coefficient index one DAE BVP [11, 18] on \([0, 1]\)
\[
\begin{pmatrix}
1 & -t & t^2 \\
0 & 1 & -t \\
0 & 0 & 0
\end{pmatrix} y' + \begin{pmatrix}
1 & -(t + 1) & (t^2 + 2t) \\
0 & -1 & t - 1 \\
0 & 0 & 1
\end{pmatrix} y = \begin{pmatrix}
0 \\
0 \\
\sin(t)
\end{pmatrix}
\]
with boundary conditions \(y_1(0) = 1\) and \(y_2(1) - y_3(1) = e\). The exact solution is
\[
y_1 = e^{-t} + te^t, \quad y_2 = e^t + t \sin(t), \quad y_3 = \sin(t).
\]
This problem was constructed in [11] to show the effect, if any, of spatially dependent coupling between the differential and algebraic parts of the system on the solution accuracy. The relative errors and the computed rates are reported in Table 1. We can see that the above-stated coupling has no effect on the solution and the global error behaves consistently as \(O(h^k)\). In the last entry of the table (8 blocks for the 8-order method) we do not obtain a significative improvement since the error obtained with the same method and 4 blocks is near the machine precision.
Table 1
Example 1, relative errors for GBDFs of order 4, 6, 8

<table>
<thead>
<tr>
<th>No. of blocks</th>
<th>Order 4</th>
<th>Order 6</th>
<th>Order 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error</td>
<td>Rate</td>
<td>Error</td>
</tr>
<tr>
<td>1</td>
<td>8.4e - 4</td>
<td></td>
<td>3.1e - 6</td>
</tr>
<tr>
<td>2</td>
<td>3.4e - 5</td>
<td>4.6</td>
<td>6.0e - 8</td>
</tr>
<tr>
<td>4</td>
<td>1.7e - 6</td>
<td>4.3</td>
<td>7.8e - 10</td>
</tr>
<tr>
<td>8</td>
<td>9.4e - 8</td>
<td>4.2</td>
<td>1.1e - 11</td>
</tr>
</tbody>
</table>

The second example is a nonlinear index-2 DAE BVP

\[
y_1' = y_3 - z_2 y_1, \quad y_2' = y_4 - z_2 y_2, \quad y_3' = -z_1 y_1 + e^t (1 + \sin t),
\]

\[
y_4' = -z_1 y_2 + \frac{1}{1 + t} \left( \frac{2}{(1 + t)^2} + \sin t \right),
\]

\[
y_1 y_2^3 + e^{y_2} = \frac{e^t}{(1 + t)^3} + e^{1/(1+t)},
\]

\[
y_3 y_2^3 + (3y_1 y_2^2 + e^{y_2}) y_4 = \frac{e^t}{(1 + t)^3} - \frac{3e^t}{(1 + t)^4} - \frac{e^{1/(1+t)}}{(1 + t)^2},
\]

which is given with the boundary conditions \( y_1(0) = 1, \ y_2(0) = 1 \) and \( y_1(1) = e \). The exact solution is

\[
y_1 = y_3 = e^t, \quad y_2 = (1 + t)^{-1}, \quad y_4 = -(1 + t)^{-2}, \quad z_1 = \sin t, \quad z_2 = 0.
\]

This example was derived in [4] by differentiating the constraint in the index-3 DAE

\[
y_1'' = -z_1 y_1 + e^t (1 + \sin t),
\]

\[
y_2'' = -z_1 y_2 + \frac{1}{1 + t} \left( \frac{2}{(1 + t)^2} + \sin t \right),
\]

\[0 = y_1 y_2^3 + e^{y_2} - \frac{e^t}{(1 + t)^3} - e^{1/(1+t)}.\]

Relative errors at the meshpoints and the corresponding convergence rates are listed in Table 2 for the index-2 problem and in Table 3 for the index-3 one. In Table 2 the rates approach the order of the methods when we consider a larger number of blocks. In Table 3 each rate is one order less than the corresponding in Table 2, as expected from (10) for linear DAEs of index higher than 2.

The third example is the pendulum problem on \([0, 0.55]\)

\[
y_1'' = -z y_1, \quad y_2'' = -z y_2 - g, \quad 0 = y_1^2 + y_2^2 - L^2,
\]

that can be converted in an index-2 DAE by one constraint differentiation:

\[
y_1' = y_3, \quad y_2' = y_4, \quad y_3' = -z y_1, \quad y_4' = -z y_2 - g, \quad 0 = y_1 y_3 + y_2 y_4.
\]
Table 2
Example 2, index-2 formulation, relative errors for GBDFs of order 4, 6, 8

<table>
<thead>
<tr>
<th>No. of blocks</th>
<th>Order 4</th>
<th>Order 6</th>
<th>Order 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error</td>
<td>Rate</td>
<td>Error</td>
</tr>
<tr>
<td>1</td>
<td>1.3e−1</td>
<td>2.8</td>
<td>1.6e−2</td>
</tr>
<tr>
<td>2</td>
<td>2.0e−2</td>
<td>3.2</td>
<td>8.2e−4</td>
</tr>
<tr>
<td>4</td>
<td>1.8e−4</td>
<td>3.6</td>
<td>6.3e−7</td>
</tr>
</tbody>
</table>

Table 3
Example 2, index-3 formulation, relative errors for GBDFs of order 4, 6, 8

<table>
<thead>
<tr>
<th>No. of blocks</th>
<th>Order 4</th>
<th>Order 6</th>
<th>Order 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error</td>
<td>Rate</td>
<td>Error</td>
</tr>
<tr>
<td>1</td>
<td>2.0e−1</td>
<td>1.8</td>
<td>2.4e−2</td>
</tr>
<tr>
<td>2</td>
<td>5.7e−2</td>
<td>2.3</td>
<td>1.6e−3</td>
</tr>
<tr>
<td>4</td>
<td>1.3e−2</td>
<td>2.6</td>
<td>7.7e−6</td>
</tr>
</tbody>
</table>

Table 4
Example 3, index-2 formulation, relative errors for GBDFs of order 4, 6, 8

<table>
<thead>
<tr>
<th>No. of blocks</th>
<th>Order 4</th>
<th>Order 6</th>
<th>Order 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>1</td>
<td>2.6e−2</td>
<td>1.1e−2</td>
<td>4.3e−3</td>
</tr>
<tr>
<td>2</td>
<td>1.1e−2</td>
<td>5.2e−4</td>
<td>1.9e−5</td>
</tr>
<tr>
<td>4</td>
<td>4.8e−4</td>
<td>3.2e−6</td>
<td>2.5e−8</td>
</tr>
<tr>
<td>8</td>
<td>2.3e−5</td>
<td>2.3e−8</td>
<td>7.2e−9</td>
</tr>
</tbody>
</table>

The initial conditions satisfy \( y_4(0)=0, \ y_1(0.55)=0 \). Moreover, to guarantee the equivalence with the index-3 formulation, in [19] it is suggested to add the condition \( 0 = y_1(0)^2 + y_2(0)^2 - L^2 \) to the index-2 formulation. We consider the problem with \( L=1 \) and \( g=10 \). In Tables 4 and 5 we compare the numerical solution in \( y_1(0) \) and \( y_3(0) \) with the exact one in [19] for both the formulations.

An interesting initial value problem is the following:

\[
y'_1 = (\lambda - \frac{1}{2-t}) y_1 + (2-t)\lambda y_3 + \frac{3-t}{2-t} e^t,
\]

\[
y'_2 = \frac{\lambda - 1}{2-t} y_1 - y_2 + (\lambda - 1) y_3 + 2e^t,
\]

\[
0 = (t+2) y_1 + (t^2-4) y_2 - (t^2 + t - 2) e^t,
\]
Table 5
Example 3, index-3 formulation, relative errors for GBDFs of order 4, 6, 8

<table>
<thead>
<tr>
<th>No. of blocks</th>
<th>Order 4 Error</th>
<th>Order 6 Error</th>
<th>Order 8 Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1e-1</td>
<td>9.0e-2</td>
<td>1.6e-2</td>
</tr>
<tr>
<td>2</td>
<td>2.9e-2</td>
<td>8.7e-4</td>
<td>3.2e-5</td>
</tr>
<tr>
<td>4</td>
<td>1.2e-3</td>
<td>4.3e-6</td>
<td>9.5e-8</td>
</tr>
<tr>
<td>8</td>
<td>3.5e-5</td>
<td>2.7e-10</td>
<td>6.8e-9</td>
</tr>
</tbody>
</table>

Table 6
Example 4, relative errors for GBDFs of order 4, 6, 8

<table>
<thead>
<tr>
<th>λ</th>
<th>No. of blocks</th>
<th>Order 4 Error</th>
<th>Order 6 Error</th>
<th>Order 8 Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>1</td>
<td>1.4e-4</td>
<td>4.5e-7</td>
<td>8.2e-10</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.1e-5</td>
<td>5.5e-9</td>
<td>2.4e-12</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.0e-6</td>
<td>3.0e-10</td>
<td>4.1e-13</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>1.5e-4</td>
<td>5.3e-7</td>
<td>1.0e-9</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.3e-5</td>
<td>7.3e-9</td>
<td>2.8e-12</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>8.4e-7</td>
<td>9.1e-11</td>
<td>1.6e-13</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td>1.6e-4</td>
<td>6.1e-7</td>
<td>1.3e-9</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.5e-5</td>
<td>1.0e-8</td>
<td>5.0e-12</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.2e-6</td>
<td>1.6e-10</td>
<td>1.7e-12</td>
</tr>
</tbody>
</table>

where $y_1(0) = 1$, $0 \leq t \leq 1$ and $\lambda \geq 1$ is a parameter. This example is analyzed in [4, 5] and results to be particularly nasty as $\lambda$ increases. For this problem the unprojected Gauss schemes are unstable, whereas the projected schemes presented in [4, 5] compute the numerical solution as expected from their order. The exact solution is $y_1 = y_2 = e^t$ and $y_3 = -e^t/(2 - t)$. We solve this problem for $\lambda = 50, 100, 1000$. The obtained relative errors are reported in Table 6. We note that GBDFs do not suffer instability and exhibit the same behaviour for increasing values of $\lambda$.

6. Conclusions

Boundary value methods have been applied to the solution of DAEs. A technique has also been considered to avoid the computation of consistent initial/boundary conditions. Some numerical examples have been presented to show that the approach is promising and the research is worth to continue in this direction. Further studies are still needed to state theoretically that BVMs (or, in particular, GBDFs) can be successfully applied to DAEs of higher index. The final point, we hope in a near future, will be that of the generalization of a code currently in progress for the solution
of ODEs, which could handle both initial and boundary value DAE problems, with the further advantage of having an efficient parallel implementation.

Acknowledgements

The authors thank the anonymous referee for helpful comments which greatly improved the paper.

References


