SEQUENTIAL REGULARIZATION METHODS FOR HIGHER INDEX DAES
WITH CONSTRAINT SINGULARITIES: THE LINEAR INDEX-2 CASE

URI M. ASCHER* AND PING LIN†

Abstract. Standard stabilization techniques for higher index DAES often involve elimination of the algebraic solution components. This may not work well if there are singularity points where the constraints Jacobian matrix becomes rank-deficient. This paper proposes instead a sequential regularization method (SRM) — a functional iteration procedure for solving problems with isolated singularities which have smooth differential solution components.

For linear index-2 DAES we consider both initial and boundary value problems. The convergence of the SRM is described and proved in detail. We believe that this is the first convergence proof for any method for DAES with this type of constraint singularities. Moreover, the regularization parameter in our method is not necessarily very small, so the SRM is an important improvement over usual regularization methods. Various aspects of the subsequent numerical discretization of the regularized problems are discussed as well and some numerical verifications are carried out.

Keywords. Differential-algebraic equations, constraint singularities, sequential regularization, higher index

AMS (MOS) subject classifications. 65L10, 65L20

1. Introduction. Recently more and more attention has been paid to the study of differential-algebraic equations (DAEs). Such equations arise in a variety of applications and can be difficult to solve when they have a higher index (index greater than one; cf. [9]). Higher index DAES are ill-posed in a certain sense, especially when the index is greater than two [6], and a straightforward discretization generally does not work well. An alternative treatment is the use of index reduction methods, whose essence is the repeated differentiation of the constraint equations until a well-posed problem (index-1 DAES or ODEs) is obtained. But repeated index reduction by direct differentiation leads to instability for numerical integrations (i.e. drift-off — the error in the original constraint grows). Hence, stabilized index reduction methods were proposed to overcome the difficulty. A popular stabilization technique was introduced first in the computation of constrained multibody systems by Baumgarte [7]. See [3] for corresponding improvements and additional techniques. However, for problems with constraint singularities (e.g. where some rows of the constraint Jacobian matrix become linearly dependent at some points), Bayo and Avello [8] indicate that Baumgarte’s technique may not work. Another approach is the so-called regularization of DAES where a small perturbation term (measured by a small positive parameter $\epsilon$) is added to the original DAE (see, e.g., [11, 14, 13, 12, 18]). The regularized problem usually is a singular perturbation problem and the DAE becomes the reduced problem of the singular perturbation problem. In the present paper, we propose a new regularization method which we call sequential regularization method (SRM). A motivation for this method comes from current research [8, 17] on simulation of the dynamics of constrained mechanical systems. We will show that our method works well for initial and boundary value problems with smooth solutions, with and without the presence of constraint singularities. This range covers, in particular, problems with a highly ill-conditioned constraint Jacobian.

A singularity in the constraints (or in the algebraic solution components) of a DAE may cause various phenomena to occur, including impasse points [19] and bifurcations. In this paper, however, we assume that the solution sought is smooth in the passage through isolated singularity points, and concentrate on the linear index-2 case.

We introduce at this point a simple mechanical system as a motivating example:

* Institute of Applied Mathematics and Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada V6T 1Z4 (asher@cs.ubc.ca). The work of this author was partially supported under NSERC Canada Grant OGP0004306.

† Institute of Applied Mathematics, University of British Columbia, Vancouver, B.C., Canada V6T 1Z2 (lin@math.ubc.ca). The work of this author was partially supported under a Killam postgraduate grant.
Example 0: two-link slider crank \[8\]
Consider two linked bars of length 2 and mass 1 each (see Fig. 1.1). One end of one bar is fixed at the origin, allowing only rotational motion in the plane. The other end of the other bar slides on the $x$-axis. The equations of motion form a nonlinear index-3 DAE

\[
p' = v
\]

\[
Mv' = f - G^T \lambda
\]

\[
g(p) = 0
\]

where $x_i, y_i, \phi_i$ are the coordinates of the centre of mass of the $i$th bar,

\[
p = (x_1, y_1, \phi_1, x_2, y_2, \phi_2)^T
\]

\[
M = \text{diag}\{1, 1, 1/3, 1, 1, 1/3\}
\]

\[
f = (0, -9.81, 0, 0, -9.81, 0)^T
\]

\[
g = \begin{pmatrix}
x_1 - \cos \phi_1 \\
y_1 - \sin \phi_1 \\
x_2 - 2x_1 - \cos \phi_2 \\
y_2 - 2y_1 - \sin \phi_2 \\
2 \sin \phi_1 + 2 \sin \phi_2
\end{pmatrix}
\]

\[
G = g_p = \begin{pmatrix}
1 & 0 & \sin \phi_1 & 0 & 0 & 0 \\
0 & 1 & -\cos \phi_1 & 0 & 0 & 0 \\
-2 & 0 & 1 & 0 & \sin \phi_2 \\
0 & -2 & 0 & 0 & 1 & -\cos \phi_2 \\
0 & 0 & 2 \cos \phi_1 & 0 & 0 & 2 \cos \phi_2
\end{pmatrix}
\]
Clearly, as the mechanism moves left through the point where both bars are upright (\( \phi_1 = \frac{\pi}{2}, \phi_2 = \frac{3\pi}{2} \)) the last row of \( G \) vanishes at this one point and a singularity is obtained. When arriving at this point with no momentum, this is actually a bifurcation point where two subsequent motion configurations are possible. We consider only the case where the sliding bar continues to slide along the \( x \)-axis past the singularity, and note that the solution is smooth in the passage through the singularity.

When attempting to integrate this system using methods like Baumgarte's or a stabilization like [3] which ignore the singularity, the results are unpredictable, depending on how close to the singular time point the integration process gets when attempting to cross it. In fact, radically different results may be obtained upon changing the value of an error tolerance. (Similar observations are made in [18].) In some instances a general purpose ODE code would simply be unable to "penetrate the singularity", and yield a solution which, after hovering around the upright (singular) position for a while, turns back towards the initial position depicted in Fig. 1.1 in solid line. Such a motion pattern may well look deceptively plausible.

We therefore feel that it is important to derive methods which can be proved to work reliably as well as efficiently. The SRM is such a method. We restrict ourselves in subsequent sections to the linear index-2 case and provide what we believe to be the first proof for a working method for DAEs with constraint singularities in this context. \( \square \)

The sequential regularization method is actually a functional iteration procedure in which the difference between the exact solution of a DAE and the corresponding iterate becomes \( O(\epsilon^r) \) in magnitude at the \( r \)th iteration, at least away from the starting value of the independent variable (which we shall call 'time'). Hence, unlike usual regularization, the regularization parameter \( \epsilon \) does not have to be chosen very small, so the regularized problems can be less stiff and/or more stable.

As in [6], we consider a linear (or linearized) model problem

\[
\begin{align*}
x^{(r)} &= \sum_{j=1}^{\nu} A_j x^{(j-1)} + B y + q, \\
0 &= C x + r,
\end{align*}
\]

where \( A_j \), \( B \) and \( C \) are sufficiently smooth functions of \( t, 0 \leq t \leq t_f, A_j(t) \in \mathbb{R}^{n_x \times n_x}, j = 1, \ldots, \nu, B(t) \in \mathbb{R}^{n_x \times n_y}, C(t) \in \mathbb{R}^{n_y \times n_x}, n_y \leq n_x \) and \( CB \) is nonsingular (the DAE has index \( \nu + 1 \)) except possibly at a finite number of isolated points of \( t \). For simplicity of exposition, let us say that there is one singularity point \( t_s, 0 < t_s < t_f \). The inhomogeneities are \( q(t) \in \mathbb{R}^{n_x} \) and \( r(t) \in \mathbb{R}^{n_y} \). We note that a linearized form of the Euler-Lagrange equations which govern the motion of mechanical systems with holonomic constraints is in the form (1.1) with \( \nu = 2 \).

Returning to Example 0, it can be verified by a simple calculation that, although the matrix \( GM^{-1}G^T \) is singular at the singularity point, the matrix \( M^{-1}G^T (G M^{-1}G^T)^{-1} G \) is smooth for all \( t \). Also, two types of singular constraints (i.e. with vanishing rows or with some rows linearly dependent at some points) mentioned in [1] both have a similar property. Thus, for the linear model (1.1), we assume accordingly:

\[ \textbf{H1}: \text{the matrix function } P = B(CB)^{-1}C \text{ is smooth, or more precisely, } P \text{ is continuous and } P' \text{ is bounded near the singular point } t_s, \text{ where we define}
\]

\[ P(t_s) = \lim_{t \to t_s} (B(CB)^{-1}C)(t). \]

Because we are only interested in the case where (1.1) has a smooth solution for \( x \) (as is the case in Example 0), it is necessary to assume, in view of (1.1b):
**H2:** The inhomogeneity $r(t)$ satisfies $r \in \mathcal{S}$, where

$$\mathcal{S} = \{ u(t) \in \mathbb{R}^{n_x} : \text{there exists a smooth function } z(t) \text{ s.t. } Cz = u \}.$$ 

We note that $H1$ and $H2$ are satisfied automatically if $CB$ is nonsingular for each $t$. On the other hand, neither $B(CB)^{-1}$ nor $(CB)^{-1}C$ alone are smooth near a singularity in general. We also indicate here that to formulate the SRM (see §3) we only need the continuity of $P$. The further requirement in $H1$ on the derivative of $P$ is needed for the regularity of the solution (cf. Lemma 2.1 and (2.9)) and the convergence proof of the method (cf. (3.12)).

In this paper, we concentrate on the index-2 case, i.e. (1.1) with $\nu = 1$. For higher index cases we assume that a stable reduction to index-2 has been performed. We consider both initial and boundary value problems. In §2, we briefly discuss the conditioning of the problem with singularities. In §3, the main section of this paper, we describe the sequential regularization method and estimate its error. In §4, we consider some discretization and implementation issues for both initial and boundary value problems. Finally, several numerical examples demonstrate our theoretical results.

This paper opens up a few avenues for extension which we plan to undertake. These include nonlinear and higher index problems and their numerical treatment.

2. **Problem Conditioning and Baumgarte stabilization.** We rewrite (1.1) with $\nu = 1$ as

\begin{align}
(2.1a) & \quad x' = Ax + By + q, \\
(2.1b) & \quad 0 = Cx + r, 
\end{align}

and consider this DAE subject to $n_x - n_y$ boundary conditions

\begin{equation}
B_0 x(0) + B_1 x(t_f) = \beta.
\end{equation}

These boundary conditions are assumed to be such that they yield a unique solution for the ODE (2.1a) on the manifold given by (2.1b). In particular, assuming for a moment that no singularities occur, if we were to replace (2.1b) by its differentiated form

\begin{align}
(2.3a) & \quad 0 = Cx' + C'x + r', \\
(2.3b) & \quad C(0)x(0) + r(0) = 0,
\end{align}

and use (2.3a) in (2.1a) to eliminate $y$ and obtain $n_x$ ODEs for $x$, then the boundary value problem for $x$ with (2.2) and (2.3b) specified has a unique solution. In the initial value case $B_1 = 0$, this means that (2.2) and (2.3b) can be solved uniquely for $x(0)$. We will give a more precise assumption in Lemma 2.1 below.

Similarly to [5] and to the method of pseudo upper triangular decomposition (PUTD) described in [1], (§10.6; a difference is that we do pivoting to interchange the row with singularity of lowest order to the current row when all the other rows vanish at some singular point), there exists a smooth matrix function $R(t) \in \mathbb{R}^{[n_x-n_y] \times n_x}$, which has full row rank and satisfies

\begin{equation}
R^0 = 0, \quad \text{for each } t, \quad 0 \leq t \leq t_f,
\end{equation}

where $R$ can be taken to have orthonormal rows.

As in [5, 6], define the new variable

\begin{equation}
u = R x, \quad 0 \leq t \leq t_f.
\end{equation}

Then, using (2.1b), the inverse transformation is given by

\begin{equation}
x = S u - B(CB)^{-1} r,
\end{equation}
where

\[ S = (I - B(CB)^{-1}C)R^T = (I - P)R^T. \]

By the assumptions in §1, this transformation is well-defined. Differentiating (2.4) and using (2.1a) and (2.5) we obtain the essential underlying ODE (EUODE):

\[ u' = (RA + R^c)Su - (RA + R^c)B(CB)^{-1}r + Rq. \]

Hence the underlying problem of (2.6) is

\begin{align*}
(2.7a) & \quad u' = (RA + R^c)Su + f, \\
(2.7b) & \quad B_0S(0)u(0) + B_1S(t_f)u(t_f) = \beta_1.
\end{align*}

We assume (cf. [5])

**H3:** The boundary value problem (2.7) is stable, i.e. there exists a moderate-size constant \( K \) such that

\[ ||u|| \leq K(||f|| + ||\beta_1||), \]

where \( ||u|| = \max_t \{|u(t)|, \ 0 \leq t \leq t_f\} \).

Similarly to Theorem 2.2 of [5], we can get

**Lemma 2.1.** Let the DAE (2.1) have smooth coefficients, and assume that H1 and H2 hold. If the EUODE (2.6) with the boundary conditions (2.7b) has a unique solution, then there exists a unique solution for \( x \) of problem (2.1)-(2.2) which is smooth. This implies a unique existence of a smooth \( By \) as well. Furthermore, if H3 holds then there is a constant \( K \) such that

\[ ||x|| \leq K(||q|| + ||B(CB)^{-1}r|| + ||\beta||), \]

\[ ||x'|| \leq K(||q|| + ||B(CB)^{-1}r|| + ||B(CB)^{-1}r'|| + ||\beta||). \]

The difference between the situation here and in the nonsingular case is that here perturbation inhomogeneities \( r \) yield reasonably bounded perturbations in the solution \( x \) only if they are (in general) from the subspace \( \text{Range} \ (C) \).

From (2.1a), we can write

\[ y = (CB)^{-1}C(x' - Ax - q), \ t \in [0,t_s) \cup (t_s,t_f], \]

which could be unbounded at the singular point \( t_s \) (whereas \( By \) is bounded).

**Remark 2.1** If \( B \) has full rank for each \( t \), then we get

\[ (CB)^{-1}C = (B^T B)^{-1}B^T P. \]

Hence, \( (CB)^{-1}C \) is smooth. So, there exists a unique solution for \( y \) of problem (2.1)-(2.2) which is smooth and can be expressed as (2.8) for each \( t \in [0,t_f] \). Furthermore, using Lemma 2.1, we have in this case

\[ ||y|| \leq K(||q|| + ||B(CB)^{-1}r|| + ||B(CB)^{-1}r'|| + ||\beta||). \]
In the general case, however, we will have to consider $By$, rather than $y$ alone, in the theorems of the next section. \(\Box\)

A Baumgarte stabilization applied to (2.1) consists in eliminating $y$ according to (2.3),(2.8), and stabilizing. This gives the ODE

\[
x' = (I - B(CB)^{-1}C)(Ax + q) - B(CB)^{-1}(C'x + r) - \epsilon^{-1}B(CB)^{-1}(Cx + r)
\]

where $\epsilon > 0$ is a parameter (cf. [7, 3]). If there are no singularities then it follows from the analysis in [6] that if $H3$ holds then the boundary value problem (2.10),(2.2),(2.3b) is also stable. In other words, the “initial value stabilization” works also for the boundary value case, because the new modes introduced by replacing (2.1b) with a combination of (2.3a) and (2.1b) are separable and decaying, in agreement with the additional initial conditions (2.3b).

However, in the singular case (2.10) may not work because the terms $B(CB)^{-1}C$ and $B(CB)^{-1}r$ are in general unbounded. Therefore, we develop in the next section an iterative method which builds up an approximation to $By$ and $x$ that avoids going through unbounded quantities.

3. Sequential Regularization Methods. Consider the following regularization for the DAE (2.1),

\[
x' + BE(Cx + r) = \epsilon Ax + \epsilon By + cq,
\]

where $E \in \mathbb{R}^{n_y \times n_y}$ is chosen such that $BEC$ has nonnegative eigenvalues. In the sequel we choose, relying on Assumptions H1 and H2, $E = (CB)^{-1}$ (hence, $BEC = P$). Using the fact that, for two matrices $M \in \mathbb{R}^{m \times n}$ and $N \in \mathbb{R}^{n \times m}$, $MN$ and $NM$ have the same nonzero eigenvalues, we have

\[
\lambda_i(P) = 1 > 0, \quad i = 1, \cdots, n_y, \quad \lambda_i(P) = 0, \quad i = n_y + 1, \cdots, n_x.
\]

If $CB$ is symmetric positive semi-definite (as in the case of mechanical systems) then it is possible to choose $E = I$ (see, e.g. [8]). For the purpose of proving our convergence theorem 3.1, however, we consider $E = (CB)^{-1}$, because otherwise one has to deal with a turning point phenomenon in a singular perturbation context.

The ODE (3.1) is subject to the boundary conditions (2.2),(2.3b). This boundary value problem has a unique solution if $y$ is given. We denote the exact solution of (2.1)-(2.2) by $x_e$, $y_e$. Obviously, if we take $y = y_e$ in (3.1) then this problem has the same solution as (2.1)-(2.2).

Because we do not know $y_e$ in advance we propose, motivated by [8], the following iterative algorithm for solving (3.1): For $s = 1, 2, \ldots$, solve the ODE problem

\[
x'_s = Ax_s + By_s + q
\]

where

\[
y_s = y_{s-1} - \frac{1}{\epsilon}E(Cx_s + r),
\]

subject to the same boundary conditions (2.2) and (2.3b). Note that $y_0(t)$ is a given initial iterate and that $\epsilon > 0$ is the regularization parameter.

We call this algorithm a sequential regularization method (SRM). Note that $x_s(t)$ and $y_s(t)$ are defined on the entire interval $[0, t_f]$ for each $s$. Also, in practice we multiply (3.4) by $B$ and keep track only of the approximations $By_s$ to the bounded function $By$. If $y$ is desired (at times other than $t_s$) then it can be easily retrieved from $By$ in a post-processing step.

Remark 3.1 A penalty method for (2.1) in the initial value case (cf. [17, 12]) reads (using our notation):

\[
\begin{align*}
x' &= Ax + By + q; \\
y &= -\frac{1}{\epsilon}E(Cx + r),
\end{align*}
\]
where $\epsilon > 0$ must be small. In the mechanical system index-3 context this corresponds to replacing the rigid constraint by a stiff spring [15]. The regularized problem (3.5) is just like one iteration of (3.3) with $y_0(t) \equiv 0$ and $E = I$ and $B = C^T$ in [17] or $E = (CB)^{-1}$ in [12]. From (3.5b) and (2.3b), we have $y(0) = 0$. Hence, if $y_\epsilon(0) \neq 0$, the method (3.5) will not yield a good approximation for $y_\epsilon$ near $t = 0$. In the method (3.3)-(3.4) proposed here, we can choose

$$y_0(0) = y_\epsilon(0),$$

where $y_\epsilon(0)$ can be calculated from (2.1)

$$y_\epsilon(0) = -(C(0)B(0))^{-1}(C(0)(A(0))x(0) + C(0)y(0) + r'(0)).$$

\[\Box\]

**Remark 3.2** In [17] the authors also indicated that the penalty procedure (3.5) has some drawbacks, e.g., once an error is committed in computing $y$, there is no compensation scheme by which the drifting of the numerical solution can be corrected. The proposed iteration (3.3)-(3.4) could be such a compensation scheme since the error (or defect) of $Cx + r$ would decrease as the iteration proceeds according to Theorem 3.1 below. In [17] a so-called staggered stabilization procedure is introduced for initial value problems with $B = C^T$. They first differentiate (3.5b) with $E = I$ to obtain

$$\begin{align*}
y' &= -\frac{1}{\epsilon}(Cx' + Cx + r').
\end{align*}$$

Then, they obtain $Cx'$ from (3.5a) and substitute it into (3.6) to yield

$$\begin{align*}
cy' + CBy &= -(CA + C')x - Cq - r'.
\end{align*}$$

The initial value for $y$ could be the exact value. This procedure does work well for the case where $CB$ is nonsingular if we choose $\epsilon$ to be small. But if $CB$ is singular at $t_*$, or more precisely, if $B$ is rank-deficient at $t_*$, this procedure may not work well since $y$ could be unbounded as shown in (2.8). Moreover, the poor accuracy of $y$ will also affect the accuracy of $x$. A similar idea appears in [12] with $E = (CB)^{-1}$ (recently also generalized to boundary value problems in [13]). This treatment may not work well for problems with constraint singularities either. \[\Box\]

Now we estimate the error of the sequential regularization method (3.3)-(3.4). We make the following assumption on the initial iterate $y_0$:

**H4**: $y_0(0) = y_\epsilon(0)$, $y'_0(0) = y'_\epsilon(0)$, \ldots, $y^{(M)}_0(0) = y^{(M)}_\epsilon(0)$,

where $M$ is an integer. Set $M = -1$ if $y_0(0) \neq y_\epsilon(0)$ (i.e., no assumption is made).

For initial value problems we may calculate $y^{(i)}_\epsilon(0)$, $i = 0, 1, \ldots$ in advance by using the ODE and its derivatives. For boundary value problem we set $M = -1$ since we don’t know $y_\epsilon(0)$ beforehand.

Choosing $E = (CB)^{-1}$, we rewrite (3.4) as

$$\begin{align*}
By_s &= By_{s-1} - \frac{1}{\epsilon}B(CB)^{-1}(Cx_s + r),
\end{align*}$$

We give the following lemma before the main theorem of this section.

**Lemma 3.1.** Let $u, v$ be the solution of

$$\begin{align*}
u' &= (RA + R')Su + S_1v + f_1, \\
\delta v' + \gamma v &= \epsilon S_2u + \epsilon S_3v + f_2, \\
B_0S(0)u(0) + B_1S(t_f)u(t_f) &= \beta - S_4v(0) - S_5v(t_f), \\
v(0) &= v_0.
\end{align*}$$

7
where all coefficients are sufficiently smooth, \( \delta = 1 \) or \( \delta = \epsilon, \gamma \) is a positive constant and \( H3 \) holds. Then, for \( \epsilon \) appropriately small or \( \gamma \) appropriately large, we have the following stability inequality

\[
\|u\| \leq K(\|f_1\| + \|f_2\| + |\beta| + |v_0|),
\]

\[
\|v\| \leq K(\epsilon\|f_1\| + \|f_2\| + |\beta| + |v_0|),
\]

where \( K \) is a positive constant.

**Proof:** Let \( v = (v_1, \ldots, v_n)^T \). From (3.9b), we easily have

\[
|v_i| \leq \frac{\epsilon}{\gamma}\|S_2\|\|u\| + \frac{\epsilon}{\gamma}\|S_3\|\|v\| + \frac{1}{\gamma}\|f_2\| + |v_0|, \quad i = 1, \ldots, n_y
\]

Hence, taking the maximum of the left hand side for \( 1 \leq i \leq n_y \) and choosing small \( \epsilon \) or large \( \gamma \) appropriately such that \( \epsilon\|S_2\| < \gamma \), we get

\[
(3.10) \quad \|v\| \leq \frac{\epsilon}{\gamma - \epsilon\|S_2\|}\|S_2\|\|u\| + \frac{\epsilon}{\gamma - \epsilon\|S_2\|}\|S_3\|\|v\| + \frac{1}{\gamma}\|f_2\| + |v_0|.
\]

By using \( H3 \), from (3.9a), there exists a positive constant \( K_1 \) such that

\[
\|u\| \leq K_1\((\|S_1\|\|v\| + \|f_1\| + |\beta| + |S_5|v(0) + |S_6|v(t_f))
\]

\[
\leq K_1\((\|S_1\| + |S_5|)|v| + \|f_1\| + |\beta| + |S_4|v_0|)
\]

\[
\leq \frac{K_1\|S_1\| + \|S_5\|\|S_2\|}{\gamma - \epsilon\|S_2\|}\|u\| + \frac{K_1\|S_1\| + \|S_5\|\|f_2\| + \gamma|v_0|}{\gamma - \epsilon\|S_2\|} + K_1\|f_1\| + |\beta| + |S_4|v_0|.
\]

Hence, by choosing smaller \( \epsilon \) or larger \( \gamma \) such that \( \frac{K_1\|S_1\| + \|S_5\|\|S_2\|}{\gamma - \epsilon\|S_2\|} < 1 \), the stability inequality for \( u \) follows. Now the stability inequality for \( v \) follows from that for \( u \) and (3.10). \( \Box \)

We are now ready to state the main theorem of this paper. It says that the error after \( s \) SRM iterations is \( O(\epsilon^s) \) (i.e., each iteration improves the error by \( O(\epsilon) \)) everywhere in \( t \), except possibly at an exponentially thin initial boundary layer. The magnitude of the error in the thin layer depends on how well the initial values of \( y_0 \) fit the exact solution (more on this in §4).

**Theorem 3.1.** Let the DAE (2.1) have sufficiently smooth coefficients, and assume that \( H1, H2, H3 \) and \( HA \) hold. Then, for the solution of iteration (3.3), (3.8), we have the following error estimates:

\[
x_s(t) - x_c(t) = O(\epsilon^s) + O(\epsilon^{M+2} p_s(t/\epsilon)e^{-t/\epsilon}),
\]

\[
B y_s(t) - B y_c(t) = O(\epsilon^s) + O(\epsilon^{M+1} p_s(t/\epsilon)e^{-t/\epsilon}),
\]

for \( 0 \leq t \leq t_f \) and \( s \geq 1 \). Here \( p_s(\tau) \equiv 0 \) if \( s \leq M + 1 \); otherwise \( p_s(\tau) \) is a polynomial of degree \( s - M - 2 \) with generic positive coefficients and \( |p_s(0)| = |(B y_0)^{M+1}(0) - (B y_c)^{M+1}(0)| \).

**Proof:** Let \( u_s = R x_s \) and \( w_s = P x_s \). Similarly to (2.5), we have

\[
x_s = S u_s + w_s.
\]

Furthermore, using (3.3) we obtain

\[
u_s = (RA + R')S u_s + (RA + R')w_s + Rq,
\]

\[
cu_s + w_s = c(P A + P')S u_s + c(P A + P')w_s + cB y_{s-1}
\]

\[
+ cP q - B(CB)^{-1} r,
\]

\[8\]
subject to

\begin{align}
B_0 S(0) u_s(0) + B_1 S(t_f) u_s(t_f) &= \beta - B_0 w_s(0) - B_1 w_s(t_f), \\
w_s(0) &= -B(0)(C(0) B(0))^{-1} r(0).
\end{align}

The iteration (3.8) for \( B y \) becomes

\begin{equation}
B y_s = B y_{s-1} - \frac{1}{\epsilon} (w_s + B(CB)^{-1} r).
\end{equation}

The proof proceeds along familiar lines of singular perturbation analysis. According to [20, 21, 16] we can construct the asymptotic expansion of \( w_s \) and \( u_s \) sequentially for \( s = 1, 2, \ldots \), where we use Lemma 3.1 to estimate the remainders. Then, using (3.14) and (3.11), we get the asymptotic expansion of \( B y_s \) and \( x_s \) respectively. Note that in these expansions the first terms are exactly \( x_e \) and \( B y_e \). This process eventually yields the proof of the theorem. \( \square \)

To provide a better feeling about the sequential regularization method we give in the appendix a detailed proof for the initial value case with \( s \leq M + 1 \). In that proof, the construction of the asymptotic expansion is directly for \( x \) and \( B y \). Moreover, the construction method we apply is somewhat different from [20, 21] and more relevant to the concept of DAEs.

Next, we consider the following sequential regularization: The same ODE (3.3) is solved successively under the same boundary conditions as before, but with the update

\begin{equation}
y_s = y_{s-1} - \frac{1}{\epsilon} E(\alpha_1(Cx_s + r)^j + \alpha_2(Cx_s + r)).
\end{equation}

When considering the initial value problem with \( E = I \) and \( B = C^T \), this corresponds to Algorithm ALF1 of [8] for constrained mechanical systems (although they do it for the corresponding index-3 case) derived by a penalty-augmented Lagrangian formulation. See also the earlier reference [15]. Bayo and Avello indicate that in multibody dynamics simulation they have used Algorithm ALF1 very successfully and that it has turned out to be very efficient and accurate. However, they have also noted that under repetitive singular conditions this algorithm may lead to unstable behaviour. For our index-2 case (2.1), it appears to be impossible to choose a matrix \( E \) such that problem (3.3), (3.15) is always stable, even if we assume \( B = C^T \). A numerical example in §5 will verify such instability phenomena even for the case of one singular point. However, for the case where constraints are without singularities, (3.15) is preferable to (3.4). In particular, (3.15) yields an ODE problem for \( x_s \) which is essentially not a singular perturbation problem. Take \( E = (CB)^{-1} \) as before and rewrite (3.15) as

\begin{equation}
B y_s = B y_{s-1} - \frac{1}{\epsilon} B E(\alpha_1(Cx_s + r)^j + \alpha_2(Cx_s + r)).
\end{equation}

Then we give the following error estimation for (3.3), (3.16):

**Theorem 3.2.** Let the DAE (2.1) have sufficiently smooth coefficients, and assume that \( C \) has full rank and that \( H1, H2 \) and \( H3 \) hold. Then for the solution of the iteration procedure (3.3), (3.16) with \( \alpha_1 \neq 0 \), we have the following error estimations:

\begin{align}
&x_s - x_e = O(\epsilon^i), \\
&B y_s - B y_e = O(\epsilon^i)
\end{align}

for \( 0 \leq t \leq t_f \) and \( s = 1, 2, \ldots \). Note that no boundary layer terms appear here even for \( M = -1 \) in \( H4 \) (i.e. no assumption on \( y_0(0) \) is needed).
\textbf{Proof:} Denote \( u_s = R x_s \) and \( v_s = C x_s \). Hence
\begin{equation}
x_s = S u_s + F v_s,
\end{equation}
where \( S = (I - P) R^T \) and \( F = B(CB)^{-1} = PC^T(C^T)^{-1} \) are both sufficiently smooth. From (3.3),(3.16), we get
\[ u_s' = (RA + R')Su_s + (RA + R')Fv_s + Rf, \]
\[ (\epsilon + \alpha_1)u_s' + \alpha_2 v_s = \epsilon(C^T + CA)u_s + \epsilon(C^T + CA)Fv_s + \epsilon CB y_{s-1} + \epsilon C q - \alpha_1 \epsilon' - \alpha_2 
\]
with the corresponding boundary conditions, and
\[ B y_s = B y_{s-1} - \frac{1}{\epsilon} B(CB)^{-1}(\alpha_1(v_s + r) + \alpha_2 (v_s + r)). \]
Repeating the procedure of the proof of Theorem 3.1 and using Lemma 3.1 again to estimate the remainder of the asymptotic expansion, we obtain
\[ u_s - u_e = O(\epsilon^s), \]
\[ v_s - v_e = O(\epsilon^s), \]
\[ B y_s - B y_e = O(\epsilon^s), \]
where \( u_e = R x_e, \ v_e = C x_e = -r \). Hence, using (3.17) and \( x_e = S u_e + F v_e \), we obtain
\[ x_s - x_e = S(u_s - u_e) + F(v_s - v_e) = O(\epsilon^s). \]
\[ \square \]

4. Discretization and implementation issues. The SRM iteration (3.3),(3.4) (or (3.3),(3.8)) yields a sequence of ODE problems which are to be solved numerically. Multiplying (3.4) by \( B \) and inserting into (3.3), the ODE problem to be solved at the \( s \)th iteration is written as the singularly-perturbed problem (see [21, 16])
\begin{align}
\epsilon x_s' &= -B E C x_s + \epsilon A x_s + f_s \\
B_0 x_s(0) + B_1 x_s(t_f) &= \beta, \ C(0)x_s(0) + r(0) = 0
\end{align}
where \( f_s(t) \) is a known inhomogeneity,
\[ f_s = -B E r + \epsilon(B y_{s-1} + q) \]
We consider a finite difference (or collocation) discretization of (4.1) on a mesh
\[ \pi : 0 = t_0 < t_1 < \ldots < t_N = t_f \]
\[ h_i = t_i - t_{i-1}, \quad h = \max_{1 \leq i \leq N} h_i. \]
and denote by \( x_s^j, y_s^j \) the corresponding approximations of \( x_s(t_i), y_s(t_i) \), resp. We now have essentially two small, positive parameters to choose: \( \epsilon \) and \( h \). We assume that \( h \) is chosen small enough so that the underlying numerical discretization applied to the (stable) EUODE problem (2.7) is stable. (In particular if the underlying discretization is explicit this may be a severe restriction when (2.7) is stiff.) We also assume that the problem coefficients are sufficiently smooth. For the remaining treatment we consider initial and boundary value problems separately.

4.1 Boundary value problem (BVP)
In the BVP case the situation is the familiar one, much like the iterative solution of a nonlinear boundary value ODE using quasilinearization (see, e.g., [4]). Each of the linear boundary value ODEs (4.1) is discretized on a mesh \( \pi \) using, say, a symmetric finite difference scheme or some other method. If \( h \ll \epsilon \) then we expect, as \( h \to 0 \), convergence to the solution of (4.1) and our theory then applies for the entire numerical algorithm.

For instance, suppose we use collocation at \( k \) Gaussian points per mesh interval (\( k = 1 \) yields the midpoint difference scheme). Denote the collocation points by

\[ t_i : t_{i-1} < t_1 < \ldots < t_{ik} < t_i \]

and require that (3.3), (3.8) hold for the collocation solution at the collocation points. With \( B y_i^0 = B y_0(t_{ij}) \) and \( x_i^j, y_i^j \) denoting the collocation approximations of \( x_i(t_{ij}), y_i(t_{ij}) \) resp., we let

\[ B y_i^j = B y_{i-1}(t_{ij}) - \epsilon^{-1} B(t_{ij}) E(t_{ij})(C(t_{ij}) x_i^j + r(t_{ij})) \]

and obtain the collocation approximations of (4.1), (4.2). The usual theory then yields

\[ x_i^j - x_i(t_i) = O(h^{2k}) \]

Further observing that (3.3) is satisfied at the collocation points we obtain

\[ B y_i^j - B y_i(t_{ij}) = O(h^{k+1}) \]

(see, e.g., [4]). Combining with Theorem 3.1 this yields the error estimates

\[ x_i^j - x_i(t_i) = O(h^{2k} + \epsilon^k) \]

\[ B y_i^j - B y_i(t_{ij}) = O(h^{k+1} + \epsilon^k), \quad j = 1, \ldots, k \]

away from the initial layer, i.e., for \( t_i \gg \epsilon \).

A difficulty with the BVP case is that in general we do not know \( y_i(0) \), hence we expect \( y_0(0) \neq y_i(0) \), i.e., \( M = -1 \) in \( H4 \) and in Theorem 3.1. The two error estimates (4.4) at \( t_i = O(\epsilon) \) have to be appended, therefore, by the terms \( O(\epsilon^{-2t/\epsilon}) \) and \( O(\epsilon^{-2t/\epsilon}) \), respectively. This suggests that \( \epsilon \) should not be taken too large: we must require

\[ \epsilon \ll \min(1, t_f). \]

Once an accurate SRM solution, \( \{x_i^j, B(t_{ij}) y_i^j\} \) say, for \( x_i \) has been determined outside the initial layer, though, it may be possible to obtain an accurate solution everywhere by applying a few SRM iterations numerically solving (4.1a) (changing \(-BEC\) to \(BEC\)) subject to the terminal value

\[ x(t_f) = x_N^*, \]

and choosing \( B y_0 \) satisfying \( B(t_f) y_0(t_f) = B(t_f) y_N^* \). This procedure is feasible provided that the terminal value problem (4.1a), (4.5) is well-conditioned (which holds if the terminal value problem for the EUODE (2.7a) is well-conditioned; however, this is an additional condition to those of Lemma 2.1).

If the condition \( h \ll \epsilon \) is deemed too restrictive then it is possible instead to consider taking \( \epsilon \ll h_i \) for \( h_i \) away from the initial layer (i.e., for all \( i \) s.t. \( t_{i-1} \gg \epsilon \)). This situation has been analyzed for collocation at Gauss and Lobatto points in [2]. We recall briefly that the error estimates (4.4) are replaced by

\[ x_i^j - x_i(t_i) = O(h^{k+i} + \epsilon^k) \]

\[ B y_i^j - B y_i(t_{ij}) = O(h^{k} + \epsilon^k), \quad j = 1, \ldots, k \]
for \( t \gg \epsilon \), where \( l = 1 \) if \( k \) is odd and an additional condition on the mesh holds (see (3.13) in [2]), \( l = 0 \) otherwise. Note that the condition (2.25) in [2] is satisfied by the problem (4.1). Observe that the order in (4.6b) is one lower than in (4.4b), essentially because the approximation for \( x(t) \) no longer has a smooth error.

### 4.2 Initial value problem (IVP)

For the IVP case where (4.1b) reduces to

\[
x(0) = \bar{x} \text{ given}
\]

we may, of course, proceed in the same way as for the BVP case. But now a few things are easier. Firstly, for this case we can calculate \( By_x(0) \) and then choose \( By_0 \) to be exact at \( t = 0 \). In fact, as indicated earlier we can also do this for higher derivatives of \( By \) at the initial value by repeated differentiation of (2.1). Such a preparation of the initial iterate \( By_0 \) allows removing the layer error terms (or the condition \( t \gg \epsilon \)) in the error estimates above.

Secondly, one may use a more convenient difference scheme to integrate the IVP (4.1a),(4.7). If the EUODE is sufficiently nonstiff to warrant use of a nonstiff integration method then this can be an attractive possibility here. Note, though, that \(-h/\epsilon \) must be in the absolute stability region of the method (see (3.12b)). Thus, an explicit Runge-Kutta method of order \( p \), for instance, may necessitate (at least) \( p \) SRM iterations in order for the error in the estimates of Theorem 3.1 to be of the same order as the error in the numerical approximation. Depending on the problem one may prefer to choose \( \epsilon \) smaller and apply a stiff method like BDF or collocation at Radau points (see, e.g., [10]). This again yields an advantage over the BVP case, because these methods dampen errors, unlike the symmetric difference schemes. The error estimates are the usual ones (cf. [10]).

For instance, using collocation at \( k \) Radau points \( (t_{ih} = t_i; k = 1 \) yields backward Euler) we have

\[
\begin{align*}
x_i^k - x_o(t_i) &= O(h^{2k-1} + \epsilon^k) \\
By_x(t_{ij}) - By_x(t_{ij}) &= O(h^k + \epsilon^k), & j = 1, \ldots, k
\end{align*}
\]

for \( 1 \leq i \leq N \).

**Remark 4.1** The stiffness introduced by the regularization is an IVP stiffness even in the BVP case – see the IVP for \( w_2 \) in (3.12b),(3.13b). It is therefore tempting, and for many examples certainly useful, to apply a damping difference method such as Radau collocation in the above BVP context as well. However, such a method may be dangerous for a general use if the EUODE gives rise to trouble due to over-damping (see, e.g., Example 10.10 in [4]).

The most important difference between the IVP and BVP cases is that the iterative method described here does not appear to be necessarily optimal or even natural in the IVP context, certainly not from the storage requirement point of view: Note that the entire approximation of \( By_{x,i-1} \) on \([0, t_f]\) needs to be stored. The situation here is similar to that encountered with other functional iteration methods like waveform methods.

However, this difficulty can be resolved by rearranging the computation, assuming that the number of the SRM iterations, \( m_i \), is chosen in advance. Thus, at each time step \( i \), \( 1 \leq i \leq N \), we calculate sequentially the quantities \( x_i^1, By_i^1, x_i^2, By_i^2, \ldots, x_i^n, By_i^n \). To do this using a one-step scheme, say, we need only the corresponding quantities locally, over the mesh subinterval \([t_{i-1}, t_i]\), and \( By_i^1 \). For the latter we may use, for instance, \( y_i^0 \equiv y_x(0) \), i.e. \( By_i^0 = B(t_i)y_0^0 \), \( 0 \leq i \leq N \). The storage requirements are now independent of \( N \) and other typical IVP techniques such as local error control may be applied as well.

**5. Numerical examples.** We now present a few very simple examples to demonstrate our claims in the previous sections. Throughout this section we use a constant step size \( h \) and set \( t_f = 1 \).
To make life difficult we choose \( h \) so that there is an \( i \) such that \( t_i = t_0 \) (if there is a singularity). In the implementation we monitor the size of the pivot in a Gauss elimination procedure for \( CB \) and slightly perturb \( t_i \) away from \( t_0 \) when needed. At a given time \( t \), we use \( 'x' \) to denote the maximum over all components of the error in \( x^* \) while \( 'y' \) denotes the maximum over all components of the error in \( By' \). Similarly, \( 'drib' \) denotes the maximum residual in the algebraic equations.

We first look at a boundary value problem.

**Example 1**

Consider the DAE (2.1) with

\[
A = \begin{pmatrix} -1 & 1 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 - 2t \end{pmatrix}, \quad q = \begin{pmatrix} -\sin t \\ 0 \end{pmatrix}
\]

\[
C = (1 - 2t \quad 1 - 2t), \quad r = -(1 - 2t)(e^{-t} + \sin t)
\]

subject to

\[
x_1(1) + x_2(0) = 1/e
\]

The exact solution is \( x_e = (e^{-t} \quad \sin t) \), \( y_e = \frac{\cosh t}{2} \). A singularity is located at \( t = 1/2 \), where \( y_e \) becomes infinite while \( By_e \) stays bounded. We start computing with the iterate \( y_0(0) \equiv 0 \).

In Table 5.1 we list errors when using the midpoint scheme

\[
\frac{x_i^* - x_{i-1}^*}{h} = A_i - h_B y_i^* + q_i - h
\]

\[
By_i^* = B y_i^* - c^{-1} B y_{i-1}^* - \frac{C_i - h}{2} x_{i-1}^* + r_i - h
\]

where \( x_i^* = \frac{x_i^* + x_{i-1}^*}{2} \) (but no such relation is necessary for \( y^* \)). We apply this scheme with \( h_i = h = .01 \) for various values of \( \epsilon \).

Since \( k = 1, 2k = k + 1 = 2 \) and we expect similar 2nd order accuracy in \( x \) and in \( cy \), except for the case \( \epsilon \ll h \) when the error order in \( By \) drops to 1. This is evident in the error column for \( t = 1.0 \). Note also the \( O(\epsilon) \) improvement per SRM iteration when this term dominates the error (i.e. when \( \epsilon \gg h^2 \)). Further experiments with different values of \( h \) verify the convergence orders of (4.4) and (4.6) for \( k = 1 \).

We note that the approximation for \( By \) at points within the initial layer is not accurate. To get a better approximation within the initial layer (i.e. near the initial point \( t = 0 \)), we solve a terminal value problem (4.1a), (4.5), as described in §4.1. Then we apply the SRM for the given problem with the improved values for \( By_0 \). In Table 5.2 we list the computed results after 3 iterations. They are obviously much better than the comparable ones in Table 5.1.

\[ \Box \]

Next we consider initial value problems.

**Example 2**

Consider the same DAE as for Example 1 with the same exact solution but with initial values \( x_1(0) = 1, \quad x_2(0) = 0 \) specified. From these initial conditions we can calculate \( y(0) = 1 \) in advance, and we choose the initial guess \( y_0(0) \equiv 1 \). Tables 5.3 and 5.4 display error results for \( \epsilon = .1 \) and \( h = .001 \) using the backward Euler and the forward Euler schemes, respectively. As explained in §4.2 we find all iterates at each step before proceeding to the next.

These tables show a significant improvement with each SRM iteration and no strong initial layer effect, as predicted by theory.
Example 3

Here we investigate the use of the modified formula (3.16) instead of (3.8) in the SRM. First, we solve the previous example numerically using (3.16). In Table 5.5 we record error values at the singularity point \( t = .5 \) after 3 SRM iterations, starting with \( y_0(t) \equiv 1 \) and using as before \( \epsilon = .1 \) and \( h = .001 \) (cf. Tables 5.3, 5.4).

From these results it is clear that the SRM with (3.16) does not work well when \( \alpha_1 \neq 0 \): large errors in \( By \) are obtained near the singularity and these adversely affect the accuracy in \( x \) as well. However, the comparison changes when there is no singularity in the constraints: We now replace
<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>error at $t=0.01$</th>
<th>$t=0.1$</th>
<th>$t=0.3$</th>
<th>$t=0.5$</th>
<th>$t=1.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e-1</td>
<td>ex</td>
<td>.36e-2</td>
<td>.46e-2</td>
<td>.39e-2</td>
<td>.28e-2</td>
</tr>
<tr>
<td></td>
<td>ey</td>
<td>.48e-2</td>
<td>.45e-2</td>
<td>.44e-2</td>
<td>.54e-2</td>
</tr>
<tr>
<td>5e-2</td>
<td>ex</td>
<td>.48e-3</td>
<td>.58e-3</td>
<td>.32e-3</td>
<td>.17e-3</td>
</tr>
<tr>
<td></td>
<td>ey</td>
<td>.22e-2</td>
<td>.18e-2</td>
<td>.10e-2</td>
<td>.50e-3</td>
</tr>
<tr>
<td>1e-2</td>
<td>ex</td>
<td>.57e-4</td>
<td>.49e-4</td>
<td>.35e-4</td>
<td>.26e-4</td>
</tr>
<tr>
<td></td>
<td>ey</td>
<td>.10e-3</td>
<td>.85e-4</td>
<td>.52e-4</td>
<td>.32e-4</td>
</tr>
<tr>
<td>1e-3</td>
<td>ex</td>
<td>.49e-4</td>
<td>.42e-4</td>
<td>.31e-4</td>
<td>.23e-4</td>
</tr>
<tr>
<td></td>
<td>ey</td>
<td>.56e-4</td>
<td>.46e-4</td>
<td>.30e-4</td>
<td>.10e-4</td>
</tr>
</tbody>
</table>

**Table 5.2**

SRM errors for Example 1 using the shooting-back technique

<table>
<thead>
<tr>
<th>iteration</th>
<th>error at $t=0.01$</th>
<th>$t=0.1$</th>
<th>$t=0.3$</th>
<th>$t=0.5$</th>
<th>$t=1.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ex</td>
<td>.29e-5</td>
<td>.72e-2</td>
<td>.34e-1</td>
<td>.63e-1</td>
</tr>
<tr>
<td></td>
<td>ey</td>
<td>.20e-2</td>
<td>.12</td>
<td>.15</td>
<td>.12</td>
</tr>
<tr>
<td></td>
<td>drift</td>
<td>.15e-5</td>
<td>.60e-2</td>
<td>.16e-1</td>
<td>.76e-4</td>
</tr>
<tr>
<td>2</td>
<td>ex</td>
<td>.20e-5</td>
<td>.31e-2</td>
<td>.13e-1</td>
<td>.10e-1</td>
</tr>
<tr>
<td></td>
<td>ey</td>
<td>.20e-2</td>
<td>.68e-1</td>
<td>.45e-2</td>
<td>.20e-1</td>
</tr>
<tr>
<td></td>
<td>drift</td>
<td>.15e-5</td>
<td>.42e-2</td>
<td>.58e-2</td>
<td>.14e-4</td>
</tr>
<tr>
<td>3</td>
<td>ex</td>
<td>.20e-5</td>
<td>.35e-2</td>
<td>.23e-2</td>
<td>.16e-2</td>
</tr>
<tr>
<td></td>
<td>ey</td>
<td>.20e-2</td>
<td>.32e-1</td>
<td>.26e-1</td>
<td>.10e-1</td>
</tr>
<tr>
<td></td>
<td>drift</td>
<td>.15e-5</td>
<td>.29e-2</td>
<td>.12e-2</td>
<td>.10e-5</td>
</tr>
</tbody>
</table>

**Table 5.3**

SRM errors for Example 2 using backward Euler

the constraint in Example 2 by

$$x_1 + x_2 - e^{-t} - \sin t = 0$$

leaving everything else the same (including the singularity in $B$). In Table 5.6 we record maximum errors in $x$ and $By$ over all mesh points (denote those 'exg' and 'eyg', respectively) for the starting iterates $y_0(t) \equiv 1$ and $y_0(t) \equiv 0$ (the latter does not agree with the exact $y_e(0)$).

The modified method is seen to work better for problems without singularities.

The above calculations all agree with our theoretical results described in Sections 3 and 4. The final example below is designed to compare our method with that of Park and Chiu [17] (i.e. (3.4) with $E = I$ for an example with $B = C^T$ and a smooth $y$ over the entire interval).

**Example 4**

Consider the DAE (2.1) with

$$A = 0, \quad C = \begin{pmatrix} t - 5 & t^2 - .25 \end{pmatrix}, \quad B = C^T$$

where $q(t)$ and $r(t)$ are chosen such that the exact solution is $x_1 = x_2 = y = e^t$.

To try a wider range of values for $\epsilon$, we use the backward Euler scheme. We set $h = .001$ and perform 4 SRM iterations with $y_0(t) \equiv 1$. This is compared to the numerical solution of (3.5a),(3.7) [17] using the same discretization. The resulting overall errors are listed in Table 5.7 and demonstrate the superiority of our approach.

6. **Appendix.** To provide a better feeling about the sequential regularization method we now give a detailed proof of Theorem 3.1 for the initial value case with no layers, $s \leq M + 1$. In this proof, the construction of the asymptotic expansion is directly for $x$ and $By$. Moreover, the construction method we apply is somewhat different from [20, 21] and more relevant to the concept of DAEs. For
Table 5.4  

<table>
<thead>
<tr>
<th>iteration</th>
<th>error at</th>
<th>( t=0.1 )</th>
<th>( t=0.2 )</th>
<th>( t=0.3 )</th>
<th>( t=0.4 )</th>
<th>( t=1.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \text{ex} )</td>
<td>.90e-6</td>
<td>.1e-2</td>
<td>.36e-1</td>
<td>.36e-1</td>
<td>.11</td>
</tr>
<tr>
<td></td>
<td>( \text{ey} )</td>
<td>.20e-2</td>
<td>12</td>
<td>.15</td>
<td>.12</td>
<td>.60e-1</td>
</tr>
<tr>
<td>2</td>
<td>( \text{ex} )</td>
<td>.50e-6</td>
<td>.60e-2</td>
<td>.16e-1</td>
<td>.76e-4</td>
<td>.15</td>
</tr>
<tr>
<td></td>
<td>( \text{ey} )</td>
<td>.20e-2</td>
<td>.68e-1</td>
<td>.41e-2</td>
<td>.20e-1</td>
<td>.70e-2</td>
</tr>
<tr>
<td>3</td>
<td>( \text{ex} )</td>
<td>.50e-6</td>
<td>.42e-2</td>
<td>.58e-2</td>
<td>.14e-4</td>
<td>.67e-2</td>
</tr>
<tr>
<td></td>
<td>( \text{ey} )</td>
<td>.20e-2</td>
<td>.32e-1</td>
<td>.26e-1</td>
<td>.97e-2</td>
<td>.46e-2</td>
</tr>
<tr>
<td></td>
<td>drift</td>
<td>.50e-6</td>
<td>.29e-2</td>
<td>.12e-2</td>
<td>.11e-5</td>
<td>.12e-2</td>
</tr>
</tbody>
</table>

Table 5.5  

<table>
<thead>
<tr>
<th>( (\alpha_1, \alpha_2) )</th>
<th>( (0, 1) )</th>
<th>( (h, 1) )</th>
<th>( (1, 1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>backward Euler</td>
<td>( 16e-2 )</td>
<td>( .10e-3 )</td>
<td>( .80e-3 )</td>
</tr>
<tr>
<td>forward Euler</td>
<td>( 18e-2 )</td>
<td>( .96e-2 )</td>
<td>( .25e-2 )</td>
</tr>
</tbody>
</table>

**SRM errors for Example 2 using forward Euler**

**Errors near singularity using modified formula (3.18)**

\( s > M + 1 \), additional initial layer expansions have to be developed. However, the construction of these layer expansions is precisely the same as in [20, 21] and so it is omitted here. In case that (2.2) are initial conditions (i.e. \( B_1 = 0 \)) our assumptions imply that (2.2) together with (2.3b) specify \( x(0) \), say

\[
x(0) = \tilde{x}
\]

At first, consider the case \( s = 1 \) of (3.3),(3.8):

\[
\epsilon x_1' + B(CB)^{-1}(Cx_1 + r) = \epsilon Ax_1 + \epsilon By_0 + \epsilon q,
\]

with the initial conditions (6.1). This is a singular-singularity-perturbed problem (see [21, 16]). Let

\[
x_1 = x_{10} + \epsilon x_{11} + \cdots + \epsilon^i x_{1i} + \cdots
\]

Comparing the coefficients of like powers of \( \epsilon \), we thus have

\[
B(CB)^{-1}C_{x10} = -B(CB)^{-1}r
\]

(6.2a)

\[
B(CB)^{-1}C_{x11} = -x_1' + Ax_{10} + By_0 + q,
\]

(6.2b)

\[
B(CB)^{-1}C_{x1i} = -x_{1i-1}' + Ax_{1i-1}, \quad 2 \leq i \leq s + 1,
\]

(6.2c)

where (6.2a) satisfies (6.1) and (6.2b) and (6.2c) satisfy homogeneous initial conditions corresponding to (6.1). Now, (6.2a) has infinitely many solutions in general. To realize the construction, we should choose \( x_{10} \) to satisfy (6.2a) and to ensure that the solution of (6.2b) exists. We choose \( x_{10} \) to be the solution \( x_\epsilon \) of problem (2.1)-(2.2), i.e.

\[
x_{10}' = Ax_{10} + By_\epsilon + q,
\]

(6.3a)

\[
0 = Cx_{10} + r,
\]

(6.3b)

\[
B_0 x_{10}(0) = \beta.
\]

(6.3c)

So \( x_{10} = x_\epsilon \) and (6.2b) has the following form

\[
B(CB)^{-1}C_{x11} = B(y_0 - y_\epsilon).
\]

(6.4)
\[
(\alpha_1, \alpha_2) \rightarrow (0, 1) \quad (h, 1) \quad (1, 1)
\]

<table>
<thead>
<tr>
<th>( y_0 )</th>
<th>method</th>
<th>( \epsilon = 10^{-1} )</th>
<th>( \epsilon = 10^{-2} )</th>
<th>( \epsilon = 10^{-3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \equiv 1 )</td>
<td>backward Euler</td>
<td>46c-2</td>
<td>44c-1</td>
<td>43c-2</td>
</tr>
<tr>
<td></td>
<td>forward Euler</td>
<td>45c-2</td>
<td>44c-1</td>
<td>44c-2</td>
</tr>
<tr>
<td>( \equiv 0 )</td>
<td>backward Euler</td>
<td>22c-1</td>
<td>.97</td>
<td>.94</td>
</tr>
<tr>
<td></td>
<td>forward Euler</td>
<td>22c-1</td>
<td>.97</td>
<td>.94</td>
</tr>
</tbody>
</table>

Table 5.6

Errors for problem without singularity using modified formula (2.16)

<table>
<thead>
<tr>
<th>method</th>
<th>( \epsilon = 10^{-1} )</th>
<th>( \epsilon = 10^{-2} )</th>
<th>( \epsilon = 10^{-3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRM (4 itns.)</td>
<td>.15c-2</td>
<td>.1c-2</td>
<td>.2c-2</td>
</tr>
<tr>
<td>Park &amp; Chiou</td>
<td>.12</td>
<td>.45</td>
<td>.38c-2</td>
</tr>
</tbody>
</table>

Table 5.7

Method comparison for Example 4

Now we choose \( x_{11} \) and a corresponding \( y_{01} \) to satisfy

(6.5a) \[ x_{11}' = Ax_{11} + By_{01} \]
(6.5b) \[ Cx_{11} = CB(y_0 - y_e), \]
(6.5c) \[ B_0x_{11}(0) = 0. \]

Noting that \( B_0 = x'-Ax-e \) is smooth, we have \( CB(y_0 - y_e) \in S \). Hence, using Lemma 2.1, there exists a smooth solution \( x_{11} \) of (6.5), and \( x_{11} \) satisfies (6.4). Indeed, using (6.5b) and H4, we have \( C(0)x_{11}(0) = 0, \) so \( x_{11}(0) = 0. \) And, from (6.5b) again,

\[
(CB)^{-1}Cx_{11} = y_0 - y_e, \quad \text{for each } t \in [0, t_s) \cup (t_s, t_f].
\]

That is,

\[
B(CB)^{-1}Cx_{11} = B(y_0 - y_e), t \in [0, t_s) \cup (t_s, t_f].
\]

Taking the limit of (6.6), we thus get that \( x_{11} \) satisfies (6.4) for each \( t \in [0, t_f]. \)

Moreover, using H4, we have

\[
y_{01}(0) = y_{01}'(0) = \cdots = y_{01}^{(s-1)}(0) = 0, \quad s \leq M + 1.
\]

Also we note that \( B_0y_{01} \) is smooth.

Generally, supposing we have got \( x_{1i-1}, B_0y_{0i-1} \) and

\[
y_{0i-1}(0) = y_{0i-1}'(0) = \cdots = y_{0i-1}^{(s-i+1)}(0) = 0
\]

for \( i \geq 2 \), we choose \( x_{1i}, y_{0i} \) satisfying

\[
x_{1i}' = Ax_{1i} + B_0y_{0i},
\]

\[
C_{x_{1i}} = -(CB)y_{0i-1},
\]

\[
B_0x_{1i}(0) = 0.
\]

By the same argument as before, we obtain that \( x_{1i} \) satisfies (6.2c) for \( 2 \leq i \leq s + 1, \) and

\[
y_{0i}(0) = y_{0i}'(0) = \cdots = y_{0i}^{(s-i)}(0) = 0, \quad s \leq M + 1.
\]

17
Also, $B y_0$ is smooth. Next we denote the asymptotic solution
\[ x_{1s+1} = x_{10} + c x_{11} + \cdots + \epsilon^s x_{1s} + \epsilon^{s+1} x_{1s+1} \]
and
\[ z_{1s+1} = x_1 - x_{1s+1}. \]
Then
\[ \epsilon z_{1s+1}^t + P z_{1s+1} = c A z_{1s+1} + \epsilon^{s+2}(-z_{1s+1}^t + A x_{1s+1}), \]
\[ z_{1s+1}(0) = 0 \]
Let $u_{1s+1} = R z_{1s+1}$ and $w_{1s+1} = P z_{1s+1}$. Hence, we have (cf. (3.11))
\[ z_{1s+1} = Su_{1s+1} + w_{1s+1} \]
and
\[ u_{1s+1} = (RA + R') Su_{1s+1} + (RA + R') w_{1s+1} + O(\epsilon^{s+1}), \]
\[ \epsilon u_{1s+1}^t + w_{1s+1} = c (PA + P') Su_{1s+1} + c (PA + P') w_{1s+1} + O(\epsilon^{s+2}), \]
\[ u_{1s+1}(0) = 0, \quad w_{1s+1}(0) = 0. \]
Using Lemma 3.1, we get $w_{1s+1} = O(\epsilon^{s+2})$ and $u_{1s+1} = O(\epsilon^{s+1})$, i.e.
\[ z_{1s+1} = O(\epsilon^{s+1}). \]
Therefore,
\[ (6.7) \quad x_1 = x_{10} + c x_{11} + \cdots + \epsilon^s x_{1s} + O(\epsilon^{s+1}). \]
Noting $x_{10} = x$, we thus obtain
\[ (6.8) \quad x_1 - x = O(\epsilon). \]
Then, by using (3.8),(6.7),(6.2),(6.3a) and (6.5a), it follows that
\[ By_1 = By_0 - \frac{1}{\epsilon} B (CB)^{-1} (Cx_1 + r) \]
\[ (6.9) \quad By_1 = By_0 - \frac{1}{\epsilon} (P x_{10} + B (CB)^{-1} r + \epsilon P x_{11} + \cdots + \epsilon^s P x_{1s} + O(\epsilon^{s+1})) = By_0 + c By_0 + \cdots + \epsilon^s By_{0s-1} + O(\epsilon) \]
or
\[ (6.10) \quad By_1 - By_0 = O(\epsilon). \]
Now we look at the second iteration $s = 2$ of (3.3):
\[ \epsilon x_2^t + B (CB)^{-1} (Cx_2 + r) = c A x_2 + c B y_1 + c q, \]
with initial conditions (6.1). Let
\[ x_2 = x_{20} + \varepsilon x_{21} + \varepsilon^2 x_{22} + \cdots. \]
Noting that (6.9) gives us a series expansion for \( By_1 \) we obtain,

(6.11a) \[ B(CB)^{-1} C x_{20} = -B(CB)^{-1} r, \]
(6.11b) \[ B(CB)^{-1} C x_{21} = -x_{20} + A x_{21} + B y_\varepsilon + q, \]
(6.11c) \[ B(CB)^{-1} C x_{2i} = -x_{2i-1} + A x_{2i-1} + B y_{\varepsilon i-1}, \quad 2 \leq i \leq s + 1 \]

Again, (6.11a) satisfies initial conditions (6.1) and (6.11b) and (6.11c) satisfy the corresponding homogenous ones. As the case of \( s = 1 \), we choose \( x_{20} = x_\varepsilon \). We thus have
\[ B(CB)^{-1} C x_{21} = 0. \]
Then \( x_{21} \) is constructed to satisfy

(6.12a) \[ x'_{21} = A x_{21} + B y_1, \]
(6.12b) \[ C x_{21} = 0, \]
(6.12c) \[ B_0 x_{21}(0) = 0. \]

Obviously \( x_{21} = 0 \) since (6.12) is uniquely solvable for \( x_{21} \) by Lemma 2.1. In general, similarly to the case of \( s = 1 \), we choose \( x_{2i} \) satisfying

(6.13a) \[ x'_{2i} = A x_{2i} + B y_i, \]
(6.13b) \[ C x_{2i} = -(CB)(y_{\varepsilon i-1} - y_i), \]
(6.13c) \[ B_0 x_{2i}(0) = 0. \]

for \( 2 \leq i \leq s + 1 \). By applying Lemma 3.1 and the same argument as in the case of \( s = 1 \) we get

(6.14) \[ x_2 = x_\varepsilon + \varepsilon x_{21} + \varepsilon^2 x_{22} + \cdots + \varepsilon^i x_{2i} + O(\varepsilon^{i+1}) \]
or

(6.15) \[ x_2 - x_\varepsilon = O(\varepsilon^2). \]

Then, using (3.8),(6.11),(6.12a),(6.13a), (6.14) and (6.9), we conclude

(6.16) \[ By_2 = By_1 - \frac{1}{\varepsilon} B(CB)^{-1} (C x_2 + r) = By_\varepsilon + \varepsilon^2 B y_{12} + \cdots + \varepsilon^{i-1} B y_{\varepsilon i-1} + O(\varepsilon^i) \]
or

(6.17) \[ By_2 - By_\varepsilon = O(\varepsilon^2) \]

We can repeat this procedure, and, by induction, conclude the results of the theorem. \( \Box \)

REFERENCES