European Society of Computational Methods in Sciences and Engineering (ESCMSE)

Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM)
vol. 4, no. 1-2, 2009, pp. 129-149
ISSN 1790–8140

Numerical Solution of Singular Two Point BVPs[1]

J. Cash, G. Kitzhofer, O. Koch, G. Moore, E. Weinmüller [2]

Dedicated to John Butcher on the occasion of his 75th birthday.

Received 27 April, 2009; accepted in revised form 30 April, 2009

Abstract: An algorithm is described for the efficient numerical solution of singular two-point boundary value problems. The algorithm is based on collocation at Gauss points, is applied directly to second order equations and uses a transformation of the independent variable to obtain extra smoothness if needed. Numerical comparisons between a code based on this approach and codes based on other options which have previously been thought of as possible alternatives such as collocation at Lobatto points or reduction to a first order system, are made and the efficiency of the new approach is clearly demonstrated by numerical results.

Keywords: Singular boundary value problems, polynomial collocation, Lobatto points, Gaussian points, performance comparisons.

Mathematics Subject Classification: 65L10, 65L60

PACS: 02.60.Cb, 02.70.Jn

1 Introduction

Mathematical models of classical applications from physics, chemistry and mechanics (e.g. the Thomas–Fermi differential equation, the Ginzburg–Landau equation) take the form of singular boundary value problems of second order: the singularity typically occurring at an end of the interval of integration. ODEs with singularities appear also in numerous applications which are of interest in modern applied mathematics. Computations of self-similar blow-up solutions of nonlinear PDEs lead to the computation of problems from this class ([7], [8], [9], [10], [11]). Also, the density profile equation in hydrodynamics may be reduced to a singular ODE ([24], [29]). Finally, the investigation of problems in the theory of shallow membrane caps, [35], is associated with such problems. Even in ecology, in the computation of avalanche run-up, this problem class is present ([27], [32]). Further research areas include the solution of differential equations posed on unbounded intervals ([1], [10] and [16]), the computation of connecting orbits or invariant manifolds for dynamical systems ([33] and [34]), differential-algebraic equations ([31]) or Sturm–Liouville eigenvalue problems ([14], and [30]). Hence several papers are concerned with numerical methods for approximating such problems, cf. [12], [15], [17], [22].

The aim of the present paper is to give a comprehensive assessment of the performance of some standard *collocation algorithms*, when applied to singular boundary value problems: in particular we compare numerical results for both first and second order formulations of the same singular boundary value problem and compare both Gauss and Lobatto collocation algorithms. Collocation at Lobatto points needs special consideration for singular boundary value problems, because then we need to collocate at the singular end-point. We also consider how a suitable transformation of the independent variable can affect the performance of our collocation algorithms. Again, this is especially important for singular boundary value problems, because such a transformation

[1]Published electronically June 15, 2009
[2]Corresponding author. E-mail: e.weinmueller@tuwien.ac.at

alters key eigenvalues and may increase the smoothness of our solution and consequently, the performance of numerical solution methods. Concerning software, there are some possibilities to use open domain codes to treat singular boundary value problems, cf. Section 4, but we do not intend to compare these codes here. This is an interesting but difficult question and it will be addressed in an upcoming paper.

The contents of the present paper are as follows. In Section 2 we briefly state the basic analytical properties of singular boundary value problems with a singularity of the first kind, for both first and second order systems. We emphasise how the number of boundary conditions required for a well-posed problem depends on certain eigenvalue conditions. We also show how a transformation of the independent variable alters these eigenvalue conditions, and may also alter the smoothness of our solutions. In Section 3 we discuss how collocation algorithms can be applied to approximate the solution of singular boundary value problems, both for first order and second order systems. We emphasise how the *implicit* boundary conditions in the continuous formulation must be made *explicit* in order to construct a well-posed discrete problem. We also discuss how to collocate at the singular end-point, when a Lobatto algorithm is used. Finally, we show how to transform from second order singular systems to corresponding first order singular systems, in a manner which maintains the structure of the boundary value problem. We describe the model problems that we use in Section 4, and display some numerical results in Section 5. Based on the extensive numerical testing that we have done, we recommend an algorithm which uses collocation on Gauss points, which uses variable stepsize to control the error, is applied directly to second order problems and which uses a transformation of the independent variable to obtain extra smoothness if needed. We expect that our code based on this approach will be very competitive with the best existing codes and we will investigate this in detail in future work.

2 Analytical Results for Singular Problems

In this section we will briefly recapitulate some of the analytical properties of singular systems of first and second order boundary value problems. For more details, we refer to [20] for the former and [40] for the latter.

2.1 Singular first order systems

We consider the nonlinear ODE

$$\mathbf{y}'(t) - \frac{\mathsf{M}}{t}\mathbf{y}(t) = \mathbf{f}(t, \mathbf{y}(t)) \quad 0 < t \le 1 \tag{2.1}$$

with boundary conditions

$$\mathbf{b}(\mathbf{y}(0), \mathbf{y}(1)) = \mathbf{0}, \tag{2.2}$$

where $\mathbf{y} : [0,1] \mapsto \mathbb{R}^n$, $\mathsf{M} \in \mathbb{R}^{n\times n}$, $\mathbf{f} : [0,1] \times \mathbb{R}^n \mapsto \mathbb{R}^n$ and $\mathbf{b} : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^p$ with the value of p discussed below. We shall be seeking a solution $\mathbf{y}^\star \in C[0,1] \cap C^1(0,1]$ and so we assume that $\mathbf{f}$ is continuous with respect to its first argument and continuously differentiable with respect to its second. Also $\mathbf{b}$ is assumed to be continuously differentiable with respect to its arguments.

The most important consideration for the matrix M is the sign of the real part of its eigenvalues. We must have $p = n_+ + n_0$ for a well-posed problem, where n_+ is the dimension of the invariant subspace of M corresponding to those eigenvalues with strictly positive real part (i.e. combined algebraic multiplicities) and n_0 is the dimension of the null-space of M (i.e. geometric multiplicity of the zero eigenvalue). This is because the assumption that $\mathbf{y}^\star$ is continuous at $t = 0$ implicitly provides $n - (n_+ + n_0)$ initial conditions.

We assume that we are trying to approximate a solution $\mathbf{y}^\star$ which is *isolated*, i.e. the linear problem

$$\mathbf{y}'(t) - \frac{\mathsf{M}}{t}\mathbf{y}(t) + \mathsf{N}(t)\mathbf{y}(t) = \mathbf{0} \quad 0 < t \le 1$$
$$\mathsf{B}_0\mathbf{y}(0) + \mathsf{B}_1\mathbf{y}(1) = \mathbf{0}$$

has only the trivial solution in $C[0,1] \cap C^1(0,1]$, where $\mathsf{N}(t)$ is the $n \times n$ Jacobian matrix of $\mathbf{f}$ with respect to its second argument, evaluated at $(t, \mathbf{y}^\star(t))$, B_0 is the $n \times n$ Jacobian matrix of $\mathbf{b}$ with respect to its first argument, evaluated at $(\mathbf{y}^\star(0), \mathbf{y}^\star(1))$ and B_1 is the $n \times n$ Jacobian matrix of $\mathbf{b}$ with respect to its second argument, evaluated at $(\mathbf{y}^\star(0), \mathbf{y}^\star(1))$.

The smoothness of $\mathbf{y}^\star$ depends on the smoothness of $\mathbf{f}$ and the positive real parts of eigenvalues of M, e.g. when, for $s \geq 0$, $\mathbf{f}$ has s continuous derivatives and no eigenvalue of M has real part in $(0, s+1]$, then we must have $\mathbf{y}^\star \in C^{s+1}[0,1]$, see [20]. The influence of these key eigenvalues of M may be mitigated by a change of independent variable: i.e. if we set

$$t = \tau^\gamma$$

for some $\gamma > 1$, then $\tilde{\mathbf{y}}^\star(\tau) \equiv \mathbf{y}^\star(\tau^\gamma)$ satisfies

$$\mathbf{y}'(\tau) - \frac{\widetilde{\mathsf{M}}}{\tau}\mathbf{y}(\tau) = \tilde{\mathbf{f}}(\tau, \mathbf{y}(\tau)), \quad 0 < \tau \leq 1, \tag{2.3}$$

where

$$\widetilde{\mathsf{M}} \equiv \gamma \mathsf{M} \quad \text{and} \quad \tilde{\mathbf{f}}(\tau, \mathbf{x}) \equiv \gamma \tau^{\gamma-1}\mathbf{f}(\tau^\gamma, \mathbf{x}),$$

and the boundary conditions (2.2). Thus the eigenvalues of $\widetilde{\mathsf{M}}$ are multiplied by γ compared to the eigenvalues of M and so $\tilde{\mathbf{y}}^\star$ can be smoother than $\mathbf{y}^\star$.

2.2 Singular second order systems

We consider the nonlinear ODE

$$\mathbf{z}''(t) - \frac{\mathsf{A}_1}{t}\mathbf{z}'(t) - \frac{\mathsf{A}_0}{t^2}\mathbf{z}(t) = \mathbf{g}(t, \mathbf{z}(t)) \quad 0 < t \leq 1 \tag{2.4}$$

with boundary conditions

$$\mathbf{c}(\mathbf{z}(0), \mathbf{z}(1), \mathbf{z}'(1)) = \mathbf{0}, \tag{2.5}$$

where $\mathbf{z} : [0,1] \mapsto \mathbb{R}^n$, $\mathsf{A}_0, \mathsf{A}_1 \in \mathbb{R}^{n\times n}$, $\mathbf{g} : [0,1] \times \mathbb{R}^n \mapsto \mathbb{R}^n$ and $\mathbf{c} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^p$ with the value of p discussed below. We shall be seeking a solution $\mathbf{z}^\star \in C[0,1] \cap C^2(0,1]$ and so we assume that $\mathbf{g}$ is continuous with respect to its first argument and continuously differentiable with respect to its second. Also $\mathbf{c}$ is assumed to be continuously differentiable with respect to its arguments.

The important consideration for the matrices A_0 and A_1 is the quadratic eigenvalue problem

$$\det\left(\lambda^2 \mathsf{I} - \lambda[\mathsf{I} + \mathsf{A}_1] - \mathsf{A}_0\right) = 0, \tag{2.6}$$

see [18]. We must have $p = n_+ + n_0$ for a well-posed problem, where n_+ is the number of roots (counted in multiplicity) with strictly positive real part of the $2n^{th}$ degree polynomial in (2.6) and n_0 is the dimension of the null-space of A_0 (i.e. geometric multiplicity of the zero eigenvalue). This is because the assumption that $\mathbf{z}^\star$ is continuous at $t = 0$ implicitly provides $2n - (n_+ + n_0)$ initial conditions.

We assume that we are trying to approximate a solution $\mathbf{z}^\star$ which is *isolated*, i.e. the linear problem

$$\begin{gathered}\mathbf{z}''(t) - \frac{\mathsf{A}_1}{t}\mathbf{z}'(t) - \frac{\mathsf{A}_0}{t^2}\mathbf{z}(t) - \mathsf{A}(t)\mathbf{z}(t) = \mathbf{0} \quad 0 < t \leq 1 \\ \mathsf{C}_0\mathbf{z}(0) + \mathsf{C}_1\mathbf{z}(1) + \mathsf{C}_1'\mathbf{z}'(1) = \mathbf{0}\end{gathered}$$

has only the trivial solution in $C[0,1] \cap C^2(0,1]$, where $\mathsf{A}(t)$ is the $n \times n$ Jacobian matrix of $\mathbf{g}$ with respect to its second argument, evaluated at $(t, \mathbf{z}^\star(t))$, C_0 is the $n \times n$ Jacobian matrix of $\mathbf{c}$ with respect to its first argument, evaluated at $(\mathbf{z}^\star(0), \mathbf{z}^\star(1), \mathbf{z}^{\star\prime}(1))$, C_1 is the $n \times n$ Jacobian matrix of $\mathbf{c}$ with respect to its second argument, evaluated at $(\mathbf{z}^\star(0), \mathbf{z}^\star(1), \mathbf{z}^{\star\prime}(1))$ and C_1' is the $n \times n$ Jacobian matrix of $\mathbf{c}$ with respect to its third argument, evaluated at $(\mathbf{z}^\star(0), \mathbf{z}^\star(1), \mathbf{z}^{\star\prime}(1))$.

The smoothness of $\mathbf{z}^\star$ depends on the smoothness of $\mathbf{g}$ and the positive real parts of the roots of (2.6), e.g. when, for $s \geq 0$, $\mathbf{g}$ has s continuous derivatives and no root of (2.6) has real part in $(0, s+2]$, then we must have $\mathbf{z}^\star \in C^{s+2}[0,1]$, see [40]. The influence of these key roots of (2.6) may be mitigated by a change of independent variable: i.e. if we set

$$t = \tau^\gamma$$

for some $\gamma > 1$, then $\tilde{\mathbf{z}}^\star(\tau) \equiv \mathbf{z}^\star(\tau^\gamma)$ satisfies

$$\mathbf{z}''(\tau) - \frac{\widetilde{\mathsf{A}}_1}{\tau}\mathbf{z}'(\tau) - \frac{\widetilde{\mathsf{A}}_0}{\tau^2}\mathbf{z}(\tau) = \tilde{\mathbf{g}}(\tau, \mathbf{z}(\tau)) \quad 0 < \tau \le 1, \tag{2.7}$$

where

$$\widetilde{\mathsf{A}}_1 \equiv [\gamma - 1]\mathsf{I} + \gamma\mathsf{A}_1, \quad \widetilde{\mathsf{A}}_0 \equiv \gamma^2\mathsf{A}_0, \quad \text{and} \quad \tilde{\mathbf{g}}(\tau, \mathbf{x}) \equiv \gamma^2\tau^{2\gamma-2}\mathbf{g}(\tau^\gamma, \mathbf{x}),$$

and the boundary conditions (2.5). Thus the roots of

$$\det\left(\lambda^2\mathsf{I} - \lambda[\mathsf{I} + \widetilde{\mathsf{A}}_1] - \widetilde{\mathsf{A}}_0\right) = 0$$

are multiplied by γ compared to the roots of (2.6) and so $\tilde{\mathbf{z}}^\star$ may be smoother than $\mathbf{z}^\star$.

3 Numerical Treatment of Singular Problems

In the previous section, we have seen that well-posed singular problems usually require less boundary conditions than standard ODEs, because the restriction to solutions in $C[0, 1]$ imposes some *implicit* side conditions. When our singular problem is discretised, however, these 'hidden' restrictions must be imposed explicitly in order to have a well-posed discrete problem. In particular, this is true for a numerical method based on collocation at Gaussian points. If one uses collocation at Lobatto points, moreover, there is the additional question of how to collocate at the singular point $t = 0$. The fact that the differential equation is assumed to hold at this point means that the solution must satisfy extra smoothness, and this allows the construction of an appropriate limiting operator as $t \to 0$ which replaces the singular differential equation.

3.1 Collocation for first order systems

We now show how to construct $n - (n_+ + n_0)$ extra linear, homogeneous, initial conditions for the singular ODE (2.1): i.e. which augment the $n_+ + n_0$ boundary conditions (2.2) and lead to a well-posed discrete problem for collocation at Gaussian points. This theory is now quite well known and in what follows we will gather together the results which we will need to obtain our numerical results. For the matrix M, let X_0 denote the n_0-dimensional null-space and X_+ denote the n_+-dimensional invariant subspace associated with eigenvalues having strictly positive real part. Hence $X_0 \oplus X_+$ has dimension $n_0 + n_+$. Now construct an $n \times (n - [n_+ + n_0])$ matrix Q, whose columns are orthonormal and span the subspace

$$(X_0 \oplus X_+)^\perp.$$

(This is a standard linear algebra task; see, for example, section 7.6 in [19].) Our extra $n - [n_+ + n_0]$ initial conditions, see [20], are therefore simply

$$\mathsf{Q}^T\mathbf{y}(0) = \mathbf{0}. \tag{3.1}$$

(Of course it is not essential to choose Q to be orthogonal: any well-conditioned basis for $(X_0 \oplus X_+)^\perp$ is acceptable.)

If we are using Lobatto points, we need to assume that $\mathbf{y}^\star(t)$ is differentiable at $t = 0$ in order to collocate there. This will only be true for generic $\mathbf{f}$ if M has no eigenvalues with real part in the interval $(0, 1]$. (The simple scalar example

$$y'(t) - \frac{1}{t}y(t) = 1 \qquad \text{with } y(1) = 0,$$

having solution $y^\star(t) \equiv t\ln t$, shows that an eigenvalue equal to 1 is not allowed.) Hence, if necessary, we must implement the change of independent variable (2.3) to satisfy this eigenvalue condition. Once this smoothness constraint holds, we can use the fact that $\mathbf{y}^\star(0)$ is in the kernel of M (cf. [20]) to write

$$\begin{aligned}\lim_{t\to 0}\frac{\mathsf{M}}{t}\mathbf{y}(t) &= \lim_{t\to 0}\mathsf{M}\frac{\mathbf{y}(t) - \mathbf{y}(0)}{t} \\ &= \mathsf{M}\mathbf{y}'(0).\end{aligned}$$

Thus, at the Lobatto point $t = 0$, we should replace (2.1) with the requirement

$$(\mathsf{I} - \mathsf{M})\mathbf{y}'(0) = \mathbf{f}(0, \mathbf{y}(0)), \tag{3.2}$$

and $\mathsf{I} - \mathsf{M}$ is guaranteed to be non-singular.

3.2 Collocation for second order systems

We now show how to construct $2n - (n_+ + n_0)$ extra linear, homogeneous, initial conditions for the singular ODE (2.4): i.e. which augment the $n_+ + n_0$ boundary conditions (2.5) and lead to a well-posed discrete problem for collocation at Gaussian points. This theory is new in that it extends what is given in section 3.1 for first order systems directly to the second order case. This new theory relies on the connections between the quadratic eigenvalue problem (2.6) and the eigenproblem for the $2n \times 2n$ matrix

$$\begin{pmatrix} \mathsf{O} & \mathsf{I} \\ \mathsf{A}_0 & \mathsf{I} + \mathsf{A}_1 \end{pmatrix}, \tag{3.3}$$

where $\mathsf{O} \in \mathbb{R}^{n\times n}$ is the zero matrix, which are developed in [18] and which we make use of here. It is easy to see that the roots of (2.6) and the eigenvalues of (3.3) coincide, but in fact the complete Jordan structures are related. Thus if $X_+ \subseteq \mathbb{R}^{2n}$ is the n_+-dimensional invariant subspace of (3.3) corresponding to the eigenvalues with strictly positive real part, and $\mathsf{U}, \mathsf{U}' \in \mathbb{R}^{n\times n_+}$ so that the columns of

$$\begin{pmatrix} \mathsf{U} \\ \mathsf{U}' \end{pmatrix} \in \mathbb{R}^{2n\times n_+} \tag{3.4}$$

span X_+, then

$$\begin{pmatrix} \mathsf{O} & \mathsf{I} \\ \mathsf{A}_0 & \mathsf{I} + \mathsf{A}_1 \end{pmatrix} \begin{pmatrix} \mathsf{U} \\ \mathsf{U}' \end{pmatrix} = \begin{pmatrix} \mathsf{U} \\ \mathsf{U}' \end{pmatrix} \mathsf{A}_+, \tag{3.5}$$

where $\mathsf{A}_+ \in \mathbb{R}^{n_+\times n_+}$ is the non-singular matrix which expresses the invariance of X_+ with respect to this basis. Equation (3.5) splits into two equations in $\mathbb{R}^n$:

$$\mathsf{U}' = \mathsf{U}\mathsf{A}_+,$$

which shows that the columns of U and U' span the same subspace $\hat{X}_+ \subseteq \mathbb{R}^n$ of dimension $\hat{n}_+ \leq \min\{n_+, n\}$, and

$$\mathsf{U}\mathsf{A}_+^2 - [\mathsf{I} + \mathsf{A}_1]\mathsf{U}\mathsf{A}_+ - \mathsf{A}_0\mathsf{U} = \mathsf{O},$$

which connects with (2.6). Also, if $X_0 \subseteq \mathbb{R}^n$ is the n_0-dimensional null-space of A_0, and the columns of $\mathsf{W} \in \mathbb{R}^{n\times n_0}$ span this subspace, then

$$\begin{pmatrix} \mathsf{O} & \mathsf{I} \\ \mathsf{A}_0 & \mathsf{I} + \mathsf{A}_1 \end{pmatrix} \begin{pmatrix} \mathsf{W} \\ \mathsf{O}' \end{pmatrix} = \begin{pmatrix} \mathsf{O}' \\ \mathsf{O}' \end{pmatrix}, \tag{3.6}$$

with $\mathsf{O}' \in \mathbb{R}^{n\times n_0}$ the zero matrix. Consequently the $n_+ + n_0$ columns of

$$\begin{pmatrix} \mathsf{U} & \mathsf{W} \\ \mathsf{U}' & \mathsf{O}' \end{pmatrix} \tag{3.7}$$

are linearly independent in $\mathbb{R}^{2n}$.

Now we can construct the required $2n - (n_+ + n_0)$ conditions on $\mathbf{z}(0)$ and $\mathbf{z}'(0)$, see [40], from (3.7). Let

$$\mathsf{U}'\Pi = \mathsf{Q}\mathsf{R}$$

be the QR factorisation with column pivoting (cf. [19]) of U': so that Π is a permutation on $\mathbb{R}^{n_+}$,

$$\mathsf{Q} \equiv \begin{pmatrix} \mathsf{Q}_1 & \mathsf{Q}_2 \end{pmatrix} \quad \text{with } \mathsf{Q}_1 \in \mathbb{R}^{n\times \hat{n}_+} \text{ and } \mathsf{Q}_2 \in \mathbb{R}^{n\times(n-\hat{n}_+)}$$

and

$$\mathsf{R} \equiv \begin{pmatrix} \mathsf{R}_1 & \mathsf{R}_2 \\ \mathsf{O}_1 & \mathsf{O}_2 \end{pmatrix} \quad \text{with } \mathsf{R}_1 \in \mathbb{R}^{\hat{n}_+\times \hat{n}_+} \text{ and } \mathsf{R}_2 \in \mathbb{R}^{\hat{n}_+\times(n_+-\hat{n}_+)}$$

plus the zero matrices $\mathsf{O}_1 \in \mathbb{R}^{(n-\hat{n}_+)\times\hat{n}_+}, \mathsf{O}_2 \in \mathbb{R}^{(n-\hat{n}_+)\times(n_+-\hat{n}_+)}$. Thus R_1 is a non-singular upper-triangular matrix. Since the columns of Q_1 span $\hat{X}_+$, we may also write

$$\mathsf{U}\Pi \equiv \mathsf{Q}_1\begin{pmatrix}\mathsf{V}_1 & \mathsf{V}_2\end{pmatrix} \quad \text{with } \mathsf{V}_1 \in \mathbb{R}^{\hat{n}_+\times\hat{n}_+} \text{ and } \mathsf{V}_2 \in \mathbb{R}^{\hat{n}_+\times(n_+-\hat{n}_+)},$$

and hence (3.4) can be expressed as

$$\begin{pmatrix}\mathsf{U}\\ \mathsf{U}'\end{pmatrix}\Pi = \begin{pmatrix}\mathsf{Q}_1\mathsf{V}_1 & \mathsf{Q}_1\mathsf{V}_2\\ \mathsf{Q}_1\mathsf{R}_1 & \mathsf{Q}_1\mathsf{R}_2\end{pmatrix}.$$

A final simple block elimination leads to

$$\begin{pmatrix}\mathsf{U}\\ \mathsf{U}'\end{pmatrix}\Pi\begin{pmatrix}\mathsf{I}_1 & -\mathsf{R}_1^{-1}\mathsf{R}_2\\ \mathsf{O}_1 & \mathsf{I}_2\end{pmatrix} = \begin{pmatrix}\mathsf{Q}_1\mathsf{V}_1 & \mathsf{Q}_1(\mathsf{V}_2 - \mathsf{V}_1\mathsf{R}_1^{-1}\mathsf{R}_2)\\ \mathsf{Q}_1\mathsf{R}_1 & \mathsf{O}\end{pmatrix},$$

where $\mathsf{I}_1 \in \mathbb{R}^{\hat{n}_+\times\hat{n}_+}, \mathsf{I}_2 \in \mathbb{R}^{(n_+-\hat{n}_+)\times(n_+-\hat{n}_+)}$ are identity matrices and $\mathsf{O} \in \mathbb{R}^{n\times(n_+-\hat{n}_+)}$ a zero matrix, and this gives a basis for X_+ in a suitable form. From (3.7), the $n_+ - \hat{n}_+ + n_0$ combined columns of $\mathsf{Q}_1(\mathsf{V}_2 - \mathsf{V}_1\mathsf{R}_1^{-1}\mathsf{R}_2)$ and W are linearly independent in $\mathbb{R}^n$: hence we construct the matrix $\widehat{\mathsf{Q}} \in \mathbb{R}^{n\times(n-[n_+-\hat{n}_+]-n_0)}$, whose columns form an orthonormal basis for the orthogonal complement. Our additional initial conditions on $\mathbf{z}$ are then

$$\widehat{\mathsf{Q}}^T\mathbf{z}(0) = \mathbf{0}. \tag{3.8}$$

If $\hat{n}_+ < n$, we also have the $n - \hat{n}_+$ initial conditions on $\mathbf{z}'$

$$\mathsf{Q}_2^T\mathbf{z}'(0) = \mathbf{0}. \tag{3.9}$$

To conclude, when collocating at Gaussian points, the $n_0 + n_+$ boundary conditions (2.5) are augmented with the $2n - (n_+ + n_0)$ initial conditions (3.8) and (3.9).

If we are using Lobatto points, we need to assume that $\mathbf{z}^\star(t)$ is twice differentiable at $t = 0$ in order to collocate there. This will only be true for generic $\mathbf{g}$, if (2.6) has no roots with real part in the interval $(0, 2]$. (The simple scalar example

$$z''(t) - \frac{1}{t}z'(t) = 2 \qquad \text{with } z(0) = 0, z(1) = 0,$$

having solution $z^\star(t) \equiv t^2 \ln t$, shows that a root equal to 2 is not allowed.) Hence, if necessary, we must implement the change of independent variable (2.7) to satisfy this condition. Once our smoothness constraint holds, we can use the fact that $\mathbf{z}^\star(0)$ is in the kernel of A_0 and $\mathbf{z}^{\star\prime}(0)$ is in the kernel of $\mathsf{A}_1 + \mathsf{A}_0$ (cf. [40]) to write

$$\begin{aligned}\lim_{t\to 0}&\left\{\frac{\mathsf{A}_1}{t}\mathbf{z}^{\star\prime}(t) + \frac{\mathsf{A}_0}{t^2}\mathbf{z}^\star(t)\right\}\\ &= \lim_{t\to 0}\left\{\mathsf{A}_1\frac{\mathbf{z}^{\star\prime}(t) - \mathbf{z}^{\star\prime}(0)}{t} + \mathsf{A}_0\frac{\mathbf{z}^\star(t) - t\mathbf{z}^{\star\prime}(0) - \mathbf{z}^\star(0)}{t^2}\right\}\\ &= (\mathsf{A}_1 + \tfrac{1}{2}\mathsf{A}_0)\mathbf{z}^{\star\prime\prime}(0).\end{aligned}$$

Thus, at the Lobatto point $t = 0$, we should replace (2.4) with the collocation equation

$$(\mathsf{I} - \mathsf{A}_1 - \tfrac{1}{2}\mathsf{A}_0)\mathbf{z}''(0) = \mathbf{g}(0, \mathbf{z}(0)), \tag{3.10}$$

and $\mathsf{I} - \mathsf{A}_1 - \frac{1}{2}\mathsf{A}_0$ is guaranteed to be non-singular.

3.3 Transformation to first order form

It is, of course, possible to transform (2.4) in the standard way into a system of $2n$ first order equations, i.e. by setting

$$\mathbf{y}(t) \equiv (\mathbf{y}_1(t), \mathbf{y}_2(t))^T := (\mathbf{z}(t), \mathbf{z}'(t))^T.$$

This approach, however, destroys all the structure of our singular problem. Thus it is preferable to apply the transformation

$$\mathbf{y}(t) \equiv (\mathbf{y}_1(t), \mathbf{y}_2(t))^T := (\mathbf{z}(t), t\mathbf{z}'(t))^T \tag{3.11}$$

to write the system (2.4) in the first order form (2.1), i.e.

$$\mathsf{M} = \begin{pmatrix} \mathsf{O} & \mathsf{I} \\ \mathsf{A}_0 & \mathsf{I}+\mathsf{A}_1 \end{pmatrix}, \quad \mathbf{f}(t,\mathbf{y}(t)) = \begin{pmatrix} \mathbf{0} \\ t\mathbf{g}(t,\mathbf{y}_1(t)) \end{pmatrix}.$$

(Here we see the key matrix (3.3) appearing again.) The transformation of the boundary conditions (2.5) yields

$$\mathbf{c}(\mathbf{y}_1(0),\mathbf{y}_1(1),\mathbf{y}_2(1)) = \mathbf{0}.$$

Finally we remark that the idea of scaling the independent variable is not new. It has been used for example in [16], [28], and [39]. However what is new in this paper is that we do this scaling in order to make the solution smoother and this, as we will show in section 5, allows the numerical methods that we will propose to perform much more efficiently due to an excellent rate of convergence observed when solving the transformed problem with smoothed solution.

4 Model Problems

In this section we describe the test problems used in our numerical experiments. In Example 1 we compare Gaussian and Lobatto points as well as the first and second order formulations. Example 2 is a demonstration of the effect of the change of the independent variable discussed in Section 2.1. Finally, Example 3 is a very difficult problem that requires the use of continuation for its efficient solution.

4.1 Example 1

We first discuss the scalar ($n=1$) linear problem

$$z''(t) + \frac{1}{t}z'(t) - \frac{\mu^2}{t^2}z(t) = g(t,z(t)), \quad 0 < t \leq 1, \tag{4.1}$$

where

$$g(t,z(t)) \equiv ct^{k-2}e^{-\alpha t}(k^2-\mu^2-\alpha t(1+2k)) + \alpha^2 z(t),$$

subject to the boundary condition

$$z(1) = ce^{-\alpha}. \tag{4.2a}$$

Here, α, $k > 2$ and $\mu > 2$ are parameters, and

$$c \equiv \left(\frac{\alpha}{k}\right)^k e^k.$$

The exact solution of (4.1) is $z(t) = ct^k e^{-\alpha t}$, see Figure 1.

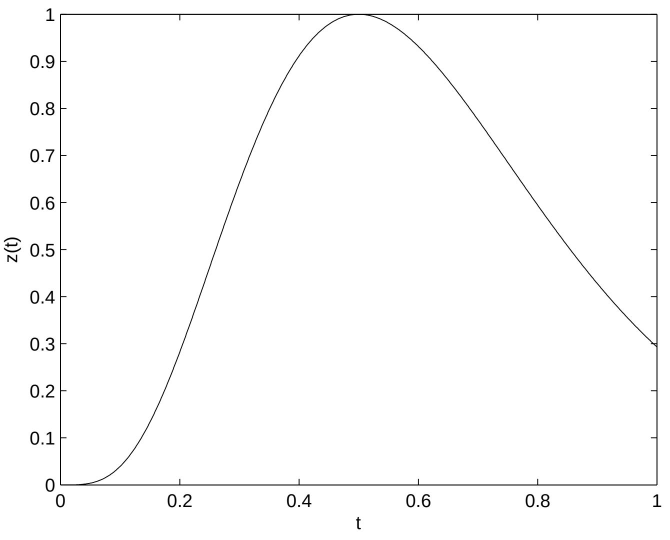

Figure 1: Example 1: Exact solution of (4.1) for $k = 4$ and $\alpha = 8$

For this problem we have

$$\mathsf{A}_1 = -1 \quad \text{and} \quad \mathsf{A}_0 = \mu^2,$$

and so (2.6) is a quadratic polynomial with roots $\lambda = \pm\mu$; hence $n_0 = 0$, $n_+ = 1$, $p = 1$ and we have the correct number of explicit boundary conditions in (4.2a). There is also the implicit boundary condition derived from (3.8), i.e.

$$z(0) = 0. \tag{4.2b}$$

When collocating at Lobatto points, we need to use (3.10) to generate our differential equation at $t = 0$, which in this case becomes

$$\left[2 - \tfrac{1}{2}\mu^2\right] z''(0) = \alpha^2 z(0).$$

If we apply the transformation (3.11) to (4.1), we obtain the related first order system (2.1), where $n = 2$ and

$$\mathbf{y}(t) = \frac{1}{t}\mathsf{M}\mathbf{y}(t) + \mathbf{f}(t, \mathbf{y}(t)), \quad \mathsf{M} = \begin{pmatrix} 0 & 1 \\ \mu^2 & 0 \end{pmatrix}, \; \mathbf{f}(t, \mathbf{y}(t)) = \begin{pmatrix} 0 \\ tg(t, y_1(t)) \end{pmatrix}. \tag{4.3}$$

Here the exact solution is $y_1(t) = ct^k e^{-\alpha t}$ and $y_2(t) = ce^{-\alpha t}(t^{k-1}k - t^k\alpha)t$. Again we have the explicit boundary condition

$$y_1(1) = ce^{-\alpha} \tag{4.4a}$$

and, because X_+ is spanned by $(1, \mu)^T$, the implicit boundary condition (3.1) can be written

$$\mu y_1(0) = y_2(0). \tag{4.4b}$$

When collocating at Lobatto points, we need to use (3.2) to generate our differential equation at $t = 0$, which in this case becomes

$$\mathbf{y}'(0) = \mathbf{0}.$$

4.2 Example 2

Now we consider a first order linear system

$$\mathbf{y}'(t) = \frac{1}{t}\mathsf{M}\mathbf{y}(t) + \mathbf{f}(t) \tag{4.5}$$

with $n = 4$, where

$$\mathsf{M} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & -\frac{9}{4} & -3 \end{pmatrix}, \quad \mathbf{f}(t) = \begin{pmatrix} 0 \\ 0 \\ 6t - 3t^2 \\ 9t + 17t^2 \end{pmatrix}$$

subject to boundary conditions

$$y_1(1) + y_3(1) = -12, \quad y_2(1) + y_4(1) = 31. \tag{4.6a}$$

Thus the exact solution is

$$\mathbf{y}(t) = \begin{pmatrix} -12\sqrt{t} + 2t^2 \\ 18\sqrt{t} + t^3 \\ -6\sqrt{t} + 4t^2 \\ 9\sqrt{t} + 3t^3 \end{pmatrix}.$$

The matrix M has double eigenvalues at $\frac{1}{2}$ and -2: hence $n_0 = 0$, $n_+ = 2$, $p = 2$ and we have the correct number of explicit boundary conditions in (4.6a). There are also the implicit boundary conditions derived from (3.1) and, because X_+ is spanned by

$$\begin{pmatrix} 0 \\ 5 \\ 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 4 \\ -1 \\ 3 \\ -2 \end{pmatrix},$$

these can be written

$$5y_3(0) = 4y_1(0) + y_2(0) \quad \text{and} \quad 20y_4(0) = -9y_1(0) + 4y_2(0). \tag{4.6b}$$

We use this example in Section 5 to show the effect of the transformation of the independent variable in Subsection 2.1: i.e., our transformed problem is

$$\mathbf{y}'(\tau) = \frac{1}{\tau}\widetilde{\mathsf{M}}\mathbf{y}(\tau) + \widetilde{\mathbf{f}}(\tau), \tag{4.7}$$

where

$$\widetilde{\mathsf{M}} = \gamma\mathsf{M}, \quad \widetilde{\mathbf{f}}(\tau) = \gamma\tau^{\gamma-1}\mathbf{f}(\tau^\gamma).$$

Thus the eigenvalues of $\widetilde{\mathsf{M}}$ are $\gamma/2$ and -2γ, while the exact solution of (4.7) is

$$\mathbf{y}(\tau) = \begin{pmatrix} -12\tau^{\gamma/2} + 2\tau^{2\gamma} \\ 18\tau^{\gamma/2} + \tau^{3\gamma} \\ -6\tau^{\gamma/2} + 4\tau^{2\gamma} \\ 9\tau^{\gamma/2} + 3\tau^{3\gamma} \end{pmatrix}.$$

4.3 Example 3

The singular boundary value problem we discuss here originates from the Cahn-Hillard theory, which is used in hydrodynamics to study the behavior of non-homogeneous fluids. In [13], the *density profile equation* for the description of the formation of microscopic bubbles in a non-homogeneous fluid (in particular, vapor inside liquid) is derived. After some simplifications, cf. [26], we arrive at the boundary value problem

$$\rho''(r) + \frac{N-1}{r}\rho'(r) = 4\lambda^2(\rho(r)+1)\rho(r)(\rho(r)-\xi), \tag{4.8}$$

$$\rho'(0) = 0, \quad \rho(\infty) = \xi, \tag{4.9}$$

where ρ is the density of the liquid surrounding the bubble. The problem (4.8), (4.9) depends on 3 parameters: λ, which may be chosen as $\lambda = 1$ without restriction of generality, N which is the dimension of the problem, which in the physically meaningful case is $N = 3$, and ξ, which is varied in the range $[0,1]$ such as to reflect different physical situations. For the numerical treatment, we transform the problem to the finite interval $s \in [0,1]$, by introducing $(z_1(s), z_2(s))^T = (\rho(s), \rho(1/s))^T$ and obtain,

$$z_1''(s) = \frac{1-N}{s} z_1'(s) + 4\lambda^2 (z_1(s)+1) z_1(s)(z_1(s)-\xi), \tag{4.10a}$$

$$z_2''(s) = \frac{1}{s}(N-3) z_2'(s) + \frac{4\lambda^2}{s^4}(z_2(s)+1) z_2(s)(z_2(s)-\xi), \tag{4.10b}$$

subject to boundary conditions

$$z_1'(0) = 0,\ z_2(0) = \xi,\ z_1(1) = z_2(1),\ z_1'(1) = -z_2'(1). \tag{4.11}$$

The above boundary value problem exhibits a singularity of the second kind (due to the $1/s^4$ term) and depends on the real parameter ξ. We solve the problem by applying the path-following strategy implemented in `bvpsuite`. Due to an interior layer, the numerical treatment becomes computationally more challenging when the value of the parameter ξ approaches 1 and it turns out that the implicit form of (4.10b),

$$s^4 z_2''(s) = s^3(N-3) z_2'(s) + 4\lambda^2 (z_2(s)+1) z_2(s)(z_2(s)-\xi)$$

considerably improves the numerical stability. For more details, see [26].

5 Numerical Results

To obtain the numerical results presented in this section, we used the MATLAB code `bvpsuite`, which is described in detail in [25]. The code is based on polynomial collocation and it is capable of solving ordinary differential equations directly in *fully implicit* form and arbitrary order (including zero, corresponding to algebraic equations). In every subinterval we make an ansatz with polynomials represented in the *Runge–Kutta basis* [4] of degree $\le s$ for problems of first order and $\le s+1$ for second order, where s is the number of collocation points used in each subinterval. The collocating function is required to be globally continuous in the first and continuously differentiable in the second case. The convergence behavior of the collocation schemes applied to boundary value problems with a singularity of the first kind have been studied in [21]. Stage order convergence $O(h^s)$ can be shown to hold uniformly in t. However, due to the singularity, the superconvergence orders $O(h^{2s})$ for Gauss and $O(h^{2s-2})$ for Lobatto cannot be guaranteed, in general. The code is capable of adaptive mesh selection as described in [6], based on a posteriori estimates of the global error presented in [5]. Due to the robustness of collocation, this method was used in one of the best established standard FORTRAN codes for (regular) BVPs, COLNEW, see [2] and [3], and in `bvp4c`, `bvp6c`, the standard MATLAB modules for (regular) ODEs with an option for singular problems, cf. [37]. Solvable by the FORTRAN code COLNEW are *explicit* systems of at most order four. The MATLAB code `bvp4c` also solves *explicit* ODE systems and is based on Lobatto collocation. As with `bvp4c`, certain subclasses of singular boundary value problems can be also treated by a newer 'User-Friendly Fortran Code', see [38]. In this paper we do not attempt to compare the above software, but rather concentrate on comparing different algorithmic variants of the collocation mentioned in the introduction.

5.1 Example 1

In the following figures and tables, the performance of different collocation schemes applied to the second order formulation (2.4) and the first order formulation (2.1) is compared. We give numerical results for the parameter values $\mu = 3$, $k = 4$ and $\alpha = 8$; for further results and other choices of μ see [23]. All runs have been carried out in MATLAB (with the floating-point relative machine accuracy of $2^{-52} \approx 2.22 \cdot 10^{-16}$) utilizing adaptive mesh selection aiming at the equidistribution of the global error, see [6]. The relative and absolute tolerance parameters which appear in `bvpsuite` were set to rTOL=aTOL=Tol. For a fair comparison we proceeded as follows. The method whose name is stated in the figure's or table's caption was chosen to determine the final mesh on which its numerical solution satisfied the tolerance requirement. Two other methods shown in tables and figures were then executed on this final mesh with no error control i.e. with no mesh refinement.

One of them has as many collocation points as the leading method, which means the same amount of work, while the other one has the same order of convergence which means the same accuracy.

In Figures 2 and 3 we give two work precision diagrams. Let the final mesh be denoted by

$$\Delta_{final} = \{t_0 = 0 < t_1 < \ldots < t_{k-1} < t_k \ldots < t_{I-1} < t_I = 1\}$$

and the numerical solution at the mesh point t_k by ξ_k. These meshes are obtained using collocation at 4 Gaussian points for Figure 2 and 4 Lobatto points for Figure 3. Then work is measured by $N := sI$, where s is the number of collocation points used in each subinterval $[t_{k-1}, t_k]$, and the precision is defined to be the norm of the known global error,

$$\text{err} = \max_{0\le k\le I} \|\xi_k - \mathbf{z}(t_k)\|_\infty, \quad \text{err} = \max_{0\le k\le I} \|\xi_k - \mathbf{y}(t_k)\|_\infty,$$

for the problem (2.4) and (2.1), respectively. In conclusion the code `bvpsuite` behaves in the following way. For singular BVPs the user can choose the default options of either equidistant or Gauss points (which are already implemented in the code) or else he can specify his own collocation points which must not include zero. For singular problems and Gaussian or equidistant points, the code assumes that the problem is well posed which means that the boundary conditions are posed in a proper way, cf. Subsections 3.1, 3.2, and [20], [40]. Lobatto points can also be implemented but they can be used only for regular problems (which means that the collocation equation at $t = 0$ causes no problems). For the Lobatto runs given in this paper we specify the necessary alternative conditions (replacing the collocation at $t = 0$) by something that is equivalent to (3.10). For Example 1, this condition reads $\mathbf{y}'(0) = \mathbf{0}$. Finally, we note that the code `bvpsuite` can handle problems directly in higher order form or else as first order systems.

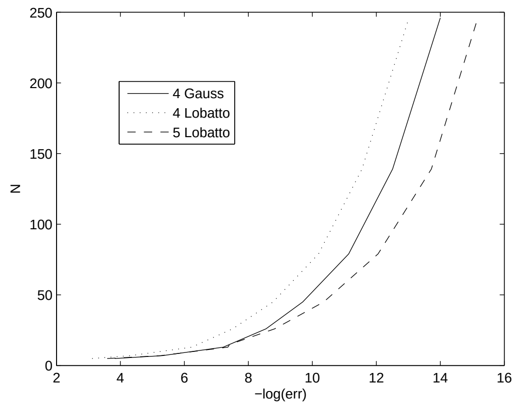

Figure 2: Example 1: Work precision diagram for the second order formulation (4.1). The norm of the absolute global error is plotted versus the number of grid points in the final mesh. Mesh adaptation based on collocation at 4 Gaussian points, starting mesh with 4 equidistant subintervals

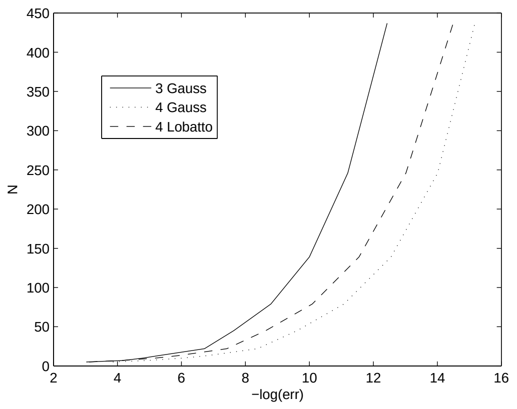

Figure 3: Example 1: Work precision diagram for the second order formulation (4.1). The norm of the absolute global error is plotted versus the number of grid points in the final mesh. Mesh adaptation based on collocation at 4 Lobatto points, starting mesh with 4 equidistant subintervals

Both diagrams show the following behavior for this particular problem. When an s point Gauss collocation is compared to an s point Lobatto method, both requiring the *same amount of work*, the *Gauss method is more accurate* and this is obviously due to the fact that the Gauss method is of higher order. Moreover, when looking at an s point Gauss collocation and comparing it to an $s+1$ point Lobatto method, both having the *same order of convergence*, the *Lobatto method is more accurate*. This observation is supported very well by all models, cf. [23]. Therefore, to avoid repetitions, we do not comment on this fact in the sequel. For further results (mesh adaptation based on 3 Gaussian or 3 Lobatto points) see [23].

The following two Tables 1 and 2 show analogous results to those given in Figures 2 and 3, but now for the first order formulation (4.3).

		4 Gauss		4 Lobatto		5 Lobatto	
Tol	I	err	err.est	err	err.est	err	err.est
10^{-1}	5	5.4e-3	5.9e-3	1.1e-2	1.2e-2	5.4e-3	5.2e-3
10^{-2}	5	5.4e-3	5.9e-3	1.1e-2	1.2e-2	5.4e-3	5.2e-3
10^{-3}	7	5.4e-4	5.4e-4	1.3e-3	1.2e-3	1.8e-4	1.8e-4
10^{-4}	12	3.1e-5	3.1e-5	7.2e-5	7.3e-5	3.6e-6	3.7e-6
10^{-5}	21	6.4e-7	6.6e-7	1.8e-6	1.9e-6	2.0e-7	2.1e-7
10^{-6}	36	2.7e-8	2.9e-8	8.5e-8	9.1e-8	6.2e-9	6.4e-9
10^{-7}	61	1.9e-9	2.1e-9	6.2e-9	6.7e-9	3.9e-10	4.0e-10
10^{-8}	106	1.2e-10	1.3e-10	3.7e-10	4.0e-10	2.5e-11	2.5e-11

Table 1: Example 1, first order formulation: Mesh adaptation based on collocation at 4 Gaussian points, starting mesh with 4 equidistant subintervals. Left subcolumn – norm of the absolute global error; right subcolumn – norm of the error estimate

		3 Gauss		4 Gauss		4 Lobatto	
Tol	I	err	err.est	err	err.est	err	err.est
10^{-1}	5	5.7e-2	6.2e-2	5.4e-3	5.9e-3	1.1e-2	1.2e-2
10^{-2}	7	3.1e-3	3.6e-3	5.6e-4	5.6e-4	1.3e-3	1.3e-3
10^{-3}	10	6.7e-4	7.7e-4	8.2e-5	8.3e-5	1.9e-4	1.9e-4
10^{-4}	15	1.8e-4	2.0e-4	5.4e-6	5.5e-6	1.3e-5	1.3e-5
10^{-5}	25	2.3e-5	2.6e-5	2.1e-7	2.2e-7	6.4e-7	6.8e-7
10^{-6}	43	3.0e-6	3.4e-6	1.3e-8	1.4e-8	4.2e-8	4.5e-8
10^{-7}	76	3.7e-7	4.2e-7	6.8e-10	7.5e-10	2.2e-9	2.4e-9
10^{-8}	133	3.7e-8	4.3e-8	4.0e-11	4.4e-11	1.2e-10	1.3e-10

Table 2: Example 1, first order formulation: Mesh adaptation based on collocation at 4 Lobatto points, starting mesh with 4 equidistant subintervals. Left subcolumn – norm of the absolute global error; right subcolumn – norm of the error estimate

We would like to point out an important difference in running the codes in the first and second order formulation. Let us look at the related figures and tables, Figure 2 and Table 1 or Figure 3 and Table 2. It is easily seen that meshes related to the same tolerances are denser for the first order formulation, especially for stricter tolerance requirements. This is because in the second order formulation only the errors in the solution z are controlled. When working with the first order form however, the errors of both components of y, $y_1 = z$ and $y_2 = tz'$, are to be controlled. To enable a better understanding of this effect we carry out another test, see Tables 3 and 4. Here, we set the tolerance for the component $y_2 = tz'$ to Tol $=1$, so the error in this component is not controlled in practice. Consequently, the adaptation strategy has to take care of $y_1 = z$ only. Still, due to the coupling of y_1 and y_2, the meshes stay denser compared to the second order formulation which clearly indicates that the direct treatment of the second order problem is more efficient, in general. For further results (mesh adaptation based on 3 Gauss or 3 Lobatto points), see [23]. A final remark that we wish to make is that the quality of the error estimate is excellent throughout.

		4 Gauss		4 Lobatto		5 Lobatto	
Tol	I	err	err.est	err	err.est	err	err.est
10^{-1}	5	5.4e-3	5.9e-3	1.1e-2	1.2e-2	5.4e-3	5.2e-3
10^{-2}	5	5.4e-3	5.9e-3	1.1e-2	1.2e-2	5.4e-3	5.2e-3
10^{-3}	7	8.6e-4	8.4e-4	2.0e-3	1.9e-3	3.7e-4	3.7e-4
10^{-4}	10	1.4e-4	1.6e-4	4.2e-4	4.5e-4	1.1e-5	1.2e-5
10^{-5}	13	1.8e-5	2.0e-5	5.7e-5	6.1e-5	2.0e-6	2.0e-6
10^{-6}	21	3.5e-6	3.9e-6	1.0e-5	1.1e-5	1.3e-7	1.3e-7
10^{-7}	36	4.2e-7	4.6e-7	1.3e-6	1.4e-6	6.1e-9	6.3e-9
10^{-8}	63	2.5e-8	2.7e-8	7.5e-8	8.1e-8	3.2e-10	3.3e-10

Table 3: Example 1, first order formulation: Mesh adaptation based on collocation at 4 Gaussian points, starting mesh with 4 equidistant subintervals. Left subcolumn – norm of the absolute global error; right subcolumn – norm of the error estimate, tolerances only prescribed for the first component

		3 Gauss		4 Gauss		4 Lobatto	
Tol	I	err	err.est	err	err.est	err	err.est
10^{-1}	5	5.7e-2	6.2e-2	5.4e-3	5.9e-3	1.1e-2	1.2e-2
10^{-2}	5	5.7e-2	6.2e-2	5.4e-3	5.9e-3	1.1e-2	1.2e-2
10^{-3}	7	2.4e-3	2.7e-3	6.1e-4	6.0e-4	1.4e-3	1.4e-3
10^{-4}	10	5.7e-4	6.5e-4	6.3e-5	6.4e-5	1.5e-4	1.5e-4
10^{-5}	16	1.4e-4	1.6e-4	4.7e-6	5.2e-6	1.5e-5	1.6e-5
10^{-6}	27	1.8e-5	2.1e-5	4.4e-7	4.9e-7	1.3e-6	1.5e-6
10^{-7}	45	2.1e-6	2.4e-6	4.3e-8	4.8e-8	1.3e-7	1.4e-7
10^{-8}	78	3.2e-7	3.6e-7	3.2e-9	3.5e-9	9.7e-9	1.0e-8

Table 4: Example 1, first order formulation: Mesh adaptation based on collocation at 4 Lobatto points, starting mesh with 4 equidistant subintervals. Left subcolumn – norm of the absolute global error; right subcolumn – norm of the error estimate, tolerances only prescribed for the first component

5.2 Example 2

Our aim in considering this example is to experimentally investigate how the transformation $t = \tau^\gamma$ described in Subsection 2.1 influences the convergence order of the collocation scheme and the performance of the mesh adaptation strategy. For illustration purposes we choose $\gamma = 10$ for Example 2 and so the solution of the transformed problem (4.7) is considerably smoother than the solution of the original problem (4.5). All calculations have been carried out with 4 equidistant collocation points and initial mesh with 10 equidistant subintervals. This means that the order of the method being used is 4. The reason that we have chosen to use equidistant collocation points is to ensure that our experiments are not complicated by superconvergence (although we do recognize the fact that superconvergence does not generally occur with singular problems, see [21]). By using equidistant collocation points we can be sure that any changes in the rate of convergence are caused by the eigenstructure of the problem and not by any superconvergence behaviour.

Let us consider a partition

$$\Delta_i := \{t_0 = 0 < t_1 < \ldots < t_{k-1} < t_k \ldots < t_{I-1} < t_I = t_{2^{i+1}} = 1, \}$$

of the interval of integration which consists of $I = 2^{i+1}$ equidistant subintervals, and let us denote the numerical solution at the mesh point k, $k = 0, 1, \ldots 2^{i+1}$ by ξ_i^k. Moreover, let

$$\xi_i := (\xi_i^0, \xi_i^1, \ldots, \xi_i^k, \ldots, \xi_i^{2^{i+1}}), \quad y_{ex} := (\mathbf{y}(t_0), \mathbf{y}(t_1), \ldots, \mathbf{y}(t_k), \ldots, \mathbf{y}(t_{2^{i+1}})),$$

and

$$\|\xi_i - \xi_{i+1}\| := \max_{0 \le k \le 2^{i+1}} \|\xi_i^k - \xi_{i+1}^{2k}\|_\infty, \quad \|\xi_i - y_{ex}\| := \max_{0 \le k \le 2^{i+1}} \|\xi_i^k - \mathbf{y}(t_k)\|_\infty.$$

In our numerical experiments the number of subintervals was doubled at each step and for two consecutive meshes the order of convergence $conv\_est_i$ was estimated. Due to the non-smoothness of the analytical solution of (4.5), we observe a severe order reduction with the order of convergence being 0.5, see Tables 5 and 6.

i	$\|\|\xi_i - \xi_{i+1}\|\|$	$\|\|\xi_i - y_{ex}\|\|$	$conv\_est_i$
1	1.8e-01	6.3e-01	0.50
2	1.3e-01	4.5e-01	0.50
3	9.3e-02	3.1e-01	0.50
4	6.5e-02	2.2e-01	0.50
5	4.6e-02	1.5e-01	—

Table 5: Example 2, first component, equidistant collocation: Left subcolumn – norm of the global error estimate $||\xi_i - \xi_{i+1}||$; middle subcolumn – norm of the global error in comparison to the exact solution; right subcolumn – order of convergence $conv\_est_i$

i	$\|\|\xi_i - \xi_{i+1}\|\|$	$\|\|\xi_i - y_{ex}\|\|$	$conv\_est_i$
1	2.7e-01	9.5e-01	0.50
2	1.9e-01	6.7e-01	0.50
3	1.3e-01	4.7e-01	0.50
4	9.8e-02	3.3e-01	0.50
5	6.9e-02	2.3e-01	—

Table 6: Example 2, second component, equidistant collocation: Left subcolumn – norm of the global error estimate $||\xi_i - \xi_{i+1}||$; middle subcolumn – norm of the global error in comparison to the exact solution; right subcolumn – order of convergence $conv\_est_i$

The results in Tables 7 and 8 are related to the solution of the transformed problem (4.7). Due to the extra smoothness of the solution a much better order of convergence (full stage order) is observed.

i	$\|\|\xi_i - \xi_{i+1}\|\|$	$\|\|\xi_i - y_{ex}\|\|$	$conv\_est_i$
1	3.1e-03	3.4e-03	3.82
2	2.2e-04	2.4e-04	3.96
3	1.4e-05	1.5e-05	3.99
4	9.0e-07	9.6e-07	3.99
5	5.6e-08	6.0e-08	—

Table 7: Example 2, first component, transformed, equidistant collocation: Left subcolumn – norm of the global error estimate $||\xi_i - \xi_{i+1}||$; middle subcolumn – norm of the global error in comparison to the exact solution; right subcolumn – order of convergence $conv\_est_i$

i	$\|\|\xi_i - \xi_{i+1}\|\|$	$\|\|\xi_i - y_{ex}\|\|$	$conv\_est_i$
1	2.4e-02	2.6e-02	3.88
2	1.6e-03	1.7e-03	3.97
3	1.0e-04	1.1e-04	3.99
4	6.5e-06	6.9e-06	3.99
5	4.1e-07	4.3e-07	—

Table 8: Example 2, second component, transformed, equidistant collocation: Left subcolumn – norm of the global error estimate $||\xi_i - \xi_{i+1}||$; middle subcolumn – norm of the global error in comparison to the exact solution; right subcolumn – order of convergence $conv\_est_i$

It is useful here to summarise these results. Due to the lack of smoothness in the solution of the original problem which is due to small positive eigenvalues of M, we observe severe reduction of the convergence order of the collocation scheme. After the transformation, the solution is appropriately smooth and the classical convergence order is observed.

We now test our mesh adaptation algorithm for the transformed problem first and for the original one afterwards. The initial mesh contained 10 equidistant subintervals. The calculations were carried out with 4 equidistant collocation points, and the absolute and relative error tolerances were set to 10^{-4}.

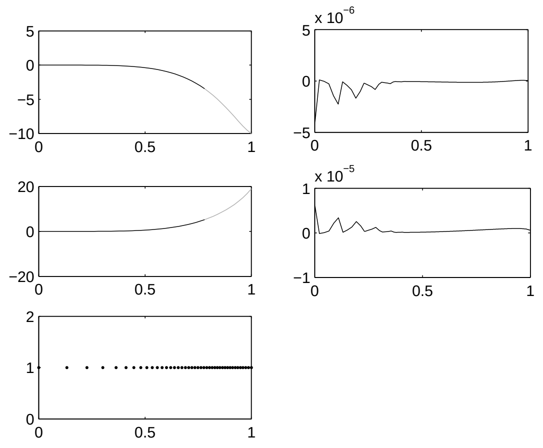

Figure 4: Example 2 (transformed version), 4 equidistant collocation points: Automatically chosen final mesh necessary to satisfy the absolute and relative tolerance requirement 10^{-4}. Left top – down: First and second solution component and the distribution of the meshpoints. Right top – down: Absolute error estimated for the first and second component

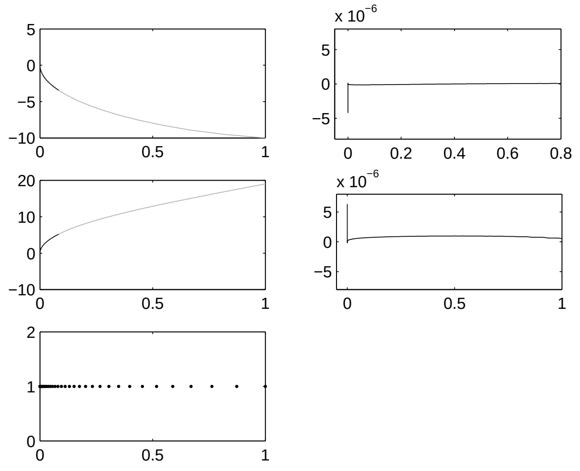

Figure 5: Example 2 (transformed version), 4 equidistant collocation points: Automatically chosen final mesh necessary to satisfy the absolute and relative tolerance requirement 10^{-4} rescaled to the original independent variable. Left top – down: First and second solution component and the distribution of the meshpoints. Right top – down: Absolute error estimated for the first and second component

The transformation does an extremely good job of improving the rate of convergence of our algorithm. The tolerances are satisfied on a mesh containing only 63 meshpoints. Figure 4 shows the solution and the automatically chosen mesh for the system (4.7). When we rescale the plots in Figure 4 back to the original independent variable we obtain Figure 5. The colors black and grey indicate the stretching effect of the transformation.

Finally, we apply our adaptive solution routine to the original system (4.5).

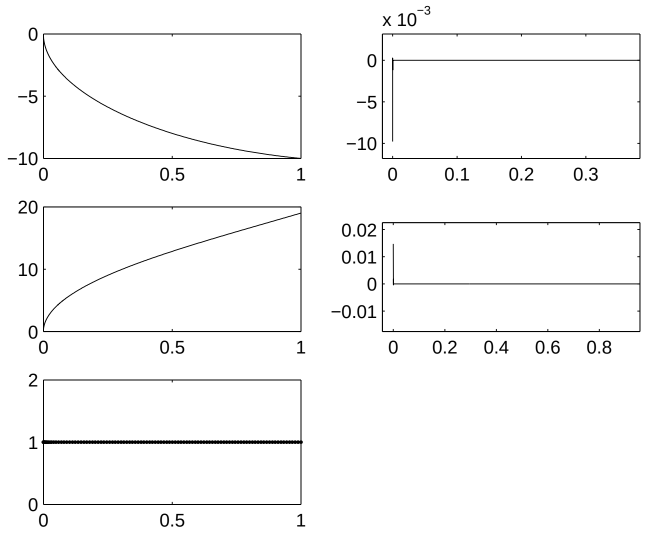

Figure 6: Example 2 (original version), 4 equidistant collocation points: Mesh containing 107 points after 4 refinements, tolerances not satisfied. Left top – down: First and second solution component and the distribution of the meshpoints. Right top – down: Absolute error estimated for the first and second component

Clearly the mesh adaptation algorithm does not work at all well, due to the slow convergence of the method and the incorrect order the mesh adaptation strategy is based on. Our mesh choosing algorithm is implemented on the assumption that the order of convergence is $s = 4$ but in this example the order of convergence is in fact 0.5. The very poor performance of the mesh choosing algorithm can be seen from Figure 6 and we stopped the calculations with a mesh containing around 700 points without reaching the required accuracy.

5.3 Example 3

In this section, we only report on the solution for two different values of the parameter ξ. The numerical solution is shown in Figure 7.

	3 Gauss		4 Gauss		5 Gauss	
Tol	I	err.est	I	err.est	I	err.est
10^{-1}	4	3.0e-3	4	4.4e-4	4	8.5e-5
10^{-2}	4	3.0e-3	4	4.4e-4	4	8.5e-5
10^{-3}	4	3.0e-3	4	4.4e-4	4	8.5e-5
10^{-4}	11	3.4e-5	8	1.2e-6	4	8.5e-5
10^{-5}	20	3.6e-6	8	1.2e-6	8	3.5e-8
10^{-6}	34	2.6e-7	8	1.2e-6	8	3.5e-8
10^{-7}	61	1.5e-8	19	6.5e-8	8	3.5e-8
10^{-8}	108	1.1e-9	31	2.6e-9	20	9.3e-9
10^{-9}	192	7.6e-11	48	4.4e-10	29	1.0e-9
10^{-10}	340	4.6e-12	76	4.3e-11	43	3.8e-11

Table 9: Example 3, $\xi = 0.1$: Numerical solution obtained with mesh adaptation based on collocation at 3, 4, and 5 Gaussian points, starting mesh with 4 equidistant subintervals

	3 Gauss		4 Gauss		5 Gauss	
Tol	I	err.est	I	err.est	I	err.est
10^{-1}	8	5.6e-2	8	1.7e-2	8	1.6e-3
10^{-2}	12	1.2e-2	8	1.7e-2	8	1.6e-3
10^{-3}	21	5.0e-4	13	5.0e-4	8	1.6e-3
10^{-4}	37	5.4e-6	21	1.5e-5	11	2.0e-4
10^{-5}	66	3.2e-7	32	4.1e-7	16	2.4e-6
10^{-6}	117	1.9e-8	51	3.3e-8	23	1.4e-7
10^{-7}	207	1.1e-9	81	2.1e-9	34	8.1e-9
10^{-8}	368	5.8e-11	128	1.4e-10	50	5.7e-10
10^{-9}	654	3.3e-12	202	9.6e-12	73	4.8e-11
10^{-10}	1163	1.9e-13	320	7.2e-13	107	3.5e-12

Table 10: Example 3, $\xi = 0.6$: Numerical solution obtained with mesh adaptation based on collocation at 3, 4, and 5 Gaussian points, starting mesh with 8 equidistant subintervals

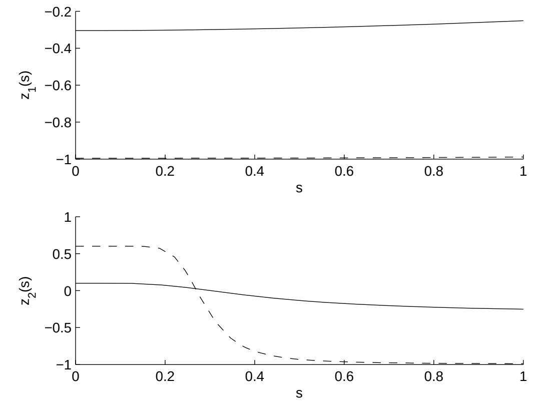

Figure 7: Example 3: $\xi = 0.1$ (solid line), $\xi = 0.6$ (dashed line)

We give the results obtained for $\xi = 0.1$ and 0.6 for illustration purposes. As can be seen from Tables 9 and 10 the algorithm we propose does a very good job of computing solutions with the required degree of accuracy.

6 Conclusions

In this paper, we have presented experimental evidence for some interesting properties of collocation methods when applied to the numerical solution of boundary value problems of first and second order with a singularity of the first kind. Our numerical experiments reveal that collocation at Gaussian points is more efficient than collocation at Lobatto points even though the collocation function for Lobatto points has extra smoothness. If however we consider Gauss and Lobatto formulae of the same order then the Lobatto formulae are more accurate but generally require more computational effort. Furthermore, we have demonstrated that solving our problems in a second order formulation is more efficient than reduction to first order form. This cannot solely be attributed to the approximation of y' which is necessary in the latter case. Instead it is due to the coupling of the components and this was observed even if no tolerance requirement was prescribed for the derivative. For application of the above technique to problems with various types of singular behavior which are relevant in applications see [10], [11] or [35].

Finally, we have proposed a transformation of the independent variable which alters the eigenstructure of the problem being solved. This leads to problems having much smoother solutions and this in turn allows the use of a numerical method with higher order of convergence. The numerical results that we have presented show that this smoothing technique coupled with the algorithms described in this paper constitutes an extremely powerful approach for solving singular problems of the type we have discussed.

It is important to emphasize that the results given in this paper have been computed using a code which is derived from the widely used (readily available) code `bvpsuite` with Gauss collocation applied directly to a higher order system and with a transformation of the independent variable if required. At present we are comparing this code with others that are generally available and we hope to discuss the comparison of codes in a forthcoming paper.

References

[1] F. Al-Musallam, M. Al-Zanaidi and C. Grossmann, *A grid generator for problems on unbounded intervals with rationally decaying solutions*, Computing, 58 (1997), pp. 157–171.

[2] U. Ascher, J. Christiansen and R. Russell, *A collocation solver for mixed order systems of boundary values problems*, Math. Comp., 33 (1978), pp. 659–679.

[3] ——, *Collocation software for boundary value ODEs*, ACM Transactions on Mathematical Software, 7 (1981), pp. 209–222.

[4] U. Ascher, R. Mattheij and R. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[5] W. Auzinger, O. Koch, D. Praetorius and E. Weinmüller, *New a posteriori error estimates for singular boundary value problems*, Numer. Algorithms, 40 (2005), pp. 79–100.

[6] W. Auzinger, O. Koch and E. Weinmüller, *Efficient mesh selection for collocation methods applied to singular BVPs*, J. Comput. Appl. Math., 180 (2005), pp. 213–227.

[7] C. Budd, J. Chen and V. A. Galaktionov, *Focusing blow-up for quasilinear parabolic equations*, Proc. R. Soc. Edinb., 128A (1998), pp. 965–992.

[8] C. Budd, V. A. Galaktionov and J. F. Williams, *Self-similar blow-up in higher-order semilinear parabolic equations*, Preprint 02/10, Dept. Math. Sci., Univ. of Bath, 2002.

[9] C. J. Budd, S. Chen and R. D. Russell, *New self-similar solutions of the nonlinear Schrödinger equation with moving mesh computations*, J. Comput. Phys., 152 (1999), pp. 756–789.

[10] C. J. Budd, O. Koch and E. Weinmüller, *Computation of self-similar solution profiles for the nonlinear Schrödinger equation*, 77 (2006), pp. 335–346.

[11] ——, *From nonlinear PDEs to singular ODEs*, Appl. Numer. Math., 56 (2006), pp. 413–422.

[12] J. Cash and H. Silva, *On the numerical solution of a class of singular two-point boundary value problems*, J. Comput. Appl. Math., 45 (1993), pp. 91–102.

[13] F. Dell'Isola, H. Gouin and G. Rotoli, *Nucleation of spherical shell-like interfaces by second gradient theory: Numerical simulations*, Eur. J. Mech. B/Fluids, 15 (1996), pp. 545–568.

[14] M. Duhoux, *Nonlinear singular Sturm-Liouville problems*, Nonlinear Anal., 38A (1999), pp. 897–918.

[15] M. El-Gebeily and I. Abu-Zaid, *On a finite difference method for singular two-point boundary value problems*, IMA J. Numer. Anal., 18 (1998), pp. 179–190.

[16] R. Fazio, *A novel approach to the numerical solution of boundary value problems on infinite intervals*, SIAM J. Numer. Anal., 33 (1996), pp. 1473–1483.

[17] A. Fink, J. Gatica, G. Hernandez and P. Waltman, *Approxomation of solutions of singular second-order boundary value problems*, SIAM J. Math. Anal., 22 (1991), pp. 440–462.

[18] I. Gohberg, P. Lancaster and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

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

[20] F. de Hoog and R. Weiss, *Difference methods for boundary value problems with a singularity of the first kind*, SIAM J. Numer. Anal., 13 (1976), pp. 775–813.

[21] ——, *Collocation methods for singular boundary value problems*, SIAM J. Numer. Anal., 15 (1978), pp. 198–217.

[22] S. Iyengar and P. Jain, *Spline finite difference methods for singular two-point boundary value problems*, Numer. Math., 50 (1987), pp. 363–376.

[23] G. Kitzhofer, *Numerical Treatment of Implicit Singular BVPs*, Ph.D. Thesis, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria. In preparation.

[24] G. Kitzhofer, O. Koch, P. Lima and E. Weinmüller, *Efficient numerical solution of the density profile equation in hydrodynamics*, J. Sci. Comput., 32 (2007), pp. 411–424.

[25] G. Kitzhofer, O. Koch and E. Weinmüller, *An Efficient General Purpose Code for Implicit ODEs of Arbitrary Order*. In preparation.

[26] ——, *Collocation methods for the computation of bubble-type solutions of a singular boundary value problem in hydrodynamics*, Techn. Rep. ANUM Preprint Nr. 14/04, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria, 2004. Available at `http://www.math.tuwien.ac.at/~inst115/preprints.htm`.

[27] O. Koch and E. Weinmüller, *Analytical and numerical treatment of a singular initial value problem in avalanche modeling*, Appl. Math. Comput., 148 (2004), pp. 561–570.

[28] N. Konyukhova, P. Lima and M. Carpentier, *Asymptotic and numerical approximation of nonlinear singular bounadry value problem*, Tendencias em Matematica Aplicada e Computacional, (2002).

[29] P. Lima, N. Chemetov, N. Konyukhova and A. Sukov, *Analytical-numerical approach to a singular boundary value problem.* Proceedings of CILAMCE XXIV, Ouro Preto, Brasil.

[30] X. Liu, *A note on the Sturmian Theorem for singular boundary value problems*, J. Math. Anal. Appl., 237 (1999), pp. 393–403.

[31] R. März and E. Weinmüller, *Solvability of boundary value problems for systems of singular differential-algebraic equations*, SIAM J. Math. Anal., 24 (1993), pp. 200–215.

[32] D. M. McClung and A. I. Mears, *Dry-flowing avalanche run-up and run-out*, J. Glaciol., 41 (1995), pp. 359–369.

[33] G. Moore, *Computation and parametrization of periodic and connecting orbits*, IMA J. Numer. Anal., 15 (1995), pp. 245–263.

[34] ——, *Geometric methods for computing invariant manifolds*, Appl. Numer. Math., 17 (1995), pp. 319–331.

[35] I. Rachůnková, O. Koch, G. Pulverer and E. Weinmüller, *On a singular boundary value problem arising in the theory of shallow membrane caps*, Math. Anal. and Appl., 332 (2007), pp. 523–541.

[36] P. Rentrop, *Eine Taylorreihenmethode zur numerischen Lösung von Zwei-Punkt Randwertproblemen mit Anwendung auf singuläre Probleme der nichtlinearen Schalentheorie.* TUM-MATH-7733. Technische Universität München 1977.

[37] L. Shampine, J. Kierzenka and M. Reichelt, *Solving Boundary Value Problems for Ordinary Differential Equations in* Matlab *with bvp4c*, 2000. Available at `ftp://ftp.mathworks.com/pub/doc/papers/bvp/`.

[38] L. Shampine, P. Muir and H. Xu, *A User-Friendly Fortran BVP Solver*, Available at `http://cs.smu.ca/~muir/BVP_SOLVER_Files/ShampineMuirXu2006.pdf`, (2006).

[39] J. Tuomela, *A geometric analysis of singular ODE related to the study of quasilinear PDE*, Electron. J. Differential Equations, 62 (2000), pp. 1–6.

[40] E. Weinmüller, *On the boundary value problems for systems of ordinary second order differential equations with a singularity of the first kind*, SIAM J. Math. Anal., 15 (1984), pp. 287–307.

[41] ——, *Testing of error estimation procedures for singular boundary value problems*, CMS Conf. Proc., 8 (1987), pp. 135–152.

Jeff R. Cash
http://www.ma.ic.ac.uk/∼jcash/, j.cash@imperial.ac.uk
Gerald Moore
http://www.ma.ic.ac.uk/∼gmoore/, g.moore@imperial.ac.uk

Imperial College London
South Kensington Campus
Department of Mathematics
London, SW7 2AZ, United Kingdom

Georg Kitzhofer
http://www.math.tuwien.ac.at/georg/, georg.kitzhofer@gmx.at
Othmar Koch
http://www.othmar-koch.org/, othmar@othmar-koch.org
Ewa Weinmüller
http://www.math.tuwien.ac.at/∼ewa/, e.weinmueller@tuwien.ac.at

Vienna University of Technology
Institute for Analysis and Scientific Computing
Wiedner Hauptstrasse 8-10
A-1040 Wien, Austria