

ASC Report No. 09/2009

Numerical Treatment of Singular BVPS: the New MATLAB code bvpsuite

Georg Kitzhofer, Othmar Koch, Ewa Weinmüller

Institute for Analysis and Scientific Computing
Vienna University of Technology — TU Wien
www.asc.tuwien.ac.at ISBN 978-3-902627-05-6

Most recent ASC Reports

- 08/2009 *Irena Rachunkova, Svatoslav Stanek, Ewa Weinmüller, Michael Zenz*
Limit Properties of Solutions of Singular Second-order Differential Equations
- 07/2009 *Anton Arnold, Jose A. Carillo, Chiara Manzini*
Refined Long-Time Asymptotics for some Polymeric Fluid Flow Models
- 06/2009 *Samuel Ferraz-Leite, Christoph Ortner, Dirk Praetorius*
Convergence of Simple Adaptive Galerkin Schemes Based on H-H/2 Error Estimators
- 05/2009 *Gernot Pulverer, Svatoslav Staněk, Ewa B. Weinmüller*
Analysis and Numerical Solutions of Positive and Dead Core Solutions of Singular Sturm-Liouville Problems
- 04/2009 *Anton Arnold, N. Ben Abdallah, Claudia Negulescu*
WKB-based Schemes for the Schrödinger Equation in the Semi-classical Limit
- 03/2009 *Emmanuil H. Georgoulis, Edward Hall, Jens Markus Melenk*
On the Suboptimality of the p-version Interior Penalty Discontinuous Galerkin Method
- 02/2009 *Samuel Ferraz-Leite, Jens Markus Melenk, Dirk Praetorius*
Reduced Model in Thin-film Micromagnetics
- 01/2009 *Markus Aurada, Samuel Ferraz-Leite, Dirk Praetorius*
Convergence of Adaptive Boundary Element Methods
- 38/2008 *Roberta Bosi*
Classical Limit for Linear and Nonlinear Quantum Fokker-Planck Systems
- 37/2008 *Federico Bassitti, Lucia Ladelli, Daniel Matthes*
Central Limit Theorem for a Class of Ome-dimensional Equations

Institute for Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8–10
1040 Wien, Austria

E-Mail: admin@asc.tuwien.ac.at
WWW: <http://www.asc.tuwien.ac.at>
FAX: +43-1-58801-10196

ISBN 978-3-902627-05-6

© Alle Rechte vorbehalten. Nachdruck nur mit Genehmigung des Autors.



NUMERICAL TREATMENT OF SINGULAR BVPS: THE NEW MATLAB CODE `BVPSUITE`

GEORG KITZHOFER, OTHMAR KOCH, AND EWA WEINMÜLLER

ABSTRACT. In this article we consider boundary value problems for systems of ordinary differential equations with singularities. We discuss the analytical properties of such systems and put forward polynomial collocation for their numerical solution. We also discuss further prerequisites necessary for an efficient open domain code — a posteriori error estimation and a grid adaptation strategy. Finally, we present the scope and the performance of our new MATLAB code `bvpsuite` designed to solve singular boundary value problems and index-1 differential algebraic equations.

Boundary value problems – singularity of the first kind – singularity of the second kind – analysis – collocation methods – a posteriori error estimation – mesh adaptation – pathfollowing – blow-up problems

1. INTRODUCTION

We deal with the numerical solution of singular boundary value problems of the form

$$(1a) \quad z'(t) = \frac{M(t)}{t^\alpha} z(t) + f(t, z(t)), \quad t \in (0, 1],$$

$$(1b) \quad B_0 z(0) + B_1 z(1) = \beta,$$

where $\alpha \geq 1$, z is an n -dimensional real function, M is a smooth $n \times n$ matrix and f is an n -dimensional smooth function on a suitable domain. B_0 and B_1 are constant matrices which are subject to certain restrictions for a well-posed problem. (1a) is said to feature a *singularity of the first kind* for $\alpha = 1$, while for $\alpha > 1$ the problem has a *singularity of the second kind*, also commonly referred to as *essential singularity*. The analytical properties of (1) have been discussed in [13, 16]. We will recapitulate the most important results in Section 2, where we focus on the most general boundary conditions which guarantee well-posedness of the problem.

To compute the numerical solution of (1), in Sections 3 and 4 we use polynomial collocation. Our decision to use collocation was motivated by its advantageous convergence properties for (1), while in the presence of a singularity other high order methods show order reductions and become

Date: July 27, 2009.

inefficient [17]. Motivated by these observations, we have implemented two MATLAB codes for singular boundary value problems [4, 22].

For higher efficiency, we also discuss estimation of the global error in Section 5 and adaptive mesh selection in Section 6. Transformation of problems posed on semi-infinite intervals to $[0, 1]$ makes the solution of such problems also accessible by our methods. All these algorithmic components have been integrated into two MATLAB codes. `sbvp` solves explicit first order ODEs [4], while `bvpsuite` can be applied to arbitrary order problems also in implicit formulation and differential algebraic equations [22]. Moreover, a pathfollowing strategy extends the scope of our code, see Section 7. In Section 8, we assess the performance of our implementation by comparisons with other, well-established collocation software. Examples from relevant applications illustrate the achievements possible with our implementation in Section 9.

1.1. Notation. Throughout the paper, the following notation is used. For functions $y \in C[0, 1]$, we define the maximum norm,

$$\|y\| := \max_{0 \leq t \leq 1} |y(t)|.$$

For the numerical analysis, we define meshes

$$(2) \quad \Delta := (\tau_0, \tau_1, \dots, \tau_N),$$

and

$$(3) \quad h_i := \tau_{i+1} - \tau_i, \quad J_i := [\tau_i, \tau_{i+1}], \quad i = 0, \dots, N-1, \quad \tau_0 = 0, \tau_N = 1.$$

For reasons of simplicity, we restrict the discussion to equidistant meshes,

$$h_i = h, \quad i = 0, \dots, N-1.$$

However, the results also hold for nonuniform meshes which have a limited variation in the stepsizes, cf. [14]. On Δ , we define corresponding grid vectors

$$(4) \quad u_\Delta := (u_0, \dots, u_N) \in \mathbb{R}^{(N+1)n}.$$

The norm on the space of grid vectors is given by

$$(5) \quad \|u_\Delta\|_\Delta := \max_{0 \leq k \leq N} |u_k|.$$

For a continuous function $y \in C[0, 1]$, we denote by R_Δ the pointwise projection onto the space of grid vectors,

$$(6) \quad R_\Delta(y) := (y(\tau_0), \dots, y(\tau_N)).$$

For collocation, m points $t_{i,j}$, $j = 1, \dots, m$, are inserted in each subinterval J_i . We choose the same distribution of collocation points in every subinterval, thus yielding the (fine) grid¹

$$(7) \quad \Delta^m = \Delta \cup \{t_{i,j} = \tau_i + \rho_j h, \quad i = 0, \dots, N-1, j = 1, \dots, m\},$$

with

$$(8) \quad 0 < \rho_1 < \rho_2 \cdots < \rho_m \leq 1.$$

For reasons of convenience, we define $\rho_{m+1} := 1$. We restrict ourselves to grids where $\rho_1 > 0$ to avoid a special treatment of the singular point $t = 0$ [11]. For a grid Δ^m , u_{Δ^m} , $\|\cdot\|_{\Delta^m}$ and R_{Δ^m} are defined analogously

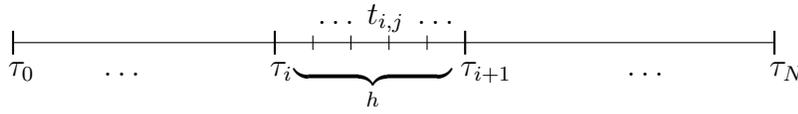


FIGURE 1. The computational grid

to (4)–(6).

2. ANALYTICAL RESULTS

First, we discuss the analytical properties of linear boundary value problems with a singularity of the first kind,

$$(9a) \quad z'(t) = \frac{M(t)}{t}z(t) + f(t), \quad t \in (0, 1],$$

$$(9b) \quad B_0 z(0) + B_1 z(1) = \beta.$$

Throughout, we assume $M \in C^1[0, 1]$. Consequently, we can rewrite $M(t)$ and obtain

$$(10) \quad M(t) = M(0) + tC(t)$$

with a continuous matrix $C(t)$.

It was shown in [13] that the form of the boundary conditions (9b) which guarantee that (9) has a unique, continuous solution depends on the spectral properties of the coefficient matrix $M(0)$. To avoid fundamental modes of (9a) which have the form $\cos(\sigma \ln(t)) + i \sin(\sigma \ln(t))$, we assume that zero is the only eigenvalue of $M(0)$ on the imaginary axis.

¹For convenience, we denote τ_i by $t_{i,0} \equiv t_{i-1,m+1}$, $i = 1, \dots, N$.

Now, let S denote a projection onto the invariant subspace which is associated with eigenvalues of $M(0)$ which have positive real parts, and R a projection onto the kernel of $M(0)$. Finally, define

$$(11) \quad P := S + R, \quad Q := I - P,$$

where I denotes the identity matrix in \mathbb{R}^n .

In [13] it was shown that the boundary value problem (9) is well-posed iff the boundary conditions (9b) can equivalently be written as

$$(12a) \quad (Q + R)z(0) = Rz(0) = \gamma \in \ker(M(0)),$$

$$(12b) \quad Sz(1) = S\eta, \quad \eta \in \mathbb{R}^n.$$

Note that (12a) could also be written equivalently as

$$(13) \quad Qz(0) = 0, \quad \tilde{B}_0 Rz(0) + \tilde{B}_1 Rz(1) = \tilde{\beta},$$

with suitable matrices $\tilde{B}_0, \tilde{B}_1 \in \mathbb{R}^{r \times n}$ and $\tilde{\beta} \in \mathbb{R}^r$. This does not change our arguments, however, so for reasons of simplicity we use (12a).

It follows from the variation of constant formula that, for any $0 < b \leq 1$, every solution z of (9a) satisfies the integral equation

$$(14) \quad z(t) = \left(\frac{t}{b}\right)^{M(0)} z(b) + t^{M(0)} \int_b^t \tau^{-M(0)} (C(\tau)z(\tau) + f(\tau)) d\tau.$$

For sufficiently small b , (14) can be shown to have a unique continuous solution on $[0, b]$, and classical theory yields the existence of a unique solution of (9) on $[0, 1]$.

For the analysis of the nonlinear case, i. e., (1) with $\alpha = 1$, we make the following assumptions:

- (1) Equation (1) has an isolated solution $z \in C[0, 1] \cap C^1(0, 1]$. This means that

$$\begin{aligned} u'(t) &= \frac{M(t)}{t}u(t) + A(t)u(t), \quad t \in (0, 1], \\ B_0u(0) + B_1u(1) &= 0, \end{aligned}$$

where

$$A(t) := \frac{\partial f(t, z)}{\partial z}(t, z(t)),$$

has only the trivial solution. This is equivalent to the local uniqueness of the solution z [18].

With the solution z and a $\rho > 0$ we associate the spheres

$$S_\rho(z(t)) := \{y \in \mathbb{R}^n : |z(t) - y| \leq \rho\}$$

and the tube

$$T_\rho := \{(t, y) : t \in [0, 1], y \in S_\rho(z(t))\}.$$

- (2) $f(t, z)$ is continuously differentiable with respect to z , and $\frac{\partial f(t, z)}{\partial z}$ is continuous on T_ρ .

For this situation, the following smoothness properties hold, for a proof see [13]:

Theorem 1. *Let f be p times continuously differentiable on T_ρ and $M \in C^{p+1}[0, 1]$. Then*

- (1) $z \in C^{p+1}(0, 1]$.
- (2) *If all the eigenvalues of $M(0)$ have nonpositive real parts, then $z \in C^{p+1}[0, 1]$.*
- (3) *Let $\sigma_+ > p$ denote the smallest of the positive real parts of the eigenvalues of $M(0)$ and n_0 the dimension of the largest Jordan box associated with the eigenvalue zero in the Jordan canonical form of $M(0)$. Then the following statements hold:*
 - *For $\sigma_+ < p + 1$, $|z^{(p+1)}(t)| \leq \text{const } t^{\sigma_+ - p - 1} (|\ln(t)|^{n_0 - 1} + 1)$.*
 - *For $\sigma_+ = p + 1$, $|z^{(p+1)}(t)| \leq \text{const } (|\ln(t)|^{n_0} + 1)$.*
 - *For $\sigma_+ > p + 1$, $z \in C^{p+1}[0, 1]$.*

Motivated by the last result, we will assume throughout this paper that $\sigma_+ > 1$ to ensure that $z \in C^1[0, 1]$. Note that if this assumption is not satisfied, we can transform the equation (1a) by letting $t \rightarrow t^\lambda$, $1 > \lambda > 0$, whence $\sigma_+ \rightarrow \sigma_+/\lambda$. Thus, the assumption $\sigma_+ > 1$ imposes no restriction of generality.

Next we formulate the corresponding result for problems with an essential singularity,

$$(15) \quad T(t)z'(t) = f(t, z(t)), \quad t \in (0, 1],$$

$$(16) \quad z \in C[0, 1] \cap C^1(0, 1],$$

$$(17) \quad B_0z(0) + B_1z(1) = \beta,$$

where z is a vector-valued function of dimension n , $B_0, B_1 \in \mathbb{R}^{p \times n}$, $\beta \in \mathbb{R}^p$,

$$(18) \quad T(t) := \text{diag}(t^{\alpha_1} I_1, t^{\alpha_2} I_2, \dots, t^{\alpha_\ell} I_\ell),$$

and the I_k are unit matrices with either $\alpha_k \geq 1$ for $1 \leq k \leq \ell$, or $\alpha_k \geq 1$ for $1 \leq k \leq \ell - 1$ and $\alpha_\ell = 0$. Again, it turns out that (16) is equivalent to $n - p$ linearly independent constraints on $z(0)$, and (17) should provide p additional conditions to ensure local uniqueness of the solution [16].

To formulate smoothness results for nonlinear problems (15)–(17), we first make a number of assumptions:

- (N1): Problem (15)–(17) has a solution $z(t)$. With this solution and some $\rho > 0$ are again associated the spheres $S_\rho(z(t))$ and tube $T_\rho(z)$.

(N2): For some $\rho > 0$, $f(t, z(t))$ is continuously differentiable with respect to z , and $\frac{\partial f}{\partial z}(t, z)$ is continuous on $T_\rho(z)$.

(N3): For all $y \in S_\rho(z(0))$, the matrix

$$M(y) := \frac{\partial f}{\partial z}(0, y)$$

has the fixed block upper triangular structure

$$(19) \quad \begin{pmatrix} M_{11} & M_{12} & \dots & M_{1\ell} \\ 0 & M_{22} & \dots & M_{2\ell} \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & M_{\ell\ell} \end{pmatrix},$$

In addition, for the matrix

$$M := M(z(0)),$$

each submatrix M_{kk} is a square matrix of the same size as I_k and is assumed to be nonsingular when $\alpha_k \neq 0$ and has no eigenvalues that are purely imaginary. When $\alpha_\ell = 0$ then $M_{\ell\ell} = 0$.

(N4): If P corresponds with the same projection as for problems with a singularity of the first kind in (11) (and likewise R used below), then $\text{rank}(P) = p$ from (17).

(N5): The solution $z(t)$ is isolated.

The requirement that the solution of (15) be continuous at $t = 0$ implies the following restriction on the solution:

$$(20) \quad (I - R)f(0, z(0)) = 0.$$

This immediately yields

Lemma 2.1. *Let z satisfy (15) and (16). Then $Tz' \in C[0, 1]$ and*

$$\lim_{t \rightarrow 0^+} T(t)z'(t) = Rz'(0).$$

This lemma says that any component z_i of z in a block associated with $\alpha_k \geq 1$ satisfies $\lim_{t \rightarrow 0^+} t^{\alpha_k} z'_i(t) = 0$, while the components (if any) associated with $\alpha_r = 0$ are in $C^1[0, 1]$. This smoothness result can be extended if further restrictions are imposed on the problem.

Lemma 2.2. *Assume that*

- (1) (N1)–(N5) hold.
- (2) All α_k are integers.
- (3) The real parts of the eigenvalues of the matrix M_{kk} are negative whenever $\alpha_k = 1$.
- (4) $f \in C^m(T_\rho(z))$.

Then, $z \in C^m[0, 1] \cap C^{m+1}(0, 1]$.

3. COLLOCATION METHODS

In this section, we discuss collocation with continuous, piecewise polynomial functions of degree $\leq m$.

Let us denote by B_m the Banach space of continuous, piecewise polynomial functions $q \in \mathbb{P}_m$ of degree $\leq m$, $m \in \mathbb{N}$ (m is called the *stage order* of the method), equipped with the norm $\|\cdot\|$. As an approximation for the exact solution z of (1), we define an element of B_m which satisfies the differential equation (1a) at a finite number of points and which is subject to the same boundary conditions. Thus, we are seeking a function $p(t) = p_i(t)$, $t \in J_i$, $i = 0, \dots, N-1$, in B_m which satisfies

$$(21a) \quad p'(t_{i,j}) = \frac{M(t_{i,j})}{t_{i,j}^\alpha} p(t_{i,j}) + f(t_{i,j}, p(t_{i,j})),$$

$$i = 0, \dots, N-1, j = 1, \dots, m,$$

$$(21b) \quad B_0 p(0) + B_1 p(1) = \beta.$$

We consider collocation on general grids Δ^m (see (7)), subject to the restriction $\rho_1 > 0$.

In [24] the following convergence result was proven for problems with a singularity of the first kind, $\alpha = 1$ in (21a):

Theorem 2. *Assume that $M \in C^{m+2}[0, 1]$, f is $m+1$ times continuously differentiable in $[0, 1] \times \mathbb{R}^n$ with $\frac{\partial f}{\partial z}$ bounded in that domain and $\sigma_+ > m+2$. Then the collocation scheme (21) has a unique solution $p \in B_m$ in a neighborhood of an isolated solution $z \in C^{m+2}[0, 1]$ of (1). This solution can be computed using Newton's method, which converges quadratically. Moreover,*

$$(22) \quad \|p - z\| = O(h^m),$$

$$(23) \quad \left| \frac{M(0)}{t} (p(t) - z(t)) \right| = O(h^m), \quad t \in [0, 1],$$

$$(24) \quad \|p^{(k+1)} - z^{(k+1)}\| = O(h^{m-k}), \quad k = 0, \dots, m-1,$$

$$(25) \quad \left| p'(t) - \frac{M(t)}{t} p(t) - f(t, p(t)) \right| = O(h^m), \quad t \in [0, 1].$$

Note that the condition $\sigma_+ > m+2$ does not impose a restriction of generality, see also the remark following Theorem 1. Furthermore, if $\sigma_+ \leq m+2$ we cannot in general guarantee that $z \in C^{m+2}$ (Theorem 1), and thus we cannot expect to observe the desired convergence orders in this case anyway and the restriction $\sigma_+ > m+2$ is thus natural in this context.

The usual high-order superconvergence at the mesh points does not hold in general for singular problems, however, the uniform superconvergence is preserved (up to logarithmic factors):

$$(26) \quad \|R_\Delta(p) - R_\Delta(z)\|_\Delta = O(h^{m+1} |\ln(h)|^{n_0-1}),$$

$$(27) \quad \|p - z\| = O(h^{m+1} |\ln(h)|^{n_0-1})$$

if

$$(28) \quad \int_0^1 s^k \prod_{l=1}^m (s - \rho_l) ds = 0, \quad k = 0, \dots, \nu - 1.$$

holds with $\nu \geq 1$, see [7], [24] for details.

For problems with an essential singularity, no theoretical results are known for general high-order collocation methods. However, we observed experimentally that the *stage order* $O(h^m)$ is retained for any choice of symmetric collocation points. The superconvergence orders for $\nu \geq 1$ in (28) are

$$(29) \quad \|R_\Delta(p) - R_\Delta(z)\|_\Delta = O(h^{m+\gamma}),$$

$$(30) \quad \|p - z\| = O(h^{m+\gamma}),$$

where $0 < \gamma = \gamma(\alpha) < 1$, and γ decreases with increasing α in (1a). For non-symmetric collocation points, we observed rapid divergence of the numerical solution. Experimental evidence for these propositions is given in [6].

The analysis of the box scheme given in [15] implies that its order of convergence is $1 + \gamma$, where $0 < \gamma < 1$. Since the box scheme is equivalent to collocation at Gaussian points with $m = 1$, this is consistent with the above conjecture.

4. BASIC SOLVER IN THE MATLAB CODE `BVPSUITE`

The code is designed to solve systems of differential equations of arbitrary mixed order including zero², subject to initial or boundary conditions,

$$F(t, p_1, \dots, p_s, z_1(t), z_1'(t), \dots, z_1^{(l_1)}(t), \dots, z_n(t), z_n'(t), \dots, z_n^{(l_n)}(t)) = 0, \quad (31a)$$

$$\begin{aligned} B(p_1, \dots, p_s, z_1(c_1), \dots, z_1^{(l_1-1)}(c_1), \dots, z_n(c_1), \dots, z_n^{(l_n-1)}(c_1), \dots, \\ z_1(c_q), \dots, z_1^{(l_1-1)}(c_q), \dots, z_n(c_q), \dots, z_n^{(l_n-1)}(c_q)) = 0, \end{aligned} \quad (31b)$$

where the solution $z(t) = (z_1(t), z_2(t), \dots, z_n(t))^T$, and the parameters p_i , $i = 1, \dots, s$, are unknown. In general, $t \in [a, b]$, $-\infty < a, b < \infty$ ³.

²This means that differential-algebraic equations are also in the scope of the code.

³For the extension to unbounded domains, see Section 7.

Moreover,

$$F : [a, b] \times \mathbb{R}^s \times \mathbb{R}^{l_1} \times \cdots \times \mathbb{R}^{l_n} \rightarrow \mathbb{R}^n$$

and

$$B : \mathbb{R}^s \times \mathbb{R}^{q_{l_1}} \times \cdots \times \mathbb{R}^{q_{l_n}} \rightarrow \mathbb{R}^{l+s},$$

where $l := \sum_{i=1}^n l_i$. Note that boundary conditions can be posed on any subset of distinct points $c_i \in [a, b]$, $a \leq c_1 < c_2 < \cdots < c_q \leq b$.

For the numerical treatment, we assume that the boundary value problem (31) is well-posed and has a locally unique solution z .

In order to find a numerical solution of (31) we introduce a mesh $[\tau_i, \tau_{i+1}]$, $i = 0, \dots, N-1$, partitioning the interval $[a, b]$ as shown in Figure 1. Every subinterval $[\tau_i, \tau_{i+1}]$ contains m collocation points $t_{i,j}$, $j = 1, \dots, m$. Let \mathbb{P}_m be the space of piecewise polynomial functions of degree $\leq m$, cf. Section 3, which are globally continuous in $[a, b]$. In every subinterval J_i we make an ansatz $P_{i,k} \in \mathbb{P}_{m+l_k-1}$ for the k -th solution component z_k , $k = 1, \dots, n$, of the problem (31). In order to compute the coefficients in the ansatz functions we require that (31) is satisfied exactly at the collocation points. Moreover we require that the collocation polynomial $p(t) := P_i(t)$, $t \in J_i$, is a globally continuous function on $[a, b]$ with components in $C^{l_i-1}[a, b]$, $i = 1, \dots, n$, and that the boundary conditions hold. All these conditions imply a nonlinear system of equations for the unknown coefficients in the ansatz function. For more details see [2] and [20]. It can be easily seen that the number of equations in this nonlinear system amounts to $Nmn + Nl + s$. Every component of the polynomial $P_{i,k}$, $i = 0, \dots, N-1$, $k = 1, \dots, n$ is characterized by $m + l_k$ unknown coefficients, and therefore for every i the polynomial $P_{i,k}$ has $nm + l$ coefficients to be determined. This together with s unknown parameters adds to a total number of unknowns which is also equal to $N(nm + l) + s$.

For the representation of the collocation polynomial p we use the Runge-Kutta basis, see [2], and solve the resulting nonlinear system for the coefficients in this representation by a Newton iteration implemented in the subroutine ‘solve_nonlinear_sys.m’ from the MATLAB code sbvp, cf. [4], which is based on the ‘fast frozen’ Newton method.

5. ERROR ESTIMATES FOR THE GLOBAL ERROR OF THE COLLOCATION

An estimate for the global error of the collocation solution for problems with a singularity of the first kind was proposed and analyzed in [7, 24]: Here, the numerical solution p obtained by collocation is used to define a ‘neighboring problem’ to (1). The original and neighboring problems are

solved by the backward Euler method at all points $t_{i,j}$, $i = 0, \dots, N-1$, $j = 1, \dots, m+1$. This yields the grid vectors $\xi_{i,j}$ and $\pi_{i,j}$ as the solutions of the following schemes, subject to boundary conditions (12),

$$(32a) \quad \frac{\xi_{i,j} - \xi_{i,j-1}}{t_{i,j} - t_{i,j-1}} = \frac{M(t_{i,j})}{t_{i,j}} \xi_{i,j} + f(t_{i,j}, \xi_{i,j}), \quad \text{and}$$

$$(32b) \quad \frac{\pi_{i,j} - \pi_{i,j-1}}{t_{i,j} - t_{i,j-1}} = \frac{M(t_{i,j})}{t_{i,j}} \pi_{i,j} + f(t_{i,j}, \pi_{i,j}) + \bar{d}_{i,j},$$

where $\bar{d}_{i,j}$ is a defect term defined by

$$(33) \quad \bar{d}_{i,j} := \frac{p(t_{i,j}) - p(t_{i,j-1})}{t_{i,j} - t_{i,j-1}} - \sum_{k=1}^{m+1} \alpha_{j,k} \left(\frac{M(t_{i,k})}{t_{i,k}} p(t_{i,k}) + f(t_{i,k}, p(t_{i,k})) \right).$$

Here, the coefficients $\alpha_{j,k}$ are chosen in such a way that the quadrature rules given by

$$\frac{1}{t_{i,j} - t_{i,j-1}} \int_{t_{i,j-1}}^{t_{i,j}} \varphi(\tau) d\tau \approx \sum_{k=1}^{m+1} \alpha_{j,k} \varphi(t_{i,k})$$

have precision $m+1$.

In the next theorem, we state the result that the difference $\xi_{\Delta^m} - \pi_{\Delta^m}$ is an asymptotically correct estimate for the global error of the collocation solution, $R_{\Delta^m}(z) - R_{\Delta^m}(p)$.

Theorem 3. *Assume that the singular boundary value problem (1) with $\alpha = 1$ has an isolated (sufficiently smooth⁴) solution z and satisfies the assumptions of Theorem 2. Then, provided that h is sufficiently small, the following estimate holds:*

$$(34) \quad \| (R_{\Delta^m}(z) - R_{\Delta^m}(p)) - (\xi_{\Delta^m} - \pi_{\Delta^m}) \|_{\Delta^m} = O(|\ln(h)|^{n_0-1} h^{m+1}),$$

with some positive integer n_0 .

The proof is given in [7, 24].

For problems with an essential singularity, it was found that a symmetric auxiliary scheme has to be used, see [5]. Thus, an asymptotically correct estimate can be constructed based on the box scheme. However, in this case (34) is replaced by⁵

$$(35) \quad \| (R_{\Delta^m}(z) - R_{\Delta^m}(p)) - (\xi_{\Delta^m} - \pi_{\Delta^m}) \|_{\Delta^m} = O(h^{m+\gamma}),$$

⁴In fact, we require $z \in C^{m+2}[0, 1]$.

⁵In the case of an essential singularity, we assume that the invariant subspace of \mathbb{R}^n associated with the eigenvalue $\lambda = 0$ coincides with the eigenspace of $\lambda = 0$ and therefore $n_0 = 1$.

where $\gamma < 1$.

Finally, we consider a classical error estimate based on mesh halving. In this approach, we compute the collocation solution at m points on a grid Δ with step sizes h_i and denote this approximation by $p_\Delta(t)$. Subsequently, we choose a second mesh Δ_2 where in every interval J_i of Δ we insert two subintervals of length $h_i/2$. On this mesh, we compute the numerical solution based on the same collocation scheme to obtain the collocating function $p_{\Delta_2}(t)$. Using these two quantities, we define

$$(36) \quad \mathcal{E}(t) := \frac{2^m}{1 - 2^m} (p_{\Delta_2}(t) - p_\Delta(t))$$

as an error estimate for the approximation $p_\Delta(t)$. Assume that the global error $\delta(t) := p_\Delta(t) - z(t)$ of the collocation solution can be expressed in terms of the principal error function $e(t)$,

$$(37) \quad \delta(t) = e(t)h_i^m + O(h_i^{m+1}), \quad t \in J_i,$$

where $e(t)$ is independent of Δ . Then obviously the quantity $\mathcal{E}(t)$ satisfies $\mathcal{E}(t) - \delta(t) = O(h^{m+1})$. This holds for problems with a singularity of the first kind and for regular problems. However, numerical results reported in [5] indicate that in case of an essential singularity (37) reads

$$(38) \quad \delta(t) = e(t)h_i^m + O(h_i^{m+\gamma}), \quad t \in J_i,$$

with the same $\gamma < 1$ as in (35). Generally, estimates of the global error based on mesh halving work well for both problems with a singularity of the first kind and for essentially singular problems [5]. Since they are also applicable to higher-order problems and problems in implicit form (as for example DAEs) without the need for modifications, we have implemented this strategy in our code `bvpsuite`.

6. ADAPTIVE MESH SELECTION

The mesh selection strategy implemented in `bvpsuite` was proposed and investigated in [26]. Most modern mesh generation techniques in two-point boundary value problems construct a smooth function mapping a uniform auxiliary grid to the desired nonuniform grid. In [26] a new system of control algorithms for constructing a grid density function $\phi(t)$ is described. The local mesh width $h_i = \tau_{i+1} - \tau_i$ is computed as $h_i = \epsilon_N / \varphi_{i+1/2}$, where $\epsilon_N = 1/N$ is the accuracy control parameter corresponding to $N - 1$ interior points, and the positive sequence $\Phi = \{\varphi_{i+1/2}\}_{i=0}^{N-1}$ is a discrete approximation to the continuous density function $\phi(t)$, representing the mesh width variation. Using an error estimate, a feedback control law generates a new density from the previous

one. Digital filters may be employed to process the error estimate as well as the density.

For boundary value problems, an adaptive algorithm must determine the sequence $\Phi^{[\nu]}$ in terms of problem or solution properties. True adaptive approaches equidistribute some *monitor function*, a measure of the residual or error estimate, over the interval. As $\Phi^{[\nu]}$ will depend on the error estimates, which in turn depend on the distribution of the grid points, the process of finding the density becomes *iterative*. For some error control criteria a local grid change typically has global effects. The techniques developed here avoid this difficulty by restricting the error estimators to those having the property that the estimated error on the interval $[\tau_i, \tau_{i+1}]$ only depends on the local mesh width, $h_i = \epsilon_N / \varphi_{i+1/2}$.

In order to be able to generate the mesh density function, we decided to use the residual $r(t)$ to define the monitor function. The values of $r(t)$ are available from the substitution of the collocation solution $p(t)$ into the analytical problem (31). We first compute

$$R_k(\tau_{i+1/2}) = \int_{\tau_i}^{\tau_{i+1}} r_k(t) dt \approx \frac{r_k(\tau_i) + r_k(\tau_{i+1})}{2} (\tau_{i+1} - \tau_i)$$

for $i = 0, \dots, N - 1$ and for each component r_k , $k = 1, \dots, n$, of the residual r . Now, for each subinterval J_i , we calculate

$$\hat{R}(\tau_{i+1/2}) := \left(\sum_{k=1}^n R_k^2(\tau_{i+1/2}) \right)^{\frac{1}{2}}, \quad i = 0, \dots, N - 1,$$

to obtain the monitor function necessary for the update of $\Phi^{[\nu]}$.

While the residual based monitor function $R(t) := \hat{R}(\tau_{i+1/2})$, $t \in J_i$, is used to update the mesh density, the number of the necessary mesh points in the final grid is determined from the requirement that the absolute global error satisfies the tolerance. The mesh halving routine provides the values of the error estimate (36) in the entire interval $[a, b]$, so we can compute

$$G_{\Delta^m} := \max_t (\max_{1 \leq k \leq n} |\mathcal{E}_k(t)|), \quad t \in \Delta^m.$$

The number of points for the next iteration step is predicted from

$$(39) \quad N_{\nu+1} = M \left(\frac{G_{\Delta^m}}{0.9\text{TOL}} \right)^{1/(m+1)},$$

where $M = 50$ is the fixed number of points in the control grid. Below, we specify in more detail the grid adaptation routine implemented in the code.

- (1) Grid generation, finding the optimal density function, is separated from mesh refinement, finding the proper number of mesh points. We first try to provide a good density function Φ on a rather coarse mesh with a fixed number of points $M = 50$. The mesh density function is chosen to equidistribute the monitor function $R(t)$.
- (2) For each density profile in the above iteration, we estimate the number of mesh points necessary to reach the tolerance, according to (39).
- (3) The calculation of the density function is terminated when $N_{\nu+1} > 0.9N_\nu$. Clearly, it can be expected that in the course of the optimization of the density function the number of the associated mesh points will monotonically decrease. This process is stopped when the next density profile $\Phi^{[\nu+1]}$ would result in saving less than 10% of the mesh points compared to the current density profile $\Phi^{[\nu]}$.
- (4) Since the calculation of a residual is reasonably cheap we always update the density profile to make use of the information provided by the most recent available numerical solution associated with the function $\Phi^{[\nu]}$.
- (5) We finally solve the problem on the mesh based on $\Phi^{[\nu+1]}$ with $N^{[\nu+1]}$ mesh points, and estimate the global error of this approximation. If the accuracy requirement is satisfied, we stop the calculations, otherwise we refine again.

For more details and the results of numerical tests, we refer the reader to [26].

7. PATHFOLLOWING AND PROBLEMS ON SEMI-INFINITE INTERVALS

In this and in the following sections, we discuss the scope of `bvpsuite` and its special features which allow to cover a very wide range of applications. First of all, our code realizes a pathfollowing strategy to follow solution branches in dependence of a known parameter. To describe the strategy in general terms, we consider (1a), (1b) as a parameter-dependent operator equation

$$(40) \quad F(y; \lambda) = 0,$$

where $F : Y \times \mathbb{R} \rightarrow Z$, and Y, Z are Banach spaces (of possibly infinite dimension).

Pathfollowing in this general setting has been discussed in detail in [34].

We are particularly interested in computing solution branches Γ with *turning points*. By definition, in a turning point the solution of (40) constitutes a local maximum (or minimum) of λ , and consequently is not locally unique as a function of the parameter λ . The situation is illustrated in

Figure 2. There, we plot some functional of the solution against the parameter λ . The arrows indicate the turning points. Thus, in a turning point we cannot parametrize Γ as a function of λ . However, it is sufficient for our procedure that a tangent is uniquely determined at all points of Γ . This is guaranteed by realistic assumptions formulated for our problem in [23].

Now, we proceed by describing our pathfollowing strategy. As explained in [23], our assumptions on the problem ensure that at a point $(y_0, \lambda_0) \in \Gamma$, a tangent can be uniquely determined up to the sign. Additional criteria determine how to choose the direction. On the tangent just computed, a predictor (y_P, λ_P) is chosen for the computation of the next point on Γ , and finally a corrector equation is solved yielding (y_C, λ_C) . One step of our procedure starting at (y_0, λ_0) is illustrated in Figure 2.

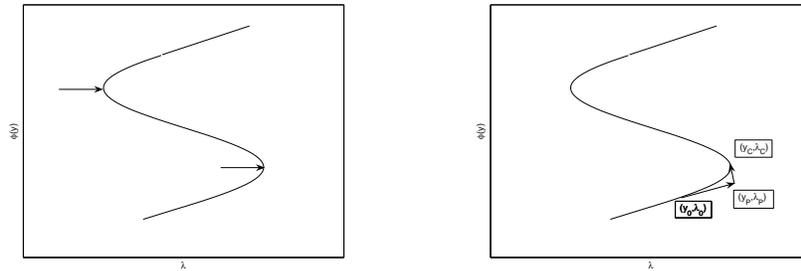


FIGURE 2. A solution branch with two turning points (left), one step of the pathfollowing procedure (right).

As one example to demonstrate that our pathfollowing strategy indeed works for singular boundary value problems and generates meshes adapted to the solution profile, in [23] we considered an example from [12], describing the buckling of a spherical shell.

We followed the solution path Γ shown in Figure 3, starting at $\lambda = 0$. Figure 3 shows the maximum norm of the first solution component, $\|\beta\|_\infty$ along the path Γ . The crosses indicate points of Γ where the solution profiles of β and the second solution component Ψ are plotted in Figure 4, together with the meshes generated by our adaptive mesh selection procedure. A comparison with [12, Figure 10] shows that the solution is computed reliably and obviously the meshes are denser where the solution varies more rapidly.

Our code can also treat problems posed on semi-infinite intervals $t \in (a, \infty)$, $a > 0$ (and by a splitting of the interval, also $a = 0$). In order to exploit our efficient and robust mesh selection strategy also in this case, we

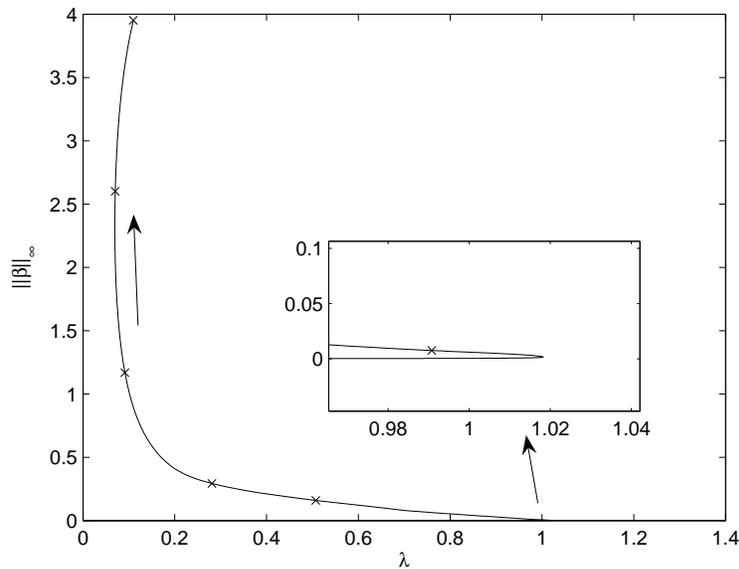


FIGURE 3. Values of $\|\beta\|_\infty$ along a solution branch.

use the transformation $t = \frac{1}{\tau}$, $z(t) = x\left(\frac{1}{\tau}\right)$ to restate

$$x'(\tau) = \tau^\beta f(\tau, x(\tau)), \quad \tau \in [a, \infty), \quad \beta > -1$$

as

$$z'(t) = -\frac{1}{t^{\beta+2}} f(1/t, z(t)), \quad t \in (0, 1/a].$$

This is in general a problem with an essential singularity, which however is in the scope for our collocation methods, error estimation procedure and adaptive mesh refinement. In this approach, the mesh is adapted only according to the unsmoothness of the solution without the need for mesh grading on long intervals, and moreover no truncation of the unbounded interval is necessary. This strategy was employed successfully for example in [9], [10] and [21].

8. CODE PERFORMANCE

In this section, we comment on the performance of our code `bvpsuite` when compared to other available software for the numerical solution of boundary value problems in ordinary differential equations. Since our focus is on singular boundary value problems, we have chosen those codes which explicitly claim that singular problems are in their scope. Therefore, we take into considerations the standard MATLAB code `bvp4c` and its higher order versions `bvp5c`, `bvp6c`, cf. [29], [19], and two FORTRAN

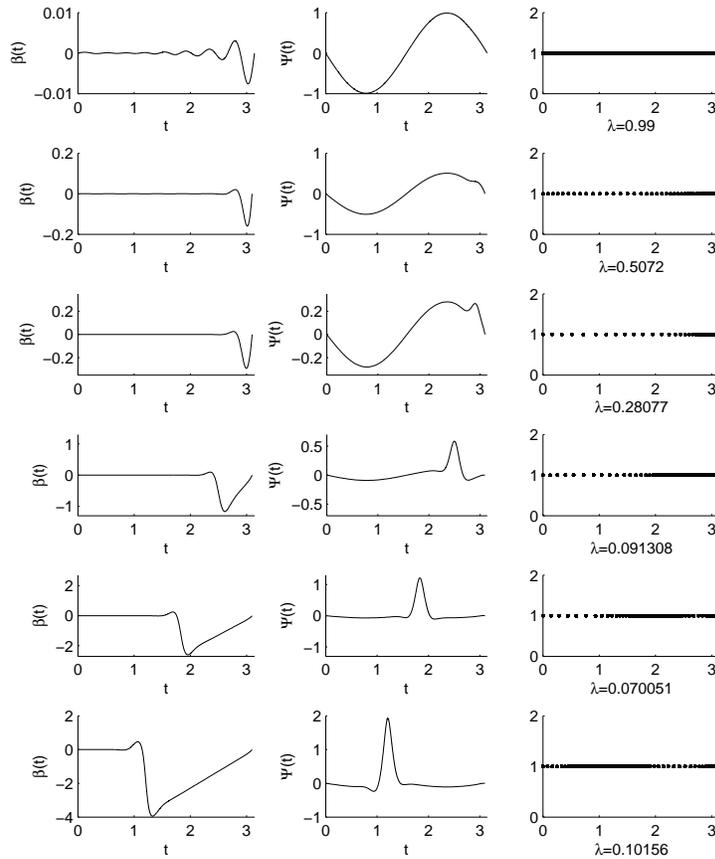


FIGURE 4. Solution profiles and automatically selected meshes at the points marked in Figure 3 along the solution branch.

codes, A User-Friendly Fortran BVP Solver [30] and the COLNEW solver described in [1].

Our main intention while designing `bvpsuite` was to provide a MATLAB code which can cope with a wide range of applications and works dependably and efficiently for a large range of tolerances with emphasis on high-precision solution. Therefore, we have chosen the *fully implicit formulation* of the nonlinear system of equations and nonlinear boundary conditions, see Section 4. The order of the differential equations in the components of the system can be arbitrary and different for different

components. Thus, there is no need to transform a higher order system to its first order form. The code can cope with free unknown parameters for which the appropriate number of additional boundary conditions need to be specified at the borders or within the interval of integration. In its scope are nonlinear singular boundary value problems with a singularity of the first or of the second kind⁶. Over the years, we have been able to give a good theoretical justification for all components of the code, also in the context of singular problems, see for instance the list of publications at <http://www.math.tuwien.ac.at/~ewa>.

The code can solve index 1 differential-algebraic equations, a coupled system of differential equations and algebraic constraints. Also, it is equipped with a pathfollowing strategy in case of known parameter values such that the turning points in the solution-parameter path do not constitute a difficulty. Recently, we have equipped the code with modules for the solution of eigenvalue problems of first and second order, see the references below. Moreover, for a problem posed on a semi-infinite interval $[0, \infty)$ the code automatically reduces the problem to the interval $[0, 1]$ and after numerical computations it provides the approximate solution on an arbitrary interval $[0, b]$, $b < \infty$.

The order of the collocation solver is chosen automatically in dependence of the tolerance specified by the user and varies between two and eight. We stress that since in our code Gaussian points (or equidistant interior collocation points) are used, we avoid the evaluation at the singular point and therefore also in the case of singular problems only *one numerical method on the whole interval* is used and no pre-handling is necessary. In other words, there is no distinction between the solution of singular or regular problems with `bvpsuite`. The error estimate and the grid adaptation routine have been described in Sections 5 and 6, respectively

The code `bvp4c`⁷ [28] can solve *explicit nonlinear systems of order one* with nonlinear boundary conditions and unknown parameters. However, the singular problems have to show a special structure,

$$(41) \quad z'(t) = \frac{S}{t}z(t) + f(t, z(t)), \quad t \in (0, T],$$

with a constant matrix S . This means that only a singularity of the first kind in this particular form is in the scope of the code. The basic solution method is based on polynomial collocation with four, five or six Lobatto points, respectively. Within one routine the order of the method is fixed to

⁶and clearly, problems with no singularity

⁷In the following, we refer to `bvp4c` only, even when all three variants of the code are addressed.

four, five or six. The quantity to be estimated and controlled is the residual, and residual and error in case of `bvp5c` [19].

The `User-Friendly Code` [30] covers the same class of problems, regular and singular, as `bvp4c`. The methods used here are implicit Runge-Kutta schemes (MIRKDC) of orders two, four, and six. The code controls the defect in the differential equations and boundary conditions and also provides an estimate for the global error using the extrapolation technique.

Finally, COLNEW [1] can solve *explicit* nonlinear systems of ordinary differential equations of varying order up to four. The basic solver is collocation based on Gaussian points whose number ranges from one to seven. The code controls the global error estimated from the mesh halving principle which in case of Gaussian points is strongly related to the residual. In this code a pathfollowing strategy is also available and the code can cope with free parameters.

We compare the performance of the codes by solving the following boundary value problem:

$$z'(t) = \frac{1}{t} \begin{pmatrix} 0 & 1 \\ 2 & 6 \end{pmatrix} z(t) + \begin{pmatrix} 0 \\ 4k^2t^5 \sin(k^2t^2) + 10t \sin(k^2t^2) \end{pmatrix}, \quad (42a)$$

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} z(0) + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} z(1) = \begin{pmatrix} 0 \\ \sin(k^2) \end{pmatrix}, \quad (42b)$$

where the analytical solution is known,

$$z(t) = (t^2 \sin(k^2t^2), 2k^2t^4 \cos(k^2t^2) + 2t^2 \sin(k^2t^2))^T.$$

For the simulation the order of the methods implemented in the codes was set to four. Additionally, we also used the possibility of our code to adapt the order of the method to the prescribed tolerance, see curve ‘`bvpsuite vo`’ in Figures 5 and 6. The results show that this approach provides the most efficient solution method. Therefore the flexibility of our code also constitutes a significant improvement of the performance. The model problem (42) discussed here, gives a typical picture observed in all tests. The results of the simulation can be found in Figures 5 and 6. When the order is fixed to four for comparison, the number of grid points required by `bvpsuite` and COLNEW is comparable and smaller throughout than for `bvp4c` and the `User-Friendly Code`. Especially, for strict tolerances, the gap between `bvpsuite` and COLNEW and the other two codes is significant. The strictest tolerance successfully reached by the `User-Friendly Code` was 10^{-10} , and by `bvp4c` 10^{-11} , while `bvpsuite` and COLNEW reached an accuracy of 10^{-13} . It is worth mentioning that all codes work very well at target, which means that the tolerance and accuracy are closely related.

9. APPLICATIONS

As already mentioned, eigenvalue problems, see [3], [31], and differential algebraic equations, cf. [8], [25], are within the scope of our code, but it can also be applied in case of non-standard singularities. In [27], we investigated the following singular equation which originates from the theory of shallow membrane caps,

$$(43) \quad (t^3 u'(t))' + t^3 \left(\frac{1}{8u^2(t)} - \frac{a_0}{u(t)} - b_0 t^{2\gamma-4} \right) = 0, \quad t \in (0, 1],$$

subject to boundary conditions

$$\lim_{t \rightarrow 0^+} t^3 u'(t) = 0, \quad u(1) = 0,$$

where a_0 , b_0 , and γ are given constants. Note that this problem has a more challenging structure than (1). After rewriting (43), we obtain the explicit version of the equation,

$$(44) \quad u''(t) + \frac{3}{t} u'(t) + \left(\frac{1}{8u^2(t)} - \frac{a_0}{u(t)} - b_0 t^{2\gamma-4} \right) = 0, \quad u(1) = 0.$$

Here, a singularity of the first kind occurs at $t = 0$, but at the same time due to the boundary condition at $t = 1$ the problem has a so-called *phase singularity* at the other end of the interval. For such more involved problems existence and uniqueness of solutions is shown by means of generalized lower and upper functions, involving limiting processes, cf. [27] and references therein. Our code `bvpsuite` could be used to approximate solutions⁸ of the membrane problem. However, a theoretical justification for the collocation method in view of the problem structure is still an open question.

Another source of challenging problems with an interesting solution structure are reaction-diffusion equations, see [32], [33]. In [32], the simple looking, parameter dependent problem of the form

$$u''(t) = \frac{\lambda}{\sqrt{u(t)}}, \quad t \in (0, 1], \quad u(0) = u(1) = 1,$$

where λ is a given parameter, turns out to have a very challenging structure. Depending on the value of λ there exist the so-called positive solutions, $u(t) > 0$ for all $t \in [0, 1]$, pseudo dead core, and dead core solutions, such that $u(t) = 0$ for a certain point $t \in (0, 1)$, or $u(t) = 0$ for a certain subinterval $t \in [\alpha, \beta]$, $0 < \alpha < \beta < 1$, respectively. In order to find the latter two solutions, we simulated the problem numerically using `bvpsuite`. Here,

⁸even though $u'(0)$ may become unbounded

we utilized the fact that the above equation can be treated in its fully implicit form,

$$u''(t)\sqrt{u(t)}u(t) = \lambda u(t), \quad t \in (0, 1], \quad u(0) = u(1) = 1.$$

Clearly, in the case that the analytical problem is especially involved, the numerical approach may sometimes constitute the only source of information about the solution structure. We faced this type of difficulty in [33]. Since the problem is again parameter dependent, we have applied the path-following strategy implemented in `bvpsuite` to solve

$$((u'(t))^3)' + \frac{u'(t)}{t^2} = \lambda \left(\frac{1}{\sqrt{u(t)}} + (u'(t))^2 \right), \quad t \in (0, 1), \quad (45a)$$

$$u'(0) = 0, \quad 0.1u(1) + u'(1) = 1. \quad (45b)$$

The results of this simulation are shown in Figure 9. We can see that for a certain range of λ the positive solution is unique, and for the other part of the path, we could find two different positive solutions, see Figures 7 and 8. According to Figure 9, we have moved around a turning point at $\lambda \approx 1.8442$. Finally, in the last step of the procedure, we obtained a solution which nearly reaches a pseudo dead core solution with the collocation solution $p(0) \approx u(0) \approx 0$.

10. CONCLUSIONS

In this paper we gave an overview of the very intense activities carried out for many years at Vienna University of Technology and focused on the analysis, numerical solution and code development for singular boundary value problems in ordinary differential equations, differential algebraic equations, and problems posed on semi-infinite intervals.

When analyzing singular problems, we first note that their direction field is very unsmooth, especially close to the singular point. Consequently, we can encounter unbounded contributions to the solution manifold, such that $z \in C(0, 1]$. However, irrespective of the spectrum of the matrix $M(0)$, by posing proper homogeneous initial conditions, we can extend the above solution to $z \in C[0, 1]$. It also turns out that in such a case the condition $M(0)z(0) = 0$ must hold. For singular problems the solution's smoothness depends not only on the smoothness of the inhomogeneity f but also on the size of real parts of the eigenvalues of $M(0)$.

Concerning the numerical treatment of singular problems one usually assumes that the underlying analytical problem is well-posed and has a smooth solution. On the basis of such an assumption, one would like to

design a high order method, and error estimate and grid adaptation strategies, which remain unaffected by the steep direction field. This means that the grids should become dense only in the regions when the solution is unsmooth. Especially they should stay coarse close to the singularity, when the solution is smooth there. It turns out that collocation at Gaussian (or inner equidistant) points remains robust for singular problems and can serve as a dependable solver in the code design, while other high order methods suffer from order reductions. Also, defect correction and mesh halving principles constitute a reliable basis for the a posteriori error estimation. We have put a lot of effort in the grid adaptation strategy. Here, the main idea is to split the adaptation of the grid density and the number of grid points necessary to satisfy the tolerance requirements. This idea has proven very fruitful and results in grids which in a very satisfactory way reflect the solution behavior.

Finally, we introduced and described in detail our new MATLAB solver `bvpsuite` and demonstrated that concerning the scope and efficiency it is a very competitive candidate among the available software for singular boundary value problems.

REFERENCES

- [1] U. Ascher, J. Christiansen, and R.D. Russell. A collocation solver for mixed order systems of boundary values problems. *Math. Comp.*, 33:659–679, 1978.
- [2] U. Ascher, R.M.M. Mattheij, and R.D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [3] W. Auzinger, E. Karner, O. Koch, and E.B. Weinmüller. Collocation methods for the solution of eigenvalue problems for singular ordinary differential equations. *Opuscula Math.*, 26(2):229–241, 2006.
- [4] W. Auzinger, G. Kneisl, O. Koch, and E. Weinmüller. A collocation code for boundary value problems in ordinary differential equations. *Numer. Algorithms*, 33:27–39, 2003.
- [5] W. Auzinger, O. Koch, D. Praetorius, and E. Weinmüller. New a posteriori error estimates for singular boundary value problems. *Numer. Algorithms*, 40:79–100, 2005.
- [6] W. Auzinger, O. Koch, and E. Weinmüller. Collocation methods for boundary value problems with an essential singularity. In I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, volume 2907 of *Lecture Notes in Computer Science*, pages 347–354. Springer Verlag, 2004.
- [7] W. Auzinger, O. Koch, and E. Weinmüller. Analysis of a new error estimate for collocation methods applied to singular boundary value problems. *SIAM J. Numer. Anal.*, 42(6):2366–2386, 2005.
- [8] W. Auzinger, H. Lehner, and E. Weinmüller. Defect-based a posteriori error estimation for index-1 DAEs. Submitted to *Numer. Math.*
- [9] C. J. Budd, O. Koch, and E. Weinmüller. Computation of self-similar solution profiles for the nonlinear Schrödinger equation. *Computing*, 77:335–346, 2006.

- [10] C. J. Budd, O. Koch, and E. Weinmüller. From nonlinear PDEs to singular ODEs. *Appl. Numer. Math.*, 56:413–422, 2006.
- [11] J. Cash, G. Kitzhofer, O. Koch, G. Moore, and E. Weinmüller. Numerical solution of singular two-point boundary value problems. To appear in *JNAIAM J. Numer. Anal. Indust. Appl. Math.*
- [12] M. Gräff, R. Scheidl, H. Troger, and E. Weinmüller. An investigation of the complete post-buckling behavior of axisymmetric spherical shells. *ZAMP*, 36:803–821, 1985.
- [13] F.R. de Hoog and R. Weiss. Difference methods for boundary value problems with a singularity of the first kind. *SIAM J. Numer. Anal.*, 13:775–813, 1976.
- [14] F.R. de Hoog and R. Weiss. Collocation methods for singular boundary value problems. *SIAM J. Numer. Anal.*, 15:198–217, 1978.
- [15] F.R. de Hoog and R. Weiss. The numerical solution of boundary value problems with an essential singularity. *SIAM J. Numer. Anal.*, 16:637–669, 1979.
- [16] F.R. de Hoog and R. Weiss. On the boundary value problem for systems of ordinary differential equations with a singularity of the second kind. *SIAM J. Math. Anal.*, 11:41–60, 1980.
- [17] F.R. de Hoog and R. Weiss. The application of Runge-Kutta schemes to singular initial value problems. *Math. Comp.*, 44:93–103, 1985.
- [18] H. Keller. Approximation methods for nonlinear problems with application to two-point boundary value problems. *Math. Comp.*, 29:464–474, 1975.
- [19] J. Kierzenka and L. Shampine. A BVP solver that controls residual and error. *JNAIAM J. Numer. Anal. Indust. Math.*, 3:27–41, 2008.
- [20] G. Kitzhofer. *Numerical Treatment of Implicit Singular BVPs*. Ph.D. Thesis, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria, 2009. In preparation.
- [21] G. Kitzhofer, O. Koch, P. Lima, and E. Weinmüller. Efficient numerical solution of the density profile equation in hydrodynamics. *J. Sci. Comput.*, 32:411–424, 2007.
- [22] G. Kitzhofer, O. Koch, and E. Weinmüller. An Efficient General Purpose Code for Implicit ODEs of Arbitrary Order. In preparation, 2009.
- [23] G. Kitzhofer, O. Koch, and E. Weinmüller. Pathfollowing for essentially singular boundary value problems with application to the complex Ginzburg–Landau equation. *BIT Numerical Mathematics*, 49:141, 2009.
- [24] O. Koch. Asymptotically correct error estimation for collocation methods applied to singular boundary value problems. *Numer. Math.*, 101:143–164, 2005.
- [25] O. Koch, R. März, D. Praetorius, and E.B. Weinmüller. Collocation methods for index-1 DAEs with a singularity of the first kind. To appear in *Math. Comp.*
- [26] G. Pulverer, G. Söderlind, and E. Weinmüller. Automatic grid control in adaptive BVP solvers. Submitted to *Numer. Algorithms*.
- [27] 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.*, 332:523–541, 2007.
- [28] L. Shampine and J. Kierzenka. A BVP solver based on residual control and the MATLAB PSE. *ACM T. Math. Software*, 27:299–315, 2001.
- [29] 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/>.
- [30] 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.

- [31] Ch. Simon. *Numerical Solution of Singular Eigenvalue Value Problems for Systems of ODEs*. Master thesis, Vienna University of Technology, Vienna, Austria, 2009.
- [32] S. Staněk, G. Pulverer, and E. Weinmüller. Analysis and numerical solution of positive and dead core solutions of singular two-point boundary value problems. *Comput. Math. Appl.*, 56:1820–1837, 2008.
- [33] S. Staněk, G. Pulverer, and E. Weinmüller. Analysis and numerical solution of positive and dead core solution of singular Sturm-Liouville problems. Submitted to *Comput. Math. Appl.*, 2009.
- [34] R. Winkler. Path-following for two-point boundary value problems. Techn. Rep. 78, Department of Mathematics, Humboldt-University Berlin, Germany, 1985.

VIENNA UNIVERSITY OF TECHNOLOGY

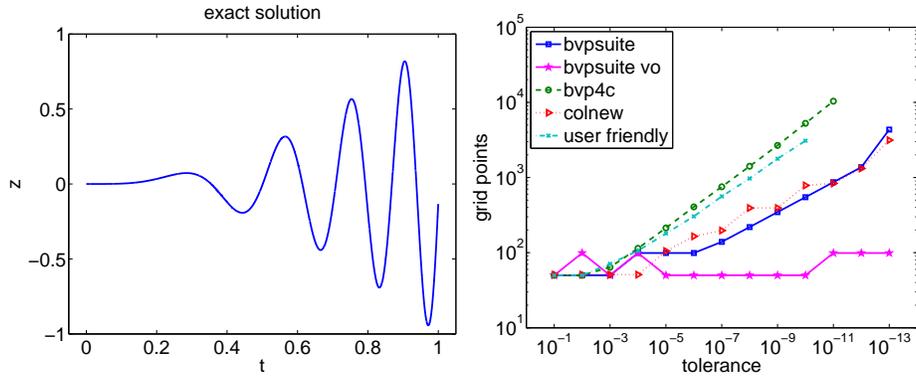


FIGURE 5. Problem (42), $k = 5$: The exact solution (left) and the number of grid points in the final mesh (right).

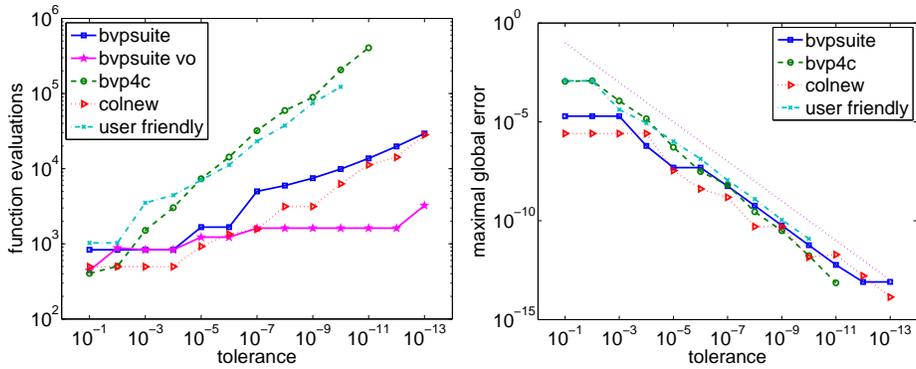


FIGURE 6. Problem (42), $k = 5$: The number of function evaluations (left) and the tolerance versus accuracy graph (right).

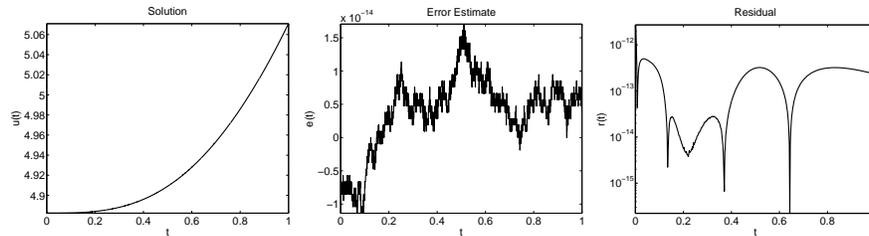


FIGURE 7. Problem (45): The numerical solution, the error estimate and the residual for $\lambda = 1.42604644036221$.

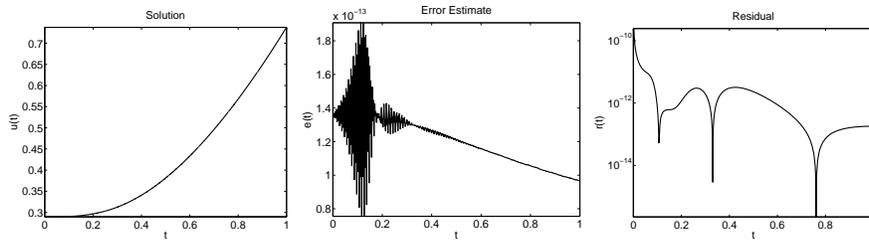


FIGURE 8. Problem (45): The numerical solution, the error estimate and the residual for $\lambda = 1.42139222684689$.

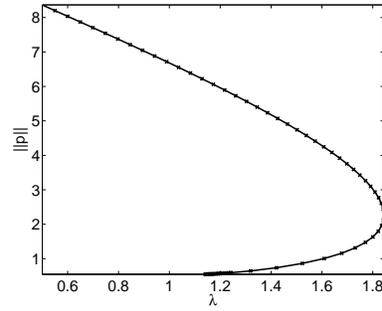


FIGURE 9. Graph of the $\|p\| - \lambda$ path obtained in 76 steps of the pathfollowing procedure, where $\|p\| = \max_{t \in [0,1]} |p(t)|$. The turning point has been found at $\lambda \approx 1.8442$.