Appl. Numer. Math. 205 (2024) 184–194

Contents lists available at ScienceDirect

Applied Numerical Mathematics

journal homepage: www.elsevier.com/locate/apnum

Research Paper

Numerical treatment of singular ODEs using finite difference and collocation methods

Matthias Hohenegger [a], Giuseppina Settanni [b], Ewa B. Weinmüller [a,*], Mered Wolde [a]

[a] Institute of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8–10, A-1040 Wien, Austria
[b] Dyrecta Lab Istituto di Ricerca, Via Vescovo Simplicio 45, Conversano, I-70014, Italy

ARTICLE INFO

Keywords:
Boundary value problems
Ordinary differential equations
Singular problems
Mesh adaptation
Finite difference scheme
Collocation method

ABSTRACT

Boundary value problems (BVPs) in ordinary differential equations (ODEs) with singularities arise in numerous mathematical models describing real-life phenomena in natural sciences and engineering. This motivates vivid research activities aiming to characterize the analytical properties of singular problems, to investigate convergence of the standard numerical methods when they are applied to simulate differential equation with singularities, and to provide software for their efficient numerical treatment. There are two well-known, high order numerical methods which we focus on in this paper, the finite difference schemes and the collocation methods. Those methods proved to be dependable and highly accurate in the context of regular differential equations, so the question arises how do they preform for singular problems. While, there is a strong evidence for the collocation schemes to be a robust method to solve singular systems in a stable and efficient way, finite difference schemes are still considered less suitable for this problem class.

In this paper, we shall compare the performance of the code `HOFiD_bvp` based on the high order finite difference schemes and `bvpsuite2.0` based on polynomial collocation, when the codes are applied to singular problems in ODEs. We are fully aware of the difficulties in a code comparison, so in this paper, we will try to only diagnose the potential improvements, we could address in the next update of the codes.

1. Introduction

The class of singular ODEs we are interested in, consists of a system of second order ODEs with a singularity of the first kind,

$$y''(t) - \frac{A_1(t)}{t} y'(t) - \frac{A_0(t)}{t^2} y(t) = f(t, y(t), y'(t)), \quad t \in (0, L], \tag{1}$$

subject to correctly posed boundary conditions. The analytical investigations are usually carried out for the related ODE system of first order which can be derived from (1) using the Euler transformation $z(t) = (y(t), t y'(t))^T$, see [37],

$$z'(t) - \frac{M(t)}{t} z(t) = g(t, z(t)), \quad t \in (0, L]. \tag{2}$$

* Corresponding author.
E-mail addresses: matthias.hohenegger@student.tuwien.ac.at (M. Hohenegger), giuseppina.settanni@dyrecta.com (G. Settanni), e.weinmueller@tuwien.ac.at (E.B. Weinmüller), mered.wolde@student.tuwien.ac.at (M. Wolde).

https://doi.org/10.1016/j.apnum.2024.07.002
Received 9 March 2024; Received in revised form 27 May 2024; Accepted 8 July 2024

Available online 11 July 2024

While in the classical literature the inhomogeneity g is assumed to be smooth, see [19,37], recent studies also deal with right-hand sides which are less regular,

$$z'(t) - \frac{M(t)}{t} z(t) = G(t, z(t)) := \frac{g(t, z(t))}{t}, \quad t \in (0, L], \tag{3}$$

cf. [13–15,36]. Problems of the form (3) arise in the modelling of snow avalanche run-up and run-out [25,28,29]. Similar problem formulation arises when models posed on a semi-infinite interval $t \in [0, \infty)$ are transformed to the finite domain $t \in (0, 1]$. Search for the numerical method to solve problems (1) and (3) is strongly motivated by further numerous applications from physics, chemistry, mechanics, or ecology. Also, research activities in related fields, like the computation of connecting orbits in dynamical systems or singular Sturm-Liouville problems benefit from techniques developed for problems of the form (3). The computation of self-similar solution profiles for the nonlinear Schrödinger equation is also essentially reduced to this problem type [10,11]. For more information see [12,21,23,27].

In this study, we focus our attention on the scalar second order ODEs of the form

$$y''(t) - \frac{a_1}{t} y'(t) - \frac{a_0}{t^2} y(t) = f(t, y(t), y'(t)), \quad t \in (0, L], \tag{4}$$

subject to boundary conditions

$$y(0) = 0, \quad b(y(L), y'(L)) = 0 \tag{5}$$

or

$$y'(0) = 0, \quad b(y(L), y'(L)) = 0. \tag{6}$$

Here, f and b are appropriately smooth functions on $[0, L]$, and a_0, a_1 are real constants.

The classes of the first and second order BVPs in singular ODEs have been extensively studied and various approaches to their numericals solution have been proposed, among them finite difference schemes and collocation methods,[1] see [18,19,26,38,39]. Collocation schemes proved especially robust and efficient and therefore, they have been used in many codes as basic solvers, see FORTRAN codes, `COLSYS` [6] and `COLNEW` [5,8], and MATLAB codes `MIRKDC` [17], `bvp4c` [34,35], `sbvp` [3], and `bvpsuite` [4,22, 24].

While due to its advantageous properties in the context of singular problems, polynomial collocation is widely acknowledged, the finite difference methods are not so popular. Recently, the MATLAB code `HOFiD_bvp`, based on high order finite difference schemes, has been successfully applied to solve singularly perturbed[2] problems with discontinuous source terms.

In order to detect advantages and possible drawbacks of the MATLAB codes `HOFiD_bvp` and `bvpsuite2.0`, we compare the performance of these software packages applied to numerically simulate the singular BVPs (4), (5) and (4), (6).

The paper is organized as follows: In Sections 2 and 3, we describe the codes `HOFiD_bvp` and `bvpsuite2.0`, respectively. Numerical simulations of model problems are discussed in Section 4 and summarized in Section 5.

2. Finite difference schemes and the `HOFiD_bvp` code

The `HOFiD_bvp` package [33] is based on the finite difference method of high order to treat BVPs of the second order ODEs.[3] The method follows the approach of the boundary value methods, see [1,2,9,32]. It uses symmetric difference schemes (called "main schemes") to approximate the solution values in the inner points of the mesh and (non-symmetric) "initial" and "final" difference schemes of order p at the beginning and at the end of the interval. As main scheme central differences of order p (computed with $p+1$ points) are used for the approximation of the second derivative, while for the first derivative backward or forward formulae of the same order are utilized following the upwind method. Moreover, the code does not require to transform the second order problem to the corresponding first order one, since the derivatives are approximated separately. This become clear when we consider a scalar ODE posed on the interval $(0, 1]$,

$$\begin{aligned} &y''(t) = f(t, y(t), y'(t)), \\ &b(y(0), y'(0), y(1), y'(1)) = 0, \end{aligned} \tag{7}$$

with a singularity at $t = 0$. Let us consider the mesh

$$\Delta_N = \{\tau_i = \tau_{i-1} + h_i,\ i = 1, \ldots, N,\ \tau_0 = 0,\ \tau_N = 1\}$$

[1] In [30], the simplified kernel reproducing method combined with the least squares was proposed to solve a class of a nonlinear singular BVP.

[2] However, note that the difficulties of this problem class are essentially different from the challenges we face while dealing with singular ODEs.

[3] Typically, the orders are $p = 4, 6, 8$ or 10.

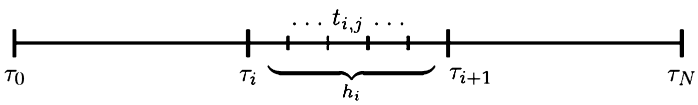


Fig. 1. Mesh with variable stepsizes the interval $[a, b]$, $a = \tau_0$, $b = \tau_N$.

and denote by M_1 and M_2 the quasi-Toeplitz matrices of size $(N+1) \times (N+1)$ approximating the second and the first derivative respectively. For $i = p/2, \dots, N - p/2$ the matrix coefficients are related to the main formulae, while for $i = 0, \dots, p/2 - 1$ and $i = N - p/2 + 1, \dots, N$ they depend on the initial and final methods of order p. Certainly, in $i = 0$ and $i = N$ boundary conditions are required to hold. This means that in case of Neumann or Fourier-type conditions, the first derivative is approximated using initial formula of order p having respectively zero initial conditions in $i = 0$ and zero final conditions in $i = N$. The variable mesh is assembled by blocks of different stepsize containing at least $p+4$ equidistant points, so that the mesh structure on the overall interval is piecewise constant. Moreover, let $Y = (y_0, y_1, \dots, y_N)^T$ be the vector of the values of the numerical solution, and $Y_\nu = (y_0^\nu, y_1^\nu, \dots, y_N^\nu)^T$, $\nu = 1, 2$, the vectors of the approximations of $y'(t)$ and $y''(t)$ in the mesh points. The coefficients of the matrices M_1 and M_2 are computed by solving a Vandermonde-type linear system, where the derivative approximations are obtained from

$$Y_1 = \text{diag}\left(\frac{1}{h_i}\right) M_1 Y, \quad Y_2 = \text{diag}\left(\frac{1}{h_i^2}\right) M_2 Y,$$

and the solution of the BVP (7) is obtained from the system

$$Y_2 = F(T, Y, Y_1).$$

It is important to stress that the variable mesh is piecewise constant, with blocks that have at least p+4 equidistant points. The starting mesh is usually uniform and it is adapted using the error equidistribution principle, based on the evaluation of an absolute error estimate calculated from two methods of consecutive orders. Moreover, the code can potentially work with four different even orders using a variable order strategy proposed in [33]. We are especially interested to see if this feature will prove useful in the context of singular problems resulting in coarser meshes and shorter execution times, see Section 4.

3. Collocation method and `bvpsuite2.0`

In the scope of the code are BVPs for systems of implicit mixed order[4] ODEs,

$$F(t, p_1, ..., p_s, y_1(t), y_1'(t), ..., y_1^{(l_1)}(t), ..., y_n(t), y_n'(t), ..., y_n^{(l_n)}(t)) = 0, \tag{8}$$

$$B(p_1, ..., p_s, y_1(c_1), ..., y_1^{(l_1-1)}(c_1), ..., y_n(c_1), ..., y_n^{(l_n-1)}(c_1),$$
$$..., y_1(c_q), ..., y_1^{(l_1-1)}(c_q), ..., y_n(c_q), ..., y_n^{(l_n-1)}(c_q)) = 0. \tag{9}$$

Here, the solution is denoted by $y(t) = (y_1(t), y_2(t), ..., y_n(t))^T$ and p_i, $i = 1, \dots, s$, are unknown parameters which have to be calculated along with the solution y. The problem can be posed on a finite, $t \in [a, b]$, or semi-finite interval, $t \in [a, \infty)$, $a \geq 0$. For the latter case, a driver routine automatically transforms the semi-infinite domain to the finite one. The boundary conditions are multipoint boundary conditions posed at points c_ν, $\nu = 1, \dots, q$.

We apply the polynomial collocation to solve the problem. To this aim, we consider a partition Δ of the interval $[a, b]$, as shown in Fig. 1.

In each subinterval $J_i = (\tau_i, \tau_{i+1})$, we introduce m so-called collocation points

$$t_{i,j} = \tau_i + p_j h_i, \quad j = 1, ..., m, \quad 0 < p_1 < p_2 < \dots < p_m < 1.$$

In many applications, singularities may occur in $t = a$ and $t = b$, which motivates the choice $p_1 > 0$ and $p_m < 1$ avoiding evaluation of F at the singular point, where F becomes unbounded. Naturally, Gaussian points or equidistant inner points are good options for the collocation points of the method.

The solution y is approximated by a piecewise polynomial function which has to satisfy the boundary conditions, certain global continuity conditions at the points τ_i, $i = 0, \dots, N-1$, and the collocation conditions, which means to satisfy the system of ODEs (up to the round-off errors) at the collocation points $t_{i,j}$, $i = 0, \dots, N-1$, $j = 1, \dots, m$.

Further problem classes in scope of `bvpsuite2.0` are Sturm-Liouville eigenvalue problems for the differential operators, Index 1 differential-algebraic equations, and parameter dependent problems solved utilizing path-following techniques.

[4] The highest derivatives of the solution components may vary.

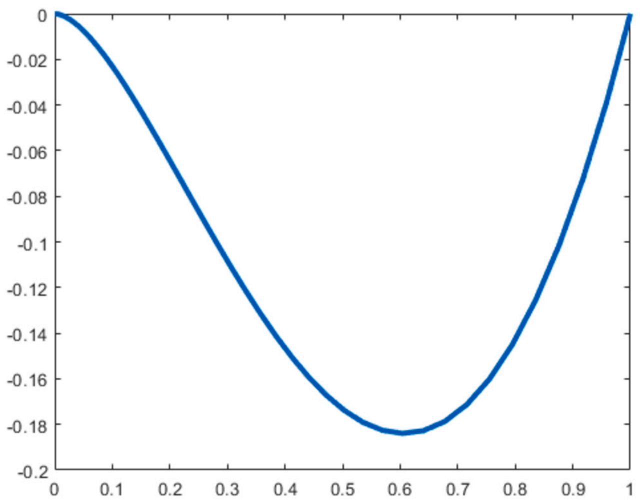


Fig. 2. Problem 1: Exact solution $y(t) = t^2 \ln t$.

Table 1
Problem 1: Results from `bvpsuite2.0`.

order	TOL	mesh	error	exact rel. error	exact abs. error
6	1e-8	51	6.1013e-10	3.3430e-14	6.1490e-15
8	1e-8	51	6.6641e-10	1.7964e-13	3.3029e-14
10	1e-8	51	9.2947e-10	1.1065e-11	2.0351e-12
6	1e-8	21	3.4399e-09	2.8829e-13	5.2993e-14
8	1e-8	21	7.3561e-09	1.8116e-13	3.3168e-14
10	1e-8	21	7.3319e-09	1.0511e-11	1.9334e-12

Each code module is equipped with an error estimation routine which is used as a basis for the mesh adaptation. The error estimation is based on the $h - h/2$ idea which means that the collocation scheme is executed on two meshes, with $N + 1$ and $2N + 1$ mesh points, and the values of both solutions on the coarser mesh deliver the error estimate for the global error there.

The mesh adaptation strategy has been developed in [31]. It consists of two steps. The first step is carried out on a coarse mesh with 51 mesh points.[5] The aim of the first step is to relocate the mesh points in such a way that they correctly reflect the solution behaviour. This is done in a few iteration steps aiming at the equidistribution of the defect. In the second step appropriately many points are added (along the grid density function from the first step) to satisfy the tolerance for the global error prescribed by the user.

4. Numerical results

Problem 1

We first consider the following BVP:

$$y''(t) - \frac{1}{t} y'(t) - 2 = 0, \quad y(0) = y(1) = 0,$$

whose exact solution is given by $y(t) = t^2 \ln t$, see Fig. 2.

The results of the numerical simulation using `bvpsuite2.0` are documented in Table 1. By `order` we denoted the order of the collocation method, `TOL` is the tolerance parameter for the mixed error control (in which relative and absolute tolerances are set equal to `TOL`), and `mesh` shows the number of mesh points N in the final mesh. The maximum of the error estimate which has to satisfy the tolerance requirement to terminate the run is denoted by `error`, while `exact rel. error` and `abs. error` is the maximum of the true relative and absolute error, respectively.[6]

The solution of Problem 1 is *not smooth* and its higher *derivatives become unbounded* close to $t = 0$. The problem is solved using mesh adaptation and two starting meshes with 21 and 51 mesh points. One can see that the tolerance is satisfied already on the starting mesh with 21 points. The exact errors are much smaller than `TOL`, and range between 10^{-14} to 10^{-11}.

[5] This number is used on the normalized interval of the length equal to 1 and has to be enlarged for longer intervals accordingly.

[6] Note that these maximal values have been obtained as discrete maxima taken over the mesh points and collocation points.

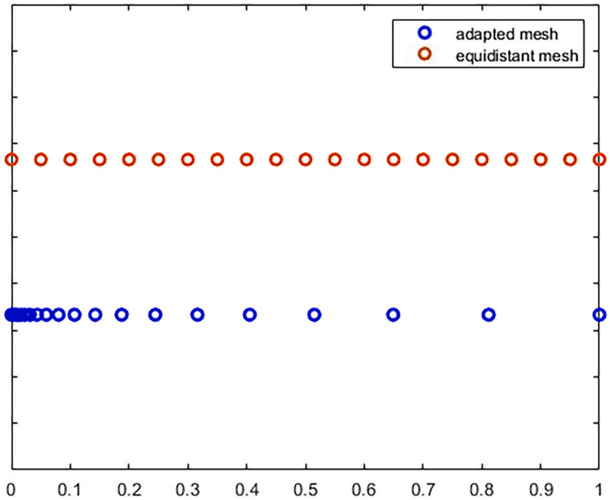


Fig. 3. Problem 1: Starting mesh and adapted final mesh with 21 mesh points.

Table 2
Problem 1: Results from `HOFiD_bvp`. The third column shows the estimated absolute error which is basically of the same size as the relative one. In this example, the error estimate satisfies the prescribed tolerance, however the true error is slightly larger.

order	TOL	mesh	error	exact abs. error	exact rel. error
4	1e-8	3585	3.0059e-09	1.4246e-08	1.4243e-08
6	1e-8	918	9.3704e-09	7.6807e-08	7.6806e-08
8	1e-8	474	9.8446e-09	1.0902e-07	1.0902e-07
10	1e-8	571	9.9055e-09	1.3958e-07	1.3958e-07

However, the mesh points have been appropriately relocated to correctly reflect the solution behaviour and the size of the `error`, see Fig. 3. The region, where the solution is non smooth is correctly detected and the mesh points lay denser close to the origin, where that error is large. In the subinterval $[0.3, 1]$ the mesh becomes coarser than the uniform mesh.

The simulation results obtained by `HOFiD_bvp` can be found in Table 2. The code controls both, the absolute and the relative error but the results turn to be less accurate although the mesh is much denser. For the lower orders 4 and 6, we obtain more accurate results than for the higher orders 8 and 10. This behaviour is a consequence of the order reduction down to 2 which can be observed in all tests.

Problem 2

Let us consider the singular BVP [34,35],

$$y''(t) + \frac{2}{t}y'(t) - \phi^2 y(t)\exp\left(\frac{\gamma\beta(1-y(t))}{1+\beta(1-y(t))}\right) = 0, \quad y'(0) = 0, \;\; y(1) = 1,$$

where $\phi = 0.6$, $\gamma = 40$, and $\beta = 0.2$. The problem has three solutions which can be computed using different starting solutions, see Fig. 4.

In Table 3, the results of the numerical simulation with `HOFiD_bvp` are collected. The code solved the problem within the prescribed tolerance from initial meshes specified below. No difficulties have been observed.

The results obtained from `bvpsuite2.0` can be found in Table 4. The tolerance requirement has been satisfied on the initial mesh with 51 equidistant points, no mesh adaptation was necessary. If we compare the solution accuracy, the collocation code delivers results whose errors are a few orders of magnitude more accurate. However, we were not able to recover $y_2(t)$. We have tried various starting profiles, but could observe convergence only to either $y_1(t)$ or $y_3(t)$. Solution $y_2(t)$ seems to be a repulsive fixed point for the collocation code.

Problem 3

Let us now consider the BVP problem arising from electromagnetic self-interaction theory [7],

$$y''(t) + \frac{4}{t}y'(t) + (ty(t) - 1)y(t) = 0, \quad y'(0) = 0,$$

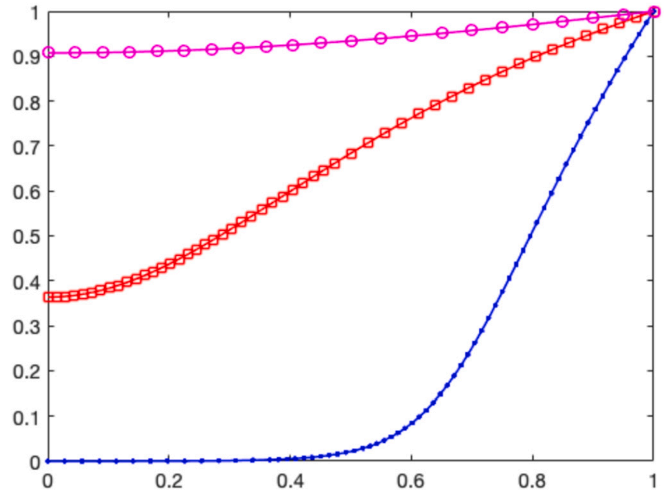


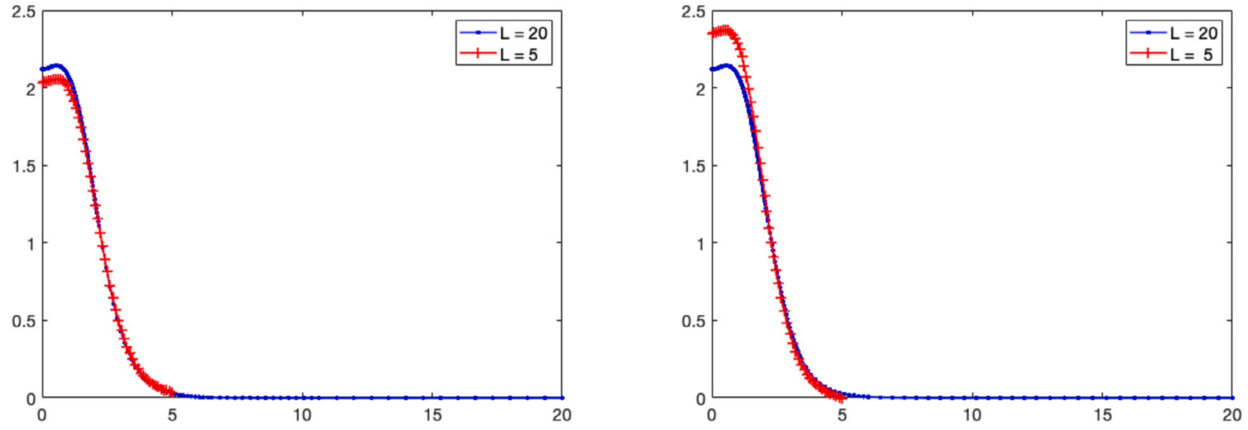
Fig. 4. Problem 2: Solutions computed using `HOFiD_bvp` of order 6. The solution $y_1(t)$, depicted in blue (lowest curve), has been computed using the initial profile $y(t) \equiv 0$. To obtain the solution $y_2(t)$, shown in red (central curve), the following starting values have been used: $y(0) = 0.38$, $y(0.5) = 0.75$, and $y(1) = 1$. Finally, the starting guess $y(t) \equiv 1$ provided the solution $y_3(t)$ depicted in magenta (upper most curve). For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.

Table 3
Problem 2: The initial mesh for solutions $y_2(t)$ and $y_3(t)$ contained 15 points. For the solution $y_1(t)$, the starting mesh with 21 points was used for the methods of order 4, 6, and 8, while for order 10 the initial mash with 51 was utilized.

order	TOL	mesh	abs. error	rel. error
$y_1(t)$				
4	1e-8	157	9.8068e-09	8.4706e-09
6	1e-8	66	8.2490e-09	8.2432e-09
8	1e-8	57	1.3310e-09	1.3243e-09
10	1e-8	54	1.4665e-10	1.2778e-10
$y_2(t)$				
4	1e-8	167	6.5398e-09	4.4201e-09
6	1e-8	47	6.5869e-09	4.8281e-09
8	1e-8	27	1.1974e-08	8.6217e-09
10	1e-8	29	6.1202e-09	4.4882e-09
$y_3(t)$				
4	1e-8	44	1.7429e-08	9.1387e-09
6	1e-8	18	6.6477e-09	3.4857e-09
8	1e-8	15	6.5300e-10	3.4240e-10
10	1e-8	15	1.3426e-10	7.0396e-11

Table 4
Problem 2: Solutions $y_1(t)$ and $y_3(t)$ obtained from the `bvpsuite2.0` code.

order	TOL	mesh	abs. error
$y_1(t)$			
6	1e-8	51	3.4230e-14
8	1e-8	51	2.2180e-14
10	1e-8	51	1.8676e-13
$y_3(t)$			
6	1e-8	51	7.5002e-14
8	1e-8	51	7.4565e-14
10	1e-8	51	6.9790e-14



(a) $y'(0) = 0, \quad y(L) + y'(L) = 0$ (b) $y'(0) = 0, \quad y(L) = 0$

Fig. 5. Problem 3: Solutions computed using `HOFiD_bvp` of order 6.

Table 5
Problem 3: Numerical results computed using `HOFiD_bvp` with an initial mesh of 32 and 48 points for $L = 5$ and $L = 20$, respectively.

L	order	TOL	mesh	abs. error	rel. error
$y'(0) = 0, \quad y(L) + y'(L) = 0$					
5	4	1e-8	4033	2.9461e-09	9.6905e-10
5	6	1e-8	143	6.8640e-09	2.2562e-09
5	8	1e-8	169	1.7346e-09	5.7010e-10
5	10	1e-8	52	6.0698e-09	2.0002e-09
5	6 8 10	1e-2 1e-5 1e-8	49	1.2210e-08	4.0235e-09
20	4	1e-8	2077	1.0856e-08	6.3889e-09
20	6	1e-8	829	6.9737e-09	2.2297e-09
20	8	1e-8	188	8.4732e-09	8.4655e-09
20	10	1e-8	171	1.5021e-08	4.8028e-09
20	6 8 10	1e-2 1e-5 1e-8	133	1.5724e-08	5.0274e-09
$y'(0) = 0, \quad y(L) = 0$					
5	4	1e-8	2753	3.1070e-08	9.2501e-09
5	6	1e-8	389	3.8279e-09	1.1397e-09
5	8	1e-8	78	6.6038e-09	4.9040e-09
5	10	1e-8	52	3.0596e-09	9.1231e-10
5	6 8 10	1e-2 1e-5 1e-8	50	2.6145e-08	7.7963e-09
20	4	1e-8	11105	2.6538e-09	8.4660e-10
20	6	1e-8	1781	4.7281e-09	1.5121e-09
20	8	1e-8	387	3.8053e-09	1.2167e-09
20	10	1e-8	261	3.0595e-08	9.7821e-09
20	6 8 10	1e-2 1e-5 1e-8	141	8.0393e-09	2.5704e-09

where $L = 5, 8, 10$, or 20. The second boundary condition is posed at $t = L$ and reads: $y(L) + y'(L) = 0$ or $y(L) = 0$.

Using the initial solution profile,

$$y(t) = \begin{cases} 2, & t \le 1.5, \\ 2e^{1.5-t}, & t > 1.5. \end{cases}$$

results in the convergence to the desired non-trivial solution. In Fig. 5, the solutions for $L = 5$ and 20 are presented. Since the solutions of the problem are not smooth, utilizing mesh adaptation is the only way to approximate the solution in an efficient way. Note that the finite difference code is adapting the mesh in a proper way by enhancing the mesh density in the regions of difficulty.

In Table 5, the numerical results obtained from the finite difference code are recorded. Here, an order variation strategy has been applied, see the fifth row of the block. The orders 6, 8, and 10 were used to reach the accuracies 10^{-2}, 10^{-5}, and 10^{-8}, respectively. This strategy allows to satisfy the final tolerance 10^{-8} on a mesh with less points, compare the third and the fifth row in a block. In general, the computational times are short.

Table 6
Problem 3: Results obtained using `bvpsuite2.0` from starting meshes with 51 points.

L	order	TOL	mesh	abs. error
$y'(0)=0$ $y(L)+y'(L)=0$				
5	6	1e-08	51	6.3159e-15
5	8	1e-08	51	5.7958e-15
5	10	1e-08	51	1.5234e-12
20	6	1e-08	51	1.7189e-10
20	8	1e-08	51	1.7451e-12
20	10	1e-08	51	2.9796e-12
$y'(0)=0$ $y(L)=0$				
5	6	1e-08	51	8.5716e-15
5	8	1e-08	51	1.2037e-14
5	10	1e-08	51	9.0060e-13
20	6	1e-08	51	1.7187e-10
20	8	1e-08	51	3.6224e-13
20	10	1e-08	51	2.9859e-12

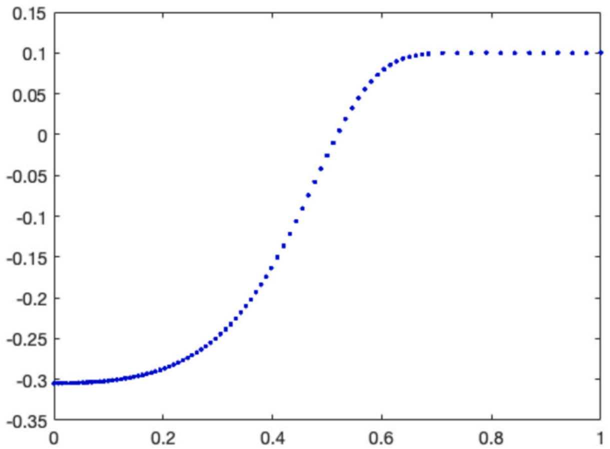


Fig. 6. Problem 4, $\xi = 0.1$: Solution of the problem computed with `HOFiD_bvp` of order 8 using an initial uniform mesh with 15 points and initial solution profile defined in [20].

When comparing the results of the finite difference method with those of collocation, see Table 6, we observe a very good accuracy of the latter solution. The absolute error ranges from 10^{-14} to 10^{-10} on meshes with 51 mesh points.[7]

Problem 4

Let us finally consider the singular BVP arising from the Cahn-Hillard theory and used in hydrodynamics to study the behaviour of non-homogeneous fluids. In particular, the equation describes the formation of microscopic bubbles in a non-homogeneous fluid (vapour inside liquid). The problem reads [16,20,21]:

$$y''(t) + \tfrac{N-1}{t} y'(t) - 4\lambda^2 (y(t)+1)y(t)(y(t)-\xi) = 0, \quad y'(0) = 0, \quad y(\infty) = \xi,$$

where $N = 3$, $\lambda = 1$, and $\xi = 0.1, 0.6$. We now transform the problem to a finite domain using the transformation

$$\tau = 1 - \frac{1}{\sqrt{1+t}}, \quad \tau \in [0,1].$$

Consequently,

$$\tfrac{1}{4}(1-\tau)^6 y''(\tau) + \left(\tfrac{N-1}{2\tau(2-\tau)} - \tfrac{3}{4}\right)(1-\tau)^5 y'(\tau) - 4\lambda^2 (y(\tau)+1)y(\tau)(y(\tau)-\xi) = 0,$$

subject to the boundary conditions $y'(0) = 0$, $y(1) = \xi$ follows. The problem is parameter dependent with $\xi \in (0,1)$. The larger the value of ξ, the steeper the inner boundary layer becomes, see Fig. 6 and 7.

[7] The tolerances have been satisfied on the initial mesh.

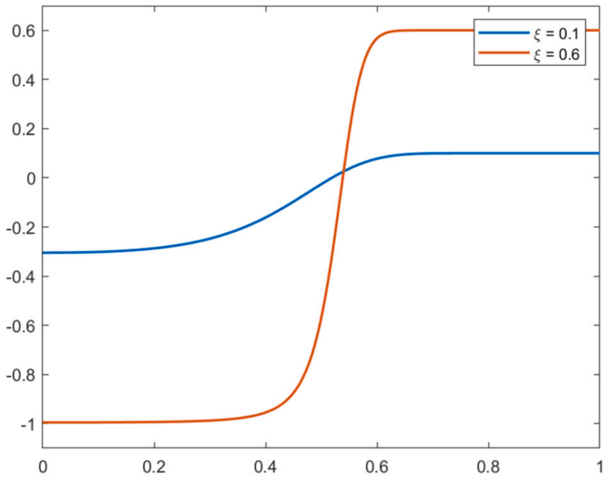


Fig. 7. Problem 4, $\xi = 0.1$ and $\xi = 0.6$: Solution of the problem computed with `bvpsuite2.0` using an initial uniform mesh with 51 points and initial solution profile defined in [20].

Table 7
Problem 4, $\xi = 0.1$: Results obtained from `HOFiD_bvp`.

order	TOL	mesh	abs. error	rel. error
4	1e-8	787	1.1073e-09	8.4874e-10
6	1e-8	247	4.6386e-09	3.5608e-09
8	1e-8	86	8.2062e-09	6.2899e-09
10	1e-8	48	1.2494e-08	9.5763e-09
6 8 10	1e-2 1e-5 1e-8	51	1.1820e-08	9.0598e-09

Table 8
Problem 4, $\xi = 0.1$ and $\xi = 0.6$: Results obtained from `bvpsuite2.0`. The tolerances have been satisfied on the initial mesh with 51 mesh points. The mesh adaptation strategy implemented in the code, requires a certain asymptotic quality of the solution, so that lowering the number of the mesh points in the starting mesh is not recommended. The code can also work with less points in the initial mesh, see Problems 1, but coarse initial meshes are risky.

order	TOL	mesh	abs. error
$\xi = 0.1$			
6	1e-8	51	2.8471e-14
8	1e-8	51	1.4489e-15
10	1e-8	51	2.2298e-13
$\xi = 0.6$			
6	1e-8	51	7.6069e-10
8	1e-8	51	2.7590e-12
10	1e-8	51	6.1910e-12

Clearly, the choice for the initial solution profile is crucial. For any initial profile which is not sufficiently close to the exact solution, the iteration diverges or converges to the trivial solution $y(t) = \xi$.

In Table 7, we record the results of the simulation carried out with `HOFiD_bvp`. Here, we see that the variable order strategy,[8] only slightly changes the number of points in the final mesh, when compared to the run, where the final accuracy of $\texttt{TOL} = 10^{-8}$ is sought to become satisfied directly, see rows four and five in Table 7.

The results obtained using `bvpsuite2.0` can be found in Fig. 7 and in Table 8. We again observe that the tolerances are satisfied on the initial mesh with errors considerably smaller than the required accuracy of 10^{-8}. Similar to the previous observations, only at

[8] Orders 6, 8, and 10 are used to satisfy the tolerances 10^{-4}, 10^{-6}, and 10^{-8}, respectively.

order 10 and for variable orders 6, 8, and 10, the `HOFiD_bvp` solves the problem on a mesh whose size is comparable to the one of `bvpsuite2.0`.

We simulated a few more singular BVPs with somewhat similar findings, so we are not documenting all results here.

5. Conclusion

The results of the simulations described above suggest that the high order difference schemes may suffer from order reductions in the context of singular ODEs which result in very fine meshes. Here, the convergence analysis would be in place to clarify the situation before designing a code version for singular ODEs. However, the code `HOFiD_bvp` proved able to cope with the singularity in a very dependable way and reasonable execution times.

The code `bvpsuite2.0` shows to be suitable for singular ODEs with a singularity of the first kind. The meshes used stay coarse and when adapted correctly reflect the solution behaviour. In most of the cases the tolerances are satisfied on the initial mesh with 51 points. Clearly, the code is 'overdoing' by delivering solutions which are a few orders of magnitude more accurate than required. This can be explained by the size of the initial mesh which seems to 'have too many points' to precisely hit the tolerance. On the other hand, reducing the number of points in the initial mesh is not recommended due to a certain asymptotic quality of the solution calculated on the initial mesh in order to obtain a proper mesh density function, see the first step of the mesh adaptation routine described at the end of Section 3. This solution and the adapted initial mesh are the basis for the second step and their good quality is necessary for the mesh adaptation module to work dependably, cf. Section 3.

CRediT authorship contribution statement

Matthias Hohenegger: Data curation, Software, Visualization. **Giuseppina Settanni:** Conceptualization, Investigation, Software, Writing – original draft. **Ewa B. Weinmüller:** Conceptualization, Investigation, Supervision, Writing – review & editing. **Mered Wolde:** Data curation, Software, Validation.

Acknowledgement

The authors acknowledge TU Wien Bibliothek for the financial support through its Open Access Funding Program.

References

[1] P. Amodio, G. Settanni, A finite differences MATLAB code for the numerical solution of second order singular perturbation problems, J. Comput. Appl. Math. 236 (2012) 3869–3879, https://doi.org/10.1016/j.cam.2012.04.011.
[2] P. Amodio, I. Sgura, High-order finite difference schemes for the solution of second-order BVPs, J. Comput. Appl. Math. 176 (1) (2005) 59–76.
[3] W. Auzinger, G. Kneisl, O. Koch, E.B. Weinmüller, A collocation code for boundary value problems in ordinary differential equations, Numer. Algorithms 33 (2003) 27–39.
[4] W. Auzinger, K.N. Burdeos, M. Fallahpour, O. Koch, R.G. Mendoza, E.B. Weinmüller, A numerical continuation method for parameter-dependent boundary value problems using bvpsuite2.0, J. Numer. Anal. Ind. Appl. Math. 16 (2022) 1–13.
[5] U.M. Ascher, R.D. Russell, R.M.M. Mattheij, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, SIAM's Classic in Applied Mathematics, Philadelphia, 1995.
[6] U.M. Ascher, J. Christiansen, R.D. Russell, Collocation software for boundary value ODEs, ACM Trans. Math. Softw. 7 (1981) 209–222.
[7] U.M. Ascher, R.D. Russell, Reformulation of boundary value problems into standard form, SIAM Rev. 23 (1981) 238–254.
[8] G. Bader, U. Ascher, A new basis implementation for a mixed order boundary value ODE solver, SIAM J. Sci. Stat. Comput. 8 (1987) 483–580.
[9] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon and Breach Science Publishers, Amsterdam, 1998.
[10] C.J. Budd, O. Koch, E.B. Weinmüller, Self-Similar Blow-up in Nonlinear PDEs, AURORA TR-2004-15, Institute for Analysis and Scientific Computing, Vienna University of Technology, Austria, 2004.
[11] Ch. Budd, O. Koch, E. Weinmüller, Computation of self-similar solution profiles for the nonlinear Schrödinger equation, Computing 77 (2006) 335–346.
[12] Ch. Budd, O. Koch, E.B. Weinmüller, From nonlinear PDEs to singular ODEs, Appl. Numer. Math. 56 (2006) 413–422.
[13] J. Burkotová, I. Rachůnková, E.B. Weinmüller, On singular BVPs with nonsmooth data: analysis of the linear case with variable coefficient matrix, Appl. Numer. Math. 114 (2017) 77–96.
[14] J. Burkotová, I. Rachůnková, E.B. Weinmüller, On singular BVPs with nonsmooth data – convergence of the collocation schemes, BIT Numer. Math. 57 (2017) 1153–1184.
[15] J. Burkotová, I. Rachůnková, S. Stanek, E.B. Weinmüller, S. Wurm, On nonlinear singular BVPs with nonsmooth data. Part 1: analytical results, Appl. Numer. Math. 130 (2018) 23–50.
[16] J. Cash, G. Kitzhofer, O. Koch, G. Moore, E. Weinmüller, Numerical solution of singular two point BVPs1, J. Numer. Anal. Ind. Appl. Math. 4 (1–2) (2009) 129–149.
[17] W.H. Enright, P.H. Muir, Runge–Kutta software with defect control for boundary value ODEs, SIAM J. Sci. Comput. 17 (1996) 479–497.
[18] F.R. De Hoog, R. Weiss, Collocation methods for singular boundary value problems, SIAM J. Numer. Anal. 15 (1978) 198–217.
[19] F.R. De Hoog, R. Weiss, Difference method for boundary value problems with a singularity of the first kind, SIAM J. Numer. Anal. 13 (1977) 775–813.
[20] G. Kitzhofer, O. Koch, E.B. Weinmüller, Collocation Methods for the Computation of Bubble-Type Solutions of a Singular Boundary Value Problem in Hydrodynamics, Technical Report ANUM No. 14/04, Vienna University of Technology, 2004.
[21] G. Kitzhofer, O. Koch, P. Lima, E.B. Weinmüller, Efficient numerical solution of the density profile equation in hydrodynamics, J. Sci. Comput. 32 (2007) 411–424.
[22] G. Kitzhofer, O. Koch, G. Pulverer, C. Simon, E.B. Weinmüller, BVPSUITE a New Matlab Solver for Singular Implicit Boundary Value Problems, ASC Report No. 35/2009, Vienna University of Technology, Vienna, 2009.
[23] G. Kitzhofer, O. Koch, E.B. Weinmüller, Pathfollowing for essentially singular boundary value problems with application to the complex Ginzburg-Landau equation, BIT Numer. Math. 49 (2009) 217–245.

[24] G. Kitzhofer, O. Koch, G. Pulverer, C. Simon, E.B. Weinmüller, The new Matlab code bvpsuite for the solution of singular implicit BVPs, J. Numer. Anal. Ind. Appl. Math. 5 (2010) 113–134.
[25] O. Koch, E.B. Weinmüller, Analytical and numerical treatment of a singular initial value problem in avalanche modeling, Appl. Math. Comput. 148 (2003) 561–570.
[26] O. Koch, Asymptotically correct error estimation for collocation methods applied to singular boundary value problems, Numer. Math. 101 (2005) 143–154.
[27] R. Lamour, R. März, E.B. Weinmüller, Boundary-value problems for differential-algebraic equations: a survey, in: A. Ilchmann, T. Reis (Eds.), Surveys in Differential-Algebraic Equations III, DAE-F, Springer International Publishing, Switzerland, 2015, pp. 117–309.
[28] D.M. McClung, A.I. Mears, Dry-flowing avalanche run-up and run-out, J. Glaciol. 41 (1995) 359–369.
[29] D.M. McClung, O. Hungr, An equation for calculating snow avalanche run-up against barriers, Avalanche Formation, Movement and Effects, IAHS Publ. No. 1621987, pp. 605–612.
[30] J. Niu, M. Xu, Y. Lin, Q. Xue, Numerical solution of nonlinear singular boundary value problems, J. Comput. Appl. Math. 331 (2018) 42–51.
[31] G. Pulverer, G. Söderlind, E.B. Weinmüller, Automatic grid control in adaptive BVP solvers, Numer. Algorithms 56 (2011) 61–92.
[32] G. Settanni, High order finite difference schemes for the numerical solution of second order ordinary differential problems, Ph.D. Thesis, Università degli Studi di Bari, Bari, Italy, 2012.
[33] G. Settanni, Potentiality of the HOFiD_bvp code in solving different kind of second order boundary value problems, Appl. Numer. Math. (2023), https://doi.org/10.1016/j.apnum.2023.08.008.
[34] L.F. Shampine, Singular boundary value problems for ODEs, Appl. Math. Comput. 138 (2003) 99–112.
[35] L.F. Shampine, J. Kierzenka, M.W. Reichelt, Solving boundary value problems for ordinary differential equations in Matlab with bvp4c, https://classes.engineering.wustl.edu/che512/bvp_paper.pdf.
[36] G. Vainikko, A smooth solution to a nonlinear system of singular ODEs, AIP Conf. Proc. 1558 (2013) 758–761.
[37] E.B. Weinmüller, On the boundary value problems for systems of ordinary differential equations with a singularity of the first kind, SIAM J. Math. Anal. 15 (2) (1984) 287–307.
[38] E.B. Weinmüller, On the numerical solution of singular boundary value problems of second order by a difference method, Math. Comput. 46 (173) (1986) 93–117, https://doi.org/10.2307/2008217.
[39] E.B. Weinmüller, Collocation for singular boundary value problems of second order, SIAM J. Numer. Anal. 23 (1986) 1062–1095.