TECHNISCHE
UNIVERSITÄT
WIEN
Vienna | Austria

DISSERTATION

Numerical Solution of Singular BVPs in ODEs and Parabolic PDEs Using Collocation

ausgeführt am Institut für
Analysis und Scientific Computing
der Technischen Universität Wien

unter Anleitung von Ao. Univ-Prof. Dipl.-Ing. Dr. Ewa Weinmüller

durch
Dipl.-Ing. Gernot Pulverer

Quentlistrasse 69
8193 Eglisau
Schweiz

Datum

Unterschrift

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass die vorliegende Arbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen von mir selbstständig erstellt wurde. Alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, sind in dieser Arbeit genannt und aufgelistet. Die aus den Quellen wörtlich entnommenen Stellen, sind als solche kenntlich gemacht. Das Thema dieser Arbeit wurde von mir bisher weder im In- noch Ausland einer Beurteilerin/einem Beurteiler zur Begutachtung in irgendeiner Form als Prüfungsarbeit vorgelegt. Diese Arbeit stimmt mit der von den Begutachterinnen/Begutachtern beurteilten Arbeit überein.

Datum

Unterschrift

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Danksagung

Danken möchte ich meiner Lebensgefährtin Monika Maria Paller, die viel Zeit investieren musste, mich im Lauf dieser Arbeit anzuhören und mich aufzubauen. Ohne sie hätte mir der letzte Mut gefehlt, doch endlich den Abschluss zu wagen.

Mein Dank gilt auch all den Kolleginnen und Kollegen, die mich im Lauf meiner Dissertation mit Tatkraft, aber auch durch nette, motivierende Worte unterstützt haben. Allen voran stehen dabei natürlich die Co-Autoren der veröffentlichten Papers, mit denen die Zusammenarbeit immer spannend, herausfordernd und prägend war.

Ein besonderer Dank gilt dabei meiner Betreuerin Ewa Weinmüller, die selbst als ich nicht mehr sicher war, ob ich die Arbeit beenden sollte, von der Sinnhaftigkeit überzeugt war und mich dazu bewegen konnte.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Contents

1 Motivation and Overview **2**

1.1 Singularity of the First Kind – MATLAB Code `sbvp1.0` 2

1.2 Singularity of the Second Kind – MATLAB Code `bvpsuite1.1` 4

2 Aims of the Thesis **7**

2.1 Grid Adaptivity for BVPs in ODEs and parabolic PDEs 7

2.1.1 Motivation 7

2.1.2 Research Problems in Adaptivity 9

2.2 Numerical Simulation of Models from Applications 10

2.3 New MATLAB Code `time-traveller` for Parabolic PDEs 11

3 Discussion of the Papers and the Code `time-traveller` **12**

4 Paper 1: Automatic grid control in adaptive BVP solvers **18**

5 Paper 2: MATLAB Code `bvpsuite1.1` for the Solution of Singular BVPs in ODEs **50**

6 Paper 3: Singular ODEs arising in the membrane theory **73**

7 Paper 4: Positive and Dead Core Solutions of Singular Eigenvalue Problems **102**

8 Paper 5: Numerical Approach to Foreign Ownership Restrictions **140**

9 Paper 6: Positive and dead-core solutions of singular two-point BVPs in ODEs **169**

10 Paper 7: Singular BVPs arising in the theory of shallow membrane caps **188**

11 The MATLAB Code `time-traveller` **208**

11.1 Transverse Method of Lines (Rothe's Method) for the Numerical Solution of Parabolic PDEs 208

11.2 Adaptive space-time grids in `time-traveller` 209

11.2.1 The adaptive solver for BVPs in ODEs 209

11.2.2 Adaptive time stepping 210

11.3 The Code `time-traveller` 211

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

11.3.1 The Routine timetraveller.m . . . 211
11.3.2 The Routine travelopt.m . . . 215
11.3.3 Common Options . . . 218
11.3.4 TimeControlOpt Options . . . 219
11.3.5 SpaceControlOpt Options . . . 220
11.3.6 Private Options . . . 221
11.4 Solving IBVPs for PDEs Using `time-traveller` . . . 223
11.4.1 Example 1: One Dimensional Linear Heat Equation . . . 223
11.4.2 Example 2: Imaginary Time Schrödinger Equation . . . 232
11.4.3 Example 3: One Dimensional Nonlinear Heat Equation . . . 236
11.4.4 Example 4: Heat Equation with Neumann BC1 . . . 238
11.4.5 Example 5: Heat Equation with Neumann BC2 . . . 239
11.4.6 Conclusions . . . 240

List of Tables **247**

List of Figures **250**

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Einer der Schwerpunkte der Forschungsgruppe "Numerische Analysis und Simulation von Differentialgleichungen" am Institut für Analysis und Scientific Computing der TU Wien ist die Theorie und die numerische Behandlung von Randwertproblemen singulärer Differentialgleichungen. Solche Probleme treten oft in der folgenden Form auf:

$$t^\alpha z'(t) = M(t)z(t) + f(t, z(t)), \quad t \in (0, 1], \quad g(z(0), z(1)) = 0, \quad z \in C[0, 1].$$

Dabei sind die stetige Matrixfunktion M und die stetigen Funktionen f und g als auch die Konstante $\alpha \geq 1$, die die Schwierigkeit des Problems charakterisiert gegeben. Gesucht ist die stetige und lokal eindeutige Lösung des Problems $z \in C[0, 1]$.

Die Suche nach der effizienten Lösung der obigen Probleme ist durch die zahlreichen Anwendungen in den Natur- und Ingenieurwissenachaften motiviert, die durch singuläre Gleichungen modeliert werden. Zu Beginn der Disseratation lagen schon zahlreiche theoretischen Ergebnisse und zwei MATLAB Programme `sbvp1.0` und `bvpsuite1.1` vor. Daraus ergaben sich die Ziele der Dissertation. Das erste Ziel lag darin die neue Schrittweitenanpasung für `bvpsuite1.1` zu entwickeln und zu testen. Als zweites sollte dieses neue Programm dazu benützt werden schwierige Anwendungsmodelle zu simulieren. Weiters sollte basierend auf `bvpsuite1.1` eine neue Routine `timetraveller` für partielle Differentailgleichungen (eindimensional im Ort) entwickelt und getestet werden.

Abstract

During recent years, scientific work carried out in the research group 'Numerics and Simulation of Differential Equations' at the Institute for Analysis and Scientific Computing, Vienna University of Technology, focused among others on the analysis and numerical treatment of boundary value problems in differential equation with singularities. Such problems take often the following form:

$$t^\alpha z'(t) = M(t)z(t) + f(t, z(t)), \quad t \in (0, 1], \quad g(z(0), z(1)) = 0, \quad z \in C[0, 1].$$

Here, the continuous matrix function M and the continuous functions f and g, as well as the constant $\alpha \geq 1$, describing the problem's difficulty, are given. The aim is to find a locally unique solution of the problem $z \in C[0, 1]$.

The search for the efficient solution of the above problem is motivated by the numerous application in the natural and engineering sciences formulated in form of singular boundary value problems. As a basis and starting point of the thesis there was extensive theoretical knowledge of the problem and two MATLAB codes `sbvp1.0` und `bvpsuite1.1`. The first aim of the thesis was to design and test a new stepsize adaptation strategy for the collocation based basic solver of `bvpsuite1.1`. Moreover, the new routine shall be used to simulate challenging problems in applications. Finally, on the basis of the improved `bvpsuite1.1` modul a new routine `timetraveller` was to implement to enable the numerical treatment of partial differential equations (one-dimensional in space).

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Chapter 1

Motivation and Overview

During recent years, scientific work carried out in the research group Numerics and Simulation of Differential Equations at the Institute for Analysis and Scientific Computing, Vienna University of Technology, concentrated among others on the analysis and numerical treatment of boundary value problems (BVPs) in ordinary differential equations (ODEs) which exhibit singularities. Such problems are often given in the following form:

$$t^\alpha z'(t) = M(t)z(t) + f(t, z(t)), \quad t \in (0,1], \tag{1.1a}$$

$$g(z(0), z(1)) = 0, \quad z \in C[0,1]. \tag{1.1b}$$

For $\alpha = 1$ the problem is called singular with a singularity of the first kind, for $\alpha > 1$ it is essentially singular (singularity of the second kind). The search for efficient numerical methods to solve (1.1) is strongly motivated by numerous applications from physics, see [18], [20], [44], [78], chemistry, cf. [32], [63], [70], mechanics, [30], ecology, see [56], or economy [34], [37], [45]. Also, research activities in related fields, like the computation of connecting orbits in dynamical systems ([57]), differential algebraic equations ([55]) or singular Sturm-Liouville problems ([14]), benefit from techniques developed for problems of the form (1.1).

1.1 Singularity of the First Kind – MATLAB Code `sbvp1.0`

The objective was to provide a sound theoretical basis and the implementation of an open domain MATLAB code for the numerical solution of BVPs with a singularity of the first kind, $\alpha = 1$. To compute the numerical solution of nonlinear singular BVPs of type (1.1), collocation is used at an even[1] number of collocation points spaced in the interior of a collocation interval. The decision to use collocation was motivated by its advantageous convergence properties for (1.1). For smooth problems, the convergence order is at least equal to the *stage order* of the method. For the collocation schemes (at equidistant inner points or Gaussian points) this convergence results mean

---

[1] The even number of points is motivated by the technical details of the error estimation procedure.

that a collocation scheme with s inner collocation points constitutes a high order basic solver ($O(h^s)$ uniformly in t), robust with respect to the singularity of the first kind. Here, h is the maximal stepsize in a (nonequidistant) grid. In the presence of a singularity other high order methods show order reductions and become inefficient, see for example [41] and [52]. In particular, this means that explicit Runge-Kutta methods and multi-step methods, as well as shooting technique and iterated defect/deferred correction cannot be efficiently used in case of singularities. Moreover, many available FORTRAN and Matlab codes for regular, stiff or/and singularly perturbed BVPs in ODEs are not well-suited for such problems either. The reasons due to both, the nature of singular ODEs and to the algorithmic realization of the codes, are manifold. The shooting technique which heavily relies on the solution of IVPs does not work for singular ODEs in general, because many BVPs important for applications do not have an equivalent formulation as a well-posed IVP, see [53]. The success of the iterated defect/deferred correction technique strongly depends on an appropriately long asymptotic error expansion for the global discretization error/local discretization error (defect). It has been shown in [33] however, that for singular BVPs such an expansion only exists under severe restrictions on the problem data (no multiple eigenvalues, absence of positive eigenvalues, or very large positive eigenvalues of the matrix $M(0)$, or more precisely, their real parts) excluding many important applications. Many codes include evaluation of (1.1a) at $t = 0$ (many implicit Runge-Kutta schemes, or Lobatto collocation without provisions for the singular point) resulting in the immediate breakdown[2] of the calculations, or control defect in the error estimation procedure. The latter strategy is especially disadvantageous for singular problems, because although (for a singularity of the first kind) both quantities have the same order of convergence, the defect is typically some orders of magnitude larger than the global error. For those reasons FORTRAN codes MIRKDC, TWPBVP and TWPBVPL, and Matlab code TOM may not be the best choice for singular problems.

In order to solve the ODE systems efficiently the meshes have to be adapted to the solution behavior. For singular problems, we aim at meshes which are not affected by the steep direction field, staying coarse also close to the singularity when the solution is smooth in that region. To design a mesh adaptation procedure, we need an efficient asymptotically correct a posteriori estimate for the error of the numerical solution. In the context of singular ODEs, global error control seems more appropriate than the control of the residual. This is the case because the values of the residual (a local error measure) are usually much larger than the values of the global error. Thus, it often turns out that grids generated via the equidistribution of the residual are too fine and generate solutions whose global errors are dramatically smaller than the prescribed tolerance, which is inefficient. The global error estimate was introduced in [13] and is based on the defect correction principle. It has been shown that for a collocation method of order $O(h^s)$, the error of the estimate (the difference between the exact global error and its estimate) is of order $O(h^{s+1})$, cf. [52]. This asymptotically correct error estimate yields a reliable basis for an efficient mesh selection procedure and the grid adaptation procedure results in grids which adequately reflect the solution behavior.

The final step was to implement the above algorithm and to provide an open domain Matlab code for nonlinear problems with an error estimation routine and a grid selection strategy. This code,

---

[2] Since the codes require an explicit form of the ODE, $z'(t) = \frac{1}{t^\alpha}(M(t)z(t)+f(t,z(t)))$, the right-hand side becomes unbounded for $t = 0$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

`sbvp1.0` for Matlab 6.0, has been published in 2002, see

`http://www.mathworks.com/matlabcentral/fileexchange` > `Mathematics` > `Differential Equations` > `SBVP1.0 Package`

and [8].

Due to its robustness, collocation was used in one of the best established standard FORTRAN codes for (regular) BVPs, COLNEW, see [4] and [5], and in `bvp4c`, the standard Matlab module for (regular) ODEs with an option for singular problems, cf. [64]. In scope of the FORTRAN code COLNEW are *explicit* systems of at most order four with multi-point boundary conditions. The code is using $h - h/2$ strategy for the error estimation which means that the expensive collocation method is carried out twice, on the original mesh and on the refined mesh with the doubled number of mesh points. The Matlab code `bvp4c` solves also *explicit* ODE systems and is based on Lobatto collocation. As already mentioned, the control of the defect used in this code is especially disadvantageous for singular problems compared to controlling the global error in `sbvp`. Therefore, the meshes provided by `bvp4c` become unnecessarily dense. Comparing `sbvp1.0` with COLNEW and `bvp4c` indicates a very satisfactory performance of `sbvp1.0`. It is competitive with COLNEW and for the above reasons strongly superior to `bvp4c`. Nevertheless, for reasons described below, further development of `sbvp1.0` was necessary to widen its scope beyond that of COLNEW, `bvp4c`, and a newer 'User-Friendly Fortran Code' by L. Shampine, P. Muir and H. Xu, see [65], including among others, fully implicit form of the ODE system with multi-point boundary conditions, arbitrary degree of the differential equations including zero, module for dealing with infinite intervals, module for eigenvalue problems, free parameters, and a path-following strategy for parameter-dependent problems with turning points.

1.2 Singularity of the Second Kind – Matlab Code `bvpsuite1.1`

The code `sbvp1.0` was very well received by the scientific community and, according to many international contacts, is broadly used in many areas of applications. However, `sbvp1.0`, in its original version, was designed to solve only first order systems in explicit form, so that many important applications were not in scope of the code. This was a strong motivation to implement a test version of a new Matlab code `bvpsuite` addressing these additional requirements, cf. [50]. This code is designed to solve implicit systems of ODEs which may have arbitrary variable order including zero. A system of fourth order, for instance, may have the following structure:

$$F(z^{(4)}(t), z^{(3)}(t), z''(t), z'(t), z(t), t) = 0, \quad a \le t \le b,$$
$$g(z^{(3)}(a), z''(a), z'(a), z(a), z^{(3)}(b), z''(b), z'(b), z(b)) = 0, \quad z \in C[a, b].$$

In particular, algebraic constraints are also admitted. Moreover the code can cope with unknown parameters in a way that it is not necessary to introduce artificial differential equations for such parameters. Clearly, in this case additional boundary conditions have to be specified. This can be done not only at the boundary but also within the integration interval. The code can be applied

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

to systems with both types of singularity and features appropriate routines for error estimation and grid adaptation, cf. [12], [13] and [10]. The basic solver routine is still collocation at s inner collocation points, the error estimate is now based on the more robust mesh halving, and the equidistribution of the global error is the basis for the grid adaptation routine.

Another important feature is that parameter-dependent problems are now within the scope of the code, since during 2006 the adaptive path-following strategy was included. The theoretical justification for the method in context of singular problems was given in [47]. This strategy is based on pseudo-arclength parametrization for the solution of parameter-dependent BVPs. In [47], we formulated criteria which ensure the successful application of this method for the computation of solution branches with turning points for problems with an essential singularity.

To illustrate that `bvpsuite1.1` can be applied to solve problems of practical relevance, consider a nontrivial example, the *complex Ginzburg-Landau equation (CLS)*. This particular problem is a very good specimen for a situation in which many of the new features of the code are necessary for a successful numerical treatment.

The CLS equation,

$$\imath \frac{\partial u}{\partial t} + (1 - \imath\varepsilon)\Delta u + (1 + \imath\delta)|u|^2 u = 0, \quad t > 0, \tag{1.2}$$

$$u(x, 0) = u_0(x), \quad x \in \mathbb{R}^3, \tag{1.3}$$

arises as a model in a variety of problems from physics, biology and chemistry. We note that equation (1.2) is a perturbed version of the *nonlinear Schrödinger equation (NLS)* which takes the same form with $\varepsilon = \delta = 0$. It has been conjectured that both equations have *blow-up solutions* which become singular in finite time and obey the same scaling laws as the differential equation, see [21], [25]. After a similarity reduction (see for example [23], [25]) the self-similar solution profile can be computed from the nonlinear BVP for $\tau > 0$,

$$(1 - \imath\varepsilon)\Big(y''(\tau) + \frac{2}{\tau} y'(\tau)\Big) - y(\tau) + \imath a(\tau y(\tau))' + (1 + \imath\delta)|y(\tau)|^2 y(\tau) = 0, \tag{1.4a}$$

$$y'(0) = 0, \quad \Im y(0) = 0, \quad \lim_{\tau\to\infty} \tau y'(\tau) = 0. \tag{1.4b}$$

Now, the BVP (1.4) is parameter-dependent with the path parameter δ, and in addition to the solution function $y(\tau) \in \mathbb{C}$, the unknown (non-negative) parameter a is also to be determined from the system. Moreover, the problem is singular with a singularity of the *first kind* and is posed on a semi-infinite interval. The idea is now to transform (1.4) to a BVP with an *essential singularity* on the interval $(0, 1]$, for details see [46], and to use the path-following strategy to compute the solution branches around the turning points. The main advantage of this idea is that the problem is now posed on a finite interval, and therefore it is not necessary to adapt the length of some truncated interval. The new problem has the form

$$z'(\tau) = \begin{pmatrix} \frac{M(\tau,a)}{\tau} & 0 \\ 0 & \frac{A(\tau,a)}{\tau^3} \end{pmatrix} z(\tau) + \begin{pmatrix} f(\tau, z_1(\tau), z_2(\tau)) \\ g(\tau, z_3(\tau), z_4(\tau)) \end{pmatrix}, \quad 0 < \tau \le 1, \tag{1.5a}$$

$$z_2(0) = 0,\ \Im z_1(0) = 0,\ z_1(1) = z_3(1),\ z_2(1) = z_4(1),\ z_4(0) = 0, \tag{1.5b}$$

with appropriately defined data. Multi-bump solutions of the nonlinear Schrödinger equation were a starting point for the computation of the solution branches for the complex Ginzburg-Landau equation. Following the branches around turning points, real-valued solutions of the nonlinear Schrödinger equation could be computed. During the numerical simulation it turned out that the most stable way to solve the problem (1.5) was to treat it in *implicit* form obtained by premultiplying both sides of (1.5a) by the factor τ^3.

Using `bvpsuite`, some joint work with C. Budd from the Bath Institute of Complex Systems (BICS), Bath, UK, P. Lima from the Instituto Superior Tecnico, Lisbon, Portugal, and I. Rachůnková from the Palacky University, Olomouc, Czech Republic, see [23], [24], [48], [22], [49], and [61] could be carried out. Here, focus only on [61], where the following singular BVP which originates from the theory of shallow membrane caps was investigated,

$$(t^3u'(t))' + t^3\Big(\frac{1}{8u^2(t)} - \frac{a_0}{u(t)} - b_0t^{2\gamma-4}\Big) = 0, \quad \lim_{t\to0+} t^3u'(t) = 0, \; u(1) = 0, \tag{1.6}$$

where a_0, b_0, and γ are given constants. Note that this problem has a more challenging structure than usual. After rewriting (1.6), the following explicit version of the ODE arises:

$$u''(t) + \frac{3}{t}\,u'(t) + \Big(\frac{1}{8u^2(t)} - \frac{a_0}{u(t)} - b_0t^{2\gamma-4}\Big) = 0, \quad 0 < t < 1, \; u(1) = 0. \tag{1.7}$$

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* on 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, cf. [61]. The code `bvpsuite` could be used to approximate solutions[3] of the membrane problem. However, a theoretical justification for the collocation method view of such a problem structure is still missing.

Since the current version the program `bvpsuite1.1` provides a very valuable extension of `sbvp1.0`, it was published in 2010, see

`http://www.asc.tuwien.ac.at/~ewa/> Software: BVPSUITE Implicit singular BVPs`

and [51].

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

---

[3] even though $u'(0)$ may become unbounded

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Chapter 2

Aims of the Thesis

2.1 Grid Adaptivity for BVPs in ODEs and parabolic PDEs

Research on adaptivity was carried out in a very active cooperation with G. Söderlind from Lund University, Lund, Sweden. We were especially interested in efficiently solving BVPs with a singularity of the first or second kind. However, the new technique could also become important in context of parabolic PDEs, see Section 11.

2.1.1 Motivation

In the numerical solution of ODEs, adaptivity means that the mesh width or stepsize is variable. For efficiency, one wants to keep the number of grid points small, but for accuracy we need a small stepsize. This trade-off is handled by putting the grid points where they really matter to accuracy. In this way their number can be kept small, without sacrificing accuracy.

The research group in Lund has developed state-of-the-art adaptive time-stepping algorithms for many years. The interdisciplinary approach uses control theory, digital signal processing, and step density control. As time-stepping is recursive, it lends itself very well to control-theoretic and signal processing techniques.

For initial value problems (IVPs), this approach can be illustrated with a conventional block diagram, see Figure 2.1. Here TOL refers to the external accuracy requirement. The controller chooses a stepsize h which is used by the computational process, to advance the solution one step from time t to $t+h$. The computational process also generates an error estimate, r, which depends on the actual differential equation being solved. This is represented as an external input, φ, which corresponds to the principal error function when the method is applied to the differential equation. Typically, for local errors or residuals, φ corresponds to a higher derivative of the solution to be determined. The error estimate r is then fed back and compared to the accuracy requirement TOL. If the error is too large, the controller will take action and reduce the stepsize; if the error is less

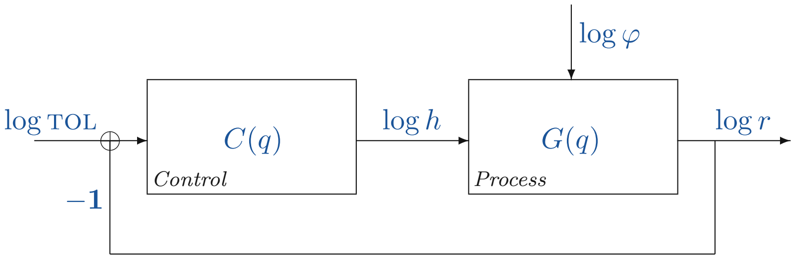

Figure 2.1: Block diagram of the grid adaptation

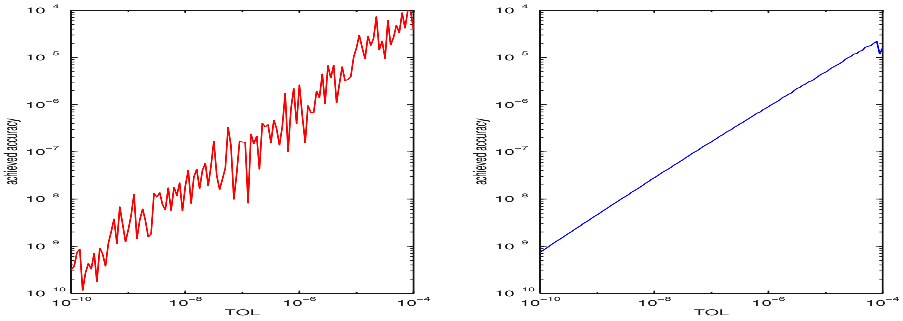

Figure 2.2: *Computational stability.* Achieved accuracy as a function of tolerance when an ODE was solved, using the same code, with conventional stepsize control (left), and digital filter control (right) showing improved regularity and robustness. Total computational work is unaffected. From [69]

than *TOL* the controller will increase the stepsize, not to increase the error, but to economize on the total number of steps necessary to complete the integration.

In two-point BVPs, the control procedure is not sequential. Instead, one needs to estimate the error on a uniform grid, and then relocate the grid points (and possibly change their number) to construct the actual adaptive grid. The optimal grid is typically obtained by Euler-Lagrange variational calculus.

Detailed studies of adaptive time-stepping started some 20 years ago, and the first results based on control theory appeared in the early 90s, see [35] and the survey [67], when a full analysis of elementary proportional-integral (PI) controllers for adaptive time-stepping was developed. This led to further studies of related algorithmic and implementation aspects, [36, 71]. About five years ago, improved techniques based on digital signal processing (filters) were introduced, [66]. This went far beyond elementary PI control, and we demonstrated its impact on real computations [69], compare Figure 2.2. It has subsequently also been used in industrial computations in VLSI circuit simulation [75, 76] as well as in stochastic differential equations [27] and multistep methods [3].

In the cooperation with Lund similar techniques are being explored for two-point boundary value problems. In `sbvp1.0` and `bvpsuite` equdistribution of the global error, closely related to Euler-Lagrange based optimization, were used. We hope to develop a technique which is entirely based on a mathematical analysis of adaptivity. This means that adaptivity is not a single, complex algorithm but a combination of various parts, such as an Euler-Lagrange criterion, a proven error estimator,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

digital filtering in order to produce a smooth grid, and a final procedure for oversampling, so that the necessary number of grid points can be determined. All these parts are based on state-of-the art techniques from control theory and signal processing.

2.1.2 Research Problems in Adaptivity

This dissertation offers a possibility to combine the focal points of the research at the Vienna University of Technology in Austria (numerical solution to BVPs), Humboldt University Berlin in Germany and Lund University in Sweden (control-based adaptivity).

Special problems in BVPs. In two-point BVPs, it is common to generate the grid from a 'grid density' function. The actual grid points are computed by drawing N equidistant samples from a nonlinear function that is computed from a monitoring function derived from a given error estimate, cf. [29]. However, it is not uncommon to encounter stability problems in this procedure, and therefore there are many algorithms that equidistribute the grid points along the solution's arc length. While this is a possible adaptivity criterion, it is unrelated to error control.

This motivated the search for alternatives to the grid adaptation strategy implemented in `bvpsuite`, cf. [13]. Although, the original strategy based on the error estimate of the global error works very dependable in practice, its theoretical justification is based on a priori assumptions concerning the structure of the global error. First of all, its leading term should have the form $\psi \cdot h^q$, where h is the maximal stepsize in the grid. This holds true for the collocation method we are using, for both regular and singular BVPs in ODEs, provided the solution of the underlying analytical problem is appropriately smooth.

However, the function $\psi \cdot h^q$ should also dependably reflect the local behavior of the error. More precisely, if we refine the mesh locally we would like to guarantee that the *global error* decreases accordingly in this very region. Unfortunately, this second important property cannot be attributed to the *global error*, in general. It holds for the superconvergent collocation schemes for instance, which means that for regular ODEs and Gaussian points, the above assumptions are satisfied. For singular ODEs superconvergence order is very often observed but cannot be guaranteed to hold in general. Therefore, we shall implement an alternative strategy and compare it to the standard implementations in `sbvp1.0` and `bvpsuite`. In this strategy, the grid density function is computed using the equidistribution (or Euler-Lagrange minimization) of the *residual*, the number of points in the new mesh is chosen using the norm of the global error. These first variants of the adaptation strategy will be comprehensively tested also in context of other problem settings. We also aim at its theoretical understanding.

Finally, we intend to create stable, reliable and efficient algorithms based on controlling the error, starting from the Euler-Lagrange criterion (equidistribution principle) on grid optimality and implement it within a working code. Although there exists a lot of experimental knowledge on adaptive algorithms controlling the error, we believe that much remains to be done in order to obtain a full understanding of how this adaptivity should be designed to provide best results. In particular, we intend to further develop error estimators, where the error in a single mesh cell only depends on the width of that cell, for regular as well as singular BVPs. This property will make it possible to construct stable and reliable controllers that yield an adapted grid after a single, or at most a few, trial attempts, starting from a coarse, uniform grid. By 'error estimator' above, we mean either an estimator of the residual, the defect or the actual error, where it is of great importance to ascertain the robustness of the estimator as well as its asymptotic properties, as the number of grid points $N \to \infty$. We will need to establish the efficiency of the different approaches, in terms of the mathematical properties of the problem, the method and estimator properties, and the interaction with the controller. Finally, we shall develop software that offers options to solve different classes of problems using various adaptive techniques, while using the most efficient and reliable techniques for singular as well as regular BVPs.

Extension to partial differential equations As we have good algorithms available for time-stepping in IVPs and a new approach to grid generation for BVPs, it is only natural to combine these techniques in order to devise adaptive algorithms in initial-boundary value problems in (at least 1D) partial differential equations (PDEs), cf. Section 11. This entails studying both full space-time adaptivity, and moving mesh algorithms.

Techniques similar to the BVP mesh generation discussed in this proposal have been suggested for 1D hyperbolic problems, but many algorithms suffer from 'grid tangling' as they move the grid points directly instead of working with a grid density function as proposed here. We aim at combining our approach to mesh generation for BVPs with the control and signal processing techniques we have used for IVPs.

Such a combination should be possible for moving mesh algorithms in parabolic problems, such as in reaction-diffusion problems. Because diffusion successively regularizes the solution, we do not expect such problems to be particularly challenging for the grid density approach in one space dimension. Hyperbolic problems, on the other hand, are completely different. In particular, if shocks develop, the numerical scheme could benefit significantly from 'front-tracking', but at this stage we do not plan to include this class of problems. Finally, in several space dimensions, the grid density approach will require very significant further studies.

2.2 Numerical Simulation of Models from Applications

After implementing the new grid adaptation in `bvpsuite`, the code shall be applied to simulate problems important in applications. This part of the work shall be carried out in a cooperation with I. Rachůnková and S. Staněk from the Palacky University, Olomouc, Czech Republic. Moreover, included are simulation of models in applications in context of the `bvpsuite` code development and publication and a simulation of the ownership shares restrictions in political economy.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

2.3 New MATLAB Code `time-traveller` for Parabolic PDEs

Having an efficient code solving BVPs in ODEs adaptively, it seems natural to utilize its advantages when solving parabolic PDEs. We shall implement and comprehensively test a routine solving PDEs of the form

$$\frac{\partial u}{\partial t}(x,t) = F\Big(\frac{\partial^2 u}{\partial x^2}(x,t),\ \frac{\partial u}{\partial x}(x,t),\ u(x,t), x, t\Big), \quad x \in [a,b],\ \ t > 0, \tag{2.1a}$$

$$u(x,0) = u_0(x),\ \ t > 0, \quad B(u(a,t), u(b,t)) = 0,\ \ a \le x \le b, \tag{2.1b}$$

where $u := u(x,t) : [a,b] \times \mathbb{R}_+ \to \mathbb{R}$ is an appropriately smooth function, and F, B and u_0 are given functions specified on suitable domains. For the numerical solution of (2.1), we apply the so-called *transverse method of lines*, or Rothe's method, which means that we discretize in time first and in space afterwards. For the solution in space, we then use our new solver `bvpsuite1.1`. With the backward Euler rule[1], we obtain

$$\frac{u(x,t_{n+1}) - u(x,t_n)}{\Delta t_n} = F\Big(\frac{\partial^2 u}{\partial x^2}(x,t_{n+1}), \frac{\partial u}{\partial x}(x,t_{n+1}), u(x,t_{n+1}), x, t_{n+1}\Big), \tag{2.2a}$$

$$u(x,0) = u_0(x), \quad B(u(a,t_{n+1}),\ u(b,t_{n+1})) = 0. \tag{2.2b}$$

Introducing $v_n(x) := u(x,t_n)$, we can rewrite (2.2) and see that for a given $v_n(x)$ the time-step $t_n \to t_{n+1}$ corresponds to a solution of a BVP for $v_{n+1}(x)$:

$$\mathcal{F}(v''_{n+1}(x),\ v'_{n+1}(x),\ v_{n+1}(x),\ v_n(x)) = 0, \quad B(v_{n+1}(a),\ v_{n+1}(b)) = 0, \tag{2.3}$$

with a suitably defined function $\mathcal{F}$. Error control based on mesh halving will be used to estimate the error in the time-stepping procedure in order to keep error in time below the prescribed tolerance. To solve the ODEs in space, collocation method is used and the grids are controlled as described before using the new strategy for the grid adaptation. This means that in the code, both errors in space and time are controlled to satisfy the prescribed absolute and relative tolerances, $STOL_a, STOL_r, TTOL_a, TTOL_r$.

---

[1] An application of a second order method is also an option and will be considered later.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Chapter 3

Discussion of the Papers and the Code `time-traveller`

The dissertation consists of seven papers followed by the chapter about the code `time-traveller`. The papers, items 1 to 7, are listed in the following way:

1. **Automatic grid control in adaptive BVP solvers, G. Pulverer, G. Söderlind, E.B. Weinmüller, Numer. Algor. 56 (61-92), 2011.**
   In this paper, we develop a new grid adaptation strategy for the boundary value solver `bvpsuite`. Grid adaptation in two-point boundary value problems is usually based on mapping a uniform auxiliary grid to the desired nonuniform grid. Typically, finding the location of grid points to correctly reflect the solution behavior and determining the number of grid points which are necessary to satisfy the tolerance requirement are done in one step. Here, we try to split these two tasks. In the first step, we construct a grid density function $\phi(x)$. The local mesh width $\Delta x_{j+1/2} = x_{j+1} - x_j$ with $0 = x_0 < x_1 < \cdots < x_N = 1$ is computed as $\Delta x_{j+1/2} = \epsilon_N/\varphi_{j+1/2}$, where $\{\varphi_{j+1/2}\}_0^{N-1}$ is a discrete approximation to the continuous density function $\phi(x)$, representing mesh width variation. The parameter $\epsilon_N = 1/N$ controls accuracy via the choice of N. For any given grid, a solver provides an error estimate. Taking this as its input, the feedback control law then adjusts the grid, and the interaction continues until the error has been equidistributed. Once $\phi(x)$ is determined, another control law determines N based on the prescribed tolerance TOL.
   In our case, the grid density function is determined via the equidistribution of the defect of the collocation solution. The iteration is carried out on a coarse grid to safe time. After that the number of points in the final grid is chosen in such a way that the estimate for the global error satisfies the tolerance.
   In the paper the interaction between control system and solver, and the controllers ability to produce an optimal grid in a stable manner is studied. Moreover, we focus on the correct prediction of the number of necessary grid points. Numerical tests demonstrate the advantages of the new control system within the `bvpsuite` solver, for a selection of problems and over a wide range of tolerances.

The contribution of G. Pulverer to this work was 50%**.**

2. **The New MATLAB Code `bvpsuite` for the Solution of Singular Implicit BVPs, G. Kitzhofer, O. Koch, G. Pulverer, Ch. Simon, E.B. Weinmüller, JNAIAM 5 (113-134), 2010.**
Our aim was to design an open domain MATLAB code `bvpsuite1.1` for the efficient numerical solution of BVPs in ODEs. Motivated by applications, we were especially interested in providing a code whose scope is appropriately wide, including fully implicit problems of mixed orders, parameter dependent problems, problems with unknown parameters, problems posed on semi-infinite intervals, eigenvalue problems and systems of differential algebraic equations of index 1. Our main focus however, was on singular BVPs in which singularities in the differential operator arise. In this paper, we first recapitulate the analytical properties of singular systems and the convergence behavior of polynomial collocation used as a basic solver in the code for both singular and regular ODEs and differential algebraic equations. We also discuss the a-posteriori error estimate and the grid adaptation strategy implemented in our code. Finally, we describe the code structure and present the performance of the code which also has been equipped with a graphical user interface for an easy use.
The contribution of G. Pulverer to this work was 30%**.**

3. **A unified approach to singular problems arising in the membrane theory, I. Rachůnková, G. Pulverer, E.B. Weinmüller, Applications of Mathematics 55 (47-75), 2010.**
In this work, we study the singular boundary value problem

$$(t^n u'(t))' + t^n f(t, u(t)) = 0, \quad \lim_{t \to 0+} t^n u'(t) = 0, \quad a_0 u(1) + a_1 u'(1) = A,$$

where $f(t, x)$ is a given continuous function defined on the set $(0, 1] \times (0, \infty)$ which can have a time singularity at $t = 0$ and a space singularity at $x = 0$. Moreover, $n \in \mathbb{N}$, $n \geq 2$ and a_0, a_1, A are real constants such that $a_0 \in (0, \infty)$, whereas $a_1, A \in [0, \infty)$. The aim of this paper was to discuss the existence of solutions to the above problem and apply the general results to cover certain classes of singular problems arising in the theory of shallow membrane caps, where we are especially interested in characterizing positive solutions. The second aim was to simulate the problem and illustrate the theoretical statements, especially those from Theorem 3.6, and to show that `bvpsuite` can cope with this very difficult problem. In order to approximate the problem, we used collocation at 4 Gaussian collocation points. The numerical solution has been calculated on a fixed equidistant mesh with 1000 points. These rather dense grids were necessary for a good visualization of approximations when transforming them from the standard interval $[0, 1]$ back to the infinite interval $[1, \infty)$. The error estimate and the residual were also recorded as indicators for the accuracy of the numerical solution. The error estimate was computed by coupling solutions related to meshes with 1000 and 2000 meshpoints. The residual was obtained by substituting the numerical solution into the system of differential equations.
The contribution of G. Pulverer to this work was 40%**.**

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

4. **Analysis and Numerical Solution of Positive and Dead Core Solutions of Singular Sturm-Liouville Problems, G. Pulverer, S. Staněk, E.B. Weinmüller, Advances in Difference Equations, Volume 2010, Article ID 969536, 37 pages (2010).**
We investigate the singular Sturm-Liouville problem

$$u''(t) = \lambda u(t), \quad u(0) = 0, \quad \beta u'(1) + \alpha u(1) = A,$$

where λ is a nonnegative parameter, $\beta \geq 0, \alpha > 0$ and $A > 0$. We discuss the existence of multiple positive solutions and show that for certain values of λ, there also exist solutions that vanish on a whole subinterval $[0, \rho] \in [0, 1)$, the so-called dead core solutions. The numerical simulation was here especially challenging because using suitably chosen starting profiles, all multiple solutions had to be recovered. On the other hand the dead core solutions were showing sharp edges in the areas around $t \approx \rho$, which certainly was a difficulty for the code. In the numerical simulations, the collocation method, error estimate strategies and the grid adaptation, implemented in `bvpsuite`, proved dependable and robust.
The contribution of G. Pulverer to this work was 40%**.**

5. **Foreign Ownership Restrictions: A Numerical Approach, B. Karabay, G. Pulverer, E. Weinmüller, Comput Acon 33 (361-388), 2009.**
We analyze the reason behind the use of foreign ownership restrictions on inward Foreign Direct Investment (FDI). We extend the results developed by Karabay (2005) by changing the condition on share distribution in the model. Due to this change, we are able to analyze the political economy aspect of this restrictive policy, i.e., we can study the effect of the host government welfare preference on the optimal foreign ownership restriction. Since the analytical solution to the optimal share restriction policy cannot be specified analytically, in general, we use a numerical approach based on collocation and implemented in the code `sbvp1.0`, to approximate the solution to the problem. Within this framework, under certain conditions, it turns out that the rent extraction-efficiency trade-off is sharper the less the host government favors the local firm. We show that not only economic factors but also political factors play an important role in the determination of the foreign ownership restrictions.
The mathematical model arising from the above theory, has the form of a nonlinear implicit scalar ordinary differential equation of second order which can be rewritten in a form of an implicit system of two differential equations of first order. The additional difficulty is that the linearized system whose solution we solve for during the Newton iteration, exhibits a singularity at the origin. Such systems are typically given in the following form:

$$b^\delta z'(b) = M(b)z(b) + b^\delta g(b), \quad b \in (0, 1], \quad R_0 z(0) + R_1 z(1) = \rho, \quad z \in C[0, 1],$$

where $\delta \geq 1$, z is an n-dimensional real function, M is a smooth $n \times n$ matrix, R_0, R_1 are constant $n \times n$ matrices, and g is a smooth n-dimensional function. For the singular problem with a continuous solution $z \in C[0, 1]$, boundary conditions have to show a certain structure which depends on the eigenvalues of the matrix $M(0)$ and are chosen in such a way that the solution $z \in C[0, 1]$ is isolated and the problem at hand is well-posed.
The contribution of G. Pulverer to this work was 50%**.**

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

6. **Analysis and numerical simulation of positive and dead-core solutions of singular two-point boundary value problems, S. Staněk, G. Pulverer, E.B. Weinmüller, Comp. Math. Appl. 56 (1820-1837), 2008.**
   We investigate the solvability of the Dirichlet boundary value problem
   $$u''(t) = \lambda g(u(t)), \quad \lambda \geq 0, \quad u(0) = 1, \quad u(1) = 1,$$
   where λ is a nonnegative parameter. We discuss the existence of multiple positive solutions and show that for certain values of λ, there also exist solutions that vanish on a subinterval $[\rho, 1-\rho] \in (0,1)$, the so-called dead-core solutions. In order to simulate model equations and to illustrate the theoretical findings, we focused on $g(u) = 1/u$ and used the collocation method implemented in `bvpsuite1.1`. The aims and the results of this paper both for the analysis and the numerical simulation are similar to those described in Paper 4.
   The contribution of G. Pulverer to this work was 40%.

7. **On a singular boundary value problem arising in the theory of shallow membrane caps, I. Rachůnková, O. Koch, G. Pulverer, E. Weinmüller, J. Math. Anal. Appl. 332 (523-541), 2007.**
   In this work, we investigate the following singular boundary value problem which originates from the theory of shallow membrane caps,
   $$(t^3u'(t))' + t^3\left(\frac{1}{8u^2(t)} - \frac{a_0}{u(t)} - b_0t^{2\gamma-4}\right) = 0, \quad \lim_{t\to 0+} t^3u'(t) = 0, \quad u(1) = 0,$$
   where a_0, b_0, and γ are given constants. We show the existence of a positive solution to the above problem by means of a generalized lower and upper functions method involving limiting processes. We illustrate the theory by numerical experiments, carried out with the MATLAB code `bvpsuite1.1` based on polynomial collocation, to approximate the solution of the membrane problem.
   The contribution of G. Pulverer to this work was 30%.

8. **The MATLAB code `time-traveller`**
   As mentioned in Section 2.3, a test version of a code for the numerical solution of parabolic PDEs in one space variable has been implemented. The underlying numerical method is Rothe's method. In Chapter 11, the detailed description of the code and the simulation of applications can be found.
   The contribution of G. Pulverer to this work was 80%.

Additional Remarks on `time-traveller`

Some interesting information about the code, can be found in the recent article [26]. Here, the properties of our fully adaptive space-time discretization for a class of nonlinear heat equations was investigated. Recall that the space discretization is based on adaptive polynomial collocation (`bvpsuite1.1`) which relies on equidistribution of the defect of the numerical solution, and the time propagation is realized by an adaptive backward Euler scheme, cf. [59]. From the known scaling laws, the theoretically optimal grids implying error equidistribution

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

are described and it is verified that the adaptive procedure implemented in `time-traveller` closely approaches these optimal grids.
In particular, the study is focused on the problem

$$u_t(x,t) = u_{xx}(x,t) + u^3(x,t), \quad u_x(0,t) = u(4,t) = 0, \quad u(x,0) = u_0(x) = 4e^{-x^2}, \qquad (3.1)$$

whose numerical solution is discussed and analyzed. The numerical solution to problem (3.1) is shown in Figure 3.1 and the asymptotic grid properties depicted in Figure 3.2 are in a good agreement with the theory.

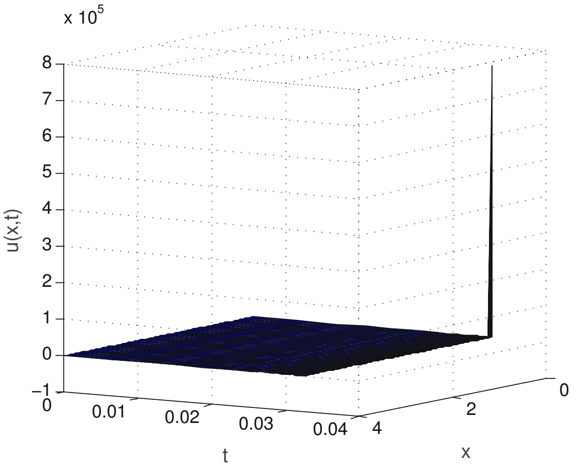

Figure 3.1: The evolution in time for the solution of the problem (3.1). At $t_{end} = 0.0325129$ the solution maximum is $u_{max} = 7.45{\cdot}10^5$.

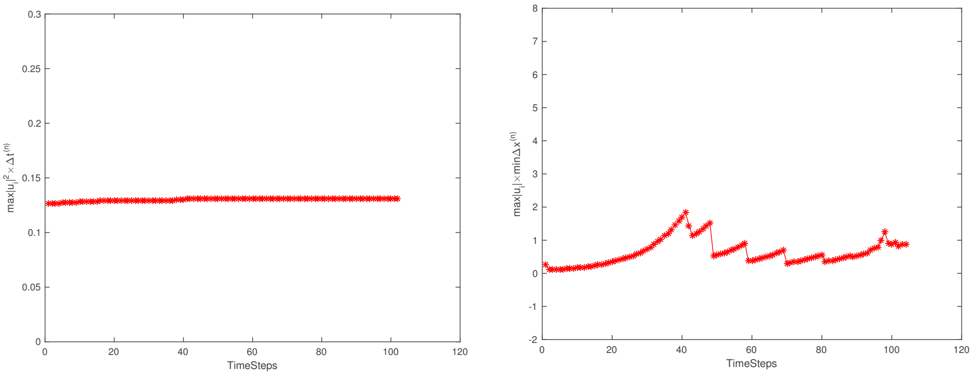

Figure 3.2: Problem (3.1): The top figure shows the evolution in time for $\max|u_i|^2 \times \Delta t^{(n)}$ and the bottom figure shows the evolution in time for $\max|u_i| \times \min \Delta x^{(n)}$ in this problem. Here, $\max|u_i|$ is the maximal solution value and Δt, Δx are stepsizes in time and space, respectively. According to the theory, $\max|u_i|^2 \times \Delta t^{(n)}$ and $\max|u_i| \times \min \Delta x^{(n)}$ shall be constant. Here, $aTOL = 10^{-4}$, $rTOL = 10^{-6}$ and $t_{end} = 0.0325129$, as the desired end point in time, has been reached after 105 steps.

The graphs verify that optimal grids derived from known scaling lows for the analytical problem and providing an efficient solution through error equidistribution can be observed in the practical realization of `time-traveller`. Note that these optimal grids rely solely on asymptotically correct error estimates.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub

Chapter 4

Paper 1: Automatic grid control in adaptive BVP solvers

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Chapter 5

Paper 2: MATLAB Code `bvpsuite1.1` for the Solution of Singular BVPs in ODEs

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Chapter 6

Paper 3: Singular ODEs arising in the membrane theory

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Chapter 7

Paper 4: Positive and Dead Core Solutions of Singular Eigenvalue Problems

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Chapter 8

Paper 5: Numerical Approach to Foreign Ownership Restrictions

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Chapter 9

Paper 6: Positive and dead-core solutions of singular two-point BVPs in ODEs

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Chapter 10

Paper 7: Singular BVPs arising in the theory of shallow membrane caps

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Chapter 11

The MATLAB Code `time-traveller`

11.1 Transverse Method of Lines (Rothe's Method) for the Numerical Solution of Parabolic PDEs

We consider initial/boundary value problems for systems of PDEs of the following form:

$$\frac{\partial}{\partial t}u(x,t) = F\left(\frac{\partial^2 u}{\partial x^2}(x,t), \frac{\partial u}{\partial x}(x,t), u(x,t)\right), \quad x \in [a,b], \quad t > 0,$$

subject to boundary conditions

$$g(u(a,t), u(b,t)) = 0, \quad t > 0,$$

and the initial condition

$$u(x,0) = u_0(x), \quad x \in [a,b].$$

Here, F, g, and u_0 are given vector-valued functions defined on suitable domains.

In order to transform the system of PDEs into a system af ODEs, we first discretize the above system in time using the backward Euler scheme. This results in

$$\frac{u(x,t_{i+1}) - u(x,t_i)}{\Delta t_i} = F\left(\frac{\partial^2 u}{\partial x^2}(x,t_{i+1}), \frac{\partial u}{\partial x}(x,t_{i+1}), u(x,t_{i+1})\right), \; g(u(a,t_{i+1}), u(b,t_{i+1})) = 0, \, i \geq 0,$$

where $u(x,t_i)$ as a given profile for the step from t_i to t_{i+1} and $u(x,t_0) = u(x,0) = u_0(x)$.

From the above system, the function $u(x,t_{i+1})$, $x \in [a,b]$ is to be approximated.

With $v_{i+1}(x) := u(x,t_{i+1})$, the discretized problem can be written as a BVP for an implicit system of ODEs,

$$G(v''_{i+1}(x), v'_{i+1}(x), v_{i+1}(x); v_i(x)) = 0, \quad g(v_{i+1}(a), v_{i+1}(b)) = 0, \quad i \geq 0, \tag{11.1}$$

where $v_i(x)$ is a given data function. For a given $\Delta t_i = t_{i+1} - t_i$, this problem is fully specified. The solution $v_i(x)$ at the time level t_i will be provided as a piecewise polynomial function from the available collocation code `bvpsuite1.1` [50].

11.2 Adaptive space-time grids in `time-traveller`

The full space-time grid adaptation implemented in `time-traveller` consists of stepping in time and on each time level adapting the grid in space. All technical details can be found in [59] and in [60], respectively, see also [26]. To avoid repetitions, we restrict ourselves here to some remarks on the most important issues of the adaptation procedure.

11.2.1 The adaptive solver for BVPs in ODEs

The mesh selection strategy in space implemented in `bvpsuite1.1` was proposed and investigated in [60]. The main idea in the context of two-point boundary value problems is to construct a smooth function mapping a uniform auxiliary grid to the desired nonuniform grid. In [60] a new way to to provide the so-called grid density function $\phi(x)$ was proposed. The local stepsize $h_i = x_{i+1} - x_i$ 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 an approximation to the continuous density function $\phi(x)$, representing the mesh width variation. We use an error estimate to generate a new density from the old one. Digital filters are employed to process the error estimate and the density [68].

For boundary value problems, an adaptive algorithm determines a sequence of mesh densities $\Phi^{[\nu]}$, $\nu = 0, 1, \ldots$, by equidistributing some *monitor function*, a measure of the residual or error estimate. Clearly, since $\Phi^{[\nu]}$ depends 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. Our aim is to avoid this difficulty and therefore, we choose the error estimators such that the error on the interval $[x_i, x_{i+1}]$ only depends on the local mesh width, $h_i = \epsilon_N/\varphi_{i+1/2}$. We decided to use the residual $r(x)$ to define the monitor function, whose values are available from the substitution of the collocation solution $p(x)$ into the analytical problem (11.1). While the residual based monitor function $R(x)$, is used to update the mesh density, the number of the necessary mesh points in the final grid is determined from the condition that the absolute global error satisfies the tolerance requirement. The mesh halving idea provides the values of the error estimate $\mathcal{E}_k(x)$, $k = 1, \ldots, d$, in the interval $[a, b]$. Now we can compute

$$G_{\Delta^m} := \max_x \big(\max_{1 \le k \le d} |\mathcal{E}_k(x)| \big), \quad x \in \Delta^m,$$

where $\Delta^m := \Delta \cup \{x_{i,j} : x_{i,j} = x_i + \rho_j h_i,\ i = 0, \ldots, N-1,\ j = 1, \ldots, m\}$ is the computational grid (including the collocation points) with $\Delta := \{x_0, x_1, ..., x_N\}$, $0 < \rho_1 < \rho_2 < ... < \rho_m < 1$, $h_i := x_{i+1} - x_i$ and $J_i := [x_i, x_{i+1}]$. The number of grid points for the next iteration step is estimated from

$$N^{[\nu+1]} = M \left(\frac{G_{\Delta^m}}{0.9\, TOL} \right)^{1/(m+1)}, \tag{11.2}$$

where $M = 50$ is the fixed number of points in the control grid.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

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 $\Phi^{[0]}$ on a rather coarse mesh with a fixed number of points $M = 50$. The mesh density function is chosen to equidistribute the monitor function $R(x)$.
2. For each density profile in the above iteration, we estimate the number of mesh points necessary to reach the tolerance, according to (11.2).
3. The calculation of the density function is terminated when $N^{[\nu+1]} > 0.9N_\nu$. We expect that in the course of the optimization of the density function the number of the associated mesh points will monotonically decrease. We stop the iteration, when the next density profile $\Phi^{[\nu+1]}$ would save less than 10% of the mesh points compared to the current profile $\Phi^{[\nu]}$.
4. Since the calculation of a residual is not very expensive, we always update the density profile. This means that we use of the information associated with the most recent available numerical solution.
5. Finally, we 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 tolerance requirement is satisfied, the calculations are terminated. Otherwise, the grid is again refined.

For more details and the results of numerical tests, we refer to [60].

11.2.2 Adaptive time stepping

To adapt the stepsize in time, we follow the standard strategy proposed in [59]. The first estimated timestep Δt_0 is updated at each timestep if the approximate solution of the problem is not accurate enough to satisfy the prescribed error tolerance. In this case, the stepsize is reduced.

We use the classical error estimate based on mesh-halving. This means that the accuracy of the solution at each timestep is estimated by comparing two approximations, one obtained by solving the ODE system with full stepsize Δt and the second obtained by solving the ODE system twice with the stepsize $\Delta t/2$. Let us denote these approximations by u_{full} and u_{half}, respectively. Since these two approximations are not given on the same grid, we interpolate their values on a reference grid. We now estimate the error for u_{full} by

$$\mathcal{E}(x) := \frac{2^q}{1-2^q}(u_{full}(x) - u_{half}(x)),$$

where $q = 1$ is the order of convergence of the Euler method. To accept or reject the step, a mixed tolerance is defined using the tolerances for the absolute and relative error,

$$\text{mixed TTOL} := aTTOL + rTTOL\ \max(\|u_{\text{full}}\|_\infty, \|u_{\text{old}}\|_\infty),$$

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

where $aTTOL$ and $rTTOL$ are the absolute and relative tolerances in time, respectively. Moreover, u_{old} is the approximate solution from the previous timestep. Our aim is that error estimate, for each x in the spatial grid, satisfies the tolerance requirement,

$$|\mathcal{E}(x)| \leq \text{mixed TTOL}, \tag{11.3}$$

where $|v|$ is the maximum norm of the vector v. To this aim, we work with the tolerance factor,

$$\text{TOL fac} := \left\| \frac{\text{error}}{\text{mixed TOL}} \right\|_2, \quad \text{error} := \max_x |\mathcal{E}(x)|. \tag{11.4}$$

If TOL fac ≤ 1, the computed step is accepted and the solution is advanced with u_{full}. Next, we try to provide the optimal guess for the new timestep. From the error behavior, TOL fac $\approx C(\Delta t)^{q+1}$ and from $1 \approx C(\Delta t)^{q+1}_{\text{opt}}$, the optimal stepsize is obtained by

$$\Delta t_{\text{opt}} = 0.9 \Delta t_{old} \left(\frac{1}{\text{TOL fac}} \right)^{\frac{1}{q+1}}, \tag{11.5}$$

where $q = \min(q_{\text{full}}, q_{\text{half}})$. In our case, $q_{\text{full}} = 1$ and $q_{\text{half}} = 1$, because we used Euler's method to compute the the full and half steps.

If TOL fac > 1, the step is rejected and the computations are repeated with the new stepsize.

11.3 The Code `time-traveller`

11.3.1 The Routine timetraveller.m

The routine timetraveller.m is a MATLAB routine that organizes the time stepping for the solution of the initial/boundary value problem of a partial differential equation introduced in Section 11.1. The PDEs are transformed to a sequence of BVPs in ODEs using the Rothe's method. At any time point the resulting BVP is solved by `bvpsuite1.1`. However, any other ODE solver could be be used by simply changing two lines in the routine timetraveller.m. Also, the data files, containing the data of the ODE, have to be changed accordingly, to fit into the structure of the new routine.

Figure 11.1 illustrates how timetraveller.m is working. Depending on the settings of an options-structure that is generated by the routine travelopt.m, see Section 11.3.2, each problem takes a different path through the program until it terminates with the desired solution or at an earlier point with all the available data. Below, a detailed description of the functionality of every single part shown Figure 11.1 is given.

Initialize Variables, Evaluate Options, Estimate Initial Stepsize

- Define global variables

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek, WIEN Your knowledge hub

- Initialize auxiliary variables
- Log the starting profile
- Check the options-struct and re-initialize it if necessary
- Assign the destination in time
- (Estimate initial stepsize)

Start Main Path

- Start time-tracking
- Show information about the actual timestep (start, stepsize, ...)

Calculate New Mesh and Starting Profile

- Calculate new number of meshpoints
  - Use the mesh of the last solution
  - Reduce the number of meshpoints used to calculate the last solution but retain the distribution of meshpoints
  - Use a fixed number of meshpoints
- Calculate new starting profile based on the new mesh
  - Use last solution
  - Extrapolate the last solutions on the new mesh along time
  - Use a simple pre-estimate based on the last solutions on a reference-mesh
  - Use the last solutions on the reference-mesh to extrapolate on the nearest points to the new mesh along time

Calculate Full Step

- Calculate the solution with full stepsize using `bvpsuite1.1`
- Log the solution

Calculate Half Steps

- Calculate the solution with the halfed stepsize
- Log the solution
- Calculate the second solution with the halfed stepsize starting in the middle of the interval
- Log the solution

For the second half-step the solution of the full step is used as a starting profile.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Estimate Error

- Interpolate the solutions of the full step and the second half step to a reference-grid using appropriately many points
- Use the interpolated solutions for error estimation
- Check if tolerances are reached

Reduce Stepsize

- Calculate a new stepsize
- Restore the old solution

Error in `bvpsuite1.1`

In case that no convergence in `bvpsuite1.1` is observed and the collocation code stops the calculations, `time-traveller` interprets such situation as an error. However, often it is possible to achieve convergence in `bvpsuite` by refining the mesh. So for timetraveller.m the controlled abort of `bvpsuite` is an error because no solution has been calculated and the routine stopped, although in principle it would be possible to further advance the computations.

- Catch the termination of `bvpsuite1.1`
- Strictly half the stepsize

Save Raw Data

- Log raw data
- Check if the destination has been reached
- Prepare the data for the next step

Program Terminates Before Reaching Destination

Although Figure 11.1 does not show this, it is possible that the routine terminates after a new stepsize has been calculated. This is the case when the stepsize becomes smaller than the minimal stepsize. In the figure, an undesired termination of the routine can only happen after logging the raw data. Then, the routine checks if the destination has been reached and if the stepsize is smaller than the minimal stepsize.

- Terminate the routine
- Pass all collected data to the user

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Program Terminates as Designated

- Terminate after reaching the desired destination
- Pass the solution and all collected data to the user

The description of the functionality of timetraveller.m given in this section is quite superficial. The overview given here was only meant to provide a basic understanding of the sequence of instructions within the routine. For details concerning different options-settings which trigger the different paths of the routine see Section 11.3.2.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

11.3.2 The Routine travelopt.m

The routine travelopt.m is used to set the options for the routine timetraveller.m. As there are multiple possibilities for timetraveller.m to advance, it is necessary to specify the related parameters. It is very important to specify the desired accuracy, by prescribing the tolerances in time an space, both absolute and relative. Other option are listed below,

- UseMesh,
- UseGrid,
- StepSize,
- Timesteps,
- Logging,
- SpaceControlOpt,
- UseLastCalculatedMesh,
- UselastCalculatedSolution,
- ImproveProfile,
- OptionCI,
- Display,
- MaxRefinements,
- TimeControlOpt,
- Private,
- producedby.

Among these options TimeControlOpt, SpaceControlOpt and Private are structs. The sub-options of the Private-struct are

- ReductionFacSpace,
- MaxNumberSpace,
- MinNumberSpace.

The sub-options of the SpaceControlOpt-struct are

- AbsTol,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

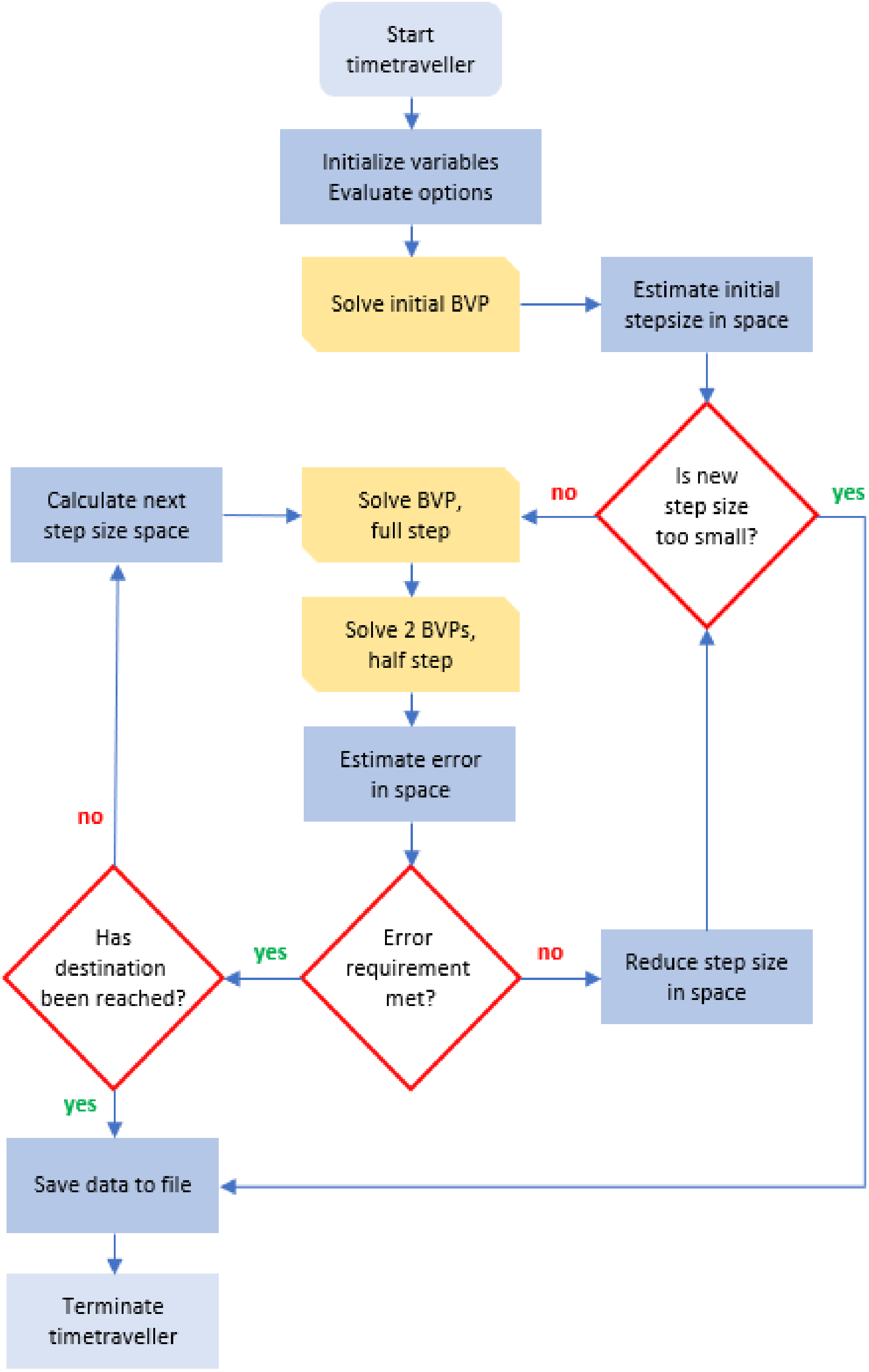

Figure 11.1: The flow-chart for the routine timetraveller.m showing how the routine is organized. If the user specifies a minimal stepsize, the routine might terminate without providing the final solution in case that the estimated new stepsize becomes smaller than that minimal stepsize.

- RelTol,
- K,
- Plot,
- InitialMesh,
- InitialCoeff.

Finally, the sub-options of the TimeControlOpt-struct are

- Method,
- AbsTol,
- RelTol,
- MinStepSize,
- Destination,
- EstimateFirsth,
- InitialStepSize.

Below, we specify the options in more detail, but first we explain how the options themselves can be assigned: There are two ways to initialize them.
The first one is straight forward. With the simple call

$$\text{options} = \text{travelopt}('\text{d}');$$

the default options-struct is assigned to options. If the user wants to change a specific option he can simply redefine the assigned value by typing

$$\text{options.DesiredOption} = \text{DesiredValue};$$

There is no obligation to use 'd' but any other value or string will assign the default options-structure.

The other way to assign the options is to define them right away with the call of travelopt.m. Typing

$$\text{options} = \text{travelopt}('\text{DesiredOption}','\text{DesiredValue}','\text{DesiredOption2}','\text{DesiredValue2}',\ldots,);$$

will assign default options and overwrite the fields with the appropriate values. It is possible to change the options using the second command above.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

11.3.3 Common Options

Common Options are options that are not included in one of the sub-options-structs, TimeControlOpt, SpaceControlOpt, and Private. Possible settings can be found right next to the option. Default settings are placed in brackets.

UseMesh, (0,[1])

This option is used to determine if the solution on the last mesh shall be used in the interpolation of the last solution in space. If set to 1 the necessary values are calculated from the solution on the mesh. If set to 0 the options UseGrid has to be set to 1.

UseGrid, ([0], 1)

UseGrid must be set to 1 if UseMesh is set to 0. Otherwise if UseMesh is set to 1 this option is obsolete. If set to 1 and not obsolete the routine uses the last calculated solution on the collocation grid to determine the desired values for the routine.

StepSize, [0.05]

Any number can be used for StepSize, though a number smaller than 1 is reasonable. If time-travelling is active, i.e. TimeControlOpt.Method is not 0, StepSize is obsolete. If time-travelling is inactive there are two different possibilities to initialize StepSize. If StepSize is scalar, timetraveller.m does each time step with the fixed StepSize. If StepSize is a vector, timetraveller.m takes StepSize as the timegrid on which the routine attempt to solve the problem.

TimeSteps, [20]

Determines the number of TimeSteps. If TimeControlOpt.Method is not 0 or StepSize is a vector, TimeSteps is obsolete.

Logging, (0, [1])

If set to 0 only the necessary data is logged, if set to 1 more data is logged.

UseLastCalculatedMesh, (0, 1, [2])

Other than UseMesh, UseLastCalculatedMesh is used to determine the new initial mesh for the next calculation in space. If set to 0 a standard mesh determined from the ODE solver or, if assigned, the inital mesh passed to SpaceControlOpt is used. At 1 the mesh of the last solution is used. When the option is set to 2 the mesh of the last solution is reduced in number by ReductionFacSpace in the Private-structure. The reduction of mesh points is carried out with unchanged mesh density.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

UseLastCalculatedSolution, (0, [1])

Determines whether to use the last calculated solution (1) as the new initial profile or not (0). If set to 1 this option becomes improved by ImproveProfile.

ImproveProfile, (0, [1], 2, 3)

If UseLastCalculatedSolution is set to 1, ImproveProfile determines if the solution shall be improved using different kinds of interpolation. At 0 no improvement is done. If ImproveProfile is set to 1 the last solutions are interpolated to the next starting mesh and the next starting profile is extrapolated along the time on each point of the mesh. With the setting at 2 a quick guess is made using the last solution on the reference grid. ImproveProfile at 3 uses the solution on the reference grid to extrapolate along the time and then interpolate the obtained solution back to the starting mesh.

OptionCI, []

OptionCI is an options-struct for the ODE solver.

Display, (0, [1])

Display controls the output of the ODE solver during the calculation of the solution.

MaxRefinements [3]

Maximum number of refinements of the ODE solver.

producedby

Safety field to check if the options were produced by travelopt.m.

11.3.4 TimeControlOpt Options

The TimeControlOpt options-structure determines the options for the time stepping.

Method (0, [1])

If Method is set to 1, time stepping using h/$\frac{h}{2}$ is chosen. If set to 0 time stepping is inactive and further options in the TimecontrolOpt-structure are obsolete.

AbsTol [1e-3]

Determines the absolute tolerance for evolution in time.

RelTol [1e-3]

Determines the relative tolerance for evolution in time.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

MinStepSize [1e-8]

MinStepSize determines the smallest allowed stepsize. If a smaller stepsize is calculated from the routine, it first tries to compute a solution with MinStepSize and terminates after failing.

Destination []

Should be specified when using time stepping. If not the routine calculates a destination in time using StepSize and TimeSteps.

EstimateFirsth (0, [1])

Determines whether the first stepsize shall be estimated. If set to 0 the InitialStepSize is used for the first time step.

InitialStepSize [1e-8]

Is used only when time stepping is active and EstimateFirsth is set to 0. Determines the size of the first time step.

11.3.5 SpaceControlOpt Options

SpaceControlOpt includes options that are necessary for the ODE solver.

AbsTol [1e-8]

Determines the absolute tolerance for calculations in space.

RelTol [1e-8]

Determines the relative tolerance for calculations in space.

K [100]

The ratio between the largest and the smallest subinterval on the space mesh.

Plot ([0], 1)

Plot is a feature from the ODE solver `bvpsuite1.1` which plots the solution and error curves when the the tolerance has been satisfied. Since, we use `bvpsuite1.1` quite often during the evolution in time, Plot should be set to 0.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

InitialMesh []

The initial mesh in space is passed over via this option. If InitialMesh is not assigned the default settings from the ODE solver are used. If UseLastCalculatedMesh is 0, InitialMesh will be used in every time step.

InitialCoeff []

The coefficients of the initial solution profile for the ODE solver. The coefficients should be fitting the InitialMesh.

11.3.6 Private Options

Private options determine settings which shall not be changed (or changed only by experienced users).

ReductionFacSpace [2]

Determines by which factor the number of meshpoints from the last calculated solution shall be reduced to determine a new initial mesh. Is obsolete if UseLastCalculatedMesh is not set to 2.

MinNumberSpace [5]

A minimal number of meshpoints which shall be used in the initial mesh.

MaxNumberSpace [100]

A maximum number of meshpoints which shall be used in the initial mesh.

Problem Data

Since timetraveller.m uses an ODE solver to calculate the solutions in space, the problem data has to be provided to the solver in an appropriate form. The solver `bvpsuite1.1` uses an extra m-file for storage of the problem data. Because of the evolution in time, the problem data changes in every step. To avoid too complicated modifications of the data files it is necessary to prepare a second data file for the changes. The original data file for the ODE solver is modified by using functions (the second data file) for the changing parts. Doing so doubles the number of data files, but the advantage is that the second data file (from now on we call it support file) does not need to be altered if another ODE solver is used. Due to organization issues the support files and the date files have the same names with a leading yy_ prefix.

The following listing shows how the support file is organized,

```
function out = yy_datafile(var,x,varargin)
```

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

```
global first_g usemesh_g usegrid_g mesh_g solmesh_g grid_g ...
    solgrid_g stepsizeintime_g

switch var
    case 'c1'
        out=stepsizeintime_g;
    case 'value'
        if first_g
           out=sin(x./5.*pi);
        elseif usemesh_g
            out=interp1(mesh_g,solmesh_g,x,'spline');
        elseif usegrid_g
            out=interp1(grid_g,solgrid_g,x,'spline');
        else
            disp('WARNING: No further evolution in time possible.');
            disp('    Try changing globals usemesh and usegrid.  ');
        end

    case 'spacegrid'
        out=frakem_1('x1');
end
```

As mentioned above the name of the support file and thus the function-call is yy_datafile. There are two variables that have to be passed to the support file. The first one, var, regulates which part of the function is needed. The second one tells on which position a value is needed. The optional varargin argument allows necessary changes without reorganizing the file. The first case for the variable var is c1. c1 stores the actual stepsize using a global variable because the adaptive mesh strategy leads to continuous changing the stepsizes. The second case is value and it carries the last solution, i. e. the starting profile for the current calculation. In case of the first run the global first_g is set to 1 to allow to read the profile from the given function. At every later timestep first_g is set to 0 and the desired value is generated via interpolating the last solution on the mesh or on the collocation grid. The else-part of the switch-case command, simply generates the feedback if the user incorrectly set the corresponding part in the options-struct.

The spacegrid part evaluates the corresponding data file to return the mesh on which the problem shall be solved. Obviously this could have been left out by calling the data file directly from timetraveller.m. Nevertheless it was reasonable to do so, because this way direct communication between timetraveller.m and the data file can be avoided, see Figure 11.2.
The global variables are necessary because the values in the support file change in every step of the solution process. It would be possible to organize the timetraveller.m routine itself to carry all the data (and avoid an additional file) but that would require changes, at least in the the data file, to pass over the current data. Actually, even a change in the ODE solver would be necessary, because

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

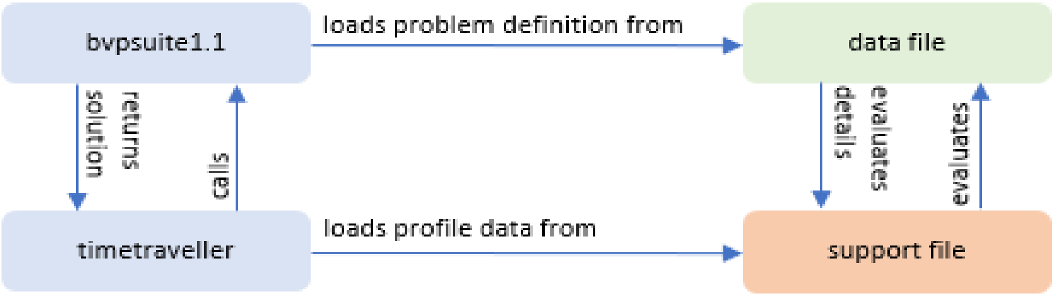

Figure 11.2: The diagram shows how the different files interact during the solution process. Note that the data file and the support file interact with each other, while neither timetraveller with the data file nor `bvpsuite1.1` with the support file do.

it also calls the data file. Consequently, using global variables keeps the timetraveller.m routine modular and ensures correct behavior of the ODE solver.

11.4 Solving Initial/Boundary Value Problems for Parabolic Partial Differential Equations Using `time-traveller`

11.4.1 Example 1: One Dimensional Linear Heat Equation

The first example is the one dimensional heat equation, with a homogeneous rod placed in $x \in [0,1]$. Its formulation is

$$\frac{\partial u}{\partial t}(x,t) = \kappa \frac{\partial^2 u}{\partial^2 x}(x,t), \quad x \in [0,1], \quad t > 0, \quad u(x,0) = u_0(x), \tag{11.6}$$

subject to boundary conditions $u(0,t) = u(1,t)$, $t > 0$, and the thermal diffusivity is set to $\kappa = 1$. To show the performance of `time-traveller`, we will run the code using different starting profiles $u_0(x)$ and different tolerances such that $TTOL = TTOL_a = TTOL_r$ and $STOL = STOL_a = STOL_r$. Note that the tolerance $STOL$ is always smaller than $TTOL$ since the solution on the time level t_n is the data for the run on the level t_{n+1} and such data has to be appropriately accurate for the solution $u(x, t_{n+1})$ to satisfy $TTOL$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Example 1 with $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-2},\ STOL = 10^{-4}$

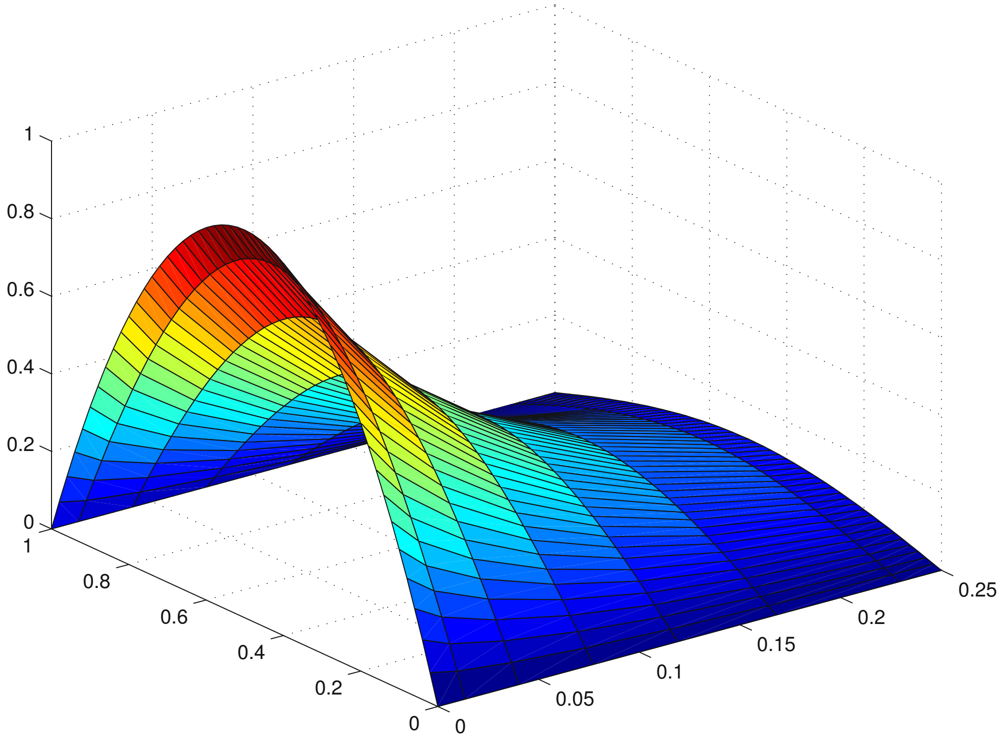

Figure 11.3: Problem (11.6): Solution $u(x,t)$ for $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-2},\ STOL = 10^{-4}$. The endpoint for the time integration was chosen as $t = 0.25$ because for $t > 0.25$ the solution quickly converges to zero.

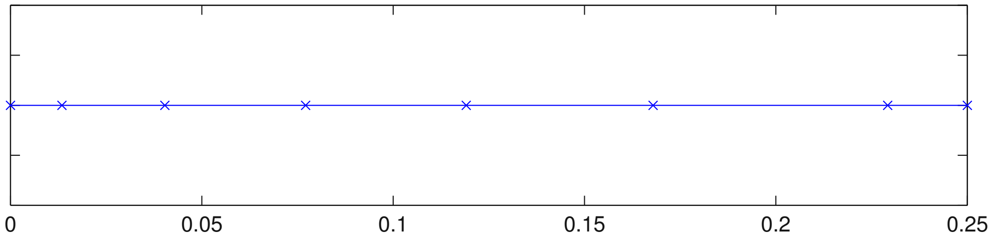

Figure 11.4: Problem (11.6): The time grid with 7 points used by the `time-traveller` to provide a solution sytisfying the tolerances $TTOL = 10^{-2},\ STOL = 10^{-4}$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Example 1 with $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-3},\ STOL = 10^{-5}$

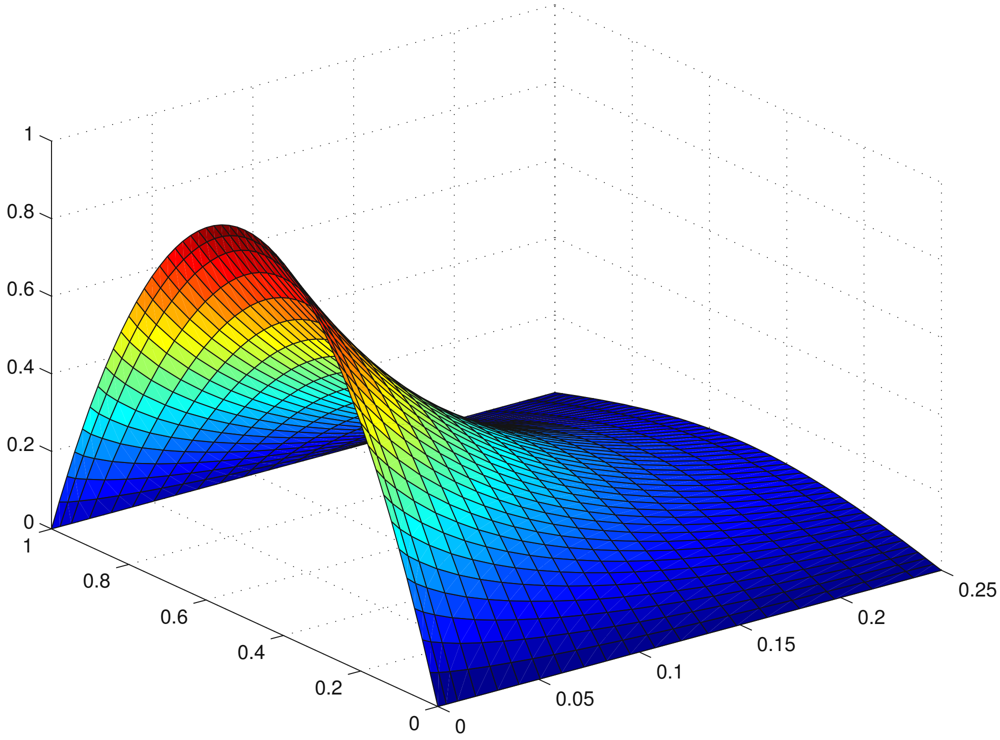

Figure 11.5: Problem (11.6): Solution $u(x,t)$ for $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-3},\ STOL = 10^{-5}$. Again, the endpoint for the time integration was chosen as $t = 0.25$.

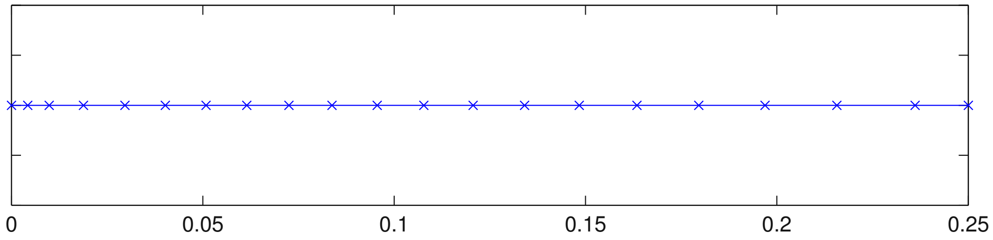

Figure 11.6: Problem (11.6): The time grid with 20 points used by the `time-traveller` to provide a solution sytisfying the tolerances $TTOL = 10^{-3},\ STOL = 10^{-5}$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Example 1 with $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-4},\ STOL = 10^{-7}$

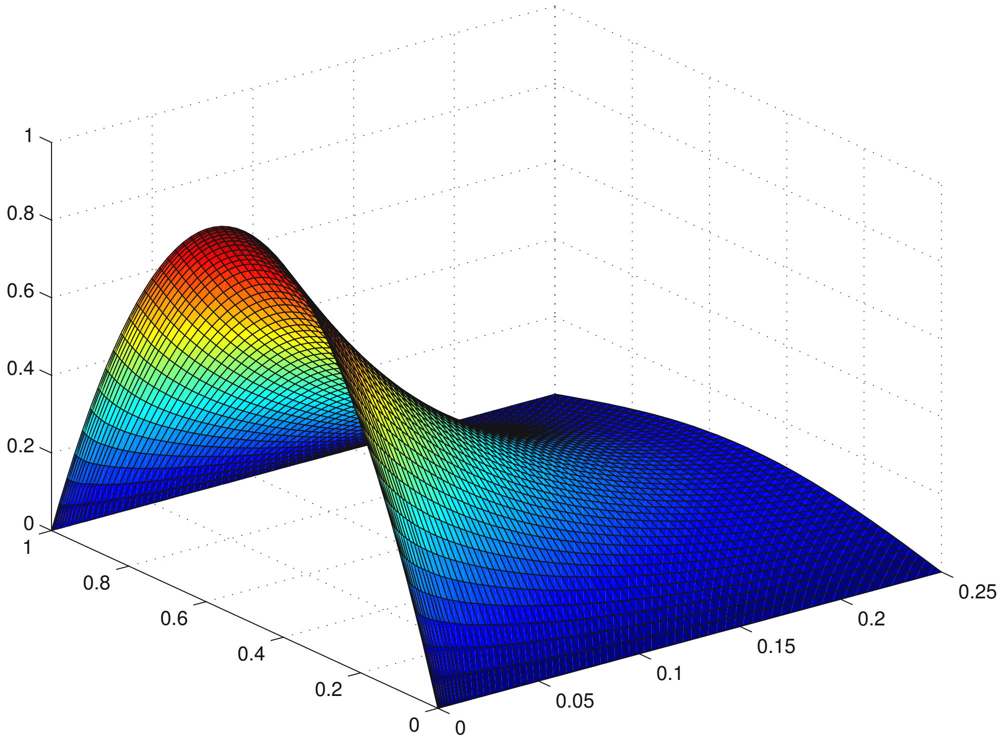

Figure 11.7: Problem (11.6): Solution $u(x,t)$ for $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-4},\ STOL = 10^{-7}$. The endpoint for the time integration was $t = 0.25$.

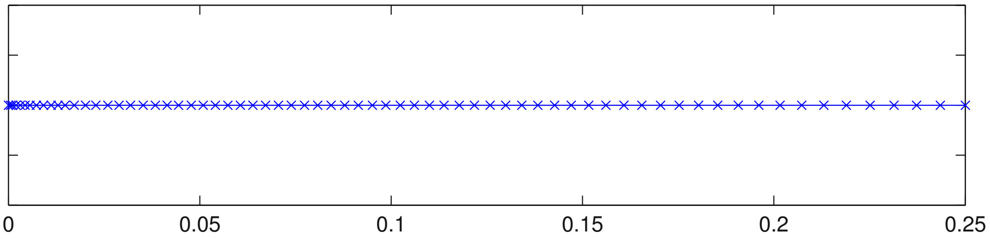

Figure 11.8: Problem (11.6): The time grid with 68 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-4},\ STOL = 10^{-7}$.

Summary for Example 1 with $u_0(x) = 4x - 4x^2$

In Table 11.1, we collect the norm differences for solutions calculated with different tolerances. The notation a/b means that the respective tolerances were $TTOL = 10^{-a}$ and $STOL = 10^{-b}$. The entries in the table are

$$\max_{0\le x\le 1} |u_1(x,0.25) - u_2(x,0.25)|,$$

where the solutions $u_1(x, 0.25)$ and $u_2(x, 0.25)$ are related to a_1/b_1 and a_2/b_2, respectively.

	2/4	3/5	4/7
2/4	–	–	–
3/5	$1.4 \cdot 10^{-2}$	–	–
4/7	$2.0 \cdot 10^{-2}$	$5.3 \cdot 10^{-3}$	–

Table 11.1: Problem (11.6): Norm differences between solutions calculated with different tolerances. In the first and second column, for solutions related to $TTOL = 10^{-2}$ and $TTOL = 10^{-3}$, we expect to see numbers of the magnitude 10^{-2} and 10^{-3}, respectively.

To summarize, Example 1 is an easy problem for the `time-traveller` and the code can provide solutions satisfying all prescribed tolerances. This is the case, because the solution u is smooth and decaying.

Example 1 with $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-2},\ STOL = 10^{-4}$

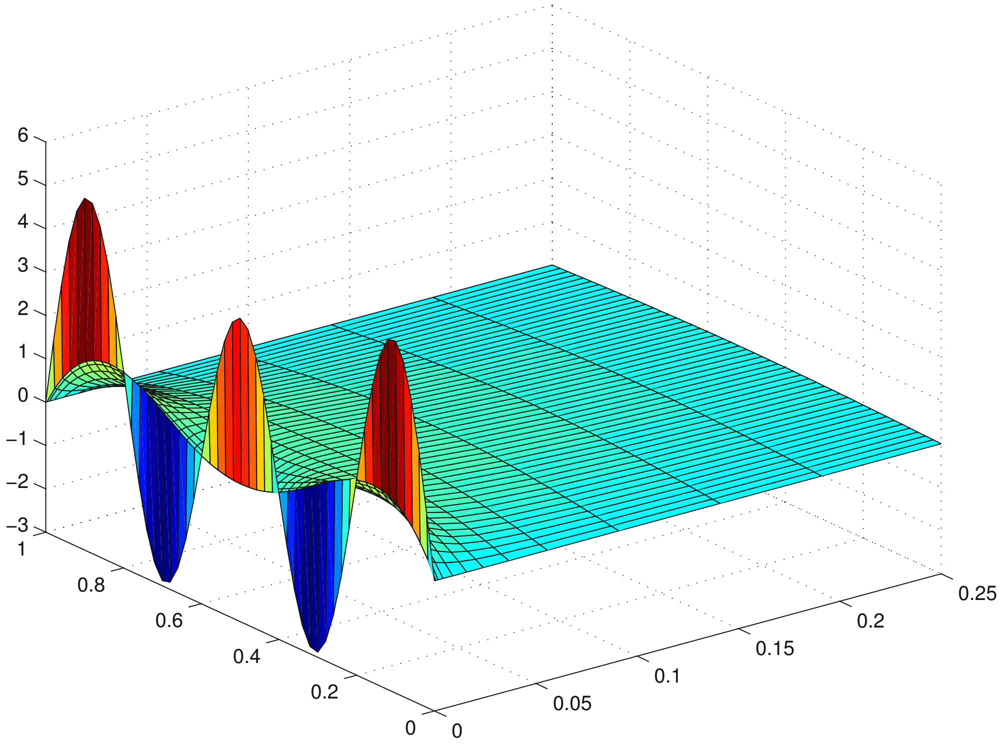

Figure 11.9: Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-2},\ STOL = 10^{-4}$. The endpoint for the time integration was $t = 0.25$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

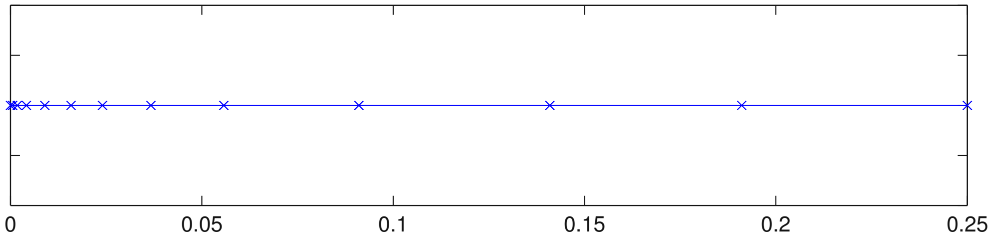

Figure 11.10: Problem (11.6): The time grid with 12 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-2},\ STOL = 10^{-4}$.

Since the solution changes rapidly at the beginning of the integration, we now take a closer look at that area, see Figure 11.11, where the solution is shown on the interval $t \in [0, 0.2)$.

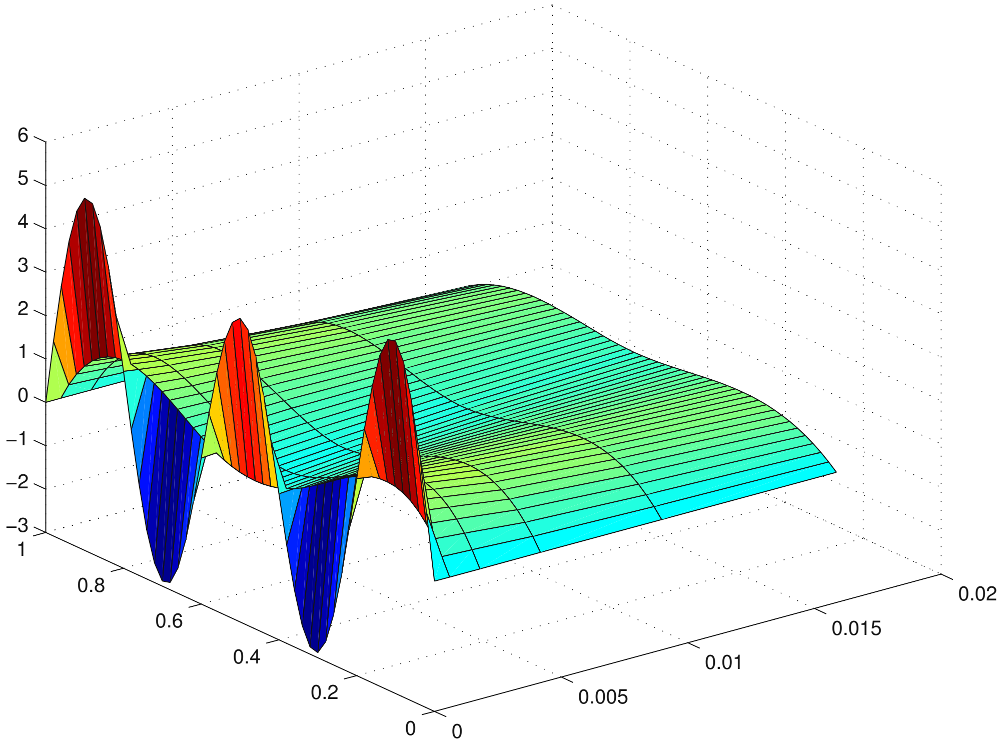

Figure 11.11: Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-2},\ STOL = 10^{-4}$. The endpoint for the time integration was $t = 0.02$.

According to Figure 11.11, the solution may not change as sharply as suggested by Figure 11.9. The tolerances are probably not sufficiently small to correctly reflect the solution behavior around $t = 0$. Therefore, in the next step, we will decrease the tolerances.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Example 1 with $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-3},\ STOL = 10^{-5}$

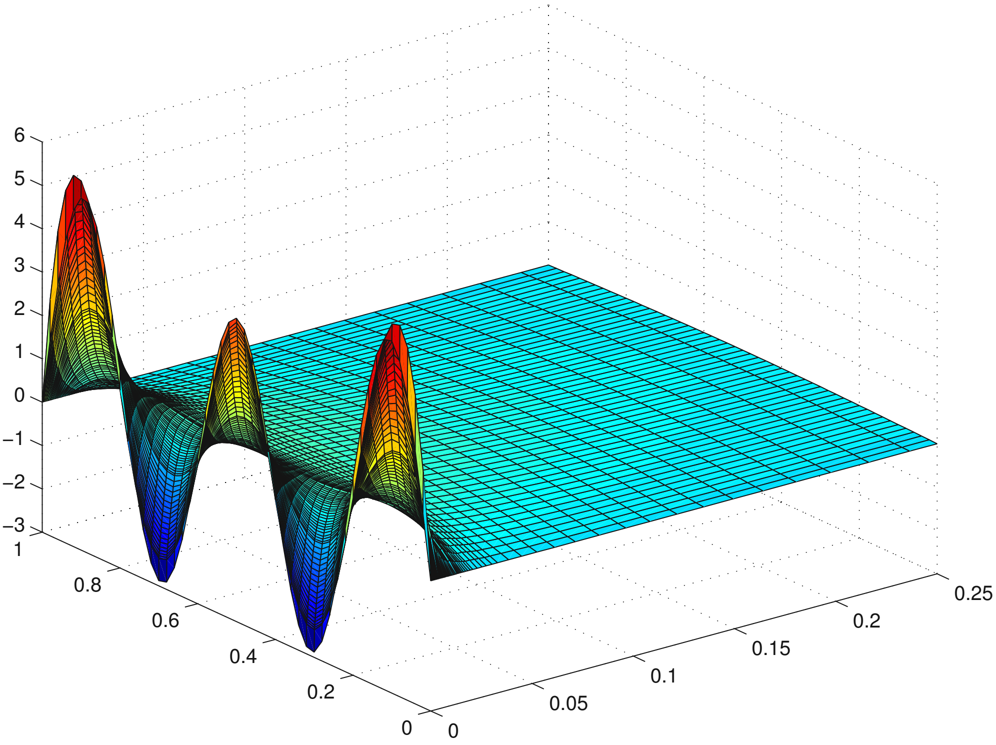

Figure 11.12: Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-3},\ STOL = 10^{-5}$. The endpoint for the time integration was $t = 0.25$.

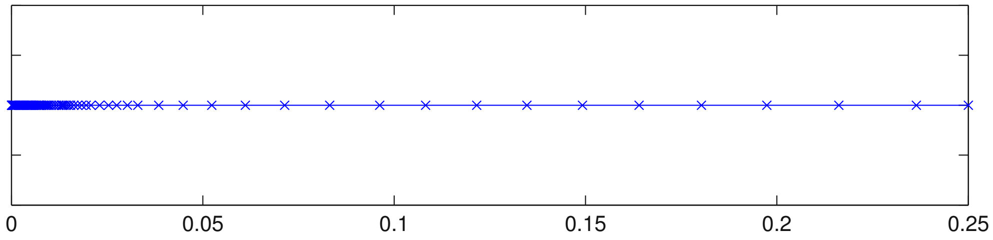

Figure 11.13: Problem (11.6): The time grid with 75 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-3},\ STOL = 10^{-5}$.

The next figure shows that indeed the results obtained $TTOL = 10^{-2}$ and $STOL = 10^{-4}$ are not very accurate. Clearly, the tolerances are too large to resolve the solution behavior, cf. Figures 11.11 and 11.14.

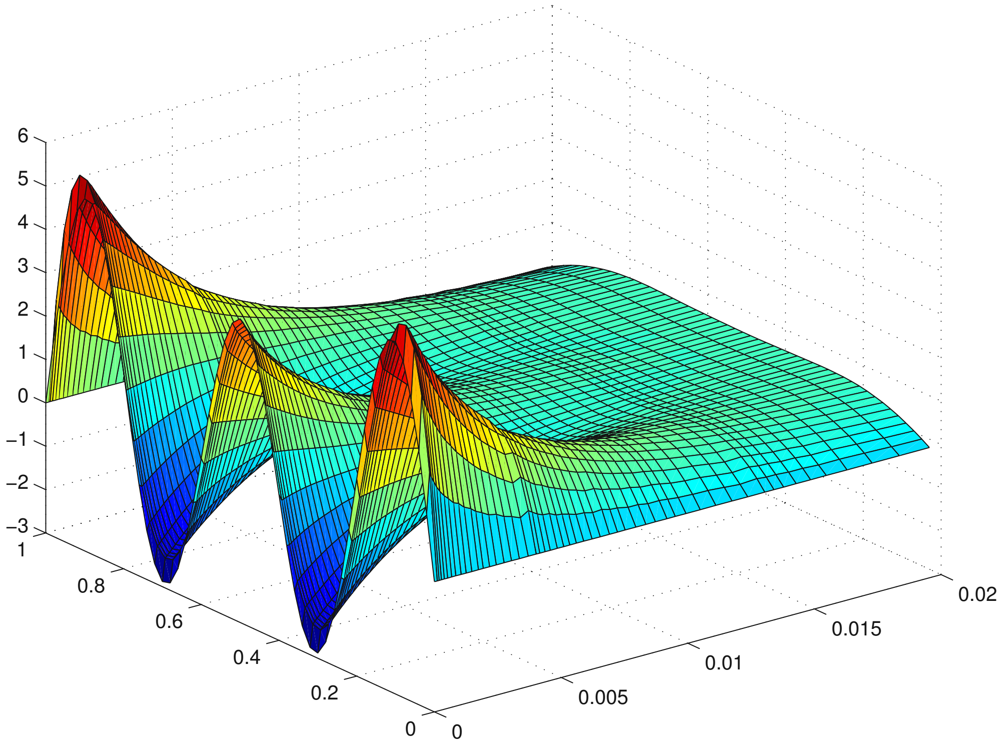

Figure 11.14: Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-3},\ STOL = 10^{-5}$. The endpoint for the time integration was $t = 0.02$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Example 1 with $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-4},\ STOL = 10^{-7}$

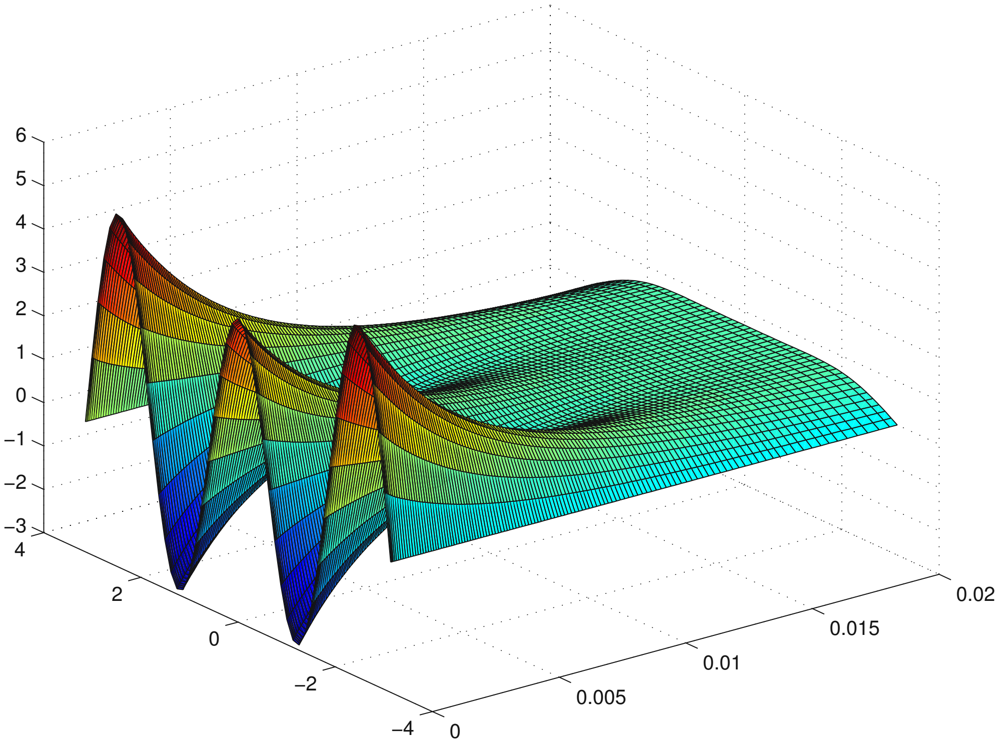

Figure 11.15: Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-4},\ STOL = 10^{-7}$. The endpoint for the time integration was $t = 0.02$.

For this data set, we calculated the solution only for $t \in [0, 0.02]$, in order to more precisely show the starting phase of the computation.
Here 122 steps in time were necessary to reach $t = 0.02$. Note, that when aftre the starting phase the solution smooths out, the grid becomes coarser.

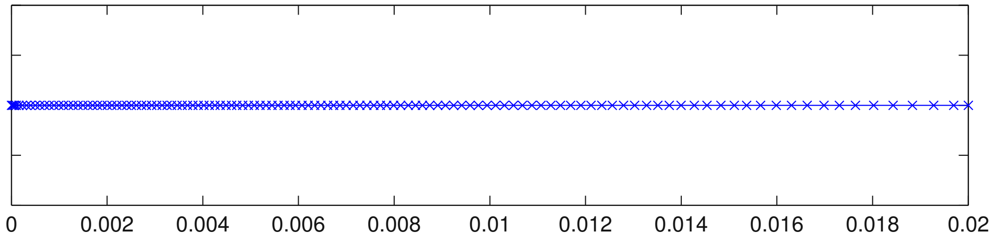

Figure 11.16: Problem (11.6): The time grid with 122 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-4},\ STOL = 10^{-7}$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Summary for Example 1 with $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$

The following table shows quantities analogous to those presented already in Table 11.1. We observe, that although the starting phase of the solution is better resolved using stricter tolerances, the solution values at the time level $t = 0.25$ are comparable.

	2/4	2/5	3/5	3/6
2/4	–	–	–	–
2/5	$3.3 \cdot 10^{-3}$	–	–	–
3/5	$1.9 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	–	–
3/6	$2.3 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$	–

Table 11.2: Problem (11.6): Norm differences between solutions calculated with different tolerances. In the first two columns and in the third column, for solutions related to $TTOL = 10^{-2}$ and $TTOL = 10^{-3}$, we expect to see numbers of the magnitude 10^{-2} and 10^{-3}, respectively.

11.4.2 Example 2: Imaginary Time Schrödinger Equation

Next problem is the imaginary time Schrödinger equation, cf. [42],

$$\frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) + V(x,u(x,t))u(x,t) = 0, \quad x \in [-\pi,\ \pi], \quad t > 0, \quad u(x,0) = u_0(x), \quad (11.7)$$

subject to the following boundary conditions $u(-\pi,t) = 0$, $u(\pi,t) = 0$, $t > 0$. Here $V : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a smooth 2π-periodic potential, $V(x,u) := (1 - \cos x)$, and the initial profile has the form, $u_0(x) = -\sin(4x) + \cos x + 1$. Interestingly, we observe, that decreasing the power of the time tolerance by one means a factor of three in the number of necessary time steps, from 6 to 17, 18 to 55, and 55 to 175.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Solution for Tolerances $TTOL = 10^{-2},\ STOL = 10^{-4}$

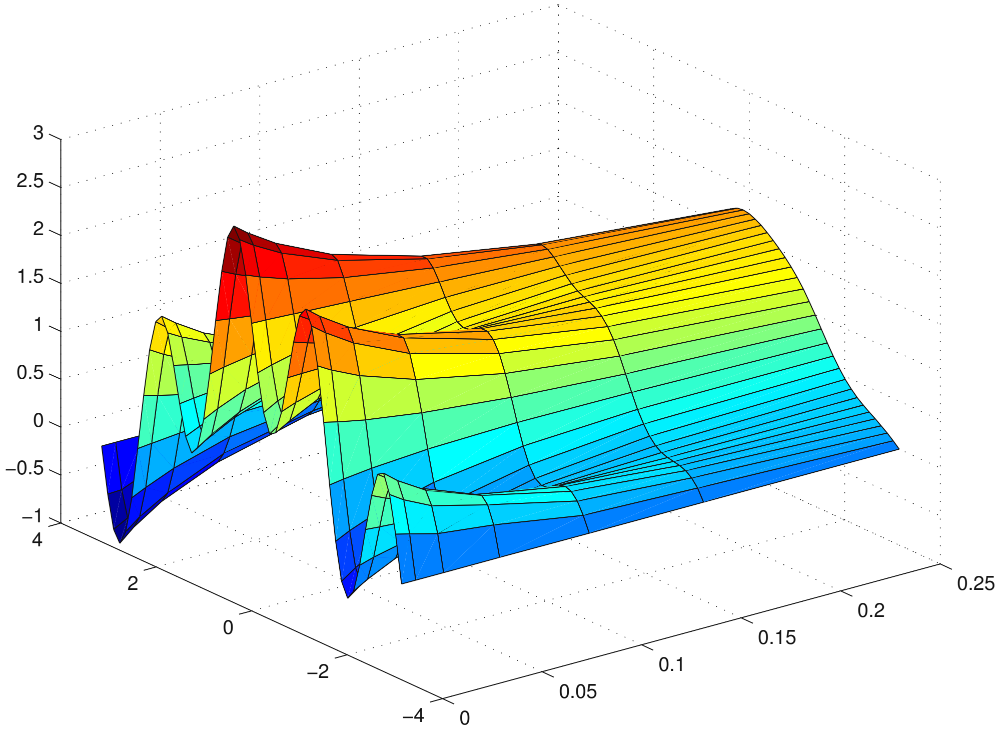

Figure 11.17: Problem (11.7): Solution $u(x,t)$ for $u_0(x) = -\sin(4x) + \cos(x) + 1$ and $TTOL = 10^{-2},\ STOL = 10^{-4}$. The endpoint for the time integration was $t = 0.25$.

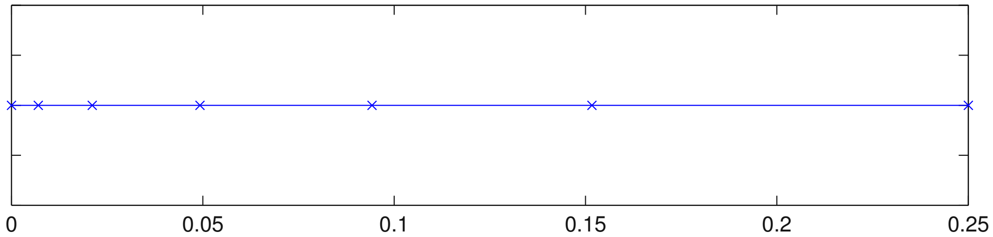

Figure 11.18: Problem (11.7): The time grid with 6 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-2},\ STOL = 10^{-4}$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Solution for Tolerances $TTOL = 10^{-3},\ STOL = 10^{-6}$

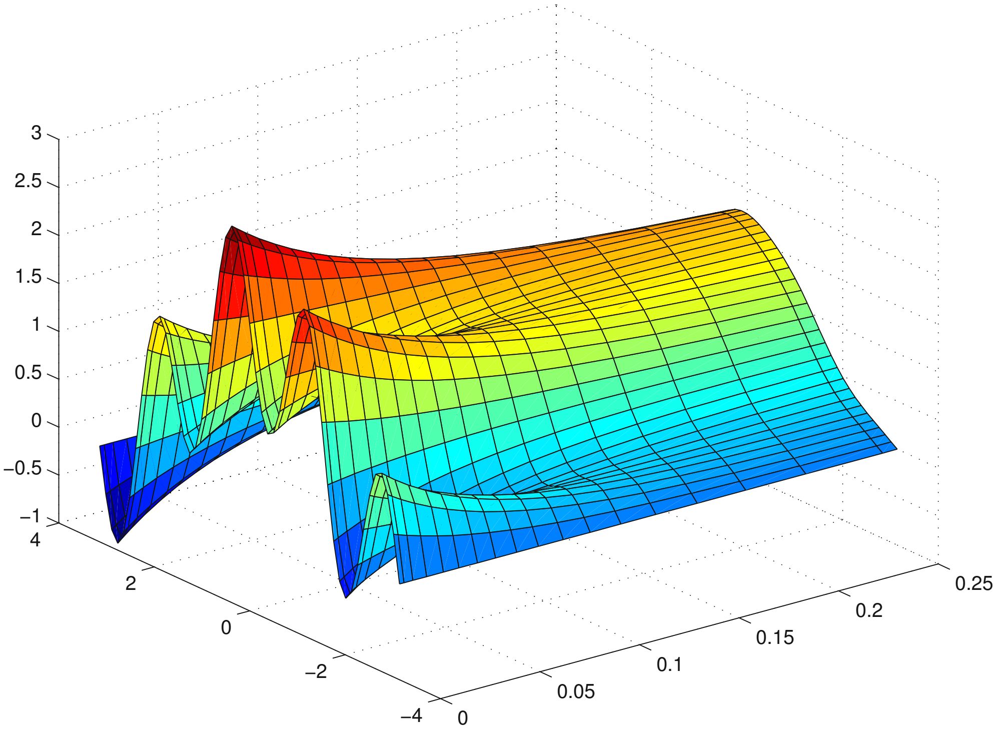

Figure 11.19: Problem (11.7): Solution $u(x,t)$ for $u_0(x) = -\sin(4x) + \cos(x) + 1$ and $TTOL = 10^{-3},\ STOL = 10^{-6}$. The endpoint for the time integration was $t = 0.25$.

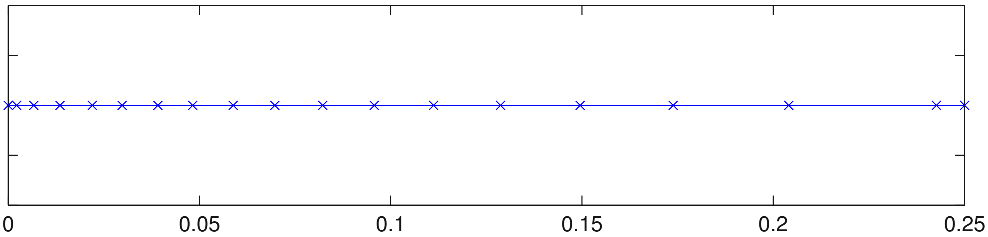

Figure 11.20: Problem (11.7): The time grid with 18 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-3},\ STOL = 10^{-6}$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Solution for Tolerances $TTOL = 10^{-5},\ STOL = 10^{-7}$

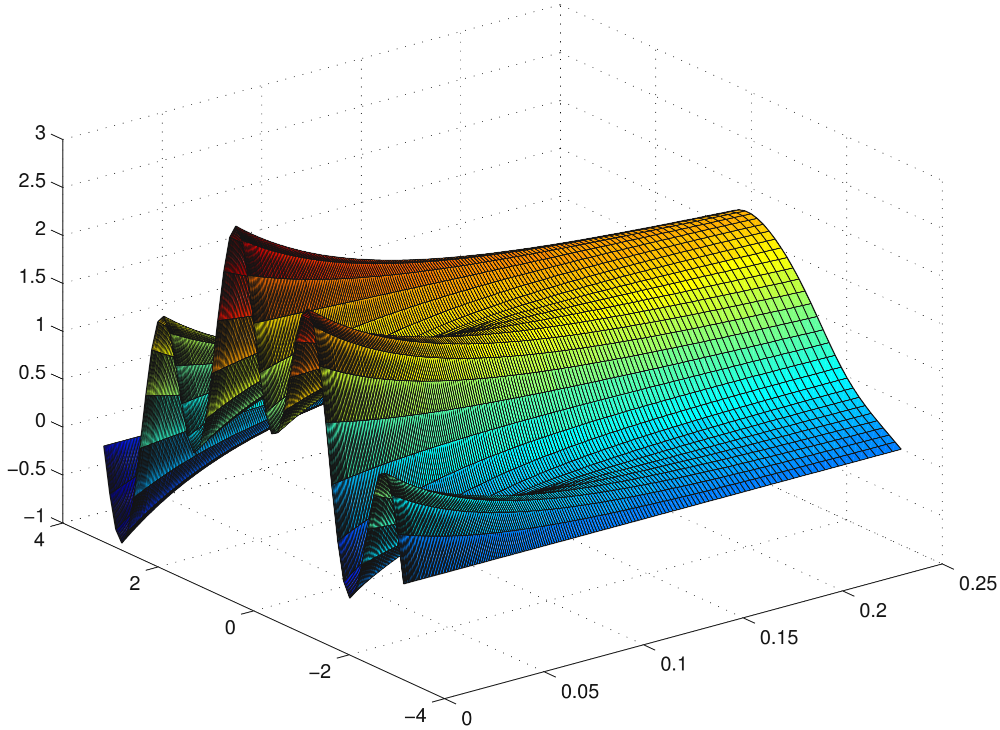

Figure 11.21: Problem (11.7): Solution $u(x,t)$ for $u_0(x) = -\sin(4x) + \cos(x) + 1$ and $TTOL = 10^{-5},\ STOL = 10^{-7}$. The endpoint for the time integration was $t = 0.25$.

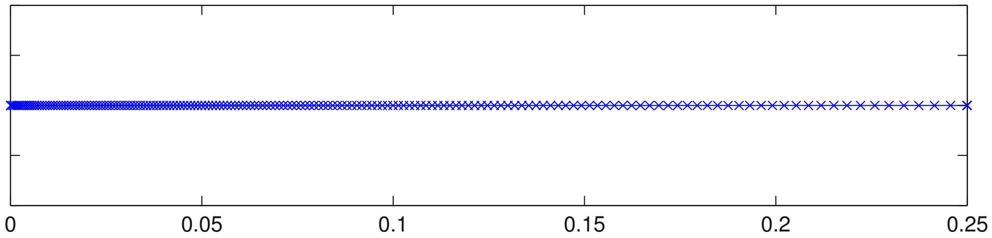

Figure 11.22: Problem (11.7): The time grid with 175 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-5},\ STOL = 10^{-7}$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Summary for Example 2

	2/4	2/5	3/5	3/6	4/6	4/7	5/7
2/4	–	–	–	–	–	–	–
2/5	$1.1 \cdot 10^{-3}$	–	–	–	–	–	–
3/5	$1.4 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	–	–	–	–	–
3/6	$1.4 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$3.4 \cdot 10^{-4}$	–	–	–	–
4/6	$1.9 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$	$4.5 \cdot 10^{-3}$	–	–	–
4/7	$1.8 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$4.2 \cdot 10^{-3}$	$4.3 \cdot 10^{-3}$	$1.5 \cdot 10^{-4}$	–	–
5/7	$2.0 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$5.6 \cdot 10^{-3}$	$5.7 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$	–

Table 11.3: Problem (11.7): Norm differences between solutions calculated with different tolerances. In the first two columns, for solutions related to $TTOL = 10^{-2}$, in the third and fourth columns for $TTOL = 10^{-3}$, in the fifth and sixth columns for $TTOL = 10^{-4}$. We expect to see numbers of the magnitude 10^{-2}, 10^{-3}, and 10^{-4}, respectively.

In Table 11.3, we collected all[1] results for Example 2. The results are similar to those observed for Example 1. The code works dependably and fast, because of the solution structure which is not very challenging.

11.4.3 Example 3: One Dimensional Nonlinear Heat Equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u^3, \quad x \in [0,\ L], \quad t > 0, \quad u(x,0) = u_0(x), \tag{11.8}$$

subject to the boundary conditions,

$$u_x(0,t) = 0, \quad u(L,t) = 0, \quad 0 \le t \le b.$$

Here, $L = 8$, $u_0(x) = 4e^{-x^2}$, and $b = 0.034$.

Solution for Tolerances $TTOL = 10^{-3},\ STOL = 10^{-8}$

Due to a difficult blow-up structure of the solution, see Figure 11.23, we decided to use a very strict tolerance in space in relation to the tolerance in time. This measure was taken to reduce the data error effects when moving from a time step to the next.

---

[1] We have not given all graphs in the text for the sake of brevity.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

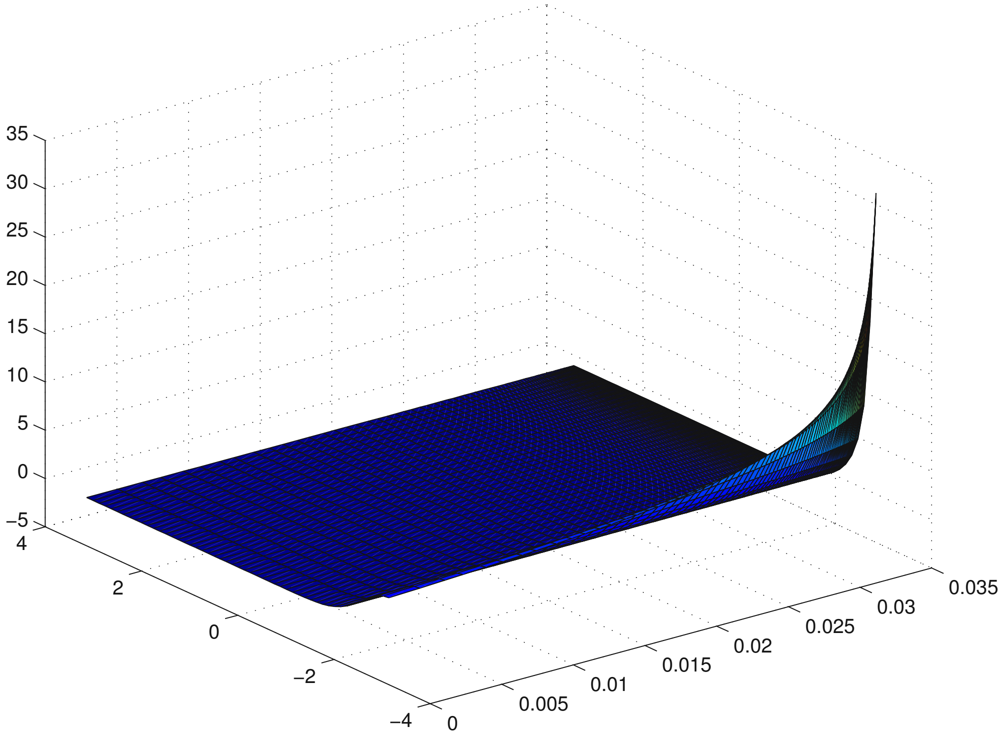

Figure 11.23: Problem (11.8): Solution $u(x,t)$ for $u_0(x) = 4e^{-x^2}$ and $TTOL = 10^{-3}$, $STOL = 10^{-8}$. The endpoint for the time integration was $t = 0.035$.

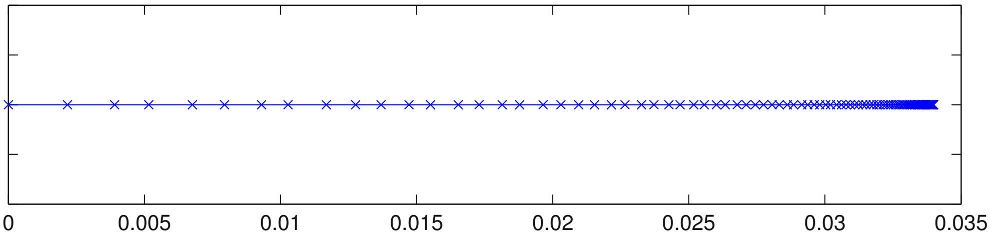

Figure 11.24: Problem (11.8): The time grid with 92 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-3}$, $STOL = 10^{-8}$.

The maximal solution value is around 33. In the last step the size of the step length was $1.82 \cdot 10^{-5}$. The estimate for the next time step length was below the minimal value set to 10^{-8}. For the simulation of a similar problem with much smaller minimal time stepsize, see [26]. There, the maximal solution value was around 10^5.

In the next figure, we show how the time step reduction developed.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

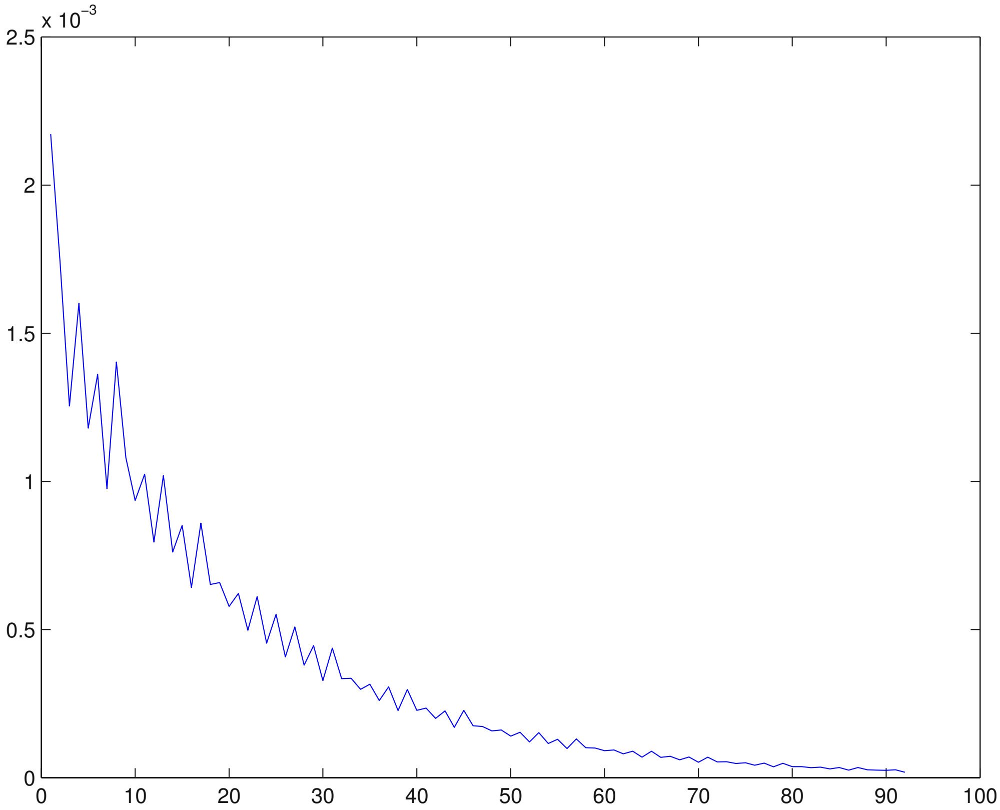

Figure 11.25: Problem (11.8): Time step reduction to reach the tolerances $TTOL = 10^{-3}, \; STOL = 10^{-8}$.

The following two models are problems involving Neumann boundary condition at the right boundary.

11.4.4 Example 4: Heat Equation with Neumann BC1

Here, we consider the following problem:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - 2\frac{\partial u}{\partial x}, \quad x \in [0, \; \pi], \quad t > 0, \quad u(x,0) = u_0(x) = x^2 - \pi x, \tag{11.9}$$

subject to the boundary conditions,

$$u(0,t) = 0, \quad u_x(\pi, t) = 0.$$

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Solution for Tolerances $TTOL = 10^{-3},\ STOL = 10^{-8}$

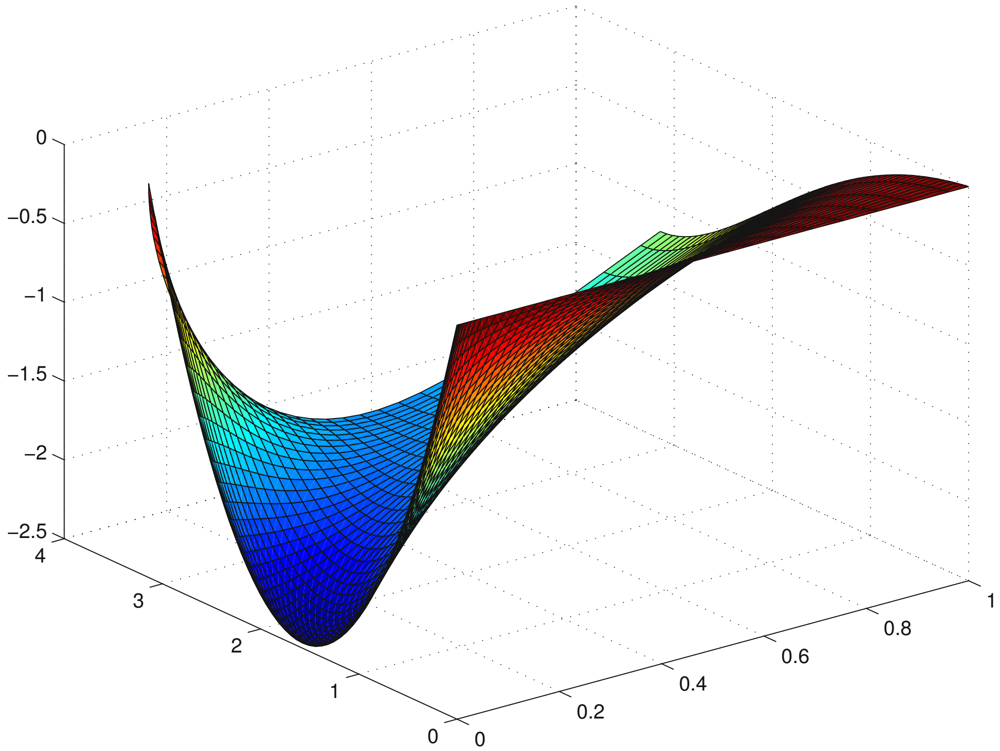

Figure 11.26: Problem (11.9): Solution $u(x,t)$ for $u_0(x) = x^2 - \pi x$ and $TTOL = 10^{-3},\ STOL = 10^{-8}$. The endpoint for the time integration was $t = 1$.

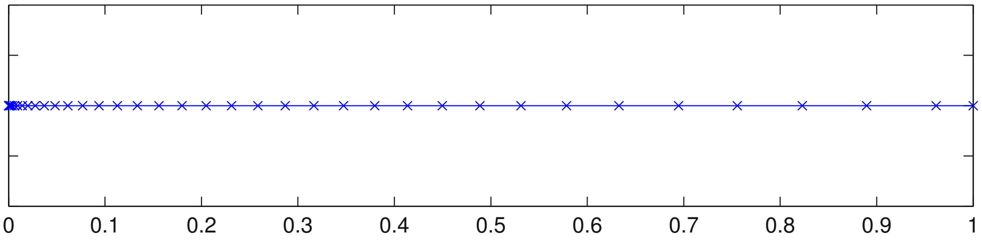

Figure 11.27: Problem (11.9): The time grid with 38 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-3},\ STOL = 10^{-8}$.

11.4.5 Example 5: Heat Equation with Neumann BC2

Finally, we simulate the problem,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad x \in [0,\ 1], \quad t > 0, \quad u(x,0) = u_0(x) = x^2 - x, \tag{11.10}$$

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

subject to the boundary conditions of the form,

$$u(0,t) = 0, \quad u_x(1,t) = 0.$$

Solution for Tolerances $TTOL = 10^{-3},\ STOL = 10^{-6}$

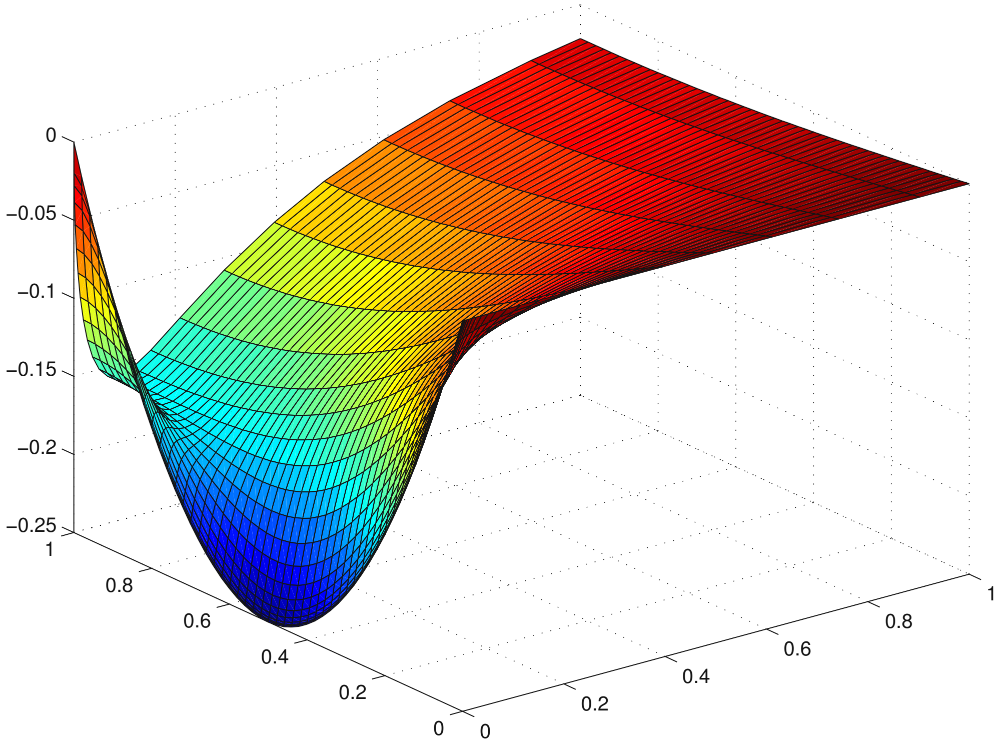

Figure 11.28: Problem (11.10): Solution $u(x,t)$ for $u_0(x) = x^2 - x$ and $TTOL = 10^{-3},\ STOL = 10^{-6}$. The endpoint for the time integration was $t = 1$.

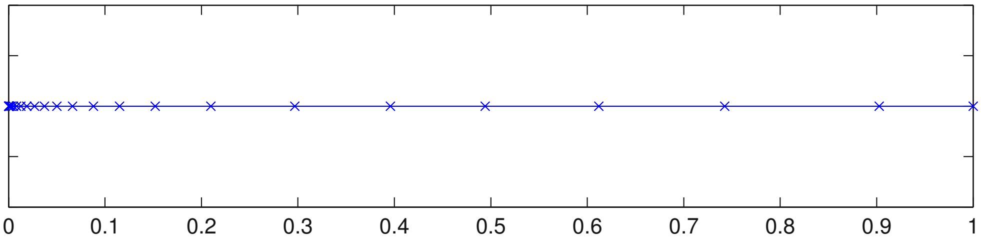

Figure 11.29: Problem (11.10): The time grid with 23 points used by the `time-traveller` to provide a solution satisfying the tolerances $TTOL = 10^{-3},\ STOL = 10^{-6}$.

11.4.6 Conclusions

Numerical simulations of the above model problems show that our test code `time-traveller` can cope with them in a dependable and robust manner for a wide range of tolerances.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Some additional information about the code was provided in [26]. In this paper the properties of our fully adaptive space-time discretization for a class of nonlinear heat equations was investigated. From the known scaling laws, the theoretically optimal grids implying error equidistribution are described using the model problem,

$$u_t(x,t) = u_{xx}(x,t) + u^3(x,t), \quad u_x(0,t) = u(4,t) = 0, \quad u(x,0) = u_0(x) = 4e^{-x^2},$$

whose solution shows a very rapid growth in time. The investigation suggest that optimal grids derived from known scaling lows for the analytical problem and providing an efficient solution through error equidistribution can be observed in the practical realization of `time-traveller`, see the end of Chapter 3.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

References

[1] R. Agarwal, D. O'Reagan, I. Rachůnková, and M. Staněk, *Two-point higher order BVPs with singularities in phase variable*, Computers and Mathematics with Applications, 46 (2003), pp. 1799–1826.

[2] M. Ainsworth and J. Oden, *A posteriori error estimation in finite element analysis*, Wiley-Interscience, New York, 2000.

[3] C. Arévalo, J. López, and G. Söderlind, *Linear multistep methods with constant coefficients and step density control*, J. Comp. Appl. Math. 205 (2007), pp. 891–900.

[4] 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.

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

[6] W. Auzinger, E. Karner, O. Koch, D. Praetorius, and E. Weinmüller, *Globale Fehlerschätzer für Randwertprobleme mit einer Singularität zweiter Art*, Techn. Rep. ANUM Preprint Nr. 6/03, Inst. for Appl. Math. and Numer. Anal., Vienna Univ. of Technology, Austria, 2003. Available at `http://www.math.tuwien.ac.at/~inst115/preprints.htm`.

[7] W. Auzinger, G. Kneisl, O. Koch, and E. Weinmüller, *A solution routine for singular boundary value problems*, Techn. Rep. ANUM Preprint Nr. 1/02, Inst. for Appl. Math. and Numer. Anal., Vienna Univ. of Technology, Austria, 2002. Available at `http://www.math.tuwien.ac.at/~inst115/preprints.htm`.

[8] W. Auzinger, G. Kneisl, O. Koch, and E. Weinmüller, *A collocation code for boundary value problems in ordinary differential equations*, Numer. Algorithms, 33 (2003), pp. 27–39.

[9] W. Auzinger, O. Koch, D. Praetorius, G. Pulverer, and E. Weinmüller, *Performance of collocation software for singular BVPs*, Techn. Rep. ANUM Preprint Nr. 4/04, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria, 2004. Available at `http://www.math.tuwien.ac.at/~inst115/preprints.htm`.

[10] 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.

[11] ——, *Collocation methods for boundary value problems with an essential singularity*, in Large-Scale Scientific Computing, I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, eds., vol. 2907 of Lecture Notes in Computer Science, Springer Verlag, 2004, pp. 347–354.

[12] ——, *Analysis of a new error estimate for collocation methods applied to singular boundary value problems*, SIAM J. Numer. Anal., 42 (2005), pp. 2366–2386.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[13] ——, *Efficient mesh selection for collocation methods applied to singular BVPs*, J. Comput. Appl. Math., 180 (2005), pp. 213–227.

[14] P. Bailey, W. Everitt, and A. Zettl, *Computing eigenvalues of singular Sturm-Liouville problems*, Results in Mathematics, 20 (1991), pp. 391–423.

[15] R. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Math. Comp., 44 (1985), pp. 283–301.

[16] Z. Bashir Ali, *Numerical Solution of Parameter Dependent Two-Point BVPs Using Iterated Deferred Correction*, ph. d. thesis, Imperial College of Science, Technology and Medicine, London, U.K., 1998.

[17] R. Bird, W. Stewart, and E. Lightfoot, *Transport phenomena*, John Wiley & Sons, New York, 2002.

[18] C. Budd and R. Kuske, *Localised periodic pattern for the non-symmetric generalized Swift-Hohenberg equations*, Physica D, 208 (2005), pp. 73–95.

[19] C. Budd, V. Rothschafer, and J. Williams, *Multi-bump self-similar solutions of the complex Ginsburg Landau equations*, SIAM J. Dyn. Sys., 4 (2005), pp. 649–678.

[20] C. Budd and J. Williams, *Parabolic monge-ampère methods for blow-up problems in several spatial dimensions*, Journal of Physics A, 39 (2006), pp. 5425–5463.

[21] C. J. Budd, *Asymptotics of multibump blow-up self-similar solutions of the nonlinear Schrödinger equation*, SIAM J. Appl. Math., 62 (2001), pp. 801–830.

[22] C. J. Budd, O. Koch, and E. Weinmüller, *Self-similar blow-up in nonlinear PDEs*, AURORA TR-2004-15, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria, 2004. Available at `http://www.vcpc.univie.ac.at/aurora/publications/`.

[23] ——, *Computation of self-similar solution profiles for the nonlinear Schrödinger equation*, Computing, 77 (2006), pp. 335–346.

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

[25] C. J. Budd, V. Rottschäfer, and J. F. Williams, *Multi-bump, blow-up, self-similar solutions of the complex Ginzburg-Landau equation*, SIAM J. Appl. Dyn. Syst., 4 (2005), pp. 649–678.

[26] C. J. Budd, O. Koch, L. Taghizadeh, and Ewa Weinmüller, *Asymptotic properties of the space-time adaptive numerical solution of a nonlinear heat equation*, Calcolo 55 (2018), article 43, 14 pages, https://doi.org/10.1007/s10092-018-0286-zlo.

[27] P. Burrage, R. Herdiana, and K. Burrage, *Adaptive stepsize based on control theory for stochastic differential equations*, J. Comput. Appl. Math., 170 (2004), pp. 317 336.

[28] C. Carstensen, *Some remarks on the history and future of averaging techniques in a posteriori finite element error analysis*, ZAMM J. Appl. Math. Mech., 84, (2004), pp 3–21. pp. 3–21.

[29] C. Christara and K. Ng, *Adaptive techniques for spline collocation*, Computing, 76 (2005), pp. 259–277.

[30] M. Drmota, R. Scheidl, H. Troger, and E. Weinmüller, *On the imperfection sensitivity of complete spherical shells*, Comp. Mech., 2 (1987), pp. 63–74.

[31] B. Finlayson, *Nonlinear analysis in chemical engineering*, McGraw-Hill Inc., New York, 1980.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[32] G. Froment and K. Bischoff, *Chemical reactor analysis and design*, John Wiley & Sons Inc., New York, 1990.

[33] F. Frommlet and E. Weinmüller, *Asymptotic error expansions for singular boundary value problems*, Math. Models and Meth. Appl. Sci., 11 (2001), pp. 71–85.

[34] S. Golub, *Measures of restrictions in inward foreign direct investment in OECD countries*, OECD Economics Dept. WP Nr. 357.

[35] K. Gustafsson, *Control theoretic techniques for stepsize selection in implicit Runge-Kutta methods*, ACM TOMS, 20 (1994), pp. 496–517.

[36] K. Gustafsson and G. Söderlind, *Control strategies for the iterative solution of nonlinear equations in ODE solvers*, SIAM J. Sci. Comput., 18 (1997), pp. 23–40.

[37] E. Helpman, M. Melitz, and Yeaple, *Export versus FDI with heterogeneous firms*, Amer. Econ. Rev., 94 (2004), pp. 300–316.

[38] F. d. 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.

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

[40] ——, *On the boundary value problem for systems of ordinary differential equations with a singularity of the second kind*, SIAM J. Math. Anal., 11 (1980), pp. 41–60.

[41] ——, *The application of Runge-Kutta schemes to singular initial value problems*, Math. Comp., 44 (1985), pp. 93–103.

[42] T. Jahnke, C. Lubich, *Error Bounds for Exponential Operator Splittings*, BIT Numer. Math., 40 (2000), pp. 735–744.

[43] R. Kannan and D. O'Reagan, *Singular and nonsingular boundary value problems with sign changing nonlinearities*, J. Ineq. Appl., 5 (2000), pp 621–637.

[44] T. Kapitula, *Existence and stability of singular heteroclinic orbits for the Ginzburg-Landau equation*, Nonlinearity, 9 (1996), pp. 669–685.

[45] B. Karabay, *Foreign direct investment and host country policies: A rationale for using ownership restrictions*, Techn. Report, University of Virginia, WP, 2005.

[46] G. Kitzhofer, *Numerical Treatment of Implicit Singular BVPs*, Ph.D. Thesis, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria 2017.

[47] G. Kitzhofer and E. Koch, O. Weinmüller, *Pathfollowing for essentially singular boundary value problems with application to the complex Ginzburg–Landau equation*, BIT Numer. Math. 49 (2009), pp. 217–245.

[48] 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. Available at `http://www.othmar-koch.org/research.html`.

[49] G. Kitzhofer, O. Koch, and E. Weinmüller, *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`.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[50] ——, *Kollokationsverfahren für singuläre Randwertprobleme zweiter Ordnung in impliziter Form*, Techn. Rep. ANUM Preprint Nr. 9/04, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria, 2004. Available at `http://www.math.tuwien.ac.at/~inst115/preprints.htm`.

[51] G. Kitzhofer, O. Koch, G. Pulverer, Ch. Simon, E.B. Weinmüller, *The New Matlab Code bvpsuite for the Solution of Singular Implicit BVPs*, JNAIAM J. Numer. Anal. Indust. Appl. Math. 5 (2010), pp. 113-134.

[52] O. Koch, *Asymptotically correct error estimation for collocation methods applied to singular boundary value problems*, Numer. Math., 101 (2005), pp. 143–164.

[53] O. Koch and E. Weinmüller, *The convergence of shooting methods for singular boundary value problems*, Math. Comp., 72 (2003), pp. 289–305.

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

[55] 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.

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

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

[58] A. Papastavrou and R. Verfürth, *A posteriori error estimators for stationary convection-diffusion problems: a computational comparison*, Comput. Methods Appl. Mech. Eng., 189 (2000), pp. 449–462.

[59] W.H. Press, B.P. Flannery, S.A. Teukosky, W.T. Vetterling, Numerical Recipes in C, The Art of Scientific Computing, Cambridge University Press 1988.

[60] G. Pulverer, G. Söderlind, and E. Weinmüller, *Automatic grid control in adaptive BVP solvers*, Numer. Algorithms, 56 (2011), pp. 61–92.

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

[62] I. Rachůnková, S. Staněk, and M. Tvrdý, *Singularities and Laplacians in Boundary Value Problems for Nonlinear Ordinary Differential Equations*, vol. 3 of Handbook of Differential Equations. Ordinary Differential Equations, Elsevier, 2006, pp. 607–723.

[63] V. Ranade, *Computational flow modeling for chemical engineering*, Academic Press, San Diego, 2002.

[64] L. Shampine, J. Kierzenka, and M. Reichelt, *Solving Boundary Value Problems for Ordinary Differential Equations in* Matlab *with bvp4c*, 2000. Available at `http://www.mathworks.com/bvp_tutorial`.

[65] L. Shampine, P. Muir, and H. Xu, *A User-Friendly Fortran BVP Solver*, Available at `http://http://cs.stmarys.ca/~muir/JNAIAM_Shampine_Muir_Xu2006.pdf`, (2006).

[66] G. Söderlind, *Time-step selection algorithms: adaptivity, control, and signal processing*, Appl. Numer. Math., 56 (2006), pp. 488–502.

[67] ——, *Automatic control and adaptive time–stepping*, Numer. Algorithms, 31 (2002), pp. 281-308.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[68] ——, *Digital filters in adaptive time-stepping*, ACM Trans. Math. Software, 29 (2003), pp. 1–26.

[69] G. Söderlind and L. Wang, *Adaptive time–stepping and computational stability*, J. Comp. Appl. Math., 185 (2006), pp. 225–243.

[70] K. Sundmacher and U. Hoffmann, *Multicomponent mass and energy transport on different length scales in a packed reactive distillation column for heterogeneously catalyzed fuel ether production*, Chem. Eng. Sci., 49 (1994), pp. 4443–4464.

[71] J. d. Swart and G. Söderlind, *On the construction of error estimators for implicit runge–kutta methods*, J. Comp. Appl. Math., 86 (1997), pp. 347–358.

[72] R. Verfürth, *A posteriori error estimates for finite element discretizations of the heat equation*, Calcolo, 40 (2003), pp. 195–212.

[73] ——, *A posteriori error estimates for linear parabolic equations*, Technical Report, 2004.

[74] ——, *Robust a posteriori error estimates for nonstationary convection-diffusion equations*, SIAM J. Numer. Anal., 43 (2005), pp. 1783–1802.

[75] A. Verhoeven, *Automatic control for adaptive time stepping in electrical circuit simulation*, Philips Research Report, TN-2004/00033 (2004).

[76] A. Verhoeven, T. Beelen, M. Hautus, and E. ter Maten, *Digital linear control theory for automatic stepsize control*, In: Anile A.M., Al G., Mascali G. (eds) Scientific Computing in Electrical Engineering. Mathematics in Industry, vol 9. Springer, Berlin, Heidelberg (2006), pp 137–142.

[77] P. Wissgott, *Adaptive finite element method for linear parabolic equations*, Technical Report, 2007.

[78] C.-Y. Yeh, A.-B. Chen, D. Nicholson, and W. Butler, *Full-potential Korringa-Kohn-Rostoker band theory applied to the Mathieu potential*, Phys. Rev. B, 42 (1990), pp. 10976–10982.

[79] O. Zienkiewicz and J. Z. Zhu, *A simple error estimator and adaptive procedure for practical engineering analysis*, Int. J. Numer. Meth. Eng., 2 (1987), pp. 337–357.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

List of Tables

11.1 Problem (11.6) for $u_0(x) = 4x - 4x^2$: Norm differences between solutions calculated with different tolerances . . . 227
11.2 Problem (11.6) for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$: Norm differences between solutions calculated with different tolerances . . . 232
11.3 Problem (11.7): Norm differences between solutions calculated with different tolerances 236

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

List of Figures

2.1 Block diagram of the grid adaptation 8

2.2 Computational stability: Achieved accuracy as a function of tolerance when an ODE was solved, using the same code, with conventional stepsize control (left), and digital filter control (right) 8

3.1 Problem (3.1): The solution of the problem. At $t_{end} = 0.0325129$ the solution maximum is $u_{max} = 7.45 \cdot 10^5$ 16

3.2 Problem (3.1): The evolution in time for $\max |u_i|^2 \times \Delta t^{(n)}$ (top) and the evolution in time for $\max |u_i| \times \min \Delta x^{(n)}$ (bottom). Here, $\max |u_i|$ is the maximal solution value and Δt, Δx are stepsizes in time and space, respectively. Moreover, $aTOL = 10^{-4}$, $rTOL = 10^{-6}$ and $t_{end} = 0.0325129$, reached after 105 steps 16

11.1 Flow-Chart for timetraveller.m 216

11.2 File Calls 223

11.3 Problem (11.6): Solution $u(x,t)$ for $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-2}$, $STOL = 10^{-4}$ 224

11.4 Problem (11.6): The time grid with 7 points to reach the tolerances $TTOL = 10^{-2}$, $STOL = 10^{-4}$ 224

11.5 Problem (11.6): Solution $u(x,t)$ for $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-3}$, $STOL = 10^{-5}$ 225

11.6 Problem (11.6): The time grid with 20 points to reach the tolerances $TTOL = 10^{-3}$, $STOL = 10^{-5}$ 225

11.7 Problem (11.6): Solution $u(x,t)$ for $u_0(x) = 4x - 4x^2$ and $TTOL = 10^{-4}$, $STOL = 10^{-7}$ 226

11.8 Problem (11.6): The time grid with 68 points to reach the tolerances $TTOL = 10^{-4}$, $STOL = 10^{-7}$ 226

11.9 Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-2}$, $STOL = 10^{-4}$ 227

11.10 Problem (11.6): The time grid with 12 points to reach the tolerances $TTOL = 10^{-2}$, $STOL = 10^{-4}$ 228

11.11 Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-2}$, $STOL = 10^{-4}$ 228

11.12 Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-2}$, $STOL = 10^{-4}$ 229

11.13 Problem (11.6): The time grid with 75 points to reach the tolerances $TTOL = 10^{-3}$, $STOL = 10^{-5}$ . . . 229
11.14 Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-3}$, $STOL = 10^{-5}$ . . . 230
11.15 Problem (11.6): Solution $u(x,t)$ for $u_0(x) = \sin(\pi x) + \sin(3\pi x) + 4\sin(5\pi x)$ and $TTOL = 10^{-4}$, $STOL = 10^{-7}$ . . . 231
11.16 Problem (11.6): The time grid with 122 points to reach the tolerances $TTOL = 10^{-4}$, $STOL = 10^{-7}$ . . . 231
11.17 Problem (11.7): Solution $u(x,t)$ for $u_0(x) = -\sin(4x) + \cos(x) + 1$ and $TTOL = 10^{-2}$, $STOL = 10^{-4}$ . . . 233
11.18 Problem (11.7): The time grid with 6 points to reach the tolerances $TTOL = 10^{-2}$, $STOL = 10^{-4}$ . . . 233
11.19 Problem (11.7): Solution $u(x,t)$ for $u_0(x) = -\sin(4x) + \cos(x) + 1$ and $TTOL = 10^{-3}$, $STOL = 10^{-6}$ . . . 234
11.20 Problem (11.7): The time grid with 18 points to reach the tolerances $TTOL = 10^{-3}$, $STOL = 10^{-6}$ . . . 234
11.21 Problem (11.7): Solution $u(x,t)$ for $u_0(x) = -\sin(4x) + \cos(x) + 1$ and $TTOL = 10^{-5}$, $STOL = 10^{-7}$ . . . 235
11.22 Problem (11.7): The time grid with 175 points to reach the tolerances $TTOL = 10^{-5}$, $STOL = 10^{-7}$ . . . 235
11.23 Problem (11.8): Solution $u(x,t)$ for $u_0(x) = 4e^{-x^2}$ and $TTOL = 10^{-3}$, $STOL = 10^{-8}$ 237
11.24 Problem (11.8): The time grid with 92 points to reach the tolerances $TTOL = 10^{-3}$, $STOL = 10^{-8}$ . . . 237
11.25 Problem (11.8): Time step reduction to reach the tolerances $TTOL = 10^{-3}$, $STOL = 10^{-8}$ . . . 238
11.26 Problem (11.9): Solution $u(x,t)$ for $u_0(x) = x^2 - \pi x$ and $TTOL = 10^{-3}$, $STOL = 10^{-8}$ 239
11.27 Problem (11.9): The time grid with 38 points to reach the tolerances $TTOL = 10^{-3}$, $STOL = 10^{-8}$ . . . 239
11.28 Problem (11.10): Solution $u(x,t)$ for $u_0(x) = x^2 - x$ and $TTOL = 10^{-3}$, $STOL = 10^{-6}$ 240
11.29 Problem (11.10): The time grid with 23 points to reach the tolerances $TTOL = 10^{-3}$, $STOL = 10^{-6}$ . . . 240

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub