Asymptotic properties of the space-time adaptive numerical solution of a nonlinear heat equation

Chris Budd · **Othmar Koch** · **Leila Taghizadeh** ·
Ewa Weinmüller

January 11, 2021

Abstract We consider the fully adaptive space-time discretization of a class of nonlinear heat equations by Rothe's method. Space discretization is based on adaptive polynomial collocation which relies on equidistribution of the defect of the numerical solution, and the time propagation is realized by an adaptive backward Euler scheme. From the known scaling laws, we infer theoretically the optimal grids implying error equidistribution, and verify that our adaptive procedure closely approaches these optimal grids.

Keywords Evolution equations · Rothe's method · collocation methods · backward Euler method · adaptivity

Mathematics Subject Classification (2000) 65M20, 65L05, 65L10, 65L50

1 Introduction

Evolutionary problems described by nonlinear parabolic partial differential equations (PDEs) of the form

$$u_t = f(x, u, u_x, u_{xx}), \quad x \in (a, b), \tag{1.1}$$

arise in a wide variety of applications, including combustion, chemical reactions, mathematical biology, meteorology and gas dynamics. Such problems are often solved by using the method lines in *time*, with a spatial discretisation using a finite difference, finite element or collocation method. If (as is often the case) the solution of the PDE has structure which evolves on small length scales, including the formation of singularities in finite time, then

Chris Budd
Mathematical Sciences, University of Bath, Bath BA2 7AY, United Kingdom
E-mail: mascjb@bath.ac.uk

Othmar Koch
Institut für Mathematik, Universität Wien, Oskar Morgenstern-Platz 1, A-1090 Wien
E-mail: othmar@othmar-koch.org

Leila Taghizadeh · Ewa Weinmüller
Institut für Analysis und Scientific Computing, Technische Universität Wien
Wiedner Hauptstrasse 8–10/E101, 1040 Wien, Austria
E-mail: e.weinmueller@tuwien.ac.at, leila.taghizadeh@tuwien.ac.at

some form of adaptivity is often needed to resolve this structure. Typical methods include moving/relocating mesh points (r-adaptivity) or locally adding mesh points (h-adaptivity). Neither method is perfect, as r-adaptive methods can suffer from having too few points to resolve the structure (even if the mesh points are moved to optimal locations), and h-adaptive methods can suffer from a loss of global regularity, with poorly graded and irregular meshes. This can lead to unacceptable errors in the solution.

In this paper we consider an alternative approach to solving problems of the form (1.1) which uses an implicit finite difference approach in *time* and the *method of lines in space* (also known as Rothe's method) combined with an adaptive approach in both time and space. This method will be described in detail in Section 3, but in summary the method works by replacing (1.1) by the finite difference in time discretisation given by the implicit θ-method

$$u^{n+1} = u^n + \Delta t_n \left[\theta f(x, u^{n+1}, u_x^{n+1}, u_{xx}^{n+1}) + (1-\theta) f(x, u^n, u_x^n, u_{xx}^n)\right] \qquad x \in (a,b), \quad (1.2)$$

where

$$u^n(x) \approx u\left(x, t_n\right). \quad (1.3)$$

The function $u^{n+1}(x)$ then satisfies a two-point boundary value problem in x, of the form

$$g(x, u^{n+1}, u_x^{n+1}, u_{xx}^{n+1}) = h(x) \quad (1.4)$$

which can be conveniently solved using a collocation or related method in which $u^{n+1}(x)$ is represented continuously as a *piece-wise smooth polynomial* in x over an appropriate mesh $X_N^{n+1} = (x_0^{n+1}, x_1^{n+1}, \ldots, x_i^{n+1}, \ldots x_N^{n+1})$.

The advantage of this method is that the smooth representation of the function $u^{n+1}(x)$ at each time level allows for a significant degree of flexibility in choosing an effective mesh on which to represent the solution of (1.2,1.4). In particular at each time level a preliminary solution u^* can be constructed and the error defect evaluated. The mesh points x_{i+1}^n can then be moved to equidistribute this error defect and hence minimise the maximum defect. If this is too large then further points can be added in a smooth manner by equidistributing the mesh density function. This whole process can then be iterated until the global defect error is below a suitable error tolerance. This procedure delivers a smoothly graded mesh which also resolves the local structures of the solution.

In particular, suppose that we solve a problem with N_n mesh points at time t_n with underlying solution $u(x,t)$. The ideal action of an effective r-adaptive method Huang and Russell (2010), when used for example to find solutions which become singular as time evolves, is to place the N_n mesh points *optimally* so that the maximum global defect error $\mathscr{E}^n$ at time t_n should take the form

$$\mathscr{E}^n < \frac{C^n}{N_n^\alpha} \quad (1.5)$$

where the constant C^n is (to the extent possible) *independent of the solution* and in particular does not (if possible) increase (without bound) as the length scales in the solution decrease (to zero) as t_n increases. Of course this is not always achievable, especially if $\mathscr{E}^n$ is considered to be the *absolute or mixed* error rather than the *relative* error of the solution. The action of the adaptive step is then to find the *best* (ie. the lowest) such constant C^n at the time t_n. The flexibility of the h-r approach over a purely r-adaptive approach, is that by changing N_n at each time level we can guarantee that $\mathscr{E}^n$ is below a prescribed tolerance. Of course the combination of the r and h adaptive approaches should yield a smaller error with a smaller number of mesh points, than if either was used on its own.

If our adaptive strategy indeed reflects correctly the solution profile and ensuing error behavior, the global error of the numerical solution should be roughly equidistributed as a function of the spatial variable. According to (Ascher et al., 1995, (9.15)), the error of a collocation solution u^n at m collocation points with global (superconvergence) order p applied to a second-order problem has the form

$$u(x_{i+1},t_n) - u^n(x_{i+1}) = \mathscr{C}^n \cdot (x_{i+1} - x_i)^{m+2} u^{(m+2)}(x_{i+1}) + O(h^p), \tag{1.6}$$

where h is the maximal step-size in the adaptive mesh and $\mathscr{C}^n$ is independent of u. With the aim to equidistribute the error over the spatial mesh for the problem (1.7), the relationship (4.5) below for the mesh width should be aspired for in order to balance the terms in the error representation (1.6). This error estimate will be used both as a guide to the construction of the spatial adaptivity strategy and also a test of the accuracy of the approach that we consider. The consequences of the error representation (1.6) will be explained in detail in Section 4 below.

In this paper we will apply this method to the well known nonlinear heat equation

$$\begin{aligned} u_t &= u_{xx} + f(u), \\ u(0,t) &= u(1,t) = 0, \\ u(x,0) &= u_0(x), \end{aligned} \tag{1.7}$$

where $f(u) = u^s$, $s > 1$ and $a \le x \le b$. If $u_0(x)$ is sufficiently large, positive and has a single non-degenerate maximum, then there is a blow-up time $T < \infty$ and a unique blow-up point x^* such that

$$u(x^*,t) \longrightarrow \infty \qquad \text{as} \quad t \longrightarrow T, \tag{1.8}$$

and

$$u(x,t) \longrightarrow u(x,T) \quad \text{if} \quad x \neq x^*.$$

Close to x^*, the solution $u(x,t)$ develops an isolated peak which becomes narrower, tending to zero width, as $t \longrightarrow T$ and T depends on the initial conditions.

In particular, we will focus on the problem

$$u_t = u_{xx} + u^3, \tag{1.9}$$

whose numerical solution we will discuss and analyze.

This equation has solutions which blow up in a finite time T and which develop increasingly small length scales. It has been studied extensively in the literature and is an important test problem for adaptive methods Budd and Williams (2010). It also has certain scaling laws and approximately self-simlar solutions. These features will allow a direct comparison of the errors obtained by the 'method of lines in space' approach considered in this paper, and other adaptive means of solving this and related problems. We shall show that the mesh points obey scaling laws consistent with those of the original equation. We will derive careful estimates for the mesh scaling and, by substituting these into (1.6) will show that estimates of the form (1.5) can be achieved with the constant C truly independent of the solution scale.

The layout of the rest of this paper is as follows. In Section 2 we review some of the theory related to the blow-up behaviour and scaling laws of the equation (1.7). In Section 3 we describe the adaptive numerical method we employ to solve this problem using an implicit discretisation in time and a collocation method in space. In Section 4 we derive some error estimates for this adaptive method. In Section 5 we report on some numerical experiments which confirm these error estimates. Finally in Section 6 we present some conclusions.

2 Blow-Up Problems

The study of blow-up phenomena is a major research focus in the study of nonlinear evolution equations. A class of problems that may display this feature consists of semilinear parabolic equations. These arise for instance in models of combustion in chemicals, chemotaxis in cellular aggregates, or the formation of shocks in the inviscid Burger equation Budd et al. (1996). Singularities can represent a change in the properties of the model such as ignition of a heated gas mixture. Blow-up commonly occurs when the initial data is sufficiently large and has a single nondegenerate maximum Budd et al. (1996).

2.1 The structure of blow-up solutions to PDEs

2.1.1 Overview

Let x^* denote the single blow-up point according to (1.8). To compute an accurate solution, an adaptive numerical method must be used that evolves the spatial mesh as the singularity develops. The singularity develops in a fairly simple manner, often independently of local structures in the initial conditions. It is conjectured in Kitzhofer et al. (2010) that the growth of $u(x,t)$ near the blow-up time T is described by

$$\max_x |u(x,t)| \propto (T-t)^{-\gamma}, \quad \gamma > 0. \tag{2.1}$$

We will analyze (1.7) with $f(u) = u^s$ for $s = 3$. This problem is well suited for testing numerical methods as the formation of the singularity is typical for a wide range of PDEs. Also, a lot is known about the underlying analytical structure of the solution for t close to T and x close to x^*. Thus it is well suited for testing the properties of our adaptive numerical methods. If the numerical method faithfully follows the underlying asymptotic structure, we can assume that it does the same for more complicated problems where the underlying structure is unknown.

2.1.2 Scaling

Let us consider the scaling properties of (1.7). Let

$$t = \lambda t', \qquad x = \lambda^\theta x', \qquad u = \lambda^\beta u'. \tag{2.2}$$

Substituting into the left-hand side of (1.7)

$$u_t = \frac{\partial u}{\partial t} = \frac{\partial(\lambda^\beta u')}{\partial(\lambda t')} = \lambda^{\beta-1}\frac{\partial u'}{\partial t'}, \tag{2.3}$$

and the right-hand side

$$u_{xx} = \frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) = \frac{\partial}{\partial x}\left(\lambda^{\beta-\theta}\frac{\partial u'}{\partial x'}\right) = \lambda^{\beta-2\theta}\frac{\partial^2 u'}{\partial x'^2}, \tag{2.4}$$

and

$$u^s = \lambda^{\beta s} u'^s. \tag{2.5}$$

Changing variables we obtain

$$\lambda^{\beta-1} u'_{t'} = \lambda^{\beta-2\theta} u'_{x'x'} + \lambda^{\beta s} u'^s.$$

For natural scaling invariance of the PDE we require

$$\beta - 1 = \beta - 2\theta = \beta s,$$

which implies $\theta = \frac{1}{2}$ and $\beta = \frac{1}{1-s} < 0$. We then set

$$\gamma = -\beta = \frac{1}{s-1} > 0.$$

A natural scaling of the solutions of (1.7) is then given by

$$\begin{aligned} &(T-t) \longrightarrow \lambda(T-t), \\ &x \longrightarrow \lambda^{\frac{1}{2}} x, \\ &u \longrightarrow \lambda^{-\gamma} u, \qquad \lambda > 0. \end{aligned} \tag{2.6}$$

A self-similar solution of (1.7) is any solution of the form

$$u = \frac{1}{(T-t)^{\gamma}} f\left(\frac{x}{(T-t)^{\frac{1}{2}}}\right), \tag{2.7}$$

which is invariant under this scaling. Such scaling invariance and corresponding self-similar solutions can be found in various equations describing blow-up.

2.1.3 Asymptotic Theory

In this section, we recapitulate the known asymptotic theory for the behavior of the solution of (1.7). This equation has been very well-studied in the mathematical literature and the solutions are known to exhibit self-similar blow-up behaviour. In particular, if we define $\gamma = \frac{1}{s-1}$, then the solution has a self-similar profile which blows up according to

$$u(x,t) \sim (T-t)^{-\gamma} \tag{2.8}$$

asymptotically as $t \longrightarrow T$ Bebernes and Bricher (1992). This information was used in Budd et al. (1996) as part of a scaling argument to show that the MMPDE corresponding to the blow-up problem (1.7) is also scale-invariant if the monitor function is chosen to be $M = u^{s-1}$. They then presented some numerical results which showed that the MMPDE method is capable of reproducing the self-similar solution profiles in a more accurate and stable manner than is possible with other more common choices of monitor function such as arclength. A more precise description of the blow-up profile of the solution is given in the following theorem (see Bebernes and Bricher (1992) and references therein):

Theorem 1 *Let* $\gamma = \frac{1}{s-1}$. *If the initial value is sufficiently large, then the solution to the initial-boundary value problem* (1.7) *satisfies*

$$\lim_{t \longrightarrow T} (T-t)^{\gamma} u(x^* + \mu[(T-t)(\alpha - \log(T-t))]^{\frac{1}{2}}, t) = \gamma^{\gamma} \left[1 + \frac{\mu^2}{4s\gamma} \right]^{-\gamma} \tag{2.9}$$

uniformly for all $|\mu| \leq C$ *for a given constant* $C > 0$, *where* α *is a constant depending only on the initial value* $u(x,0)$.

The theorem shows that the blow-up profile can best be represented in the so-called kernel coordinate $\mu = \mu(x,t)$, which is fixed as $t \longrightarrow T$ and defined as

$$\mu = (x - x^*)[(T-t)(\alpha - \log(T-t))]^{-\frac{1}{2}}. \tag{2.10}$$

We note that this relationship identifies a natural spatial length scale of approximately

$$L \sim (T-t)^{1/2} \tag{2.11}$$

close to the singularity, and that this should be reflected in the numerical method.

Close to the blow-up time T and blow-up point x^*, the solution u of (1.7) satisfies

$$u(x,t) \sim \frac{1}{\left((T-t)+x^2\right)^{\frac{1}{s-1}}} = \frac{1}{(T-t)^{\frac{1}{s-1}}} \frac{1}{\left(1+\frac{x^2}{T-t}\right)^{\frac{1}{s-1}}}. \tag{2.12}$$

In particular, for $f(u) = u^3$

$$u(x,t) \sim \frac{1}{\sqrt{(T-t)+x^2}} = \frac{1}{\sqrt{(T-t)}} \frac{1}{\sqrt{1+(\frac{x}{\sqrt{T-t}})^2}}. \tag{2.13}$$

3 The adaptive numerical method

3.1 The adaptive BVP solver

The mesh selection strategy implemented in `bvpsuite` was proposed and investigated in Pulverer et al. (2011). Most modern mesh generation techniques in two-point boundary value problems construct a smooth function mapping a uniform auxiliary grid to the desired nonuniform grid. In Pulverer et al. (2011) a new system of control algorithms for constructing a grid density function $\phi(x)$ is described. The local mesh width $h_i = x_{i+1} - x_i$ is computed as $h_i = \varepsilon_N / \varphi_{i+1/2}$, where $\varepsilon_N = 1/N$ is the accuracy control parameter corresponding to $N-1$ interior points, and the positive sequence $\Phi = \{\varphi_{i+1/2}\}_{i=0}^{N-1}$ is a discrete approximation to the continuous density function $\phi(x)$, representing the mesh width variation. Using an error estimate, a feedback control law generates a new density from the previous one. Digital filters may be employed to process the error estimate as well as the density Söderlind (2003).

For boundary value problems, an adaptive algorithm must determine a sequence of mesh densities $\Phi^{[\nu]}$, $\nu = 0, 1, \ldots$ in terms of problem or solution properties. True adaptive approaches equidistribute some *monitor function*, a measure of the residual or error estimate, over the interval, such that (1.5) holds. As $\Phi^{[\nu]}$ will depend on the error estimates, which in turn depend on the distribution of the grid points, the process of finding the density becomes *iterative*. For some error control criteria a local grid change typically has global effects. The techniques developed here avoid this difficulty by restricting the error estimators to those having the property that the estimated error on the interval $[x_i, x_{i+1}]$ only depends on the local mesh width, $h_i = \varepsilon_N / \varphi_{i+1/2}$.

In order to be able to generate the mesh density function, we decided to use the residual $r(x)$ to define the monitor function. The values of $r(x)$ are available from the substitution of the collocation solution $p(x)$ into the analytical problem (1.4). We first compute

$$R_k(x_{i+1/2}) = \int_{x_i}^{x_{i+1}} r_k(x)\,dx \approx \frac{r_k(x_i) + r_k(x_{i+1})}{2}(x_{i+1} - x_i)$$

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

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

to obtain the monitor function necessary for the update of $\Phi^{[\nu]}$. While the residual based monitor function $R(x) := \hat{R}(x_{i+1/2})$, $x \in J_i$, is used to update the mesh density, the number of the necessary mesh points in the final grid is determined from the requirement that the absolute global error satisfies the tolerance. The mesh halving routine provides the values of the error estimate $\mathscr{E}_k(x)$, $k = 1, \ldots, d$, in the entire interval $[a, b]$, so we can compute

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

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 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 points for the next iteration step is predicted from

$$N^{[\nu+1]} = M \left(\frac{G_{\Delta^m}}{0.9\,TOL} \right)^{1/(m+1)}, \tag{3.1}$$

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

Below, we specify in more detail the grid adaptation routine implemented in the code.

1. Grid generation, finding the optimal density function, is separated from mesh refinement, finding the proper number of mesh points. We first try to provide a good density function $\Phi^{[0]}$ 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 (3.1).
3. The calculation of the density function is terminated when $N^{[\nu+1]} > 0.9 N_\nu$. Clearly, it can be expected that in the course of the optimization of the density function the number of the associated mesh points will monotonically decrease. This process is stopped when the next density profile $\Phi^{[\nu+1]}$ would result in saving less than 10% of the mesh points compared to the current density profile $\Phi^{[\nu]}$.

4. Since the calculation of a residual is reasonably cheap we always update the density profile to make use of the information provided by the most recent available numerical solution associated with the function $\Phi^{[\nu]}$.
5. We finally solve the problem on the mesh based on $\Phi^{[\nu+1]}$ with $N^{[\nu+1]}$ mesh points, and estimate the global error of this approximation. If the accuracy requirement is satisfied, we stop the calculations, otherwise we refine again.

For more details and the results of numerical tests, we refer the reader to Pulverer et al. (2011).

3.2 Adaptive time-stepping

The step size in time is defined or estimated at the first timestep as Δt_0 and it is updated at each timestep through a step size in time control strategy. Also, the step size in time should be updated if the approximate solution of the problem at a timestep is not accurate enough. In this case, the step size should be reduced.

We are interested in applying a strategy which automatically adjusts the step size in order to achieve a prescribed error tolerance. We use a classical error estimate based on mesh-halving. In this approach, the accuracy of the solution at each timestep is calculated via computation of two approximations to the solution. The first approximation is obtained by solving the ODE system with full step size Δt and the second one is obtained by solving the ODE system twice with halved step size $\Delta t/2$. We 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. Using these two approximations, we estimate the error for u_{full} by

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

where $q = 1$ is the order of convergence of the Euler method. To proceed to the next timestep, a mixed tolerance is defined using the tolerances for the absolute and relative error,

$$\text{mixed TOL} := aTOL + rTOL \max(\|u_{\text{full}}\|_\infty, \|u_{\text{old}}\|_\infty),$$

where $aTOL$ and $rTOL$ are the absolute and relative tolerances in time, respectively. Moreover, u_{old} is the approximate solution from the previous timestep. We require that the error estimate satisfies for each x in the spatial grid

$$|\mathscr{E}(x)| \leq \text{mixed TOL}. \tag{3.2}$$

We define a tolerance factor by

$$\text{TOL fac} := \left\| \frac{\text{error}}{\text{mixed TOL}} \right\|_2. \tag{3.3}$$

The tolerance factor is required to be less than 1 in order to accept the predicted step size. 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 step size is obtained by

$$\Delta t_{\text{opt}} = 0.9\Delta t \left(\frac{1}{\text{TOL fac}}\right)^{\frac{1}{q+1}}, \tag{3.4}$$

where $q = \min(q_{\text{full}}, q_{\text{half}})$. In our case, q_{full} and q_{half} are equal to the order of Euler's method in calculating the full and half steps, respectively, i.e. $q = 1$.

If TOL fac ≤ 1, the computed step is accepted and the solution is advanced with u_{full} and a new step is tried with Δt_{opt} as step size. Otherwise, the step is rejected and the computations are repeated with the new step size Δt_{opt}.

4 Error equidistribution

First, we look at the time propagation of the semi-discrete problem. For this purpose, we neglect the spatial errors in the approximations of the solution u and its spatial derivatives, as these are reliably controlled by the adaptive BVP solver to the desired level of accuracy, see below. The principal local error term of the backward Euler method with step size Δt_n Quarteroni and Saleri (2006) is

$$\mathscr{E}(x,t_n,\Delta t_n) = \frac{(\Delta t_n)^2}{2} u_{tt}(x,t_n). \tag{4.1}$$

Differentiation of (1.7) w. r. t. t yields

$$\begin{aligned} u_{tt} &= u_{xxt} + f'(u)u_t \\ &= u_{xxxx} + (f(u))_{xx} + f'(u)(u_{xx} + f(u)) \\ &= u_{xxxx} + (f'(u)u_x)_x + f'(u)(u_{xx} + f(u)) \\ &= u_{xxxx} + 2f'(u)u_{xx} + f''(u)u_x^2 + f'(u)f(u). \end{aligned}$$

We consider the special case of $f(u) = u^3$. This gives

$$\mathscr{E}(x,t_n,\Delta t_n) = \frac{(\Delta t_n)^2}{2} \Big(u_{xxxx}(x,t_n) + 6u^2(x,t_n)u_{xx}(x,t_n) + 6u(x,t_n)u_x^2(x,t_n) + 3u^5(x,t_n) \Big). \tag{4.2}$$

Since close to the blow-up time T, we have $u(x,t) \sim (T-t)^{-\frac{1}{2}}$, see (2.8), we can conclude $(T-t) \sim u(x,t)^{-2}$. So, we have

$$u^5 = u \times u^4, \quad = u \times (u^{-2})^{-2}, \quad \sim (T-t)^{-\frac{1}{2}}(T-t)^{-2}, \quad = (T-t)^{-\frac{5}{2}}.$$

We now estimate the size of the error $\mathscr{E}$. To do this we use the natural solution scale, say U and the length scale L from (2.11). We then observe the following scalings:

$$u_{xxxx} \sim U/L^4, \quad u^2 u_{xx} \sim U^3/L^2, \quad uu_x^2 \sim U^3/L^2, \quad u^5 \sim U^5.$$

Now, it follows from Section 2 that U scales as $1/(T-t)^{1/2}$ and L scales as $(T-t)^{1/2}$, thus all of the components involving u and its derivatives in (4.2) scale as $(T-t)^{-5/2}$. Consequently,

$$\begin{aligned} \max_{x\in[0,1]} \mathscr{E}(x,t_n,t_{n+1}-t_n) &\sim \frac{1}{2}(T-t_{n+1})^{-5/2}(t_{n+1}-t_n)^2 \\ &\sim \frac{1}{2}u(0,t_{n+1})(T-t_{n+1})^{-2}(t_{n+1}-t_n)^2. \end{aligned} \tag{4.3}$$

Thus, to control the relative error of the time integration close to the blow-up time, the step size should satisfy

$$\Delta t_{n+1} := t_{n+1} - t_n \sim (T - t_{n+1}) \sim \frac{1}{u(0,t_{n+1})^2}. \tag{4.4}$$

In our numerical experiments reported in Section 5, this relationship is indeed observed, see Figure 5.2.

Next, we analyze the spatial mesh width that we can predict from asymptotic theory and compare this to our numerical experiments reported in Section 5. For each time t_n, the corresponding minimal spatial mesh width $\Delta x^{(n)}$ should be close to the natural length scale given by L in (2.11), and hence we obtain

$$\Delta x^{(n)} \sim \frac{1}{\max_{x\in[0,1]}|u(x,t_n)|} \sim \sqrt{T-t_n} \sim \frac{1}{u(0,t_n)}. \tag{4.5}$$

Indeed this result follows from the collocation error estimate (1.6) . In this estimate the derivative $u^{(m+2)}$ scales as U/L^{m+2}. It follows from (1.6) that the relative collocation error is given by

$$(u(x,t_n)-u^n(x_n))/U = C((x_{n+1}-x_n)/L)^{m+2},$$

where C is independent of U. Thus we get an equidistributed solution error which is independent of u and x provided that $\Delta x \sim L$, which is precisely (4.5).

The time evolution of the spatial grid shown in Figure 5.2 confirms this proposition.

5 Numerical Results

Let us consider (1.9), where we slightly change the boundary conditions to accommodate for the known symmetry of the problem, from

$$u(-4,t) = u(4,t) = 0,$$

to

$$u_x(0,t) = u(4,t) = 0, \tag{5.1}$$

and initial condition

$$u(x,0) = u_0(x) = 4\exp(-x^2), \quad x \in [0,4]. \tag{5.2}$$

Discretization by the transverse method of lines approach in conjunction with the backward Euler scheme ((1.2) with $\theta = 1$), yields in each timestep the ODE boundary value problem for $z(x) := u_{i+1}(x)$, as follows

$$\Delta t\, z''(x) + \Delta t\, z^3(x) - z(x) + u_i(x) = 0, \tag{5.3}$$

with initial condition

$$u_0(x) = 4\exp(-x^2), \tag{5.4}$$

and boundary conditions

$$z'(0) = 0, \quad z(4) = 0. \tag{5.5}$$

In Figure 5.1, the solution process can be seen. We have limited our calculations to $t_{end} = 0.0325129$ as the desired end point in time. Tolerances have been set as follows: $aTOL = 10^{-4}$, absolute tolerance in time, and $rTOL = 10^{-6}$, relative tolerance in space. The maximum solution value at t_{end} is approximately $7.45\cdot 10^5$. The calculations took 105 steps.

In Figure 5.2, we show the time evolution of the quantities $\max_x |u^n(x)^2| \times \Delta t_n$ and $\max_x |u^n(x)| \times \min \Delta x^{(n)}$ which clearly reflect the results from the error analysis: We consider the evolution of Δx throughout the course of the calculation. Recall that this is controlled by the error control strategy based on equidistributing the estimate (1.6). In Section 4 we predicted that $\min \Delta x^{(i)} \sim u = (T-t)^{1/2}$. As $\max_x |u(x,t)| = (T-t)^{-1/2}$, we expect to observe an upper bound on $\max_x |u^n(x)| \min(\Delta x^{(n)} = O(1)$. In Figure 5.2 (bottom), we observe exactly the predicted behaviour. Note the jumps in the graph, which are caused by changes in the number of mesh points used at that stage in the calculation, as shown in Figure 5.3 (right). Further information can be found in Figure 5.3 (left).

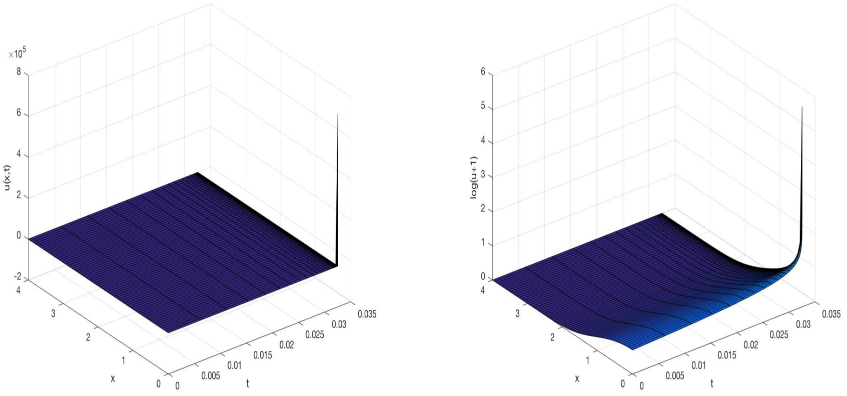

Fig. 5.1 The evolution in time for the solution of the problem (1.9) subject to boundary conditions (5.2) and the starting profile $u_0(x) = 4\exp(-x^2)$. At $t_{end} = 0.0325129$ the solution maximum is $u_{max} = 7.45 \cdot 10^5$. The left plot shows the solution u; in order to better resolve the shape of the surface particularly at the boundary while avoiding negative values of the logarithm, we plot $\log(u+1)$ in the right picture.

6 Conclusions

We have considered the space and time adaptive numerical solution of a nonlinear heat equation by Rothe's method based on the backward Euler scheme in conjunction with a polynomial collocation method implemented in the Matlab code `bvpsuite`. We inferred the asymptotics of optimal grids providing an efficient solution through error equidistribution from known scaling laws for the analytic problem, and verified that our adaptive approach closely approaches these optimal grids relying solely on asymptotically correct error estimates.

References

U. Ascher, S. Ruuth, and T.R. Wetton. Implicit–explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.*, 32(3):797–823, 1995.

J. Bebernes and S. Bricher. Final time blowup profiles for semilinear parabolic equations via center manifold theory. *SIAM J. Math. Anal.*, 23(4):852–869, 1992.

C. J. Budd and J. Williams. How to adaptively resolve evolutionary singularities in differential equations with symmetry ? *J. Eng. Math.*, 66:217–236, 2010.

C. J. Budd, W. Huang, and R. D. Russell. Moving mesh methods for problems with blow-up. *SIAM J. Sci. Comput.*, 17:305–327, 1996.

W. Huang and R. D. Russell *Adaptive Moving Mesh Methods*. Springer Verlag, New York, 2010.

G. Kitzhofer, O. Koch, G. Pulverer, C. Simon, and E. Weinmüller. The new MATLAB code BVPSUITE for the solution of singular implicit boundary value problems. *JNAIAM J. Numer. Anal. Indust. Appl. Math.*, 5:113–134, 2010.

G. Pulverer, G. Söderlind, and E. Weinmüller. Automatic grid control in adaptive bvp solvers. *Numer. Algorithms*, 56(1):61–92, 2011.

A. Quarteroni and F. Saleri. *Approximation of functions and data*. Springer, 2006.

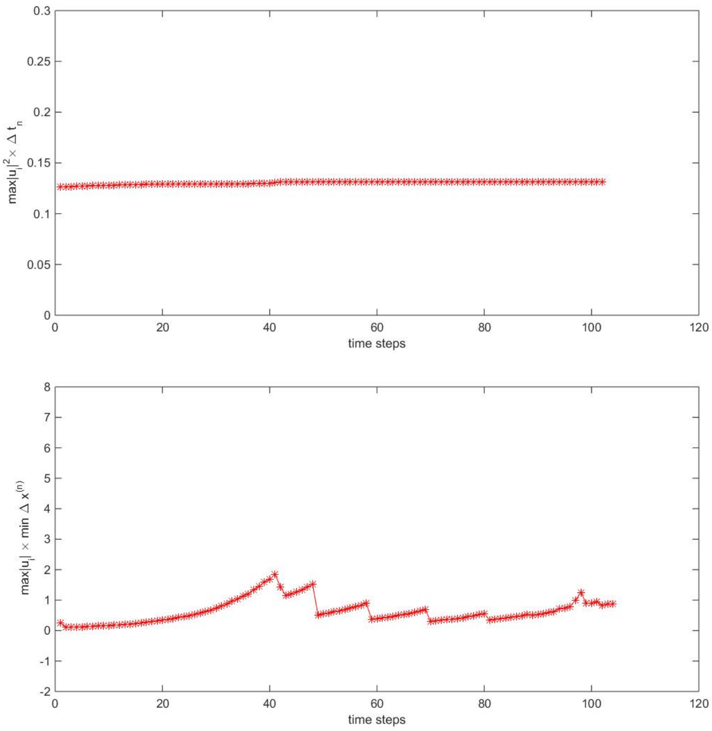

Fig. 5.2 The top figure shows the evolution in time for $\max|u_i|^2 \times \Delta t_n$ for the problem (1.9)–(5.2) with starting profile $u_0(x) = 4\exp(-x^2)$ and the bottom figure shows the evolution in time for $\max|u_i| \times \min \Delta x^{(n)}$ in this problem. We have considered $aTOL = 10^{-4}$, $rTOL = 10^{-6}$ and $t_{end} = 0.0325129$ as the desired end point and 105 steps are necessary to reach it.

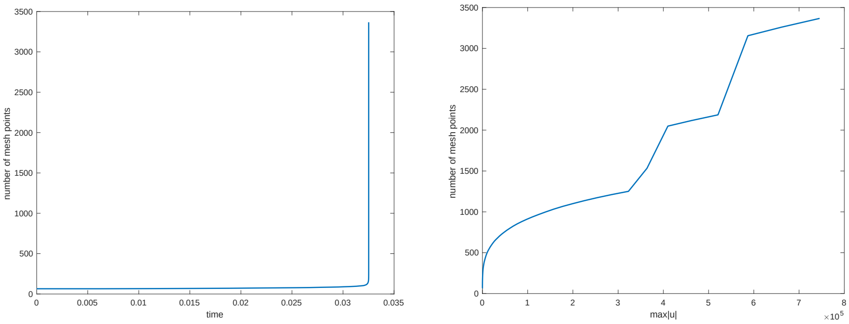

Fig. 5.3 The left figure shows the number of meshpoints in the space grid as a function of time until $t_{end} = 0.0325129$. The right figure displays the number of meshpoints in the space grid as a function of solution's maximal value, $\max|u| = \max_{0\leq k\leq N}|u_k|$.

G. Söderlind. Digital filters in adaptive time-stepping. *ACM Trans. Math. Software*, 29: 1–26, 2003.

T.K. Ushijima. On the approximation of blow-up time for solutions of nonlinear parabolic equations. *Publ. RIMS, Kyoto Univ.*, 36(5):613–640, 2000.