From Nonlinear PDEs to Singular ODEs ⋆

Chris Budd

University of Bath, Department of Mathematical Sciences, Bath, BA 27 AY, UK

Othmar Koch ∗ Ewa Weinmüller

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

Abstract

We discuss a new approach for the numerical computation of self-similar blow-up solutions of certain nonlinear partial differential equations. These solutions become unbounded in finite time at a single point at which there is a growing and increasingly narrow peak. Our main focus is on a quasilinear parabolic problem in one space dimension, but our approach can also be applied to other problems featuring blow-up solutions. For the model we consider here, the problem of the computation of the self-similar solution profile reduces to a nonlinear, second-order ordinary differential equation on an unbounded domain, which is given in implicit form. We demonstrate that a transformation of the independent variable to the interval $[0, 1]$ yields a singular problem which facilitates a stable numerical solution. To this end, we implemented a collocation code which is designed especially for implicit second order problems. This approach is compared with the numerical solution by standard methods from the literature and by well-established numerical solvers for ODEs. It turns out that the new solution method compares favorably with previous approaches in its stability and efficiency. Finally, we comment on the applicability of our method to other classes of nonlinear PDEs with blow-up solutions.

Key words: Quasilinear partial differential equations, self-similarity, blow-up solutions, singular ordinary differential equations, collocation methods

⋆ This project was supported in part by the Special Research Program SFB F011 “AURORA” of the Austrian Science Fund FWF.

∗ Corresponding author.
Email addresses: cjb@maths.bath.ac.uk (Chris Budd), othmar@othmar-koch.org (Othmar Koch), e.weinmueller@tuwien.ac.at (Ewa Weinmüller).

1 Self-Similar Blow-Up in Quasilinear Parabolic Equations

In this paper we discuss the numerical computation of blow-up solutions of the quasi-linear partial differential equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(u^\sigma \frac{\partial u}{\partial x}\right) + u^\beta, \tag{1}$$

which is a model for the temperature profile of a fusion reactor plasma with one source term [15]. We prescribe the boundary conditions

$$u(-L,t) = u(L,t) = 0,\ t > 0,\ u(x,0) = u_0(x),\ x \in [-L, L],$$

and the parameters are chosen such that $\sigma > 0,\ \beta \geq \sigma + 1$. To study blow-up behavior, the initial function $u_0(x)$ is assumed to be sufficiently large. The numerical approximation of solutions to (1) is also discussed in [3]. Note that the equation (1) is degenerated at the boundary if $\sigma > 0$. This does not affect the calculation of blow-up however, as the singularity formation is well away from the boundary.

Remark 1 *Using similar ideas, we could also treat the problem*

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\left|\frac{\partial u}{\partial x}\right|^\sigma \frac{\partial u}{\partial x}\right) + \mathrm{e}^u, \tag{2}$$

which occurs in turbulent diffusion or the flow of a non-Newtonian fluid. Note that for $\sigma = 0$, (2) *is the Frank-Kamenetskii equation* [9]. *In* [4], *qualitative results for* (2) *are derived. The problem does not lend itself to numerical computations readily, however. Instead, a regularization*

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\left(\left(\frac{\partial u}{\partial x}\right)^2 + \varepsilon^2\right)^{\sigma/2} \frac{\partial u}{\partial x}\right) + \mathrm{e}^u,$$

where $\varepsilon = 10^{-6}$, *could be used. This will be the subject of further investigation. Note that if necessary, equation* (1) *can also be regularized in a similar fashion by replacing* u^σ *by* $(u^2 + \varepsilon^2)^{\sigma/2}$.

The differential equation (1) is invariant under the Lie group of transformations

$$t \to \lambda t,\ x \to \lambda^m x,\ u \to \lambda^{-1/(\beta-1)} u,$$

where $m = 1/2 - \sigma/(2\beta - 2) > 0$.

The self-similar solutions which are invariant under these transformations take the form

$$u(x,t) = \frac{z(\tau)}{(T-t)^{1/(\beta-1)}} \quad \text{with } \tau = \frac{|x-x_*|}{(T-t)^m}, \tag{3}$$

where $x_* \in (-L, L)$ denotes the point where blow-up occurs.

Substitution of this ansatz into the differential equation (1) yields the ordinary differential equation

$$(z^\sigma(\tau) z'(\tau))' - m\tau z'(\tau) - \frac{1}{\beta-1} z(\tau) + z^\beta(\tau) = 0, \quad \tau > 0. \tag{4}$$

Symmetry yields the boundary condition

$$z'(0) = 0. \tag{5}$$

Moreover, a matching condition corresponding to the boundary conditions for the original PDE is prescribed for the self-similar solutions:

$$\lim_{\tau\to\infty} z(\tau)\tau^{1/m(\beta-1)} = C, \tag{6}$$

where C is a suitable constant. From the assumptions on the parameters m and β used in our discussion here, it follows that

$$\lim_{\tau\to\infty} z(\tau) = 0. \tag{7}$$

Moreover, to observe blow-up we require $z(\tau) > 0$.

2 Transformation to a Finite Domain

The solution approach we propose for (4) is transformation to a singular boundary value problem on the interval $[0, 1]$, where the numerical solution method is subsequently applied directly to the resulting second order problem in implicit form.

To transform (4) to a finite domain, we split the interval $(0, \infty) = (0, 1] \cup [1, \infty)$, and use the transformation $\tau \to 1/\tau$ on the second, unbounded interval. This yields a boundary value problem with an essential singularity for $(z_1(s), z_2(s)) := (z(s), z(1/s)),\ s \in (0, 1]$, where the new variable s is the same as τ for z_1 and $s = 1/\tau$ for z_2. Together with the appropriate matching conditions at $\tau = 1$, the resulting problem reads

$$z_1''(s)z_1(s)^\sigma + z_1(s)^{\sigma-1}\sigma(z_1'(s))^2 - msz_1'(s) - \frac{1}{\beta-1}z_1(s) + z_1^\beta(s) = 0, \quad (8)$$

$$\left(2sz_2'(s) + s^2 z_2''(s)\right) z_2(s)^\sigma + s^2 z_2(s)^{\sigma-1}\sigma(z_2'(s))^2 + \\ + m\frac{z_2'(s)}{s} - \frac{1}{\beta-1}\frac{z_2(s)}{s^2} + \frac{z_2^\beta(s)}{s^2} = 0, \quad (9)$$

$$z_1'(0) = 0,\ z_2(0) = 0,\ z_1(1) = z_2(1),\ z_1'(1) = -z_2'(1). \quad (10)$$

We found this to be the most suitable formulation for the numerical treatment, see §4 and [10].

3 Numerical Solution

To solve the transformed problem (8)–(10) numerically, we implemented a solver based on polynomial collocation for second order ordinary differential equations (see for example [2]). In contrast to standard collocation implementations (cf. [1]), this solver was designed especially to be applicable to problems posed in *implicit form*. We will demonstrate in §4 that this may be an advantage for the numerical treatment of (8)–(10). Our code was implemented in MATLAB 6.5 R13 on a PC, and the computations reported in this paper were performed in IEEE double precision arithmetic with relative machine accuracy $\approx 1.11 \cdot 10^{-16}$.

The numerical solution of (8)–(10) turned out to be rather contrived, since the solution of the nonlinear system of algebraic equations for the coefficients of the collocation solution requires a reasonable starting guess. We found that a good initial approximation is given by choosing $z_1 \equiv \alpha,\ z_2(t) = \alpha t$, where $\alpha \approx z_1(0) = z(0)$. A useful heuristic for the choice of α is given by the following reasoning (cf. [10]): Ignoring boundary conditions, a constant solution of (4) is given by

$$z_{\text{const}} \equiv \mathrm{e}^{\frac{\ln(\beta-1)}{1-\beta}}\,.$$

Thus, $\alpha = z(0)$ must satisfy

$$z''(0)\alpha^\sigma - \frac{1}{\beta-1}\alpha + \alpha^\beta = 0$$

due to (5). Since we observed that $z''(0)$ usually is small and negative, this means that a choice of α slightly larger than z_{const} gives a useful starting approximation. This corresponds with the values for $z(0)$ given in [3] for various choices of $\sigma,\ \beta$, but can also be successfully applied in other situations.

Note that the choice of α is indeed critical, a value too close to z_{const} leads to a failure of the solution routine, and the collocation equations cannot be solved successfully if α is chosen too large either. Also, we found constant starting profiles for $z_1(t)$, $z_2(t)$ to yield no solution of the nonlinear collocation equations. If we choose for example $\sigma = 0.1$, $\beta = 2$, the value for $z_1(0)$ computed using our code is $z_1(0) = 1.02656202260728$ (the value given in [3] is $z_1(0) = 1.0265620225916$). The corresponding solution can be computed starting with $\alpha = 1.04$. However, for the slightly perturbed values $\alpha = 1.015$ or $\alpha = 1.06$ our solution procedure was not successful, see [10]. In Table 1 we give a list of values for α which were successful for various values of β, where throughout we chose $\sigma = 0.1$. For comparison, we also give the values of z_{const} used to estimate α and the values $z_1(0)$ successfully computed by our code.

β	z_{const}	α	$z_1(0)$
1.7	1.664518071	1.76	1.73978649986601
1.8	1.321714079	1.34	1.37085731540303
1.9	1.124195018	1.145	1.15924643609817
2.0	1	1.04	1.02656202260728
2.5	0.7631428284	0.765	0.77381693907163
3.0	0.7071067810	0.709	0.71323890650409

Table 1
Values of z_{const}, α and $z_1(0)$ for various choices of β, $\sigma = 0.1$.

With the suitable starting guess, we found that our collocation code works dependably, but is subject to an order reduction. In Table 2, we give the empirical convergence orders for the solution of (8)–(10) by collocation based on four equidistant collocation points. The convergence order was computed from the approximations for three consecutive step sizes, cf. for example [7]. Table 2 gives, for every equidistant step size h, the value diff=$\|\xi_{i+1} - \xi_i\|$ of the maximal absolute difference between successive numerical approximations and the estimated convergence order, separately for the two components z_1, z_2. We find that, while z_1 shows the classical convergence order four, the second component is affected by an order reduction down to about two. The reason for this behavior is not quite clear, but may be caused by an unsmoothness of the solution of (4) near infinity.

In Figure 1, we give a plot of the two components of the solution of (8)–(9) computed for $h = 2^{-6}$, and Figure 2 shows this solution transformed back to the interval $[0, 70]$.

Finally, in Figure 3 the evolution of the peak of the solution u of the original PDE (1) computed according to the ansatz (3) is shown. Here for the purpose of the graphical representation we have set $x_* = 0$, $T = 20$.

h	diff z_1	ord z_1	diff z_2	ord z_2
2^{-1}	1.834426e−01	2.51	1.995535e−01	0.38
2^{-2}	3.202632e−02	2.79	1.528185e−01	1.43
2^{-3}	4.600655e−03	8.54	5.650699e−02	7.62
2^{-4}	1.229689e−05	4.61	2.856124e−04	2.00
2^{-5}	5.032704e−07	3.98	7.140073e−05	2.36
2^{-6}	3.167926e−08	3.99	1.387161e−05	2.15
2^{-7}	1.991901e−09	3.99	3.121607e−06	2.05
2^{-8}	1.246463e−10	3.99	7.528824e−07	1.99
2^{-9}	7.801537e−12	3.91	1.889616e−07	1.96
2^{-10}	5.160317e−13	—	4.849716e−08	—

Table 2
Convergence orders for (8)–(10), four equidistant collocation points.

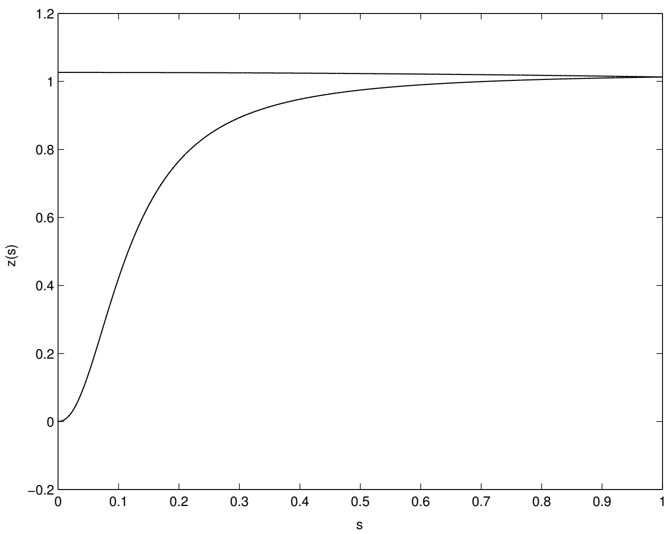

Fig. 1. Solution of (8)–(10).

4 Comparisons

Here, we compare our method for the computation of the numerical solution of (4) with different approaches either reported in the literature or adapted from obvious ideas. In [3], a shooting type approach is used to approximate the self-similar solution profile: Instead of using the asymptotic boundary condition (7), an initial value problem starting at $z(0) = \alpha,\ z'(0) = 0$ is solved. This method is numerically highly unstable. If the value of α is not chosen with high precision, the solution satisfying the asymptotic boundary condition (7) is not approximated by the shooting method. But even if α

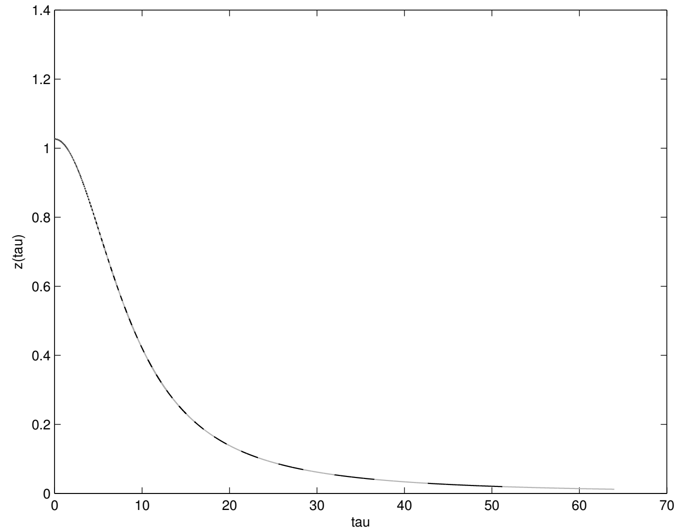

Fig. 2. Solution of (8)–(10) transformed back to $[0, 70]$.

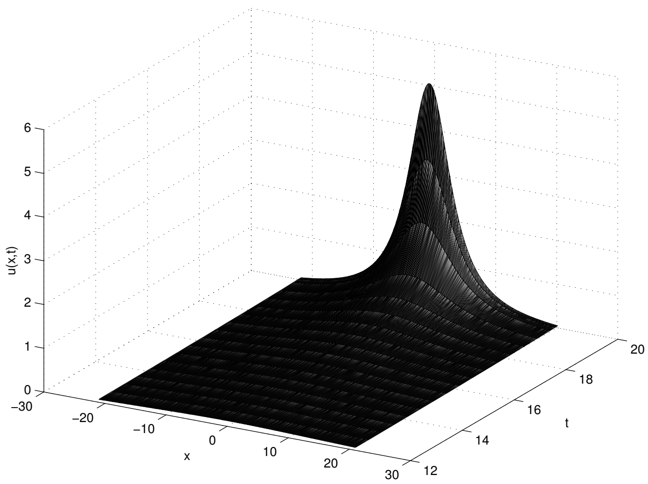

Fig. 3. Blow-up solution of (1).

is chosen carefully, the errors introduced during the numerical calculation imply that eventually the correct solution is no longer followed. Rather, undesired solution modes lead to unusable results or even failure of the numerical method. Two examples demonstrating this unfavorable behavior are given in Figure 4. Both curves were computed by the shooting approach using $z_1(0) = z(0) = 1.02656202260728$, $z_2(0) = z'(0) = 0$. The lower, black curve was computed by MAPLE using a Runge-Kutta-Fehlberg pair of orders four and five, see for example [12]. Up to $t \approx 8$, the desired, decaying solution is approximated quite well, cf. Figure 4. However, beyond this point the numerical solution starts to grow exponentially. Numerical inaccuracy causes it to drift away from the bounded solution and instability results. The second curve in Figure 4 shows a different behavior which we encountered when the implicit solver DDASSL, see [11], was used: After showing the desired solution behav-

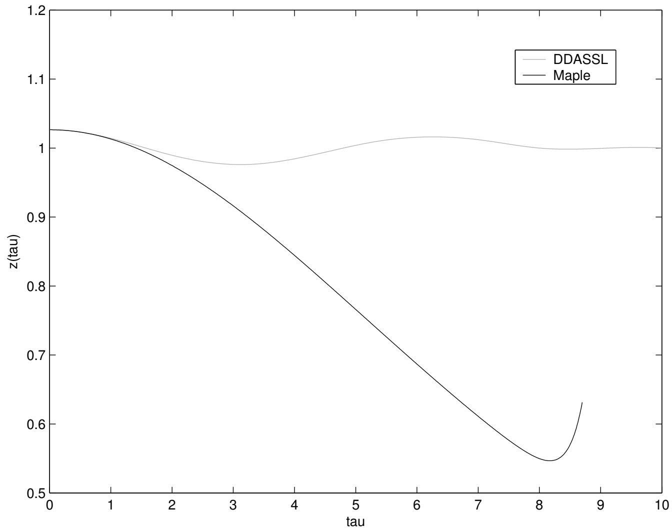

Fig. 4. Shooting approach by Maple and DDASSL.

ior up to $t \approx 1$, the numerical solution resolves the trivial, constant solution $z \equiv 1 = z_{\text{const}}$. Both results demonstrate that the use of shooting to solve an equivalent initial value problem instead of the boundary value problem with the correct asymptotic boundary condition leads to an unstable solution behavior. This is also obvious from the results in [3], where the shooting approach using DDASSL to solve the initial value problem yields the correct solution up to $t \approx 12$, whence exponential growth sets in.

The shooting approach seems to be the only reasonable choice for the numerical approximation of the original problem (4). The obvious alternative is to replace the problem (4) on the infinite interval by the same problem posed on a truncated interval $[0, T],\ T \gg 1$, with the condition (7) replaced by $z(T) = 0$. Unfortunately, this technique does not seem to work in a stable way either: Figure 5 shows the numerical approximation computed by our boundary value problem solver, where $T = 64$ and the mesh consists of only five subintervals. We can see that the collocation solution shows rapid oscillations, whence a numerical solution at a finer mesh is impossible altogether. The same behavior was observed when using the standard solvers `bvp4c`, see [13], or COLNEW, cf. [1], for the solution of the boundary value problem on a truncated interval. Only in the cases where $z(T)$ is chosen as the value of the "exact" solution (computed numerically as in §3) this approach is successful, but not practicable of course. Numerical experiments supporting our claims are given in [10].

Finally, we treated our transformed problem (8)–(10) by standard collocation solvers. First, we applied the Matlab code `bvp4c` available with Matlab 6.5 R13. To this end, the problem has to be transformed to first order and explicit form. `bvp4c` uses collocation with an evaluation at the left, singular

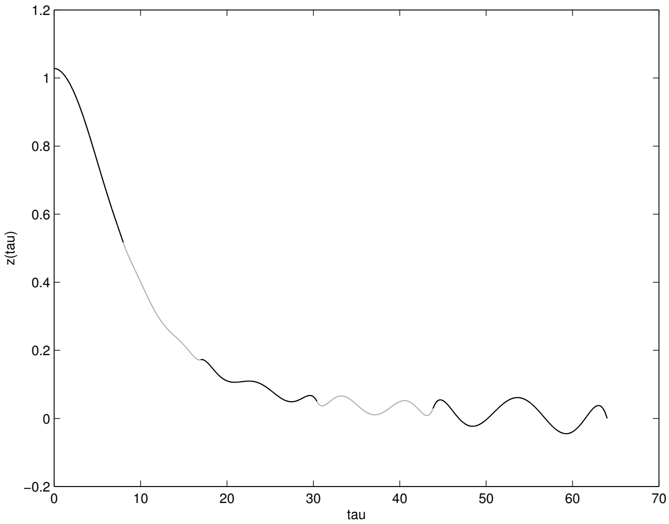

Fig. 5. Solution on truncated interval using $z(T) = 0$.

endpoint. To avoid this evaluation at $t = 0$ we chose an interval $[\varepsilon, 1]$ with $\varepsilon = 0.0078125$, instead of $[0, 1]$, and shifted the boundary conditions to this interval, $z_1'(\varepsilon) = z_2(\varepsilon) = 0$. Still, the numerical solution turned out to be highly unstable. Even for starting profiles very close to the exact solution the numerical solution was oscillating wildly for small t, see the results given in Figure 6. We stress that throughout, `bvp4c` could not reach the default tolerances (absolute tolerance 10^{-6}, relative tolerance 10^{-3}), the displayed result is the last approximation returned by the code. Note that choosing the boundary conditions $z_1'(\varepsilon)$ and $z_2(\varepsilon)$ as the values of the numerical solution determined in §3 also fails. Still, the failure of `bvp4c` should not be overestimated, since this code is not intended for the solution of problems of the present type. We have included these computations in order to provide the best possible overview of standard solution approaches, however.

Remark: Using a special reformulation of our problem based on the known asymptotics (6) of the solution at infinity, it is possible however to use the features of `bvp4c` for parameter dependent problems to approximate the solution of (4)–(6): In [14], the interval is truncated at both ends, and thus the computations are carried out for $t \in [\varepsilon, b]$, where $\varepsilon = 10^{-3}$, $b = 70$ (different choices of the parameter ε may yield a higher accuracy of the numerical results). Then, a transformation to a first order system is achieved via $y_1(\tau) = z(\tau),\ y_2(\tau) = z^{\sigma}(\tau)z'(\tau)$. The boundary conditions at $t = \varepsilon$ and $t = b$ are now posed depending on two unknown parameters, where the asymptotics (6) serve to derive a set of four boundary conditions for the unknowns in this new problem. The features of `bvp4c` for the solution of problems with unknown parameters can now be utilized to compute the numerical approximation of the solution of the problem.

We also solved (8)–(10) brought to explicit form by the Fortran 90 code COL-

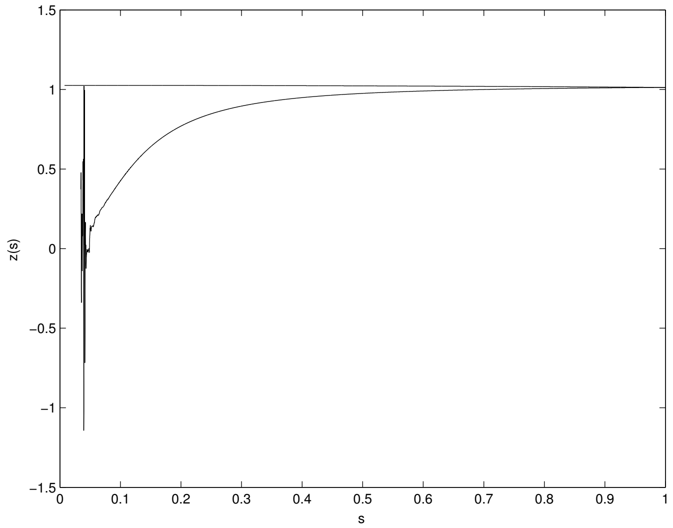

Fig. 6. Solution of (8)–(10) using `bvp4c`.

NEW, cf. [1]. In spite of the fact that in this form a singular term $1/z^\sigma$ is present in the right-hand side, this solution approach provided virtually identical results as our solver, see [10].

5 Conclusions and Outlook

In this paper we have described a new approach for the solution of the implicit, second order ordinary differential equations on semi-infinite intervals associated with the computation of self-similar solution profiles of quasilinear parabolic PDEs. To compute the solution numerically, the problem is transformed to a finite domain and the resulting boundary value problem with an essential singularity is solved by collocation. To this end, a Matlab solver was implemented which can treat directly second order problems posed in implicit form. If the starting approximations are chosen with some care, the collocation methods work dependably and robustly. Our approach is compared with other standard solution methods which mostly turn out to show an unfavorable behavior as compared to our technique.

The same idea of transformation to a finite interval and solution of the resulting singular boundary value problem can be applied in a variety of problems where self-similar solution profiles of nonlinear PDEs are computed via ODE problems. In [7], we have for instance considered the nonlinear Schrödinger equation. In this case, however, it is unfavorable to use the second order form, see [10]. Rather, a transformation to a first order problem and a subsequent transformation to a finite interval makes it possible to translate the asymptotic boundary conditions at $t = \infty$ correctly into boundary conditions for the associated singular problem. Its solution can be successfully computed by collocation, see [6]. Other problems like semi-linear, higher order parabolic equations (cf. [5]) or the complex Ginzburg-Landau equation (cf. [8]) are in principle accessible to our solution method, see [7], but the numerical difficulties arising still require further investigations.

Acknowledgements

We would like to thank Georg Kitzhofer for the implementation of the MATLAB collocation code for the solution of implicit second-order ODEs, and for providing the numerical results reported in this paper.

References

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

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

[3] C. J. Budd, G. J. Collins, V. A. Galaktionov, An asymptotic and numerical description of self-similar blow-up in quasilinear parabolic equations, J. Comput. Appl. Math. 97 (1998) 51–80.

[4] C. J. Budd, V. A. Galaktionov, Stability and spectra of blow-up in problems with quasi-linear gradient diffusivity, Proc. R. Soc. Lond. A 45A (1998) 2371–2407.

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

[6] C. J. Budd, O. Koch, E. Weinmüller, From the nonlinear Schrödinger equation to singular BVPs, AURORA TR-2004-02, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria, available at `http://www.vcpc.univie.ac.at/aurora/publications/` (2004).

[7] C. J. Budd, O. Koch, E. Weinmüller, Self-similar blow-up in nonlinear PDEs, AURORA TR-2004-07, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria, available at `http://www.vcpc.univie.ac.at/aurora/publications/` (2004).

[8] C. J. Budd, V. Rottschäfer, J. F. Williams, Multi-bump, blow-up, self-similar solutions of the complex Ginzburg-Landau equation, submitted to Nonlinearity.

[9] D. A. Frank-Kamenetskii, Towards temperature distributions in a reaction vessel and the stationary theory of thermal explosion, Doklady Akad. Nauk SSSR 18 (1938) 411–412.

[10] G. Kitzhofer, O. Koch, E. Weinmüller, Kollokationsverfahren für singuläre Randwertprobleme zweiter Ordnung in impliziter Form, Techn. Rep. ANUM Preprint Nr. 9/04, in preparation.

[11] L. Petzold, A description of DASSL: A differential algebraic system solver, Report SAND82-8637, Amer. Math. Soc., Sandia National Labs (1982).

[12] L. Shampine, R. Corless, Initial value problems for odes in problem solving environments, J. Comp. Appl. Maths. 125 (2000) 31–40.

[13] L. Shampine, J. Kierzenka, A BVP solver based on residual control and the MATLAB PSE, ACM T. Math. Software 27 (2001) 299–315.

[14] L. Shampine, Solving a hard BVP with `bvp4c`, private communication.

[15] N. V. Zmitrenko, S. P. Kurdyumov, A. P. Mikhailov, A. A. Samarskii, Localization of thermonuclear combustion in a plasma with electronic thermal conductivity, JETF Lett. 26 (1977) 469–472.