

Numerical Blow-up of Semilinear Parabolic PDEs on Unbounded Domains in \mathbb{R}^2

Jiwei Zhang · Houde Han · Hermann Brunner

Received: 3 February 2010 / Revised: 11 January 2011 / Accepted: 24 January 2011 /
Published online: 4 February 2011
© Springer Science+Business Media, LLC 2011

Abstract We study the numerical solution of semilinear parabolic PDEs on unbounded spatial domains Ω in \mathbb{R}^2 whose solutions blow up in finite time. Of particular interest are the cases where $\Omega = \mathbb{R}^2$ or Ω is a sectorial domain in \mathbb{R}^2 . We derive the nonlinear absorbing boundary conditions for corresponding, suitably chosen computational domains and then employ a simple adaptive time-stepping scheme to compute the solution of the resulting system of semilinear ODEs. The theoretical results are illustrated by a broad range of numerical examples.

Keywords Semilinear PDEs · Unbounded spatial domains · Sectorial domains · Finite-time blow-up · Local nonlinear boundary conditions · Finite difference spatial discretization · Adaptive time-stepping

J. Zhang

Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA
e-mail: jzhang@cims.nyu.edu

H. Han

Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
e-mail: hhan@math.tsinghua.edu.cn

H. Brunner (✉)

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR, P.R. China
e-mail: hbrunner@math.hkbu.edu.hk

H. Brunner

Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL A1C 5S7, Canada
e-mail: hermann@math.mun.ca

1 Introduction

We consider the semilinear parabolic problem

$$u_t = \Delta u + u^p, \quad \text{in } \Omega \times (0, T], \quad (1.1)$$

$$u|_{t=0} = u_0(r, \theta), \quad (1.2)$$

$$u \rightarrow 0 \quad \text{when } r \rightarrow +\infty, \quad (1.3)$$

where Δ denotes the two-dimensional spatial Laplace operator and $p > 1$. The initial function u_0 is assumed to be compactly supported in some bounded domain. The unbounded spatial domains Ω we will consider in this paper are:

1. $\Omega = \mathbb{R}^2$; and
2. $\Omega = \{(r, \theta) : 0 < \theta < \nu\pi, 0 \leq r < \infty\} \subset \mathbb{R}^2$ with $0 < \nu < 2$: for this sectorial domain Ω we need to specify two boundary conditions on the boundaries $\Gamma_0 = \{(r, \theta) | \theta = 0 \text{ or } \theta = \nu\pi, 0 \leq r < \infty\}$,

$$u|_{\theta=0} = g_0(r, t) \quad \text{and} \quad u|_{\theta=\nu\pi} = g_\nu(r, t),$$

where we assume that the given functions $g_0(r, t)$, $g_\nu(r, t)$, $u_0(r, \theta)$ are such that

$$g_0(r, 0) = u_0(r, 0), \quad g_\nu(r, 0) = u_0(r, \nu\pi) \quad (0 \leq r \leq +\infty)$$

holds.

The theory of finite-time blow-up in semilinear parabolic PDEs on unbounded domains is well understood; see Fujita's classical paper [11], and compare the survey papers by Levine [20], Bandle and Brunner [3] and Deng and Levine [9], as well as the 1995 monograph [23]. In sharp contrast, many problems arising in the computational treatment of such blow-up problems remain to be addressed [4, 7, 8, 19]. This is particularly true when the underlying spatial domain is unbounded.

There exist numerous papers (see for example [2, 5, 6, 8, 19, 21, 22, 24–26]) that are devoted to the computational solution of blow-up problems (1.1) on *bounded spatial domains* Ω in \mathbb{R}^1 and \mathbb{R}^2 .

In the computational treatment of finite-time blow-up problems on *unbounded domains* Ω , the given PDE needs to be restricted to an appropriate bounded computational domain containing the domain of physical interest, and this requires the construction of appropriate artificial boundary conditions ([13, 15]; compare also [16] for a description of the general framework). The first paper to deal with computational blow-up when $\Omega = \mathbb{R}^d$ ($d = 1, 2$) is, to the best of our knowledge, the one by Brunner, Wu and Zhang [7]; here, *rectangular* computational domains are used.

In the present paper we describe the construction of these artificial boundary conditions first again for $\Omega = \mathbb{R}^2$, using (arguably more natural) *circular* computational domains, and then extend this construction to the more challenging case where Ω is a *sectorial* domain in \mathbb{R}^2 .

The circular artificial boundary Γ_R will be chosen as

$$\Gamma_R := \{(r, \theta) : r = R, 0 \leq \theta \leq \nu\pi\}. \quad (1.4)$$

It divides the original domains Ω into two parts, respectively:

1. If $\Omega = \mathbb{R}^2$, then we define

$$\Omega_i := \{(r, \theta) : 0 \leq \theta < 2\pi, \ 0 \leq r < R\}, \quad (1.5)$$

$$\Omega_e := \{(r, \theta) : 0 \leq \theta < 2\pi, \ R < r < \infty\}. \quad (1.6)$$

2. If Ω is the sectorial domain, we set

$$\Omega_i := \{(r, \theta) : 0 < \theta < \nu\pi, \ 0 \leq r < R\}, \quad (1.7)$$

$$\Omega_e := \{(r, \theta) : 0 < \theta < \nu\pi, \ R < r < \infty\}. \quad (1.8)$$

In order to restrict the given problem to the domain Ω_i , appropriate boundary conditions have to be constructed. Ideally, they should not only be easy to implement, but also make the solution of the new problem coincide with that of the model problem in Ω_i ; in other words, they should imitate the perfect absorption of waves (or heat flows) traveling out of the computational domain through the artificial boundary. A comprehensive description of the artificial boundary methods in numerical partial differential equations can be found in the recent monograph [16] by Han and Wu.

For nonlinear problems it is in general difficult to design such ideal absorbing boundary conditions (ABCs). Using a linearizing approach or an inverse scattering method, the papers [12, 14, 17, 28, 30, 34, 35] present the exact artificial boundary conditions for certain nonlinear PDEs. To construct efficient nonlinear absorbing boundary condition, a novel unified approach is proposed in the recent papers by Zhang, Xu and Wu [32, 33]; it can also be applied to problems like the nonlinear Schrödinger equation, Burgers' equation and semilinear parabolic problems.

The basic idea underlying the unified approach is based on the time-splitting method. Thus, we first consider the splitting of linear problem on the exterior domain Ω_e , to obtain the one-way approximate operator (to make wave outgoing) for the linear differential operator. Then we unite the approximate operator with the nonlinear operator. This paper aims to construct the circular nonlinear ABCs by extending the unified approach.

The outline of the paper is as follows. In Sect. 2 we review the general principle of the novel unified approach mentioned above, and use this approach to construct the local absorbing boundary conditions when $\Omega = \mathbb{R}^2$ and Ω_i given by (1.5). The case where Ω is a sectorial domain in \mathbb{R}^2 (and Ω_i is as in (1.7)) is considered in Sect. 3. In Sect. 4 we discuss the problem of how to select the key parameters, and we then present a range of numerical examples to illustrate various aspects of the resulting numerical schemes in Sect. 5. A brief discussion of future work concludes the paper.

2 Construction of Nonlinear Absorbing Boundary Conditions: $\Omega = \mathbb{R}^2$

In this section we describe the derivation of the local absorbing boundary conditions (LABCs) for the semilinear parabolic (1.1), using the novel unified approach introduced in the papers [18, 32, 33].

2.1 Unified Approach: General Principle

Underlying this unified approach to deriving LABCs is the well-known time-splitting method (or split-step method) of [29, 31] for various types of nonlinear PDEs. Consider (1.1) in its generic operator form,

$$u_t = \mathcal{L}u + \mathcal{N}u, \quad (2.1)$$

where \mathcal{L} denotes a linear (spatial) partial differential operator and \mathcal{N} stands for a nonlinear operator. In analogy to the widely used Strang splitting [27] on a small time interval $[t, t + \tau]$ with $\tau > 0$,

$$u(x, t + \tau) \approx e^{\mathcal{L}\tau/2} e^{\mathcal{N}\tau} e^{\mathcal{L}\tau/2} u(x, t),$$

we use the operator splitting in the form

$$u(x, t + \tau) \approx e^{[\mathcal{L} + \mathcal{N}]\tau} u(x, t)$$

(compare also [7]). We first construct an approximation operator $\mathcal{L}^{(n)}$ for the linear operator \mathcal{L} by means of a one-directional equation (to make the flow outgoing), and then combine the approximation operator $\mathcal{L}^{(n)}$ with the nonlinear term \mathcal{N} , restricted to the artificial boundaries, to obtain the equation

$$u_t = \mathcal{L}^{(n)} u + \mathcal{N} u \quad (2.2)$$

as the nonlinear absorbing boundary conditions. The key problem obviously consists in how to obtain a good approximation operator $\mathcal{L}^{(n)}$. In the following Sect. 2.2, we focus on the derivation of a suitable one-way approximation operator $\mathcal{L}^{(n)}$.

2.2 The Linear Diffusion Equation in \mathbb{R}^2

Setting

$$\mathcal{L}_1 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r}, \quad \mathcal{L}_2 = \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2},$$

we first consider the linear diffusion equation on the exterior domain Ω_e ,

$$u_t = \mathcal{L}_1 u + \mathcal{L}_2 u, \quad \text{in } \Omega_e \times (0, T], \quad (2.3)$$

$$u|_{\Gamma_R} = u(R, \theta, t) \quad (2.4)$$

$$u|_{t=0} = 0, \quad (2.5)$$

$$u \rightarrow 0, \quad \text{when } r \rightarrow +\infty. \quad (2.6)$$

We also introduce an associated simplified problem,

$$u_t = \mathcal{L}_1 u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r}, \quad R < r < +\infty, \quad (2.7)$$

$$u|_{r=R} = u(R, t), \quad (2.8)$$

$$u|_{t=0} = 0, \quad R < r < +\infty, \quad (2.9)$$

$$u \rightarrow 0 \quad \text{when } r \rightarrow +\infty. \quad (2.10)$$

Recall that the Laplace transformation is given by

$$\hat{u}(r, s) := \int_0^{+\infty} e^{-st} u(r, t) dt, \quad \operatorname{Re}\{s\} > 0. \quad (2.11)$$

Thus, by (2.7), the transformed function $\hat{u}(r, s)$ satisfies

$$s\hat{u} = \frac{\partial^2 \hat{u}}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{u}}{\partial r}, \quad (2.12)$$

$$\hat{u} \rightarrow 0 \quad \text{when } r \rightarrow 0. \quad (2.13)$$

The linear differential equation

$$\frac{\partial^2 \hat{u}}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{u}}{\partial r} - s \hat{u} = 0. \quad (2.14)$$

possesses two linearly independent solutions $K_0(\sqrt{s}r)$ and $I_0(\sqrt{s}r)$, where $K_0(x)$, $I_0(x)$ are the modified Bessel functions of order zero (cf. [1, Chap. 9]). Owing to the condition (2.13) obtain

$$\hat{u}(r, s) = C K_0(\sqrt{s}r). \quad (2.15)$$

Differentiating this solution (2.15) with respect to r and substituting in the above differential equation, we are led to the desired one-directional equation on the boundary Γ_R : it is given by

$$\frac{\partial \hat{u}}{\partial r}(R, s) = \frac{\sqrt{s} K'_0(\sqrt{s}R)}{K_0(\sqrt{s}R)} \hat{u}(R, s) =: w(s) \hat{u}, \quad (2.16)$$

where

$$w(s) = \frac{\sqrt{s} K'_0(\sqrt{s}R)}{K_0(\sqrt{s}R)} = -\frac{\sqrt{s} K_1(\sqrt{s}R)}{K_0(\sqrt{s}R)}.$$

The exact ABCs can now be found by means of the inverse Laplace transformation applied to (2.16).

In the practical numerical implementation, it is expensive to evaluate the inverse Laplace transformation of the function $w(s)$. Instead, we will use an approximation to $w(s)$ given by the simplest Padé approximant,

$$w(s) \approx \frac{\alpha s + \beta}{\gamma s + \delta}, \quad |s - s_0| \leq l. \quad (2.17)$$

For a given value of the parameter s_0 , the coefficients $(\alpha, \beta, \gamma, \delta)$ are uniquely determined.

Upon inserting (2.17) into (2.16), we have

$$(\gamma s + \delta) \frac{\partial \hat{u}}{\partial r} = (\alpha s + \beta) \hat{u}. \quad (2.18)$$

The application of the inverse Laplace transformation leads to

$$\frac{\partial}{\partial t} \left(\gamma \frac{\partial u}{\partial r} - \alpha u \right) = -\delta \frac{\partial u}{\partial r} + \beta u, \quad (2.19)$$

and thus we use (2.19) to obtain the approximation $\mathcal{L}_1^{(3)}$ for the operator \mathcal{L}_1 by setting

$$\mathcal{L}_1 \approx \mathcal{L}_1^{(3)} := \left(\gamma \frac{\partial}{\partial r} - \alpha \right)^{-1} \left(-\delta \frac{\partial}{\partial r} + \beta \right). \quad (2.20)$$

Recalling (2.2) in the procedure of the unified approach, we arrive at the one-way equation

$$u_t = \mathcal{L}_1^{(3)} u + \mathcal{L}_2 u. \quad (2.21)$$

Multiply the operator $(\gamma \frac{\partial}{\partial r} - \alpha)$ to (2.21), we thus obtain the local absorbing boundary condition of the linear diffusion equation on the circular artificial boundary Γ_R :

$$\gamma u_{tr} - \alpha u_t = -\delta u_r + \beta u + \gamma \left[-\frac{2}{R^3} u_{\theta\theta} + \frac{1}{R^2} u_{\theta\theta r} \right] - \alpha \frac{1}{R^2} u_{\theta\theta}. \quad (2.22)$$

By observing the boundary condition (2.22), the problem for the diffusion equation on \mathbb{R}^2 has now been reduced to the problem on the bounded domain $\Omega_i \times (0, T]$; it is given by

$$u_t = \Delta u \quad \text{in } \Omega_i \times (0, T], \quad (2.23)$$

$$\gamma u_{tr} - \alpha u_t = -\delta u_r + \beta u + \gamma \left[-\frac{2}{R^3} u_{\theta\theta} + \frac{1}{R^2} u_{\theta\theta r} \right] - \alpha \frac{1}{R^2} u_{\theta\theta} \quad \text{on } \Gamma_R, \quad (2.24)$$

$$u|_{t=0} = u_0(x) \quad \text{in } \Omega_i. \quad (2.25)$$

2.3 Semilinear Parabolic Equations in \mathbb{R}^2

Now we consider the model equation on the exterior domain Ω_e :

$$u_t = \mathcal{L}_1 u + \mathcal{L}_2 u + \mathcal{N} u \quad \text{in } \Omega_e \times (0, T], \quad (2.26)$$

$$u|_{\Gamma_R} = u(R, \theta, t), \quad (2.27)$$

$$u|_{t=0} = 0, \quad (2.28)$$

$$u \rightarrow 0 \quad \text{when } r \rightarrow +\infty, \quad (2.29)$$

with

$$\mathcal{L}_1 := \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r}, \quad \mathcal{L}_2 := \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}, \quad \mathcal{N} = u^{p-1}.$$

Employing the unified approach of Sect. 2.1 and using again the approximation (2.20), the approximating one-way equations become

$$u_t = \mathcal{L}_1^{(3)} u + \mathcal{L}_2 u + \mathcal{N} u. \quad (2.30)$$

By a simple rearrangement, we then construct the nonlinear local artificial boundary condition on the circular artificial boundary Γ_R : it is seen to be

$$\begin{aligned} \gamma u_{tr} - \alpha u_t = & -\delta u_r + \beta u + \gamma \left(-\frac{2}{R^3} u_{\theta\theta} + \frac{1}{R^2} u_{\theta\theta r} + p u^{p-1} u_r \right) \\ & - \alpha \left(\frac{1}{R^2} u_{\theta\theta} + u^p \right). \end{aligned} \quad (2.31)$$

By the nonlinear local artificial boundary condition (2.31), the problem (1.1)–(1.3) is reduced to the following problem on the bounded domain $\Omega_i \times (0, T]$:

$$u_t = \Delta u + u^p, \quad \text{in } \Omega_i \times (0, T], \quad (2.32)$$

$$\begin{aligned} \gamma u_{tr} - \alpha u_t = & -\delta u_r + \beta u + \gamma \left(-\frac{2}{R^3} u_{\theta\theta} + \frac{1}{R^2} u_{\theta\theta r} + p u^{p-1} u_r \right) \\ & - \alpha \left(\frac{1}{R^2} u_{\theta\theta} + u^p \right) \quad \text{on } \Gamma_R, \end{aligned} \quad (2.33)$$

$$u|_{t=0} = u_0 \quad \text{in } \Omega_i. \quad (2.34)$$

3 Construction of Nonlinear Absorbing Boundary Conditions: Sectorial Domain

3.1 The Linear Diffusion Equation on Sectorial Domains

Proceeding along the lines of Sect. 2.2, it is readily verified that the linear diffusion equation in the bounded sectorial (computational) domain (1.7) is given by

$$u_t = \Delta u \quad \text{in } \Omega_i \times (0, T], \quad (3.1)$$

$$\begin{aligned} \gamma u_{tr} - \alpha u_t = -\delta u_r + \beta u + \gamma \left(-\frac{2}{R^3} u_{\theta\theta} + \frac{1}{R^2} u_{\theta\theta r} \right) \\ - \alpha \frac{1}{R^2} u_{\theta\theta} \quad \text{on } \Gamma_R, \end{aligned} \quad (3.2)$$

$$u|_{\theta=0} = g_0(r, t), \quad u|_{\theta=v\pi} = g_v(r, t), \quad (3.3)$$

$$u|_{t=0} = u_0 \quad \text{in } \Omega_i. \quad (3.4)$$

3.2 The Semilinear Parabolic Equation on Sectorial Domains

The analogous result of (2.32)–(2.33), that is, the reduced problem for the semilinear parabolic equation on the bounded sectorial (computational) domain (1.7) is found to be

$$u_t = \Delta u + u^p \quad \text{in } \Omega_i \times (0, T], \quad (3.5)$$

$$\begin{aligned} \gamma u_{tr} - \alpha u_t = -\delta u_r + \beta u + \gamma \left(-\frac{2}{R^3} u_{\theta\theta} - \frac{1}{R^2} u_{\theta\theta r} + p u^{p-1} u_r \right) \\ - \alpha \left(\frac{1}{R^2} u_{\theta\theta} + u^p \right) \quad \text{on } \Gamma_R, \end{aligned} \quad (3.6)$$

$$u|_{\theta=0} = g_0(r, t), \quad u|_{\theta=v\pi} = g_v(r, t), \quad (3.7)$$

$$u|_{t=0} = u_0 \quad \text{in } \Omega_i. \quad (3.8)$$

We end this section by adding a brief remark on the stability of the obtained nonlinear LABCs at the artificial boundaries. According to the construction of the nonlinear LABCs, the resulting solution makes the energy arising in the computational domain out-going. Thus, the corresponding exterior solution will have no influence on the interior solution. We therefore may conclude that the nonlinear LABCs are stable. The numerical experiments presented in Sect. 5 confirm this observation regarding the stability of the boundary conditions.

4 Computation of the Parameter s_0

In view of practical applications it is natural to ask how, for given initial data u_0 , does one find a suitable value of the parameter s_0 appearing in the LABCs (cf. Sect. 2.2)? In the paper [31], the choice of the parameter s_0 is related to the frequency of the wave impinging on the artificial boundaries. We remind the reader of two strategies described in the papers [10, 31]. The first strategy is to take the dominant Fourier mode by using the Fourier series expression

in terms of the physical variable in space. The second strategy consists in employing the Gabor transform near the artificial boundaries. The Gabor transform is given by

$$\tilde{u}(k, t) := \int_0^R W(x) u(x, t) e^{-ikx} dx = \int_{R-b}^R u(x, t) e^{-ikx} dx, \quad (4.1)$$

where the window function is

$$W(x) := \begin{cases} 1, & x \in [R-b, R], \\ 0, & \text{otherwise.} \end{cases}$$

Thus we select the wave number as

$$k_0 = \frac{\int_0^\infty |\tilde{u}(k, t)|^m k dk}{\int_0^\infty |\tilde{u}(k, t)|^m dk}, \quad (4.2)$$

with m denoting some positive real number. For the wave equation, we take $s_0 = k_0^2$. For the heat equation, the relation $s_0 = f(k_0)$ between the wave number and the expansion point s_0 is still unknown.

In the following computations we use, for convenience, a fixed value s_0 ; the problem of choosing the parameter s_0 in a more systematic way is currently being studied. This value of the parameter s_0 will then be taken as the expansion point in (2.17) when approximating the function $w(s)$ by the $[1, 1]$ -Padé approximant. Resorting to the function `PadéApproximant` in the toolbox of Mathematica, we show in Table 1 the values of the parameters $\alpha, \beta, \gamma, \delta$ in (2.24) corresponding to the given s_0 and R .

Recall that

$$w(s) = \frac{\sqrt{s} K'_0(\sqrt{s} R)}{K_0(\sqrt{s} R)} \sim -\sqrt{s} - \frac{1}{2R} + \mathcal{O}(s^{-\frac{1}{2}}), \quad s \rightarrow \infty.$$

If instead of $w(s)$ we approximate the function $-\sqrt{s}$, we obtain

$$w(s) \approx -\frac{3s_0^{\frac{1}{2}} s + s_0^{\frac{3}{2}}}{s + 3s_0} - \frac{1}{2R} + \mathcal{O}(s^{-\frac{1}{2}}) + \mathcal{O}(s - s_0)^3 \quad (4.3)$$

and

$$\alpha = 6R\sqrt{s_0} + 1, \quad \beta = s_0(3 + 2R\sqrt{s_0}), \quad \gamma = 2R, \quad \delta = 6Rs_0,$$

where s_0 is the expansion point.

Table 1 Evaluation of parameters for the given s_0, R

(s_0, R)	α	β	γ	δ
(4, 3)	-0.386322	-0.615363	0.0627889	0.748844
(10, 3)	-0.241595	-0.908999	0.0250528	0.749472
(4, 4)	-0.383364	-0.588051	0.0626775	0.749290
(10, 4)	-0.240449	-0.88049	0.0250316	0.749684
(4, 5)	-0.381621	-0.571233	0.0626202	0.749519
(10, 5)	-0.239772	-0.863063	0.025021	0.749790

We briefly comment on the computation of the parameters involved in the Padé approximation (in particular in (4.3)).

1. Once the expansion point s_0 has been chosen, the values of the corresponding parameters $\alpha, \beta, \gamma, \delta$ are determined by approximating the function $w(s)$ by the $[1, 1]$ -Padé expansion at the point s_0 .
2. Since the function $w(s)$ includes the special functions K_0 and K_1 , we can turn to computer software (e.g. Mathematica) to expand $w(s)$, by using the function `PadéApproximant` for the $[1, 1]$ -Padé expansion.
3. In the present paper, we have carried out the all calculations by fixing the expansion points $s_0 = 4, 10, \dots$.
4. However, the authors are aware that the expansion point s_0 is related to the frequency of the wave (or heat flow) impinging on the artificial boundaries. By always taking a fixed expansion point s_0 , the boundary conditions will not be able to minimize all the different waves (or heat flows) impinging on the boundaries. Thus, ideally, we ought to capture the value of the expansion point s_0 adaptively. According to the papers mentioned at the beginning of this section, the wave number k_0 can be obtained by using, e.g., the Gabor transform. For the wave equation we know that the relation between the wave number k_0 and the frequency s_0 is approximately $s_0 = k_0^2$. For the heat equation considered in this paper, the analogous relation is unknown. Hence we are not (yet) able to use the adaptive method to compute feasible values of the parameters.
5. From the point of view of developing a code based on the adaptive method, the computational cost will be high if we employ the function `PadéApproximant` in Mathematica each time (recall the asymptotic expansion for $w(s)$). For smaller s , the asymptotic expansion will have an error order of $\mathcal{O}(s^{-1/2})$. In general, when the expansion point s_0 (frequency) is greater than 1, we use (4.3) to approximate $w(s)$. The last term in (4.3) reflects the error order caused by the $[1, 1]$ -Padé approximation.

5 Numerical Discretizations and Examples

5.1 Numerical Discretizations

For the reduced problems (2.32)–(2.33), we not only give the discretizations of the model equation and the nonlinear ABCs, but we also propose a novel approach to deal with the moving non-essential singularity at the point $(0, 0)$. For the reduced problems (3.5)–(3.8), the equations and ABCs are discretized in a similar way.

We cover the truncated computational domain $[0, R] \times [0, 2\pi]$ by a rectangular grid, parallel to the axes of the polar coordinate system. Let $\Delta r := R/I$ and $\Delta\theta := 2\pi/J$ be the spatial mesh sizes. Then the corresponding grid points are given by $r_i = i\Delta r$ and $\theta_j = j\Delta\theta$.

Denote the approximation of $u(r_i, \theta_j, t_m)$ by $u_{i,j}^m$ and set $h := \min\{\Delta r, \Delta\theta\}$. The time steps $\tau_m := t_{m+1} - t_m$ are chosen such that the mapping from $\mathbb{R}^{I \times J}$ to $\mathbb{R}^{I \times J}$ that defines the approximate solution u^{m+1} is contractive. Let $\rho := \max_{0 \leq i \leq I, 0 \leq j \leq J} |u_{i,j}^m|$, and define $L = L(M)$ to be the Lipschitz constant of u^ρ in $[0, M]$. The time-step criterion [3, 4] given by

$$\tau_m < \min \left\{ \frac{(\kappa - 1)\rho}{2(8\kappa\rho/h^2 + (\kappa\rho)^\rho)}, \frac{1}{2(8/h^2 + L(\kappa\rho))} \right\}, \quad (5.1)$$

makes the above mapping from $B(\kappa\rho) := \{v \in \mathbb{R}^{I \times J} : |v| < \kappa\rho, \kappa > 1\}$ to $B(\kappa\rho)$ contractive (at the time level $t = t_m$). Applying the Crank-Nicolson scheme to (2.32) we obtain the

discretization

$$\begin{aligned} \frac{u_{i,j}^{m+1} - u_{i,j}^m}{\tau_m} &= \frac{u_{i+1,j}^{m+\frac{1}{2}} - 2u_{i,j}^{m+\frac{1}{2}} + u_{i-1,j}^{m+\frac{1}{2}}}{\Delta r^2} + \frac{1}{r_i} \frac{u_{i+1,j}^{m+\frac{1}{2}} - u_{i-1,j}^{m+\frac{1}{2}}}{2\Delta r} \\ &\quad + \frac{1}{r_i^2} \frac{u_{i,j+1}^{m+\frac{1}{2}} - 2u_{i,j}^{m+\frac{1}{2}} + u_{i,j-1}^{m+\frac{1}{2}}}{\Delta \theta^2} + (u_{i,j}^{m+\frac{1}{2}})^p, \end{aligned} \quad (5.2)$$

$$u_{i,0}^m = u_{i,J}^m, \quad u_{i,-1}^m = u_{i,J-1}^m, \quad (5.3)$$

with $0 < i < I$, $0 \leq j < J$, and $u_{i,j}^{m+\frac{1}{2}} = \frac{u_{i,j}^m + u_{i,j}^{m+1}}{2}$.

The discretizations of the boundary conditions at the point (r_{I-1}, θ_j) are given by

$$\begin{aligned} &\gamma \frac{u_{I,j}^{m+1} - u_{I,j}^m + u_{I-2,j}^{m+1} - u_{I-2,j}^m}{2\tau_m \Delta r} - \alpha \frac{u_{I-1,j}^{m+1} - u_{I-1,j}^m}{\tau_m} + \delta \frac{u_{I,j}^{m+\frac{1}{2}} - u_{I-2,j}^{m+\frac{1}{2}}}{2\Delta r} \\ &\quad - \beta u_{I-1,j}^{m+\frac{1}{2}} + \frac{2\gamma + \alpha r_{I-1}}{r_{I-1}^3} \frac{u_{I-1,j+1}^{m+\frac{1}{2}} - 2u_{I-1,j}^{m+\frac{1}{2}} + u_{I-1,j-1}^{m+\frac{1}{2}}}{\Delta \theta^2} + \alpha (u_{I-1,j}^{m+\frac{1}{2}})^p \\ &\quad - \frac{\gamma}{r_{I-1}^2} \frac{(u_{I,j+1}^{m+\frac{1}{2}} - 2u_{I,j}^{m+\frac{1}{2}} + u_{I,j-1}^{m+\frac{1}{2}}) - (u_{I-2,j+1}^{m+\frac{1}{2}} - 2u_{I-2,j}^{m+\frac{1}{2}} + u_{I-2,j-1}^{m+\frac{1}{2}})}{2\Delta r \Delta \theta^2} \\ &\quad - p\gamma (u_{I-1,j}^{m+\frac{1}{2}})^{p-1} \frac{u_{I,j}^{m+\frac{1}{2}} - u_{I-2,j}^{m+\frac{1}{2}}}{2\Delta r} = 0. \end{aligned} \quad (5.4)$$

Note that

$$u_{0,j}^m = u_{0,0}^m \quad (j = 1, \dots, J; m = 1, 2, \dots). \quad (5.5)$$

At the origin we cannot discretize (2.32) directly, due to the singularity of the coefficient of the equation at $r = 0$. In order to circumvent this problem, we look at removing the non-essential singularity at the point $(0, 0)$ caused by the coordinate transformation. In the small disc

$$\Omega_{\frac{\Delta r}{2}} := \left\{ (r, \theta) \mid 0 \leq r \leq \frac{\Delta r}{2}, 0 \leq \theta < 2\pi \right\},$$

we may write (1.1), resorting to integration by parts and a simple transformation of variables, as

$$\frac{\Delta r}{2} \int_0^{2\pi} \frac{\partial u}{\partial r} d\theta = \int_{\Omega_{\frac{\Delta r}{2}}} (u_t - u^p) dx dy.$$

Using the composite mid-point rule, we arrive at

$$\frac{\Delta r \Delta \theta}{2} \sum_{j=0}^N \frac{\partial u}{\partial r} \Big|_{(\frac{\Delta r}{2}, \theta_j)} = \frac{\pi \Delta r^2}{4} (u_t(0, 0) - u^p(0, 0)).$$

Furthermore,

$$\frac{\Delta\theta}{2} \sum_{j=0}^J (u(\Delta r, \theta_j) - u(0, 0)) = \frac{\pi \Delta r^2}{4} (u_t(0, 0) - u^p(0, 0)),$$

that is,

$$\frac{2\Delta\theta}{\pi \Delta r^2} \left(\sum_{j=0}^J u(\Delta r, \theta_j) - (J+1)u(0, 0) \right) = (u_t(0, 0) - u^p(0, 0)).$$

Taking $J = 3$ as an illustrative example, we have $\Delta\theta = \pi/2$ and

$$\begin{aligned} & \frac{u(\Delta r, 0) + u(\Delta r, \pi/2) + u(\Delta r, \pi) + u(\Delta r, 3\pi/2) - 4u(0, 0)}{\Delta r^2} \\ &= (u_t(0, 0) - u^p(0, 0)). \end{aligned}$$

Thus, we construct the discretization of (2.32) at the point $(0, 0)$:

$$\frac{2\Delta\theta}{\pi \Delta r^2} \left(\sum_{j=0}^J u_{1,j}^{m+\frac{1}{2}} - (J+1)u_{0,0}^{m+\frac{1}{2}} \right) = \left(\frac{u_{0,0}^{m+1} - u_{0,0}^m}{\tau_m} - (u_{0,0}^{m+\frac{1}{2}})^p \right). \quad (5.6)$$

Hence, the discretizations of problems (2.32)–(2.33) are described by (5.2)–(5.6). In order to deal with the nonlinear difference terms, we use a suitable iteration method at each step, for a given error tolerance $\epsilon := 10^{-11}$.

5.2 Numerical Tests

In this section we illustrate various aspects of the performance of our numerical schemes. In Sect. 5.2.1, we discuss the numerical tests for the domain $\Omega = \mathbb{R}^2$. Test 1 is devoted to investigating the efficiency of the local absorbing boundary conditions, by selecting different parameters s_0 for the linear diffusion equation. At the same time, we also look at the performance of the strategy of removing the non-essential singularity at the point $(0, 0)$. Test 2 illustrates the dependence of the blow-up time on the size R of the computational domain. Test 3 is concerned with numerical solutions for the cases of multi-point blow-up and line blow-up, respectively.

In Sect. 5.2.2, we focus on the numerical tests for the sectorial domains. In Test 4, we see the dependence of the blow-up time on the choice of the parameter ν , and we show the dependence of the numerical blow-up time on the refinement of the spatial mesh.

The following initial functions u_0 are used in our tests:

1. Single Gaussian function:

$$u_0 = \chi(r, \theta) \exp(-r^2); \quad (5.7)$$

2. Double Gaussian function:

$$u_0 = \chi_1(r, \theta) \exp(-2(r-2)^2) + \chi_2(r, \theta) \exp(-2r^2); \quad (5.8)$$

3. Initial function u_0 for the sectorial domain:

$$u_0 = \chi(r, \theta) \sin\left(\frac{\theta}{\nu}\right) \exp(-2(r-1.2)^2); \quad (5.9)$$

where χ, χ_i ($i = 1, 2$) are given functions. In the computations we set these initial functions $u_0 = 0$ when $(r, \theta) \notin \Omega_i$. Let M_i denote the prescribed blow-up thresholds, $\tilde{T}_b^{CN}(M_i)$ is the blow-up time for the above Crank-Nicolson scheme with respect to the thresholds M_i , and $\mu(M_i)$ denotes the number of time steps needed to reach the blow-up threshold M_i (which we choose to be $M_1 = 10^7$ and $M_2 = 10^8$, respectively).

The adaptive time step selection is based on (5.1); hence, we choose the sequence $\{\tau_m\}$ by

$$\tau_m := \min \left\{ \frac{(\kappa - 1)\rho}{2(8\kappa\rho/h^2 + (\kappa\rho)^p)}, \frac{1}{2(8/h^2 + L(\kappa\rho))} \right\}, \quad (5.10)$$

corresponding to the parameters $\kappa = 50$, $p = 2$ and $h = \min\{\Delta r, \Delta\theta\}$.

5.2.1 Numerical Tests for $\Omega = \mathbb{R}^2$

Test 1: This test aims at testing the strategy of removing the non-essential singularity at the point $(0, 0)$, and investigating the efficiency of local absorbing boundary conditions. For the initial function

$$u_0 = \frac{1}{4\pi t_0} \exp\left(-\frac{r^2}{4t_0}\right),$$

the linear diffusion equation possesses the exact solution

$$u_{ex}(r, \theta, t) = \frac{1}{4\pi(t + t_0)} \exp\left(-\frac{r^2}{4(t + t_0)}\right).$$

In the calculation, we chose $R = 3$, $\tau = 1.5625 \times 10^{-3}$ and $t_0 = 0.1$. For different parameters s_0 and refinement meshes, Table 2 lists the relative errors $|u_{nu} - u_{ex}|/|u_{ex}|$ at the point $(r, \theta) = (3, 0)$. Another important way to measure the performance for the chosen s_0 is to look at the (discrete) L_1 -norm of the error, defined by

$$L_1(t) := \frac{1}{I \times J} \sum_{i=0}^I \sum_{j=0}^J |u_{ex}(r_i, \theta_j, t) - u_h(r_i, \theta_j, t)|.$$

Table 3 shows the L_1 -norm for different parameters s_0 and different mesh sizes. Tables 2 and 3 reveal that the proposed absorbing boundary conditions and the strategy of removing the singularity work well. Moreover, the accuracy of the local ABCs is dependent on the choice of the parameter s_0 , which is taken in a large interval (for example $[4, 10]$), such that the relative error is less than 1% (see Table 2).

Test 2: To see the dependence of the blow-up time on the length of the computational domain, we use fixed spatial mesh sizes and the adaptive time steps τ_m of (5.10). The same

Table 2 Relative error on artificial boundary point (3,0)

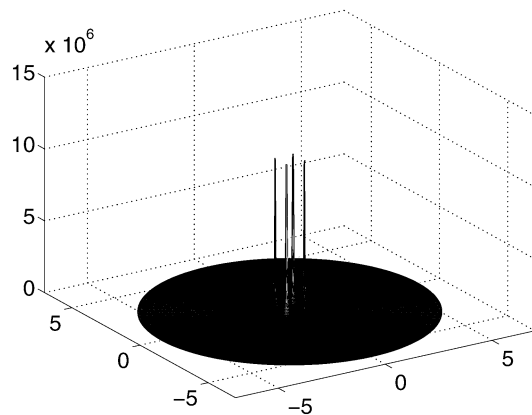
s_0	2	4	6	8	10	12
32×36	0.0248	0.00817	0.0164	0.00987	0.00397	0.0214
64×72	0.0242	0.002999	0.0113	0.00721	0.00327	0.0170
128×144	0.0233	0.00123	0.00938	0.00629	0.00279	0.01498

Table 3 L_1 -norm for different parameters s_0

s_0	2	4	6	8	10	12
32×36	1.3427e-4	1.0691e-4	1.2367e-4	1.2336e-4	1.1452e-4	1.2097e-4
64×72	5.4769e-5	2.8395e-5	3.5861e-5	3.7276e-5	3.1521e-5	3.7418e-5
128×144	3.3995e-5	1.1344e-5	1.3326e-5	1.5287e-5	1.0606e-5	1.5925e-5

Table 4 Single Gaussian function and $s_0 = 10$

	$r = 1.5$	$r = 2$	$r = 2.5$	$r = 3$
$\tilde{T}_b^{CN}(M_1)$	0.02330861728	0.02330861728	0.02330861728	0.02330861728
$\tilde{T}_b^{CN}(M_2)$	0.02330871838	0.02330871838	0.02330871838	0.02330871838

Fig. 1 Numerical solution for multi-point blow-up

time step size strategy will be utilized in the following numerical examples. We use the above single-Gaussian function with $\chi(r, \theta) = 50$ as the initial data.

The calculations are based on the values $s_0 = 10$, $R = 4$, $I = R \times 40$, $J = 60$, $\kappa = 50$, and we chose as the blowup thresholds $M_1 = 10^7$, $M_2 = 10^8$. For the different lengths of the computational domains, we show, in Table 4, a sample of computed blow-up times $\tilde{T}_b^{CN}(M_i)$ ($i = 1, 2$) for single-Gaussian initial data. Table 4 reveals that the numerical blow-up times are insensitive to the length of the computational domain with the given absorbing boundary conditions. It also illustrates the efficiency of the constructed ABCs.

Test 3: This example shows the multi-point numerical blow-up with two different initial functions.

1. Multi-point blow-up:

We use the single Gaussian function (5.7) as the initial data corresponding to the value $\chi(r, \theta) = 50 \sin^2(\pi r) \cos(\theta) \sin(\theta)$. In the calculation we choose $s_0 = 10$, $R = 5$, $I = 100$, $J = 80$, $\kappa = 50$, $M = 10^6$. Figure 1 shows the numerical solution with multi-point blow-up for the given threshold $M = 10^6$.

2. Line blow-up:

For the double Gaussian initial function (5.8), we choose $\chi_1(r, \theta) = \chi_2 = 50$, $s_0 = 10$,

$R = 4$, $I = 200$, $J = 120$, $\kappa = 50$, $M_1 = 10^7$, and we show, in Fig. 2, the numerical solution with line blow-up for the given threshold M_1 .

5.2.2 Numerical Tests for Sectorial Domain

In this part we compute the blow-up times by selecting different parameters ν to investigate the dependence on the size of the computational domain.

Test 4: We use the given initial function in (5.9) with $\chi = 50$, and choose $s_0 = 10$, $\kappa = 50$ for the following calculation. Table 5 gives the numerical blow-up times and the number of time steps for the sector parameters $\nu = 1$, $1/2$, $1/6$, $R = 4$ and fixed spatial mesh sizes $\Delta r = R/160$, $\Delta\theta = \pi/150$.

We display the dependence of the numerical blow-up time and the number of time steps on the spatial mesh refinement for $\nu = 1/6$ in Table 6.

In Table 7, we compare the blow-up times and the number of time steps by choosing different values of the radius R , with $\Delta r = 1/80$, $\Delta\theta = \pi/300$.

Fig. 2 Numerical solution for line blow-up

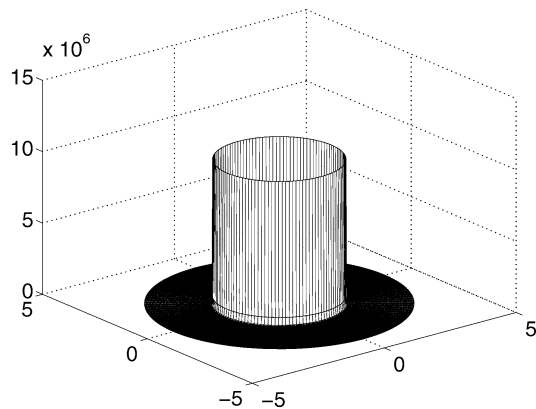


Table 5 Dependence of blow-up time on ν for fixed mesh size

	$\nu = 1$	$\nu = \frac{1}{2}$	$\nu = \frac{1}{6}$
$\tilde{T}_b^{CN}(M_1)$	2.18171879578e-02	2.286195092e-02	3.78867668840e-02
$\tilde{T}_b^{CN}(M_2)$	2.18172779163e-02	2.286204022e-02	3.78868560826e-02
$\mu(M_1)$	462	475	616
$\mu(M_2)$	520	533	674

Table 6 Dependence of blow-up time on refinement with $\nu = 1/6$

$I \times J$	$\tilde{T}_b^{CN}(M_1)$	$\mu(M_1)$	$\tilde{T}_b^{CN}(M_2)$	$\mu(M_2)$
160×25	3.78867668840e-02	616	3.78868560826e-02	674
320×50	3.79439555392e-02	1366	3.79440422849e-02	1423

Table 7 The blow-up time and number of time steps for different radii R

	$R = 2.5$	$R = 3$	$R = 3.5$
$\tilde{T}_b^{CN}(M_1)$	3.79439555848e–02	3.79439555392e–02	3.79439555392e–02
$\tilde{T}_b^{CN}(M_2)$	3.79440423306e–02	3.79440422849e–02	3.79440422849e–02
$\mu(M_1)$	1336	1366	1366
$\mu(M_2)$	1423	1423	1423

6 Concluding Remarks

This paper studies the numerical solution of blow-up problems associated with initial-boundary value problems for semilinear parabolic PDEs where the spatial domain is either \mathbb{R}^2 or a sectorial domain in \mathbb{R}^2 . For suitably chosen natural (bounded) computational domains (different from the ones in [7]) we use a unified approach to construct the nonlinear local absorbing boundary conditions. The time-stepping part of our numerical scheme is based on a simple adaptive implementation of a one-point collocation method. Extensive numerical tests illustrate the accuracy and efficiency of our approach.

Many questions regarding theoretical and numerical aspects of these blow-up problems remain to be answered. Here we mentioned just a few of these:

- For sectorial domains $\Omega = \{(r, \theta) : 0 < \theta < \nu\pi, 0 \leq r < \infty\}$ ($\nu \in (0, 2)$) the analysis of the location of the blow-up point(s) corresponding to a given nonnegative initial function u_0 remains, as far as we know, to be established. The resulting insight will of course play a crucial role when selecting a suitable R for the computational domain (1.7).
- Construction of the nonlinear absorbing boundary conditions for semilinear parabolic problems more general than (1.1), for example for

$$u_t = \Delta u + V(t, \cdot)u + f(u),$$

where $V(t, \cdot)$ represents a potential and $f(u) = u^p$ ($p > 1$) or $f(u) = e^u$?

- Use of moving mesh PDEs (cf. [19] and references) to generate more appropriate spatial meshes, especially for sectorial domains?
- Extension of the results in Sects. 2 and 3 to the domains $\Omega = \mathbb{R}^3$ and $\Omega = \text{cone} \subset \mathbb{R}^3$?

Acknowledgements The first author (J. Zhang) was supported in part by RGC and FRG of Hong Kong Baptist University. The research of the second author (H. Han) was supported by the NSFC Project No. 10971116. The work of the third author (H. Brunner) was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and by the Hong Kong Research Grants Council (RGC).

References

1. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions, 10th edn. National Bureau of Standards Applied Mathematics Series, vol. 55. Dover, New York (1972)
2. Acosta, G., Duran, R.G., Rossi, J.D.: An adaptive time step procedure for a parabolic problem with blow-up. *Computing* **68**, 343–373 (2002)
3. Bandle, C., Brunner, H.: Numerical analysis of semilinear parabolic problems with blow-up solutions. *Rev. R. Acad. Cienc. Exactas Fís. Nat. Madr.* **88**, 203–222 (1994)
4. Bandle, C., Brunner, H.: Blowup in diffusion equations: a survey. *J. Comput. Appl. Math.* **97**, 2–22 (1998)
5. Berger, M., Kohn, R.V.: A rescaling algorithm for the numerical calculation of blowing-up solutions. *Commun. Pure Appl. Math.* **41**, 841–863 (1988)

6. Brändle, C., Quiórs, F., Rossi, J.D.: An adaptive numerical method to handle blow-up in a parabolic system. *Numer. Math.* **102**, 39–59 (2005)
7. Brunner, H., Wu, X., Zhang, J.: Computational solution of blow-up problems for semilinear parabolic PDEs on unbounded domains. *SIAM J. Sci. Comput.* **31**, 4478–4496 (2010)
8. Budd, C.J., Huang, W., Russell, R.D.: Moving mesh methods for problems with blow-up. *SIAM J. Sci. Comput.* **2**, 305–327 (1996)
9. Deng, K., Levine, H.A.: The role of the critical exponent in blow-up problems: the sequel. *J. Math. Anal. Appl.* **243**, 85–126 (2000)
10. Fevens, T., Jiang, H.: Absorbing boundary conditions for the Schrödinger equation. *SIAM J. Sci. Comput.* **21**, 255–282 (2000)
11. Fujita, H.: On the blowing up of solutions of the Cauchy problem for $u_t = \Delta u + u^{1+\alpha}$. *J. Fac. Sci. Univ. Tokyo Sect. IA, Math.* **13**, 109–124 (1966)
12. Hagstrom, T., Keller, H.B.: Asymptotic boundary conditions and numerical methods for nonlinear elliptic problems on unbounded domains. *Math. Comput.* **48**, 449–470 (1987)
13. Han, H.: The artificial boundary method—numerical solutions of partial differential equations on unbounded domains. In: *Frontiers and Prospects of Contemporary Applied Mathematics*, pp. 33–58 (2005)
14. Han, H., Huang, Z., Yin, D.: Exact artificial boundary conditions for quasilinear elliptic equations in unbounded domains. *Commun. Math. Sci.* **6**, 71–83 (2008)
15. Han, H., Wu, X.: Approximation of infinite boundary condition and its applications to finite element methods. *J. Comput. Math.* **3**, 179–192 (1985)
16. Han, H., Wu, X.: *Artificial Boundary Method—Numerical Solution of Partial Differential Equations on Unbounded Domains*. Tsinghua University Press, Beijing (2009)
17. Han, H., Wu, X., Xu, Z.: Artificial boundary conditions for Burgers' equation using nonlinear boundary conditions. *J. Comput. Math.* **24**, 295–304 (2006)
18. Han, H., Zhang, Z.: Split local absorbing boundary conditions for one-dimensional nonlinear Klein-Gordon equation on unbounded domain. *J. Comput. Phys.* **227**, 8992–9004 (2008)
19. Huang, W., Ma, J., Russell, R.D.: A study of moving mesh PDE methods for numerical simulation of blow-up in reaction diffusion equations. *J. Comput. Phys.* **227**, 6532–6552 (2008)
20. Levine, H.A.: The role of the critical exponent in blow-up problems. *SIAM Rev.* **32**, 262–288 (1990)
21. Nakagawa, T.: Blowing up of a finite difference solution to $u_t = u_{xx} + u^2$. *Appl. Math. Optim.* **2**, 337–350 (1975)
22. Nakagawa, T., Ushijima, T.: Finite element analysis of the semi-linear heat equation of blow-up type. In: Miller, J.J.H. (ed.) *Topics in Numerical Analysis III*, pp. 275–291. Academic Press, London (1977)
23. Samarskii, A.A., Galaktionov, V.A., Kurdyumov, S.P., Mikhailov, A.P.: *Blow-up in Quasilinear Parabolic Equations*. Walter de Gruyter, Berlin (1995)
24. Souplet, P.: Uniform blow-up profiles and boundary behavior for diffusion equations with nonlocal nonlinear source. *J. Differ. Equ.* **153**, 374–406 (1999)
25. Souplet, P.: Uniform blow-up profile and boundary behavior for a non-local reaction-diffusion equation with critical damping. *Math. Methods Appl. Sci.* **27**, 1819–1829 (2004)
26. Souplet, P.: A note on diffusion-induced blow-up. *J. Dyn. Differ. Equ.* **19**, 819–823 (2007)
27. Strang, G.: On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.* **5**, 506–517 (1968)
28. Wu, X., Zhang, J.: Artificial boundary method for two-dimensional Burgers' equation. *Comput. Math. Appl.* **56**, 242–256 (2008)
29. Xu, Z., Han, H.: Absorbing boundary conditions for nonlinear Schrödinger equations. *Phys. Rev. E* **74**, 037704 (2006)
30. Xu, Z., Han, H., Wu, X.: Numerical method for the deterministic Kardar-Parisi-Zhang equation in unbounded domains. *Commun. Comput. Phys.* **1**, 479–493 (2006)
31. Xu, Z., Han, H., Wu, X.: Adaptive absorbing boundary conditions for Schrödinger-type equations: application to nonlinear and multi-dimensional problems. *J. Comput. Phys.* **225**, 1577–1589 (2007)
32. Zhang, J., Xu, Z., Wu, X.: Unified approach to split absorbing boundary conditions for nonlinear Schrödinger equations. *Phys. Rev. E* **78**, 026709 (2008)
33. Zhang, J., Xu, Z., Wu, X.: Unified approach to split absorbing boundary conditions for nonlinear Schrödinger equations: Two dimensional case. *Phys. Rev. E* **79**, 046711 (2009)
34. Zheng, C.: Exact nonreflecting boundary conditions for one-dimensional cubic nonlinear Schrödinger equations. *J. Comput. Phys.* **215**, 552–565 (2006)
35. Zheng, C.: Numerical solution to the sine-Gordon equation defined on the whole real axis. *SIAM J. Sci. Comput.* **29**, 2494–2506 (2009)