J Sci Comput
DOI 10.1007/s10915-016-0179-8

Numerical Blow-Up of Nonlinear Parabolic Integro-Differential Equations on Unbounded Domain

Hermann Brunner[1,2] · **Tao Tang**[3] · **Jiwei Zhang**[4]

Received: 23 November 2014 / Revised: 11 December 2015 / Accepted: 5 February 2016

Abstract The efficient numerical methods of the nonlinear parabolic integro-differential PDEs on unbounded spatial domains whose solutions blow up in finite time are considered. Based on the unified approach proposed in Zhang et al. (Phys Rev E 78:026709, 2008), Zhang et al. (Phys Rev E 79:046711, 2009), the nonlinear absorbing boundary conditions for one-dimensional and two-dimensional nonlinear parabolic integro-differential PDEs are derived. Thus the original problem on the unbounded spatial domain is reduced to an initial-boundary-value (IBV) problem on a bounded computational domain. Secondly, a simple but efficient adaptive time-stepping scheme for the reduced IBV problem is achieved by using the fixed point method to make the finite difference approximation stable at each time level. At each time level, we also prove that the lower bound and upper bound of the blow-up time can be bounded by the numerical blow-up times of the forward and backward Euler schemes. Finally, the theoretical results are illustrated by a broad range of numerical examples, including a problem with a circle line blow-up.

✉ Jiwei Zhang
jwzhang@csrc.ac.cn

Hermann Brunner
hbrunner@math.hkbu.edu.hk

Tao Tang
ttang@math.hkbu.edu.hk

1 Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong

2 Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL A1C 5S7, Canada

3 Department of Mathematics & Institute for Computational and Theoretical Studies, Hong Kong Baptist University, Kowloon Tong, Hong Kong

4 Beijing Computational Science Research Center, No. 10 Dongbeiwang West Road, Haidian District, Beijing 100094, China

Published online: 16 February 2016

Keywords Nonlinear parabolic integro-differential PDEs · Unbounded spatial domains · Unified approach · Finite time blow-up · Nonlinear absorbing boundary conditions · Adaptive time-stepping

Mathematics Subject Classification 65M06 · 65L10 · 35Q53 · 35Q51

1 Introduction

The integro-differential PDEs come from many problems in the applied sciences to model the dynamical systems, and reflect the effects of the memory of the system in model, such as the heat transfer, nuclear reactor dynamics and thermoelasticity and so on. To remedy the difficulty of the effect of past history will lead to the integro-differential PDEs. In this paper we consider efficient numerical methods for the nonlinear parabolic integro-differential equation with blow-up solution on an unbounded domain,

$$u_t = \Delta u + \int_0^t \kappa(t-s)u^p(\mathbf{x},s)ds,\ \ \mathbf{x} \in \mathbb{R}^N,\ \ p > 1,\ \ t \in (0,T], \tag{1.1}$$

where the initial function $u|_{t=0} = u_0$ is assumed to be compactly supported and non-negative. The convolution kernel satisfies $\kappa(z) > 0$ and $\kappa'(z) \leq 0$ for all $z \geq 0$. This parabolic equation is used to describe the blow-up in a chemical system, which has a dramatic increase in temperature leading to ignition or explosion. The finite time blowup in this paper means the maximal interval of existence is bounded, in other words, the solution becomes unbounded at some point in the spatial domain at finite time. The Laplacian term is dissipative, giving a negative contribution to the time derivative at the local spatial maximum, namely, it tends to drive the solution towards to a constant value. The nonlinear term will make the solution blowup, where the popular kernels are the constant kernel $\kappa(t) \equiv 1$, the exponent kernel $\kappa(t) = \exp(-t)$ and the Abel kernel $\kappa(t) = t^{\alpha-1}$ with $\alpha > 0$.

Nonlinear integro-differential PDEs have been widely investigated both from the analytical and numerical view points. For the smoothly bounded domain, Bellout [4] studies the blow-up solutions of the integro-differential equation. Li and Xie [15] consider the existence of a blow-up solution for a nonlinear parabolic equation with nonlinear memory. Souplet examines the monotonity of the solution and blow-up for the semilinear parabolic equations with nonlinear memory [22]. For more recent results, one can refer to [17].

For the unbounded spatial domain, Yanik et al. [31] analyze the space-time dependent nuclear dynamics, blow-up problem. Souplet [23,24] shows the critical exponent $p^* = \infty$ for problem (1.1) with $\kappa(t) \equiv 1$, which means that the solution will blowup at finite time for any nonnegative initial value. Yang et al. [30] show the critical exponent is $p^* = \infty$ for problem (1.1) with $\kappa(t) = t^{\alpha-1}$ when $\alpha \geq 1$. One can see that the constant kernel is a simple case of the Abel kernel with $\alpha = 1$. Yang et al. [30] also present that there exists a critical exponent $1 + \frac{2}{N} \leq p^* \leq 1 + \frac{2+2\alpha}{N}$ for the kernel $\kappa(t) = t^{\alpha-1}\exp(-t)$.

For the numerical simulations, Hua and Liang [20], Liang and Zhu [16] study the numerical blow-up using the moving collocation method. Han et al. [9,10] propose the exact ABCs of the linear parabolic Volterra integro-differential equations. Farhad et al. [7] consider the spectral methods for the problem (1.1) with $p = 1$. For the nonlinear problem (1.1) with $p > 1$, the efficient numerical method has not been treated when the solution will blow up at finite time. In this paper we consider the efficient numerical computation of the problem (1.1) with $p > 1$.

The core of solving problems on the unbounded spatial domain numerically is to construct the appropriate absorbing boundary conditions (ABCs) to reformulate the original problem to a problem on a bounded computational domain. In this paper we use artificial boundary methods (ABMs), see [11,27,35,36], to derive ABCs. For nonlinear problems, the traditional Laplace transform and Fourier transform can not be used directly to design ABCs, hence we need introduce new strategies. Here we extend the methodology of the unified approach proposed in [32,33] to derive the effective ABCs for problem (1.1). The main idea underlying unified approach is based on the operator splitting method, and the detailed description is given in Sect. 2.1.

On the other hand, we not only provide an adaptive time step procedure to make the numerical schemes stable, but also address a main question: Given the initial function u_0 and a threshold M, which solutions of systems (3.1) and (3.2) can reach the "blow-up thresholds" M first? That is to say, which solutions $U(t)$ and $V(t)$ blow numerically up first? We will describe this question from the viewpoint of the error $e(t) := U(t) - V(t)$ for forward Euler method and backward Euler method, respectively, and prove that the numerical blow-up times can be bounded by the blow-up times of the forward and backward Euler schemes.

The paper is organized as follows. In Sect. 2, the unified approach is introduced to derive the nonlinear ABCs for one-dimensional and two-dimensional cases. In Sect. 3, the numerical schemes for approximating the solution of the resulting initial-boundary-value (IBV) problem for (1.1) are presented, and the adaptive time-stepping scheme is provided by an extension of the fixed piont method of Bandle and Brunner [2]. Furthermore, some geometrical aspects of the numerical (time-stepping) scheme is described, i.e., the numerical blow-up time of the reduced problem can be bounded by the forward and backward Euler methods. Then we use Sect. 4 to illustrate the effectiveness of our approach in various aspects. Future work and some open problems are addressed in the concluding Sect. 5.

2 Construction of Nonlinear ABCs

In this section, we will address the general principle of the unified approach, which is proposed in [32,33] for the study of the nonlinear Schrödinger equation, and has been successfully applied to semilinear parabolic Eqs. [5,21,34]. Using this approach, we derive the effective nonlinear ABCs for the Eq. (1.1) on the chosen computational domain Ω_i defined below.

2.1 Preliminary

Based on the well-known time-splitting method (or split-step method), the following procedure is called the unified approach. According to the philosophy of the unified approach, we first rewrite the general equation in operator form,

$$u_t = \mathcal{L}u + \mathcal{N}u, \tag{2.1}$$

where $\mathcal{L}$ and $\mathcal{N}$ correspond to a linear differential operator and a nonlinear operator. In this paper, the operators are specially defined by

$$\mathcal{L}u = \Delta u, \quad \mathcal{N}u = \int_0^t \kappa(t-s)u^p(x,s)ds,$$

In analogy to the widely used Strang splitting [25],

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

and the Baker-Campbell-Hausdorff theorem using the first-order approximation [13]

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

in a small time interval from t to $t+\tau$ with $\tau > 0$, we use the approximation

$$u(x, t+\tau) \approx e^{(\mathcal{L}+\mathcal{N})\tau} u(x, t). \tag{2.2}$$

In the approximate (2.2), we approximate the linear operator $\mathcal{L}$ by a suitable approximation operator $\mathcal{L}^{(n)}$ to make the wave outgoing when the waves reach to the artificial boundary. Then combining the approximation operator $\mathcal{L}^{(n)}$ with the nonlinear operator $\mathcal{N}$ and letting $\tau \to 0$ reads

$$u_t = \mathcal{L}^{(n)} u + \mathcal{N} u. \tag{2.3}$$

Equation (2.3) will be our ABCs when we constrain it at the artificial boundaries. Now we consider how to derive the one-way approximation operator $\mathcal{L}^{(n)}$ for one-dimensional and two-dimensional cases.

2.2 Nonlinear ABCs for One-Dimensional Case

To obtain the approximation $\mathcal{L}^{(n)}$ of linear operator $\mathcal{L}$, we first consider the design of local ABCs for heat equation on exterior domain $\Omega_e := (-\infty, x_l] \bigcup [x_r, \infty)$, given by

$$\begin{cases} u_t - u_{xx} = 0, & x \in \Omega_e; \\ u(x, 0) = 0, & x \in \Omega_e; \\ u \to 0, & \text{as} \quad |\mathbf{x}| \to \infty. \end{cases} \tag{2.4}$$

Define Laplace transform by

$$\widetilde{u}(x, s) = \int_0^\infty e^{-st} u(x, t) dt. \tag{2.5}$$

Applying the Laplace transform to the first equation in (2.4), we have

$$s\tilde{u} - \tilde{u}_{xx} = 0. \tag{2.6}$$

The Eq. (2.6) has two independent eigensolutions $e^{\sqrt{s}x}$ and $e^{-\sqrt{s}x}$. Noting the last condition in (2.4), and taking the derivative of the eigensolutions with respect to x, we get

$$\widetilde{u}_x \pm \sqrt{s}\tilde{u} = 0 \tag{2.7}$$

on the artificial boundaries $\Gamma := \{x_l, x_r\}$. The plus sign in "$\pm$" corresponds to the right boundary condition, and the minus sign corresponds to the left one. Applying the inverse Laplace transform to Eq. (2.7) results in the exact ABCs

$$\partial_n u(x, t) + \frac{1}{\sqrt{\pi}} \int_0^t \frac{\partial_\tau u(x, \tau)}{\sqrt{t-\tau}} d\tau = 0, \tag{2.8}$$

where ∂_n is the outer normal derivative. It is impossible to combine the exact ABCs (2.8) with the nonlinear operator in (2.3). Alternately, we approximate $\sqrt{s}$ by the padé expansion at point s_0:

$$\sqrt{s} \approx \sqrt{s_0} - \sqrt{s_0} \sum_{k=1}^{K} \frac{b_k (s_0 - s)}{s_0 - a_k (s_0 - s)}, \tag{2.9}$$

where $a_k = \cos^2(\frac{k\pi}{2K+1})$, $b_k = \frac{2}{2K+1}\sin^2(\frac{k\pi}{2K+1})$, $k = 1, 2, \cdots, K$. The approximation (2.9) will lead to high-order ABCs, and their stability analysis and effectiveness are discussed in [28]. Here we only consider the simplest case $K = 1$, i.e.,

$$\sqrt{s} \approx \sqrt{s_0}\,(s_0 + 3s)\,/\,(3s_0 + s)\,. \tag{2.10}$$

Substituting the approximation (2.10) into (2.7), and applying inverse laplace transform yield

$$(\partial_x \pm 3\sqrt{s_0})\partial_t u = -(3s_0\partial_x \pm s_0\sqrt{s_0})u,$$

which implies the one-way approximate operator

$$\mathcal{L} \approx \mathcal{L}^{(3)} = -\left(\partial_x \pm 3\sqrt{s_0}\right)^{-1}\left(3s_0\partial_x \pm s_0\sqrt{s_0}\right). \tag{2.11}$$

Substituting (2.11) into the one-way Eq. (2.3), we have the third-order nonlinear ABCs

$$\pm 3\sqrt{s_0}u_t + 3s_0u_x + u_{xt} \pm s_0\sqrt{s_0}u = \int_0^t \kappa(t-s)u^{p-1}\left(pu_x \pm 3\sqrt{s_0}u\right)ds. \tag{2.12}$$

We remark that ABCs (2.12) do not bring any extra computational burden since we only use one-directional approximation operator to approximate the linear operator, make a little change of the nonlinear integro-differential term to make the wave outgoing.

Thus the problem (1.1) restricted to the truncated computational interval $[x_l, x_r]$ is reduced to an IBV problem with nonlinear ABCs, given by

$$\begin{cases} u_t = u_{xx} + \int_0^t \kappa(t-\tau)u^p(x,s)ds, \quad x \in (x_l, x_r),\ p > 1, \\ u(x,0) = u_0(x), \quad x \in (x_l, x_r), \\ 3s_0u_x + 3\sqrt{s_0}u_t + u_{xt} + s_0\sqrt{s_0}u = \int_0^t \kappa(t-s)u^{p-1}\left(pu_x + 3\sqrt{s_0}u\right)ds, \quad x = x_r, \\ 3s_0u_x - 3\sqrt{s_0}u_t + u_{xt} - s_0\sqrt{s_0}u = \int_0^t \kappa(t-s)u^{p-1}\left(pu_x - 3\sqrt{s_0}u\right)ds, \quad x = x_l. \end{cases} \tag{2.13}$$

2.3 Nonlinear ABCs for the Two-Dimensional Case

To compare the two most commonly chosen computational domains (rectangle and disc), we use this section to design the respective nonlinear ABCs for the problem (1.1) in two dimension, see analogous strategies in [5,34].

2.3.1 ABCs for a Rectangle

We first construct nonlinear ABCs on a rectangle $\Omega_i := \{(x, y) : 0 \le x \le L; 0 \le y \le L\}$, where L is a given positive constant. Using an argument similar to one-dimensional case, we consider heat equation on the exterior domain $\Omega_e := \mathbb{R}^2/\Omega_i$:

$$u_t = u_{xx} + u_{yy}, \quad (x, y) \in \Omega_e, \tag{2.14}$$

with the initial value $u_0 = 0$. Define the Fourier transform by

$$\widehat{u}(\xi, \eta, t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} u(x, y, t)e^{-i\xi x - i\eta y}dxdy.$$

Applying the Fourier transform and the Laplace transform to (2.14), we have

$$s + \xi^2 + \eta^2 = 0. \tag{2.15}$$

Solving (2.15), and only allowing waves outgoing, we have the one-directional dispersion relations on the east and west artificial boundaries by

$$-i\xi \pm \sqrt{s+\eta^2} = 0. \tag{2.16}$$

Here, the plus sign in "±" stands for the positive direction, and the minus sign stands for the negative direction. Setting $z = \eta^2 + s$, expanding $\sqrt{z}$ by using formula (2.9) with $K = 1$, substituting the approximation into (2.16), and solving the resulting algebraic equation, we obtain

$$s = -(-i\xi\eta^2 \pm 3\sqrt{\xi_0}\eta^2 - 3i\xi_0\xi - \sqrt{\xi_0}\xi_0)/(-i\xi \pm 3\sqrt{\xi_0}). \tag{2.17}$$

We can get the analogous algebraic equation at the northern and southern boundaries:

$$s = -(-i\xi^2\eta \pm 3\sqrt{\eta_0}\xi^2 - 3i\eta_0\eta - \sqrt{\eta_0}\eta_0)/(-i\eta \pm 3\sqrt{\eta_0}), \tag{2.18}$$

where (ξ_0, η_0) are the padé expansion points. For the corners, we extend the approach in [33] by using the (1,1)-padé approximation to expand ξ^2 and η^2, respectively, and have the algebraic identity at the north-east and the south-west corners

$$s = -\xi_0(-3i\xi \pm \sqrt{\xi_0})/(-i\xi \pm 3\sqrt{\xi_0}) - \eta_0(-3i\eta \pm \sqrt{\eta_0})/(-i\eta \pm 3\sqrt{\eta_0}). \tag{2.19}$$

At the north-west and south-east corners, the approximation algebraic identities read

$$s = -\xi_0(-3i\xi \mp \sqrt{\xi_0})/(-i\xi \mp 3\sqrt{\xi_0}) - \eta_0(-3i\eta \pm \sqrt{\eta_0})/(-i\eta \pm 3\sqrt{\eta_0}). \tag{2.20}$$

The Eqs. (2.17)–(2.20) are the approximation algebraic identities in Laplace and Fourier spaces. Using the duality relation $s \leftrightarrow \partial_t$, $-i\xi \leftrightarrow \partial_x$ and $-i\eta \leftrightarrow \partial_y$, the corresponding local ABCs for heat equation can accordingly be obtained (see [28]). Taking (2.17) for example, the third-order padé approximations on the eastern and western artificial boundaries are given by

$$\mathcal{L}^{(3)} = (\partial_x\partial_y^2 \pm 3\sqrt{\xi_0}\partial_y^2 - 3\xi_0\partial_x + \sqrt{\xi_0}\xi_0)(\partial_x \pm 3\sqrt{\xi_0})^{-1}. \tag{2.21}$$

Replacing $\mathcal{L}^{(3)}$ in (2.3) by form of (2.21), we have the eastern and western nonlinear ABCs

$$3\xi_0 u_x - u_{xyy} + u_{xt} \pm \sqrt{\xi_0}\left(\xi_0 u + 3u_t - 3u_{yy}\right) + \int_0^t \kappa(t-s)\left(\pm 3\sqrt{\xi_0}u^p - pu^{p-1}u_x\right) ds = 0. \tag{2.22}$$

Similarly, we have the northern and southern nonlinear ABCs,

$$3\eta_0 u_y - u_{xxy} + u_{yt} \pm \sqrt{\eta_0}\left(\eta_0 u + 3u_t - 3u_{xx}\right) + \int_0^t \kappa(t-s)\left(\pm 3\sqrt{\xi_0}u^p - pu^{p-1}u_y\right) ds = 0, \tag{2.23}$$

the nonlinear ABCs at the north-east and south-west corners

$$\begin{aligned} &u_{xyt} + (3\xi_0 + 3\eta_0)u_{xy} \pm 3\sqrt{\xi_0}u_{yt} \pm 3\sqrt{\eta_0}u_{xt} \pm \sqrt{\eta_0}(9\xi_0 + \eta_0)u_x \\ &\quad + 9\sqrt{\xi_0\eta_0}u_t \pm \sqrt{\xi_0}(9\eta_0 + \xi_0)u_y + 3\sqrt{\xi_0\eta_0}(\xi_0 + \eta_0)u \\ &= \int_0^t \kappa(t-s)\left((u^p)_{xy} \pm 3\sqrt{\xi_0}pu^{p-1}u_y \pm 3\sqrt{\eta_0}pu^{p-1}u_x + 9\sqrt{\xi_0\eta_0}u^p\right) ds, \end{aligned} \tag{2.24}$$

as well as the nonlinear ABCs at the northwest and southeast corners

$$\begin{aligned} &u_{xyt} + (3\xi_0 + 3\eta_0)u_{xy} \mp 3\sqrt{\eta_0}u_{yt} \pm 3\sqrt{\xi_0}u_{xt} \pm \sqrt{\xi_0}(9\eta_0 + \xi_0)u_x - 9\sqrt{\xi_0\eta_0}u_t \\ &-9\sqrt{\xi_0\eta_0}u_t \mp \sqrt{\eta_0}(9\xi_0 + \eta_0)u_y - 3\sqrt{\xi_0\eta_0}(\xi_0 + \eta_0)u \\ &= \int_0^t \kappa(t-s)\left((u^p)_{xy} \mp 3\sqrt{\xi_0}pu^{p-1}u_y \pm 3\sqrt{\eta_0}pu^{p-1}u_x - 9\sqrt{\xi_0\eta_0}u^p\right)ds. \end{aligned} \tag{2.25}$$

Thus, the problem (1.1) is reduced to an IBV problem on a rectangle with the nonlinear ABCs (2.22)–(2.25). In the following, we consider to introduce a circle to limit the computational domain and design ABCs on the circle.

2.3.2 ABCs for a Circular Disc

Denote the circular artificial boundary by $\Gamma_R := \{(r,\theta) :\ r = R,\ 0 \le \theta < 2\pi\}$, the interior (computational) domain by $\Omega_i := \{(r,\theta) :\ 0 \le \theta < 2\pi,\ \ 0 \le r < R\}$, and the exterior domain by $\Omega_e := \{(r,\theta) :\ 0 \le \theta < 2\pi,\ \ R < r < \infty\}$. Then the constrain of the problem (1.1) on Ω_e is written in the polar coordinate by

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

with $\mathcal{L}_1 u := \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r}$, $\mathcal{L}_2 u := \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}$, $\mathcal{N}u := \int_0^t \kappa(t-s)u^p(r,\theta,s)ds$. To obtain the approximate operator of linear term $\mathcal{L}_1$, we first study an associated problem,

$$\begin{cases} u_t = \mathcal{L}_1 u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r}, \quad R < r < +\infty, \\ u|_{t=0} = 0, \quad R < r < +\infty, \\ u \to 0, \quad \text{when } r \to +\infty. \end{cases} \tag{2.26}$$

Appling Laplace transform to the first equation in (2.26), we have

$$\frac{\partial^2 \hat{u}}{\partial r^2} + \frac{1}{r}\frac{\partial \hat{u}}{\partial r} - s\hat{u} = 0, \tag{2.27}$$

which possesses two linearly independent solutions $K_0(\sqrt{s}r)$ and $I_0(\sqrt{s}r)$. The special functions $K_0(x)$, $I_0(x)$ are the modified Bessel functions of order zero. From the last condition in (2.26), we know that the solution $\hat{u}$ satisfies

$$\hat{u}(r,s) = CK_0(\sqrt{s}r).$$

Differentiating the above $\hat{u}(r,s)$ with respect to r, we get

$$\frac{\partial \hat{u}}{\partial r}(R,s) = w(s)\hat{u}, \tag{2.28}$$

where $w(s) = \sqrt{s}K_0'(\sqrt{s}R)/K_0(\sqrt{s}R) = -\sqrt{s}K_1(\sqrt{s}R)/K_0(\sqrt{s}R)$. We now approximate $w(s)$ using the simplest padé approximant by

$$w(s) \approx \frac{\lambda s + \beta}{\gamma s + \delta}. \tag{2.29}$$

For a given s_0, the coefficients $(\lambda, \beta, \gamma, \delta)$ will be uniquely determined. Upon inserting (2.29) into Eq. (2.28), and with the application of the inverse Laplace transform, we have

$$(\gamma\partial_r - \lambda)\partial_t u = -\delta\partial_r u + \beta u, \tag{2.30}$$

which implies

$$\mathcal{L}_1 \approx \mathcal{L}_1^{(3)} := (\gamma\partial_r - \lambda)^{-1}(-\delta\partial_r + \beta). \tag{2.31}$$

Applying the philosophy of unified approach again, we get the approximating one-way equation

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

After a simple calculation, the nonlinear ABCs on the circular artificial boundary are achieved by

$$\gamma u_{tr} - \lambda u_t = -\delta u_r + \beta u - \frac{2\gamma + R\lambda}{R^3} u_{\theta\theta} + \frac{\gamma}{R^2} u_{\theta\theta r} + \int_0^t \kappa(t-s)\left[\gamma p u^{p-1} u_r(s) - \lambda u^p(s)\right] ds.$$

Thus the problem of (1.1) is reduced to an IBV problem on a chosen disc

$$\begin{cases} u_t = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} + \int_0^t \kappa(t-s) u^p(r,\theta,s) ds, & (r,\theta) \in \Omega_i, \\ u|_{t=0} = u_0(r,\theta), & (r,\theta) \in \Omega_i, \\ \gamma u_{tr} - \lambda u_t - \delta u_r - \beta u + \frac{2\gamma + R\lambda}{R^3} u_{\theta\theta} - \frac{\gamma}{R^2} u_{\theta\theta r} = \\ \quad + \int_0^t \kappa(t-s)\left[\gamma p u^{p-1} u_r(r,\theta,s) - \lambda u^p(r,\theta,s)\right] ds, & (r,\theta) \in \Gamma_R. \end{cases} \tag{2.32}$$

Remark From the procedure of constructing the nonlinear ABCs, the ABCs only annihilate the energy arising in the interior computational domain by using one-directional equation to make the wave outgoing; and it will not propagate energy into the interior domain to disrupt the true solution. Thus, the perturbation caused by ABCs will have no effect on the interior solution, hence the obtained ABCs are *stable*.

2.4 The Choice of the Parameter s_0

In (2.9) and (2.29), the parameter s_0 is the expansion point for padé approximant. The choice of s_0 produces effects on the performance of the obtained ABCs. In practical applications, one can choose s_0 adaptively for any given initial values u_0, see the energy-weighted wave number approach given in [29] by

$$k_0 = \int_0^\infty |\bar{u}(k,t)|^m k \mathrm{d}k \Big/ \int_0^\infty |\bar{u}(k,t)|^m \mathrm{d}k, \tag{2.33}$$

where $m = 4$ is more efficient and accurate, and $\bar{u}$ denotes Gabor transform given by

$$\bar{u}(k,t) = \int_{x_r - \mathbf{b}}^{x_r} u(x,t) e^{-ikx} dx,$$

with $\mathbf{b}$ representing the window width. For wave equation, we take $s_0 = k_0^2$ for the dispersion relation between wave-number and frequency. For heat equation, although the relation $s_0 = f(k_0)$ is still unknown, we still can take $s_0 = k_0^2$ for an approximation. In our situation shown in Sect. 4, the numerical simulation is not sensitive to the choice of s_0, we may use a fixed value of s_0.

Similarly, for a rectangle, the algorithm of determining ξ_0 and η_0 can be extended straightforwardly from that of one dimension. For any given value s_0 in the approximation (2.29) with the circular case, we can resort to the function `PadeApproximant` in the toolbox of Mathematica to obtain the values of the parameters $\lambda, \beta, \gamma, \delta$ in (2.29) (see Table 1 for the given s_0 and R).

Table 1 Evaluation of parameters in (2.29) for the given s_0, R

(s_0, R)	λ	β	γ	δ
(10, 2.5)	−0.242527	−0.931359	0.0250725	0.749275
(10, 4.5)	−0.240072	−0.87084	0.0250255	0.749745
(10, 6)	−0.239326	−0.851302	0.025015	0.749850

3 Setting of the Approximating Problems

Denote δ_k the unit vector in the direction of the x_k axis, and let $\{P_i\}$ be the set of grid points, where $P_1, \cdots, P_I$ denote the grid points contained in the domain Ω_i. At a grid point P_i, we use the second-roder central difference to approximate the spatial derivatives $u_{x_k x_k}$ with mesh diameter $h > 0$ by

$$D_k^2 u := h^{-2} \cdot (u(P_i + h\delta_k, t) - 2u(P_i, t) + u(P_i - h\delta_k, t)).$$

The spatially discretized version of Eq. (1.1) is then given by the large system of nonlinear Volterra integro-differential equations

$$\dot{U}(P_i, t) = \sum_{k=1}^{N} D_k^2 U(P_i, t) + \int_0^t \kappa(t-s) U^p(P_i, s) ds =: G(U(P_i, t)) \tag{3.1}$$

with initial conditions $U(P_i, 0) = u(P_i, 0)$ $(i = 1, \cdots, I)$. Let $U(t) = (U(P_1, t), \cdots, U(P_I, t))^T$, and consider a (generally non-uniform, adaptively chosen) temporal grid $\tau_m := t_{m+1} - t_m$ for $[0, T]: \ 0 = t_0 < t_1 < \cdots < t_M = T$. Thus $U(t)$ can be approximated by piecewise linear functions $V(t)$ in $C^0(0, T)$. For $P_i \in \Omega_i$, the system of algebraic equations are given by

$$\begin{aligned} V(P_i, t_{m+1}) - V(P_i, t_m) = \tau_m \sum_{k=1}^{N} D_k^2 (V(P_i, t_m) + c(V(P_i, t_{m+1}) - V(P_i, t_m)) \\ + \tau_m \left[F^m(P_i, t_{m+1}) + V^p(P_i, t_m + c\tau_m) \int_{t_m}^{t_{m+1}} \kappa(t_{m+1} - s) ds \right], \end{aligned} \tag{3.2}$$

with $m = 0, \cdots, M-1$ and $F^m(P_i, t_{m+1}) = \sum_{l=1}^{m} V^p(P_i, t_l + c\tau_l) \int_{t_{l-1}}^{t_l} \kappa(t_{m+1} - s) ds$.

The parameters $c \in \{0, \ 1/2, \ 1\}$ lead to three specific time-stepping methods known in the literatures: the forward Euler, the Crank-Nicolson and the backward Euler methods.

For the reduced problem (2.32), 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 $\triangle r := R/I$ and $\triangle\theta := 2\pi/J$ be the spatial mesh sizes, and the corresponding grid points $r_i = i\triangle r$ and $\theta_j = j\triangle\theta$.

Denote the approximation of $u(r_i, \theta_j, t_m)$ by $u_{i,j}^m$ and set $h := \min\{\triangle r, \triangle\theta\}$, the time steps $\tau_m := t_{m+1} - t_m$. The Crank-Nicolson scheme of the Eq. in (2.32) is discretized by

$$\begin{aligned} D_m u_{i,j}^m =& D_i^2 u_{i,j}^{m+\frac{1}{2}} + \frac{1}{r_i} D_i u_{i,j}^{m+\frac{1}{2}} + \frac{1}{r_i^2} D_j^2 u_{i,j}^{m+\frac{1}{2}} + F^m(r_i, \theta_j, t_{m+1}) \\ & + (u_{ij}^{m+\frac{1}{2}})^p \int_{t_m}^{t_{m+1}} \kappa(t_{m+1} - s) ds, \end{aligned} \tag{3.3}$$

with $u_{i,0}^m = u_{i,J}^m,\ u_{i,-1}^m = u_{i,J-1}^m,\ \ 0 < i < I,\ 0 \le j < J$ and

$$u_{i,j}^{m+\frac{1}{2}} = \frac{u_{i,j}^m + u_{i,j}^{m+1}}{2}, \quad F^m(r_i, \theta_j, t_{m+1}) = \int_0^{t_m} \kappa(t_{m+1} - s) \left(u_{ij}(s)\right)^p ds,$$
$$D_i^2 u_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\triangle r^2}, \quad D_j^2 u_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\triangle \theta^2},$$
$$D_m u^m = \frac{u^{m+1} - u^m}{\tau_m}, \qquad D_i u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\triangle r}.$$

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

$$\gamma D_{I-1} D_m u_{I-1,j}^{m+1} - \lambda D_m u_{I-1,j}^m + \delta D_{I-1} u_{I-1,j}^{m+\frac{1}{2}} - \beta u_{I-1,j}^{m+\frac{1}{2}} + \frac{2\gamma + \lambda r_{I-1}}{r_{I-1}^3} D_j^2 u_{I-1,j}^{m+\frac{1}{2}}$$
$$- \frac{\gamma}{r_{I-1}^2} D_{I-1} D_j^2 u_{I-1,j}^{m+\frac{1}{2}} + (u_{I-1,j}^{m+\frac{1}{2}})^{p-1} \left[\lambda(u_{I-1,j}^{m+\frac{1}{2}}) - p\gamma D_{I-1} u_{I-1,j}^{m+\frac{1}{2}}\right]$$
$$\int_{t_m}^{t_{m+1}} \kappa(t_{m+1} - s) ds$$
$$+ \lambda F^m(r_{I-1}, \theta_j, t_{m+1}) - p\gamma \int_0^{t_m} \kappa(t_{m+1} - s) u^{p-1} u_r(r_{I-1,j}, \theta_j, s) ds = 0. \tag{3.4}$$

For the θ-direction, the periodic boundary conditions are applied: $u_{0,j}^m = u_{0,0}^m$ $(j = 1, \ldots, J;\ \ m = 1, 2, \ldots)$. For the case $i = 0$, there is a singularity of the coefficient of the equation at the origin $(0, 0)$ (i.e., $r = 0$) in (2.32). To circumvent this non-essential singularity, we employ the following strategy to discretize (2.32). In the small disc $\Omega_{\triangle r/2} := \{(r, \theta) | 0 \le r \le \triangle r/2,\ 0 \le \theta < 2\pi\}$, we may write (2.32), resorting to integration by parts, as

$$\frac{\triangle r}{2} \int_0^{2\pi} \frac{\partial u}{\partial r} d\theta = \int_{\Omega_{\triangle r/2}} \left(u_t(r, \theta, t) - \int_0^t \kappa(t - s) u^p(r, \theta, s) ds \right) dr d\theta.$$

Using the composite mid-point rule, we get

$$\frac{2\triangle\theta}{\pi(\triangle r)^2} \left(\sum_{j=0}^{J} u(\triangle r, \theta_j, t) - (J + 1) u(0, 0, t)\right) = u_t(0, 0, t) - \int_0^t \kappa(t - s) u^p(0, 0, s) ds.$$

Thus, we have the discretization of Eq. (2.32) at the point $(0, 0)$ as

$$\frac{2\triangle\theta}{\pi \triangle r^2} \left(\sum_{j=0}^{J} u_{1,j}^{m+\frac{1}{2}} - (J + 1) u_{0,0}^{m+\frac{1}{2}}\right) = \frac{u_{0,0}^{m+1} - u_{0,0}^m}{\tau_m} - F^m(r_0, \theta_0, t_{m+1}) - (u_{0,0}^{m+\frac{1}{2}})^p$$
$$\int_{t_m}^{t_{m+1}} \kappa(t_{m+1} - s) ds. \tag{3.5}$$

Thus the reduced problem (2.32) is approximated by the discretized version of (3.3)–(3.5).

3.1 Adaptive Time-Stepping

An illuminating expression why numerical methods using a fixed time stepsize are not appropriate is studied in [26] when solving nonlinear ODEs with blow-up in a finite time. The

adaptive time stepsize strategies for computing nonlinear parabolic PDEs with blow-up solutions also discussed in [1–3,6,8,14,18,19]. In this paper, we extend the approach proposed in [2,3] to derive a novel adaptive temporal stepsize for the problem (1.1), such that the numerical solutions of the implicit methods (backward Euler method and Crank-Nicolson method) exist uniquely at each time level.

In fact, the above system of nonlinear equations with $c > 0$ are implicit, and one needs an iterative strategy (modified Newton iteration or direct fixed-point iteration) to solve those Eqs. (3.2). Denote

$$\begin{aligned} z_m(P_i) &:= V(P_i, t_m) + c(V(P_i, t_{m+1}) - V(P_i, t_m)) \\ &= V(P_i, t_m) + c\tau_m \cdot \left[\sum_{k=1}^{N} D_k^2 z_m(P_i) + F_m(P_i, t_{m+1}) + \int_{t_m}^{t_{m+1}} \kappa(t_{m+1} - s) z_m^p(P_i) ds \right]. \end{aligned} \tag{3.6}$$

Given $V(P_i, t_m)$ for all $P_i \in \Omega_i$, the Eq. (3.6) represents a nonlinear system of algebraic equations for unknowns $z_m(P_i)$, $i = 1, \cdots, I$. Existence will be proved by means of a fixed point argument. For this purpose the time step $\tau_m = t_{m+1} - t_m$ must be chosen such that the mapping $H: \mathbb{R}^I \to \mathbb{R}^I$ that defines the solution $z_m(P_i)$ is contractive. The ith component of the mapping H is given by

$$\begin{aligned} H_i(z_m) := V(P_i, t_m) + c\tau_m \cdot \Big[&\textstyle\sum_{k=1}^{N} D_k^2 z_m(P_i) + F_m(P_i, t_{m+1}) \\ &+ \textstyle\int_{t_m}^{t_{m+1}} \kappa(t_{m+1} - s) z_m^p(P_i) ds \Big] \end{aligned}$$

with $z_m := (z_m(P_1), \cdots, z_m(P_r))^T$. Let $\rho_m := |V(P_i, t_m)|_\infty = \max\limits_{0<i<I} |V(P_i, t_m)|$, and the ball $B(\alpha\rho_m) := \left\{ v \in \mathbb{R}^I : |v| < \alpha\rho_m \right\}$ for given $\alpha > 1$. For $\forall z_m \in B(\alpha\rho_m)$, we have

$$|H(z_m)|_\infty \le c\tau_m \left(\frac{4N}{h^2} \alpha\rho_m + \sum_{m=0}^{M} (\alpha\rho_m)^p \int_{t_m}^{t_{m+1}} \kappa(t_{m+1} - s) ds \right) + \rho_m.$$

This estimate leads to the adaptive time step

$$\tau_m < \frac{\rho_m(\alpha - 1)}{c(4N\alpha\rho_m / h^2 + \sum_{m=1}^{N} (\alpha\rho_m)^p \int_{t_m}^{t_{m+1}} \kappa(t_{m+1} - s) ds)}, \tag{3.7}$$

which makes the above mapping from $B(\alpha\rho_m)$ to $B(\alpha\rho_m)$ contractive at the time level $t = t_{m+1}$.

3.2 Geometry of the Time-Discretization

The main question is: Given the initial function u_0 and a threshold M, which solutions of systems (3.1) and (3.2) can reach the "blow-up thresholds" M first? Namely, which solutions $U(t)$ and $V(t)$ blow numerically up first? We will address this question from the viewpoint of the error $e(t) := U(t) - V(t)$ for forward Euler method and backward Euler method, respectively. The basic idea is to investigate the sign of $e(t)$ at each time level, thus we can obtain the lower bound and upper bound of the solutions $U(t)$ of Eq. (3.2) for the given M. In this paper we consider the simplest case $\kappa(t) \equiv 1$, although the results as follows can be extend to the case that $\kappa(t)$ is positive and nonincreasing. Let us now integrate the Eq. (3.1) from t_m to t_{m+1}, to obtain

$$U(P_i, t_{m+1}) = U(P_i, t_m) + \int_{t_m}^{t_{m+1}} G(U(P_i, t)) dt, \tag{3.8}$$

where $G(U(P_i,t)) = \sum_{k=1}^{N} D_k^2 U(P_i,t) + \int_0^t U^p(P_i,s))ds$. The two specific finite difference schemes: forward Euler method ($c = 0$) and backward Euler method ($c = 1$) are discussed.

Lemma 3.1 *Let $L(\rho(P_i,\varsigma_m))$ $(t_m < \varsigma_m < t_{m+1})$ the function at the point P_i such that $u^p(a) - u^p(b) = L(\rho)(a-b)$, we have*

1. *for $c = 0$:*

$$e(P_i,t_{m+1}) = \left[1 + \tau_m\left(\sum_{k=1}^{N} D_k^2 + \tau_m L(\rho,\varsigma_m))\right)\right] e(P_i,t_m) + \frac{\tau_m^2}{2}\dot{G}(U(P_i,\xi)) + \tau_m \cdot \sum_{j=1}^{m}\left[\frac{p\tau_{j-1}^2}{2}U^{p-1}(P_i,\varsigma_{j-1}) + \tau_{j-1}\cdot L(\rho(P_i,\varsigma_{j-1}))e(P_i,t_{j-1})\right]; \tag{3.9}$$

2. *for $c = 1$:*

$$e(P_i,t_{m+1}) = e(P_i,t_m) + \tau_m\left[\sum_{k=1}^{N} D_k^2 + \tau_m L(\rho)\right] e(P_i,t_{m+1}) - \frac{\tau_m^2}{2}\dot{G}(U(P_i,\xi)) + \tau_m \cdot \sum_{j=1}^{m}\left[-\frac{p\tau_{j-1}^2}{2}U^{p-1}(P_i,\varsigma_{j-1}) + \tau_{j-1}\cdot L(\rho(P_i,\varsigma_{j-1})e(P_i,t_{j-1})\right]. \tag{3.10}$$

Proof Subtracting (3.8) from (3.2), we get the error equations in form of

$$\begin{aligned} e(P_i,t_{m+1}) &= e(P_i,t_m) \\ &+\tau_m\left(\sum_{k=1}^{N} D_k^2 + \tau_m\cdot L(\rho(P_i,\varsigma_m))\right)(e\,(P_i,t_m) + c\,(e\,(P_i,t_{m+1}) - e\,(P_i,t_m))) \\ &+\int_{t_{m+1}}^{t_m}\left[U^p(P_i,s) - U^p(P_i,t_m + c\tau_m) + \sum_{k=1}^{N} D_k^2\,[U(P_i,s) - U(P_i,t_m + c\tau_m)]\right]ds \\ &+\tau_m\cdot\sum_{j=1}^{m}\int_{t_{j-1}}^{t_j}\left[U^p(P_i,s) - U^p(P_i,t_{j-1} + c\tau_{j-1})\right]ds \\ &+\tau_m\cdot\sum_{j=1}^{m}\int_{t_{j-1}}^{t_j}\left[U^p(P_i,t_{j-1} + c\tau_{j-1}) - V^p(P_i,t_{j-1} + c\tau_{j-1})\right]ds. \end{aligned} \tag{3.11}$$

Using the trapezoidal rule respectively for $c = 0$ and $c = 1$ at at each time level for the time discretization of the Eq. (3.11), the results (3.9) and (3.10) can be obtained. The proof is completed. □

From the definition of $G(P_i,\xi)$, and taking the derivative of the Eq. (1.1) with respect to t, we get $\dot{G}(P_i,\xi)) = \ddot{U}(P_i,\xi)$ in (3.9) and (3.10). From this observation and Lemma 3.1, we have

Lemma 3.2 *If $\ddot{U}(P_i,\xi) \geq 0$, we have*

1. *for $c = 0$:*

$$e(P_i,t_1) = 0,\ e(P_i,t_{m+1}) \geq 0 \ \text{ if } \tau_m\left(-\tfrac{2N}{h^2} + \tau_m L(\rho)\right) + 1 > 0; \tag{3.12}$$

2. *for* $c = 1$*:*

$$e(P_i, t_1) = 0,\ e(P_i, t_{m+1}) \le 0 \ \text{if}\ \tau_m^2 L(\rho) < 1. \tag{3.13}$$

Proof For the case $c = 0$, the conclusion (3.12) is simple by noting that $U^{p-1}(P_i, \varsigma_{j-1})$ and $L(\rho(P_i, \varsigma_{j-1})$ are positive for all $j \le m$ in Eq. (3.9). For the case $c = 1$, the mathematical induction is used. We first consider the time t_2. From Eq. (3.10), setting $\ddot{U}(P_i, \xi) \ge 0$ and $e(P_i, t_1) = 0$, if $\tau_m^2 L(\rho) < 1$ with $m = 1$, one obtains a strictly diagonally dominant matrix A and a vector $b < 0$ such that

$$Ae = b. \tag{3.14}$$

By using Gauss-Seidel iteration to solve Eq. (3.14) with initial approximation $e^0 = 0$, we obtain $e(P_i, t_2) < 0$. By induction, we have $e(P_i, t_j) \le 0$ with $j = 1, \cdots, m$ and for all i. For time level $m+1$, we can have $b \le 0$ by noting that $U^{p-1}(P_i, \varsigma_{j-1})$ and $L(\rho(P_i, \varsigma_{j-1})$ are positive for all $j \le m$ in Eq. (3.10). Repeating the above process, we arrive at the conclusion (3.13). The proof is completed. □

Assume that $\tilde{T}_b < \infty$ is the numerical blow-up time for problem (3.1) for given threshold M, and let $\tilde{T}_b^{FE}$ and $\tilde{T}_b^{BE}$ denote, respectively, the numerical blow-up times for the discretized problem (3.2) corresponding to the forward Euler method and the backward Euler method. Using the result of Lemma 3.2, we obtain

Theorem 3.3 *Assume that, for fixed threshold* $M \gg 1$*, the solution* $U(t)$ *is a monotonically increasing and concave function. Then*

$$\tilde{T}_b^{BE}(M) < \tilde{T}_b(M) < \tilde{T}_b^{FE}(M). \tag{3.15}$$

4 Numerical Examples

Here the performance of our numerical methods are investigated in various aspects. We first investigate the efficiency of the nonlinear ABCs in Sect. 4.1, and the dependence of the numerical blow-up time on the length of the computational interval in Sect. 4.2. Then we give a detailed comparison of the numerical blow-up times corresponding to the backward Euler, the forward Euler, and the Crank-Nicolson schemes, with different initial values u_0 and different kernels in Sect. 4.3, and also list the dependence of the approximate blow-up time on the refinement of the spatial mesh. Finally, we give a two-dimensional case to show the blow-up solutions on a circular domain.

For prescribed blow-up thresholds M_i, let $\tilde{T}_b^{CN}(M_i)$, $\tilde{T}_b^{FE}(M_i)$ and $\tilde{T}_b^{BE}(M_i)$ denote the blow-up time for the Crank-Nicolson scheme, the forward Euler and the backward Euler schemes, respectively. Let $\mu(M_i)$ denote the number of time steps needed to reach the blow-up threshold M_i. For one-dimensional case, the following initial functions are used:

(a) Single Gaussian function: $u_0(x) = g(x)\exp(-x^2)$; and
(b) Double Gaussian function: $u_0(x) = g_1(x)\exp\left(-(x-c_1)^2\right) + g_2(x)\exp\left(-(x+c_2)^2\right)$,

where c_i are positive constants, and $g(x)$, $g_i(x)$ $(i = 1, 2)$ are given functions. To adaptively choose the time steps τ_m (following (3.7)), we set the corresponding parameters $\alpha = 5$, $d = 1$, $s_0 = 1$, $p = 3$, and $h = (x_r - x_l)/r$. For the (explicit) forward Euler method $(c = 0)$, we set $c = 1$ in (3.7). The thresholds $M_1 = 10^5$ and $M_2 = 10^6$ are given.

Table 2 Comparison of "exact solution" with numerical solution at $x_l = 1.5$

	Exact value	$h = 0.01$	$h = 0.005$	$h = 0.0025$
$s_0 = 1$	108.49493	108.47761	108.47934	108.48030
$s_0 = 2$	108.49493	108.46924	108.47205	108.47350
$s_0 = 3$	108.49493	108.49982	108.49822	108.49763
$s_0 = 4$	108.49493	108.52572	108.52067	108.51848

Table 3 The L_1-errors and convergence rates for $s_0 = 2.5$

	$\Delta x = 0.02$	Order	$\Delta x = 0.01$	Order	$\Delta x = 0.05$	Order
$x_l = -2.0$	2.095e-6	–	5.432e-7	1.94	1.336e-7	1.82
$x_l = -1.5$	2.102e-6	–	5.346e-7	1.98	1.412e-7	1.92

4.1 Efficiency of the Constructed ABCs

To see the influence of s_0 arising in the nonlinear ABCs (2.28), we text the single Gaussian initial function with $g(x) = 10^3$, and the Crank-Nicolson scheme taking $\Delta t = 10^{-6}$, $T = 0.001714$, the kernel $\kappa(t) \equiv 1$. The numerical values at the artificial boundary x_l are compared with the corresponding "exact" solutions, which are computed in a larger interval and with smaller mesh sizes. Table 2 shows the "exact value" and the numerical values at x_l with different parameters s_0 in the computational interval $[-1.5, 1.5]$. Another manner to measure the performance of s_0 is to see the convergence order of the L_1-error, defined by $L_1(t) = ||u(\cdot, t) - u_h(\cdot, t)||_1 / ||u(\cdot, t)||_1$. Table 3 shows the L_1-errors and the almost second convergence order with $s_0 = 2.5$.

From Tables 2, 3, one sees that the numerical solutions approximate the "exact" solution well, and the choice of parameter s_0 has a small effect on the performance of ABCs, but is not sensitive. The parameter s_0 can be chosen in a larger interval such that the boundary conditions work well, and the relative error is less than $1‰$.

4.2 Dependence of the Numerical Blow-Up Time on the Length of the Computational Domain

We now discuss the influence of the length of the computational interval on the blow-up time $T_b^{CN}(M_i)$. In the calculations, we use the fixed spatial mesh size $h = 0.005$, and $s_0 = 3.0$, while the adaptive time steps τ_m are given as in (3.7) and the same temporal step size strategy will be used in the following sections. Tables 4 and 5 show the different lengths of the computational domains, and a sample of computed blow-up times $\tilde{T}_b^{CN}(M_1)$, respectively, for the single Gaussian initial function and kernel $\kappa(t) \equiv 1$ with $g(x) = 100$, the double Gaussian initial function and kernel $\kappa(t) \equiv 1$ with the values $g_1(x) = g_2(x) = 50$, $c_1 = c_2 = 2$. Tables 4, 5 reveal that the blow-up times are insensitive to the choice of the computational intervals. Thus, in practical computations we may produce the numerical blow-up times by different schemes in a smaller computational domain.

Table 4 Dependence of interval length with single Gaussian function: $g(x) = 100$

$[x_l, x_r]$	[-3,3]	[-4,4]	[-5,5]	[-6,6]
$\tilde{T}_b^{CN}(M_1)$	0.0189301117	0.0189301117	0.0189301117	0.0189301117
$\tilde{T}_b^{CN}(M_2)$	0.0189430511	0.0189430511	0.0189430511	0.0189430511

Table 5 Dependence of interval length with double Gaussian function: $g_1(x) = g_2(x) = 50$

$[x_l, x_r]$	[-4,4]	[-4.5,4.5]	[-5,5]	[-5.5,5.5]
$\tilde{T}_b^{CN}(M_1)$	0.03871330171	0.03871330171	0.03871330171	0.03871330171
$\tilde{T}_b^{CN}(M_2)$	0.03872620309	0.03872620309	0.03872620309	0.03872620309

Table 6 Blow-up times and number of time steps with $g(x) = 100x^2$ and $\kappa \equiv 1$

I	$\tilde{T}_b^{BE}(M_1)$	$\mu_b^{BE}(M_1)$	$\tilde{T}_b^{CN}(M_1)$	$\mu_b^{CN}(M_1)$	$\tilde{T}_b^{FE}(M_1)$	$\mu_b^{FE}(M_1)$
400	0.05591669945	722	0.05610484874	731	0.05650402604	1481
800	0.05618198563	1468	0.05627259559	1427	0.05642949838	2965
1600	0.05630743234	4438	0.05634157735	4446	0.05639713663	8094

Table 7 Blow-up times and number of time steps with double Gaussian function and $\kappa \equiv 1$

I	$\tilde{T}_b^{BE}(M_1)$	$\mu_b^{BE}(M_1)$	$\tilde{T}_b^{CN}(M_1)$	$\mu_b^{CN}(M_1)$	$\tilde{T}_b^{FE}(M_1)$	$\mu_b^{FE}(M_1)$
400	0.03834299744	605	0.03855922372	710	0.03885542352	1244
800	0.03856647326	1116	0.03864258422	1123	0.03879141428	2262
1600	0.03867749070	3157	0.03870720439	3163	0.03876119954	6342

4.3 Approximation of the Blow-Up Time by Using Different Schemes

Using the Crank-Nicolson scheme, backward and forward Euler schemes, we investigate the numerical blow-up times for the given blow up threshold M_1 in the computational interval $[-4.5, 4.5]$. A selection of numerical blow-up times for different initial functions and different kernels is presented below. Table 6 shows the computed blow-up times $\tilde{T}_b(M_1)$ and the number of time steps $\mu(M_1)$ for single Gaussian function and kernel $\kappa(t) \equiv 1$ with $g(x) = 100x^2$. Table 7 shows for double Gaussian function and kernel $\kappa(t) \equiv 1$ with $g_1(x) = g_2(x) = 50$, $c_1 = c_2 = 2$. Table 8 shows for Double Gaussian function and kernel $\kappa(t) = \exp(-t)$ with $g_1(x) = g_2(x) = 100$, $c_1 = c_2 = 2$. From Tables 6, 7 and 8 for the given spatial mesh size h, one sees that the relationship between the blow-up times satisfies

$$\tilde{T}_b^{BE}(M_1) < \tilde{T}_b^{CN}(M_1) < \tilde{T}_b^{FE}(M_1).$$

Table 8 Blow-up times and number of time steps with double Gaussian function and $\kappa(t) = \exp(-t)$

I	$\tilde{T}_b^{BE}(M_1)$	$\mu_b^{BE}(M_1)$	$\tilde{T}_b^{CN}(M_1)$	$\mu_b^{CN}(M_1)$	$\tilde{T}_b^{FE}(M_1)$	$\mu_b^{FE}(M_1)$
400	0.01853901668	224	0.01866742607	174	0.01925028503	372
800	0.01876748007	465	0.01886564194	473	0.01907341481	1270
1600	0.01891034620	1666	0.01894281143	1672	0.01900937344	4867

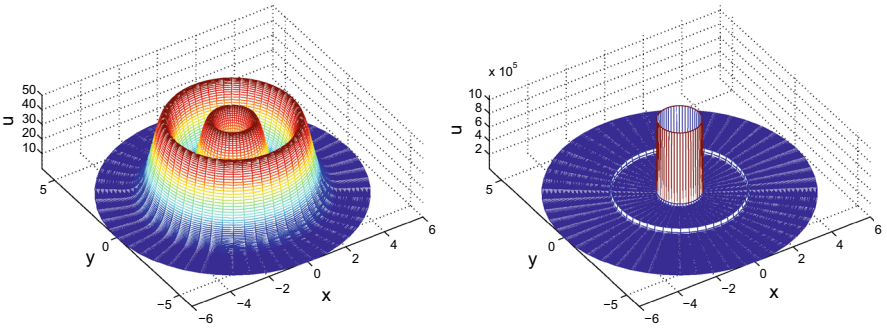

Fig. 1 Initial values and the numerical line blow-up solution

4.4 One Example for Two-Dimensional Spatial Domain

We finally illustrate the effectiveness of our ABCs on a circular artificial boundaries. The initial value as shown in Fig. 1, is given by

$$u_0(r, \theta) = 50\exp(-2(r-2)^2) + 50\exp(-(r-3)^2).$$

The computational parameters are chosen as $p = 3, I = 200, J = 150, R = 6, s_0 = 10$ with the threshold $M_2 = 10^6$ at $T = 0.019489$ and the kernel $\kappa(t) \equiv 1$. Figure 1 shows the line blow-up for the given initial value, and the parameters in the calculation can be found in Table 1.

5 Future Work and Open Problem

The various simulations have demonstrated that our numerical methods are stable, efficient and accuracy. In the future we will study the derivation of error bounds and order results for the numerical blow-up time. Furthermore, if it is not known a priori whether or not the exact solution blows up in finite time, can we devise an approximation scheme that allows for the *numerical* detection of finite-time blow-up (or the existence of a global solution)? On the other hand, we will consider to introduce the moving mesh method for the simulations in the practical applications and construct a fast evaluation of the nonlinear term when the kernel is Abel-type $t^{\alpha-1}$ using the method proposed in [12].

Acknowledgments HB and TT are supported by Hong Kong Research Council GRF Grants and Hong Kong Baptist University FRG grants. JZ is partially supported by the start-up fund of Beijing CSRC and NSFC 91430216, and thanks Dr. Zhan-wen Yang for valuable discussions. The authors thank referees for their carefully reading of the original version of the paper and for their valuable suggestions.

References

1. Acosta, G., Duran, R.G., Rossi, J.D.: An adaptive time step procdure for a parabolic problem with blow-up. Computing **68**, 343–373 (2002)
2. Bandle, C., Brunner, H.: Numerical analysis of semilinear parabolic problems with blow-up solutions. Rev. Real Acad. Cienc. Exact. Fis. Natur. Madrid **88**, 203–222 (1994)
3. Bandle, C., Brunner, H.: Blow-up in diffusion equations: a survey. J. Comput. Appl. Math. **97**, 2–22 (1998)
4. Bellout, H.: Blow-up solutions of parabolic equations with nonlinear memory. J. Differ. Equ. **70**, 42–68 (1987)
5. 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)
6. Chen, Y.G.: Asymptotic behavior of blowing-up solutions for finite difference analogue of $u_t = u_{xx} + u^{1+\alpha}$. J. Fac. Sci. Uni. Tokyo Sect. IA Math. **33**, 541–574 (1986)
7. Fakhar-Izadi, F., Dehghan, M.: The spectral methods for parabolic Volterra integro-differential equations. J. Comput. Appl. Math. **235**, 4032–4044 (2011)
8. Groisman, P.: Totally discrete explicit and semi-implicit Euler methods for a blow-up problem in several space dimensions. Computing **76**, 325–352 (2006)
9. Han, H., Zhu, L., Brunner, H., Ma, J.T.: The numerical solution of parabolic Volterra integro-differential equations on unbounded spatial domains. Appl. Numer. Math. **55**, 83–99 (2005)
10. Han, H., Zhu, L., Brunner, H., Ma, J.T.: Artificial boundary conditions for parabolic Volterra integro-differential equations on unbounded two-dimensional domains. J. Comput. Appl. Math. **197**, 406–420 (2006)
11. Han, H., Wu, X.: Artificial Boundary Method. Springer, Heidelberg and Tsinghua University Press, Beijing (2013)
12. Jiang, S., et al.: Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations, Submitted
13. Khan, L.A., Liu, P.L.F.: Numerical analyses of operator-splitting algorithms for the two-dimensional advection-diffusion equation. Comput. Method Appl. Mech. Eng. **152**, 337–359 (1998)
14. Kyza, I., Makridakis, C.: Analysis of time discrete approximations of blow-up solutions of semilinear parabolic equations. SIAM J. Numer. Anal. **49**, 405–426 (2011)
15. Li, Y., Xie, C.: Blow-up for semilinear parabolic equations with nonlinear memory. Z. Angew. Math. Phys. **55**, 15–27 (2004)
16. Liang, K., Zhu, H.: Moving collocation method for a reaction diffusion equation with a traveling heat source. Numer. Methods PDEs. **29**, 2047–2060 (2013)
17. Liu, D., Mu, C.: Blow-up analysis for a semilinear parabolic equation with nonlinear memory and nonlocal nonlinear boundary condition. J. Qual. Theory Differ. Equ. **51**, 1–17 (2010)
18. Nakagawa, T.: Blowing up of a finite difference solution to $u_t = u_{xx} + u^2$. Appl. Math. Optim. **2**, 337–350 (1975)
19. 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)
20. Qiang, H., Liang, K.: Moving mesh method for problems with blow-up on unbounded domains. Numer. Algorithms **59**, 63–77 (2012)
21. Sun, Z., Wu, X., Zhang, J., Wang, D.: A linearized difference scheme for semilinear parabolic equations with nonlinear absorbing boundary conditions. Appl. Math. Comput. **218**, 5187–5201 (2012)
22. Souplet, P.: Monotonity of solutions and blow-up for semilinear parabolic equations with nonlinear memory. Z. Argew. Math. Phys. **55**, 28–31 (2004)
23. Souplet, P.: Blow-up in nonlocal reaction-diffusion equations. SIAM J. Math. Anal. **29**, 1301–1334 (1998)
24. 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)
25. Strang, G.: On the construction and comparison of difference schemes. SIAM J. Numer. Anal. **5**, 506–517 (1968)
26. Stuart, A., Floater, M.: On the computation of blow-up. Eur. J. Appl. Math. **1**, 47–71 (1990)
27. Tsynkov, S.V.: Numerical solution of problems on unbounded domain: a review. Appl. Numer. Math. **27**, 465–532 (1998)
28. Wu, X., Zhang, J.: Higher-order local absorbing boundary conditions for heat equations in unbounded domain. J. Comput. Math. **29**, 74–90 (2011)
29. 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)

30. Yang, Z., et al.: Blowup and global existence of solutions to semilinear Volterra diffusion equations, Preprint
31. Yanik, E.G., Fairweather, G.: Finite element methods for parabolic and hyperbolic partial integro-differential equations. Nonlinear Anal. **12**, 785–809 (1988)
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. Zhang, J., Han, H., Brunner, H.: Numerical blow-up of semilinear parabolic PDEs on unbounded domains in $\mathbb{R}^2$. J. Sci. Comput. **49**, 367–382 (2011)
35. Zheng, C.: Exact nonreflecting boundary conditions for one-dimensional cubic nonlinear Schrödinger equations. J. Comput. Phys. **215**, 552–565 (2006)
36. Zheng, C.: Numerical solution to the sine-Gordon equation defined on the whole real axis. SIAM J. Sci. Comput. **29**, 2494–2506 (2007)