Computational solution of blow-up problems for semilinear parabolic PDEs on unbounded domains

Brunner, Hermann; Wu, Xiaonan; Zhang, Jiwei

Published in:
SIAM Journal on Scientific Computing

DOI:
10.1137/090761367

Published: 01/01/2010

Document Version:
Publisher's PDF, also known as Version of record

SIAM J. SCI. COMPUT.
Vol. 31, No. 6, pp. 4478–4496

COMPUTATIONAL SOLUTION OF BLOW-UP PROBLEMS FOR SEMILINEAR PARABOLIC PDEs ON UNBOUNDED DOMAINS*

HERMANN BRUNNER†, XIAONAN WU‡, AND JIWEI ZHANG§

Abstract. This paper is concerned with the numerical solution of semilinear parabolic PDEs on unbounded spatial domains whose solutions blow up in finite time. The focus of the presentation is on the derivation of the nonlinear absorbing boundary conditions for one-dimensional and two-dimensional computational domains and on a simple but efficient adaptive time-stepping scheme. The theoretical results are illustrated by a broad range of numerical examples, including problems with multiple blow-up points.

Key words. semilinear partial differential equations, unbounded spatial domains, finite-time blow-up, local absorbing boundary conditions, finite difference spatial discretization, adaptive time stepping

AMS subject classifications. 65M06, 65L10, 35Q53, 35Q51

DOI. 10.1137/090761367

1. Introduction.

We consider the numerical solution of the semilinear parabolic PDE for $u = u(x,t)$,

$$u_t - \Delta u = u^p, \quad x \in \Omega, \quad t > 0 \quad (p > 1), \tag{1.1}$$

subject to the initial condition $u(x,0) = u_0(x)$. The spatial domain Ω is given either by $\Omega = \mathcal{R}^d$ $(d = 1, 2)$, and we assume that the initial function u_0 has compact support.

While the theory of finite-time blow-up for such problems is well understood (see, for example, the papers by Fujita [8], Levine [14], Bandle and Brunner [5], the monograph by Samarskii et al. [18], and their references), the numerical analysis of blow-up on unbounded spatial domains has so far received little attention. The main difficulty is the selection of a suitable bounded computational domain and, above all, the derivation of appropriate nonlinear absorbing BCs for the semilinear PDE (1.1). For the bounded domain case, there are numerous studies on the numerical simulation of blow-up. Nakagawa [15] and Nakagawa and Ushijima [16] first studied this subject by using the finite difference and finite element method on a uniform spatial mesh. A mesh refinement strategy is presented in [17]. A moving mesh method (dynamic regridding) has also been proposed (see, for example, [7]): here, the spatial mesh is generated by appropriately chosen moving mesh PDEs. Information on more

*Received by the editors June 5, 2009; accepted for publication (in revised form) August 17, 2009; published electronically January 20, 2010.
http://www.siam.org/journals/sisc/31-6/76136.html

†Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL, A1C 5S7, Canada. Current address: Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR, People's Republic of China (hbrunner@math.hkbu.edu.hk). The work of this author was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and by the Hong Kong Research Grants Council (RGC).

‡Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong, People's Republic of China (xwu@math.hkbu.edu.hk). The work of this author was supported partially by RGC and Faculty Research Grant of Hong Kong Baptist University.

§Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371, Singapore (jwzhang@math.hkbu.edu.hk). The work of this author was supported partially by RGC and Faculty Research Grant of Hong Kong Baptist University.

recent work and additional references may be found in the review paper [12] and in [1, 6, 19, 20, 21].

The essential difficulties in the numerical solution of the model problem (1.1) consist of three parts: the nonlinearity, the unboundedness, and the multidimensionality of the physical domain. To deal with the unboundedness, one of the most powerful approaches is the use of artificial boundary conditions (ABCs) for an appropriate bounded computational domain; compare, for example, [11, 10]. (The use of (local) ABCs is computationally much more efficient, especially in problems with finite-time blow-up and nonlocal initial functions, than employing homogeneous BCs, since in the former case much smaller computational domains can be chosen.) In general, the ABCs can be grouped into two categories: implicit BCs and explicit BCs (including global, or nonlocal, ABCs, local ABCs, and discrete ABCs). For the multidimensional case, it is hard to find suitable ABCs at the corners of a rectangle. In recent years, there has been some new progress on ABC methods for nonlinear PDEs, such as the linearized or reduced method [9, 23, 25, 2]. In the inverse scattering approach, the exact ABCs are derived for the one-dimensional cubic nonlinear Schrödinger equation [28] and the sine-Gordon equation [29].

Recently, Zhang, Xu, and Wu [26, 27] proposed a novel unified approach to the design of efficient local ABCs for time-dependent nonlinear Schrödinger equations. The idea underlying the unified approach is to distinguish between incoming and outgoing waves along the boundaries for the linear kinetic subproblem and to approximate the corresponding linear operator by using a "one-way operator" (to make the wave outgoing); then unite the approximate operator and the potential energy subproblem to yield nonlinear BCs. The resulting ABCs are shown to be efficient and stable.

The objective of this paper is to extend the spirit of this methodology to the design of analogous local ABCs for semilinear parabolic PDEs on unbounded spatial domains, in order to derive an efficient numerical scheme for solving problems in which the solutions blow up in finite time.

We address this problem in section 2 of our paper: there we introduce our unified approach for deriving the nonlinear absorbing BCs for one-dimensional and two-dimensional (2D) spatial computational domains. The numerical scheme for approximating the solution of the resulting initial–boundary-value problem for (1.1) on the chosen spatial domains will be presented in section 3; the adaptive time-stepping scheme is an extension of an approach by Bandle and Brunner [4]. In section 4 we describe some geometrical aspects of the numerical (time-stepping) scheme, and then we use section 5 to illustrate various aspects: efficiency of the constructed ABCs; sensitivity of the numerical blow-up time with respect to the size of the computational domain; dependence of numerical blow-up time on the spatial mesh and the chosen (collocation-based) time-stepping method; and single-point versus multiple-point blow-up of our approach. Future work and some open problems are described in the concluding section 6.

2. Nonlinear absorbing BCs. In this section, we present a novel unified approach (using ideas of [26, 27]) to derive the local absorbing BCs (LABCs) for the nonlinear blow-up equation (1.1) on suitably chosen computational domains Ω_i defined below. This unified approach is based on the well-known time-splitting method (or split-step method) of [24]. We first approximate the Laplace operator by a one-directional operator and then combine the approximation operator with the reaction term, u^p, to obtain the LABCs. These appear in the form of a nonlinear expression and are shown to absorb the heat flow perfectly.

2.1. Unified approach for the one-dimensional blow-up model. To understand the philosophy of the unified approach, let us rewrite (1.1) in the operator form

$$u_t(x,t) = [\hat{T} + \hat{V}]u(x,t), \tag{2.1}$$

where $\hat{T}$ represents the linear differential operator and $\hat{V}$ is the nonlinear operator that governs the effects of the nonlinearity. These operators are given by

$$\hat{T} := \partial_x^2 \quad \text{and} \quad \hat{V} := u^{p-1}. \tag{2.2}$$

In a small time interval from t to $t+\tau$ with $\tau > 0$ we use, in analogy to the widely used Strang splitting [22],

$$u(x,t+\tau) \approx e^{\hat{T}\tau/2} e^{\hat{V}\tau} e^{\hat{T}\tau/2} u(x,t), \tag{2.3}$$

the operator splitting

$$u(x,t+\tau) \approx e^{[\hat{T}+\hat{V}]\tau} u(x,t), \tag{2.4}$$

which, by the Baker–Campbell–Hausdorff formula, is closely related (but not identical, owing to the noncommutativity of the operators $\hat{T}$ and $\hat{V}$ [13]) to the splitting

$$u(x,t+\tau) \approx e^{\hat{T}\tau} e^{\hat{V}\tau} u(x,t), \tag{2.5}$$

with first-order error $\mathcal{O}(\tau)$. These splittings mean that during a small time step τ the approximation carries out the diffusion step and nonlinearity step separately, reflecting the basic idea of the well-known time-splitting method which is often very effective in the numerical solution of semilinear PDEs. Thus, the key idea underlying this unified approach is to approximate the linear operator directly and then to combine the results with the nonlinear operator.

We now focus on the derivation of the approximation of the operator $\hat{T}$. We first consider the heat problem on the exterior domain $\Omega_e := (-\infty, x_l] \cup [x_r, +\infty)$ (or, equivalently, $\Omega_e := \mathcal{R}^d \setminus \Omega_i$, where Ω_i denotes the suitably chosen computational domain). Here, the solution $u(x,t)$ satisfies

$$\begin{cases} u_t - u_{xx} = 0, & x \in \Omega_e, \\ u(x,0) = 0, & x \in \Omega_e, \\ u|_{x=x_r} = u(x_r,t), & \\ u|_{x=x_l} = u(x_l,t), & \\ u \to 0 & \text{as} \quad |\mathbf{x}| \to \infty. \end{cases} \tag{2.6}$$

If we apply the Laplace transformation with respect to t to problem (2.6), we arrive at

$$s\tilde{u} - \tilde{u}_{xx} = 0, \tag{2.7}$$

where the Laplace transformation is given by

$$\tilde{u} = \widetilde{u}(x,s) := \int_0^\infty e^{-st} u(x,t)dt. \tag{2.8}$$

The equation (2.7) is homogeneous and has two linearly independent eigensolutions, $\widetilde{u}^{(1)}(x) = e^{-x\sqrt{s}}$ and $\widetilde{u}^{(2)}(x) = e^{x\sqrt{s}}$. One can see that BCs in (2.6) are satisfied if and

only if the eigensolutions $\widetilde{u}^{(1)}(x)$ do not contribute to the solution $\widetilde{u}(x)$ of problem (2.7) in the semi-infinite interval $(-\infty, x_l])$, and $\widetilde{u}^{(2)}(x)$ does not contribute in the interval $[x_r, +\infty)$.

We now obtain the following relations on the artificial boundaries:

$$\partial_x \widetilde{u} \pm \sqrt{s}\widetilde{u} = 0. \tag{2.9}$$

The plus sign in "±" corresponds to the right BC, and the minus sign corresponds to the left one. By using $\sqrt{s} = \frac{s}{\sqrt{s}}$ and the definition of the inverse Laplace transformation,

$$\mathcal{L}^{-1}\left\{\frac{1}{\sqrt{s+\alpha}}\right\} = \frac{1}{\sqrt{\pi t}} e^{-\alpha t}, \tag{2.10}$$

formula (2.9) yields the exact ABCs at the artificial boundaries, as follows:

$$\partial_x \mathrm{u} \pm \sqrt{\frac{1}{\pi}} \int_0^t \frac{\partial_\tau \mathrm{u}}{\sqrt{t-\tau}} d\tau = 0. \tag{2.11}$$

The equation (2.11) is global and is not coupled with the nonlinear term in (2.4).

Motivated by the known advantages of LABCs, our aim is now to construct the LABCs for our problem. In formula (2.9), we set $z = s$ and expand the irrational function $\sqrt{z}$ by using the Padé approximation (with respect to a given point z_0)

$$g(z) = \sqrt{z_0}\sqrt{1-\left(1-\frac{z}{z_0}\right)} \approx \sqrt{z_0} - \sqrt{z_0}\sum_{k=1}^{N} \frac{b_k\left(1-\frac{z}{z_0}\right)}{1-a_k\left(1-\frac{z}{z_0}\right)} = g_N(z),\ |z-z_0| < |z_0|, \tag{2.12}$$

where

$$a_k = \cos^2\left(\frac{k\pi}{2N+1}\right), \quad b_k = \frac{2}{2N+1}\sin^2\left(\frac{k\pi}{2N+1}\right), \quad k = 1, 2, \ldots, N.$$

In general, the larger N, the closer is the approximate curve to the exact one. More precisely, the important approximation property (cf. [3] and its references) is

$$||g - g_N||_{L^\infty([0,2z_0])} = \frac{1}{N}. \tag{2.13}$$

We substitute the approximation (2.12) in (2.9) to obtain

$$\partial_x \widetilde{u} \pm \sqrt{z_0}\left(1 - \sum_{k=1}^{N} \frac{b_k\left(1-\frac{z}{z_0}\right)}{1-a_k\left(1-\frac{z}{z_0}\right)}\right)\tilde{u} = 0. \tag{2.14}$$

As a simple case, we choose $N = 1$ and substitute $z = s$ and $z_0 = s_0$ to obtain

$$(1-a_1)s_0\partial_x\widetilde{u} + a_1 s\partial_x\widetilde{u} \pm (1-a_1-b_1)s_0\sqrt{s_0}\widetilde{u} \pm (a_1+b_1)\sqrt{s_0}s\widetilde{u} = 0. \tag{2.15}$$

The parameter s_0 is chosen according to the underlying physical meaning, representing the wave number of the wave imposed on the artificial boundary in the wave equations. Applying the inverse Laplace transformation, we find the third-order ABCs as follows:

$$3s_0\partial_x u + \partial_x\partial_t u \pm s_0\sqrt{s_0}u \pm 3\sqrt{s_0}\partial_t u = 0. \tag{2.16}$$

From formulae (2.16), the linear operator $\hat{T}$ can be approximated by one-way operators

$$\hat{T} \approx \hat{\mathcal{T}}^{(3)} = -\left(\partial_x \pm 3\sqrt{s_0}\right)^{-1}\left(3s_0\partial_x \pm s_0\sqrt{s_0}\right), \tag{2.17}$$

which correspond to third-order local BCs in (2.16). The basic idea underlying this unified approach is to substitute $\hat{\mathcal{T}}^{(3)}$ for $\hat{T}$ in (2.4), in order to obtain, for small $\tau > 0$, the approximating operators to the solution operator

$$e^{[\hat{T}+\hat{V}]\tau} \approx e^{[\hat{\mathcal{T}}^{(3)}+\hat{V}]\tau}. \tag{2.18}$$

This implies the one-directional equations,

$$\partial_t u(x,t) = \left[\mathcal{T}^{(3)} + \hat{V}\right] u(x,t). \tag{2.19}$$

They represent approximations to (2.1) at the artificial boundaries and are the required absorbing BCs. Concretely, the nonlinear absorbing BCs have the form

$$\pm 3\sqrt{s_0}u_t + 3s_0u_x + u_{xt} \pm s_0\sqrt{s_0}u - \left(\pm 3\sqrt{s_0}u^p + pu^{p-1}u_x\right) = 0. \tag{2.20}$$

It is convenient to express the BCs (2.20) in operator forms:

$$\mathcal{B}_-^{(3)}(u, u_t, u_x, u_{xt}) = 0, \tag{2.21}$$

$$\mathcal{B}_+^{(3)}(u, u_t, u_x, u_{xt}) = 0, \tag{2.22}$$

where $\mathcal{B}_-$ and $\mathcal{B}_+$, respectively, represent the left and right BCs; that is, the minus sign in "$\pm$" is taken for $\mathcal{B}_-$ and the plus sign for $\mathcal{B}_+$. The initial-value problem in the open domain of (1.1) restricted to the truncated interval $[x_l, x_r]$ is then approximated by an initial–boundary-value problem with LABCs:

$$u_t = u_{xx} + u^p \quad \text{for} \quad x \in [x_l, x_r], \tag{2.23}$$

$$u(x,0) = u_0(x) \quad \text{for} \quad x \in [x_l, x_r], \tag{2.24}$$

$$\mathcal{B}_-^{(3)}(u, u_t, u_x, u_{xt}) = 0, \tag{2.25}$$

$$\mathcal{B}_+^{(3)}(u, u_t, u_x, u_{xt}) = 0. \tag{2.26}$$

2.2. Unified approach for the 2D blow-up equation. We now consider the design of the LABCs for the 2D blow-up problem on a rectangle. In analogy to (2.1) the solution $u(x, y, t)$ satisfies

$$u_t = \hat{T}u + \hat{V}u \tag{2.27}$$

with $\hat{T}u := \partial_x^2 u + \partial_y^2 u$ and $\hat{V}u := u^{p-1}$. Using an argument similar to the one for the one-dimensional blow-up problem, we reduce (2.27) to an approximate problem in a bounded domain. The key step is to obtain the suitable approximations of the Laplace operator $\hat{T}$. Then we consider the 2D heat problem on the exterior domain Ω_e:

$$u_t = \hat{T}u, \quad \mathbf{x} \in \Omega_e. \tag{2.28}$$

Applying the Fourier transform with respect to x and y and the Laplace transform with respect to t, the resulting dispersion relation becomes

$$s + \xi^2 + \eta^2 = 0, \tag{2.29}$$

where the Fourier transform is

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

while the Laplace transform is the same as in (2.8). Solving (2.29), adopting the same strategy as in the one-dimensional case, and allowing only outgoing waves, we have the following dispersion relations on the east and west artificial boundaries:

$$-i\xi \pm \sqrt{s + \eta^2} = 0. \tag{2.30}$$

Here, the plus sign in "±" stands for the positive direction and the minus sign for the negative direction. Setting $z = \eta^2 + s$, expanding $\sqrt{z}$ by using formula (2.12) with $\mathrm{N} = 1$, substituting the approximation in (2.30), and solving the resulting algebraic equation, we obtain

$$s = -\frac{-i\xi\eta^2 \pm 3\sqrt{\xi_0}\eta^2 - 3i\xi_0\xi - \sqrt{\xi_0}\xi_0}{-i\xi \pm 3\sqrt{\xi_0}}. \tag{2.31}$$

By the same argument, we can derive the analogous dispersion relation at the northern and southern boundaries; they are

$$s = -\frac{-i\xi^2\eta \pm 3\sqrt{\eta_0}\xi^2 - 3i\eta_0\eta - \sqrt{\eta_0}\eta_0}{-i\eta \pm 3\sqrt{\eta_0}}. \tag{2.32}$$

In fact, the algebraic equation (2.32) can be derived by exchanging ξ and η.

For the corners, we use the (1,1)-Padé approximation to expand ξ^2 and η^2 at the expansion point (ξ_0, η_0). At the northeast and the southwest corners, we have the algebraic identity

$$s = -\xi_0 \frac{-3i\xi \pm \sqrt{\xi_0}}{-i\xi \pm 3\sqrt{\xi_0}} - \eta_0 \frac{-3i\eta \pm \sqrt{\eta_0}}{-i\eta \pm 3\sqrt{\eta_0}}. \tag{2.33}$$

At northwest and southeast corners, the approximations are given, respectively, by

$$s = -\xi_0 \frac{-3i\xi - \sqrt{\xi_0}}{-i\xi - 3\sqrt{\xi_0}} - \eta_0 \frac{-3i\eta + \sqrt{\eta_0}}{-i\eta + 3\sqrt{\eta_0}}, \tag{2.34}$$

$$s = -\xi_0 \frac{-3i\xi + \sqrt{\xi_0}}{-i\xi + 3\sqrt{\xi_0}} - \eta_0 \frac{-3i\eta - \sqrt{\eta_0}}{-i\eta - 3\sqrt{\eta_0}}. \tag{2.35}$$

By utilizing the duality of $s \leftrightarrow \partial_t, -i\xi \leftrightarrow \partial_x$, and $-i\eta \leftrightarrow \partial_y$, the corresponding LABCs can then be obtained for the heat equation on the artificial boundaries.

The equations (2.31)–(2.35) are the approximation of the Laplace operator in Fourier space. By using the duality relation, the approximation of the Laplace operator $\hat{\mathcal{T}}^{(3)}$ can be obtained in physical space, resulting in the third-order local BCs. According to the unified approach method and in analogy to (2.18), the solution operator $\hat{T}$ is replaced by $\hat{\mathcal{T}}^{(3)}$. This leads to the 2D version of (2.19), namely, to the one-directional equation

$$u_t = \hat{\mathcal{T}}^{(3)} u + \hat{V} u, \quad \mathbf{x} \in \Omega_e. \tag{2.36}$$

By a simple calculation and using the duality relation, we have the eastern and western LABCs, namely,

$$\left(3\xi_0 - pu^{p-1}\right) u_x - u_{xyy} + u_{xt} \pm \sqrt{\xi_0}\left(\xi_0 u - 3u^p + 3u_t - 3u_{yy}\right) = 0, \tag{2.37}$$

as well as the northern and southern LABCs

$$\left(3\eta_0 - pu^{p-1}\right) u_y - u_{xxy} + u_{yt} \pm \sqrt{\eta_0}\left(\eta_0 u - 3u^p + 3u_t - 3\sqrt{\eta_0}u_{xx}\right) = 0. \tag{2.38}$$

The LABCs at the northeast and southwest corners are given by

$$\begin{aligned} &u_{xyt} + \left(3\xi_0 + 3\eta_0 - u^{p-1}\right) u_{xy} \pm 3\sqrt{\xi_0}u_{yt} \pm 3\sqrt{\eta_0}u_{xt} \pm \sqrt{\eta_0}\left(9\xi_0 + \eta_0 - 3u^{p-1}\right) u_x \\ &\quad + 9\sqrt{\xi_0\eta_0}u_t \pm \sqrt{\xi_0}\left(9\eta_0 + \xi_0 - 3u^{p-1}\right) u_y + 3\sqrt{\xi_0\eta_0}\left(\xi_0 + \eta_0 - 3u^{p-1}\right) u = 0, \end{aligned} \tag{2.39}$$

while the LABCs at the northwest and southeast corners are

$$\begin{aligned} &u_{xyt} + \left(3\xi_0 + 3\eta_0 - u^{p-1}\right) u_{xy} \pm 3\sqrt{\xi_0}u_{xt} \pm \sqrt{\xi_0}\left(9\eta_0 + \xi_0 - u^{p-1}\right) u_x - 9\sqrt{\xi_0\eta_0}u_t \\ &\quad - 3\sqrt{\xi_0\eta_0}\left(\xi_0 + \eta_0 - 3u^{p-1}\right) u = \pm 3\sqrt{\eta_0}u_{yt} \pm \sqrt{\eta_0}\left(9\xi_0 + \eta_0 - u^{p-1}\right) u_y. \end{aligned} \tag{2.40}$$

Note that we linearized the nonlinear term for LABCs at corners to avoid the appearance of u^{p-2}. Thus, in the 2D case the initial value problem (1.1) is reduced to an approximating initial–boundary-value problem on a rectangle. The absorbing BCs are given by (2.37)–(2.40) on the corresponding artificial boundaries.

We conclude this section with a brief remark regarding the *stability* of the reduced initial–boundary-value problem with the obtained LABCs on the bounded computational domains. The procedure for designing the nonlinear absorbing BCs reveals that the BCs annihilate only the energy arising in the truncated computational domain; it will not propagate energy into the interior domain to disrupt the true solution. Thus, the perturbation appearing in the artificial boundaries will have no effect on the interior solution, and hence the obtained LABCs are efficient and stable. In section 5, numerical tests are given to illustrate the efficiency and stability of these BCs.

3. Setting of the approximating problems. We cover the truncated computational domain by a rectangular grid with mesh diameter $h > 0$, parallel to the axes of the Cartesian coordinate system, and denote by δ_k the unit vector in the direction of the x_k axis. Let $\{P_i\}$ be the set of grid points and assume that $P_1, \ldots, P_r$ denote the grid points contained in the domain Ω_i. At such a grid point P_i we will replace the derivative $u_{x_k x_k}$ 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 (1.1) is then given by

$$\dot{U}(P_i, t) = \sum_{k=1}^{d} D_k^2 U(P_i, t) + f(U(P_i, t)) =: G(U(P_i, t)), \tag{3.1}$$

with $f(u) = u^p$. In the following, let $U(t)$ denote the vector whose components are $U(P_i, t)(i = 1, \ldots, r)$. It is approximated by (continuous) piecewise linear functions $V(t)$ in $C^0(0, T)$. Consider a (generally nonuniform, adaptively chosen) temporal grid

$\tau_m := t_{m+1} - t_m$ for $[0,T]:\ 0 = t_0 < t_1 < \cdots < t_M = T$, and let $c \in [0,1]$ be a given collocation parameter. Assume that the approximating piecewise linear polynomial $\mathbf{V}(t) := (V(P_1,t),\ldots,V(P_r,t))^T$ has its knots at $t_1,\ldots,t_{M-1}$, and on $[t_m,t_{m+1}]$ has the local representation

$$V(P_i,t) = V(P_i,t_m) + \frac{t-t_m}{\tau_m}(V(P_i,t_{m+1}) - V(P_i,t_m)). \tag{3.2}$$

It is determined by collocation with respect to the collocation points $t = t_m + c\tau_m$. Hence, the collocation equation reads

$$\dot{V}(P_i,t_m + c\tau_m) = \sum_{k=1}^{d} D_k^2 V(P_i,t_m + c\tau_m) + f(V(P_i,t_m + c\tau_m)) \tag{3.3}$$

$(m = 0,\ldots,M-1)$. For $P_i \in \Omega_i$, the collocation equation (3.3), together with (3.2), leads to the system of algebraic equations:

$$\begin{aligned} V(P_i,t_{m+1}) &= V(P_i,t_m) \\ &\quad + \tau_m \cdot \sum_{k=1}^{d} D_k^2(V(P_i,t_m) + c(V(P_i,t_{m+1}) - V(P_i,t_m))) \\ &\quad + \tau_m \cdot f(V(P_i,t_m) + c(V(P_i,t_{m+1}) - V(P_i,t_m))). \end{aligned} \tag{3.4}$$

The collocation parameters $c \in \{0,\ 1/2,\ 1\}$ lead to three specific time-stepping methods known in the literature, namely the continuous forward Euler method, the continuous implicit Crank–Nicolson method, and the continuous backward Euler method, respectively. It is obvious that the above systems of algebraic equations cannot be solved uniquely, since the number of equations is smaller than the number of unknowns. Thus, we focus on the discretization of the absorbing BCs (2.20) and (2.37)–(2.40). Denote by the operators $D_+,\ D_-$, and D_0 the forward, backward, and centered differences, respectively, and by $S_+,\ S_-$, and S_0 the forward, backward, and centered sums, respectively; for example,

$$D_+^x u_{i,j}^m = (u_{i+1,j}^m - u_{i,j}^m)/h, \quad S_+^t u_{i,j}^m = (u_{i,j}^{m+1} + u_{i,j}^m)/2.$$

Here, $u_{i,j}^m$ represents the approximation of the function u at the grid point (x_i,y_j,t^m), with $i = 0,1,\ldots,I$ and $j = 0,1,\ldots,J$. Take the east BC in (2.37) and the northeast corner in (2.40) as examples. Without loss of generality, the discretized forms with $c = 1/2$ (implicit Crank–Nicolson) are

$$\begin{aligned} &u_x = D_-^x S_+^t u_{I,j}^n, \quad u = S_-^x S_+^t u_{I,j}^m, \quad u_{yy} = S_-^x D_-^y D_+^y S_+^t u_{I,j}^m, \\ &u_t = S_-^x D_+^t u_{I,j}^m, \quad u_{xyy} = D_-^x D_+^y D_-^y S_+^t u_{I,j}^m, \quad u_{xt} = D_-^x D_+^t u_{I,j}^m, \end{aligned}$$

and the discretized forms in the corner BC (2.40) are given by

$$\begin{aligned} &u_x = D_-^x S_-^y S_+^t u_{I,J}^m, \quad u = S_-^x S_-^y S_+^t u_{I,J}^m, \quad u_y = S_-^x D_+^y S_+^t u_{I,j}^m, \\ &u_t = S_-^x S_-^y D_+^t u_{I,J}^m, \quad u_{xy} = D_-^x D_-^y S_+^t u_{I,J}^m, \quad u_{xt} = D_-^x S_-^y D_+^t u_{I,J}^m, \\ &u_{xyt} = D_-^x D_-^y D_+^t u_{I,J}^m, \quad u_{yt} = D_-^y S_-^x D_+^t u_{I,J}^m. \end{aligned}$$

Hence, the discretizations of the other boundary and corner conditions can be obtained in an analogous way. The above system of nonlinear equations with $c > 0$ is implicit,

and one needs an iterative strategy (modified Newton iteration or direct fixed-point iteration) to solve (3.4). For this purpose the time step $\tau_m = t_{m+1} - t_m$ must be chosen such that the mapping from $\mathcal{R}^r$ to $\mathcal{R}^r$ that defines the solution V^{m+1} is contractive. Following [4], let $\rho := \max_{0<i<r} |V(P_i, t_m)|$, define $L = L(M)$ to be the Lipschitz constant of u^p in $[0, M]$, and denote by the ball $B(\alpha\rho) := \{v \in \mathcal{R}^r : |v| < \alpha\rho\}$ for given $\alpha > 1$. In [4, 5] the time-step criterion is given by

$$\tau_m < \min\left\{\frac{(\alpha-1)\rho}{c(4d\alpha\rho/h^2 + f(\alpha\rho))}, \frac{1}{c(4d/h^2 + L(\alpha\rho))}\right\}, \tag{3.5}$$

which makes the above mapping from $B(\alpha\rho)$ to $B(\alpha\rho)$ contractive (at the time level $t = t_m$).

4. Geometry of the time-discretization. In this section we consider the geometry of the forward ($c = 0$) and backward ($c = 1$) Euler schemes. This will allow us to obtain lower and upper bounds for the blow-up time corresponding to a given "blow-up thresholds" M_i.

Let $e(t) := U(t) - V(t)$, where $U(t)$ and $V(t)$ are, respectively, the solutions of (3.1) and (3.4) (that is, $e(t)$ is the collocation error corresponding to the collocation solution $V(t)$ to (3.1)).

Integrating (3.1) from t_m to t_{m+1}, we obtain

$$\begin{aligned} U(P_i, t_{m+1}) &= U(P_i, t_m) + \sum_{k=1}^{d} \int_{t_m}^{t_{m+1}} D_k^2 U(P_i, \tau) d\tau + \int_{t_m}^{t_{m+1}} f(U(P_i, \tau)) d\tau \\ &= U(P_i, t_m) + \int_{t_m}^{t_{m+1}} G(U(P_i, \tau)) d\tau. \end{aligned} \tag{4.1}$$

Combining (3.4) and (4.1) leads to

$$\begin{aligned} e(P_i, t_{m+1}) &= e(P_i, t_m) + \int_{t_m}^{t_{m+1}} (G(U(P_i, \tau)) - G(V(P_i, t_m) \\ &\qquad + c(V(P_i, t_{m+1}) - V(P_i, t_m)))) d\tau \\ &= e(P_i, t_m) + \tau_m \left(\sum_{k=1}^{d} D_k^2 + L(\rho(P_i, \varsigma))\right) \\ &\quad (e(P_i, t_m) + c(e(P_i, t_{m+1}) - e(P_i, t_m))) \\ &\quad + \int_{t_m}^{t_{m+1}} (G(U(P_i, \tau)) - G(U(P_i, t_m) + c(U(P_i, t_{m+1}) - U(P_i, t_m)))) d\tau. \end{aligned} \tag{4.2}$$

Here, $L(\rho(P_i, \varsigma))$ $(t_m < \varsigma < t_{m+1})$ is the function at the point P_i such that $f(a) - f(b) = L(\rho)(a - b)$.

Now we discuss the two specific time-stepping schemes employing $c = 0,\ c = 1$. From (4.2), we obtain

1. $c = 0$:

$$e(P_i, t_{m+1}) = e(P_i, t_m) + \tau_m \left(\sum_{k=1}^{d} D_k^2 + L(\rho)\right) e(P_i, t_m) + \frac{\tau_m^2}{2} \dot{G}(U(P_i, \xi)). \tag{4.3}$$

2. $c = 1$:

(4.4)
$$e(P_i, t_{m+1}) = e(P_i, t_m) + \tau_m \left(\sum_{k=1}^{d} D_k^2 + L(\rho) \right) e(P_i, t_{m+1}) - \frac{\tau_m^2}{2} \dot{G}(U(P_i, \xi)).$$

Observe that $\dot{G}(P_i, \xi)) = \ddot{U}(P_i, \xi)$ in the above expressions. Hence, upon setting $\ddot{U}(P_i, \xi) \geq 0$, we find what follows:

LEMMA 4.1.

1. *For $c = 0$:*

$$e(P_i, t_1) = 0, \ e(P_i, t_{m+1}) \geq 0 \quad \text{if } \tau_m \left(-\frac{2N}{h^2} + L(\rho) \right) > -1. \tag{4.5}$$

2. *For $c = 1$:*

$$e(P_i, t_1) = 0, \ e(P_i, t_{m+1}) \leq 0 \quad \text{if } \tau_m L(\rho) < 1. \tag{4.6}$$

Proof. For the case $c = 0$, the conclusion (4.5) is simple. Now we focus on the case $c = 1$. We first consider the time t_2. From (4.4), setting $\ddot{U}(P_i, \xi) \geq 0$ and $e(P_i, t_1) = 0$, if $\tau_m L(\rho) < 1$, one obtains a strictly diagonally dominant matrix A and a vector $b < 0$ such that

$$Ae = b. \tag{4.7}$$

By using Gauss–Seidel iteration to solve (4.7) with initial approximation $e^0 = 0$, we obtain $e(P_i, t_2) < 0$. Repeating the above process, we arrive at the conclusion (4.6). The proof is completed. ∎

Remark. If we consider $c = 1/2$, then the collocation method corresponds to the (continuous) implicit Crank–Nicolson scheme. In this case we have

$$e(P_i, t_{m+1}) = e(P_i, t_m) + \frac{\tau_m}{2} \left(\sum_{k=1}^{d} D_k^2 + L(\rho) \right) (e(P_i, t_m) + e(P_i, t_{m+1})) + R_t$$

and

$$R_t = \tau_m \left(f\left(U\left(P_i, \frac{t_m + t_{m+1}}{2} \right) \right) - \frac{f\left(U(P_i, t_m)\right) + f\left(U(P_i, t_{m+1})\right)}{2} \right) - \frac{\tau_m^3}{12} \ddot{G}(P_i, \xi).$$

By using Taylor's expansion, we see that $R_t = \mathcal{O}(\tau_m^3)$. However, it is not possible to deduce whether the value R_t is positive or not.

The following lemma can be found in Bandle and Brunner [4].

LEMMA 4.2. *Suppose that*

$$|U(\cdot, t)|_\infty, |V(\cdot, t)|_\infty \leq M \quad \text{for} \quad t \in (0, t'(M)].$$

Then

$$|e(t)| \leq B(M, t')\tau,$$

where $\tau := \max_{(m)} \tau_m$ for some finite constant $B(M, t')$.

Assume that $\tilde{T}_b < \infty$ is the blow-up time for problem (3.1), and $\tilde{T}_b^{FE}$ and $\tilde{T}_b^{BE}$ denote, respectively, the blow-up times for the discretized problem (3.4) corresponding to the forward Euler method ($c = 0$) and the backward Euler method ($c = 1$).

Combining the results of Lemmas 4.1 and 4.2, we obtain the following comparison results.

THEOREM 4.3. *Assume that, for fixed threshold $M \gg 1$, the solution $U(t)$ is a monotonically increasing and concave function. Then*

$$\text{(4.8)} \qquad \text{(a)} \quad \tilde{T}_b^{BE}(M) < \tilde{T}_b(M) < \tilde{T}_b^{FE}(M),$$

$$\text{(4.9)} \qquad \text{(b)} \quad \tilde{T}_b^{BE}(\tau, M) < \tilde{T}_b^{BE}(\tau/2, M) < \tilde{T}_b^{FE}(\tau/2, M) < \tilde{T}_b^{FE}(\tau, M).$$

Proof. In fact, from Lemma 4.1, one can obtain the lower and upper bounds for the blow-up time under a given "blow-up threshold" M, which leads to the result (a). Lemma 4.2 shows that

$$\lim_{\tau \to 0} \tilde{T}_b^{BE}(\tau, M) = \lim_{\tau \to 0} \tilde{T}_b^{FE}(\tau, M) = \tilde{T}_b(M).$$

This implies the conclusion (b), and the proof is completed. ∎

5. Numerical examples. In this section we illustrate various aspects of the performance of our numerical schemes. Subsection 5.1 is devoted to discussing the efficiency of the constrained absorbing BCs. In subsection 5.2 we show the dependence of the blow-up time on the length of the computational interval when using the Crank–Nicolson scheme. Subsection 5.3 contains a detailed comparison of the numerical blow-up times corresponding to the implicit Euler scheme, the explicit Euler scheme, and the Crank–Nicolson scheme. We also list the dependence of the approximate blow-up time on the refinement of the spatial mesh. In subsection 5.4, we discuss the dynamics of the numerical blow-up point(s) for the cases of single-point and two-point blow-up. In subsection 5.5, we look at the dependence of the numerical blow-up time on the chosen blow-up threshold M. Finally, we give an example corresponding to an initial function u_0 that changes sign on its domain.

In our examples we used the following initial functions u_0:

1. Single Gaussian function:

$$\text{(5.1)} \qquad u_0(x) = g(x) \exp\left(-x^2\right).$$

2. Double Gaussian function:

$$\text{(5.2)} \qquad u_0(x) = g_1(x) \exp\left(-(x - c_1)^2\right) + g_2(x) \exp\left(-(x + c_2)^2\right).$$

3. Nonsymmetric u_0:

$$\text{(5.3)} \qquad u_0(x) = \begin{cases} 0, & -x_l \le x < 0, \\ A \sin(\pi x/(2x_a)), & 0 \le x < x_a, \\ A \cos(\pi(x - x_a)/(2(1 - x_a))), & x_a \le x < 1, \\ 0, & 1 \le x \le x_r, \end{cases}$$

where c_i, A, and x_a are positive constants and g, g_i $(i = 1, 2)$ are given functions. In the computations for the first two initial functions we set $u_0(x) = 0$ when $x \notin \Omega_i$.

For prescribed blow-up thresholds M_i, let $\tilde{T}_b^{CN}(M_i)$ denote the blow-up time for the Crank–Nicolson scheme ($c = 1/2$) with respect to the thresholds M_i, and assume that $\mu(M_i)$ is the number of time steps needed to reach the blow-up threshold M_i (which we choose to be $M_1 = 10^5$ and $M_2 = 10^6$). Moreover, the numerical blow-up times $\tilde{T}_b^{FE}(M_i)$ and $\tilde{T}_b^{BE}(M_i)$ correspond to the forward Euler scheme ($c = 0$) and the backward Euler scheme ($c = 1$), respectively.

TABLE 1
Comparison of "exact solution" with numerical solution at $x_l = -2$.

	Exact value	$h = 0.01$	$h = 0.005$	$h = 0.0025$
$s_0 = 0.5$	2.15331	2.20357	2.19965	2.19779
$s_0 = 1.0$	2.15331	2.17902	2.17749	2.17669
$s_0 = 2.0$	2.15331	2.15975	2.15907	2.15876
$s_0 = 2.5$	2.15331	2.15740	2.15687	2.15664
$s_0 = 3.0$	2.15331	2.15765	2.15709	2.15684
$s_0 = 3.5$	2.15331	2.15917	2.15850	2.15820
$s_0 = 4.0$	2.15331	2.16099	2.16020	2.15984

TABLE 2
Comparison of "exact solution" with numerical solution at $x_l = -1.5$.

	Exact value	$h = 0.01$	$h = 0.005$	$h = 0.0025$
$s_0 = 1.0$	12.9083	13.2508	13.2198	13.2053
$s_0 = 2.5$	12.9083	13.1567	13.1342	13.1237

TABLE 3
The errors and convergence rates for $s_0 = 2.5$.

	$\Delta x = 0.02$	Order	$\Delta x = 0.01$	Order	$\Delta x = 0.05$	Order
$x_l = -2.0$	1.083e−2	–	2.653e−3	2.029	6.411e−4	2.049
$x_l = -1.5$	1.084e−2	–	2.658e−3	2.028	6.445e−4	2.044

To adaptively choose the time steps τ_m (following 3.5), we set the corresponding parameters $\alpha = 2,\ d = 1,\ s_0 = 1,\ p = 2$, and $h = (x_r - x_l)/r$. For the (explicit) forward Euler method ($c = 0$), it is readily seen that the adaptive time-step selection cannot work.

5.1. Efficiency of the constructed ABCs. We first consider the choice of the parameter s_0 arising in the ABCs (2.20). To see the influence of s_0 on the ABCs, the initial function is the single Gaussian function (5.1). In the calculation, we use the Crank–Nicolson scheme ($c = 1/2$) and choose $g(x) = 100,\ \Delta t = 10^{-6},\ \tilde{T}_b^{CN}(10^5) = 0.010184$, and $x_l = -x_r$. The Crank–Nicolson method is unconditionally stable and of second-order accuracy in both space and time. We compare the numerical values at the artificial boundaries x_l with the corresponding "exact" (reference) solution obtained by computing the numerical solution in the larger interval $[-10, 10]$ with smaller mesh size $h = 0.0005$.

In Tables 1 and 2, we list the "exact value" and the numerical values at the artificial boundaries x_l with different parameters s_0 in the time point $\tilde{T}_b^{CN}(M_1) = 0.010184$, where the computational interval is $[x_l, x_r] = [-2, 2]$ and $[-1.5, 1.5]$, with spatial mesh sizes $h = 0.01,\ 0.005,\ 0.0025$. It is readily seen that the larger the computational domain, the closer the numerical solutions tend to the "exact" solution, and the numerical solutions approximate the "exact" solution well even in a smaller domain. Another important way to measure the performance of s_0 is to see the convergence rate with respect to the L_1-norm,

$$L_1(t) = \frac{||u(\cdot, t) - u_h(\cdot, t)||_1}{||u(\cdot, t)||_1}.$$

Table 3 shows the L_1-errors and the second-order convergence rate, for $s_0 = 2.5$, in the computational domains $[-1.5, 1.5]$ and $[-2, 2]$. Figure 1 plots the change of the

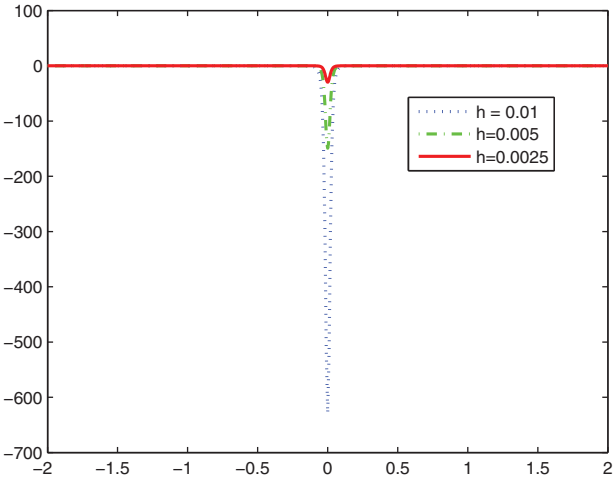

FIG. 1. *Error with* $s_0 = 1.0$ *on the interval* $[-2, 2]$.

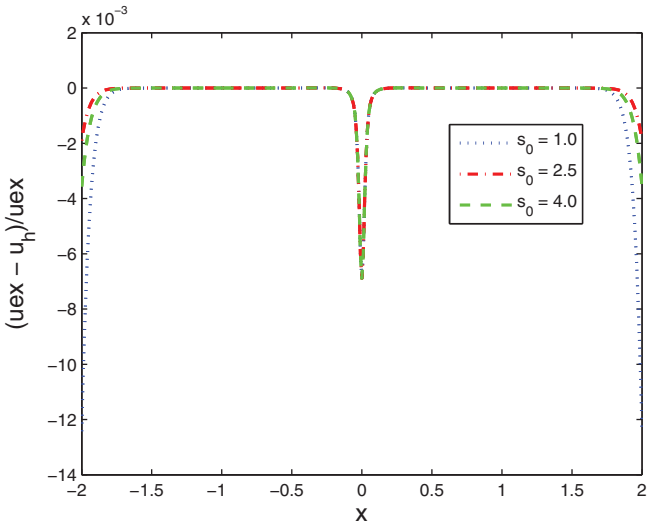

FIG. 2. *Relative error for different* s_0 *on the interval* $[-2, 2]$.

error $uex(x) - u(x)$ for different spatial mesh sizes for the fixed parameter $s_0 = 1.0$. Figure 2 shows the relative error $(uex(x)-u(x))/uex(x)$ for different parameter values s_0 for fixed $h = 0.05$.

From Tables 1–3 and Figures 1–2, we observe that the choice of parameter s_0 plays an important role in the performance of the ABC. The parameter s_0 can be chosen in a larger interval such that the BCs work well even when the computational domain is smaller. That is to say, the relative error is less than 1% for the choice of the parameters s_0 in a larger interval.

5.2. Dependence of the 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 $\tilde{T}_b^{CN}(M_i)$. In the calculations, we use the fixed spatial

TABLE 4
Single Gaussian function: $g(x) = 100$ *and* $s_0 = 2.0$.

$[x_l, x_r]$	$[-2, 2]$	$[-3, 3]$	$[-4, 4]$	$[-5, 5]$
$\tilde{T}_b^{CN}(M_1)$	0.0101840	0.0101840	0.0101840	0.0101840
$\tilde{T}_b^{CN}(M_2)$	0.01019229	0.01019229	0.01019229	0.01019229

TABLE 5
Double Gaussian function: $g_1(x) = g_2(x) = 50, c_1 = -c_2 = 2,\ s_0 = 3$.

$[x_l, x_r]$	$[-4, 4]$	$[-4.5, 4.5]$	$[-5, 5]$	$[-5.5, 5.5]$
$\tilde{T}_b^{CN}(M_1)$	0.02075139	0.02075139	0.02075139	0.02075139
$\tilde{T}_b^{CN}(M_2)$	0.02076241	0.02076241	0.02076241	0.02076241

TABLE 6
Nonsymmetric initial function: $x_a = 0.2,\ A = 14$.

$[x_l, x_r]$	$[-0.5, 1.5]$	$[-1, 2]$	$[-1.5, 2.5]$	$[-2, 3]$
$\tilde{T}_b^{CN}(M_1)$	0.129240478	0.129247219	0.129247220	0.129247220
$\tilde{T}_b^{CN}(M_2)$	0.12925010	0.129257226	0.129257227	0.129257227

mesh size $h = 0.005$, while the adaptive time steps τ_m are as in (3.5). (The same temporal step size strategy will be used in the following subsections.) For the different lengths of the computational domains we list, in Table 4, a sample of computed blow-up times $T_b^{CN}(M_1)$ for the single Gaussian initial function corresponding to the values $g(x) = 100$ and $s_0 = 2.0$. In Table 5 we show the blow-up times for the double Gaussian initial function corresponding to the values $g_1(x) = g_2(x) = 50,\ c_1 = c_2 = 2$, and $s_0 = 3$. In Table 6, we give the blow-up times for the nonsymmetric initial function corresponding to the values $x_a = 0.2,\ A = 14$. Tables 4–6 reveal that the blow-up times are insensitive to the choice of the computational intervals with the ABCs. They also illustrate the efficiency of the constructed ABCs. Thus, in practical computations we may generate the numerical blow-up times by different schemes in the smaller spatial domain.

5.3. Approximation of the exact blow-up time by using the Crank–Nicholson, backward Euler and forward Euler schemes, and the dependence of the numerical blow-up time on spatial mesh refinement. Using the different values $c = 1,\ 1/2,\ 0$ of the collocation parameter, we obtain the computed blow-up times with respect to the blow-up threshold M_1 for the one-dimensional model. A selection of numerical blow-up times for different initial functions is presented below.

1. Single Gaussian function:
   Table 7 shows the computed blow-up times $\tilde{T}_b(M_1)$ and the number of time steps $\mu(M_1)$ corresponding to the values $g(x) = 100$ in the computational domain $[-3, 3]$. Table 8 shows the computed blow-up times and the number of time steps corresponding to the values $g(x) = 100x^2$ in the computational domain $[-4.5, 4.5]$.
2. Double Gaussian function:
   Table 9 shows the computed blow-up times and the number of time steps corresponding to the values $g_1(x) = g_2(x) = 50,\ c_1 = c_2 = 2$ in the computational domain $[-5.5, 5.5]$.

TABLE 7
Blow-up times and number of time steps with $g(x) = 100$.

r	$\tilde{T}_b^{BE}(M_1)$	$\mu_b^{BE}(M_1)$	$\tilde{T}_b^{CN}(M_1)$	$\mu_b^{CN}(M_1)$	$\tilde{T}_b^{FE}(M_1)$	$\mu_b^{FE}(M_1)$
200	0.009804	107	0.010147	56	0.010525	124
400	0.010055	381	0.010174	193	0.010304	396
800	0.010143	1468	0.010181	737	0.010222	1483

TABLE 8
Blow-up times and number of time steps with $g(x) = 100x^2$.

r	$\tilde{T}_b^{BE}(M_1)$	$\mu_b^{BE}(M_1)$	$\tilde{T}_b^{CN}(M_1)$	$\mu_b^{CN}(M_1)$	$\tilde{T}_b^{FE}(M_1)$	$\mu_b^{FE}(M_1)$
200	0.029031	139	0.029840	72	0.030722	157
400	0.029619	495	0.029898	250	0.030207	513
800	0.029829	1915	0.029920	961	0.030014	1932

TABLE 9
Blow-up times and number of time steps.

r	$\tilde{T}_b^{BE}(M_1)$	$\mu_b^{BE}(M_1)$	$\tilde{T}_b^{CN}(M_1)$	$\mu_b^{CN}(M_1)$	$\tilde{T}_b^{FE}(M_1)$	$\mu_b^{FE}(M_1)$
200	0.019674	73	0.020642	39	0.021719	91
400	0.020363	240	0.020719	122	0.021116	257
800	0.020626	900	0.020748	453	0.020874	916

We observe from Tables 7–9 that, for the given spatial mesh size h, relationship between the blow-up times computed by the three schemes is given by

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

One also sees that the number of time steps $\mu(M_i)$ and the blow-up times (Crank–Nicolson and backward Euler schemes) increase when the mesh is refined. The blow-up times for the forward Euler scheme decrease when the spatial mesh is refined.

5.4. Single-point blow-up versus two-point blow-up. In this subsection we consider the dynamics of (numerical) single blow-up points for different types of initial functions.

1. Single-point blow-up:
   If we use the single Gaussian function as the initial data corresponding to the value $g(x) = 100$, we observe that the location of the single blow-up point does not change. If we use the nonsymmetric function with the values $A = 14,\ x_a = 0.2$ in $[x_l, x_r] = [-2, 3]$, the movement of the single blow-up point is shown in Figure 3.
2. Two-point blow-up:
   For the single Gaussian initial function, we choose $g(x) = 100x^2$. There exist two isolated blow-up points (see Figures 4 and 5). These two points retain their location (as t increases): they are moving parallel to each other. The same is true for a symmetric double Gaussian initial function.

5.5. Dependence of the numerical blow-up time on the blow-up threshold. We now consider the dependence of the numerical blow-up time on the blow-up threshold. In Table 10 we list a sample of computed blow-up times for different initial functions and blow-up thresholds. For the single Gaussian function, the shown values correspond to $g(x) = 100,\ [x_l, x_r] = [-4, 4]$. For the double Gaussian function, the val-

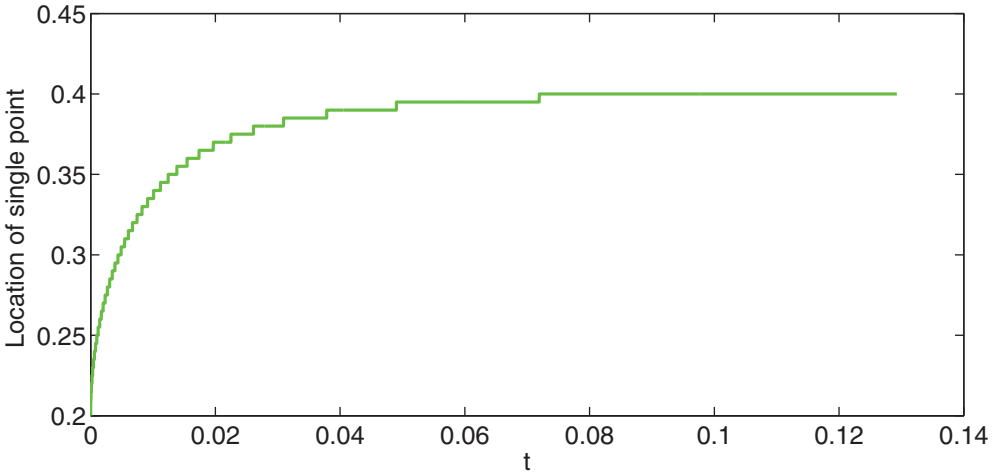

FIG. 3. *The movement of the single blow-up point with nonsymmetric initial function.*

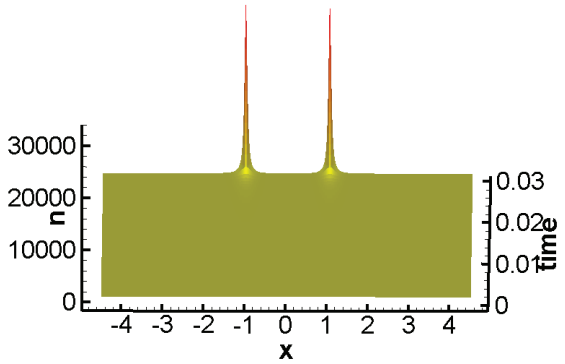

FIG. 4. *The solution with initial function* $g(x) = 100x^2$.

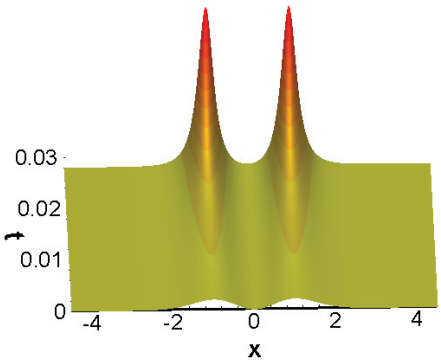

FIG. 5. *The part solution with initial function* $g(x) = 100x^2$.

ues are those for $g_1(x) = g_2(x) = 50,\ c_1 = c_2 = 2,\ [x_l, x_r] = [-5.5, 5.5]$. For the nonsymmetric initial function, the values are chosen as $A = 14, x_a = 0.2, [x_l, x_r] = [-2.3]$. In the calculations, we chose the spatial mesh size $h = 0.001$ and the parameter $s_0 = 2.5$.

TABLE 10
Dependence of the blow-up time on the blow-up threshold.

Initial data	Single Gaussian	Double Gaussian	Nonsymmetric function
$\tilde{T}_b^{CN}(10^{10})$	0.0101949388	0.0207649007	0.1292587529
$\tilde{T}_b^{CN}(10^{15})$	0.0101949390	0.0207649008	0.1292587530

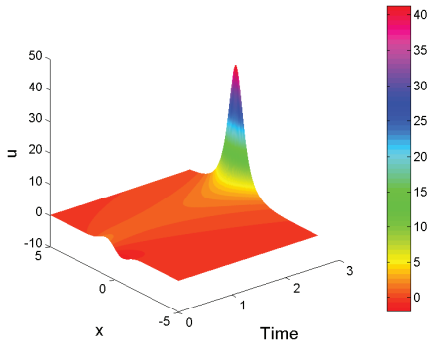

FIG. 6. *The partial evolution of the solution.*

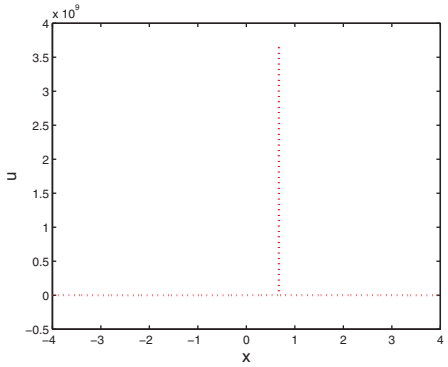

FIG. 7. *The solution with threshold* $M = 10^{10}$.

5.6. Dependence of the blow-up on the initial data $u_0(x)$. The above numerical tests show the finite-time blow-up of the numerical solution when the initial data $u_0(x)$ are positive. It is also interesting to see what happens if the initial function changes its sign; for example, if it is given by

$$u_0(x) = a\sin(x)\exp\left(-x^2\right).$$

In the calculation, let $h = 0.002$, $s_0 = 2.5$, and $[x_l, x_r] = [-4, 4]$. We observe the following behavior:

I. Blow-up: $(a = 5)$.
The solution will blow up. Figure 6 shows the evolution of the solutions in a short time interval. In Figure 7 we plot the solution at blow-up time with the blow-up threshold $M = 10^{10}$.

II. Blow-up: $(a = 1)$.
Figure 8 presents the evolution of solution over a long time. We see that here the solution exists globally and will tend to zero.

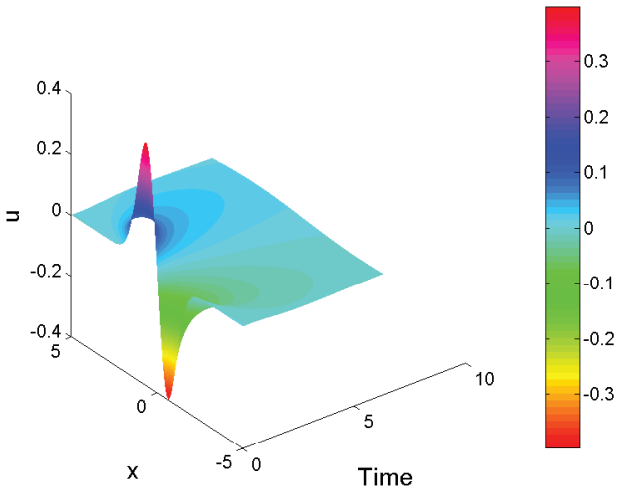

FIG. 8. *The partial evolution of the solution.*

6. Concluding remarks. The numerical treatment of initial-value problems for semilinear parabolic PDEs on unbounded spatial domains given by, e.g., $\mathcal{R}^d$ $(d = 1, 2)$, whose solutions blow up in finite time requires selecting an appropriate (bounded) computational domain. The efficiency and accuracy of the corresponding numerical scheme depend crucially on the construction of suitable LABCs. This paper presents a unified approach to derive such LABCs for one-dimensional and 2D spatial domains. This is combined with a simple, robust adaptive time-stepping method. The blow-up time for the semidiscretized equation (3.1) is shown bounded below by the blow-up time for the discretized equation (3.4) using the backward time stepping, and it is bounded above by the blow-up time for the discretized equation (3.4) using the forward time stepping. A broad range of numerical examples illustrates various aspects of our numerical approach for the one-dimensional problem. The blow-up situation for 2D problems is much more complicated, which will be considered in our future work.

Acknowledgments. The authors gratefully acknowledge the valuable comments by the two referees; they led to a considerably improved version of our paper.

REFERENCES

[1] G. ACOSTA, R. G. DURAN, AND J. D. ROSSI, *An adaptive time step procedure for a parabolic problem with blow-up*, Computing, 68 (2002), pp. 343–373.
[2] X. ANTOINE, C. BESSE, AND S. DESCOMBES, *Artificial boundary conditions for one-dimensional cubic nonlinear Schrödinger equations*, SIAM J. Numer. Anal., 43 (2006), pp. 2272–2293.
[3] A. BAMBERGER, B. ENGQUIST, L. HALPERN, AND P. JOLY, *Higher order paraxial wave equation approximations in heterogeneous media*, SIAM J. Appl. Math., 48 (1988), pp. 129–154.
[4] C. BANDLE AND H. BRUNNER, *Numerical analysis of semilinear parabolic problems with blow-up solutions*, Rev. R. Acad. Cienc. Exactas Fis. Nat. (Esp.), 88 (1994), pp. 203–222.
[5] C. BANDLE AND H. BRUNNER, *Blow-up in diffusion equations: A survey*, J. Comput. Appl. Math., 97 (1998), pp. 2–22.
[6] C. BRÄNDLE, F. QUIÓRS, AND J. D. ROSSI, *An adaptive numerical method to handle blow-up in a parabolic system*, Numer. Math., 102 (2005), pp. 39–59.
[7] C. J. BUDD, W. HUANG, AND R. D. RUSSELL, *Moving mesh methods for problems with blow-up*, SIAM J. Sci. Comput., 17 (1996), pp. 305–327.
[8] H. FUJITA, *On the blowing up of solutions of the Cauchy problem for* $u_t = \Delta u + u^{1+\alpha}$, J. Fac. Sci. Univ. Tokyo Sect. I, 13 (1966), pp. 109–124.

[9] T. Hagstrom and H. B. Keller, *Asymptotic boundary conditions and numerical methods for nonlinear elliptic problems on unbounded domains*, Math. Comp., 48 (1987), pp. 449–470.
[10] H. Han, *The artificial boundary method – numerical solutions of partial differential equations on unbounded domains*, in Frontiers and Prospects of Contemporary Applied Mathematics Ser. Contemp. Appl. Math. CAM 6, T. Li and P. Zhang, eds., Higher Education Press, Beijing, 2005, pp. 33–58.
[11] H. D. Han and X. N. Wu, *Approximation of infinite boundary condition and its applications to finite element methods*, J. Comput. Math., 3 (1985), pp. 179–192.
[12] W. Huang, J. Ma, and R. D. Russell, *A study of moving mesh PDE methods for numerical simulation of blow-up in reaction diffusion equations*, J. Comput. Phys., 227 (2008), pp. 6532–6552.
[13] L. A. Khan and P. L. F. Liu, *Numerical analyses of operator-splitting algorithms for the two-dimensional advection-diffusion equation*, Comput. Methods Appl. Mech. Engrg., 152 (1998), pp. 337–359.
[14] H. A. Levine, *The role of critical exponents in blowup theorems*, SIAM Rev., 32 (1990), pp. 262–288.
[15] T. Nakagawa, *Blowing up of a finite difference solution to* $u_t = u_{xx} + u^2$, Appl. Math. Optim., 2 (1975), pp. 337–350.
[16] T. Nakagawa and T. Ushijima, *Finite element analysis of the semi-linear heat equation of blow-up type*, in Topics in Numerical Analysis, III, J. J. H. Miller, ed., Academic Press, London, 1977, pp. 275–291.
[17] M. Berger and R. V. Kohn, *A rescaling algorithm for the numerical calculation of blowing-up solutions*, Comm. Pure Appl. Math., 41 (1988), pp. 841–863.
[18] A. A. Samarskii, V. A. Galaktionov, S. P. Kurdyumov, and A. P. Mikhailov, *Blow-up in Quasilinear Parabolic Equations*, Walter de Gruyter, Berlin, 1995.
[19] P. Souplet, *A note on diffusion-induced blow-up*, J. Dynam. Differential Equatioins, 19 (2007), pp. 819–823.
[20] P. Souplet, *Uniform blow-up profiles and boundary behavior for diffusion equations with nonlocal nonlinear source*, J. Differential Equations, 153 (1999), pp. 374–406.
[21] P. Souplet, *Uniform blow-up profile and boundary behavior for a non-local reaction-diffusion equation with critical damping*, Math. Methods Appl. Sci., 27 (2004), pp. 1819–1829.
[22] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal., 5 (1968), pp. 506–517.
[23] X. Wu and J. Zhang, *Artificial boundary method for two-dimensional Burgers' equation*, Comput. Math. Appl., 56 (2008), pp. 242–256.
[24] Z. Xu and H. Han, *Absorbing boundary conditions for nonlinear Schrödinger equations*, Phys. Rev. E (3), 74 (2006), 037704.
[25] Z. Xu, H. Han, and X. Wu, *Numerical method for the determimistic Kardar-Parisi-Zhang equation in unbounded domains*, Commun. Comput. Phys., 1 (2006), pp. 479–493.
[26] J. Zhang, Z. Xu and X. Wu, *Unified approach to split absorbing boundary conditions for nonlinear Schrödinger equations*, Phys. Rev. E (3), 78 (2008), 026709.
[27] J. Zhang, Z. Xu and X. Wu, *Unified approach to split absorbing boundary conditions for nonlinear Schrödinger equations: Two dimensional case*, Phys. Rev. E (3), 79 (2009), 046711.
[28] C. Zheng, *Exact nonreflecting boundary conditions for one-dimensional cubic nonlinear Schrödinger equations*, J. Comput. Phys., 215 (2006), pp. 552–565.
[29] C. Zheng, *Numerical solution to the sine-Gordon equation defined on the whole real axis*, SIAM J. Sci. Comput., 29 (2007), pp. 2494–2506.