

Numerical Solution of the Heat Equation with Nonlinear Boundary Conditions in Unbounded Domains

Miglena Koleva, Lubin Vulkov

*Center of Applied Mathematics and Informatics, University of Rousse,
Rousse 7017, Bulgaria*

Received 25 August 2005; accepted 9 July 2006

Published online 26 October 2006 in Wiley InterScience (www.interscience.wiley.com).

DOI 10.1002/num.20183

The numerical solution of the heat equation in unbounded domains (for a 1D problem-semi-infinite line and for a 2D one semi-infinite strip) is considered. The artificial boundaries are introduced and the exact artificial boundary conditions are derived. The original problems are transformed into problems on finite domains. The space semi-discretization by finite element method and the full approximation by the implicit-explicit Euler's method are presented. The solvability of the full discretization schemes is analyzed. Computational examples demonstrate the accuracy and the efficiency of the algorithms. Also, the behavior of blowing up solutions is examined numerically. © 2006 Wiley Periodicals, Inc. *Numer Methods Partial Differential Eq* 23: 379–399, 2007

Keywords: heat equation; nonlinear boundary conditions; unbounded domains; artificial boundary conditions; finite element schemes; numerical blow-up

I. INTRODUCTION

Many problems in heat-mass transfer, fluid dynamics, engineering etc. can be described by parabolic equations in unbounded domains with nonlinear boundary conditions. When the analytical solution is not available or it is too complicated to use, then a numerical method is necessary for solving the problem. The most widely used methods are the finite element and finite difference schemes. Since the grids are finite, then on the grid boundary the same type boundary conditions as on the infinity in the differential problem, are often imposed see for example [1,2]. This however, leads to loss of accuracy. More accurate are the artificial boundary conditions. But their construction is a difficult and complicated question, solved separately for each class of problems, (see [3–6]). For linear parabolic problems with linear boundary conditions such results can be found in [4,6]. In this article we consider linear equations subjected with nonlinear boundary conditions. Our focus is on the computational efficiency of the numerical method.

Correspondence to: Miglena Koleva, Center of Applied Mathematics and Informatics, University of Rousse, 8 Studentska strasse, Rousse, 7017, Bulgaria (e-mail: mkoleva@ru.acad.bu)

Contract grant sponsor: Bulgarian National Fund of Science; contract grant number: VU-MI-106/2005.

© 2006 Wiley Periodicals, Inc.

In the one-dimensional (1D) case, we consider the problem

$$u_t = u_{xx} \quad \text{for } x > 0, \quad t > 0, \quad (1.1)$$

$$-u_x = f(u) \quad \text{for } x = 0, \quad t > 0, \quad (1.2)$$

$$u(x, 0) = u_0(x) \geq 0 \quad \text{for } x > 0; \quad \text{supp } u_0 < \infty, \quad (1.3)$$

$$-u'_0(0) = f(u_0(0)). \quad (1.4)$$

The function $f(u)$ often is positive and tends to infinity as $u \rightarrow \infty$. Thus, in heat flow interpretation of the condition (1.2) is an absorption law, which makes heat flow in the body, [7–9] (in the present article the body is infinite). In [10], existence and nonexistence in the large in time of solutions of such problems are studied. The authors of [11] discuss a rescaling method for estimation of the blow-up time of problem (1.1)–(1.4). It can be used for numerical calculation of blow-up solutions, (see [2, 12]).

In the theory of blowing-up solutions [7, 10, 13, 14] for problems of this type, when $f(u) = u^p$, $p > 0$ was proved that there exist positive critical values p_0, p_c (with $p_0 < p_c$) such that for $p \in (0, p_0]$, all solutions are global while for $p \in (p_0, p_c]$ any solution $u \not\equiv 0$ blows up in a finite time and for $p > p_c$ small data solutions exist globally in time, while large data solutions are nonglobal. In this concrete case (1.1)–(1.5) was found in [10] that $p_c = 2$, $p_0 = 1$. Throughout in this article we will assume

$$p > p_0. \quad (1.5)$$

We also will investigate the case of dynamical boundary conditions [7]:

$$u_t - u_x = u^p \quad \text{for } x = 0, \quad t > 0. \quad (1.2')$$

In Section 5 we shall examine numerically the following analytical results for problem (1.1)–(1.5) obtained in [10].

R 1. $u(x, t)$ blows up for large u_0 and $p > 1$. More precisely, if

$$\int_0^\infty u_0(x)\phi(x)dx > (\sqrt{k\pi})^{1/(p-1)}, \quad \phi(x) = 2\sqrt{\frac{k}{\pi}}e^{-k^2x}, \quad k > 0 \text{ const}, \quad (1.6)$$

$u(0, t)$ blows up in finite time. More precisely, in [10] is shown that for $t \geq 0$,

$$u(0, t) \geq F(t) = \int_0^\infty u(x, t)\phi(x)dx \rightarrow \infty, \quad \text{as } t \rightarrow T < \infty.$$

R 2. Any $u \not\equiv 0$ blows up if $1 < p < 2$. Indeed, consider (1.6) more carefully. It is equivalent to the inequality

$$2\sqrt{\frac{k}{\pi}} \int_0^\infty u_0(x)e^{-kx^2}dx > (\sqrt{k\pi})^{1/(p-1)}. \quad (1.7)$$

Since $u_0 \not\equiv 0$, we conclude that (1.7) is valid for arbitrarily small $k > 0$ if $p < 2$.

R 3. Global small solutions for $p > 2$. We introduce the supersolution, [10].

$$\bar{u}(x, t) = (T + t)^{-1/2(p-1)} f(\eta), \quad f(\eta) = Ae^{-\alpha(\eta+b)^2}, \quad \eta = \frac{x}{(T + t)^{1/2}}, \quad (1.8)$$

where T, A, α, b are positive constants and $4\alpha - 1 < 0$, $0 < b \ll 1$. It is shown that if A sufficiently small and $u_0(x) \leq \bar{u}(x, 0)$ for $x > 0$ then $u \leq \bar{u}$ on $R_+ \times [0, T]$.

R 4. If $p \leq 1$ all solutions with bounded initial values are global.

R 5. Any $u \not\equiv 0$ blows up for $p = 2$. Now, starting from the representation formula ($p = 2$)

$$u(x, t) = \int_0^\infty G(x, y, t) u_0(y) dy + \int_0^t u^p(0, \eta) G(x, 0, t - \eta) d\eta, \quad (1.9)$$

is proved

$$u(0, t) \geq C_0 t^{-(1/2)} F(t), \quad F(t) \equiv \int_0^t u^2(0, \eta) d\eta \rightarrow \infty, \quad \text{as } t \rightarrow T < \infty, \quad C_0 = \sqrt{\frac{2}{\pi}}.$$

We also consider the 2D problem

$$u_t = a \Delta u \quad (x, y, t) \in \Omega = \{0 < x < l, y > 0, 0 \leq t < \infty\}, \quad (1.10)$$

$$u(0, y, t) = u(l, y, t) = 0, \quad y > 0, t > 0, \quad (1.11)$$

$$c_0 u_t - u_y = f(u), \quad y = 0, \quad 0 < x < l, \quad t > 0, \quad (1.12)$$

$$u(x, y, 0) = u_0(x, y), \quad \text{supp } u_0 = (0, l) \times (0, L), \quad L < \infty, \quad (1.13)$$

where p, l , and a are real numbers. This problem in comparison with the 1D one is less studied. For our calculations and representation formulas below, we shall assume that the solution u has all necessary derivatives.

Our goal is the numerical solution of the problem, using the method of the artificial boundary conditions.

The remainder part of the article is organized as follows. In Section II we construct the artificial boundary conditions of problems (1.1)–(1.5) and (1.10)–(1.13). In Section III we derive the space semi-discretization of both problems. Section IV is devoted to the full discretization. Analysis for the solvability and stability of the algorithm is presented in Section V. Finally, in Section VI we give some numerical results.

II. ARTIFICIAL BOUNDARY CONDITIONS

In this section we introduce the artificial boundary conditions. Such condition, for problem (1.1)–(1.4) with Dirichlet boundary condition instead of (1.2), is constructed in [6]. Here we also derive an artificial boundary condition for the two-dimensional problem (1.10)–(1.13).

A. 1D Problem

For computation one can use formula (1.9), where

$$G(x, y, t) = (4\pi t)^{-1/2} [e^{-(x-y)^2/4t} + e^{-(x+y)^2/4t}].$$

However, it requires an infinite mesh.

From (1.3) $\text{supp } u_0(x) < \infty$ follows that there exists $0 < L < \infty$, $u_0(x) = 0$. Then $u \rightarrow 0$ when $x \rightarrow +\infty$, i.e. $u(+\infty, t) = 0$, $t > 0$. Now we shall consider the restriction of the solution of problem (1.1)–(1.5) on the domain $\Omega^l = \{l < x < +\infty, 0 \leq t < \infty\}$, $l > L$. The function $u(x, t)$ satisfies

$$\begin{aligned} u_t - u_{xx} &= 0, & (x, t) \in \Omega^l, \\ u(x, 0) &= 0, & l \leq x < +\infty, \\ u(x, t) &\rightarrow 0 & \text{when } x \rightarrow +\infty. \end{aligned} \quad (2.1)$$

Then, following a technique from [6], we find an exact boundary condition satisfied by the solution $u(x, t)$ on the artificial boundary $x = l$

$$\frac{\partial u(l, t)}{\partial x} = -\frac{1}{\sqrt{\pi}} \int_0^t \frac{\partial u(l, \lambda)}{\partial \lambda} \frac{1}{\sqrt{t-\lambda}} d\lambda, \quad 0 \leq t < +\infty. \quad (2.2)$$

In order to make the article self-contained, we include its derivation. Let us write the solution $u(x, t)$ for given $u(l, t)$ of (2.1):

$$u(x, t) = \frac{x-l}{2\sqrt{\pi}} \int_0^t u(l, \lambda) (t-\lambda)^{-(3/2)} e^{-[(x-l)^2]/[4(t-\lambda)]} d\lambda. \quad (2.3)$$

Setting $\rho = (x-l)/(2\sqrt{t-\lambda})$, then we have

$$\begin{aligned} u(x, t) &= \frac{2}{\sqrt{\pi}} \int_{\frac{x-l}{2\sqrt{t}}}^\infty u\left(l, t - \frac{(x-l)^2}{4\rho^2}\right) e^{-\rho^2} d\rho, \\ \frac{\partial u(x, t)}{\partial x} &= \frac{2}{\sqrt{\pi}} u(l, 0) e^{-[(x-l)^2]/[4t]} \frac{1}{2\sqrt{t}} + \frac{2}{\sqrt{\pi}} \int_{\frac{x-l}{2\sqrt{t}}}^\infty \frac{\partial u}{\partial t} \left(l, t - \frac{(x-l)^2}{4\rho^2}\right) \left(-\frac{2(x-l)}{4\rho^2}\right) e^{-\rho^2} d\rho. \end{aligned}$$

Returning to the variable λ , we get

$$\frac{\partial u(x, t)}{\partial x} = -\frac{1}{\sqrt{\pi}} \int_0^t \frac{\partial u(l, \lambda)}{\partial \lambda} \frac{1}{\sqrt{t-\lambda}} e^{-[(x-l)^2]/[4(t-\lambda)]} d\lambda.$$

Taking the limit $x \rightarrow +l$, we obtain (2.2).

Using the exact boundary condition (2.2), we reduce the original problem (1.1)–(1.5) to a problem on the bounded domain $\Omega^0 = \{0 < x < l, 0 \leq t < \infty\}$

$$u_t - u_{xx} = 0, \quad (x, t) \in \Omega^0, \quad (2.4)$$

$$c_0 u_t - u_x = f(u), \quad x = 0, \quad 0 \leq t < \infty, \quad (2.5)$$

$$u_x = -\frac{1}{\sqrt{\pi}} \int_0^t u_\lambda(l, \lambda) \frac{1}{\sqrt{t-\lambda}} d\lambda, \quad x = l, \quad 0 \leq t < \infty, \quad (2.6)$$

$$u(x, 0) = u_0(x), \quad x \in \bar{D} = [0, l], \quad (2.7)$$

where $c_0 = 0, 1$ [at $c_0 = 0$ we obtain (1.2) and for $c_0 = 1 - (1.2')$]. The solution of the problem (1.1)–(1.5) in the domain Ω^l can be computed by formula (2.3) for given $u(l, t)$.

B. 2D Problem

Similarly to the 1D problem, we restrict the domain $\Omega : \Omega^r = \{0 < x < l, r < y < \infty, 0 \leq t < \infty\}$, $r \geq L$ and consider the solution $u(x, y, t)$ of the problem

$$u_t = a \Delta u, \quad (x, y, t) \in \Omega^r, \quad (2.8)$$

$$u(0, y, t) = u(l, y, t) = 0, \quad y > r, \quad t > 0, \quad (2.9)$$

$$u(x, y, t) \rightarrow 0, \quad \text{when } y \rightarrow \infty, \quad 0 < x < l, \quad (2.10)$$

in order to construct artificial boundary condition.

Let us write the solution $u(x, y, t)$ of (2.8)–(2.10) for given $u(x, r, t)$, the value on $y = r$ of the solution to problem (1.10)–(1.13), (see [9]):

$$u(x, y, t) = a \int_0^t \int_0^l u(\xi, r, \tau) \left[\frac{\partial}{\partial \eta} G(x, y - r, \xi, \eta, t - \tau) \right]_{\eta=0} d\xi d\tau,$$

where

$$G(x, y - r, \xi, \eta, t - \tau) = G_1(x, \xi, t - \tau) G_2(y - r, \eta, t - \tau),$$

$$G_1(x, \xi, t - \tau) = \frac{2}{l} \sum_{d=1}^{\infty} \sin \frac{d\pi x}{l} \sin \frac{d\pi \xi}{l} e^{-[ad^2\pi^2(t-\tau)]/l^2},$$

$$G_2(y - r, \eta, t - \tau) = \frac{1}{2\sqrt{\pi a(t - \tau)}} \left\{ e^{-[(y-r-\eta)^2]/[4a(t-\tau)]} - e^{-[(y-r+\eta)^2]/[4a(t-\tau)]} \right\}.$$

After some calculations, the solution $u(x, y, t)$ of (2.8)–(2.10) takes the form

$$u(x, y, t) = \frac{1}{l\sqrt{a\pi}} \sum_{d=1}^{\infty} \sin \frac{d\pi x}{l} \int_0^l \sin \frac{d\pi \xi}{l} \mathcal{U}(\xi, y, t) d\xi, \quad (2.11)$$

where

$$\mathcal{U}(\xi, y, t) = (y - r) \int_0^t u(\xi, r, \lambda) e^{-[ad^2\pi^2(t-\lambda)]/l^2} e^{-[(y-r)^2]/[4a(t-\lambda)]} (t - \lambda)^{-3/2} d\lambda.$$

Setting $\rho = [(y - r)]/[2\sqrt{a(t - \lambda)}]$, then we have

$$\mathcal{U}(\xi, y, t) = 4\sqrt{a} \int_{\frac{y-r}{2\sqrt{at}}}^{\infty} u \left(\xi, r, t - \frac{(y-r)^2}{4a\rho^2} \right) e^{-[d^2\pi^2(y-r)^2]/[4\rho^2r^2]} e^{-\rho^2} d\rho,$$

$$\begin{aligned} \frac{\partial \mathcal{U}(\xi, y, t)}{\partial y} &= 2\sqrt{a} u(\xi, r, 0) e^{-[ad^2\pi^2t]/l^2} e^{-[(y-r)^2]/[4at]} \frac{1}{\sqrt{at}} \\ &+ 2\sqrt{a} \cdot \int_{\frac{y-r}{2\sqrt{at}}}^{\infty} \left[\frac{\partial u}{\partial \lambda} \left(\xi, r, t - \frac{(y-r)^2}{4a\rho^2} \right) \left(-\frac{y-r}{4a\rho^2} \right) e^{-[d^2\pi^2(y-r)^2]/[4\rho^2l^2]} e^{-\rho^2} \right. \\ &\left. + u \left(\xi, r, t - \frac{(y-r)^2}{4a\rho^2} \right) \left(-\frac{d^2\pi^2(y-r)}{\rho^2l^2} \right) e^{-[d^2\pi^2(y-r)^2]/[4\rho^2l^2]} e^{-\rho^2} \right] d\rho. \end{aligned}$$

Returning to the variable λ , we get

$$\frac{\partial \mathcal{U}(\xi, y, t)}{\partial y} = -2 \int_0^t e^{-[(y-r)^2]/[4a(t-\lambda)]} e^{-[d^2\pi^2 a(t-\lambda)]/[l^2]} \times \left[\frac{\partial u}{\partial \lambda}(\xi, r, \lambda) + u(\xi, r, \lambda) \left(\frac{ad^2\pi^2}{l^2} \right) \right] \frac{1}{\sqrt{t-\lambda}} d\lambda.$$

Taking the limit $y \rightarrow +r$, we find from (2.11)

$$\frac{\partial u(x, r, t)}{\partial y} = -\frac{2}{l\sqrt{\pi a}} \sum_{d=1}^{\infty} \sin \frac{d\pi x}{l} \int_0^l \sin \frac{d\pi \xi}{l} \int_0^t e^{-[d^2\pi^2 a(t-\lambda)]/[l^2]} \times \left[\frac{\partial u}{\partial \lambda}(\xi, r, \lambda) + u(\xi, r, \lambda) \frac{ad^2\pi^2}{l^2} \right] \frac{1}{\sqrt{t-\lambda}} d\lambda d\xi. \quad (2.12)$$

Having obtained the exact boundary condition (2.12), we reduce the original problem (1.10)–(1.13) to a problem on the bounded domain $\Omega^0 = \{0 < x < l, 0 < y < r, 0 \leq t < \infty\}$, just as in the 1D case:

$$u_t = a\Delta u, \quad (x, y, t) \in \Omega^0, \quad (2.13)$$

$$u(0, y, t) = u(l, y, t) = 0, \quad 0 < y < r, \quad t > 0, \quad (2.14)$$

$$c_0 u_t - u_y = f(u), \quad y = 0, \quad 0 < x < l, \quad t > 0, \quad (2.15)$$

$$u_y(x, r, t) = -\frac{2}{l\sqrt{\pi a}} \sum_{d=1}^{\infty} \sin \frac{d\pi x}{l} \int_0^l \sin \frac{d\pi \xi}{l} \int_0^t e^{-[d^2\pi^2 a(t-\lambda)]/[l^2]} \times \left[u_{\lambda}(\xi, r, \lambda) + \frac{d^2\pi^2 a}{l^2} u(\xi, r, \lambda) \right] \frac{1}{\sqrt{t-\lambda}} d\lambda d\xi, \quad 0 < x < l, \quad t > 0 \quad (2.16)$$

$$u(x, y, 0) = u_0(x, y), \quad (x, y) \in \bar{D} = (0 \leq x \leq l, 0 \leq y \leq r). \quad (2.17)$$

III. SPACE DISCRETIZATION

In this section, using linear finite element schemes, we construct a second order in space approximations of the problems (2.4)–(2.7) and (2.13)–(2.17).

A. Problem (2.4)–(2.7)

Let V_h is a piecewise linear finite element space, defined on an uniform mesh with size h in $\bar{D} = [0, l]$: $\bar{\omega}_h = \{x_i, x_i = (i-1)h, i = 1, 2, \dots, N; (N-1)h = l\}$. For a discrete function, defined on $\bar{\omega}_h$, we introduce the norms:

$$\|v\|_{L_{\infty}(\bar{\omega}_h)} = \max_{\bar{\omega}_h} |v(x_i)|, \quad \|v\|_{L_2(\bar{\omega}_h)} = \left(\sum_{i=1}^N h v^2(x_i) \right)^{1/2}.$$

The standard finite element discretization of the problem (2.4)–(2.7) is to find

$$\begin{aligned} u^h \in V_h, \quad u^h &= \sum_{i=1}^N U_i(t) \varphi_i(x). \\ c_0 u_t^h(0, t) \varphi(0) + (\varphi, u_t^h) + \frac{1}{\sqrt{\pi}} \left(\int_0^t u_\lambda^h(l, \lambda) \frac{1}{\sqrt{t-\lambda}} d\lambda \right) \varphi(l) - f(u^h(0, t)) \varphi(0) \\ &+ (\nabla u^h, \nabla \varphi) = 0, \quad \forall \varphi \in V_h, \quad \text{where } (u, v) = \int_{\bar{D}} uv dx. \end{aligned} \quad (3.1)$$

Now, doing a mass lumping in (3.1), we obtain for $U_i, i = 1, \dots, N$ the following system of ordinary differential equations (ODEs):

$$\dot{U}_1 = \frac{2}{(h + 2c_0)} \left(f(U_1) + \frac{U_2 - U_1}{h} \right), \quad (3.2)$$

$$\dot{U}_i = \frac{1}{h^2} (U_{i-1} - 2U_i + U_{i+1}), \quad i = 2, \dots, N-1, \quad (3.3)$$

$$\dot{U}_N = -\frac{2}{h} \left(\frac{1}{\sqrt{\pi}} \int_0^t \dot{U}_N(\lambda) \frac{1}{\sqrt{t-\lambda}} d\lambda + \frac{U_N - U_{N-1}}{h} \right). \quad (3.4)$$

B. Problem (2.13)–(2.17)

In the domain Ω^0 we introduce a regular rectangular grid with nodes (x_i, y_j) :

$$\begin{aligned} \bar{\omega}_x &= \{x_i, x_i = (i-1)h_1, \quad i = 1, \dots, N, (N-1)h_1 = l\}, \\ \bar{\omega}_y &= \{y_j, y_j = (j-1)h_2, \quad j = 1, \dots, M, (M-1)h_2 = r\}, \quad \Omega_h^0 = \bar{\omega}_x \times \bar{\omega}_y. \end{aligned}$$

Let introduce the mesh norms

$$\|v\|_{L_\infty(\bar{\Omega}_h^0)} = \max_{\bar{\Omega}_h^0} |v(x_i, y_j)|, \quad \|v\|_{L_2(\bar{\Omega}_h^0)} = \left(\sum_{i=1}^N \sum_{j=1}^M h_1 h_2 v^2(x_i, y_j) \right)^{1/2}.$$

We shall use a linear triangle elements with vertexes: $(x_i, y_j), (x_{i+1}, y_j), (x_{i+1}, y_{j+1})$ and $(x_{i+1}, y_{j+1}), (x_i, y_{j+1}), (x_i, y_j), i = 1, \dots, N-1, j = 1, \dots, M-1$. Let denote the total number of grid nodes by $I = NM$ and the piecewise finite element space, defined on mesh, described above, with V_h .

Now, we can describe the numerical approximation $u_{ij}^h(x_i, y_j, t)$ of $u(x, y, t)$. Using the relation $s = (j-1)N + i, i = 1, \dots, N, j = 1, \dots, M$, the 2D array u_{ij} can be transformed to 1D array $u_s, s = 1, \dots, I$. The standard finite element approximation of the problem (2.13)–(2.17) is to find $u^h \in V_h$,

$$\begin{aligned} u^h &= \sum_{s=1}^I U_s(t) \Phi_s(x, y) \\ (u_t^h, \Phi) + (\nabla u^h, \nabla \Phi) + c_0 \int_0^l u_t^h(x, 0, t) \Phi(x, 0) dx - \int_0^l f(u^h(x, 0, t)) \Phi(x, 0) dx \\ &+ \frac{2}{l\sqrt{\pi}a} \sum_{d=1}^\infty \int_0^l \Phi(x, r) \sin \frac{d\pi x}{l} dx \int_0^l \sin \frac{d\pi \xi}{l} \int_0^t e^{-[d^2\pi^2 a(t-\lambda)]/l^2} \left[u_\lambda^h(\xi, r, \lambda) \right. \\ &\left. + \frac{d^2\pi^2 a}{l^2} u^h(\xi, r, \lambda) \right] \frac{1}{\sqrt{t-\lambda}} d\lambda d\xi = 0, \quad \forall \Phi \in V_h, \quad \text{where } (u, v) = \int_{\bar{D}} uv dx dy. \end{aligned}$$

Consequently, we have

$$\begin{aligned} & \sum_{s=1}^I (\Phi_s, \Phi_i) \dot{U}_s(t) + \sum_{s=1}^I (\nabla \Phi_s, \nabla \Phi_i) U_s(t) + c_0 \sum_{s=1}^I \left[\int_0^l \Phi_s(x, 0) \Phi_i(x, 0) dx \right] \dot{U}_s(t) \\ & - \sum_{s=1}^I \left[\int_0^l \Phi_s(x, 0) \Phi_i(x, 0) dx \right] f(U_s(t)) + \frac{2}{l\sqrt{\pi a}} \sum_{d=1}^{\infty} \int_0^l \Phi_i(x, r) \sin \frac{d\pi x}{l} dx \\ & \quad \times \sum_{s=(M-1)N+1}^I \int_0^l \Phi_s(\xi, r) \sin \frac{d\pi \xi}{l} d\xi \int_0^t e^{-[d^2\pi^2 a(t-\lambda)/l^2]} (\dot{U}_s(\lambda) \\ & \quad + \frac{d^2\pi^2 a}{l^2} U_s(\lambda)) \frac{1}{\sqrt{t-\lambda}} d\lambda = 0, \quad i = 1, \dots, I, \end{aligned}$$

where the so called "product approximation" formula was used:

$$f(u(x, y, t)) \approx \sum_{s=1}^I f(U_s(t)) \Phi_s(x, y).$$

Applying mass lumping, we obtain the following ODEs (written in 2D array for clarity):

$$U_{1j} = U_{Nj} = 0, \quad j = 1, \dots, M, \quad (3.5)$$

$$\dot{U}_{i1} = \frac{ah_2}{h_2 + 2ac_0} \left(\frac{U_{i+1,1} - 2U_{i1} + U_{i-1,1}}{h_1^2} + \frac{2}{h_2} \frac{U_{i2} - U_{i1}}{h_2} + \frac{2}{h_2} f(U_{i1}) \right), \quad (3.6)$$

$$\dot{U}_{ij} = a \frac{U_{i-1,j} - 2U_{ij} + U_{i+1,j}}{h_1^2} + a \frac{U_{i-1,j} - 2U_{ij} + U_{i+1,j}}{h_2^2}, \quad (3.7)$$

$$\begin{aligned} \dot{U}_{iM} &= a \frac{U_{i-1,M} - 2U_{iM} + U_{i+1,M}}{h_1^2} - \frac{2a}{h_2} \frac{U_{iM} - U_{iM-1}}{h_2} \\ & - \frac{4h_1\pi^2 a^2}{h_2 l^3 \sqrt{a\pi}} \sum_{d=1}^{\infty} d^2 \sin \frac{d\pi x_i}{l} \sum_{m=2}^{N-1} \sin \frac{d\pi x_m}{l} \int_0^t e^{-[d^2\pi^2 a(t-\lambda)]/l^2} U_{mM}(\lambda) \frac{1}{\sqrt{t-\lambda}} d\lambda \\ & - \frac{4h_1 a}{h_2 l \sqrt{a\pi}} \sum_{d=1}^{\infty} \sin \frac{d\pi x_i}{l} \sum_{m=2}^{N-1} \sin \frac{d\pi x_m}{l} \int_0^t e^{-[d^2\pi^2 a(t-\lambda)]/l^2} \dot{U}_{mM}(\lambda) \frac{1}{\sqrt{t-\lambda}} d\lambda, \\ & i = 2, \dots, N-1, \quad j = 2, \dots, M-1. \quad (3.8) \end{aligned}$$

IV. FULL DISCRETIZATION

The choice of the variable time step (Δt_n) for numerical solution of the ODEs (3.2)–(3.4) and (3.5)–(3.8) is governed by the growth of the solution. Let t_n denote the n th mesh point. Then

$$t_0 = 0, \quad t_n = t_{n-1} + \Delta t_{n-1} \equiv \sum_{k=0}^{n-1} \Delta t_k, \quad (\Delta t_k > 0, \quad k = 0, 1, 2, \dots).$$

For the sake of simplicity, we shall take below $f(u) = u^p$. In the case of blow-up solutions ($p > 1$), the time increment Δt_k is chosen to be variable as in [12, 15–17]:

$$\Delta t_n = \tau \times \min \left\{ 1, \frac{1}{\|U\|_{L_\infty(\text{or } L_2)}^{p-1}} \right\}, \quad \tau = \max_{0 \leq k \leq n-1} \Delta t_k.$$

The time-step size decreases according to the growth of the numerical solution.

For approximation of the integrals (with respect to t) in (3.8) we shall apply the following modifications of Lemma 1 [6].

Lemma 4.1. *Let $f(t) \in C^2[0, t_n]$ and $E(\lambda) = e^{-[d^2\pi^2 a(t_n-\lambda)]/l^2}$. Then*

$$\left| \int_0^{t_n} f'(\lambda) \cdot E(\lambda) \frac{d\lambda}{\sqrt{t_n - \lambda}} - \sum_{k=1}^n \frac{f(t_k) - f(t_{k-1})}{\Delta t_k} \int_{t_{k-1}}^{t_k} E(\lambda) \frac{d\lambda}{\sqrt{t_n - \lambda}} \right| \leq \frac{1}{6} (10\sqrt{2} - 11) \max_{0 \leq t \leq t_n} |f''(t)| \tau^{3/2}.$$

Lemma 4.1, for $E(\lambda) = 1$ is formulated and proved in [6]. In our case it is similarly.

Lemma 4.2. *Let $f(t) \in C^2[0, t_n]$ and $g(t) = e^{-[d^2\pi^2 a(t_n-\lambda)]/l^2} \frac{1}{\sqrt{t_n-t}}$. Then*

$$\left| \int_0^{t_n} f(\lambda) g(\lambda) d\lambda - \sum_{k=1}^n \frac{f(t_k) + f(t_{k-1})}{2} \int_{t_{k-1}}^{t_k} g(\lambda) d\lambda \right| \leq \tau^{3/2} \frac{l}{ad\sqrt{\pi}} \left[l \max_{0 \leq t \leq t_n} |f'(t)| \frac{1}{d\pi\sqrt{d\pi}} + \frac{\tau^{1/2}}{8} \right].$$

Proof. Again the proof is very similar to those in [6] and that's way we will be short. Using (just as in [6]) the theory of polynomial interpolation, for any $t \in [t_{k-1}, t_k]$, we have

$$\begin{aligned} \int_0^{t_n} f(\lambda) g(\lambda) d\lambda - \sum_{k=1}^n \frac{1}{2} [f(t_k) + f(t_{k-1})] \int_{t_{k-1}}^{t_k} g(\lambda) d\lambda \\ = \sum_{k=1}^n \frac{1}{\Delta t_k} [f(t_k) + f(t_{k-1})] \int_{t_{k-1}}^{t_k} (\lambda - t_{k-1/2}) g(\lambda) d\lambda \\ + \frac{1}{2} \sum_{k=1}^n \int_{t_{k-1}}^{t_k} f''(\xi) (\lambda - t_k) (\lambda - t_{k-1}) g(\lambda) d\lambda, \quad \text{where } \xi_k \in (t_{k-1}, t_k). \end{aligned}$$

Next,

$$\begin{aligned} \left| \int_0^{t_n} f(\lambda) g(\lambda) d\lambda - \sum_{k=1}^n \frac{1}{2} [f(t_k) + f(t_{k-1})] \int_{t_{k-1}}^{t_k} g(\lambda) d\lambda \right| \\ \leq \max_{0 \leq t \leq t_n} |f'(t)| \sum_{k=1}^n \left| \int_{t_{k-1}}^{t_k} (\lambda - t_{k-1/2}) g(\lambda) d\lambda \right| + \frac{1}{8} \tau^2 \max_{0 \leq t \leq t_n} |f''(t)| \sum_{k=1}^n \int_{t_{k-1}}^{t_k} g(\lambda) d\lambda \end{aligned}$$

$$\begin{aligned}
&< \max_{0 \leq t \leq t_n} |f'(t)| \left| \sum_{k=1}^n \left| \int_{t_{k-1}}^{t_k} \frac{(\lambda - t_{k-1/2})l^2}{d^2 \pi^2 a(t_n - \lambda)} \frac{d\lambda}{\sqrt{t_n - \lambda}} \right| + \tau^2 \frac{l}{8d\sqrt{a\pi}} \right| \\
&\leq \tau^{3/2} \frac{l}{ad\sqrt{\pi}} \left[l \max_{0 \leq t \leq t_n} |f'(t)| \frac{1}{d\pi\sqrt{d\pi}} + \frac{\tau^{1/2}}{8} \right]. \quad \blacksquare
\end{aligned}$$

Now, from (3.2)–(3.4) using implicit Euler's method and Lemma 4.1, we obtain the following discretization of the problem (2.4)–(2.7):

$$\begin{aligned}
\left(\frac{1}{\Delta t_{n-1}} + \frac{2}{h^2} + \frac{2c_0}{h\Delta t_{n-1}} - \frac{2(U_1^{n-1})^{p-1}}{h} \right) U_1^n - \frac{2}{h^2} U_2^n &= \left(\frac{1}{\Delta t_{n-1}} + \frac{2c_0}{h\Delta t_{n-1}} \right) U_1^{n-1}, \\
-\frac{1}{h^2} U_{i-1}^n + \left(\frac{2}{h^2} + \frac{1}{\Delta t_{n-1}} \right) U_i^n - \frac{1}{h^2} U_{i+1}^n &= \frac{1}{\Delta t_{n-1}} U_i^{n-1}, \quad i = 2, \dots, N-1, \\
\left(\frac{h}{2\Delta t_{n-1}} + \frac{2}{\sqrt{\pi}\Delta t_{n-1}} + \frac{1}{h} \right) U_N^n - \frac{1}{h} U_{N-1}^n &= \left(\frac{h}{2\Delta t_{n-1}} + \frac{2}{\sqrt{\pi}\Delta t_{n-1}} \right) U_N^{n-1} \\
&\quad - \frac{1}{2\sqrt{\pi}} \sum_{k=1}^{n-1} \frac{\sqrt{t_n - t_{k-1}} - \sqrt{t_n - t_k}}{\Delta t_{k-1}} (U_N^k - U_N^{k-1}).
\end{aligned}$$

If $\hat{x} > l$, then from (2.3) we have

$$u(\hat{x}, t_n) = \frac{1}{2} \sum_{k=1}^n \left[\operatorname{erf} \left(\frac{\hat{x} - l}{2\sqrt{t_n - t_{k-1}}} \right) - \operatorname{erf} \left(\frac{\hat{x} - l}{2\sqrt{t_n - t_k}} \right) \right] (U_N^k - U_N^{k-1}),$$

where $\operatorname{erf}(x) = 2/\sqrt{\pi} \int_0^x e^{-\rho^2} d\rho$.

By analogy, using Lemmas 4.1 and 4.2 the full discretization of (3.5)–(3.8), can be written as follows:

$$U_{1j}^n = U_{Nj}^n = 0, \quad j = 1, \dots, M, \quad (4.1)$$

$$(1 + 2(v_1 + v_2) - 2v_2 h_2 (U_{i1}^{n-1})^{p-1}) U_{i1}^n - v_1 (U_{i+1,1}^n + U_{i-1,1}^n) - 2v_2 U_{i2}^n = U_{i1}^{n-1}, \quad (4.2)$$

$$(1 + 2(\mu_1 + \mu_2)) U_{ij}^n - \mu_1 (U_{i-1,j}^n + U_{i+1,j}^n) - \mu_2 (U_{i,j-1}^n + U_{i,j+1}^n) = U_{ij}^{n-1}, \quad (4.3)$$

$$\begin{aligned}
&(1 + 2(\mu_1 + \mu_2) + H \cdot F f_i(x_i, P_+)) U_{iM}^n + (-\mu_1 + H \cdot F f_i(x_{i-1}, P_+)) U_{i-1,M}^n \\
&\quad + (-\mu_1 + H \cdot F f_i(x_{i+1}, P_+)) U_{i+1,M}^n - 2\mu_2 U_{i,M-1}^n + H \left(\sum_{m=2}^{i-2} + \sum_{m=i+2}^{N-1} \right) F f_i(x_m, P_+) U_{mM}^n \\
&= H \sum_{m=2}^{N-1} F f_i(x_m, P_-) U_{mM}^{n-1} \\
&\quad - H \cdot P_2 \Delta t_n \sum_{d=1}^{\infty} d \sin \frac{d\pi x_i}{l} \sum_{m=2}^{N-1} \sin \frac{d\pi x_m}{l} \sum_{\substack{k=1 \\ n>1}}^{n-1} \operatorname{Erf}(d, k) (U_{mM}^k + U_{mM}^{k-1})
\end{aligned}$$

$$\begin{aligned}
 & -H \cdot P_1 \Delta t_n \sum_{d=1}^{\infty} \frac{1}{d} \sin \frac{d\pi x_i}{l} \sum_{m=2}^{N-1} \sin \frac{d\pi x_m}{l} \sum_{k=1}^{n-1} \operatorname{Erf}(d, k) \frac{U_{mM}^k - U_{mM}^{k-1}}{\Delta t_k} + U_{iM}^{n-1}, \\
 & \quad n > 1 \\
 & \quad i = 2, \dots, N-1, \quad j = 2, \dots, M-1, \quad (4.4)
 \end{aligned}$$

where

$$\begin{aligned}
 \mu_i &= \Delta t_n \frac{a}{h_i^2}, \quad v_i = \Delta t_n \frac{ah_2}{h_i^2(h_2 + 2ac_0)}, \quad i = 1, 2, \quad H = 4 \frac{h_1}{h_2}, \\
 P_1 &= \frac{1}{\pi}, \quad P_2 = \frac{a\pi}{2l^2}, \quad P_{\pm} = \frac{1}{d} P_1 \pm d \Delta t_n P_2, \\
 Ff_i(x_j, P) &= \sum_{d=1}^{\infty} P \cdot \operatorname{erf} \left(\frac{d\pi \sqrt{a\Delta t_n}}{l} \right) \sin \frac{d\pi x_i}{l} \sin \frac{d\pi x_j}{l}, \\
 \operatorname{Erf}(d, k) &= \operatorname{erf} \left(\frac{d\pi \sqrt{a(t_n - t_{k-1})}}{l} \right) - \operatorname{erf} \left(\frac{d\pi \sqrt{a(t_n - t_k)}}{l} \right).
 \end{aligned}$$

If $(\hat{x}, \hat{y}) \in \Omega^l$, then from (2.11) we obtain

$$\begin{aligned}
 (\hat{x}, \hat{y}, t_n) &= \frac{\hat{y} - r}{4l\sqrt{a\pi}} \sum_{d=1}^{\infty} \sin \frac{d\pi \hat{x}}{l} \sum_{m=2}^{N-1} \sin \frac{d\pi x_m}{l} \sum_{k=1}^n \Delta t_k (Q(t_k) - Q(t_{k-1})) (U_{iM}^k - U_{iM}^{k-1}), \\
 Q(\lambda) &= e^{-[d^2\pi^2 a(t_n - \lambda)]/[l^2]} e^{-[(\hat{y} - r)^2]/[4a(t_n - \lambda)]} (t_n - \lambda)^{-(3/2)}.
 \end{aligned}$$

Next, we are going to study the solvability of the numerical scheme (4.1)–(4.4). With that end in view, we rewrite the system (4.1)–(4.4) in matrix form

$$\begin{aligned}
 C_1 Y_1 - B Y_2 &= F_1, \\
 -A Y_{j-1} + C Y_j - A Y_{j+1} &= F_j, \quad j = 2, \dots, M-1, \\
 -A_M Y_{M-1} - C_M Y_M &= F_M,
 \end{aligned}$$

where

$$\begin{aligned}
 Y_j &= [U_{1j}^n, \dots, U_{ij}^n, \dots, U_{Nj}^n]^T, \quad Y_j, F_j \in \mathbf{R}^N, \quad j = 1, \dots, M; \\
 C_1, B, A, C, A_M, C_M &\in \mathbf{R}^{N \times N},
 \end{aligned}$$

and

- $C_1 = \operatorname{tridiag}[C_{1(-1)}, C_{1(0)}, C_{1(+1)}]$, $C_{1(-1)}, C_{1(+1)} \in \mathbf{R}^{N-1}$, $C_{1(0)} \in \mathbf{R}^N$,
 $C_{1(-1)} = v_1 \{-1, \dots, -1, 0\}$,
 $C_{1(+1)} = v_1 \{0, -1, \dots, -1\}$,
 $C_{1(0)} = \{1, 1 + 2(v_1 + v_2) - 2v_2 h_2 (Y_1^{n-1}(2))^{p-1}, \dots, 1$
 $+ 2(v_1 + v_2) - 2v_2 h_2 (Y_1^{n-1}(N-1))^{p-1}, 1\}$;

- $B = 2\nu_2 \text{diag}[0, 1, \dots, 1, 0]$;
- $A = \mu_2 \text{diag}[0, 1, \dots, 1, 0]$;
- $C = \text{tridiag}[C_{(-1)}, C_{(0)}, C_{(+1)}]$, $C_{(-1)}, C_{(+1)} \in \mathbf{R}^{N-1}$, $C_{(0)} \in \mathbf{R}^N$,
 $C_{(-1)} = \mu_1\{-1, \dots, -1, 0\}$,
 $C_{(+1)} = \mu_1\{0, -1, \dots, -1\}$,
 $C_{(0)} = \{1, 1 + 2(\mu_1 + \mu_2), \dots, 1 + 2(\mu_1 + \mu_2), 1\}$;
- $A_M = 2A$,
- $C_M = \begin{pmatrix} 1 & \mathbf{0} & 0 \\ \mathbf{0} & \boxed{C^*} & \mathbf{0} \\ 0 & \mathbf{0} & 1 \end{pmatrix}$, $C^* \in \mathbf{R}^{(N-2) \times (N-2)}$, $C^* = \{c_{ij}\}$, $i, j = 2, \dots, N-2$,

$$c_{ij} = \begin{cases} 1 + 2(\mu_1 + \mu_2) + H \cdot Ff_i(x_j, P_+), & j = i, \\ -\mu_1 + H \cdot Ff_i(x_j, P_+), & j = i \pm 1, \\ H \cdot Ff_i(x_j, P_+), & \text{otherwise.} \end{cases}$$
- $F_j = [0, U_{2j}^{n-1}, \dots, U_{ij}^{n-1}, \dots, U_{N-1,j}^{n-1}, 0]^T$, $j = 1, \dots, M-1$;
- $F_M = [0, FF(x_2), \dots, FF(x_i), \dots, FF(x_{N-1}), 0]^T$,

where with $FF(x_i)$ we denote the term in the right-hand side of (4.4).

The matrix \mathbf{C} (submatrix \mathbf{C}^*) has the following properties:

- Symmetry, because of the symmetry of the term $Ff_i(x_j, P_+)$.
- Diagonal domination.

To show the **b** property, we shall estimate the convergence of the series $Ff_i(x_j, P_+)$.

Let us rearrange $Ff_i(x_j, P_+)$ as follows:

$$\begin{aligned} Ff_i(x_j, P_+) &= \frac{\sqrt{a\pi}}{l} \sum_{d=1}^{\infty} \left(\Delta t_n \frac{a\pi}{2l^2} + \frac{1}{\pi d^2} \right) \sin \frac{\pi dx_i}{l} \sin \frac{\pi dx_j}{l} d \int_0^{[\pi d \sqrt{a \Delta t_n}] / [l]} e^{-\rho^2} d\rho \\ &< \frac{\sqrt{a\pi}}{l} \sum_{d=1}^{\infty} \left(\Delta t_n \frac{a\pi}{2l^2} + \frac{1}{\pi d^2} \right) d \int_0^{[\pi d \sqrt{a \Delta t_n}] / [l]} e^{-\rho^2} d\rho. \end{aligned}$$

After the change of variable $\rho = k/d^\alpha$, $\alpha > 2$ (integer), we find

$$\begin{aligned} \frac{\sqrt{a\pi}}{l} \sum_{d=1}^{\infty} \left(\Delta t_n \frac{a\pi}{2l^2} + \frac{1}{\pi d^2} \right) \frac{1}{d^{\alpha-1}} \int_0^{[\pi d^{\alpha-1} \sqrt{a \Delta t_n}] / [l]} e^{-[k^2] / [d^{2\alpha}]} dk \\ < \frac{\sqrt{a\pi}}{l} \sum_{d=1}^{\infty} \left(\Delta t_n \frac{a\pi}{2l^2} + \frac{1}{\pi d^2} \right) \frac{1}{d^{\alpha-1}} \int_0^{[\pi d^{2\alpha} \sqrt{a}] / [\Delta t_n l]} e^{-[k^2] / [d^{2\alpha}]} dk. \end{aligned}$$

Next, letting $k = d^{2\alpha}(t_n - \lambda)^{1/2}$ we obtain for the last expression

$$\begin{aligned} & \frac{\sqrt{a\pi}}{2l} \sum_{d=1}^{\infty} \left(\Delta t_n \frac{a\pi}{2l^2} + \frac{1}{\pi d^2} \right) \frac{1}{d^{\alpha-1}} \int_{t_n}^{t_n - [l^2 \Delta t_n^2] / [a\pi^2]} e^{-[d^{2\alpha}] / [t_n - \lambda]} d^{2\alpha} (t_n - \lambda)^{-(3/2)} d\lambda \\ & < \frac{\sqrt{a\pi}}{2l} \sum_{d=1}^{\infty} \left(\Delta t_n \frac{a\pi}{2l^2} + \frac{1}{\pi d^2} \right) \frac{1}{d^{\alpha-1}} \int_{t_n}^{t_n - \frac{l^2 \Delta t_n^2}{a\pi^2}} (t_n - \lambda)^{-\frac{1}{2}} d\lambda \\ & = \frac{\Delta t_n}{\sqrt{\pi}} \left(\Delta t_n \frac{a\pi}{2l^2} \sum_{d=1}^{\infty} \frac{1}{d^{\alpha-1}} + \frac{1}{\pi} \sum_{d=1}^{\infty} \frac{1}{d^{\alpha+1}} \right) \stackrel{\alpha=3}{<} \Delta t_n \frac{\pi\sqrt{\pi}}{6} \left(\Delta t_n \frac{a\pi}{2l} + \frac{1}{15} \right). \end{aligned}$$

Therefore for $\Delta t_n < Ch_2$, the property **b** is fulfilled.

At this stage, it became clear that there exist the matrixes C^{-1} , C_1^{-1} , C_M^{-1} . Further, we have

$$\begin{aligned} 2 \quad \|C^{-1}A\| & < 2\|C^{-1}\| \cdot \|A\| = 2 \frac{\max(\lambda_A)}{\min(\lambda_C)} \\ & = \frac{2 \max\{0, \mu_2\}}{\min \left\{ 1, 1 + 2(\mu_1 + \mu_2) + 2\mu_1 \cos \frac{s\pi}{N} \right\}_{s=2, \dots, N-1}} = 2\mu_2, \\ \|C_1^{-1}B\| & < \|C_1^{-1}\| \cdot \|B\| = \frac{\max(\lambda_B)}{\min(\lambda_{C_1})} \\ & = \frac{\max\{0, 2\nu_2\}}{\min \left\{ 1, 1 + 2(\nu_1 + \nu_2) - \frac{2a\tau}{h_2 + 2ac_0} + 2\nu_1 \cos \frac{s\pi}{N} \right\}_{s=2, \dots, N-1}} = 4\nu_2, \\ \|C_M^{-1}A_M\| & < \|C_M^{-1}\| \cdot \|A_M\| = \frac{\max(\lambda_{A_M})}{\min(\lambda_{C_M})} \\ & < \frac{\max\{0, 2\mu_2\}}{\min \left\{ 1, 1 + 2(\mu_1 + \mu_2) + 2\mu_1 \cos \frac{s\pi}{N} \right\}_{s=2, \dots, N-1}} = 2\mu_2, \end{aligned}$$

and from Lemma 5 on p 107 in [18] follows that the numerical method is stable. This results we formulate as a theorem.

Theorem 4.3. *Let $\Delta t_n < Ch_2$, $n = 0, 1, \dots$, where C is a certain constant, independent of h_1 and h_2 . Then the matrix form of Thomas algorithm [applied to (4.1)–(4.4)] is correct and stable.*

V. COMPUTATIONAL RESULTS

A series of computational tests were performed to study numerically the problems, described in Section I. Two kinds of experiments are considered. The first one discusses the accuracy and the convergence rate; the second one simulates global and blow-up solutions of the problems (1.1)–(1.4), (1.10)–(1.13).

A. One-dimensional Experiments

Example 1. Let $f(t)$ be chosen such that the problem

$$\begin{aligned} u_t &= u_{xx}, & x > 0, \quad 0 \leq t < \infty, \\ c_0 u_t - u_x &= u^p + f(t), & x = 0, \quad 0 \leq t < \infty, \\ u_x &= -\frac{1}{\sqrt{\pi}} \int_0^t u_\lambda(l, \lambda) \frac{1}{\sqrt{t-\lambda}} d\lambda, & x = l, \quad 0 \leq t < \infty, \quad u \rightarrow \infty \text{ when } x \rightarrow \infty, \\ u(x, 0) &= 0, & 0 \leq x < \infty, \end{aligned}$$

has exact solution $u(x, t) = \operatorname{erfc}[(x+2)]/[2\sqrt{t}]$, where $\operatorname{erfc}(x) = 2/\sqrt{\pi} \int_x^\infty e^{-\rho^2} d\rho$. $f(t)$ is different for $c_0 = 0$ and $c_0 = 1$. Our first example is designed to illustrate the convergence rate of the problem, given above, for $p = 3$. The ratio $\tau/h^2 = 1$ is fixed. In Table I, the errors under different norms and convergence rate at $t = 1$ are presented. The solution is computed in the interval $[0, 1]$. The errors are defined as follows:

$$E_\infty^N = \|u - U\|_{L_\infty(\bar{\omega}_h)} \quad (5.1)$$

$$E_2^N = \|u - U\|_{L_2(\bar{\omega}_h)}. \quad (5.2)$$

The convergence rate is computed, using the formula

$$CR = \log_2 \frac{E_{2(or \infty)}^N}{E_{2(or \infty)}^{2N}}. \quad (5.3)$$

The convergence rate is $O(\tau + h^2)$. For this example, we may take any line $x = l > 0$ as the artificial boundary.

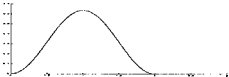
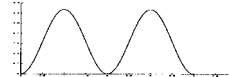
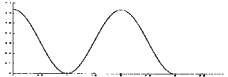
Example 2. Here we test the critical exponents of Fujita type for the problem (1.1)–(1.5) [with (1.2) or (1.2')], $f(u) = u^p$, using artificial boundary condition. Computations are performed for $h = 0.02$, $\tau = 0.05$. The initial functions are chosen such that

1. The compatibility condition (1.4) is fulfilled.
2. $\operatorname{supp} u_0(x) = [0, L]$, $L < \infty$.
3. $u_0(x)$ is nonnegative.

TABLE I. Errors in different norms and convergence rate.

N/CR	$c_0 = 0$		$c_0 = 1$	
	E_∞^N	E_2^N	E_∞^N	E_2^N
11	3.677291e-4	3.277388e-4	6.362449e-4	6.432701e-4
21	9.044109e-5	7.901208e-5	1.587495e-4	1.569222e-4
CR	2.0236	2.0524	2.0028	2.0354
41	2.247152e-5	1.945045e-5	3.952951e-5	3.871844e-5
CR	2.0088	2.0223	2.0057	2.0189
81	5.602326e-6	4.809982e-6	9.866635e-6	9.665531e-6
CR	2.0040	2.0157	2.0023	2.0021
161	1.398738e-6	1.194935e-6	2.463754e-6	2.413872e-6
CR	2.0019	2.0091	2.0017	2.0015

TABLE II. Global existence (GE) or blow-up set ($\{\cdot\}$) of the solution.

<div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  $u_0 = \begin{cases} \frac{1-\cos(\pi x)}{J\pi}, & x \in [0, 2], \\ 0, & x \in [2, \infty). \end{cases}$ $D^* = [0, 3], \quad l = 2.4$ </div> <div style="text-align: center;">  $u_0 = \begin{cases} \frac{1-\cos(\pi x)}{J\pi}, & x \in [0, 4], \\ 0, & x \in [4, \infty). \end{cases}$ $D^* = [0, 5], \quad l = 4.4$ </div> <div style="text-align: center;">  $u_0 = \begin{cases} \frac{1+\cos(\pi x)}{J\pi}, & x \in [0, 3], \\ 0, & x \in [3, \infty). \end{cases}$ $D^* = [0, 4], \quad l = 3.4$ </div> </div>								
p	$c_0 = 0 \text{ or } 1$ $J = 1$	$c_0 = 1$ $J = \frac{1}{5}$	$c_0 = 0 \text{ or } 1$ $J = 1$	$c_0 = 1$ $J = \frac{1}{5}$	$c_0 = 0$ $J = 1$	$c_0 = 1$ $J = 1$	$c_0 = 1$ $J = \frac{1}{5}$	$c_0 = 1$ $J = 2$
0.5	GE	GE	GE	GE	GE	GE	GE	GE
1	GE	GE	GE	GE	GE	GE	GE	GE
1.5	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$
2	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$
3	GE	$\{0\}$	GE	$\{0\}$	GE	$\{0\}$	$\{0\}$	GE
4	GE	$\{0\}$	GE	$\{0\}$	GE	GE	$\{0\}$	GE
5	GE	$\{0\}$	GE	$\{0\}$	GE	GE	$\{0\}$	GE

4. In accordance with R1-R3, i.e. there exist positive constants $k, A, b \ll 1, \alpha < \frac{1}{4}$ such that the inequalities (1.6) [or (1.7)], (1.8) are fulfilled, for corresponding value of p . Denote with $D^* = [0, L_e]$ the domain in which we compute the solution, $0 < L < l < L_e$. We take small enough initial functions, such that the numerical solution of the problem (1.1)–(1.5), $c_0 = 0$ exists globally for $p > 2$. Then we compute the solution with the same initial data, but $1 < p < 2$. The results are given in Table II. It is clear that $p_0 = 1, p_c = 2$.

For the problem with dynamical boundary condition ($c_0 = 1$), the results are also in Table II. For the first two initial functions obviously $p_0 = 1$ and $p_c = 2$. For the third example we can not conclude that $p_c = 3$ (although the numerical solution blows-up for $J = 1$ and $p = 3$), because each problem has its own measure for “small” u_0 (see the case $J = 2$). Also the criterion (1.7), (1.8) is different for problems with dynamical boundary condition. Surely, for large enough initial functions and $p > 1$ the numerical solutions of problem (1.1)–(1.5), blow-up in finite time.

In Fig. 1, we show the evolution graphics of the numerical solution of the problem (1.1)–(1.5), (1.2') computed using artificial boundary condition, $L = 4, l = 4.4, D^* = [0, 5], h = 0.02, \tau = 0.05$. The initial function is the second one from Table II, $J = 1/5$.

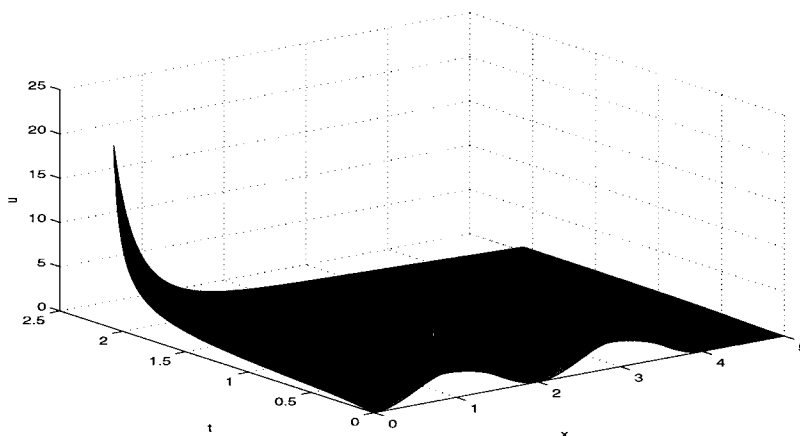
B. 2D Experiments

Example 3. The aim of this example is to test the convergence rate of the numerical scheme (4.1)–(4.4) with respect to the space variable. The test problem is (1.10)–(1.13), where instead of (1.12), we take $u(x, 0, t) = 0, 0 < x < l, t > 0, l = r = a = 1$,

$$u_0(x, y) = \begin{cases} \sin(\pi x) \sin(\pi y), & 0 \leq y \leq 1, \quad 0 \leq x \leq 1, \\ 0, & y > 1, \quad 0 \leq x \leq 1. \end{cases}$$

The exact solution of the problem in the domain

$$\bar{\Omega}^0 = \{0 \leq x \leq 1, \quad 0 \leq y \leq 1, \quad 0 \leq t < \infty\}$$

FIG. 1. Evolution graphic of the solution, $p = 2$, $t = 2.07272463798456$, $c_0 = 1$.

is $u(x, y, t) = e^{-2\pi^2 t} \sin(\pi x) \sin(\pi y)$. Let $h_1 = h_2 = h$, $\tau/h^2 = 1$ is fixed. We use (5.1)–(5.3) for error estimate and convergence rate, respectively. In Table III we show the errors in different discrete norms [(5.1), (5.2)] and convergence rate at $t = 0.1$. The numerical scheme is (4.1)–(4.4), where instead of (3.6), we have $U_{i1} = 0$, $i = 2, \dots, N - 1$. For numerical implementation, the infinite sum is truncated at a fixed number of terms, say S , so it becomes $\sum_{d=1}^S$, $S = 20$ for this example. The error and convergence rate are computed, using (5.1)–(5.3), but instead of $\bar{\omega}_h$ we take Ω_h^0 .

Obviously, the convergence rate is $O(\tau + h^2)$.

Example 4. Now, we shall observe the blow-up behavior of the numerical solution of the problem (2.13)–(2.17), $l = 1$, $r = 1.1$, computed with numerical scheme (4.1)–(4.4), $N = 41$, $M = 45$, $\tau = 0.001$, $S = 20$. The initial function u_0 is

$$u_0(x, y) = \begin{cases} \bar{u}_0, & 0 \leq y \leq 1, \quad 0 \leq x \leq 1, \\ 0, & y > 1, \quad 0 \leq x \leq 1. \end{cases}$$

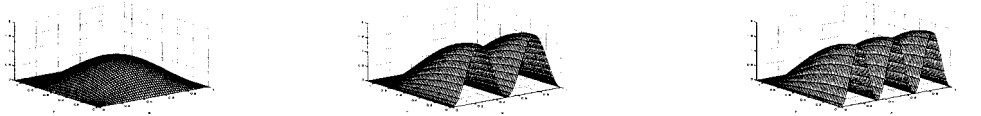
In Table IV we present the blow-up results, depending on the values of p , c_0 for different initial functions.

We observe the phenomena blow-up of the numerical solution of the problem (2.13)–(2.17) in the case of big enough initial function or p . Blow-up set consists of points, situated on the boundary with nonlinear boundary condition.

TABLE III. Errors and convergence rate of (3.9)–(4.4).

N	E_∞^N	CR	E_2^N	CR
11	3.011686e-2		1.530150e-2	
21	7.990291e-3	1.9143	3.981159e-3	1.9424
41	2.003096e-3	1.9960	1.000571e-3	1.9924
81	5.017121e-4	1.9973	2.508894e-4	1.9957

TABLE IV. Global existence (GE) or blow-up set $((x_1, y_1), (x_2, y_2), \dots)$ of the solution.

					
\tilde{u}_0	$\sin(\pi x) \sin(\pi y)$	$(\cos(4\pi(x - \frac{1}{4})) + 1) \cos \frac{\pi y}{2}$		$(\cos(6\pi(x - \frac{1}{6})) + 1) \cos \frac{\pi y}{2}$	
p	$c_0 = 0$ or $c_0 = 1$	$c_0 = 0$	$c_0 = 1$	$c_0 = 0$	$c_0 = 1$
1	GE	GE	GE	GE	GE
2	GE	GE	GE	GE	GE
3	GE	GE	GE	GE	GE
4	GE	$\{(0.25, 0), (0.75, 0)\}$	GE	$\{0.5, 0\}$	GE
5	GE	$\{(0.25, 0), (0.75, 0)\}$	$\{(0.25, 0), (0.75, 0)\}$	$\{0.5, 0\}$	$\{0.5, 0\}$
6	GE	$\{(0.25, 0), (0.75, 0)\}$	$\{(0.25, 0), (0.75, 0)\}$	$\{0.5, 0\}$	$\{0.5, 0\}$
10	GE	$\{(0.25, 0), (0.75, 0)\}$	$\{(0.25, 0), (0.75, 0)\}$	$\{0.5, 0\}$	$\{0.5, 0\}$

C. Efficiency Discussion

The proposed semi-discrete numerical schemes (3.2)–(3.4) and (3.5)–(3.8) include convolution integrals of types

$$J_1(t) = \int_0^t H(t - \lambda) U_{AB}(\lambda) d\lambda \quad \text{and} \quad J_2(t) = \int_0^t H(t - \lambda) \dot{U}_{AB}(\lambda) d\lambda,$$

where the kernel $H(t - \lambda) = 1/(\sqrt{t - \lambda})$ in 1D case and $H(t - \lambda) = e^{-[d^2\pi^2 a(t-\lambda)]/l^2} 1/(\sqrt{t - \lambda})$ in 2D case; U_{AB} and \dot{U}_{AB} are the values of U and \dot{U} , respectively, at the artificial boundary.

Due to this terms, which makes the problem nonlocal in time and the interaction of the integrals and different terms, the solution process involves, at any given time step, the history of U_{AB} and t . Thus, there is a potential need to store and operate on the entire history of the numerical solution. To cope with this problem in many articles (see [19, 20], for example) is obtained a recursive formula for $J_1(t_n)$ and $J_2(t_n)$, which involves only values at the current and previous time step, and not the entire history. Then this formula is incorporated into a numerical scheme, which then become local in time.

Another approach, [19], is to split systems ODEs (3.2)–(3.4) and (3.5)–(3.8) into integro-differential systems, involving new unknown functions $J_1(t_n)$ and $J_2(t_n)$.

Unfortunately, we failed to find such alternative relations, because of the singularity of the integral kernel $H(t - \lambda)$. This problem stays open. Instead, to obtain the full discretizations of the problems, we approximate U_{AB} and \dot{U}_{AB} according to Lemma 4.1 and Lemma 4.2 and then calculating the integrals exactly. It's worth to note that the CPU time of computing $erf(x)$ integral is about the same as the CPU time of the $\sin(x)$ or $\cos(x)$ function. More over, the convolution concerns only one point (1D case) or $N - 2$ points (2D case) along the artificial boundary.

An alternative approach for solving numerically the problems (1.1)–(1.4) and (1.10)–(1.13) is to use standard finite element schemes with no artificial boundary conditions but with a sufficiently long domain and the condition $u(L_e, t) = 0$ (1D case) or $u(x, R_e, t) = 0$ (2D case) on the remote boundary L_e or R_e .

Some natural questions arise

- How long this domain should be?
- If $\text{supp}(U^n)$ increases in time (just as in our test example), shall we enlarge the computational domain?

TABLE V. Max errors and CPU times (sec) for different time levels, $\tau = h = 0.025$

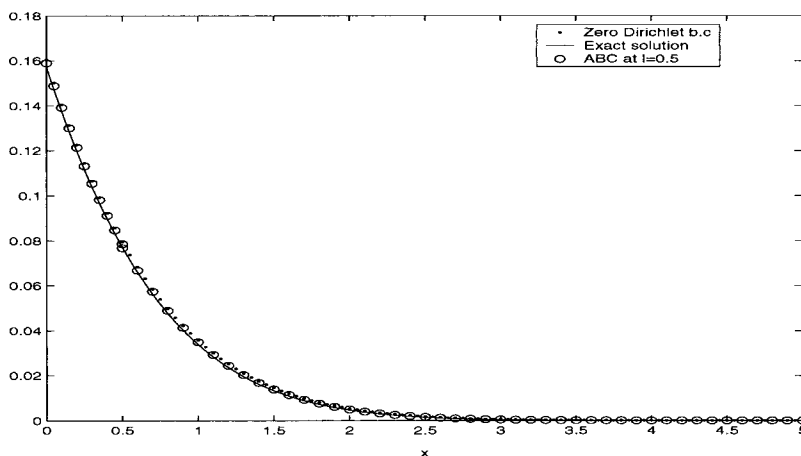
t	Artificial boundary condition at $l = 0.5$		Zero Dirichlet boundary condition at L_e			
	E_{∞}^{41}	CPU time	$L_e = 5$		$L_e = 10$	
			E_{∞}^{201}	CPU time	E_{∞}^{401}	CPU time
0.1	5.944420e-5	0.047	6.103829e-5	0.078	6.103828e-5	0.172
1	1.659975e-3	0.188	3.122574e-3	0.797	3.122574e-3	1.609
5	1.604122e-3	1.703	2.685670e-2	3.969	9.631096e-3	8.406
10	1.874281e-2	6.230	1.175249e-1	9.187	6.459017e-2	16.687

- Which method is cheaper (CPU time): using zero Dirichlet boundary condition and computing the solution on large region or construct artificial boundary condition and operate in very small region, but storing the information along the artificial boundary for all times since $t = 0$ and re-processing this information at each time step?
- Is the accuracy of the numerical solution, computed by imposing the zero Dirichlet boundary condition, as high as the one obtained by constructing an artificial boundary condition?

In Table V are given the max errors and CPU times (sec) of the solution of the test problem from Example 1, $c_0 = 1$, computed in interval $[0, L_e]$, using zero Dirichlet boundary condition and artificial boundary condition (ABC) at point $l = 0.5$ for different times. The solution in the counterpart domain $[l, L_e]$ is obtained by formula, presented in Section IV. The mesh density is fixed: $\tau = h = 0.025$.

In Figs. 2–4 are plotted the graphics of the numerical solutions in interval $[0, 5]$ at times $t = 1$, $t = 5$, and $t = 10$, respectively, compared with exact solution of the problem. Therefore, imposing the condition $u(L_e, t) = 0$, even for large computational domain, the high precision of the artificial boundary condition method (applicable to a very small region) can not be reached and procedure become prohibitively expensive.

For 2D case the convolution integrals are involved for $N - 2$ points from artificial boundary. Thus, advancing in time, each time loop become larger and delay in time proportional to N . For

FIG. 2. Numerical and exact solutions in $D^* = [0, 5]$, $t = 1$.

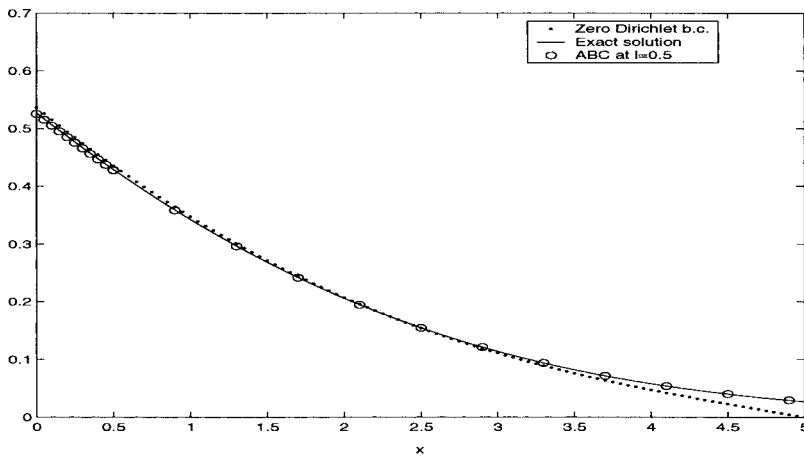


FIG. 3. Numerical and exact solutions in $D^* = [0, 5]$, $t = 5$.

example, if $N = 11$, the delay at each time loop is ≈ 0.18 sec, if $N = 21$, this delay is ≈ 0.8 sec, for $N = 41$: ≈ 3.1 sec i.e., when N increases two times, the CPU delay at each time cycle increases $\approx 2^2$ times.

For short time computations (or for large τ) or no blow-up solutions, this problem is surmountable, because we can reach a high accuracy even by operating on very small region and small N . When the solution grows rapidly, then blow-up phenomena develop for a short time, and the method is not so expensive. The worse case is when the solutions goes to infinity very slowly. Then a long time computations are required. On the other hand, the blow-up set is situated on the boundary $y = 0$ (where the nonlinear dynamical boundary condition is imposed) and the mesh must be very fine around this region. In such case, the *local mesh refinement technique* [21] is very effective. For similar problem, defined in bounded domain, the efficiency of this approach is presented in [22]. Let the coarse mesh step size is $h_1 = h_2 = 0.1$ and the fine mesh step size is $k_1 = k_2 = 0.025$, i.e., $M = 14$, $N = 41$, if $0 \leq y \leq 0.1$ and $N = 11$, if $0.1 \leq y \leq r = 1$.

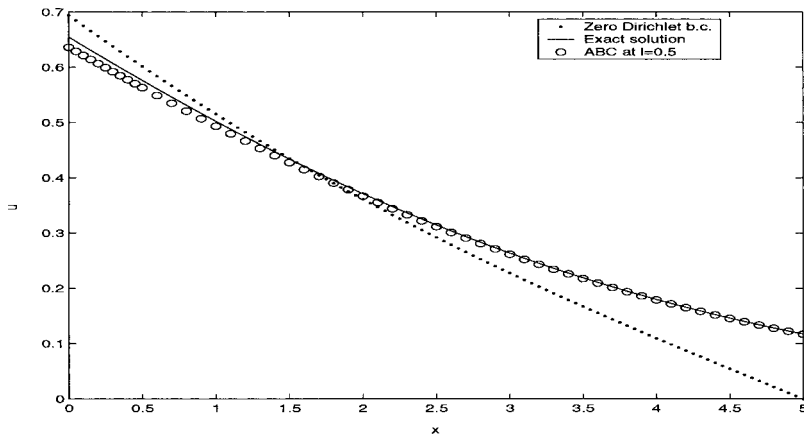


FIG. 4. Numerical and exact solutions in $D^* = [0, 5]$, $t = 10$.

TABLE VI. CPU times (sec) of the computations in domain $[0, 1] \times [0, 3]$, $S = 20$.

Time level	Artificial boundary condition at $r = 1$		Zero Dirichlet boundary condition at R_e	
	<i>Coarse-fine mesh</i>			
	$h_1 = h_2 = 0.1$ $k_1 = k_2 = 0.025$	<i>Uniform mesh</i> $h_1 = h_2 = 0.025$	<i>Uniform mesh</i> $h_1 = h_2 = 0.1$	<i>Uniform mesh</i> $h_1 = h_2 = 0.025$
50	2.317e+2	4.137e+3	3.157	4.580e+3
100	8.723e+2	1.572e+4	5.157	7.060e+3
150	2.013e+3	3.414e+4	7.141	9.560e+3
500	2.175e+4	3.721e+5	23.312	2.710e+4
1000	8.707e+4	1.522e+6	45.375	5.177e+4
1500	2.002e+5	3.317e+6	66.890	7.680e+4

For such kind of nonuniform mesh, the theoretical results given in the previous sections still hold. In Table VI we give the CPU times (sec) of the computations with scheme (4.1)–(4.4) with and without mesh refinement technique, for different number of time cycles. Only for comparison; we also show the CPU times of the computations with zero Dirichlet boundary condition on the remote boundary in region $[0, 1] \times [0, 3]$, which of course is not large enough, in general, so that the desired accuracy to be reached. The mesh is uniform, $h_1 = h_2$.

VI. CONCLUSIONS

For 1D and 2D heat equations with nonlinear boundary conditions in unbounded domains we constructed artificial boundary conditions. On this base we derived numerical schemes for solution of the heat problems. The solutions in the counterpart domains were computed by integral formulas. Below we summarize advantages and disadvantages of our numerical algorithms in comparison with known ones: using approximate finite boundary or infinite quasi-uniform grids (uniform or quasi-uniform), etc.

On the positive side are the following features of the new scheme:

- The high accuracy of the numerical solution.
- The scheme on the bounded subdomain has second-order local truncation error in space and first in time. It is not difficult to construct the scheme from high-order accuracy as well in time, using a three-level time scheme; see [1, 6];
- The solution in counterpart domains can be computed at any point directly, using formula.
- The computations can be performed on a very small region.
- The artificial boundary, say l , can be chosen in a very simple way: $\text{supp } u_0(x) < l$.

On the negative side are the following three main disadvantages:

- The construction of artificial boundary condition is possible for a restricted class of problems and its derivation is often not easy;
- Geometrically not universal.
- Algorithmically simple, but numerically expensive, because of involving the convolution integral with “memory property”.

We thank the anonymous reviewers for their comments and suggestions to revise and enhance portions of this article.

References

1. E. Alshina, N. Kalatkin, and S. Panchenko, Numerical solution of boundary value problem in unlimited area, *Math Model* 14(11) (2002), 10–22 (in Russian).
2. M. Berger and R. Kohn, A rescaling algorithm for the numerical calculation of blowing up solutions, *Comm Pure Appl Math* 41 (1988), 841–863.
3. D. Givoli, *Numerical methods for problems in infinite domains*, Elsevier, Amsterdam, 1992.
4. D. Han and Z. Huang, A class of artificial boundary conditions for heat equation in unbounded domains, *Comp Math Appl* 43 (2002), 889–900.
5. S. Tsynkov, Numerical solution of problems on unbounded domains, a review, *Appl Numer Math* 27 (1989), 465–532.
6. X. Wu and Z. Sun, Convergence of different scheme for heat equation in unbounded domains using artificial boundary conditions, *Appl Numer Math* 50 (2004), 261–277.
7. D. Andreuci and R. Gianni, Global existence and blow up in a parabolic problem with nonlocal dynamical boundary conditions, *Adv Diff Eq* 1(5) (1996), 729–752.
8. J. Crank, *The mathematics of diffusion*, Clarendon Press, Oxford, 1973.
9. A. Samarskii and A. Tihonov, *Equations in mathematical physics*, MGU, Moscow, 1999 (in Russian).
10. V. Galaktionov and H. Levine, On critical Fujita exponents for heat equations with nonlinear flux conditions of the boundary, *Israel J Math* 94 (1996), 125–146.
11. J. Bonder and J. Rossi, Life span for solutions of the heat equation with a nonlinear boundary condition Tsukuba *J Math* 25(1) (2001), 215–220.
12. C. Bandle and H. Brunner, Blow-up in diffusion equations: a survey, *J Comp Appl Math* 97 (1998), 3–22.
13. K. Deng, M. Fila, and H. Levine, On critical exponents for a system of heat equations coupled in the boundary conditions, *Acta Math Univ Comeniani* 63 (1994), 169–192.
14. M. Fila and P. Quittner, Large time behavior of solutions of semilinear parabolic equations with nonlinear dynamical boundary conditions, *Topics Nonlinear Anal Progr Nonl Diff Eq Appl* 35 (1999), 251–272.
15. M. Koleva and L. Vulkov, On the blow-up of finite difference solutions to the heat-diffusion equation with semilinear dynamical boundary conditions, *Appl Math Comput* 161 (2005), 69–91.
16. M. Koleva and L. Vulkov, Blow-up of continuous and semidiscrete solutions to elliptic equations with Ssemilinear dynamical boundary conditions of parabolic type, *J Comp Appl Math* available online.
17. T. Nakagava, Blowing up of a finite difference solution to $u_t = u_{xx} + u^2$, *Appl Math Optimization* 2 (1976), 337–350.
18. A. Samarskii and E. Nikolaev, *Methods for the solution of grid equation* Nauka, Moscow, 1978 (in Russian) English translation: *Numerical methods for grid equations*, Vol. I, Birkhäuser Verlag, Basel-Boston-Berlin, 1989.
19. I. Patlashenko, D. Givoli, and P. Barbone, Time-stepping schemes for systems of Volterra integro-differential equations, *Comp Methods Appl Mech Eng* 190 (2001), 5691–5718.
20. P. Wang, C. Zheng, and S. Gorelick, A general approach to advective-dispersive transport with multirate mass transfer, *Adv Water Res* 28 (2005), 33–42.
21. R. Ewing, R. Lazarov, and P. Vassilevski, Local refinement techniques for elliptic problems on cell-centered grids I. Error analysis, *Math Comput* 194(56) (1991), 437–461.
22. M. Koleva, On the computation of blow-up solutions to elliptic equations with semilinear dynamical boundary conditions, *LNCS* 2907 (2004), 473–480.