Pergamon

Computers Math. Applic. Vol. 31, No. 1, pp. 15–32, 1996
Copyright©1996 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0898-1221/96 $15.00 + 0.00

0898-1221(95)00180-8

Numerical Experiments in 2-D IHCP on Bounded Domains Part I: The "Interior" Cube Problem

Y. LIU AND D. A. MURIO*
Department of Mathematical Sciences, University of Cincinnati
Cincinnati, OH 45221-0025, U.S.A.
diego@dmurio.csm.uc.edu

(Received June 1994; revised and accepted July 1995)

Abstract—Different space marching implementations of the Mollification Method are introduced to numerically recover the temperature and heat flux histories at interior points of bounded subdomains of a finite two-dimensional rectangular body when the temperature and heat flux functions are approximately measured at one boundary side.

Keywords—Ill-posed problems, Two-dimensional inverse heat conduction, Discrete mollification, Singular perturbation, Finite differences.

1. INTRODUCTION

Theoretical concepts and computational implementation related to the inverse heat conduction problem (IHCP) have been mostly restricted to one-dimensional models. The difficulties of the two-dimensional IHCP are more pronounced and very few results are available in this case.

The first analytical solution—requiring exact data—that is applicable to two-dimensional conduction systems for geometries of arbitrary shape was introduced by Imber [1]. Most of the literature related to the numerical treatment of the two-dimensional IHCP is based on different ways of combining finite elements realizations with the Future Temperatures Method of Beck [2]. For some of the early applications of these ideas, see [3]. More numerical experimentation can be found in [4]. An elaborated and comprehensive exposition of the method was presented later by Baumeister and Reinhardt [5], and, more recently, by Zabaras and Liu [6]. See [7] for a historical perspective.

In Section 2 of this paper, we review the space marching implementation of the Mollification Method for the two-dimensional inverse heat conduction problem in a slab—including the numerical procedure and the corresponding numerical analysis—as introduced by Guo and Murio [8]. In Section 3, we concentrate our attention on the actual application of the space-marching fully explicit algorithm to the more realistic situation related with a bounded rectangular space domain in the (x, y)-plane. We study the task of recovering transient temperatures and heat fluxes at "interior" space domains for times greater than a certain $t_{\min} > 0$. Section 4 is devoted to the analysis of other techniques, such as singular perturbation schemes, to enhance the stability and consistency of the numerical method. The question of attempting to recover the initial temperature distribution on the interior domains and the testing of numerical strategies for the

*Partially supported by a C. Taft Fellowship.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

approximate reconstruction of the temperature functions at the boundaries of the whole original bounded domain—a much more difficult task—will be reported in a subsequent publication, Part II: The "Lost Boundary" Problem. In all cases, we provide several model examples to generously illustrate the results of the numerical experiments.

We assume all the functions involved to be L_2 functions in R^2 suitably extended—when necessary for the analysis—as being zero everywhere if $t \leq 0$ or $y \leq 0$.

2. TWO-D IHCP IN A SEMI-INFINITE SLAB

We consider a two-dimensional IHCP in a semi-infinite slab, in which the temperature and heat flux histories $f(y,t)$ and $q(y,t)$ on the right-hand side ($x = x_1$) are desired and unknown and the temperature and heat flux on the left-hand surface ($x = 0$) are approximately measurable.

The normalized linear problem can be described mathematically as follows. The unknown temperature $u(x,y,t)$ satisfies:

$$u_t = u_{xx} + u_{yy}, \qquad 0 < x < x_1, \quad y > 0, \quad t > 0, \tag{1a}$$

$$u(0,y,t) = F(y,t), \qquad y > 0, \qquad t > 0, \tag{1b}$$

with corresponding approximate data function $F_m(y,t)$,

$$u_x(0,y,t) = Q(y,t), \qquad y > 0, \qquad t > 0, \tag{1c}$$

with corresponding approximate data function $Q_m(y,t)$,

$$u(x,y,0) = 0, \qquad 0 \leq x \leq x_1, \quad y \geq 0, \tag{1d}$$

$$u(x_1,y,t) = f(y,t), \qquad y > 0, \qquad t > 0, \tag{1e}$$

the desired but unknown temperature function,

$$u_x(x_1,y,t) = q(y,t), \qquad y > 0, \qquad t > 0, \tag{1f}$$

the desired but unknown heat flux function,

$$u(x,0,t) = 0, \qquad 0 \leq x \leq x_1, \quad t > 0. \tag{1g}$$

We hypothesize that the exact data functions $F(y,t)$ and $Q(y,t)$ and the measured data functions $F_m(y,t)$ and $Q_m(y,t)$ satisfy the L_2 data error bounds

$$\|F - F_m\| \leq \varepsilon \quad \text{and} \quad \|Q - Q_m\| \leq \varepsilon. \tag{2}$$

The Fourier analysis of system (1), presented in [8], demonstrates that if s and w represent the Fourier transform variables associated with y and t, respectively, attempting to solve problem (1)—obtaining $f(y,t)$ and $q(y,t)$ from $F(y,t)$ and $Q(y,t)$—amplifies the error in a high frequency component by the factor

$$\exp\left[\left(\sqrt{s^4 + w^2} + s^2\right)^{1/2}\right].$$

Thus, the inverse problem is highly ill-posed in the high frequency components.

2.1. Stabilized Problem

Introducing the two-dimensional Gaussian kernel

$$\rho(y,t,\delta_1,\delta_2) = \frac{1}{\pi\delta_1\delta_2} \exp\left[-\left(\frac{y^2}{\delta_1^2} + \frac{t^2}{\delta_2^2}\right)\right]$$

and denoting $\rho(y,t,\delta_1,\delta_2)$ by $\rho_\delta(y,t)$ if $\delta_1 = \delta_2 = \delta$, the two-dimensional convolution of any locally integrable function $g(y,t)$ with the Gaussian kernel $\rho_\delta(y,t)$—the mollification of the function $g(y,t)$—is written as

$$J_\delta g(y,t) = (\rho_\delta * g)(y,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \rho_\delta(y',t')\, g(y-y',t-t')\, dy'\, dt'.$$

Mollifying system (1), we obtain the following associated problem: attempt to find $J_\delta f_m(y,t) = J_\delta u(x_1,y,t)$ and $J_\delta q_m(y,t) = J_\delta u_x(x_1, y, t)$ at some point (y,t) of interest and for some radius $\delta > 0$ given that $J_\delta u(x,y,t)$ satisfies

$$\begin{aligned}
(J_\delta u)_t &= (J_\delta u)_{xx} + (J_\delta u)_{yy}, && 0 < x < x_1, \quad y > 0, \quad t > 0,\\
J_\delta u(0,y,t) &= J_\delta F_m(y,t), && y > 0, \quad t > 0,\\
J_\delta u_x(0,y,t) &= J_\delta Q_m(y,t), && y > 0, \quad t > 0,\\
J_\delta u(x,y,0) &= 0, && 0 \le x \le x_1, \quad y > 0,\\
J_\delta u(x,0,t) &= 0, && 0 \le x \le x_1, \quad t > 0.
\end{aligned} \tag{3}$$

This problem and its solutions satisfy the following theorem.

THEOREM 1. *Suppose that $\|F - F_m\| \le \epsilon$ and $\|Q - Q_m\| \le \epsilon$. Then:*

1. *Problem (3) is a formally stable problem with respect to perturbations in the data.*
2. *If the exact boundary temperature $f(y,t)$ and heat flux $q(y,t)$ have uniformly bounded first order partial derivatives on the bounded domain $D = [0,Y] \times [0,T]$, then $J_\delta f_m$ and $J_\delta q_m$ verify*

$$\|f - J_\delta f_m\|_D \le O(\delta) + 2\epsilon \exp\left[7\delta^{-2}\right]$$

and

$$\|q - J_\delta q_m\|_D \le O(\delta) + 2\epsilon \exp\left[7\delta^{-2}\right].$$

The proof of this statement can be found in [8].

2.2. Numerical Procedure and Error Analysis

With $v = J_\delta u$ and $w = v_x$, system (1) is equivalent to

$$\begin{aligned}
v_t &= w_x + v_{yy}, && 0 < x < x_1, \quad y > 0, \quad t > 0,\\
w &= v_x, && 0 < x < x_1, \quad y > 0, \quad t > 0,\\
v(0,y,t) &= J_\delta F_m(y,t), && y > 0, \quad t > 0,\\
w(0,y,t) &= J_\delta Q_m(y,t), && y > 0, \quad t > 0,\\
v(x,y,0) &= 0, && 0 \le x \le x_1, \quad y \ge 0,\\
v(x_1,y,t) &= J_\delta f_m(y,t), && y > 0, \quad t > 0, \text{ unknown},\\
w(x_1,y,t) &= J_\delta q_m(y,t), && y > 0, \quad t > 0, \text{ unknown},\\
v(x,0,t) &= 0, && 0 \le x \le x_1, \quad t > 0.
\end{aligned} \tag{4}$$

Without loss of generality, we will seek to reconstruct the unknown mollified boundary temperature function $J_\delta f_m$ and mollified boundary heat flux function $J_\delta q_m$ in the unit square $D = [0,1] \times [0,1]$ of the (y,t)-plane $x = x_1$. Consider a uniform grid in the (x,y,t)-space:

$$\Big\{ (\xi_i = ih,\ y_j = js,\ t_n = nk),\ i = 0,1,\dots,N,\ Nh = x_1;$$
$$j = 0,1,\dots,M,\ Ms = L;\ n = 0,1,\dots,P,\ Pk = C \Big\},$$

where L and C depend on h, s, and k in a way to be specified later, $L, C > 1$. Let the grid functions V and W be defined by

$$V_{i,j}^n = v(\xi_i, y_j, t_n), \quad W_{i,j}^n = (\xi_i, y_j, t_n), \qquad 0 \le i \le N, \quad 0 \le j \le M, \quad 0 \le n \le P.$$

We notice that

$$\begin{aligned} V_{0,j}^n &= J_\delta F_m(y_j, t_n), && 0 \le j \le M, \quad 0 \le n \le P, \\ W_{0,j}^n &= J_\delta Q_m(y_j, t_n), && 0 \le j \le M, \quad 0 \le n \le P, \\ V_{i,0}^n &= 0, && 0 \le i \le N, \quad 0 \le n \le P, \\ V_{i,j}^0 &= 0, && 0 \le i \le N, \quad 0 \le j \le M. \end{aligned}$$

We approximate the system of partial differential equations (4) with the consistent finite difference schemes

$$\begin{aligned} W_{i+1,j}^n &= W_{i,j}^n + \frac{h}{2k}\left(V_{i,j}^{n+1} - V_{i,j}^{n-1}\right) - \frac{h}{s^2}\left(V_{i,j-1}^n - 2V_{i,j}^n + V_{i,j+1}^n\right) \\ V_{i+1,j}^n &= V_{i,j}^n + hW_{i+1,j}^n, \\ & \qquad i = 0, 1, \dots, N\text{–}1; \quad j = 1, 2, \dots, M\text{–}i\text{–}1; \quad n = 1, 2, \dots, P\text{–}i\text{–}1, \\ V_{0,j}^n &= (J_\delta F_m)_j^n, \qquad j = 1, 2, \dots, M\text{–}1; \quad n = 1, 2, \dots, P\text{–}1, \\ W_{0,j}^n &= (J_\delta Q_m)_j^n, \qquad j = 1, 2, \dots, M\text{–}1; \quad n = 1, 2, \dots, P\text{–}1, \\ V_{i,0}^n &= 0, \qquad i = 0, 1, \dots, N\text{–}1; \quad n = 1, 2, \dots, P\text{–}i, \\ V_{i,j}^0 &= 0, \qquad i = 1, 2, \dots, N; \quad j = 1, 2, \dots, M. \end{aligned} \tag{5}$$

Notice that as we march forward in the x-direction in space, we must drop the estimation of the interior temperature from the highest previous point in time and the associated right-most point in the y-direction. Since we want to evaluate $\{V_{N,j}^n\}$ and $\{W_{N,j}^n\}$ at the grid points of the unit square $D = [0,1] \times [0,1]$—in the (y,t)-plane at $x = x_1$—after N iterations, the minimum initial length C of the data sample interval in the time axis needs to satisfy the condition $C = Pm = 1 - k + k/h$. Similarly, the minimum initial length L of the data sample interval in the y-direction satisfies $L = Ms = 1 - s + s/h$.

The stability of the finite difference scheme (5) and the convergence of the numerical solution to the solution of the mollified problem (3) are shown in [8].

REMARKS.

1. The radius of mollification, δ, can be selected automatically as a function of the level of noise in the data. In fact, for a given $\epsilon > 0$, there is a unique $\delta > 0$, such that
$$\|J_\delta F_m - F_m\|_D = \epsilon. \tag{6}$$
2. It is also possible to replace the discrete two-dimensional mollification of the data functions by two successive one-dimensional mollifications in time and y-space. In this manner, the data filtering task can be executed as a parallel process.
3. If $F(y,t)$ and $Q(y,t)$ denote the exact temperature and heat flux functions, respectively, the noisy data is obtained by adding different random errors to $F(y_j, t_n)$ and $Q(y_j, t_n)$ at each grid point. That is,
$$F_m(y_j, t_n) = F(y_j, t_n) + \epsilon_{j,n}^F,$$
and
$$Q_m(y_j, t_n) = Q(y_j, t_n) + \epsilon_{j,n}^Q,$$
where $\epsilon_{j,n}^F$ and $\epsilon_{j,n}^Q$ are independent Gaussian random variables with variance $\sigma^2 = \epsilon^2$ and zero mean.

3. TWO-D IHCP IN A BOUNDED DOMAIN

We investigate now the two-dimensional IHCP when the space domain in the (x, y)-plane is restricted to the bounded prototype rectangle $R = [0, x_1] \times [0, 1]$. We must add the boundary condition

$$u(x, 1, t) = \alpha(x, t), \qquad 0 \le x \le x_1, \quad t > 0,$$

and, at the same time, we shall not require the homogeneity of the boundary conditions (1d) and (1g) that now should read, respectively,

$$u(x, y, 0) = \beta(x, y), \qquad 0 \le x \le x_1, \quad 0 \le y \le 1,$$

and

$$u(x, 0, t) = \gamma(x, t), \qquad 0 \le x \le x_1, \quad t > 0.$$

However, since the functions f, α, β, γ, $u(x, y, 0)$, and F uniquely determine the solution of the direct problem—including the heat fluxes q and Q at $x = x_1$ and $x = 0$, respectively—it is clear that, for the inverse problem, the data functions F and Q together with the initial temperature distribution possess all the necessary information for the recovery of the temperature and heat flux functions f and q at $x = x_1$ and the boundary conditions α, β, and γ. Consequently, the functions $u(x, 0, t) = \gamma(x, t)$, $u(x, 1, t) = \alpha(x, t)$, and $u(x, y, 0) = \beta(x, y)$ will be treated as *unknowns* and their recovery becomes a natural task for the two-dimensional IHCP. Mathematically, the new inverse problem can be stated as follows.

The temperature function $u(x, y, t)$ satisfies:

$$u_t(x, y, t) = u_{xx}(x, y, t) + v_{yy}(x, y, t), \qquad 0 < x < x_1, \quad 0 < y < 1, \quad t > 0, \tag{7a}$$

$$u(0, y, t) = F(y, t), \qquad 0 < y \le 1, \quad t > 0, \tag{7b}$$

with corresponding approximate data function $F_m(y, t)$,

$$u_x(0, y, t) = Q(y, t), \qquad 0 < y \le 1, \quad t > 0, \tag{7c}$$

with corresponding approximate data function $Q_m(y, t)$,

$$u(x, y, 0) = \beta(x, y), \text{ unknown}, \qquad 0 \le x \le x_1, \quad 0 \le y \le 1, \tag{7d}$$

$$u(x, 0, t) = \gamma(x, t), \text{ unknown}, \qquad 0 < x < x_1, \quad t > 0, \tag{7e}$$

$$u(x, 1, t) = \alpha(x, t), \text{ unknown}, \qquad 0 < x < x_1, \quad t > 0, \tag{7f}$$

$$u(x_1, y, t) = f(y, t), \text{ unknown}, \qquad 0 < y \le 1, \quad t > 0, \tag{7g}$$

$$u(x_1, y, t) = q(y, t), \text{ unknown}, \qquad 0 < y \le 1, \quad t > 0. \tag{7h}$$

For a discussion of the uniqueness of the solution of this problem, the reader is referred to [7].

3.1. "Interior" Cube

In this section, we begin the numerical testing of the space marching implementation of the mollification method. The numerical process, as well as the results of the recovery, depends on a set of parameters that includes the sample step-sizes, the radii of mollification and some others. One of the major tasks of this numerical study is to show practical stability, and to find an optimal set of parameters for the fully explicit algorithm. Accordingly, we test the method on five different model examples. The overall performance of our algorithm is "problem independent."

3.1.1. The list of examples

Here we list the five examples that are used to illustrate the results of our numerical experiments. As mentioned before, the general conclusions do not depend on any particular example and, most of the time, we will present the numerical results associated with the prototype **Example 1**.

EXAMPLE 1.

$$\begin{aligned} u(x,y,t) &= e^{5-2t}\sin x \sin y \\ u_x(x,y,t) &= e^{5-2t}\cos x \sin y. \end{aligned}$$

EXAMPLE 2.

$$\begin{aligned} u(x,y,t) &= e^{5-4.25t}\sin \frac{1}{2}x \sin 2y \\ u_x(x,y,t) &= \frac{1}{2}e^{5-4.25t}\cos \frac{1}{2}x \sin 2y. \end{aligned}$$

EXAMPLE 3.

$$\begin{aligned} u(x,y,t) &= e^{t+(1/\sqrt{2})(x+y)} \\ u_x(x,y,t) &= \frac{1}{\sqrt{2}}e^{t+(1/\sqrt{2})(x+y)}. \end{aligned}$$

EXAMPLE 4.

$$\begin{aligned} u(x,y,t) &= \frac{1}{3}x^3y + 3txy + \frac{1}{6}xy^3 \\ u_x(x,y,t) &= x^2y + 3ty + \frac{1}{6}y^3. \end{aligned}$$

EXAMPLE 5.

$$\begin{aligned} u(x,y,t) &= 50x - 100x^2 + 50y^2 - 100t \\ u_x(x,y,t) &= 50 - 200x. \end{aligned}$$

All these functions satisfy the heat-conduction equation (1a).

We study now the two-dimensional IHCP when the space domain in the (x,y)-plane is the rectangle $R = [0,x_1] \times [0,1]$ with initial data temperature and heat flux functions $F_m(y,t)$ and $Q_m(y,t)$ measured at the discrete grid points of the domain $D_{\text{data}} = [0,1] \times [0,T]$ in the (y,t)-space at $x = 0$, with sample step sizes s and k in the y and t directions, respectively. Once the radii of mollification δ_F and δ_Q have been obtained—after solving the discrete version of equation (6)—setting $\delta = \max(\delta_F, \delta_Q)$, δ a multiple of s and k, the discrete mollified data functions $J_\delta F_m$ and $J_\delta Q_m$—computed using a rectangular quadrature rule in the original grid—are evaluated on a coarser grid with step sizes $\Delta y = n_y s$ and $\Delta t = n_t k$ $(n_y, n_t > 1)$, in the domain $D_\delta = [3\delta, 1-3\delta] \times [3\delta, T-3\delta]$ of the (y,t)-plane at $x = 0$. The numbers n_y and n_t are the refine constants of the algorithm.

Our new immediate task is to recover the temperature and heat flux functions at the grid points of the three-dimensional "interior" cube $\Omega_\delta = [0,x_1] \times [3\delta, 1-3\delta] \times [3\delta, 1]$ of the (x,y,t)-space with the knowledge of the discrete data functions F_m and Q_m at the grid points of the domain D_δ.

3.1.2. Numerical procedure

Following Section 2.2, with $T = 1 + 3\delta - \Delta t + \Delta t/h$, we consider the uniform grid in the (x, y, t)-space:

$$\Big\{(\xi_i = ih,\ y_j = 3\delta + j\Delta y,\ t_n = 3\delta + n\Delta t),\ i = 0, 1, \dots, N_x,\ N_x h = x_1;$$
$$j = 0, 1, \dots, N_y,\ N_y \Delta y = 1 - 6\delta;\ n = 0, 1, \dots, N_t,\ N_t \Delta t = T - 6\delta\Big\}.$$

We define the grid functions

$$V_{i,j}^n = v(\xi_i, y_j, t_n),\quad W_{i,j}^n = (\xi_i, y_j, t_n),\qquad 0 \le i \le N_x,\ 0 \le j \le N_y,\ 0 \le n \le N_t,$$

and with $v = J_\delta u$ and $w = v_x$, we introduce the consistent finite difference scheme

$$\begin{aligned}
W_{i+1,j}^n &= W_{i,j}^n + \frac{h}{2k}\left(V_{i,j}^{n+1} - V_{i,j}^{n-1}\right) - \frac{h}{s^2}\left(V_{i,j-1}^n - 2V_{i,j}^n + V_{i,j+1}^n\right)\\
V_{i+1,j}^n &= V_{i,j}^n + hW_{i-1,j}^n,\\
&\qquad i = 0, 1, \dots, N_x - 1;\quad j = 1, 2, \dots, N_y - 1;\quad n = 1, 2, \dots, N_t - 1,\\
V_{0,j}^n &= (J_\delta F_m)_j^n,\qquad j = 0, 1, \dots, N_y;\qquad n = 1, 2, \dots, N_t,\\
W_{0,j}^n &= (J_\delta Q_m)_j^n,\qquad j = 1, 2, \dots, N_y;\qquad n = 1, 2, \dots, N_t.
\end{aligned}\tag{8}$$

As we march forward in the x-direction in space, we must drop the estimation of $V_{i,j}^n$ associated with the boundary values of j and n. We have already anticipated this behavior for the largest time index and we have read, accordingly, all the necessary time data at $x = 0$. However, we need to extrapolate to recover the boundary conditions at $y = 3\delta$, $y = 1 - 3\delta$, and $t = 3\delta$. For example, at each step of the marching scheme, the boundary value for $j = N_y$ is recovered by extrapolation from the corresponding values of the discrete heat flux function at $j = N_y - 1$, $N_y - 2$, $N_y - 3$, and $N_y - 4$, introducing a $O(\Delta y^4)$ error in the process. We recover the discrete temperatures at $j = 0$ and $n = 0$ in a similar way. The algorithm is continued for $i = 1, 2, \dots, N_{x_1}$ until all the required quantities $\{W_{N_{x_1},j}^n\}$ and $\{U_{N_{x_1},j}^n\}$ are computed. These values are then taken as the accepted approximations for the heat flux and temperature functions, respectively, at each point of the discrete domain $[3\delta, 1 - 3\delta] \times [3\delta, 1]$ of the (y, t)-space at $x = x_1$.

It is desirable to make the grids denser in order to get more detailed information about the computed temperature and heat flux functions. However, there is a trade-off between the step-size and the accuracy of the recovery since the IHCP is an ill-posed problem. Keeping the accuracy at a certain level, we try to make the sample step-sizes $\Delta y, \Delta t$ and space-marching step-size h as small as possible. The performance of the algorithm is mostly justified by the consistency and the magnitude of these parameters.

We would like to test the stability of the algorithm under moderately high levels of noise. However, higher noise level means that a larger mollification constant is required to stabilize the numerical process, and, therefore, we have a smaller inner cube to recover.

The accuracy of a recovery is measured by the l_2 error norm of the recovered function **over the whole interior cube Ω_δ**.

The stability of the algorithm is indirectly measured by the size of the working grid, i.e., by the sample step-size and the space-marching step-size of the recovery. For the same level of accuracy, an algorithm is considered more stable if it allows for a denser working grid, i.e., smaller sample step-size and space marching step-size.

To put ourselves in a comparable contention, we consider (most of the time) the noise level constant, ε, up to 0.005 in our experiments. The corresponding mollification constant δ is 0.0667. A recovery is considered "accurate" (acceptable) if the l_2 error norm of the recovered temperature function is less than 0.01 and that of the heat flux function is less than 1. An algorithm is "**quasi-stable**" if it allows an accurate recovery with $h, \Delta y, \Delta t \le 0.1$.

No specific requirements are imposed on the grid refine constants. We will take the smallest integers that stabilize the numerical process. In most cases, $n_y = n_t = 10$. We also fix $x_1 = 0.5$.

3.1.3. The mollification results

For the mollification method, with $\varepsilon = 0.005$ and $\delta = 0.0667$, the optimal combination of the parameter values, according to our numerical experience, are $N_x = 10$, $N_y = N_t = 40$, $n_y = n_t = 10$.

For this specific choice of the parameter set, the l_2 error norms of the recovered temperature and heat flux functions on the whole cube Ω_δ are shown in Table 1.

Table 1. The l_2 error norms. Example 1. $N_x = 10$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$.

Example 1	Temperature	Heat Flux
l_2 Error	0.0079	0.4490
Relative l_2 Error	0.0009	0.0153

Here and in what follows,

$$\text{Relative } l_2 \text{ error norm } = \frac{l_2 \text{ error norm}}{l_2 \text{ norm of the function}}.$$

Table 2 contains the l_2 error norms for a list of 2-surfaces for fixed values of x.

Table 2. The l_2 error norms for 2-surfaces for fixed value of x. Example 1. $N_x = 10$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$.

x	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
0.1	0.0153	0.0050	0.1776	0.0058
0.2	0.0573	0.0094	0.2514	0.0084
0.3	0.1295	0.0143	0.3246	0.0111
0.4	0.2310	0.0193	0.4367	0.0154
0.5	0.3612	0.0245	1.0794	0.0400

Next we take the previous parameter set values as a reference to test the performance of the algorithm under a different choice of parameters. At each experience, only one of the parameters is changed.

Table 3 lists the dependency between the two parameters ε and δ. All the error norms are for the temperature functions.

Table 3. The optimal δ for different ε. Example 1. $N_x = 10$, $N_y = N_t = 40$, $n_y = n_t = 10$.

ε	δ	l_2 error	R. l_2 error
0.0005	0.0417	0.0061	0.0006
0.0010	0.0417	0.0061	0.0006
0.0020	0.0500	0.0069	0.0007
0.0030	0.0583	0.0071	0.0008
0.0040	0.0583	0.0071	0.0008
0.0050	0.0667	0.0079	0.0009

Table 4. The change of the space marching step-size. Example 1. $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$.

N_x	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
5	0.0138	0.0015	0.5242	0.0179
10	0.0079	0.0009	0.4490	0.0153
15	0.0058	0.0007	1.7076	0.0580
20	0.0060	0.0007	7.2921	0.2477
25	0.0093	0.0011	26.6089	0.9035

Table 5. The change of the initial sample step-sizes. Example 1. $N_x = 10$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$.

$N_y = N_t$	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
10	0.0301	0.0033	0.3188	0.0105
20	0.0151	0.0017	0.4476	0.0150
40	0.0079	0.0009	0.4490	0.0153
60	0.0053	0.0010	0.5186	0.0177
80	0.0040	0.0005	0.3826	0.0131
100	0.0032	0.0004	0.5165	0.0177

Table 6. The l_2 error norms, all the examples. $N_x = 10$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$.

Example	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
1	0.0079	0.0009	0.4490	0.0153
2	0.0042	0.0012	0.3857	0.0332
3	0.0028	0.0009	0.3307	0.1571
4	0.0017	0.0063	0.3306	0.3569
5	0.0917	0.0024	0.4842	0.0153

Table 4 shows the performance of the algorithm when the space marching step-size h changes. Table 5 is for the change of the sample step-sizes Δy and Δt.

We recall that all the tables listed above are for Example 1. The performance of the algorithm is the same when applied to other examples. The l_2 error norms of the temperature and heat flux functions for the other examples are in Table 6. In all cases, we set $\varepsilon = 0.005$, $\delta = 0.0667$, $h = 0.05$, $\Delta y = \Delta t = 0.025$, $n_y = n_t = 10$.

REMARKS.

1. As we mentioned earlier, one would like to make the number of steps of the space marching, N_x, as big as possible to get more detailed information of the temperature and heat flux functions on the cube Ω_δ. However, due to the ill-posedness of the problem, the accuracy of the recovery is very sensitive to N_x. As illustrated in Table 4, the accuracy drops dramatically on every increment of five units in N_x. According to our experience, N_x is one of the most important stability measurements. To keep the accuracy at the same level, allowing an increment of ten in N_x should be regarded as a substantial improvement on the stability of a given algorithm.
2. As illustrated in Table 5, the accuracy of the recovery seems to be less sensitive to the initial sample sizes N_t and N_y. However, this is not true in general. If we change N_x from 10

Table 7. The change of the initial sample size. Example 1. $N_x = 15$, $\varepsilon = 0.005$, $\delta = 0.0667$.

$n_t = n_y$	$N_t = N_y$	l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$
10	40	0.0058	1.7076
10	60	0.0036	1.3039
8	80	0.0059	4.2234
6	100	0.0027	32.1356

to 15, as illustrated in Table 7, the sensitivity of the accuracy on N_t and N_y becomes clear.

Hence, the grid sizes N_t and N_y are also important measurements for the stability of the algorithm.

The major objective for a quasi-stable algorithm is to make N_x, as well as N_y and N_t, as large as possible while keeping the same level of accuracy.

1. For our Example 1, the exact temperature function on the initial surface is identically zero. Nevertheless, we do add randomly generated noise to the **zero data**. The perturbed "zero data" causes specific trouble on the stability and accuracy of the recovery. As illustrated in Table 6, the accuracy is only related to the magnitude of the noise added into the initial data sample, but not to the individual example model. The performance of the mollification method, when applied to different problems, is consistent.

4. THE STABLE ALGORITHM

In this section, we describe different ways to improve the stability of the mollification algorithm. In particular, we will combine the mollification algorithm with a singular perturbation method. The addition of a suitably chosen term to the heat conduction equation will greatly improve the stability of the mollification algorithm.

4.1. The Singular Perturbation Scheme

The discussion in this section is motivated by the realization that it is possible to modify the previous numerical algorithm in such a way as to allow for some extra filtering of the high frequency components of the noise. In particular, we would like to improve the stability related with the approximation of the second order partial derivative of the solution temperature with respect to the y-variable as we march in the x-direction. What we have in mind is a potential perturbation of the differential operator itself, so that the finite-difference implementation of the algorithm is not substantially changed.

We recall that in the frequency space, the relationship

$$\widehat{u_{xx}}(x,s,w) = \left(iw + s^2\right)\widehat{u}(x,s,w), \tag{9}$$

where s and w are the frequency variables corresponding to the real variables y and t, respectively, leads to the general solution

$$\widehat{u}(x,s,w) = A(s,w)e^{\sqrt{iw+s^2}x} + B(s,w)e^{-\sqrt{iw+s^2}x}.$$

We propose to replace the solution of equation (9) by the function

$$\widehat{v}(x,s,w) = \left(A(s,w)e^{\sqrt{iw+s^2}x} + B(s,w)\,e^{-\sqrt{iw+s^2}x}\right)\,e^{-\lambda s^2 x}, \tag{10}$$

where λ is a small positive constant. We observe that in real space, *the function v is simply the convolution of the function u with a Gaussian kernel that acts only in the y-variable* and, consequently, the analysis of the high frequency components of Section 2 applies directly to this situation. The fundamental task is now reduced to finding the governing differential equation for v.

To this end, we proceed as follows. Introducing $\alpha = \sqrt{iw + s^2}$, taking partial derivatives with respect to x and using (10), we have

$$\begin{aligned}\widehat{v_x}(x,s,w) &= \left(\alpha - \lambda s^2\right) A(s,w)^{\left(\alpha-\lambda s^2\right)x} - \left(\alpha + \lambda s^2\right) B(s,w) e^{-\left(\alpha+\lambda s^2\right)x} \\ &= \alpha\left(A(s,w)e^{\alpha x} - B(s,w)\, e^{-\alpha x}\right) e^{-\lambda s^2 x} - \lambda s^2 \widehat{v}(x,s,w). \qquad (11)\end{aligned}$$

Similarly, utilizing (10) and (11),

$$\begin{aligned}\widehat{v_{xx}}(x,s,w) &= \alpha^2 \widehat{v}(x,s,w) - \alpha\lambda s^2 \left(A(s,w)e^{\alpha x} - B(s,w)e^{-\alpha x}\right) e^{-\lambda s^2 x} - \lambda s^2 \widehat{v_x}(x,s,w) \\ &= \widehat{v_t}(x,s,w) - \widehat{v_{yy}}(x,s,w) - 2\lambda s^2 \widehat{v_x}(x,s,w) - \lambda^2 s^4 \widehat{v}(x,s,w).\end{aligned}$$

By the linearity of the Fourier transformations, the function $v(x,y,t)$ satisfies the linear partial differential equation

$$v_{xx}(x,y,t) = v_t(x,y,t) - v_{yy}(x,y,t) + 2\lambda v_{xyy}(x,y,t) - \lambda^2 v_{yyyy}(x,y,t),$$

a singular perturbation of the original diffusion equation, that depends on the small positive parameter λ.

In actual implementations, assuming the solution is sufficiently smooth, we only retain the first order term in λ, i.e., numerically we use the approximation

$$v_{xx}(x,y,t) \cong v_t(x,y,t) - v_{yy}(x,y,t) + 2\lambda v_{xyy}(x,y,t).$$

Thus, instead of solving the equation

$$u_{xx} = u_t - u_{yy},$$

we will work with the perturbed equation

$$u_{xx} = u_t - u_{yy} + 2\lambda u_{xyy}.$$

For the numerical implementation, we use the following consistent finite difference schemes to obtain the necessary formulas for the space-marching algorithm:

$$\begin{aligned}W^n_{i+1,j} &= W^n_{i,j} + \frac{h}{2\Delta t}\left(V^{n+1}_{i,j} - V^{n-1}_{i,j}\right) - \frac{h}{\Delta y^2}\left(V^n_{i,j-1} - 2V^n_{i,j} + V^n_{i,j+1}\right) \\ &\quad + 2\lambda\frac{h}{\Delta y^2}\left(W^n_{i,j-1} - 2W^n_{i,j} + W^n_{i,j+1}\right), \qquad (12) \\ V^n_{i+1,j} &= V^n_{i,j} + hW^n_{i-1,j}.\end{aligned}$$

4.2. Combined Mollification-Singular Perturbation Method

To add the singular perturbation to the mollification algorithm, we only change the corresponding space-marching formulas as indicated in (12). Everything else is kept the same as in Section 3. We again consider $\varepsilon = 0.005$ and $\delta = 0.0667$.

To see the improvements, we set $N_y = N_t = 40$ and $\lambda = 0.010$. One can compare Table 8 with Table 4 to appreciate the differences made by the singular perturbation term.

Table 8. The l_2 error norms, with singular perturbation. Example 1. $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, $\lambda = 0.010$.

N_x	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
5	0.0047	0.0013	0.1602	0.0138
10	0.0023	0.0007	0.5075	0.0437
15	0.0015	0.0004	0.1927	0.0166
20	0.0015	0.0004	0.2544	0.0219
25	0.0017	0.0005	0.3531	0.0304

Table 8 illustrates a significant improvement in the stability of the algorithm. Much smaller space marching step size is now allowed, and all the results satisfy our accuracy criterion.

We provide one more comparison. Table 9 contains the results without singular perturbation, and Table 10, those with singular perturbation. The parameter values are $\varepsilon = 0.005$, $\delta = 0.0667$, $N_y = N_t = 60$, and $\lambda = 0.004$.

Table 9. The l_2 error norms, without singular perturbation. Example 1. $N_y = N_t = 60$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, $\lambda = 0.000$.

N_x	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
5	0.0093	0.0010	0.5186	0.0177
10	0.0053	0.0006	0.3436	0.0117
15	0.0036	0.0004	1.3039	0.0445
20	0.0037	0.0004	9.9876	0.3405
25	0.0134	0.0015	78.3677	2.6714

Table 10. The l_2 error norms, with singular perturbation. Example 1. $N_y = N_t = 60$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, $\lambda = 0.004$.

N_x	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
5	0.0091	0.0010	0.4455	0.0125
10	0.0051	0.0006	0.2363	0.0081
15	0.0035	0.0004	0.2495	0.0085
20	0.0027	0.0003	0.5511	0.0188
25	0.0024	0.0003	1.3109	0.0447

4.3. Performance on Dense Grids

As illustrated in Table 7, the accuracy of the recovery is also sensitive to the density of the grid for the mollification algorithm when $N_x > 10$. Adding the singular perturbation term also improves the performance of the algorithm on dense grids. To see this, we test the algorithm on the grids with $N_y = N_t > 60$. N_x is set to 20, and again, $\varepsilon = 0.005$ and $\delta = 0.0667$ for all the cases considered.

Tables 11 and 12 are for $N_y = N_t = 80$. All the model examples are tested and the l_2 error norms are presented. Table 11 contains the results without the singular perturbation, and Table 12 those with the singular perturbation. $\lambda = 0.003$ for the results illustrated in Table 12. (Entries larger than 10^3 are replaced by asterisks.)

Table 11. The l_2 error norms, without singular perturbation. $N_x = 20$, $N_y = N_t = 80$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 8$, $\lambda = 0.000$.

Example	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
1	0.0059	0.0007	*******	3.6241
2	0.0059	0.0017	*******	9.2092
3	0.0059	0.0020	*******	50.6177
4	0.0059	0.0224	*******	*******
5	0.0183	0.0005	*******	3.5097

Table 12. The l_2 error norms, with singular perturbation. $N_x = 20$, $N_y = N_t = 80$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 8$, $\lambda = 0.003$.

Example	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
1	0.0021	0.0002	2.5275	0.0863
2	0.0015	0.0004	2.5108	0.2195
3	0.0013	0.0004	2.5258	1.2032
4	0.0013	0.0049	2.5258	2.7430
5	0.0186	0.0005	2.5259	0.0834

Table 13. The l_2 error norms, with singular perturbation. $N_x = 20$, $N_y = N_t = 100$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 6$, $\lambda = 0.001$.

Example	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
1	0.0031	0.0004	31.4977	1.0773
2	0.0031	0.0009	31.5609	2.7675
3	0.0031	0.0010	31.4842	15.0037
4	0.0031	0.0116	31.4842	34.2225
5	0.0146	0.0004	31.4847	1.0399

Table 13 illustrates the results for the even denser grid $N_y = N_t = 100$ with singular perturbation. The testing of the mollification algorithm without singular perturbation for the same grid sizes gives unacceptable data and no corresponding table is provided.

REMARKS.

1. As mentioned earlier, the performance of the algorithm is consistent when applied to different example models. Tables 11–13 can also be taken as evidence of such a strong consistency of the algorithm. The absolute l_2 error norms for the temperature and heat flux functions are practically constant for all the examples.
2. The performance of the combined singular perturbation scheme also depends on the perturbation constant λ. For different grid sizes, one has to choose a proper value of λ to get the expected improvement. Table 14 contains the corresponding optimal values of λ for different grid sizes $N_y = N_t$.

Table 14. The relationship between $N_y = N_t$ and λ.

$N_y = N_t$	λ
40	0.010
60	0.004
80	0.003
100	0.001

4.4. Correction for Heat Flux

According to Tables 12 and 13, the numerical solutions give a good approximation for the temperature functions on the dense grid. However, the results for the heat flux functions do not meet our accuracy standard with $N_y = N_t = 80$, and are not acceptable at all with $N_y = N_t = 100$. To look at the recovered functions more closely, we take a list of 2-surfaces with x fixed and observe the computed heat flux function. The results with $N_x = 20$, $N_y = N_t = 100$ are listed in Table 15.

Table 15. The l_2 error norms with fixed x for the heat flux. Example 1. $N_x = 20$, $N_y = N_t = 100$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 6$, $\lambda = 0.001$.

x	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
0.05	0.1004	0.0033
0.10	0.1321	0.0044
0.15	0.1473	0.0049
0.20	0.1621	0.0054
0.25	0.1776	0.0060
0.30	0.2166	0.0074
0.35	0.2683	0.0094
0.40	3.5576	0.1266
0.45	19.2814	0.7021
0.50	*******	4.2664

One concludes that the large l_2 error norm is mainly caused by the last couple of steps when marching in the x-direction. This explains that, although the earlier recovered heat flux function has to be used to compute the temperature in the current marching step, the high error did not contribute to the recovery of the temperature function. The wild last two values of the heat flux function are not used to obtain the temperature output at those locations.

Consequently, a natural way to fix the error in the recovery of the heat flux function is to use the already evaluated temperatures to recompute the heat flux function at the same level. This should improve the accuracy of the computed heat flux function. The l_2 error norms, for the algorithm with correction, are in Tables 16 and 17. Table 16 is for the case $N_y = N_t = 80$ and Table 17 for $N_y = N_t = 100$. We now compare Table 16 to 12 and Table 17 to 13 to see the improvements.

Table 16. The l_2 error norms, with corrector. $N_x = 20$, $N_y = N_t = 80$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 8$, $\lambda = 0.003$.

Example	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
1	0.0021	0.0002	0.7596	0.0259
2	0.0015	0.0007	0.6358	0.0556
3	0.0013	0.0004	0.6202	0.2955
4	0.0013	0.0049	0.6184	0.6715
5	0.0186	0.0005	7.2472	0.2394

Table 17. The l_2 error norms, with corrector. $N_x = 20$, $N_y = N_t = 100$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 6$, $\lambda = 0.001$.

Example	l_2 for $u(x,y,t)$	R. l_2 for $u(x,y,t)$	l_2 for $u_x(x,y,t)$	R. l_2 for $u_x(x,y,t)$
1	0.0031	0.0004	1.1888	0.0407
2	0.0031	0.0009	1.1147	0.0977
3	0.0031	0.0010	1.0949	0.5218
4	0.0031	0.0116	1.0937	1.1889
5	0.0146	0.0004	7.3011	0.2411

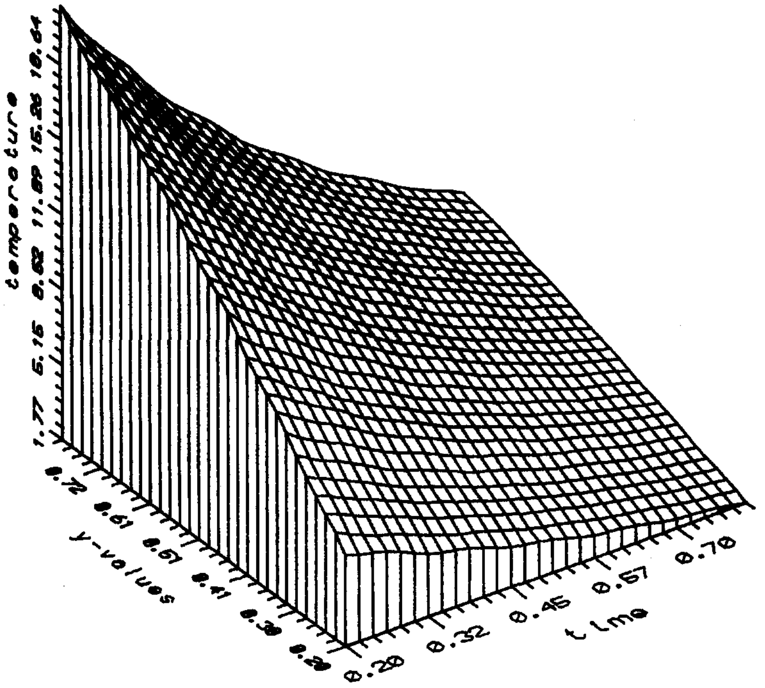

Figure 1. Reconstructed temperature function at $x = 0.5$. Example 1. $N_x = 20$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, and $\lambda = 0.010$.

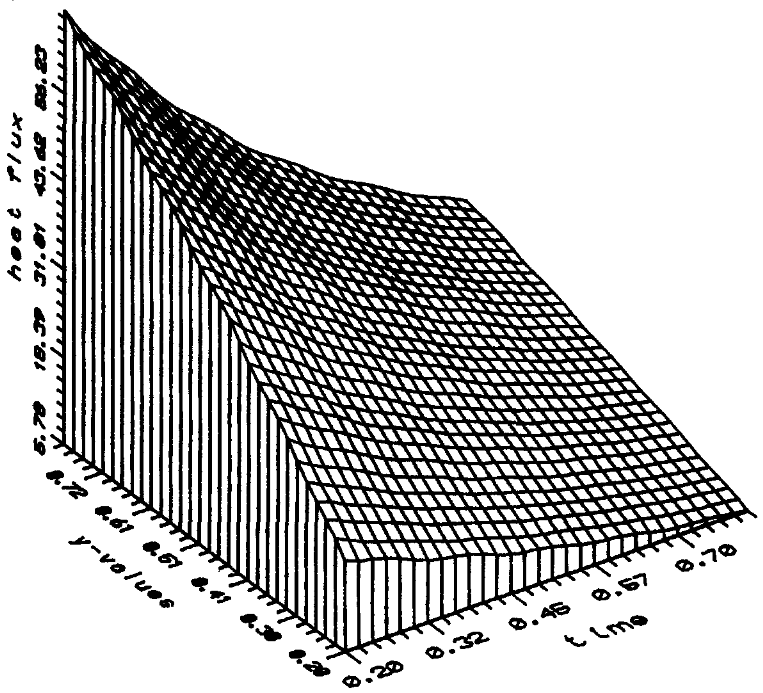

Figure 2. Reconstructed heat flux function at $x = 0.5$. Example 1. $N_x = 20$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, and $\lambda = 0.010$.

Note that the formula used for the correction of the heat-flux functions is given by

$$W_{i,j}^n = \frac{V_{i+1,j}^n - V_{i-1,j}^n}{2h},$$

and the function values at the very end of the space marching ($x = 0.5$) are recomputed by extrapolation. Example 5 (with $u_{xx} = -200$ and $u_{yy} = 100$) behaves exceptionally. The correction scheme does not improve the l_2 error norm for the heat flux function in this case.

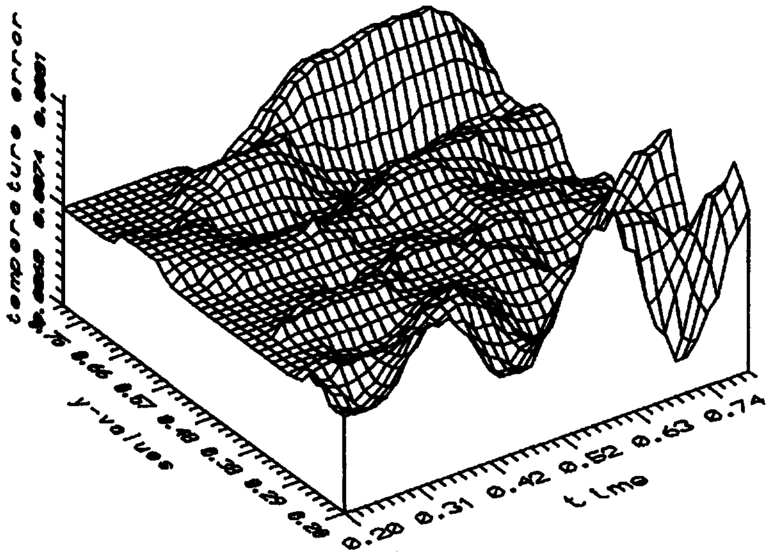

Figure 3. Relative errors for the temperature function at $x = 0.5$. Example 1. $N_x = 20$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, and $\lambda = 0.010$.

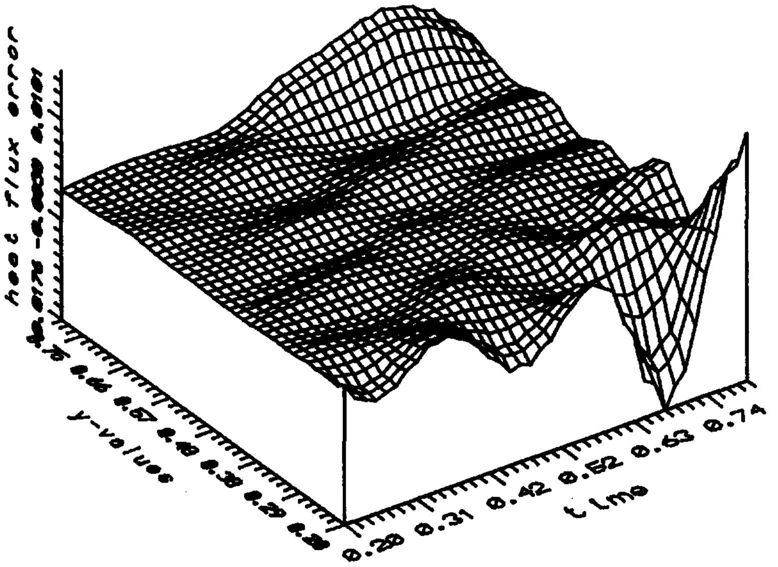

Figure 4. Relative errors for the heat flux function at $x = 0.5$. Example 1. $N_x = 20$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, and $\lambda = 0.010$.

The graphs in this section illustrate the qualitative behavior of the reconstructed temperatures and heat fluxes at $x = 0.5$, corresponding to Example 1 and the set of parameters $N_x = 20$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, and $\lambda = 0.01$.

Figures 1 and 2 show the computed temperature and heat flux functions, respectively, for y and t between 0.2 and 0.8.

The discrete relative error functions associated with the temperature and heat flux functions $\left(\frac{V_{i,j}^n - u(ih,j\Delta y,n\Delta t)}{u(ih,j\Delta y,n\Delta t)} \text{ and } \frac{W_{i,j}^n - u_x(ih,j\Delta y,n\Delta t)}{u_x(ih,j\Delta y,n\Delta t)}\right)$, respectively, are displayed in Figures 3 and 4 for y and t between 0.2 and 0.8.

The transient grid relative errors for the temperature and heat flux functions, corresponding to the space locations $x = y = 0.5$ and $x = 0.5$, $y = 0.8$, respectively, are shown in Figures 5 and 6, for $0.2 \leq t \leq 0.8$.

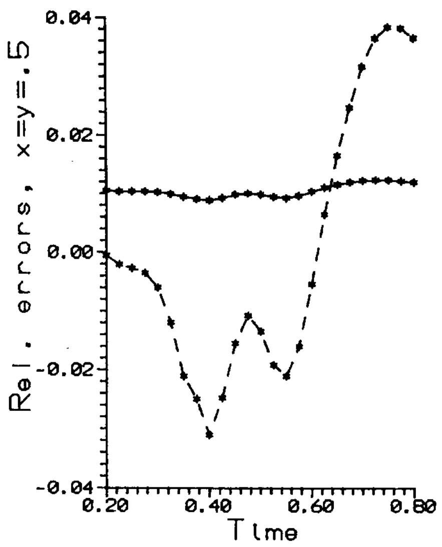

Figure 5. Transient relative errors for the temperature (*solid line*) and heat flux functions (*dashed line*) at $x = y = 0.5$. Example 1. $N_x = 20$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, and $\lambda = 0.010$.

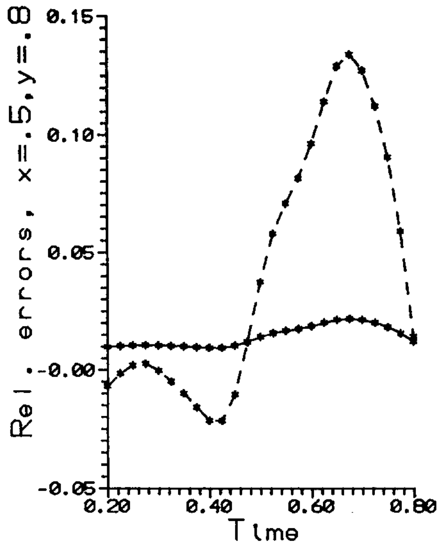

Figure 6. Transient relative errors for the temperature (*solid line*) and heat flux (*dashed line*) functions at $x = 0.5$, $y = 0.8$. Example 1. $N_x = 20$, $N_y = N_t = 40$, $\varepsilon = 0.005$, $\delta = 0.0667$, $n_y = n_t = 10$, and $\lambda = 0.010$.

5. SUMMARY AND CONCLUSIONS

A combined mollification-singular perturbation regularization technique is derived and investigated. The method is implemented as a space-marching-fully-explicit finite differences algorithm, providing computational flexibility and overall efficiency.

Several test cases are investigated for a finite three-dimensional cube in the (x, y, t)-space when the temperature and heat flux data functions are measured only on the square $[0,1]\times[0, 1 + 3\delta]$, in the (y, t)-space at $x = 0$. The required temperature and heat flux functions are numerically approximated in the "interior" cube $\Omega_\delta = [0, 0.5] \times [3\delta, 1 - 3\delta] \times [3\delta, 1]$ of the (x, y, t)-space.

A number of parameters are varied, including radii of mollification, amount of noise in the data, step marching and grid sizes. A reliable set of parameters values is experimentally determined to guarantee acceptable accuracy and good stability properties for the algorithm.

A general conclusion is that the numerical procedure presented in this work remains a viable procedure for the numerical solution of the 2-D IHCP in "interior" bounded subdomains.

The more difficult task involving the recovery of the initial and boundary temperature functions on the whole original three-dimensional unit cube in the (x, y, t)-space will appear in another report, Part II, of this research.

REFERENCES

1. M. Imber, Temperature extrapolation mechanism for two-dimensional heat flow, *AIAA J.* **12**, 1089–1093 (1974).
2. J.V. Beck, Nonlinear estimation applied to the nonlinear inverse heat conduction problem, *Int. J. Heat Mass Transfer* **13**, 703–716 (1970).
3. B.R. Bass and L.J. Ott, A finite element formulation of the two-dimensional inverse heat conduction problem, *Adv. Comput. Technol.* **2**, 238–248 (1980).
4. T. Yoshimura and K. Ituka, Inverse heat conduction problem by finite element formulation, *Int. J. Systems Sci.* **16**, 1365–1376 (1985).
5. J. Baumeister and H.J. Reinhardt, On the approximate solution of a two-dimensional inverse heat conduction problem, In *Inverse and Ill-posed Problems*, (Edited by H.G. Engl and C.W. Groetsch), pp. 325–344, Academic Press, Orlando, FL, (1987).
6. N. Zabaras and J.C. Liu, An analysis of two-dimensional linear inverse heat transfer problems using an integral method, *Numer. Heat Transfer* **13**, 527–533 (1988).

7. D.A. Murio, *The Mollification Method and the Numerical Solution of Ill-Posed Problems*, John Wiley & Sons, Interscience, New York, (1993).
8. L. Guo and D.A. Murio, A mollified space-marching finite difference algorithm for the two-dimensional inverse heat conduction problem with slab symmetry, *Inverse Problems* **7**, 247–259 (1991).