Inverse Problems, Design and Optimization Symposium
Rio de Janeiro, Brazil, 2004

LATTICE-FREE FINITE DIFFERENCE METHOD FOR NUMERICAL SOLUTION OF INVERSE HEAT CONDUCTION PROBLEM

Kentaro Iijima
c/o Professor Kazuei Onishi
Graduate School of Science and Engineering,
Ibaraki University
Bunkyo 2-1-1, Mito-shi, 310-8512, Japan
E-mail: nd2401a@mcs.ipc.ibaraki.ac.jp

Kazuei Onishi
Department of Mathematical Sciences,
Ibaraki University
Bunkyo 2-1-1, Mito-shi, 310-8512, Japan
E-mail: onishi@mx.ibaraki.ac.jp

ABSTRACT

Inverse heat conduction problem consists of finding an initial temperature distribution from the knowledge of a distribution of the temperature at the present time. Here we assume that the associated boundary conditions are known. The heat conduction problem backward in time is a typical example of ill-posed problems in the sense that the solution exists only for regular functions of some kind describing the present temperature distribution and also the solution is unstable for the present temperature distribution function. Conventional numerical methods often suffer from instability of the problem itself when high accuracy is intended in the approximation. Our aim is to create a meshless method which is applicable to the ill-posed inverse heat conduction problem. We construct a high order finite difference method in which quadrature points do not need to have a lattice structure. In order to develop our new method we show a tool in using exponential functions in Taylor's expansion. From numerical experiments we confirmed that our method is effective for solving two-dimensional inverse heat conduction problem numerically subject to mixed boundary conditions.

NOMENCLATURE

D	domain in $\boldsymbol{R}^2$
L, Q	matrices on $\boldsymbol{R}^N$
$\boldsymbol{l}_j$, $\boldsymbol{l}(\boldsymbol{x})$	vectors of $\boldsymbol{R}^N$
m	dimension
N	total number of quadrature points
$\boldsymbol{N}$	the set of natural numbers
$\boldsymbol{n}(\boldsymbol{x})$	outward unit normal on $\partial\Omega$
$P(\boldsymbol{\xi})$	polynomial in $\boldsymbol{\xi}$
$P(\boldsymbol{\partial})$	differential operator
$\bar{q}$	Neumann data
$\boldsymbol{R}$	the set of real numbers
u	solution of the problem
$\boldsymbol{u}$	vector of approximate solution
u_F	final data
$\bar{u}$	Dirichlet data
$\boldsymbol{w}(\boldsymbol{x})$	vector of weights
$\boldsymbol{x}$, $\boldsymbol{\xi}$	points in $\boldsymbol{R}^m$
$\boldsymbol{x}^{(j)}$	the jth quadrature point
$\boldsymbol{Z}_+$	the set of non-negative integers
$\boldsymbol{\alpha}$	multi-index
$\Gamma_\square$	a part of $\partial\Omega$
Δ	Laplacian
Δx_1	lattice width in x_1 direction
Ω	domain in $\boldsymbol{R}^m$
$\boldsymbol{\partial}$	gradient operator
$\partial\Omega$	boundary of Ω

INTRODUCTION

We take a bounded domain D in $\boldsymbol{R}^2$ and a space-time domain $\Omega = D \times (0, T)$ in $\boldsymbol{R}^3$ for a final time $T > 0$, where D represents a heat conductor. A point in the space-time domain Ω is written by $\boldsymbol{x} = {}^t(x_1, x_2, t) = {}^t(x_1, x_2, x_3)$ for the sake of conciseness. We write $\Gamma_d = \Gamma_B \cup \Gamma_F$ by using two surfaces $\Gamma_B = \partial D \times [0, T]$ and $\Gamma_F = D \times \{T\}$ of the boundary $\partial\Omega$. The boundary Γ_B moreover consists of surfaces $\Gamma_{B1} \subset \Gamma_B$ and $\Gamma_{B2} = \Gamma_B \setminus \Gamma_{B1}$. Then for given Dirichlet data $\bar{u} : \Gamma_{B1} \to \boldsymbol{R}$, Neumann data $\bar{q} : \Gamma_{B2} \to \boldsymbol{R}$ and final data $u_F : \Gamma_F \to \boldsymbol{R}$, we consider a problem to look for a function $u(\boldsymbol{x})$

that satisfies

$$\frac{\partial u}{\partial x_3} = \Delta u \quad \text{in} \quad \Omega, \tag{1}$$

$$u = \bar{u} \quad \text{on} \quad \Gamma_{B1}, \quad \frac{\partial u}{\partial n} = \bar{q} \quad \text{on} \quad \Gamma_{B2}, \tag{2}$$

and

$$u = u_F \quad \text{on} \quad \Gamma_F. \tag{3}$$

Here the symbol Δ denotes the Laplacian $\dfrac{\partial^2}{\partial x_1^2} + \dfrac{\partial^2}{\partial x_2^2}$. An outward unit normal of the boundary ∂D at $\boldsymbol{x}$ is denoted by $\boldsymbol{n}(\boldsymbol{x}) = {}^t(n_1(\boldsymbol{x}), n_2(\boldsymbol{x}), 0)$. We call the problem (1)–(3) a two-dimensional backward heat conduction problem under the mixed boundary conditions (2).

The backward heat conduction problem is ill-posed in the sense that the solution does not always exist for any final data u_F. It is known for one-dimensional backward heat conduction equation $\dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2}$ in [1] for $0 < x < 1$, $t \leq T$ with $\dfrac{\partial u}{\partial x}(0, t) = \dfrac{\partial u}{\partial x}(1, t) = 0$ that the final data $u(x, T)$ is required to belong to the function space $\left\{ \sum_{k=0}^{\infty} a_k \cos k\pi x : \sum_{k=0}^{\infty} e^{2k^2\pi^2 T} a_k^2 < +\infty \right\}$ for the existence of the solution $u(x, t)$.

The backward heat conduction problem is also ill-posed in the sense that the solution is unstable for given final data u_F [2]. As a matter of fact we illustrate instability of the problem: Let the domain D be $(-\pi, \pi) \times (-\pi, \pi)$. We set $\Gamma_{B1} = \Gamma_B$ and $\Gamma_{B2} = \phi$. We prescribe the final data $u_F^{(l)}(\boldsymbol{x}) := e^{-2l^2 T} \sin lx_1 \sin lx_2$, $\boldsymbol{x} \in \Gamma_F$ and the boundary data $\bar{u}^{(l)}(\boldsymbol{x}) = 0$, $\boldsymbol{x} \in \Gamma_B$ for an $l \in \boldsymbol{N}$. Then the exact solution of the heat equation (1) is given by $u^{(l)}(\boldsymbol{x}) = e^{-2l^2 x_3} \sin lx_1 \sin lx_2$. We choose the two L^2 norms

$$\|u\|_{L^2(\Omega)} := \left\{ \int_\Omega u(\boldsymbol{x})^2 d\boldsymbol{x} \right\}^{\frac{1}{2}}$$

and

$$\|v\|_{L^2(\Gamma_F)} := \left\{ \int_D v(\boldsymbol{x}', T)^2 d\boldsymbol{x}' \right\}^{\frac{1}{2}}$$

for functions $u : \Omega \to \boldsymbol{R}$ and $v : \Gamma_F \to \boldsymbol{R}$, respectively, where $\boldsymbol{x}' = {}^t(x_1, x_2) \in D$. The solution can be estimated as follows.

$$\begin{aligned} &\|u^{(l)}\|_{L^2(\Omega)}^2 \\ &= \int_\Omega \left(e^{-2l^2 x_3} \sin lx_1 \sin lx_2 \right)^2 d\boldsymbol{x} \\ &= \int_0^T e^{4l^2(T - x_3)} dx_3 \\ &\quad \times \int_D \left(e^{-2l^2 T} \sin lx_1 \sin lx_2 \right)^2 d\boldsymbol{x}' \\ &= \frac{1}{4l^2} \left(e^{4l^2 T} - 1 \right) \|u_F^{(l)}\|_{L^2(\Gamma_F)}^2. \end{aligned} \tag{4}$$

Since for any $C > 0$ we can choose $l \in \boldsymbol{N}$ such that $\dfrac{1}{2l}\sqrt{e^{4l^2 T} - 1} > C$, the inequality $\|u^{(l)}\|_{L^2(\Omega)} > C\|u_F^{(l)}\|_{L^2(\Gamma_F)}$ holds for any $C > 0$. This means that the solution does not depend on the final data continuously in the L^2-sense. Therefore the solution of the backward heat conduction problem is unstable for the final data.

In order to solve the backward heat conduction problem numerically, we consider an application of conventional finite difference schemes. For any time step size $\Delta x_3 > 0$ in the time range $[0, T]$ and for any lattice widths $\Delta x_1 > 0$ and $\Delta x_2 > 0$ in each direction of x_1 and x_2 in D, we know by the von Neumann condition [3] that the following finite difference scheme, explicit backwards, approximating the equation (1) is unstable.

$$\begin{aligned} &\frac{u(x_1, x_2, x_3) - u(x_1, x_2, x_3 - \Delta x_3)}{\Delta x_3} \\ &= \Big\{ u(x_1 - \Delta x_1, x_2, x_3) - 2u(x_1, x_2, x_3) \\ &\qquad + u(x_1 + \Delta x_1, x_2, x_3) \Big\} / {\Delta x_1}^2 \\ &\quad + \Big\{ u(x_1, x_2 - \Delta x_2, x_3) - 2u(x_1, x_2, x_3) \\ &\qquad + u(x_1, x_2 + \Delta x_2, x_3) \Big\} / {\Delta x_2}^2. \end{aligned} \tag{5}$$

We notice that the following scheme, implicit backwards, is also unstable.

$$\begin{aligned} &\frac{u(x_1, x_2, x_3) - u(x_1, x_2, x_3 - \Delta x_3)}{\Delta x_3} \\ &= \Big\{ u(x_1 - \Delta x_1, x_2, x_3 - \Delta x_3) \\ &\qquad - 2u(x_1, x_2, x_3 - \Delta x_3) \\ &\qquad + u(x_1 + \Delta x_1, x_2, x_3 - \Delta x_3) \Big\} / {\Delta x_1}^2 \\ &\quad + \Big\{ u(x_1, x_2 - \Delta x_2, x_3 - \Delta x_3) \\ &\qquad - 2u(x_1, x_2, x_3 - \Delta x_3) \\ &\qquad + u(x_1, x_2 + \Delta x_2, x_3 - \Delta x_3) \Big\} / {\Delta x_2}^2. \end{aligned} \tag{6}$$

We state a motivation to our research. There are numbers of researches which challenge to analyze ill-posed problems numerically. For example, techniques are described in these researches [4], [5] for making both discretization error and rounding error arbitrarily small by using the spectral collocation method and an arbitrary precision arithmetic, respectively. The backward heat conduction problem is solved very precisely under no observation errors by their techniques. However in the spectral collocation method the Chebyshev-Gauss-Lobatto points [6] are employed as quadrature points for the inversion. Therefore it is difficult to apply the techniques to the problem on a domain with curved boundaries. As an applicable method for engineering problems, we propose a more flexible high order finite difference scheme instead which allows quadrature points at arbitrary locations.

NOTATION

We introduce a set $\boldsymbol{Z}_+ := \{z \in \boldsymbol{Z} : \ z \geq 0\}$ and let $\boldsymbol{Z}_+^m = \overbrace{\boldsymbol{Z}_+ \times \boldsymbol{Z}_+ \times \cdots \times \boldsymbol{Z}_+}^{m}$, where $\boldsymbol{Z}$ denotes the set of all integers. Then an element $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m) \in \boldsymbol{Z}_+^m$ is called a multi-index. A symbol $\mathbf{0}$ denotes $(0, 0, \ldots, 0)$. For the multi-index $\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m$, a few operations and relations are defined in the following: A length of $\boldsymbol{\alpha}$ is defined by $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \cdots + \alpha_m$. Let $\boldsymbol{x} = {}^t(x_1, x_2, \ldots, x_m)$ be a vector in $\boldsymbol{R}^m$. We distinguish the above-mentioned length of a multi-index $|\cdot|$ from the length of the vector $|\boldsymbol{x}| = \sqrt{\sum_{k=1}^{m} x_k^2}$. A power of $\boldsymbol{x}$ is defined by $\boldsymbol{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_m^{\alpha_m}$. A factorial of $\boldsymbol{\alpha}$ is defined by $\boldsymbol{\alpha}! := \alpha_1! \alpha_2! \cdots \alpha_m!$. A differential symbol $\dfrac{\partial^{|\boldsymbol{\alpha}|}}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}}$ is meant by $\dfrac{\partial^{\alpha_1+\alpha_2+\cdots+\alpha_m}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_m^{\alpha_m}} = \dfrac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \dfrac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} \cdots \dfrac{\partial^{\alpha_m}}{\partial x_m^{\alpha_m}}$ operated to sufficiently smooth functions. Setting $\partial_j = \dfrac{\partial}{\partial x_j}$ and $\boldsymbol{\partial} = (\partial_1, \ \partial_2, \ \ldots, \ \partial_m)$ formally, we write $\dfrac{\partial^{|\boldsymbol{\alpha}|}}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \boldsymbol{\partial}^{\boldsymbol{\alpha}}$.

FINITE DIFFERENCE APPROXIMATION

In this section, we introduce a finite difference approximation to the derivatives. Let u be an analytic function defined on a convex bounded domain $\Omega \subset \boldsymbol{R}^m$ into $\boldsymbol{R}$. Namely, the function u can be expanded into the Taylor series

$$u(\boldsymbol{y}) = \sum_{\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m} \frac{(\boldsymbol{y} - \boldsymbol{x})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \boldsymbol{\partial}^{\boldsymbol{\alpha}} u(\boldsymbol{x}), \quad \boldsymbol{x}, \ \boldsymbol{y} \in \Omega \tag{7}$$

in the sense of absolute and uniform convergence for any compact subset of Ω. We take a point $\boldsymbol{x} = {}^t(x_1, x_2, \ldots, x_m)$ and N quadrature points $\boldsymbol{x}^{(j)} = {}^t(x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)})$ for $j = 1, 2, \ldots, N$ randomly in Ω. For real constants $a_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha}$ in $\boldsymbol{Z}_+^m$, we set a differential operator $P(\boldsymbol{\partial})$ of order μ_0 as

$$P(\boldsymbol{\partial}) := \sum_{\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m} a_{\boldsymbol{\alpha}} \boldsymbol{\partial}^{\boldsymbol{\alpha}}, \tag{8}$$

where $a_{\boldsymbol{\alpha}} = 0$ for $|\boldsymbol{\alpha}| > \mu_0$ with some $\mu_0 \in \boldsymbol{N}$. We consider approximating the value $P(\boldsymbol{\partial})u(\boldsymbol{x})$ at the point $\boldsymbol{x}$ by using a linear combination of values $u(\boldsymbol{x}^{(j)})$, $j = 1, 2, \ldots, N$. More specifically, by choosing weights $w_j(\boldsymbol{x}) \in \boldsymbol{R}$, $j = 1, 2, \ldots, N$ appropriately, we try to represent the value $P(\boldsymbol{\partial})u(\boldsymbol{x})$ as

$$P(\boldsymbol{\partial})u(\boldsymbol{x}) = \sum_{j=1}^{N} w_j(\boldsymbol{x}) u(\boldsymbol{x}^{(j)}) + \varepsilon(\boldsymbol{x}; P(\boldsymbol{\partial})u), \tag{9}$$

where $\varepsilon(\boldsymbol{x}; P(\boldsymbol{\partial})u)$ denotes a discretization error. We call the approximation (9) a *high order finite difference approximation* of $P(\boldsymbol{\partial})$ with respect to the quadrature points $\boldsymbol{x}^{(j)}$, $j = 1, 2, \ldots, N$ for larger N.

Concretely we can determine the weights $w_j(\boldsymbol{x})$, $j = 1, 2, \ldots, N$ as follows. Substituting the operator (8) to the equality (9), we can see that the left hand side of the equality (9) becomes

$$P(\boldsymbol{\partial})u(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m} a_{\boldsymbol{\alpha}} \boldsymbol{\partial}^{\boldsymbol{\alpha}} u(\boldsymbol{x}). \tag{10}$$

From Taylor's expansion (7) the first term on the right hand side of the equality (9) becomes

$$\sum_{j=1}^{N} w_j(\boldsymbol{x}) u(\boldsymbol{x}^{(j)}) = \sum_{j=1}^{N} w_j(\boldsymbol{x}) \times \left\{ \sum_{\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m} \frac{1}{\boldsymbol{\alpha}!} (\boldsymbol{x}^{(j)} - \boldsymbol{x})^{\boldsymbol{\alpha}} \boldsymbol{\partial}^{\boldsymbol{\alpha}} u(\boldsymbol{x}) \right\}. \tag{11}$$

From relations (10) and (11), the error in the equation (9) can be written as

$$\varepsilon(\boldsymbol{x};\, P(\boldsymbol{\partial})u) = \sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m} \left\{ a_{\boldsymbol{\alpha}} - \sum_{j=1}^{N} w_j(\boldsymbol{x}) \frac{1}{\boldsymbol{\alpha}!}(\boldsymbol{x}^{(j)} - \boldsymbol{x})^{\boldsymbol{\alpha}} \right\} \boldsymbol{\partial}^{\boldsymbol{\alpha}} u(\boldsymbol{x}).$$

In the conventional finite difference approximation, weights $w_j(\boldsymbol{x})$, $j = 1, 2, \ldots, N$ are given by a solution of the linear system

$$a_{\boldsymbol{\alpha}} = \sum_{j=1}^{N} w_j(\boldsymbol{x}) \frac{1}{\boldsymbol{\alpha}!}(\boldsymbol{x}^{(j)} - \boldsymbol{x})^{\boldsymbol{\alpha}}, \quad |\boldsymbol{\alpha}| \le \mu \quad (12)$$

for the largest possible integer μ.

Here we take the following trick in order to choose the weights: Let $\boldsymbol{\xi}^{(i)}$, $i = 1, 2, \ldots, N$ be vectors in $\boldsymbol{R}^m$ such that $\boldsymbol{\xi}^{(i)} \neq \boldsymbol{\xi}^{(j)}$ for $i \neq j$. Multiplied by $(\boldsymbol{\xi}^{(i)})^{\boldsymbol{\alpha}}$ and summed up for all $\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m$, the equality (12) becomes

$$\sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m} a_{\boldsymbol{\alpha}} (\boldsymbol{\xi}^{(i)})^{\boldsymbol{\alpha}} = \sum_{j=1}^{N} w_j(\boldsymbol{x}) \times \left\{ \sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m} \frac{1}{\boldsymbol{\alpha}!} (\boldsymbol{\xi}^{(i)})^{\boldsymbol{\alpha}} (\boldsymbol{x}^{(j)} - \boldsymbol{x})^{\boldsymbol{\alpha}} \right\}. \quad (13)$$

Generally, the equality

$$\begin{aligned} \sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m} \frac{1}{\boldsymbol{\alpha}!} \boldsymbol{\xi}^{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} &= \sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m} \prod_{k=1}^{m} \frac{1}{\alpha_k!} \xi_k^{\alpha_k} x_k^{\alpha_k} \\ &= \prod_{k=1}^{m} \sum_{\alpha_k=0}^{\infty} \frac{1}{\alpha_k!} \xi_k^{\alpha_k} x_k^{\alpha_k} = \prod_{k=1}^{m} e^{\xi_k x_k} \\ &= e^{\boldsymbol{\xi}\cdot\boldsymbol{x}}, \qquad \boldsymbol{x},\, \boldsymbol{\xi} \in \boldsymbol{R}^m \quad (14) \end{aligned}$$

holds. By the equality (14) and a polynomial $P(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m} a_{\boldsymbol{\alpha}} \boldsymbol{\xi}^{\boldsymbol{\alpha}}$, $\boldsymbol{\xi} \in \boldsymbol{R}^m$, the equality (13) is transformed into $P(\boldsymbol{\xi}^{(i)}) = \sum_{j=1}^{N} w_j(\boldsymbol{x}) e^{\boldsymbol{\xi}^{(i)}\cdot(\boldsymbol{x}^{(j)} - \boldsymbol{x})}$, or equivalently

$$P(\boldsymbol{\xi}^{(i)}) e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}} = \sum_{j=1}^{N} w_j(\boldsymbol{x}) e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}^{(j)}}, \quad i = 1, 2, \ldots, N. \quad (15)$$

We set a column vector $\boldsymbol{l}_j := {}^t\left(e^{\boldsymbol{\xi}^{(1)}\cdot\boldsymbol{x}^{(j)}}, e^{\boldsymbol{\xi}^{(2)}\cdot\boldsymbol{x}^{(j)}}, \ldots, e^{\boldsymbol{\xi}^{(N)}\cdot\boldsymbol{x}^{(j)}}\right) = \left(e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}^{(j)}}\right)_{i=1}^{N}$ for $j = 1, 2, \ldots, N$ and construct a matrix $L = (\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_N) = \left(e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}^{(j)}}\right)_{i,\,j=1}^{N}$. Moreover we set a diagonal matrix $Q := \left(P(\boldsymbol{\xi}^{(i)})\delta_{ij}\right)_{i,\,j=1}^{N}$ with Kronecker's symbol δ_{ij} and two column vectors $\boldsymbol{l}(\boldsymbol{x}) := \left(e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}}\right)_{i=1}^{N}$ and $\boldsymbol{w}(\boldsymbol{x}) := \left(w_j(\boldsymbol{x})\right)_{j=1}^{N}$. Then we can write the linear system (15) as

$$Q\boldsymbol{l}(\boldsymbol{x}) = L\boldsymbol{w}(\boldsymbol{x}). \quad (16)$$

Therefore the weights $w_j(\boldsymbol{x})$, $j = 1, 2, \ldots, N$ are given by

$$\boldsymbol{w}(\boldsymbol{x}) = L^{-1} Q \boldsymbol{l}(\boldsymbol{x}) \quad (17)$$

provided that L is invertible. By using a vector $\boldsymbol{u} := \left(u(\boldsymbol{x}^{(j)})\right)_{j=1}^{N}$ the high order finite difference approximation (9) is thus represented by

$$\begin{aligned} P(\boldsymbol{\partial})u(\boldsymbol{x}) &= {}^t\boldsymbol{w}(\boldsymbol{x})\boldsymbol{u} + \varepsilon(\boldsymbol{x};\, P(\boldsymbol{\partial})u) \\ &= {}^t\left(L^{-1}Q\boldsymbol{l}(\boldsymbol{x})\right)\boldsymbol{u} + \varepsilon(\boldsymbol{x};\, P(\boldsymbol{\partial})u). \quad (18) \end{aligned}$$

EXPONENTIAL INTERPOLATION

We characterize our high order finite difference approximation (9) by showing a relation between an exponential interpolation and the high order finite difference approximation.

Let a function $\tilde{u}$ be a linear combination of exponential functions and be equal to the function u at each quadrature point $\boldsymbol{x}^{(j)}$ for $j = 1, 2, \ldots, N$. More specifically, there exist constants $b_i \in \boldsymbol{R}$, $i = 1, 2, \ldots, N$ such that

$$\begin{aligned} \tilde{u}(\boldsymbol{x}) &= \sum_{i=1}^{N} b_i e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}} = {}^t\boldsymbol{l}(\boldsymbol{x})\boldsymbol{b}, \\ \tilde{u}(\boldsymbol{x}^{(j)}) &= u(\boldsymbol{x}^{(j)}), \quad j = 1, 2, \ldots, N \end{aligned}$$

with a vector $\boldsymbol{b} := (b_i)_{i=1}^{N}$. We call the function $\tilde{u}$ an *exponential interpolation* in u at the point $\boldsymbol{x}^{(j)}$, $j = 1, 2, \ldots, N$. Since $\boldsymbol{u} = \left(\tilde{u}(\boldsymbol{x}^{(j)})\right)_{j=1}^{N} = ({}^t\boldsymbol{l}_j\boldsymbol{b})_{j=1}^{N} = {}^tL\boldsymbol{b}$, the coefficients of the linear combination become $\boldsymbol{b} = {}^tL^{-1}\boldsymbol{u}$. Therefore the exponential interpolation $\tilde{u}$ can be written by

$$\tilde{u}(\boldsymbol{x}) = {}^t\left(L^{-1}\boldsymbol{l}(\boldsymbol{x})\right)\boldsymbol{u}. \quad (19)$$

Now we operate $P(\boldsymbol{\partial})$ on both sides of the formula (19). From the equality $P(\boldsymbol{\partial})\boldsymbol{l}(\boldsymbol{x}) = \left(P(\boldsymbol{\partial})e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}}\right)_{i=1}^{N} = \left(P(\boldsymbol{\xi}^{(i)})e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}}\right)_{i=1}^{N} = Q\boldsymbol{l}(\boldsymbol{x})$, we obtain

$$P(\boldsymbol{\partial})\tilde{u}(\boldsymbol{x}) = {}^{t}\left(L^{-1}Q\boldsymbol{l}(\boldsymbol{x})\right)\boldsymbol{u}. \qquad (20)$$

Furthermore, from an equality ${}^{t}\left(L^{-1}Q\boldsymbol{l}(\boldsymbol{x})\right) = {}^{t}\left(L^{-1}\boldsymbol{l}(\boldsymbol{x})\right){}^{t}(L^{-1}QL)$, the equation (20) becomes

$$P(\boldsymbol{\partial})\tilde{u}(\boldsymbol{x}) = {}^{t}(L^{-1}\boldsymbol{l}(\boldsymbol{x})){}^{t}(L^{-1}QL)\boldsymbol{u}.$$

We see from this expression and (19) that the matrix $\hat{P} := {}^{t}(L^{-1}QL)$ is a counterpart to the differential operator $P(\boldsymbol{\partial})$ through the exponential interpolation. In Figure 1 we illustrate the equivalence of the matrix $\hat{P}$ and the differential operator $P(\boldsymbol{\partial})$ inside the space $\Lambda_N := \mathrm{span}\left\{e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}} : i = 1, 2, \ldots, N\right\}$.

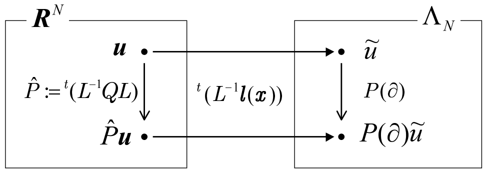

Figure 1: Equivalence of $\hat{P}$ and $P(\boldsymbol{\partial})$

Here we illustrate an approximation of a derivative numerically by using the exponential interpolation. From the equality (20), the high order finite difference approximation (18) is represented by

$$P(\boldsymbol{\partial})u(\boldsymbol{x}) = P(\boldsymbol{\partial})\tilde{u}(\boldsymbol{x}) + \varepsilon(\boldsymbol{x};\, P(\boldsymbol{\partial})u). \qquad (21)$$

When the dimension $m = 2$, we randomly locate quadrature points $\boldsymbol{x}^{(j)} \in (-0.5,\, 0.5) \times (-0.5,\, 0.5)$, $j = 1, 2, \ldots, N$ and set $\boldsymbol{\xi}^{(i)} = \boldsymbol{x}^{(i)}$, $i = 1, 2, \ldots, N$. We take a function $u(\boldsymbol{x}) = \sin 3x_1 \sin 3x_2$ and a differential operator $P(\boldsymbol{\partial}) = \Delta$ the Laplacian. In Figure 2 the approximation $P(\boldsymbol{\partial})\tilde{u}(\boldsymbol{x}) = {}^{t}(L^{-1}\boldsymbol{l}(\boldsymbol{x}))\hat{P}\boldsymbol{u}$ for $N = 100$ and the derivative $P(\boldsymbol{\partial})u(\boldsymbol{x}) = -18\sin 3x_1 \sin 3x_2$ are presented by their contour lines. We see that the approximation agrees well with the true derivative. In fact, the maximum error $\max\limits_{j=1,2,\ldots,N} |P(\boldsymbol{\partial})u(\boldsymbol{x}^{(j)}) - P(\boldsymbol{\partial})\tilde{u}(\boldsymbol{x}^{(j)})|$ is 7.0×10^{-2}. From this numerical example we can expect that the error in the derivative is small in the high order finite difference approximation.

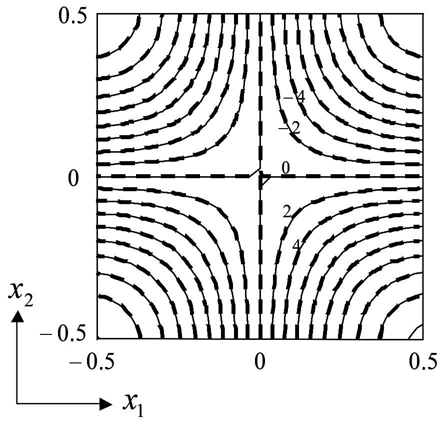

Figure 2: $P(\boldsymbol{\partial})\tilde{u}(\boldsymbol{x})$ (—) and $P(\boldsymbol{\partial})u(\boldsymbol{x})$ (- - -)

HIGH ORDER FINITE DIFFERENCE METHOD

We use the finite difference approximation (9) in our method to solve backward heat conduction problems (1)–(3) numerically.

Let quadrature points $\boldsymbol{x}^{(k)}$, $k = 1, 2, \ldots, N$ belong to the closure $\bar{\Omega}$ of the domain Ω. We set differential operators

$$P_k(\boldsymbol{\partial}) := \begin{cases} \boldsymbol{\partial}^{(0,0,1)} - \left(\boldsymbol{\partial}^{(2,0,0)} + \boldsymbol{\partial}^{(0,2,0)}\right), & \boldsymbol{x}^{(k)} \in \Omega, \\ I, & \boldsymbol{x}^{(k)} \in \Gamma_{B1} \cup \Gamma_F, \\ n_1(\boldsymbol{x}^{(k)})\boldsymbol{\partial}^{(1,0,0)} + n_2(\boldsymbol{x}^{(k)})\boldsymbol{\partial}^{(0,1,0)}, & \boldsymbol{x}^{(k)} \in \Gamma_{B2}, \end{cases}$$

and data

$$f_k := \begin{cases} 0, & \boldsymbol{x}^{(k)} \in \Omega, \\ \bar{u}(\boldsymbol{x}^{(k)}), & \boldsymbol{x}^{(k)} \in \Gamma_{B1}, \\ \bar{q}(\boldsymbol{x}^{(k)}), & \boldsymbol{x}^{(k)} \in \Gamma_{B2}, \\ u_F(\boldsymbol{x}^{(k)}), & \boldsymbol{x}^{(k)} \in \Gamma_F, \end{cases}$$

for $k = 1, 2, \ldots, N$, where I denotes the identity operator. We restrict the domain Ω considered in the problem (1)–(3) onto the set of quadrature points $\boldsymbol{x}^{(k)}$, $k = 1, 2, \ldots, N$ to obtain

$$P_k(\boldsymbol{\partial})u(\boldsymbol{x}^{(k)}) = f_k, \quad k = 1, 2, \ldots, N. \qquad (22)$$

Let u_j be an approximate value of the true $u(\boldsymbol{x}^{(j)})$ for $j = 1, 2, \ldots, N$. We set vectors $\boldsymbol{\xi}^{(i)} = \rho\boldsymbol{x}^{(i)}$, $i = 1, 2, \ldots, N$ for a parameter $\rho > 0$. Then we consider finding the approximate values u_j, $j = 1, 2, \ldots, N$ from the given data f_k, $k = 1, 2, \ldots, N$ based on the equalities (22).

Let $k \in \{1, 2, \ldots, N\}$ be fixed arbitrarily. From the high order finite difference approximation

$$P_k(\boldsymbol{\partial})u(\boldsymbol{x}^{(k)}) \approx \sum_{j=1}^{N} w_{kj}u(\boldsymbol{x}^{(j)}) \tag{23}$$

we can calculate weights w_{kj}, $j = 1, 2, \ldots, N$ as follows. Since weights in the approximation (9) are determined as a solution of the equation (15), we set weights w_{kj}, $j = 1, 2, \ldots, N$ as a solution of the linear system

$$P_k(\rho\boldsymbol{x}^{(i)})e^{\rho\boldsymbol{x}^{(i)}\cdot\boldsymbol{x}^{(k)}} = \sum_{j=1}^{N} w_{kj}e^{\rho\boldsymbol{x}^{(i)}\cdot\boldsymbol{x}^{(j)}}, \tag{24}$$

$$i = 1, 2, \ldots, N.$$

From the equations (22) and the approximation (23) it is suitable to find approximate values u_j, $j = 1, 2, \ldots, N$ as the solution of the linear system

$$\sum_{j=1}^{N} w_{kj}u_j = f_k, \quad k = 1, 2, \ldots, N. \tag{25}$$

Setting a matrix $W := (w_{kj})_{k,\,j=1}^{N}$ and two column vectors $\hat{\boldsymbol{u}} := (u_j)_{j=1}^{N}$, $\boldsymbol{f} := (f_k)_{k=1}^{N}$, the linear system (25) is represented by $W\hat{\boldsymbol{u}} = \boldsymbol{f}$. Therefore approximate values u_j, $j = 1, 2, \ldots, N$ are given by

$$\hat{\boldsymbol{u}} = W^{-1}\boldsymbol{f}. \tag{26}$$

We call the above method a *high order finite difference method.*

NUMERICAL RESULTS

We apply the high order finite difference method to the backward heat conduction problem (1)–(3).

Heat Conduction with a Fin

Let the domain D be $\{{}^t(x_1, x_2) : x_1^2 + x_2^2 < 0.5^2\} \cup (0, 1) \times (-0.25, 0.25)$ and the final time be $T = 1$. We set $\Gamma_{B2} = \{{}^t(x_1, x_2, x_3) \in \partial\Omega : x_1^2 + x_2^2 = 0.5^2\}$ and $\Gamma_{B1} = \Gamma_B \setminus \Gamma_{B2}$. The outward unit normal $\boldsymbol{n}(\boldsymbol{x})$ of Γ_{B2} is represented by ${}^t(2x_1, 2x_2, 0)$. For a parameter $\boldsymbol{y} = {}^t(y_1, y_2, y_3) \notin \Omega$ the function

$$G(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{4\pi(x_3 - y_3)}\mathrm{Exp}\left[-\frac{(x_1 - y_1)^2 + (x_2 - y_2)^2}{4(x_3 - y_3)}\right]$$

with respect to $\boldsymbol{x}$ satisfies the heat conduction equation (1). For $\boldsymbol{y} = {}^t(0, 0, -0.3)$ we give the boundary data and the final data exactly as $\bar{u}(\boldsymbol{x}) = G(\boldsymbol{x}, \boldsymbol{y})$ at $\boldsymbol{x} \in \Gamma_{B1}$, $\bar{q}(\boldsymbol{x}) = \dfrac{\partial G}{\partial n(\boldsymbol{x})}(\boldsymbol{x}, \boldsymbol{y})$ at $\boldsymbol{x} \in \Gamma_{B2}$, and $u_F(\boldsymbol{x}) = G(\boldsymbol{x}, \boldsymbol{y})$ at $\boldsymbol{x} \in \Gamma_F$. Then we calculate numerical solutions of the problem (1)–(3) by using the high order finite difference method for $N = 500$ and $\rho = 3$.

We show the numerical solution and the exact solution in Figure 3 at $x_3 = 0.5$ and in Figure 4 at $x_3 = 0$. At $x_3 = 0$ the error of the

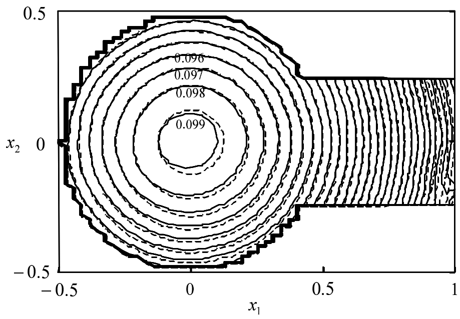

Figure 3: Numerical(—) and exact(- - -) solutions ($x_3 = 0.5$)

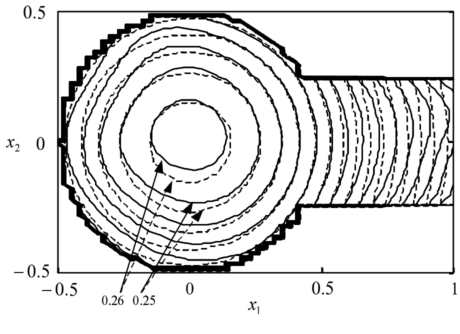

Figure 4: Numerical(—) and exact(- - -) solutions ($x_3 = 0$)

numerical solution is err $= 4.9 \times 10^{-3}$. Accordingly, for the problem in which the domain is not rectangle, we can obtain the accurate numerical solution.

Heat Conduction in a Square

Let the domain D be $(-0.5, 0.5) \times (-0.5, 0.5)$ and let the final time be $T = 1$. We set boundaries $\Gamma_{B1} = \{{}^t(x_1, x_2, x_3) \in \partial\Omega : x_1 = -0.5, 0.5\}$ and $\Gamma_{B2} = \{{}^t(x_1, x_2, x_3) \in \partial\Omega : x_2 = -0.5, 0.5\}$. For a parameter $l \in \boldsymbol{N}$ the function

$$u^{(l)}(\boldsymbol{x}) = e^{-2l^2 x_3} \sin l x_1 \sin l x_2$$

satisfies the heat conduction equation (1). We prescribe the boundary data and the final data exactly as $\bar{u}^{(l)}(\boldsymbol{x}) = u^{(l)}(\boldsymbol{x})$ at $\boldsymbol{x} \in \Gamma_{B1}$, $\bar{q}^{(l)}(\boldsymbol{x}) = \dfrac{\partial u^{(l)}}{\partial n(\boldsymbol{x})}(\boldsymbol{x})$ at $\boldsymbol{x} \in \Gamma_{B2}$, and $u_F^{(l)}(\boldsymbol{x}) = u^{(l)}(\boldsymbol{x})$ at $\boldsymbol{x} \in \Gamma_F$. Then we calculate numerical solutions of the problem (1)–(3) by using the high order finite difference method for the number $N = 500$ of quadrature points.

In Figure 5 the distribution of quadrature points is presented.

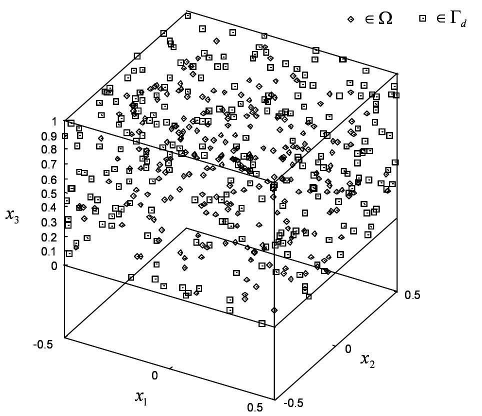

Figure 5: Quadrature points

We show the exact solution and the numerical solution for $\rho = 4$ at $x_3 = 0$ in Figures 6–8 for $l = 1, 2, 3$, respectively. Each error in the numerical solutions is 1.6×10^{-4}, 2.4×10^{-2}, and 8.2 respectively for $l = 1, 2, 3$, where error is defined by err $:= \max_{j=1,2,\ldots,N} |u(\boldsymbol{x}^{(j)}) - u_j|$. We observe an increase in the error of the numerical solutions as the parameter l becomes large. In particular the error on the boundary Γ_{B2} is larger than the error in the vicinity of the boundary Γ_{B1}.

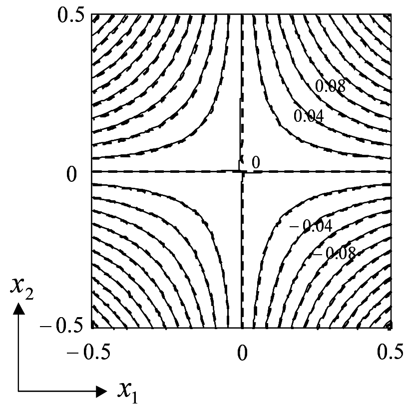

Figure 6: Numerical(—) and exact(- - -) solutions($l = 1$, $x_3 = 0$)

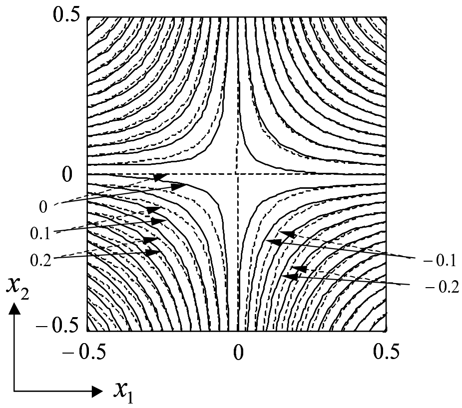

Figure 7: Numerical(—) and exact(- - -) solutions ($l = 2$, $x_3 = 0$)

In Figure 8 the difference between the numerical solution and the exact solution is considerably large. From the estimation (4) the ratio between the solution $u^{(l)}$ and the final data $u_F^{(l)}$ is $C_l = \dfrac{1}{2l}\sqrt{e^{4l^2T} - 1} = O\left(\dfrac{e^{2l^2}}{l}\right)$ with respect to L^2 norm. For $l = 2, 3$ the ratios are estimated as $C_2 \approx 700$ and $C_3 \approx 10^7$. As a source of the large error in the numerical solution for $l = 3$ we guess an accumulation of round-off errors in the computational arithmetic.

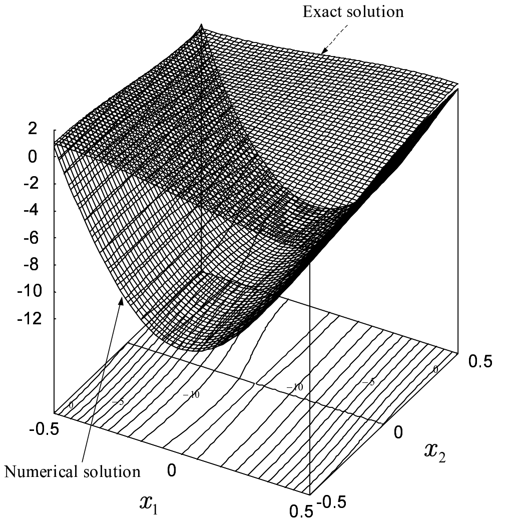

Figure 8: Numerical and exact solutions ($l = 3$, $x_3 = 0$)

CONCLUSIONS

We considered a high order finite difference method in order to solve the backward heat conduction problem. The high order finite difference approximation is based on the idea that the derivative of an unknown regular function can be approximated with high accuracy by a linear combination of values of the function at quadrature points. Since we use the quadrature points which can be chosen at arbitrary locations, the method gains a meshless property. It is shown that the approximation coincides with the derivative of the exponential interpolation. In numerical experiments, two-dimensional backward heat conduction problem was solved as three-dimensional problem in the space-time domain. When magnification of the solution to the final data is very large, we guess that the numerical solution can conceivably influenced by round-off errors. We confirmed that our method is applicable to the problem in the domain with curved boundary under the mixed boundary conditions.

REFERENCES

1. M. Yamamoto, *Introduction to Inverse Problems* (in Japanese), Iwanami-Shoten, Tokyo, 2002, p. 77.

2. R. Kress, *Linear Integral Equations*, Second Edition, Applied Mathematical Sciences, **82**, Springer-Verlag, New York, 1989.

3. R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-value Problems*, Second edition, Interscience Tracts in Pure and Applied Mathematics, **4**, Interscience Publishers, New York, London, Sydney, 1967.

4. H. Imai, T. Takeuchi, and M. Kushida, On numerical simulation of partial differential equations in infinite precision, *Advances in Mathematical Sciences and Applications*, Gakkōtosho, Tokyo, **9-2**, 1007 (1999).

5. H. Fujiwara and Y. Iso, Numerical challenge to ill-posed problems by fast multiple-precision system, *Theoretical and Applied Mechanics Japan*, **50**, 419 (2001).

6. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1988.