NUMERICAL SOLUTION OF BACKWARD HEAT CONDUCTION PROBLEMS BY A HIGH ORDER LATTICE-FREE FINITE DIFFERENCE METHOD

Kentaro Iijima

ABSTRACT

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 method we show two tools using Fourier transform and Taylor expansion, respectively. On the other hand, the backward heat conduction problem is a typical example of ill-posed problems in the sense that the solution is unstable for errors of data. Our aim is creation of a meshless method which can be applied to the ill-posed problem. From numerical experiments we confirmed that our method is effective in solving two-dimensional backward heat conduction equations subject to the Dirichlet boundary condition.

Key Words: high order finite difference method, inverse problem, meshless method.

I. 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 domain Ω is written by $\boldsymbol{x}={}^t(x_1, x_2, t)={}^t(x_1, x_2, x_3)$. We set two surfaces $\Gamma_B=\partial D\times[0, T]$ and $\Gamma_F=D\times\{T\}$ of the boundary $\partial\Omega$. A symbol Δ denotes the Laplacian $\frac{\partial^2}{\partial x_1^2}+\frac{\partial^2}{\partial x_2^2}$. Then for given boundary data u_B: $\Gamma_B\to\boldsymbol{R}$ and final data u_F: $\Gamma_F\to\boldsymbol{R}$, we consider a problem to look for a function u such that

$$\frac{\partial u}{\partial t}=\Delta u \text{ in } \Omega \tag{1}$$

$$u=u_B \text{ on } \Gamma_B \tag{2}$$

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

We call the problem (1)-(3) a two-dimensional backward heat conduction problem.

The backward heat conduction problem is an ill-posed problem in the sense that the solution is unstable for the given final data u_F (Kress, 1989). We illustrate instability of the backward heat conduction problem: Let the domain D be $(-\pi, \pi)\times(-\pi, \pi)$. We give final data $u_F^{(n)}(\boldsymbol{x}):=e^{-2n^2T}\sin nx_1 \sin nx_2$, $\boldsymbol{x}\in\Gamma_F$ and boundary data $u_B^{(n)}(\boldsymbol{x})=0$, $\boldsymbol{x}\in\Gamma_B$ for $n\in\boldsymbol{N}$. Then the exact solution of the heat Eq. (1) is represented by $u^{(n)}(\boldsymbol{x})=e^{-2n^2t}\sin nx_1\sin nx_2$. We choose two L^2 norms

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

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

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 is estimated by

$$\begin{aligned}
&\| u^{(n)} \|^2_{L^2(\Omega)} \\
&= \int_\Omega (e^{-2n^2t}\sin nx_1 \sin nx_2)^2 d\boldsymbol{x} \\
&= \int_0^T e^{4n^2(T-t)}dt \times \int_D (e^{-2n^2T}\sin nx_1\sin nx_2)^2 d\boldsymbol{x}' \\
&= \frac{1}{4n^2}(e^{4n^2T}-1)\| u_F^{(n)} \|^2_{L^2(\Gamma_F)}
\end{aligned} \tag{4}$$

Since for any $C>0$ there exists $n\in\boldsymbol{N}$ such that $\frac{1}{2n}\sqrt{e^{4n^2T}-1}>C$, an inequality $\| u^{(n)} \|_{L^2(\Omega)}>$

K. Iijima is with the Graduate School of Science and Engineering, Ibaraki University, Mito 310-8512, Japan (Email: nd2401a@mcs.ipc.ibaraki.ac.jp)

$C\| u_F^{(n)} \|_{L^2(\Gamma_F)}$ holds for any $C>0$. This means that the solution does not depend on the final data continuously. Therefore the solution of the backward heat conduction problem is unstable for the final data with respect to the L^2 norm.

In order to solve the backward heat conduction problem numerically, we consider an application of conventional finite difference schemes. For any time step size $\Delta t>0$ in the time range $[0, T]$ and for any lattice width $\Delta x_1>0$ and $\Delta x_2>0$ in each direction of D we can show by the von Neumann condition (Richtmyer and Morton, 1967) that the following finite difference scheme approximating the Eq. (1) is unstable.

$$\frac{u(x_1, x_2, t) - u(x_1, x_2, t-\Delta t)}{\Delta t}$$
$$= \{u(x_1 + \Delta x_1, x_2, t) - 2u(x_1, x_2, t)$$
$$+ u(x_1 + \Delta x_1, x_2, t)\}/\Delta x_1^2$$
$$+ \{u(x_1, x_2 - \Delta x_2, t) - 2u(x_1, x_2, t)$$
$$+ u(x_1, x_2 + \Delta x_2, t)\}/\Delta x_2^2 \quad (5)$$

We state a motivation for research. There are researches which challenge numerical analysis of ill-posed problems. The techniques in these researches make discretization error and rounding error arbitrarily small by the spectral collocation method and an arbitrary precision arithmetic, respectively (Imai *et al.*, 1999; Fujiwara and Iso, 2001). The backward heat conduction problem is solved very precisely with no observation error by their techniques. However in the spectral collocation method we must take the Chebyshev-Gauss-Lobatto points (Canuto *et al.*, 1988). Therefore it is difficult to apply the technique to the problem in a domain with curved boundaries. As a method applicable to such problems, we propose a high order finite difference scheme which can choose quadrature points at arbitrary locations.

II. NOTATION

We introduce a set $\boldsymbol{Z}_+ := \{z \in \boldsymbol{Z}: z \geq 0\}$ and let $\boldsymbol{Z}_+^m = \overbrace{\boldsymbol{Z}_+ \times \boldsymbol{Z}_+ \times \cdots \times \boldsymbol{Z}_+}$, where $\boldsymbol{Z}$ denotes the set of all non-negative integers. Then an element $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_m) \in \boldsymbol{Z}_+^m$ is called a multi-index. A symbol $\boldsymbol{0}$ denotes $(0, \cdots, 0)$. For $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_m) \in \boldsymbol{Z}_+^m$, 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 $\boldsymbol{x} = {}^t(x_1, x_2, \cdots, x_m)$ be a vector in $\boldsymbol{R}^m$. We distinguish the length of a multi-index $|\cdot|$ from the length of 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 operator $\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}}$ denotes $\frac{\partial^{\alpha_1+\alpha_2+\cdots+\alpha_m}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_m^{\alpha_m}} = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} \cdots \frac{\partial^{\alpha_m}}{\partial x_m^{\alpha_m}}$. Setting $\partial_j = \frac{\partial}{\partial x_j}$ and $\boldsymbol{\partial} = (\partial_1, \partial_2, \cdots, \partial_m)$ formally, we write $\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \boldsymbol{\partial}^{\boldsymbol{\alpha}}$.

Taylor's theorem can be written as follows. Let Ω be a domain in $\boldsymbol{R}^m$ and let f be a function of class C^μ in Ω for some $\mu \in \boldsymbol{Z}_+$. Then for each $\boldsymbol{x} \in \Omega$ there exists $r>0$ such that

$$f(\boldsymbol{x} + \boldsymbol{h}) = \sum_{|\boldsymbol{\alpha}| \leq \mu - 1} \frac{\boldsymbol{h}^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \boldsymbol{\partial}^{\boldsymbol{\alpha}} f(\boldsymbol{x})$$
$$+ \sum_{|\boldsymbol{\alpha}| = \mu} \frac{\mu \boldsymbol{h}^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \int_0^1 (1-t)^{\mu-1} \boldsymbol{\partial}^{\boldsymbol{\alpha}} f(\boldsymbol{x} + t\boldsymbol{h}) dt$$

for all $h \in \boldsymbol{R}^m$ with $|\boldsymbol{h}|<r$, $\boldsymbol{x}+\boldsymbol{h} \in \Omega$.

III. FINITE DIFFERENCE APPROXIMATION

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

$$u(\boldsymbol{y}) = \sum_{\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m} \frac{(\boldsymbol{y} - \boldsymbol{x})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \partial^{\boldsymbol{\alpha}} u(\boldsymbol{x}), \ \boldsymbol{x}, \boldsymbol{y} \in \Omega \quad (6)$$

in the sense of absolute and uniform convergence. We take $N+1$ quadrature points $\boldsymbol{x} = {}^t(x_1, x_2, \cdots, x_m)$, $\boldsymbol{x}^{(j)} = {}^t(\boldsymbol{x}_1^{(j)}, \boldsymbol{x}_2^{(j)}, \cdots, \boldsymbol{x}_m^{(j)})$ in Ω for $j=1, 2, \cdots, N$ randomly. For real constants $a_{\boldsymbol{\alpha}}$, $\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m$, we set a differential operator $P(\boldsymbol{\partial})$ of the order μ_0 as

$$P(\boldsymbol{\partial}) := \sum_{\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m} a_{\boldsymbol{\alpha}} \boldsymbol{\partial}^{\boldsymbol{\alpha}} \quad (7)$$

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

$$P(\boldsymbol{\partial})u(\boldsymbol{x}) = \sum_{j=1}^N w_j(\boldsymbol{x}) u(\boldsymbol{x}^{(j)}) + \varepsilon(\boldsymbol{x}; u) \quad (8)$$

where $\varepsilon(\boldsymbol{x}; u)$ denotes the discretization error. When $\varepsilon(\boldsymbol{x}; u)$ is sufficiently small, we call the approximation (8) a high order finite difference approximation of $P(\boldsymbol{\partial})$ at $\boldsymbol{x}$ with respect to $\boldsymbol{x}^{(j)}$, $j=1, 2, \cdots, N$.

Concretely we can determine weights $w_j(\boldsymbol{x})$, $j=1, 2, \cdots, N$ in the equality (8). Substituting the operator (7) into the equality (8) we can see that the left hand side of the equality (8) becomes

$$P(\boldsymbol{\partial})u(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \boldsymbol{Z}_+^m} a_{\boldsymbol{\alpha}} \boldsymbol{\partial}^{\boldsymbol{\alpha}} u(\boldsymbol{x}), \ \boldsymbol{\alpha} \in \boldsymbol{Z}_+^m \quad (9)$$

From Taylor's expansion (6) the first term on the right hand side of the equality (8) becomes

$$\sum_{j=1}^{N} w_j(\boldsymbol{x})u(\boldsymbol{x}^{(j)})$$

$$=\sum_{j=1}^{N} w_j(\boldsymbol{x})\{\sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m}\frac{1}{\boldsymbol{\alpha}!}(\boldsymbol{x}^{(j)}-\boldsymbol{x})^{\boldsymbol{\alpha}}\boldsymbol{\partial}^{\boldsymbol{\alpha}}u(\boldsymbol{x})\} \tag{10}$$

From the approximation (8) and the above two equations, we obtain

$$\varepsilon(\boldsymbol{x}; u)$$

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

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

$$a_{\boldsymbol{\alpha}}=\sum_{j=1}^{N} w_j(\boldsymbol{x})\frac{1}{\boldsymbol{\alpha}!}(\boldsymbol{x}^{(j)}-\boldsymbol{x})^{\boldsymbol{\alpha}} ,\ |\boldsymbol{\alpha}|\le\mu \tag{11}$$

for the largest possible integer μ.

We present the following method in order to choose our weights: Let $\boldsymbol{\xi}^{(i)}$, i=1, 2, ⋯, N be vectors in $\boldsymbol{R}^m$ such that $\boldsymbol{\xi}^{(i)}\neq\boldsymbol{\xi}^{(j)}$ for $i\neq j$. Multiplying by $(\boldsymbol{\xi}^{(i)})^{\boldsymbol{\alpha}}$ and summing for all $\alpha\in\boldsymbol{Z}_+^m$, the equality (11) becomes

$$\sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m} a_{\boldsymbol{\alpha}}(\boldsymbol{\xi}^{(i)})^{\boldsymbol{\alpha}}$$

$$=\sum_{j=1}^{N} w_j(\boldsymbol{x})\{\sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m}\frac{1}{\boldsymbol{\alpha}!}(\boldsymbol{\xi}^{(i)})^{\boldsymbol{\alpha}}(\boldsymbol{x}^{(j)}-\boldsymbol{x})^{\boldsymbol{\alpha}}\} \tag{12}$$

Generally, the equality

$$\sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m}\frac{1}{\boldsymbol{\alpha}!}\boldsymbol{\xi}^{\boldsymbol{\alpha}}\boldsymbol{x}^{\boldsymbol{\alpha}}=\sum_{\boldsymbol{\alpha}\in \boldsymbol{Z}_+^m}\prod_{k=1}^{m}\frac{1}{\alpha_k!}\xi_k^{\alpha_k}x_k^{\alpha_k}$$

$$=\prod_{k=1}^{m}\sum_{\alpha_k=0}^{\infty}\frac{1}{\alpha_k!}\xi_k^{\alpha_k}x_k^{\alpha_k}$$

$$=\prod_{k=1}^{m} e^{\xi_k x_k}$$

$$=e^{\boldsymbol{\xi}\cdot\boldsymbol{x}},\quad \boldsymbol{x},\ \boldsymbol{\xi}\in\boldsymbol{R}^m \tag{13}$$

holds. By the equality (13) 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 (12) is transformed into

$$P(\boldsymbol{\xi}^{(i)})e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}}=\sum_{j=1}^{N} w_j(\boldsymbol{x})e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}^{(j)}},\ i=1, 2, \cdots, N \tag{14}$$

Setting matrices $L=(\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_N):=(e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}^{(j)}})_{i,j=1}^N$, $Q:=(P(\boldsymbol{\xi}^{(i)})\delta_{ij})_{i,j=1}^N$ and column vectors $\boldsymbol{l}(\boldsymbol{x}):=(e^{\boldsymbol{\xi}^{(i)}\cdot\boldsymbol{x}})_{i=1}^N$, $\boldsymbol{w}(\boldsymbol{x}):=(w_j(\boldsymbol{x}))_{j=1}^N$, we can write the linear system (14) as

$$Q\boldsymbol{l}(\boldsymbol{x})=L\boldsymbol{w}(\boldsymbol{x}) \tag{15}$$

where δ_{ij} denotes the Kronecker delta. Therefore the weights $w_j(\boldsymbol{x})$, j=1, 2, ⋯, N are given by

$$w(\boldsymbol{x})=L^{-1}Q\boldsymbol{l}(\boldsymbol{x}) \tag{16}$$

if L has the inverse.

By using a vector $\boldsymbol{u}:=(u(\boldsymbol{x}^{(j)}))_{j=1}^N$ the high order finite difference approximation (8) is represented by

$$P(\boldsymbol{\partial})u(\boldsymbol{x})={}^t\boldsymbol{w}(\boldsymbol{x})\boldsymbol{u}+\varepsilon(\boldsymbol{x};\ u)$$

$$={}^t(L^{-1}Q\boldsymbol{l}(\boldsymbol{x}))\boldsymbol{u}+\varepsilon(\boldsymbol{x};\ u) \tag{17}$$

We have not obtained even a sufficient condition for the existence of the inverse of L. However we know that the determinant $|L|$ of L becomes 0 under the following condition:

There exist $i, j\in\{1, 2, \cdots, N\}$, $i\neq j$ and $\boldsymbol{\eta}, \boldsymbol{\eta}'\in\boldsymbol{R}^m$ such that the vectors $\{\boldsymbol{\xi}^{(p)}\}$ and the points $\{\boldsymbol{x}^{(q)}\}$ satisfy either

$$(\boldsymbol{\xi}^{(k)}-\boldsymbol{\eta})\perp(\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)})\ \text{ or }\ (\boldsymbol{\xi}^{(i)}-\boldsymbol{\xi}^{(j)})\perp(\boldsymbol{x}^{(k)}-\boldsymbol{\eta}')$$

$$\text{for } k=1;\ 2, \cdots, N \tag{18}$$

The first condition of (18) implies that N points $\boldsymbol{\xi}^{(k)}$, k=1, 2, ⋯, N are contained in a plane that contain $\boldsymbol{\eta}$ and the vector $\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)}$ is perpendicular to the plane. In fact, we assume that the first condition in (18) is satisfied. Then the determinant $|L|$ becomes

$$|L|=\left|(e^{\boldsymbol{\xi}^{(p)}\cdot\boldsymbol{x}^{(q)}})_{p,q=1}^N\right|$$

$$=\left|(e^{\boldsymbol{\xi}^{(p)}\cdot\boldsymbol{x}^{(j)}}\delta_{pq})_{p,q=1}^N(e^{\boldsymbol{\xi}^{(p)}\cdot(\boldsymbol{x}^{(q)}\cdot\boldsymbol{x}^{(j)})})_{p,q=1}^N\right|$$

$$=\left|(e^{\boldsymbol{\xi}^{(p)}\cdot\boldsymbol{x}^{(j)}}\delta_{pq})_{p,q=1}^N(e^{(\boldsymbol{\xi}^{(p)}-\boldsymbol{\eta})\cdot(\boldsymbol{x}^{(q)}\cdot\boldsymbol{x}^{(j)})})_{p,q=1}^N\right.$$

$$\left.\cdot\ (e^{\boldsymbol{\eta}\cdot(\boldsymbol{x}^{(q)}-\boldsymbol{x}^{(j)})}\delta_{pq})_{p,q=1}^N\right|$$

$$=\prod_{p=1}^{N} e^{\boldsymbol{\xi}^{(p)}\cdot\boldsymbol{x}^{(j)}}\left|(e^{(\boldsymbol{\xi}^{(p)}-\boldsymbol{\eta})\cdot(\boldsymbol{x}^{(q)}-\boldsymbol{x}^{(j)})})_{p,q=1}^N\right|$$

$$\times\prod_{q=1}^{N} e^{\boldsymbol{\eta}\cdot(\boldsymbol{x}^{(q)}-\boldsymbol{x}^{(j)})}$$

In the matrix $(e^{(\boldsymbol{\xi}^{(p)}-\boldsymbol{\eta})\cdot(\boldsymbol{x}^{(q)}-\boldsymbol{x}^{(j)})})_{p,q=1}^N$ the jth column vector (as $q=j$) becomes $(1)_{p=1}^N$. From the condition

(18) the ith column vector (as $q=i$) also becomes $(1)_{p=1}^{N}$. Therefore $|L|=0$ follows.

IV. EXPONENTIAL INTERPOLATION

We show a relation between the high order finite difference approximation and an exponential interpolation.

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

$$\tilde{u}(\boldsymbol{x}) = \sum_{i=1}^{N} b_i e^{\boldsymbol{\xi}^{(i)}\cdot \boldsymbol{x}} = {}^t l(\boldsymbol{x}) b \,,$$

$$\tilde{u}(\boldsymbol{x}^{(j)}) = u(\boldsymbol{x}^{(j)}) \,, \quad j=1, 2, \cdots, N$$

where $\boldsymbol{b}:=(b_i)_{i=1}^{N}$ is an N-vector to be determined. We call the function $\tilde{u}$ an exponential interpolation formula of the N-vector $\boldsymbol{u}=(u(\boldsymbol{x}^{(j)}))_{j=1}^{N}$. Since $\boldsymbol{u}=(\tilde{u}(\boldsymbol{x}^{(j)}))_{j=1}^{N}=({}^t\boldsymbol{l}_j\boldsymbol{b})_{j=1}^{N}={}^tL\boldsymbol{b}$, the coefficients of the linear combination become $\boldsymbol{b}={}^tL^{-1}\boldsymbol{u}$. Therefore the linear combination can be written by

$$\tilde{u}(\boldsymbol{x})={}^t\boldsymbol{l}(\boldsymbol{x})\,{}^tL^{-1}\boldsymbol{u}={}^t(L^{-1}\boldsymbol{l}(\boldsymbol{x}))\boldsymbol{u} \tag{19}$$

Here we make $P(\boldsymbol{\partial})$ operate on both sides of the formula (19). From the equality

$$P(\boldsymbol{\partial})\boldsymbol{l}(\boldsymbol{x}) = (P(\boldsymbol{\partial})e^{\boldsymbol{\xi}^{(i)}\cdot \boldsymbol{x}})_{i=1}^{N}$$

$$= (P(\boldsymbol{\xi}^{(i)})e^{\boldsymbol{\xi}^{(i)}\cdot \boldsymbol{x}})_{i=1}^{N} = Q\boldsymbol{l}(\boldsymbol{x})$$

we obtain

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

Furthermore, from

$${}^t(L^{-1}Q\boldsymbol{l}(\boldsymbol{x}))={}^t(L^{-1}QLL^{-1}\boldsymbol{l}(\boldsymbol{x}))={}^t(L^{-1}\boldsymbol{l}(\boldsymbol{x}))\,{}^t(L^{-1}QL)$$

the Eq. (20) can be written as

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

Therefore we can see that the matrix $\hat{\boldsymbol{P}}:={}^t(L^{-1}QL)$ is equivalent to the differential operator $P(\boldsymbol{\partial})$ via exponential interpolation (19). In Fig. 1 we illustrate the equivalence of $\hat{P}$ and $P(\boldsymbol{\partial})$, where $\Lambda_N:=\mathrm{span}\{e^{\boldsymbol{\xi}^{(i)}\cdot \boldsymbol{x}}: i=1, 2, \cdots, N\}$. From the equality (20) the high order finite difference approximation (17) is represented by

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

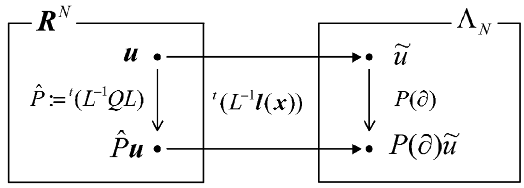

Fig. 1 Equivalence of $\hat{P}$ and $P(\boldsymbol{\partial})$

V. DEDUCTION VIA FOURIER TRANSFORM

We deduce the high order finite difference approximation from another viewpoint by using the Fourier transform.

From Plancherel's theorem (Ito, 1963) the Fourier transform

$$\mathcal{F}[u](\boldsymbol{\xi}) := \int_{\boldsymbol{R}^m} u(\boldsymbol{x})e^{-\sqrt{-1}\boldsymbol{\xi}\cdot \boldsymbol{x}}d\boldsymbol{x} \,, \quad \boldsymbol{\xi}\in \boldsymbol{R}^m$$

is a bijective operator on $L^1(\boldsymbol{R}^m)\cap L^2(\boldsymbol{R}^m)$. We assume that $\boldsymbol{\partial}^{\boldsymbol{\alpha}}u\in L^1(\boldsymbol{R}^m)\cap L^2(\boldsymbol{R}^m)$ for all $|\boldsymbol{\alpha}|\le\mu_0$. The Fourier transform is linear and it has properties

$$\mathcal{F}[P(\boldsymbol{\partial})u](\boldsymbol{\xi})=P(\sqrt{-1}\,\boldsymbol{\xi})\mathcal{F}[u](\boldsymbol{\xi}) \tag{22}$$

$$\mathcal{F}[u(\cdot+\boldsymbol{h})](\boldsymbol{\xi})=e^{-\sqrt{-1}\boldsymbol{\xi}\cdot \boldsymbol{h}}\mathcal{F}[u](\boldsymbol{\xi}) \tag{23}$$

We assume $u\not\equiv 0$ and take a function $v(\boldsymbol{x}+\boldsymbol{y}):=u(\boldsymbol{x})$ for $\boldsymbol{y}\in \boldsymbol{R}^m$. Then the approximation (8) is represented as

$$P(\boldsymbol{\partial})v(\boldsymbol{x}+\boldsymbol{y}) \approx \sum_{j=1}^{N} w_j(\boldsymbol{x})v(\boldsymbol{x}^{(j)}+\boldsymbol{y}) \tag{24}$$

By operating the Fourier transform on each side of the approximation (24) with respect to $\boldsymbol{y}$, the left and right hand sides become

$$\mathcal{F}[P(\boldsymbol{\partial})v(\boldsymbol{x}+\cdot\,)](\boldsymbol{\xi}) = P(\sqrt{-1}\,\boldsymbol{\xi})e^{-\sqrt{-1}\boldsymbol{\xi}\cdot \boldsymbol{x}}\mathcal{F}[v](\boldsymbol{\xi})$$

and

$$\mathcal{F}[\sum_{j=1}^{N} w_j(\boldsymbol{x})v(\boldsymbol{x}^{(j)}+\cdot\,)](\boldsymbol{\xi})$$

$$= \sum_{j=1}^{N} w_j(\boldsymbol{x})e^{\sqrt{-1}\boldsymbol{\xi}\cdot \boldsymbol{x}^{(j)}}\mathcal{F}[v](\boldsymbol{\xi})$$

respectively. When the sides are equated with each other for $\boldsymbol{\xi}=\boldsymbol{\xi}^{(i)}$, $i=1\ 2, \cdots, N$, equations

$$P(\sqrt{-1}\,\boldsymbol{\xi}^{(i)})e^{\sqrt{-1}\boldsymbol{\xi}\cdot \boldsymbol{x}^{(i)}}\mathcal{F}[v](\boldsymbol{\xi}^{(i)})$$

$$= \sum_{j=1}^{N} w_j(\boldsymbol{x})e^{\sqrt{-1}\boldsymbol{\xi}^{(i)}\cdot \boldsymbol{x}^{(j)}}\mathcal{F}[v](\boldsymbol{\xi}^{(i)}) \,, \quad i=1, 2, \cdots, N$$

follow. If the vector $\boldsymbol{\xi}^{(i)}$ is chosen as $\mathcal{F}[v](\boldsymbol{\xi}^{(i)}) \neq 0$ for i=1, 2, ⋯, N, the above equations become

$$P(\sqrt{-1}\,\boldsymbol{\xi}^{(i)})e^{\sqrt{-1}\,\xi^{(i)}\cdot\boldsymbol{x}} = \sum_{j=1}^{N} w_j(\boldsymbol{x})e^{\sqrt{-1}\,\xi^{(i)}\cdot\boldsymbol{x}^{(j)}},$$

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

Replacing $\sqrt{-1}\,\xi^{(i)}$ with $\boldsymbol{\xi}^{(i)}$ for i=1, 2, ⋯, N formally, we obtain the linear system (14).

VI. HIGH ORDER FINITE DIFFERENCE METHOD

We use the finite difference approximation (8) in a method to solve backward heat conduction problems numerically.

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

$$P_k(\boldsymbol{\partial}) := \begin{cases} \boldsymbol{\partial}^{(0,0,1)} - (\boldsymbol{\partial}^{(2,0,0)} + \partial^{(0,2,0)}), & \boldsymbol{x}^{(k)} \in \Omega \\ I, & \boldsymbol{x}^{(k)} \in \Gamma_B \cup \Gamma_F, \end{cases}$$

and data

$$f_k := \begin{cases} 0, & \boldsymbol{x}^{(k)} \in \Omega \\ u_B(\boldsymbol{x}^{(k)}), & \boldsymbol{x}^{(k)} \in \Gamma_B \\ u_F(\boldsymbol{x}^{(k)}), & \boldsymbol{x}^{(k)} \in \Gamma_F \end{cases},$$

for k=1, 2, ⋯, N, where I denotes the identity operator. We restrict the domain Ω considered in the problem (1)-(3) to quadrature points $\boldsymbol{x}^{(k)}$, k=1, 2, ⋯, N as

$$P_k(\boldsymbol{\partial})u(\boldsymbol{x}^{(k)})=f_k,\ k=1, 2, \cdots, N \tag{25}$$

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

Let $S_k \subset \boldsymbol{R}^m$ be a neighborhood of each quadrature point $\boldsymbol{x}^{(k)}$ for k=1, 2, ⋯, N, where their union $\bigcup_{k=1}^{N} S_k$ includes the closed domain $\overline{\Omega}$. Let M_k be the number of elements which belong to the set $N(S_k)$:= $\{j \in \{1, 2, \cdots, N\}: \boldsymbol{x}^{(j)} \in S_k\}$ for k=1, 2, ⋯, N. We make a table $\{m_{kj}\}$ such that $m_{kj} \in N(S_k)$, j=1, 2, ⋯, M_k, k=1, 2, ⋯, N.

Let $k \in \{1, 2, \cdots, N\}$ be fixed arbitrarily. From an approximation

$$P_k(\boldsymbol{\partial})u(\boldsymbol{x}^{(k)}) \approx \sum_{j \in N(S_k)} w_{kj}u(\boldsymbol{x}^{(j)}) \tag{26}$$

we calculate weights w_{kj}, $j \in \{m_{k1}, m_{k2}, \cdots, m_{kM_k}\} = N(S_k)$. Since weights in the approximation (8) are determined as a solution of the Eq. (14), we set weights w_{kj}, $j \in N(S_k)$ as a solution of the linear system

$$P_k(\rho\boldsymbol{x}^{(i)})e^{\rho\boldsymbol{x}^{(i)}\cdot\boldsymbol{x}^{(k)}} = \sum_{j \in N(S_k)} w_{kj}e^{\rho\boldsymbol{x}^{(i)}\cdot\boldsymbol{x}^{(j)}},\ i \in N(S_k) \tag{27}$$

and w_{kj}=0, $j \notin N(S_k)$. When k moves through 1, 2, ⋯, N, from the Eqs. (25) and the approximation (26) it is suitable to define approximate values u_j, j=1, 2, ⋯, N as the solution of the linear system

$$\sum_{j=1}^{N} w_{kj}u_j = f_k,\ k=1; 2; \cdots; N \tag{28}$$

Setting a matrix $W:=(w_{kj})_{k,\,j=1}^{N}$ and vectors $\hat{\boldsymbol{u}}:=(u_j)_{j=1}^{N}$, $\boldsymbol{f}:=(f_k)_{k=1}^{N}$, the linear system (28) is represented by $W\hat{\boldsymbol{u}}=\boldsymbol{f}$. Therefore approximate values u_j, j=0, 1, ⋯, N–1 are obtained by

$$\hat{\boldsymbol{u}}=W^{-1}\boldsymbol{f} \tag{29}$$

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

VII. NUMERICAL RESULTS

We apply the high order finite difference method to the backward heat conduction problem (1)-(3). In the high order finite difference method we choose such a neighborhood, S_k, that includes all quadrature points for k=1, 2, ⋯, N in this paper. Therefore the set $N(S_k)$ coincides with {1, 2, ⋯, N}.

Example 1

We compare the precision in numerical solutions between the following direct and inverse problems. We choose the domain D=(−0.5, 0.5)×(−0.5, 0.5) and a time interval (T_I, T_F)=(−0.5, 0.5), where T_I and T_F denote the initial time and the final time, respectively. We write each surface of the space-time domain $\Omega=D\times(T_I, T_F)$ by $\Gamma_B=\partial D\times[T_I, T_F]$, $\Gamma_I=D\times\{T_I\}$, and $\Gamma_F=D\times\{T_F\}$. Let a function u satisfy (1) and (2) for given boundary data u_B.

Direct problem: Determine the function u which satisfies $u=u_I$ for given initial data u_I on Γ_I.

Inverse problem: Determine the function u which satisfies $u=u_F$ for given final data u_F on Γ_F.

We choose the quadrature points at cubic

Table 1 Maximum norm of the exact solution (n=1)

t	$\max_{x'\in \overline{D}} \lvert u^{(1)}(\boldsymbol{x}', t)\rvert$
−0.5	6.25×10^{-1}
−0.25	3.79×10^{-1}
0	2.30×10^{-1}
0.25	1.39×10^{-1}
0.5	8.46×10^{-2}

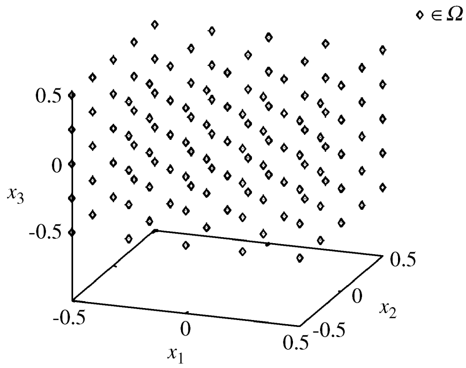

Fig. 2 Quadrature points

lattice points by equally dividing $\overline{\Omega}$ in p−1 parts in each direction of x_k-axis. Then the number of quadrature points is $N=p^3$. They are represented by $\boldsymbol{x}^{([i_1\ i_2\ i_3]_p)} = {}^t(\frac{i_1}{p-1}-0.5;\ \frac{i_2}{p-1}-0.5;\ \frac{i_3}{p-1}-0.5)$ for i_1, i_2, i_3=0, 1, ⋯, p−1, where $[i_1\ i_2\ i_3]_p:=1+i_1+i_2p+i_3p^2$ is the serial number. Here we choose the number $N=5^3=125$ and the parameter ρ=1. Fig. 2 shows the allocation of the quadrature points. A maximum absolute error is defined by

$$\text{abs}-\text{err}(t):=\max_{x_3^{(k)}=t}\left| u_k-u(\boldsymbol{x}^{(k)})\right|.$$

A maximum relative error is defined by

$$\text{rel}-\text{err}(t)$$

$$:=\begin{cases}\max_{x_3^{(k)}=t}\dfrac{\left|u_k-u(\boldsymbol{x}^{(k)})\right|}{\left|u(\boldsymbol{x}^{(k)})\right|}, & \left|u(\boldsymbol{x}^{(k)})\right|\geq\varepsilon\\ \max_{x_3^{(k)}=t}\left|u_k-u(\boldsymbol{x}^{(k)})\right|, & \left|u(\boldsymbol{x}^{(k)})\right|<\varepsilon\end{cases}$$

for $\varepsilon=10^{-13}$ in the double precision arithmetic.

For a parameter $n\in \boldsymbol{N}$ the function

$$u^{(n)}(\boldsymbol{x})=e^{-2n^2t}\sin nx_1 \sin nx_2$$

Table 2 Errors in the direct problem (n=1)

t	abs-err(t)	rel-err(t)
−0.5	5.38×10^{-9}	1.72×10^{-8}
−0.25	2.90×10^{-5}	2.87×10^{-4}
0	1.01×10^{-5}	1.64×10^{-4}
0.25	1.28×10^{-5}	3.44×10^{-4}
0.5	9.61×10^{-6}	4.27×10^{-4}

Table 3 Errors in the inverse problem (n=1)

t	abs-err(t)	rel-err(t)
−0.5	2.71×10^{-6}	1.63×10^{-5}
−0.25	2.90×10^{-5}	2.87×10^{-4}
0	1.01×10^{-5}	1.65×10^{-4}
0.25	1.27×10^{-5}	3.41×10^{-4}
0.5	2.89×10^{-9}	1.29×10^{-7}

Table 4 Maximum norm of the exact solution (n=2)

t	$\max_{x'\in \overline{D}} \lvert u^{(2)}(\boldsymbol{x}', t)\rvert$
−0.5	3.87×10^{1}
−0.25	5.23×10^{0}
0	7.08×10^{-1}
0.25	9.58×10^{-2}
0.5	1.30×10^{-2}

satisfies the heat Eq. (1). The solution of the direct problem is $u^{(n)}$ for the boundary data $u_B=u^{(n)}$ on Γ_B and the initial data $u_I=u^{(n)}$ on Γ_I. The solution of the inverse problem is also $u^{(n)}$ for the boundary data $u_B=u^{(n)}$ on Γ_B and the final data $u_F=u^{(n)}$ on Γ_F.

In Table 1 we show maximum absolute values of $u^{(n)}(\boldsymbol{x}', t)$ with respect to $\boldsymbol{x}'\in \overline{D}$ for n=1. We show the maximum absolute error and the maximum relative error in Table 2 for the direct problem and in Table 3 for the inverse problem.

From Table 2 we can see that the numerical solution is highly accurate for the direct problem with n=1. At the time t=−0.5 the numerical errors are not 0 although the discretization error is 0 since the initial data are given at t=−0.5. We guess that errors are caused by rounding errors. We can see that the precision of the numerical solution for the inverse problem is almost equivalent to the precision for the direct problem except for times t=−0.5, 0.5.

In Table 4 we show the maximum absolute value of $u^{(n)}(\boldsymbol{x}', t)$ with respect to $\boldsymbol{x}'\in \overline{D}$ for n=2. The maximum absolute values decrease faster than the case of n=1 when t increases. Solving the direct problem and the inverse problem for n=2, we show both the maximum absolute error and the maximum relative error

Table 5 Errors in the direct problem (n=2)

t	abs-err(t)	rel-err(t)
−0.5	1.39×10^{-5}	1.24×10^{-5}
−0.25	6.26×10^{-2}	3.69×10^{-2}
0	2.25×10^{-2}	9.81×10^{-2}
0.25	2.39×10^{-2}	7.68×10^{-1}
0.5	6.64×10^{-2}	1.58×10^{1}

Table 6 Errors in the inverse problem (n=2)

t	abs-err(t)	rel-err(t)
−0.5	3.48×10^{-1}	2.77×10^{-2}
−0.25	5.62×10^{-2}	3.31×10^{-2}
0	2.06×10^{-2}	8.95×10^{-2}
0.25	2.16×10^{-2}	6.93×10^{-1}
0.5	1.92×10^{-5}	4.57×10^{-3}

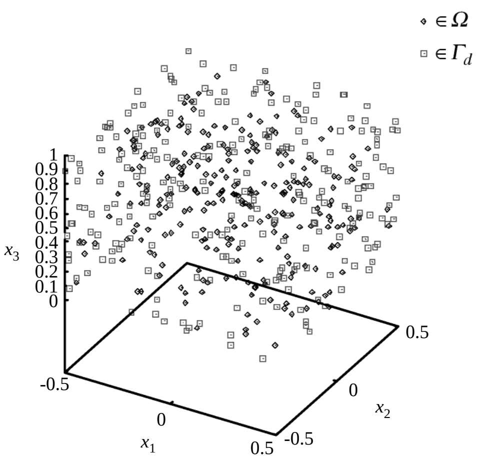

Fig. 3 Quadrature points

in Table 5 for the direct problem and in Table 6 for the inverse problem.

The error abs-err(−0.5)=1.39×10^{-5} in Table 5 and the error abs-err(0.5)=1.92×10^{-5} in Table 6 are accumulations of rounding errors for the direct problem and for the inverse problem, respectively. From Tables 5, 6 we can estimate the absolute errors between 10^{-2} and 10^{-1} except for t=−0.5, 0.5. For n=2 we can see that the precision of the numerical solution for the inverse problem is almost equivalent to the precision for the direct problem except for times t=−0.5, 0.5.

Example 2.

We consider the problem (1)-(3). Let the domain D be (−0.5, 0.5)×(−0.5, 0.5) and let the final time be T=1. We take the solution $u^{(n)}(\boldsymbol{x})$ in Example 1. We give the boundary data and the final data as

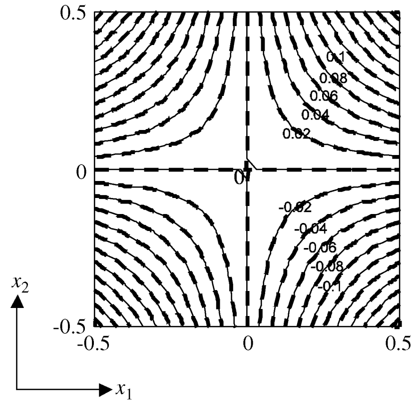

Fig. 4 Numerical(—) and exact(- - -) solutions (n=1, t=0)

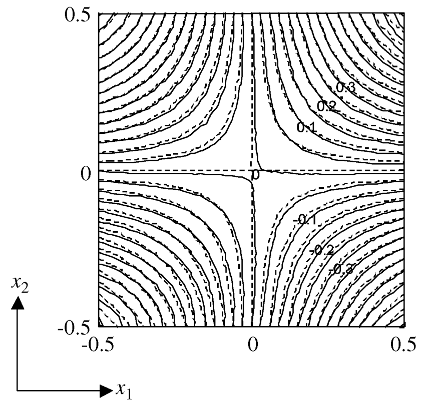

Fig. 5 Numerical(—) and exact(- - -) solutions (n=2, t=0)

$u_B^{(n)}(\boldsymbol{x})=u^{(n)}(\boldsymbol{x})$, $\boldsymbol{x}\in\Gamma_B$ and $u_F^{(n)}(\boldsymbol{x})=u^{(n)}(\boldsymbol{x})$, $\boldsymbol{x}\in\Gamma_F$. 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 Fig. 3 distribution of quadrature points is presented. We show the exact solution and the numerical solution in Fig. 4 for parameters n=1 and ρ=4 at t=0, and in Fig. 5 for parameters n=2 and ρ=4 at t=0. In Fig. 6 and Fig. 7 we show the numerical solution and the exact solution for parameters n=3 and ρ=4 at t=0, respectively. We observe increase in the error of the numerical solution when the parameter n becomes large.

From the estimation (4) the ratio between the solution $u^{(n)}$ and the final data $u_F^{(n)}$ is $C_n=\frac{1}{2n}\cdot\sqrt{e^{4n^2T}-1}=O(\frac{e^{2n^2}}{n})$ with respect to L^2 norm. For n=2, 3 ratios are estimated as $C_2\approx700$ and $C_3\approx10^7$. For n=3 we guess an accumulation of rounding error in the computational arithmetic as a source factor for the large error in numerical solutions.

Example 3.

We consider the problem (1)-(3). Let the domain D be $\{(x_1, x_2): x_1^2+x_2^2<0.5^2\}\cup(0, 1)\times(-0.25, 0.25)$

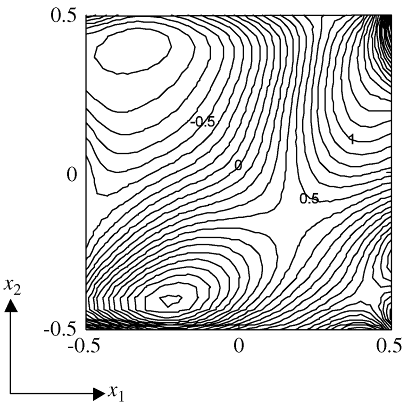

Fig. 6 Numerical solution (n=3, t=0)

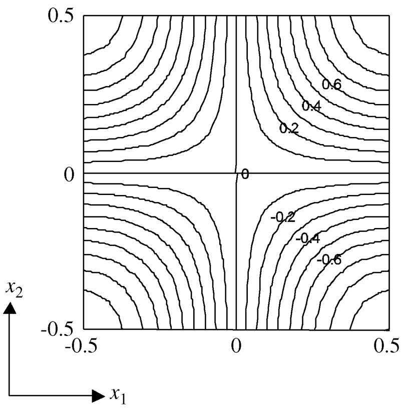

Fig. 7 Exact solution (n=3, t=0)

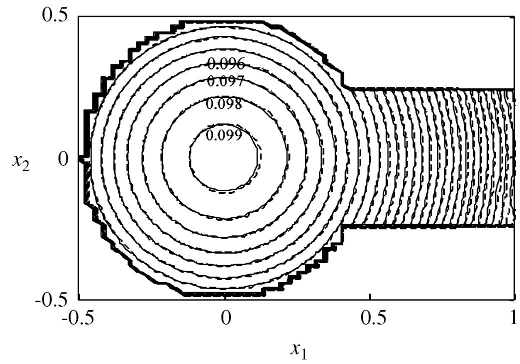

Fig. 8 Numerical(—) and exact(- - -) solution (t=0.5)

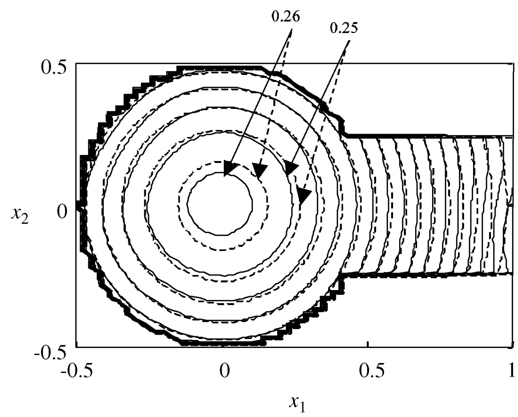

Fig. 9 Numerical(—) and exact(- - -) solution (t=0)

and the final time be T=1. For a parameter $\boldsymbol{y}={}^t(y_1, y_2, \tau)\notin \Omega$ the function

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

with respect to $\boldsymbol{x}$ satisfies the heat Eq. (1). For $\boldsymbol{y}={}^t(0, 0, -0.3)$ we give the boundary data and the final data as $u_B^{(n)}(\boldsymbol{x})=G(\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{x}\in \Gamma_B$ and $u_F^{(n)}(\boldsymbol{x})=G(\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{x}\in \Gamma_F$. Then we calculate numerical solutions of the problem (1)-(3) by using the high order finite difference method for N=500 and ρ=3.

We show the numerical solution and the exact solution in Fig. 8 at t=0.5 and in Fig. 9 at t=0. For the problem in which the domain is not a rectangle we obtained the accurate solution although the problem is unstable.

VIII. MULTI-PRECISION ARITHMETIC

We consider behavior of rounding errors in the high order finite difference method by numerical computations.

We calculate the high order finite difference approximation $\tilde{u}''(x)$ to the derivative $u''(x)$ of a one-dimensional function $u(x)$ on the interval [−3, 3] by using a multi-precision arithmetic (Fujiwara and Iso, 2001). A maximum absolute error is defined by $\max_{j=1,2,\cdots,N} |\tilde{u}''(x^{(j)})-u''(x^{(j)})|$ with the accumulation of rounding errors for the quadrature points $x^{(j)}\in[-3, 3]$, j=1, 2, ⋯, N.

In Figs. 10 and 11 we show error for $u(x)=\sin x$ and $u''(x)=-\sin x$. Here we choose the quadrature points in case 1 as dividing the interval [−3, 3] equally, namely $x^{(j)}=-3+\frac{6(j-1)}{N-1}$, j=1, 2, ⋯, N, and in case 2 as $x^{(j)}=3\cos\frac{j-1}{N-1}\pi$, j=1, 2, ⋯, N. We use 16 and 200 digits in the multi-precision arithmetic, where using 16 digits means double precision arithmetic. The axis of the abscissa indicates the number, N, of quadrature points and the axis of the ordinate indicates the error by the logarithmic scale.

In Fig. 10 the increment of accuracy of $\tilde{u}''$ stops for N>20 for both cases 1 and 2 in the arithmetic by 16 digits. The approximations $\tilde{u}''$ in the arithmetic by 200 digits sharply improve in accuracy for both types of quadrature points as N increases. The above

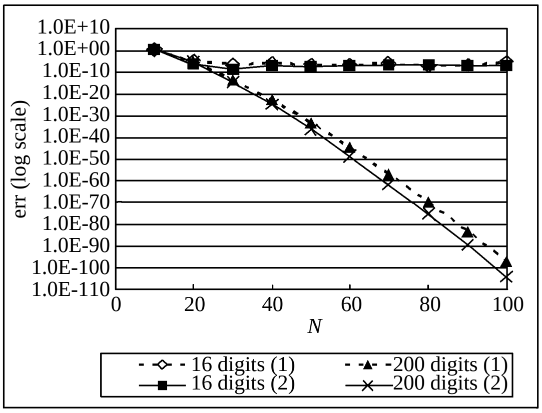

Fig. 10 Errors of $\tilde{u}''$ for $u(x)=\sin x$

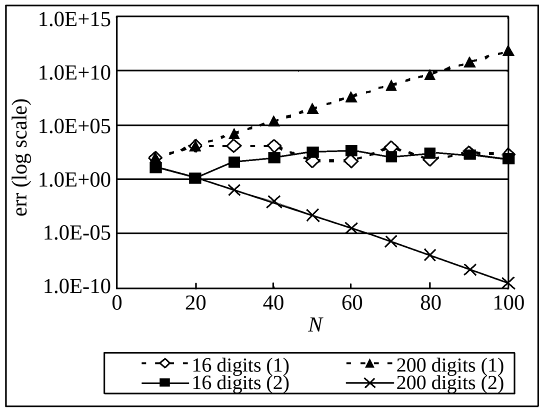

Fig. 11 Errors of $\tilde{u}''$ for $u(x)=\frac{1}{1+x^2}$

results imply that we can obtain the approximate function with high accuracy by using our finite difference method.

In Fig. 11 we show the error of the approximation to $u(x)=\frac{1}{1+x^2}$ for both ways of choosing the quadrature points. In the double precision arithmetic the accuracy of the approximation does not improve for either way of choosing the quadrature points. In the arithmetic by 200 digits, we observe that the approximation in case 1 diverges, although the error of the approximation in case 2 decreases exponen-tially. In Fig. 12 we show the approximation in case 1 for N=50. We can see that the Runge's phenomenon occurs in case 1. From the above results we can not always choose the quadrature points randomly.

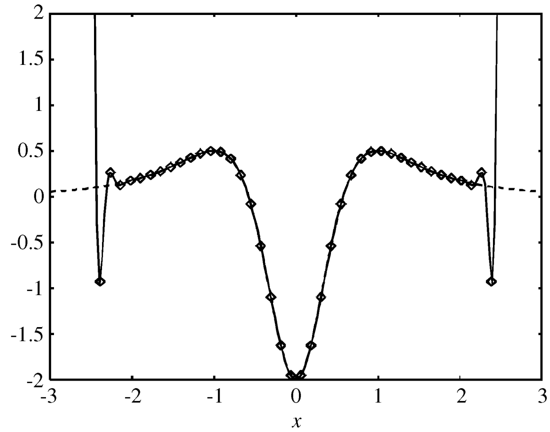

Fig. 12 Derivative $u''(x) = \frac{-2+8x^2}{(1+x^2)^3}$ (- - -) and the approximation $\tilde{u}''(x)$ (—) in case 1 for N=50

IX. 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 function can be approximated by a linear combination of values of the function at quadrature points. Since we can use quadrature points which are chosen at arbitrary locations, the approximation gains a meshless property. Two examples from the Taylor expansion and the Fourier transform are presented. It is shown that the approximation coincides with the derivative of the exponential interpolation formula. In numerical experiments, the two-dimensional backward heat conduction problem was solved as a three-dimensional problem in the space-time domain. By using N=125 cubic lattice points as the quadrature points, the accuracy of the numerical solution for the inverse problem is shown to be almost equivalent to the accuracy for the corresponding direct problem. When magnification of the solution for the final data is very large, we guess that the numerical solution is strongly contaminated by round-off errors. We confirmed that our method is applicable to the problem in a domain with a curved boundary.

ACKNOWLEDGMENTS

The author is grateful to Professor Y. Iso and Dr. H. Fujiwara at Graduate School of Informatics, Kyoto University. In this paper their multiprecision system has been used.

REFERENCES

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

Fujiwara, H., and Iso, Y., 2001, "Numerical Challenge to Ill-Posed Problems by Fast Multipleprecision System," *Theoretical and Applied Mechanics Japan*, Vol. 50, pp. 419-424.

Imai, H., Takeuchi, T., and Kushida, M., 1999, "On Numerical Simulation of Partial Differential Equations in Infinite Precision," *Advances in Mathematical Sciences and Applications*, Gakkōtosho, Tokyo, Vol. 9, No. 2, pp. 1007-1016.

Ito, S., 1963, *Lebesgue Integral Primer*, Shokabo, Japan (in Japanese).

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

Richtmyer, R. D., and Morton, K. W., 1967, *Difference Methods for Initial-value Problems*, Second edition, Interscience Tracts in Pure and Applied Mathematics, No. 4, Interscience Publishers, New York, London, USA.

Manuscript Received: Oct. 02, 2003
Revision Received: May 04, 2004
and Accepted: May 17, 2004