mathematics

MDPI

Article

Numerical Gradient Schemes for Heat Equations Based on the Collocation Polynomial and Hermite Interpolation

Hou-Biao Li [1,*], Ming-Yan Song [1,*], Er-Jie Zhong [1] and Xian-Ming Gu [2]

[1] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China; zhongej@uestc.edu.cn

[2] School of Economic Mathematics, Southwestern University of Finance and Economics, Chengdu 611130, China; guxm@swufe.edu.cn

* Correspondence: lihoubiao0189@163.com (H.-B.L.); mysong619@163.com (M.-Y.S.); Tel.: +86-131-9488-4912 (H.-B.L.)

Received: 26 November 2018; Accepted: 2 January 2019; Published: 17 January 2019

Abstract: As is well-known, the advantage of the high-order compact difference scheme (H-OCD) is that it is unconditionally stable and convergent on the order $O(\tau^2 + h^4)$ (where τ is the time step size and h is the mesh size), under the maximum norm for a class of nonlinear delay partial differential equations with initial and Dirichlet boundary conditions. In this article, a new numerical gradient scheme based on the collocation polynomial and Hermite interpolation is presented. The convergence order of this kind of method is also $O(\tau^2 + h^4)$ under the discrete maximum norm when the spatial step size is twice the one of H-OCD, which accelerates the computational process. In addition, some corresponding analyses are made and the Richardson extrapolation technique is also considered in the time direction. The results of numerical experiments are consistent with the theoretical analysis.

Keywords: heat equation; compact difference schemes; numerical gradient; collocation polynomial; Hermite interpolation; Richardson extrapolation

1. Introduction

Recently, a great deal of effort has been devoted to the development of numerical approximations to heat equation problems (see [1–4]). It is well known that the traditional numerical schemes have low accuracy, and thus need fine discretization in order to obtain the desired accuracy, which leads to many computational challenges due to prohibitive computer memory and time requirements (see [3]).

For heat equations, the forward Euler, backward Euler, and Crank–Nicolson methods were presented many years ago (see Reference [2]). In addition, three layer implicit schemes also appeared in Reference [4]. The forward and backward Euler methods only have first-order accuracy in time and second-order accuracy in space. Also, the forward Euler method is not stable when $c\tau/h^2 > 1/2$. The three layer implicit compact format can reach $O(\tau^2 + h^4)$, but the format is complex. The Crank–Nicolson method has second-order accuracy in time and space, which is not better when compared to the high-order compact difference scheme (see Reference [3]), with second-order accuracy in time and fourth-order accuracy in space. The high-order compact difference format (H-OCD) has many advantages such as its use of less grid backplane points, its high accuracy, its unconditionally stability, and its convergence order $O(\tau^2 + h^4)$ under the maximum norm for a class of nonlinear delay partial differential equations with initial and Dirichlet boundary conditions.

In this paper, based on the H-OCD rough grids, we introduce numerical gradients and utilize local information to improve the calculation accuracy of the rough grids, and the purpose of acceleration is achieved. Our strategy is as follows: first, we obtain the intermediate points of the H-OCD rough mesh grid by cubic and bi-cubic Hermite interpolation. Then, according to these intermediate points, a new explicit scheme on the gradient of the discrete solutions of the heat equation is deduced based on the collocation polynomial. This greatly reduces the amount of calculation for the same accuracy as the high-order compact difference schemes.

The outline of the article is organized as follows. In Section 2, the compact difference scheme is derived for one-dimensional heat equations, the numerical gradient method is presented, and then its convergence is analyzed in detail. In Section 3, we generalize the previous one-dimensional numerical gradient scheme to a two-dimensional one, and some similar results are obtained. In addition, the Richardson extrapolation on the time term is also considered. Finally, some numerical results are reported in Section 4.

2. One-Dimensional Numerical Gradient Schemes Based on the Local Hermite Interpolation and Collocation Polynomial

For the convenience of description, let us firstly consider the one-dimensional case and then generalize to the two-dimensional case.

2.1. The High-Order Compact Difference Scheme in One-Dimension

First, let us consider the following one-dimensional heat equation problem:

$$\begin{cases} \frac{\partial u}{\partial t}(x,t) &= c\frac{\partial^2 u}{\partial x^2}(x,t), & (x,t) \in (0,1) \times (0,T], \\ u(x,0) &= \varphi(x), & x \in [0,1], \\ u(0,t) &= g_1(t),\ u(1,t) = g_2(t), & t \in (0,T], \end{cases} \tag{1}$$

Here T is a positive number. Denote $\Omega = (0,1) \times (0,T]$. In addition, the solution $u(x,t)$ is assumed to be sufficiently smooth and has the required continuous partial derivative.

Next, let us recall the compact difference scheme, which has been introduced in Reference [5].

Let $\Omega_h = \{x_j | x_j = jh, 0 \le j \le N\}$ be a uniform partition of $[0,1]$ with the mesh size $h = 1/N$ and $\Omega_\tau = \{t_k | t_k = k\tau,\ 0 \le k \le M\}$ be a uniform partition of $[0,T]$ with the time step size $\tau = T/M$. We denote $\Omega_{h\tau} = \Omega_h \times \Omega_\tau$. Let $\{u_j^k | 0 \le j \le N,\ 0 \le k \le M\}$ be a mesh function defined on $\Omega_{h\tau}$. For convenience, some other notations are introduced below:

$$\begin{aligned} [u]_j^k &= u(x_j,t_k), \quad u_j^k \approx u(x_j,t_k),\ u_j^{k+\frac{1}{2}} = \frac{u_j^k + u_j^{k+1}}{2}, \\ \delta_t u_j^{k+\frac{1}{2}} &= u_j^{k+1} - u_j^k,\ u_{j-\frac{1}{2}}^k = \frac{u_j^k + u_{j-1}^k}{2}, \\ \delta_x u_{j-\frac{1}{2}}^k &= u_j^k - u_{j-1}^k,\ \text{and}\ \delta_x^2 u_j^k = u_{j-1}^k - 2u_j^k + u_{j+1}^k. \end{aligned}$$

In addition, we sometimes use the index pair (j,k) to represent the mesh point (x_j,t_k). In order to obtain the high-order compact difference scheme on the Equation (1), let us first recall the following lemma.

Lemma 1 ([3,5]). *Suppose $g(x) \in \mathbb{C}^6[x_{i-1},x_{i+1}]$, then*

$$\tfrac{1}{12}[g''(x_{i-1}) + 10g''(x_i) + g''(x_{i+1})] - \tfrac{1}{h^2}[g(x_{i-1}) - 2g(x_i) + g(x_{i+1})] = \tfrac{h^4}{240}g^6(\omega_i), \tag{2}$$

where $\omega_i \in (x_{i-1},x_{i+1})$.

Next, let us consider the Equation (1) at the point $(x_j,t_{k+\frac{1}{2}})$:

$$\frac{\partial u}{\partial t}(x_j,t_{k+\frac{1}{2}}) = c\frac{\partial^2 u}{\partial x^2}(x_j,t_{k+\frac{1}{2}}), \quad 0 \le i \le N, \quad 0 \le k \le M-1. \tag{3}$$

Then for $g = [g_0,\ g_1,\ \ldots,\ g_N]$, we introduce the operator β with the help of Lemma 1, where we denote:

$$\beta g_j = \frac{1}{12}[g_{j-1} + 10g_j + g_{j+1}], \quad 1 \le j \le N-1. \tag{4}$$

By the famous Taylor formula, we have that:

$$\frac{1}{12\tau}\left[\delta_t u_{j-1}^{k+\frac{1}{2}} + 10\delta_t u_j^{k+\frac{1}{2}} + \delta_t u_{j+1}^{k+\frac{1}{2}}\right] = \frac{c}{h^2}\delta_x^2 u(x_j, t_{k+\frac{1}{2}}) + R_j^k, \tag{5}$$

and:

$$R_j^k = \tau^2 \beta r_j^k + \frac{ch^4}{480}\left[\frac{\partial^6 u}{\partial x^6}(\theta_j^k, t_k) + \frac{\partial^6 u}{\partial x^6}(\theta_j^{k+1}, t_{k+1})\right], \tag{6}$$

where $r_j^k = \frac{1}{24}\frac{\partial^3 u}{\partial t^3}(x_j, \xi_j^k) - \frac{c}{8}\frac{\partial^4 u}{\partial x^2 \partial t^2}(x_j, \eta_j^k)$, and $\xi_j^k, \eta_j^k, \theta_j^k, \theta_j^{k+1} \in (x_{j-1}, x_{j+1})$, $1 \le j \le N-1$, and $0 \le k \le M-1$.

Noting the initial and boundary conditions in Equation (1), we obtain the following high-order compact difference scheme:

$$\begin{aligned} &(u_{j-1}^{k+1} - u_{j-1}^k) + 10(u_j^{k+1} - u_j^k) + (u_{j+1}^{k+1} - u_{j+1}^k) \\ &= \tfrac{6c\tau}{h^2}\delta_x^2(u_j^k + u_j^{k+1}), \end{aligned} \tag{7}$$

where $1 \le j \le N-1$, and $0 \le k \le M-1$ and with:

$$u_j^0 = \varphi(x_j), \quad 0 \le j \le N, \tag{8}$$

$$u_0^k = g_1(k\tau),\ \ u_N^k = g_2(k\tau), \quad 0 \le k \le M. \tag{9}$$

Denoting $u_h^k = [u_1^k,\ u_2^k, \ldots,\ u_{N-1}^k]^T$, for $k = 0,\ 1,\ 2, \ldots,\ M-1$, the above Equations (7)–(9) can be written as:

$$\left(T_1 - \frac{6c\tau}{h^2}T_2\right)u_h^{k+1} = \left(T_1 + \frac{6c\tau}{h^2}T_2\right)u_h^k + F_0, \tag{10}$$

where:

$$T_1 = \begin{pmatrix} 10 & 1 & 0 & \ldots & 0 & 0 \\ 1 & 10 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 10 \end{pmatrix} \text{ and } T_2 = \begin{pmatrix} -2 & 1 & 0 & \ldots & 0 & 0 \\ 1 & -2 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & -2 \end{pmatrix}.$$

In addition, if we denote:

$$C = \max\left\{\frac{c}{240}\max_{0\le x\le 1, 0\le t\le T}\left|\frac{\partial^6 u(x,t)}{\partial x^6}\right|, \frac{1}{24}\max_{0\le x\le 1, 0\le t\le T}\left|\frac{\partial^3 u(x,t)}{\partial t^3}\right| + \frac{c}{8}\max_{0\le x\le 1, 0\le t\le T}\left|\frac{\partial^4 u(x,t)}{\partial x^2 \partial t^2}\right|\right\},$$

then, according to Reference [5], we have:

$$|R_j^k| \le C(\tau^2 + h^4), \quad 1 \le j \le N-1,\ \ 0 \le k \le M-1.$$

That is, the truncation error of the compact difference scheme in Equation (7) is $O(\tau^2 + h^4)$.

2.2. One-Dimensional Numerical Gradient Scheme Based on Local Hermite Interpolation and the Collocation Polynomial

As is stated in previous Sections 1 and 2.1, the compact difference method has some advantages. However, the amount of calculation will be rapidly increased with an increase in the number of mesh grid points, see the numerical experiments in Section 4. In order to deal with this problem, next we give a new numerical gradient scheme based on the collocation polynomial and Hermite interpolation.

Let U_h be the vector space of the grid function on $\Omega_{\tau h}$, and let u_h denote the discrete solution satisfying Equations (7)–(9). Denote:

$$P_j = \frac{\partial u(x_j, t)}{\partial x}, \text{ and } P_j^k = \frac{\partial u(x_j, t_k)}{\partial x}. \tag{11}$$

Our strategy is as follows:

1. First, get the values of points u_j^T by the H-OCD scheme, Equations (7)–(9);
2. Then obtain a formula (see Equation (17)) for P_j with the help of the collocation polynomial; i.e.,:

$$P_j = \frac{1}{12h}[8u(x_{j+1}, t) - 8u(x_{j-1}, t) + u(x_{j-2}, t) - u(x_{j+2}, t)];$$

3. Finally, determine the values (see Equation (14)) of the intermediate points $u^t_{j+\frac{1}{2}}$ based on Hermite interpolation; i.e.,:

$$u^t_{j+\frac{1}{2}} = \frac{1}{2}(u^t_j + u^t_{j+1}) + \frac{h}{8}(P_j - P_{j+1}).$$

Thus, combining the H-OCD scheme with the above improvements, a new explicit numerical gradient scheme for the gradient terms of the discrete solutions of heat equations is deduced, which will greatly reduce the amount of calculation at the same accuracy as the high-order compact difference method. Next, let us give a concrete analysis of this approach.

2.2.1. Local Hermite Interpolation and Refinement in the One-Dimensional Case

For convenience, we just consider Hermite cubic and bi-cubic interpolation functions $u_H(x, t)$ on the interval $[x_j, x_{j+1}] \subset \Omega_h$; its vertexes are as follows:

$$z_1(x_j, t),\ z_2(x_{j+1}, t) \in \Omega_{\tau h}.$$

On the segment $z_1 - z_2$, let the cubic interpolation function satisfy the conditions:

$$u_H(z_1) = u(z_1),\ u_H(z_2) = u(z_2),$$

$$(u_H)_x(z_1) = u_x(z_1), \text{ and } (u_H)_x(z_2) = u(z_2).$$

Based on Reference [6], we can get the Hermite interpolation polynomial as follows:

$$u_H(x_{j+\frac{1}{2}}, t) = \frac{1}{2}[u(z_1) + u(z_2)] + \frac{h}{8}[u_x(z_1) - u_x(z_2)], \tag{12}$$

where $j = 1, 2, \ldots, N-1$. The interpolation errors are

$$\begin{aligned} u_H(x_{j+\frac{1}{2}}, t) - u(x_{j+\frac{1}{2}}, t) &= \tfrac{1}{4!}u_{xxxx}(\xi_j)(x_{j+\frac{1}{2}} - x_j)^2(x_{j+\frac{1}{2}} - x_{j+1})^2 \\ &= \tfrac{h^4}{24\times 16}u_{xxxx}(\xi_j), \quad j = 1, 2, \ldots, N-1, \end{aligned} \tag{13}$$

where ξ_j lies between z_1 and z_2 (see Reference [6]). So, by Equation (11), we have the refinemed computation format:

$$u^t_{j+\frac{1}{2}} = \frac{1}{2}[u(x_j, t) + u(x_{j+1}, t)] + \frac{h}{8}(P_j - P_{j+1}), \quad j = 1,\ 2, \ldots,\ N-2. \tag{14}$$

From Equation (13), we know that the above refinement schemes have fourth-order accuracy in the space direction.

2.2.2. The Collocation Polynomial in the One-Dimensional Case

From Equation (14), we know that we must obtain the expression of P_j in order to get the specific formula of the intermediate points. Here, we choose the collocation polynomial method. For convenience, we first consider the sub-domain:

$$[x_{j-1}, x_{j+1}] \subset \Omega,\ j = 1, 2, \dots, N-1.$$

Then, we denote:

$$\xi = x - x_j,\ x \in \Omega,\ j = 1, 2, \dots, N-1.$$

In order to get the approximation polynomial of u, we consider the polynomial space:

$$H_4 = span\{1,\ \xi,\ \xi^2,\ \xi^3,\ \xi^4\}, \tag{15}$$

and the approximation polynomial of u:

$$H(\xi) = a_0 + a_1\xi + a_2\xi^2 + a_3\xi^3 + a_4\xi^4. \tag{16}$$

Let:

$$\begin{array}{rcl} H(x_{j-1}) & = & u^t_{j-1}, \qquad H(x_{j+1}) = u^t_{j+1}, \\ c\frac{\partial^2 H(x_{j-1})}{\partial x^2} & = & \frac{\partial u}{\partial t}(x_{j-1}, t), \qquad c\frac{\partial^2 H(x_j)}{\partial x^2} = \frac{\partial u}{\partial t}(x_j, t),\ \text{and} \\ c\frac{\partial^2 H(x_{j+1})}{\partial x^2} & = & \frac{\partial u}{\partial t}(x_{j+1}, t), \qquad j = 2,\ 3,\ ,\dots\ N-2. \end{array}$$

Thus, by Equations (1) and (11), the approximation of P_j can be described as follows:

$$\begin{array}{rcl} P_j & = & \frac{1}{2h}[u(x_{j+1}, t) - u(x_{j-1}, t)] + \frac{h}{12c}[\frac{\partial u}{\partial t}(x_{j-1}, t) - \frac{\partial u}{\partial t}(x_{j+1}, t)] \\ & = & \frac{1}{12h}[8u(x_{j+1}, t) - 8u(x_{j-1}, t) + u(x_{j-2}, t) - u(x_{j+2}, t)], \\ P_1 & = & \frac{1}{6h}[-2g_1(t) - 3u(x_1, t) + 6u(x_2, t) - u(x_3, t)], \\ P_{N-1} & = & \frac{1}{6h}[2g_2(t) + 3u(x_{N-1}, t) - 6u(x_{N-2}, t) + u(x_{N-3}, t)], \end{array} \tag{17}$$

where $j = 2,\ 3, \dots\ N-3$.

Thus:

$$\begin{array}{rcl} u^t_{j+\frac{1}{2}} & = & \frac{1}{2}[u(x_j, t) + u(x_{j+1}, t)] + \frac{h}{8}(P_j - P_{j+1}) \\ & = & \frac{1}{2}[u(x_j, t) + u(x_{j+1}, t)] + \frac{1}{96}[8u(x_{j+1}, t) - 8u(x_{j-1}, t) + u(x_{j-2}, t) - \\ & & u(x_{j+2}, t)] - \frac{1}{96}[8u(x_{j+2}, t) - 8u(x_j, t) + u(x_{j-1}, t) - u(x_{j+3}, t)] \\ & = & \frac{1}{96}[56u(x_j, t) + 56u(x_{j+1}, t) + u(x_{j-2}, t) - 9u(x_{j+2}, t) + u(x_{j+3}, t)], \\ u^t_{\frac{3}{2}} & = & \frac{1}{2}[u(x_1, t) + u(x_2, t)] + \frac{h}{8}(P_1 - P_2) \\ & = & \frac{1}{96}[50u(x_1, t) + 60u(x_2, t) - 10u(x_3, t) - u(x_0, t) + u(x_4, t) - 4g_1(t)], \\ u^t_{N-\frac{3}{2}} & = & \frac{1}{2}[u(x_{N-2}, t) + u(x_{N-1}, t)] + \frac{h}{8}(P_{N-2} - P_{N-1}) \\ & = & \frac{1}{96}[60u(x_{N-2}, t) + 50u(x_{N-1}, t) - 10u(x_{N-3}, t) + u(x_{N-4}, t) - \\ & & u(x_N, t) - 4g_2(t)], \end{array} \tag{18}$$

where $j = 2,\ 3,\ ,\dots\ N-3$.

Next, according to our improvement scheme, let us analyze the convergence order of this kind of numerical gradient scheme.

Theorem 1. *If $u(x,t) \in \mathbb{C}^6_x(\Omega)$ and $u(x,t) \in \mathbb{C}^3_t(\Omega)$, then we have:*

$$\left| \frac{\partial u(x_j, t_k)}{\partial x} - P^k_j \right| \le O(h^4), \tag{19}$$

where $j = 2,\ 3,\ 4,\ \dots\ N-2$.

Proof. When $T = k\tau$, we know that:

$$\begin{array}{rcl} P_j^k & = & \frac{1}{12h}[8u(x_{j+1},t_k) - 8u(x_{j-1},t_k) + u(x_{j-2},t_k) - u(x_{j+2},t_k)] \\ & = & \frac{4}{3}\frac{u(x_{j+1},t_k)-u(x_{j-1},t_k)}{2h} - \frac{1}{3}\frac{u(x_{j+2},t_k)-u(x_{j-2},t_k)}{4h}. \end{array}$$

So, by the H-OCD method and the energy method with the Sobolev embedding theorem in Reference [7]:

$$\left|\frac{\partial u(x_j,t_k)}{\partial x} - \left\{\frac{4}{3}\frac{u(x_{j+1},t_k)-u(x_{j-1},t_k)}{2h} - \frac{1}{3}\frac{u(x_{j+2},t_k)-u(x_{j-2},t_k)}{4h}\right\}\right| = O(h^4),$$

where $j = 2,\ 3,\ 4,\ \dots\ N-2$. Thus the proof is completed. □

By the above theorem, we know that the accuracy of the partial derivative of u (i.e., P_j) in the space direction is $O(h^4)$ when $T = k\tau$. In fact, due to Equation (14), it is easy to prove that the accuracy of the intermediate points is $O(h^4)$ as well. The corresponding analysis is as follows.

Theorem 2. *If* $u(x,t) \in \mathbb{C}_x^6(\Omega)$ *and* $u(x,t) \in \mathbb{C}_t^3(\Omega)$, *and* $u(x,t)$ *is the exact solution of Equation (1), then:*

$$\left|u(x_{j+\frac{1}{2}},t_k) - u_{j+\frac{1}{2}}^k\right| \leq O(\tau^2 + h^4), \tag{20}$$

where $j = 2,\ 3,\ \dots\ N-3$.

Proof. First, note that:

$$\begin{array}{rcl} \left|u(x_{j+\frac{1}{2}},t_k) - u_{j+\frac{1}{2}}^k\right| & = & \left|u(x_{j+\frac{1}{2}},t_k) - u_H(x_{j+\frac{1}{2}},t_k) + u_H(x_{j+\frac{1}{2}},t_k) - u_{j+\frac{1}{2}}^k\right| \\ & \leq & \left|u(x_{j+\frac{1}{2}},t_k) - u_H(x_{j+\frac{1}{2}},t_k)\right| + \left|u_H(x_{j+\frac{1}{2}},t_k) - u_{j+\frac{1}{2}}^k\right|, \end{array}$$

then, by the Taylor expansion at $(x_{j+\frac{1}{2}},T)$ (where $T = k\tau$), we have:

$$u(x_j,t_k) = u(x_{j+\frac{1}{2}},t_k) - \frac{h}{2}u_x(x_{j+\frac{1}{2}},t_k) + \frac{h^2}{8}u_{xx}(x_{j+\frac{1}{2}},t_k) - \frac{h^3}{48}u_{xxx}(x_{j+\frac{1}{2}},t_k) + O(h^4),$$

$$u(x_{j+1},t_k) = u(x_{j+\frac{1}{2}},t_k) + \frac{h}{2}u_x(x_{j+\frac{1}{2}},t_k) + \frac{h^2}{8}u_{xx}(x_{j+\frac{1}{2}},t_k) + \frac{h^3}{48}u_{xxx}(x_{j+\frac{1}{2}},t_k) + O(h^4).$$

Thus, by Equation (14), we can obtain:

$$\begin{array}{rcl} u(x_{j+\frac{1}{2}},t_k) & = & \frac{1}{2}[u(x_j,t_k) + u(x_{j+1},t_k)] - \frac{h^2}{8}u_{xx}(x_{j+\frac{1}{2}},t_k) + O(h^4) \\ & = & \frac{1}{2}[u(x_j,t_k) + u(x_{j+1},t_k)] - \frac{h^2}{8}\frac{\partial}{\partial x}\frac{\partial u}{\partial x}(x_{j+\frac{1}{2}},t_k) + O(h^4) \\ & = & \frac{1}{2}[u(x_j,t_k) + u(x_{j+1},t_k)] - \frac{h^2}{8}\frac{\partial}{\partial x}[\frac{1}{h}\delta_x u(x_{j+\frac{1}{2}},t_k) + O(h^2)] + O(h^4) \\ & = & \frac{1}{2}[u(x_j,t_k) + u(x_{j+1},t_k)] + \frac{h^2}{8}\frac{\partial}{\partial x}\frac{u(x_j,t_k)-u(x_{j+1},t_k)}{h} + O(h^4) \\ & = & \frac{1}{2}[u(x_j,t_k) + u(x_{j+1},t_k)] + \frac{h}{8}[u_x(x_j,t_k) - u_x(x_{j+1},t_k)] + O(h^4) \\ & = & u_H(x_{j+\frac{1}{2}},t_k) + O(h^4), \end{array}$$

where $j = 2,\ 3,\ \dots\ N-3$.

Note that there is no change in the time direction corresponding to the H-OCD method. Therefore, by Equations (12) and (14), and Theorem 1, we have:

$$\begin{aligned} u_H(x_{j+\frac{1}{2}},T)-u^T_{j+\frac{1}{2}} &= \tfrac{1}{2}[u(z_1)+u(z_2)]+\tfrac{h}{8}[u_x(z_1)-u_x(z_2)]- \\ & \quad \tfrac{1}{2}[u(x_j,T)+u(x_{j+1},T)]-\tfrac{h}{8}(P_j-P_{j+1}) \\ &= \tfrac{h}{8}[u_x(z_1)-u_x(z_2)]-\tfrac{h}{8}(P_j-P_{j+1}) \\ &= \tfrac{h}{8}[u_x(z_1)-P_j-u_x(z_2)+P_{j+1}] \\ &\le O(\tau^2+h^4). \end{aligned}$$

So:

$$\begin{aligned} \left|u(x_{j+\frac{1}{2}},t_k)-u^k_{j+\frac{1}{2}}\right| &\le \left|u(x_{j+\frac{1}{2}},t_k)-u_H(x_{j+\frac{1}{2}},t_k)\right|+\left|u_H(x_{j+\frac{1}{2}},t_k)-u^k_{j+\frac{1}{2}}\right| \\ &\le O(h^4)+O(\tau^2+h^4) \\ &\le O(\tau^2+h^4). \end{aligned}$$

Thus the proof is completed. □

2.3. Richardson Extrapolation on the H-OCD Scheme in the One-Dimensional Case

For the compact difference H-OCD scheme considered in Section 2.1, the numerical solution and its difference quotient in space direction are unconditionally convergent with a convergence order $O(\tau^2+h^4)$ under the maximum norm. Furthermore, the convergence of the difference quotient in space direction can be proved by the energy method with the Sobolev embedding theorem, that is:

$$\left|\frac{\partial u(x_j,t_k)}{\partial x}-\left\{\frac{4}{3}\frac{u(x_{j+1},t_k)-u(x_{j-1},t_k)}{2h}-\frac{1}{3}\frac{u(x_{j+2},t_k)-u(x_{j-2},t_k)}{4h}\right\}\right|=O(\tau^2+h^4),$$

and:

$$\left|u(x_j,t_k)-u^k_j\right|=O(\tau^2+h^4),\quad 1\le j\le N-1,\ 1\le k\le M.$$

Next, we consider Richardson extrapolation [8] on this H-OCD scheme, Equations (7)–(9), in the time direction in order to reduce the total computation time, as in Reference [9].

Lemma 2 ([10])**.** *Let $\{V^k_j|0\le j\le N, 0\le k\le M\}$ be the solution of the equation below:*

$$\begin{array}{ll} \frac{1}{\tau}\delta_t V^{k+\frac{1}{2}}_j-\frac{a}{h^2}\delta^2_x V^{k+\frac{1}{2}}_j=g^{k+\frac{1}{2}}_j, & 0\le j\le N-1, 0\le k\le M-1, \\ V^0_j=\varphi_j, & 0\le j\le N-1, \\ V^k_0=0, V^k_N=0, & 0\le k\le M. \end{array} \tag{21}$$

Then:

$$|V^k|_\infty\le\frac{1}{2}\left(|V^0|^2_1+\frac{\tau}{2a}\sum_{l=0}^{k-1}|g^{l+\frac{1}{2}}|^2\right)^{\frac{1}{2}},\qquad 0\le k\le N,$$

where:

$$|g^{l+\frac{1}{2}}|^2=h\sum_{j=1}^{N-1}(g^{l+\frac{1}{2}}_j)^2.$$

Theorem 3. *Let $u^k_j(h,\tau)$ be the solution of H-OCD scheme in Equations (7)–(9) with the time step τ and the spatial step h. Then:*

$$\left|u(x_j,t_k)-\left[\frac{4}{3}u^{2k}_j(h,\frac{\tau}{2})-\frac{1}{3}u^k_j(h,\tau)\right]\right|=O(\tau^4+h^4),\ 1\le j\le N-1,\ 1\le k\le M.$$

Proof. Let us consider the following initial–boundary problem:

$$\begin{array}{rcl} u_t - \triangle u &=& F_u(x,t), \quad (x,t) \in (0,1) \times (0,T], \\ u(0,t) &=& u(1,t) = 0, \quad 0 \le t \le T, \\ u(x,0) &=& 0, \quad x \in (0,1) \end{array}$$

with the smooth solution $p(x,t)$, where:

$$F_p(x,t) = \frac{1}{24}\frac{\partial^3 u(x,t)}{\partial t^3} - \frac{c}{8}\frac{\partial^4 u(x,t)}{\partial x^2 \partial t^2}.$$

By Equation (6), we know:

$$R_j^k = F_p(x_j, t_{k+\frac{1}{2}})\tau^2 + O(\tau^4 + h^4), \quad 1 \le j \le N-1, \ 0 \le k \le M-1.$$

So:

$$\begin{array}{l} \frac{\delta_t e_j^{k+\frac{1}{2}}}{\tau} - \frac{c}{h^2}\delta_x^2 e_j^{k+\frac{1}{2}} = F_p(x_j, t_{k+\frac{1}{2}})\tau^2 + O(\tau^4 + h^4), \ 1 \le j \le N-1, \ 0 \le k \le M-1, \\ e_j^0 = 0, \quad 0 \le j \le N, \\ e_0^k = 0, \quad e_m^k = 0, \quad 1 \le k \le M. \end{array}$$

Here $e_j^k = u(x_j, t_k) - u_j^k$, $0 \le j \le N$, and $0 \le k \le M$.
In addition, according to the H-OCD in Equations (7)–(9), we obtain:

$$\begin{array}{l} \frac{\delta_t p_j^{k+\frac{1}{2}}}{\tau} - \frac{c}{h^2}\delta_x^2 p_j^{k+\frac{1}{2}} = F_p(x_j, t_{k+\frac{1}{2}}), \ 1 \le j \le N-1, \ 0 \le k \le M-1, \\ p_j^0 = 0, \quad 0 \le j \le N, \\ p_0^k = 0, \quad p_m^k = 0, \quad 1 \le k \le M. \end{array}$$

Then:

$$u(x_j, t_k) - p_j^k(h,\tau) = O(\tau^2 + h^4), \quad 1 \le j \le N-1, \ 0 \le k \le M.$$

By denoting:

$$r_j^k = e_j^k + \tau^2 p_j^k, \quad 1 \le j \le N, \ 0 \le k \le M,$$

and combining the above equations, we get:

$$\begin{array}{l} \frac{\delta_t r_j^{k+\frac{1}{2}}}{\tau} - \frac{c}{h^2}\delta_x^2 r_j^{k+\frac{1}{2}} = O(\tau^4 + h^4), \quad 1 \le j \le N-1, \ 0 \le k \le M-1, \\ r_j^0 = 0, \quad 0 \le j \le N, \\ r_0^k = 0, \quad r_N^k = 0, \quad 1 \le k \le M. \end{array}$$

Then by Lemma 2 we have:

$$|r^k|_\infty \le \frac{1}{2}\left(|r^0|_1^2 + \frac{\tau}{2a}\sum_{l=0}^{k-1}|O^{l+\frac{1}{2}}(\tau^4 + h^4)|^2\right)^{\frac{1}{2}}, \qquad 0 \le k \le N,$$

where:

$$|O^{l+\frac{1}{2}}(\tau^4 + h^4)|^2 = h\sum_{j=1}^{N-1}(g_j^{l+\frac{1}{2}}(\tau^4 + h^4))^2.$$

That is:

$$|r^k|_\infty = O(\tau^4 + h^4), \quad 1 \le k \le M,$$

i.e.,:

$$u_j^k(h,\tau) = u(x_j,t_k) + \tau^2 p(x_j,t_k) + O(\tau^4 + h^4), \quad 1 \le j \le N-1,\ 0 \le k \le M,$$

$$u_j^{2k}(h,\frac{\tau}{2}) = u(x_j,t_k) + (\frac{\tau}{2})^2 p(x_j,t_k) + O((\frac{\tau}{2})^4 + h^4), \quad 1 \le j \le N-1,\ 0 \le k \le M.$$

Finally:

$$\left| u(x_j,t_k) - \left[\frac{4}{3} u_j^{2k}(h,\frac{\tau}{2}) - \frac{1}{3} u_j^k(h,\tau) \right] \right| = O(\tau^4 + h^4), \quad 1 \le j \le N-1,\ 1 \le k \le M.$$

Thus, the conclusion is proved. □

Remark 1. *With the Richardson extrapolation method above, the truncation errors in the time direction for the H-OCD scheme is* $O(\tau^4 + h^4)$ *in terms of the maximum norm. Similarly, the extrapolation* $\frac{16}{15} u_{2j}^{4k}(\frac{\tau}{4},\frac{h}{2}) - \frac{1}{15} u_j^k(\tau,h)$ *can obtain the following result for any* $1 \le j \le N-1, 1 \le k \le M$*:*

$$\left| u(x_j,t_k) - \left[\frac{16}{15} u_{2j}^{4k}(\frac{\tau}{4},\frac{h}{2}) - \frac{1}{15} u_j^k(\tau,h) \right] \right| = O(\tau^4 + h^6).$$

3. Two-Dimensional Numerical Gradient Scheme Based on Local Hermite Interpolation and the Collocation Polynomial

3.1. The High-Order Compact Difference Scheme in Two-Dimensions

Next, let us generalize the previous one-dimensional H-OCD scheme to the two-dimensional one. Similar to the previous Section 2, the following two-dimensional heat equation problem is considered:

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, & (x,y,t) \in \Omega \times (0,T], \\ u(x,y,0) = \varphi(x,y), & (x,y) \in [a,b] \times [c,d], \\ u(a,y,t) = g_1(y,t),\ u(b,y,t) = g_2(y,t), & (y,t) \in [c,d] \times (0,T], \\ u(x,c,t) = g_3(x,t),\ u(x,d,t) = g_4(x,t), & (x,t) \in [a,b] \times (0,T], \end{cases} \tag{22}$$

where T is a positive number. Denote $\Omega = (a,b) \times (c,d)$. In addition, the solution $u(x,y,t)$ is assumed to be sufficiently smooth and has the required continuous partial derivative.

Let $h_x = \frac{b-a}{N_x}, h_y = \frac{d-c}{N_y}$, and $\Omega_h = \{(x_i,y_j)\,|x_i = ih_x, y_j = jh_y, 0 \le i,j \le N\}$. When $\tau = T/M$, define $\Omega_\tau = \{t_k | t_k = k\tau, 0 \le k \le M\}$ and $\Omega_{h\tau} = \Omega_h \times \Omega_\tau$. In addition, we denote $\{u_{ij}^k | 0 \le i,j \le N, 0 \le \mathrm{k} \le \mathrm{M}\}$ as the mesh function defined on $\Omega_{h\tau}$. Moreover, some other notations are introduced below:

$$[u]_{ij}^k = u\left(x_i,y_j,t_k\right), u_{ij}^k \approx u\left(x_i,y_j,t_k\right);$$
$$\delta_x^2 u_{ij}^k = u_{i-1,j}^k - 2u_{ij}^k + u_{i+1,j}^k, \text{ and } \delta_y^2 u_{ij}^k = u_{i,j-1}^k - 2u_{ij}^k + u_{i,j+1}^k.$$

For convenience, define the operators $D_x = \frac{d}{dx}, D_y = \frac{d}{dy}$, $\mathrm{E} = \sum\limits_{k=0}^{\infty} \frac{1}{k!}(hD)^k = e^{hD}$ and its inverse operator $E^{-1} = e^{-hD}$. Obviously:

$$\delta^2 = E^{-1} - 2 + E = e^{-hD} - 2 + e^{hD} = h^2D^2 + 1/12h^4D^4 + O(h^6).$$

Note that $\delta^2 = h^2D^2 + O\left(h^4\right)$, therefore:

$$\delta^2 = h^2D^2 + \frac{1}{12}h^2D^2\left[\delta^2 - O\left(h^4\right)\right] + O\left(h^6\right) = \left(1 + \frac{1}{12}\delta^2\right)h^2D^2 + O\left(h^6\right). \tag{23}$$

That is:

$$(1+\frac{1}{12}\delta^2)^{-1}\delta^2 = h^2D^2 + O\left(h^6\right).$$

Applying Equation (23) to Equation (22), we obtain:

$$\left(1+\frac{1}{12}\delta_x^2+\frac{1}{12}\delta_y^2\right)\frac{\partial u}{\partial t} = \frac{\left(1+\frac{1}{12}\delta_y^2\right)}{h_x^2}\delta_x^2 u + \frac{\left(1+\frac{1}{12}\delta_x^2\right)}{h_y^2}\delta_y^2 u + O\left(h^4\right), \tag{24}$$

where $O\left(h^4\right) = O\left(h_x^4 + h_y^4\right)$. In addition, according to the Crank–Nicolson difference scheme [1,7], we further have:

$$\left(1+\frac{1}{12}\delta_x^2+\frac{1}{12}\delta_y^2\right)\frac{u^{n+1}-u^n}{\tau} = \left[\frac{\left(1+\frac{1}{12}\delta_y^2\right)}{h_x^2}\delta_x^2 + \frac{\left(1+\frac{1}{12}\delta_x^2\right)}{h_y^2}\delta_y^2\right]\frac{u^{n+1}+u^n}{2},$$

which can be written as:

$$\begin{aligned}&\left(1+\tfrac{1}{12}\delta_x^2+\tfrac{1}{12}\delta_y^2-\tau\tfrac{\left(1+\frac{1}{12}\delta_y^2\right)}{2h_x^2}\delta_x^2-\tau\tfrac{\left(1+\frac{1}{12}\delta_x^2\right)}{2h_y^2}\delta_y^2\right)u^{n+1}\\ &=\left(1+\tfrac{1}{12}\delta_x^2+\tfrac{1}{12}\delta_y^2+\tau\tfrac{\left(1+\frac{1}{12}\delta_y^2\right)}{2h_x^2}\delta_x^2+\tau\tfrac{\left(1+\frac{1}{12}\delta_x^2\right)}{2h_y^2}\delta_y^2\right)u^n.\end{aligned}$$

Next, in order for convenience of description, let $h = h_x = h_y, N = N_x = N_y$, and define $r = \tau/2h^2$, then the above equation can be reduced to the following discrete form by the initial and boundary conditions:

$$\begin{aligned}&\left(\tfrac{2}{3}+\tfrac{10}{3}r\right)u_{ij}^{n+1}-\left(\tfrac{2r}{3}-\tfrac{1}{12}\right)\left(u_{i-1,j}^{n+1}+u_{i+1,j}^{n+1}+u_{i,j-1}^{n+1}+u_{i,j+1}^{n+1}\right)-\tfrac{r}{6}(u_{i-1,j-1}^{n+1}+u_{i-1,j+1}^{n+1}\\ &+u_{i+1,j-1}^{n+1}+u_{i+1,j+1}^{n+1})=\left(\tfrac{2r}{3}+\tfrac{1}{12}\right)\left(u_{i-1,j}^{n}+u_{i+1,j}^{n}+u_{i,j-1}^{n}+u_{i,j+1}^{n}\right)+\\ &\left(\tfrac{2}{3}-\tfrac{10}{3}r\right)u_{ij}^{n}+\tfrac{r}{6}\left(u_{i-1,j-1}^{n}+u_{i-1,j+1}^{n}+u_{i+1,j-1}^{n}+u_{i+1,j+1}^{n}\right),\end{aligned} \tag{25}$$

where $i,j = 1,2,\ldots,N-1$. Furthermore, $u_{ij}^0 = u_0\left(x_i,y_j\right), u_{0j}^k = g_1\left(y_j,t_k\right), u_{Nj}^k = g_2\left(y_j,t_k\right), u_{i0}^k = g_3\left(x_i,t_k\right)$, and $u_{iN}^k = g_4\left(x_i,t_k\right)$, with $i,j = 1,2,\ldots,N-1, k = 1,2,\ldots,M$. The concrete computation process of the above discrete scheme may be described as follows:

$$\begin{aligned}&\begin{pmatrix} A_1 & -B_1 & \cdots & 0 & 0\\ -B_1 & A_1 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & -B_1 & A_1\end{pmatrix}\begin{pmatrix} u_{h1}^{n+1}\\ u_{h2}^{n+1}\\ \vdots\\ u_{hN-1}^{n+1}\end{pmatrix} =\\ &\begin{pmatrix} A_2 & B_2 & \cdots & 0 & 0\\ B_2 & A_2 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & B_2 & A_2\end{pmatrix}\begin{pmatrix} u_{h1}^{n}\\ u_{h2}^{n}\\ \vdots\\ u_{hN-1}^{n}\end{pmatrix} +\\ &\begin{pmatrix} U_{h1}^{n+1}\\ U_{h2}^{n+1}\\ \vdots\\ U_{hN-1}^{n+1}\end{pmatrix}+\begin{pmatrix} B_1u_{h0}^{n+1}\\ 0\\ \vdots\\ B_1u_{hN}^{n+1}\end{pmatrix}+\begin{pmatrix} U_{h1}^{n}\\ U_{h2}^{n}\\ \vdots\\ U_{hN-1}^{n}\end{pmatrix}+\begin{pmatrix} B_2u_{h0}^{n}\\ 0\\ \vdots\\ B_2u_{hN}^{n}\end{pmatrix},\end{aligned} \tag{26}$$

where:

$$\begin{aligned}
U_{hj}^{n+1} &= \left[\tfrac{8r-1}{12}u_{0j}^{n+1} + \tfrac{r}{6}u_{0j+1}^{n+1} + \tfrac{r}{6}u_{0j-1}^{n+1}, 0\ldots 0, \tfrac{8r-1}{12}u_{Nj}^{n+1} + \tfrac{r}{6}u_{N,j+1}^{n+1} + \tfrac{r}{6}u_{N,j-1}^{n+1}\right]^T,\\
U_{hj}^{n} &= \left[\tfrac{8r+1}{12}u_{0j}^{n} + \tfrac{r}{6}u_{0j+1}^{n} + \tfrac{r}{6}u_{0j-1}^{n}, 0\ldots 0, \tfrac{8r+1}{12}u_{Nj}^{n} + \tfrac{r}{6}u_{N,j+1}^{n} + \tfrac{r}{6}u_{N,j-1}^{n}\right]^T, j = 1, 2, \ldots, N-1.
\end{aligned}$$

$$\begin{aligned}
u_{h0}^{n+1} &= \left[u_{10}^{n+1}, u_{20}^{n+1}, \ldots, u_{N-1,0}^{n+1}\right]^T,\ u_{hN}^{n+1} = \left[u_{1N}^{n+1}, u_{2N}^{n+1}, \ldots, u_{N-1,N}^{n+1}\right]^T,\\
u_{h0}^{n} &= \left[u_{10}^{n}, u_{20}^{n}, \ldots, u_{N-1,0}^{n}\right]^T,\ u_{hN}^{n} = \left[u_{1N}^{n}, u_{2N}^{n}, \ldots, u_{N-1,N}^{n}\right]^T.
\end{aligned}$$

and:

$$A_1 = \begin{pmatrix} \frac{2+10r}{3} & \frac{1-8r}{12} & & & \\ \frac{1-8r}{12} & \frac{2+10r}{3} & \frac{1-8r}{12} & & \\ & \frac{1-8r}{12} & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{1-8r}{12} \\ & & & \frac{1-8r}{12} & \frac{2+10r}{3} \end{pmatrix}, A_2 = \begin{pmatrix} \frac{2-10r}{3} & \frac{1+8r}{12} & & & \\ \frac{1+8r}{12} & \frac{2-10r}{3} & \frac{1+8r}{12} & & \\ & \frac{1+8r}{12} & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{1+8r}{12} \\ & & & \frac{1+8r}{12} & \frac{2-10r}{3} \end{pmatrix};$$

$$B_1 = \begin{pmatrix} \frac{8r-1}{12} & \frac{r}{6} & & & \\ \frac{r}{6} & \frac{8r-1}{12} & \frac{r}{6} & & \\ & \frac{r}{6} & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{r}{6} \\ & & & \frac{r}{6} & \frac{8r-1}{12} \end{pmatrix}, \quad B_2 = \begin{pmatrix} \frac{8r+1}{12} & \frac{r}{6} & & & \\ \frac{r}{6} & \frac{8r+1}{12} & \frac{r}{6} & & \\ & \frac{r}{6} & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{r}{6} \\ & & & \frac{r}{6} & \frac{8r+1}{12} \end{pmatrix}.$$

This is the compact difference scheme (H-OCD) for Equation (22). The truncation errors $O(\tau^2 + h^4)$ can be directly obtained by the derivation process.

3.2. Two-Dimensional Numerical Gradient Scheme

Next, analogous to Section 2, let us consider the two-dimensional numerical gradient scheme on the above discrete form, Equation (25), by local Hermite interpolation and the collocation polynomial. The intermediate points $u(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ can be expressed (see Equation (28)) by the values of the mesh points and the partial derivatives $P(x_i, y_j)$ (i.e., K_{ij} and L_{ij}, see Equations (34) and (35)) around it (see Figure 1), where $P(x_i, y_j)$ is computed by the difference points around $u(x_i, y_j)$ (see Figure 2).

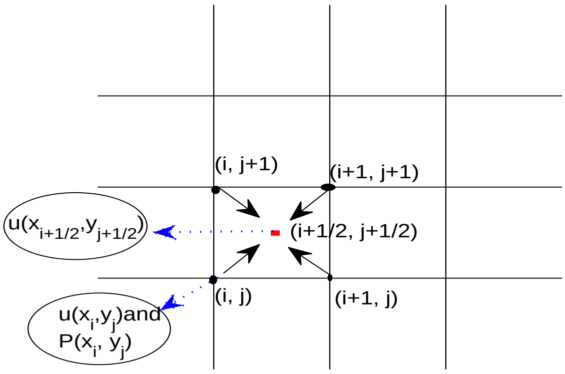


Figure 1. The relationships between $u(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$, and the surrounding points $u(x_i, y_j)$ and partial derivatives $P(x_i, y_j)$.

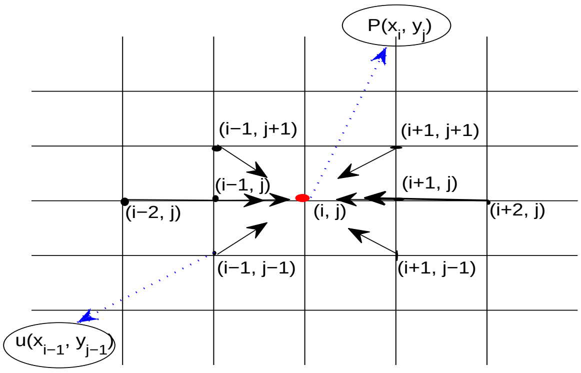


Figure 2. The relationships between $P(x_i, y_j)$ and the surrounding points $u(x_i, y_j)$.

3.2.1. Local Hermite Interpolation and Refinement in the Two-Dimensional Case

For convenience, we denote:

$$K_{i,j} = \frac{\partial u\left(x_i, y_j, t\right)}{\partial x} \overset{\Delta}{=} \left.\frac{\partial u}{\partial x}\right|_{\left(x_i, y_j, t\right)}, \text{ and } L_{i,j} = \frac{\partial u\left(x_i, y_j, t\right)}{\partial y}.$$

Let us consider Hermite bilinear interpolation functions $\Psi_H(x, y, t)$ on the rectangular mesh $[x_i, x_{i+1}] \times [y_j, y_{j+1}] \subset \Omega_h$, where its four vertexes are as follows:

$$z_1 : \left(x_i, y_j, t\right), z_2 : \left(x_{i+1}, y_j, t\right), z_3 : \left(x_i, y_{j+1}, t\right), z_4 : \left(x_{i+1}, y_{j+1}, t\right) \in \Omega_h.$$

On the segment $z_1 - z_2$, let the bilinear interpolation function satisfy the following conditions:

$$\Psi_H(z_1) = u(z_1),\ \Psi_H(z_2) = u(z_2),$$

$$(\Psi_H)_x(z_1) = u_x(z_1), \text{ and } (\Psi_H)_x(z_2) = u(z_2).$$

Based on Reference [6], we can obtain the following Hermite interpolation polynomial:

$$\Psi_H(x_{i+\frac{1}{2}}, y_j, t) = \frac{1}{2}[u(z_1) + u(z_2)] + \frac{h}{8}[u_x(z_1) - u_x(z_2)], \tag{27}$$

where $i = 1, 2, \ldots, N-1$. The interpolation errors are:

$$\begin{aligned} \Psi_H\left(x_{i+\frac{1}{2}}, y_j, t\right) - u\left(x_{i+\frac{1}{2}}, y_j, t\right) &= \frac{u_{xxxx}\left(\xi_i, y_j, t\right)}{4!}\left(x_{i+\frac{1}{2}} - x_i\right)^2\left(x_{i+\frac{1}{2}} - x_{i+1}\right)^2 \\ &= \frac{h_x^4}{384} u_{xxxx}\left(\xi_i, y_j, t\right), i, j = 1, 2, \ldots, N-1. \end{aligned} \tag{28}$$

where ξ_j lies between z_1 and z_2 (see Reference [6]). Thus, we obtain the following approximate computation formula for any $i = 2, 3, \ldots, , N-1, j = 1, 2, \ldots, N-1$:

$$u^k_{i+\frac{1}{2},j} = \frac{1}{2}\left(u^k_{ij} + u^k_{i+1,j}\right) + \frac{h_x}{8}\left(K_{ij} - K_{i+1,j}\right). \tag{29}$$

Similarly, we have also that:

$$u^k_{i,j+\frac{1}{2}} = \frac{1}{2}\left(u^k_{ij} + u^k_{i,j+1}\right) + \frac{h_y}{8}\left(L_{ij} - L_{i,j+1}\right). \tag{30}$$

Therefore, for $i, j = 2, 3, \ldots, N-3$, $u^k_{i+\frac{1}{2},j+\frac{1}{2}}$ can be approximated as follows:

$$\begin{array}{l} u^k_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\left(u^k_{i,j+\frac{1}{2}} + u^k_{i+1,j+\frac{1}{2}}\right) + \frac{h}{8}\left(K_{i,j+\frac{1}{2}} - K_{i+1,j+\frac{1}{2}}\right) \\ = \frac{1}{4}\left(u^k_{ij} + u^k_{i,j+1}\right) + \frac{h}{16}\left(L_{ij} - L_{i,j+1} + L_{i+1,j} - L_{i+1,j+1}\right) + \\ \frac{1}{4}\left(u^k_{i+1,j} + u^k_{i+1,j+1}\right) + \frac{h}{16}\left(K_{ij} - K_{i+1j} + K_{ij+1} - K_{i+1j+1}\right). \end{array} \tag{31}$$

In Section 3.3, we will prove that the above refinement scheme has fourth-order accuracy in the space direction, see Theorem 5.

3.2.2. The Collocation Polynomial in the Two-Dimensional Case

Next, we use the collocation polynomial method to obtain the approximate values of K_{ij} and L_{ij}. For convenience, we consider the sub-domain:

$$[x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}] \subset \Omega,\ i, j = 1, 2, \ldots, N-1,$$

and denote:

$$\xi = x - x_i, \eta = y - y_j, (x, y) \in \Omega_h, (i, j = 1, 2, \ldots, N-1).$$

In order to get the approximation polynomial of u, we consider the polynomial space:

$$H_4 = span\left\{1, \xi, \eta, \xi^2, \xi\eta, \eta^2, \xi^3, \xi^2\eta, \xi\eta^2, \eta^3, \xi^4, \xi^2\eta^2, \eta^4\right\}, \tag{32}$$

and define the approximation polynomial as follows:

$$\begin{array}{c} H(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi^2 + a_4\xi\eta + a_5\eta^2 + a_6\xi^3 + a_7\xi^2\eta + \\ a_8\xi\eta^2 + a_9\eta^3 + a_{10}\xi^4 + a_{11}\xi^2\eta^2 + a_{12}\eta^4. \end{array} \tag{33}$$

Let:

$$\begin{array}{l} H(x_m, y_n) = u(x_m, y_n); m = i-1, i, i+1; n = j-1, j, j+1;\ (m-i)(n-j) \neq 0; \\ \frac{\partial^2 H(x_m,y_n)}{\partial x^2} + \frac{\partial^2 H(x_m,y_n)}{\partial y^2} = \frac{\partial u}{\partial t}(x_m, y_n, t), \end{array}$$

where $n = j; m = i-1, i, i+1$ and $n = j \pm 1, m = i$. Then, we can obtain the following numerical gradient approximate scheme:

$$\begin{array}{l} K_{ij} = \frac{1}{12h}\left[8u(x_{i+1}, y_j, t) - 8u(x_{i-1}, y_j, t) + u(x_{i-2}, y_j, t) - u(x_{i+2}, y_j, t)\right], \\ \quad i = 2, 3, \ldots, N-2, j = 1, 2, \ldots, N-1; \end{array} \tag{34}$$

$$\begin{array}{l} L_{ij} = \frac{1}{12h}\left[8u(x_i, y_{j+1}, t) - 8u(x_i, y_{j-1}, t) + u(x_i, y_{j-2}, t) - u(x_i, y_{j+2}, t)\right], \\ \quad i = 1, 2, \ldots, N-1, j = 2, 3, \ldots, N-2. \end{array} \tag{35}$$

3.3. The Truncation Errors of the Numerical Gradient Scheme

As stated in the previous Section 2, the truncation errors of the compact difference method in Reference [5] are $O(\tau^2 + h^4)$. In fact, the above numerical gradient schemes, Equations (34) and (35), also have the same convergence order.

Theorem 4. *If $u(x, y, t) \in \mathbb{C}^6_x(\Omega)$ and $u(x, y, t) \in \mathbb{C}^6_y(\Omega)$, then we have:*

$$\left|K_{ij} - \frac{\partial u(x_i, y_j, t)}{\partial x}\right| < O\left(h^4\right), (i = 2, 3, \ldots, N-2; j = 1, 2, \ldots, N-1), \tag{36}$$

$$\left| L_{ij} - \frac{\partial u\left(x_i, y_j, t\right)}{\partial y} \right| < O\left(h^4\right), (j = 2, 3, \ldots, N-2; i = 1, 2, \ldots, N-1). \quad (37)$$

Proof. According to Equation (34), we know:

$$\begin{aligned} K_{ij} &= \tfrac{1}{12h}\left[8u\left(x_{i+1}, y_j, t\right) - 8u\left(x_{i-1}, y_j, t\right) + u\left(x_{i-2}, y_j, t\right) - u\left(x_{i+2}, y_j, t\right)\right] \\ &= \tfrac{4}{3}\tfrac{u\left(x_{i+1}, y_j, t\right) - u\left(x_{i-1}, y_j, t\right)}{2h} - \tfrac{1}{3}\tfrac{u\left(x_{i-2}, y_j, t\right) - u\left(x_{i+2}, y_j, t\right)}{4h}. \end{aligned}$$

In addition, according to the Taylor series expansion theorem, we have:

$$\frac{\partial u\left(x_i, y_j, t\right)}{\partial x} = \frac{4}{3}\frac{u\left(x_{i+1}, y_j, t\right) - u\left(x_{i-1}, y_j, t\right)}{2h} - \frac{1}{3}\frac{u\left(x_{i-2}, y_j, t\right) - u\left(x_{i+2}, y_j, t\right)}{4h} + O\left(h^4\right)$$

So:

$$\left| \frac{\partial u\left(x_i, y_j, t\right)}{\partial x} - K_{ij} \right| = O\left(h^4\right).$$

Similarly, we may also prove that:

$$\left| \frac{\partial u\left(x_i, y_j, t\right)}{\partial y} - L_{ij} \right| = O\left(h^4\right).$$

Thus the proof is completed. □

By the above theorem, we know that the accuracy of the numerical gradient schemes in Equations (34) and (35) is $O(h^4)$ in the space direction. In fact, in the intermediate points $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, t)$, the above refinement scheme, Equation (31), also has fourth-order accuracy in space direction.

Theorem 5. *If* $u(x, y, t) \in \mathbb{C}_x^6(\Omega)$ *and* $u(x, y, t) \in \mathbb{C}_y^6(\Omega)$*, then:*

$$\left| u\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, t_k\right) - u^k_{i+\frac{1}{2}, j+\frac{1}{2}} \right| \leq O\left(h^4\right), i, j = 2, 3, \ldots, N-3. \quad (38)$$

Proof. First, by the Taylor expansion of $u\left(x_i, y_j, t_k\right)$ at $(x_{j+\frac{1}{2}}, T)$, we have:

$$\begin{aligned} u\left(x_i, y_j, t_k\right) = {} & u\left(x_{i+\frac{1}{2}}, y_j, t_k\right) - \tfrac{h}{2}u_x\left(x_{i+\frac{1}{2}}, y_j, t_k\right) + \tfrac{h^2}{8}u_{xx}\left(x_{i+\frac{1}{2}}, y_j, t_k\right) - \\ & \tfrac{h^3}{48}u_{xxx}\left(x_{i+\frac{1}{2}}, y_j, t_k\right) + O\left(h^4\right), \end{aligned}$$

$$\begin{aligned} u\left(x_{i+1}, y_j, t_k\right) = {} & u\left(x_{i+\frac{1}{2}}, y_j, t_k\right) + \tfrac{h}{2}u_x\left(x_{i+\frac{1}{2}}, y_j, t_k\right) + \tfrac{h^2}{8}u_{xx}\left(x_{i+\frac{1}{2}}, y_j, t_k\right) + \\ & \tfrac{h^3}{48}u_{xxx}\left(x_{i+\frac{1}{2}}, y_j, t_k\right) + O\left(h^4\right). \end{aligned}$$

Therefore:

$$\begin{aligned} & u\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, t_k\right) = \tfrac{1}{4}\left[u\left(x_i, y_j, t_k\right) + u\left(x_i, y_{j+1}, t_k\right) + u\left(x_{i+1}, y_j, t_k\right) + u\left(x_{i+1}, y_{j+1}, t_k\right)\right] - \\ & \quad \tfrac{h^2}{8}u_{xx}\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, t_k\right) - \tfrac{h^2}{16}[u_{yy}\left(x_i, y_{j+\frac{1}{2}}, t_k\right) - u_{yy}\left(x_{i+1}, y_{j+\frac{1}{2}}, t_k\right)] + O\left(h^4\right) \\ & = \tfrac{1}{4}\left[u\left(x_i, y_j, t_k\right) + u\left(x_i, y_{j+1}, t_k\right) + u\left(x_{i+1}, y_j, t_k\right) + u\left(x_{i+1}, y_{j+1}, t_k\right)\right] - \\ & \quad \tfrac{h}{16}\tfrac{\partial}{\partial y}\delta_y\left[u\left(x_i, y_{j+\frac{1}{2}}, t_k\right) - u\left(x_{i+1}, y_{j+\frac{1}{2}}, t_k\right)\right] - \tfrac{h}{8}\tfrac{\partial}{\partial x}\left[\delta_x u\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, t_k\right)\right] + O\left(h^4\right) \\ & = \tfrac{1}{4}\left(u^k_{ij} + u^k_{i,j+1} + u^k_{i+1,j} + u^k_{i+1,j+1}\right) - \tfrac{h}{16}\left(L_{ij} - L_{i,j+1} + L_{i+1j} - L_{i+1j+1}\right) - \\ & \quad \tfrac{h}{16}\left(K_{ij} - K_{i+1,j} + K_{i,j+1} - K_{i+1,j+1}\right) + O\left(h^4\right) \\ & = u^k_{i+\frac{1}{2}, j+\frac{1}{2}} + O\left(h^4\right). \end{aligned}$$

That is, the conclusion holds. □

In addition, to reduce the total computing time, we also consider the Richardson extrapolation on the H-OCD scheme of Equation (25) in the two-dimensional case. For convenience, we take the following initial–boundary problem as a simple example:

$$\begin{array}{l} u_t - \Delta u = F_u\left(x,y,t\right), \left(x,y,t\right) \in \left(a,b\right) \times \left(c,d\right) \times \left(0,T\right], \\ u\left(x,c,t\right) = u\left(x,d,t\right) = u\left(a,y,t\right) = u\left(b,y,t\right) = 0, \left(x,y\right) \in \left[a,b\right] \times \left[c,d\right], 0 \leq t \leq T, \\ u\left(x,y,0\right) = 0, \left[x,y\right] \in \left[a,b\right] \times \left[c,d\right] \end{array} \tag{39}$$

with the smooth solution $u(x,y,t)$, where:

$$F_u\left(x,y,t\right) = \frac{1}{24}\frac{\partial^3 u\left(x,y,t\right)}{\partial t^3} - \frac{1}{8}\frac{\partial^4 u\left(x,y,t\right)}{\partial x^2 \partial t^2}.$$

Theorem 6. *Let $u\left(x,y,t\right) \in \mathbb{C}^{8,6}\left(\Omega_h \times [0,T]\right)$ be the solution of Equation (22) for the initial–boundary problem of Equation (39), and let $u_{ij}^k\left(h,\tau\right)$ be the numerical solution of the H-OCD scheme, Equation (25), with time step τ and spatial step h. Then:*

$$\max_{1 \leq i,j \leq N-1, 1 \leq k \leq M} \left| u\left(x_i, y_j, t_k\right) - \left[\frac{4}{3} u_{ij}^{2k}\left(h, \frac{\tau}{2}\right) - \frac{1}{3} u_j^k\left(h,\tau\right)\right]\right| = O\left(\tau^4 + h^4\right).$$

Proof. The proof is similar to that of Theorem 3. □

In addition, the corresponding numerical experiments will be shown in Table 10.

4. Numerical Experiments

4.1. Numerical Experiments for the One-Dimensional Case

Example 1. *Let $u(x,0) = \sin(\pi x), u(0,t) = u(1,t) = 0$ for Equation (1) with $(x,t) \in (0,1) \times (0,T]$. Then the exact solution of Equation (1) is:*

$$u(x,t) = \exp(-\pi^2 t)\sin(\pi x).$$

Next, let us observe and compare the numerical solutions and computation times for the same number of points in the above two schemes.

First, we note that matrix computations are based on LAPACK, and the optimized basic linear algebra subroutines (BLAS) on the Matlab platform which speeds up matrix multiplications and the LAPACK routines themselves, according to Matlab user manual. Therefore, all the numerical experiments were performed in Matlab 2011b. In addition, for convenience, we denote $Rate(h) = \log_2\left(\frac{Error(h)}{Error(\frac{h}{2})}\right)$ and $Error(h) = \max_{x_k = x_0 + kh, k=0,1,\ldots N}\{|\ (u(x_k,T) - u_k^T)\ |\}$, where $u(x_k,T)$ represents the exact solution and u_k^T is the numerical solution. Let:

$$Error(P) = \max_{x_k = x_0 + kh, k=0,1,\ldots N}\left\{\left|\frac{\partial u}{\partial x}(x_j,T) - P_k^T\right|\right\}.$$

Table 1 lists the computational results for the mesh grid points, the intermediate points, and u_x with different spatial step sizes, when time step size is fixed as $\tau = 1/100,000$. We can see that the convergence order in space can reach $O(h^4)$ which is consistent with the theoretical analysis in this article.

Table 1. Errors and rate for calculations of the intermediate points and numerical gradients P in space direction, with $\tau = 1/100,000$.

h	Mesh Grid Points	Rate	Intermediate Points	Rate	P (i.e., $\frac{\partial u}{\partial x}$)	Rate
	Error		Error		Error	
1/4	8.3491×10^{-7}	4.0355	5.4790×10^{-7}	3.5729	3.7721×10^{-6}	3.7762
1/8	5.0915×10^{-8}	4.0073	4.6043×10^{-8}	3.9694	2.5732×10^{-7}	3.9370
1/16	3.1660×10^{-9}	4.0045	2.9394×10^{-9}	3.9952	1.7976×10^{-8}	3.9823
1/32	1.9725×10^{-10}	4.0466	1.8433×10^{-10}	4.0474	1.1374×10^{-9}	3.9715
1/64	1.1936×10^{-11}	-	1.1148×10^{-11}	-	7.2504×10^{-11}	-

Figure 3 displays the errors curves with different step sizes for the mesh grid points (by the H-OCD method) and for all the points (by the new method) when $T = 1$. This displays that the changes of the truncation errors in the mesh grid points and the other points are large with large h and τ. At the same time, the shape of the curves is approximately the same. That means that the points obtained through the new method are not worse than the H-OCD method.

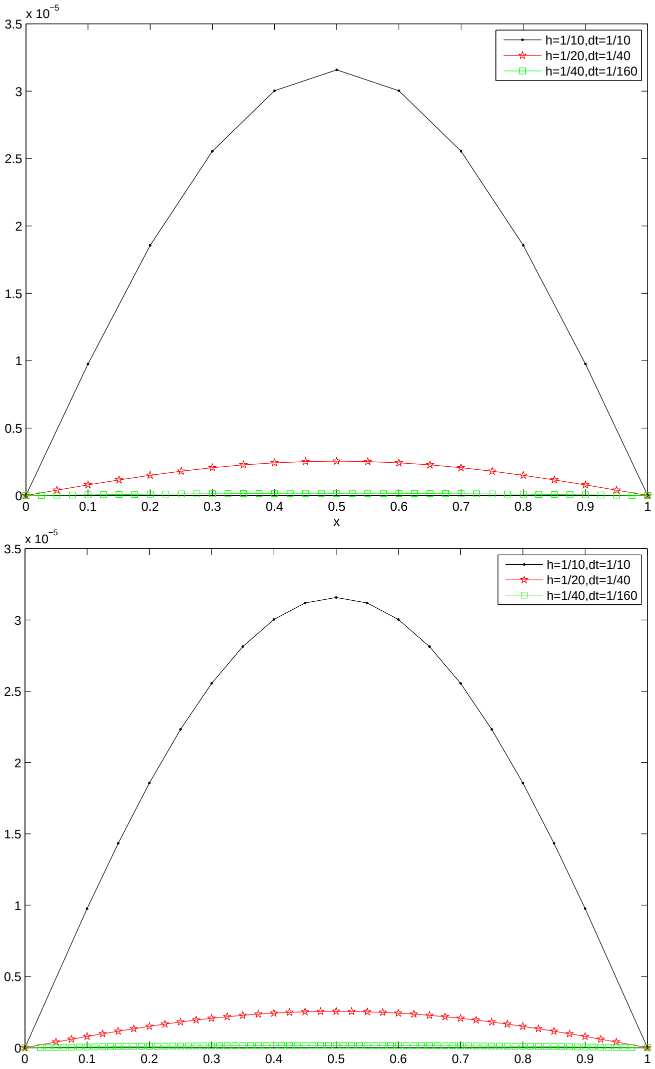


Figure 3. (**Top**) The error curves of the mesh grid points in the high-order compact difference (H-OCD) method, when $T = 1$; (**Bottom**) The error curves of all the points in the new method, when $T = 1$.

Figure 4 shows that the numerical solution (the red line), for all the points calculated by the new method, is closer to the exact solution (the green line) when $h = 1/8,\ \tau = 1/100, T = 1$. That is to say, the simulation result of the red line is better than the other one. In order to make Figure 4 more clear, we choose $h = 1/8$.

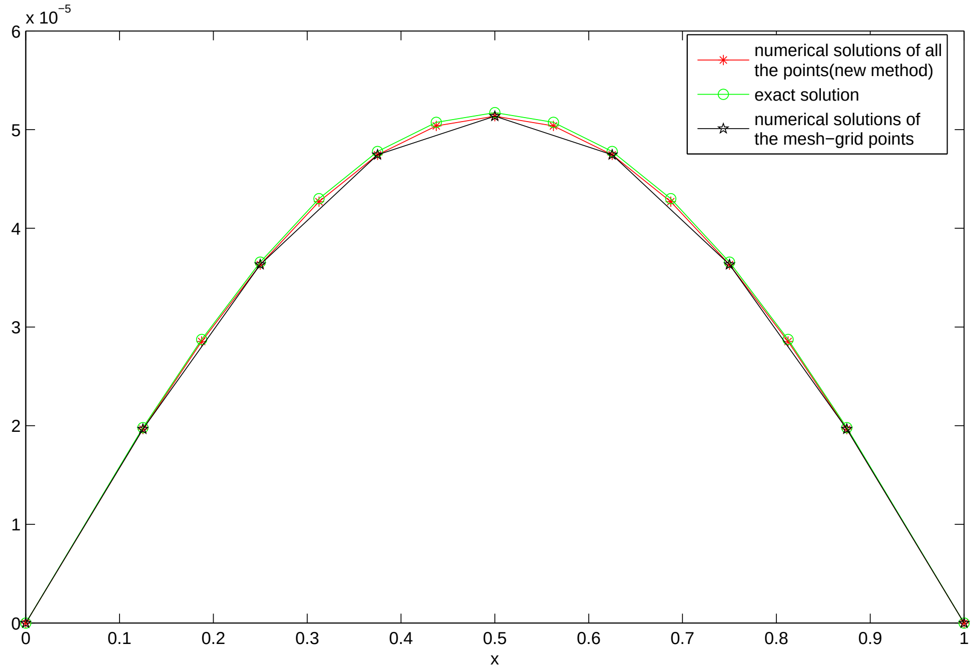


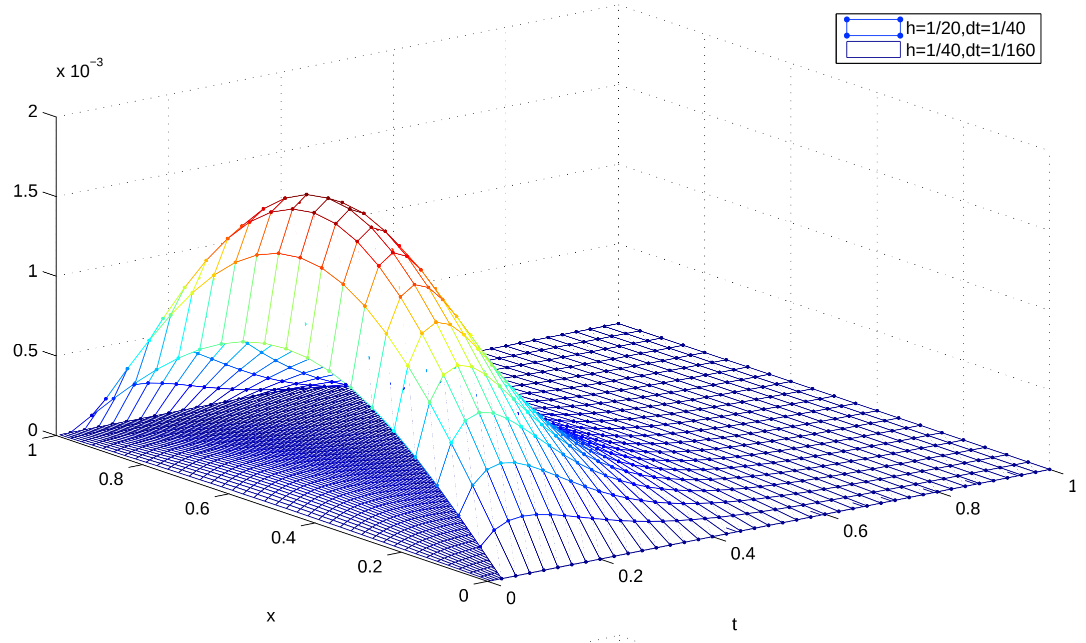
Figure 4. The numerical solutions and the exact solution for all the points in the new method and for the mesh grid points in the H-OCD method, when $h = 1/8$, $\tau = 1/100$, $t = 1$.

Figure 5 displays the error surface maps with different step sizes in both the spatial and time directions for the mesh grid points (by the H-OCD method) and for all the points (by the new method) when $t = 1$. This display that the truncation errors in the mesh grid points and the other points are large when h and τ are large. At the same time, the shape of the curves is approximately the same. That means that the points obtained through the new method work very well too.

In addition, from Table 2, we know that the H-OCD method takes more time to compute the same number of difference points, compared to the new method. For example, if we need the numerical solutions of 255 points to simulate the real figure, we only need $h = 1/128$. Through the method proposed in this article, we can get the numerical solutions for the 255 points.

Table 2. Errors of the numerical solutions in the mesh grid points (H-OCD method), and in all the points (numerical gradient scheme), and the total time to calculate the solutions when $\tau = h^2$, $T = 1, h = 1/(n-1)$.

Grid Node Number	H-OCD Method	Time	Numerical Gradient Scheme	Time
	Error		Error	
N = 15	6.0041×10^{-8}	0.1544	9.5518×10^{-7}	0.0312
N = 31	3.7541×10^{-9}	0.5725	6.0041×10^{-8}	0.1560
N = 63	2.3464×10^{-10}	2.5389	3.7623×10^{-9}	0.5839
N = 127	1.4665×10^{-11}	11.505	2.3536×10^{-10}	2.7233
N = 255	9.1633×10^{-13}	$\boxed{142.64}$	1.4713×10^{-11}	$\boxed{12.8879}$



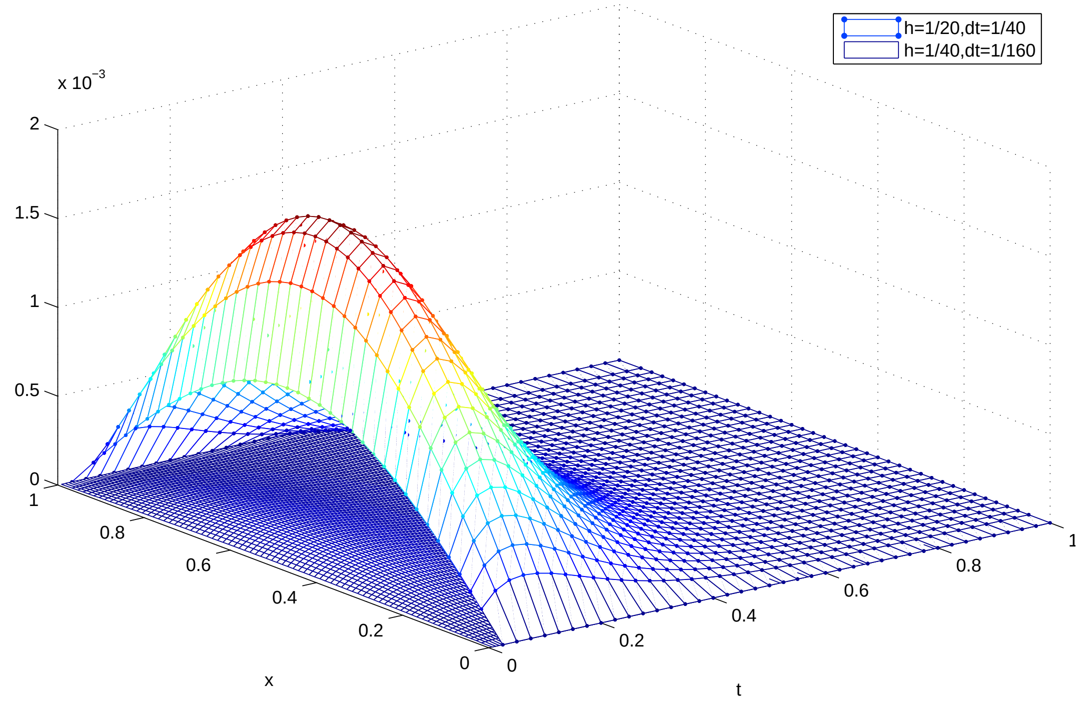


Figure 5. (**Top**) The error surface map of the compact difference scheme for the mesh grid points; (**Bottom**) The error surface map of the numerical gradient scheme for the intermediate points.

Example 2. *For* $u(x,0) = \exp(x), u(0,t) = \exp(t), u(1,t) = \exp(1+t)$ *with* $(x,t) \in (0,1) \times (0,T]$*, the exact solution of Equation (1) is:*

$$u(x,t) = \exp(x+t).$$

In the following, we compare the numerical solution with the exact solution as follows (see Tables 3–6).

Table 3. Errors and rates of the H-OCD scheme, Equations (7)–(9), the intermediate points (new method), and the numerical gradient P_j in the spatial direction, with $\tau = 1/100,000$.

h	Mesh-Grid Points	Rate	Intermediate Points	Rate	P (i.e., $\frac{\partial u}{\partial x}$)	Rate
	Error		Error		Error	
1/4	8.4064×10^{-6}	3.9974	2.9136×10^{-4}	3.8091	6.8324×10^{-3}	2.7543
1/8	5.2636×10^{-7}	3.9895	2.0787×10^{-5}	3.9099	1.1027×10^{-3}	2.8763
1/16	3.3138×10^{-8}	3.9993	1.3828×10^{-6}	3.9564	1.3792×10^{-4}	2.9379
1/32	2.0721×10^{-9}	4.0472	8.9081×10^{-8}	3.9787	1.7998×10^{-5}	2.9689
1/64	1.2534×10^{-10}	-	5.6505×10^{-9}	-	2.2987×10^{-6}	-

Table 4. Errors and rates of the H-OCD scheme, Equations (7)–(9), the intermediate points (new method), and the numerical gradient P_j in the time direction, with $h = 1/10,000$.

τ	Compact Difference	Rate	Intermediate Points	Rate	P (i.e., $\frac{\partial u}{\partial x}$)	Rate
	Error		Error		Error	
1/10	4.3449×10^{-4}	1.9988	4.3449×10^{-4}	1.9988	2.0491×10^{-3}	1.9769
1/20	1.0871×10^{-4}	1.9998	1.0871×10^{-4}	1.9998	5.2055×10^{-4}	1.9887
1/40	2.7183×10^{-5}	1.9999	2.7183×10^{-5}	1.9999	1.3116×10^{-4}	1.9945
1/80	6.7960×10^{-6}	2.0005	6.7960×10^{-6}	2.0005	3.2914×10^{-5}	1.9987
1/160	1.6984×10^{-6}	2.0053	1.6984×10^{-6}	2.0053	8.2362×10^{-6}	2.0053
1/320	4.2303×10^{-7}	2.0246	4.2303×10^{-7}	2.0246	2.0515×10^{-6}	2.0259
1/640	1.0397×10^{-7}	-	1.0397×10^{-7}	-	5.0375×10^{-7}	-

From Tables 3 and 4, we know that the numerical results are consistent with our theoretical results.

Table 5. Errors of the numerical solutions for the mesh grid points and for all points in the new method, and the time to get these solutions when $\tau = h^2, t = 1, h = 1/(n-1)$.

Grid Node Number	H-OCD Method	Time	Numerical Gradient Scheme	Time
	Error		Error	
N = 15	6.2975×10^{-7}	0.1248	1.3540×10^{-5}	0.0406
N = 31	3.9376×10^{-8}	0.5725	1.1199×10^{-6}	0.1265
N = 63	2.4630×10^{-9}	3.1590	8.0259×10^{-8}	0.5959
N = 127	1.5453×10^{-10}	20.117	5.3654×10^{-9}	3.1844
N = 255	1.2050×10^{-11}	$\boxed{128.44}$	3.4669×10^{-10}	$\boxed{21.542}$

In addition, the conclusion in the space direction we get from Table 5 is the same as that from Table 2. Thus, combining with Figure 6, the advantage of the numerical gradient scheme is obvious. In Table 6, we consider the Richardson extrapolation on the H-OCD scheme of Equations (7)–(9) in the time direction. This result is consistent with Theorem 3.

Table 6. Errors and rates for all points in the new method for Examples 1 and 2 when $\tau = h, T = 1$.

$\tau = h$	Example 1	$\frac{error(h,\tau)}{error(h/2,\tau/2)}$	Example 2	$\frac{error(h,\tau)}{error(h/2,\tau/2)}$
	Error		Error	
1/8	5.5147×10^{-6}	14.2567	2.0369×10^{-5}	14.9661
1/16	3.8682×10^{-7}	15.6251	1.3610×10^{-6}	15.4867
1/32	2.4756×10^{-8}	15.9066	8.7881×10^{-8}	15.7438
1/64	1.5563×10^{-9}	15.9741	5.5819×10^{-9}	15.8711
1/128	9.7429×10^{-11}	15.9934	3.5170×10^{-10}	15.9209
1/256	6.0918×10^{-12}	-	2.2091×10^{-11}	-

4.2. Numerical Experiments for the Two-Dimensional Case

Example 3. *When:*

$$u(x,y,0) = \sin(\pi x)\sin(\pi y),$$
$$u(0,y,t) = u(1,y,t) = u(x,0,t) = u(x,1,t) = 0,$$

the exact solution of Equation (22) is:

$$u(x,y,t) = e^{-2\pi^2 t}\sin(\pi x)\sin(\pi y), (x,y,t) \in \Omega \times (0,T].$$

Next, let us observe and compare the numerical solutions from the different methods.

Table 7 lists the computational results for the mesh grid points and intermediate points with different spatial step sizes, when the time step size is fixed as $\tau = 1/100,000$. We can see that the convergence orders in space can reach $O(h^4)$ which is consistent with the theoretical analysis (see Theorems 4 and 5) in this article. In addition, from Table 8, we see also that the numerical gradient scheme has the same convergence order $O(\tau^2 + h^4)$ as the H-OCD method when the time and space step sizes are the same.

Table 7. Errors and rates for intermediate points and numerical gradients P in spatial direction with $\tau = 1/100,000$.

h	H-OCD Mesh-Grid Points	Rate	Intermediate Points (New Method)	Rate
	Error		Error	
1/4	5.3017×10^{-11}	3.9403	-	-
1/8	3.4536×10^{-12}	3.9878	4.6576×10^{-12}	3.9629
1/16	2.1769×10^{-13}	3.9805	2.9869×10^{-13}	3.9791
1/32	1.3791×10^{-14}	-	1.8939×10^{-14}	-

Table 8. Errors and rates of all points (new method) for Example 3 when $\tau = h^2, h = 1/n - 1, T = 1$.

N	H-OCD Method	$\frac{error(h,\tau)}{error(h/2,\tau/2)}$	Numerical Gradient	$\frac{error(h,\tau)}{error(h/2,\tau/2)}$
	Error		Error	
N = 5	1.6485×10^{-9}	9.7908	1.8257×10^{-9}	11.0802
N = 10	1.6838×10^{-10}	15.6079	1.6477×10^{-10}	15.3196
N = 20	1.0788×10^{-11}	15.9751	1.0755×10^{-11}	15.9015
N = 40	6.7530×10^{-13}	-	6.7638×10^{-13}	-

In addition, Table 9 and Figure 6 also show similar results to those of Table 5 and Figures 3 and 4, respectively. Table 10 lists the computational results for the Richardson extrapolation scheme.

These results show that the convergence order in time direction can reach $O(\tau^4)$, which is consistent with the theoretical analysis (see Theorem 6).

Table 9. A comparison of the computation time between the H-OCD method and the numerical gradient scheme.

Grid Number	H-OCD Method		Grid Number	Numerical Gradient	
	Error	Time		Error	Time
n = 16	1.6485×10^{-9}	0.0374	n = 17	1.8257×10^{-9}	0.0421
n = 81	1.6838×10^{-10}	0.4563	n = 117	1.6838×10^{-10}	0.5756
n = 224	3.3602×10^{-11}	2.8782	n = 433	3.4081×10^{-11}	2.8860
n = 361	1.0788×10^{-11}	9.0527	n = 745	1.0788×10^{-11}	10.8556
n = 624	4.4056×10^{-12}	25.2347	n = 1233	4.4370×10^{-12}	26.8556
n = 899	2.1338×10^{-12}	63.2506	$\boxed{\text{n = 1783}}$	2.1347×10^{-12}	$\boxed{66.8199}$
$\boxed{\text{n = 1599}}$	6.7530×10^{-13}	$\boxed{422.8602}$	n = 3183	6.7638×10^{-13}	424.2073

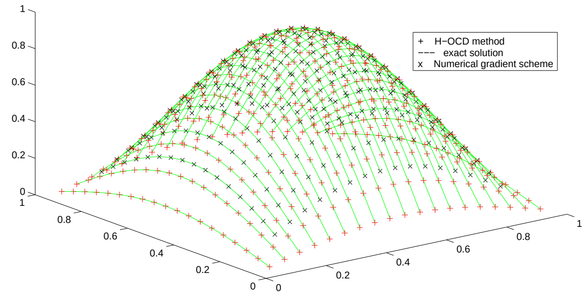


Figure 6. Curves of the numerical solutions and the exact solution for Example 3 when $h = 1/20$, $\tau = 1/400, T = 1$.

Table 10. The convergence order for the Richardson extrapolation scheme of Example 3 when $\tau = h/20, T = 1$.

h	H-OCD Method	$\frac{error(h,\tau)}{error(h/2,\tau/2)}$	Numerical Gradient	$\frac{error(h,\tau)}{error(h/2,\tau/2)}$
	Error		Error	
h = 1/5	2.1070×10^{-11}	14.1419	3.2895×10^{-11}	16.4508
h = 1/10	1.4899×10^{-12}	15.9254	1.9996×10^{-12}	15.7474
h = 1/20	9.3555×10^{-14}	15.9822	1.2698×10^{-13}	15.9392
h = 1/40	5.8537×10^{-15}	15.9955	7.9665×10^{-15}	15.9848
h = 1/40	3.6596×10^{-16}	-	4.9838×10^{-16}	-

For this two-dimension problem, we have obtained similar experimental results as the previous one-dimension problem, which shows that this method is effective.

5. Conclusions

Recently, many people have devoted themselves to the development of numerical approximations of heat equation problems. By numerical comparisons, we know that the high-order compact difference scheme (H-OCD) of Reference [5] is better than the traditional numerical schemes. In this article, we further improve this method to a new numerical gradient scheme. Moreover, our theoretical analysis and numerical experiments show that this numerical gradient scheme has the same convergence order as the H-OCD in Reference [5]. We hope that this is a useful supplement to the existing results. The results also provide potential for several applications for example those in References [11–14].

In addition, many new methods have recently emerged for solving differential equations such as the Lie algebra method [15,16]. Absorbing or drawing on the advantages of these methods to obtain better results is a problem worthy of further study.

Author Contributions: Conceptualization, E.-J.Z.; Funding acquisition, H.-B.L.; Date, X.-M.G.; Supervision, H.-B.L. and X.-M.G.; Writing-original draft, M.-Y.S. and H.-B.L.; and Writing-review and editing, H.-B.L. and X.-M.G.

Funding: This work was financially supported by the National Natural Science Foundation of China (11271001, 11101071), and the Fundamental Research Funds for the Central Universities(ZYGX2016J138).

Acknowledgments: The authors would like to thank reviewers and editors for providing us some good suggestions and help.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, J.W.; Tang, H.M. *Numerical Methods of Differential Equations*; Science Press: Berlin, Germany, 2007.
2. Morton, K.W. *Numerical Solutions of Partial Differential Equations*, 2nd ed.; Posts and Telecom Press: Beijing, China, 2006.
3. Sun, Z.Z. Compact Difference Schemes for Heat Equations with the Neumann Boundary Conditions. *Numer. Method Partial Differ. Equ.* **2009**, *25*, 1320–1341. [CrossRef]
4. Zhang, W.S. *Finite Difference Methods for Partial Difference Equations in Science Computation*; Higher Education Press: Beijing, China, 2006.
5. Sun, Z.Z.; Zhang, Z.B. A Linearized Compact Difference Schemes for a Class of Nonlinear Delay Partial Differential Equations. *Appl. Math. Model.* **2013**, *37*, 742–752. [CrossRef]
6. Timothy, S. *Numerical Analysis*; Posts and Telecom Press: Beijing, China, 2010.
7. Liao, H.L.; Sun, Z.Z.; Shi, H.S. Error Estimate of Fourth-order Compact Scheme for Linear Schromdinger Equations. *SIAM Numer. Anal.* **2010**, *47*, 4381–4401. [CrossRef]
8. Richardson, L.F. The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. *Philos. Trans. R. Soc. A* **1911**, *210*, 307–357. [CrossRef]
9. Liao, H.L.; Sun, Z.Z. Maxmum Norm Error Bounds of ADI and Compact ADI methods for Solving Parabolic Equations. *Numer. Method Partial Differ. Equ.* **2010**, *26*, 37–60. [CrossRef]
10. Sun, Z.Z. *Numerical Methods of Partial Differential Equations*; Science Press: Berlin, Germany, 2012.
11. Ikota, R. E. Yanagida. Stability of stationary interfaces of binary-tree type. *Calc. Var. Partial Differ. Equ.* **2004**, *22*, 375–389 [CrossRef]
12. Dassios, I. Stability of basic steady states of networks in bounded domains. *Comput. Math. Appl.* **2015**, *70*, 2177–2196. [CrossRef]
13. Dassios, I. Stability of Bounded Dynamical Networks with Symmetry. *Symmetry* **2018**, *10*, 121. [CrossRef]
14. Xiaofeng, R., Wei, J. A Double Bubble in a Ternary System with Inhibitory Long Range Interaction. *Arch. Ration. Mech. Anal.* **2013**, *208*, 201–253.
15. Shang, Y. A Lie algebra approach to susceptible-infected-susceptible epidemics. *Electron. J. Differ. Equ.* **2012**, *2012*, 147–154.
16. Shang, Y. Analytical Solution for an In-host Viral Infection Model with Time-inhomogeneous Rates. *Acta Phys. Pol. Ser. B* **2015**, *46*, 1567–1577. [CrossRef]