## Article

# A High-Order Compact (HOC) Implicit Difference Scheme and a Multigrid Method for Solving 3D Unsteady Reaction Diffusion Equations 

Lili Wu ${ }^{1,2}$ and Xiufang Feng ${ }^{\text {1,* }}$<br>1 School of Mathematics and Statistics, Ningxia University, Yinchuan 750021, China; wlltynx@163.com<br>2 School of Mathematics and Computer Science, Ningxia Normal University, Guyuan 756000, China<br>* Correspondence: xf_feng@nxu.edu.cn; Tel.: +86-139-0959-7502

Received: 7 November 2019; Accepted: 5 December 2019; Published: 9 December 2019


#### Abstract

A high-order compact (HOC) implicit difference scheme is proposed for solving three-dimensional (3D) unsteady reaction diffusion equations. To discretize the spatial second-order derivatives, the fourth-order compact difference operators are used, and the third- and fourth-order derivative terms, which appear in the truncation error term, are also discretized by the compact difference method. For the temporal discretization, the multistep backward Euler formula is used to obtain the fourth-order accuracy, which matches the spatial accuracy order. To accelerate the traditional relaxation methods, a multigrid method is employed, and the computational efficiency is greatly improved. Numerical experiments are carried out to validate the accuracy and efficiency of the present method.


Keywords: reaction diffusion equation; multigrid method; high-order compact scheme; implicit scheme; nonlinear reaction term

## 1. Introduction

Reaction diffusion systems are widely used to describe the similar diffusion of all kinds of phenomena in modern science, such as the diffusion of gas, the infiltration of liquid concentration, gene diffusion in population, etc. Most of these systems contain nonlinear reaction terms, which make it difficult to obtain the exact solutions [1,2]. Therefore, it is most meaningful to investigate the accurate, efficient and stable numerical approaches of these systems.

In past decades, many researchers have been focusing on using efficient numerical methods for solving reaction diffusion equations. For instance, Madzvamuse and Maini [2] studied the implications of a mesh structure on the numerically computed solutions of a reaction diffusion model equation. Wang and Guo [3] proposed a spatially fourth-order and temporally second-order accuracy implicit finite difference monotone scheme for solving nonlinear reaction diffusion equations. Araújo et al. [4] proved the convergence properties of a class of implicit and semi-implicit finite difference schemes for solving nonlinear reaction diffusion equations. Zhu et al. [5] proposed Galerkin methods for solving the reaction diffusion system. Fernandes et al. [6] proposed an alternating direction implicit (ADI) method based on an extrapolated Crank-Nicolson orthogonal spline collocation method. Gu et al. [7] proposed a high-order compact ADI difference scheme to solve a 3D reaction diffusion equation with Dirichlet boundary condition. Liao [8] extended the method to solve this 3D reaction diffusion equation with Neumann boundary condition.

Wang [9] proposed a higher-order compact ADI method and a monotonous iterative algorithm for solving the reaction diffusion equations. Tian [10] proposed a cubic spline difference scheme for solving the one-dimensional (1D) reaction diffusion equation. Hasnain et al. [11] used the fourth-order

Douglas implicit scheme to solve the 3D reaction diffusion equation with nonlinear source term. Ge et al. [12] proposed a high-order compact (HOC) scheme and an adaptive method to solve the 1D nonlinear degenerate singular reaction diffusion equations.

There are also other high-order methods that have been developed to solve the reaction diffusion equation with the convection term. For instance, Kaya [13] developed two finite difference schemes with the Crank-Nicolson method and backward Euler formula, respectively, for time discretization. Zhu and Rui [14] proposed an HOC difference scheme and an adaptive algorithm to solve a 1D convection diffusion reaction equation with nonlinear source terms. Iagar and Sánchez [15] proposed a method using the upper and lower solutions to study the existence and characteristics of blow up of the quasilinear reaction diffusion equation. Jiang and Zhang [16] proposed a method with arbitrary order of accuracy using the weighted essentially non-oscillatory approach. Jha and Singh [17] addressed the exponential basis and compact formulation when solving 3D convection diffusion reaction equations. Piquet et al. [18] presented a highly parallel finite difference method and a domain decomposition method and then applied them to solve the problems in computational fluid dynamics. Taken together, there are many references on nonlinear systems with source terms, and most of these methods can achieve fourth-order accuracy in spatial directions but only second-order accuracy in temporal direction. Therefore, it is still meaningful to identify a novel numerical method for solving the 3D nonlinear reaction diffusion equation.

This paper aims to develop an HOC difference scheme combined with the multigrid method to solve the 3D unsteady reaction diffusion equations; therefore, an unconditionally stable and high-order five-level fully implicit scheme is proposed. The main significances of this study are that the temporal accuracy of the developed scheme is able to reach the fourth-order and a multigrid method is used to accelerate the traditional iterative methods for solving the linear systems. The outline of this paper is as follows. In Section 2, the spatial derivative term is discretized by the fourth-order compact difference formula, and the time derivative term is discretized by the fourth-order backward Euler formula. For the calculation of the start-up levels, the first and second time levels are discretized using the Crank-Nicolson method which is the second-order accuracy, and the third time level is discretized by the third-order backward Euler formula. In Section 3, a multigrid algorithm is designed to solve the linear systems. In Section 4, numerical experiments are conducted to validate the present method. In Section 5, a brief conclusion is given.

## 2. HOC Scheme for Unsteady Reaction Diffusion Equation

We consider the 3D unsteady reaction diffusion equation using Dirichlet boundary conditions. We will discuss how to design the fourth-order compact scheme for solving the 3D unsteady reaction diffusion equation in the following.

### 2.1. HOC Main Scheme

Consider the unsteady form of the following equations:

$$
\begin{gather*}
\frac{\partial u}{\partial t}=\varepsilon\left(\frac{\partial^{2} u}{\partial x^{2}}+\frac{\partial^{2} u}{\partial y^{2}}+\frac{\partial^{2} u}{\partial z^{2}}\right)+f(x, y, z, t, u), \quad(x, y, z, t) \in \Omega \times(0, T]  \tag{1}\\
u(x, y, z, 0)=\varphi(x, y, z),(x, y, z) \in \Omega  \tag{2}\\
u(x, y, z, t)=\phi(x, y, z, t),(x, y, z, t) \in \partial \Omega \times(0, T] \tag{3}
\end{gather*}
$$

Assume that the problem domain is rectangular $\Omega \in[a, b]^{3}$ and $a$ and $b$ are constants. $\partial \Omega$ is the boundary and $(0, T]$ is the time interval. $\varepsilon$ is the positive constant diffusion term coefficient, $u(x, y, z, t)$ is the unknown function, $f(x, y, z, t, u)$ is the nonlinear source function term, $\varphi(x, y, z)$ and $\phi(x, y, z, t)$ are the smooth functions. We discretize the domain $\Omega$ with uniform grids and split $[a, b]^{3} \times(0, T]$ as meshes $N^{3} \times M\left(N, M \in Z^{+}\right)$; set $\tau=\frac{T}{M}$ to be the temporal grid step length and $h$ to be the spatial
grid step length in the $x$-, $y$ - and $z$-directions. The grid points are labeled as $\left(x_{i}, y_{j}, z_{k}, t_{n}\right)$, where $x_{i}=a+i h ; \quad y_{j}=a+j h ; z_{k}=a+k h ; t_{n}=n \tau ; i, j$ and $k=0,1, \cdots, N$ and $n=0,1, \cdots, M$.

For the sake of simplicity, we use Figure 1 to denote the placement of the 19 points. We number the mesh points $(x, y, z),(x+h, y, z),(x, y+h, z),(x-h, y, z),(x, y-h, z),(x, y, z+h),(x, y, z-h)$, $(x+h, y+h, z),(x-h, y+h, z),(x-h, y-h, z),(x+h, y-h, z),(x+h, y, z+h),(x, y+h, z+h)$, $(x-h, y, z+h),(x, y-h, z+h),(x+h, y, z-h),(x, y+h, z-h),(x-h, y, z-h)$ and $(x, y-h, z-h)$ as $0,1,2,3 \cdots 18$, respectively.


Figure 1. Grid point stencil.
The details of the finite difference operators are given as follows:

$$
\begin{gather*}
\delta_{x} u_{0}=\frac{u_{1}-u_{3}}{2 h}, \delta_{y} u_{0}=\frac{u_{2}-u_{4}}{2 h}, \delta_{z} u_{0}=\frac{u_{5}-u_{6}}{2 h},  \tag{4}\\
\delta_{x}^{2} u_{0}=\frac{u_{1}-2 u_{0}+u_{3}}{h^{2}}, \delta_{y}^{2} u_{0}=\frac{u_{2}-2 u_{0}+u_{4}}{h^{2}}, \delta_{z}^{2} u_{0}=\frac{u_{5}-2 u_{0}+u_{6}}{h^{2}} \tag{5}
\end{gather*}
$$

The model Equation (1) can be rewritten as follows:

$$
\begin{equation*}
\frac{\partial^{2} u}{\partial x^{2}}+\frac{\partial^{2} u}{\partial y^{2}}+\frac{\partial^{2} u}{\partial z^{2}}=\frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right) \tag{6}
\end{equation*}
$$

Using the Taylor series expansions at the grid point $(x, y)$, we have

$$
\begin{align*}
& \left(\frac{\partial^{2} u}{\partial x^{2}}\right)_{0}=\delta_{x}^{2} u_{0}-\frac{h^{2}}{12}\left(\frac{\partial^{4} u}{\partial x^{4}}\right)_{0}+O\left(h^{4}\right)  \tag{7}\\
& \left(\frac{\partial^{2} u}{\partial y^{2}}\right)_{0}=\delta_{y}^{2} u_{0}-\frac{h^{2}}{12}\left(\frac{\partial^{4} u}{\partial y^{4}}\right)_{0}+O\left(h^{4}\right)  \tag{8}\\
& \left(\frac{\partial^{2} u}{\partial z^{2}}\right)_{0}=\delta_{z}^{2} u_{0}-\frac{h^{2}}{12}\left(\frac{\partial^{4} u}{\partial z^{4}}\right)_{0}+O\left(h^{4}\right) \tag{9}
\end{align*}
$$

To keep the scheme compact, the fourth derivatives need to be discretized. We differentiate Equation (6) with $x, y, z$, respectively as

$$
\begin{align*}
& \frac{\partial^{4} u}{\partial x^{4}}=\frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right)_{x x}-\frac{\partial^{4} u}{\partial x^{2} \partial y^{2}}-\frac{\partial^{4} u}{\partial x^{2} \partial z^{2}}  \tag{10}\\
& \frac{\partial^{4} u}{\partial y^{4}}=\frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right)_{y y}-\frac{\partial^{4} u}{\partial y^{2} \partial x^{2}}-\frac{\partial^{4} u}{\partial y^{2} \partial z^{2}} \tag{11}
\end{align*}
$$

$$
\begin{equation*}
\frac{\partial^{4} u}{\partial z^{4}}=\frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right)_{z z}-\frac{\partial^{4} u}{\partial z^{2} \partial x^{2}}-\frac{\partial^{4} u}{\partial z^{2} \partial y^{2}} \tag{12}
\end{equation*}
$$

Then, we substitute Equations (7)-(12) into Equation (6) to obtain the following difference equation:

$$
\begin{gather*}
\delta_{x}^{2} u_{0}-\frac{h^{2}}{12}\left[\frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right)_{x x 0}-\left(\frac{\partial^{4} u}{\partial x^{2} \partial y^{2}}\right)_{0}-\left(\frac{\partial^{4} u}{\partial x^{2} \partial z^{2}}\right)_{0}\right]+\delta_{y}^{2} u_{0}-\frac{h^{2}}{12}\left[\frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right)_{y y 0}-\left(\frac{\partial^{4} u}{\partial y^{2} \partial x^{2}}\right)_{0}-\left(\frac{\partial^{4} u}{\partial y^{2} \partial z^{2}}\right)_{0}\right]  \tag{13}\\
+\delta_{z}^{2} u_{0}-\frac{h^{2}}{12}\left[\frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right)_{z z 0}-\left(\frac{\partial^{4} u}{\partial z^{2} \partial x^{2}}\right)_{0}-\left(\frac{\partial^{4} u}{\partial z^{2} \partial y^{2}}\right)_{0}\right]=\frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right)_{0}+O\left(h^{4}\right)
\end{gather*}
$$

We notice that

$$
\begin{equation*}
\left(\frac{\partial^{4} u}{\partial x^{2} \partial y^{2}}\right)_{0}=\delta_{x}^{2} \delta_{y}^{2} u_{0}+O\left(h^{2}\right), \quad\left(\frac{\partial^{4} u}{\partial x^{2} \partial z^{2}}\right)_{0}=\delta_{x}^{2} \delta_{z}^{2} u_{0}+O\left(h^{2}\right), \quad\left(\frac{\partial^{4} u}{\partial y^{2} \partial z^{2}}\right)_{0}=\delta_{x}^{2} \delta_{z}^{2} u_{0}+O\left(h^{2}\right) \tag{14}
\end{equation*}
$$

Omitting the truncation error term, we derive a compact difference scheme as the following:

$$
\begin{equation*}
\frac{1}{h^{2}}\left(2 \sum_{l=1}^{6} u_{l}+\sum_{l=1}^{18} u_{l}-24 u_{0}\right)=\frac{6}{\varepsilon}\left(\frac{\partial u}{\partial t}-f\right)_{0}+\frac{h^{2}}{2 \varepsilon}\left[\left(\frac{\partial u}{\partial t}-f\right)_{x x}+\left(\frac{\partial u}{\partial t}-f\right)_{y y}+\left(\frac{\partial u}{\partial t}-f\right)_{z z}\right]_{0} \tag{15}
\end{equation*}
$$

To obtain an HOC scheme based on the mixed time derivatives $\frac{\partial^{3} u}{\partial t \partial x^{2}}, \frac{\partial^{3} u}{\partial t \partial y^{2}}$ in Equation (15), we only need to use the following second-order backward Euler formula [19] for discretization:

$$
\begin{equation*}
\left(\frac{\partial u}{\partial t}\right)^{n+1}=\frac{3 u^{n+1}-4 u^{n}+u^{n-1}}{2 \tau}+O\left(\tau^{2}\right) \tag{16}
\end{equation*}
$$

The following is the fourth-order backward Euler formula [19] for the time derivative in Equation (15):

$$
\begin{equation*}
\left(\frac{\partial u}{\partial t}\right)^{n+1}=\frac{25 u^{n+1}-48 u^{n}+36 u^{n-1}-16 u^{n-2}+3 u^{n-3}}{12 \tau}+O\left(\tau^{4}\right) \tag{17}
\end{equation*}
$$

Finally, substituting Equations (16) and (17) into Equation (15), neglecting the higher-order truncation error terms, and letting the mesh ratio be $r=\frac{\tau}{h^{2}}$, the present HOC scheme can be rewritten as follows:

$$
\begin{gather*}
\left(-24-\frac{17}{2 \varepsilon r}\right) u_{0}^{n+1}+\left(2-\frac{3}{4 \varepsilon r}\right) \sum_{l=1}^{6} u_{l}^{n+1}+\sum_{l=7}^{18} u_{l}^{n+1}=-\frac{18}{\varepsilon r} u_{0}^{n}-\frac{1}{\varepsilon r} \sum_{l=1}^{6} u_{l}^{n}+\frac{33}{2 \varepsilon r} u_{0}^{n-1}+\frac{1}{4 \varepsilon r} \sum_{l=1}^{6} u_{l}^{n-1} \\
-\frac{8}{\varepsilon r} u_{l}^{n-2}+\frac{3}{2 \varepsilon r} u_{l}^{n-3}+\frac{h^{2}}{2 \varepsilon}\left(\sum_{l=1}^{6} f_{l}^{n+1}+6 f_{0}^{n+1}\right) \tag{18}
\end{gather*}
$$

Equation (18) is the new HOC scheme we developed. It is easy to see from the derivation process that the truncation error of this scheme is $O\left(\tau^{4}+\tau^{2} h^{2}+h^{4}\right)$, which means that the present HOC scheme has fourth-order accuracy in both time and space. It should be noted that only the spatial dimensions are compact in this scheme while the temporal dimension is not compact, since it involves five time levels. According to the conclusion in [19], the backward Euler formula is unconditionally stable when the accuracy is not higher than the sixth-order; therefore, we used the second-, third- and fourth-order backward Euler formulas in this paper; the HOC scheme is unconditionally stable.

### 2.2. HOC Schemes for the Start-Up Layers

We note that the scheme involves five time layers. In addition to knowing the initial value of the moment, it is necessary to obtain the values of the first, second and third time layers before we can move forward time. To ensure the accuracy of the schemes, we consider the value of Equation (18) at
the moment $n+1 / 2$ using the Crank-Nicolson method and Equations (16) and (17) are replaced by the following equation:

$$
\begin{equation*}
\left(\frac{\partial u}{\partial t}\right)^{n+\frac{1}{2}}=\frac{u^{n+1}-u^{n}}{\tau}+O\left(\tau^{2}\right) \tag{19}
\end{equation*}
$$

We simplify this equation to obtain

$$
\begin{align*}
\left(12-\frac{4}{\varepsilon r}\right) u_{0}^{n+1}+\left(2-\frac{1}{2 \varepsilon r}\right) & \sum_{l=1}^{6} u_{l}^{n+1}+\frac{1}{2} \sum_{l=7}^{18} u_{l}^{n+1}=\left(-12-\frac{4}{\varepsilon r}\right) u_{0}^{n}-\left(2+\frac{1}{2 \varepsilon r}\right) \sum_{l=1}^{6} u_{l}^{n}-\frac{1}{2} \sum_{l=7}^{18} u_{l}^{n}  \tag{20}\\
& -\frac{1}{4 r}\left[\sum_{l=1}^{4} f_{l}^{n}+\sum_{l=1}^{4} f_{l}^{n-1}+8\left(f_{0}^{n}+f_{0}^{n-1}\right)\right] .
\end{align*}
$$

As seen from Equation (20), at time $n=0$, the value of the first time layer can be solved; in addition, at time $n=1$, the value of the second time level can be solved. The truncation error of this scheme is $O\left(\tau^{2}+\tau^{2} h^{2}+h^{4}\right)$.

To obtain the value of the third time layer, for the mixed derivative term containing the time, the second-order backward Euler Formula (16) is adopted for discretization, and the third-order backward Euler formula can be used for the time derivative. Thus, Equation (17) is replaced by the following Equation [19]:

$$
\begin{equation*}
\left(\frac{\partial u}{\partial t}\right)^{n+1}=\frac{11 u^{n+1}-18 u^{n}+9 u^{n-1}-2 u^{n-2}}{6 \tau}+O\left(\tau^{3}\right) \tag{21}
\end{equation*}
$$

Letting $n=2$, Equation (20) can be written as

$$
\begin{gather*}
\left(-24-\frac{13}{2 \varepsilon r}\right) u_{0}^{3}+\left(2-\frac{3}{4 \varepsilon r}\right) \sum_{l=1}^{6} u_{l}^{3}+\sum_{l=7}^{18} u_{l}^{3}=-\frac{12}{\varepsilon r} u_{0}^{2}-\frac{1}{\varepsilon r} \sum_{l=1}^{6} u_{l}^{2}+\frac{15}{2 \varepsilon r} u_{0}^{1}+\frac{1}{4 \varepsilon r} \sum_{l=1}^{6} u_{l}^{1}  \tag{22}\\
-\frac{2}{\varepsilon r} u_{l}^{0}-\frac{h^{2}}{2 \varepsilon}\left(\sum_{l=1}^{6} f_{l}^{3}+6 f_{0}^{3}\right) .
\end{gather*}
$$

The truncation error of this scheme is $O\left(\tau^{3}+\tau^{2} h^{2}+h^{4}\right)$.
According to the above analysis, the HOC scheme is a fourth-order five-layer fully implicit scheme, and the scheme is unconditionally stable. That is, the HOC scheme in this paper can achieve fourth-order accuracy in both the spatial and temporal dimensions. For the HOC scheme, if we set $\tau=O(h)$, the first time and the second time layers keep an order of magnitude of $h^{2}$; the third time layer keeps an order of magnitude of $h^{3}$; and after the fourth time layer, the temporal dimension can achieve fourth-order accuracy. Therefore, the first three start-up layers have lower-order accuracy than the fourth-order accuracy. According to [19], the use of a second-order accuracy scheme to initialize a fourth-order accuracy scheme appears not to affect the overall order of accuracy because the cumulative error phase is relatively small due to the small number of advanced layers. The numerical experiments in Section 4 will also fully illustrate this point.

## 3. Multigrid Method

Various difference schemes are known to give rise to a number of linear algebraic equations, and thus, certain iterative methods are commonly used to solve them. However, the traditional iterative methods have been found to be inefficient due to their slow convergence speeds. For this reason, the multigrid method has been adopted as the primary solution to these problems. The convection diffusion equations [20,21] and Poisson equation [22] are just a couple of examples of the many that have been solved using this method. Wang and Ge [23] developed a fourth-order compact difference scheme and employed the multigrid method to solve the general 2D elliptic equations.

In general, the multigrid method can be regarded as one of the better nontraditional iterative methods used to address a wide range of partial differential equations [24], which is achieved by going through several specific loops and recursive algorithms, including the V-cycle, W-cycle, full multigrid

V-cycle (FMV) and so on. The relaxation operator, restriction operator and interpolation operator are the three essential elements for establishing the multigrid method. The main characteristic of the multigrid method is that it uses different meshes. First, this method conducts "polishing" on the fine mesh level. Then, this method transfers the residual to the coarse mesh using the restriction operator, solves the residual equation on the coarse mesh, and finally returns the error correction result to the fine mesh level through the interpolation operator. According to the characteristics of the scheme in this paper, the full weighting restriction operator in [24] is used.

$$
\begin{gather*}
\left(R_{k}^{k-1} r^{k}\right)_{i+1, j+1}=\frac{1}{4} r_{2 i+1,2 j+1}^{k}+\frac{1}{8}\left(r_{2 i+2,2 j+1}^{k}+r_{2 i+1,2 j+2}^{k}+r_{2 i, 2 j+1}^{k}+r_{2 i+1,2 j}^{k}\right)  \tag{23}\\
+\frac{1}{16}\left(r_{2 i+2,2 j+2}^{k}+r_{2 i, 2 j+2}^{k}+r_{2 i+2,2 j}^{k}+r_{2 i, 2 j}^{k}\right) .
\end{gather*}
$$

Then, we remove the four corner points in (23) to increase the impact of the center point to twice that of the original. It can be expressed as follows:

$$
\begin{equation*}
\left(R_{k}^{k-1} u^{k}\right)_{i+1, j+1}=\frac{1}{2} u_{2 i+1,2 j+1}^{k}+\frac{1}{8}\left(u_{2 i+2,2 j+1}^{k}+u_{2 i+1,2 j+2}^{k}+u_{2 i, 2 j+1}^{k}+u_{2 i+1,2 j}^{k}\right) . \tag{24}
\end{equation*}
$$

The bilinear interpolation operators also in [24] are as follows:

$$
\begin{gather*}
\left(P_{k-1}^{k} r^{k-1}\right)_{2 i+1,2 j+1}=r_{i+1, j+1}^{k-1}  \tag{25}\\
\left(P_{k-1}^{k} r^{k-1}\right)_{2 i+2,2 j+1}=\frac{1}{2}\left(r_{i+1, j+1}^{k-1}+r_{i+2, j+1}^{k-1}\right),  \tag{26}\\
\left(P_{k-1}^{k} r^{k-1}\right)_{2 i+1,2 j+2}=\frac{1}{2}\left(r_{i+1, j+1}^{k-1}+r_{i+1, j+2}^{k-1}\right),  \tag{27}\\
\left(P_{k-1}^{k} r^{k-1}\right)_{2 i+1,2 j+2}=\frac{1}{4}\left(r_{i+1, j+1}^{k-1}+r_{i+2, j+1}^{k-1}+r_{i+1, j+2}^{k-1}+r_{i+2, j+2}^{k-1}\right), \tag{28}
\end{gather*}
$$

$R_{k}^{k-1}$ represents the restriction operator from level $k$ to level $k-1$,
$P_{k-1}^{k}$ represents the interpolation operator from level $k-1$ to level $k$,
$r^{k}$ represents the error residuals for the fine grid level, and
$r^{k-1}$ represents the error residuals for the adjacent coarse grid level.
Then, as far as the relaxation operator is concerned, we use the alternating direction line Gauss-Seidel relaxations to remove the residuals on each coarse grid.

## 4. Numerical Experiments

The results of numerical experiments on the 3D unsteady semi-linear diffusion reaction equation problems are given in this section to demonstrate the performances of the present HOC schemes and the multigrid method. All of the model problems with Dirichlet boundary conditions have the exact solutions. At each time layer, when the L2-norm of the residual on the finest grids is lower than a tolerance $\sigma$, it is assumed that the calculation is convergent at this time layer and then proceeds to the calculation of the next time layer. The multigrid V cycles (Num), the CPU time in seconds, the maximum absolute errors (Error) and the convergence order (Order) for the different numbers of grids are given in the tables for all problems. The code is written in the Fortran 77 programming language using double precision arithmetic. The configuration of the computer is an $\operatorname{Intel}(\mathrm{R})$ core(TM) i7-7600 CPU with 4 GB of memory. The error, ratio of errors with two different grid sizes and the convergence order are computed using the following formulas:

$$
\begin{equation*}
\text { Error }=\underset{0 \leq i, j, k \leq N}{\operatorname{Max}}\left|u\left(x_{i}, y_{j}, z_{k}, t_{n}\right)-u_{i, j, k}^{n}\right|, \tag{29}
\end{equation*}
$$

$$
\begin{gather*}
\text { Ratio }=\frac{\operatorname{Error}\left(N_{1}\right)}{\operatorname{Error}\left(N_{2}\right)}  \tag{30}\\
\operatorname{Order}=\frac{\log \left[\operatorname{Error}\left(N_{1}\right) / \operatorname{Error}\left(N_{2}\right)\right]}{\log \left(N_{2} / N_{1}\right)} \tag{31}
\end{gather*}
$$

where $N_{i, j, k}^{n}$ is the numerical solution, and $u\left(x_{i}, y_{j}, z_{k}, t_{n}\right)$ is the exact solution. $N_{1}$ and $N_{2}$ represent different numbers of coarse and fine grids, and $\operatorname{Error}\left(N_{1}\right)$ and $\operatorname{Error}\left(N_{2}\right)$ are the maximum absolute errors estimated for the two numbers of grids of coarse grid $N_{1}$ and adjacent fine grid $N_{2}$, respectively.

### 4.1. Problem 1

Consider the following 3D reaction diffusion problem:

$$
\begin{gather*}
\frac{\partial u}{\partial t}=\frac{1}{2}\left(\frac{\partial^{2} u}{\partial x^{2}}+\frac{\partial^{2} u}{\partial y^{2}}+\frac{\partial^{2} u}{\partial z^{2}}\right)+f(x, y, z, t, u), \quad 0 \leq x, y, z \leq 1, t \geq 0  \tag{32}\\
f(x, y, z, t, u)=3 e^{-u}+7 e^{-2 u} \tag{33}
\end{gather*}
$$

where its exact solution is

$$
\begin{equation*}
u(x, y, z, t)=\ln (1+x+2 y+3 z+3 t) \tag{34}
\end{equation*}
$$

The maximum absolute errors, the convergence order and the CPU time are calculated for $\tau=h, T=1$ and $\tau=0.5 h, T=0.5$ when the tolerance of residual is set to $\sigma=10^{-12}$ for the HOC scheme and the Crank-Nicolson scheme, and the results are given in Tables 1 and 2, respectively. The numerical results show that the Crank-Nicolson scheme is convergent with only the second-order accuracy for the spatial variables, but the proposed HOC scheme can achieve fourth-order accuracy in the spatial dimension. The HOC scheme yields more accurate solutions than those of the Crank-Nicolson scheme. Furthermore, we can also see the present HOC difference scheme is more cost-effective than the Crank-Nicolson scheme for computing an approximate solution with a given accuracy; e.g., the maximum absolute error computed from the present HOC difference with $N=20$ is $1.22 \times 10^{-7}$ and cost CPU time 44.89 s , which is more accurate than the result $2.09 \times 10^{-7}$ computed using the Crank-Nicolson scheme with $N=40$ and the CPU cost is 954.30 s . This result shows the superiority of the present high-order methods.

Table 1. Maximum absolute error (Error), convergence order (Order) and CPU time at $\tau=h, T=1$ for Example 1.

| $\boldsymbol{N}$ | Crank-Nicolson Scheme |  |  |  | Present HOC Scheme |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Error | Ratio | Order | CPU(s) | Error | Ratio | Order | CPU(s) |
| 10 | $1.72(-05)$ | - | - | 2.184 | $3.48(-06)$ | - | - | 0.747 |
| 20 | $2.25(-06)$ | 7.6444 | 2.93 | 24.04 | $1.22(-07)$ | 28.5246 | 4.83 | 44.89 |
| 40 | $2.09(-07)$ | 10.7656 | 3.43 | 954.30 | $6.85(-09)$ | 17.8120 | 4.15 | 1814.57 |
| 80 | $3.44(-08)$ | 6.0856 | 2.60 | 68943.48 | $3.12(-10)$ | 21.9551 | 4.46 | 74856.58 |

Table 2. Maximum absolute error (Error), convergence order (Order) and CPU time at $\tau=0.5 h, T=0.5$ for Example 1.

| $\boldsymbol{N}$ | Crank-Nicolson Scheme |  |  |  | Present HOC Scheme |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Error | Ratio | Order | CPU(s) | Error | Ratio | Order | CPU(s) |
| 10 | $4.05(-05)$ | - | - | 2.380 | $9.93(-07)$ | - | - | 0.620 |
| 20 | $1.04(-05)$ | 3.8942 | 1.96 | 21.64 | $4.75(-08)$ | 20.9053 | 4.39 | 29.36 |
| 40 | $2.62(-06)$ | 3.9695 | 1.99 | 636.07 | $2.86(-09)$ | 16.6084 | 4.05 | 951.87 |
| 80 | $6.57(-07)$ | 3.9878 | 2.00 | 27192.28 | $1.37(-10)$ | 20.8759 | 4.38 | 28255.23 |

### 4.2. Problem 2

We consider the following 3D reaction diffusion problem:

$$
\begin{gather*}
\frac{\partial u}{\partial t}=\frac{\partial^{2} u}{\partial x^{2}}+\frac{\partial^{2} u}{\partial y^{2}}+\frac{\partial^{2} u}{\partial z^{2}}+f(x, y, z, t, u), 0 \leq x, y, z \leq 1, t \geq 0  \tag{35}\\
f(x, y, z, t, u)=-0.5 u^{3}-\frac{9}{4} u^{5} \tag{36}
\end{gather*}
$$

where its exact solution is

$$
\begin{equation*}
u(x, y, z, t)=\frac{1}{\sqrt{1+x+y+z+t}} \tag{37}
\end{equation*}
$$

For comparison purposes, we use the HOC scheme and the Crank-Nicolson scheme to compute the numerical solutions of Problem 2, and set the tolerance of residual to be $\sigma=10^{-13}$.

In Tables 3 and 4, maximum absolute error, convergence order, and CPU time are given with $\tau=h, T=1$ and $\tau=1 / 80, T=0.5$, respectively. It can be seen that the HOC scheme achieves fourth-order accuracy while the Crank-Nicolson scheme achieves only second-order accuracy with different grids. Moreover, this result shows that the maximum absolute error of the HOC scheme is $1-2$ orders of magnitude lower than that of the Crank-Nicolson scheme. The advantages of the HOC scheme are more obvious when the number of grids increases. This result also shows that the present high-order scheme is more accurate.

Table 3. Maximum absolute error (Error), convergence order (Order) and CPU time at $\tau=h, T=1$ for Example 2.

| $N$ | Crank-Nicolson Scheme |  |  |  | Present HOC Scheme |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Error | Ratio | Order | CPU(s) | Error | Ratio | Order | CPU(s) |
| 10 | $3.49(-06)$ | - | - | 4.55 | $1.38(-07)$ | - | - | 0.70 |
| 20 | $7.79(-07)$ | 4.4801 | 2.16 | 70.42 | $4.49(-09)$ | 30.7350 | 4.94 | 66.60 |
| 40 | $1.97(-07)$ | 3.9543 | 1.98 | 2061.96 | $2.71(-10)$ | 16.5683 | 4.05 | 2108.23 |
| 80 | $4.92(-08)$ | 4.0040 | 2.00 | 82056.84 | $1.74(-11)$ | 15.5747 | 3.96 | 83483.57 |

Table 4. Maximum absolute error (Error), convergence order (Order) and CPU time at $\tau=1 / 80, T=0.5$ for Example 2.

| $\boldsymbol{N}$ | Crank-Nicolson Scheme |  |  |  | Present HOC Scheme |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Error | Ratio | Order | CPU(s) | Error | Ratio | Order | CPU(s) |
| 10 | $9.67(-06)$ | - | - | 5.244 | $5.48(-08)$ | - | - | 1.82 |
| 20 | $2.45(-06)$ | 3.9469 | 1.98 | 42.01 | $3.22(-09)$ | 17.0186 | 4.09 | 67.26 |
| 40 | $5.89(-07)$ | 4.1596 | 2.06 | 1211.30 | $1.20(-10)$ | 26.8333 | 4.75 | 2062.45 |
| 80 | $1.21(-07)$ | 4.8678 | 2.28 | 40177.75 | $6.87(-12)$ | 17.4672 | 4.13 | 49564.22 |

### 4.3. Problem 3

Consider the following 3D reaction diffusion problem:

$$
\begin{gather*}
\frac{\partial u}{\partial t}=\frac{\partial^{2} u}{\partial x^{2}}+\frac{\partial^{2} u}{\partial y^{2}}+\frac{\partial^{2} u}{\partial z^{2}}+f(x, y, z, t, u), 0 \leq x, y, z \leq 1, t \geq 0  \tag{38}\\
f(x, y, z, t, u)=\cos (u)-\cos \left(e^{-3 t} \sin (x) \sin (y) \sin (z)\right) \tag{39}
\end{gather*}
$$

where its exact solution is

$$
\begin{equation*}
u(x, y, z, t)=e^{-3 t} \sin (x) \sin (y) \sin (z) \tag{40}
\end{equation*}
$$

Problem 3 is the nonlinear unsteady reaction diffusion equation. We use the proposed HOC scheme and the scheme proposed in [6] to compute the numerical solution of Problem 3. The maximum absolute errors with different values of $h$ and $\tau$ at $T=1$ are listed in Table 5, and we set the tolerance of residual to be $\sigma=10^{-13}$. We can clearly see that when $h \geq 1 / 16$, the maximum absolute errors of the present scheme and Gu et al.'s scheme [6] are almost identical. With the increase of the number of grids, the present scheme produces much better solutions. In Table 6, speed-up efficiency of the multigrid method is demonstrated. For comparisons, the traditional iterative method is also used to calculate the solution of this problem. All computed results are obtained with $\tau=h$ at $T=1$. We can see that both the number of iterations and CPU time of the multigrid method are less than that of the traditional iterative method. These results show the superiority of the multigrid method.

Table 5. Maximum absolute error (Error) and convergence order (Order) at $T=1$ for Example 3.

| $(\boldsymbol{h}, \boldsymbol{\tau})$ | Scheme in Reference [6] |  | Present HOC Scheme |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Error | Ratio | Order | Error | Ratio | Order |
| $\left(\frac{1}{4}, \frac{1}{5}\right)$ | $6.28(-05)$ | - | - | $1.98(-04)$ | - | - |
| $\left(\frac{1}{8}, \frac{1}{10}\right)$ | $4.04(-06)$ | 15.5286 | 3.96 | $5.79(-06)$ | 34.1969 | 5.10 |
| $\left(\frac{1}{16}, \frac{1}{20}\right)$ | $2.60(-07)$ | 15.5246 | 3.96 | $1.53(-07)$ | 37.8431 | 5.24 |
| $\left(\frac{1}{32}, \frac{1}{40}\right)$ | $1.71(-08)$ | 15.2120 | 3.93 | $8.54(-09)$ | 17.6211 | 4.16 |
| $\left(\frac{1}{64}, \frac{1}{80}\right)$ | $1.35(-09)$ | 12.7118 | 3.67 | $5.36(-10)$ | 15.9328 | 3.99 |

Table 6. Speed-up efficiency of multigrid method compared to traditional method with $\tau=h, T=1$ for Example 3.

| $\boldsymbol{N}$ | Multigrid Method |  |  | Traditional Method |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Num | Error | CPU(s) | Num | Error | CPU(s) |
| 8 | 224 | $6.07(-06)$ | 0.18 | 552 | $6.07(-06)$ | 0.546 |
| 16 | 930 | $4.45(-07)$ | 12.98 | 2979 | $4.45(-07)$ | 22.19 |
| 32 | 3789 | $2.21(-08)$ | 400.52 | 14278 | $2.21(-08)$ | 1074.61 |
| 64 | 17118 | $1.31(-09)$ | 55198 | 45879 | $1.31(-09)$ | 100198 |

## 5. Conclusions

In this paper, a fully implicit five-layer high-order compact difference scheme is proposed for solving the 3D unsteady reaction diffusion equations. The truncation error of the main scheme is $O\left(\tau^{4}+\tau^{2} h^{2}+h^{2}\right)$, which means that the overall accuracy of the HOC scheme achieves the fourth-order when $\tau=\mathrm{O}(h)$. Although the accuracy order of the first three start-up layers is lower than the fourth, we numerically validate that it does not affect the overall accuracy of the present method. We also emphasize that the temporal accuracy of the present method in this paper is the fourth-order, which allows us to use a bigger time step length than the second-order schemes in the literature (see [3,7,10,12,14]). Consequently, the computational cost would be decreased because less time layers are needed for the present temporal high-order scheme than those temporal low-order schemes. The computational efficiency is improved when the multigrid acceleration technique is adopted in this paper. Numerical experiments are conducted to demonstrate the accuracy of the present scheme and the efficiency of the multigrid method. It is shown that the present HOC scheme is more accurate than those in the literature and the efficiency of the multigrid method is higher than the traditional iterative method. In addition, the HOC method can be directly extended to the numerical solutions of unsteady convection diffusion reaction equations, and unsteady incompressible Navier-Stokes equations, such as the work of Méndez et al. work in [25], where the spectral element method is used.

Author Contributions: Methodology, L.W.; writing—review and editing, L.W.; conceptualization, X.F.; data curation, X.F.; investigation, L.W.; formal analysis, X.F.

Funding: Thanks to the National Natural Science Foundation of China (11772165 and 11701306). Thanks to the Foundation of Ningxia Normal University (NXSFZDA1803)

Acknowledgments: The authors are deeply indebted to the anonymous referees for their very careful reading and valuable comments and suggestions which immensely improved the previous version of our manuscript.

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

## Nomenclature

| $a, b$ | constants |  | the second- and fourth-order spatial derivatives |
| :---: | :---: | :---: | :---: |
| Error | maximum absolute error | $\begin{aligned} & \frac{\partial^{2} u}{\partial x^{2}}, \frac{\partial^{2} u}{\partial y^{2}}, \frac{\partial^{2} u}{\partial z^{2}} \\ & \frac{\partial^{4} u}{\partial x^{4}}, \frac{\partial^{4} u}{\partial y^{4}}, \frac{\partial^{4} u}{\partial z^{4}} \end{aligned}$ |  |
| Num | numbers of the multigrid V cycles |  | in the $x$-, $y$-, z-directions, respectively |
| CPU(s) | the CPU time in seconds |  |  |
| $f$ | forcing function | $\frac{\partial^{4} u}{\partial x^{2} \partial y^{2}}, \frac{\partial^{4} u}{\partial x^{2} \partial z^{2}}, \frac{\partial^{4} u}{\partial y^{2} \partial z^{2}}$ | the fourth-order spatial mixed derivatives |
| $f_{x x}, f_{y y}, f_{z z}$, | spatial second derivative of $f$ | $\overline{\partial x^{2} \partial y^{2}}, \overline{\partial x^{2} \partial z^{2}}, \overline{\partial y^{2} \partial z^{2}}$ | the fourth-order spatial mixed derivatives |
| $h$ | spatial grid step length |  |  |
| $i, j, k$ | index of grid point in $x-y-, z$-directions, respectively | $\frac{\partial^{3} u}{\partial t \partial x^{2}}, \frac{\partial^{3} u}{\partial t \partial y^{2}}$ | the second-order spatial derivatives of $\frac{\partial u}{\partial t}$ |
| $n$ | time level |  |  |
| $N, M$ | numbers of grids for space and time | $\frac{\partial u}{\partial t}, \partial^{2} u$ | the first- and the second-order temporal |
| $N_{1}, N_{2}$ | numbers of grids of coarse and fine meshes, respectively | $\frac{\partial t}{\partial t}, \frac{\partial t^{2}}{}$ | derivatives of $u$ |
| order | convergence order |  |  |
| $p_{k-1}^{k}$ | interpolation operator from level $k-1$ to level $k$ | $x, y, z$ | spatial coordinate in the $x$-, $y$-, z-directions |
| $r$ | mesh ratio | $x_{i}, y_{j}, z_{k}$ | spatial discrete points |
| Ratio | the error ratio of different grids | Greek symbols |  |
| $r^{k-1}$ | residuals for coarse grid levels |  | the first- and the second-order central different |
| $r^{k}$ | residuals for fine grid levels | $\delta_{x}, \delta_{x}^{2}$ | operators |
| $R_{k}^{k-1}$ | restriction operator from level $k$ to $k-1$ | $\varepsilon$ | positive constant diffusion term coefficient |
| Max | maximum value function | $\varepsilon$ | positive constant diffusion term coefficient |
| $t$ | time coordinate | $\phi$ | boundary conditions |
| T | total time | $\varphi$ | initial conditions |
| $u$ | solution of Equation (18) | $\tau$ | temporal grid step length |
| $u\left(x_{i}, y_{j}, z_{k},\right)$ | exact solution at grid point $\left(x_{i}, y_{j}, z_{k}\right.$, ) | $\Omega$ | computational domain |
| $u_{i, j, k}^{n}$ | numerical solution at grid point ( $x_{i}, y_{j}, z_{k}$, | $\partial \Omega$ | domain boundary |

## References

1. Ames, W.F. Nonlinear Partial Differential Equations in Engineering; Academic Press: New York, NY, USA, 1965.
2. Madzvamuse, A.; Maini, P.K. Velocity-induced numerical solutions of reaction-diffusion systems on continuously growing domains. J. Comput. Phys. 2007, 225, 100-119. [CrossRef]
3. Wang, Y.; Guo, B. A monotone compact implicit scheme for nonlinear reaction-diffusion equations. J. Comp. Math. 2008, 26, 123-148.
4. Araújo, A.; Barbeiro, S.; Serranho, P. Convergence of finite difference schemes for nonlinear complex reaction-diffusion processes. SIAM J. Numer. Anal. 2015, 53, 228-250. [CrossRef]
5. Zhu, J.; Zhang, Y.T.; Newman, S.A.; Alber, M. Application of discontinuous galerkin methods for reaction-diffusion systems in developmental biology. J. Sci. Comput. 2009, 40, 391-418. [CrossRef]
6. Fernandes, R.I.; Bialecki, B.; Fairweather, G. An ADI extrapolated Crank-Nicolson orthogonal spline collocation method for nonlinear reaction-diffusion systems on evolving domains. J. Comput. Phys. 2015, 299, 561-580. [CrossRef]
7. Gu, Y.; Liao, W.; Zhu, J. An efficient high-order algorithm for solving systems of 3-D reaction-diffusion equations. J. Comput. Appl. Math. 2003, 155, 1-17. [CrossRef]
8. Liao, W. A high-order ADI finite difference scheme for a 3D reaction-diffusion equation with neumann boundary condition. Numer. Methods Part. Differ. Equ. 2013, 29, 778-798. [CrossRef]
9. Wang, Y.M.; Wang, J. A higher-order compact ADI method with monotone iterative procedure for systems of reaction-diffusion equations. Comput. Math. Appl. 2011, 62, 2434-2451. [CrossRef]
10. Tian, Z. Cubic-spline difference schemes of high-order accuracy for diffusion reaction equation. J. Hubei Univ. 1997, 19, 111-115. (In Chinese)
11. Hasnain, S.; Saqib, M.; Mashat, D.S. Fourth order Douglas implicit scheme for solving three dimension reaction diffusion equation with non-linear source term. AIP Adv. 2017, 7, 075011. [CrossRef]
12. Ge, Y.; Cai, Z.; Sheng, Q. A compact adaptive approach for degenerate singular reaction-diffusion equations. Numer. Methods Part. Differ. Equ. 2018, 34, 1166-1187. [CrossRef]
13. Kaya, A. Finite difference approximations of multidimensional unsteady convection-diffusion-reaction equations. J. Comput. Phys. 2015, 285, 331-349. [CrossRef]
14. Zhu, X.; Rui, H. High-order compact difference scheme of 1D nonlinear degenerate convection-reaction-diffusion equation with adaptive algorithm. Numer. Heat Transf. Part B Fundam. 2019, 75, 43-46. [CrossRef]
15. Iagar, R.G.; Sánchez, A. Blow up profiles for a quasilinear reaction-diffusion equation with weighted reaction with linear growth. J. Dyn. Differ. Equ. 2019, 31, 2061-2094. [CrossRef]
16. Jiang, T.; Zhang, Y.T. Krylov single-step implicit integration factor WENO methods for advection-diffusion-reaction equations. J. Comput. Phys. 2016, 311, 22-44. [CrossRef]
17. Jha, N.; Singh, B. Exponential basis and exponential expanding grids third (fourth)-order compact schemes for nonlinear three-dimensional convection-diffusion-reaction equation. Adv. Differ. Equ. 2019, 2019, 339. [CrossRef]
18. Piquet, A.; Zebiri, B.; Hadjadj, A.; Shadloo, M.S. A parallel high-order compressible flows solver with a domain decomposition method in the generalized curvilinear coordinates system. Int. J. Numer. Methods Heat Fluid Flow 2019. [CrossRef]
19. Strikwerda, J.C. High order accurate schemes for incompressible viscous flow. Int. J. Numer. Meth. Fluids 1997, 24, 715-734. [CrossRef]
20. Gupta, M.M.; Zhang, J. High accuracy multigrid solution of the 3D convection-diffusion equation. Appl. Math. Comput. 2000, 113, 249-274. [CrossRef]
21. Ge, Y.; Cao, F. Multigrid method based on the transformation-free HOC scheme on nonuniform grids for 2D convection diffusion problems. J. Comput. Phys. 2011, 230, 4051-4070. [CrossRef]
22. Ge, Y.; Cao, F.; Zhang, J. A transformation-free HOC scheme and multigrid method for solving the 3D Poisson equation on nonuniform grids. J. Comput. Phys. 2013, 234, 199-216. [CrossRef]
23. Wang, Y.; Ge, Y. High-order compact difference scheme and multigrid method for solving the 2D elliptic problems. Math. Probl. Eng. 2018. [CrossRef]
24. Wesseling, P.W. An Introduction to Multigrid Methods; Wiley: Chichester, UK, 1992.
25. Méndez, M.; Shadloo, M.S.; Hadjadj, A.; Ducoin, A. Boundary layer transition over a concave surface caused by centrifugal instabilities. Comput. Fluids 2018, 171, 135-153. [CrossRef]
