JOURNAL OF COMPUTATIONAL PHYSICS **130,** 109–122 (1997)
ARTICLE NO. CP965564

Finite Difference Schemes for Three-dimensional Time-dependent Convection-Diffusion Equation Using Full Global Discretization

H. Y. Xu, M. D. Matovic, and A. Pollard[1]

Centre for Advanced Gas Combustion Technology, Department of Mechanical Engineering, Queen's University at Kingston, Ontario, Canada K7L-3N6

Received February 20, 1996; revised August 12, 1996

The three-dimensional, time-dependent convection-diffusion equation (CDE) is considered. An exponential transformation is used to collectively transform the CDE. The idea of global discretization is used, where attention is paid to the whole transformed CDE, but not to the individual spatial and temporal derivatives in the equation. Four finite difference schemes for both CDE and transformed CDE are established. The modified partial differential equations of these schemes are obtained, which indicate that the trunction errors of the schemes can be of second and fourth order, depending on the prescription of the time step length. Some characteristic physical parameters, i.e., local Reynolds number, local Strouhal number, and viscous diffusive length, are introduced into the schemes and the viscous diffusive length is found to be a significant parameter in relating temporal discretisation with spatial discretisation. A series of benchmark analytical solutions of Navier–Stokes and Burgers equations, as well as the numerical solutions using the well-known discretisation schemes, are used to investigate the properties of the derived schemes. The high-order schemes achieve higher resolutions over the conventional schemes without decreasing much the sparsity of the matrix structures. Grid refinement studies reveal that the inverse exponential transformation of the finite difference schemes tends to destroy some resolution of the schemes, especially for large local Reynolds number.

1. INTRODUCTION

One of the major issues is computational fluid dynamics (CFD) is the discretization of the Navier–Stokes (N-S) equations, which are sets of three-dimensional, time-dependent, convection-diffusion equations (CDE). This paper presents a new way to establish finite difference schemes for the CDE, that is, global discretization. The discretization of a partial differential equation (PDE) with conventional finite differencing [1] is well known. Typically, discretisation pays attention to individual derivative terms in the PDE, where the objective is to approximate the PDE by replacing it with a set of discretized equations that are created using some prescribed pattern. Here, the PDE is treated in totality and the integral truncation error of the discretization is then minimized. This idea is called global discretisation where attention is no longer paid to the individual spatial and temporal derivatives but to the whole equation. With this idea, four finite difference schemes are established for both the CDE and the transformed CDE. The modified PDEs, following Warming and Hyett [10], are obtained, which indicate that the schemes are of either second or fourth order if the time step is properly prescribed. A series of analytical solutions to both the Navier–Stokes and Burgers equations are chosen as benchmark cases against which the new method is assessed. The paper is organized as follows. Sections 2 and 3 demonstrate the major steps in establishing the finite difference schemes. Section 4 provides the formulation of the schemes for the CDE and the schemes for the transformed CDE are given in the Appendix. Section 5 presents the analytical solutions and the numerical algorithms chosen to benchmark the schemes. The numerical experiments are described in Section 6. The results are discussed in Section 7 and conclusions drawn from the work close out the paper.

2. CONVECTION-DIFFUSION EQUATION AND ITS TRANSFORMATION

Consider the general equation, i.e., the CDE:

$$\frac{\partial \phi}{\partial t} + a_x \frac{\partial \phi}{\partial x} + a_y \frac{\partial \phi}{\partial y} + a_z \frac{\partial \phi}{\partial z} = b\left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}\right) + s(t, x, y, z). \tag{1}$$

The coeffecients a_x, a_y, a_z, and b can be reasonably assumed locally constant. In most existing solution algorithms, e.g., SIMPLE [3], PISO [11], and fractional step method [12], the pressure derivatives are treated as a source term in the momentum equations. Here, it is assumed that the source term in Eq. (1) is a function of both space and time.

[1] To whom correspondence should be addressed.

0021-9991/97 $25.00

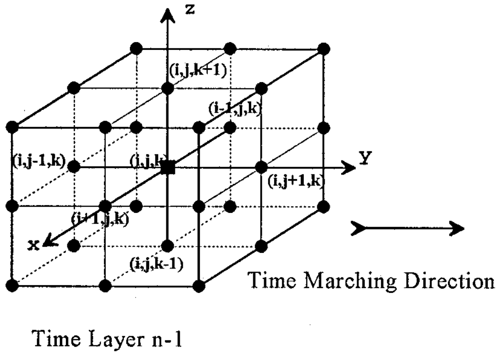

FIGURE 1

Introducing the exponential transformation

$$\phi = Te^{((a_x/2b)x+(a_y/2b)y+(a_z/2b)z)} \tag{2}$$

and applying the transformation to Eq. (1) leads to the transformed version of the convection-diffusion equation or CED$^{\mathrm{T}}$,

$$\frac{\partial T}{\partial t} = b\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right) + cT + s^T, \tag{3}$$

where

$$c = -\frac{a_x^2 + a_y^2 + a_z^2}{4b}, \quad s^T = se^{-((a_x/2b)x+(a_y/2b)y+(a_z/2b)z)}. \tag{4}$$

The exponential transformation eliminates the convection term in the CDE, thereby giving a conduction equation.

3. DISCRETISATION OF CDE$^{\mathrm{T}}$

Generally, the discretization of a PDE is to replace the PDE point-by-point with a discretized equation, which is basically a relation between the value at a central point (i, j, k) and those of its neighbours in both space and time, as shown in Fig. 1. The terms in (3) are discretised here using the method outlined in Yang *et al.* [13], wherein the nodal points are arranged to ensure that the odd derivatives cancel in the Taylor series expansions about the nodal points. Thus, dispersive errors are eliminated. The schemes considered here are the explicit, implicit, weakly implicit, and strongly implicit schemes, which depend on what time layers of the nodal points are employed. With reference to Fig. 1, the relationship between the value at node (i, j, k) and those at its neighbours can be assumed, together with their order; these are:

(a) second-order explicit scheme (SOES),

$$\begin{aligned} T^n_{ijk} &= c_0 + c_1 T^{n-1}_{ijk} + c_2(T^{n-1}_{i-1jk} + T^{n-1}_{i+1jk}) \\ &\quad + c_3(T^{n-1}_{ij-1k} + T^{n-1}_{ij+1k}) + c_4(T^{n-1}_{ijk-1} + T^{n-1}_{ijk+1}); \end{aligned} \tag{5}$$

(b) second-order implicit scheme (SOIS),

$$\begin{aligned} T^n_{ijk} &= c_0 + c_1(T^n_{i-1jk} + T^n_{i+1jk}) + c_2(T^n_{ij-1k} + T^n_{ij+1k}) \\ &\quad + c_3(T^n_{ijk-1} + T^n_{ijk+1}) + c_4 T^{n-1}_{ijk}; \end{aligned} \tag{6}$$

(c) fourth-order strongly implicit schemes (FOSIS),

$$\begin{aligned} T^n_{ijk} &= c_0 + c_1(T^n_{i-1jk} + T^n_{i+1jk}) + c_2(T^n_{ij-1k} + T^n_{ij+1k}) \\ &\quad + c_3(T^n_{ijk-1} + T^n_{ijk+1}) \\ &\quad + c_4(T^n_{i-1j-1k} + T^n_{i-1j+1k} + T^n_{i+1j-1k} + T^n_{i+1j+1k}) \\ &\quad + c_5(T^n_{ij-1k-1} + T^n_{ij-1k+1} + T^n_{ij+1k-1} + T^n_{ij+1k+1}) \\ &\quad + c_6(T^n_{i-1jk-1} + T^n_{i-1jk+1} + T^n_{i+1jk-1} + T^n_{i+1jk+1}) \\ &\quad + c_7 T^{n-1}_{ijk} + c_8(T^{n-1}_{i-1jk} + T^{n-1}_{i+1jk}) + c_9(T^{n-1}_{ij-1k} + T^{n-1}_{ij+1k}) \\ &\quad + c_{10}(T^{n-1}_{ijk-1} + T^{n-1}_{ijk+1}); \end{aligned} \tag{7}$$

(d) fourth-order weakly implicit schemes (FOWIS),

$$\begin{aligned} T^n_{ijk} &= c_0 + c_1(T^n_{i-1jk} + T^n_{i+1jk}) + c_2(T^n_{ij-1k} + T^n_{ij+1k}) \\ &\quad + c_3(T^n_{ijk-1} + T^n_{ijk+1}) \\ &\quad + c_4(T^{n-1}_{i-1j-1k} + T^{n-1}_{i-1j+1k} + T^{n-1}_{i+1j-1k} + T^{n-1}_{i+1j+1k}) \\ &\quad + c_5(T^{n-1}_{ij-1k-1} + T^{n-1}_{ij-1k+1} + T^{n-1}_{ij+1k-1} + T^{n-1}_{ij+1k+1}) \\ &\quad + c_6(T^{n-1}_{i-1jk-1} + T^{n-1}_{i-1jk+1} + T^{n-1}_{i+1jk-1} + T^{n-1}_{i+1jk+1}) \\ &\quad + c_7 T^{n-1}_{ijk} + c_8(T^{n-1}_{i-1jk} + T^{n-1}_{i+1jk}) + c_9(T^{n-1}_{ij-1k} + T^{n-1}_{ij+1k}) \\ &\quad + c_{10}(T^{n-1}_{ijk-1} + T^{n-1}_{ijk+1}). \end{aligned} \tag{8}$$

The subscripts i, j, and k represent the nodal points in directions x, y, and z, respectively, and the superscript stands for the temporal dimension. The coefficients, c_i, in the above schemes need to be determined. The derivation of the finite difference form of the CDE$^{\mathrm{T}}$ using the second-order implicit scheme (SOIS) is used to illustrate the major steps and ideas. The other schemes can be obtained similarly.

A Taylor series expansion about a nodal point when using the SOIS reads

$$\frac{1}{2}(T^n_{i-1jk} + T^n_{i+1jk}) = \left(T + \frac{1}{2!}h_x^2\frac{\partial^2 T}{\partial x^2} + \frac{1}{4!}h_x^4\frac{\partial^4 T}{\partial x^4} + \cdots\right)^n_{ijk} \quad (9)$$

$$\frac{1}{2}(T^n_{ij-1k} + T^n_{ij+1k}) = \left(T + \frac{1}{2!}h_y^2\frac{\partial^2 T}{\partial y^2} + \frac{1}{4!}h_y^4\frac{\partial^4 T}{\partial y^4} + \cdots\right)^n_{ijk} \quad (10)$$

$$\frac{1}{2}(T^n_{ijk-1} + T^n_{ijk+1}) = \left(T + \frac{1}{2!}h_z^2\frac{\partial^2 T}{\partial z^2} + \frac{1}{4!}h_z^4\frac{\partial^4 T}{\partial z^4} + \cdots\right)^n_{ijk} \quad (11)$$

$$T^{n-1}_{ijk} = \left[\sum_{p=0}^{\infty}\frac{1}{p!}(-\tau)^p\frac{\partial^p T}{\partial t^p}\right]^n_{ijk}, \quad (12)$$

where the spatial intervals of the grid points are denoted by h_x, h_y, and h_z in x, y, and z directions, respectively, and τ is the temporal step length.

Define the operator

$$\frac{\partial'}{\partial t'} = b\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) + c \quad (13)$$

so that Eq. (3) can be rewritten as

$$\frac{\partial T}{\partial t} = \frac{\partial' T}{\partial t'} + s^T, \quad (14)$$

and, by successively substituting Eq. (14) into Eq. (12), we have

$$\begin{aligned} T^{n-1}_{ijk} &= \left(\sum_{p=0}^{\infty}\frac{1}{p!}(-\tau)^p\frac{\partial'^p}{\partial t'^p}T\right)^n_{ijk} \\ &\quad + \left[\sum_{q=0}^{\infty}\frac{\partial'^q}{\partial t'^q}\left(\sum_{p=q+1}^{\infty}\frac{1}{p!}(-\tau)^p\frac{\partial^{p-(q+1)}}{\partial t^{p-1(q+1)}}s^T\right)\right]^n_{ijk}. \end{aligned} \quad (15)$$

By successively substituting Eq. (13) into Eq. (15), we obtain

$$\begin{aligned} T^{n-1}_{ijk} &= \left(\sum_{p=0}^{\infty}\frac{(-k_2)^p}{p!}\right)T^n_{ijk} \\ &\quad + \left(\sum_{p=1}^{\infty}\frac{(-k_2)^{p-1}(-k_1)}{(p-1)!}\right)\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right)^n_{ijk} \\ &\quad + \left(\sum_{p=2}^{\infty}\frac{(-k_2)^{p-2}(-k_1)^2}{(p-2)!2!}\right)\left(\frac{\partial^4 T}{\partial x^4} + \frac{\partial^4 T}{\partial y^4} + \frac{\partial^4 T}{\partial z^4}\right. \\ &\quad \left. + 2\frac{\partial^4 T}{\partial x^2\partial y^2} + 2\frac{\partial^4 T}{\partial y^2\partial z^2} + 2\frac{\partial^4 T}{\partial x^2\partial z^2}\right)^n_{ijk} \\ &\quad + \sum_{p=1}^{\infty}\frac{(-\tau)^p}{p!}\frac{\partial^{p-1}}{\partial t^{p-1}}s^T_{ijk} \\ &\quad + \sum_{q=1}^{\infty}\frac{\partial'^q}{\partial t'^q}\left(\sum_{p=q+1}^{\infty}\frac{1}{p!}(-\tau)^p\frac{\partial^{p-(q+1)}}{\partial t^{p-(q+1)}}s^T\right)^n_{ijk} + \cdots, \end{aligned} \quad (16)$$

where $k_1 = \tau b$ and $k_2 = \tau c$, the physical meanings of which will be explained in the following section.

It should be noted that only the series of $q = 0$ in Eq. (15) is needed if the time step is prescribed small enough.

The two series have the properties:

$$\begin{aligned} e^k &= \sum_{p=0}^{\infty}\frac{1}{p!}k^p; \\ \int_{t_n}^{t_{n-1}} f\,dt &= \left[\sum_{k=1}^{\infty}\frac{1}{k!}(-\tau)^k\frac{\partial^{k-1}}{\partial t^{k-1}}f\right]_{t_n}, \quad \tau = t_n - t_{n-1}. \end{aligned} \quad (17)$$

We can rewrite Eq. (16) as

$$\begin{aligned} T^{n-1}_{ijk} &= \int_{t_n}^{t_{n-1}} s^T_{ijk}\,dt + \sum_{q=1}^{\infty}\frac{\partial'^q}{\partial t'^q}\left(\sum_{p=q+1}^{\infty}\frac{1}{p!}(-\tau)^p\frac{\partial^{p-(q+1)}}{\partial t^{p-(q+1)}}s^T\right)^n_{ijk} \\ &\quad + e^{-k_2}\left[T + (-k_1)\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right)\right. \\ &\quad + \frac{(-k_1)^2}{2!}\left(\frac{\partial^4 T}{\partial x^4} + \frac{\partial^4 T}{\partial y^4} + \frac{\partial^4 T}{\partial z^4} + 2\frac{\partial^4 T}{\partial x^2\partial y^2}\right. \\ &\quad \left.\left. + 2\frac{\partial^4 T}{\partial y^2\partial z^2} + 2\frac{\partial^4 T}{\partial x^2\partial z^2}\right)\right]^n_{ijk} + ... \end{aligned} \quad (18)$$

Now, substituting Eq. (8), (9), (10), and (18) into (6) leads to

$$\begin{aligned} T^n_{ijk} &\equiv A_0 + A_1 T^n_{ijk} + A_2\left(\frac{\partial^2 T}{\partial x^2}\right)^n_{ijk} + A_3\left(\frac{\partial^2 T}{\partial y^2}\right)^n_{ijk} \\ &\quad + A_4\left(\frac{\partial^2 T}{\partial z^2}\right)^n_{ijk} + A_5\left(\frac{\partial^4 T}{\partial x^4}\right)^n_{ijk} + A_6\left(\frac{\partial^4 T}{\partial y^4}\right)^n_{ijk} \\ &\quad + A_7\left(\frac{\partial^4 T}{\partial z^4}\right)^n_{ijk} + A_8\left(\frac{\partial^4 T}{\partial x^2\partial y^2}\right)^n_{ijk} + A_9\left(\frac{\partial^4 T}{\partial y^2\partial z^2}\right)^n_{ijk} \\ &\quad + A_{10}\left(\frac{\partial^4 T}{\partial x^2\partial z^2}\right)^n_{ijk} \\ &\quad + \sum_{q=1}^{\infty}\frac{\partial'^q}{\partial t'^q}\left(\sum_{p=q+1}^{\infty}\frac{1}{p!}(-\tau)^p\frac{\partial^{p-(q+1)}}{\partial t^{p-(q+1)}}s^T\right)^n_{ijk} + ... \end{aligned} \quad (19)$$

For Eq. (6) to satisfy Eq. (19) at the maximum level, it is necessary that $A_0 = A_2 = A_3 = A_4 = 0$, and $A_1 = 1$, which leads to a linear system of equations,

$$\begin{pmatrix} 2 & 2 & 2 & e^{-k_2} \\ h_x^2 & 0 & 0 & -k_1 e^{-k_2} \\ 0 & h_y^2 & 0 & -k_1 e^{-k_2} \\ 0 & 0 & h_z^2 & -k_1 e^{-k_2} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix},$$

and an integral relation

$$c_0 = A_0 + \int_{t_{n-1}}^{t_n} c_4 s_{ijk}^T \, dt$$

which permits the coefficients in Eq. (6) to be determined.

Thus, the finite difference form of the CDE^T using SOIS is deduced. The formulation of the finite difference equations for CDE^T using the other schemes can be found in the Appendix. The truncation errors of the four schemes are evaluated by obtaining their corresponding modified PDEs, as outlined in [10]. Here, typically, the modified CDE^T, when employing SOIS, reads

$$\begin{aligned} &\frac{\partial T}{\partial t} - \left[b \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) + cT + s^T \right] \\ &\quad = b \left[k_1 \left(\frac{\partial^4 T}{\partial x^2 \partial y^2} + \frac{\partial^4 T}{\partial y^2 \partial z^2} + \frac{\partial^4 T}{\partial x^2 \partial z^2} \right) \right. \\ &\qquad + \left(\frac{h_x^2}{12} + \frac{k_1}{2} \right) \frac{\partial^4 T}{\partial x^4} + \left(\frac{h_y^2}{12} + \frac{k_1}{2} \right) \frac{\partial^4 T}{\partial y^4} \\ &\qquad \left. + \left(\frac{h_z^2}{12} + \frac{k_1}{2} \right) \frac{\partial^4 T}{\partial z^4} \right]_{ijk}^n \\ &\qquad + \sum_{q=1}^{\infty} \frac{\partial'^q}{\partial t'^q} \left(\sum_{p=q+1}^{\infty} \frac{1}{p!} (-\tau)^{p-1} \frac{\partial^{p-(q+1)}}{\partial t^{p-(q+1)}} s^T \right)_{ijk}^n + \cdots \end{aligned} \tag{20}$$

which indicates that the total truncation error of the scheme becomes second order if the time step τ is of the order of the spatial interval squared.

4. FINITE DIFFERENCE EQUATIONS OF CDE

The inverse exponential transformation

$$T(t, x, y, z) = \phi(t, x, y, z)\, e^{-((a_x/2b)x + (a_y/2b)y + (a_z/2b)z)}$$

can be applied to the CDE^T to obtain the finite difference approximation to the CDE. In the following, k_x, k_y, and k_z are defined as

$$k_x = \frac{a_x}{2b} h_x, \quad k_y = \frac{a_y}{2b} h_y, \quad k_z = \frac{a_z}{2b} h_z:$$

(a) second-order explicit scheme (SOES),

$$\begin{aligned} \phi_{ijk}^n &= c_c^{n-1} \phi_{ijk}^{n-1} + c_{wc}^{n-1} \phi_{i-1jk}^{n-1} + c_{ec}^{n-1} \phi_{i+1jk}^{n-1} + c_{sc}^{n-1} \phi_{ij-1k}^{n-1} \\ &\quad + c_{nc}^{n-1} \phi_{ij+1k}^{n-1} + c_{bc}^{n-1} \phi_{ijk-1}^{n-1} + c_{tc}^{n-1} \phi_{ijk+1}^{n-1} + S_{ijk} \end{aligned}$$

$$\begin{aligned} d_0 &= \frac{1}{h_x^2 h_y^2 h_z^2}, \\ c_c^{n-1} &= [h_x^2 h_y^2 h_z^2 - 2k_1 (h_x^2 h_y^2 + h_y^2 h_z^2 + h_x^2 h_z^2)] e^{k_2} d_0 \\ d_x^{n-1} &= k_1 h_y^2 h_z^2 e^{k_2} d_0, \quad d_y^{n-1} = k_1 h_x^2 h_z^2 e^{k_2} d_0, \\ d_z^{n-1} &= k_1 h_x^2 h_y^2 e^{k_2} d_0 \\ c_{wc}^{n-1} &= d_x^{n-1} e^{k_x}, \quad c_{ec}^{n-1} = d_x^{n-1} e^{-k_x}, \quad c_{sc}^{n-1} = d_y^{n-1} e^{k_y}, \\ c_{nc}^{n-1} &= d_y^{n-1} e^{-k_y}, \quad c_{bc}^{n-1} = d_z^{n-1} e^{k_z}, \quad c_{tc}^{n-1} = d_z^{n-1} e^{-k_z} \\ S_{ijk} &= \int_{t_{n-1}}^{t_n} (c_c^{n-1} s_{ijk} + c_{wc}^{n-1} s_{i-1jk} + c_{ec}^{n-1} s_{i+1jk} + c_{sc}^{n-1} s_{ij-1k} \\ &\quad + c_{nc}^{n-1} s_{ij+1k} + c_{bc}^{n-1} s_{ijk-1} + c_{tc}^{n-1} s_{ijk+1}) \, dt; \end{aligned}$$

(b) second-order implicit scheme (SOIS),

$$\begin{aligned} \phi_{ijk}^n &= c_{wc}^n \phi_{i-1jk}^n + c_{ec}^n \phi_{i+1jk}^n + c_{sc}^n \phi_{ij-1k}^n + c_{nc}^n \phi_{ij+1k}^n \\ &\quad + c_{bc}^n \phi_{ijk-1}^n + c_{tc}^n \phi_{ijk+1}^n + c_c^{n-1} \phi_{ijk}^{n-1} + S_{ijk} \end{aligned}$$

$$\begin{aligned} d_0 &= \frac{1}{h_x^2 h_y^2 h_z^2 + 2k_1 (h_x^2 h_y^2 + h_y^2 h_z^2 + h_x^2 h_z^2)}, \\ c_c^{n-1} &= h_x^2 h_y^2 h_z^2 e^{k_2} d_0 \\ S_{ijk} &= \int_{t_{n-1}}^{t_n} c_c^{n-1} s_{ijk} \, dt, \quad d_x^n = k_1 h_y^2 h_z^2 d_0, \\ d_y^n &= k_1 h_x^2 h_z^2 d_0, \quad d_x^n = k_1 h_x^2 h_y^2 d_0 \\ c_{wc}^n &= d_x^n e^{k_x}, \quad c_{ec}^n = d_x^n e^{-k_x}, \quad c_{sc}^n = d_y^n e^{k_y}, \\ c_{nc}^n &= d_y^n e^{-k_y}, \quad c_{bc}^n = d_z^n e^{k_x}, \quad c_{tc}^n = d_z^n e^{-k_z}; \end{aligned}$$

(c) fourth-order weakly implicit scheme (FOWIS),

$$\begin{aligned} \phi_{ijk}^n &= c_{wc}^n \phi_{i-1jk}^n + c_{ec}^n \phi_{i+1jk}^n + c_{sc}^n \phi_{ij-1k}^n + c_{nc}^n \phi_{ij+1k}^n \\ &\quad + c_{bc}^n \phi_{ijk-1}^n + c_{tc}^n \phi_{ijk+1}^n + c_{wsc}^{n-1} \phi_{i-1j-1k}^{n-1} + c_{wnc}^{n-1} \phi_{i-1j+1k}^{n-1} \end{aligned}$$

$$
\begin{aligned}
&+ c_{esc}^{n-1}\phi_{i+1j-1k}^{n-1} + c_{enc}^{n-1}\phi_{i+1j+1k}^{n-1} + c_{csb}^{n-1}\phi_{ij-1k-1}^{n-1}\\
&+ c_{cst}^{n-1}\phi_{ij-1k+1}^{n-1} + c_{cnb}^{n-1}\phi_{ij+1k-1}^{n-1} + c_{cnt}^{n-1}\phi_{ij+1k+1}^{n-1}\\
&+ c_{wcb}^{n-1}\phi_{i-1jk-1}^{n-1} + c_{wct}^{n-1}\phi_{i-1jk+1}^{n-1} + c_{ecb}^{n-1}\phi_{i+1jk-1}^{n-1}\\
&+ c_{ect}^{n-1}\phi_{i+1jk+1}^{n-1} + c_{c}^{n-1}\phi_{ijk}^{n-1} + c_{wc}^{n-1}\phi_{i-1jk}^{n-1}\\
&+ c_{ec}^{n-1}\phi_{i+1jk}^{n-1} + c_{sc}^{n-1}\phi_{ij-1k}^{n-1} + c_{nc}^{n-1}\phi_{ij+1k}^{n-1}\\
&+ c_{bc}^{n-1}\phi_{ijk-1}^{n-1} + c_{tc}^{n-1}\phi_{ijk+1}^{n-1} + S_{ijk}
\end{aligned}
$$

$$d_0 = \frac{1}{6h_x^2h_y^2h_z^2 + 12k_1(h_x^2h_y^2 + h_y^2h_z^2 + h_x^2h_z^2)}$$

$$c_c^{n-1} = [6h_x^2h_y^2h_y^2 - 4k_1(h_x^2h_y^2 + h_y^2h_z^2 + h_x^2h_z^2)]e^{k_2}\, d_0$$

$$d_x^n = h_y^2h_z^2(6k_1 - h_x^2)\, d_0, \quad d_y^n = h_x^2h_z^2(6k_1 - h_y^2)\, d_0,$$
$$d_z^n = h_x^2h_y^2(6k_1 - h_z^2)\, d_0$$

$$c_{wc}^n = d_x^n e^{k_x}, \quad c_{ec}^n = d_x^n e^{-k_x}, \quad c_{sc}^n = d_y^n e^{k_y},$$
$$c_{nc}^n = d_y^n e^{-k_y}, \quad c_{bc}^n = d_z^n e^{k_z}, \quad c_{tc}^n = d_z^n e^{-k_z}$$

$$d_{xy}^{n-1} = k_1 h_x^2(h_x^2 + h_y^2)e^{k_2}\, d_0, \quad d_{yz}^{n-1} = k_1 h_x^2(h_y^2 + h_z^2)e^{k_2}\, d_0,$$
$$d_{xz}^{n-1} = k_1 h_y^2(h_x^2 + h_z^2)e^{k_2}\, d_0$$

$$c_{wsc}^{n-1} = d_{xy}^{n-1}e^{k_x+k_y}, \quad c_{wnc}^{n-1} = d_{xy}^{n-1}e^{k_x-k_y}, \quad c_{esc}^{n-1} = d_{xy}^{n-1}e^{-k_x+k_y},$$
$$c_{enc}^{n-1} = d_{xy}^{n-1}e^{-k_x-k_y} \quad c_{csb}^{n-1} = d_{yz}^{n-1}e^{k_y+k_z}, \quad c_{cst}^{n-1} = d_{yz}^{n-1}e^{k_y-k_z},$$
$$c_{cnb}^{n-1} = d_{yz}^{n-1}e^{-k_y+k_z}, \quad c_{cnt}^{n-1} = d_{yz}^{n-1}e^{-k_y-k_z} \quad c_{wcb}^{n-1} = d_{xz}^{n-1}e^{k_x+k_z},$$
$$c_{wct}^{n-1} = d_{xz}^{n-1}e^{k_x-k_z}, \quad c_{ecb}^{n-1} = d_{xz}^{n-1}e^{-k_x+k_z}, \quad c_{ect}^{n-1} = d_{xz}^{n-1}e^{-k_x-k_z}$$

$$d_x^{n-1} = [h_x^2h_y^2h_y^2 + 2k_1(h_y^2h_z^2 - h_x^2h_y^2 - h_x^2h_z^2)]e^{k_2}\, d_0,$$
$$c_{wc}^{n-1} = d_x^{n-1}e^{k_x}, \quad c_{ec}^{n-1} = d_x^{n-1}e^{-k_x}$$
$$d_y^{n-1} = [h_x^2h_y^2h_y^2 + 2k_1(h_x^2h_z^2 - h_x^2h_y^2 - h_y^2h_z^2)]e^{k_2}\, d_0,$$
$$c_{sc}^{n-1} = d_y^{n-1}e^{k_y}, \quad c_{nc}^{n-1} = d_y^{n-1}e^{-k_y}$$
$$d_z^{n-1} = [h_x^2h_y^2h_y^2 + 2k_1(h_x^2h_y^2 - h_y^2h_z^2 - h_x^2h_z^2)]e^{k_2}\, d_0,$$
$$c_{bc}^{n-1} = d_z^{n-1}e^{k_z}, \quad c_{tc}^{n-1} = d_z^{n-1}e^{-k_z}$$

$$
\begin{aligned}
S_{ijk} = \int_{t_{n-1}}^{t_n} &(c_c^{n-1}s_{ijk} + c_{wc}^{n-1}s_{i-1jk} + c_{ec}^{n-1}s_{i+1jk}\\
&+ c_{sc}^{n-1}s_{ij-1k} + c_{nc}^{n-1}s_{ij+1k} + c_{bc}^{n-1}s_{ijk-1} + c_{tc}^{n-1}s_{ijk+1}\\
&+ c_{wsc}^{n-1}s_{i-1j-1k} + c_{wnc}^{n-1}s_{i-1j+1k} + c_{esc}^{n-1}s_{i+1j-1k}\\
&+ c_{enc}^{n-1}s_{i+1j+1k} + c_{csb}^{n-1}s_{ij-1k-1} + c_{cst}^{n-1}s_{ij-1k+1}\\
&+ c_{cnb}^{n-1}s_{ij+1k-1} + c_{cnt}^{n-1}s_{ij+1k+1} + c_{wcb}^{n-1}s_{i-1jk-1}\\
&+ c_{wct}^{n-1}s_{i-1jk+1} + c_{ecb}^{n-1}s_{i+1jk-1} + c_{ect}^{n-1}s_{i+1jk+1})\, dt;
\end{aligned}
$$

(d) fourth-order strongly implicit scheme (FOSIS),

$$
\begin{aligned}
\phi_{ijk}^n = {}& c_{wc}^n\phi_{i-1jk}^n + c_{ec}^n\phi_{i+1jk}^n + c_{sc}^n\phi_{ij-1k}^n + c_{nc}^n\phi_{ij+1k}^n\\
&+ c_{bc}^n\phi_{ijk-1}^n + c_{tc}^n\phi_{ijk+1}^n + c_{wsc}^n\phi_{i-1j-1k}^n + c_{wnc}^n\phi_{i-1j+1k}^n\\
&+ c_{esc}^n\phi_{i+1j-1k}^n + c_{enc}^n\phi_{i+1j+1k}^n + c_{csb}^n\phi_{ij-1k-1}^n\\
&+ c_{cst}^n\phi_{ij-1k+1}^n + c_{cnb}^n\phi_{ij+1k-1}^n + c_{cnt}^n\phi_{ij+1k+1}^n\\
&+ c_{wcb}^n\phi_{i-1jk-1}^n + c_{wct}^n\phi_{i-1jk+1}^n + c_{ecb}^n\phi_{i+1jk-1}^n\\
&+ c_{ect}^n\phi_{i+1jk+1}^n + c_c^{n-1}\phi_{ijk}^{n-1} + c_{wc}^{n-1}\phi_{i-1jk}^{n-1}\\
&+ c_{ec}^{n-1}\phi_{i+1jk}^{n-1} + c_{sc}^{n-1}\phi_{ij-1k}^{n-1} + c_{nc}^{n-1}\phi_{ij+1k}^{n-1}\\
&+ c_{bc}^{n-1}\phi_{ijk-1}^{n-1} + c_{tc}^{n-1}\phi_{ijk+1}^{n-1} + S_{ijk}
\end{aligned}
$$

$$d_0 = \frac{1}{6h_x^2h_y^2h_z^2 + 4k_1(h_x^2h_y^2 + h_y^2h_z^2 + h_x^2h_z^2)}$$

$$c_c^{n-1} = [6h_x^2h_y^2h_y^2 - 12k_1(h_x^2h_y^2 + h_y^2h_z^2 + h_x^2h_z^2)]e^{k_2}\, d_0$$

$$d_x^n = [2k_1(h_y^2h_z^2 - h_x^2h_y^2 - h_x^2h_z^2) - h_x^2h_y^2h_z^2]\, d_0,$$
$$c_{wc}^n = d_x^n e^{k_x}, \quad c_{ec}^n = d_x^n e^{-k_x}$$
$$d_y^n = [2k_1(h_x^2h_z^2 - h_x^2h_y^2 - h_y^2h_z^2) - h_x^2h_y^2h_z^2]\, d_0,$$
$$c_{sc}^n = d_y^n e^{k_y}, \quad c_{nc}^n = d_y^n e^{-k_y}$$
$$d_z^n = [2k_1(h_x^2h_y^2 - h_y^2h_z^2 - h_x^2h_z^2) - h_x^2h_y^2h_z^2]\, d_0,$$
$$c_{bc}^n = d_z^n e^{k_z}, \quad c_{tc}^{n-1} = d_z^n e^{-k_z}$$

$$d_{xy}^n = k_1 h_x^2(h_x^2 + h_y^2)\, d_0, \quad d_{yz}^n = k_1 h_x^2(h_y^2 + h_z^2)\, d_0,$$
$$d_{xz}^n = k_1 h_y^2(h_x^2 + h_z^2)\, d_0$$

$$c_{wsc}^n = d_{xy}^n e^{k_x+k_y}, \quad c_{wnc}^n = d_{xy}^n e^{k_x-k_y}, \quad c_{esc}^n = d_{xy}^n e^{-k_x+k_y},$$
$$c_{enc}^n = d_{xy}^n e^{-k_x-k_y} \quad c_{csb}^n = d_{yz}^n e^{k_y+k_z}, \quad c_{cst}^n = d_{yz}^n e^{k_y-k_z},$$
$$c_{cnb}^n = d_{yz}^n e^{-k_y+k_z}, \quad c_{cnt}^n = d_{yz}^n e^{-k_y-k_z} \quad c_{wcb}^n = d_{xz}^n e^{k_x+k_z},$$
$$c_{wct}^n = d_{xz}^n e^{k_x-k_z}, \quad c_{ecb}^n = d_{xz}^n e^{-k_x+k_z}, \quad c_{ect}^n = d_{xz}^n e^{-k_x-k_z}$$

$$d_x^{n-1} = h_y^2h_z^2(6k_1 + h_x^2)e^{k_2}\, d_0,$$
$$c_{wc}^{n-1} = d_x^{n-1}e^{k_x}, \quad c_{ec}^{n-1} = d_x^{n-1}e^{-k_x},$$
$$d_y^{n-1} = h_x^2h_z^2(6k_1 + h_y^2)e^{k_2}\, d_0,$$
$$c_{sc}^{n-1} = d_y^{n-1}e^{k_y}, \quad c_{nc}^{n-1} = d_y^{n-1}e^{-k_y}$$
$$d_z^{n-1} = h_x^2h_y^2(6k_1 + h_z^2)e^{k_2}\, d_0,$$
$$c_{bc}^{n-1} = d_z^{n-1}e^{k_z}, \quad c_{tc}^{n-1} = d_z^{n-1}e^{-k_z}$$

$$
\begin{aligned}
S_{ijk} = \int_{t_{n-1}}^{t_n} &(c_c^{n-1}s_{ijk} + c_{wc}^{n-1}s_{i-1jk} + c_{ec}^{n-1}s_{i+1jk}\\
&+ c_{sc}^{n-1}s_{ij-1k} + c_{nc}^{n-1}s_{ij+1k} + c_{bc}^{n-1}s_{ijk-1} + c_{tc}^{n-1}s_{ijk+1})\, dt.
\end{aligned}
$$

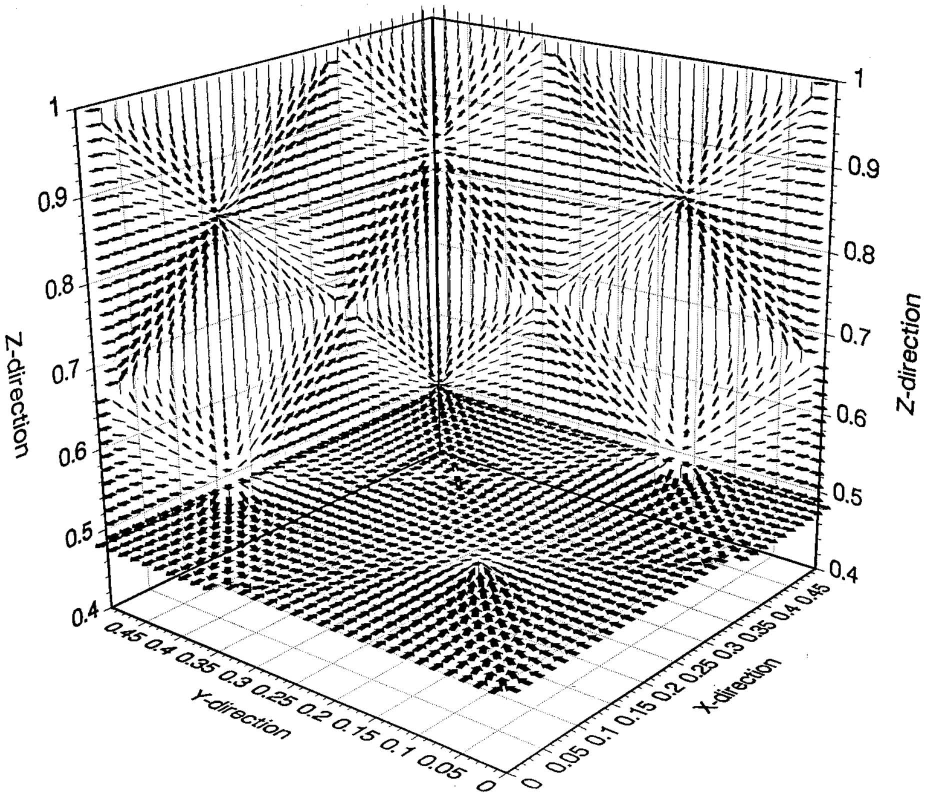

FIGURE 2

The truncation error of the second-order schemes is

$$O\left(\sum_{i=0}^{1} h_x^{2i} k_1^{1-i}\right) + O\left(\sum_{i=0}^{1} h_y^{2i} k_1^{1-i}\right) + O\left(\sum_{i=0}^{1} h_z^{2i} k_1^{1-i}\right) + O(\tau),$$

while the truncation error of the fourth-order schemes is

$$\begin{aligned} &O\left(\sum_{i=0}^{2} h_x^{2i} k_1^{2-i}\right) + O\left(\sum_{i=0}^{2} h_y^{2i} k_1^{2-i}\right) + O\left(\sum_{i=0}^{2} h_z^{2i} k_1^{2-i}\right) \\ &\quad + O\left[\left(\sum_{i=0}^{1} h_x^{2i} k_1^{1-i}\right)\left(\sum_{i=0}^{1} h_y^{2i} k_1^{1-i}\right)\right] \\ &\quad + O\left[\left(\sum_{i=0}^{1} h_y^{2i} k_1^{1-i}\right)\left(\sum_{i=0}^{1} h_z^{2i} k_1^{1-i}\right)\right] \\ &\quad + O\left[\left(\sum_{i=0}^{1} h_x^{2i} k_1^{1-i}\right)\left(\sum_{i=0}^{1} h_z^{2i} k_1^{1-i}\right)\right] + O(\tau). \end{aligned}$$

Here, it should be noted that the trucation errors can be separated into two parts. The first part is basically the products of $h_x k_1$, $h_y k_1$, $h_z k_1$ and the spatial derivatives of the unknown variable ϕ. The second part is the product of τ and the derivatives of the source term s, which can be easily controlled if the source term is known.

5. BENCHMARK ANALYTICAL SOLUTIONS AND NUMERICAL ALGORITHMS

The accuracy and numerical behavior of the four schemes outlined in the previous section are compared to those of conventional numerical algorithms. The new schemes and the conventional methods are benchmarked against analytical solutions to Burgers equations and Navier–Stokes equations. Additionally, a grid refinement study is done.

The first nonlinear analytical solution to Burgers equations was given separately by Hopf [14] and Cole [15]. Cole [15] pointed out that this transformation could be interpreted as a multidimensional transformation. Following Cole's idea, Fletcher [16] obtained two-dimensional steady solutions to Burgers equations and this approach is extended here to three dimensions and time. The equations are

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} + v\frac{\partial \phi}{\partial y} + w\frac{\partial \phi}{\partial z} = \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}\right)$$

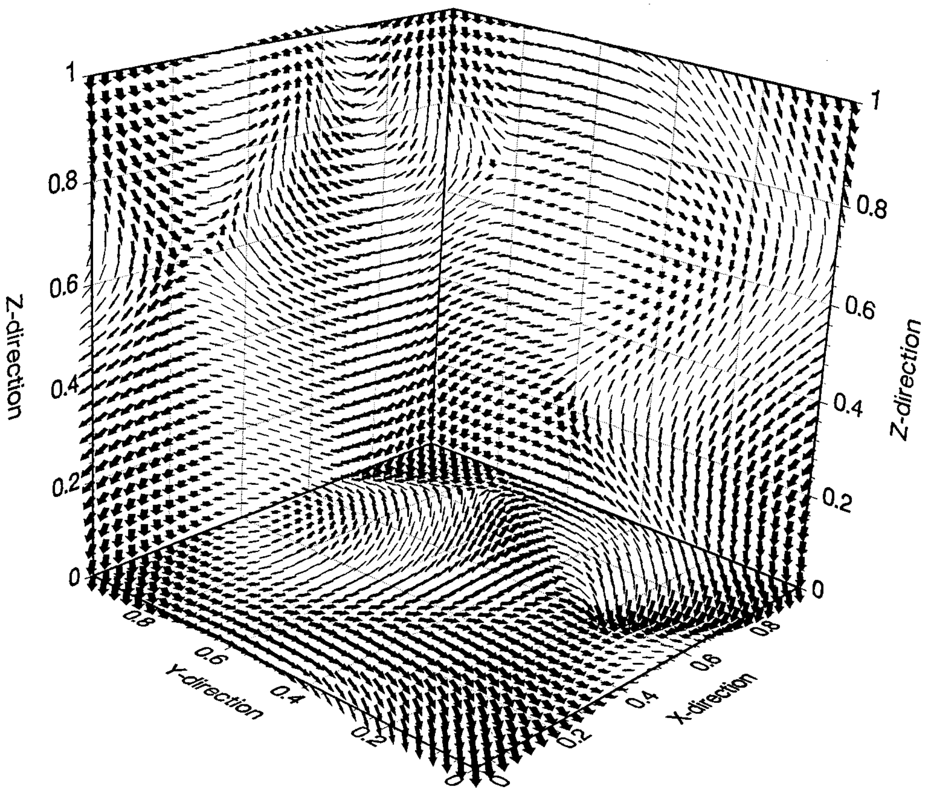

FIGURE 3

where, in general, $\phi = u, v, w$.

The nonlinear, but exact, solutions are

$$u = -\frac{2}{\text{Re}}[a_2 e^{-\lambda(t/\text{Re})} n_x \pi \cos(n_x \pi x) \sin(n_y \pi y) \sin(n_z \pi z)]\, d_0$$

$$v = -\frac{2}{\text{Re}}[a_2 e^{-\lambda(t/\text{Re})} n_y \pi \sin(n_x \pi x) \cos(n_y \pi y) \sin(n_z \pi z)]\, d_0$$

$$w = -\frac{2}{\text{Re}}[a_2 e^{-\lambda(t/\text{Re})} n_z \pi \sin(n_x \pi x) \sin(n_y \pi y) \cos(n_z \pi z)]\, d_0,$$

where

$$\lambda = \pi^2(n_x^2 + n_y^2 + n_z^2),$$

$$d_0 = \frac{1}{a_1 + a_2 e^{-\lambda(t/\text{Re})} \sin(n_x \pi x) \sin(n_y \pi y) \sin(n_z \pi z)},$$

where a_1 and a_2 are adjustable constants which control the amplitude of the solutions.

The exact solution is demonstrated, in part, by the vector plot in Fig. 2, which displays the velocity field on three surfaces of a cube, where three wavenumbers and two parameters have been used as $n_x = n_y = n_z = 3$, $a_1 = 1$, and $a_2 = 0.1$.

As is well known, the three-dimensional time-dependent Navier–Stokes equations are

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{v},$$

where $\mathbf{v} = (u, v, w)$ is the velocity vector and $\nabla = (\partial/\partial_x)\mathbf{i} + (\partial/\partial_y)\mathbf{j} + (\partial/\partial z)\mathbf{k}$.

Steinman *et al.* [17] provide a class of exact solutions for these equations, combined with conservation of mass,

$$u = -a_2 e^{-a_1^2(t/\text{Re})}[e^{a_2 x} \sin(a_2 y \pm a_1 z) + e^{a_2 z} \cos(a_2 x \pm a_1 y)]$$

$$v = -a_2 e^{-a_1^2(t/\text{Re})}[e^{a_2 y} \sin(a_2 z \pm a_1 x) + e^{a_2 x} \cos(a_2 y \pm a_1 z)]$$

$$w = -a_2 e^{-a_1^2(t/\text{Re})}[e^{a_2 z} \sin(a_2 x \pm a_1 y) + e^{a_2 y} \cos(a_2 z \pm a_1 x)]$$

$$p = -\tfrac{1}{2}(u^2 + v^2 + w^2),$$

where a_1 and a_2 are adjustable constants, which control the amplitude and frequency of the solutions.

The flow field produced by these exact solutions is displayed in part in Fig. 3, where $a_1 = 2\pi$ and $a_2 = 0.1$.

Several types of numerical algorithms for the solution to the CDE are available; here, we choose those given

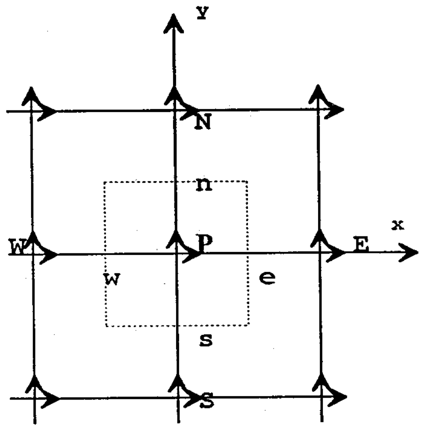

FIGURE 4

in Patankar [3]. The exponential scheme of Patankar is adopted for spatial discretization. In the temporal direction, both first-order implicit and Crank–Nicholson implicit schemes are used. Since the issue of velocity-pressure coupling is not considered in this paper, there is no need to use a staggered grid. The arrangment of the collocated grid and the corresponding control volume in two dimensions is given in Fig. 4.

6. NUMERICAL TESTS

6.1. *Boundary and Initial Conditions*

The Burgers and Navier–Stokes equations are solved, both analytically and computationally, over a cubic domain. The time dependent boundary conditions are prescribed by the analytical solutions on the surfaces of the cube. The initial conditions of the flow are those corresponding to analytical solutions at $t = 0$.

6.2. *Error Evaluation*

The relative error fields are determined between the numerical and the analytical solutions, that is, $\varepsilon_f = (f_{\text{num}} - f_{\text{ana}})/f_{\text{avg}}$, where f can be any variable of u, v, w; f_{num} and f_{ana} are numerical and analytical solutions and $f_{\text{avg}} = [(1/N)(\sum_{n=1}^{N} f_n^2)]^{1/2}$, where f_n is the value of analytical solution on a nodal point n and N is the number of nodal points. The error of a scheme is estimated as the maximum value of the three rms relative errors, ε_u, ε_v, and ε_w.

6.3. *Scheme and Nomenclature Specification*

Six numerical schemes are used to solve both Burgers and Navier–Stokes equations. The schemes are: second-order explicit scheme (SOES); second-order implicit scheme (SOIS); fourth-order strongly implicit scheme (FOSIS); fourth-order weakly implicit scheme (FOSIS), which are supplemented by two other, more traditional, schemes: a temporally first-order implicit scheme (FOIS) and the Crank–Nicholson implicit scheme (CNIS).

6.4. *Burgers Equations*

In the numerical tests with Burgers equations, some common test parameters are chosen. The three spatial wave numbers in the analytical solution to Burgers equation are $n_x = n_y = n_z = 3$. The two adjustable parameters, which control the amplitude of the solution, are $a_1 = 1.0$ and $a_2 = 0.1$. Of course, other values can be easily chosen. The total integration time is 0.1 and the computational domain is cubic at $1 \times 1 \times 1$ (dimensionless) and equally divided into $19 \times 19 \times 19$ control volumes.

In Test I, the time step, typically, is chosen as 1.0×10^{-3} and the Reynolds number is 10. Table I gives the rms relative errors for these six schemes, after 100 time steps. It should be noted that the time step for the FOSIS is chosen as 1.0×10^{-4}; otherwise a stable solution could not be obtained.

The rms error distributions on one plane that result from applying the six schemes are presented in Fig. 5 with the error axes scaled to the same level. Increaseing the Reynolds number to 100, while maintaining the time step as 1.0×10^{-3} produces little change in the magnitudes of the rms relative errors, after 100 time steps as noted in Table II.

A number of grid refinement tests are performed. Parameter a_1 in the analytical solution of Burgers equation can be adjusted to control the amplitude of the velocity. Numerical tests show that the behaviour of SOES, SOIS, FOIS, and CNIS are similar to one another, while those of FOWIS are almost the same to FOSIS. Figure 6 displays the typical numerical behaviour of SOES and FOWIS when the grid spacings are refined. These results will be discussed later.

TABLE I

The Comparison of RMS Errors

Schemes	SOES	SOIS	FOWIS	FOSIS	FOIS	CNIS
Rms errors	9.3×10^{-3}	5.6×10^{-2}	1.7×10^{-4}	9.0×10^{-4}	6.0×10^{-2}	3.4×10^{-2}

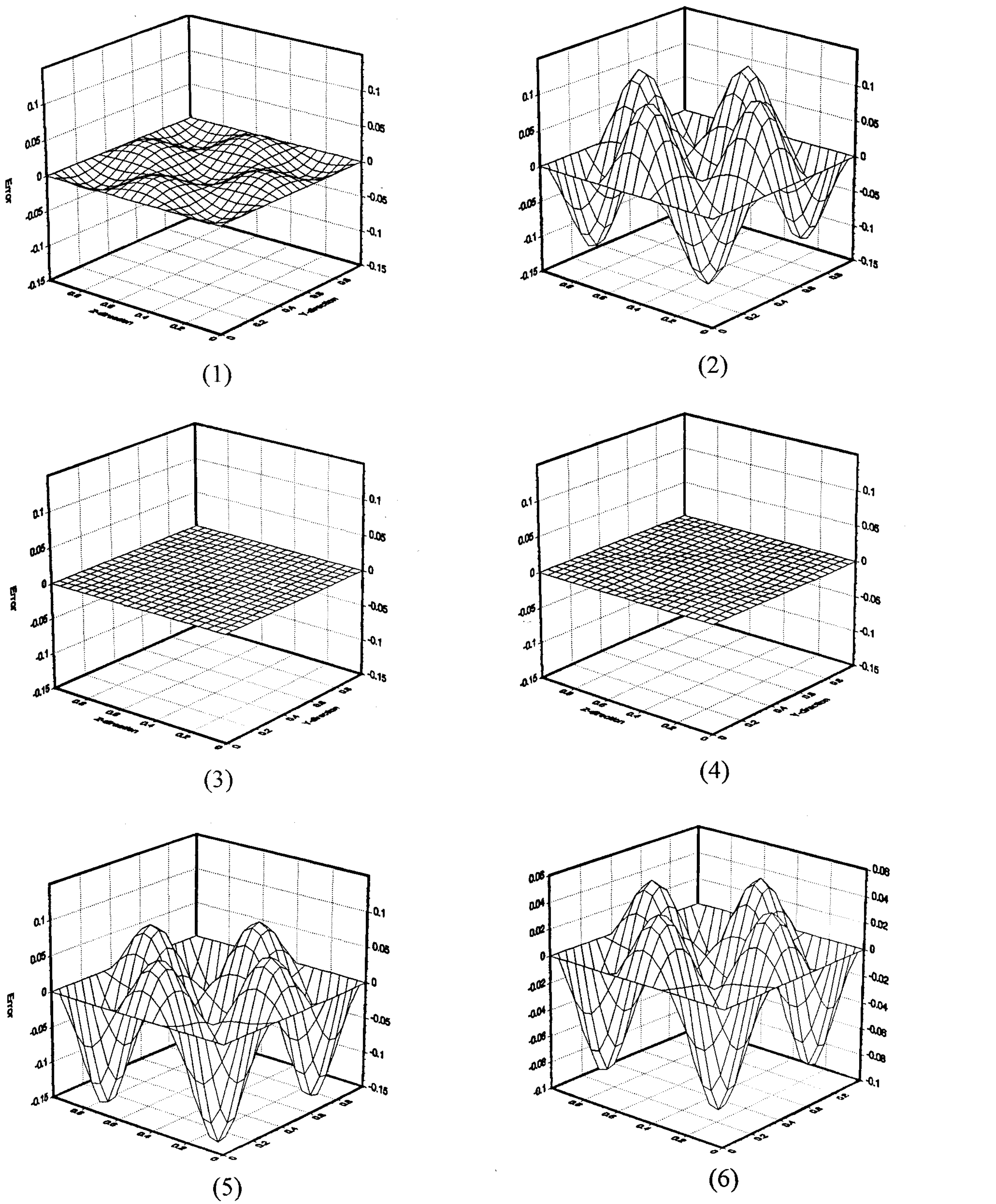

FIG. 5. Error comparisons of the six schemes against the exact solution of Burgers equations. A typical cross section is chosen at $i = 10$ and the error distributions are on an y-z plane: (1) SOES; (2) SOIS; (3) FOWIS; (4) FOSIS; (5) FOIS; (6) CNIS.

TABLE II

The Comparison of RMS Errors

Schemes	SOES	SOIS	FOWIS	FOSIS	FOIS	CNIS
Rms errors	3.3×10^{-3}	3.8×10^{-3}	5.3×10^{-4}	5.3×10^{-4}	1.2×10^{-2}	6.9×10^{-2}

TABLE III

The Comparison of RMS Errors

Schemes	SOES	SOIS	FOWIS	FOSIS	FOIS	CNIS
Rms errors	3.1×10^{-3}	1.7×10^{-2}	3.6×10^{-4}	1.7×10^{-4}	1.7×10^{-2}	7.6×10^{-3}

6.5. *Navier–Stokes Equations*

In the numerical tests with Navier–Stokes equations, some common test parameters are chosen. The number of control volumes is chosen as $19 \times 19 \times 19$. The computational domain is set to a cube with side lengths of unity. The adjustable parameter, which controls the amplitude of the solution, is $a_1 = 2\pi$ and $a_2 = 0.1$, which controls both the amplitude and the wave number of the solution. The time step is chosen as 5.0×10^{-3} and total integration time is 0.5. The Reynolds number is set at 10. Table III gives the rms relative errors of the six schemes after 100 time steps. It should be noted that the time step is chosen as 2.5×10^{-3} for SOES and as 1.0×10^{-4} for FOSIS to enforce a stable solution.

The rms error distributions on one plane are presented in Fig. 7 with the error axes scaled to the same level. Increasing the Reynolds number, integration time, and time step to 100, 10, and 0.1, respectively, while decreasing a_2 to 0.01, gives the rms relative errors, after 100 time steps, as noted in Table IV. It should be noted that the time step is chosen as 1.0×10^{-4} for SOES and as 1.0×10^{-3} for FOSIS to enforce a stable solution.

The grid refinement studies are done also in this case. Parameter a_2 in the exact solution of N-S equation is used to control the amplitude of velocity. Again, it is observed that the behavior of SOES, SOIS, FOIS, and CNIS are similar to one another, while that of FOWIS is almost the same as FOSIS. Figure 8 shows the numerical behaviour of SOES and FOWIS when the grid spacing is refined. The results will be commented upon in the following section.

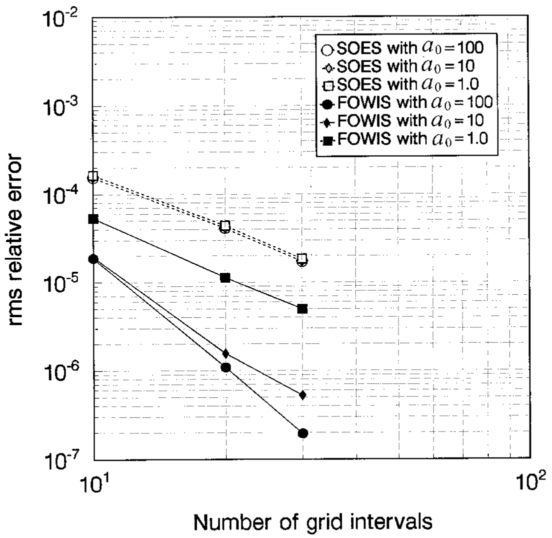

FIGURE 6

7. DISCUSSION

Usually, the conventional higher order difference schemes achieve higher order accuracy by introducing more neighbouring grid points into the schemes, for example, Rai and Kim [7], which results in decreasing the sparsity of the discretized equation. And still further, the compact difference schemes proposed in Lele [9] lose the sparse property, which leads to a dense matrix when introducing the derivatives on the center and its neighbour points. The major advantage of the higher order difference schemes (FOWIS and FOSIS) is that higher order accuracy is obtained by decreasing the sparsity of the discretized equation as little as possible, which will significantly reduces the computational work required to solve the sparse matrix.

The formulations of the difference schemes reveal three types of characteristic parameters in the discretization of CDE. The first type, k_x, k_y, k_z, represents the local Reynolds numbers based on the grid sizes in three spatial directions. The second type is the diffusive length, $k_1 = \tau b$, which

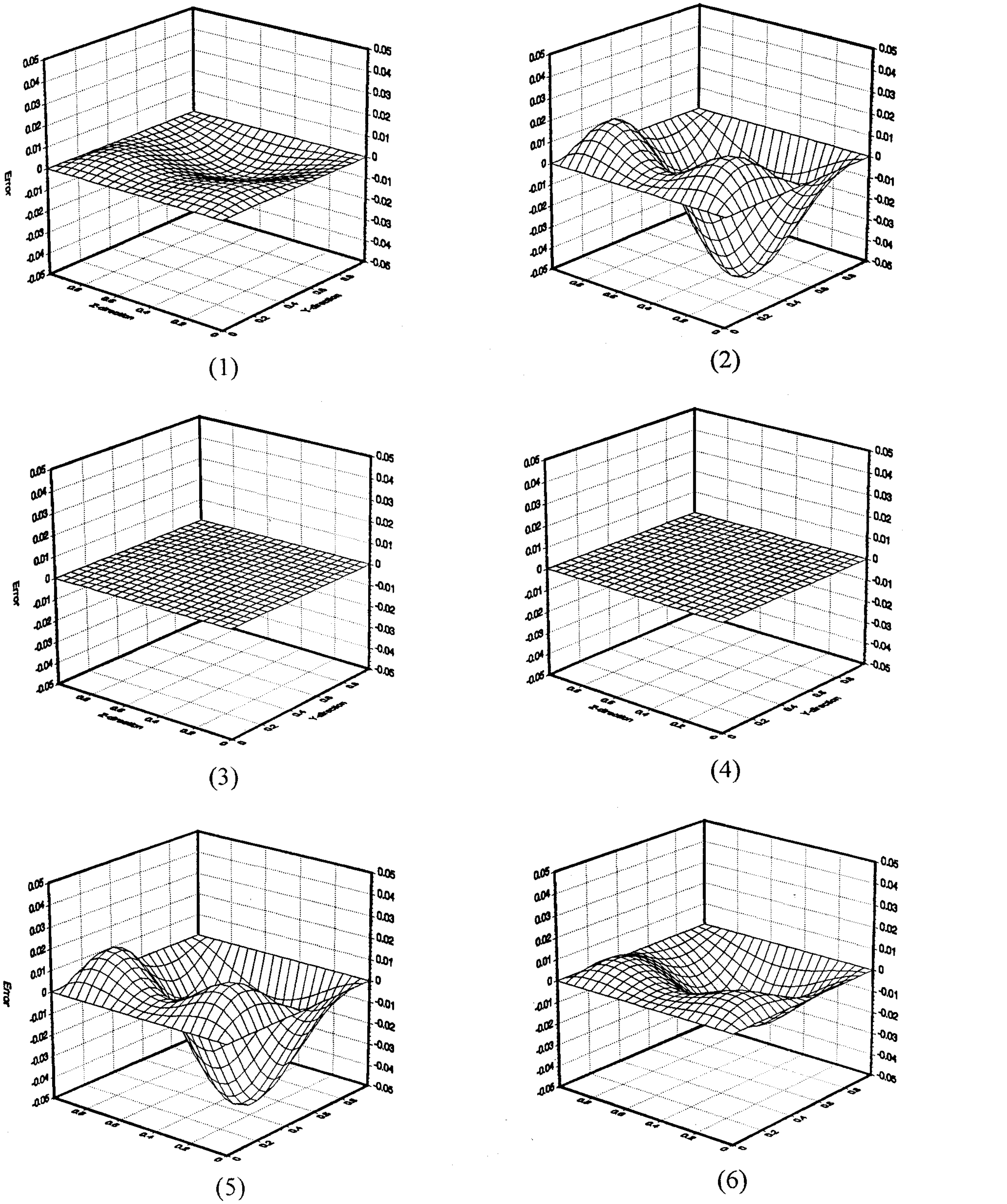

FIG. 7. Error comparisons of the six schemes against the exact solution of Navier–Stokes equations. A typical cross section is chosen at $i = 10$ and the error distributions are on an y-z plane: (1) SOES; (2) SOIS; (3) FOWIS; (4) FOSIS; (5) FOIS; (6) CNIS.

TABLE IV

The Comparison of RMS Errors

Schemes	SOES	SOIS	FOWIS	FOSIS	FOIS	CNIS
Rms errors	1.1×10^{-2}	5.1×10^{-2}	7.9×10^{-4}	1.8×10^{-4}	5.0×10^{-2}	1.4×10^{-2}

can be interpreted as the region affected by molecular diffusion within the prescribed time interval of τ. Actually, via this parameter, the time interval τ can be equivalent to the spatial step length. The third type, k_2, which, by introducing a characteristic grid length size Δ, can be rewritten as $k_2 = -U^2\tau/(4b) = -1/4(U\Delta/b)(\tau U/\Delta)$ and $U^2 = a_x^2 + a_y^2 + a_z^2$. Thus, the physical interpretation of k_2 is the ratio of the local Reynolds number $U\Delta/b$ and the local Strouhal number $\Delta/(\tau U)$.

The transformation brings some numerical dissipation into the difference schemes. This can be understood by noticing that the effect of the transforamtion is to weight the neighbours with respect to the central nodal point; this weight is dependent upon the magnitude of the local grid Reynolds number. To qualitatively check the property of this dissipation, the method introduced in [2] can be used. The discretization form for the unsteady one-dimensional convection-diffusion model equation (see Patankar [3]) reads

$$\phi_i^n = \frac{k_1}{h_x^2 + 2k_1}\frac{2e^{k_x}}{e^{k_x} + e^{-k_x}}\phi_{i-1}^n + \frac{k_1}{h_x^2 + 2k_1}\frac{2e^{-k_x}}{e^{k_x} + e^{-k_x}}\phi_{i+1}^n + \frac{h_x^2}{h_x^2 + 2k_1}\phi_i^{n-1},$$

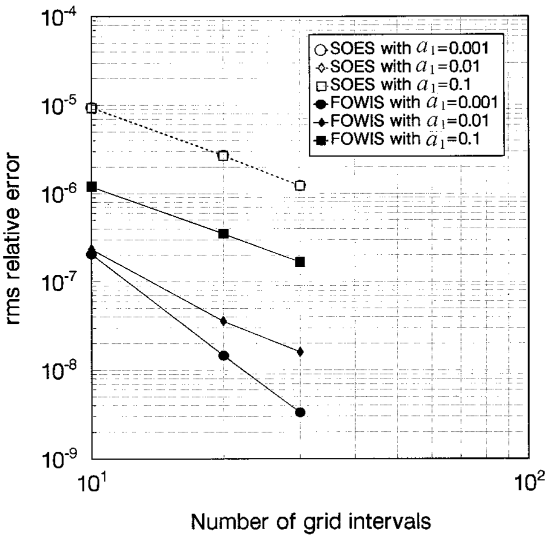

FIGURE 8

while by the method introduced in this paper, it reads

$$\phi_i^n = \frac{k_1}{h_x^2 + 2k_1}e^{k_x}\phi_{i-1}^n + \frac{k_1}{h_x^2 + 2k_1}e^{-k_x}\phi_{i+1}^n + \frac{h_x^2}{h_x^2 + 2k_1}e^{k_2}\phi_i^{n-1},$$

where k_2 in a one-dimensional case should be written as $k_2 = \tau c$, $c = -a_x^2/(4b)$.

Now, we see that the appropriate dissipating (or weighting) function form should be $2e^{k_x}/(e^{k_x} + e^{-k_x})$ rather than e^{k_x} for convection-dominated one-dimensional flows. The two functions are shown in Fig. 9, where it is seen that the functions are in accord only when k_x (local Reynolds number) is small. This indicates that the schemes for CDE are only suitable for small local Reynolds number. Furthermore, the grid refinement studies reveal (Fig. 6 and 8) that the higher order schemes are sensitive to the local Reynolds number and the order of the scheme resolution tends to deteriorate with increasing local Reynolds number. This is deemed to be an impediment and future work will be done to alleviate this defect.

It is interesting to note that the form of the transformation function, i.e., Eq. (2), is similar in form to the differential filter function proposed by Germano [18] for use in large eddy simulation (LES). In LES, the effect of the filter

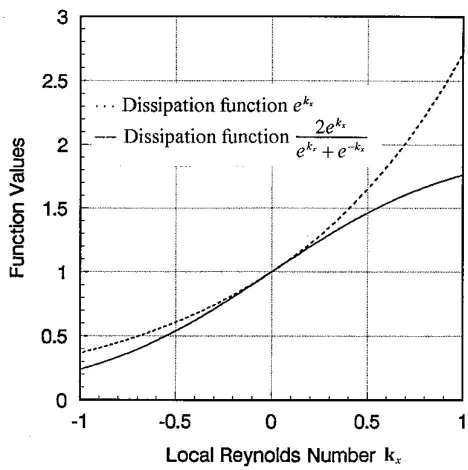

FIGURE 9

is to separate the large and small scales. The small scales are usually represented by some eddy-viscosity model, which is basically a model of energy dissipation. Obviously, the transformation in the present schemes plays the same role as a filter, as they introduce some numerical dissipation. This dissipation vanishes when the local Reynolds number is small enough, i.e., the grid resolution is sufficiently high.

8. CONCLUSION

The idea of global discretization is explored in this paper. Via this idea, four finite difference schemes for CDE$^{\mathrm{T}}$, i.e., SOES, SOIS, FOWIS, and FOSIS, were established. The diffusive length is found to be a significant parameter in linking temporal discretisation with spatial discretisation. Further, via an inverse exponential transformation, four finite difference schemes for the CDE are obtained. The numerical experiments indicate that higher resolutions are achieved by the high order schemes, i.e., FOWIS and FOSIS, compared against the other four schemes, especially for long-time integrations. The numerical tests show that the stability requirement for SOIS and FOWIS is much less restrictive than that for SOES and FOSIS. The grid refinement study on the schemes for the CDE revealed that the inverse exponential transformation on the finite difference schemes for the CDE$^{\mathrm{T}}$ tends to destroy some resolution of the schemes for the CDE and some future work will address this defect.

APPENDIX: FINITE DIFFERENCE SCHEMES FOR CDE$^{\mathrm{T}}$

Second-order explicit scheme (SOES),

$$T^n_{ijk} = c^{n-1}_c T^{n-1}_{ijk} + c^{n-1}_{wc} T^{n-1}_{i-1jk} + c^{n-1}_{ec} T^{n-1}_{i+1jk} + c^{n-1}_{sc} T^{n-1}_{ij-1k} + c^{n-1}_{nc} T^{n-1}_{ij+1k} + c^{n-1}_{bc} T^{n-1}_{ijk-1} + c^{n-1}_{tc} T^{n-1}_{ijk+1} + S_{ijk}$$

$$d_0 = \frac{1}{h_x^2 h_y^2 h_z^2},$$

$$c^{n-1}_c = [h_x^2 h_y^2 h_z^2 - 2k_1(h_x^2 h_y^2 + h_y^2 h_z^2 + h_x^2 h_z^2)]e^{k_2}\, d_0$$

$$c^{n-1}_{wc} = c^{n-1}_{ec} = k_1 h_y^2 h_z^2 e^{k_2}\, d_0, \quad c^{n-1}_{sc} = c^{n-1}_{nc} = k_1 h_x^2 h_z^2 e^{k_2}\, d_0,$$

$$c^{n-1}_{bc} = c^{n-1}_{tc} = k_1 h_x^2 h_y^2 e^{k_2}\, d_0$$

$$S_{ijk} = \int_{t_{n-1}}^{t_n} (c^{n-1}_c s^T_{ijk} + c^{n-1}_{wc} s^T_{i-1jk} + c^{n-1}_{ec} s^T_{i+1jk} + c^{n-1}_{sc} s^T_{ij-1k} + c^{n-1}_{nc} s^T_{ij+1k} + c^{n-1}_{bc} s^T_{ijk-1} + c^{n-1}_{tc} s^T_{ijk+1})\, dt.$$

Second-order implicit scheme (SOIS),

$$T^n_{ijk} = c^n_{wc} T^n_{i-1jk} + c^n_{ec} T^n_{i+1jk} + c^n_{sc} T^n_{ij-1k} + c^n_{nc} T^n_{ij+1k} + c^n_{bc} T^n_{ijk-1} + c^n_{tc} T^n_{ijk+1} + c^{n-1}_c T^{n-1}_{ijk} + S_{ijk}$$

$$d_0 = \frac{1}{h_x^2 h_y^2 h_z^2 + 2k_1(h_x^2 h_y^2 + h_y^2 h_z^2 + h_x^2 h_z^2)},$$

$$c^{n-1}_c = h_x^2 h_y^2 h_z^2 e^{k_2}\, d_0 \quad c^n_{wc} = c^n_{ec} = k_1 h_y^2 h_z^2\, d_0,$$

$$c^n_{sc} = c^n_{nc} = k_1 h_x^2 h_z^2\, d_0, \quad c^n_{bc} = c^n_{tc} = k_1 h_x^2 h_y^2\, d_0$$

$$S_{ijk} = \int_{t_{n-1}}^{t_n} c^{n-1}_c s_{ijk}\, dt.$$

The truncation error of the second-order schemes,

$$O\left(\sum_{i=0}^{1} h_x^{2i} k_1^{1-i}\right) + O\left(\sum_{i=0}^{1} h_y^{2i} k_1^{1-i}\right) + O\left(\sum_{i=0}^{1} h_z^{2i} k_1^{1-i}\right) + O(\tau).$$

Fourth-order weakly implicit scheme (FOWIS),

$$\begin{aligned} T^n_{ijk} = {} & c^n_{wc} T^n_{i-1jk} + c^n_{ec} T^n_{i+1jk} + c^n_{sc} T^n_{ij-1k} + c^n_{nc} T^n_{ij+1k} \\ & + c^n_{bc} T^n_{ijk-1} + c^n_{tc} T^n_{ijk+1} + c^{n-1}_{wsc} T^{n-1}_{i-1j-1k} + c^{n-1}_{wnc} T^{n-1}_{i-1j+1k} \\ & + c^{n-1}_{esc} T^{n-1}_{i+1j-1k} + c^{n-1}_{enc} T^{n-1}_{i+1j+1k} + c^{n-1}_{csb} T^{n-1}_{ij-1k-1} \\ & + c^{n-1}_{cst} T^{n-1}_{ij-1k+1} + c^{n-1}_{cnb} T^{n-1}_{ij+1k-1} + c^{n-1}_{cnt} T^{n-1}_{ij+1k+1} \\ & + c^{n-1}_{wcb} T^{n-1}_{i-1jk-1} + c^{n-1}_{wct} T^{n-1}_{i-1jk+1} + c^{n-1}_{ecb} T^{n-1}_{i+1jk-1} \\ & + c^{n-1}_{ect} T^{n-1}_{i+1jk+1} + c^{n-1}_c T^{n-1}_{ijk} + c^{n-1}_{wc} T^{n-1}_{i-1jk} \\ & + c^{n-1}_{ec} T^{n-1}_{i+1jk} + c^{n-1}_{sc} T^{n-1}_{ij-1k} + c^{n-1}_{nc} T^{n-1}_{ij+1k} \\ & + c^{n-1}_{bc} T^{n-1}_{ijk-1} + c^{n-1}_{tc} T^{n-1}_{ijk+1} + S_{ijk} \end{aligned}$$

$$d_0 = \frac{1}{6h_x^2 h_y^2 h_z^2 + 12k_1(h_x^2 h_y^2 + h_y^2 h_z^2 + h_x^2 h_z^2)}$$

$$c^{n-1}_c = [6h_x^2 h_y^2 h_y^2 - 4k_1(h_x^2 h_y^2 + h_y^2 h_z^2 + h_x^2 h_z^2)]e^{k_2}\, d_0$$

$$c^n_{wc} = c^n_{ec} = h_y^2 h_z^2 (6k_1 - h_x^2)\, d_0,$$

$$c^n_{sc} = c^n_{nc} = h_x^2 h_z^2 (6k_1 - h_y^2)\, d_0,$$

$$c^n_{bc} = c^n_{tc} = h_x^2 h_y^2 (6k_1 - h_z^2)\, d_0$$

$$c^{n-1}_{wsc} = c^{n-1}_{wnc} = c^{n-1}_{esc} = c^{n-1}_{enc} = k_1 h_z^2 (h_x^2 + h_y^2) e^{k_2}\, d_0$$

$$c^{n-1}_{csb} = c^{n-1}_{cst} = c^{n-1}_{cnb} = c^{n-1}_{cnt} = k_1 h_x^2 (h_y^2 + h_z^2) e^{k_2}\, d_0$$

$$c^{n-1}_{wcb} = c^{n-1}_{wct} = c^{n-1}_{ecb} = c^{n-1}_{ect} = k_1 h_y^2 (h_x^2 + h_z^2) e^{k_2}\, d_0$$

$$c^{n-1}_{wc} = c^{n-1}_{ec} = [h_x^2 h_y^2 h_y^2 + 2k_1(h_y^2 h_z^2 - h_x^2 h_y^2 - h_x^2 h_z^2)]e^{k_2}\, d_0$$

$$c^{n-1}_{sc} = c^{n-1}_{nc} = [h_x^2 h_y^2 h_y^2 + 2k_1(h_x^2 h_z^2 - h_x^2 h_y^2 - h_y^2 h_z^2)]e^{k_2}\, d_0$$

$$c^{n-1}_{bc} = c^{n-1}_{tc} = [h_x^2 h_y^2 h_y^2 + 2k_1(h_x^2 h_y^2 - h_y^2 h_z^2 - h_x^2 h_z^2)]e^{k_2}\, d_0$$

$$S_{ijk} = \int_{t_{n-1}}^{t_n} (c^{n-1}_c s^T_{ijk} + c^{n-1}_{wc} s^T_{i-1jk} + c^{n-1}_{ec} s^T_{i+1jk} + c^{n-1}_{sc} s^T_{ij-1k} + c^{n-1}_{nc} s^T_{ij+1k} + c^{n-1}_{bc} s^T_{ijk-1} + c^{n-1}_{tc} s^T_{ijk+1}$$

$$+ c_{wsc}^{n-1} s_{i-1j-1k}^{T} + c_{wnc}^{n-1} s_{i-1j+1k}^{T} + c_{esc}^{n-1} s_{i+1j-1k}^{T}$$
$$+ c_{enc}^{n-1} s_{i+1j+1k}^{T} + c_{csb}^{n-1} s_{ij-1k-1}^{T} + c_{cst}^{n-1} s_{ij-1k+1}^{T}$$
$$+ c_{cnb}^{n-1} s_{ij+1k-1}^{T} + c_{cnt}^{n-1} s_{ij+1k+1}^{T} + c_{wcb}^{n-1} s_{i-1jk-1}^{T}$$
$$+ c_{wct}^{n-1} s_{i-1jk+1}^{T} + c_{ecb}^{n-1} s_{i+1jk-1}^{T} + c_{ect}^{n-1} s_{i+1jk+1}^{T}) \, dt.$$

Fourth-order implicit scheme (FOSIS),

$$T_{ijk}^{n} = c_{wc}^{n} T_{i-1jk}^{n} + c_{ec}^{n} T_{i+1jk}^{n} + c_{sc}^{n} T_{ij-1k}^{n} + c_{nc}^{n} T_{ij+1k}^{n}$$
$$+ c_{bc}^{n} T_{ijk-1}^{n} + c_{tc}^{n} T_{ijk+1}^{n} + c_{wsc}^{n} T_{i-1j-1k}^{n} + c_{wnc}^{n} T_{i-1j+1k}^{n}$$
$$+ c_{esc}^{n} T_{i+1j-1k}^{n} + c_{enc}^{n} T_{i+1j+1k}^{n} + c_{csb}^{n} T_{ij-1k-1}^{n}$$
$$+ c_{cst}^{n} T_{ij-1k+1}^{n} + c_{cnb}^{n} T_{ij+1k-1}^{n} + c_{cnt}^{n} T_{ij+1k+1}^{n}$$
$$+ c_{wcb}^{n} T_{i-1jk-1}^{n} + c_{wct}^{n} T_{i-1jk+1}^{n} + c_{ecb}^{n} T_{i+1jk-1}^{n}$$
$$+ c_{ect}^{n} T_{i+1jk+1}^{n} + c_{c}^{n-1} T_{ijk}^{n-1} + c_{wc}^{n-1} T_{i-1jk}^{n-1}$$
$$+ c_{ec}^{n-1} T_{i+1jk}^{n-1} + c_{sc}^{n-1} T_{ij-1k}^{n-1} + c_{nc}^{n-1} T_{ij+1k}^{n-1}$$
$$+ c_{bc}^{n-1} T_{ijk-1}^{n-1} + c_{tc}^{n-1} T_{ijk+1}^{n-1} + S_{ijk}$$

$$d_0 = \frac{1}{6h_x^2 h_y^2 h_z^2 + 4k_1(h_x^2 h_y^2 + h_y^2 h_z^2 + h_x^2 h_z^2)}$$
$$c_c^{n-1} = [6h_x^2 h_y^2 h_z^2 - 12k_1(h_x^2 h_y^2 + h_y^2 h_z^2 + h_x^2 h_z^2)] e^{k_2} \, d_0$$
$$c_{wc}^{n} = c_{ec}^{n} = [2k_1(h_y^2 h_z^2 - h_x^2 h_y^2 - h_x^2 h_z^2) - h_x^2 h_y^2 h_z^2] \, d_0$$
$$c_{sc}^{n} = c_{nc}^{n} = [2k_1(h_x^2 h_z^2 - h_x^2 h_y^2 - h_y^2 h_z^2) - h_x^2 h_y^2 h_z^2] \, d_0$$
$$c_{bc}^{n} = c_{tc}^{n} = [2k_1(h_x^2 h_y^2 - h_y^2 h_z^2 - h_x^2 h_z^2) - h_x^2 h_y^2 h_z^2] \, d_0$$
$$c_{wsc}^{n} = c_{wnc}^{n} = c_{esc}^{n} = c_{enc}^{n} = k_1 h_z^2 (h_x^2 + h_y^2) \, d_0$$
$$c_{csb}^{n} = c_{cst}^{n} = c_{cnb}^{n} = c_{cnt}^{n} = k_1 h_x^2 (h_y^2 + h_z^2) \, d_0$$
$$c_{wcb}^{n} = c_{wct}^{n} = c_{ecb}^{n} = c_{ect}^{n} = k_1 h_y^2 (h_x^2 + h_z^2) \, d_0$$
$$c_{wc}^{n-1} = c_{ec}^{n-1} = h_y^2 h_z^2 (6k_1 + h_x^2) e^{k_2} \, d_0$$
$$c_{sc}^{n-1} = c_{nc}^{n-1} = h_x^2 h_z^2 (6k_1 + h_y^2) e^{k_2} \, d_0$$
$$c_{bc}^{n-1} = c_{tc}^{n-1} = h_x^2 h_y^2 (6k_1 + h_z^2) e^{k_2} \, d_0$$
$$S_{ijk} = \int_{t_{n-1}}^{t_n} (c_c^{n-1} s_{ijk}^{T} + c_{wc}^{n-1} s_{i-1jk}^{T} + c_{ec}^{n-1} s_{i+1jk}^{T}$$
$$+ c_{sc}^{n-1} s_{ij-1k}^{T} + c_{nc}^{n-1} s_{ij+1k}^{T} + c_{bc}^{n-1} s_{ijk-1}^{T} + c_{tc}^{n-1} s_{ijk+1}^{T}) \, dt.$$

The truncation error of the fourth-order schemes,

$$O\left(\sum_{i=0}^{2} h_x^{2i} k_1^{2-i}\right) + O\left(\sum_{i=0}^{2} h_y^{2i} k_1^{2-i}\right) + O\left(\sum_{i=0}^{2} h_z^{2i} k_1^{2-i}\right)$$
$$+ O\left[\left(\sum_{i=0}^{1} h_x^{2i} k_1^{1-i}\right)\left(\sum_{i=0}^{1} h_y^{2i} k_1^{1-i}\right)\right]$$
$$+ O\left[\left(\sum_{i=0}^{1} h_y^{2i} k_1^{1-i}\right)\left(\sum_{i=0}^{1} h_z^{2i} k_1^{1-i}\right)\right]$$
$$+ O\left[\left(\sum_{i=0}^{1} h_x^{2i} k_1^{1-i}\right)\left(\sum_{i=0}^{1} h_z^{2i} k_1^{1-i}\right)\right] + O(\tau).$$

ACKNOWLEDGMENTS

This research was supported by grants from the Natural Science and Engineering Research Council of Canada.

REFERENCES

1. P. J. Roache, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM, 1972).
2. A. J. Baker, *Finite Element Computational Fluid Mechanics* (Hemisphere, New York, 1983).
3. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, New York, 1980).
4. C. J. Chen, *J. Num. Heat Transfer* **4,** 179 (1981).
5. D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods* (SIAM, Philadelphia, 1977).
6. R. Rogallo and P. Moin, *Annu. Rev. Fluid Mech.* **16,** 99 (1984).
7. M. M. Rai and P. Moin, *J. Comput. Phys.* **96,** 15 (1991).
8. A. T. Patera, *J. Comput. Phys.* **54,** 468 (1984).
9. S. K. Lele, *J. Comput. Phys.* **103,** 16 (1992).
10. R. F. Warming and B. J. Hyett, *J. Comput. Phys.* **14,** 159 (1974).
11. R. I. Issa, *J. Comput. Phys.* **62,** 40 (1985).
12. J. Kim and P. Moin, *J. Comput. Phys.* **59,** 308 (1985).
13. Z. F. Yang, X. Y. Chou, and X. Q. Xu, *J. Hydrodyn. Ser. B* (Engl. ed.) **6,** 113 (1991).
14. E. Hopf, *Commun. Pure Appl. Math.* **3,** 201 (1950).
15. J. D. Cole, *Quart. Appl. Math.* **9,** 225 (1951).
16. C. A. J. Fletcher, *Int. J. Numer. Methods Fluids* **3,** 213 (1983).
17. D. A. Steinman, C. R. Ethier, X. Zhang, and S. R. Karpik, ''Code Testing with an Exact Solution to the 3D Navier–Stokes Equations,'' in *Proceedings, 2nd Annual Conference of the CFD Society of Canada, Toronto, Canada, 1994,* edited by J. J. Gottlieb *et al.,* p. 115.
18. M. Germano, *J. Phys. Fluids* **29,** 1755 (1986).