

# A Two Colorable Fourth Order Compact Difference Scheme and Parallel Iterative Solution of the 3D Convection Diffusion Equation

Jun Zhang\* and Lixin Ge†

Department of Computer Science, University of Kentucky  
773 Anderson Hall, Lexington, KY 40506-0046, USA

Jules Kouatchou‡

NASA Goddard Space Flight Center - Code 931  
Greenbelt, MD 20771, USA

March 12, 2000

## Abstract

A new fourth order compact difference scheme for the three dimensional convection diffusion equation with variable coefficients is presented. The novelty of this new difference scheme is that it only requires 15 grid points and that it can be decoupled with two colors. The entire computational grid can be updated in two parallel subsweeps with the Gauss-Seidel type iterative method. This is compared with the known 19 point fourth order compact difference scheme which requires four colors to decouple the computational grid. Numerical results, with multigrid methods implemented on a shared memory parallel computer, are presented to compare the 15 point and the 19 point fourth order compact schemes.

**Key words:** 3D convection diffusion equation, fourth order compact difference schemes, multigrid method, parallel computation.

## 1 Introduction

The three dimensional (3D) convection diffusion equation with variable coefficients can be written as

$$u_{xx} + u_{yy} + u_{zz} + p(x, y, z)u_x + q(x, y, z)u_y + r(x, y, z)u_z = f(x, y, z), \quad (1)$$

---

\*E-mail: jzhang@cs.uky.edu, URL: <http://www.cs.uky.edu/~jzhang>. The research of this author was supported in part by the U.S. National Science Foundation under the grant CCR-9902022, and in part by the University of Kentucky Center for Computational Sciences.

†E-mail: lixin@csr.uky.edu. The research of this author was supported by the University of Kentucky Center for Computational Sciences.

‡E-mail: kouatchou@gssc.nasa.gov, URL: <http://cesdis1.gsfc.nasa.gov/~kouatch>. This author is also affiliated with Morgan State University and his research was supported by NASA under the grant No. NAGS-3508.

for a specified forcing function  $f$  in a continuous domain  $\Omega$  in 3D space with suitable boundary conditions prescribed on  $\partial\Omega$ , the boundary of  $\Omega$ . Here the coefficients  $p, q, r$ , the forcing function  $f$ , as well as the unknown function  $u$ , are assumed to be continuously differentiable and have the required partial derivatives on  $\Omega$ , where  $\Omega$  is a union of rectangular solids.

Equation (1) is encountered most commonly in the modeling of transport processes, including heat transfer and fluid flows [22, 25], such as the groundwater pollution problems and reservoir displacement problems [2, 3]. It describes the convection and diffusion of various physical quantities, e.g., momentum, heat, material concentrations, etc. Traditional numerical discretization schemes for approximating convection diffusion equations usually employ centered differencing for the second order diffusion terms and some form of upwind differencing for the first order convection terms [23]. For convection dominated problems, basic iterative methods fail to converge when used to solve linear systems resulting from the standard central difference discretization. The computed solutions from the standard upwind difference scheme is only first order accurate. Very fine discretization has to be employed to compute approximate solution with high accuracy, which in turn requires enormous computational power for 3D problems. Thus, the use of high order discretization schemes is one way to obtain high accuracy solution with moderate computational cost. A 19 point fourth order compact finite difference scheme for (1) has been published in [34], based on the truncated Taylor series expansions. Other fourth order compact schemes for the 3D elliptic partial differential equations can be found in [1, 11]. Alternative high accuracy discretization schemes for 2D convection diffusion problems have also been reported in [19, 20, 21]. A parallelizable multigrid method with the 19 point fourth order compact scheme using a four color decoupling of computational grid has been developed by Gupta and Zhang in [15].

In parallel calculations with a Gauss-Seidel type iterative method, a computational grid decoupled with four colors needs four parallel subsweeps to update the entire grid. If the standard second order central difference or upwind difference schemes are used, the computational grid can be decoupled with two colors and updated in two parallel subsweeps. For the 2D convection diffusion equation, it can be shown that a fourth order compact scheme needs the closest 9 grid points, for which the computational grids can be decoupled with minimum 4 colors. It would be advantageous to find a fourth order compact scheme that does not need more than two colors to decouple the computational grid and still offers computed solution with high accuracy. This does not seem to be possible for the 2D convection diffusion problems.

The work of Gupta and Kouatchou [13] shows that it is possible to derive a fourth order compact difference scheme for the 3D Poisson equation that requires only 15 grid points in the approximation scheme. The current work is to derive a 15 point compact difference scheme for the 3D convection diffusion equation with variable coefficients, to design a parallel multigrid solution method to solve the resulting sparse linear systems, and to compare its numerical performance with the existing 19 point compact scheme.

This paper is organized as follows. In Section 2, we present the method for deriving the 15 point compact difference scheme. In Section 3, we discuss the multigrid solution method. Section 4 contains strategies for decoupling the computational grid to extract parallelism in a Gauss-Seidel iteration. Numerical results are presented in Section 5 to compare the solution accuracy and parallel efficiency of the 15 and 19 point compact difference schemes. A brief conclusion is given in Section 6. The stencil coefficients of the 15 point fourth order compact finite difference scheme are listed in Appendix A.

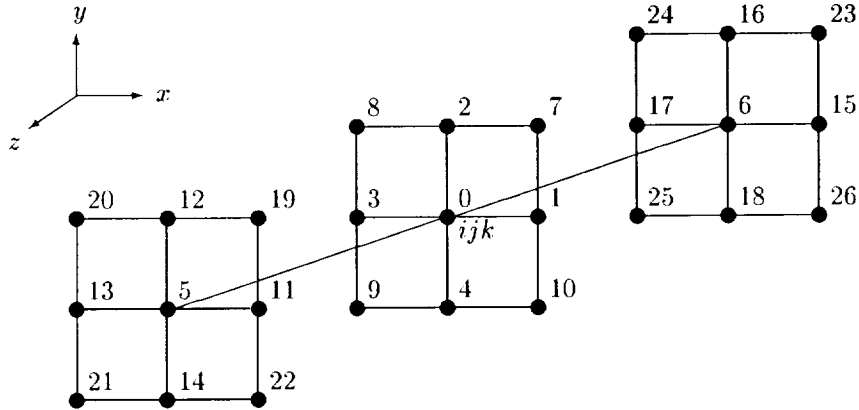


Figure 1: The 27 point stencil of the 3D grid points in a reference cube.

## 2 Description of Derivation Procedure

The discretization is carried out on a uniform 3D grid with a uniform mesh size  $h$ . We use a local coordinate system where the unit cubic grids are labeled as in Figure 1. The approximate value of a function  $u(x, y, z)$  at an interior mesh point  $(i, j, k)$  is denoted by  $u_0$ . The approximate values of its 26 immediate neighboring mesh points are denoted by  $u_l, l = 1, 2, \dots, 26$ , as in Figure 1. The discrete values of  $p_l, q_l, r_l$  and  $f_l$  for  $l = 0, 1, \dots, 26$ , are defined analogously. A 3D finite difference scheme is compact if it only involves at most the 26 nearest neighboring grid points (of the center point) in the approximation formula. For convenience we divide the grid points into three groups:  $A = \{0, 1, 2, 3, 4, 5, 6\}$ ,  $B = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18\}$ ,  $C = \{19, 20, 21, 22, 23, 24, 25, 26\}$ , see Figure 1. Group  $A$  contains the essential grid points needed for a 3D finite difference scheme for (1). Well known examples are the standard central difference scheme and the upwind difference scheme. The 19 point compact scheme in [34] utilizes the grid points in groups  $A$  and  $B$ . The 15 point compact scheme that we will derive later utilizes the grid points in groups  $A$  and  $C$  only.

The approach that we take to develop high order compact difference schemes was advocated by Spitz and Carey [27, 28, 29]. It has been used with a symbolic computation procedure by Ge and Zhang [11] to derive high order compact difference schemes for the 3D linear elliptic partial differential equations with variable coefficients. The entire procedure for deriving high order compact schemes is straightforward and can be done step by step. The truncation errors of the lower order approximations are approximated to higher orders to yield a high order finite difference scheme for the initial approximation. A unique finite difference scheme is given by the symbolic computation package Maple. All that needs to be done is complex substitution and term collection processes, which are especially suitable for symbolic computation packages.

For a sufficiently smooth solution  $u$ , its first and second order partial derivatives with respect to  $x$  at an interior grid point 0 can be approximated by

$$\frac{\partial u}{\partial x} = \delta_x u - \frac{h^2}{6} \frac{\partial^3 u}{\partial x^3} - \frac{h^4}{120} \frac{\partial^5 u}{\partial x^5} + O(h^6), \quad (2)$$

$$\frac{\partial^2 u}{\partial x^2} = \delta_x^2 u - \frac{h^2}{12} \frac{\partial^4 u}{\partial x^4} - \frac{h^4}{360} \frac{\partial^6 u}{\partial x^6} + O(h^6), \quad (3)$$

where  $\delta_x$  and  $\delta_x^2$  are the first and second order central difference operators with respect to  $x$ . Similar partial derivatives with respect to  $y$  and  $z$  can be approximated to  $O(h^6)$  order analogously.

Different finite difference schemes can be derived by substituting the approximation formulas (2), (3), and their counterparts for the  $y$  and  $z$  variables, for the first and second order partial derivatives in (1) and dropping the reminders of appropriate order. As an example, we derive in the following some compact difference schemes up to the fourth order. For this purpose, the  $O(h^4)$  and the higher order truncation error terms in (2) and (3) can be ignored. The substitution yields

$$\begin{aligned} & (\delta_x^2 u + \delta_y^2 u + \delta_z^2 u + p\delta_x u + q\delta_y u + r\delta_z u - f) \\ & - \frac{h^2}{6} \left( \frac{1}{2} \frac{\partial^4}{\partial x^4} + \frac{1}{2} \frac{\partial^4}{\partial y^4} + \frac{1}{2} \frac{\partial^4}{\partial z^4} + p \frac{\partial^3}{\partial x^3} + q \frac{\partial^3}{\partial y^3} + r \frac{\partial^3}{\partial z^3} \right) u + O(h^4) = 0. \end{aligned} \quad (4)$$

The standard 7 point second order central difference scheme is obtained by dropping all the  $O(h^2)$  and the higher order terms in (4). To obtain a difference scheme with a higher order, the  $O(h^2)$  terms in (4) cannot be dropped and has to be approximated further. Since the  $O(h^2)$  terms have an  $h^2$  factor, they can be approximated to the second order accuracy and still yield the fourth order accuracy for the whole approximation scheme.

The key idea for increasing approximation accuracy is that the truncation errors pertaining to the discrete operator may be represented in the final discrete equation. For instance, in the case of the central difference operator for the first order derivative, the  $O(h^2)$  and the higher order truncation error terms can be represented using the original differential equation such that the order of accuracy is increased depending on how many terms are represented. To illustrate this idea, we differentiate (1) with respect to  $x$ ,  $y$ , and  $z$ , in sequence, to obtain

$$\frac{\partial^3 u}{\partial x^3} = \frac{\partial f}{\partial x} - \frac{\partial^3 u}{\partial y^2 \partial x} - \frac{\partial^3 u}{\partial z^2 \partial x} - \frac{\partial p}{\partial x} \frac{\partial u}{\partial x} - p \frac{\partial u^2}{\partial x^2} - \frac{\partial q}{\partial x} \frac{\partial u}{\partial y} - q \frac{\partial^2 u}{\partial y \partial x} - \frac{\partial r}{\partial x} \frac{\partial u}{\partial z} - r \frac{\partial^2 u}{\partial z \partial x}, \quad (5)$$

$$\frac{\partial^3 u}{\partial y^3} = \frac{\partial f}{\partial y} - \frac{\partial^3 u}{\partial y \partial x^2} - \frac{\partial^3 u}{\partial z^2 \partial y} - \frac{\partial p}{\partial y} \frac{\partial u}{\partial x} - p \frac{\partial^2 u}{\partial y \partial x} - \frac{\partial q}{\partial y} \frac{\partial u}{\partial y} - q \frac{\partial^2 u}{\partial y^2} - \frac{\partial r}{\partial y} \frac{\partial u}{\partial z} - r \frac{\partial^2 u}{\partial z \partial y}, \quad (6)$$

$$\frac{\partial^3 u}{\partial z^3} = \frac{\partial f}{\partial z} - \frac{\partial^3 u}{\partial z \partial y^2} - \frac{\partial^3 u}{\partial z \partial x^2} - \frac{\partial p}{\partial z} \frac{\partial u}{\partial x} - p \frac{\partial^2 u}{\partial z \partial x} - \frac{\partial q}{\partial z} \frac{\partial u}{\partial y} - q \frac{\partial^2 u}{\partial z \partial y} - \frac{\partial r}{\partial z} \frac{\partial u}{\partial z} - r \frac{\partial^2 u}{\partial z^2} \quad (7)$$

for the third order partial derivatives, and repeat the process to obtain

$$\begin{aligned} \frac{\partial^4 u}{\partial x^4} &= \frac{\partial^2 f}{\partial x^2} - \frac{\partial u^4}{\partial z^2 \partial x^2} - 2 \frac{\partial p}{\partial x} \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 p}{\partial x^2} \frac{\partial u}{\partial x} - 2 \frac{\partial q}{\partial x} \frac{\partial^2 u}{\partial y \partial x} - \frac{\partial^2 q}{\partial x^2} \frac{\partial u}{\partial y} \\ &\quad - p \frac{\partial^3 u}{\partial x^3} - 2 \frac{\partial r}{\partial x} \frac{\partial^2 u}{\partial z \partial x} - \frac{\partial^2 r}{\partial x^2} \frac{\partial u}{\partial z} - q \frac{\partial^3 u}{\partial y \partial x^2} - r \frac{\partial^3 u}{\partial z \partial x^2} - \frac{\partial^4 u}{\partial y^2 \partial x^2}, \end{aligned} \quad (8)$$

$$\begin{aligned} \frac{\partial^4 u}{\partial y^4} &= \frac{\partial^2 f}{\partial y^2} - 2 \frac{\partial p}{\partial y} \frac{\partial^2 u}{\partial y \partial x} - \frac{\partial^4 u}{\partial y^2 \partial x^2} - \frac{\partial^4 u}{\partial z^2 \partial y^2} - \frac{\partial^2 p}{\partial y^2} \frac{\partial u}{\partial x} - p \frac{\partial^3 u}{\partial y^2 \partial x} \\ &\quad - \frac{\partial^2 q}{\partial y^2} \frac{\partial u}{\partial y} - 2 \frac{\partial q}{\partial y} \frac{\partial^2 u}{\partial y^2} - q \frac{\partial^3 u}{\partial y^3} - \frac{\partial^2 r}{\partial y^2} \frac{\partial u}{\partial z} - 2 \frac{\partial r}{\partial y} \frac{\partial^2 u}{\partial z \partial y} - r \frac{\partial^3 u}{\partial z \partial y^2}, \end{aligned} \quad (9)$$

$$\begin{aligned} \frac{\partial^4 u}{\partial z^4} &= \frac{\partial^2 f}{\partial z^2} - \frac{\partial^4 u}{\partial z^2 \partial y^2} - \frac{\partial^4 u}{\partial z^2 \partial x^2} - \frac{\partial^2 p}{\partial z^2} \frac{\partial u}{\partial x} - 2 \frac{\partial p}{\partial z} \frac{\partial^2 u}{\partial z \partial x} - p \frac{\partial^3 u}{\partial z^2 \partial x} \\ &\quad - \frac{\partial^2 q}{\partial z^2} \frac{\partial u}{\partial y} - 2 \frac{\partial q}{\partial z} \frac{\partial^2 u}{\partial z \partial y} - q \frac{\partial^3 u}{\partial z^2 \partial y} - \frac{\partial^2 r}{\partial z^2} \frac{\partial u}{\partial z} - 2 \frac{\partial r}{\partial z} \frac{\partial^2 u}{\partial z^2} - r \frac{\partial^3 u}{\partial z^3} \end{aligned} \quad (10)$$

for the fourth order partial derivatives. If the right hand sides of these partial derivatives can be approximated to the  $O(h^2)$  order, then the finite difference scheme (4) can be approximated to the  $O(h^4)$  order, per our previous discussion.

The 15 point compact scheme can be derived by considering cross derivatives of the same order together and by utilizing their symmetry relation. This is different from the strategies used

for deriving the 19 point compact scheme, in which cross derivatives are approximated individually [11].

Using the Taylor series expansion for the fourth order cross derivatives, we have

$$\frac{\partial^4 u}{\partial z^2 \partial x^2} + \frac{\partial^4 u}{\partial z^2 \partial y^2} + \frac{\partial^4 u}{\partial y^2 \partial x^2} \approx \frac{1}{2h^4} [u_{19} + u_{23} + u_{26} + u_{22} + u_{20} + u_{21} + u_{24} + u_{25} + 16u_0 - 4(u_1 + u_2 + u_3 + u_4 + u_5 + u_6)]. \quad (11)$$

In addition, exploiting the relations among the third order cross derivatives, we get

$$\frac{\partial^3 u}{\partial y^2 \partial x} + \frac{\partial^3 u}{\partial z^2 \partial x} \approx \frac{1}{4h^3} [(u_{19} + u_{23} + u_{22} + u_{26} - 4u_1) - (u_{20} + u_{21} + u_{24} + u_{25} - 4u_3)], \quad (12)$$

$$\frac{\partial^3 u}{\partial z^2 \partial y} + \frac{\partial^3 u}{\partial z^2 \partial y} \approx \frac{1}{4h^3} [(u_{20} + u_{19} + u_{23} + u_{24} - 4u_2) - (u_{21} + u_{22} + u_{26} + u_{25} - 4u_4)], \quad (13)$$

$$\frac{\partial^3 u}{\partial x^2 \partial z} + \frac{\partial^3 u}{\partial y^2 \partial z} \approx \frac{1}{4h^3} [(u_{19} + u_{20} + u_{21} + u_{22} - 4u_5) - (u_{23} + u_{24} + u_{25} + u_{26} - 4u_6)]. \quad (14)$$

Using Taylor series expansion again, the second order cross derivatives can be approximated as

$$\frac{\partial^2 u}{\partial y \partial x} = \frac{1}{8h^2} [(u_{19} + u_{23} + u_{21} + u_{25}) - (u_{22} + u_{26} + u_{20} + u_{24})] + O(h^2). \quad (15)$$

With the same strategy,  $\frac{\partial^2 u}{\partial z \partial y}$  and  $\frac{\partial^2 u}{\partial z \partial x}$  can be approximated using the grid points in groups *A* and *C* only. Therefore, all the partial derivatives in (5) to (10) can be approximated to  $O(h^2)$  order using the first and second order central difference operators involving the grid points in groups *A* and *C*. Now if we substitute the finite difference expressions of (5) - (10), using (11) - (15) and their counterparts, into (4), we will have a fourth order compact finite difference scheme for (1) defined at the grid points in groups *A* and *C*.

We used the Maple symbolic computation package for the extensive algebraic manipulations. The Maple code is similar to that in [11] used to derive the 19 point compact scheme. The computations were performed on an HP Exemplar supercomputer at the University of Kentucky Center for Computational Sciences. Appendix A lists the resulting formula of the 15 point fourth order compact difference scheme that can be used directly. The coefficients are scaled appropriately so that they have the same scale as those of the 19 point compact scheme of [34]. The scaling is done for the convenience of comparing truncation errors of the two difference schemes.

Next, we compare the leading  $O(h^4)$  terms in the truncation errors of the 15 and 19 point compact schemes. To simplify notations and still make meaningful comparison, we assume that the convection coefficients  $p, q$  and  $r$  are constant in  $\Omega$ . The leading truncation error terms were computed by our Maple code.

The leading truncation error with the  $O(h^4)$  term of the 19 point compact scheme is

$$\begin{aligned} E_{19} = & -\frac{1}{120} \left( p \frac{\partial^5 u}{\partial x^5} + q \frac{\partial^5 u}{\partial y^5} + r \frac{\partial^5 u}{\partial z^5} \right) - \frac{1}{144} \left( p^2 \frac{\partial^4 u}{\partial x^4} + q^2 \frac{\partial^4 u}{\partial y^4} + r^2 \frac{\partial^4 u}{\partial z^4} \right) \\ & - \frac{1}{36} \left( pq \frac{\partial^4 u}{\partial y^3 \partial x} + pq \frac{\partial^4 u}{\partial y \partial x^3} + qr \frac{\partial^4 u}{\partial z \partial y^3} + qr \frac{\partial^4 u}{\partial z^3 \partial y} + pr \frac{\partial^4 u}{\partial z^3 \partial x} + pr \frac{\partial^4 u}{\partial z \partial x^3} \right) \\ & - \frac{1}{72} \left( p \frac{\partial^5 u}{\partial y^4 \partial x} + q \frac{\partial^5 u}{\partial y \partial x^4} + r \frac{\partial^5 u}{\partial z \partial x^4} + p \frac{\partial^5 u}{\partial z^4 \partial x} + q \frac{\partial^5 u}{\partial z^4 \partial y} + r \frac{\partial^5 u}{\partial z \partial y^4} \right) \\ & - \frac{1}{36} \left( p \frac{\partial^5 u}{\partial y^2 \partial x^3} + q \frac{\partial^5 u}{\partial z^2 \partial y^3} + r \frac{\partial^5 u}{\partial z^3 \partial y^2} + p \frac{\partial^5 u}{\partial z^2 \partial x^3} + q \frac{\partial^5 u}{\partial y^3 \partial x^2} + r \frac{\partial^5 u}{\partial z^3 \partial x^2} \right) \end{aligned}$$

$$\begin{aligned}
& -\frac{1}{72} \left( \frac{\partial^6 u}{\partial y^4 \partial x^2} + \frac{\partial^6 u}{\partial y^2 \partial x^4} + \frac{\partial^6 u}{\partial z^2 \partial x^4} + \frac{\partial^6 u}{\partial z^4 \partial x^2} + \frac{\partial^6 u}{\partial z^4 \partial y^2} + \frac{\partial^6 u}{\partial z^2 \partial y^4} \right) \\
& -\frac{1}{360} \left( \frac{\partial^6 u}{\partial x^6} + \frac{\partial^6 u}{\partial y^6} + \frac{\partial^6 u}{\partial z^6} \right).
\end{aligned}$$

Similarly, the leading truncation error with the  $O(h^4)$  term of the 15 point compact scheme is

$$\begin{aligned}
E_{15} = E_{19} - \frac{1}{12} \left( pq \frac{\partial^4 u}{\partial z^2 \partial y \partial x} + pr \frac{\partial^4 u}{\partial z \partial y^2 \partial x} + qr \frac{\partial^4 u}{\partial z \partial y \partial x^2} \right. \\
\left. + p \frac{\partial^5 u}{\partial z^2 \partial y^2 \partial x} + q \frac{\partial^5 u}{\partial z^2 \partial y \partial x^2} + r \frac{\partial^5 u}{\partial z \partial y^2 \partial x^2} + \frac{\partial^6 u}{\partial z^2 \partial y^2 \partial x^2} \right).
\end{aligned}$$

We note that  $E_{15}$  contains all terms of  $E_{19}$ , as well as a few extra terms involving the cross derivatives of all three variables. The most visible observation is that the largest coefficient factor for  $E_{15}$  is 3 times larger than that for  $E_{19}$ , if the magnitude of the convection coefficients is around 1. It follows that, statistically, the errors of the 15 point compact scheme are a factor of 3 larger than the corresponding errors of the 19 point compact scheme. However, if the magnitude of the convection coefficients is large, the coefficient factor of  $E_{15}$  behaves like  $O(\frac{1}{12} \text{Re}^2)$ , while that of  $E_{19}$  behaves like  $O(\frac{1}{36} \text{Re}^2)$ . Here,  $\text{Re}$  denotes the magnitude of the convection coefficients. Hence, the magnitude of the convection coefficients affects the accuracy of both schemes inversely.

### 3 Multigrid Solution Method

Each of the fourth order compact finite difference discretization schemes result in a system of linear equations of the form

$$\mathbf{A}\mathbf{u} = \mathbf{b},$$

where  $\mathbf{A}$  is the coefficient matrix,  $\mathbf{u}$  is the solution vector (unknown), and  $\mathbf{b}$  is the right hand side vector, which includes the forcing term and boundary condition information. Each row of  $\mathbf{A}$  corresponding to an interior node away from the boundary contains 15 (or 19) nonzero entries for the 15 (or 19) point compact schemes. Those rows corresponding to the nodes next to the boundary contain fewer nonzero entries. In general,  $\mathbf{A}$  has 15 (or 19) nonzero diagonals. The linear system has to be solved by some solution technique to yield a solution, i.e.,  $\mathbf{u} = \mathbf{A}^{-1}\mathbf{b}$ . For the current 3D problems, the dimension of  $\mathbf{A}$  is in general very large. Direct solution method based on Gaussian elimination is usually refrained from consideration due to the excessive requirements on computer memory and CPU time. Iterative techniques are viewed as a more viable means in solving 3D problems. The major disadvantage of many iterative techniques is that their convergence may not be guaranteed for solving general sparse linear systems. Even if an iterative method converges for solving a given problem, its convergence rate is usually dependent on many factors, e.g., on the size of the linear system. In the current situation, the size of the linear system is reflected by the mesh size  $h$ .

The coefficient matrix will be used many times in an iterative method. It is usually computed explicitly and stored before it is utilized. On average, each row of the coefficient matrix of the 19 point compact scheme,  $\mathbf{A}_{19}$ , has 4 more nonzero entries than that of the 15 point compact scheme,  $\mathbf{A}_{15}$ . Then  $\mathbf{A}_{15}$  uses at least 21.05% less storage space. This may be an advantage in 3D computations where computer memory is usually a major constraint.

For the 19 point compact scheme, it is possible to show that the convergence of some basic iterative methods, such as Jacobi and Gauss-Seidel methods, is guaranteed if the cell Reynolds

number condition

$$\mathbf{Re}_h = \frac{\max_{(x,y,z) \in \Omega} \{|c|, |d|, |e|\}}{2h} \leq 1$$

is satisfied. Under this condition,  $\mathbf{A}$  may be weakly diagonally dominant, i.e., the inequality  $|\alpha_0| \geq \sum_{l=1}^{18} |\alpha_l|$  holds for all rows of  $\mathbf{A}$ , using the stencil notation in (16). A rigorous convergence proof for the 2D analogous problem with constant coefficients is given in [33] and its generalization to the 3D problem is straightforward. However, even if this cell Reynolds number condition is violated, numerical experiments show that Gauss-Seidel method still converges, regardless of the magnitude of  $\mathbf{Re}_h$  [31, 32, 34]. (A rigorous proof for the 1D case can be found in [35].)

It is well known that classical iterative (relaxation) methods converge slowly for solving large sparse linear systems. Many iterative methods have also been used to solved 2D convection diffusion equations discretized by other schemes [4, 6, 7, 12, 24, 26, 37]. Our choice of linear system solver is multigrid method which has been shown to be very effective for solving discretized elliptic problems [8, 30].

The multigrid method is based on the idea that classical relaxation methods such as Gauss-Seidel iteration strongly damp the oscillatory error components, but converge slowly for smooth error components. Hence, after a few relaxation sweeps, we compute the smooth residual and project it to a coarser grid on which the smooth error components become more oscillatory. Solving the residual equation on a coarse grid, interpolating the error correction back to the fine grid, and adding it to the current approximate solution give the two level method. The multigrid method exploits the idea that the residual equation on the coarse grid has a similar structure as the original problem on the fine grid and the basic idea of the two level method can be applied recursively. Therefore, on the coarse grid, relaxation sweeps are carried out and the smooth residual is projected to a coarser grid. This process may go down to a coarsest grid where a direct solver or several relaxation sweeps are employed to obtain a solution (both approaches are cheap because the size of the linear system on the coarsest grid is small). Then the corrections are interpolated back to finer grids until the process reaches the finest grid and the fine grid approximate solution is corrected. The procedure just described is a simple multigrid V cycle algorithm. A multigrid  $V(\nu_1, \nu_2)$  cycle algorithm is to do  $\nu_1$  relaxation sweeps on a given grid before going to a coarser grid and to do  $\nu_2$  relaxation sweeps after adding the coarse grid correction to the current approximation. For an introduction to the multigrid method and other multigrid cycling algorithms, see Briggs [5] and Wesseling [30].

For the 19 point compact scheme with constant convection coefficients, Fourier smoothing analysis was conducted in [15]. It shows that the Gauss-Seidel relaxation has a smoothing factor that is strictly less than 1, which indicates that the multigrid method with a Gauss-Seidel relaxation will converge regardless of the magnitude of the cell Reynolds number (constant convection coefficients throughout the computational domain).

Similar Fourier smoothing analysis may be conducted to show that the Gauss-Seidel relaxation with the 15 point compact scheme also has a smoothing factor that is smaller than 1 for any constant convection coefficients.

Multigrid techniques for solving 2D and 3D convection diffusion equations, discretized by the fourth order compact schemes, has been studied extensively recently [14, 15, 16, 18, 17, 31, 32, 36]. For more detailed description of the multigrid method with the 19 point compact scheme, see [15]. The 15 point compact scheme can be accommodated straightforwardly by modifying the relevant (relaxation and residual computation) parts in the existing multigrid method.

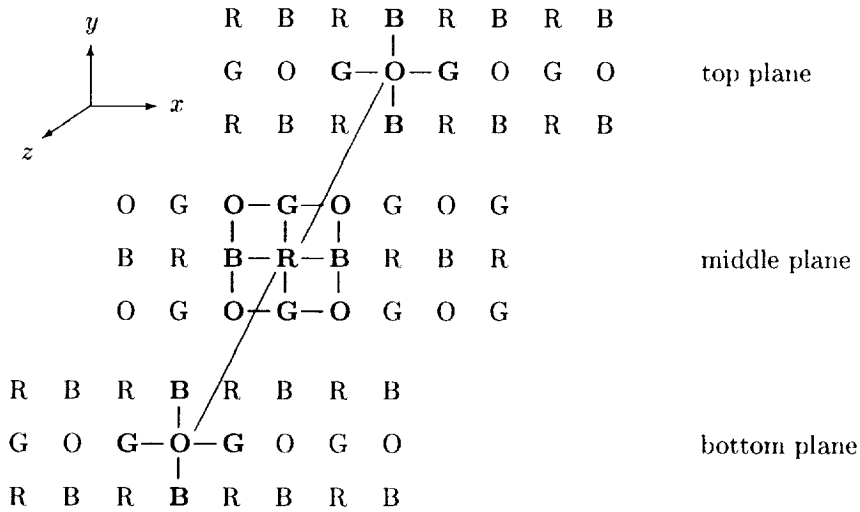


Figure 2: Decoupling the 3D grid points with four colors for Gauss-Seidel relaxation with the 19 point compact scheme.

## 4 Multicoloring Strategies for Parallelism

It is well known that Jacobi iterative method can be fully parallelized. The drawback of the Jacobi method is that when it is used as a smoother in the multigrid method, it usually needs to be damped by a damping factor which is difficult to estimate for most practical problems. Even with a damping factor obtained by trial and error, the smoothing effect of the (damped) Jacobi relaxation is usually poor.

The lexicographic Gauss-Seidel relaxation, which has a better smoothing effect than the Jacobi relaxation, is often used as the smoother in multigrid method. For parallelization and vectorization benefit, we may reorder the grid points by dividing them into several colored groups so that parallel relaxation sweeps can be carried out within each group. In the 2D case, four colors are needed to decouple a 9 point compact scheme. In the 3D case with our 19 point compact scheme, four colors are sufficient to completely decouple the 3D grid points [15]. For simplicity, we assume that (R)ed, (B)lack, (G)reen and (O)range colors are used. For a grid point with a given color, it is necessary that the nearest grid points along the three coordinate directions are marked with different colors. Figure 2 depicts a reference grid point colored with red and its 18 nearest neighboring grid points are colored with black, green and orange. Note that updating a red point needs the values of 2 nearest and 4 next nearest grid points marked with each of the other three colors. For the 19 point compact discretization scheme, we noted above that if the grid is colored by four colors, all grid points with each color can be updated simultaneously on parallel computers and four subsweeps can be carried out to perform a Gauss-Seidel relaxation over the whole grid. This approach is referred to as four color Gauss-Seidel relaxation.

It is interesting that a 3D computational grid with the 15 point compact difference scheme can be decoupled with only two colors [13]. This is shown in Figure 3. It can be seen that a reference R(ed) point is linked to 14 other B(lack) points with the 15 point compact scheme. Hence, two color red-black Gauss-Seidel relaxation with the 15 point compact scheme can update the entire computational grid in two parallel subsweeps, while the four color Gauss-Seidel relaxation needs



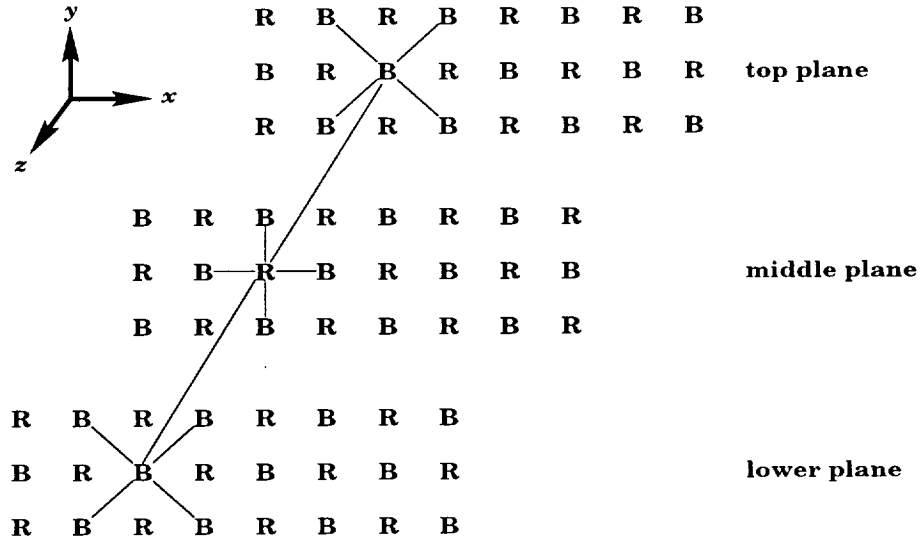


Figure 3: Decoupling the 3D grid points with two colors for Gauss-Seidel relaxation with the 15 point compact scheme.

four parallel subsweeps with the 19 point compact scheme. This difference could be utilized for the advantage of the 15 point compact scheme on parallel computers.

The colored Gauss-Seidel relaxation leads to highly parallelizable solvers. Parallelization is obtained since the grid points with each color are decoupled and all the equations of a single color can be computed independently of the other colors. The computations are performed in a number of parallel operations equal to the number of independent colors. In addition to the gains in parallelization, practical experience showed that better multigrid convergence and smoothing properties are usually obtained with multiple color ordering.

## 5 Numerical Validation

Two test problems are chosen to numerically validate the new 15 point compact scheme and the parallel multigrid solution method. We compare its performance with that of the 19 point compact scheme. For our experiments, we expect the 15 point compact scheme to be slightly less accurate than the 19 point compact scheme but to cost less per multigrid cycle.

The multigrid method described in [15] is used to solve the sparse linear systems arising from the fourth order compact discretizations. Red-black and four color Gauss-Seidel relaxations are used with the 15 and 19 point compact schemes respectively as the multigrid smoothers. The computations are terminated when the mean norm of the difference of the successive approximations, defined as

$$\sqrt{\frac{\sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} |u_{ijk}^{(n)} - u_{ijk}^{(n-1)}|^2}{(N-1)^3}},$$

is smaller than  $10^{-10}$ . Here  $(N-1)^3$  is the number of interior grid points (unknowns) and  $n$  is the number of iterations. The errors reported are the maximum absolute errors over the entire discrete grid points.

Table 1: Comparison of maximum absolute errors for Test 1 with different mesh sizes and Re.

	15 Point Scheme			19 Point Scheme		
	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/32$	$h = 1/64$	$h = 1/128$
Re = 0	4.02(-5)	2.52(-6)	1.57(-7)	1.04(-5)	6.52(-7)	4.07(-8)
Re = 1	3.88(-5)	2.42(-6)	1.51(-7)	1.01(-5)	6.28(-7)	3.93(-8)
Re = 10	1.95(-5)	1.22(-6)	7.62(-8)	6.73(-6)	4.22(-7)	2.63(-8)
Re = 100	1.28(-3)	8.13(-5)	5.11(-6)	2.72(-4)	1.22(-5)	1.08(-6)
Re = 1000	1.27(-2)	9.07(-4)	5.73(-5)	2.60(-3)	1.92(-4)	1.29(-5)

The computations are conducted on an SGI Power Challenge parallel computer with 4 processors and a 512 MB shared memory. The code is written in standard Fortran 77 and is run in double precision. Parallelization is achieved by adding parallel derivatives to the loops in (colored) relaxation and in residual computation subroutines. The interpolation procedure is not parallelized.

**Test 1.** For the first test problem, the following coefficients are specified for (1)

$$p = \text{Re} \sin y \sin z \cos x, \quad q = \text{Re} \sin x \sin z \cos y, \quad r = \text{Re} \sin x \sin y \cos z.$$

The computational domain is the unit cube  $\Omega = (0, 1)^3$ . The constant Re represents the magnitude of the convection coefficients and simulates the Reynolds number in a flow simulation. The Dirichlet boundary conditions and the forcing term  $f$  are set to satisfy the exact solution  $u = \cos(4x + 6y + 8z)$ .

Table 1 shows the maximum absolute error comparison with the 15 and 19 point compact schemes with different mesh sizes and different Re. Note that both schemes are of fourth order accuracy, in the fact that the maximum absolute errors decrease approximately by a factor of 16 when the mesh size is halved. The computed solutions from the 19 point compact scheme are slightly more accurate than the corresponding solutions from the 15 point compact scheme. The difference is about a factor of from slightly larger than 3 to slightly less than 5. This result agrees with our truncation error analysis in Section 2.

The number of multigrid V(1,1) iterations for both schemes is listed in Table 2 with different mesh size and different Re. Multigrid grid independent convergence rate is achieved for both schemes with  $\text{Re} \leq 100$ , although the 19 point compact scheme converged more quickly. The convergence rates of both schemes are inversely affected by the magnitude of Re. Note that the accuracy of the computed solution from both schemes is inversely affected by the magnitude of Re.

The CPU time in seconds with multiple processors is compared in Table 3 with different mesh sizes and a fixed  $\text{Re} = 10$ . For reference, the number of iterations in each case is listed in the last row. We notice that, with one processor (similar to serial computations), the 15 point compact scheme actually took more CPU time to converge (since more iterations are needed for convergence), in spite of the fact that it requires less arithmetic operations in each iteration. However, as more processors are utilized, the 15 point compact scheme is actually faster than the 19 point compact scheme. The difference would be larger should we be able to use more processors.

For large Re, Table 2 indicates a substantial increase in the number of multigrid V(1,1) iterations. To reduce the number of iterations, a multigrid V(2,2) cycle may be used. The test results (CPU seconds) with different mesh sizes and a fixed  $\text{Re} = 1000$  are listed in Table 4. We

Table 2: Number of multigrid V(1,1) iterations for Test 1 with different mesh sizes and Re.

	15 Point Scheme			19 Point Scheme		
	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/32$	$h = 1/64$	$h = 1/128$
Re = 0	12	12	12	10	11	11
Re = 1	12	12	12	10	11	11
Re = 10	12	12	12	11	11	11
Re = 100	29	30	29	26	27	26
Re = 1000	51	57	71	52	56	66

Table 3: Multiprocessor CPU time in seconds (multigrid V(1,1)) for solving Test 1 with different mesh sizes and Re = 10.

	$h = 1/32$		$h = 1/64$		$h = 1/128$	
	15 Point	19 Point	15 Point	19 Point	15 Point	19 Point
1 processor	1.92	1.92	16.49	15.40	139.34	129.31
2 processors	1.15	1.17	9.24	8.99	75.99	71.25
3 processors	0.92	0.87	6.92	7.01	55.88	53.74
4 processors	0.73	0.71	5.63	5.81	43.61	44.02
Iterations	12	11	12	11	12	11

point out that the speedup with 4 processors for the 15 point compact scheme with  $h = 1/128$  is 3.21. It is 2.99 for the 19 point compact scheme. Hence the two colorable 15 point compact scheme appears to be more scalable.

**Test 2.** The convection coefficients of (1) for this test problem are chosen as

$$p = \text{Re } x(1 - y)(1 - 2z), \quad q = \text{Re } y(1 - z)(1 - 2x), \quad r = \text{Re } z(1 - x)(1 - 2z).$$

The Dirichlet boundary conditions and the forcing function are specified to satisfy the exact function  $u = \sin \pi x \sin \pi y \sin \pi z$ . In all calculations for this problem, multigrid V(2,2) cycle iterations are used. Table 5 tabulates the maximum absolute errors of the computed solutions from both the

Table 4: Multiprocessor CPU time in seconds (multigrid V(2,2)) for solving Test 1 with different mesh sizes and Re = 1000.

	$h = 1/32$		$h = 1/64$		$h = 1/128$	
	15 Point	19 Point	15 Point	19 Point	15 Point	19 Point
1 processor	8.29	8.87	82.56	85.19	747.24	703.27
2 processors	4.82	5.23	45.79	49.35	402.78	385.26
3 processors	3.73	3.73	34.31	38.16	288.71	288.50
4 processors	3.09	3.03	27.71	31.47	233.15	235.23
Iterations	31	30	35	35	37	34

Table 5: Comparison of maximum absolute errors for Test 2 with different mesh sizes and Re.

	15 Point Scheme			19 Point Scheme		
	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/32$	$h = 1/64$	$h = 1/128$
Re = 0	3.48(-6)	2.18(-7)	1.36(-8)	9.04(-7)	5.65(-7)	3.53(-9)
Re = 1	3.43(-6)	2.14(-7)	1.34(-8)	8.77(-7)	5.48(-8)	3.43(-9)
Re = 10	3.41(-6)	2.13(-7)	1.33(-8)	1.03(-6)	6.46(-8)	4.04(-9)
Re = 100	2.60(-5)	1.64(-6)	1.03(-7)	1.56(-5)	9.78(-7)	6.12(-8)
Re = 1000	4.42(-4)	2.83(-5)	1.77(-6)	2.97(-4)	1.87(-5)	1.17(-6)

Table 6: Number of multigrid V(2,2) iterations for Test 2 with different mesh sizes and Re.

	15 Point Scheme			19 Point Scheme		
	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/32$	$h = 1/64$	$h = 1/128$
Re = 0	8	8	8	8	8	8
Re = 1	8	8	8	8	8	8
Re = 10	9	9	9	8	9	8
Re = 100	14	14	14	16	16	16
Re = 1000	28	27	32	31	25	29

15 and 19 point compact schemes, when the mesh sizes and Re change. The computed solutions from the 15 point compact scheme are very close to those from the 19 point compact scheme, with a difference of a factor of approximately 3. The difference is actually smaller when Re is large.

Increasing the number of relaxation sweeps at each level of the multigrid algorithm helps reduce the convergence difference between the 15 and 19 point compact schemes. The results in Table 6 show that, in some cases, the 15 point compact scheme converge even faster than the 19 point compact scheme.

In Table 7, we compare the parallel efficiency of the two schemes with different Re and a fixed  $h = 1/128$  when different number of processors are utilized. The parallel run time is affected by the convergence rates of the multigrid iterations, which in turn are affected by the magnitude of Re for both schemes. If the convergence rates are the same, the 15 point compact scheme is faster.

## 6 Conclusion

We derived a new fourth order compact finite difference scheme for the 3D convection diffusion equation with which the computational grid can be decoupled with only two colors, when Gauss-Seidel type iterative method is used to solve the resulting sparse linear systems. A parallel implementation of a multigrid method is discussed. Numerical experiments are conducted to compare the new compact difference scheme with the existing 19 point compact difference scheme.

Our studies show that the 15 point compact scheme may have the advantage of delivering fast solution on parallel computers. Our tests show that with 4 processors, multigrid method with the 15 point compact scheme needs less run time than with the 19 point compact scheme, if the multigrid convergence rates do not differ a lot. The richer parallelism of the 15 point compact

Table 7: Multiprocessor CPU time in seconds (multigrid V(2,2)) for solving Test 2 with different mesh size  $h = 1/128$  and different Re.

	Re = 10		Re = 100		Re = 1000	
	15 Point	19 Point	15 Point	19 Point	15 Point	19 Point
1 processor	182.86	188.20	305.19	333.23	648.63	604.37
2 processors	99.04	102.19	165.24	181.84	352.26	328.10
3 processors	72.32	76.31	112.46	136.18	256.83	245.26
4 processors	56.45	61.46	94.52	108.74	199.67	197.46
Iterations	9	9	14	16	32	29

scheme will be more advantageous if more processors are employed. Another noticeable advantage of the 15 point compact scheme is that it uses 21.05% less memory than the 19 point compact scheme and requires less computational time per multigrid cycle.

For certain convection diffusion problems with boundary layers, uniform discretization may not be suitable. However, grid transformation can be used to transform a graded grid into a uniform grid, on which the fourth order compact scheme can be applied, see [10, 9, 38] for details.

## References

- [1] U. Ananthakrishnaiah, R. Monahar, and J. W. Stephenson. Fourth-order finite difference methods for three-dimensional general linear elliptic problems with variable coefficients. *Numer. Methods Partial Differential Equations*, 3:229–240, 1987.
- [2] K. Aziz and A. Settari. *Petroleum Reservoir Simulation*. Applied Science Publisher Ltd., London, 1979.
- [3] J. Bear. *Dynamics of Fluids in Porous Media*. American Elsevier Publishing Company, New York, 1972.
- [4] M. Bhuruth and D. J. Evans. Block alternating group explicit preconditioning (BLAGE) for a class of fourth order difference schemes. *Intern. J. Comput. Math.*, 63:121–136, 1997.
- [5] W. L. Briggs. *A Multigrid Tutorial*. SIAM, Philadelphia, PA, 1987.
- [6] M. P. Chernesky. On preconditioned Krylov subspace methods for discrete convection-diffusion problems. *Numer. Methods Partial Differential Equations*, 13:321–330, 1997.
- [7] R. C. Y. Chin, T. A. Manteuffel, and J. de Pillis. ADI as a preconditioning for solving the convection-diffusion equation. *SIAM J. Sci. Statist. Comput.*, 5(2):281–299, 1984.
- [8] P. M. de Zeeuw. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *J. Comput. Appl. Math.*, 33:1–25, 1990.
- [9] L. Ge and J. Zhang. Accuracy, robustness, and efficiency comparison in iterative computation of convection diffusion equation with boundary layers. Technical Report No. 291-99, Department of Computer Science, University of Kentucky, Lexington, KY, 1999. Accepted for publication in *Numer. Methods Partial Differential Equations*.

- [10] L. Ge and J. Zhang. High accuracy iterative solution of convection diffusion equation with boundary layers on nonuniform grids. Technical Report No. 288-99, Department of Computer Science, University of Kentucky, Lexington, KY, 1999. Submitted to *J. Comput. Phys.*
- [11] L. Ge and J. Zhang. Symbolic computation of high order compact difference schemes for three dimensional linear elliptic partial differential equations with variable coefficients. Technical Report No. 299-00, Department of Computer Science, University of Kentucky, Lexington, KY, 2000. Submitted for publication in *J. Comput. Appl. Math.*
- [12] C. I. Goldstein. Preconditioning convection dominated convection-diffusion problems. *Internat. J. Numer. Methods Heat Fluid Flow*, 5:99–119, 1995.
- [13] M. M. Gupta and J. Kouatchou. Symbolic derivation of finite difference approximations for the three dimensional Poisson equation. *Numer. Methods Partial Differential Equations*, 18(5):593–606, 1998.
- [14] M. M. Gupta, J. Kouatchou, and J. Zhang. A compact multigrid solver for convection-diffusion equations. *J. Comput. Phys.*, 132:123–129, 1997.
- [15] M. M. Gupta and J. Zhang. High accuracy multigrid solution of the 3D convection-diffusion equation. *Appl. Math. Comput.* to appear.
- [16] J. Kouatchou. Asymptotic stability of a 9-point multigrid algorithm for convection-diffusion equations. *Electron. Trans. Numer. Anal.*, 6:153–161, 1997.
- [17] J. Kouatchou. A dynamic injection operator in a multigrid solution of the convection-diffusion equation. *Internat. J. Numer. Methods Fluids*, 26:1205–1216, 1998.
- [18] J. Kouatchou. *High-Order Multigrid Techniques for Partial Differential Equations*. PhD thesis, The George Washington University, Washington, DC, 1998.
- [19] V. Kriventsev and H. Ninokata. An effective, locally exact finite-difference scheme for convection-diffusion problems. *Numer. Heat Transfer, Part B*, 36:183–205, 1999.
- [20] D. Liang and W. Zhao. A high-order upwind method for the convection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 147:105–115, 1997.
- [21] B. J. Noye and H. H. Tan. A third-order semi-implicit finite difference method for solving the one-dimensional convection-diffusion equation. *Internat. J. Numer. Methods Engrg.*, 26(7):1615–1629, 1988.
- [22] S. V. Pakantar. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill, New York, 1980.
- [23] P. J. Roache. *Computational Fluid Dynamics*. Hermosa, Albuquerque, NM, 1976.
- [24] S.-C. Sheen and J.-L. Wu. Preconditioning techniques for the BiCGSTAB algorithm used in convection-diffusion problems. *Numer. Heat Transfer, Part B*, 34:241–256, 1998.
- [25] T. M. Shih. *Numerical Heat Transfer*. Hemisphere, Washington, DC, 1984.
- [26] W. Shyy and J. Burke. Study of iterative characteristics of convective-diffusive and conjugate heat transfer problems. *Numer. Heat Transfer, Part B*, 26:21–37, 1994.
- [27] W. F. Spotz. *High-Order Compact Finite Difference Schemes for Computational Mechanics*. PhD thesis, University of Texas at Austin, Austin, TX, 1995.

- [28] W. F. Spitz and G. F. Carey. High-order compact scheme for the steady stream-function vorticity equations. *Internat. J. Numer. Methods Engrg.*, 38:3497–3512, 1995.
- [29] W. F. Spitz and G. F. Carey. A high-order compact formulation for the 3D Poisson equation. *Numer. Methods Partial Differential Equations*, 12:235–243, 1996.
- [30] P. Wesseling. *An Introduction to Multigrid Methods*. Wiley, Chichester, England, 1992.
- [31] J. Zhang. Accelerated high accuracy multigrid solution of the convection-diffusion equation with high Reynolds number. *Numer. Methods Partial Differential Equations*, 77(1):73–89, 1997.
- [32] J. Zhang. *Multigrid Acceleration Techniques and Applications to the Numerical Solution of Partial Differential Equations*. PhD thesis, The George Washington University, Washington, DC, 1997.
- [33] J. Zhang. On convergence of iterative methods with a fourth-order compact scheme. *Appl. Math. Lett.*, 10(2):49–55, 1997.
- [34] J. Zhang. An explicit fourth-order compact finite difference scheme for three dimensional convection-diffusion equation. *Commun. Numer. Methods Engrg.*, 14:209–218, 1998.
- [35] J. Zhang. On convergence and performance of iterative methods with fourth-order compact schemes. *Numer. Methods Partial Differential Equations*, 14:262–283, 1998.
- [36] J. Zhang. A note on an accelerated high accuracy multigrid solution of the convection-diffusion equation with high Reynolds number. *Numer. Methods Partial Differential Equations*, 16(1):1–10, 2000.
- [37] J. Zhang. Preconditioned iterative methods and finite difference schemes for convection-diffusion. *Appl. Math. Comput*, 109(1):11–30, 2000.
- [38] J. Zhang, L. Ge, and M. M. Gupta. Fourth order compact difference scheme for 3D convection diffusion equation with boundary layers on nonuniform grids. Technical Report No. 298-00, Department of Computer Science, University of Kentucky, Lexington, KY, 2000. Submitted to *Numer. Heat Transfer, II*.

## Appendixes

### A The 15 Point Fourth Order Compact Difference Scheme

The 15 point fourth order compact finite difference scheme for Equation (1) can be written as

$$\sum_{l=0}^6 \alpha_l u_l + \sum_{l=19}^{26} \alpha_l u_l = F_0, \quad (16)$$

where the stencil coefficients are given as:

$$\begin{aligned} \alpha_0 &= -[28 + (p_1 - p_3 + q_2 - q_4 + r_5 - r_6)h + (p_0^2 + q_0^2 + r_0^2)h^2], \\ \alpha_1 &= 4 + \frac{1}{4}(2p_0 + 3p_1 + p_2 - p_3 + p_4 + p_5 + p_6)h \end{aligned}$$

$$\begin{aligned}
& + \frac{1}{8}[4p_0^2 + p_0(p_1 - p_3) + q_0(p_2 - p_4) + r_0(p_5 - p_6)]h^2, \\
\alpha_2 & = 4 + \frac{1}{4}(2q_0 + q_1 + 3q_2 + q_3 - q_4 + q_5 + q_6)h \\
& + \frac{1}{8}[4q_0^2 + p_0(q_1 - q_3) + q_0(q_2 - q_4) + r_0(q_5 - q_6)]h^2, \\
\alpha_3 & = 4 - \frac{1}{4}(2p_0 - p_1 + p_2 + 3p_3 + p_4 + p_5 + p_6)h \\
& + \frac{1}{8}[4p_0^2 - p_0(p_1 - p_3) - q_0(p_2 - p_4) - r_0(p_5 - p_6)]h^2, \\
\alpha_4 & = 4 - \frac{1}{4}(2q_0 + q_1 - q_2 + q_3 + 3q_4 + q_5 + q_6)h \\
& + \frac{1}{8}[4q_0^2 - p_0(q_1 - q_3) - q_0(q_2 - q_4) - r_0(q_5 - q_6)]h^2, \\
\alpha_5 & = 4 + \frac{1}{4}(2r_0 + r_1 + r_2 + r_3 + r_4 + 3r_5 - q_6)h \\
& + \frac{1}{8}[4r_0^2 + p_0(r_1 - r_3) + q_0(r_2 - r_4) + r_0(r_5 - r_6)]h^2, \\
\alpha_6 & = 4 - \frac{1}{4}(2r_0 + r_1 + r_2 + r_3 + r_4 - r_5 + 3q_6)h \\
& + \frac{1}{8}[4r_0^2 - p_0(r_1 - r_3) - q_0(r_2 - r_4) - r_0(r_5 - r_6)]h^2, \\
\alpha_{19} & = \frac{1}{2} + \frac{1}{16}[4(p_0 + q_0 + r_0) + p_2 - p_4 + p_5 - p_6 + q_1 - q_3 \\
& + q_5 - q_6 + r_1 - r_3 + r_2 - r_4]h + \frac{1}{8}(p_0q_0 + p_0r_0 + q_0r_0)h^2, \\
\alpha_{20} & = \frac{1}{2} - \frac{1}{16}[4(p_0 - q_0 - r_0) + p_2 - p_4 + p_5 - p_6 + q_1 - q_3 \\
& - q_5 + q_6 + r_1 - r_3 - r_2 + r_4]h - \frac{1}{8}(p_0q_0 + p_0r_0 - q_0r_0)h^2, \\
\alpha_{21} & = \frac{1}{2} - \frac{1}{16}[4(p_0 + q_0 - r_0) - p_2 + p_4 + p_5 - p_6 - q_1 + q_3 \\
& + q_5 - q_6 + r_1 - r_3 + r_2 - r_4]h + \frac{1}{8}(p_0q_0 - p_0r_0 - q_0r_0)h^2, \\
\alpha_{22} & = \frac{1}{2} + \frac{1}{16}[4(p_0 - q_0 + r_0) - p_2 + p_4 + p_5 - p_6 - q_1 + q_3 \\
& - q_5 + q_6 + r_1 - r_3 - r_2 + r_4]h - \frac{1}{8}(p_0q_0 - p_0r_0 + q_0r_0)h^2, \\
\alpha_{23} & = \frac{1}{2} + \frac{1}{16}[4(p_0 + q_0 - r_0) + p_2 - p_4 - p_5 + p_6 + q_1 - q_3 \\
& - q_5 + q_6 - r_1 + r_3 - r_2 + r_4]h + \frac{1}{8}(p_0q_0 - p_0r_0 - q_0r_0)h^2, \\
\alpha_{24} & = \frac{1}{2} - \frac{1}{16}[4(p_0 - q_0 + r_0) + p_2 - p_4 - p_5 + p_6 + q_1 - q_3 \\
& + q_5 - q_6 - r_1 + r_3 + r_2 - r_4]h - \frac{1}{8}(p_0q_0 - p_0r_0 + q_0r_0)h^2, \\
\alpha_{25} & = \frac{1}{2} - \frac{1}{16}[4(p_0 + q_0 + r_0) - p_2 + p_4 - p_5 + p_6 - q_1 + q_3 \\
& - q_5 + q_6 - r_1 + r_3 - r_2 + r_4]h + \frac{1}{8}(p_0q_0 + p_0r_0 + q_0r_0)h^2, \\
\alpha_{26} & = \frac{1}{2} + \frac{1}{16}[4(p_0 - q_0 - r_0) - p_2 + p_4 - p_5 + p_6 - q_1 + q_3
\end{aligned}$$



$$\begin{aligned}
& + q_5 - q_6 - r_1 + r_3 + r_2 - r_4]h - \frac{1}{8}(p_0q_0 + p_0r_0 - q_0r_0)h^2, \\
E_0 & = \frac{1}{2}(6f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6)h^2 \\
& + \frac{1}{4}[p_0(f_1 - f_3) + q_0(f_2 - f_4) + r_0(f_5 - f_6)]h^3.
\end{aligned}$$

