Numerical Analysis of The Convection Diffusion Equation Using Multigrid Method

Fazal Haq[1] Shakoor Muhammad, Yousaf Khan, Taimur Ali, Wajid Khan

1dr.fazalhaq@uoas.edu.pk

[2]shakoor@awkum.edu.pk, [2]yousafkhan8262@gmail.com,
[2]taimurali999ali@gmail.com, [2]wajidkhanlr@gmail.com

[1]Department of Mathematics The University of Agriculture Swat , KP Pakistan
[2]Department of Mathematics, Abdul Wali Khan University, Mardan 23200, Pakistan

Abstract

This article is concerned with the development of fast iterative methods for the numerical analysis of linear partial differential equation. In this work convection–diffusion equation with a small parameter multiplying with diffusion term is considered. A fourth-order compact difference scheme with uniform mesh points is employed to discretize a 2–dimmensional convection-diffusion equation.

Finally, we compared multigrid method with fourth order compact difference scheme with the standard second order central difference scheme. Numerical results show the efficiency of this method.

Keywords: Multigrid Method, Convection-Diffusion equation, Fourth order compact scheme.
Corresponding Author Email: shakoor@awkum.edu.pk

1. Introduction

Convection is a physical way by which property is transfer according to its flow, on the other hand diffusion is usually investigated phenomena play an important role in scientific modeling problems in which heat transfer and fluid flow take place. There are large numbers of physical problems in which convection and diffusion are involved, such as in the modeling of semi-conductors, transport of air and ground water pollutant etc, [12].

There are two terms involved in convection-diffusion equation, first is convection and second is diffusion. Convection-diffusion equation is widely used in modeling and simulations of various complex phenomena in science and engineering. As it has many applications, so we discuss techniques for the numerical solution of this equation.

This problem is concerned with convection-diffusion equation is:

$$\begin{cases} -\frac{1}{pe}\Delta u + \frac{1}{2}[v1ux + v2uy + (v1u)x + (v2uy)] = f, \\ vi = vi(x,y), \quad i = 1,2, \quad u = u(x,y), \quad f = f(x,y), \\ (x,y) \in \Omega = [0,1] \times [0,1], \\ u \big| \partial\Omega = 0. \end{cases} \tag{1}$$

where pe denote the peclet numbers, $v = (v1, v2)$ show velocity vector, u is the solution of the equation (1) and Ω is a rectangular domain with suitable boundary conditions. We discretized Ω with uniform mesh sizes, Δx and Δy in the x and y directions respectively.

Consider a special case where $v1 = 1$ & $v2 = -1$, in equation (1). Equation (1) can be extended to any values of v_1, v_2 then we have :

$$-\frac{1}{pe}\left(u_{xx} + u_{yy}\right) + u_x - u_y = f(x,y). \tag{2}$$

Mathematical models which contain a combination of convection and diffusion procedure are widely used in all sciences [14, 1]. Research for that procedure is very important but complicated, when convection is dominant [7]. Convection diffusion equation is a second order partial differential equation. We use fourth order compact difference scheme for approximation of convection diffusion equation, after the implementation of this scheme we get system of linear equations. Matrix of these equations is diagonally dominant.

The numerical solution of convection-diffusion equation has been developed by using different approaches such as second order upwind difference scheme and five point central difference scheme. But the upwind difference scheme cannot give desirable results, because it frequently prevents oscillation and another disadvantage is that it reduces accuracy to the $O(\Delta x)$ [17]. The central difference scheme has a truncation error of order $O(\Delta x^2)$, its results are less accurate for large coefficient of highest order derivatives [21]. To obtain more accurate solution for convection diffusion equation, more complex procedures are required.

In this work multigrid method with fourth order compact approximation is used to solve the 2D convection-diffusion equation on uniform grids. We use multi-grid method with fourth order compact scheme, because its numerical results are more accurate and efficient then the second order scheme. In this work a multi-grid method is developed for the solution of 2D convection diffusion equation based on fourth order compact scheme. The method is faster than any other direct or iterative solving methods. Multi-grid method using Gauss-Seidel is considered to solve the linear system of equations. Multi-grid method works by decomposing a problem into separate length scales, and using an iterative solver method that optimizes error reduction for that length scale.

2. Fourth order compact difference scheme

We discretize now the two dimensional convection diffusion equation in both x and y directions with uniform mesh Δx and Δy respectively, using second order central differences . We can write equation (1) as:

$$(\delta_x^2 u_{i,j} + \delta_y^2 u_{i,j}) - p(\delta_x u_{i,j} - \delta_y u_{i,j}) = -pf_{i,j}, \tag{2}$$

where $u_{i,j} = u(xi, yj)$ and $f_{i,j} = f(xi, yj)$.

The fourth order compact approximation for 1D convection-diffusion equation can be written as done in [15] :

$$\delta_x^2 u_i = \left(\mathbf{1} + \frac{\Delta \boldsymbol{x}^2}{\mathbf{12}} \boldsymbol{\delta}_x^2\right) f_i + O(\Delta x^4)\ , \tag{3}$$

equation (3) can also be written as:

$$\left(\mathbf{1} + \frac{\Delta \boldsymbol{x}^2}{\mathbf{12}} \boldsymbol{\delta}_x^2\right)^{-1} \delta_x^2 u_i = f_i + O(\Delta x^4),$$

and

$$\left(\mathbf{1} + \frac{\Delta \boldsymbol{y}^2}{\mathbf{12}} \boldsymbol{\delta}_y^2\right)^{-1} \delta_y^2 u_i = f_i + O(\Delta y^4).$$

Equation (2) becomes

$$\left(\mathbf{1} + \frac{\Delta x^2}{\mathbf{12}} \delta_x^2\right)^{-\mathbf{1}} \boldsymbol{\delta}_x^2 \boldsymbol{u}_{i,j} + \left(\mathbf{1} + \frac{\Delta y^2}{\mathbf{12}} \boldsymbol{\delta}_y^2\right)^{-\mathbf{1}} \boldsymbol{\delta}_y^2 \boldsymbol{u}_{i,j} - \boldsymbol{p}\left(\boldsymbol{\delta}_x \boldsymbol{u}_{i,j} - \boldsymbol{\delta}_y \boldsymbol{u}_{i,j}\right) = -\boldsymbol{p}\boldsymbol{f}_{i,j} + \boldsymbol{O}(\Delta \boldsymbol{x}^4), \tag{4}$$

the equation (4) can be written as :

$$\left(\mathbf{1}+\frac{\Delta y^2}{12}\boldsymbol{\delta}_y^2\right)\boldsymbol{\delta}_x^2\boldsymbol{u}_{i,j}+\left(\mathbf{1}+\frac{\Delta x^2}{12}\boldsymbol{\delta}_x^2\right)\boldsymbol{\delta}_y^2\boldsymbol{u}_{i,j}-\boldsymbol{p}\left(\mathbf{1}+\frac{\Delta x^2}{12}\boldsymbol{\delta}_x^2+\frac{\Delta y^2}{12}\boldsymbol{\delta}_y^2\right)\boldsymbol{\delta}_x\boldsymbol{u}_{i,j}+\boldsymbol{p}\left(\mathbf{1}+\frac{\Delta x^2}{12}\boldsymbol{\delta}_x^2+\frac{\Delta y^2}{12}\boldsymbol{\delta}_y^2\right)\boldsymbol{\delta}_y\boldsymbol{u}_{i,j}$$

$$=-\boldsymbol{p}\left(\mathbf{1}+\frac{\Delta x^2}{12}\boldsymbol{\delta}_x^2\right)\left(\mathbf{1}+\frac{\Delta y^2}{12}\boldsymbol{\delta}_y^2\right)\boldsymbol{f}_{i,j}+\boldsymbol{O}(\Delta\boldsymbol{x}^4)$$

$$=-\boldsymbol{p}\left(\mathbf{1}+\frac{\mathbf{1}}{\mathbf{12}}\left(\Delta\boldsymbol{x}^2\boldsymbol{\delta}_x^2+\Delta\boldsymbol{y}^2\boldsymbol{\delta}_y^2\right)\right)\boldsymbol{f}_{i,j}+\boldsymbol{O}(\Delta\boldsymbol{x}^4). \tag{5}$$

Now we simplify the fourth-order compact approximation scheme and neglecting the order $O(\Delta x^4)$ terms, then equation (5) can be written as

$$\underbrace{(\boldsymbol{\delta}_x^2+\boldsymbol{\delta}_y^2)\boldsymbol{u}_{i,j}}_{l_1}+\frac{\mathbf{1}}{\mathbf{12}}\underbrace{(\Delta^2)\boldsymbol{\delta}_x^2\boldsymbol{\delta}_y^2}_{l_2}-\underbrace{\boldsymbol{p}\left(\mathbf{1}+\frac{\Delta x^2\delta^2 x}{12}+\frac{\Delta y^2\delta^2 y}{12}\right)\boldsymbol{\delta}_x\boldsymbol{u}_{i,j}}_{l_3}+\underbrace{\boldsymbol{p}\left(\mathbf{1}+\frac{\Delta x^2}{12}\boldsymbol{\delta}_x^2+\frac{\Delta y^2}{12}\boldsymbol{\delta}_y^2\right)\boldsymbol{\delta}_y\boldsymbol{u}_{i,j}.}_{l_4}$$

$$=\underbrace{-p\left(\boldsymbol{f}_{i,j}+\frac{\mathbf{1}}{\mathbf{12}}(\Delta\boldsymbol{x}^2\boldsymbol{\delta}_x^2+\Delta\boldsymbol{y}^2\boldsymbol{\delta}_y^2)\boldsymbol{f}_{i,j}\right)}_{l_5}+O(\Delta x^4).$$

$$l_1+l_2+l_3+l_4=l_5 \tag{6}$$

$$l_1=(\boldsymbol{\delta}_x^2+\boldsymbol{\delta}_y^2)\boldsymbol{u}_{i,j}=\frac{\boldsymbol{u}_{i+1,j}-2\boldsymbol{u}_{i,j}+\boldsymbol{u}_{i-1,j}}{\Delta\boldsymbol{x}^2}+\frac{\boldsymbol{u}_{i,j+1}-2\boldsymbol{u}_{i,j}+\boldsymbol{u}_{i,j-1}}{\Delta\boldsymbol{y}^2},$$

generally, we denote the mesh ratio by $\boldsymbol{\gamma}=\frac{\Delta x}{\Delta y}$,

$$l_2=\frac{\mathbf{1}}{\mathbf{12}}(\boldsymbol{\gamma}\Delta\boldsymbol{y}^2+\Delta\boldsymbol{y}^2)\boldsymbol{\delta}_x^2\boldsymbol{\delta}_y^2\boldsymbol{u}_{i,j}=\left(\frac{1+\gamma^2}{12}\right)\Delta\boldsymbol{y}^2\boldsymbol{\delta}_x^2\boldsymbol{\delta}_y^2\boldsymbol{u}_{i,j}$$

$$=\left(\frac{1+\gamma^2}{12}\right)\Delta\boldsymbol{y}^2\left(\boldsymbol{\delta}_x^2\left(\frac{\boldsymbol{u}_{i,j+1}-2\boldsymbol{u}_{i,j}+\boldsymbol{u}_{i,j-1}}{\Delta\boldsymbol{y}^2}\right)\right)$$

$$=\left(\frac{1+\gamma^2}{12}\right)\left(\boldsymbol{\delta}_x^2\boldsymbol{u}_{i,j+1}-2\boldsymbol{\delta}_x^2\boldsymbol{u}_{i,j}+\boldsymbol{\delta}_x^2\boldsymbol{u}_{i,j-1}\right)$$

$$=\left(\frac{1+\gamma^2}{12}\right)\left(\frac{\boldsymbol{u}_{i+1,j+1}-2\boldsymbol{u}_{i,j+1}+\boldsymbol{u}_{i-1,j+1}}{\Delta\boldsymbol{x}^2}-2\frac{\boldsymbol{u}_{i+1,j}-2\boldsymbol{u}_{i,j}+\boldsymbol{u}_{i-1,j}}{\Delta\boldsymbol{x}^2}+\frac{\boldsymbol{u}_{i+1,j-1}-2\boldsymbol{u}_{i,j-1}+\boldsymbol{u}_{i-1,j-1}}{\Delta\boldsymbol{x}^2}\right).$$

$$l_2=\left(\frac{1+\gamma^2}{12}\right)\left(\frac{\boldsymbol{u}_{i+1,j+1}+\boldsymbol{u}_{i-1,j+1}+\boldsymbol{u}_{i+1,j-1}+\boldsymbol{u}_{i-1,j-1}}{\Delta\boldsymbol{x}^2}-2\frac{\boldsymbol{u}_{i+1,j}+\boldsymbol{u}_{i-1,j}}{\Delta\boldsymbol{x}^2}-2\frac{\boldsymbol{u}_{i,j+1}+\boldsymbol{u}_{i,j-1}}{\Delta\boldsymbol{x}^2}+4\frac{\boldsymbol{u}_{i,j}}{\Delta\boldsymbol{x}^2}\right).$$

Now

$$l_3=-\boldsymbol{p}\left(\mathbf{1}+\frac{\Delta x^2\delta^2 x}{12}+\frac{\Delta y^2\delta^2 y}{12}\right)\boldsymbol{\delta}_x\boldsymbol{u}_{i,j},$$

expanding by 2nd order difference formula, we get

$$=-\boldsymbol{p}\left(\mathbf{1}+\frac{\Delta x^2\delta^2 x}{12}+\frac{\Delta y^2\delta^2 y}{12}\right)\left(\frac{\boldsymbol{u}_{i+1,j}-\boldsymbol{u}_{i-1,j}}{2\Delta\boldsymbol{x}}\right),$$

by discretization, we get the following:

$$= -\frac{12p}{24\Delta x}\left(u_{i+1,j} - u_{i-1,j}\right) - \frac{p\Delta x^2\delta^2 x\left(\boldsymbol{u_{i+1,j}}-\boldsymbol{u_{i-1,j}}\right)}{24\Delta x} - \frac{p\Delta^2 y\delta^2 y\left(\boldsymbol{u_{i+1,j}}-\boldsymbol{u_{i-1,j}}\right)}{24\Delta x}.$$

Putting $\Delta x = \Delta y$, then we have:

$$l_3 = -\frac{p}{24\Delta x}\left(9u_{i+1,j} - 9u_{i-1,j} + u_{i+1,j+1} + u_{i+1,j-1} - u_{i-1,j+1} - u_{i-1,j-1}\right).$$

and

$$l_4 = p\left(1 + \frac{\Delta x^2\delta^2 x}{12} + \frac{\Delta y^2\delta^2 y}{12}\right)\left(\frac{\boldsymbol{u_{i,j+1}}-\boldsymbol{u_{i,j-1}}}{2\Delta y}\right)$$

$$l_4 = \frac{p}{24\Delta y}\left(9u_{i,j+1} - 9u_{i,j-1} + u_{i+1,j+1} + u_{i-1,j+1} - u_{i+1,j-1} - u_{i-1,j-1}\right).$$

Now the right hand side (l_5) of (6) can be simplified as:

$$l_5 = -p(f_{i,j} + \frac{1}{12}\Delta y^2\left(\gamma^2\delta_x^2 + \delta_y^2\right)f_{i,j}\big) + O(\Delta^4)$$

$$= -p(f_{i,j} + \frac{1}{12}\left(\delta_x^2 f_{i,j} + \delta_y^2 f_{i,j}\right)\big) .$$

Using second order central difference approximation, we can write as:

$$= -pf_{i,j} - \frac{p}{12}\left(f_{i+1,j} - 2f_{i,j} + f_{i-1,j} + f_{i,j+1} - 2f_{i,j} + f_{i,j+1}\right)$$

$$= \frac{-p}{12}\left(12f_{i,j} + \left(f_{i+1,j} - 4f_{i,j} + f_{i-1,j} + f_{i,j-1} + f_{i,j+1}\right)\right).$$

Or

$$l_5 = \frac{-p}{12}\left(f_{i+1,j} + 8f_{i,j} + f_{i-1,j} + f_{i,j-1} + f_{i,j+1}\right).$$

Now putting $l_1, l_2, l_3,\ l_4$ and l_5 in equation (6) we have:

$$\frac{1}{6}\left(\frac{\boldsymbol{u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}}}{\boldsymbol{\Delta x^2}}\right) + \frac{2}{3}\left(\frac{\boldsymbol{u_{i+1,j}+u_{i-1,j}}}{\boldsymbol{\Delta x^2}}\right) + \frac{2}{3}\left(\frac{\boldsymbol{u_{i,j+1}+u_{i,j-1}}}{\boldsymbol{\Delta x^2}}\right) - \frac{10}{3}\left(\frac{\boldsymbol{u_{i,j}}}{\boldsymbol{\Delta x^2}}\right)$$

$$+ \frac{p}{24\Delta x}\left(-9u_{i+1,j} + 9u_{i-1,j} - 2u_{i+1,j-1} + 2u_{i-1,j+1} + 9u_{i,j+1} - 9u_{i,j-1}\right)$$

$$= \frac{-p}{12}\left(f_{i+1,j} + 8f_{i,j} + f_{i-1,j} + f_{i,j-1} + f_{i,j+1}\right) . \tag{7}$$

Multiplying both side of equation (7) by $6\Delta x^2$, we get the following equation:

$$\left(u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}\right) + 4\left(u_{i+1,j} + u_{i-1,j}\right) + 4\left(u_{i,j+1} + u_{i,j-1}\right)$$

$$-20u_{i,j} + \frac{p\Delta x}{4}\left(-9u_{i+1,j} + 9u_{i-1,j} - 2u_{i+1,j-1} + 2u_{i-1,j+1} + 9u_{i,j+1} - 9u_{i,j-1}\right)$$

$$= \frac{-p\Delta x^2}{2}\left(f_{i+1,j} + 8f_{i,j} + f_{i-1,j} + f_{i,j-1} + f_{i,j+1}\right) \tag{8}$$

equation (8) can be written as:

$$= \left(u_{i+1,j+1} + u_{i-1,j-1}\right) + \left(1 - \frac{2p\Delta x}{4}\right) u_{i+1,j-1} + \left(1 + \frac{2p\Delta x}{4}\right) u_{i-1,j+1} + \left(4 - \frac{9p\Delta x}{4}\right) u_{i+1,j}$$

$$+ \left(4 + \frac{9p\Delta x}{4}\right) u_{i-1,j} + \left(4 + \frac{9p\Delta x}{4}\right) u_{i,j+1} + \left(4 - \frac{9p\Delta x}{4}\right) u_{i,j-1} - 20u_{i,j}\ .$$

$$= \frac{-p\Delta x^2 (f_{i+1,j} + 8f_{i,j} + f_{i-1,j} + f_{i,j-1} + f_{i,j+1})}{2}\ . \tag{9}$$

Let $\boldsymbol{a = 1 - \frac{2p\Delta x}{4}}$, $\boldsymbol{b = 1 + \frac{2p\Delta x}{4}}$, $\boldsymbol{c = 4 - \frac{9p\Delta x}{4}}$, $\boldsymbol{d = 4 + \frac{9p\Delta x}{4}}$, $\boldsymbol{e = 4 + \frac{9p\Delta x}{4}}$, and $\boldsymbol{g = 4 - \frac{9p\Delta x}{4}}$. For equation (9) we have

$$\boldsymbol{\left(u_{i+1,j+1} + u_{i-1,j-1}\right) + a u_{i+1,j-1} + b u_{i-1,j+1} + c u_{i+1,j} + d u_{i-1,j} + e u_{i,j+1} + g u_{i,j-1} - 20 u_{i,j}}$$

$$\boldsymbol{= \frac{-p\Delta x^2}{2}\left(f_{i+1,j} + 8f_{i,j} + f_{i-1,j} + f_{i,j-1} + f_{i,j+1}\right).} \tag{10}$$

Using equation (10), we get the system of linear equations by

$$Mu = f\ , \tag{11}$$

where M is the coefficient matrix and is very large symmetric matrix, u is the vector and f is the right hand side vector. The matrix M can be written in the form of block tri-diagonal matrix, each block of order N_y so order of the co-efficient matrix M is $N_x \times N_y$, where

$M = diag[M_2, M_0, M_1]$, and

$M_0 = diag[4\ , -20,\ 4\ ]$, $M_1 = diag[1\ , e, 1]$, $M_2 = diag[1\ , g, 1]$ are symmetric tri-diagonal sub matrices of the order N_x, where each M_2, M_0 and M_1 denote the sub-matrices of each line along one direction. The matrix M contains constant blocks at each grid line, where N used to show the number of grid points. The scheme has nine points stencil for the fourth order compact scheme which is given below.

$$\begin{bmatrix} b & e & 1 \\ d & -20 & c \\ 1 & g & a \end{bmatrix},$$

this is the stencil notations for 2D convection diffusion equation in (x, y) plane, numerical results show that fourth order compact scheme has a good accuracy.

3 Multigrid method

Numerical solution of convection-diffusion equation has been developed by using different approaches such as second order upwind difference scheme and five point central difference schemes [17,21]. For the solution of the system of linear equations obtained from the fourth order compact scheme, if we use classical iterative methods like Jacobi and Gauss-Seidel method this will slow the convergence due to large linear system. So we have to solve equation (11) with multigrid method. In this work we use multigrid algorithm, with Gauss-Seidel method as a smoother. In order to remove high frequency error using fourth order compact difference scheme, multigrid method uses some relaxation methods and to remove the errors, it uses coarse grid correction. Multigrid method with fourth order compact scheme is more efficient than the corresponding second order scheme. In this method first high frequency components of the error are reduced by applying iterative techniques like Gauss-Seidel or Jacobi methods. At the same time, the low frequency error components are removed by coarse-grid correction procedures. We suppose that the grid points are ordered lexicographically, i.e. first from left to right along the x direction, then from bottom to top along y direction. In multi-grid method, we use bilinear interpolation through which corrections transfer from coarse grid to a fine grid, we also use full-weighting scheme to update the residual on a coarse grid [8]. All multigrid methods use V-cycle or W-cycle algorithm.

Multi-grid algorithm:

The multi-grid algorithm for solving $M^h u^h = f^h$.

Let parameter γ represent the number of cycle of the multi-grid on each level, if $\gamma = 1$ it is called V-cycle and if $\gamma = 2$ is called a W-cycle.

$v_1 =$ pre-smoothing step on each level.

$v_2 =$ post-smoothing step on each level.

FAS multi-grid cycle

$$u^h \leftarrow FASCYC(u^h, f^h, v_1, v_2, \gamma)$$

1. If Ω^h is the coarsest grid solve the equation then stop.

   Else do the pre-smoothing step:

$$u^h \leftarrow GS^{v_1}(u^h, f^h, v_1, tol), \quad \text{(Pre-smoothing)}$$

2. **Restriction :**

$$u^{2h} = I_h^{2h} u^h, U^{2h} = u^{2h},$$
$$f^{2h} = I_h^{2h}(f^h - N^h u^h),$$
$$u^{2h} \leftarrow FASCYC_{\gamma}^{2h}(u^{2h}, f^{2h}, v_1, v_2).$$

3. **Interpolation**:

$$u^h \leftarrow u^h + I_{2h}^h(u^{2h} - \bar{u}^{2h})$$

$$u^{\boldsymbol{h}} \leftarrow GS^{v_2}(u^{\boldsymbol{h}},\ f^h, v_2) \qquad \text{(Post-smoothing)}.$$

4. Numerical experiments

In order to obtain results with multigrid method, we perform some numerical experiments by solving a 2D convection-diffusion equation on the unit square domain$[0,1] \times [0,1]$. The right hand side function with the Dirichlet boundary conditions are described to satisfy the exact solutions,

(1) $u(x,y) = e^{xy} \sin \pi x \,.\sin \pi y$.

(2) $u(x,y) = (X^2 - X^4)(Y^4 - Y^2)$.

(3) $u(x,y) = \ y^2(1-y)^2 \sin 2\pi x$.

In this work we will observe the results regarding, approximate solution, CPU time and residual by multigrid method using fourth order compact difference scheme and compare it with multigrid method using second order central difference scheme. If we put $v1 = v2 = 0$ in equation (2) the convection-diffusion equation reduce to Poisson equation. We examine the behavior of the scheme for different values of p, Error is reduces for $0 < p < 1$. Especially when $p = 0.00001$ the error is reduces efficiently. The fourth order compact difference scheme converges faster than the second order central difference scheme, which is clear from the tables (4.1), (4.2), (4.3). The maximum absolute error between the exact solution and approximate solution is given by :

the error vector is $e_{i,j} = U_{i,j} - u_{i,j}$ with l_2 norm as:

$$\frac{1}{N}\sqrt{\sum_{i,j=0}^{N} e_{i,j}^2}\,, \qquad (12)$$

Comparison of fourth order compact scheme with second order central difference schemes:

We see that at each grid level, fourth order compact scheme is much more accurate than the second order central difference scheme for the number of interior grid points $N = 3, 7, 15, 31, 63, 127$. The fourth order compact scheme is less time consuming than the second order central difference scheme using the same discretization parameters, N_x and N_y, in fourth order compact scheme the number of arithmetic operation is more than the second order central difference scheme. Despite it gives the best results especially at

$p = 0.00001$

From "table 4.1" we observe that the approximate solution for fourth order compact scheme is 8.1871×10^{-5} achieved with $N = 127$, similarly with $N = 127$ the approximate solution 3.8741×10^{-6} is achieved from table 4.2.

Example 4.1.

$$u_{xx} + u_{yy} - pux + puy = f(x, y), \text{ where}$$

$$f(x, y) = ye^{xy} \sin \pi x \,.\, \sin \pi y \,(y - p) + e^{xy} \sin \pi x \,.\, \sin \pi y \,(x^2 + px - \pi^2)$$
$$+xe^{xy} \sin \pi x \,.\, \pi.\cos \pi y + e^{xy}\pi.\sin \pi x \,(\cos \pi y - \pi sin\pi y),$$
$$0 \le x, y \le 1,$$

with the Drichlet boundary conditions on all sides of a unit square i.e

$$u(0, y) = u(1, y) = u(x, 0) = u(x, 1) = 0,$$

with the exact solution $u(x, y) = e^{xy} \sin \pi x \,.\, \sin \pi y$.

Comparison of maximum absolute errors and CPU time (Seconds) for a multigrid method P=0.00001

Table 4.1

N	2^{nd} Order	CPU	Residual	4^{th} Order	CPU	Residual
3	1.0317×10^{0}	0.0019	2.5371×10^{-14}	1.4267×10^{-1}	0.0019	6.1362×10^{-20}
7	3.8891×10^{-1}	0.0059	6.5948×10^{-13}	2.6204×10^{-2}	0.0063	1.8345×10^{-19}
15	1.7337×10^{-1}	0.0068	8.8991×10^{-12}	5.8764×10^{-3}	0.0070	5.7492×10^{-19}
31	8.2541×10^{-2}	0.0075	1.1188×10^{-10}	1.3735×10^{-3}	0.0092	2.1394×10^{-18}
63	4.0314×10^{-2}	0.0092	1.9492×10^{-9}	3.3256×10^{-4}	0.0124	8.4774×10^{-18}
127	1.9921×10^{-2}	0.0131	3.2232×10^{-8}	8.1871×10^{-5}	0.0188	3.3890×10^{-17}

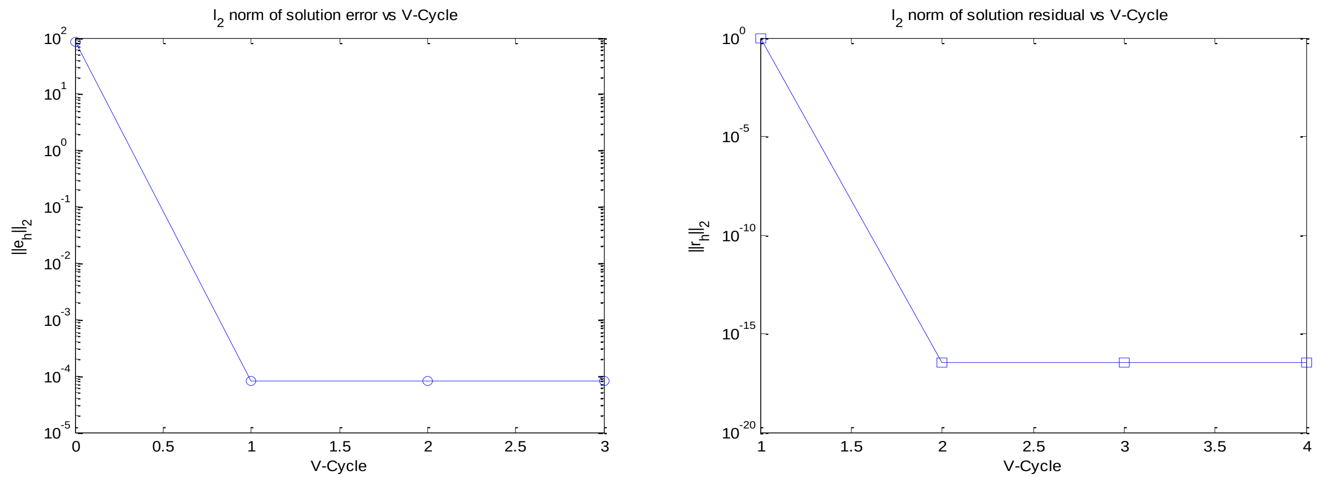


Figure 4.1: Left side figure shows the error graph and right side figure shows graph of the residuals norm. N=127 are the number of nodes.

Example 4.2:

$$u_{xx} + u_{yy} - pux + puy = f(x,y), \text{ where}$$

$$f(x,y) = 2(y^4 - y^2)[(6x^2 - 1) - 2p(x - 2x^3)]$$

$$+2(x^2 - x^4)[(6y^2 - 1) + 2p(2y^3 - y)] \,, \quad (x,y) \in [0,1],$$

with the Drichlet boundary conditions on all sides of a unit square i.e

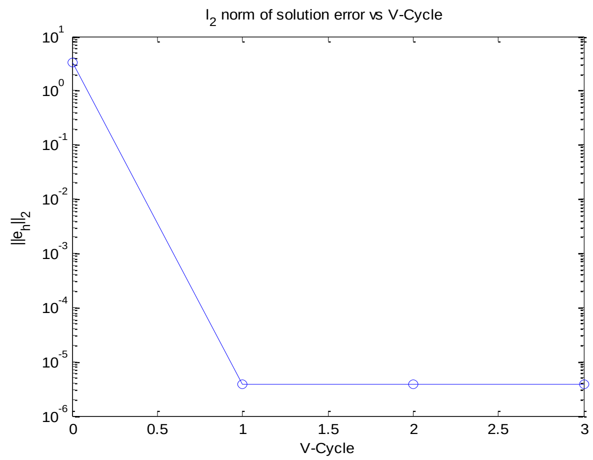
$$u(0,y) = u(1,y) = u(x,0) = u(x,1) = 0,$$

with the exact solution $u(x,y) = (X^2 - X^4)(Y^4 - Y^2)$.

Comparison of maximum absolute errors and CPU time (Seconds) for a multigrid method, P=0.00001

Table 4.2

N	2^{nd} Order	CPU	Residual	4th Order	CPU	Residual
3	5.9920×10^{-2}	0.0019	4.9970×10^{-15}	2.7264×10^{-3}	0.0019	1.6236×10^{-21}
7	1.8357×10^{-2}	0.0067	2.0803×10^{-14}	1.2360×10^{-3}	0.0062	7.2746×10^{-21}
15	8.5354×10^{-3}	0.0064	2.0568×10^{-13}	2.7612×10^{-4}	0.0069	1.7267×10^{-20}
31	4.0898×10^{-3}	0.0074	4.5121×10^{-12}	6.4892×10^{-5}	0.0075	9.7272×10^{-20}
63	1.9991×10^{-3}	0.0087	7.7435×10^{-11}	1.5743×10^{-5}	0.0079	3.2098×10^{-19}
127	9.8786×10^{-4}	0.0138	1.1557×10^{-9}	3.8741×10^{-6}	0.0234	1.3985×10^{-18}



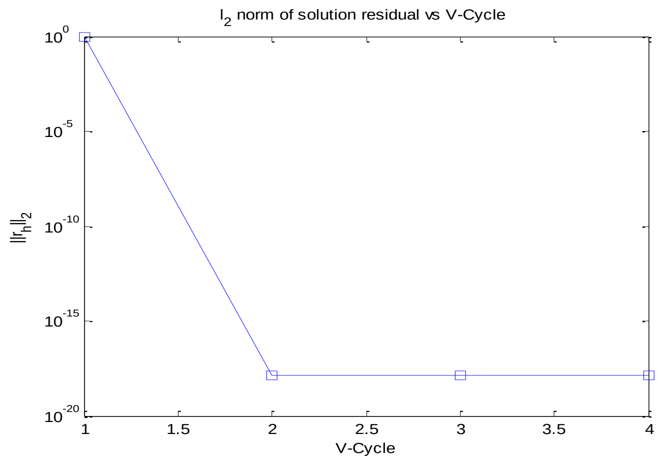


Figure 4.2: Left side figure shows the error graph and right side figure shows graph of the residuals norm. N=127 are the number of nodes.

Example 4.3:

$$u_{xx} + u_{yy} - pux + puy = f(x,y), \text{ where}$$

$$f(x,y) = 2(1-y^2)\sin 2\pi x\,(1-2\pi^2y^2+2py) + 2y^2 \sin 2\pi x\,(2y-3-py)$$

$$-2p\pi.\cos 2\pi x\, y^2(1-y^2) \qquad (x,y) \in [0,1],$$

with the Drichlet boundary conditions on all side of a unit square i.e

$$u(0,y) = u(1,y) = u(x,0) = u(x,1) = 0,$$

with the exact solution $u(x,y) = y^2(1-y)^2 \sin 2\pi x.$

Comparison of maximum absolute errors and CPU (Seconds) for a multigrid method, P=0.00001

Table 4.3

N	2^{nd} Order	CPU	Residual	4^{th} Order	CPU	Residual
3	6.1093×10^{-2}	0.0018	1.8771×10^{-15}	4.9444×10^{-3}	0.0018	1.1645×10^{-21}
7	2.0184×10^{-2}	0.0058	1.7104×10^{-14}	1.2325×10^{-3}	0.0060	1.0226×10^{-21}
15	3.7230×10^{-3}	0.0065	3.5191×10^{-13}	2.7777×10^{-4}	0.0067	2.2554×10^{-20}
31	4.1103×10^{-3}	0.0100	5.7448×10^{-12}	6.5036×10^{-4}	0.0072	8.5589×10^{-20}
63	2.0001×10^{-3}	0.0080	8.1182×10^{-11}	1.5847×10^{-5}	0.0140	3.5831×10^{-19}
127	9.9829×10^{-3}	0.0106	1.2931×10^{-9}	3.5750×10^{-6}	0.0207	1.4555×10^{-18}

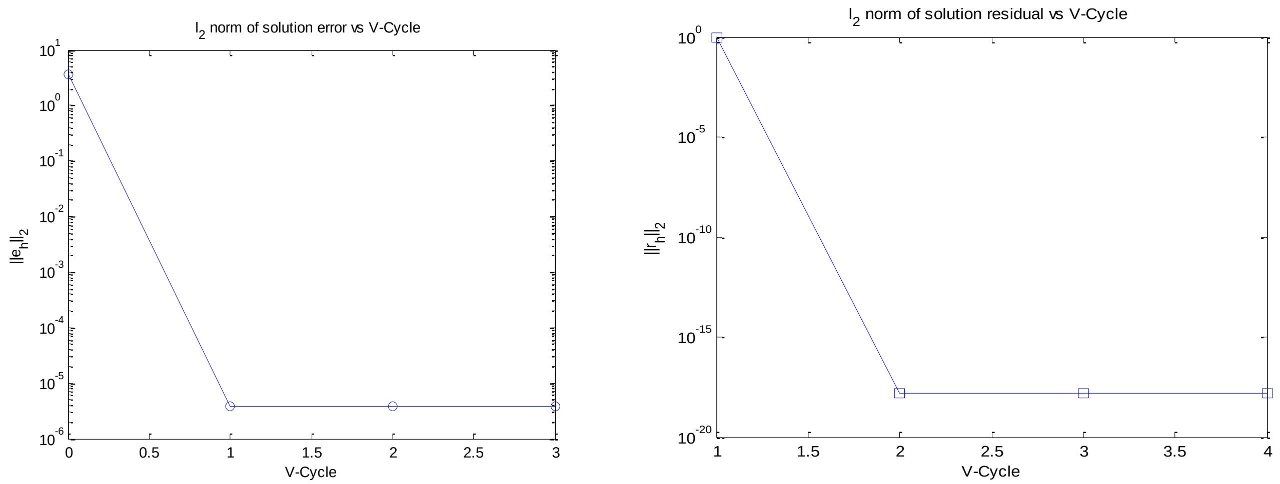


Figure 4.3: Left side figure shows the error graph and right side figure shows graph of the residuals norm. N=127 are the number of nodes.

We studied fourth order compact difference scheme for discretization of 2D convection diffusion equation. Multigrid method used to solve the resulting sparse linear systems. Multigrid method using Gauss-Seidel smoother is proved to be more effective for the solution of convection diffusion equation with given peclet number. Our numerical works show that multigrid method with fourth order compact scheme is more accurate than the second order central difference scheme, different figures show error graphs and residual norm.

Conclusion

In this research work, we have studied fourth order compact difference scheme with uniform mesh points by discretization of the two dimensional convection-diffusion equation. We have studied this problem with boundary conditions and developed a multigrid method to solve the given system of equations efficiently. The main advantage of this method is to solve the convection diffusion equation with peclet number ($p = 0.00001$) with high efficiency. Moreover it was found that multigrid method with Gauss-Seidel smoother works very fast which is based on fourth order compact difference scheme.

We conducted numerical experiments to test the accuracy of the multigrid method and found that multigrid method with fourth order compact scheme is more accurate and faster than the second order central difference scheme.

Bibliography:

[1] N. S. Bakhvalov, *Optimization Methods for Boundary Value Problems in the Presence of a Boundary* Layer, Zh.Vychisl. Mat. Mat. Fiz. **9**, 841–859 (1969).

[2] A. Brandt. *Multi-level Adoptive solutions to boundary value problems* . Math. Comp. 31(138):333-390.

[3] T. F. Chan and W. L . Wan. *Robust multigrid methods for smooth coefficient elliptic linear system. Journal of computational and applied mathematics* :123:323,352,2000.

[4] P. M. De Zeeuw. *Matrix dependent prolongations and restriction in a block box Multigrid Solver*. *Journal of Computational and Applid Mathematics*, 33:1-27, 1990.

[5] J. E. Dendy. Black Box Multigrid. *Journal of Computational Physics*, 48:366-386,1982.

[6] J. Douglas Faires and Richard L. Burden. *Numerical methods*. Brooks / Cole Publishing Co. pacific Grove, CA, sixth edition, 1998.

[7] H. Elman, D. Silvester, A. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2005.

[8] M. M. Gupta, J. Kouatchou, and J. Zhang, *Comparison of second and fourth order discretizations for multigrid Poisson solver*, *J. Comput. Phys.* **132**, 226 (1997).

[9] R. Hahn, *Sub-diffusive reaction diffusion equation* Ph.D thesis. 2010.

[10] V. E. Henson. *Multigrid methods nonlinear problem* : *an overview. Computational Imaging*, 5016:36-48, 2003.

[11] J. A. N. Kolmogorov .I. Petrovskii.and N.Piskounov. *A Study of the diffusion equation with increase in the amount of substance and its application to a biological problem.* In selected work of A. N. Kol- mogorov I. Kluwer,1991.

[12] K. W. Morton.*Numerical Solution of Convection-Diffusion Problems* (chapmen- Hall. London. 1996).

[13] Ke Chen. *Matrix Preconditioning Techniques and Applications*. Cambridge Univerty press. The Edinburgh Building, Cambridge CB2 2RU,UK, first edition, 2005.

[14] G. I. Shishkin, *Grid Approximation of a Singularly Perturbed Elliptic Equation with convective terms in the Presence of Various Boundary Layers*, Zh. Vychisl. Mat. Mat. Fiz. **45**, 110–125 (2005) [Comput. Math. Math.Phys. **45**, 104–119 (2005)].

[15] W. F. Spotz and G. F. Carey, *High-order compact scheme for the steady stream function vorticity equations*, *Int. J. Numer. Methods Eng.* **38**, 34-97 (1995).

[16] W. F. Spotz. *High-Order Compact Finite Difference Schemes for Computational Mechanics.* PhD thesis, University of Texas at Austin, Austin, TX, 1995.

[17] U. Trottenbrge and Anton Schuller. *Multigrid*. Acadmic Press, Inc., Orlando, FL, USA, 2001.

[18] T. Lu, P. Neittaanmaki, and X.-C. Tai, *A parallel splitting up method for partial diffferential equations and its application to navier-stokes equations*. RAIRO Math. Model and Numerical Analysis **26** (1992), no. 6, 673–708.

[19] P. Wesseling, M. *An introduction to multigrid methods* (Wily, Chichester, Uk, 1992).

[20] H. William. Press. Saul A. Teukolsky. William T. Vetterling. And Brian P. Flannery. *Numerical Recipes in C, The art of Scientific Computing.* Cambridge University press, 1992.

[21] J. Zhang. *Preconditioned Iterative Methods and Finite Difference Schemes for Convection Diffusion,* Appl. Math. Comput. **109**, 11–30 (2000).