A Sixth Order Compact Multigrid Solver for the Convection Diffusion Equation with Boundary Layers

Christopher Kawatsu* Yin Wang† Ruxin Dai‡ and Jun Zhang§

January 9, 2012

Abstract

A geometric multigrid method for solving the convection diffusion equation with boundary layers to sixth order accuracy is presented. A nine point finite difference discretization scheme is used to obtain fourth order accurate solutions on a coarse and a fine grid. Richardson extrapolation is used to increase the order of accuracy to sixth order on the coarse. An iterative smoothing technique is then used to obtain a sixth order solution on the fine grid. The discretization we used allows the grid to be a graded mesh. This is the first time the post extrapolation smoothing technique has been applied to a graded mesh. Numerical results are presented to demonstrate the use of a graded mesh can significantly decrease the maximum error compared to a regular mesh.

*Department of Mathematics and Computer Science, Lawrence Technological University, 21000 West Ten Mile Road, Southfield, MI, 48075-1058, USA. E-mail: ckawatsu@ltu.edu.

†Department of Mathematics and Computer Science, Lawrence Technological University, 21000 West Ten Mile Road, Southfield, MI, 48075-1058, USA. E-mail: ywang12@ltu.edu.

‡Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0633, USA. E-mail: ruxin.dai@uky.edu.

§Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA. E-mail: jzhang@cs.uky.edu.

1 Introduction

The convection diffusion equation is widely used in many science and engineering problems modeling fluid flows and heat transfer. Numerical solutions to the convection diffusion equation can be obtained using a variety of methods. In the present paper, a finite difference discretization and multiscale multigrid method are used.

The two dimensional (2D) convection diffusion equation with Dirichlet boundary conditions is given by

$$\begin{aligned} u_{xx} + u_{yy} + p(x,y)u_x + q(x,y)u_y &= f(x,y), \quad (x,y) \in \Omega, \\ u(x,y) &= g(x,y), \quad (x,y) \in \partial\Omega, \end{aligned} \tag{1}$$

where Ω is a rectangle or a union of rectangles in R^2 and $\partial\Omega$ is the boundary of Ω. It is additionally assumed that $p(x,y)$ and $q(x,y)$ are smooth enough to accommodate the discretization scheme.

The usual choices of finite difference discretization for the convection diffusion equation are the central difference scheme (CDS) and the upwind difference scheme (UDS). The CDS has a truncation error of $O(h^2)$ but has convergence problems for large Reynolds numbers [14]. The UDS converges for problems with large Reynolds number, but the order of accuracy is reduced to $O(h)$ [9]. Both of these schemes are compact since every point depends only on the adjacent grid points.

There are interests in finding higher order compact finite difference discretizations which provide better accuracy than the discretizations mentioned above. Many fourth order compact discretizations have been developed for the 2D [4, 8] and 3D [6, 13] convection diffusion equation. Sixth order schemes [10, 11, 12] have also been developed. In the present paper, sixth order schemes which use Richardson extrapolation and post extrapolation interpolation to increase the solution accuracy from fourth order to sixth order are of interest.

When discretizing a problem, it is desirable to have many points in areas with high gradient in order to obtain high solution accuracy. Conversely, it is desirable to have fewer grid points in smooth areas of the domain. This reduces the computational and storage costs required to obtain the solution. Many of the higher order discretizations require regular grid spacing which prevents increasing the number of grid points in regions with high gradient. In particular, techniques using Richardson extrapolation require grids such that the truncation error on a fine grid is a known multiple of the truncation error on a coarser grid.

In the present paper, a sixth order compact scheme that allows graded computation domains is presented. The scheme uses a coordinate transformation from [2] to transform the graded mesh into a regular grid. A fourth order discretization is used on the regular grid. Richardson extrapolation is applied on the regular grid to obtain a sixth order solution. Finally, the solution is transfered back to the graded mesh.

An outline of the paper is as follows. In Section 2 the coordinate transformation and the associated discretization scheme is presented. Section 3 describes the Richardson extrapolation technique used to obtain a sixth order solution on a non uniform grid. The multigrid method used to solve the system is discussed in Section 4. The post extrapolation smoothing used to obtain a sixth order solution on the fine grid is discussed in Section 5. Numerical results are presented in Section 6.

2 Coordinate Transform

Let $M : (x, y) \to (\xi, \eta)$ be a non degenerate map from a graded mesh on $(x, y) \in [0, 1]^2$ to a uniform mesh on $(\xi, \eta) \in [0, 1]^2$. On the uniform mesh Equation (1) becomes [3]

$$\alpha(\xi,\eta)u_{\xi\xi} + \beta(\xi,\eta)u_{\eta\eta} + c(\xi,\eta)u_{\xi\eta} + \lambda(\xi,\eta)u_{\xi} + \mu(\xi,\eta)u_{\eta} = f(\xi,\eta) \qquad (2)$$

where the coefficients are given by

$$\begin{aligned}
\alpha(\xi,\eta) &= \xi_x^2 + \xi_y^2, \\
\beta(\xi,\eta) &= \eta_x^2 + \eta_y^2, \\
c(\xi,\eta) &= 2(\xi_x\eta_x + \xi_y\eta_y), \\
\lambda(\xi,\eta) &= p(\xi,\eta)\xi_x + q(\xi,\eta)\xi_y + \xi_{xx} + \xi_{yy}, \\
\mu(\xi,\eta) &= p(\xi,\eta)\eta_x + q(\xi,\eta)\eta_y + \eta_{xx} + \eta_{yy}.
\end{aligned}$$

In the case where both the uniform and graded meshes are orthogonal and have the same unit vectors for each axis, then $c = 0$ and Equation (2) becomes

$$\alpha(\xi,\eta)u_{\xi\xi} + \beta(\xi,\eta)u_{\eta\eta} + \lambda(\xi,\eta)u_{\xi} + \mu(\xi,\eta)u_{\eta} = f(\xi,\eta). \qquad (3)$$

Gupta et al. [5] have obtained a fourth order finite difference scheme for Equation (3) by substituting Taylor series expansions into Equation (3). The discretization is given by

$$\sum_{i=0}^{8} \alpha_i u_i = 6h^2 f_{00} + h^4(f_{20} + f_{02} + T_1 f_{10} + T_2 f_{01}), \tag{4}$$

where the coefficients are given by

$$\begin{aligned}
\alpha_0 &= -(2R_1 + 2R_2 + 4S_1),\\
\alpha_1 &= R_1 + R_3,\\
\alpha_2 &= R_2 + R_4,\\
\alpha_3 &= R_1 - R_3,\\
\alpha_4 &= R_2 - R_4,\\
\alpha_5 &= S_1 + S_2 + S_3 + S_4,\\
\alpha_6 &= S_1 + S_2 - S_3 - S_4,\\
\alpha_7 &= S_1 - S_2 + S_3 - S_4,\\
\alpha_8 &= S_1 - S_2 - S_3 + S_4,\\
T_1 &= \frac{\lambda_{00} - 2\alpha_{10}}{2\alpha_{00}},\\
T_2 &= \frac{\mu_{00} - 2\beta_{01}}{2\beta_{00}},\\
R_1 &= 5\alpha_{00} - \beta_{00} + T_1 h^2(\lambda_{00} + \alpha_{10}) + T_2 h^2 \alpha_{01} + h^2(\alpha_{20} + \alpha_{02} + \lambda_{10}),\\
R_2 &= 5\beta_{00} - \alpha_{00} + T_2 h^2(\mu_{00} + \beta_{01}) + T_1 h^2 \beta_{10} + h^2(\beta_{20} + \beta_{02} + \mu_{01}),\\
R_3 &= \frac{h}{2}(5\lambda_{00} - \beta_{10} - 2\beta_{00} T_1) + \frac{h^3}{2}(T_1 \lambda_{10} + T_2 \lambda_{01} + \lambda_{20} + \lambda_{02}),\\
R_4 &= \frac{h}{2}(5\mu_{00} - 2\alpha_{01} - 2\alpha_{00} T_2) + \frac{h^3}{2}(T_2 \mu_{01} + T_1 \mu_{10} + \mu_{20} + \mu_{02}),\\
S_1 &= \frac{1}{2}(\alpha_{00} + \beta_{00}),\\
S_2 &= \frac{h}{4}(\mu_{00} + \beta_{00}),\\
S_3 &= \frac{h^2}{4}(\mu_{10} + \lambda_{01} + T_1 \mu_{00} + T_2 \lambda_{00}),\\
S_4 &= \frac{h}{4}(\lambda_{00} + 2\beta_{10} + 2T_1 \beta_{00}).
\end{aligned}$$

The double subscripts on α, β, λ, μ, and f denote partial derivatives in the ξ and η directions respectively. For example,

$$\alpha_{ij} = \frac{1}{i!j!}\frac{\partial^{i+j}\alpha}{\partial\xi^i\partial\eta^j}. \tag{5}$$

Instead of using the analytical solution given in (5) the solution method presented here uses the central difference scheme to approximate the partial derivatives.

3 Richardson Extrapolation

Richardson extrapolation [7] uses the relationship between the truncation error on different scale grids to increase the order of accuracy of the solution. Suppose the truncation error of a fourth order finite difference discretization is given by

$$f = \tilde{f}_h + Kh^4 + O(h^6), \tag{6}$$

then the truncation error on the $2h$ grid can be written as

$$f = \tilde{f}_{2h} + K(2h)^4 + O(h^6). \tag{7}$$

Using (6) and (7) a sixth order solution can be obtained by the substitution

$$f = \frac{16\tilde{f}_h - \tilde{f}_{2h}}{15} + O(h^6). \tag{8}$$

Consider the transformation M and discretization from Section 2. If the transformed convection diffusion equation is solved on (ξ, η) the solution is $O(h^4)$. However, this h corresponds to the spacing between grid points on the (ξ, η) grid, not the spacing between grid points on the (x, y) grid. To make this clear denote the grid spacing as $h_{(\xi,\eta)}$. Now suppose the approximate solution at $(\alpha, \beta) \in (\xi, \eta)$ is $\tilde{f}$ and the exact solution is f. Then Equation (8) becomes

$$f = \frac{16\tilde{f}_h - \tilde{f}_{2h}}{15} + O(h^6_{(\xi,\eta)}).$$

An expansion in h for the (x, y) coordinates is not meaningful since the spacing between points is potentially different in every direction. The scheme in the present paper is said to be sixth order since the order of accuracy on the uniform mesh is sixth order.

4 Multigrid Solution Method

Standard iterative methods such as Jacobi or Gauss-Seidel typically require $O(n^2)$ time to converge for a system with n unknowns. This occurs because these methods are able to eliminate high frequency error quickly but take many iterations to remove low frequency error. The multigrid method [1] uses multiple scale grids to alleviate this problem. On coarser grids, error that is low frequency on fine grids becomes high frequency and can be eliminated efficiently using iterative methods.

The multigrid method used in the present paper is the V-cycle. A V(i,j)-cycle starts on the finest grid which is smoothed using an iterative method i times. The residual is then restricted onto a coarser grid. This process is repeated until the coarsest grid level is reached. The residual from the coarsest grid is projected onto the second coarsest grid and smoothed j times. This is repeated until the finest grid level is reached. Once the finest grid level is reached, the V-cycle is complete. Typically, multiple V-cycles are required to obtain a converged solution, but the number of V-cycles is independent from the number of unknowns for many problems.

In the present paper V(1,1) cycles are used. The smoothing method used is line Gauss-Seidel. Full weighting is used as the restriction operator. Bilinear interpolation is used as the prolongation operator.

5 Post Extrapolation Smoothing

The multigrid method discussed in the previous section is used to compute fourth order solutions on a fine and coarse grid with grid spacings h and $2h$ respectively. Richardson extrapolation results in a sixth order solution on the $2h$ grid. In order to obtain a sixth order solution on the h grid, the operator interpolation[1] technique from [11] is applied.

In order to apply post extrapolation smoothing, the sixth order $2h$ grid points are interpolated to the $(even, even)$ grid points of the fourth order h grid. After interpolation, the (odd, odd) points are updated using Gauss-Seidel, followed by the $(even, odd)$ and $(odd, even)$ points. The Gauss-Seidel smoothing continues until the residual norm reaches a specified tolerance.

[1] In the present paper, operator interpolation is referred to as post extrapolation smoothing.

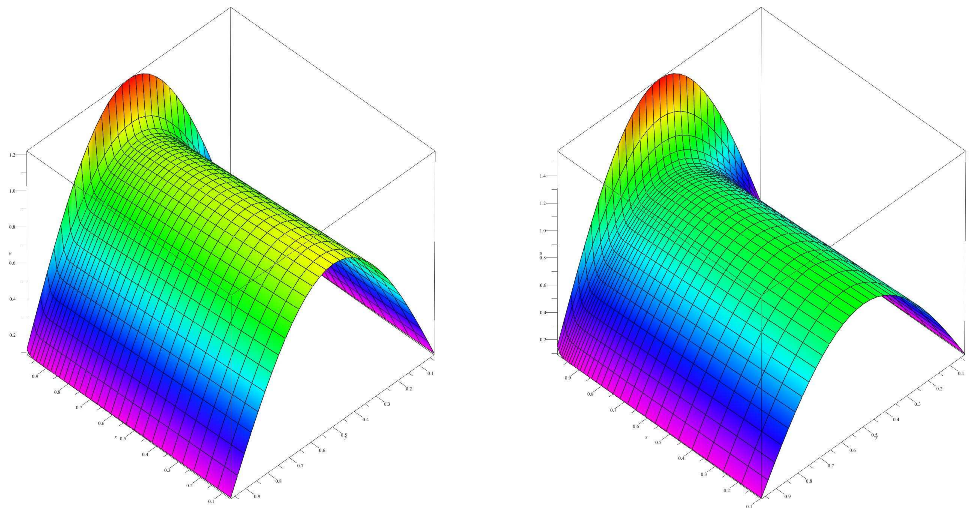

Figure 1: Computed solution for Problem 1 after post extrapolation smoothing for $\epsilon = .01$ on the graded mesh with $1/h = 32$. Left side shows $\gamma = 5$, right side shows $\gamma = 20$.

6 Numerical Results

The domain for all problems is the unit cube. Standard V(1,1) cycles are used for the multigrid V-cycles. The initial guess for u is the zero vector. The stopping condition is a residual norm of 10^{-10}. The reported maximum error is given by

$$E_{max} = \sup_{(i,j)\in\Omega_h} |\tilde{u}_{i,j} - u_{i,j}|.$$

In order to calculate the order of accuracy experimentally, the average error is used:

$$E^h_{avg} = \frac{\sum_{(i,j)\in\Omega_h} |\tilde{u}_{i,j} - u_{i,j}|}{\sum_{(i,j)\in\Omega_h} |u_{i,j}|}.$$

The reported order of accuracy is then computed using

$$O = \frac{\log E^{2h}_{avg}/E^h_{avg}}{\log 2}.$$

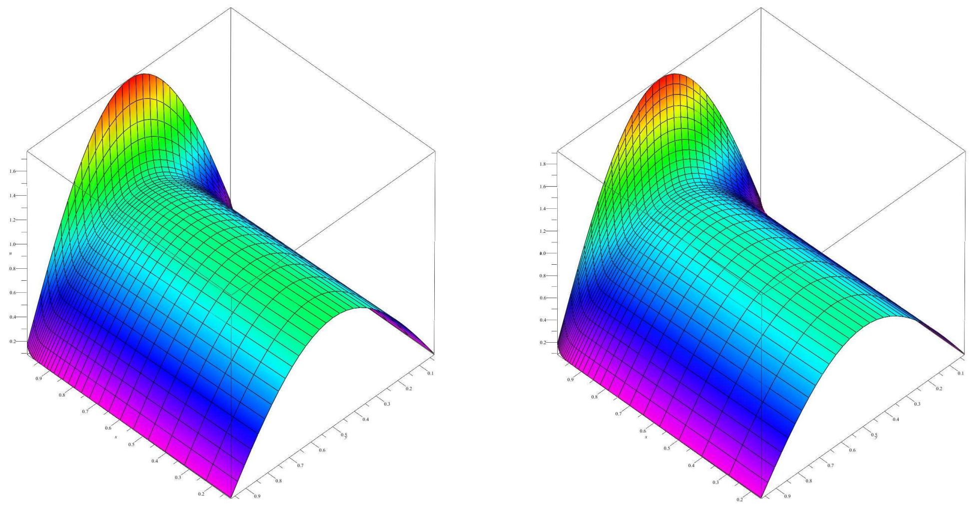

Figure 2: Computed solution for Problem 1 after post extrapolation smoothing for $\epsilon = .01$ on the graded mesh with $1/h = 32$. Left side shows $\gamma = 50$, right side shows $\gamma = 200$.

6.1 Problem 1

In this section the same test problem as [2, 3, 5] is used for comparison. The test problem uses the constant convection diffusion equation given by

$$-\epsilon(u_{xx} + u_{yy}) + u_x = 0.$$

The exact solution is then

$$u(x, y) = e^{x/2\epsilon} \sin(\pi y) \frac{2e^{-1/2\epsilon} \sinh(\sigma x) + \sinh[\sigma(1 - x)]}{\sinh \sigma},$$

where $\sigma^2 = \pi^2 + \epsilon^2/4$. The following coordinate transformation is used,

$$\begin{aligned} x &= (1 - e^{-Q\xi})/(1 - e^{-Q}), \\ y &= \eta, \\ Q &= \ln(\gamma)/(1 - \Delta\xi). \end{aligned}$$

Using the same method as Dai *et al.* [2] the mesh stretching in the x direction is quantified using γ. Where γ is the ratio between the largest and smallest

Δx in the discretization. Examples of computed values for $u(x, y)$ are shown in Figures 1 and 2.

Table 1: Problem 1 numerical results for extrapolation. CPU times are not included because the variance of the measurement was comparible to the measured value in most cases. In the case $\gamma = 50$, $\epsilon = 0.1$ the residual when restricting from fine to coarse grids was scaled by 0.5 to obtain a better convergence rate.

		$1/h = 32$			$1/h = 64$			$1/h = 128$		
ϵ	γ	E_{max}	O	Iter.	E_{max}	O	Iter.	E_{max}	O	Iter.
1	5	5.02(-8)	5.74	(9,8)	7.88(-10)	5.96	(10,9)	1.22(-11)	6.01	(11,9)
0.1	5	8.46(-8)	6.14	(11,9)	1.30(-9)	6.07	(11,9)	2.02(-11)	6.03	(11,8)
	10	2.09(-7)	6.06	(14,10)	3.23(-9)	6.03	(14,9)	4.94(-11)	6.03	(14,9)
	50	3.35(-6)	6.53	(14,13)	4.13(-8)	6.11	(17,13)	6.46(-10)	6.04	(254,256)
0.01	5	4.99(-4)	5.83	(15,17)	1.12(-5)	5.89	(21,17)	1.88(-7)	5.99	(23,16)
	10	8.71(-5)	6.72	(17,21)	1.45(-6)	6.09	(26,22)	2.42(-8)	6.04	(29,21)
	20	1.66(-5)	7.08	(19,23)	2.72(-7)	6.06	(30,25)	4.45(-9)	6.01	(33,25)
	50	8.94(-6)	6.86	(21,24)	1.60(-7)	5.62	(31,33)	2.66(-9)	5.84	(44,35)
	100	1.38(-5)	6.56	(22,23)	2.82(-7)	5.37	(30,37)	4.99(-9)	5.71	(50,46)
	200	2.51(-5)	6.46	(25,29)	5.49(-7)	5.30	(36,39)	1.03(-8)	5.61	(54,58)
0.001	50	1.08(-1)	2.34	(6,11)	4.69(-4)	7.26	(13,29)	1.10(-5)	5.89	(37,39)
	100	2.28(-3)	6.62	(6,12)	4.72(-5)	7.13	(14,28)	8.24(-7)	5.81	(36,42)
	200	3.92(-4)	8.53	(6,13)	6.35(-6)	6.28	(15,26)	1.31(-7)	5.10	(33,44)
	300	2.31(-4)	8.82	(6,13)	3.78(-6)	5.79	(15,25)	1.01(-7)	4.81	(34,45)
	400	2.01(-4)	8.71	(6,13)	3.46(-6)	5.58	(16,27)	1.09(-7)	4.68	(34,46)

The order of accuracy of the extrapolated solution was improved in all cases; however for small ϵ, the improvment in the order of accuracy is greatly reduced on large grids. This occurs because the fourth order discretization produces results less than fourth order. The change in the order of accuracy of the unextrapolated solution greatly reduces the accuracy gained by extrapolation. For small ϵ the number of iterations required to converge increases with grid size. (This is typical for problems with high Reynolds number.) It is interesting to note that the iterations required to converge on the $1/h = 32$ grid are smaller for higher ϵ. For small grids and high ϵ, the problem can actually be solved in very few iterations by a standard Gauss-Seidel solver [2]. This contributes to the small number of V-cycles required for convergence.

Table 2: Problem 1 numerical results after post extrapolation smoothing. The iterations reported are the number of Gauss-Seidel iterations required to reduce the residual norm to 10^{-10} after extrapolation.

		$1/h = 32$			$1/h = 64$			$1/h = 128$		
ϵ	γ	E_{max}	O	Iter.	E_{max}	O	Iter.	E_{max}	O	Iter.
1	5	5.09(-7)	5.78	29	9.12(-9)	5.93	27	1.52(-10)	6.04	25
0.1	5	1.00(-7)	5.87	29	1.70(-9)	5.94	29	2.87(-11)	5.96	26
	10	7.61(-7)	5.61	42	1.49(-8)	5.83	46	2.73(-10)	5.93	46
	50	8.42(-6)	5.64	122	1.74(-7)	5.73	180	3.04(-9)	5.86	226
0.01	5	1.72(-3)	5.03	23	6.86(-5)	5.02	33	2.27(-6)	5.38	40
	10	3.96(-4)	5.23	28	1.67(-5)	5.05	52	4.69(-7)	5.43	65
	20	1.09(-4)	5.36	45	3.96(-6)	5.13	85	9.78(-8)	5.54	113
	50	2.23(-5)	5.92	83	6.72(-7)	5.18	146	2.77(-8)	5.38	212
	100	6.65(-5)	5.70	120	2.74(-6)	4.83	241	8.88(-8)	5.21	398
	200	1.44(-4)	5.46	153	5.65(-7)	4.79	373	1.75(-7)	5.18	693
0.001	50	2.44(-2)	4.09	561	2.17(-3)	5.65	146	1.40(-4)	4.40	94
	100	5.48(-3)	5.91	504	3.87(-4)	5.17	155	2.40(-5)	4.27	164
	200	1.21(-3)	7.48	447	8.12(-5)	4.80	166	5.15(-6)	4.28	290
	300	5.09(-4)	7.87	422	3.62(-5)	4.78	175	2.28(-6)	4.34	396
	400	2.89(-4)	7.93	425	2.10(-5)	4.82	190	1.32(-6)	4.39	485

6.2 Problem 2

In this section a test problem with a non-zero right hand side and variable p and q values is used. The problem is as follows,

$$\begin{aligned} p(x,y) &= \operatorname{Re} x(x-1)(1-2y), \\ q(x,y) &= \operatorname{Re} y(y-1)(1-2x), \\ u(x,y) &= \exp(-\sigma(x-\tfrac{1}{2})^2 - y^2), \\ f(x,y) &= u_{xx} + u_{yy} + p(x,y)u_x + q(x,y)u_y. \end{aligned}$$

For large Re the solution to this problem has a very steep slope near $x = 0.5$. In order to place more points near this line, the following coordinate transform is used:

$$\begin{aligned} x &= \xi + \tfrac{\gamma}{2\pi}\sin(2\pi\xi), \\ y &= \eta. \end{aligned}$$

In Table 3 results for varying degrees of mesh stretching are shown. By increasing γ from 0.1 to 0.7 the maximum error is decreased by three orders of magnitude. Table 4 shows results for various mesh sizes. The V-cycles required to converge are independent of grid size. The Gauss-Seidel iterations required for post extrapolation smoothing actually decrease with grid size in most cases.

Table 3: Problem 2 numerical results for different values of γ. Higher γ values indicate more mesh stretching.

$1/h = 512, \sigma = 1000, \mathrm{Re} = 1000$					
γ	E_{max}	O	V_{2h} Cycles	V_h Cycles	Smoothing Iter.
0.1	1.67(-7)	5.89	10	9	26
0.2	9.68(-8)	5.90	10	9	26
0.3	5.30(-8)	5.90	10	8	28
0.4	2.77(-8)	5.91	10	8	30
0.5	1.30(-8)	5.92	11	9	35
0.6	5.52(-9)	5.93	11	9	44
0.7	4.11(-9)	5.94	11	10	64
0.8	4.90(-9)	5.91	12	11	130

Table 4: Problem 2 numerical results for different grid sizes.

$\gamma = 0.7, \sigma = 1000, \mathrm{Re} = 1000$					
$1/h$	E_{max}	O	V_{2h} Cycles	V_h Cycles	Smoothing Iter.
64	3.72(-4)	6.19	12	13	117
128	1.04(-5)	5.24	11	11	70
256	2.35(-7)	5.73	11	10	69
512	2.77(-8)	5.91	10	8	30
1024	6.67(-11)	5.24	11	9	58

7 Concluding Remarks

A sixth order multigrid solver for the convection diffusion equation was presented. Numerical results for the first test problem show sixth order solutions for $\epsilon = 1$ to $\epsilon = 0.1$. Solutions for smaller ϵ failed to achieve sixth order solutions, but still show greater than fourth order accuracy. Additionally, the results show that an appropriately chosen graded mesh can reduce the maximum error by over one order of magnitude. For the second test problem similar results were observed. In this case a graded mesh was able to reduce the maximum error by three orders of magnitude.

References

[1] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial.* SIAM, Philadelphia, PA, 2nd edition, 2000.

[2] R. Dai, Y. Wang, J. Zhang, and Y. Ge. Comparison of Two Higher Order Compact Computation Strategies for Handling Boundary Layers. Copper Mountain Conference on Multigrid Method, Copper Mountain, Colorado, March 2011.

[3] L. Ge and J. Zhang. High accuracy iterative solution of convection diffusion equation with boundary layers on nonuniform grids. *J. Comput. Phys.*, 171(2):560-578, 2001.

[4] M. M. Gupta, R. P. Manohar, and J. W. Stephenson. A single cell high order scheme for the convection-diffusion equation with variable coefficients. *Int. J. Numer. Methods Fluids*, 4:641-651, 1984.

[5] M. M. Gupta, R. P. Manohar, and J. W. Stephenson. High-order difference schemes for two-dimensional elliptic equations. *Numer. Methods Partial Differential Eq.*, 1:71-78, 1985.

[6] M. M. Gupta and J. Zhang. High accuracy multigrid solution of the 3D convection-diffusion equation, *Appli. Math. and Comput.*, 113:249-274, 2000.

[7] L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. *Phil. Trans. R. Soc. Lond. A*, 210:459-470, 1911.

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

[9] W. F. Spotz and G. F. Carey. High-order compact scheme for the steady stream-function vorticity equations. *Int. J. Numer. Methods Engrg.*, 38:3497-3512, 1995.

[10] H. Sun and J. Zhang. A high order finite difference discretization strategy based on extrapolation for convection diffusion equations. *Numer. Methods Partial Differential Eq.*, 20(1):18-32, 2004.

[11] Y. Wang and J. Zhang. Integrated fast and high-accuracy computation of convection diffusion equations using multiscale multigrid method. *Numer. Methods Partial Differential Eq.*, DOI: 10.1002/num.20529.

[12] Y. Wang and J. Zhang. High Accuracy and Scalable Multiscale Multigrid Computation for 3D Convection Diffusion Equation. *J. Comput. and Appli. Math.*, 234:3496-3506, 2010.

[13] J. Zhang. An explicit fourth-order compact finite difference scheme for three dimensional convection-diffusion equation. *Commun. Numer. Methods Engrg.*, 14:209-218, 1998.

[14] J. Zhang. Preconditioned iterative methods and finite difference schemes for convection-diffusion. *Appl. Math. Comput.*, 109:11-30, 2000.