CILAMCE 2017
Florianópolis - Santa Catarina - Brazil

ANALYSIS OF PARAMETERS OF GEOMETRIC MULTIGRID FOR THE CPU TIME FOR THREE-DIMENSIONAL POISSON EQUATION

Fabiane de Oliveira

fabiane1910@yahoo.com.br

State University of Ponta Grossa. Department of Mathematics and Statistics

Av. Gen. Carlos Cavalcanti, N^o 4.748 - Uvaranas, Zip-Code: 84030-900, Ponta Grossa – PR, Brazil

Sebastião Romero Franco

romero@unicentro.br

State University of Centro-Oeste, Department of Mathematics

Irati – PR, Brazil

Federal University of Paraná, Graduate Program in Numerical Methods in Engineering Curitiba – PR, Brazil

Marcio Augusto Villela Pinto

marcio_villela@ufpr.br

Federal University of Paraná, Department of Mechanical Engineering

Curitiba – PR, Brazil

Abstract. *The aim of this paper is to minimize the CPU time necessary for solving the 3D Poisson equation with Dirichlet boundary conditions. The Finite Difference Method is used to discretizate the differential equation with central differencing scheme. The systems of equations are solved with the lexicographic and red–black Gauss–Seidel methods associated to the geometric multigrid with correction scheme and V-cycle. It used trilinear interpolation and full coarsening with ratio r = 2. Comparisons are made among: (1) Restriction injection and full weighting; (2) Inner iteration number (v); (3) Number of mesh levels (L) and (4) unknowns number (N). With the analysis of algorithm complexity was possible to verify the optimum values this multigrid method in relation of optimization of CPU time.*

Keywords: *Multigrid, Finite difference method, 3D Poisson.*

INTRODUCTION

To reduce the discretization error in computational fluid dynamics (CFD) problems, very refined meshes are required that generate very large equation systems. Solving these systems through basic iteration (solvers) methods requires large CPU time. This is because at the beginning of the iterative process the convergence rate is large, decreasing considerably as the number of iterations increases.

There are many research papers that aim to increase the convergence rate of iterative methods. For this, a method that can be used is the multigrid (Briggs et al., 2000; Wesseling, 1992). It consists of the use of thicker auxiliary meshes (with fewer nodes) than the mesh in which the problem is solved. Restriction and extension processes are used to transfer information between the various meshes. Various solvers may be used in the relaxation or smoothing process. The sequence with which the various meshes are visited is called the multigrid cycle (cycle V, W, F among others). In each type of cycle one can start from the thicker mesh, in the scheme called full multigrid (FMG), or the finer mesh, in the so-called standard scheme. Two types of schemas can be used with the multigrid method: CS (Correction Storage) and FAS (Full Approximation Storage). They are indicated respectively for linear and non-linear problems (Briggs et al., 2000). Finally, we can distinguish the geometric and algebraic multigrid methods, respectively indicated for structured and unstructured meshes.

The efficiency of the multigrid method has not been fully achieved in practical engineering applications in the area of CFD. With the increasing complexity of applications, the demand for more efficient and robust methods is also increasing Trottenberg et al. (2001). These methods are expected to have a good reduction in computational time, to use low memory, and to address non-linearity and couplings with no major impairment in performance.

Wesseling and Oosterlee (2001) reviewed the developments of the geometric multigrid method in CFD in the 1990s, showing the state of the art for incompressible and compressible flows. According to them the geometric multigrid remains an active topic of research in CFD and is one of the most significant developments in numerical analysis in the second half of the twentieth century. Theoretically, the current algorithms used with the geometric multigrid can still be much optimized in relation to the computational time needed to solve a CFD problem; for example, to solve the Navier-Stokes equations, the CPU time can be reduced still from 10 to 100 times (Brandt et al., 2002) of the current one. Reducing computational time to solve the same problem results in reduced project costs. An increase in the efficiency of the method also allows, in the same computational time, to solve a problem in a more refined mesh, that is, with greater number of nodes; This means obtaining a numerical solution with less discretization error (Roache, 1998), improving the quality and reliability of the projects.

According to Trottenberg et al. (2001), experiments with the multigrid method show that its parameters (number of meshes, the smoothing or solver, the number of iterations in the solver, cycles, schedules and restriction and interpolation schemes) can have a strong influence on the efficiency of the algorithm . There are no general rules for choosing these parameters, but certain values may be recommended for certain situations. The rate of convergence depends on the choices made. A simple choice of the number of meshes to use can significantly affect computational time. This justifies the importance of studying the various parameters of the multigrid method. In the literature we find some work on the influence of parameters in the multigrid method, for example: Pinto et al. (2005) studied the optimal parameters of the multigrid for the diffusion, advection-diffusion and Burgers 1D equations; Rabi and De Lemos (2001) studied the optimal parameters for a 2D advection-diffusion; and Santiago and Marchi (2007) performed an analysis of the parameters for the Navier 2D problem involving two equations.

The objective this work is to verify the effect of the parameters on the CPU time for a multigrid method. The optimum value of a parameter is found when the solution is obtained in the lowest CPU time for fixed values of the other parameters. Such optimization involves the study of the multigrid components: the number of inner iterations of the solver (v), the number of grid levels (L), and also the type of restriction. In the present study, the term “CPU time” is used instead of “work units” due to the reasons pointed by Trottenberg et al. (2001) and Larsson et al. (2005).

The present work involves a linear three-dimensional heat conduction problem, governed by the Laplace equation with Dirichlet boundary conditions. The numerical method used is that of finite differences (Golub and Ortega, 1992; Tannehill et al., 1997) with CDS approximation and uniform meshes. This paper is organized as follows: in section 1, the mathematical models and discretizations are presented. In section 2 the numeric results. And section 3 concludes the paper.

1 MATHEMATICAL MODELS AND DISCRETIZATIONS

One three-dimensional Poisson equation problem in the unitary cube domain, with Dirichlet boundary conditions, is investigated in this work. It is governed by the following differential equation in the Cartesian coordinate system (Incropera et al., 2007):

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}=S, \tag{1}$$

where x, y and z are the coordinate directions, T is the temperature, and S is a source term. The Poisson equation has a source term, defined by

$$S=-3\pi^2.sen(\pi.x).sen(\pi.y).sen(\pi.z). \tag{2}$$

The Finite Difference Method (FDM) (Tannehill et al., 1997) is employed in the discretization of Eq. (1). It used second-order central differencing scheme (CDS). The unitary cube domain is divided into Nx nodes in the x-direction, Ny nodes in the y-direction and Nz nodes in the z-direction, using uniform grids in each direction, totalizing N nodes $\left(N=Nx.Ny.Nz\right)$.

$$\frac{T_{i-1,j,k}-2T_{i,j,k}+T_{i+1,j,k}}{h^2}+\frac{T_{i,j+1,k}-2T_{i,j,k}+T_{i,j-1,k}}{h^2}+\frac{T_{i,j,k-1}-2T_{i,j,k}+T_{i,j,k+1}}{h^2}=S, \tag{3}$$

where h is the distance between two different nodes of the grid in any direction. Eq. (3) can be represented by one system of linear equations of the type:

$$AT = b, \tag{4}$$

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering
P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

which must be solved. By means of the discretization procedures applied, it can be noticed that the coefficients matrix *A* in Eq. (4) is heptadiagonal, symmetric and positive-definite; *T* is the solution temperature vector and *b* is the independent vector.

1.1 Multigrid method and computational details

The linear equation, Eq. (1), is solved by using the geometric multigrid method (GMG), as described by Wesseling (1992). It used correction scheme (CS). The V-cycle was chosen because of its simplicity for programming implementation and for its smaller computational work for isotropic problems.

Trilinear interpolation, injection operator and full weighting (Trottenberg et al, 2001) were chosen for all the studied cases. The solver algorithms used in this work are lexicographical and red-black Gauss–Seidel (Parter, 1988; Zhang, 1996). In all the numerical simulations, the number of grid levels (*L*) was taken in such way that 1 to $L_{\max}$, where $L_{\max}$ is the maximum number of different grids which can be employed in the multigrid cycle. Standard coarsening with ratio $r = 2$ is used.

Each V-cycle is repeated until the achievement of a given stop criterion, which is based on the non-dimensional $l2-norm$ of the residual – the reference is the $l2-norm$ of the initial guess – as found in (Trottenberg et al., 2001). The null-value was taken as the initial guess for the whole domain. The admitted tolerance was equal to 10^{-10} for all the analyzed cases in this work.

The numerical codes were generated by using the Fortran Visual Studio 2008, using quadruple precision. The simulations were realized in the microcomputer with processor Intel Core 2 Duo of 2.66 GHz, 8 GB of RAM and xp64 bits Windows operating system.

2 NUMERIC RESULTS

With the purpose of analyzing the influence of different multigrid components on the CPU time in one Poisson equation, about several numerical simulations were performed. The methodology employed consists in, for a given component of interest, keeping the other ones with a fixed value and, by comparison, choosing the set of components which have shown the best performance. Same methodology used in Oliveira et al., (2017). The numerical simulations belong to categories: number of smoothing steps, type restrictions, number of grids and comparison among solvers lexicographical and red-black Gauss-Seidel.

2.1 Number of smoothing steps (*v*)

The main purpose of this subsection is to establish the optimum number of smoothing steps $\left(\nu_{optimum}\right)$, which provides the lowest CPU time (t_{cpu}) for a given set of components. In order to reduce the number of numerical simulations and to achieve t_{cpu} minimization for all the dependent variables, the number of levels is fixed equal to $L_{\max}$, full weighting is chosen as the restriction operator and the lexicographical Gauss-Seidel (LEX-GS) and red-black Gauss-Seidel (RB-GS) as solvers.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering
P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

Figure 1 shows the influence of the number of smoothing (v) on CPU time for the 3D Poisson equation. For the lexicographical Gauss-Seidel solver, the number of smoothing that presented the lowest CPU time was $\nu_1 = \nu_2 = 3$, whereas for the red-black Gauss-Seidel solver it was $\nu_1 = \nu_2 = 2$. Notice that ν_1 and ν_2 are the numbers of pre- and post-smoothing, respectively (see Oliveira et al., 2017).

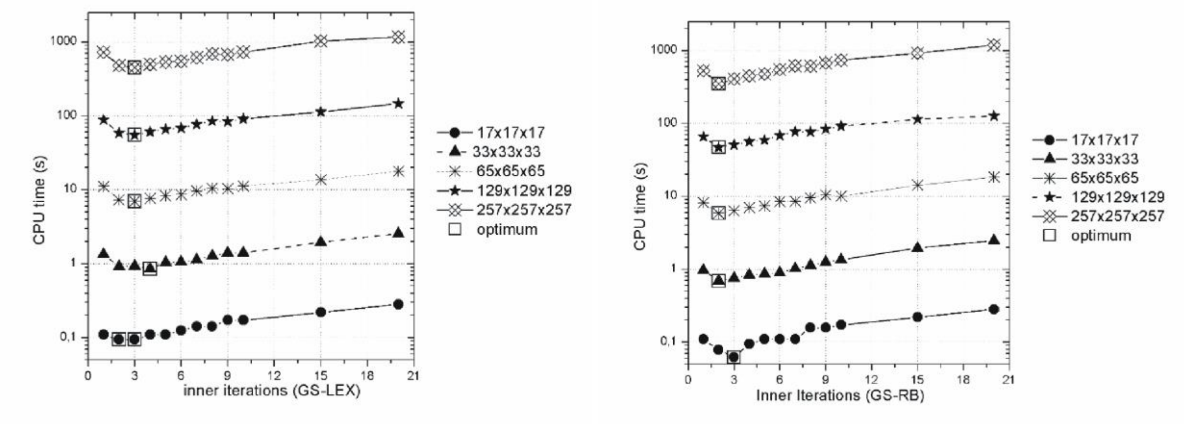

Figure 1: Analysis of number of smoothing for (a) lexicographical Gauss-Seidel and (b) red-black Gauss-Seidel, both using full weighting restriction

2.2. Type of restriction

Were considered two types of restriction: injection (INJ) and full weighting (FW). The injection restriction identifies and relates grid functions to coarse grid points and with their corresponding grid functions at fine grid points.

The influence of the number of smoothings (v) on CPU time using the two types of restriction was tested using the lexicographical Gauss-Seidel solver. The red-black Gauss-Seidel solver was tested only with full weighting restriction as it diverges when used with injection restriction and lexicographical ordination.

The optimum number of smoothings using injection restriction was $\nu_1 = \nu_2 = 5$, whereas for full weighting restriction it was $\nu_1 = \nu_2 = 3$. The lowest CPU time using the lexicographical Gauss-Seidel solver was obtained using injection restriction. For the 257x257x257 grid, it was about one time faster than the full weighting restriction (see Oliveira et al., 2017).

2.3. Grid levels (L)

Some authors, such as Suero et al. (2012) and Gaspar et al. (2009), have also analyzed the number of grid levels for problems involving the multigrid method. In the previous subsections, the number of grids used for the multigrid was kept invariable and equal to L_{max}. Since other components that influence CPU time performance have been previously studied, in this subsection only the effect of L on CPU time performance was evaluated.

Figure 2 summarizes the numerical results obtained of the effect of L on CPU time for the Poisson equation and red-black Gauss-Seidel solver. This solver was used because it results in lower CPU time than the lexicographical Gauss Seidel. It is easily observed that the lowest CPU time is achieved when using L_{max}. Similar results were achieved with lexicographical Gauss-Seidel solver.

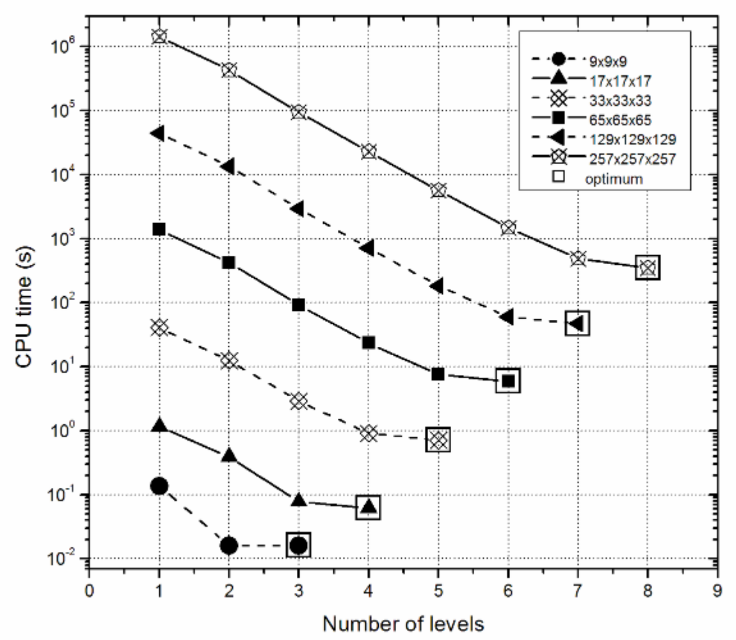

Figure 2: Number of levels versus CPU time for the grids ($N = 5x5x5$ a $N = 257x257x257$ *N* = 257x257x257) for the Poisson equation and red-black Gauss-Seidel solver

Results of the current work agree with those presented by Tannehill et al. (1997). These authors have investigated the existence of optimal components in multigrid method in a two-dimensional Laplace problem, using an isotropic 128x128 element grid, and from two to seven grids, and reported, regarding the CPU time, that the performance when using four or five grids was nearly the same as when using seven grids (which corresponds to L_{max}). According to the numerical results of the current work, the recommended value of L is always L_{max} (see Oliveira et al., 2017).

2.4. Comparison between solvers and complexity analysis

This subsection aims to compare how much CPU time is necessary to solve the 3D Poisson equation using the lexicographical Gauss-Seidel solver with injection restriction, and red-black Gauss-Seidel solver with weighting restriction. Such types of restriction were chosen as they result in lower CPU time, as shown in subsection 2.2.

Figure 3 shows the CPU time obtained with lexicographical Gauss-Seidel and red-black Gauss-Seidel solvers for the 3D Poisson equation. The CPU time obtained with the red-black

Gauss-Seidel solver is slightly lower than with the lexicographical Gauss-Seidel solver. For the 257x257x257 grid, it is about 1.17 times faster (see Oliveira et al., 2017).

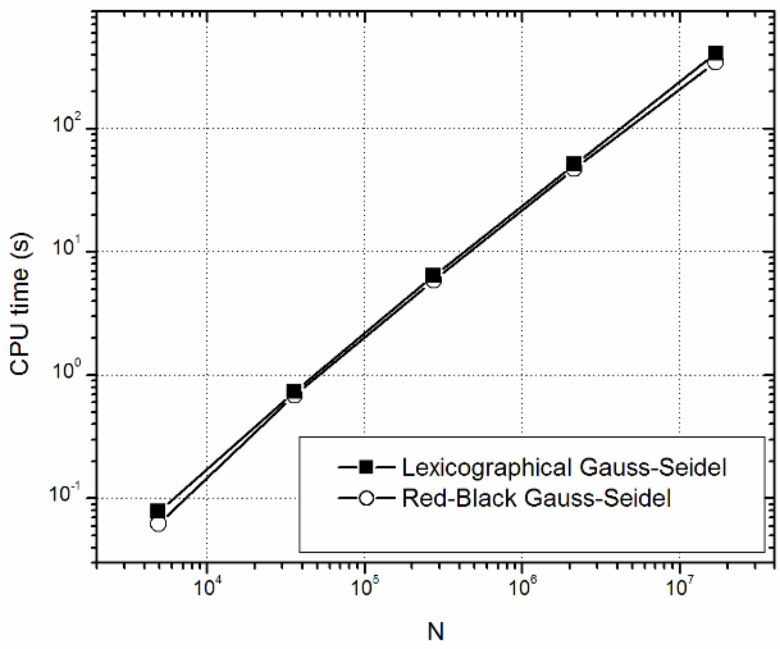

Figure 3: Comparison between lexicographical Gauss-Seidel (injection restriction) and Red-Black Gauss-Seidel (full weighting restriction)

Exponent *p*, which was obtained using the least square method, was computed for the function $t_{cpu}(N) = c\,N^p$, where *p* is the order solver associated with the method used, *c* is the coefficient that depends on each method and *N* is the number of unknowns of the system. For the ideal multigrid $p = 1$, which means that the computational effort increases linearly with the size of grid (Trottenberg et al., (2001) and Hirsh (1988)). So, for a given hardware and compiler, as the value of *p* decreases, the efficiency of the algorithm increases.

Table 1 shows the coefficient for the geometric curve fitting for the multigrid method. The best result was obtained using the multigrid method with red-black Gauss-Seidel solver and full weighting restriction (see Oliveira et al., 2017).

The CPU time necessary to solve the 3D Poisson equation using the singlegrid and multigrid methods was assessed. The red-black Gauss-Seidel solver was used in both cases. Table 2 shows the coefficient for the geometric curve fitting for the single grid and multigrid methods, both using the red-black Gauss-Seidel solver, which is in agreement with the literature. For more refined grids, the coefficient *p*, on the multigrid method, tends to one (see Oliveira et al., 2017).

Table 1. Coefficient for the geometric curve fitting	c	p
Multigrid Method (lexicographical Gauss-Seidel) full weighting restriction.	1.5261E-05	1.03558
Multigrid Method (lexicographical Gauss-Seidel) Injection Restriction.	1.16102E-05	1.04817
Multigrid Method (red-black Gauss-Seidel)	9.53101E-06	1.05397

Table 2. Coefficient for the geometric curve fitting

Method	c	p
Multigrid Method (red-black Gauss-Seidel)	9. 3101E-06	1.05397
Singlegrid Method (red-black Gauss-Seidel)	1.60094E-06	1.64497

3 CONCLUSION

This work aimed to assess the effect of multigrid components on CS scheme. For the numerical analysis, three-dimensional problem governed by Poisson equation, with Dirichlet boundary conditions, were employed and discretized with the Finite Difference Method and second order CDS approximations. Based on the results of this work, these are the most interesting remarks:

1) The number of smoothings influences CPU time. This parameter depends on the solver and restriction type. It was obtained:
    a) $\nu = 5$ as the optimum number for the lexicographical Gauss-Seidel solver with injection restriction;
    b) $\nu = 3$ and as $\nu = 2$ the optimum number for the lexicographical Gauss-Seidel solver and red-black Gauss-Seidel solver respectively. Both using full weighting restriction.
2) The CPU time obtained with red-black Gauss-Seidel solver is slightly faster than with lexicographical Gauss-Seidel solver. For the 257x257x257 grid, it is about 1.17 times faster.

3) The number of levels influences CPU time. The lowest CPU time was obtained with L_{max}, regardless of the solver and restriction type used.
4) The CPU time obtained with multigrid method is lower than with single grid method. The orders of complexity obtained from these two methods are 1.05397 and 1.64497, respectively.

PERMISSION

We are responsible for making sure that we have the right to publish everything in our paper.

ACKNOWLEDGEMENTS

This work was partially supported by the Brazilian agencies: CNPq (National Council for Scientific and Technological Development - Brazil) and CAPES (Coordination for the Improvement of Higher Education Personnel - Brazil). The first author acknowledges the State University of Ponta Grossa for the financial support.

REFERENCES

Brandt, A., 1977. Multi-level adaptive solutions to boundary-value problems. *Math. Comput*. n. 31, pp. 333–390.

Briggs, W.L., Henson, V.E. & McCormick, S.F., 2000. *A Multigrid Tutorial*. second ed., - Philadelphia:SIAM.

Gaspar, F.J., Gracia, J.L., Lisbona, F.J. & Rodrigo, C., 2009. On geometric Multigrid methods for triangular grids three-coarsening strategy. *Appl. Numer. Math*. n. 59, pp.1693-1708.

Golub, G. H. & Ortega, J.M., 1992. *Scientific Computing and Differential Equations: an Introduction to Numerical Methods*. Academic Press, Inc., 1992.

Hirsch, C., 1988. *Numerical Computational of Internal and External Flows*. vol. 1, – Chichester: John Wiley & Sons.

Incropera, F.P., DeWitt, D.P., Bergman, T.L. & Lavine, A.S., 2007. *Fundamentals of Heat and Mass Transfer*. Sixth ed., John Wiley & Sons.

Larsson, J., Lien, F.S., & Yee, E., 2005. Conditional Semicoarsening Multigrid Algorithm for the Poisson Equation on Anisotropic Grids. *J. Comput. Phys*. n. 208, pp.368-383.

Oliveira, F, Franco, S.R., & Pinto, M.A.V., 2017. Effect of Multigird Parameters in 3D Heat Diffusion Equation – *International Journal of Applied Mechanics and Engineering.* (in print).

Parter, S.V., 1988. Estimates for Multigrid Methods Based on Red-Black Gauss-Seidel Smooth. *Numer. Math*. n. 52, pp.701-723.

Pinto, A.M., Santiago, C.D. & Marchi, C.H., 2005. Effect of Parameters of a Multigrid Method on CPU Time for One-dimensional Problems. *Proceedings of COBEM*.

Rabi, J.A., & De Lemos, M.J.S., 2001. Optimization of convergence acceleration in multigrid numerical solutions of conductive–convective problems. *Appl. Math. Comput.* n.124, pp. 215-226.

Roache, P.J., 1998. *Fundamentals of Computational Fluid Dynamics*. Albuquerque, USA: Hermosa Publishers.

Santiago, C.D. & Marchi, C.H., 2007. Optimum Parameters of a Geometric Multigrid for a Two-Dimensional Problem of Two-Equations. *Proceedings of COBEM*.

Suero, R., Pinto, M.A.V., Marchi, C.H., Araki, L.K., & Alves, A.C., 2012. Analysis of the algebraic Multigrid parameters for two-dimensional steady-state diffusion equations. *Appl. Math. Modell.* n. 36, pp. 2996-3006.

Tannehill, J. C., Anderson, D.A. & Pletcher, R.H., 1997. *Computational Fluid Mechanics and Heat Transfer*. second ed. Washington: Taylor & Francis.

Trottenberg, U., Oosterlee, C. & Schüller A., 2001. *Multigrid*. San Diego: Academic Press.

Wesseling, P., 1992. *An Introduction to Multigrid Methods*. Philadelphia: John Wiley & Sons.

Wesseling, P., Oosterlee, C.W., 2001. Geometric Multigrid with Applications to Computational Fluid Dynamics. *Journal of Computation and Applied Mathematics*, vol. 128, pp. 311-334.

Zhang, J., 1996. *Multigrid Acceleration Techniques and Applications to the Numerical Solution of Partial Differential Equations*. Dissertation. Chongqing Univesity. China.