ENCIT 2018

ABCM Associação Brasileira de Engenharia e Ciências Mecânicas

17th Brazilian Congress of Thermal Sciences and Engineering
November 25th-28th, 2018, Águas de Lindóia, SP, Brazil

ENCIT-2018–0116
STUDY OF PARAMETERS OF THE MULTIGRID METHOD FOR THE SOLUTION OF 2D HEAT DIFFUSION PROBLEM USING NON-ORTHOGONAL STRUCTURED GRIDS

Daiane Cristina Zanatta
State University of Centro-Oeste, Irati-PR, Brazil.
daiaczanatta@gmail.com
Luciano Kiyoshi Araki
Marcio Augusto Villela Pinto
Federal University of Parana, Graduate Program in Numerical Methods in Engineering, Curitiba, PR, Brazil.
lucaraki@ufpr.br
marcio_villela@ufpr.br
Diego Fernando Moro
Positivo University, Curitiba, PR, Brazil.
diego.moro@up.edu.br

Abstract. *The purpose of this work was to study the behavior of the geometric multigrid method on the CPU time for a two-dimensional linear heat diffusion problem, governed by Poisson equation, with Dirichlet boundary conditions in a trapezoidal geometry. For this, two grid generation methods were used to generate the non-orthogonal structured grids. In the first method, algebraic method, Lagrange interpolation was used. The second one was based on solving a system of Laplace equations. The considered parameters that were assessed in order to reduce the CPU time are: number of inner iterations of the solver (ν), number of unknowns (N) and number of grids (L). The differential equation was discretized by the finite volume method (FVM) and two-dimensional uniform grid in each direction, with second order approximation scheme (CDS). The Dirichlet type boundary conditions were applied through ghost cell method. The system of algebraic equations was solved using lexicographical Gauss-Seidel smoothing. In order to accelerate the convergence of the iterative scheme, the multigrid method with Correction Scheme, V-cycle and coarsening ratio $r = 2$ was used. Results show that: $\nu_{optimum} = 2$ or 3 for grids generated by employing Lagrange interpolation and $\nu_{optimum} = 2$ or 4 for grids generated by employing differential equations; $L_{optimum} = L_{max}$ usually presents the lowest values of time CPU; the orders of complexity of the algorithm for with grids generated by employing Lagrange interpolation as well as a system of differential equations for the multigrid method are $p = 1.116$ and $p = 1.073$, respectively, which is in accordance with the theoretical efficiency of the multigrid method.*

Keywords: *CFD, geometric multigrid, finite volume method, non-orthogonal grids.*

1. INTRODUCTION

Many engineering problems concern to more complex geometries in which the use of the cartesian, cylindrical or spherical coordinate systems is not practical or appropriate.

The use of coordinate systems that coincide with the domain boundaries is an alternative for discretizing more complex geometries (Versteeg and Malalasekera, 2007). With this option, there is a global coordinate system and it is possible to write the differential equations that will be solved, as well as discretize them, in it (Maliska, 2010).

In this new system, the discretization of mathematical models by finite volume method (FVM) approximates a differential equation system by means of an algebraic equations system of the form of

$$A\phi = f \tag{1}$$

where A is the matrix of coefficients, ϕ is the vector of unknowns and corresponds to the approximate solution in each point of the grid, and f is the vector of independent terms.

Solving algebraic systems for computational fluid dynamics problems by means of numerical methods requires, in many occasions, a high computational cost due to the high number of equations solved at every iterative step. Although the convergence rate is high in the beginning of the calculations, it considerably decreases as the iterative process advances (Briggs *et al.*, 2000). In order to improve the convergence rate of numerical methods, the multigrid method is used.

The multigrid method refers to a family of iterative methods used to solve, efficiently, algebraic equation systems as expressed by Eq. (1). This method sweeps, during the iterative process, several grids with different refinement levels, thus sweeping every frequency error (smooth or oscillatory) and smoothing the error more effectively. Restriction operators are used to transfer information from finer grids to immediately coarser grids and prolongation operators are used to transfer information from coarser grids to immediately finer grids (Briggs *et al.*, 2000). With a smoother, iterations are performed at each grid level until a specific convergence criterion for the system of equations in the finer grid is reached. Therefore, at each grid, different error components are efficiently smoothed thus accelerating the convergence of the iterative process used in the resolution of the systems (Tannehill *et al.*, 1997).

This work focuses on the study of parameters of the geometric multigrid method for Poisson equation with trapezoidal domain and grids generated using elliptic differential equations and Lagrange interpolation. The parameters that will be assessed in order to reduce the CPU time are: number of inner iterations of the smoother, number of grids and number of unknowns.

2. GENERATING THE CURVILINEAR COORDINATE SYSTEM

Several methods are used to generate grids. These methods can be classified as algebraic or differential. Algebraic methods utilize different types of interpolation and are versatile as well as fast (Thompson *et al.*, 1985; Maliska, 2010). Differential methods, on the other hand, employ a system of differential equations and can be used more broadly, however, they need more mathematical elaboration and have a higher computational time (Maliska, 2010). In this work, both methods were used. In order to generate the grids by means of the the algebraic and differential methods, Lagrange interpolation and a system of Laplace equations were used, respectively (Maliska, 2010).

Figure 1, for example, depicts a grid generated using Lagrange interpolation for a trapeze-shaped domain.

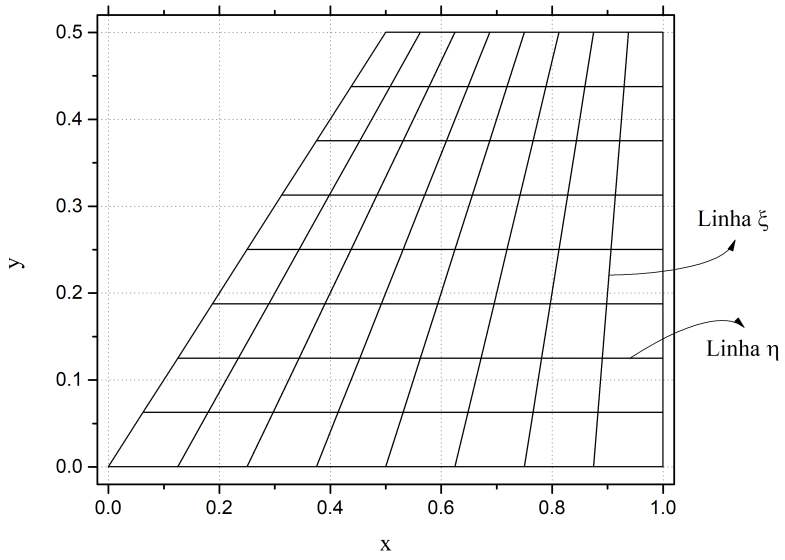

Figure 1: Grid generated using Lagrange interpolation with 8 volumes in the ξ and η directions.

3. MATHEMATICAL AND NUMERICAL MODELS

3.1 Mathematical models

The mathematical problem considered concerns to the two-dimensional linear heat steady diffusion problem, in cartesian coordinates, governed by Poisson equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = S. \tag{2}$$

In Eq. (2), the variables x and y are the spatial coordinates and T is the temperature. The computational domain of this problem, shown in Fig. 2, is defined by

$$\begin{cases} 0 \leq x \leq 0.5 & \text{if} \quad 0 \leq y \leq x; \\ 0.5 < x \leq 1 & \text{if} \quad 0 \leq y \leq 0.5 \end{cases} \tag{3}$$

and

$$\begin{cases} 0 \leq y \leq x & \text{if} \quad 0 \leq x \leq 0.5; \\ 0 \leq y \leq 0.5 & \text{if} \quad 0.5 < x \leq 1 \end{cases} \tag{4}$$

and the source term is $S = -\frac{\pi^2}{2} sin\left(\frac{\pi x}{2}\right) sin\left(\frac{\pi y}{2}\right)$.

The Dirichlet type boundary conditions are given by

$$\begin{cases} T(x,0) = 0 & \text{if} \quad 0 \leq x \leq 1 \\ T\left(x, \frac{1}{2}\right) = \frac{\sqrt{2}}{2} sin\left(\frac{\pi x}{2}\right) & \text{if} \quad \frac{1}{2} \leq x \leq 1 \\ T\left(x, x\right) = sin^2\left(\frac{\pi x}{2}\right) & \text{if} \quad 0 \leq x \leq \frac{1}{2} \\ T(1, y) = sin\left(\frac{\pi y}{2}\right) & \text{if} \quad 0 \leq y \leq \frac{1}{2} \end{cases} . \tag{5}$$

The analytical solution is given by

$$T(x,y) = sin\left(\frac{\pi x}{2}\right) sin\left(\frac{\pi y}{2}\right). \tag{6}$$

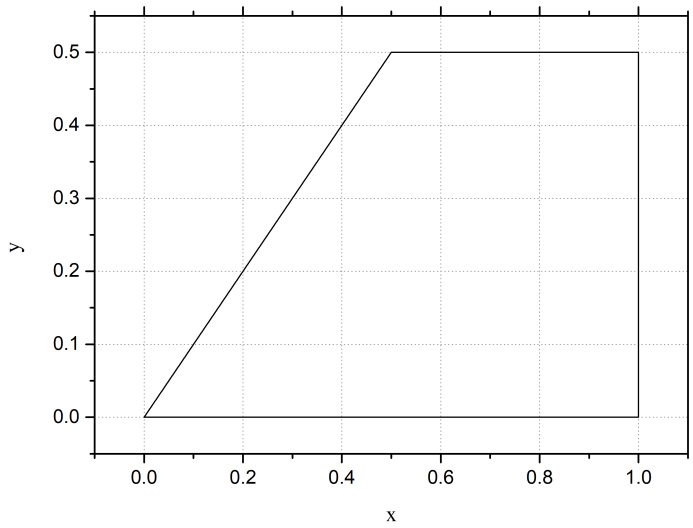

Figure 2: Domain of calculation.

Based on the following coordinate transformation

$$\begin{cases} \xi = \xi(x,y) \\ \eta = \eta(x,y) \end{cases} \tag{7}$$

and using the chain rule and some algebraic manipulations, Eq. (2) can be written in the system (ξ, η) as

$$\frac{\partial}{\partial \xi}\left[J\left(\alpha\frac{\partial T}{\partial \xi} - \beta\frac{\partial T}{\partial \eta}\right)\right] + \frac{\partial}{\partial \eta}\left[J\left(\gamma\frac{\partial T}{\partial \eta} - \beta\frac{\partial T}{\partial \xi}\right)\right] = \frac{S}{J}. \tag{8}$$

The jacobian matrix of the coordinate transformation is given by

$$J_1 = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix}, \tag{9}$$

whose elements are the metrics of the system transformation (x, y) for the system (ξ, η), obtained by the inverse function theorem. Here ξ_x represents the derivative of ξ with respect to x. The other simbols have similar definitions. These metrics are necessary in the transformed equation. The jacobian matrix of the inverse transformation is given by

$$J_2 = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}. \tag{10}$$

The jacobian matrices, given by Eqs. (9) and (10), are related as follows

$$J_1 = J_2^{-1}. \tag{11}$$

Thus, the Jacobian J of the transformation is related to the determinant of the jacobian matrix of the inverse transformation as follows

$$J = \det(J_1) = \frac{1}{\det(J_2)} = (x_\xi y_\eta - y_\xi x_\eta)^{-1}. \tag{12}$$

3.2 Numerical model

Integrating Eq. (8) on the control volume P in the transformed plane, as shown in Fig. 3, we have the expression

$$\left[J\left(\alpha\frac{\partial T}{\partial \xi} - \beta\frac{\partial T}{\partial \eta}\right)\right]_w^e \Delta\eta + \left[J\left(\gamma\frac{\partial T}{\partial \eta} - \beta\frac{\partial T}{\partial \xi}\right)\right]_s^n \Delta\xi = \frac{S}{J}\Delta\xi\Delta\eta, \tag{13}$$

where w, e, s and n are the faces west, east, south and north of the volume P, respectively.

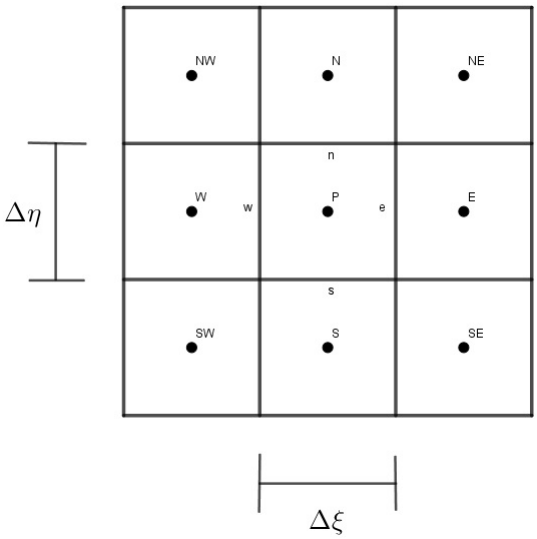

Figure 3: Control volume.

Approximating the derivatives of Eq. (13) means of central differences, the following equation is obtained

$$\begin{aligned} &J_e\alpha_e\frac{T_E - T_P}{\Delta\xi}\Delta\eta - J_w\alpha_w\frac{T_P - T_W}{\Delta\xi}\Delta\eta - J_e\beta_e\frac{T_N + T_{NE} - T_S - T_{SE}}{4\Delta\eta}\Delta\eta + \\ &J_w\beta_w\frac{T_N + T_{NW} - T_S - T_{SW}}{4\Delta\eta}\Delta\eta + J_n\gamma_n\frac{T_N - T_P}{\Delta\eta}\Delta\xi - J_s\gamma_s\frac{T_P - T_S}{\Delta\eta}\Delta\xi - \\ &J_n\beta_n\frac{T_E + T_{NE} - T_W - T_{NW}}{4\Delta\xi}\Delta\xi + J_s\beta_s\frac{T_E + T_{SE} - T_W - T_{SW}}{4\Delta\xi}\Delta\xi = S_P\frac{\Delta\xi\Delta\eta}{J_P}. \end{aligned} \tag{14}$$

The Eq. (14) can be rewritten as

$$\begin{aligned} a_pT_p &= a_wT_W + a_eT_E + a_sT_S + a_nT_N + a_{sw}T_{SW} + a_{se}T_{SE} + a_{nw}T_{NW} + \\ &\quad a_{ne}T_{NE} + b_p = \sum_{nb} a_{nb}T_{NB} + b_P, \end{aligned} \tag{15}$$

where

$$a_w = J_w\alpha_w\frac{\Delta\eta}{\Delta\xi} + \frac{1}{4}J_n\beta_n - \frac{1}{4}J_s\beta_s, \tag{16}$$

$$a_e = J_e\alpha_e\frac{\Delta\eta}{\Delta\xi} - \frac{1}{4}J_n\beta_n + \frac{1}{4}J_s\beta_s, \tag{17}$$

$$a_s = J_s\gamma_s\frac{\Delta\xi}{\Delta\eta} + \frac{1}{4}J_e\beta_e - \frac{1}{4}J_w\beta_w, \tag{18}$$

$$a_n = J_n\gamma_n\frac{\Delta\xi}{\Delta\eta} - \frac{1}{4}J_e\beta_e + \frac{1}{4}J_w\beta_w, \tag{19}$$

$$a_{sw} = -\frac{1}{4}J_w\beta_w - \frac{1}{4}J_s\beta_s, \tag{20}$$

$$a_{se} = \frac{1}{4}J_e\beta_e + \frac{1}{4}J_s\beta_s, \tag{21}$$

$$a_{nw} = \frac{1}{4}J_w\beta_w + \frac{1}{4}J_n\beta_n, \tag{22}$$

$$a_{ne} = -\frac{1}{4}J_e\beta_e - \frac{1}{4}J_n\beta_n, \tag{23}$$

$$a_p = J_e\alpha_e\frac{\Delta\eta}{\Delta\xi} + J_w\alpha_w\frac{\Delta\eta}{\Delta\xi} + J_n\gamma_n\frac{\Delta\xi}{\Delta\eta} + J_s\gamma_s\frac{\Delta\xi}{\Delta\eta} = \sum_{nb=1}^{8} a_{nb}, \tag{24}$$

and

$$b_p = -S_p\frac{\Delta\xi\Delta\eta}{J_p}, \tag{25}$$

where nb represents the 8 volumes neighboring P.

In this work, the Dirichlet type boundary conditions are applied by means of ghost cells technique.

4. MULTIGRID METHOD

Basic iterative methods, such as Gauss-Seidel and Jacobi, are efficient in smoothing the high-frequency components in refined grids in the first few iterations. However, after some iterations the process becomes slow, signaling the predominance of low-frequency modes (Briggs *et al.*, 2000). At this moment, it is advisable to transfer the information to immediately coarser grid because the smooth modes of the error become more oscillatory and the smoothing scheme will be more efficient (Wesseling and Oosterlee, 2001; Briggs *et al.*, 2000; Trottenberg *et al.*, 2001).

The multigrid method sweeps through a set of grids with different spacing. With a smoother, iterations are performed at each grid level until a specific convergence criterion for the system of equations in the finer grid is reached. The sequence in which the various grids are visited is called the multigrid cycle. Details of V, W and F-cycles, among others, can be found in Wesseling and Oosterlee (2001) and Trottenberg *et al.* (2001).

The operators that transfer information from the Ω^h fine grid to the immediately Ω^{2h} coarser grid are called restriction operators, denoted generally by I_h^{2h}, where h and $2h$ denote fine and coarse grid, respectively. A common practice to restrict the residual, according to Kumar *et al.* (2009), and that was used in this work, is to add the relative residuals of the control volumes of the fine grid that correspond to those of the coarse grid, that is,

$$I_h^{2h}\phi^h = \phi_A^h + \phi_B^h + \phi_C^h + \phi_D^h. \tag{26}$$

Figure 4 depicts this procedure for the two-dimensional case.

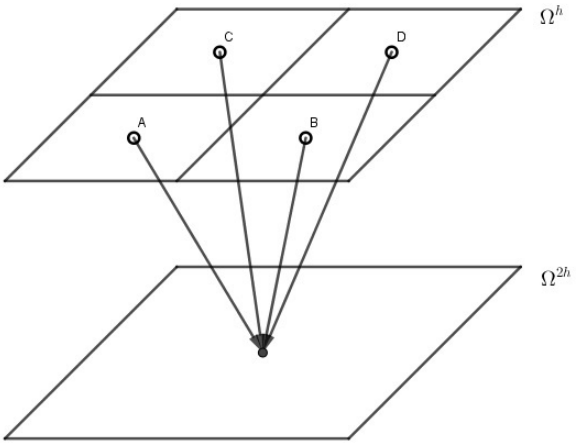

Figure 4: Restriction by the arithmetic mean of the values of the property of the four volumes of the fine grid.

Operators that transfer information from the Ω^{2h} coarse grid to the Ω^h finer grid are called prolongation operators, also known as interpolations, and are denoted generically by I_{2h}^h. The prolongation operator used in this work was the

bilinear interpolation operator. Bilinear interpolation is one of the most common prolongation operators in the literature (Wesseling and Oosterlee, 2001). The expressions for bilinear prolongation of property ϕ are given by

$$I_{2h}^{h}\phi^{2h} = \begin{cases} \dfrac{9\phi_A^{2h} + 3\phi_B^{2h} + 3\phi_C^{2h} + \phi_D^{2h}}{16} \text{ for } \phi_1^h \\ \dfrac{3\phi_A^{2h} + 9\phi_B^{2h} + \phi_C^{2h} + 3\phi_D^{2h}}{16} \text{ for } \phi_2^h \\ \dfrac{3\phi_A^{2h} + \phi_B^{2h} + 9\phi_C^{2h} + 3\phi_D^{2h}}{16} \text{ for } \phi_3^h \\ \dfrac{\phi_A^{2h} + 3\phi_B^{2h} + 3\phi_C^{2h} + 9\phi_D^{2h}}{16} \text{ for } \phi_4^h \end{cases} . \tag{27}$$

Figure 5 shows the prolongation operator with the bilinear interpolation for point 1.

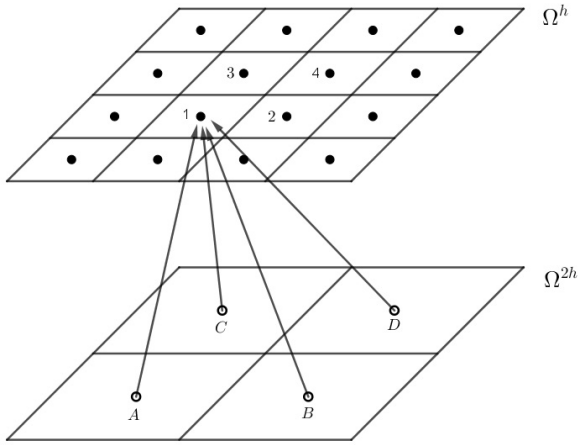

Figure 5: Transfer of information from the coarse grid to the fine grid with bilinear prolongation operator.

The multigrid method can be implemented with the Correction Scheme (CS) and Full Approximation Scheme (FAS). These schemes describe how the system of algebraic equations is operated in the coarser grids. In the CS only the residue is transferred to the coarser grids, where the residual equation is solved. In the FAS, the residue is transferred and the solution is approached for the coarser grids (Trottenberg *et al.*, 2001).

5. RESULTS

5.1 Implementation data

The algorithm was implemented in Fortran 95 language, using 11.1 Intel Compiler, with double precision. Simulations were performed in a microcomputer with Intel (R) Core (TM) i7-3770 processor, CPU 3.40 GHz, 8 GB RAM and Windows xp 64 bit operating system.

The system of algebraic equations represented by Eq. (15), for the domain given by Fig. 2, was solved by means of geometric multigrid method (Wesseling and Oosterlee, 2001; Briggs *et al.*, 2000; Trottenberg *et al.*, 2001), using correction scheme and V-cycle. Restriction was carried out by adding up the relative residuals of the control volumes of the fine grid that correspond to those of the coarse grid. Prolongation was done by means of bilinear interpolation. The coarsening ratio $r = 2$ was used, that is, the size of the coarser element is twice the size of the immediately finer element. The systems of algebraic equations were solved by using the lexicographical Gauss-Seidel smoothing (GS-Lex). The number of iterations used in pre-smoothing, ν_1, is the same as in post-smoothing, ν_2, that is, $\nu = \nu_1 = \nu_2$ where, pre- and post-smoothing is the number of smoothing in the restriction and prolongation process, respectively. The number of unknowns is given by $N = N_\xi N_\eta$, where N_ξ and N_η are the number of volumes in the directions ξ and η, respectively.

The stop criterion used to interrupt the iterative process is the L_1 norm of the residual in the current dimensionless iteration, $\mathbf{r}_{(k)}$, by the residual in the initial estimate, $\mathbf{r}_{(0)}$ (Trottenberg *et al.*, 2001)

$$\frac{||\mathbf{r}_{(k)}||_1}{||\mathbf{r}_{(0)}||_1} < \epsilon, \tag{28}$$

being $\epsilon = 10^{-12}$.

5.2 Number of inner iterations

For all the grids considered, simulations with a fixed number of inner iterations of the smoother, varying from $\nu = 1$ to $\nu = 15$, as well as with an isolated value, $\nu = 20$, were carried out in order to confirm the trend of the parameter. The

ν value that results in the lowest CPU time is considered the optimum number of inner iterations, denoted by ν_{optimum}. In all simulations performed for this parameter, $L = L_{\text{max}}$ was employed, in which L_{max} represents the highest possible number of grids that can be used for a particular problem, with only four real control volumes on the coarser grid (2×2). For example, for a problem with $N = 64 \times 64$ unknowns and coarsening ratio $r = 2$, we have the grids 64×64, 32×32, 16×16, 8×8, 4×4 and 2×2, therefore, the highest possible number of grids L_{max} is 6. The stop criterion is given by Eq. 28. Figure 6 depicts the influence of the number of inner iterations of the GS-Lex smoother on the Poisson equation with grids generated by employing Lagrange interpolation as well as a system of differential equations.

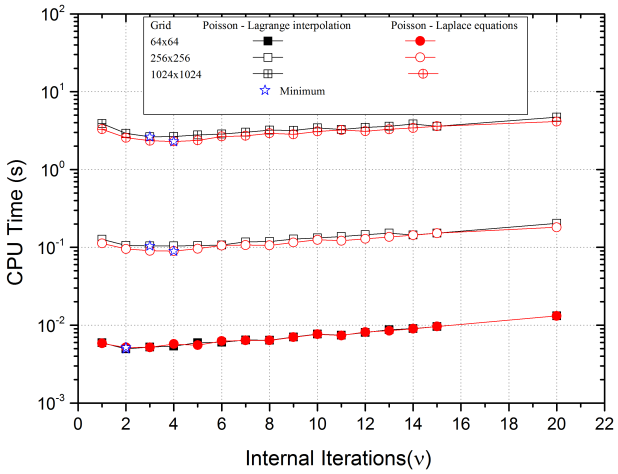

Figure 6: Effect of ν on CPU time.

As it can be observed in Fig. 6, after the ν_{optimum}, the CPU time shows a tendency to grow with the increase of the number of inner iterations. In the Poisson problem with grids generated by employing Lagrange interpolation as well as a system of differential equations, the lowest CPU times were obtained using $\nu_{\text{optimum}} = 2$ or 3 and $\nu_{\text{optimum}} = 2$ or 4, respectively. The optimum value adopted hereinafter for the Poisson equation with grids generated by employing Lagrange interpolation is $\nu_{\text{optimum}} = 3$ for all grids sizes. Considering $\nu_{\text{optimum}} = 3$ for the $N = 64 \times 64$ problem (this was the only case where the recommended optimal was different from the optimal found), the increase in CPU time was about 5% (corresponding to 0.000267 seconds). The optimum value adopted hereinafter for the Poisson equation with grids generated solving a system of differential equations is $\nu_{\text{optimum}} = 4$ for all grids sizes. Considering $\nu_{\text{optimum}} = 4$ for the $N = 64 \times 64$ problem (same as above), the increase in CPU time was about 10% (corresponding to 0.000553 seconds).

5.3 Number of grids

The study of the influence of the number of grids on the multigrid performance considers the ν_{optimum} obtained. Figure 7 shows the effect of the number of grids on the CPU time for the Poisson equation with grids generated by employing Lagrange interpolation as well as a system of elliptic equations.

Figure 7 shows that for the Poisson equation with grids generated by employing Lagrange interpolation $L_{\text{optimum}} = L_{\text{max}} - 1$ for the $N = 64 \times 64$ problem. For other problem sizes, the optimum value of number of grids is $L_{\text{optimum}} = L_{\text{max}}$. For the Poisson equation with grids generated by employing a system of elliptic equations $L_{\text{optimum}} = L_{\text{max}}$ regardless of the problem size. Moreover, for $L < L_{\text{optimum}} - 1$, the CPU time tends to grow significantly with the reduction of the number of grids. Therefore, L_{max} was used hereinafter for every problem. Considering $L_{\text{optimum}} = L_{\text{max}}$ for the Poisson equation with grids generated using Lagrange interpolation for the $N = 64 \times 64$ problem (this was the only case where the recommended optimal was different from the optimal found) the CPU time increased by about 0.2%.

5.4 Number of unknowns

Aiming to evaluate the influence of the number of unknowns of the system of equations on the CPU time, the results obtained for grids of 4×4 up to 1024×1024 control volumes, with correction scheme and singlegrid method, were assessed. Figure 8 presents the multigrid and singlegrid CPU times for the Poisson equation with grids generated by employing Lagrange interpolation and a system of elliptic equations.

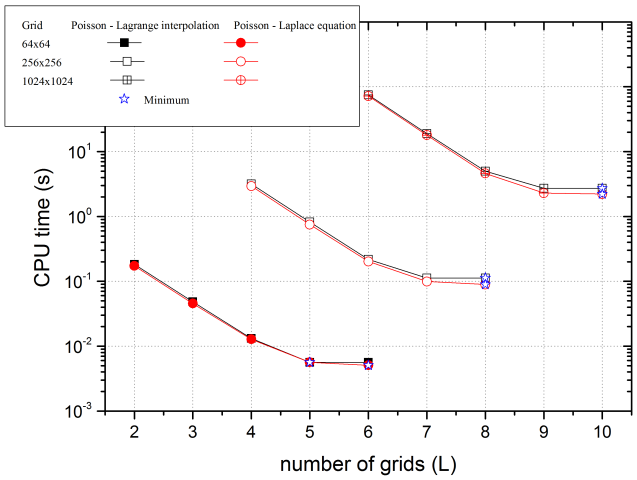

Figure 7: Effect of L on CPU time.

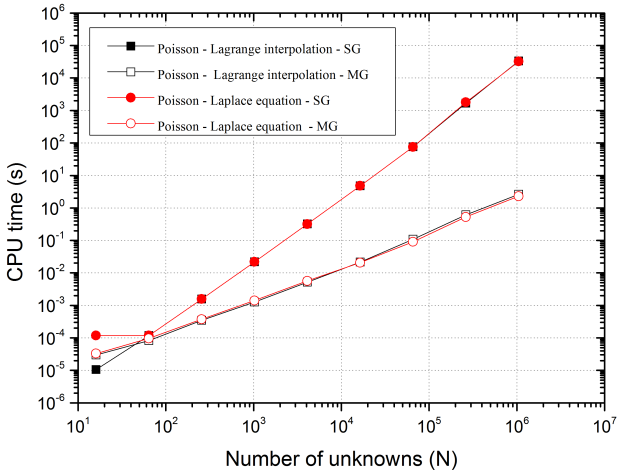

Figure 8: Number of unknowns.

According to Ferziger and Peric (2002), the finest the grid, that is, the higher the number of control volumes, the greater the advantage of the multigrid method over the singlegrid method.

In order to determine the orders of complexity of the algorithms concerning to each method as well as the behavior of the curve CPU time $\times$ N, the adjustment of a curve of the type

$$CPU(N) \text{ time } = cN^p, \tag{29}$$

was made, where N is the problem size, p is the order of complexity of the algorithm and c is a constant that depends on the method. To calculate the coefficient c and the exponent p, grids $N > 32 \times 32$ were considered. Table 1 depicts the values of c and p obtained by the adjustment of the curve, with correction scheme and GS-Lex smoother.

Table 1: Coefficient c and exponent p for the singlegrid and multigrid methods

Problem	MG		SG	
	c	p	c	p
Poisson - Lagrange interpolation	4.918E-07	1.116	1.234E-08	2.052
Poisson - Laplace equations	7.330E-07	1.073	1.221E-08	2.054

The values obtained for the exponent p by adjusting the curves are as expected, that is, close to 1 for the multigrid method and close to 2 for the singlegrid method (Wesseling and Oosterlee, 2001; Briggs *et al.*, 2000; Trottenberg *et al.*, 2001).

6. CONCLUSIONS

The investigation of the parameters of the geometric multigrid method for the Poisson equation with grids generated by employing Lagrange interpolation as well as a system of differential equations, with correction scheme and GS-Lex smoothing, verified that:

1. The optimum number of inner iteration for the Poisson equation with grids generated by employing Lagrange interpolation was $\nu_{\text{optimum}} = 2$ or 3, with recommended value 3. For the Poisson equation with grids generated by employing differential equations, the optimum number was $\nu_{\text{optimum}} = 2$ or 4, with recommended value 4. CPU time can increase considerably when using values different than the optimum number of inner iterations.

2. For the Poisson equation with grids generated by employing Lagrange interpolation, $L_{\text{optimum}} = L_{\text{max}} - 1$ or $L_{\text{optimum}} = L_{\text{max}}$, with recommended value $L_{\text{optimum}} = L_{\text{max}}$. For the Poisson equation with grids generated by employing differential equations, $L_{\text{optimum}} = L_{\text{max}}$ for the all grid sizes. For $L < L_{\text{optimum}}$, the CPU time increases considerably as the number of grids employed decreases.

3. The orders of complexity of the algorithm for the Poisson equation with grids generated by employing Lagrange interpolation as well as a system of differential equations, for the multigrid method are $p = 1.1160$ and $p = 1.073$, respectively; for the the singlegrid method they are $p = 2.052$ and $p = 2.054$, respectively, which is in accordance with the theoretical efficiency of the multigrid method.

7. ACKNOWLEDGEMENTS

The first author would like to thank the State University of Centro-Oeste (UNICENTRO) for granting a full leave of absence from teaching activities. The second author thanks CNPq for project financial resources. The authors acknowledge the infrastructure and support of the Numerical Experimentation Laboratory (LENA) of the Federal University of Parana (UFPR) and the Graduate Program in Numerical Methods in Engineering.

8. REFERENCES

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

Ferziger, J.H. and Peric, M., 2002. *Computational Methods for Fluid Dynamics*. Springer, New York, USA, 3rd edition.

Kumar, D., Suresh Kumar, K. and Kumar Das, M., 2009. "A fine grid solution for a lid-driven cavity flow using multigrid method". *Engineering Applications of Computational Fluid Mechanics*, Vol. 3, No. 3, pp. 336–354.

Maliska, C.R., 2010. *Transferência de Calor e Mecânica dos Fluidos Computacionais*. LTC, Rio de Janeiro, 2nd edition.

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

Thompson, J.F., Warsi, Z.U. and Mastin, C.W., 1985. *Numerical grid generation: foundations and applications*, Vol. 45. Elsevier North-Holland, New York, USA.

Trottenberg, U., Oosterlee, C.W. and Schuller, A., 2001. *Multigrid*. Academic Press, San Diego.

Versteeg, H.K. and Malalasekera, W., 2007. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, Harlow, England.

Wesseling, P. and Oosterlee, C.W., 2001. "Geometric multigrid with applications to computational fluid dynamics". *Computational and Applied Mathematics*, Vol. 128, No. 1, pp. 311–334.

9. RESPONSIBILITY NOTICE

The authors are the only ones responsible for the printed material included in this paper.