OPTIMUM PARAMETERS OF A GEOMETRIC MULTIGRID FOR THE TWO-DIMENSIONAL LAPLACE'S EQUATION

Márcio Augusto Villela Pinto, marciovp@demec.ufpr.br
Department of Mathematics and Statistics, State University of Ponta Grossa, Ponta Grossa, PR, Brazil

Carlos Henrique Marchi, marchi@ufpr.br
Department of Mechanical Engineering, Federal University of Paraná, Curitiba, PR, Brazil

Abstract. *On the necessary CPU time to solve a problem of heat transfer, one verifies the effect caused by: coarsening ratio (r); number of nodes (N) and grids (L); number of inner iterations (ITI); and correction (CS) and full approximation schemes (FAS) of a geometric multigrid method with V-cycle. The considered problem involves a two-dimensional linear problem, Laplace's equation, with Dirichlet boundary conditions. The finite difference method is used to discretizate the differential equation with central difference scheme scheme for uniform grids. The systems of algebraic equations are solved with MSI solver. It was verified that the minor CPU time in general occurs with: a) FAS scheme and r =3, that it results approximately in 25% of the CPU time occurring with CS scheme and r = 2; b) for FAS scheme, ITI = 3 or 4 with r = 2 or 3; and c) using the maximum number of grids.*

Keywords*: solver, CFD, finite difference method, heat transfer, numerical methods*

1. INTRODUCTION

Mathematical models in the computational fluid dynamics occur in physical phenomena that involve fluids in movement, with or without heat transfer (Fortuna, 2000; Maliska, 2004). These mathematical models, in general, do not have known analytical solutions. Then one searches numerical solutions transforming the continuous model into a discrete model. The discretization method used here is the finite difference method (Tannehill *et al.*, 1997), where the domain $\{(x,y) \in \Re^2 : 0 \le x, y \le 1\}$ is partitioned in a number of nodes (*N*), given by

$$N = N_x N_y \tag{1}$$

where N_x and N_y are the number of nodes in coordinate directions *x* and *y* of a grid Ω^h, respectively, with its coordinates given by

$$(x_i, y_j) = ((i-1)h_x, (j-1)h_y), \text{ with } h_x = \frac{1}{N_x - 1} \text{ and } h_y = \frac{1}{N_y - 1} \tag{2}$$

where $i = 1,...,N_x$, $j = 1,...,N_y$ and h_x and h_y are the lengths of each element in the coordinate directions.

The discretization of these mathematical models leads to large systems of algebraic equations of the type

$$\mathbf{A}\vec{T} = \vec{f} \tag{3}$$

where **A** is a square matrix with *N* x *N* dimension, $\vec{f}$ is the independent vector and $\vec{T}$ is the variable vector.

Several numerical techniques have been studied to solve the system given by Eq. (3) with the accuracy solution and minor CPU time (Trottenberg *et al.*, 2001). This requires a high and impracticable computational cost because of the large number of equations to be solved in each iterative step. The resolution by direct methods is not recommendable because of the large size of coefficients matrix and its high cost for inversion (Golub and Van Loan, 1989). For large scale problems, the iterative methods are recommended (Burden and Faires, 1997).

The multigrid method, studied originally by Fedorenko (1964), speeds up the resolution of the systems of equations given by Eq. (3). The basic idea of the method is to use a set of grids and to execute iterations in each level of grid, in order to approximate the solutions of this equation in coarser grids (Briggs *et al.*, 2000). Operators to transfer information (residue or solution) from the fine grid to the coarse grid (restriction) and vice-versa (prolongation) are used. The system of equations is solved with an iterative method (called here solver) in each grid. To get a good performance of multigrid, several grids must be used (Tannehill *et al.*, 1997). Pinto *et al.* (2005) recommend the use of the maximum possible number of level grids.

Two schemes can be used in multigrid (Briggs *et al.*, 2000): the Correction (CS) and the Full Approximation Schemes (FAS). In CS scheme, the Eq. (3) is solved only in the finest grid; in the coarser grids, it is solved the residual equation (Briggs *et al*., 2000), given by

$$\vec{R} = \vec{f} - \mathbf{A}\vec{T} \tag{4}$$

In FAS scheme, Eq. (3) is solved in all grids. CS scheme is generally used for linear problems and FAS scheme for nonlinear problems (Brandt, 1977). However Yan and Thiele (1998) and Mesquita and de-Lemos (2004) used a variant of the CS scheme for the resolution of the Navier-Stokes equation (nonlinear problem) and Souza *et al.* (2006) used FAS scheme for the resolution of the Poisson equation (linear problem).

The coarsening ratio (*r*), for two-dimensional problems, considering $h = h_x = h_y$, is defined by

$$r = \frac{h_2}{h_1} \tag{5}$$

where h_1 represents the size of the elements of the fine grid Ω^h and h_2 the size of the elements of the immediately coarser grid Ω^H. Brandt (1977) recommends the use of the coarsening ratio $r = 2$, by being around to the optimum one. Briggs *et al.* (2000) worked with the coarsening ratio $r = 2$, affirming that it is a universal practice and $r \neq 2$ does not any bring advantage.

Ferziger and Peric (1999) affirm that manipulations in the values of the parameters that can be selected in multigrid improve the convergence rate of the method by a factor around 2 between the worst and the best combination of these parameters. Tannehill *et al.* (1997) solve the 2D Laplace's equation with $N = 129\text{x}129$ and affirm that the use of 5 or 6 levels of grids results in almost the same performance of 7 grids. Hirsch (1988) recommends to use 4 or 5 grids. However, for the two schemes (CS and FAS), Pinto *et al.* (2005) and Oliveira *et al.* (2006) recommend to use the maximum number of grids for one-dimensional linear problems and coarsening ratios $r = 2$, 3, 4 and 5. In Oliveira *et al.* (2006) is made a study to find the optimum values of some parameters of the multigrid method, to minimize the CPU time for one-dimensional linear and nonlinear problems.

Several works (Montero *et al.*, 2001; Nishida and Satofuka, 1992; Sathyamurthy and Patankar, 1994) about multigrid method presented good numerical results with respect to convergence rate, for fluid dynamic problems. The ideal (theoretical) convergence rate of multigrid is independent of the number of nodes of the finest grid (Hirsch, 1988; Ferziger and Peric, 1999). The efficiency of the multigrid method has not been totally reached in realistic applications of engineering in computational fluid dynamics. This motivates to study the properties of the geometric multigrid method in problems modeled by simple equations. This also makes one to be able to elaborate efficient algorithms for including large class of problems, as advection-diffusion and two-dimensional termoelasticity problems and Poisson equation involved in the resolution of the cycle of the mass for the Navier-Stokes equation (Larsson *et al*. 2005; Souza *et al*., 2006).

The purpose of this work is to find optimum values of some parameters of the multigrid method, in order to minimize the CPU time, for a two-dimensional linear problem of heat conduction, governed by Laplace's equation, with Dirichlet boundary conditions. One also intends to verify if the results found for the one-dimensional case (Pinto *et al*., 2005; Oliveira *et al*., 2006) are extensive for the two-dimensional case. The following parameters are studied: coarsening ratio (*r*), number of variable (*N*), number of inner iterations (number of iterations of the iterative method, *ITI*), number of grids (level number of visited grids, *L*) and correction (CS) and full approximation schemes (FAS), both with V-cycle (Wesseling, 1992) without the use of Full Multigrid (FMG), that is, it is used the standard multigrid method that initiates the solution in the finest grid. The results are compared to the obtained ones in the bibliography. Operators of restriction by injection and bilinear interpolation for prolongation (Briggs *et al*., 2000; Trottenberg *et al.*, 2001) are employed as commonly used. By good smooth properties (Pinto *et al.*, 2006), the MSI solver (Schneider and Zedan, 1981) is used in this work.

This paper is organized as follows: in the section 2, the mathematical and numerical models are presented. In section 3, the numerical experiments and its results are described. The conclusion of this work is presented in section 4.

2. MATHEMATICAL AND NUMERICAL MODELS

The two-dimensional linear problem of heat conduction (Laplace's equation) in steady state with Dirichlet boundary conditions, in Cartesian coordinates, considered in this work is (Maliska, 2004)

$$\begin{cases} \dfrac{\partial^2 T}{\partial x^2} + \dfrac{\partial^2 T}{\partial y^2} = 0, \quad 0 < x, y < 1 \\ T(x,1) = \sin(\pi x), \quad T(x,0) = T(0,y) = T(1,y) = 0 \end{cases} \tag{6}$$

where T is the unknown and represents the temperature. The analytical solution of the problem is

$$T(x,y) = \sin(\pi x)\frac{\sinh(\pi y)}{\sinh(\pi)} \tag{7}$$

The discretization of the domain is made using uniform grids whose nodes are given by the Eq. (2). For each one of the $(N_x - 2)$ x $(N_y - 2)$ interior nodes of the grid, Eq. (6) is discretizated with the finite difference method with central difference scheme (CDS) (Tannehill *et al*., 1997), resulting in

$$\begin{cases} \dfrac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{h_x^2} + \dfrac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{h_y^2} = 0, \quad 2 \le i \le N_x - 1, \quad 2 \le j \le N_y - 1 \\ T_{i,N_y} = \sin(\pi x_{i,N_y}), \quad T_{i,1} = T_{1,j} = T_{N_x,j} = 0 \end{cases} \tag{8}$$

where $T_{i,j}$ is the numerical solution in each node (x_i, y_j). Rearranging the terms of the Eq. (8), it is obtained

$$a_{P,P}T_{i,j} + a_{P,n}T_{i,j+1} + a_{P,s}T_{i,j-1} + a_{P,w}T_{i-1,j} + a_{P,e}T_{i+1,j} = b_P \tag{9}$$

where the coefficients are given by $a_{P,P} = 2/h_x^2 + 2/h_y^2$, $a_{P,n} = a_{P,s} = -1/h_y^2$, $a_{P,w} = a_{P,e} = -1/h_x^2$, $b_P = 0$ and $P = (j-1)N_x + i$.

If $\vec{T}$ and $\vec{f}$ are denoted by $\vec{T} = (T_1,...,T_N)^t$ and $\vec{f} = (f_1,..., f_N)^t$, respectively, where $\vec{f}$ is the independent vector composed by the terms b_P, then the system of Eq. (9) can be represented by a system of algebraic equations of the type given by Eq. (3), where $\mathbf{A}$ is a pentadiagonal, symmetrical and positive-definite matrix N times N, (Briggs *et al.*, 2000).

Equation (3) is solved with multigrid method using CS and FAS schemes. The systems of equations of the type of Eq. (3) are solved with MSI solver, because this solver has obtained good smooth properties (Pinto and Marchi, 2006). In this work, it is adopted the geometric multigrid (Wesseling and Oosterlee, 2001).

The convergence criterion for the external iterations (*ITE*) (number of necessary V-cycles) is based on the ratio between *a* and *b*, where: *a* is the norm of the residue L_1 (Ferziger and Peric, 1999) in determined iteration; and *b* is the norm of the residue of the initial estimate. The average residue of each node is calculated by Eq. (4). In this work one adopts $\varepsilon = 10^{-7}$ and zero vector $\vec{T}$ for the tolerance and the initial estimate, respectively.

The algorithms have been implemented in FORTRAN 95 language with the use of the Visual Compaq Fortran 6.6 using double precision. The simulations have been carried out in a microcomputer with processor Intel Pentium 4 2.66 GHz with 1 GB of RAM.

The focus of this work is the minimization of the CPU time which is understood as the time spent to generate grids, to attribute the initial estimate, to calculate the coefficients and to solve the linear system of Eq. (3). This time is measured using the TIMEF function of PORTLIB library of FORTRAN 95. Through carried out tests, it was verified that the uncertainty of this function is approximately ± 0.05 s.

3. RESULTS

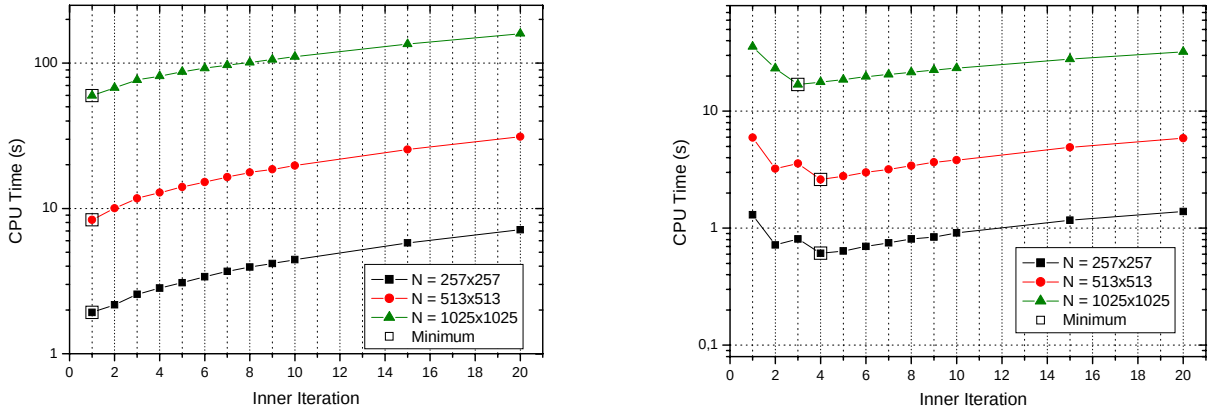
About 400 simulations have been carried out. Most representative results are presented in the sequence.

3.1. Inner iterations (*ITI*)

Figure 1a shows the influence of the number of inner iterations (*ITI*) on the CPU time for three different grids (*N*), $r = 2$, MSI solver and CS scheme with $L = L_{maximum}$. For $N = 513 \text{x} 513$ nodes, for example, $L_{maximum} = 9$, that is, using coarsening ratio *r* = 2 to solve the finest grid of 513 x 513 nodes (on level 1, the level of the finest grid, that is, Ω^h), using also the following grids: 257x257 (level 2, the level of the immediately coarser grid, that is, Ω^{2h}), 129x129 (level 3, that is, Ω^{4h}), 65x65 (level 4, Ω^{8h}), 33x33 (level 5, Ω^{16h}), 17x17 (level 6, Ω^{32h}), 9x9 (level 7, Ω^{64h}), 5x5 (level 8,

Ω^{128h}) and 3x3 (level 9, the level of the coarsest grid with only one inner node, that is, Ω^{256h}). It was verified that, for each finest grid, the minor CPU time occurs with the minor value of *ITI*, that is, the unit. Therefore, $ITI_{optimum}$ = 1, where $ITI_{optimum}$ is the value of *ITI* that results in the minor CPU time. Increasing the value of *ITI*, the CPU time also increases.

Figure 1b shows the influence of *ITI* for FAS scheme with $L = L_{maximum}$. It was verified that, for both 257x257 and 513x513 grids, $ITI_{optimum}$ = 4 and, for the 1025x1025 grid, $ITI_{optimum}$ = 3. Diminishing or increasing the value of *ITI*, in relation to the optimum one, implies in a significant CPU time increment.



a) CS scheme

b) FAS scheme

Figure 1. CPU time *versus* number of inner iterations (*ITI*) with MSI solver and $r = 2$.

Table 1 shows $ITI_{optimum}$ for the several values of *N* and for CS and FAS schemes for the several coarsening ratios $r = 2$, 3, 4 and 5. For FAS scheme, in the finest grid, one has $ITI_{optimum}$ = 3, 4 and 4, respectively for *r* = 3, 4 and 5. In the CS scheme case, $ITI_{optimum}$ does not vary with *N*. Therefore, the coarsening ratio *r* has a small influence on the $ITI_{optimum}$ value, that is more affected by the type of used scheme, CS or FAS.

Table 1. Optimum number of inner iterations ($ITI_{optimum}$) to MSI solver.

	N		$ITI_{optimum}$	
r	$N_{minimum}$	$N_{maximum}$	CS	FAS
2	257x257	1025x1025	1	3 or 4
3	163x163	1459x1459	2	3 or 4
4	129x129	2049x2049	2	4, 5 or 6
5	251x251	1251x1251	2	4 or 5

3.2. Number of grids (*L*)

Figure 2 shows the influence of the number of grids, or number of levels (*L*), on the CPU time with $r = 2$, MSI solver, for CS and FAS schemes with $N = 513 \times 513$ nodes. For this grid, the maximum number of grids is $L_{maximum} = 9$.

For CS scheme it was verified that for $ITI_{optimum}$ = 1, the optimum number of grids is $L_{optimum} = L_{maximum} = L_{maximum} - 1 = L_{maximum} - 2$, where $L_{optimum}$ is the *L* that results in the minor CPU time. Diminishing the value of *L*, in relation to $L_{optimum}$, generally implies in a significant CPU time increment.

For FAS scheme it was verified that for $ITI_{optimum}$ = 4 the optimum number of grids is $L_{optimum} = L_{maximum} - 2$, but with small difference in the CPU time for $L_{maximum}$. Like in the CS scheme, diminishing the value of *L*, in relation to $L_{optimum}$.

For both schemes (CS and FAS) and the other coarsening ratios, it was verified that $L_{optimum} \approx L_{maximum}$. This result is the same found by Pinto *et al.* (2005) in one-dimensional linear problems and coarsening ratios $r = 2$, 3, 4 and 5 with Gauss-Seidel (GS) solver and the same multigrid method. This result contradicts Hirsch (1988) who recommends the use of L = 4 or 5 grids. It is noticed by Fig. 2 (CS and FAS), that when adopting L = 4, the CPU time increases

approximately between 8 and 20 times, respectively. It also contradicts Ferziger and Peric (1999), who affirm that manipulations in the values of the parameters of multigrid improve its performance in a factor around 2. However, $L_{optimum} \approx L_{maximum}$ confirms the result of Tannehill *et al.* (1997) for the 2D Laplace's equation and $N = 129\text{x}129$, where $L_{maximum}$ = 7; in this case, Tannehill *et al.* (1997) affirm that the use of L = 5 or 6 obtain almost the same L = 7 performance.

Summarizing, it was verified that $L_{optimum} \approx L_{maximum}$. This means that the CPU time with $L_{optimum}$ is practically the same that with $L_{maximum}$. This result is general, based on results of literature and the authors of this themselves and other works, it was verified that it is valid for one or two dimensions, linear and nonlinear equations, CS and FAS schemes, coarsening ratios $r = 2$, 3, 4 and 5, any size of grid (N) and GS and MSI solvers.

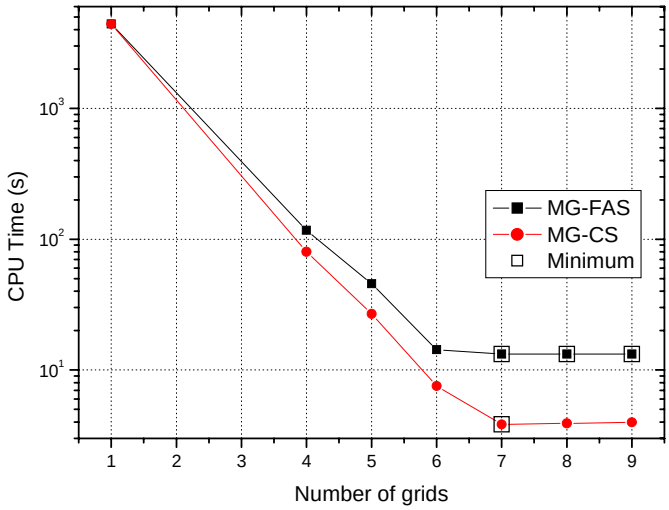


Figure 2. CPU time *versus* number of grids (L) for N = 513 x 513 with MSI solver and $r = 2$.

3.3. Coarsening ratio (r) and number of variables (N)

Figure 3 shows the influence of the number of variables (N) and of the coarsening ratio ($r = 2$, 3, 4 and 5) on the CPU time. Each point in this figure is an independent simulation that use the $ITI_{optimum}$ of each N, r and $L = L_{maximum}$. Results are also shown using only one grid (finest grid), called singlegrid (SG), with MSI solver. The used grids are: 5x5, 9x9,… until 513x513 (SG) or 2049x2049 (MG) nodes. In this figure, only points whose CPU time is not influenced by the uncertainty of its measurement are shown.

It was verified that the CPU time of the multigrid method (CS and FAS), with $N > 10^4$ and any coarsening ratio, is significantly lesser than singlegrid method. For example, for the CS scheme with r = 2 and grid 513x513 nodes, the CPU time for CS MG and SG are, respectively, 13.3 s and 7,006 s, that is, the CPU time of SG is 527 times CS MG one. For this same grid and r, the CPU time for FAS MG is 3.8 s; therefore, the CPU time of SG is 1,844 times FAS MG one. When N increases, the differences between MG and SG increase even more, because the inclinations of the curves with MG are lesser than SG.

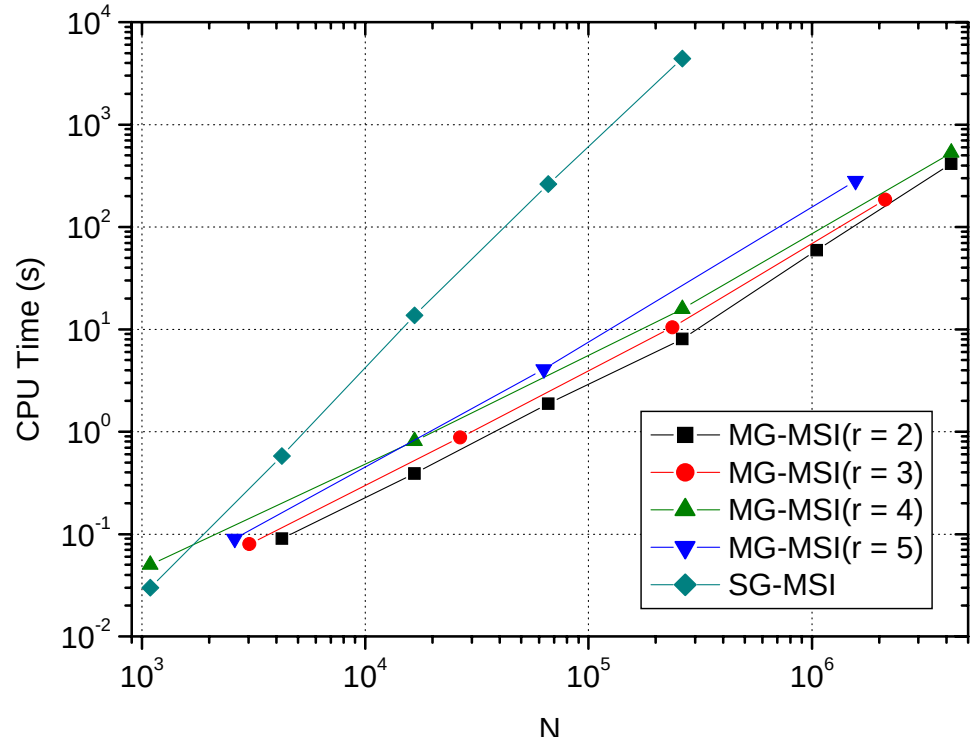
It was verified that

$$t_{CPU}(r=2) < t_{CPU}(r=3) < t_{CPU}(r=4) < t_{CPU}(r=5) < t_{CPU}(SG) \qquad \textbf{(CS scheme)} \qquad (10)$$

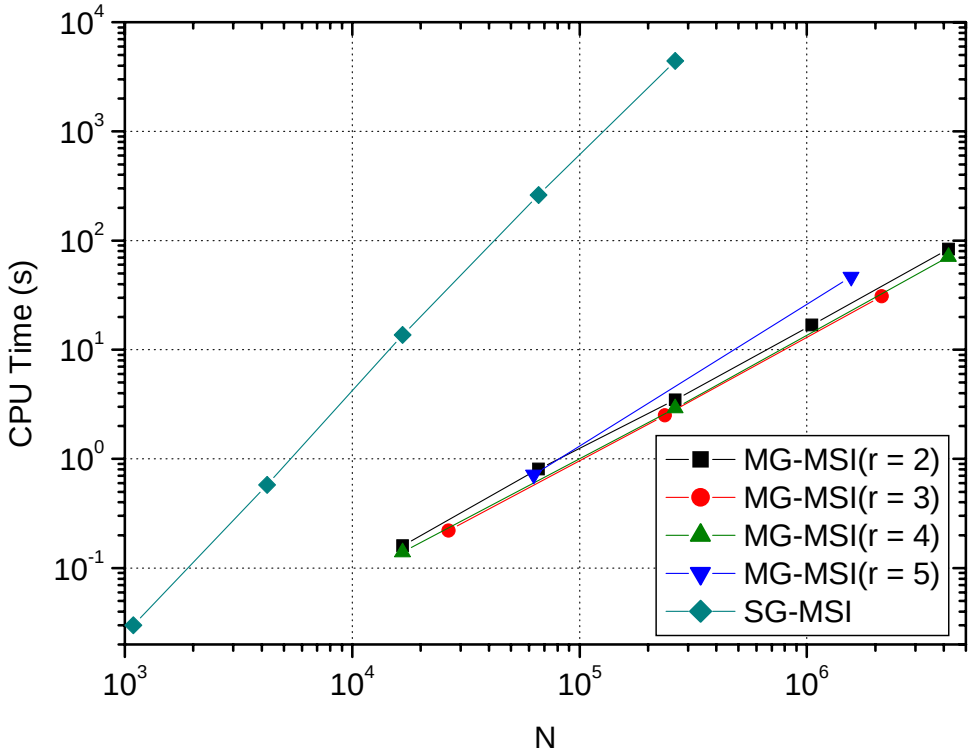
This result ratifies the results of Brandt (1977). Also it was verified that

$$t_{CPU}(r=3) < t_{CPU}(r=4) < t_{CPU}(r=2) < t_{CPU}(r=5) < t_{CPU}(SG) \qquad \textbf{(FAS scheme)} \qquad (11)$$

This result of the Eq. (11) shows that the standard coarsening ratio $r = 2$ is not the fastest in the FAS scheme, even for a linear equation. It must be noticed that the difference among the CPU times for the coarsening ratios $r = 2$, 3 and 4 in CS scheme is bigger than the ones of the FAS scheme, where these differences are small.



a) CS scheme



b) FAS scheme

Figure 3. CPU time *versus* number of variables (*N*) *versus r*.

For the points of Fig. 3, Tab. 2 presents the *c* coefficient and *p* inclination of the curves, obtained by geometric least square fitting method considering the following function

$$t_{CPU} = c\,N^{p} \tag{12}$$

The ideal multigrid method is that one whose p = 1, that is, that one whose CPU time increases linearly with the number of variables N. Therefore, for each numerical model, nearer p to unit, better the performance will be. Table 2 shows that p order is more affected by the adopted method, for singlegrid or multigrid, than by the coarsening ratio or the scheme (CS or FAS). For CS scheme, among the tested coarsening ratios, $r = 4$ is the ratio that results in a minor p order, equal to 1.12; and for FAS scheme, $r = 2$ for the same p.

Table 2. Coefficients (c) and order (p) of the Eq. (12).

	MG-CS		MG-FAS	
r	c	p	c	p
2	$2.90\text{x}10^{-6}$	1.22	$2.90\text{x}10^{-6}$	1.12
3	$5.85\text{x}10^{-6}$	1.18	$7.16\text{x}10^{-7}$	1.21
4	$1.72\text{x}10^{-5}$	1.12	$2.43\text{x}10^{-6}$	1.13
5	$4.26\text{x}10^{-6}$	1.26	$3.19\text{x}10^{-7}$	1.32
SG	$7.59\text{x}10^{-9}$	2.18	---	---

Summarizing, it was verified that for CS scheme, $r = 2$ is the coarsening ratio that results in the minor CPU time for one given grid with N points; and for FAS scheme, $r = 3$.

3.4. CS *versus* FAS scheme

Figure 4 shows a comparison between the CS (with its best coarsening ratio, $r = 2$) and FAS scheme (with its best coarsening ratio $r = 3$, and the standard ratio, $r = 2$) in function of several values of N. One can note that FAS scheme is faster than CS scheme for any N and r. For example, for $N = 10^5$, the MG-FAS (r = 3), MG-FAS (r = 2) and MG-CS (r = 2) CPU times are, respectively, 0.85 s, 1.16 and 3.45 s, that is, the MG-CS (r = 2) and the MG-FAS (r = 2) CPU times are, respectively, 4.1 and 1.4 time the MG-FAS (r = 3) CPU time. Another example: for $N = 10^6$, the MG-FAS (r = 3), MG-FAS (r = 2) and MG-CS (r = 2) CPU times are, respectively, 13.87 s, 15.22 s and 56.65 s, that is, the MG-CS (r = 2) and the MG-FAS (r = 2) CPU times are, respectively, 4.1 and 1.1 time of the MG-FAS (r = 3) CPU time. This verification is unexpected because the CS scheme is indicated to solve linear equations, as the equation of this work, and FAS scheme, for nonlinear equations (Brandt, 1977).

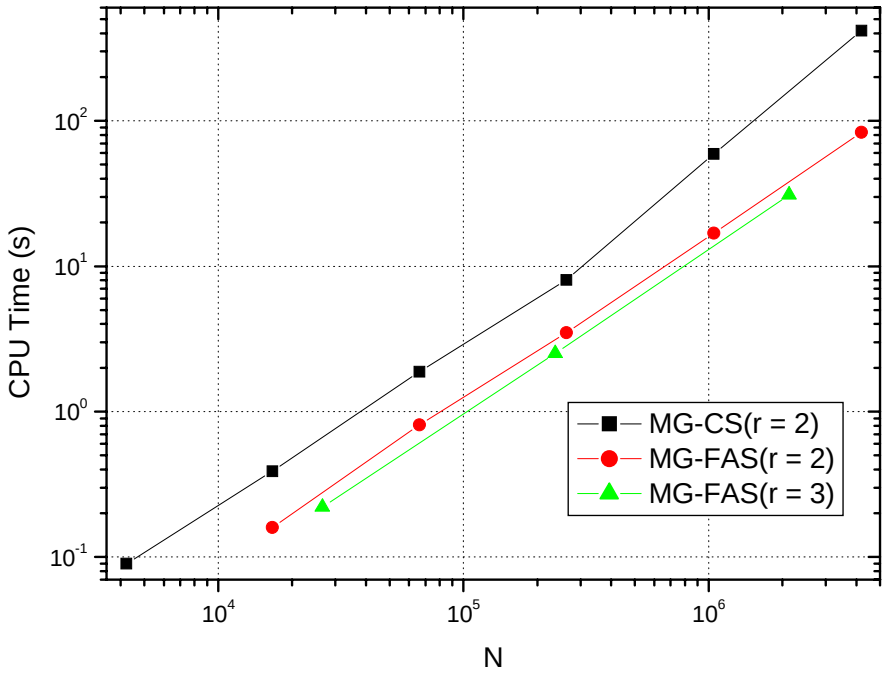


Figure 4. CPU time *versus N versus* scheme.

The results of the present work differ from those of Brandt (1977). This author made theoretical and experimental (numerical) analyses among the coarsening ratios r = 2, 3 and 3/2 for several problems, but he has not mentioned necessarily the Eq. (6). Brandt shows preference of the CS in relation to FAS scheme for linear problems. According to Brandt, each iterative cycle of FAS has a higher computational cost compared to CS scheme due to the calculations demanded in FAS scheme. But it was seen in this work that the number of iterative cycles of this scheme is significantly lesser than CS scheme. For example, for the grid with 2049x2049 nodes, the number of external iterations (*ITE*) of MG-FAS (r = 2) and MG-CS (r = 2) are 2 and 15, respectively. Another example: for the grid with 1459x1459 nodes the *ITE* of the MG-FAS (r = 3) and the MG-CS (r = 3) are 2 and 25, respectively.

4. CONCLUSION

In this work, the effect of several parameters on the necessary CPU time to solve a problem with the geometric multigrid method was verified. The considered parameters have been: coarsening ratios (r), number of nodes (N), number of inner iterations (*ITI*), number of grids (L) and correction (CS) and full approximation schemes (FAS). The considered mathematical model is a two-dimensional linear problem, governed by the Laplace's equation with Dirichlet boundary conditions. This equation was discretizated with the finite difference method and central difference scheme.

Based on the results of this work, it was verified that:

1) Among the tested values of r (2, 3, 4 and 5), for a given N and CS scheme, the minor CPU time is obtained with $r = 2$; and for FAS scheme, with $r = 3$.
2) The FAS scheme is faster than CS scheme for any N and r.
3) *ITI* can affect significantly the CPU time for CS and FAS schemes. The used scheme influences the optimum number of inner iterations ($ITI_{optimum}$): for CS scheme, $ITI_{optimum}$ = 1 or 2, depending on r and N; and for FAS scheme, $ITI_{optimum}$ = 3 to 6.
4) L can affect significantly the CPU time. $L_{optimum} \approx L_{maximum}$, that is, the CPU time with $L_{optimum}$ is practically the same that with $L_{maximum}$. This result is valid for one and two-dimensional, linear and nonlinear equations, CS and FAS schemes, coarsening ratios $r = 2$, 3, 4 and 5, any size of grid (N) and GS and MSI solvers.
5) The CPU time with FAS scheme and $r = 3$ is approximately 25% of the CPU time obtained with CS scheme and $r = 2$.
6) The results of the present work, for a two-dimensional problem, in general confirm the results already obtained by the authors themselves for one-dimensional problems. This shows the importance of studies with one-dimensional problems.

According to knowledge of the authors, the above verifications are unknown in available literature, except by the works published by authors of the present paper.

5. ACKNOWLEDGEMENTS

The first author is thankful to Laboratory of Numerical Experimentation (LENA), of the Department of Mechanical Engineering of the Federal University of Paraná, for yield its infrastructure, to the State University of Ponta Grossa and the CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for the financial support. The authors would acknowledge to CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) by material and financial support given for this work. The second author is scholarship of CNPq – Brazil.

6. REFERENCES

Brandt, A., "Multi-Level adaptive solutions to boundary-value problems, Mathematics of Computation", Vol. 31, pp. 333-390, 1977.

Brandt, A., "Barriers to Achieving Textbook Multigrid Efficiency (TME) in CFD", ICASE Interim Report, Vol. 32, NASA/CR, 1998.

Briggs, W. L. and Henson, V.E., Mccormick, S.F., "A Multgrid Tutorial", 2ª ed., SIAM, 2000.

Burden, R. L. and Faires, J. D., "Numerical Analysis", 6ª ed., Brooks/Cole Publishing Company, 1997.

Fedorenko, R. P., "On the Speed of Convergence of an Iteration Process", USSR Comput. Math. And Math. Phys., Vol. 4 (3), 1964.

Ferziger, J. H. and Peric, M., "Computational Methods for Fluid Dynamicas", 2ª ed., Springer, 1999.

Fortuna, A. O., "Técnicas Computacionais para Dinâmica dos Fluidos", Edusp, 2000.

Golub, G. H. and Van Loan, C., "Matrix Computations", 2ª ed., Johns Hopkins Press, Baltimore, 1989.

Hirsch, C., "Numerical Computational of Internal and External Flows", Vol.1, Wiley, 1988.

Larsson, J., Lien, F. S., Yee, E., "Conditional Semicoarsening Multigrid Algorithm for the Poisson Equation on Anisotropic Grids", Journal of the Computatational Physics, Vol. 208, pp. 368-383, 2005.

Maliska, C. R., "Transferência de calor e mecânica dos fluidos computacional", LTC, 2ª ed., 2004.

Montero, R. S., Llorense, I. M., Salas, M. D., "Robust Multigrid Algorithm for Navier-Stokes Equations", Journal of Computational Physics, Vol. 173, pp. 4412-432, 2001.

Nishida, H., Satofuka, N., "Higher-order Solutions of Square Driven Cavity Flow Using a Variable-order Multigrid Method", International Journal for Numerical Methods in Engineering, Vol. 34, pp. 637-653, 1992.

Oliveira, F., Pinto, M. A. V., Santiago, C. D., Marchi, C. H., "Efeito de Parâmetros do Método Multigrid CS e FAS sobre o Tempo de CPU em Problemas 1D Lineares e Não-lineares", Proceedings of the XXVII Iberian Latin American Congress on the Computational Methods in Engineering (CILAMCE), Belém, Brazil, 2006.

Pinto, M. A. V., Marchi, C. H., "Efeito de Parâmetros do Método Multigrid CS e FAS sobre o Tempo de CPU para a Equação de Laplace Bidimensional", Proceedings of the 11th Brazilian Congress of Thermal Sciences and Engineering (ENCIT), Curitiba, Brazil, 2006.

Pinto, M. A. V.; Santiago, C. D.; Marchi, C. H., "Effect of Parameters of a Multigrid Method on the CPU Time for One-dimensional Problems", Proceedings of the International Congress of Mechanical Engineering (COBEM), Paper 0619, Ouro Preto, Brazil, 2005.

Sathyamurthy, P. S., Patankar, S. V., "Block-Correction-Based Multigrid Method for Fluid Flow Problems", Numerical Heat Transfer, Part B, Vol. 25, pp. 375-394, 1994.

Schneider, G. E., Zedan, M., "A Modified Strongly Implicit Procedure for Numerical Solution of Field Problems", Numerical Heat Transfer, Vol. 4, pp. 1-19, 1981.

Souza, L. F., Oishi, C. M., Cuminato, J. A., Silveira Neto, A., "Comparison of Poisson Solvers in the Transient 2D Fluid Flow Problem with Dirichlet Boundary Condictions", Proceedings of the 11th Brazilian Congress of Thermal Sciences and Engineering (ENCIT), Curitiba, Brazil, 2006.

Stüben, K., "A Review of Algebraic Multgrid", Journal of Computation and Applied Mathematics, Vol. 128, pp. 281-309, 2001.

Tannehill, J. C., Anderson, D. A. and Pletcher, R. H., "Computational Fluid Mechanics and Heat Transfer", 2ª ed., Washington: Taylor & Francis, 1997.

Trottenberg, U., OOsterlee, C. and Schüller, A., "Multigrid", Academic Press, London, 2001.

Wesseling, P., "An Introduction to *Multgrid* Methods", John Wiley & Sons, 1992.

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

7. RESPONSIBILITY NOTICE

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