ANALYSIS OF PARALLEL GEOMETRIC MULTIGRID METHODS FOR POISSON EQUATION

Josuel Kruppa Rogenski, josuelkr@icmc.usp.br
Larissa Alves Petri, larris@gmail.com
Leandro Franco de Souza, lefraso@icmc.usp.br
Instituto de Ciências Matemáticas e de Computação - USP, Av. Trabalhador São-carlense, 400 - CEP: 13560-970 - São Carlos - SP.

Abstract. Direct Numerical Simulation (DNS) of incompressible flows demands a high computational cost. The formulation adopted frequently requires the numerical solution of Poisson equations. Normally, this solution is the most intensive subroutine in the numerical code. Aiming the optimization of the solution of the Poisson equation, this paper deals with multigrid methods and parallelization. With this motivation in mind, and knowing that in theory, multigrid methods solve linear systems in computational time proportional to the dimensional grid, two parallel geometric multigrid methods, Correction Scheme (CS) and Full Approximation Scheme (FAS) for solving 2D Poisson equation were implemented and a performance analysis was conducted. The speedup results obtained indicate that, for coarse meshes, the values are significantly lower due to overhead. When the number of unknowns increases the speedup improves for both methods.
Keywords: Poisson equation, geometric multigrid, parallel algorithms.

1. INTRODUCTION

In recent years, significant advances in Numerical Analysis have been achieved, in order to optimize the time necessary to obtain solutions of fluid dynamic problems. These advances can be motivated, for example, by using Direct Numerical Simulation (DNS) of incompressible flows. Moreover, depending on the formulation adopted, the numerical solution of a Poisson equation is necessary. This solution is the most intensive subroutine since it depends on solving a linear system. Aiming the optimization of the solution of elliptic partial differential equations (PDE), a multigrid method can be used.

Historically, according to Wesseling (1992), multigrid method was first developed by Fedorenko in the 1960's. Motivated by Fedorenko (1964), Brandt (1977) developed multigrid methods and studied the convergence rate, grid coarsening and local Fourier Analysis for linear and non-linear problems. Through Brandt's contribution, the multigrid method became widely used by the scientific community to solve problems in fluid dynamics. Hirsch (1988) and Tannehill et al. (1997) claim that the multigrid method is considered one of the most efficient method proposed recently.

According to Van der Velde (1994), the benefits of using multigrid methods are justified because it combines iterative solvers in meshes with different number of discretization points. Furthermore, the works of Hirsch (1988) and Ferziger and Peric (1999) show that there is an independence between the number of iterations to achieve convergence in the finest mesh and the number of mesh points. Pavlov et al. (2001) and Wesseling (1984) multigrid methods solve an elliptic problem discretized with N points using N operations. Stüben (2001) presents advantages of the algebraic multigrid for unstructured meshes. Wesseling (2001) recommends the use of geometric multigrid for structured meshes. The present work aims to investigate geometric multigrid methods. It were implemented two parallel geometric multigrid methods, Correction Scheme (CS) and Full Approximation Scheme (FAS) for solving 2D Poisson equation and a performance analysis was conducted.

The paper is structured as follows: section 2 presents a description of the multigrid methods adopted in this work - CS and FAS - and their parallelization. In addition, this section describes the numerical approximations used. Section 3 presents a performance analysis of the implemented methods. Finally, the main conclusions, acknowledgments and references are described in sections 4, 5 and 6 respectively.

2. FORMULATION AND NUMERICAL METHODS

The mathematical equation adopted in the present work is the Poisson equation:

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = s, \tag{1}$$

where v represents the solution and s is the source term.

For the methods under consideration, a V-cycle composed of four uniform Cartesian grid meshes was used. Figure 1 represents this multigrid structure, where S represents the iterative/smoother method, R is the restriction operation and P is the prolongation operation. The finest grid is represented by h and the coarsest one by $8h$.

2.1 Correction Scheme

The idea of the CS method is associated with the estimation of the error generated in the finer mesh as a way to obtain the solution of the linear system. This estimation is obtained by transmitting the residual of a finer mesh to a coarser one.

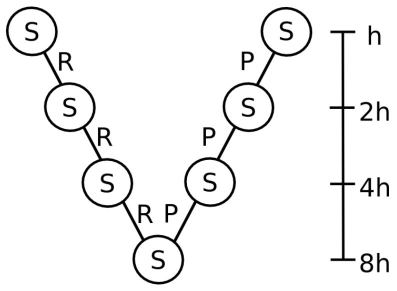


Figure 1. V-cycle - multigrid scheme

The adopted method is illustrated in Fig. 2.

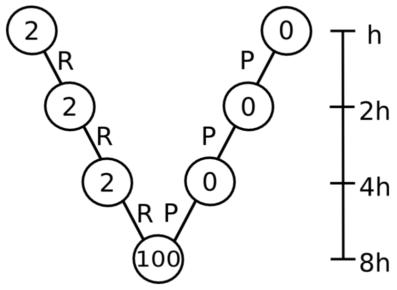


Figure 2. V-cycle - Correction Scheme

The CS multigrid method works as follow (Van der Velde, 1994):

1. Starting at the finest grid, two iterations of a Jacobi under-relaxed method with relaxation factor 0.9 is applied;
2. With the approximation of v the residual (d_h) can be achieved as:

$$d_h = s_h - \nabla^2 v_h. \tag{2}$$

3. If the residual is smaller than a prefixed tolerance, the algorithm is ended. Otherwise, the residual d_h is transmitted to a coarse grid $(2h)$ through an operation called restriction. This operation is illustrated in Fig. 3.

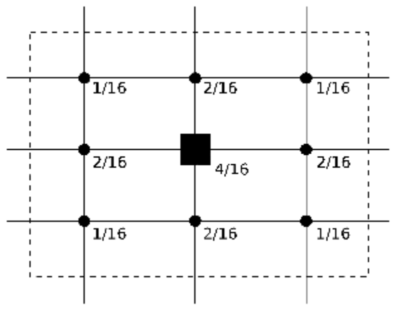


Figure 3. Full Weight - restriction operation

This specific operation is called Full Weight (FW). The only element to be transmitted, represented by a square, is defined as a source term in the coarse level $(2h)$ and it takes information of all neighbors using a specific weight for each one as shown in Fig. 3. The initial guess at $(2h)$ level is taken zero.

$$d_h \Rightarrow s_{2h} \quad (FW) \tag{3}$$

The procedures 1 - 3 are applied until it reaches the coarsest grid. At this level, 100 iterations are applied using a Successive Over Relaxation method (SOR) with relaxation factor 1.1.

4. The return to a more refined mesh is done by an operation called prolongation using bilinear interpolation and then correcting the solution.

$$v_{8h} \quad \Rightarrow \quad corr_{4h}, \tag{4}$$
$$v_{4h} \quad \Leftarrow \quad v_{4h} + corr_{4h}. \tag{5}$$

The bilinear interpolation is illustrated by Fig. 4. Each colored square represents a mesh point that exists only in the next refined grid. Moreover, the circles represent points existing in both levels. For blue and red squares, the information comes from the two inner circles and for the central green one, all four circles are used. The respective weights of the interpolation are shown above each arrow. This prolongation operation must be applied until the finest level. At this point, the cycle starts again.

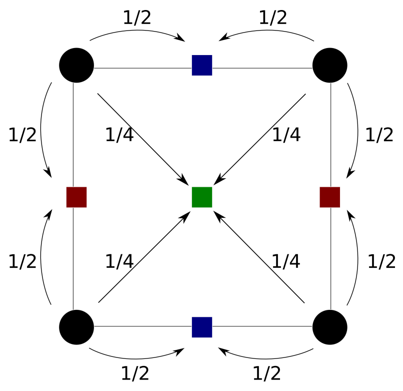


Figure 4. Bilinear interpolation - prolongation operation

2.2 Full Approximation Scheme

In the present case the V-cycle structure can be illustrated by Fig. 5. The FAS multigrid method, described in Souza (2003), works as follow:

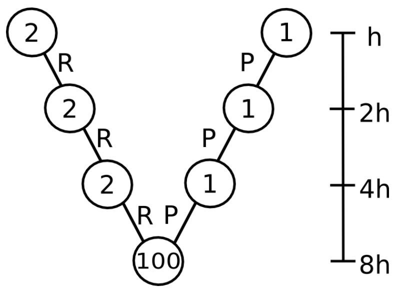


Figure 5. V-cycle - Full Approximation Scheme

1. Starting at the finest grid, two iterations of SOR method with relaxation factor 1.1 are applied.
2. With this approximation of v, the residual (d_h) is computed as:

$$d_h = s_h - \nabla^2 v_h. \tag{6}$$

3. If the residual is smaller than a tolerance, the algorithm is ended. Otherwise, the residual d_h is transmitted to a coarse grid $(2h)$ through FW operation. Moreover, the approximate solution v_h is transmitted directly, without any kind of weighting, using Straight Injection operation (SI).

$$d_h \quad \Rightarrow \quad s_{2h} \quad (FW) \tag{7}$$
$$v_h \quad \Rightarrow \quad v_{2h} \quad (SI) \tag{8}$$

The SI operation is illustrated in Fig. 6. Note that no information of the neighborhood is taken into account.

The procedures 1 - 3 are applied until it reaches the coarsest grid. At this level, 100 iterations of the SOR method with relaxation factor 1.1 are applied

4. The correction is calculated by:

$$corr_{8h} = v_{8h}^n - v_{8h}^{'}, \tag{9}$$

where $v_{8h}^{'}$ represents the solution generated by the restriction operation and v_{8h}^n is the newest solution.

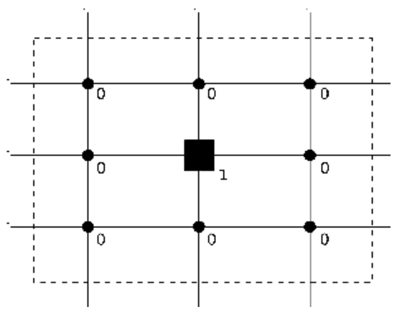


Figure 6. Straight Injection - restriction operation

5. To return to a more refined level, first it is adopted the same prolongation operation used in the CS scheme and then the solution is corrected.

$$corr_{8h} \Rightarrow corr_{4h}, \tag{10}$$
$$v_{4h} \Leftarrow v_{4h} + corr_{4h}. \tag{11}$$

6. Finally, one iteration of SOR method with factor 1.0 is applied. The V-cycle is finished after the iteration of SOR method at the finest grid.

2.3 Numerical approximations

The second-order derivatives in the Poisson equation were discretized using the following approximations (Lele, 1992; Souza, 2003). The i index means the grid position in the x direction, which ranges from 1 to $imax$. In the same sense, j means the grid position in the y direction and ranges from 1 to $jmax$.

In the x direction:

- for $2 < i < imax - 1$:

$$\frac{\partial^2 f}{\partial x^2}|_{i,j} = \frac{-f_{i-2,j} + 16f_{i-1,j} - 30f_{i,j} + 16f_{i+1,j} - f_{i+2,j}}{12\Delta x^2} + O(\Delta x^4). \tag{12}$$

- for $i = 2$:

$$\frac{\partial^2 f}{\partial x^2}|_{2,j} = \frac{10f_{1,j} - 15f_{2,j} - 4f_{3,j} + 14f_{4,j} - 6f_{5,j} + f_{6,j}}{12\Delta x^2} + O(\Delta x^4). \tag{13}$$

- for $i = imax - 1$: a similar approximation of $i = 2$ was adopted.

In the y direction:

- for $2 < j < jmax - 1$:

$$\frac{2}{15}\frac{\partial^2 f}{\partial y^2}|_{i,j-1} + \frac{11}{15}\frac{\partial^2 f}{\partial y^2}|_{i,j} + \frac{2}{15}\frac{\partial^2 f}{\partial y^2}|_{i,j+1} = \frac{3f_{i,j-2} + 48f_{i,j-1} - 102f_{i,j} + 48f_{i,j+1} + 3f_{i,j+2}}{60\Delta y^2} + O(\Delta y^6). \tag{14}$$

- for $j = 2$:

$$\frac{4}{5}\frac{\partial^2 f}{\partial y^2}|_{i,2} + \frac{1}{5}\frac{\partial^2 f}{\partial x^2}|_{i,3} = \frac{254f_{i,1} - 432f_{i,2} + 162f_{i,3} + 16f_{i,4}}{180\Delta y^2} + O(\Delta y^5). \tag{15}$$

- for $j = jmax - 1$: a similar approximation of $j = 2$ was adopted.

In the y direction, in problems with Neumann boundary conditions:

- for $j = jmax - 1$:

$$\frac{\partial^2 f}{\partial^2 y} = \frac{f_{i,jmax-2} - 2f_{i,jmax-1} + f_{i,jmax}}{\Delta y^2} + O(\Delta y^2). \tag{16}$$

- for $j = jmax$:

$$\frac{\partial^2 f}{\partial^2 y} = -\frac{f_{i,jmax-2} + 8f_{i,jmax-1} - 7f_{i,jmax}}{2\Delta y^2} + 3\frac{\partial f}{\partial y} + O(\Delta y^2). \tag{17}$$

There is no need to adopt higher-order approximations near to $j = jmax$, since the higher gradients are concentrated in the neighborhood of $j = 1$ in the fluid flow simulations that we focus in.

2.4 Parallelization of multigrid methods

The multigrid algorithms have been parallelized with Message Passing Interface (MPI) using a domain decomposition technique in the x direction. This choice is justified since the number of points in this direction is bigger than the number of points in the y direction, in the problems under consideration.

In this sense, considering a sequential rectangular mesh domain with $imax \times jmax$ points. Each process is responsible for Nx points in the x direction and $jmax$ points in the y direction. The value of Nx can be calculated by:

$$Nx = \frac{imax + (inter + 1)(p - 1)}{p}, \tag{18}$$

$$inter = 2^{(n-1)}(m - 1), \tag{19}$$

where p represents the number of processes, $inter$ is the superposition between adjacent subdomains, m is the length of the computational molecule adopted in the x direction and n is the number of a V-cycle's levels.

For example, if a multigrid method using a V-cycle composed of 4 levels takes the following second-order approximation

$$\frac{\partial^2 f}{\partial x^2} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2} + O(\Delta x^2), \tag{20}$$

then $m = 3$ and $n = 4$, which implies $inter = 8$ cells.

With regard to the algorithm's parallelization, the same sequential multigrid methods were applied at each subdomain. However, since for elliptic problems there is a high dependence between the variables involved, it was necessary to introduce communication points between adjacent processes. For the CS method, these points are:

- before applying a restriction operation;
- at each step of the iterative/smoother method;
- before applying the interpolation.

For the FAS method, the communication occurs:

- after applying a restriction operation;
- at each step of the iterative/smoother method;
- before applying the interpolation.

The squares in Fig. 7 represent the communication points for both methods.

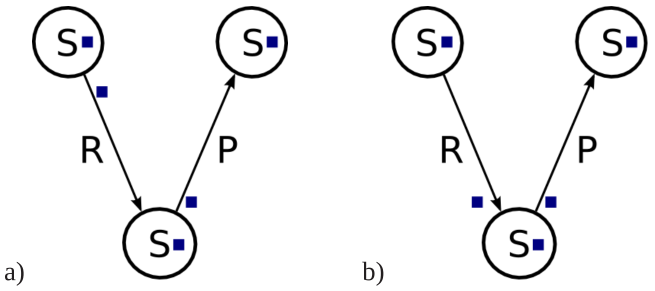


Figure 7. Communication points: a) CS scheme; b) FAS scheme.

The algorithm for the parallelization of the CS scheme was found in the literature, but it was not found any description of parallel FAS algorithm, therefore the proposed parallel algorithm presents a contribution in this area.

3. NUMERICAL RESULTS

Tests were carried out with the Poisson equation using two different boundary conditions:

- case 1: Dirichlet condition in all contours.
- case 2: Neumann condition in one boundary and Dirichlet conditions in the others.

The evaluation was done considering 36 different Cartesian meshes using 1, 2, 4 and 8 processors. The processors used are Intel Xeon E5345 with 2.33GHz, 16GB RAM and 8 processing core. In order to verify the algorithms it was done an analysis of execution time and speedup. Speedup can be defined by:

$$Su = \frac{T_s}{T_p}, \tag{21}$$

where T_s represents the execution time for the algorithm with the lowest number of processes - for this case, a sequential program - and T_p represents the execution time considering the parallel program with p processes. The tolerance adopted in this work was 10^{-5}. This criteria is associated with the absolute maximum residual of each subdomain.

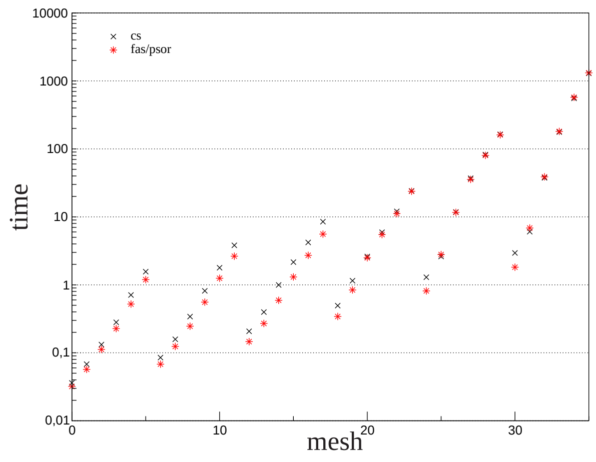
The results obtained for execution time are shown in Fig. 8, where in the x direction, each point corresponds to a mesh in Tab. 1.

Table 1. Points mesh size - 5 points molecule

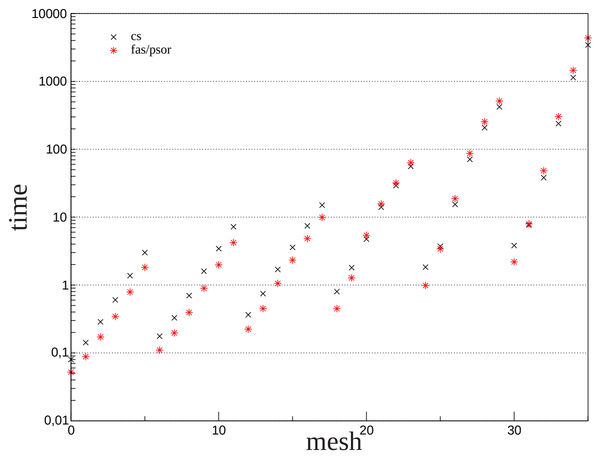
N	Nx	Ny
0	153	33
1	281	33
2	537	33
3	1049	33
4	2073	33
5	4121	33
6	153	65
7	281	65
8	537	65
9	1049	65
10	2073	65
11	4121	65

N	Nx	Ny
12	153	129
13	281	129
14	537	129
15	1049	129
16	2073	129
17	4121	129
18	153	257
19	281	257
20	537	257
21	1049	257
22	2073	257
23	4121	257

N	Nx	Ny
24	153	513
25	281	513
26	537	513
27	1049	513
28	2073	513
29	4121	513
30	153	1025
31	281	1025
32	537	1025
33	1049	1025
34	2073	1025
35	4121	1025



(a) Case 1



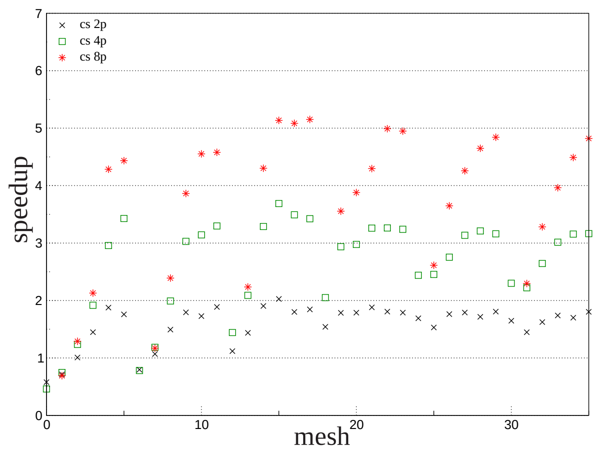
(b) Case 2

Figure 8. Execution time

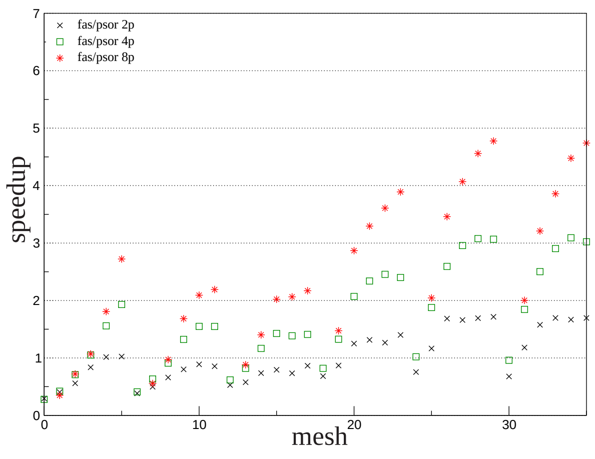
Analyzing the execution time, presented in Fig. 8, it is possible to see that when the number of variables increases, the CS method becomes better than FAS method, for both cases analyzed.

Furthermore, speedup for both cases with CS and FAS methods are shown in Figs. 9 and 10 respectively.

The speedup results obtained indicate that, for coarse meshes, the values are significantly lower due to overhead. In this sense, for meshes with small number of variables, sequential multigrid methods are better than parallel multigrid

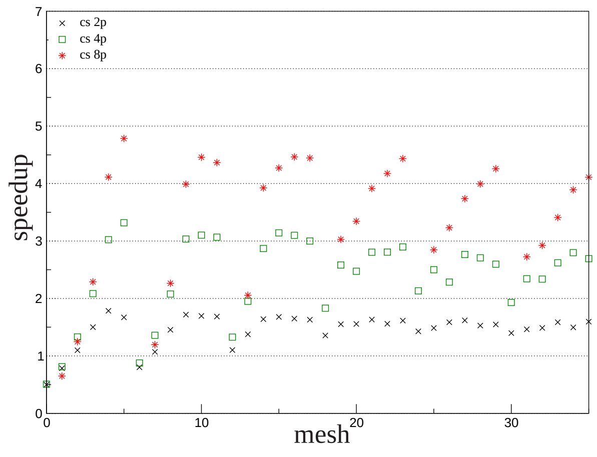


(a) CS

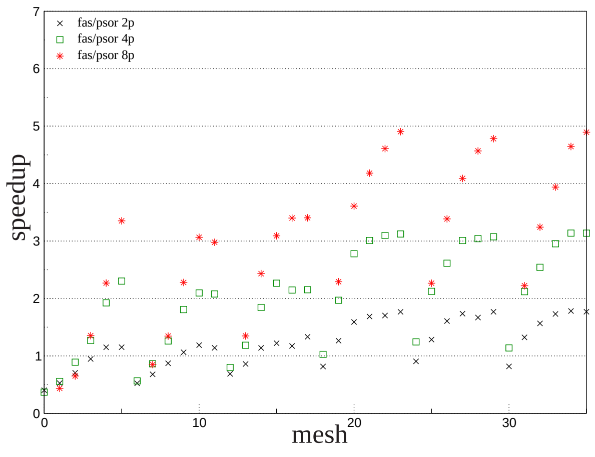


(b) FAS/PSOR

Figure 9. CS and FAS/PSOR speedup for case 1



(a) CS



(b) FAS/PSOR

Figure 10. CS and FAS/PSOR speedup for case 2

methods. For example, considering the application of FAS/PSOR method with two processes, illustrated in Fig. 9(b), the use of sequential methods is better for the meshes 0 to 3 and 6 to 19. When the number of unknowns raises the speedup improves for both multigrid methods. The growth of speedup has an upper bound which is different for each case.

Considering the CS method, in both cases, the speedup's growth behavior is similar for all values of y, considering the curve formed by fixing the number of points in y direction and ranging the number of points in x direction. This fact also can be seen in FAS/PSOR method until 257 points in y direction. After that, the speedup rises rapidly with the increase in the number of points in x direction.

For the first case, when the number of variables is small (for meshes from 0 to 25, for example), the CS method has a better speedup. While the number of unknowns increases, the speedup of FAS/PSOR method becomes similar to those of CS method. In addition, for the second case, in spite of being worse for small number of variables as in the first case, FAS/PSOR method achieves a better speedup than CS method, when the number of unknowns raises.

In general, considering just the first case, CS method showed the best results. Significant differences are observed for intermediate meshes. Nevertheless, for the second case, FAS method can be regarded as the best choice for finer meshes, as in cases 26-29 and 32-35.

4. CONCLUSIONS

Summarizing, for a large number of variables involved, the use of parallelism implies significant gains. For some cases this increase reaches more than five times over the sequential algorithm.

For the algorithms considered, the type of boundary condition used interferes in finding the best method. For the Dirichlet problem investigated, in general, CS method shows the best results. In the other hand, the use of a boundary condition of Neumann's type implies the use of FAS/PSOR multigrid method for finer meshes.

5. ACKNOWLEDGMENTS

The authors acknowledge the financial support received from FAPESP under grants 2009/03208-6 and 2008/00233-7.

6. REFERENCES

Brandt, A., 1977, "Multilevel adaptative solutions to boundary values problems", Mathematics of Computation, Vol.31, pp. 333–390.

Fedorenko, R. P., 1962, "A relaxation method for solving elliptic difference equations", USSR Computational Math. and Math. Phys, Vol. 4.

Ferziger, J. H.; PERIC, M., 1999, "Computational Methods for Fluids Dynamics", Springer, New York, USA, 389 p.

Hirsch, C., 1988, "Numerical computational of internal and external flows", John Wiley & sons, New York, USA.

Lele, S., 1992, "Compact Finite Difference Schemes with Spectral-like Resolution", J. Computational Physics, Vol. 103, pp. 16-42.

Pavlov, A.; et al., 2001, "A conservative finite difference method and its application for the analysis of a transient flow around a square prism", International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 10. pp. 6–46.

Souza, L.F., 2003, "Instabilidade Centrífuga e transição para turbulência em Escoamentos Laminares sobre Superfícies Côncavas", Instituto Tecnológico de Aeronáutica, São Paulo, Brasil.

Stüben, K., 2001, "A review of algebraic multigrid", Journal of Computation and Applied Mathematics, Vol. 128. pp. 281–309.

Tannehill, J. C.; Anderson, D. A.; Pletcher, R. H. , 1997, "Computational Fluid Mechanics and Heat Transfer", Taylor & Francis, Washington, USA, 816 p.

Van de Velde, E.F., 1994, "Concurrent Scientific Computing", Springer Verlag, California, USA, 328 p.

Wesseling, P., 1992, "An Introduction to Multigrid Methods", John Wiley & sons, England, 296 p.

7. Responsibility notice