PARALLELIZATION STRATEGIES FOR A HIGH ORDER INCOMPRESSIBLE FLOW CODE

Josuel Kruppa Rogenski, josuelkr@gmail.com
Larissa Alves Petri, lariss@gmail.com
Leandro Franco de Souza, lefraso@gmail.com
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo

Abstract. *The direct numerical simulation of transitional and turbulent incompressible flows is an area that is increasing with the advance in computational resources. The code parallelization has became a useful tool in these simulations. In the present work the unsteady two dimensional Navier-Stokes equations are used as physical model. The Tollmien-Schlichting waves propagating in a Poiseuille flow was adopted as test case. Three frequencies showing different behavior according to the Linear Stability Theory are used: an unstable, a neutral and a stable behavior. The parallelization is done via domain decomposition in the streamwise direction. The time derivative is integrated by a classical fourth order Runge-Kutta method. High order compact finite difference are used for the spatial derivatives discretization. The Poisson equation is solved by a multigrid method. The present work explores different parallelization techniques for solving the tri and penta-diagonal matrix, and the multigrid method. The results are compared to Linear Stability theory, and also between different strategies to show the advantages and disadvantages of each one.*

Keywords: *high-order compact finite difference, multigrid methods, Navier-Stokes equation, parallel computation.*

1. INTRODUCTION

The multicore processors evolution combined with the aim in computational time reduction, in computational fluid dynamics, are driving research related to the development and parallelization of numerical methods, as described by John and Tobiska (2000); Zhang (2002); Ge (2010); Henniger *et al.* (2010) and Buckeridge and Scheichl (2010). Specifically, in order to solve the Navier-Stokes equations, most of numerical formulations requires a linear system solution. The high cost of a numerical linear system solution motivates the use of efficient high-order methods and multigrid methods.

High-order approximations can be achieved through the use of compact finite difference schemes. Despite increasing the computational cost, the use of this strategy is justified by the low numerical dispersion and dissipation, stencil and error reduction and high resolution as can be seen in Lele (1992); Hirsh (1975); Wray and Hussaini (1994); Kloker (1998); Mahesh (1998) and Souza *et al.* (2005).

According to Zhang (1996, 1997); Spitaleri (2000) and Gupta *et al.* (1997), multigrid methods can be considered as a viable alternative for solving elliptic partial differential equations discretized with n points with a n complexity order. The efficiency of these methods, according to de Velde (1994), refers to combine iterative solvers in meshes with different number of discretization points. According to the author, it occurs because the fact that iterative methods reduce effectively - with a small number of iterations - only errors associated with high frequency.

This work aims the study and implementation of a parallel high resolution code for the solution of the 2D Navier-Stokes equations. The code is developed to investigate the convection of Tollmien-Schilichting waves in a Poiseuille flow. The governing equations are writing in a vorticity-velocity formulation, mainly related to the benefits of the colocatization and the elimination of the pressure treatment. The linear system arising from the numerical solution of Poisson equation is solved by multigrid methods. Two different methods are implemented, to evaluate the performance of each one. The spatial derivatives are discretized by compact finite difference schemes. The integration in time is carried out by a fourth order Runge-Kutta method. The algorithms are parallelized with Message Passing Interface (MPI) using a 1D domain decomposition technique in the main flow direction. It is conducted a comparison between different parallel strategies in order to obtain a good performance.

The paper is structured as follows: Section 2 discusses Navier-Stokes formulation topics. The numerical code characterization as the description of the multigrid methods, the compact finite difference schemes and the boundary vorticity treatment are discussed in Sec. 3. Section 4 deals with code parallelization. A performance analysis of the implemented method and comparisons with the linear stability theory are presented in Sec. 5. Finally, the main conclusions, acknowledgments and references are described in Sec. 6, 7 and 8, respectively.

2. FORMULATION

Isothermal and incompressible flows of a Newtonian fluid can be modeled through Navier-Stokes equations. The system of equations generated with continuity equation is expressed in two dimensions by:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \qquad (1)$$

$$
\begin{aligned}
\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} &= -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), &(2)\\
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0. &(3)
\end{aligned}
$$

The dimensionless parameters used are:

$$
\begin{aligned}
&x = \frac{x^*}{L^*},\ \ y = \frac{y^*}{L^*},\ \ u = \frac{u^*}{U^*_\infty},\\
&v = \frac{v^*}{U^*_\infty},\ \ p = \frac{p^*}{\rho^* U^*_\infty},\ \ t = \frac{t^* U^*_\infty}{L^*},
\end{aligned}
$$

with,

$$Re = \frac{U^*_\infty L^*}{\nu^*}, \qquad (4)$$

where the dimensional variables are indicated by $*$. The parameter L^* is a reference length, U^*_∞ is the free-stream velocity and Re is the Reynolds number. Also, x^* and y^* represent coordinates in the longitudinal and normal direction, respectively. Aiming to eliminate the pressure treatment of the Navier-Stokes equations, the vorticity-velocity formulation is adopted. Therefore, it is defined the vorticity ω_z, such as:

$$\omega_z = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}. \qquad (5)$$

Using the vorticity definition and the continuity equation, it can be derived a v Poisson equation as:

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = -\frac{\partial \omega_z}{\partial x}. \qquad (6)$$

Convenient transformations in the Navier-Stokes equations result in the vorticity transport equation:

$$\frac{\partial \omega_z}{\partial t} = -\frac{\partial (u\omega_z)}{\partial x} - \frac{\partial (v\omega_z)}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2}\right). \qquad (7)$$

Thus, by the use of this formulation, the system to be solved consists of the Eqs. (3), (6) and (7). The boundary conditions are specified according to the problem under consideration.

3. NUMERICAL METHOD

The Navier-Stokes equations using vorticity-velocity formulation are solved numerically with the objective of investigating the flow behavior under a infinitesimal disturbance. In this sense, it is considered a rectangular area represented by the Fig. 1.

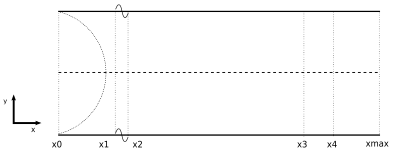


Figure 1. Integration domain

3.1 Disturbance Introduction

According to Fasel *et al.* (1990), the introduction of disturbances in the flow can be done through suction and blowing of mass at the walls. At time $t = 0$ the flow is undisturbed. From a time $t > 0$, the disturbances are introduced in a region near inflow, known as disturbance strip, through the imposition of v velocity:

$$
\begin{aligned}
v &= Af(x)\text{sin}(x), \quad x_1 < x < x_2 &(8)\\
&\text{e}\\
v &= 0, \quad x \leq x_1 \ \text{ or } \ x \geq x_2, &(9)
\end{aligned}
$$

where A is a constant used to adjust the amplitude of the disturbances, $f(x)$ is a 9^{th} order function and the points x_1 and x_2 are the initial and the last point of the disturbance strip. The values of $f(x)$, its first and second derivatives are zero in these extreme points.

3.2 Damping and relaminarization zones

In order to avoid reflections, it is used a damping regions near the inflow and near the outflow boundaries. Considering the damping region lying between the discrete points $1 \le i \le i_1$, it is defined a function:

$$f_1(x) = f_1(\epsilon_1) = 6\epsilon_1^5 - 15\epsilon_1^4 + 10\epsilon_1^4, \tag{10}$$

where $\epsilon_1 = \frac{i-1}{i_1-1}$, $1 \le i \le i_1$.

At the relaminarization flow region, located between $i_3 \le i \le i_4$, it is defined another function

$$f_2(x) = f_2(\epsilon_2) = e^{-\frac{\epsilon^4}{10}}(1-\epsilon^{50})^4, \tag{11}$$

where $\epsilon_2 = \frac{i-i_3}{i_4-i_3}$, $i_3 \le i \le i_4$. In both cases, the functions in question alter the values of the vorticity after the integration in time.

3.3 Multigrid method

Two geometric multigrid methods are tested: a Correction Scheme (CS) and a Full Approximation Scheme (FAS). For the cases under consideration, the linear system is solved in a V-cycle composed by 4 levels and represented by Fig. 2, 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 $8h$.

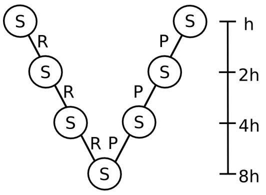


Figure 2. Multigrid method - V-cycle

3.3.1 Correction Scheme - multigrid method

Adopting a Poisson equation

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = s, \tag{12}$$

where v represents the function and s the source term of the system. The adopted V-cycle structure is illustrated in Fig. 3. The constants N_1 and N_2 represent the number of iterations applied at a respective level. A line Jacobi method is applied at all levels, except at the coarsest, where a Line Successive over-Relaxation (LSOR) method is used.

For the Jacobi method it is used a relaxation factor equal r_1. In the case of the LSOR method, it is used a relaxation factor equal r_2. It is important to note that there is no application of iterative methods in the right branch of the V-cycle, as explained in Fig. 3. Taking as reference the presented informations, the CS algorithm can be described as follows:

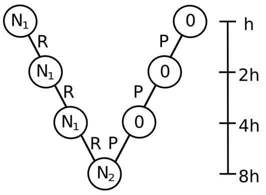


Figure 3. Multigrid CS - V-cycle

1. Starting at the finest grid, it applies N_1 iterations of a Jacobi under-relaxed method with factor r_1.
2. With this approximation, computes the residue (d_h) as

$$d_h = s_h - \nabla^2 v_h. \tag{13}$$

3. If the residue is smaller than a tolerance, the algorithm is ended. Otherwise, the residue d_h is transmitted to a coarse grid $(2h)$ through an operation called restriction. This operation is illustrated in Fig. 4.

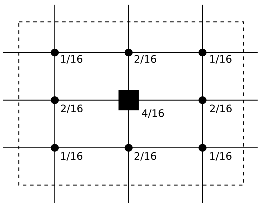


Figure 4. Full Weight - restriction operation

This specific ponderation 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 $(2h)$. The initial guess at $(2h)$ level is taken equal zero.

$$d_h \Rightarrow s_{2h} \quad (FW) \tag{14}$$

The procedures 1 - 3 are applied until it reaches the coarsest grid. At this level, N_2 LSOR iterations are applied with factor r_2.

4. The return to a more refined mesh is by applying a prolongation operation and then correcting the solution.

$$v_{8h} \quad \Rightarrow \quad corr_{4h}, \tag{15}$$

$$v_{4h} \quad \Leftarrow \quad v_{4h} + corr_{4h}. \tag{16}$$

This operations, illustrated by Fig. 5 must be applied until the finest level. At this point, the cycle starts again.

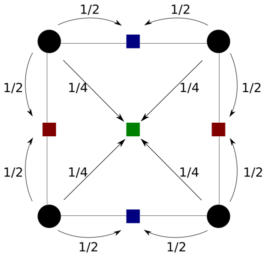


Figure 5. Bilinear interpolation - prolongation operation

3.3.2 Full Approximation Scheme - multigrid method

Considering the Eq. (12) and the V-cycle represented by Fig. 6.

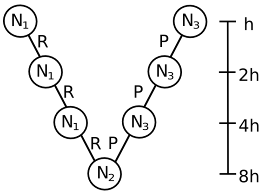


Figure 6. Multigrid FAS - V-cycle

Using the FAS, the solution of the Poisson equation is performed by a LSOR method in all levels. Specifically, it is applied a Gauss-Seidel method after the prolongation operation. In other cases, the relaxation factor is given by r_2. The multigrid FAS can be described as follows:

1. Starting at the finest grid, N_1 iterations of the LSOR method are applied, with factor r_2.
2. With this approximation, the residue (d_h) is calculated by:

$$d_h = s_h - \nabla^2 v_h. \tag{17}$$

3. If the residue is smaller than a tolerance, the algorithm is ended. Otherwise, the residue d_h is transmitted to a coarse grid $(2h)$ through a FW operation. The approximate solution v_h is restricted by a Straight Injection, ie. without any kind of ponderation, as can be seen in Fig. 7.

$$d_h \quad \Rightarrow \quad s_{2h} \quad (FW) \tag{18}$$

$$v_h \quad \Rightarrow \quad v_{2h} \quad (SI) \tag{19}$$

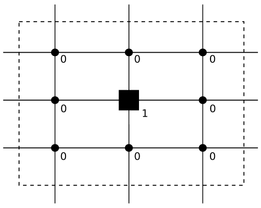


Figure 7. Straight Injection - restriction operation

The procedures 1 - 3 are applied until they reach the coarsest grid. At this level, N_2 LSOR iterations are applied with r_2 factor.

4. The correction at this levels is calculated by:

$$corr_{8h} = v^n_{8h} - v^{'}_{8h}, \tag{20}$$

where $v^{'}_{8h}$ represents the solution generated by the restriction operation and v^n_{8h} is the newest solution.

5. The prolongation to a more refined mesh is done by applying the same bilinear interpolation adopted in the multigrid CS, and then correcting the solution.

$$corr_{8h} \quad \Rightarrow \quad corr_{4h}, \tag{21}$$

$$v_{4h} \quad \Leftarrow \quad v_{4h} + corr_{4h}. \tag{22}$$

6. Finally, one iteration of Gauss-Seidel method is applied. The V-cycle is finished after the application of LSOR method at the finest grid.

3.3.3 Numerical approximations

The spatial derivatives in the Poisson equation are discretized using compact high-order finite differences approximations (Souza, 2003).

3.4 Compact high-order approximations for spatial derivatives

The use of compact finite differences to estimate the first and second spatial derivatives requires the resolution of tridiagonal linear systems. The systems in question can be found in the work of Souza *et al.* (2005).

The time integration is carried out by a classical fourth order Runge-Kutta method.

3.5 Spatial filter

In order to eliminate numerical (spurious) oscillations, a computational filter is applied after the last Runge-Kutta step (Lele, 1992). The filter adopted requires the solution of a pentadiagonal system. The filter is applied in the vorticity component.

3.6 Numerical method

In agreement with the studies of Souza (2003) and considering the vorticity-velocity formulation, Eqs. (6), (7) and (3) are solved numerically by the application of the following steps: (a) apply a step of the time integrator; (b) apply the functions responsible for the damping and relaminarization zones; (c) introduce disturbances by suction and blowing at the walls; (d) calculate the right hand side of Eq. (6); (e) calculate the v velocity by solving the linear system generated by Eq. (6); (f) calculate the value of u velocity by Eq. (3); (g) update the vorticity value ω_z on the walls; (h) apply, after the last sub-step of the time integrator, the computational filter. The numerical simulation ends when it reaches the desired computational time.

4. PARALLELIZATION

This section describes the parallel strategies adopted for solving the multigrid methods (Poisson Equation), the high-order compact finite difference approximations and the computational filter.

4.1 Parallel multigrid methods

Multigrid codes have been parallelized using a 1D domain decomposition technique in the main flow direction. This choice is justified by the fact that the number of points in this direction is much bigger than the number of points in the normal flow direction, in the problems under investigation. The MPI library has been used for parallel implementation.

It is considered a rectangular domain of $imax \times jmax$ points. Each processing element (e.g., core, processor) $k = 1, ..., p$ is responsible for Nx points in the x direction and $jmax$ points in the y direction. The Nx value is calculated by

$$Nx = \frac{imax + (inter + 1)(p - 1)}{p}, \tag{23}$$

$$inter = 2^{(N-1)}(m - 1), \tag{24}$$

where p is the number of processing elements, $inter$ is the intersection between adjacent domains, m is the computation molecule size in the x direction and N is the number of V-cycle levels.

In elliptical problems the solution of any point in the domain can be affected by a change in any other point of the domain, therefore, it is necessary to introduce communication points between adjacent processing elements. For this reason, for the parallel CS, the communication points are: before applying a restriction operation; at each step of the iterative/smoother method and before applying the interpolation operation. For the parallel FAS, the communications occur: after applying a restriction operation; at each step of the iterative/smoother method and between applying the interpolation and correction operation.

4.2 Parallel high-order compact finite difference approximations

It is considered a discrete sequential domain containing $imax \times jmax$ points in the stream and normal flow directions, respectively. Since the parallelization was done in the x direction, this subsection will focus in the numerical solution of the first and second derivatives in this direction. It is observed that the use of the classical Thomas algorithm for this problem requires the solution of $jmax$ linear systems.

Two strategies are developed to parallelize the numerical compact finite difference code. The first one uses the pipeline concept, since each line of the domain represents the solution of a tridiagonal system. In this strategy a process k is responsible to calculate just a part of a linear system associated with each discrete line of the domain. It is emphasized that just one system is solved per domain line. The second strategy considers the processing elements independent of each other. Unlike the previous strategy, p sub-systems are solved per domain line. At the end, $inter/2$ columns are changed between adjacent processing elements, as illustrated by Fig. 8.

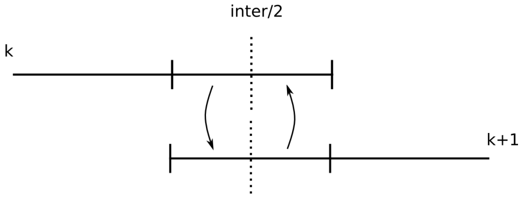


Figure 8. Second parallel strategy - communication between adjacent processing elements

4.3 Parallel computational filter

Regarding the resolution of a pentadiagonal system cited in subsection 3.5, a parallel strategy based in the pipeline concept is adopted, therefore using the first strategy described by the previous subsection.

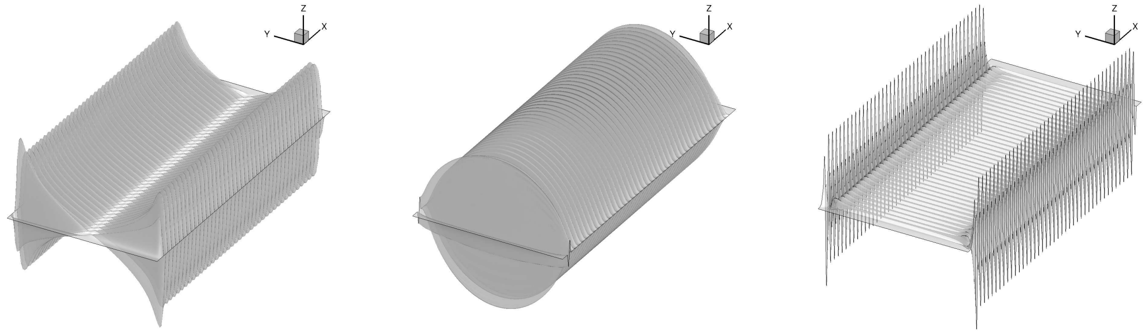
5. RESULTS

In this section, a comparison of a 2D Navier-Stokes equation solver and the linear stability theory and an analysis of the code parallelization are presented.

5.1 Poiseuille flow and linear stability theory

In order to verify the parallel numerical code a comparison of the numerical results and the linear stability theory is carried out. It is considered a rectangular domain, as presented by Fig. 1. The main flow under consideration is gave by the function $u(y) = -y^2 + 2y$. It is adopted $A = 0.0005$ to adjust the amplitude of the Tollmien-Schlichting waves in Eq. 9. The adopted number of time steps per wave period is 128. It is used 32 points per Tollmien-Schlichting wavelength. It is also defined $imax = 1433$ and $jmax = 81$ points in the x and y directions, respectively. The distance between consecutive points in the x and y directions are $\Delta y = \frac{2}{jmax-1}$ and $\Delta x = \frac{2\pi}{32\alpha_r}$, where α_r is the real part of the wavenumber. The simulation time is equal to 42 times the step per period under consideration. The other constants used are $x_0 = 0,\ x_1 = 39\Delta x,\ x_2 = 71\Delta x,\ x_3 = (imax - 100)\Delta x,$ and $x_4 = (imax - 40)\Delta x$.

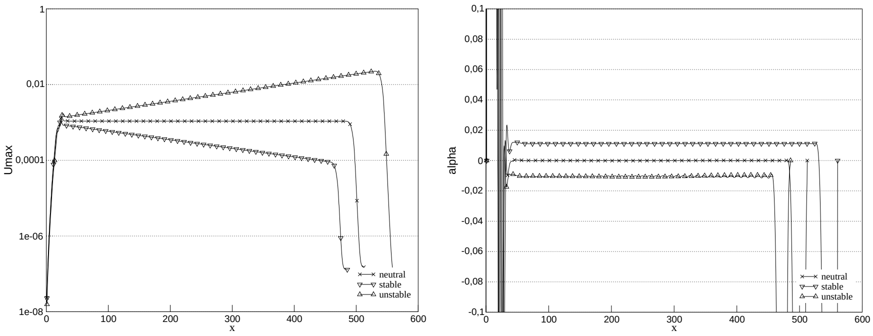
The FAS multigrid maximum defect is settled to 10^{-6}. The LSOR relaxation factor adopted is $r_2 = 1.1$. The number of iterations at each V-cycle level is $N_1 = 2,\ N_2 = 40$ and $N_3 = 1$. The second parallel strategy is used to solve the spatial derivatives in the x direction. The tests consider a neutral, stable and unstable cases. Figures 9(a), 9(b) and 9(c) present qualitative data results for the neutral case simulation. The simulations were done using 8 processing elements.



(a) velocity u (b) velocity v (c) vorticity

Figure 9. Poiseuille flow - neutral case - qualitative data

In addition, Figs. 10(a) and 10(b) show, respectively, the maximum speed and amplification rate associated with the first Fourier mode of the u velocity, for each point in the x direction. The neutral, unstable and stable cases are presented in agreement with Tab. 1.



(a) maximum velocity (b) amplification rate

Figure 10. Poiseuille flow - Fourier analysis - first mode behavior

Figure 10(b) show that the numerical results are consistent with data generated by the linear stability theory. The small oscillations are result, fundamentally, by the mesh considered. Moreover, the oscillations near the extremes are justified due to the introduction of disturbances and flow relaminarization.

In Fig. 10(a) it can be observed that the amplitude of the maximum disturbance velocity u_{max} grows, remain stable or decreases according to the unstable, neutral and stable cases, respectively. Additionally, the amplification rates obtained for the three cases are presented in Tab. 1. The values obtained are in agreement with the reference values.

5.2 Poiseuille flow - multigrid CS and FAS comparison

Only the neutral case is considered to verify the performance of the multigrid schemes in the parallel code. The same rectangular domain represented by Fig. 1 and the same parameters adopted in the previous section are used. Tests are carried out considering a set of nine meshes, described by the Tab. 2.

The results also consider as a final simulation time: 19, 30 and 48 times the step per period associated with the meshes

Table 1. Poiseuille flow - linear stability theory and code data

	Re	$\alpha_i(LST)$	$\alpha_i(numerical)$
A	5000	0.0100000	0.0104254
B	10000	0.0000988	0.0000513
C	10000	-0.0100000	-0.0111146

Table 2. Poiseuille flow - nominal axis

N	1	2	3	4	5	6	7	8	9
imax	665	665	665	1049	1049	1049	1433	1433	1433
jmax	49	65	81	49	65	81	49	65	81

considering 665, 1049 and 1433 in the x direction, respectively. The multigrid CS uses $r_1 = 0.75$ and $r_2 = 1.1$ as relaxation factors related to Jacobi and LSOR, respectively.

Figure 11 illustrates the execution time associated with the simulation of the 9 cases presented in the Tab. 2. This figure shows the sequential code behavior of the CS and FAS multigrid methods.

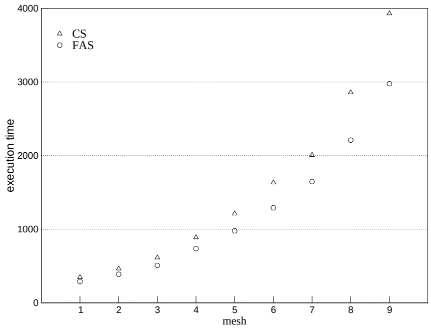
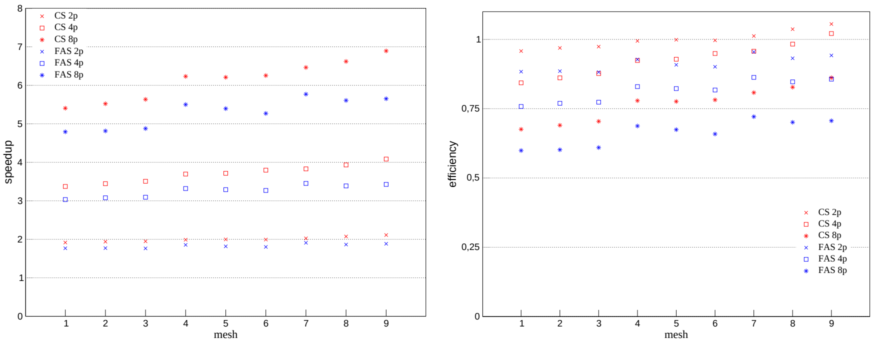


Figure 11. Poiseuille flow - sequential multigrid comparative

Figure 11 shows that the FAS is faster than the CS for all cases tested. The execution time for the both methods increases with the number of points in the y direction. It is noteworthy that there is no sense in carrying out comparisons between cases with different number of points in the x direction, since the total time simulation is different. Moreover, to evaluate the behavior of parallel multigrid methods using the first parallel strategy for compact finite difference derivatives, the speedup and efficiency are presented by Figs. 12(a) and 12(b).



(a) speedup

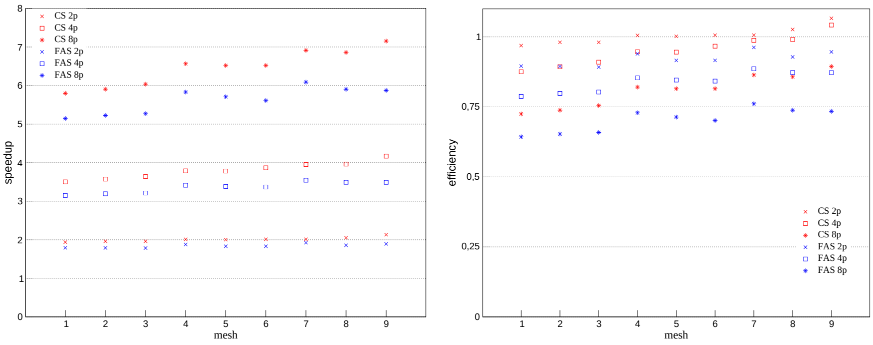
(b) efficiency

Figure 12. Poiseuille flow - multigrid parallel comparative using the first compact finite parallel strategy

It is observed a certain uniformity of the speedup values for both multigrid schemes analyzed. The rates, especially with the use of 2 and 4 processing elements, shows very positive results. The best speedup results, for all test cases, are obtained through the CS parallelization. Efficiency rates close to or above the optimum are obtained with the use of 2 and 4 processing elements. The efficiency obtained by the FAS was not as good as the CS, and this can be related to the communication time spent by the application of a Gauss-Seidel iteration in the multigrid FAS.

Through analysis aimed at the Figs. 11 and 12(a), it is possible to see that the parallel CS even though more efficient than the parallel FAS, demands a greater computational time. However, these results suggest a tendency to use the parallel CS method, in cases with a very large numbers of points.

Regarding the second parallel strategy for the solution of the compact approximations, the speedup and efficiency measures are presented by Figs. 13(a) and 13(b).



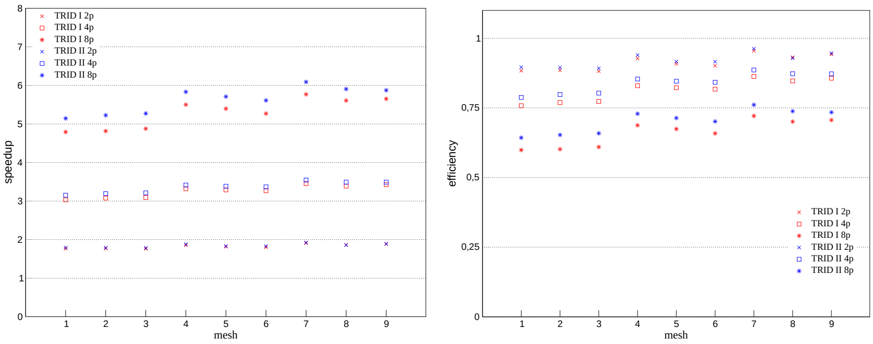
(a) speedup (b) efficiency

Figure 13. Poiseuille flow - multigrid parallel comparative using the second compact finite parallel strategy

The results in question, considering the second strategy of parallelization, presenting similar results to the previous case.

5.3 Poiseuille flow - comparative of parallel compact finite difference strategies

Comparisons between parallel strategies focused on compact high-order differences approximations are described as follows. The speedup and efficiency, considering only the FAS multigrid method are shown by Figs. 14(a) and 14(b).



(a) speedup (b) efficiency

Figure 14. Poiseuille flow - comparative of parallel compact finite difference strategies using multigrid FAS method

According to Figs. 14(a) and 14(b), the second parallel strategy is superior to the first, for all the tests. The increase in the number of processing elements implies an increase in the difference between speedup and efficiency rates of parallel strategies in question.

6. CONCLUSIONS

In the present study it is conducted a parallel performance analysis directed to the numerical solution of partial differential equations aimed to investigate a two dimensional incompressible and isothermal flow of a Newtonian fluid. A 4^{th} order Runge–Kutta method is adopted for time integration of a vorticity–velocity formulation. The Poisson equation is solved by multigrid methods and the spatial derivatives are discretized by high-order compact finite difference schemes. The algorithms are parallelized with Message Passing Interface (MPI) using a 1D domain decomposition technique in the main flow direction.

Regarding the solution of the Poisson equation the results show that the FAS multigrid method shows the best indices of execution time for all the set of meshes tested. The increase of the communication points in the left part of the V cycle affects the speedup and efficiency of the FAS method. It is possible that the CS method is more competitive for problems with a large number of points. In respect the parallelization of the spatial derivatives calculation, the best use of computing resources and the reduction of communication points obtained through the second parallel strategy confirm the feasibility in using this strategy.

Comparatives associated with the linear stability theory show agreement with the numerical method against theoretical results. As a result, for all cases testes, there are significant gains in the use of parallel strategies. In some cases, gains of more than seven times may be observed with the use of 8 processing elements.

7. ACKNOWLEDGEMENTS

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

8. REFERENCES

Buckeridge, S. and Scheichl, R., 2010. "Parallel geometric multigrid for global weather prediction". *Numerical Linear Algebra with Applications*, Vol. 17, pp. 325–342.

de Velde, E.F.V., 1994. *Concurrent Scientific Computing*. Springer–Verlag.

Fasel, H.F., Rist, U. and Konzelmann, U., 1990. "Numerical investigation of the three-dimensional development in boundary-layer transition". *AIAA*, Vol. 28, pp. 29–37.

Ferziger, J.H. and Peric, M., 1997. *Computational Methods for Fluid Dynamics*. Springer-Verlag Berlin Heidelberg New York.

Ge, Y., 2010. "Multigrid method and fourth-order compact difference discretization scheme with unequal meshsizes for 3D Poisson equation". *Journal of Computational Physics*, Vol. 229, pp. 6381–6391.

Gupta, M.M., Kouatchou, J. and Zhang, J., 1997. "Comparison of second- and fourth-order discretizations for multigrid Poisson solvers". *J. Computational Physics*, Vol. 132, pp. 226–232.

Henniger, R., Obrist, D. and Kleiser, L., 2010. "High-order accurate solution of the incompressible Navier-Stokes equations on massively parallel computers". *Journal of Computational Physics*, Vol. 229, pp. 3543–3572.

Hirsh, R.S., 1975. "High order accurate difference solutions of fluid mechanics problem by a compact differencing technique". *Journal of Computational Physics*, Vol. 19, pp. 90–109.

John, V. and Tobiska, L., 2000. "Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier-Stokes equations". *International Journal for Numerical Methods in Fluids*, Vol. 33, pp. 453–473.

Kloker, M., 1998. "A robust high-tesolution split-type compact FD scheme for spatial Direct Numerical Simulation of boundary-layer transition". *Applied Scientific Research*, Vol. 59, pp. 353–377.

Lele, S., 1992. "Compact finite difference schemes with spectral-like resolution". *J. Computational Physics*, Vol. 103, pp. 16–42.

Mahesh, K., 1998. "A family of high order finite difference schemes with good spectral resolution". *J. Computational Physics*, Vol. 145, pp. 332–358.

Souza, L.F., 2003. *Instabilidade Centrífuga e Transição para Turbulência em Escoamentos Laminares sobre Superfícies Côncavas*. Ph.D. thesis, Instituto Tecnológico de Aeronáutica.

Souza, L.F., Mendonça, M.T. and Medeiros, M.A.F., 2005. "The advantages of using high-order finite differences schemes in laminar-turbulent transition studies". *International Journal for Numerical Methods in Fluids*, Vol. 48, pp. 565–592.

Spitaleri, R.M., 2000. "Full-FAS multigrid grid generation algorithms". *Applied Numerical Math.*, Vol. 32, pp. 483–494.

Wray, A. and Hussaini, M.Y., 1994. "Highly accurate compact methods and boundary conditions". *Proc. Royal Soc. London*, Vol. A 392, pp. 373–389.

Zhang, J., 1996. "Acceleration of five-point red-black Gauss-Seidel in multigrid for Poisson equation". *Applied Math. and Comp.*, Vol. 80, pp. 73–93.

Zhang, J., 1997. "Residual scaling techniques in multigrid, I: equivalence proof". *Applied Math. and Comp.*, Vol. 80, pp. 283–303.

Zhang, J., 2002. "Multigrid method and fourth-order compact scheme for 2d Poisson equation with unequal mesh-size discretization". *Journal of Computational Physics*, Vol. 179, pp. 170–179.

9. Responsibility notice

The author(s) is (are) the only responsible for the printed material included in this paper.