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-2018-0827
PERFORMANCE ANALYSIS OF A PARALLEL CODE FOR LAMINAR-TURBULENT TRANSITION STUDIES

Matheus Tozo de Araujo
Leandro Franco de Souza
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo
Av. Trabalhador São Carlense, 400
13566-590, São Carlos - SP
matheustozo@gmail.com, lefraso@icmc.usp.br

Abstract. *In the present paper, the two-dimensional Navier-Stokes equations with the Giesekus constitutive equation are used to simulate a viscoelastic fluid flow in a straight channel. As test cases Tollmien-Schlichting waves are introduced near the upstream boundary and propagated in the flow field. The simulation is done using a high-order compact finite difference scheme for the spatial derivatives discretization and with a parallelization method that makes a domain decomposition in both streamwise and normal flow direction. The present work investigates the efficiency of the parallelization method in the simulations that were made with different meshes, using different number of processing elements in both directions of the domain.*

Keywords: *Parallelization, domain decomposition, Giesekus fluid, laminar-turbulent transition*

1. INTRODUCTION

With the advancement of science and technology, the numerical simulation has become an important ally in the development of many kinds of experiments that are often impossible to do in practice, either by their magnitude or only by the costs or time involved (Quinn, 2003). In this way, computational techniques to optimize processing time and increasingly efficient algorithms become increasingly necessary in order to eliminate dependence on technological advancement. Even because, the increase in individual processing power can be costly or even technically unfeasible.

One solution for this problem is the use of parallelization or distribution of processing data. Parallelising processing data means dividing the computational effort between several processors and making it happen approximately simultaneously. At the end, the data from the various processes must be aggregated to obtain the final result. Some of these methods can be seen in (Buckeridge and Scheichl, 2010; Ge, 2010; Henniger *et al.*, 2010; Zhang, 2002; John and Tobiska, 2000).

The present work aims the study and implementation of a parallel code of high resolution for the solution of the Navier-Stokes equations. The code, originally developed to investigate the convection of Tollmien-Schlichting waves in a straight channel flow, is used to analyze the performance and efficiency of parallelization in the simulations. The governing equations are written in a velocity-vorticity formulation, with the viscoelastic flow governed by the constitutive equation of Giesekus. The linear system resulting from the numerical solution of the Poisson equation is solved by multigrid methods. Spatial derivatives are discretized by compact finite difference schemes. Time integration is performed by a fourth-order Runge-Kutta method. The algorithms are parallelized using the message passing interface (MPI) library with a domain decomposition technique in the main and normal direction of the flow. For a better analysis, different meshes were used.

2. MATHEMATICAL FORMULATION

Incompressible and isothermal flows of non-Newtonian fluids can be modeled by the continuity and Navier-Stokes equations, that is given respectively by

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) \right) = -\nabla p + \nabla^2 \mathbf{u} + \nabla \cdot \mathbf{T}, \tag{2}$$

where $\mathbf{u}$ denotes the velocity field, t is the time, ρ is the fluid density, p is the pressure and $\mathbf{T}$ is the extra-stress tensor which must obey an appropriate constitutive equation. In this paper, we worked with viscoelastic flows governed by the

non-linear Giesekus constitutive equation (Giesekus, 1982), given by:

$$\mathbf{T} + \lambda \left(\overset{\nabla}{\mathbf{T}} + \frac{\alpha_G}{\eta_p}(\mathbf{T}\cdot\mathbf{T}) \right) = 2\eta_p \mathbf{D}, \tag{3}$$

where $\mathbf{D} = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ is the rate of deformation tensor, λ is the relaxation-time of the fluid, α_G is the so-called mobility parameter, η_p is the polymer-contributed viscosity and $\overset{\nabla}{\mathbf{T}}$ is the upper-convected derivative of $\mathbf{T}$, defined by:

$$\overset{\nabla}{\mathbf{T}} = \frac{\partial \mathbf{T}}{\partial t} + \nabla\cdot(\mathbf{u}\mathbf{T}) - \mathbf{T}\cdot(\nabla\mathbf{u})^T - (\nabla\mathbf{u})\cdot\mathbf{T}. \tag{4}$$

Introducing the following non-dimensionalized scalings:

$$\mathbf{x}^* = \frac{x}{L}, \quad \mathbf{u}^* = \frac{\mathbf{u}}{U}, \quad t^* = \frac{U}{L}t, \quad p^* = \frac{p}{\rho U^2}, \quad \mathbf{T}^* = \frac{\mathbf{T}}{\rho U^2}, \tag{5}$$

where L and U denote length and velocity scales, respectively. These equations can then be written (omitting the symbol $*$ for convenience)

$$\nabla\cdot\mathbf{u} = 0, \tag{6}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\mathbf{u}) = -\nabla p + \frac{\beta}{Re}\nabla^2\mathbf{u} + \nabla\cdot\mathbf{T}, \tag{7}$$

$$\mathbf{T} + Wi\,\overset{\nabla}{\mathbf{T}} + \frac{\alpha_G\, Wi\, Re}{(1-\beta)}(\mathbf{T}\cdot\mathbf{T}) = 2\frac{(1-\beta)}{Re}\mathbf{D}, \tag{8}$$

where the dimensionless parameters $Re = (\rho U L)/(\eta)$ and $Wi = (\lambda U)/(L)$ denote the associated Reynolds and Weissenberg numbers, respectively. The amount of Newtonian solvent is controlled by the dimensionless solvent viscosity coefficient, $\beta = \eta_s/\eta_0$, where $\eta_0 = \eta_s + \eta_p$ denotes the total shear viscosity; η_s and η_p represent the Newtonian solvent and polymeric viscosities, respectively.

2.1 Direct Numerical Simulation

To simulate all scales of the flow, from the largest and most energetic to the smallest, it is employed a high order finite difference schemes, without adding closing equations.

In order to eliminate the pressure term in the Navier-Stokes equations, it is adopted the vorticity-velocity formulation. Thus, the vorticity in direction z, ω_z, is defined by:

$$\omega_z = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}. \tag{9}$$

Therefore, the equations (6)-(8) in the two-dimensional and non-dimensional form can be rewritten as:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{10}$$

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = -\frac{\partial \omega_z}{\partial x}, \tag{11}$$

$$\frac{\partial \omega_z}{\partial t} + \frac{\partial (u\omega_z)}{\partial x} + \frac{\partial (v\omega_z)}{\partial y} = \frac{\beta}{Re}\left(\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right) + \frac{\partial^2 T^{xx}}{\partial x \partial y} + \frac{\partial^2 T^{xy}}{\partial y^2} - \frac{\partial^2 T^{xy}}{\partial x^2} - \frac{\partial^2 T^{yy}}{\partial x \partial y}, \tag{12}$$

$$T^{xx} + Wi\Big(\frac{\partial T^{xx}}{\partial t} + \frac{\partial (uT^{xx})}{\partial x} + \frac{\partial (vT^{xx})}{\partial y} - 2T^{xx}\frac{\partial u}{\partial x} - 2T^{xy}\frac{\partial u}{\partial y} \Big) + \alpha_G \frac{WiRe}{(1-\beta)}(T^{xx^2} + T^{xy^2}) = \\ = 2\frac{(1-\beta)}{Re}\frac{\partial u}{\partial x}, \tag{13}$$

$$T^{xy} + Wi\Big(\frac{\partial T^{xy}}{\partial t} + \frac{\partial (uT^{xy})}{\partial x} + \frac{\partial (vT^{xy})}{\partial y} - T^{xx}\frac{\partial v}{\partial x} - T^{yy}\frac{\partial u}{\partial y} \Big) + \alpha_G \frac{WiRe}{(1-\beta)}(T^{xx}T^{xy} + T^{xy}T^{yy}) = \\ = \frac{(1-\beta)}{Re}\Big(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \Big), \tag{14}$$

$$T^{yy} + Wi\Big(\frac{\partial T^{yy}}{\partial t} + \frac{\partial (uT^{yy})}{\partial x} + \frac{\partial (vT^{yy})}{\partial y} - 2T^{xy}\frac{\partial v}{\partial x} - 2T^{yy}\frac{\partial v}{\partial y}\Big) + \alpha_G \frac{WiRe}{(1-\beta)}(T^{xy^2} + T^{yy^2}) = \\ = 2\frac{(1-\beta)}{Re}\frac{\partial v}{\partial y}. \tag{15}$$

Considering a straight channel flow, the following boundaries conditions of the computational domain, Fig. 1, were used: $u = U(y)$, $v = 0$, $T^{xx} = T^{xx}(y)$, $T^{xy} = T^{xy}(y)$, and $T^{yy} = T^{yy}(y)$, at the inflow; no-slip condition and impermeability $(u = 0, v = 0)$, on wall boundaries; and the second derivative of variables in respect to x are set to zero, at the outflow.

2.2 Base flow

For the Giesekus fluid the base flow was generated numerically by two-dimensional code, without disturbances, and the simulations performed until the flow reached the steady state, close to the end of a very long domain in the streamwise direction.

3. NUMERICAL METHOD

The system of equations (10)–(15) is solved numerically in the domain as shown in Fig. 1. The calculations are done on an orthogonal uniform grid, parallel to the wall. The fluid enters the computational domain at $x = x_0$ and exits at the outflow boundary $x = x_{max}$.

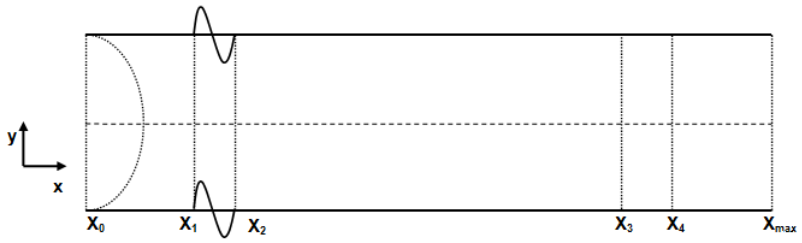


Figure 1. Computational domain.

The flow variables for time $t = 0$, are undisturbed, therefore from time $t > 0$, the disturbances are introduced in a region near the inflow (in both walls), known as disturbance strip, through the imposition of the v velocity, given by:

$$v = A\, f(x)\, sin(\omega_t\, t), \quad x_1 < x < x_2, \quad \text{and} \quad v = 0, \quad x \le x_1 \quad \text{or} \quad x \ge x_2, \tag{16}$$

where A is a constant used to adjust the amplitude of the disturbances, $f(x)$ is a 9th-order function and ω_t is the disturbance temporal frequency. The points x_1 and x_2 are the initial and the last point of the disturbance strip, respectively. The values of $f(x)$, its first, and second derivatives are zero in these extreme points. To avoid wave reflections from the inflow and outflow boundaries, it was implemented a buffer domain technique, from (Kloker *et al.*, 1993), in the region located between x_0 and x_1 and x_3 and x_4, respectively. The simulation finishes when a periodic flow is reached and the disturbances reach the region close to the end of the domain in the streamwise position.

3.1 Parallel strategies

The code was parallelized using a domain decomposition technique in both directions (the main flow direction, x-direction, and the normal flow direction, y-direction). The MPI library (MPICH2) has been used for parallelization.

Considering the rectangular domain represented in Fig. 1 with $i_{max} \times j_{max}$ points in the x- and y- directions, respectively, each processing element (e.g., core, processor) $k = 1, \cdots, p$ is responsible for N_x points in the x-direction and N_y points in the y- direction, where N_x and N_y are calculated, respectively, by

$$N_x = \frac{i_{max} + (npx - 1) \times 25}{npx} \quad \text{and} \quad N_y = \frac{j_{max} + (npy - 1) \times 25}{npy}, \tag{17}$$

where npx and npy are the number of processing elements in x- and y-direction, respectively.

To ensure the correct computation of the solution, the introduction of communication points between two adjacent processing element is required. In the parallel FAS method, the communications occur: after the restriction operation; at each step of the iterative/smoother method; and between the interpolation and the correction operations.

4. NUMERICAL RESULTS

Different numerical simulations were performed by varying the number of processing in the x- and y-directions, in order to verify the simulation time and its efficiency. In all the simulations were considered fixed constants $Re = 6500$, $\alpha_G = 0$, $Wi = 1.0$ and $\beta = 0.9$.

To evaluate the behavior of the parallel multigrid method the speedup and efficiency parameters are compared. The speedup (S_p) metric is defined as:

$$S_p = \frac{ET_1}{ET_p}, \tag{18}$$

where ET_1 and ET_p are the wall-clock time using 1 and p processing elements, respectively. Moreover, the efficiency (ET_p) is given by:

$$E_p = \frac{S_p}{p}. \tag{19}$$

The simulations were carried out in 6 different meshes, where the numbers of points adopted were $imax = 985$ and 1945 at x-direction and $jmax = 217$, 313 and 409 at y-direction. Moreover, the others parameters adopted were: the distance between two consecutive points in the x- and and y-direction were $dx = 2\pi/(32\alpha_r)$ and $dx = 2/(jmax - 1)$, respectively, where α_r is the real part of the wavenumber; the time steps per wave period are 128, disturbance frequency $\omega_t = 217$. The parameter A to adjust the amplitude of the Tollmien-Schlichting waves was 1×10^{-4}.

Figures 2 - 7 illustrates the a) wall-clock time, b) speedup and c) efficiency of the parallelization process using simulations which data were described previously. The different symbols in the figures represent the number of processing elements in the y-direction. The number of processing elements adopted varied from 1 to 40. The wall-clock time results were obtained by an average of 5 runs. The simulations were carried out in a Intel Xeon E5-2680v2 - 2.8 GHz with 2 processor with 10 cores each (plus 10 Hyper-threading).

The difference of the results shown in Figs. 2 and 5 is the number of points in the x-direction. The comparison of these figures show that the speedup and efficiency shows similar behavior, with a slightly better results when the number of points increases. The same behavior can be observed comparing Figs. 3 and 6 and Figs. 4 and 7.

The results presented show that the wall-clock time can decrease while increasing the number of processing elements in y-direction, however the a good efficiency is not achieved. This behavior was observed for all simulations carried out in the present study. It can also be observed that the efficiency increase in the y-direction with the number of points adopted in this direction.

5. CONCLUSIONS

The present paper shows a performance analysis of a parallelized code using MPI library, that investigates the Tollmien-Schlichting waves convection in a two-dimensional straight channel flow for non-Newtonian viscoelastic fluid where it was considered the Giesekus constitutive equation for the numerical simulations and with the governing equations written in a vorticity-velocity formulation. In order to evaluate the performance of the parallelization method, different values of processing elements were tested for both x- and y-direction in different meshes.

The results show that a significant reduction in the wall-clock time can be obtained through parallelization method, principally for cases where we have a great number of points in the domain. The parallelization method showed to be less efficient in the cases where did not have a great number of points in a determined direction, y-direction for example, but showed to be an excellent tool for highly-demanding applications.

6. ACKNOWLEDGEMENTS

The authors acknowledge CAPES. Research carried out using the computational resource of the Center for Mathematical Applied to Industry (CeMEAI) funded by FAPESP (Process Number 2013/07375-0).

7. REFERENCES

Buckeridge, S. and Scheichl, R., 2010. "Parallel geometric multigrid for global weather prediction". *Numerical Linear Algebra with Applications*, Vol. 17, No. 2-3, pp. 325–342.

Ge, Y., 2010. "Multigrid method and fourth-order compact difference discretization scheme with unequal mesh-sizes for 3D Poisson equation". *J. Comput. Phys.*, Vol. 229, No. 18, pp. 6381–6391. ISSN 0021-9991. doi: 10.1016/j.jcp.2010.04.048. URL `http://dx.doi.org/10.1016/j.jcp.2010.04.048`.

Giesekus, H., 1982. "A simple constitutive equation for polymer fluids based on the concept of deformation-dependent tensorial mobility". *Journal of Non-Newtonian Fluid Mechanics*, Vol. 11, pp. 69–109.

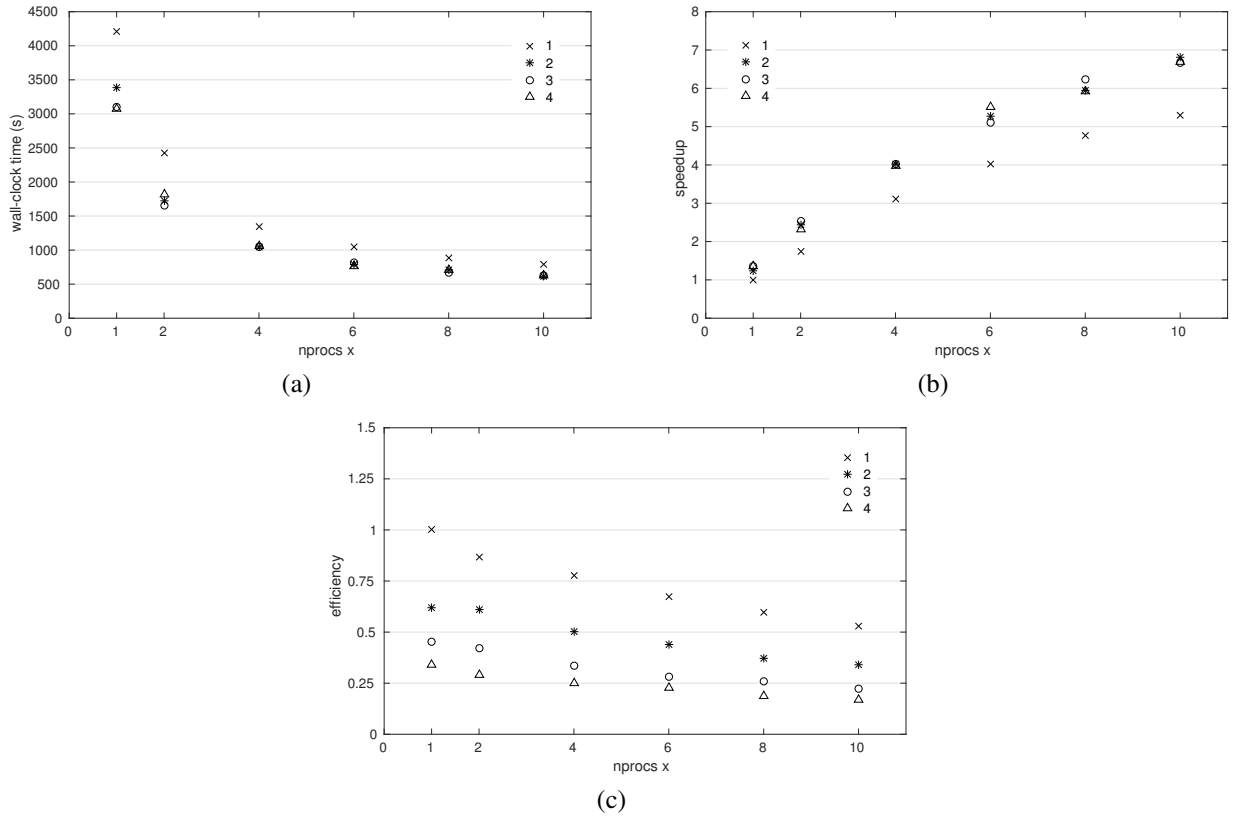


Figure 2. Comparative plots of the number of process in x-direction vs. a) wall-clock time, b) speedup and c) efficiency for four different quantities of process in y-direction in a mesh with 985 and 217 points in the x- and y-direction, respectively.

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, No. 10, pp. 3543 – 3572. ISSN 0021-9991.

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, No. 4, pp. 453–473.

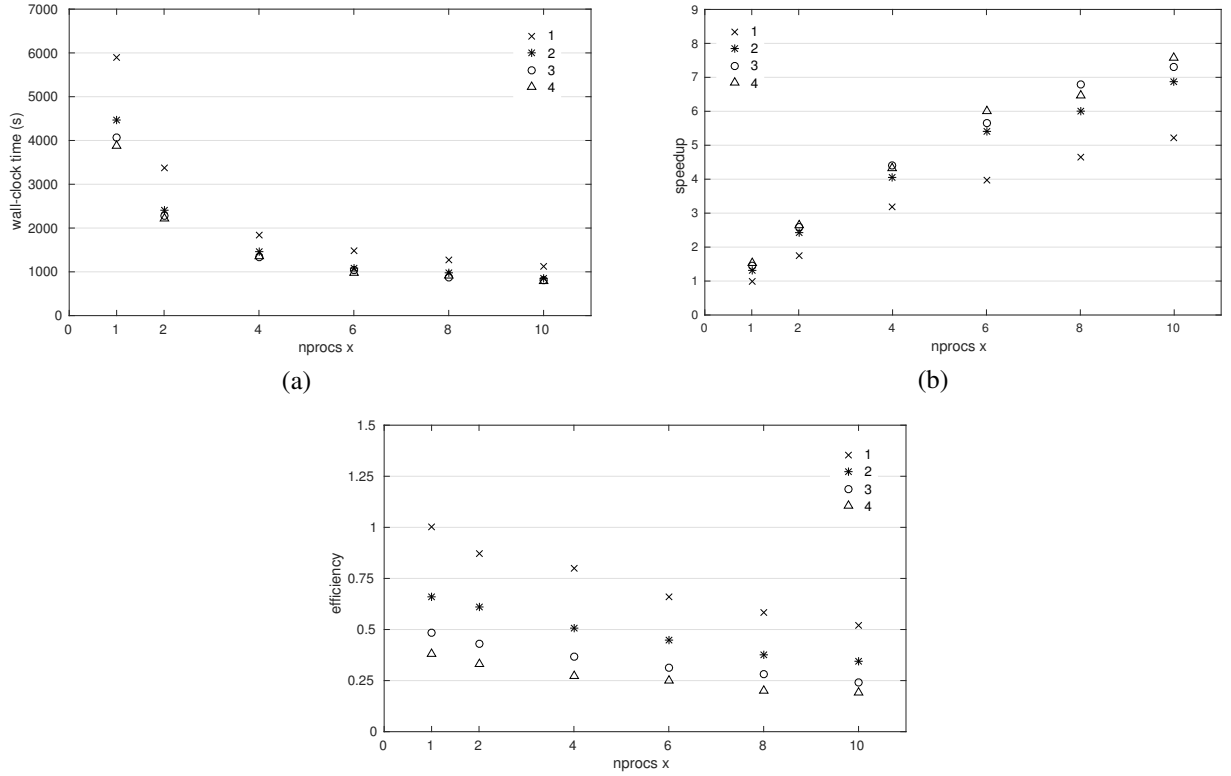
Kloker, M., Konzelmann, U. and Fasel, H.F., 1993. "Outflow boundary conditions for spatial Navier-Stokes simulations of transition boundary layers". *AIAA Journal*, Vol. **31**, pp. 620–628.

Quinn, M.J., 2003. *Parallel Programming in C with MPI and OpenMP (1a edição)*. McGraw Hill.

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, No. 1, pp. 170 – 179. ISSN 0021-9991.

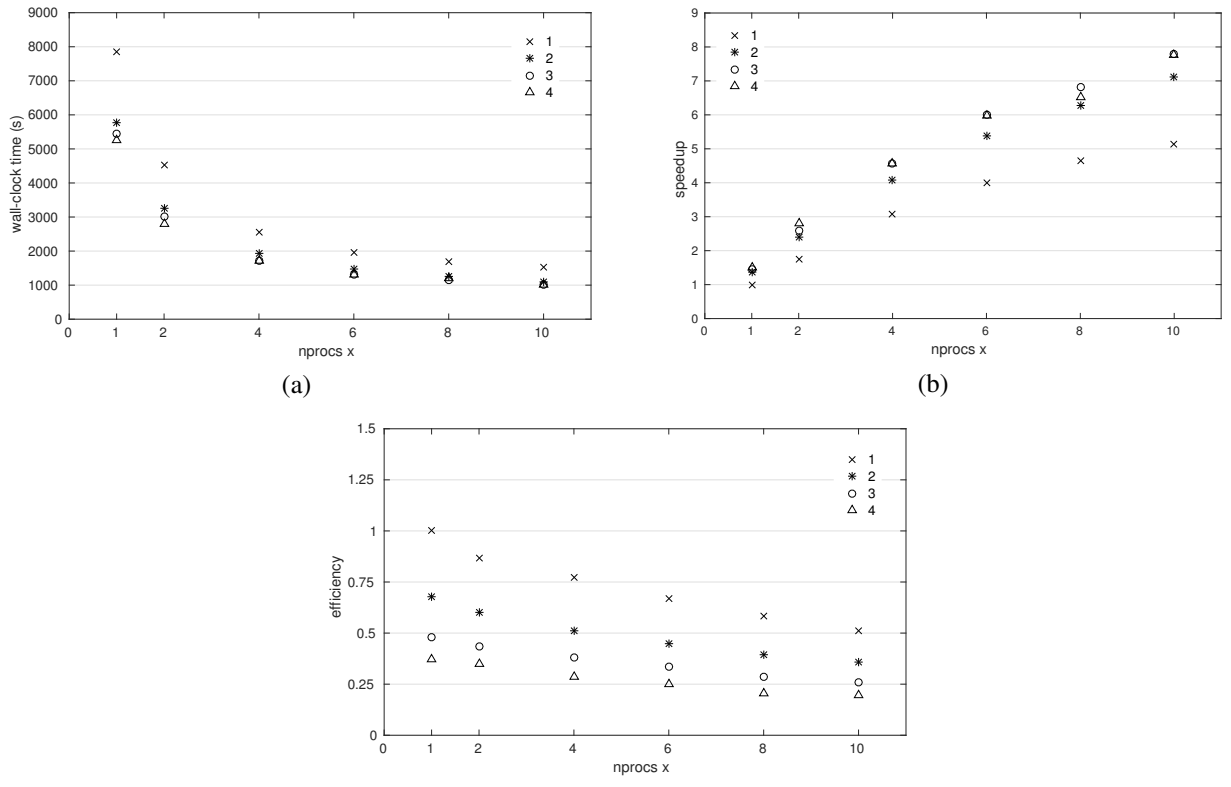
8. RESPONSIBILITY NOTICE

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



(a) (b) (c)

Figure 3. Comparative plots of the number of process in x-direction vs. a) wall-clock time, b) speedup and c) efficiency for four different quantities of process in y-direction in a mesh with 985 and 313 points in the x- and y-direction, respectively.



(a) (b) (c)

Figure 4. Comparative plots of the number of process in x-direction vs. a) wall-clock time, b) speedup and c) efficiency for four different quantities of process in y-direction in a mesh with 985 and 409 points in the x- and y-direction, respectively.

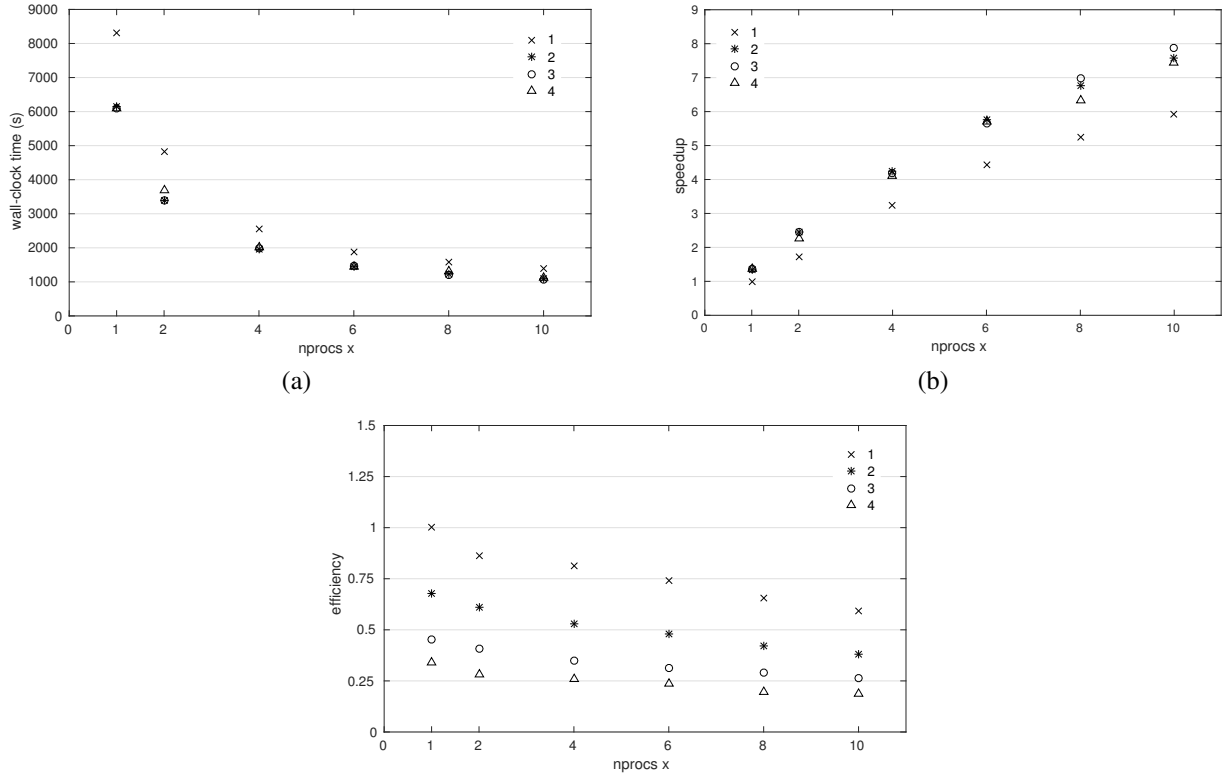


Figure 5. Comparative plots of the number of process in x-direction vs. a) wall-clock time, b) speedup and c) efficiency for four different quantities of process in y-direction in a mesh with 1945 and 217 points in the x- and y-direction, respectively.

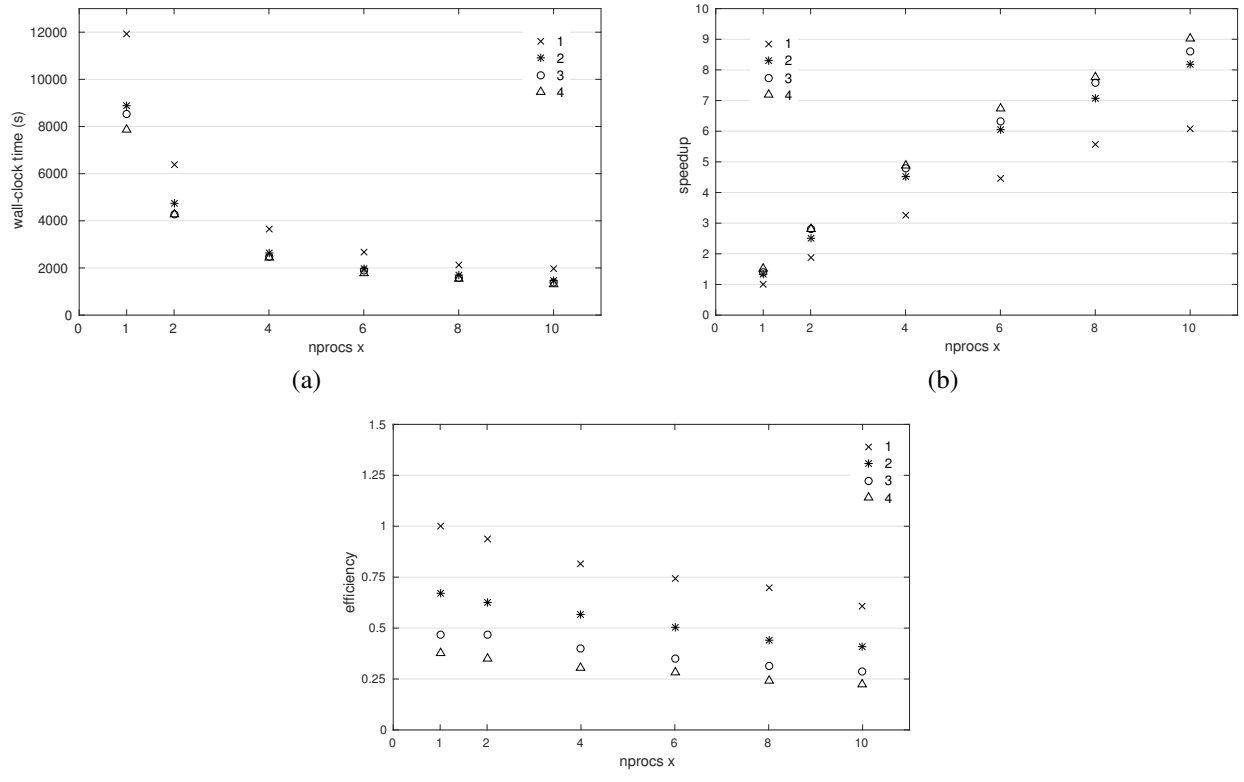


Figure 6. Comparative plots of the number of process in x-direction vs. a) wall-clock time, b) speedup and c) efficiency for four different quantities of process in y-direction in a mesh with 1945 and 313 points in the x- and y-direction, respectively.

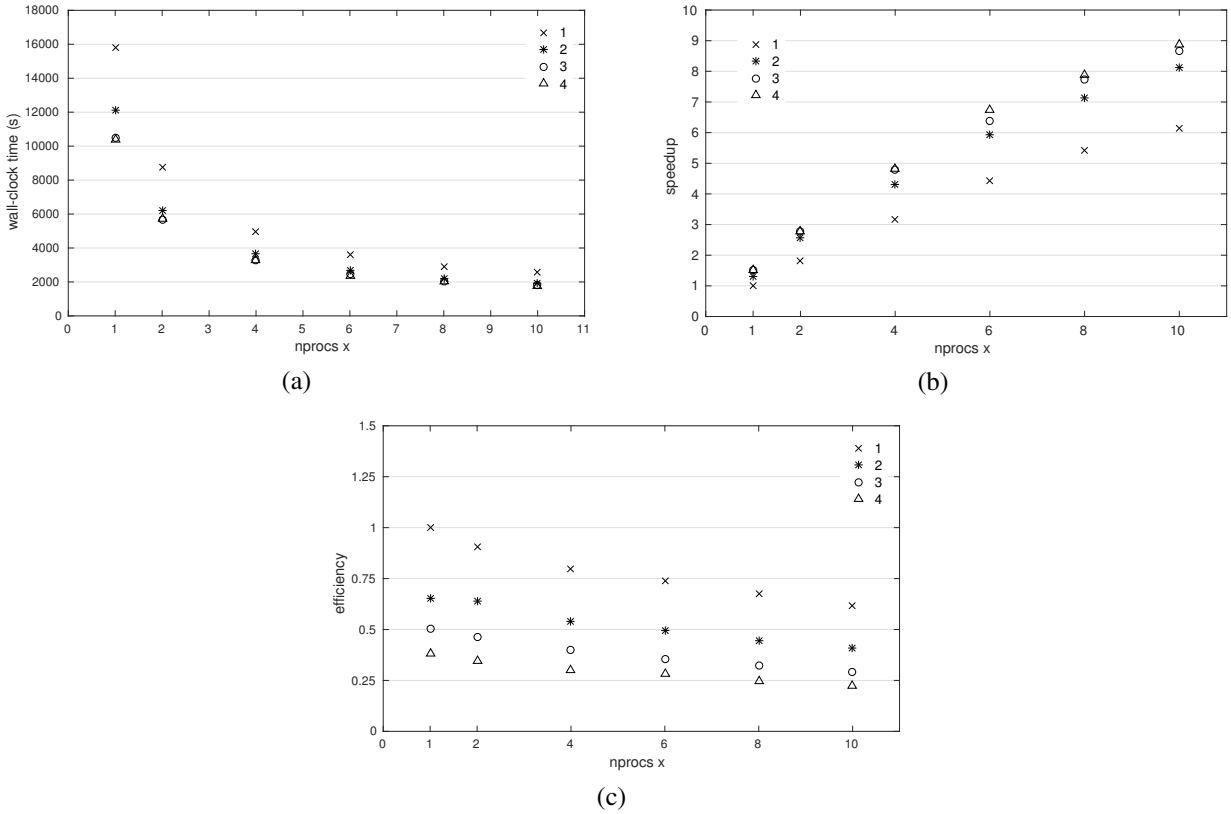


Figure 7. Comparative plots of the number of process in x-direction vs. a) wall-clock time, b) speedup and c) efficiency for four different quantities of process in y-direction in a mesh with 1945 and 409 points in the x- and y-direction, respectively.