19th Brazilian Congress of Thermal Sciences and Engineering
November 6th–10th, 2022, Bento Gonçalves - RS - Brazil

ENC-2022-0610
A GPU ACCELERATED ALGORITHM FOR SOLVING NAVIER-STOKES EQUATIONS

Daniel Botezelli
Elisan dos Santos Magalhães
Davi Antônio dos Santos
Aeronautics Institute of Technology – ITA
botezelli@ita.br

Abstract. *Modern parallel programming technology has granted a recent speed enhancement in the system of equations solutions. The present paper proposes a CUDA-C-based numeric algorithm to analyze GPU devices' acceleration in solving thermo-fluid problems through the heat transfer and Navier-Stokes equations. The thermal study is approached through the enthalpy function of non-linear thermal conductivity. In order to study this algorithm's performance, this paper solves a two-dimensional lid-driven cavity benchmark with heat conduction in various Reynolds Number (Re), linear system of equations solvers, and mesh refinement. One of the objectives is to develop a high accelerated graphics processing units (GPU) based algorithm. The problems are numerically solved using the semi-implicit finite volume method in a GPU device. The results show that the parallel solver achieved 400-450 times faster than CPU-based codes under double variables precision.*

Keywords: *GPU solver, lid-driven cavity, real-time simulation*

1. INTRODUCTION

The graphics processing units (GPU) have drawn attention by exponentially enhancing their processing power in the last decades [1]. GPU-based codes have higher algorithm velocity due to their bandwidth that solves simultaneous operations. Despite this evolution, parallel coding in GPU is relatively new in the Computational Fluid Dynamics (CFD) scenery. The CFD is one of the most crucial engineering tools used to solve therm-fluids problems. It has become a reliable tool for projecting heat transfer and fluid dynamics [2]. Optimizing the CFD simulation time while reducing their costs is mandatory. Thus, GPU-based codes have shown to be a promising way to address this problem.

The use of GPU devices allows significantly lower simulations costs. Since GPU devices provides faster simulations, the use of clusters may be less needed. Vallero et al. [3] show that the convergence rate enhancement using a GPU-based algorithm is a new avenue and promising way to faster CFD analysis. As the simulation time is shorter with parallel solvers, the computational cost is minimal. In fluid mechanics, many researchers are investigating this topic. For instance, Cohen et al. [1] got double-precision computations for solving the Boussinesq equation up to eight times faster than a single-precision solver. They acquired simulation time reduction and concluded that this reductions might allow real-time solutions. This is indeed a major challenge for engineering control and fast investigations. Real-time simulations may improve the CFD investigations since it allows for instant valuable data visualization. On the other hand, GPU devices provide ample data parallelism (bandwidth), allowing for the dispersion of this effort in parallel evaluation. Therefore, one of the main challenges of fluid flow modeling is to develop numerical methods in an accurate and computationally efficient way.

According to Niksiar et al. [4], the parallelism degree depends on the solver algorithm. For them, Jacobi solvers are more accessible to parallelize than Gauss-Seidel solvers [5]. On the other hand, the successive over-relaxation (SOR) [6], a variant of the Gauss-Seidel algorithm, can be fully parallelized in the GPU device. Despite its slow convergence rate, when parallelized, SOR becomes much faster compared to serial form. Magalhães and Lemos [7] proposed a modified SOR (SOR-M) to solve, in a GPU device, the unsteady state heat diffusion equation. They have found that SOR-M reduced by 30% the computational time in comparison with the classical SOR. Tutkun et al. [8] applied a high-order compact finite difference scheme on GPU to achieve a solution between 9–16.5 faster than a CPU-based code. Besides the direct solvers, one can find the use of multigrid solvers applied in GPU devices in the Shi et al. [9] paper. As a general rule, each linear equation solver has particularities and is going to perform well for a determined case. Besides the stationary methods, SOR and Jacobi solvers, there are non-stationary methods. This class of solvers applies a sequence of orthogonal vectors to determine the residuals of the iterates. Among this class, there is the Conjugate Gradient Squared (CGS) solver. The CGS solver is a useful tool in approximating solutions to linear partial differential equations. According to Zabidin et al. [10], this method is highly efficient since it does not cost memory or storage of the second derivative. The drawback is that the particular systems of linear equations must be positive-definite and the use of a pre-conditioner is nearly mandatory. CGS solvers have high parallelism degree and using SOR solver as pre-conditioner seems a promising way to faster solutions than Jacobi solvers in GPU-based codes. Therefore, it is hard to establish a general solver to be used in CFD applications.

As the solvers are continually being proposed, many researchers compared their solution to the classical lid-driven cavity problem. The lid-driven cavity is a renowned benchmark problem for viscous incompressible fluid flow [11]. For instance, Lin et al. [12] applied the Multi Relaxation Time (MRT) and Lattice Boltzmann Equation (LBE) to simulate lid-driven cavity flows in GPU. Franco et al.[13] solved a 2D lid-driven cavity flow simulation in a GPU device and obtained a seventy times faster solution concerning their CPU-based algorithm. AbdelMigid et al. [2] proposed an accelerated GPU-based code to the lid-driven cavity flow problem. Ghia et al. [14] solved this problem for several Reynolds numbers (Re). Their work is a reference for the lid-driven solution [4] [2] [15] [16]. Thus, this paper proposes an accelerated GPU-based algorithm for precise fluid flow model and validates the results through Ghia et al. [14] work.

2. Methodology

2.1 Lid-driven cavity model

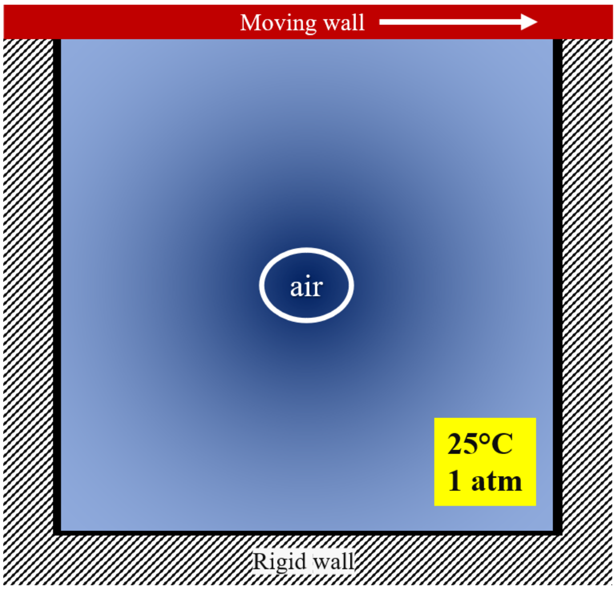


Figure 1: Schematic representation of the lid-driven cavity flow.

Figure 1 presents a schematic view of the lid-driven cavity flow. It consists of a square with a $1\,m$ side cavity with three rigid walls and a lid moving with a constant tangential velocity. The problem is solved for the same pressure and temperature in different Reynolds numbers: $100, 1000$, and 3200. The range chosen allows for simulating the flow with the maximum agreement with the real fluid dynamics [17]. In a fully developed flow, laminar flows occur around $Re_L = 3200$ [18], depending on other factors such as surface roughness and flow uniformity [19].

The non-linear two-dimensional incompressible Navier-Stokes equations can model this fluid problem. Most studies use these equations in dimensionless form, where Reynolds can be set directly. However, this paper uses a dimensional form. Consequently, the Reynolds number is set by changing the lid horizontal velocity u_L, since the Reynolds number is given by

$$Re_L = u_L\left(\frac{\rho L}{\mu}\right), \tag{1}$$

where ρ is the fluid density, L is the characteristic linear dimension and μ is the fluid's dynamic viscosity. Consider a transient two-dimensional laminar incompressible flow lid-driven cavity, the the continuity equation and the Navier-Stokes momentum equation for the velocity field $\boldsymbol{u}$ are given by

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{2}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\mu \boldsymbol{\nabla} \boldsymbol{u}) + \rho \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} = -\nabla p, \tag{3}$$

where ρ is the density, μ is the dynamic viscosity, p is the pressure, and t is the time.

2.2 Numerical model

One can discretize equation 3 to solve the problem. This work applied the co-located grid for pressure and velocity components [20]. Figure 2 presents the co-located grid. All solution vector and scalar numbers are stored in the cell's centers.

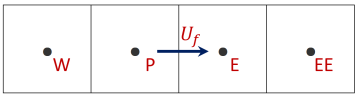


Figure 2: Co-located grid concept.

Integrating these equations over the control volume $\forall$, it is possible to obtain:

$$\begin{aligned} \int_{t}^{t+\Delta t} \int_{\Delta \forall} \frac{\partial (\rho \boldsymbol{u})}{\partial t} d\forall dt + \int_{t}^{t+\Delta t} \int_{\Delta \forall} \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \boldsymbol{u})\, d\forall dt = \\ \int_{t}^{t+\Delta t} \int_{\Delta \forall} \boldsymbol{\nabla} \cdot (\mu \nabla \boldsymbol{u}) d\forall dt - \int_{t}^{t+\Delta t} \int_{\Delta \forall} \frac{\partial p}{\partial \boldsymbol{x}} d\forall dt. \end{aligned} \tag{4}$$

Equation 4 can be written in finite difference applying the finite volume method for solving the above problem. For a bi-dimension cavity, we have

$$a_P u_P = \sum_{nb} a_{nb} u_{nb} + A\,(p_w - p_e) + b_x, \tag{5}$$

$$a_P v_P = \sum_{nb} a_{nb} v_{nb} + A\,(p_s - p_n) + b_y, \tag{6}$$

where a is the linker coefficient, b is the source term, and A is the cell's area. The subscript nb refers to the neighbors of the P cell. For the mass flux calculation, we need to use the cell face velocities through the volume domain. The momentum equations 5 and 6 can be re-written in a semi discretised form as follows:

$$\boldsymbol{\mathcal{A}} \boldsymbol{u} = \boldsymbol{\mathcal{H}} - \nabla p, \tag{7}$$

where

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} a_{1,P} & 0 & 0 & ... & 0 \\ 0 & a_{2,P} & 0 & ... & 0 \\ 0 & 0 & a_{3,P} & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & a_{n,P} \end{bmatrix}, \tag{8}$$

$$\boldsymbol{\mathcal{H}} = -\sum_{nb} a_{nb}\boldsymbol{u}_{nb} + \frac{\boldsymbol{u}^{o}}{\Delta t}. \tag{9}$$

One can remember that all information is stored in the cell's center. In Figure 2, velocity U_f is the east face velocity of P-cell. We need to interpolate the cell-centered velocities to calculate the face velocity. To address this problem, Rhie and Chow [**?**] propose a high order pressure interpolation, where west (W), prime (P), east (E), and far-east (EE) pressures are used for velocity U_f interpolation given by

$$U_f = \overline{U_f} - \overline{d_f}\left(\left.\frac{\partial p}{\partial x}\right|_f - \overline{\left.\frac{\partial p}{\partial x}\right|_f}\right), \tag{10}$$

where, the over-line is the distance interpolation modificator, $d = \dfrac{\not{V}}{a_P}$, and $\not{V}$ is the cell volume. Furthermore, right after solving the momentum equations, it is necessary to correct the velocity field so it satisfies the continuity equation. Let us take

$$\boldsymbol{u} = \boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{H}} - \boldsymbol{\mathcal{A}}^{-1}\nabla p. \tag{11}$$

Hence, by substituting this equation into the discretised continuity equation, we obtain the pressure correction p' equation (Poisson equation):

$$\boldsymbol{\nabla}\cdot\left(\boldsymbol{\mathcal{A}}^{-1}\nabla p\right) = \boldsymbol{\nabla}\cdot\left(\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{H}}\right) = \sum_{f}\boldsymbol{S}\left(\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{H}}\right)_f, \tag{12}$$

where $\boldsymbol{S}$ is the outward-pointing face area vector. The SIMPLE algorithm, by Patankar et al. [21], is used to solve different kinds of fluid flow and heat transfer problems. Figure 3 shows the the basic steps in the iterative solution update.

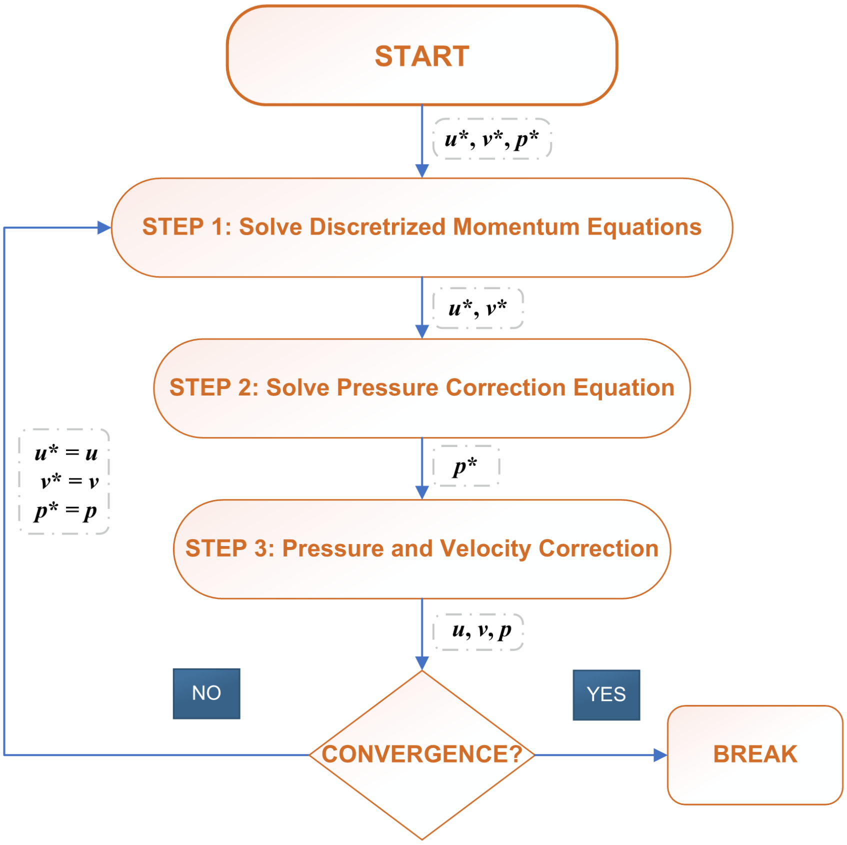

START
u*, v*, p*
STEP 1: Solve Discretrized Momentum Equations
u*, v*
STEP 2: Solve Pressure Correction Equation
p*
STEP 3: Pressure and Velocity Correction
u, v, p
u* = u
v* = v
p* = p
NO
YES
CONVERGENCE?
BREAK

Figure 3: SIMPLE method flowchart.

2.3 Boundary conditions

The boundaries conditions for the lid-driven scenery are presented. According to the no-slip condition and the velocity across the thickness lid, the velocities at the walls are zero. Furthermore, the reference point in the lower-left corner of the cavity has zero pressure. In redundancy, the pressure gradient across the wall is set to zero. Figure 4 shows the schematic boundary conditions view.

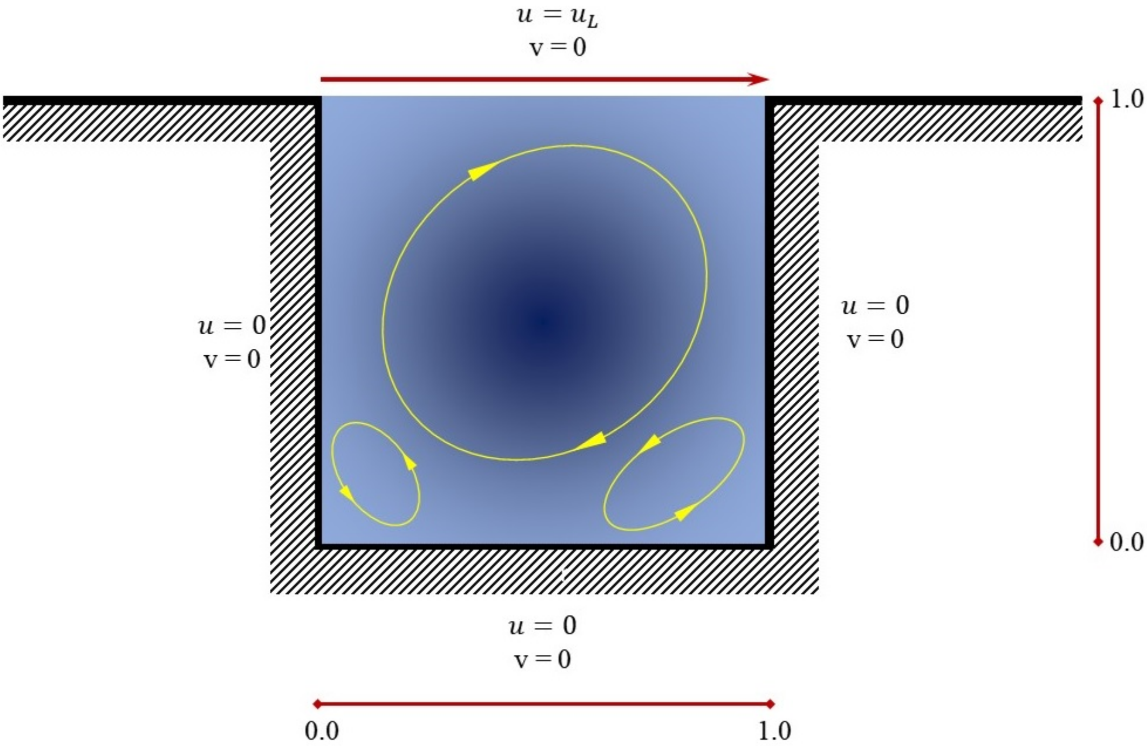


Figure 4: Lid-Driven Cavity boundary conditions.

2.4 Linear Equations System Solver

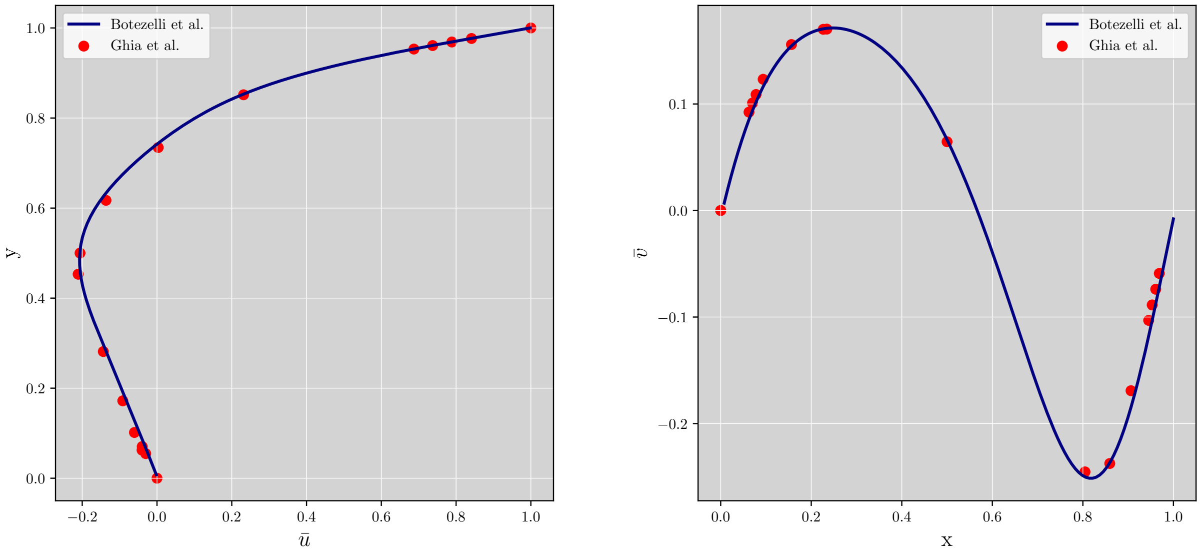
Each loop for the SIMPLE algorithm, we need to solve the linear system of equations 7 and 12, namely those whose matrix is positive-definite. To address this problem, the Conjugate Gradient Squared (CGS) using Sucessive Over-Relaxation (SOR) as pre-conditioner. Usually, these methods can be massively parallelized in GPUs, as shown in [22] paper. These methods are based on the Krylov subspace, which can converge in a theoretically finite number of steps. Thus, this paper applies the CGS-SOR solver method to investigate GPU devices' potential precision and fastness in coping with the numerical solution of the Navier-Stokes equation..

3. Results

3.1 The Lid-driven Cavity Problem

Using the standard ambient temperature and pressure (SATP) for the constant air properties and convergence residual $R = 1.0 \times 10^{-6}$, the obtained results are compared by plotting the velocity components along the lines passing through the geometric center of the cavity. The agreement between this paper and Ghia et al. [14] results are shown in figures 5, 6 and 7.

Figures 5a and 5b presents the results for $Re = 100$. At low Reynolds number, the viscous forces are dominant. Therefore, this case tested the simulation performance at this range. For the velocity components u and v along the center vertical line and horizontal line, respectively, figures 5a and 5b show good agreement with Ghia et al. [14] results.

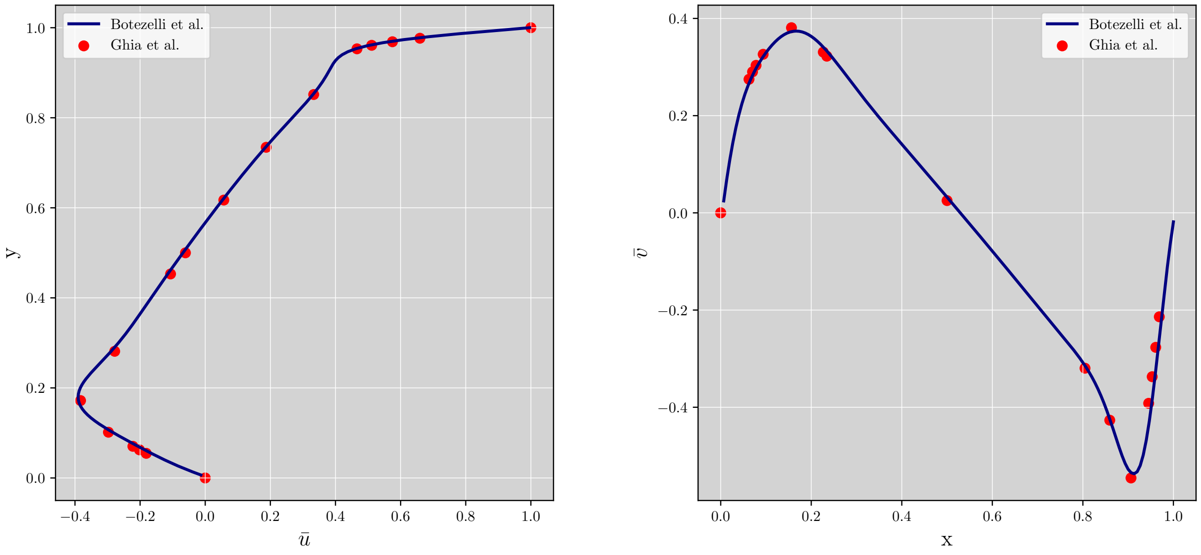


(a) u-velocity along vertical Line

(b) v-velocity along horizontal Line

Figure 5: Results for velocity along line through geometric center of cavity for **Re = 100.**

Figures 6a and 6b displays the obtained results for Re = 1000.

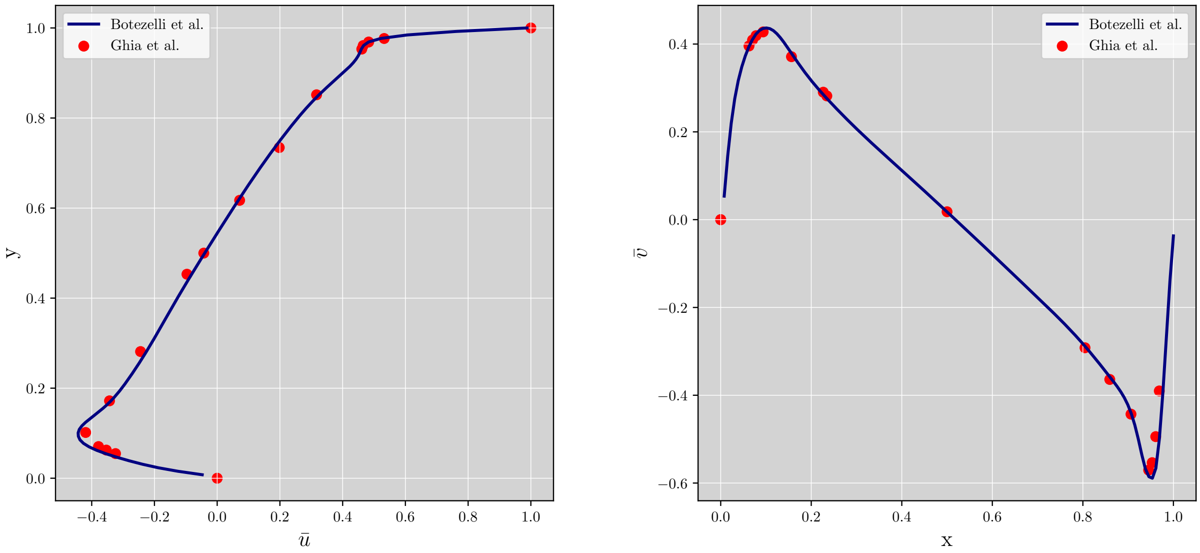


(a) u-velocity along vertical Line

(b) v-velocity along horizontal Line

Figure 6: Results for velocity along line through geometric center of cavity for **Re = 1000.**

Finally, figures 7a and 7b present the obtained results for $Re = 3200$ case. Once again, the an agreement is achieved.

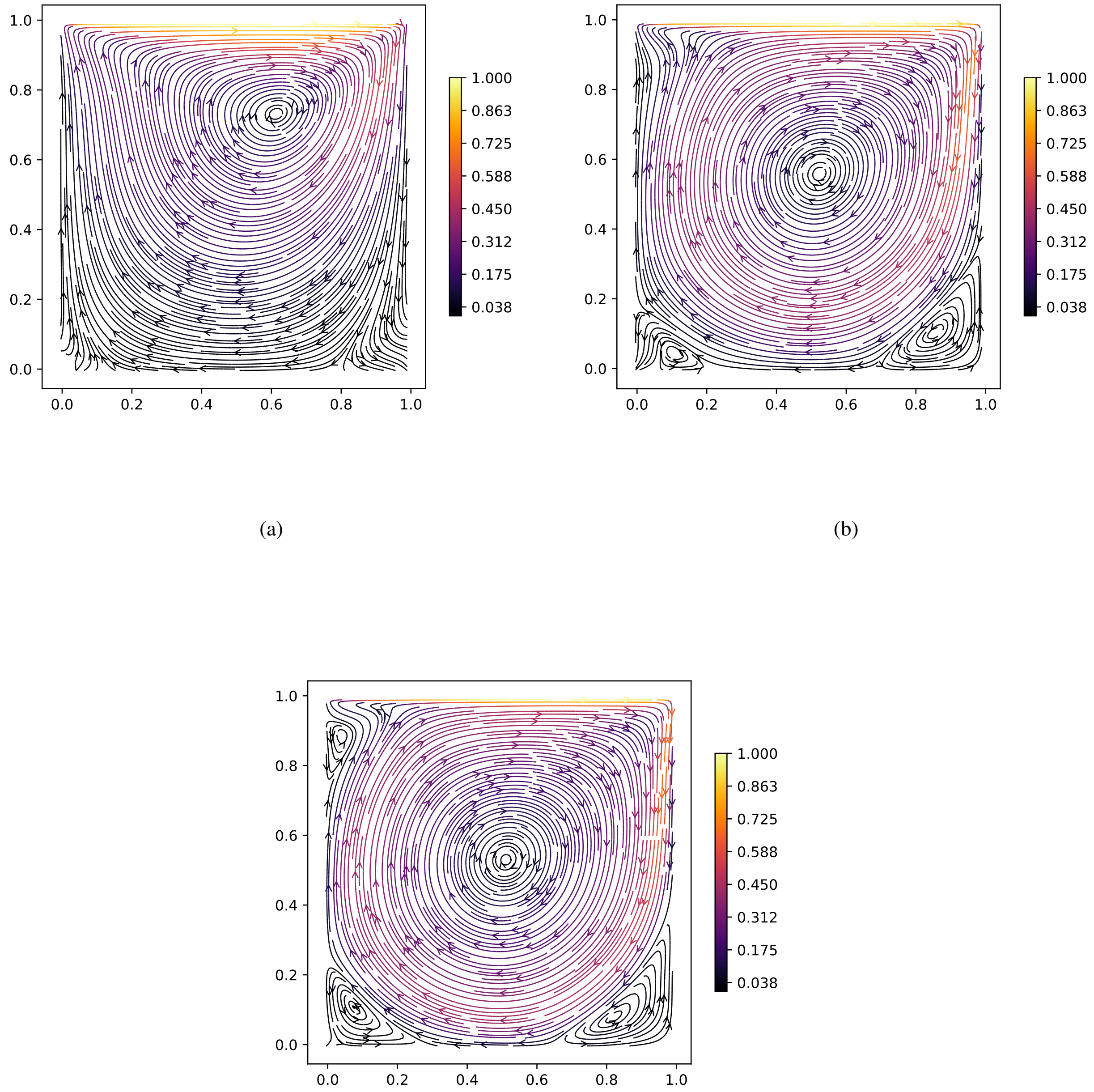


(a) u-velocity along vertical Line

(b) v-velocity along horizontal Line

Figure 7: Results for velocity along line through geometric center of cavity for **Re = 3200.**

Figure 8 presents the streamlines, which represent the path of a massless particle in a fluid relative to a solid body in a laminar regime and steady-state. These streamlines are tangent to the velocity vector of the flow. Knowing that a fluid particle can not have two different velocities simultaneously, different streamlines at the same instant do not intersect. Note that the Reynolds number directly influences the streamlines. The solutions exhibit additional counter-rotating vortices in or near the cavity corners as Re increases. Also, the primary vortex center moves towards the center with this increase.



(a)

(b)

(c)

Figure 8: Cavity streamlines: (a) Re = 100; (b) Re = 1000; (c) Re = 3200.

3.2 Performance Test

Solving for Re = 1000 and 40000 cells, the CPU-based solver converges to the solution in 4414.2 seconds. On the other hand, when paralleled in device, the solution converges in 10.1 seconds. Therefore, when comparing GPU and CPU-based algorithms for the same problem and parameters, the speedy-up is around 420–450 times. An interesting fact is that the mesh size does not significantly affect the GPU simulation time.

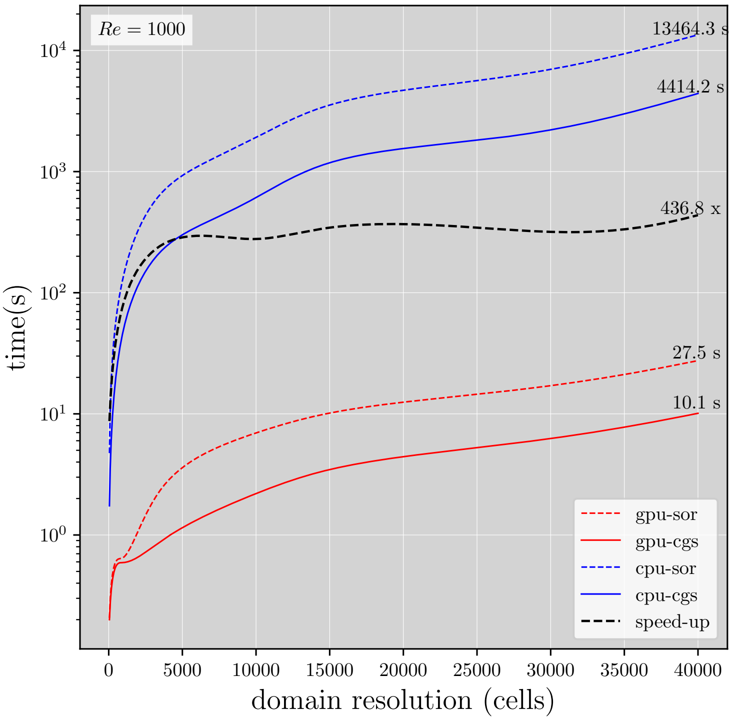


Figure 9: Simulation time for respective domain resolution.

4. Conclusion

This paper presented a methodology to solve the two-dimensional Navier-Stokes equations to study the difference between GPU and CPU-based algorithms. The traditional lid-driven problem was solved by this methodology and compared to others works results. The numerical analysis showed that the numerical algorithm to solve the minimization problem is one of the keys for fast and lower costs simulations. The CGS and SOR solvers have shown good performance. Although the SOR is a slow method, it is highly effective when applied to the GPU device. Furthermore, the 430x times speed-up reveals good enhancement in convergence rate using a GPU-based algorithm for the presented problem. One can say that GPU devices are efficient for large-scale computations, since the domain resolution scale did not effected simulation time significantly. Parallel simulation using graphic devices may be a new generation for engineering simulations with faster results, lower costs, and real-time engineering. Future works with multi-grid application may reveal even better results in solver performance.

5. ACKNOWLEDGEMENTS

The authors would like to thank CAPES and CNPq for their financial support.

6. REFERENCES

[1] Jonathan Cohen and M Molemaker. A fast double precision cfd code using cuda. *Parallel Computational Fluid Dynamics: Recent Advances and Future Directions*, 01 2009. doi: 10.1.1.515.4365.

[2] Tamer A. AbdelMigid, Khalid M. Saqr, Mohamed A. Kotb, and Ahmed A. Aboelfarag. Revisiting the lid-driven cavity flow problem: Review and new steady state benchmarking results using gpu accelerated code. *Alexandria Engineering Journal*, 56(1):123–135, 2017. doi: 10.1016/j.aej.2016.09.013.

[3] Daniel A. Vallero. Chapter 14 - air pollution dispersion models. *Air Pollution Calculations*, pages 429–448, 2019. doi: https://doi.org/10.1016/B978-0-12-814934-8.00014-4. URL `https://www.sciencedirect.com/science/article/pii/B9780128149348000144`.

[4] Pooya Niksiar, Ali Ashrafizadeh, Mehrzad Shams, and Amir Madani. Implementation of a gpu-based cfd code. *Proceedings - 2014 International Conference on Computational Science and Computational Intelligence, CSCI 2014*, 1:84–89, 03 2014. doi: 10.1109/CSCI.2014.21.

[5] I.C. Kampolis, X.S. Trompoukis, V.G. Asouti, and K.C. Giannakoglou. Cfd-based analysis and two-level aerodynamic optimization on graphics processing units. *Computer Methods in Applied Mechanics and Engineering*, 199(9):712–722, 2010. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2009.11.001. URL `https://www.sciencedirect.com/science/article/pii/S0045782509003648`.

[6] D. Young. Iterative methods for solving partial difference equations of elliptic type. *Transactions of the American Mathematical Society*, 76:92–111, 1954.

[7] Elisan dos Santos Magalhães and Marcelo J.S. de Lemos. A thermal study of a new oil well plugging & abandonment operation. *International Journal of Thermal Sciences*, 155: 106421, 2020. ISSN 1290-0729. doi: https://doi.org/10.1016/j.ijthermalsci.2020.106421. URL `https://www.sciencedirect.com/science/article/pii/S1290072919313225`.

[8] Bulent Tutkun and Firat Oguz Edis. A gpu application for high-order compact finite difference scheme. *Computers & Fluids*, 55:29–35, 2012. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2011.10.016. URL `https://www.sciencedirect.com/science/article/pii/S0045793011003227`.

[9] Xiaolei Shi, Tanmay Agrawal, Chao-An Lin, Feng-Nan Hwang, and Tzu-Hsuan Chiu. A parallel nonlinear multigrid solver for unsteady incompressible flow simulation on multi-GPU cluster. *Journal of Computational Physics*, 414: 109447, August 2020. doi: 10.1016/j.jcp.2020.109447.

[10] Qingli Zhao, Ahmad Alhawarat, Thoi Trung Nguyen, and Ramadan Sabra. An efficient modified azprp conjugate gradient method for large-scale unconstrained optimization problem. *Journal of Mathematics*, 04 2021. ISSN 2314-4629. doi: 10.1155/2021/6692024.

[11] Olek Zienkiewicz and R. Taylor. *The Finite Element Method*, volume I. Butterworth-Heinemann; 7th Revised ed., Oxford, United Kingdom, 08 2013. ISBN 9780080531670.

[12] Li-Song Lin, Hung-Wen Chang, and Chao-An Lin. Multi relaxation time lattice boltzmann simulations of transition in deep 2d lid driven cavity using gpu. *Computers and Fluids*, 80(Complete):381–387, 2013. doi: 10.1016/j.compfluid.2012.01.018.

[13] Ediguer E. Franco, Helver M. Barrera, and Santiago Laín. 2d lid-driven cavity flow simulation using gpu-cuda with a high-order finite difference scheme. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 37 (4):1329–1338, 2015. doi: 10.1007/s40430-014-0260-x. Technical Editor: Francisco Ricardo Cunha.

[14] U Ghia, K.N Ghia, and C.T Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3): 387–411, 1982. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(82)90058-4. URL `https://www.sciencedirect.com/science/article/pii/0021999182900584`.

[15] Carlos H. Marchi, Roberta Suero, and Luciano K. Araki. The Lid-Driven Square Cavity Flow: Numerical Solution with a 1024 x 1024 Grid. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 31(3), September 2009. ISSN 1678-5878. doi: 10.1590/s1678-58782009000300004. URL `http://dx.doi.org/10.1590/s1678-58782009000300004`.

[16] Roland Lewis, K. Ravindran, and Asif Usmani. Finite element solution of incompressible flows using an explicit segregated approach. *Archives of Computational Methods in Engineering*, 2:69–93, 12 1995. doi: 10.1007/BF02736197.

[17] Victor L. Streeter and E Benjamin Wylie. *Fluid mechanics*. Tata McGraw-Hill, New Delhi, 1983.

[18] Ajay K. Prasad and Jeffrey R. Koseff. Reynolds number and end-wall effects on a lid-driven cavity flow. *Physics of Fluids A: Fluid Dynamics*, 1(2):208–218, 1989. doi: 10.1063/1.857491. URL `https://doi.org/10.1063/1.857491`.

[19] Hermann Schlichting and Klaus Gersten. *Boundary-Layer Theory*. Springer, Springer, Berlin, Heidelberg, 01 2017. ISBN 978-3-662-52917-1. doi: 10.1007/978-3-662-52919-5.

[20] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics - the finite volume method*. Addison-Wesley-Longman, New York, 1995. ISBN 978-0-582-21884-0.

[21] Suhas V Patankar. *Numerical heat transfer and fluid flow*. Series on Computational Methods in Mechanics and Thermal Science. Hemisphere Publishing Corporation (CRC Press, Taylor & Francis Group), 1980. ISBN 978-0891165224. URL `http://www.crcpress.com/product/isbn/9780891165224`.

[22] Jiqun Tu, M. A. Clark, Chulwoo Jung, and Robert D. Mawhinney. *Solving DWF Dirac Equation Using Multi-Splitting Preconditioned Conjugate Gradient with Tensor Cores on NVIDIA GPUs*. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450385633. URL `https://doi.org/10.1145/3468267.3470613`.

7. RESPONSIBILITY NOTICE

The following text, properly adapted to the number of authors, must be included in the last section of the paper:
The author(s) is (are) solely responsible for the printed material included in this paper.