25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-1522
NUMERICAL EFFICIENCY OF THE GMRES ALGORITHM IN HEAT TRANSFER PROBLEMS

Tobias Luiz Marchioro Toassi
Francisco Augusto Aparecido Gomes
Paulo Gemo Conci
Federal University of Technology - Paraná , Pato Branco, Brazil
tobiasluizot@gmail.com
franciscogomes@utfpr.edu.br
pauloagconci@gmail.com

Abstract. *In this paper we aim to study the efficiency and accuracy of Generalized Minimal Residual (GMRES), one of the most effective iterative methods for solving large sparse linear system of equations and the family of Incomplete LU preconditioners (ILU) to solve pure diffusive problems. Two cases are going to be analyzed, a rectangular domain subject to a steady heat conduction problem, and a unitary square domain subject to a transient problem. For the discretization was utilized the finite difference method, that consists of replacing the derivatives in the differential equations by finite difference approximations. A fully implicit scheme was utilized on the transient analysis, so it could eliminate the stability problem that occur on explicit schemes, as in these cases the maximum value of time step is define by the largest eigenvalue of the differential equation over the analyzed mesh, a reduction in the amount of computational time may often be observed when employing an implicit, rather than explicit finite difference scheme. Appliyng the method on several meshes, it was possible to analyze the rate of convergence or order of discretization, the results were compatible with the expected order. The GMRES presented several advantages over other iterative methods such as Gauss-Siedel, and was even more efficient when the linear system was preconditioned.*

Keywords: *Finite Difference Method, GMRES, Gauss-Siedel, Numerical Analysis, Diffusive Problems*

1. INTRODUCTION

In the recent years, numerical analysis are surely in a exponential grow, since more and more studies are published with this matter (Content *et al.*, 2013; Zahr and Persson, 2013). So, find a way to do this with less costs is an important subject of study, if not necessary. Hence, the computational power has grown and the necessity to simulate simplified problems has disappeared, once is now being possible to see more complex domains being discretized (Fossati, 2015).

Hence, large linear systems became more recurrent once the Computational Fluid Dynamic (CFD) has evolved, in the past years papers like (Nejat and Ollivier-Gooch, 2005; Chassing *et al.*, 2005) have reported the necessity to solve large number of equations simultaneously, this kind of approach is derived by a more complex analysis of the problem, once this occur, the computational cost becomes more expensive, and the solution has taken more time to converge. So much research has been done in this field, and the utilization of the GMRES algorithm, proposed first by Saad (Saad and Schultz, 1986), has given much desired results when the goal is the speed of converge (Wigton *et al.*, 1985). The GMRES applicability is increasing in studies involving heat transfer (Blomquist, 2019), and is plenty useful to solve heat transfer simulation problems, such as multi-harmonic micro-scale wavy channels (Moon *et al.*, 2018), radiative heat transfer on three-dimensional semitransparent media (Han *et al.*, 2018), two-dimensional linear transient inverse heat conduction problem applied on quenching process by water jets (Jahedi *et al.*, 2018), among others.

Since GMRES has such good propriety, it's application in this problems are widely used such in CFD problems (Nejat and Ollivier-Gooch, 2005; Wigton *et al.*, 1985), but the subject in this paper is yet to be study, that is the numerical efficiency of this algorithm when solving large linear system obtained with the Finite Difference Method (FDM) in heat transfer problems, this problems have a characteristic that the linear system is symmetric and definite positive (LeVeque, 2007), this problems are more easily solved by any iterative solver. This paper aim to study two different cases, one is a steady-state heat transfer problem, which results in a linear system that only depends of the spatial parameters, the other is a transient heat transfer problem, which has to be discretized in space and time.

The finite difference method produces a linear system diagonal dominant, and the number of diagonals depend of the number of nodes used on the discretization process (Reddy, 2006). In this work was utilized Central Difference Scheme of second order (CDS-2), which results a penta-diagonal symmetric linear system. In transient analysis, the time discretization

has to be chosen carefully. Is well know three different explicit schemes produces the solution in the n-th time step with relation to the solution in the time step before, this schemes don't require much work mathematically, but when compared with other methods is much more expensive, once is unstable with relation to time step. The Crank-Nicolson, solve in part this problem, but not totally, once is require to set another parameters. The implicit method produces a linear system of equations that are unconditionally stable independent of the time step, and that was the method used in this work, once we rely on the numerical method to solve the linear system, don't matter how large the system is.

That said, the subject of study of this work is of great relevance in the study of numerical problems involving heat transfer. The use of a numerical method to solve linear systems with high convergence velocity for the solution represents an advance in the study of numerical problems that are presenting increasingly complex domains, optimizing the current computational resources and consequently the cost of these resources.

2. MATHEMATICAL MODEL

In the field of mechanical engineering, heat transfer problems occurs very often. Solving them with accuracy and a reasonable time is the aim of every research in this field. This work has the purpose of investigate the efficiency of an algorithm known as Generalized Minimal Residual (GMRES). Heat transfer analysis problems tries to estimate the temperature distribution and/or the heat flow on the domain surface (Kreith *et al.*, 2011). We focus in the temperature distribution that will be used after to calculate some numerical properties, such is the rate of convergence.

When the domains is a rectangular square in two-dimensional space and have no heat sources, the general governing equation of the temperature distribution is the Laplace's equation.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{1}$$

The solution of Eq. (1) will give $u(x, y)$, the temperature as a function of the coordinates x and y. The solution of Eq. (1), must satisfy the boundary conditions specified by the physical conditions in every single problem.

The domain analyzed was a rectangular square, of dimensions L and b and subject to the following boundary conditions.

$$\begin{aligned} &u(x,0) = 0, \quad u(x,b) = T_m \sin\left(\frac{\pi x}{L}\right) \\ &u(0,y) = 0, \quad u(L,y) = 0 \end{aligned} \tag{2}$$

Where $L = 30$, $b = 20$ and $T_m = 200$.

Hence subject to this boundary conditions, it can find the exact solution, or the analytical solution. For this it can use different methods, one of them is separation of variables.

It going to be obtained the following function of x and y, that represent the distribution of temperature over the domain.

$$u(x,y) = T_m \frac{\sinh(\pi y/L)}{\sinh(\pi b/L)} \sin(\pi x/L) \tag{3}$$

Transient heat conduction problems, are governed by another equation, know as Poisson equation.

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{4}$$

where α is the thermal diffusivity, in m^2/s. In this case a two-dimensional unity square was subject to the following boundary conditions.

$$\begin{aligned} &\frac{\partial u}{\partial x}(0,y,t) = 0, \quad \frac{\partial u}{\partial y}(x,0,t) = 0 \\ &u(1,y,t) = 0, \quad u(x,1,t) = 0 \end{aligned} \tag{5}$$

And the initial condition

$$u(x,y,0) = 0, \qquad \text{for all } (x,y) \text{ in } \Omega \tag{6}$$

3. COMPUTATIONAL PROCEDURE

3.1 Finite Diffence Method

Elliptic equations arise very often from steady-state problems such is the first problem analyzed in this work. With the necessary mathematical considerations, we going to have the Laplace Eq. (1), considering the first domain Ω, subject to essentials boundary conditions (Dirichlet), i.e., the primitive variable is specified on the boundary.

The finite differences method consist in replace the derivatives in the differential equation with finite difference approximations (LeVeque, 2007). Considering an expansion using the Taylor series about a point $\bar{x}$.

$$u(\bar{x}+h) = u(\bar{x}) + hu^{'}(\bar{x}) + \frac{1}{2}h^2u^{''}(\bar{x}) + \frac{1}{6}h^3u^{'''}(\bar{x}) + O(h^4) \tag{7.1}$$

$$u(\bar{x}-h) = u(\bar{x}) - hu^{'}(\bar{x}) + \frac{1}{2}h^2u^{''}(\bar{x}) - \frac{1}{6}h^3u^{'''}(\bar{x}) + O(h^4) \tag{7.2}$$

Using the above Eqs. (7.1) and (7.2), we going to define the following operator.

$$\begin{aligned} D_+u(\bar{x}) &= \frac{u(\bar{x}+h)-u(\bar{x})}{h} \\ &= u^{'}(\bar{x}) + \frac{1}{2}hu^{''}(\bar{x}) + \frac{1}{6}h^2u^{'''}(\bar{x}) + O(h^3) \end{aligned} \tag{8.1}$$

$$\begin{aligned} D_-u(\bar{x}) &= \frac{u(\bar{x}-h)-u(\bar{x})}{h} \\ &= -u^{'}(\bar{x}) + \frac{1}{2}hu^{''}(\bar{x}) - \frac{1}{6}h^2u^{'''}(\bar{x}) + O(h^3) \end{aligned} \tag{8.2}$$

With the Eqs. (8.1) and (8.2), is possible to obtain the general operator for finite difference approximation, that is define as following.

$$\begin{aligned} D^2u(\bar{x}) &= D_+D_-u(\bar{x}) \\ &= D_+\left(D_-u(\bar{x})\right) \\ &= \frac{1}{h}\left[D_-u(\bar{x}+h) - D_-u(\bar{x})\right] \end{aligned} \tag{9.1}$$

$$= \frac{1}{h}\left[\left(\frac{u(\bar{x}+h)-u(\bar{x})}{h}\right) - \left(\frac{u(\bar{x})-u(\bar{x}-h)}{h}\right)\right] \tag{9.2}$$

Rearranging the Eq. (9.2), and going to obtain the standard second order centered approximation (CDS-2).

$$D^2u(\bar{x}) = \frac{1}{h^2}\left[u(\bar{x}-h) - 2u(\bar{x}) + u(\bar{x}+h)\right] \tag{10.1}$$

$$= u^{''}(\bar{x}) + \frac{1}{12}h^2u^{''''}(\bar{x}) + O(h^4) \tag{10.2}$$

Now is possible to rewrite the Eq. (1) in the discretized form using the Eq. (10.1). Considering the two-dimensional (2D) space, let u_{ij} represent the approximation to $u(x_i, y_j)$, where $x_i = i\Delta x$ and $y_j = j\Delta y$, that becomes.

$$\frac{1}{h^2}\left(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}\right) = 0 \tag{11}$$

The Eq. (11) is the discretized form of the Laplace equation for a 2D grid for interior nodes. As can be seen in the Eq. (10.2), this equation have a *priori* truncation error of second order with relation of grid size.

For the transient problem, a balance of energy scheme was utilized, which consist in create a control volume around the analyzed node, in this control volume is possible to calculate the balance of the secondary variables, this make possible to ensure the continuity of the primary variables.

$$\sum_{all}\dot{Q}^{n+1} + \dot{E}_{ger}^{n+1} = \rho VC_p\frac{\partial T}{\partial t} \tag{12}$$

Now using the technique seen above to discretize the partial derivative, for the interior nodes, we going to obtain the governing equation for transient heat diffusion. Noting that all the terms are evaluated in the next time step

$$Fo\left(T_{i-1,j}^{n+1} + T_{i+1,j}^{n+1} + T_{i,j-1}^{n+1} + T_{i,j+1}^{n+1}\right) - \left(4Fo+1\right)T_{i,j}^{n+1} = -T_{i,j}^{n} - \frac{\alpha\Delta t\dot{q}_{ger}}{\kappa} \tag{13}$$

where, Fo = Fourier Number; α = Thermal Diffusivity; κ = Thermal Conductivity.

The algorithm was implement in a programming language, there are many different languages with different characteristics, FORTRAN is one of the most utilized languages in general scientific computations, hence is very useful for numerical analysis (Chapman, 2007). Was possible to implement a couple of iterative methods, such as Gauss-Siedel and GMRES with a few preconditioners, such as ILU family. Including ILUT and MILU0.

3.2 Linear system solver algorithm: Generalized Minimal Residual (GMRES)

The Generalized Minimal Residual is a iterative method who efficiently solve large sparse positive definite linear systems of equations. It uses the Arnoldi Method for build a l_2 - orthogonal basis of Krylov subspaces, reducing the residual norm and time for convergence. This algorithm shows several advantages over previous developed methods, like the Gauss-Seidel, Generalized Conjugate Residual (GCR) and ORTHODIR (Golub and Van Loan, 2013).

The GMRES is a method based on the Eq. (14) (Saad and Schultz, 1986):

$$\mathcal{K} = \mathcal{K}_m \tag{14}$$

with $\mathcal{L}$ = $A\mathcal{K}_m$, where $\mathcal{K}_m$ is a Krylov subspace with m size and v_1 = $r_0/\|r_0\|_2$. The vectors { $v_1, v_2, ..., v_k$ } are a orthonormal basis calculated by the Gram-Schmidt method from the Arnoldi Method. The Krylov subspace is given by { $v_1, Av_1, ..., A^{k-1}v_1$ }. The Arnoldi Algorithm is defined by:

Start: choose a initial vector;
for $j = 1, 2, ...$ **do**
  $h_{i,j} = A(v_j, v_i) i = 1, 2, ..., j$;
  $\widehat{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{i,j} v_i$;
  $h_{j+1,j} = \|\widehat{v}_{j+1}\|$;
  $v_{j+1} = \widehat{v}_{j+1}/h_{j+1}$
end

Algorithm 1: Arnoldi Algorithm

The arrangement of the linear system to be solved is given on the Eq. (15)

$$Ax = b \tag{15}$$

To solve a linear system given on the Eq. (15) by the Galerkin method using a l_2 orthogonal basis given by V_k, the solution x_k is aprroximated on the $x_k = x_0 + z_k$ form, with x_0 is the initial estimate and z_k is a Krylov subspace member given by $\mathcal{K}_k \equiv span\{r_0, Ar_0, ..., A^{k-1}r_0\}$ with the residual (Eq. (16)):

$$r_0 = b - Ax_0 \tag{16}$$

By executing k iterations of the Arnoldi Algorithm on a initial vector v_1 = $r_0/\|r_0\|$, this leads to notice that the residual vector $r_k \equiv b - Ax_k$ is orthogonal to $\mathcal{K}_k$, so $z_k = V_k y_k$, where $y_k = H_k^{-1}\|r_0\|e_1$, where e_1 is the unit vector $e_1 \equiv (1, 0, ..., 0)^T$.

The GMRES algorithm goal is to approximate the solution on form $x_0 + z$ by minimizing the residual norm about z on $\mathcal{K}_k$. After k iterations of the Arnoldi algorithm we obtain a orthonormal - l_2 system V_{k+1} and a $(k+1) \times k$ matrix $\overline{H}_k$, with the only non-zero terms is the $h_{i,j}$ terms generated by the algorithm. Therefore, the difference between $\overline{H}_k$ and H_k is the line that contains the $h_{k+1,k}$ on $k+1, k$ position and the v_i vectors and $\overline{H}_k$ matrix satisfies (Eq. (17)):

$$AV_k = V_{k+1}\overline{H}_k \tag{17}$$

So, it has to be solved the following minimal quadratic problem:

$$\min_{z \in \mathcal{K}_k} \|b - A[x_0 + z]\| = \min_{z \in \mathcal{K}_k} \|r_0 - Az\| \tag{18}$$

If we assume $z = V_k y$, the residual norm by the function of y can be possible to minimize :

$$J(y) = \|\beta v_1 - AV_k y\| \tag{19}$$

assuming $\beta = \|r_0\|$, we have:

$$J(y) = \|V_{k+1}[\beta e_1 - \overline{H}_k y]\| \tag{20}$$

If e_1 is the unit vector and $V(k+1)$ is orthonormal - l_2,

$$J(y) = \|\beta e_1 - \overline{H}_k y\| \tag{21}$$

So the solution will be given as follows:

$$x_k = x_0 + V_k y_k \tag{22}$$

where y_k minimizes the $J(y)$ function on $y \in R^k$.

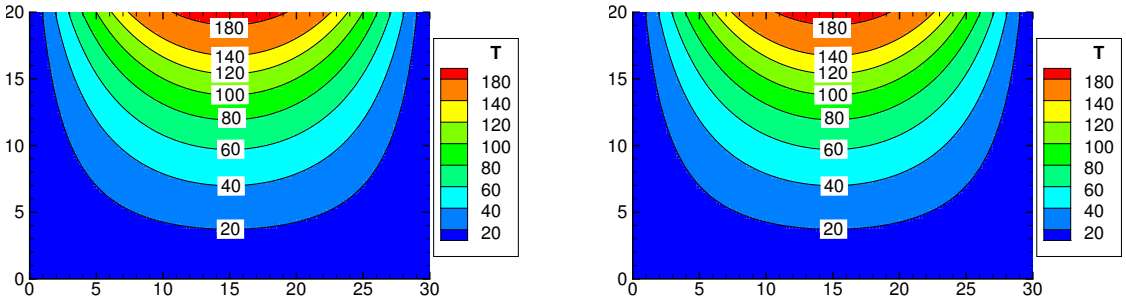
Once the weakness of iterative methods to solve linear systems is the lack of robustness, Pre-conditioning the linear system improves robustness to the method and leads to a more efficient computational work.

To preconditionate a Matrix by the method called Incomplete LU factorization - ILU), it have to calculate a Gaussian elimination on an $[A]$ matrix and dropping some elements in predetermined nondiagonal positions. However, it is possible to reduce the effect of dropping those elements by compensating for the discarded entries, who conducts to the Modified Incomplete LU factorization - MILU (Saad, 2003).

4. RESULTS AND DISCUSSION

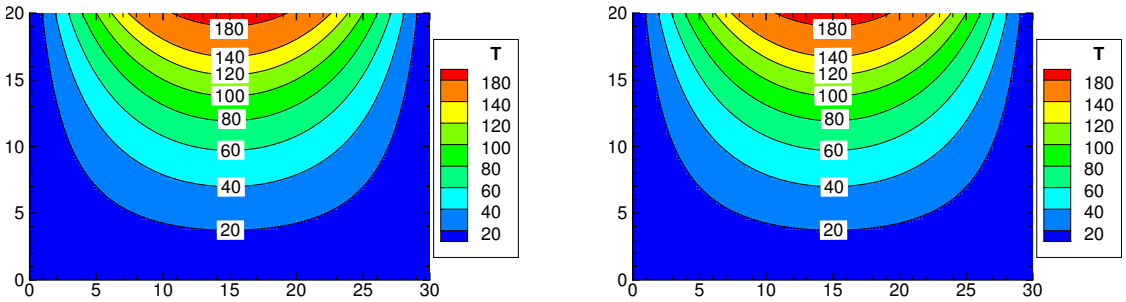
Two different domains were analyzed in this work, one was a rectangular domain, where the problem was numerically approximate the steady solution, and the second a square domain subject to a transient heat conduction problem. Both were discretized into a structure grid, and applied a finite difference method.

Once solved iteratively the Eq. (11), respecting the boundary conditions, is obtained the temperature field show in Fig. 1, hence is possible to compare with the temperature field obtained with the Eq. (3), shown in Fig. 1 (a), visually this field is very similar, and going to be analyzed *posteriori*, with error theory.



(a) (b)

Figure 1: (a) Analytical solution obtained with Eq. (3); (b) Approximate solution obtained iteratively with numerical methods.



(a) (b)

Figure 2: (a) Approximated solutions obtained from steady heat conduction problem for a mesh with 15000 nodes using GMRES without preconditioner; (b) Preconditioned with ILU0.

Can be observed in Fig. 2 that the temperature field for all methods is very similar, this reflects that don't matters what method is utilized, the results are the same. Was not shown all the fields due to similarity between the methods, so

we restrict to show the most significant methods with respect to convergence history, show in Fig. 3a, 3b and 3c, which represents the most efficient method to obtain the same solution.

In the steady-state heat transfer problem, is possible to identified which method converge most rapidly, this means that is necessary less computational cost, once the numerical method takes less time to converge. As it can be seen in Fig. 3, there is difference in the form which the same method converge with respect to the different grid size, one case that deserves mention is the GMRES without preconditioning, that in the grid with 10000 nodes converge to a value of $\epsilon = 10^{-11}$ with approximately 800 iterations, in a more refined mesh don't show such a good convergence, as it takes more than 1000 iterations to reduce the residual to a value of $\epsilon = 10^{-3}$ in a grid with 15000 nodes. This exhibit the necessity to use some preconditioner on the linear system, once become much more sparse because the refinement of the grid.

Once the preconditioner is used, it's possible to see that, don't matter what the grid size is, the number of iterations will be kept low, both for the grid with 10000 nodes as to the grid with 27000, and the round-off machine will be archived with less than 200 iterations. Worth mention that Gauss-Siedel, one of the most common method to solve linear system iteratively presented the same aspect to all grids analyzed, demonstrating your consistency, but since convergence is slow, the computational cost is to high. This makes impractical to simulate cases like the transient analysis, once it's necessary to solve one linear system in every time step.

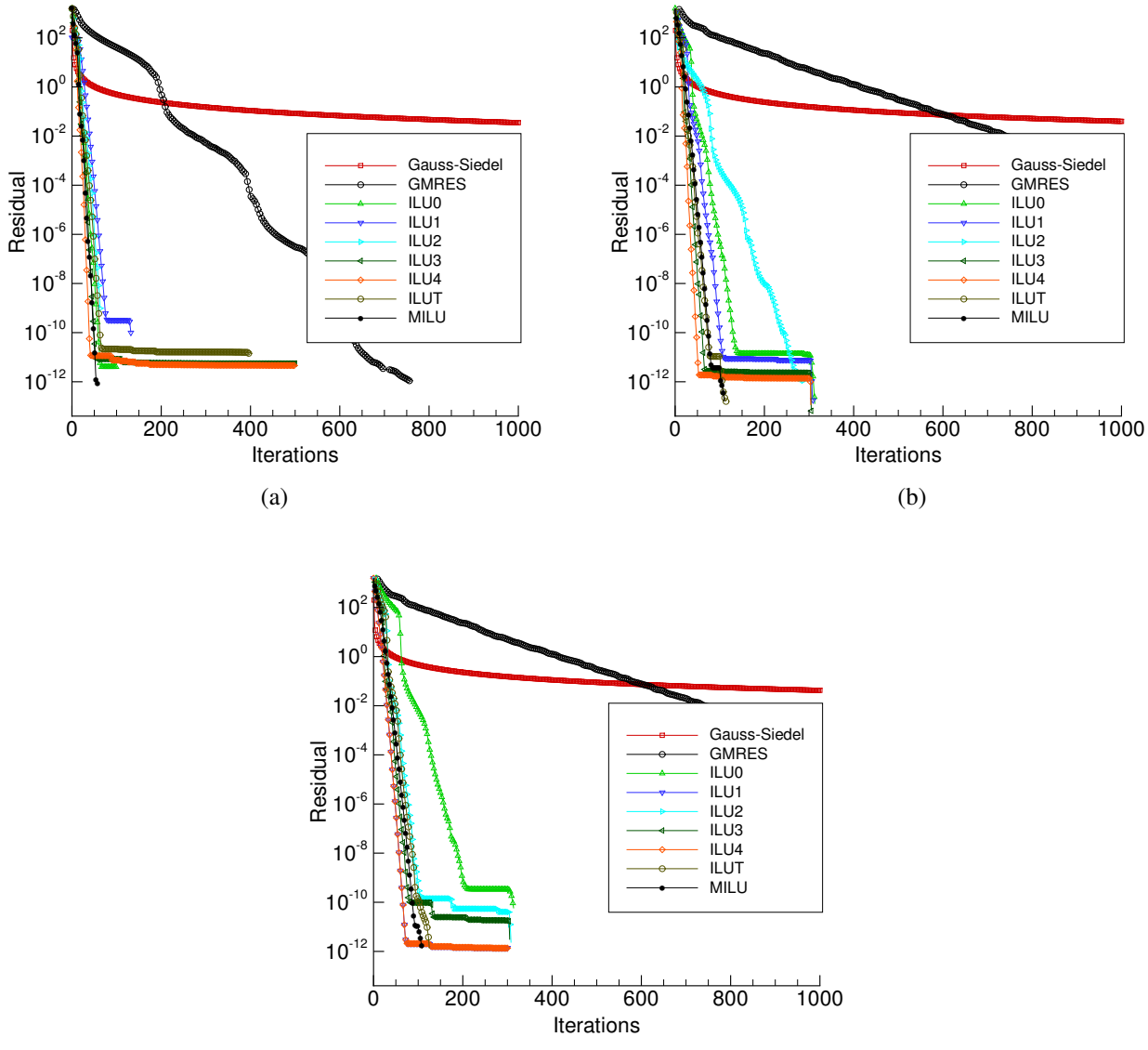


Figure 3: Convergence history for all methods over a mesh of (a) 10000, (b) 15000 and (c) 27000 nodes.

With the both solutions, analytical and numerical, i.e., solution approximated with a numerical method, is possible to

analyzed the order of convergence of the method and identify a possible degeneration in the numerical solution, which is the lost of the accuracy. Is important to test a computer code by doing the refinement study, even if the results looks good on a particular grid (LeVeque, 2007). Is common that some error in programming can lead to not so reasonable results when the grid is refined or modified. Is recommended when the exact solution is unknown utilized the numerical solution from a refined grid to compare, usually the results in this grid is more accurate, this occur very often in transient cases (LeVeque, 2007).

In the steady-state problem, we know the analytical solution, so is possible to define e_i which represents the calculation error in a specific grid, finite difference method produces a vector U, of dimension N, as an approximation to $u(x)$ only where x is a grid point and $u(x)$ is the function related with the analytical solution, which makes $U_i \approx u(x_i)$, this could be interpreted as an approximation to the point-wise value of the function at x_i

$$e_i = |U_i - u(x_i)| \qquad \text{where i=1,2,...,N.} \tag{23}$$

Now that the error vector is defined is possible to measure its magnitude using some norm. Since this is simply a vector of N components $(e_1, \ldots, e_N)$ is common to use the vector norms, which gives a misleading idea of error magnitude, once the decreasing the grid spacing h would give a larger error. So it's necessary to normalize the error vector, this can be done by simply multiplying the error vector norm by a factor of $s = \frac{1}{N}$, which gives.

$$\|e\|_q = \left(s \sum_{i=1}^{N} |e_i|^q \right)^{\frac{1}{q}} \tag{24}$$

This made possible to define the L_1 and L_2 norm, for grid functions just by replacing q which the respectively norm. Once we made $q \to \infty$ is logical that $s^{\frac{1}{q}} \to 1$, and this define the max-norm

$$\|e\|_\infty = \max_{1 \le i \le N} |e_i| \tag{25}$$

In the Fig. 4(a) is possible to see that all error norms behave like expect, once with the refinement of the grid, occur more accurate solutions, since the error has an asymptotic behavior.

Estimates the order of accuracy is now possible, but first is necessary to define that if some method is *p*th order accurate, is expected that

$$E(h) = Ch^p + o(h^p) \qquad \text{as} \quad h \to 0.$$

and if h is sufficiently small, we get

$$E(h) \approx Ch^p \tag{26}$$

By using different grids, we can estimate p as a function of the grid refinement and the error, which gives

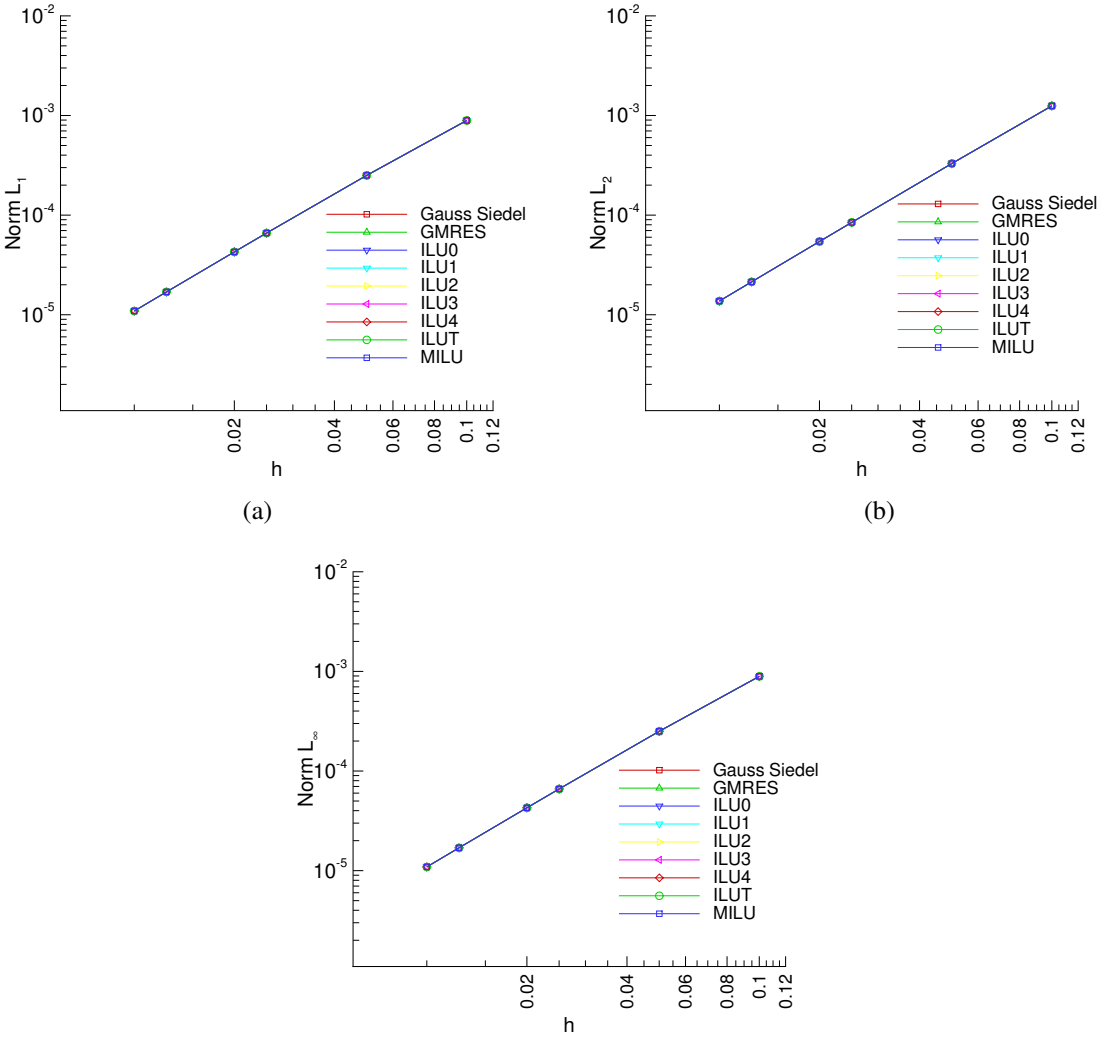
$$p \approx \frac{\log\left(E(h_1)/E(h_2)\right)}{\log\left(h_1/h_2\right)} \tag{27}$$

With this Eq. (27) is possible to estimate the order of accuracy of the numerical method utilized, utilizing the L_2 norm, Fig. 4(b), which presents a better behavior in the Cartesian space, all values of p where around 2, i.e., $p \approx 2$, this represents that there is no other numerical error in the code, once there is only the presence of the truncation error, eliminating the other sources of error such is, programming, iteration and round-off, once this happen is possible to state that the truncation error is the discretization error. This was expected once we truncate the expansion in the Taylor series with the high order terms in function of h^2, which represents a second order accurate scheme.

The second problem analyzed in this work is a transient heat conduction problem, since this is a time dependent problem, we solve one linear system in every time step, this only justified the necessity to choose an efficient solver. The problem below was solved in the literature (Reddy, 2006) so is possible to compare the results. In the literature was utilized both Finite Element Method (FEM) and FDM, the aim here is to compare a fully implicit discretization, that is unconditionally stable with the explicit discretization utilized with FDM approach in the literature, which have the disadvantage of be unstable when the time step is choose to be much large then the maximum eigenvalue of the governing Eq. (4).

In the Fig. 5 the domain presented the expected values, this is compared with the literature in Fig. 6, one characteristic worth mentioning, the time step used to obtain the approximated solution is 100 times greater than the utilized in the explicit discretization, called forward difference scheme, this was only possible by the backward difference scheme here adopted.

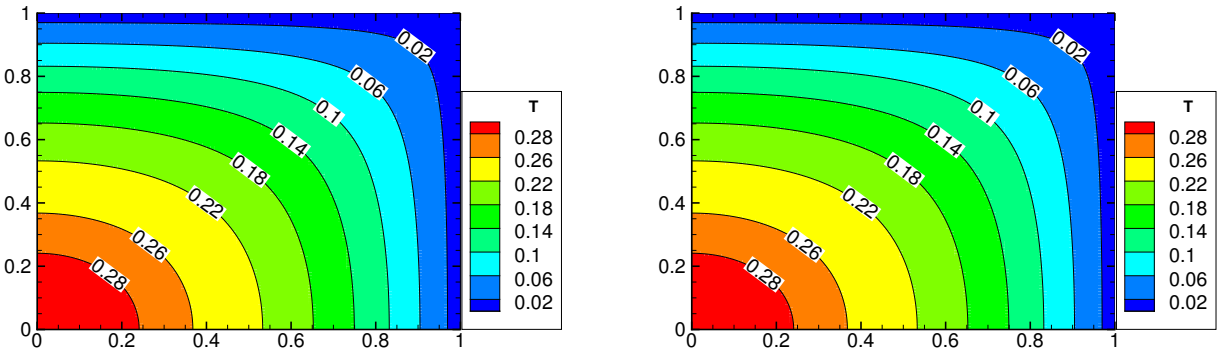
Hence the time step is much greater, is expected that the temperature don't have such a smooth evolving in time, this is easily seen in the graph of Fig. 6 (a), when the temperature have such slow growing when nearness to the steady-state, once in the presented code, such approximation don't occur with such suavity, this occurred due to the time step.



(a) (b)

(c)

Figure 4: Error analysis with: (a) L_1 norm, (b) L_2 norm and (c) L_∞ norm.



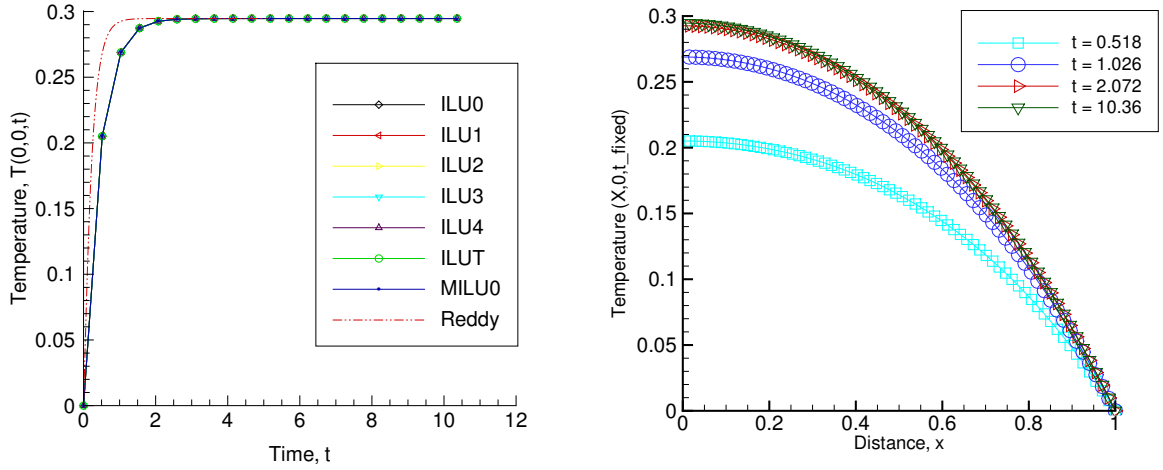
(a) (b)

Figure 5: (a) Isovalues of temperature when reached steady-state condition in a grid with 10000 nodes using GMRES precond with ILU1; (b) Precond with MILU0.

Another consideration is that, in the literature, is presented only four significant digits, in the present work all calculations were made with double precision, so this is four times more accurate. One more thing should be considered, a greater number of the terms in the expansion to obtain the exact solution, or steady-state solution, and yet a grater number of grid

points, which was made in the present work.

Nevertheless both results present similar aspects, but with more precision is expect that it takes more time to converge, or reach the steady-state, since every fluctuation in the less significant digits made to the code to recalculate the solution. Is perceptive in the Fig. 6(b) the between 2 seconds and 10 seconds the solution in the lower part of the domain has small changes.



(a) (b)

Figure 6: (a) Comparison between the two solutions at $(x, y) = 0$; (b) Time evolving in the lower part of the domain.

5. CONCLUSIONS

Solving sparse linear system of differential equations, in some cases, have high computational costs. So the aim to find a iterative method that solve this with high efficiency and still maintain the accuracy of the solution, is the goal of any research. In this work, it was studied the application of GMRES and some preconditioners, mostly of the ILU family to solve highly sparse linear systems of equations. It was possible to see that the preconditioned GMRES have the most time machine efficiency, when compared to other methods, such Gauss-Siedel and the GMRES without preconditioning.

It was notice that the utilization of preconditioned GMRES has a high accuracy, once do not present any kind of degeneration in the solution, even when was applied non-physics parameters in it, this can only prove on that the GMRES algorithm is very stable, and even more powerful when utilized with some kind of preconditioning.

In practice most large sparse system come from the discretization of the physics equations that governing some phenomena, finite difference method is not the exception, it generates one equation to every grid point, and approximate the exact solution only in that point, for this it makes sense in some cases that requires more accurate solutions in some areas, do the refinement of the grid, when this occur, we going to get a great amount of equations, and solve iteratively is a practical choice. It's know that other numerical methods, such FEM and Finite Volume Method (FVM) produces this large linear systems too, worth the effort to study if GMRES present the same accuracy and primarily the efficiency to solve this large systems.

Hence in the present work GMRES have shown to be a good option to solve any numerical application, be a steady-state heat transfer problem, or a transient. This last case is even more advised to use it, once we have to solve a linear system every time step, which demonstrated to be impractical using another solver, when is required to reduce the residual to such order of magnitude. When we unite the GMRES algorithm with the ILU family of preconditioners, the rate of convergence is undoubtedly impressive, once is possible to solve ten thousand linear equations iteratively with less than one hundred iteration obtaining a final residual of $\epsilon \approx 10^{-11}$. For future works, is surely a good choice to study the application of even more ill-conditioned system, or bigger grids.

6. ACKNOWLEDGEMENTS

The autors would like to acknowledge Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico, which provide a financial support under Grant 376/2014.

7. REFERENCES

Blomquist, M., 2019. *Performance Optimization of Numerical Simulations of Heat Transfer and Fluid Flow*. Ph.D. thesis, California State University, Northridge.

Chapman, S., 2007. *Fortran 95/200 for Scientists and Engineers*. McGraw-Hill, 3rd edition.

Chassing, J., Gerolymos, G. and Jérémiasz, J., 2005. "Gmres solution of compressible linearized navier-stokes equations without pseudo-time-marching". *AIAA*.

Content, C., Outtier, P.Y. and Cinnella, P., 2013. "Coupled/uncoupled solutions of rans equations using a jacobian-free newton-krylov method". *AIAA*.

Fossati, M., 2015. "Development of a jacobian-free finite element solver for aerothermodynamic design". *AIAA*.

Golub, G.H. and Van Loan, C.F., 2013. *Matrix Computations*. Johns Hopkins University Press, 4th edition.

Han, Y., Nie, Y. and Dong, H., 2018. "A fast multipole algorithm for radiative heat transfer in 3d semitransparent media". *Journal of Quantitative Spectroscopy and Radiative Transfer*, Vol. 221, pp. 8–17.

Jahedi, M., Berntsson, F., Wren, J. and Moshfegh, B., 2018. "Transient inverse heat conduction problem of quenching a hollow cylinder by one row of water jets". *International Journal of Heat and Mass Transfer*, Vol. 117, pp. 748–756.

Kreith, F., Manglik, R. and Bohn, M., 2011. *Principles of Heat Tranfer*. Cengage Learning, Stamford, 7th edition.

LeVeque, R., 2007. *Finite difference methods for ordinary and partial differential equations: steady and time-dependent problems*. Society for Industrial and Applied Mathematics, Philadelphia.

Moon, J., Pacheco, J.R. and Pacheco-Vega, A., 2018. "Heat transfer enhancement in wavy micro-channels through multiharmonic surfaces". In *ASME 2018 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, pp. V08BT10A004–V08BT10A004.

Nejat, A. and Ollivier-Gooch, C., 2005. "A high-order unstructured gmres algorithm for inviscid compressile flows". *AIAA Computational Fluid Dynamics Conference*, Vol. 17.

Reddy, J., 2006. *An Introduction to the Finite Element Method*. McGraw-Hill, Texas, 3rd edition.

Saad, Y. and Schultz, M., 1986. "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM J. Sci. Stat. Comput.*, Vol. 7.

Saad, Y., 2003. *Iterative methods for sparse linear systems*. siam, 2nd edition.

Wigton, L., Yu, N. and Young, D., 1985. "Gmres acceleration of computational fluid dynamics codes". *AIAA Journal*.

Zahr, M. and Persson, P., 2013. "Performance tuning of newton-gmres methods for discontinuous galerkin discretizations of the navier-stokes equations". *AIAA*.

8. RESPONSIBILITY NOTICE

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