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

COB-2019-0907
THREE-DIMENSIONAL NUMERICAL SOLUTION BASED ON GMRES ALGORITHM APPLIED TO HEAT SINKS

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

Igor Santana Sousa
Hacker Industrial, Xanxere, Santa Catarina, Brazil
ssigor.tm@gmail.com

Abstract. *Engineering problems involving heat transfer process appears very often, and the analytical solutions of this problems can be very complex, or in some cases, unsolvable. However, numerical methods can be used to solve problems like these. This work presents a numerical study of heat transfer applied to heat sinks. To execute these simulations was implemented a code based on finite difference method to discretize the three-dimensional heat diffusion equation. The system of linear equations is solved by the Generalized Minimal Residual Method (GMRES) with preconditioner. The first results are focused on verifying and validating the code using physical problems that contain analytical solutions. To demonstrate the robustness of the code, several geometries of heat sinks was simulated considering some thermal conditions. The thermal behavior of the heat sinks are presented, as well as the efficiency of the GMRES to solve problems related to heat transfer phenomena.*

Keywords: *Finite Differece Method, Implicit Method, GMRES, Thermal Analysis, Heat Sinks.*

1. INTRODUCTION

Heat transfer is a common problem in the field of engineering, from work generation to data processing. Even with the technology development and growing presence of electronic components, heat transfer stills a important aspect to be studied on information processing capacity. Basically, problems involving heat transfer remains on a domain, with one region with more energy and another with less energy, so a temperature gradient is between these regions, and that gradient constitutes the solution.

A heat transfer problem is solved by a differential equation, which the results are the temperature gradient on the solution domain, and this domain can became very complex, on real cases for instance. In some cases those solutions are difficult to solve analytically (Incropera *et al.*, 2007), so there is two ways to solve the problem: one is making simplifications of the problem until the equation can be solved by analytical methods, and this option leads to a physical condition mischaracterization, i.e. the solution does not represent the true domain state.

The other way is to divide the continuum domain in a discrete domain by numerical methods and find the solution on each of this subdivisions, so the numerical solution shows itself as a option to solve efficiently and accurately this kind of problem (LeVeque, 2007). Since more publicated studies of numerical analysis has been growing in the recent years (Content *et al.*, 2013), it's important to find methods to do these approaches with less costs. With the computational power exponentially increasing simplified problems are no longer needed, being possible to analyze complex domains these days (Fossati *et al.*, 2015).

The progress of Computational Fluid Dynamics (CFD) lead to larger linear systems to solve, Nejat and Ollivier-Gooch (2008) and Chassaing *et al.* (2006) reported the need to solve large linear systems simultaneously, due to more complex analyzes of their problem. This leads to a greater computational cost and more time to convergence of the solution. The use of the GMRES algorithm delivers the expected results when the goal is speed of convergence (Wigton *et al.*, 1985).

The linear system provided by the discretization has a large number of terms, and could be solved with numeric methods based on matrix decomposition, such that Gaussian Elimination, LU and Cholesky, or by iterative methods as Gauss-Seidel and Conjugate-Gradient, (Golub and Van Loan, 2013). But the generated linear system has a charateristic to be sparse, so a Krylov subspace based method such as GMRES are recommended to this case, (Saad and Schultz, 1986). Once the solution is obtained, is necessary to confirm if the obtained results represent the physical condition. So a

verification/validation of the algorithm is necessary to prove the accuracy of the solution.

To solve large linear systems the computational effort is enormous being necessary a powerful computer to execute the task. In this situation the microprocessor (CPU) heats up very fast and this heat charge must be dissipated efficiently to preserve the normal patterns of the system operation. To dissipate the heat of the CPU one uses, commonly, a passive heat exchanger (or heat sink) together with an exhaust system. The objective of the heat sink is conducts the heat away of the CPU surfaces, that is dissipated to the medium by the exhaustor, (Incropera *et al.*, 2007). We have two ways to understand the behavior of this type of heat sink, experimental analysis or numerical simulation. The first one demand investments in instrumentation besides the difficulty in correctly instrumentating the system. The second is more simple and offers a wide range oportunite to analyse the more severals characteristics of the heat sink. In this case, the numerical solution is a good choice to execute the analysis these type of heat sink and characterizes a good choice to apply a numerical strategy involving sparse linear systems. As well as, is an engineering problem of great importance that to be considered in the hardware developement, (Ayala *et al.*, 2010).

The purpose of this work is the development, verification and validation of a computational algorithm to simulate three-dimensional steady and transient heat transfer problems through the discretization of the domain via finite difference method. After verification and validation steps the code was used to simulate the important application involving heat sinks. In this case, the behavior of the temperature profile is the interest solution. In all cases presented in this work, the solution of the system of linear equations generated by the discretization is solved through the GMRES iterative method with MILU type preconditioner.

2. MATHEMATICAL MODEL

2.1 Heat Transfer

The heat transfer phenomena can be modeled by the heat diffusion equation, which describes the behavior of the heat transfer on the system. Let us consider an infinitesimal part of the system, i.e., the control volume presented in Fig. 1.

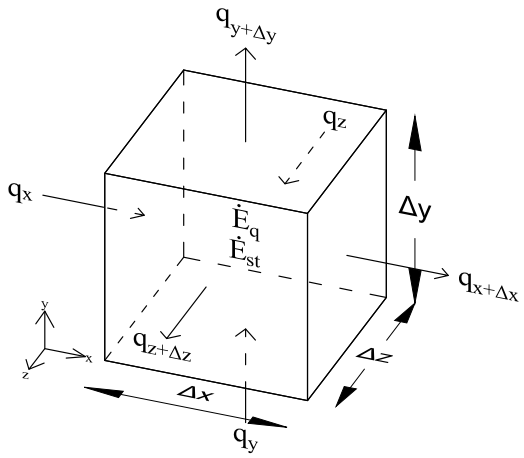


Figure 1. Heat conduction in a control volume.

In Fig. 1, q is the conduction heat rates through the faces of the control volume, $\dot{E}_g$ is the thermal energy genetation and $\dot{E}_{st}$ is the internal thermal energy stored in the control volume.

The heat conduction equation considering a homogeneous medium, without bulk motion, in Cartesian coordinates can be otained applying the conservartion of energy on a differential control volume, Fig. 1,

$$\frac{1}{\alpha}\frac{\partial T}{\partial t} - \frac{\dot{q}}{k} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}. \tag{1}$$

where α is the thermal diffusivity, T is the temperature, t is the time, $\dot{q}$ is a source term related to the energy generation per unit volume, k is the thermal conductivity, x, y, and z are the coordinates on the Cartesian field.

3. COMPUTATIONAL PROCEDURE

The domain was discretized by finite difference method, with consists on solution of diferential equations by the substitution of derivatives by algebric equations, which this equations are solved on the nodes of the mesh. This method allows a good understanding of the physical problems adequated to be discretized with structured meshes (Mathews *et al.* (2004) and Jajja *et al.* (2014)). The solution is obtained by solving a linear system, which is solved in this work by a

iterative method called Generalized Minimal Residual (GMRES) with the MILU preconditioner (Saad, 2003) and (Washio and Hayami, 1994). These methods were implemented using the Fortran 95/2003 programming language (Chapman, 2008).

3.1 Discretization of the physical problem

The Eq. (1) can be written as follows:

$$\nabla^2 T = \frac{1}{\alpha}\frac{\partial T}{\partial t} - \frac{\dot{q}}{k} \tag{2}$$

With:

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \tag{3}$$

The Laplacian on Eq. (2) and Eq. (3) can be written by a function of 7 (Eq. (4)), 19 (Eq. (5)) and 27 nodes (Eq. (6)), and a node scheme used can be shown on Fig. 2 (O'Reilly and Beck, 2006):

$$\nabla^2 T_i = \frac{1}{l^2}\sum_{j\in N_f}(T_j - T_i) \tag{4}$$

$$\nabla^2 T_i = \frac{1}{6l^2}\left(2\sum_{j\in N_f} T_j + \sum_{j\in N_a} T_j - 24T_i\right) \tag{5}$$

$$\nabla^2 T_i = \frac{3}{13l^2}\left(\sum_{j\in N_f} T_j + \frac{1}{2}\sum_{j\in N_a} T_j + \frac{1}{3}\sum_{j\in N_v} T_j - \frac{44}{3}T_i\right) \tag{6}$$

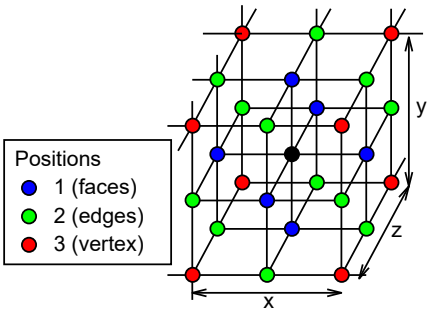


Figure 2. Node scheme according to their position. Avaliable at: O'Reilly and Beck (2006)

After the domain discretization and select the equations used on each case, the global linear system - Eq. 7 - is mounted to be solved , once if is a stationary problem or each time step if is a transient problem.

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \dots & \alpha_{2n} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \dots & \alpha_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \alpha_{n3} & \dots & \alpha_{nn} \end{bmatrix} \times \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_n \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_n \end{pmatrix} \tag{7}$$

Where α and β are real values calculated as function of the nodal scheme to be considered in the discretization [Eqs. (4), (5) or (6)] and T is the nodal temperature to be obtained.

3.2 Finite Difference Method

As a partial differential equation, the heat diffusion equation can be solved by a numerical method like the finite difference method, who consists in take the continuous problem and separate it - discretize it, and the variables are solved on the discrete points - nodes of the mesh. (Pletcher *et al.*, 2012).

The Taylor-series expansion is used to discretize the differential equations. It calculates the assumed value of the function on the interest point with the function values and the derivatives on other nodes, (LeVeque, 2007). The assumed

precision to the expansion is their number of terms. For this study, the derivative values are obtained within the points of interest (central-difference representation), with the first derivative approximation and a second-order precision truncation error (Pletcher *et al.*, 2012).

$$f'(x_i) = \frac{f(x_{i+1}) + f(x_{i-1})}{2l} + O(l^2) \tag{8}$$

$$O(l^2) = -\frac{f'''(\xi_1)}{3!}l^2 - \frac{f'''(\xi_2)}{3!}l^2 \tag{9}$$

where:
ξ_1 is a value of x between x_i and x_{i+1};
ξ_2 is a heat of x between x_{i-1} and x_i.

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

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

According to the work (Saad and Schultz, 1986), the GMRES is a method based on the Eq. (10):

$$\mathcal{K} = \mathcal{K}_m \tag{10}$$

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. So, 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 form of the linear system to be solved is given on the Eq. (11)

$$Ax = b \tag{11}$$

To solve a linear system given on the Eq. (11) 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. (12)):

$$r_0 = b - Ax_0 \tag{12}$$

The execution of k iterations of the Arnoldi Algorithm on a initial vector v_1 = $r_0/\|r_0\|$ 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 aims to approximate the solution on form $x_0 + z$ minimizing the residual norm about z on $\mathcal{K}_k$. After k iterations of the Arnoldi algorithm is obtained 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. Thus, the difference between $\overline{H}_k$ and H_k is the line that contains the $h_(k + 1, k)$ on $k + 1, k)$ position. So the v_i vectors and $\overline{H}_k$ matrix satisfies (Eq. (13)):

$$AV_k = V_{k+1}\overline{H}_k \tag{13}$$

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

$$\min_{z \in \mathcal{K}_k} \|b - A[x_0 + z]\| = \min_{z \in \mathcal{K}_k} \|r_0 - Az\| \tag{14}$$

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

$$J(y) = \|\beta v_1 - A V_k y\| \tag{15}$$

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

$$J(y) = \|V_{k+1}[\beta e_1 - \overline{H}_k y]\| \tag{16}$$

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

$$J(y) = \|\beta e_1 - \overline{H}_k y\| \tag{17}$$

So the solution will be given as follows:

$$x_k = x_0 + V_k y_k \tag{18}$$

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

Pre-conditioning of the linear system leads to a more efficient computational work and improving robustness to the method, once the weakness of iterative methods to solve linear systems is the lack of robustness. Calculating a Gaussian elimination on an $[A]$ matrix and dropping some elements in predetermined nondiagonal positions can be done to pre-conditionate a Matrix (called Incomplete LU factorization - ILU). 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

4.1 Steady-State case with exact solution

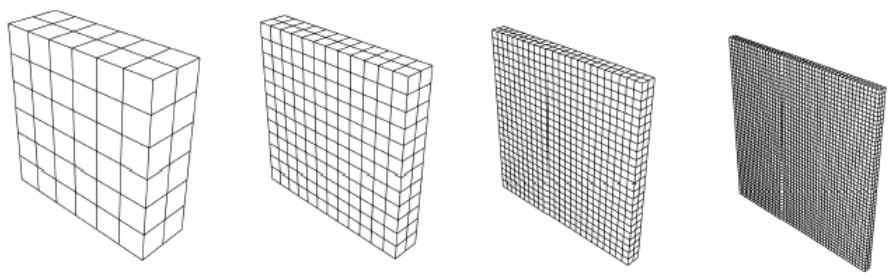
The first case is represented by a plane square plate on XY plane with L=1m, with negligible thick and without heat transfer on perpendicular direction and his borders. The boundary conditions is expressed in Eq. (19) and Eq. (20). The exact solution is expressed by Eq. (21).

$$T(0, y, z) = T(L, y, z) = T(x, 0, z) = 0 \tag{19}$$

$$T(x, L, z) = sen\left(\frac{\pi x}{L}\right) \tag{20}$$

$$T(x, y, z) = \frac{senh(\pi y/L)}{senh(\pi)} sen(\pi x/L) \tag{21}$$

The used meshes in simulation of this case is shown in Fig. 3, and the convergence rates in L_1, L_2 and L_∞ norms is shown in Fig. 4.



(a) (b) (c) (d)

Figure 3. Meshes used in first case. (a) 147 nodes, (b) 507 nodes,(c) 1875 nodes, (d) 7203 nodes.

The values in the tables 1, 2 and 3 shows a $2nd$-order convergence for the Laplacian with 7 and 27-nodes, and a convergence of $6th$-order for the Laplacian with 19-nodes. For the last one, the error to the exact solution was of the order 10^{-11}. The Fig. 5 shows the difference of the exact to the numeric solution in each Laplacian, which can be noted the high convergence order on the 19-nodes Laplacian.

Table 1. Norm values and angular coefficients for the 7-nodes Laplacian.

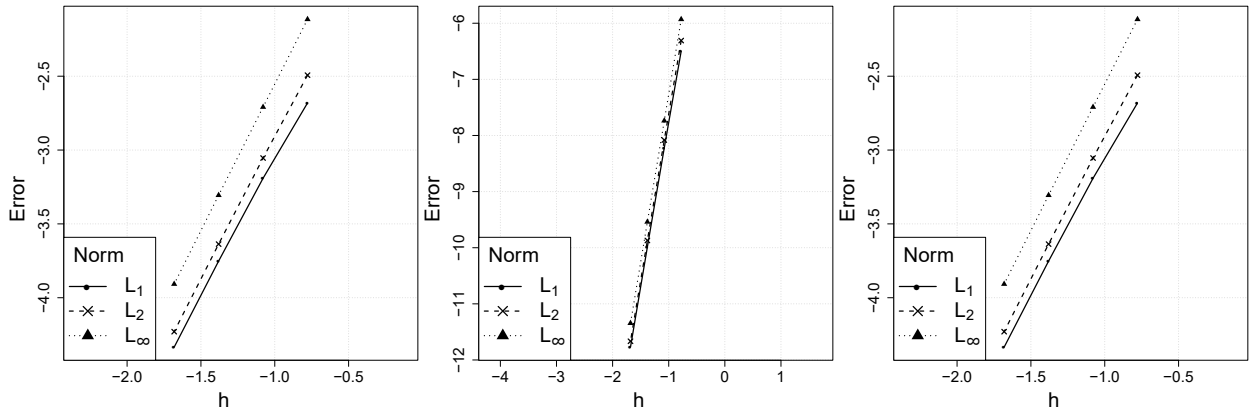
node division	L_1	$p(L_1)$	L_2	$p(L_2)$	L_∞	$p(L_\infty)$
L/6	-2,6812	-	-2,4931	-	-2,1148	-
L/12	-3,1909	1,6934	-3,0540	1,8635	-2,7090	1,9741
L/24	-3,7518	1,8635	-3,6370	1,9369	-3,3060	1,9834
L/48	-4,3346	1,9362	-4,2298	1,9694	-3,9075	1,9983

Table 2. Norm values and angular coefficients for the 19-nodes Laplacian.

node division	L_1	$p(L_1)$	L_2	$p(L_2)$	L_∞	$p(L_\infty)$
L/6	-6,4976	-	-6,3091	-	-5,9305	-
L/12	-8,2189	5,7186	-8,0820	5,8900	-7,7368	6,0010
L/24	-9,9858	5,8701	-9,8710	5,9435	-9,5399	5,9904
L/48	-11,771	5,9309	-11,667	5,9668	-11,3449	5,9967

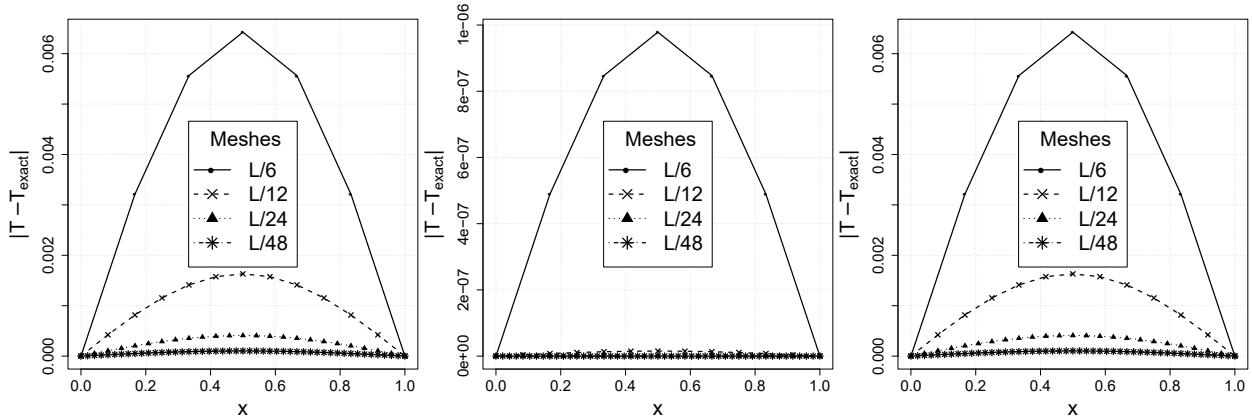
Table 3. Norm values and angular coefficients for the 27-nodes Laplacian.

node division	L_1	$p(L_1)$	L_2	$p(L_2)$	L_∞	$p(L_\infty)$
L/6	-2,8759	-	-2,6871	-	-2,3082	-
L/12	-3,3976	1,7332	-3,2606	1,9053	-2,9154	2,0173
L/24	-3,9616	1,8738	-3,8468	1,9475	-3,5157	1,9944
L/48	-4,5452	1,9389	-4,4404	1,9721	-4,1181	2,0013



(a) (b) (c)

Figure 4. Error as a function of h. Nodes by Laplacian: (a) 7, (b) 19 and (c) 27. Steady solution.



(a) (b) (c)

Figure 5. Difference between numeric and exact solution by each mesh, using the three-Laplacians on line y = $0,5\ L_y$ and z = $0,5\ L_z$. Steady solution.

4.2 Transient case with exact solution

The second case (transient) is a cube with L = 1 m, boundary conditions is expressed in Eq. (22), initial temperature $T(x, y, z, 0) = 1$ and exact solution expressed by Eq. (23).

$$T(0, y, z, t) = T(L, y, z, t) = T(x, 0, z, t) = T(x, L, z, t) = 0 \tag{22}$$

$$T(x, y, z, t) \ = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{4}{mn\pi^2}\{[(-1)^m - 1][(-1)^n - 1]\} sen\left(\frac{n\pi x}{L}\right) sen\left(\frac{m\pi y}{L}\right) exp\left(-\frac{k\pi^2 t}{L^2}(m^2 + n^2)\right) \tag{23}$$

The used meshes in this case is shown in Fig. 3 (same geometry of the Steady-State case), and the heat transfer through the model is shown in Fig. 6.

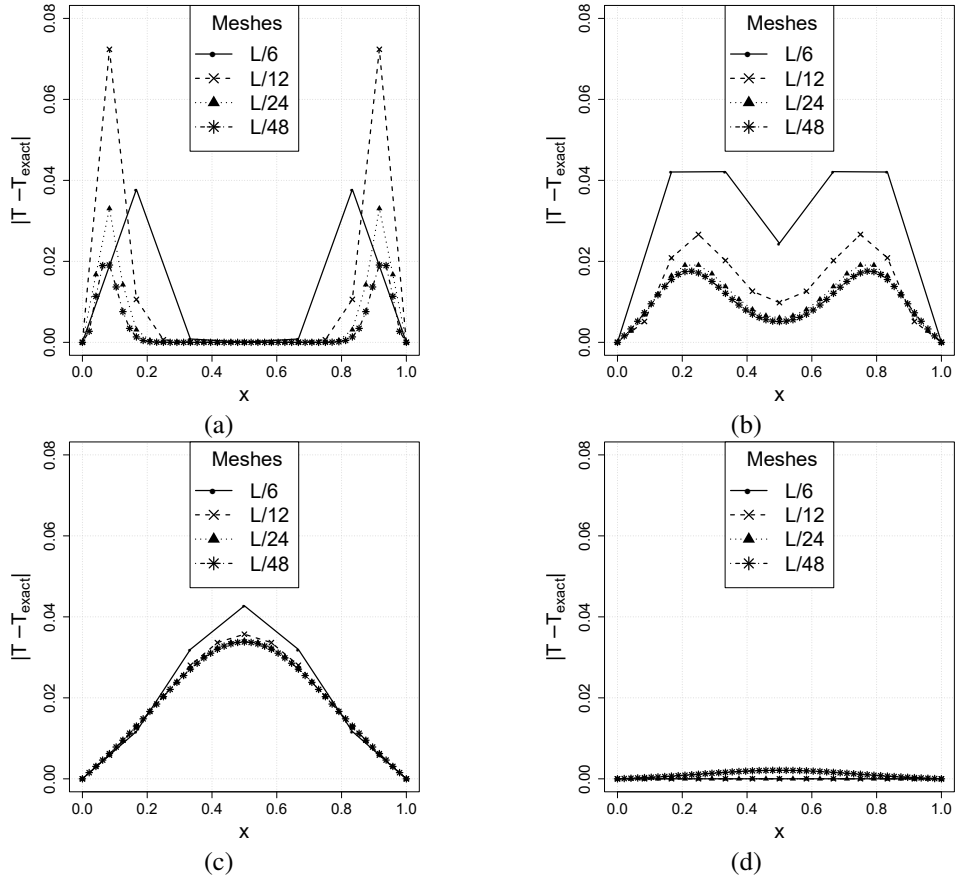


Figure 6. Difference between numeric and exact solution in t = 0.001 s, t = 0.01 s, t = 0.05 s and t = 0.1 s.

The figure 6 shows the obtained results. It can be noted a difference between the exact and numeric solution on the first time and a convergence as the time passes. The problem doesn't approaches the exact solution due to the millions of terms of the rounding error summation value .

4.3 Applications: Heat Sinks

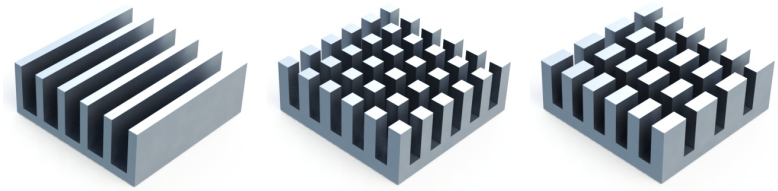
After the verification and validation of the algorithm, it can be used on practical applications, such as heat sinks, which as for example, are used to refrigerate computers CPUs. The heat sinks employed in this simulation are used in CPUs of the *Raspberry Pi* computers, which are low cost with reduced geometries without a forced mechanism of heat dissipation, (Pi, 2019). Will be studied the heat transfer through three different geometries of heat sinks, shown in Fig. 7. These heat sinks has similar dimensions, Tab. 4, to allow comparison among them. In Fig. 8 are presented the major boundary conditions applied in each heat sink in this simulation. The Table 4 express the geometric characteristics of the heat sinks to be studied, where N is the number of fins, h_t is the total height of the geometry, h_a is the height of the fins, A_1 is the primitive area of heat transfer by convection, A_2 is the area submitted to the prescribed heat flux, A_3 is the extended area due to the fins and A_4 is the total area of the geometry that is subjected to convective heat transfer.

In this simulation, the prescribed heat flux is 5 W, the convective heat transfer coefficient is 27×10^{-4} W/mm^2 $\cdot$ K, the thermal conductivity is $2,04$ W/mm $\cdot$ K, the thermal diffusivity is $9,88 \times 10^{-7}$ mm^2/s, the initial and ambient

Table 4. Geometric characteristics of the heat sinks.

Heat sink	N	h_t(mm)	h_a(mm)	A_1(mm^2)	A_2(mm^2)	A_3(mm^2)	A_4(mm^2)
1	3	5,25	4,38	20,00	12,25	195,2	215,2
2	9	5,09	3,82	19,57	12,25	174,9	194,5
3	6	5,09	3,82	19,57	12,25	145,8	165,4

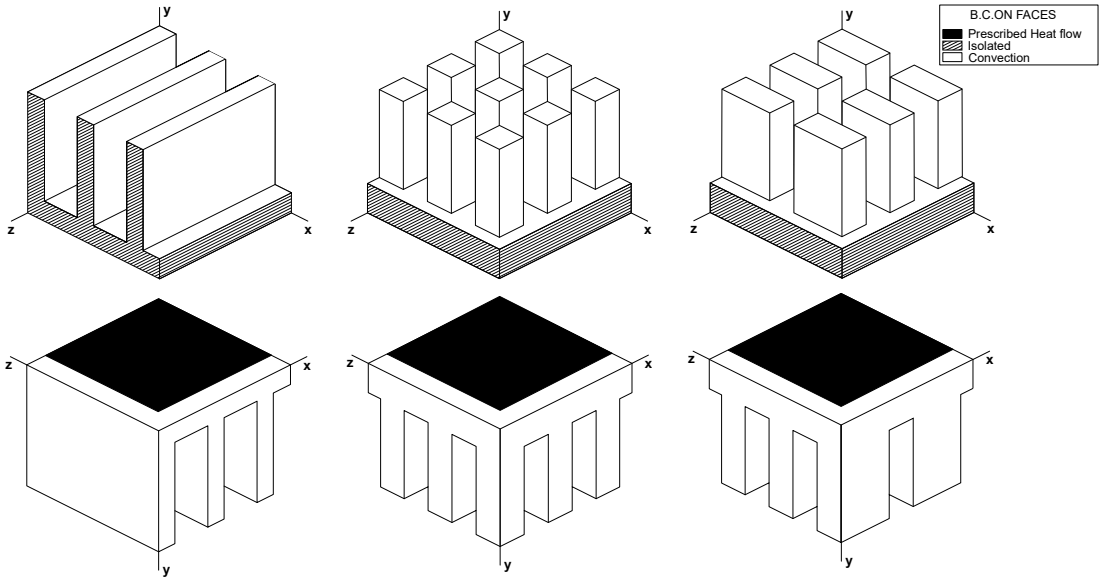
temperatures are equal,respectively, 300 K. The tolerance for the convergence of the numerical solution was set to 10^{-6}, with the time step equal to 10 seconds. The meshes generated for each heat sink has the following numbers of nodes: heat sink 1 (2397 nodes), heat sink 2 (5397 nodes) and heat sink 3 (5885 nodes).



(a) Heat sink 1 (b) Heat sink 2 (c) Heat sink 3

Figure 7. Heat sink models.

In the three heat sinks of this simulation, the prescribed heat fluxes and the areas subjected to these flux are similar (square base), the results presented for this simulation considers only a quarter of the original geometry (simmetry), Fig. 9.



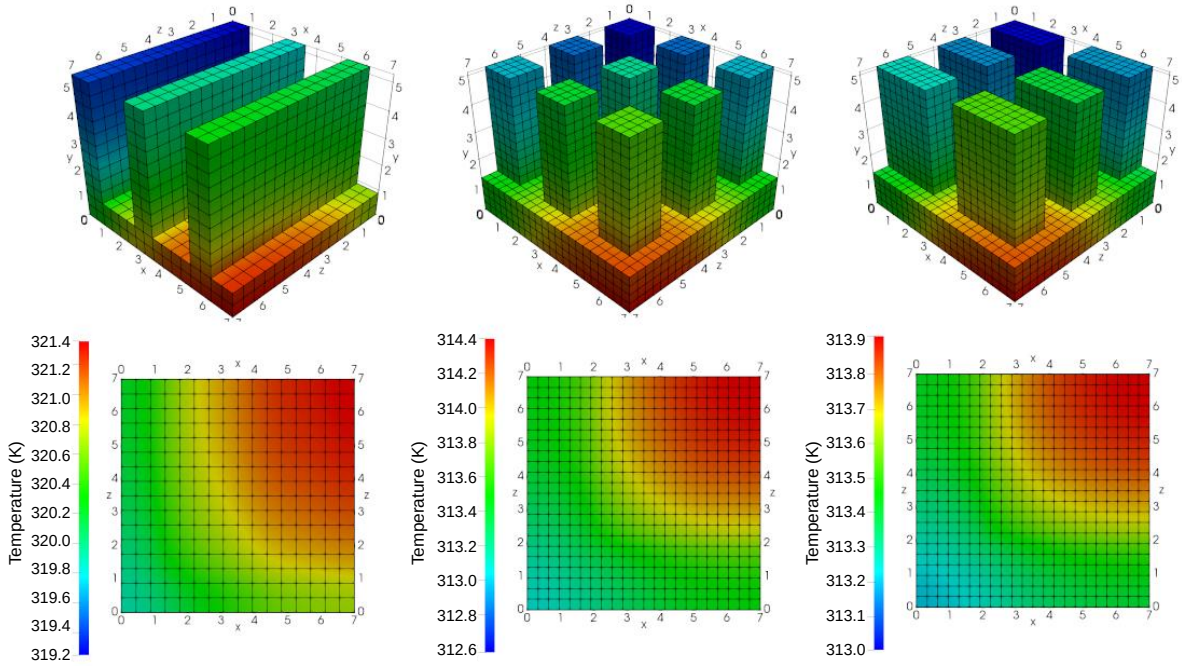
(a) Heat sink 1 (b) Heat sink 2 (c) Heat sink 3

Figure 8. Heat sinks 1, 2 and 3 - Boundary conditions.

In Fig. 9 one can observe the temperature distribution over the three heat sinks. The prescribed heat flux was set on center of the square base of each heat sink. The temperature propagates along the base of each heat sink, from the center to the ends of the base in homogeneous fashion. All superior faces of the heat sinks are subjected to the convective boundary condition, which imposes a dissipation of the heat from base to top of the heat sinks.

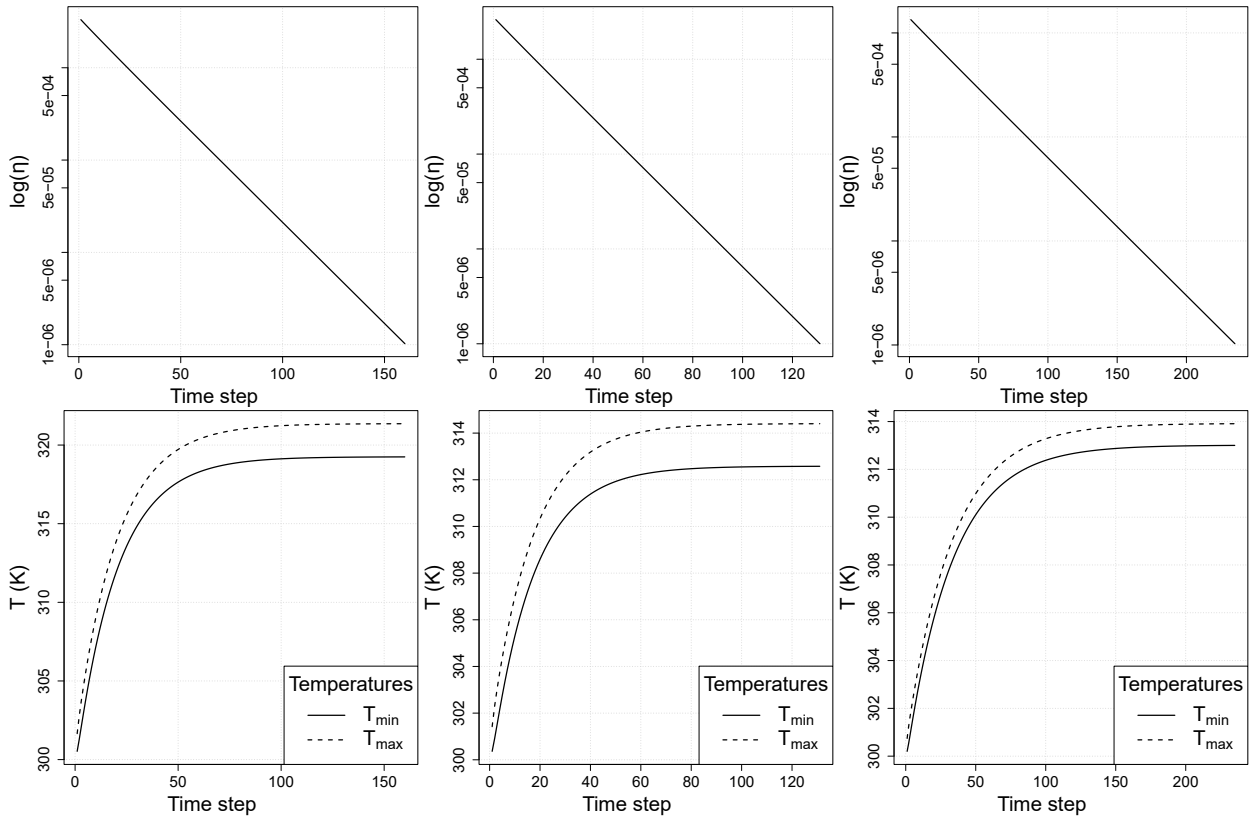
The heat conduction and thermal diffusivity are the same for all heat sinks, necessary to compare them as to their effectiveness in thermal dissipation. However, in the heat transfer simulated over this type of heat exchanger, the geometric arrangement of the fins is the main characteristic for the effectiveness of heat dissipation. This effect can be seen in the temperature distribution, Fig. 9, as well as the maximum and minimum temperatures, Fig. 10, over each heat sinks obtained in this simulation for each geometric arrangement.

Results presented in Figs. 9 and 10 indicates that the heat sink 3 is more effective in heat dissipation process. Prolonged rectangular fins, with spaces defined between them, optimized the effectiveness of the convection process in the arrangement. In heat sink 3, the maximum temperature does not exceed 314 K, while in the heat sinks 1 and 2, this temperature exceeded 314 K. It is important to note that, in heat sink 3 we have the smallest difference between maximum and minimum temperatures, that is very desirable for these applications.



(a) Heat sink 1 (b) Heat sink 2 (c) Heat sink 3

Figure 9. Heat sinks 1, 2 and 3 - Temperature Distribution.



(a) Heat sink 1 (b) Heat sink 2 (c) Heat sink 3

Figure 10. Heat sinks 1, 2 and 3 - Historic of convergence of the simulations and range of temperatures in the geometries over the time steps.

In the Fig. 10, one has the historic of convergence of the numerical simulations. The convergence for all simulations reaches the prescribed value for the stop criterium, 10^{-6}, in no more than 250 iteractions. In all simulations, GMRES solver preconditioned by MILU provide convergence of the numerical solution without oscillations.

5. CONCLUSIONS

In this work was presented a general discretization based on finite difference method applied to the heat diffusion equation. The main goal is an implementation of a solver that can be used to simulate engineering problems, in principle, to solve heat transfer applications.

The code was verified, validated and used to simulate an important engineering problem, namely, heat sinks applied to microprocessors, in this case, that ones specifically used in the *Raspberry Pi* computers. In the verification analysis, the solver provide excellent convergence rates and reaches the spatial discretization order linearly for all Laplacians used in this case. Important attention to the 19-nodes Laplacian, that presented a spectral convergence for a $6th$-order discretization. In the validation process, the numerical solution reaches the exact solution with excellent agreement. When the code was used to simulate the heat sink application, the same was observed, i.e., the behavior of the results show that the code provide accurate numerical results, mainly with physical agreement. Besides that, the GMRES with MILU preconditioner, reaches the convergence without oscillations, presenting high numerical stability in all numerical tests showed in this work. Those results shows the efficiency of the algorithm to solve problems related with heat transfer problems in engineering.

6. ACKNOWLEDGEMENTS

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

7. REFERENCES

Ayala, J.L., Sridhar, A. and Cuesta, D., 2010. "Thermal modeling and analysis of 3D multi-processor chips". *Integration*, Vol. 43, No. 4, pp. 327–341.

Chapman, S.J., 2008. *Fortran 95/2003 for scientists and engineers*. McGraw-Hill, 3rd edition. ISBN 0073191574.

Chassaing, J., Gerolymos, G. and Jeremiasz, J., 2006. "Gmres solution of compressible linearized navier-stokes equations without pseudo-time-marching". *AIAA Paper*, Vol. 688, p. 2006.

Content, C., Outtier, P.Y. and Cinnella, P., 2013. "Coupled/uncoupled solutions of rans equations using a jacobian-free newton-krylov method". In *21st AIAA Computational Fluid Dynamics Conference*. p. 2423.

Fossati, M., Habashi, W.G., Gao, S., Yin, P., Isola, D., Baruzzi, G. and Ozcer, I., 2015. "Development of a jacobian-free finite element solver for aerothermodynamic design". In *53rd AIAA Aerospace Sciences Meeting*. p. 1261.

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

Incropera, F.P., Lavine, A.S., Bergman, T.L. and DeWitt, D.P., 2007. *Fundamentals of heat and mass transfer*. Wiley, 6th edition.

Jajja, S.A., Ali, W., Ali, H.M. and Ali, A.M., 2014. "Water cooled minichannel heat sinks for microprocessor cooling: Effect of fin spacing". *Applied Thermal Engineering*, Vol. 64, No. 1-2, pp. 76–82.

LeVeque, R., 2007. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (Classics in Applied Mathematics Classics in Applied Mathemat)*, Vol. 51. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. ISBN 0898716292, 9780898716290.

Mathews, J.H., Fink, K.D. *et al.*, 2004. *Numerical methods using MATLAB*, Vol. 3. Pearson Prentice Hall Upper Saddle River, NJ, 4th edition.

Nejat, A. and Ollivier-Gooch, C., 2008. "A high-order accurate unstructured finite volume newton–krylov algorithm for inviscid compressible flows". *Journal of Computational Physics*, Vol. 227, No. 4, pp. 2582–2609.

O'Reilly, R.C. and Beck, J.M., 2006. "A family of large-stencil discrete laplacian approximations in three dimensions". *International Journal for Numerical Methods in Engineering*, Vol. 0, No. 2, pp. 1–16.

Pi, R., 2019. *Raspberry Pi*. URL `https://www.raspberrypi.org/`.

Pletcher, R.H., Tannehill, J.C. and Anderson, D.A., 2012. *Computational fluid mechanics and heat transfer*. Taylor Francis Group, 3rd edition.

Saad, Y. and Schultz, M.H., 1986. "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM J. Sci. Stat. Comput.*, Vol. 7, No. 3, pp. 856–869. ISSN 0196-5204. doi:10.1137/0907058.

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

Washio, T. and Hayami, K., 1994. "Parallel block preconditioning based on ssor and milu". *Numerical linear algebra with applications*, Vol. 1, No. 6, pp. 533–553.

Wigton, L., Yu, N. and Young, D., 1985. "Gmres acceleration of computational fluid dynamics codes". In *7th Computational Physics Conference*. p. 1494.

8. RESPONSIBILITY NOTICE

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