UDC 519.62

G. Domínguez Rodríguez, J. A. Medina García, J. A. Tapia González,
O. Bilyy, G. I. Canto Santana✉

TIME-EFFICIENT SPARSE MATRIX COMPUTATION USING MESH RESAMPLING IN FINITE DIFFERENCE ANALYSIS

A methodology is proposed to reduce the computational time required for finite differences. It implies the gradual increment of the number of nodes at each side of a rectangular representative volume and the interpolation of the guess values from the previous nodal solution. This process continues until the final number of nodes is reached. Three solution methods are used along with three different types of boundary conditions, including a discontinuous one, to evaluate the efficiency of the suggested methodology. A reduction of five orders of magnitude in computational time was registered for both Jacobi and Gauss–Seidel algorithms. Whereas, its decrement was of one order of magnitude for the conjugate-gradient method.

***Key words*:** *finite difference method, mesh resampling, Jacobi algorithm, Gauss–Seidel algorithm, conjugate gradient.*

Introduction. Physical and biological phenomena, such as atmospheric movements, electromagnetic fields, tides, earthquakes, protein interactions, and absorption of molecules in the human body, are commonly described as continuous systems containing an infinite number of degrees of freedom [16]. These continuous systems can be modelled through partial differential equations delimited by initial values and boundary conditions. While analytical solutions for such systems are not possible in most cases, numerical and computational methods can be used for analysis. Among those methods, the Finite Difference method is widely used to solve the aforementioned differential equations [1, 2, 9, 21]. This method approximates the continuous partial derivatives to finite ones [12, 14]. Consequently, systems of linear equations are formed for those finite derivatives, and several methods are used to solve the sparse matrix produced, e.g., Jacobi, Gauss–Seidel, and conjugate-gradient methods [3, 4].

The Jacobi and Gauss–Seidel methods are two of the earliest iterative processes to solve sparse matrices [5, 19, 20, 27, 29, 30, 31: Chap. 7]. The main difference between them is that the Jacobi method only uses the current values to produce the next step, while the Gauss–Seidel method always applies the latest updated values during each iteration [10: Chap. 20, 24: Chap. 10]. The conjugate-gradient method is an iterative algorithm aimed at numerically solving linear systems of equations, whose matrices are both symmetric and positive definite [6, 8, 15, 22: Chap. 22, 25, 26, 28, 13]. All three methods can be applied to sparse systems that are too large to be solved by direct methods such as Cholesky decomposition [15, 13].

It is widely known that the proper guess for the values at the first iteration effects the efficiency of the computational time of iterative methods [6, 10: Chap. 20, 22: Chap. 22, 24: Chap. 10, 31: Chap. 7]. Therefore, the methodology proposed in this paper consists of “resampling” the initial guess through the interpolation from a previous solution with a lower number of nodes. Because it is a linear system associated with a smaller sparse matrix, its computing time is drastically lower. If the process is repeated until the desired system dimensions are reached, the cumulative computing time for all instances using this mesh resampling methodology is expected to be lower than the total time of a single instance conducted with the same number of nodes.

✉ gcanto@uacam.mx

DOI: 10.15407/mmpmf2024.67.3-4.53-63

Yet another alternative is adaptive re-meshing. However, it conveys an additional computational burden similar to that of the sparse matrix solving process [11, 17, 18]. Regular meshing requires a more straightforward routine to get a distribution of nodes for optimal parallelization.

Finally, different resampling scaling factors are studied for each of the considered methods, i.e., Jacobi, Gauss–Seidel, and conjugate gradient.

1. Materials and Methods. In this work, an iterative methodology, named *mesh resampling*, is proposed to speed up finite differences' calculations based on the gradual increment on the number of nodes at a representative volume element (RVE). A brief description of this methodology is illustrated in Fig. 1. Herein, the first guess of a current solution method is obtained through a linear interpolation from the previous solution. Therefore, the convergence time is expected to be reduced compared to a similar computation. This reduction is due to the similarities between the current guess and the converged solution. The proposed methodology's main goal consists of achieving a total computing time lower than the computing time of a single calculation when the number of nodes and other conditions are kept identical.

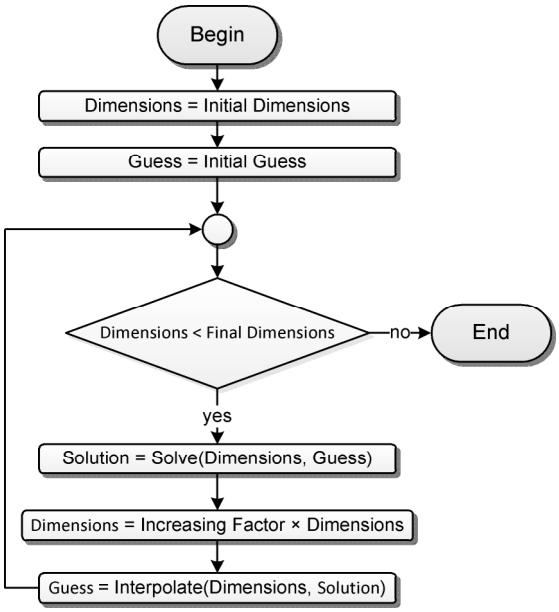


Fig. 1. Flowchart of the proposed algorithm for the resampling methodology.

In order to properly assess the proposed mesh resampling methodology's performance on the finite differences, we consider a widely used differential equation governing the steady-state heat flow [7, 23]. This equation can be modelled through the Laplace equation in the form as follows:

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0\,, \tag{1}$$

where $T(x, y)$ is the temperature, and x and y are the independent variables for 2D positions.

The second-order derivatives can be linearly approximated by the second-order central finite differences [12, 14], defined as

$$\frac{\partial^2 T}{\partial x^2} = \frac{T(x+h, y) - 2T(x, y) + T(x-h, y)}{h^2} + O(h^2)\,,$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T(x, y-h) - 2T(x, y) + T(x, y-h)}{h^2} + O(h^2)\,, \tag{2}$$

where $O(h)$ is the truncation error, and h is the smallest grid step between nodes. Within the context of formulas (2), the Laplace equation (1) can be rewritten as

$$T(x+h,y)+T(x-h,y)+T(x,y+h)+T(x,y-h)-4T(x,y)=0.$$

By using this equation along with the boundary conditions, which will be considered below, a sparse matrix can be constructed. In order to solve this matrix, three of the most commonly used methods are considered, i.e., the Jacobi, Gauss–Seidel, and conjugate-gradient methods [5, 6, 10: Chap. 20, 19, 20, 22: Chap. 22, 24: Chap. 10, 27, 29, 30, 31: Chap. 7].

The calculations will be conducted following a stop criterion of a maximum error of $e=10^{-6}$ for square RVEs. The total number of nodes for the whole RVE will be $n\times n$, where n is the number of nodes on each side.

Concerning the proposed algorithm, previously presented in Fig. 1, the increment on n for each resampling step will be obtained by a multiplicative scaling factor, labeled as α.

Then, the number of nodes on each side of the RVE on every resampling step will be the closest integer number to

$$n=n_0\alpha^i,$$

where n_0 is the number of nodes before the first resampling step and will be set 16 for all calculations, powers of two are selected because of their straightforward logarithmic representation.

The numbers of nodes on each side at the final resampling step (n_f) must be equal for all chosen α for a proper comparison. If this criterion is not achieved, the cumulative computation time for each case will not be comparable. Therefore, only those values are selected that satisfy the following equation

$$n_f=n_0\alpha^{i_f}.$$

Herein, i_f stands for the integer number of the resampling step needed to achieve a final number of nodes n equal to n_f (set $n_f=4096$) with a scaling factor of α. The proposed values are $\alpha=2^{1/4},2^{1/3},2^{1/2},2^1,2^{8/7},2^{4/3},2^{8/5}$, and 2^2.

Consider three types of boundary conditions shown in Fig. 2 and labeled as ***A***, ***B***, and ***C***. For boundary condition ***A*** (Fig. 2***a***), temperature profile T is defined along the edges of the RVE as

$$T(0,y)=1,\quad T(1,y)=0,$$

$$T(x,0)=\begin{cases}1, & 0\le x\le 0.75,\\ 4(1-x), & 0.75\le x\le 1,\end{cases}\quad T(x,1)=\begin{cases}1-4x, & 0\le x,\\ 0, & 0.25\le x\le 1.\end{cases} \tag{3}$$

Boundary condition (3) is expected to be the one with the lower computing time.

Boundary condition ***B*** (Fig. 2***b***) is similar to boundary condition ***A***, except for the sides $x=0,1$, where the Dirichlet conditions are imposed as,

$$\left.\frac{\partial T}{\partial x}\right|_{x=0}=\left.\frac{\partial T}{\partial x}\right|_{x=1}=0.$$

This boundary condition is expected to convey a higher computational burden in comparison to boundary condition ***A***.

Finally, boundary condition ***C*** implies the Dirichlet conditions imposed at the sides $x=0,1$, while the discontinuous temperature profiles are applied as

$$T(x,0) = T(x,1) = \begin{cases} 0, & 0 \le x < 0.45, \\ 1, & 0.45 \le x \le 0.55, \\ 0, & 0.55 < x \le 1. \end{cases}$$

This boundary condition is considered in order to evaluate the mesh resampling methodology performance when attempting functional discontinuities. The temperature profile is also expected to be better described as the number of nodes increases, especially around the discontinuities. Therefore, the effectiveness of the resampling methodology can be assessed in such cases.

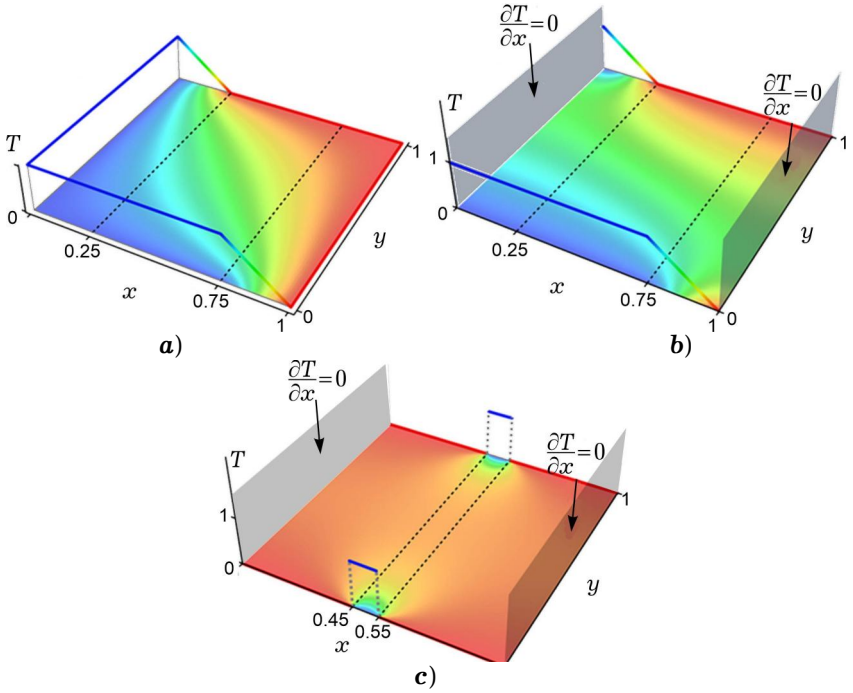


Fig. 2. Boundary conditions ***a***) ***A***, ***b***) ***B***, and ***c***) ***C*** for the considered case studies.

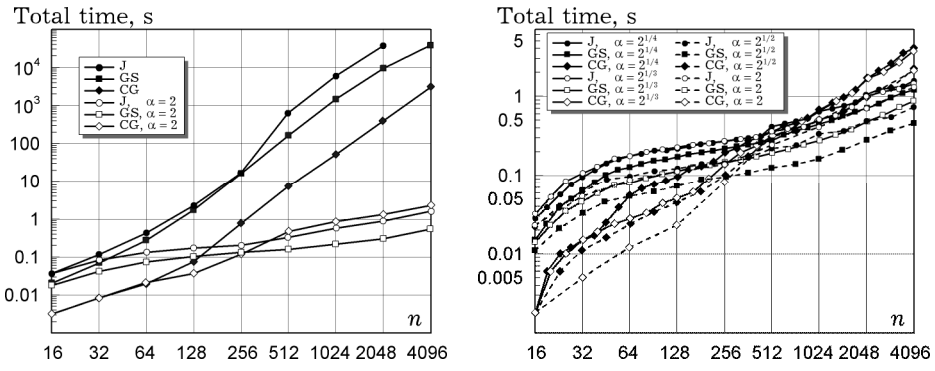
In order to ensure the results not to deviate from the converged solution when the resampling methodology is used, all calculations for each value of n, type of boundary condition, and the solution method will be repeated without considering the resampling methodology with a maximum error of $e = 10^{-6}$. The calculations will be repeated a second time without the resampling methodology but increasing the maximum error to $e = 10^{-10}$. These computations will only be conducted using the conjugate-gradient method because its results are expected to be more accurate. The maximum error on the RVE for all computations with $e = 10^{-6}$, with both use and no use of the resampling methodology, will be obtained by comparing the results to those obtained when the maximum error is $e = 10^{-10}$ and the conjugate-gradient method is used. Therefore, the resampling methodology will be validated if the maximum errors are not increased when it is employed.

2. Results and discussion.

2.1. Comparison of total computing time for all three chosen methods and boundary condition A.

Figure 3 shows the total computing time evolution for the three methods under the boundary condition ***A***. It is important to highlight two features. First, the proposed methodology yields an enormous reduction in computing time by several orders of magnitude, especially when large numbers of nodes are used. Second, while the conjugate-gradient method is already more time-

efficient before implementing the proposed approach, its advantage over Gauss–Seidel decreases as the number of elements increases. This trend continues until reaching a value of n around 256, for which the total computing time starts to be lower for Gauss–Seidel, compared to the conjugate gradient time.



a) ***b***)

Fig. 3. Total computing time as a function of n for boundary condition ***A***:
a) comparing results with the resampling methodology and without it,
b) comparing results for different values of α.

The total computing time analysis was also carried out as a function of the value of α for all three considered solving methods. These results can be seen in Fig. 4, being the Gauss–Seidel method, the one with the lower computing time for all values of α. Herein, the higher values of computing time as a function of α are present for the larger and smaller values among all studied methods. While, for lowering the total computing time, an optimal window of values of α is found from 1.5 to 2.5. This higher total computing time for smaller α is a consequence of more resampling steps involved, contributing to the total computing time, even though each step is less time consuming than a step with a higher value of α. Therefore, there is a balance between the total number of steps and the computing time involved for each one.

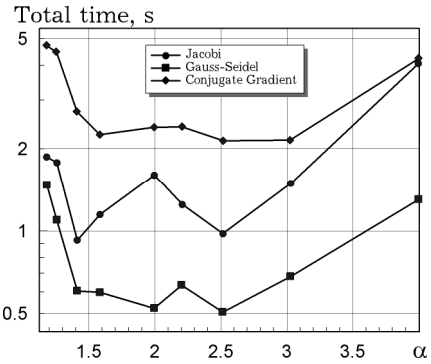


Fig. 4. Total computing time as a function of α for all three methods and boundary condition ***A***.

2.2. Error at the first iteration of the solving method for each resampling step and boundary condition A. Figure 5 shows the calculated error at the first iteration of the solving method, after the guess predicted by linear interpolation from the previous resampling step's solution. Each plot line corresponds to one of the chosen solution methods in conjunction with a specific scaling factor α. As can be seen, such error at the first iteration keeps decreasing for every consecutive resampling step. This trend is identified by a decreasing error as a function of the number of nodes on each

side of the RVE. This decrement continues until the error at the first iteration is lower than the stopping condition of $e = 10^{-6}$. Therefore, it can be concluded that after reaching a particular value of n through consecutive resampling steps, no additional improvement in the description of the nodal solution is obtained. Then, the first iteration error can be employed as a mechanism to conduct a convergence analysis.

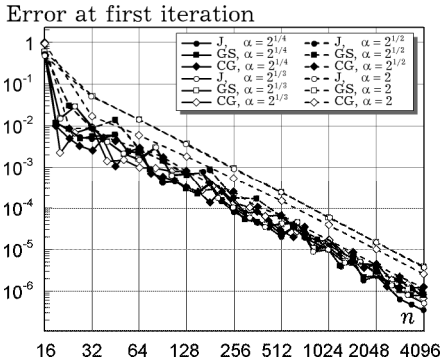


Fig. 5. Error at the first iteration for boundary condition ***A***.

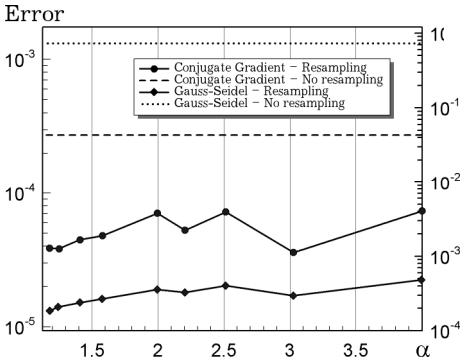


Fig. 6. Maximum temperature error on the RVE as a function of referred to the results of the conjugate gradient with a stopping condition of $e = 10^{-10}$ for boundary condition ***A***.

2.3. Maximum error on the RVE for each solving method and boundary condition A. In order to assess the influence of the proposed methodology on the precision of the results, the maximum error on the RVE for each case is estimated by comparing its temperature distribution to a reference profile. This reference profile is obtained without any resampling methodology and a stopping condition of $e = 10^{-10}$. Figure 6 shows the maximum errors for both Gauss–Seidel and conjugate gradient. Each solid line corresponds to the maximum error for the proposed methodologies, while the dotted lines stand for the error calculated without the resampling approach. Hereinafter, the Jacobi method results will not be analyzed since its performance is similar to that of Gauss–Seidel. Figure 6 shows an improvement of several orders of magnitude in the maximum error for both Gauss–Seidel and conjugate gradient using the proposed methodology.

Concerning the Gauss–Seidel method, the maximum error reduction is almost four orders of magnitude when using the resampling methodology. The larger improvements are mainly found at the center of the RVE. The main reason for this error distribution is the Gauss–Seidel solving algorithm in conjunction with the RVE dimensions. Herein, the solution is first reached near the boundary conditions. Then, the solution is propagated from them toward the whole RVE. Therefore, when the stopping condition for Gauss-Seidel is achieved, the regions more distant from the boundary conditions present larger errors. Thus, using the resampling methodology, the guess values for each instance of the solving method are closer to the solution. Then, the errors at the end of each resampling step are lower than those without the resampling methodology. For coarse changes in successive resampling steps, large α, the Gauss–Seidel's guess is more distant to the solution, then, an increasing trend of the maximum error is found. Therefore, it is highly advised to use small values of α for both Gauss–Seidel and Jacobi methods for increasing accuracy.

On the other hand, for the conjugate-gradient method, this improvement was about one order of magnitude for the maximum errors when using the resampling methodology. The maximum error using the proposed approach of

the conjugate-gradient method was fluctuating around $e = 10^{-5}$ for all values of α. Therefore, the resampling methodology is beneficial for increasing the accuracy of the results and reducing the total computing time. However, the value of α has a negligible influence on the precision. Then, the total computing time reduction criteria are recommended for selecting the value of α.

2.4. Discussion of all three boundary conditions. The temperature solutions on the RVE are mapped in Fig. 7 for all three boundary conditions and three representative values of n (16, 32, and 64). Besides a better resolution, the maps of temperature for boundary conditions ***A*** and ***B*** present minor variations when the number of elements increases. On the other hand, for boundary condition ***C***, discontinuities are placed on the temperature profiles through $y = 0,1$ (see Fig. 2***c***). The area's description around these singularities becomes more accurate as the number of nodes increases. Then, the effect of these discontinuities is further propagated towards the whole RVE. Consequently, an increase in the number of nodes improves the description of the map of temperature. Therefore, the resampling methodology proposed here is not recommended when abrupt discontinuities are present on the RVE.

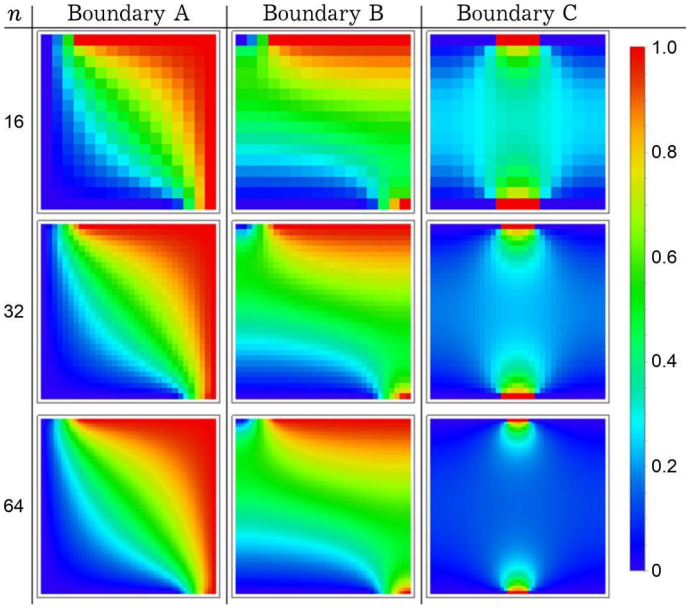


Fig. 7. Maps of temperature for all three boundary conditions considering $n = 16, 32, 64$.

In order to conduct a proper assessment of the influence of both n and α on the total computing time, the calculations were also carried out varying α with a set value of $n = 4096$ for the ***B*** and ***C*** boundary conditions. Two of the three solving methods are considered: Gauss–Seidel and conjugate gradient. Jacobi is not considered, as previously mentioned; its total computing time presents a similar behaviour to that of Gauss–Seidel, however, being several orders of magnitude higher. These computing times can be seen in Fig. 8***a*** and Fig. 8***b*** for Gauss–Seidel and conjugate gradient, respectively, and their reference values without using the proposed resampling methodology (dotted lines). In agreement with previous results, the Gauss–Seidel's total computing time was lower than that of the conjugate-gradient method. However, both methods were always kept less than an order of magnitude apart from each other. The exception was boundary condition ***C***, whose total computing time stayed in the same order of magnitude, regardless of the resampling methodology and the solving method employed.

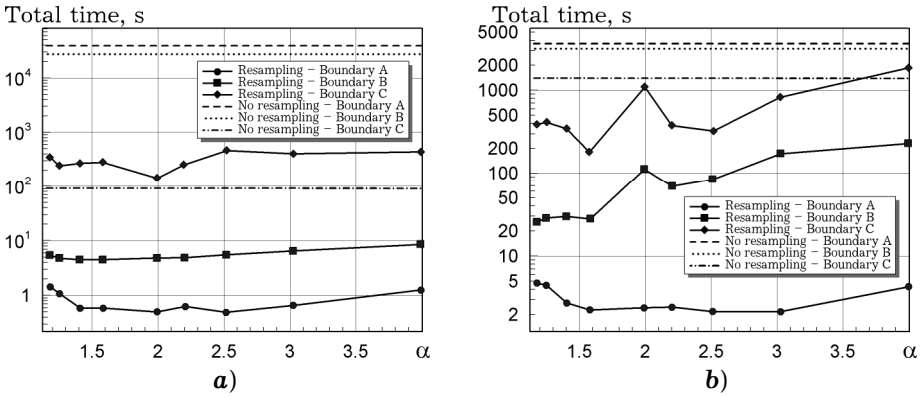


Fig. 8. Total computing time as a function of α, for all three boundary conditions, ***a***) Gauss–Seidel, ***b***) conjugate gradient.

Therefore, these results confirm that the resampling methodology is not beneficial when the solution is highly sensitive to the number of nodes, such as those with boundary conditions ***C***.

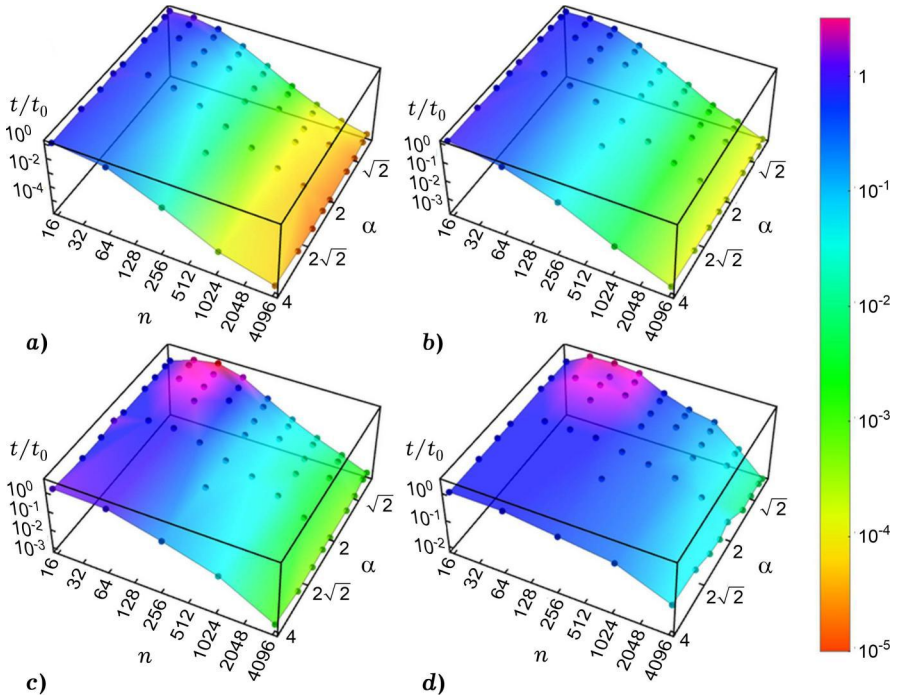


Fig. 9. Normalized computer time as a function of both α and n, considering ***A*** and ***B*** boundary conditions, ***a***) Gauss–Seidel, boundary condition ***A***; ***b***) Gauss–Seidel, boundary condition ***B***; ***c***) conjugate gradient, boundary condition ***A***; ***d***) conjugate gradient, boundary condition ***B***.

Finally, additional analyses were conducted in order to assess suitable values of α, for each boundary condition, solving method, and value of n. These results are shown in Fig. 9, where the computing time using the proposed methodology is normalized with respect to the calculations without the resampling approach. This normalized value is plotted as a function of both α and n. Figure 9***a*** and Fig. 9***c*** show the normalized computing time for boundary condition ***A*** using Gauss–Seidel and conjugate gradient, respectively. In comparison, the values for boundary condition ***B*** are presented in Fig. 9***b*** (Gauss–Seidel) and Fig. 9***d*** (conjugate gradient). Herein, boundary

condition ***C*** is not present due to its low improvement. Also, Jacobi's results are not shown because of their similar behaviour to Gauss–Seidel's.

According to Fig. 9***a*** and Fig. 9***b***, the implementation of the proposed methodology starts to be advantageous around $n = 34$ for Gauss–Seidel, and this behaviour is kept for higher values of n, whereas the influence of α is marginal. On the other hand, the normalized computing time for conjugate gradient is significantly influenced by both α and n (Fig. 9***c*** and Fig. 9***d***), and, for small values of them, a slight reduction in its performance can be obtained. Finally, for conjugate gradient, the results showed that the proposed resampling methodology starts to be beneficial at $n = 256$.

Conclusions. In this work, a mesh resampling methodology is proposed to speed up the Finite Difference method's solution, considering three widely used algorithms: Jacobi, Gauss–Seidel, and conjugate gradient. This methodology can be summarized as follows: an RVE with a low number of nodes is solved as a starting point. Then, the number of nodes is increased to create a new RVE, and the guessing values are interpolated from the previous solution. A new solution is obtained. Finally, this process is repeated until the desired criterion is fulfilled. The increment in the number of nodes between resampling steps is determined by a scaling factor proportional to the previous number of nodes.

Besides reducing the total computing time, the proposed resampling methodology is also beneficial in decreasing the numerical errors associated with the solving methods. Both improvements are found in most cases and are only missing when the solution is highly dependent on the number of nodes, e.g., when discontinuities are present.

The proposed resampling methodology decreases the computing time and the solving errors for all considered methods. Nevertheless, both reductions are more prominent for Jacobi and Gauss–Seidel, generally considered simpler than conjugate gradient. Furthermore, the reduction in computing time decreases around five orders of magnitude for both Jacobi and Gauss–Seidel with the RVEs studied here. In contrast, this reduction is slightly lower for conjugate gradient with the same RVEs, i.e., over an order of magnitude. Whereas the reduction on the solving error is found to be four orders of magnitude for Jacobi and Gauss–Seidel, and one order of magnitude for conjugate gradient.

As an additional advantage for the proposed methodology, a simultaneous convergence analysis can be implemented, taking advantage of the gradual increment of the number of variables that is common on those analyses. Furthermore, the maximum error at the first iteration after each resampling step is a possible criterion for assessing convergence analyses.

Herein, the mesh resampling methodology is applied to the Finite Difference Method. However, its implementation on other numerical methods is also possible, e.g., Finite Element Analysis or Finite Volumes Methods.

Acknowledgements. Dr. Domínguez-Rodríguez would like to thank to "Secretaría de Educación Pública" for its support through the program "Apoyo a la incorporación de nuevos profesores de tiempo completo, convocatoria 2018" with Project Number 511-6/18-8491.

1. *Abbaszadeh M., Dehghan M.* A finite-difference procedure to solve weakly singular integro partial differential equation with space-time fractional derivatives // Eng. Comput. – 2021. – **37**, No. 3. – P. 2173–2182. – https://doi.org/10.1007/s00366-020-00936-w.
2. *Abbaszadeh M., Dehghan M.* Meshless upwind local radial basis function-finite difference technique to simulate the time-fractional distributed-order advection–diffusion equation // Eng. Comput. – 2021. – **37**, No. 2. – P. 873–889. – https://doi.org/10.1007/s00366-019-00861-7.

3. *Atkinson K. E.* An introduction to numerical analysis. – Hoboken: John Wiley & Sons, 1989. – xvi + 693 p.
4. *Avriel M.* Nonlinear programming: analysis and methods. – New York: Dover Publ., 2003. – 528 p.
5. *Bagnara R.* A unified proof for the convergence of Jacobi and Gauss–Seidel methods // SIAM Rev. – 1995. – **37**, No. 1. – P. 93–97. – https://doi.org/10.1137/1037008.
6. *Christensen J., Bastien C.* Nonlinear optimization of vehicle safety structures: modeling of structures subjected to large deformations. – Oxford – Amsterdam: Butterworth-Heinemann, 2016. – 482 p. – https://doi.org/10.1016/C2013-0-00069-2.
7. *Croft D. R., Lihey D. G.* Heat transfer calculations using finite difference equations. – London: Applied Science Publishers Ltd, 1977. – xi + 283 p.
8. *Faber V., Manteuffel T.* Necessary and sufficient conditions for the existence of a conjugate gradient method // SIAM J. Numer. Anal. – 1984. – **21**, No. 2. – P. 352–362. – https://doi.org/10.1137/0721026.
9. *Fan E., Wang J., Liu Y., Li H., Fang Z.* Numerical simulations based on shifted second-order difference/finite element algorithms for the time fractional Maxwell's system // Eng. Comput. – 2022. – **38**, No. 1 (supplement). – P. 191–205. – https://doi.org/10.1007/s00366-020-01147-z.
10. *Ford W.* Numerical linear algebra with applications. – Amsterdam, etc.: Elsevier Academic Press, 2014. – 629 p.
11. *Forsyth P. A.* A control-volume, finite-element method for local mesh refinement in thermal reservoir simulation // SPE Res. Eng. – 1990. – **5**, No. 4. – P. 561–566. – https://doi.org/10.2118/18415-PA.
12. *Forsythe G. E., Wasow W. R.* Finite difference methods for partial differential equations. – Whitefish: Literary Licensing, LLC, 2013. – 454 p.
13. *Golub G. H., Van Loan C. F.* Matrix computations. – Baltimore: The Johns Hopkins University Press, 1996. – 728 p.
14. *Grossmann C., Roos H.-G., Stynes M.* Numerical treatment of partial differential equations. – Berlin – Heidelberg: Springer, 2007. – 608 p.
15. *Hestenes M. R., Stiefel E.* Methods of conjugate gradients for solving linear systems // J. Res. Nat. Bur. Standards. – 1952. – **49**, No. 6. – P. 409–436. – https://doi.org/10.6028/JRES.049.044.
16. *Hutter K., Jöhnk K.* Continuum methods of physical modeling. – Berlin – Heidelberg: Springer, 2004. – xv + 636 p.
17. *Jones M. T., Plassmann P. E.* Parallel algorithms for adaptive mesh refinement // SIAM J. Scie. Comput. – 1997. – **18**, No. 3. – P. 686–708. – https://doi.org/10.1137/S106482759528065X.
18. *Keppens R., Nool M., Tóth G., Goedbloed J. P.* Adaptive mesh refinement for conservative systems: multi-dimensional efficiency evaluation // Comput. Phys. Commun. – 2003. – **153**, No. 3. – P. 317–339. – https://doi.org/10.1016/S0010-4655(03)00139-5.
19. *Li W., Sun W.* Modified Gauss–Seidel type methods and Jacobi type methods for Z-matrices // Linear Algebra Appl. – 2000. – **317**, Nos. 1-3. – P. 227–240. – https://doi.org/10.1016/S0024-3795(00)00140-3.
20. *Milaszewicz J. P.* Improving Jacobi and Gauss–Seidel iterations // Linear Algebra Appl. – 1987. – 93. – P. 161–170. – https://doi.org/10.1016/S0024-3795(87)90321-1.
21. *Mirzaee F., Samadyar N.* Combination of finite difference method and meshless method based on radial basis functions to solve fractional stochastic advection–diffusion equations // Eng. Comput. – 2020. – **36**, No. 4. – P. 1673–1686. – https://doi.org/10.1007/s00366-019-00789-y.
22. *Modest M. F.* Radiative heat transfer. – New York, etc.: Elsevier Academic Press, 2013. – 904 p.
23. *Özişik M. N., Orlande H. R. B., Colaço M. J., Cotta R. M.* Finite difference methods in heat transfer. – Boca Raton: CRC Press, 2017. – 600 p.
24. *Phillips G. M., Taylor P. J.* Matrix norms and applications // In: Theory and applications of numerical analysis. – Amsterdam, etc.: Elsevier Academic Press, 1996. – 447 p. – (Chap. 10. – P. 265–298.) – https://doi.org/10.1016/B978-012553560-1/50011-2.
25. *Polyak B. T.* The conjugate gradient method in extremal problems // USSR Comput. Math. Math. Phys. – 1969. – **9**, No. 4. – P. 94–112. – https://doi.org/10.1016/0041-5553(69)90035-4.

26. *Powell M. J. D.* Restart procedures for the conjugate gradient method // Math. Program. – 1977. – **12**, No. 1. – P. 241–254. – https://doi.org/10.1007/BF01593790.
27. *Salkuyeh D. K.* Generalized Jacobi and Gauss–Seidel methods for solving linear system of equations // Numer. Math. J. Chinese Univ. – 2007. – **16**, No. 2. – P. 164–170.
28. *Shewchuk J. R.* An introduction to conjugate gradient method without the agonizing pain. – Pittsburgh: Carnegie Mellon University, 1994. – iv + 58 p.
29. *Tian Z., Tian M., Liu Z., Xu T.* The Jacobi and Gauss–Seidel-type iteration methods for the matrix equation $AXB = C$ // Appl. Math. Comput. – 2017. – **292**. – P. 63–75. – https://doi.org/10.1016/j.amc.2016.07.026.
30. *Tritsiklis J. N.* A comparison of Jacobi and Gauss–Seidel parallel iterations // Appl. Math. Let. – 1989. – **2**, No. 2. – P. 167–170. – https://doi.org/10.1016/0893-9659(89)90014-1.
31. *Yang K. H.* Basic Finite Element Method as applied to injury biomechanics. – London: Elsevier Academic Press, 2017. – 748 p.

СКОРОЧЕННЯ ЧАСУ ОБЧИСЛЕНЬ РОЗРІДЖЕНИХ МАТРИЦЬ ІЗ ВИКОРИСТАННЯМ ПЕРЕДИСКРЕТИЗАЦІЇ СІТКИ У СКІНЧЕННО-РІЗНИЦЕВОМУ АНАЛІЗІ

Запропоновано методологія скорочення часу обчислення скінченних різниць, яка полягає в поступовому збільшенні кількості вузлів на кожній стороні прямокутного репрезентативного об'єму та інтерполяції припущених значень із попереднього вузлового рішення. Цей процес повторюється до досягнення кінцевої кількості вузлів. Розглянуто три методи розв'язування для трьох різних типів крайових умов, у тому числі розривних, для оцінки ефективності запропонованої методики. Виявлено скорочення часу обчислення на п'ять порядків для методів Якобі і Ґаусса–Зайделя. Для методу спряжених ґрадієнтів таке скорочення становить один порядок.

Ключові слова: *метод скінченних різниць, передискретизація сітки, метод Якобі, метод Ґаусса–Зейделя, спряжений ґрадієнт.*

Autonomous University of Campeche,
San Francisco de Campeche, Cam, Mexico

Received
01.04.24