

Research Article

Efficient Alternative for Construction of the Linear System Stemming from Numerical Solution of Heat Transfer Problems via FEM

Estaner Claro Romão

Department of Basic and Environmental Sciences, Engineering School of Lorena, University of São Paulo, 12602-810 Lorena, SP, Brazil

Correspondence should be addressed to Estaner Claro Romão; estaner23@yahoo.com.br

Received 27 August 2015; Revised 22 October 2015; Accepted 1 November 2015

Academic Editor: Zhiqiang Hu

Copyright © 2016 Estaner Claro Romão. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an efficient alternative to construction of the linear system coming from a solution via the Finite Element Method that is able to significantly decrease the time of construction of this system. From the presentation of the methodology used and a numerical application it will be clear that the purpose of this work is to be able to decrease 6-7 times (on average) the linear system building time.

1. Introduction

The several activities related to engineering and research connected with it are not motivated solely and exclusively by human curiosity, but mainly real needs, which often need to be resolved quickly and accurately. The heat transfer study is of great importance in various branches of engineering. The interest to know the heat transfer mechanisms may involve equipment operations, such as in boilers, condensers, and air preheater. In cooling systems and those of air conditioning that involve heat exchangers, the heat transfer study is extremely important for mechanical engineering. In electrical engineering, in turn, there is an interest to know the heat dissipation in chips and semiconductor devices, in chemical engineering the interest is in heat transfer processes in various chemical reactions, and in environmental engineering there is an interest to study the effect of heat in the dispersion of pollutants in the air and the diffusion of pollutants in soils and thermal pollution in lakes and seas and their impact on human life, in addition to many other applications in other branches of engineering.

The majority of physical problems are governed or may be represented by partial differential equations. The 3D convection-diffusion-reaction problems can be represented,

in general, using the following partial differential equation [1]:

$$k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2} + k_z \frac{\partial^2 T}{\partial z^2} + A_x \frac{\partial T}{\partial x} + A_y \frac{\partial T}{\partial y} + A_z \frac{\partial T}{\partial z} + BT = 0, \quad (1)$$

where k_x , k_y , and k_z are the thermal conductivity terms in each space direction, A_x , A_y , and A_z are the velocity components in each space direction, and B is a reactive term.

Some mathematical methods are able to supply analytical solutions of physical problems, especially problems from heat transfer, but only of some and very specific problem, due to which it becomes unviable for engineering practice. So numerical methods are essential tools for solving heat transfer problems.

For decades the numerical methods have been used to solve such problems, among which stand out the Finite Difference Method [2–7], the Finite Volume Method [8–11], and the Finite Element Method [12–17]. You can find the open literature numerous contributions in the study of convection-diffusion-reaction problems in permanent and transient states in the three numerical methods mentioned above. But specifically in the Finite Element Method, the

TABLE 1: Connectivity matrix for a domain divided into 12 hexahedrons elements with 8 nodes (see Figure 1).

Element	Nodes							
	1	2	3	4	5	6	7	8
1	1	2	5	6	13	14	17	18
2	2	3	6	7	14	15	18	19
3	3	4	7	8	15	16	19	20
4	5	6	9	10	17	18	21	22
5	6	7	10	11	18	19	22	23
6	7	8	11	12	19	20	23	24
7	13	14	17	18	25	26	29	30
8	14	15	18	19	26	27	30	31
9	15	16	19	20	27	28	31	32
10	17	18	21	22	29	30	33	34
11	18	19	22	23	30	31	34	35
12	19	20	23	24	31	32	35	36

Galerkin method is known to present excellent numerical precision with low computational time for pure diffusion problems or diffusion-reaction [18–20]; however the same cannot happen with dominant convection problems [1, 21] and in the analysis of flows [22].

One may here mention other variants of the variants of Finite Element Method, but the main focus of this paper is related to a new proposal for constructing a global linear system coming from a formulation made with some of the variants of the Finite Element Method. Here, just for simplicity, Galerkin method will be used for such study.

In the following, a summary of the manner which was performed for the construction of the linear system in [1, 22] will be presented, which is a traditional form found in several books in this research area (like, e.g., in [12, 23]), and then the purpose of this paper will be described.

2. Methodology

In the works cited above [1, 22], the procedure adopted for the construction of global linear system stemming from contributions of each element was as follows: for each matrix element constructed, assign its values in the global matrix; this process occurs element by element with the aid of an array traditionally known as a connectivity matrix (Table 1).

Therefore, each contribution of an element (matrix element) is inserted in the global linear system that represents the physical characteristics of the problem. However, following this manner and remembering that each line of the global matrix is directly connected to its node, that is, row 1 to node 1, row 2 to node 2, and so on, construction of the mesh occurs in a manner unordered form because the contributions that each line of the global matrix receives do not occur in an orderly manner. See, for example, from Table 1, that line 7 (line of the global matrix) relative to node 7 of the global matrix will receive contributions from elements 2, 3, 5, and 6 (they are bold in the table); that is, in this interval of construction there will be contributions from other nodes before finishing the relative to node 7 (line 7), so the construction becomes of unordered form and slow.

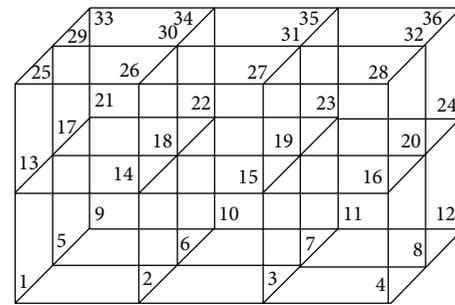


FIGURE 1: 3D domain divided into 12 hexahedrons elements with 8 nodes (each).

See also that the global matrix in position, line 7 column 7, will receive contribution of four cited elements, but at different times of the construction of this system, it would be quiet if the storage of the global matrix is made in a full matrix ($NNost \times NNost$ dimension), where $NNost$ is the number of nodes in the mesh, but it will be unfeasible for reasons of computational memory, where an array of this type, being a matrix extremely sparse, has a large number of zero coefficients that are unnecessary storage. So archiving only the nonzero coefficients of the global matrix becomes necessary; however, create it in a disorderly manner (see an example of this at the top of Table 2); the solution process will make the linear system extremely slow, so the construction in an orderly manner also becomes necessary.

Thus, the purpose of this paper is to build the global linear system in orderly form and faster in order to accelerate the computation process, for it is proposed that instead of building the global linear system in the traditional form, that is, element by element, build it node to node.

For this, it is proposed from connectivity matrix (see again Table 1); create a matrix that presents for each node which elements that are contained, as the example shown in Table 3 (this table was constructed conforming to Figure 1).

With a matrix constructed conforming to the idea presented in Table 3, define KNN of dimensions of $NNost \times 8$,

TABLE 2: Example of matrix constructed of disorderly manner (above) and orderly one (below).

Disorderly	Row	1	2	1	3	7	2	...
	Column	1	3	5	1	2	9	...
	Coefficients	0.5	0.2	0.7	0.8	0.9	1.5	...
Orderly	Row	1	1	2	2	3	7	...
	Column	1	5	3	9	1	2	...
	Coefficients	0.5	0.7	0.2	1.5	0.8	0.9	...

```

ka = 1
do i = 1,NNost
  do j = 1,8
    if(KNN(i,j).eq.0) then
      go to (1)
    end if
    do jj = 1,NNos
      if(KCONEC(KNN(i,j),jj).eq.i) then !KCONEC is the connectivity matrix conform Table 1
        iAUX = jj ! discover of which line of the element is the contribution
        go to (2)
      end if
    end do
  (2) do ii = 1,NNos
    do jj = ka,k
      if(Mjcol(jj).eq.KCONEC(KNN(i,j),ii).and.Mirow(jj).eq.i) then
        GG(jj) = GG(jj) + G(iAUX,ii) !G is the element matrix
        iAUXj = jj + 1
        go to (3)
      end if
    end do
  (3) continue
  end do
(1) continue
ka = iAUXj
end do

```

CODE 1: Piece of the code that constructs the global linear system.

where “8” is the maximum number of elements to which a node can belong (it is important to note that this quantity may vary according to each generator mesh). As an example let us consider node 2 that is contained in elements 1 and 2; according to Table 3, thus, it is known that node 2 will receive contributions of these two elements through line 2 of the global matrix. With that, from Table 1, in element 1 node 2 is in position 2 (i.e., it will receive the information of line 2 of element 1) and in element 2 node 2 is in position 1 (i.e., receiving information of line 1 of element 2).

Code 1 presents a portion of the code used to construct the linear system from the use of the connectivity matrix built conforming to the models presented in Tables 1 and 3. In Code 1, *NNodes* is the number of nodes in the element (in this paper will be used a hexahedron with 8 nodes) and in this table and in other codes that will be presented during this paper was used a Fortran programming language.

The piece of Code 1, after insertion of each contribution of each element (matrix element, *G*, *NNodes* × *NNodes* dimension), will create three vectors: *GG*, a vector of real

TABLE 3: Connectivity matrix constructed to present for each node which elements that are contained.

Nodes	Elements							
	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	0
2	1	2	0	0	0	0	0	0
3	2	3	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
6	1	2	4	5	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	1	2	4	5	7	8	10	11
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

values that will load the information of nonzero coefficients global matrix, and *Mirow* and *Mjcol* vectors of integer values that loaded, respectively, the information of the position

```

!* * * * * GAUSS SEIDEL METHOD * * * * *
T = 1.d0; Ti = T; rAux = 0.d0; Emax = 0.000000001d0  !T: Temperature; Ti: initial shot
do i = 1, 200000
  ii = 1; rAuxE = 0.d0
  do j = 1, nnost
    soma = 0.d0
    do jj = ii, k
      if(Mirow(jj).eq.j.and.Mjcol(jj).eq.j) then
        rAux = GG(jj)
      end if
      if(Mirow(jj).eq.j) then
        if(Mjcol(jj).eq.j) then
          continue
        else
          soma = soma + GG(jj)*T(Mjcol(jj))
        end if
      else
        ii = jj
        go to (2)
      end if
    end do
  (2) T(j) = (1.d0/rAux)*(FG(j) - soma)
  end do
  do j = 1, nnost
    Erro = dabs(Ti(j) - T(j))
    if(Erro > rAuxE) then
      rAuxE = Erro
    end if
  end do
  if(rAuxE > Emax) then  !Emax: stop criterion
    Ti = T
  else
    go to (1)
  end if
end do
(1) continue

```

CODE 2: Gauss-Seidel method used for calculating the global linear system.

TABLE 4: Example of construction of the vectors GG, Mirow, and Mjcol.

GG	-0.15	0.04	-0.02	...	1.12	2.00	1.13	...
Mirow	1	1	1	...	2	2	2	...
Mjcol	1	2	5	...	1	7	2	...

(row and column) of each one of the coefficients of the GG vector. It is noteworthy that the vector GG constructed only accumulates nonzero values. The right side of the global linear system is not being mentioned here in this paper because there is no change in its construction compared to that implemented in [1, 22]. In Table 4, an example of the construction of vectors GG, Mirow, and Mjcol for a unit cube divided into 12 elements is presented.

Note that, in the example shown in Code 1, the vectors are constructed in an orderly manner with respect to line, but the same does not happen with the column; let us see the case of line 2, where column 7 arises before column 2. The code

shown in Code 1 was constructed to only be concerned with the ordering of the rows, since the solution method of the linear system used in this paper is the Gauss-Seidel method, so the ordering of the columns is not necessary and does not interfere with computation speed of the global linear system. For details, see, in Code 2, a piece of code to calculate the global linear system using the Gauss-Seidel method.

It is noteworthy that if it was also necessary to have the ordinates columns, a simple code would solve this. Furthermore, the choice of the Gauss-Seidel method is due to its simplicity of implementation; however, this proposal could be adapted to other methods of solving linear systems, as, for example, the method of conjugate gradients when the formulation is done via LSFEM (Least Squares Finite Element Method).

3. Numerical Results

To analyze the efficiency of this proposal, we analyze application 2 of [22] with $k_x = k_y = k_z = 1$, $B = 1$, and

```

Mirow = 0; Mjcol = 0; KNN = 0; k = 1; ka = 1
do i = 1,NNost
  ki = 1; iel = 1
  do j = 1,Nelem
    do ii = 1,NNos
      if(KCONEC(j,ii).eq.i) then
        iAUX = j
        KNN(i,ki) = j
        ki = ki + 1
        iel = iel + 1
        go to (1)
      end if
      if(ii.eq.NNos) then
        go to (3)
      end if
    end do
  (1) continue
  if(iel.eq.2) then
    do ii = 1,NNos
      Mirow(k) = i
      Mjcol(k) = KCONEC(iAUX,ii)
      k = k + 1
    end do
  else
    do ii = 1,NNos
      do kk = ka,k
        if(Mirow(kk).eq.i.and.Mjcol(kk).eq.KCONEC(iAUX,ii)) then
          go to (2)
        end if
      end do
      Mirow(k) = i
      Mjcol(k) = KCONEC(iAUX,ii)
      k = k + 1
    (2) continue
  end do
  end if
  (3) continue
  end do
  ka = k - 1
end do
k = k - 1
continue

```

CODE 3: Code used for the construction of KNN for CASE 3.

$A_x = A_y = A_z = 0$, and this paper was using a process element by element for construction of the linear system and thus ordering the vectors thus constructed; it will be called CASE 1.

The proposal presented in this paper through the previous item will be called CASE 2. However, in this paper, CASE 2 was performed in such a way that the computer's own code constructs the mesh and its connectivity matrix, and with that, mesh was performed which is constructed building code lines using this knowledge; however, there may be situations that a mesh stemming from an external mesh generator to the FEM code is used and therefore did not have knowledge of how this mesh is constructed, so here it will be called CASE 3, the situation where how the mesh is constructed is not known, so for each node of the mesh, the connectivity matrix

(Table 1) has to be read in its entirety to thus determine for each node which elements that the contains.

For clarity, in the code of CASE 2, it is known that the mesh is constructed from left to right (in x), from front to back (in y), and from the bottom up (in z) to an orthogonal three-dimensional domain composed of hexahedrons elements; also, for example, it is known that node 1 is only element 1 (see Figure 1); it is not necessary to read all elements as in CASE 3. Next, in Code 3, a code for construction of KNN in CASE 3 is presented.

The numerical results of this application are shown in Table 5, where h is the edge length of the element built in cubic form, L_{inf} and CR are, respectively, the maximum error committed and the convergence rate as described in [1], and C1C3% is the percentage comparison of the numerical results

TABLE 5: Numerical results for the three case studies.

Nnost	8 nodes per element				In seconds				
	Nelem	h	L_{inf}	CR	CASE 1	CASE 3	% C1C3	CASE 2	% C1C2
1331	1000	0.100	$7.08E - 05$	—	24.39	1.09	4.47%	1.07	4.39%
2744	2197	0.077	$4.22E - 05$	1.97	52.45	3.60	6.86%	3.45	6.58%
4096	3375	0.067	$3.16E - 05$	2.02	94.45	6.54	6.92%	6.51	6.89%
6859	5832	0.056	$2.20E - 05$	1.98	176.08	15.00	8.52%	15.51	8.81%
9261	8000	0.050	$1.78E - 05$	2.01	252.98	25.09	9.92%	19.85	7.85%
13824	12167	0.043	$1.34E - 05$	2.03	218.00	55.88	25.63%	45.92	21.06%
17576	15625	0.040	$1.14E - 05$	1.93	313.46	83.87	26.76%	65.42	20.87%
24389	21952	0.036	$9.06E - 06$	2.02	485.78	144.51	29.75%	117.15	24.12%
29791	27000	0.033	$7.90E - 06$	1.98	664.28	202.25	30.45%	146.48	22.05%
<i>Average*</i>							16.59%		13.62%

* Average is the average of all percentual values.

of the time spent between CASE 1 and CASE 3 and C1C2% of CASE 1 and CASE 2 (e.g., in $N_{\text{Nost}} = 1331$, $N_{\text{elem}} = 1000$, $h = 0.100$, $L_{\text{inf}} = 7.08E - 05$, and % C1C3 = 4.47% because CASE 1 (C1) = 24.39 and CASE 3 (C3) = 1.09; in other words, $1.09/24.39 = 4.47\%$). It is clear to perceive by the results, analyzing the average of these, that revolve around a decrease in the time 6-7 times, and it is good to make it clear that this time shown in Table 5 is the time to construct the mesh, construct the connectivity matrix (KNN in CASE 2 and CASE 3), and construct and calculate the global linear system.

4. Conclusions

It is believed that this proposal is very important and it could bring many benefits in the art of programming numerical of heat transfer problems via Finite Element Method. It is clear that, in character of numerical programming, lines of code could be presented maybe a little more technically better, but this is not the main objective of this paper.

The most interesting part of this paper is the results presented in CASE 3, where, without having knowledge of how the mesh is built, we obtained an average of 16.59% of the time spent compared with CASE 1 which is used in a traditional manner, element by element, for construction of the global linear system, that is, a significant improvement in computation time.

Conflict of Interests

The author has not declared any conflict of interests.

Acknowledgment

The FAPESP (Proc. 2014/06679-8) supported the present work.

References

- [1] E. C. Romão and L. F. M. de Moura, "Galerkin and least squares methods to solve a 3D convection-diffusion- reaction equation with variable coefficients," *Numerical Heat Transfer—Part A: Applications*, vol. 61, no. 9, pp. 669–698, 2012.
- [2] G. D. Smith, *Numerical Solution of Partial Differential Equations*, Oxford Mathematical Handbooks, New York, NY, USA, 1971.
- [3] A. R. Bahadır, "A fully implicit finite-difference scheme for two-dimensional Burgers' equations," *Applied Mathematics and Computation*, vol. 137, no. 1, pp. 131–137, 2003.
- [4] M. N. Ozisik, *Finite Difference Methods in Heat Transfer*, CRC Press, Boca Raton, Fla, USA, 1994.
- [5] S. F. Radwan, "Comparison of higher-order accurate schemes for solving the two-dimensional unsteady Burgers' equation," *Journal of Computational and Applied Mathematics*, vol. 174, no. 2, pp. 383–397, 2005.
- [6] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Method*, Clarendon Press, Oxford, UK, 3rd edition, 1998.
- [7] M. Cui, "Convergence analysis of high-order compact alternating direction implicit schemes for the two-dimensional time fractional diffusion equation," *Numerical Algorithms*, vol. 62, no. 3, pp. 383–409, 2013.
- [8] B. R. Baliga, T. T. Pham, and S. V. Patankar, "Solution of some two-dimensional incompressible fluid flow and heat transfer problems, using a control volume finite-element method," *Numerical Heat Transfer B: Fundamentals*, vol. 6, no. 3, pp. 263–282, 1983.
- [9] O. C. Zienkiewicz and R. L. Taylor, *Finite-Element Method, Volume 3: Fluid Dynamics*, Butterworth Heinemann, Oxford, UK, 2000.
- [10] B. R. Baliga, T. T. Pham, and S. V. Patankar, "Solution of some two-dimensional incompressible fluid flow and heat transfer problems, using a control volume finite-element method," *Numerical Heat Transfer, Part B*, vol. 6, no. 3, pp. 263–282, 1983.
- [11] T. J. Chung, *Computational Fluid Dynamics*, Cambridge University Press, Cambridge, UK, 2002.
- [12] J. N. Reddy, *An Introduction to the Finite Element Method*, McGraw-Hill, 2nd edition, 1993.
- [13] M. R. Vujicic and S. G. R. Brown, "Iterative solvers in the finite element solution of transient heat conduction," *FME Transactions*, vol. 32, pp. 61–68, 2004.
- [14] Z. Si, X. Feng, and A. Abduali, "The semi-discrete streamline diffusion finite element method for time-dependent convection-diffusion problems," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 771–779, 2008.

- [15] G. Dhatt and G. Touzot, *The Finite Element Method Displayed*, John Wiley & Sons, New York, NY, USA, 1984.
- [16] R. W. Lewis, P. Niyhiarasu, and K. N. Seetharamu, *Fundamentals of the Finite Element Method for Heat and Fluid Flow*, John Wiley & Sons, New York, NY, USA, 2004.
- [17] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*, John Wiley & Sons, New York, NY, USA, 2003.
- [18] L. L. Burrell, L. Q. Tang, and T. T. H. Tsang, "On a least-squares finite element method for advective transport in air pollution modeling," *Atmospheric Environment*, vol. 29, no. 12, pp. 1425–1439, 1995.
- [19] E. C. Romão and L. F. M. de Moura, "3D contaminant transport by GFEM with hexahedral elements," *International Communications in Heat and Mass Transfer*, vol. 42, pp. 43–50, 2013.
- [20] E. C. Romão, "3D unsteady diffusion and reaction-diffusion with singularities by GFEM with 27-node hexahedrons," *Mathematical Problems in Engineering*, vol. 2014, Article ID 560492, 12 pages, 2014.
- [21] N. Camprub, I. Colominas, F. Navarrina Casteleiro, and M. Galerkin, "Least-Squares and G.L.S. numerical approaches for convective-diffusive transport problems in engineering," in *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS '00)*, Barcelona, Spain, September 2000.
- [22] E. C. Romão, M. D. Campos, and L. F. M. Moura, "Application of galerkin and least-squares finite element method in the solution of 3D poisson and helmholtz equations," *Computers & Mathematics with Applications*, vol. 62, no. 11, pp. 4288–4299, 2011.
- [23] B. N. Jiang, *The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics*, Springer, New York, NY, USA, 1998.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

