Nova scientia
ISSN: 2007-0705

Universidad de La Salle Bajío A. C., Coordinación de Investigación

Ortigoza Capetillo, Gerardo M.; Lorandi Medina, Alberto P.; García Reynoso, Alfonso C.
Reordering edges and elements in unstructured meshes
to reduce execution time in Finite Element Computations
Nova scientia, vol. 10, no. 20, 2018, pp. 263-279
Universidad de La Salle Bajío A. C., Coordinación de Investigación

DOI: 10.21640/ns.v10i20.1317

Nova Scientia
Revista de Investigación de la Universidad De La Salle Bajío

Nº 20, Vol. 10 (1), 2018. **ISSN 2007 – 0705**, pp.: 263 – 279

Reordering edges and elements in unstructured meshes to reduce execution time in Finite Element Computations
Reordenando bordes y elementos en mallas no estructuradas para reducir el tiempo de ejecución en Cálculos de Elementos Finitos

Gerardo M. Ortigoza Capetillo[1], Alberto P. Lorandi Medina[2] y Alfonso C. García Reynoso[2]

Palabras clave: re-enumeración de lados y elementos; formulación de elementos finitos; reducción de ancho de banda y tiempo de ejecución; matriz esparcida por vector
Keywords: RCM reordering; finite element; bandwidth reduction of stiffness matrices; edge and element graphs of the mesh

Recepción: 02-01-2018 / Aceptación: 04-02-2018

Resumen

En este trabajo mostramos cómo el reordenamiento RCM puede aplicarse a los lados y a los elementos de una malla no estructurada (tetraedros/triángulos) para reducir el ancho de banda de las matrices de esfuerzos para las formulaciones de elementos finitos definidas sobre lados o sobre elementos. Los generadores de malla han sido diseñados principalmente para elementos finitos nodales, sus salidas son una lista de nodos (2d/3d) y un arreglo de la conectividad de los elementos (triángulos/tetraedros). Sin embargo, para las formulaciones de elementos finitos basados en los lados se requiere una enumeración de los lados de la malla. Reportamos observaciones realizadas con los generadores de mallas Triangle y Tetgen y las estructuras esparcidas de las matrices de esfuerzos obtenidas en formulaciones sobre los lados y sobre elementos. El RCM es un algoritmo de re-enumeración tradicionalmente aplicado a los nodos de la malla. Así para aplicar este algoritmo de re-enumeración al caso de elementos finitos sobre lados y elementos, definimos los grafos de los lados y el grafo de los elementos. Obtenemos así una noTable reducción de ancho de banda de las matrices de esfuerzos, lo que se traduce en reducción de tiempo de ejecución en la multiplicación de matrices esparcidas por un vector. Usamos matrices comprimidas por filas e implementamos la multiplicación de matriz esparcida por vector en una rutina en paralelo usando OpenMp.

Abstract

Reverse Cuthill McKee (RCM) reordering can be applied to either edges or elements of unstructured meshes (triangular/tetrahedral), in accordance to the respective finite element formulation, to reduce the bandwidth of stiffness matrices. Grid generators are mainly designed for

[1] Facultad de Ingeniería, Universidad Veracruzana, Boca del Rio, Veracruz
[2] Instituto de Ingeniería, Universidad Veracruzana, Boca del Río, Veracruz. E-mail: alorandi@uv.mx

nodal based finite elements. Their output is a list of nodes (2d or 3d) and an array describing element connectivity, be it triangles or tetrahedrons. However, for edge-defined finite element formulations a numbering of the edges is required. Observations are reported for Triangle/Tetgen Delaunay grid generators and for the sparse structure of the assembled matrices in both edge- and element-defined formulations. The RCM is a renumbering algorithm traditionally applied to the nodal graph of the mesh. Thus, in order to apply this renumbering to either the edges or the elements of the respective finite element formulation, graphs of the mesh were generated. Significant bandwidth reduction was obtained. This translates to reduction in the execution effort of the sparse-matrix-times-vector product. Compressed Sparse Row format was adopted and the matrix-times-vector product was implemented in an OpenMp parallel routine.

Introduction

The finite element method is one of the most popular general purpose techniques for computing accurate solutions to partial differential equations (pdes). Since pdes form the basis for many mathematical models in the physical sciences and other fields, it is not hard to realize the importance of the finite element method. The finite element method reduces a boundary value problem for a partial differential equation or system of pdes to a system of linear equations, written in a matrix form $KU = f$ that can be solved numerically. In many cases the stiffness $\left(K\right)$ matrix is symmetric and sparse, with the sparse structure highly influenced by the node (nodal finite element method [21], edge (edge elements [10]), and element (discontinuous finite element method [4, 3]) numberings provided by the grid generator.

Here, sparse matrix techniques are preferable since the storage required increases as $0\left(N\right)$ where N denotes the degrees of freedom of the problem. Storage can be reduced by storing only the nonzero elements with a compress sparse row format. Sparse matrix-vector multiplication is an important kernel in many iterative solvers; with parallel implementations

We can obtain a solution in a reasonable amount of time but they suffer from low cache utilization due to unstructured data access patterns. [7, 2] proposed to reorder sparse matrices using a bandwidth reduction technique in order to reduce the number of cache misses to improve the memory-system performance of the sparse matrix-vector multiplication. Sometimes it is not easy to assemble the stiffness matrix and reorder its elements since the matrix is assembled at running time, or if it is too large it must be split among different processors. Here we discuss how to use RCM to reorder nodes, edges and elements of the mesh in order to reduce the bandwidth of the assembled matrices. The work is organized as follows: section 2 introduces the edge- and element-defined finite element formulations employed, in section 3 the different meshes used in our numerical computations and the influence of the edge and element ordering in the sparse structure of the finite element matrices are presented, then section 4 introduces the RCM renumbering of edge and elements graphs, with some numerical experiments also included, and finally section 5 establishes the conclusions of this work.

Finite Element formulations

Edge-element formulation

For the edge-element formulation let us consider the problem of calculating resonant frequencies of cavities ([8, 1]). The nodal formulation is:

$$\nabla\times\left(\frac{1}{\mu_r}\nabla\times E\right)-k_c^2\varepsilon_r E=0 \qquad (1)$$

Where $E=E_x\hat{x}+E_y\hat{y}+E_z\hat{z}$, μ and ε_r are the material permittivity and permeability respectively. The finite element formulation is given by

$$\int_V\frac{1}{\mu_r}\left(\nabla\times W_n\right)\cdot\left(\nabla\times E\right)dv=k_c^2\varepsilon_r\int_V W_n\cdot Edv \qquad (2)$$

The electric field in a single triangular/tetrahedral element is represented as

$$E=\sum_{m=1}^{3}e_mW_m,\quad E=\sum_{m=1}^{6}e_mW_m \qquad (3)$$

Here $W_m=l_m\left(L_{m_1}\nabla L_{m_2}-L_{m_2}\nabla L_{m_1}\right)$, l_m is the length of edge m that connects nodes m_1 and m_2; L_{m_1} and L_{m_2} are the simplex coordinates associated with nodes m_1 and m_2. Some other applications of edge finite elements in electromagnetics include ~~wave~~ propagation in both closed and open structures, such as metallic waveguides, open and shielded microstrip transmission lines, and optical waveguides or fibres.

Discontinuous formulation

For discontinuous formulation ([4], [3]) we consider the conservation laws

$$\frac{du}{dt}+\nabla\cdot F\left(u\right)=0,\quad x\in\Omega \tag{4}$$

At each time $t\in\left[0,T\right]$ the approximate solution is sought in the finite element space of discontinuous functions

$$V_h=\left\{v_h\in L^{\infty}:v_h\,|_T\in P\left(k\right),\forall T\in\tau_h\right.$$

Where τ_h is the triangular discretization of Ω, $P\left(k\right)$ is the local space of polynomials of degree $\leq k$. The weak formulation of (4) is

$$\frac{d}{dt}\int_T u\left(x,t\right)\psi\left(x\right)dx-\int_T F\left(u\right)\cdot\nabla\psi dx+\sum_{e\in\partial T}\int_e F\left(u\right)\cdot n\psi ds=0 \tag{5}$$

Where ψ is a smooth function, n is the outward unit normal to face e of element T. In the above expression we replaced u by its approximation u_k, the first two terms correspond to volume integrals, and the last term requires the evaluation of numerical fluxes.

<u>Sparse Structure and Bandwidth of Finite Element Matrices</u>

Meshes

The most popular element shapes employed for two and three dimensional applications are triangles and tetrahedra respectively; this is due to the fact that they are the simplest tessellation shapes for modeling arbitrary two and three dimensional geometries and they are also well suited for automatic mesh generation. To investigate the influence of the edge and element numbering in the structure of the stiffness matrix, we employ the grid generators Triangle [18] and Tetgen [19], which are two and three dimensional mesh generators that use Ruppert's Delaunay refinement.

These grid generators, in addition to providing nodes and elements, compute edges and neighbors of the elements (which is very convenient for element-defined and edge-defined finite element formulations).

We consider two simple geometries; for the two-dimensional case a circle with three interior circles removed, and for the three-dimensional case, a cylinder with three interior cylinders removed. An unstructured triangular mesh is generated by Triangle, whereas an unstructured tetrahedral mesh is obtained with Tetgen. Figures 1-2 show with a few elements the meshes considered. Also Figure 3 shows a two-dimensional view of the three-dimensional domain.

Table (1) shows the information for the meshes. Here *n*-nodes, *n*-elements and *n*-edges denote number of nodes, elements and edges respectively.

Table 1. Meshes information.

Mesh	*n*-nodes	*n*-elements	*n*-edges
2D	18686	36888	55576
3D	10720	43119	61369

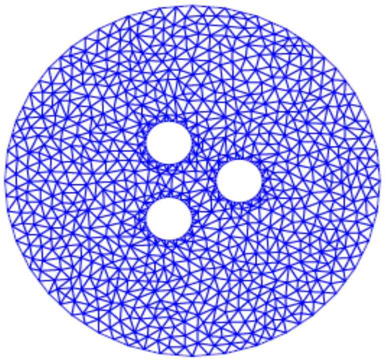

Figure 1. Two dimensional mesh for computational calculations.

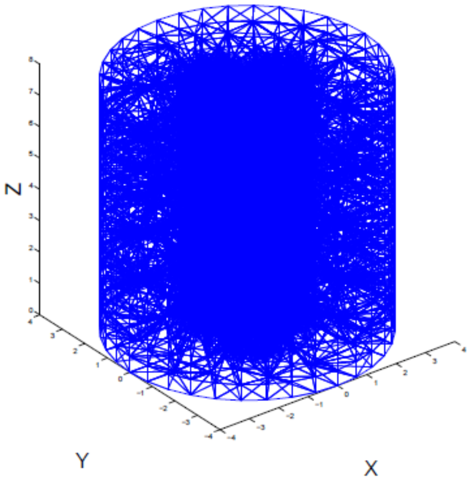

Figure 2. Three dimensional mesh for computational calculations mesh.

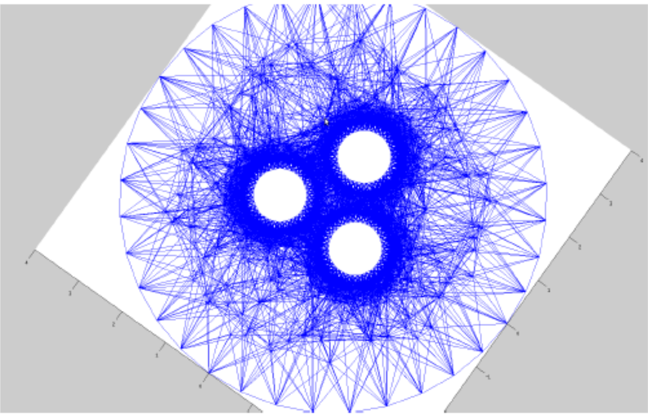

Figure 3. Two dimensional view of the three dimensional domain for computational calculations

Sparse Structure of edge-defined and element-defined formulations

In this section we discuss how the numbering of edges and elements affect the sparse structure of the stiffness matrix.

The structure of the three-dimensional case is discussed in detail and some notes are stated for the two-dimensional case. Let us introduce the following 3d sample mesh.

```
7 8 18
0 0 1
1 0 0
0 1 0
-1 0 0
0 -1 0
0 0 0
0 0 -1
1 6 2 3 -1 5 0 2 4
1 6 3 4 -1 6 0 3 1
1 6 4 5 -1 7 0 4 2
1 6 5 2 -1 8 0 1 3
3 6 2 7 -1 8 0 6 1
4 3 7 6 -1 5 7 2 0
5 4 7 6 -1 6 8 3 0
2 6 5 7 -1 7 0 5 4
1 2
1 3
1 4
1 5
1 6
5 6
2 6
```

```
3 6
4 6
6 7
2 7
3 7
4 7
5 7
2 5
2 3
3 4
4 5
```

This mesh file contains information of nodes, elements and edges. It can be obtained from the node, element, edges, and neighbors files provided by the grid generator Tetgen (two-dimensional versions of these files can be obtained by using Triangle). Its first line contains *n*-nodes, *n*-element and *n*-edges (number of nodes, elements and edges respectively), the following *n*-nodes lines contain the nodes coordinates (two or three), and the next *n*-elements lines contain the element node connectivity (nodes that define each element).

For triangles there are three nodes, a boundary marker and its three neighbors. For tetrahedral: four nodes, a boundary marker and its four neighbors. Finally *n* edges lines that contain the nodes that define each edge. Figure (4) shows the 3d simple mesh with its nodal, edge and element numberings.

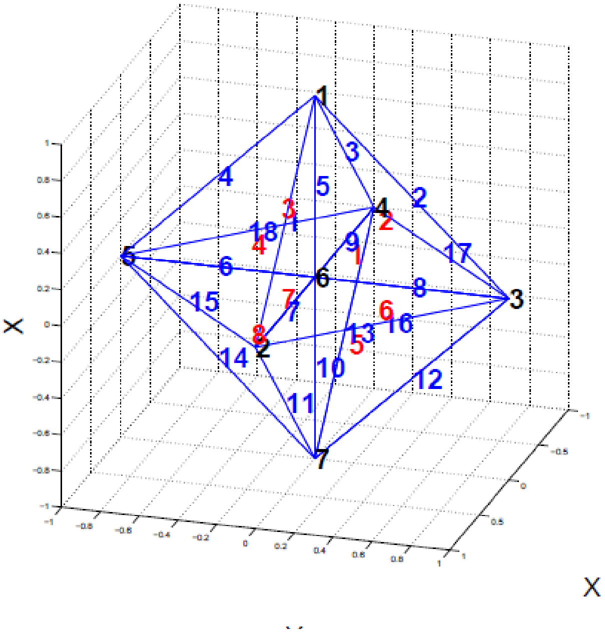

Figure 4. Three dimensional sample mesh with nodal, edge and element numberings.

In order to assemble the stiffness matrices in nodal, edge and element formulations we need the corresponding element-node, element-edge and element-neighbors connectivity Tables. The element-node and element neighbor's Tables are obtained directly from the mesh files, but the element-edge needs to be computed from the element node connectivity and the list of edges. In [14] two methods to compute an element-edge Table were discussed.

Let us start our discussion of the assembly process considering first the nodal case with linear elements. In the literature there are several works reporting how the RCM reordering has been applied to the nodal case, and we mention it here for completeness of our discussion; from the element to node connectivity Table provided by the grid generator we can get the structure of the finite element matrix K. A loop over the n elements rows of this Table says that for the row i the matrix K has nonzero elements on $K\left(table\left(i,1\right),table\left(i,2\right)\right)$, $K\left(table\left(i,1\right),table\left(i,3\right)\right)$, $K\left(table\left(i,1\right),table\left(i,4\right)\right)$, $K\left(table\left(i,2\right),table\left(i,3\right)\right)$, $K\left(table\left(i,2\right),table\left(i,4\right)\right)$ and $K\left(table\left(i,3\right),table\left(i,4\right)\right)$,where of course we have to take into account the elements of the diagonal (each node is related to itself) and the symmetries of the matrix.

Table 2. Element to node connectivity Table of our 3d sample mesh.

1 6 2 3
1 6 3 4
1 6 4 5
1 6 5 2
3 6 2 7
4 3 7 6
5 4 7 6
2 6 5 7

Table 3. Element to edge connectivity Table of our sample mesh.

1 6 1 2 1 3 2 6 2 3 3 6	5	1	2	7	16	8
1 6 1 3 1 4 3 6 3 4 4 6	5	2	3	8	17	9
1 6 1 4 1 5 4 6 4 5 5 6	5	3	4	9	18	6
1 6 1 5 1 2 5 6 2 5 2 6	5	4	1	6	15	7
3 6 2 3 3 7 2 6 2 7 6 7	8	16	12	7	11	10
3 4 4 7 4 6 3 7 6 7 4 6	17	13	9	12	10	8
4 5 5 7 5 6 4 7 6 7 4 6	18	14	6	13	10	9
2 6 2 5 2 7 5 6 5 7 6 7	7	15	11	6	14	10

Now for the edge-element formulation with linear elements, the finite element matrix can be built in a similar way as it was built in the nodal case. Here an element-to-edge Table is required. Table 3 shows the element-to-edge connectivity Table for the sample mesh. The last six columns contain the numbering of each element-edge. A loop over the *n*-elements rows of this element-edges Table says that for the row *i* the matrix K has nonzero elements on $K\left(table\left(i,1\right),table\left(i,2\right)\right)$, $K\left(table\left(i,1\right),table\left(i,3\right)\right)$, $K\left(table\left(i,1\right),table\left(i,4\right)\right)$, $K\left(table\left(i,1\right),table\left(i,5\right)\right)$, $K\left(table\left(i,2\right),table\left(i,3\right)\right)$, $K\left(table\left(i,2\right),table\left(i,4\right)\right)$, $K\left(table\left(i,2\right),table\left(i,6\right)\right)$, $K\left(table\left(i,3\right),table\left(i,5\right)\right)$, $K\left(table\left(i,3\right),table\left(i,6\right)\right)$, $K\left(table\left(i,4\right),table\left(i,5\right)\right)$, $K\left(table\left(i,4\right),table\left(i,6\right)\right)$, $K\left(table\left(i,5\right),table\left(i,6\right)\right)$, where we also consider the elements of the diagonal (each edge is related to itself) and the symmetries of the matrix.

At [14] the local number of edges is specified and also more implementation details are included.

Finally in an element wise finite element formulation (finite volume or high order discontinuous Galerkin); we need the neighbors of each element. Table 4 shows the element to node connectivity Table with their four neighbors (three for triangles) for the 3d sample mesh. Here a zero means that the element does not have a neighbor (its edge/face is on the boundary). From the element to neighbors connectivity Table we can get the structure of the finite element matrix K in the discontinuous case; a loop over the rows of the Table says that for the row *i* the matrix K have nonzero elements on $K\left(i,table\left(i,1\right)\right)$, $K\left(i,table\left(i,2\right)\right)$, $K\left(i,table\left(i,3\right)\right)$ and $K\left(i,table\left(i,4\right)\right)$ where of course we consider only the nonzero elements of the neighbors connectivity Table and again we take into account the elements of the diagonal and the symmetries of the matrix. Note that the numbering of edges and elements is given by the order as they appear at the files provided by the grid generators.

Table 4. Element to neighbors Table of our sample mesh.

16 2 3	5	0	2	4
1 6 3 4	6	0	3	1
1 6 4 5	7	0	4	2
1 6 5 2	8	0	1	3
3 6 2 7	8	0	6	1
4 3 7 6	5	7	2	0
5 4 7 6	6	8	3	0
2 6 5 7	7	0		4

Reordering and Numerical Experiments

Nodal Renumbering

Traditionally, bandwidth reduction is obtained by a reordering of nodes of the finite element mesh. Some of the most common used algorithms include: Reverse Cuthill McKee [5], and Gibbs-Poole [15]. TRIANGULATION RCM and TET MESH RCM are programs which compute the reverse Cuthill-McKee reordering for nodes in triangular/tetrahedral meshes; they are freely available in C++, FORTRAN90 and MATLAB versions [12]. The nodal renumbering scheme is directly applied to the node graph. It is desirable that grid generators incorporate node renumbering algorithms.

Edge renumbering

In ([13, 14]) two edge-numbering schemes provided by Jin [8] were tested for two- and three-dimensional implementations respectively. Numerical experiments there showed that if the node labels are close enough, a good edge-numbering scheme (low bandwidth) can be defined by using an indicator to each edge (the indicator was $i_1 \times i_2$ with i_1 and i_2 denoting the end nodes of the edge) with the array of indicators rearranged by a sorting algorithm very similar to the edge-reordering scheme used in [6]. In this work we use a different approach. Our starting point is any edge numbering, for example the one provided by Tetgen or Triangle.

To fix the ideas, let us refer to our 3d sample mesh from Figure 4. The last six (three for triangles) columns of element-to-edge connectivity Table (3) define the connectivity of what we call the *Edge-Graph*. A graphical representation of this mesh can be generated by joining the middle points of edges of the original Node-Mesh, Figure 5 shows this graph. This is our main contribution; because once we have defined the connectivity of the *Edge-Graph* we proceed to apply a renumbering algorithm as RCM.

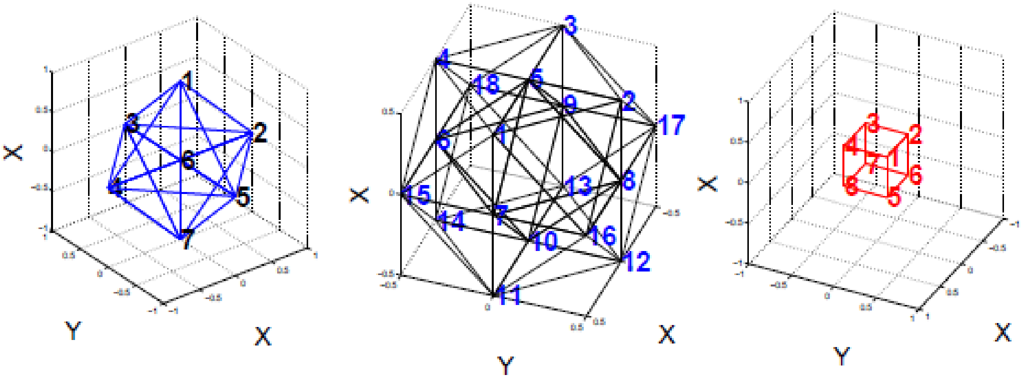

Figure 5. Nodal, Edge and Element Graphs.

Elements Renumbering

For the discontinuous formulation, in addition to the description of the elements by the connectivity of their nodes, a list of the neighbors of each element is required. Both grid generators Triangle and Tetgen provide the neighbors for triangles and tetrahedra respectively. In our 3d sample mesh, the last four (three for triangles) columns of Table (4) define the connectivity of what we call the Element-Graph. This is our main contribution because once we have defined the Element-Graph we can apply the renumbering algorithm to its connectivity Table. Notice that we can generate its graphical representation by joining the barycenters of the elements of the original node graph. Figure 5 shows this graph.

Numerical Experiments

We use meshes generated by the grid generators Triangle and Tetgen to assemble the matrices for the edge and discontinuous formulations given in section 2.

Let us make some comments about the edge-element formulation. Figure 6 shows a sparse matrix structure obtained by using the original edge-numbering provided by the grid generator [14], whereas the matrix structure showed at Figure 7 was assembled using a node-reordered mesh (RCM) followed by the S1 edge-numbering scheme, and finally the matrix structure showed in

Figure 8 was assembled using the RCM applied to the *Edge-Graph*. As mentioned in [14], scheme S1 is a good edge-numbering provided the nodes are close enough numbered, but the best result is obtained by directly applying the RCM reordering to the *Edge-graph*. In fact, after using a node-reordered mesh with the edge-numbering S1 the original bandwidth was reduced in a 332%, but when a RCM-renumbering is directly applied to the edge graph the bandwidth is reduced in a 610%.

As it has been widely reported, RCM applied to nodes of a mesh reduces the bandwidth of assembled matrices in nodal finite elements formulation. In order to obtain a bandwidth reduction in finite volume or discontinuous Galerkin formulations, we defined the Element Graph and applied the RCM reordering of elements. Figures 9 shows the sparse matrix structure for the element-defined formulations for both the original tetrahedral mesh and the graph element RCM reordered.

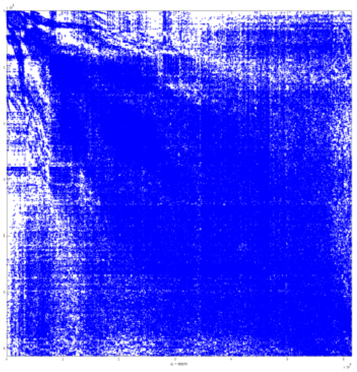

Figure 6. Original edge formulation, as provided by the grid generator.

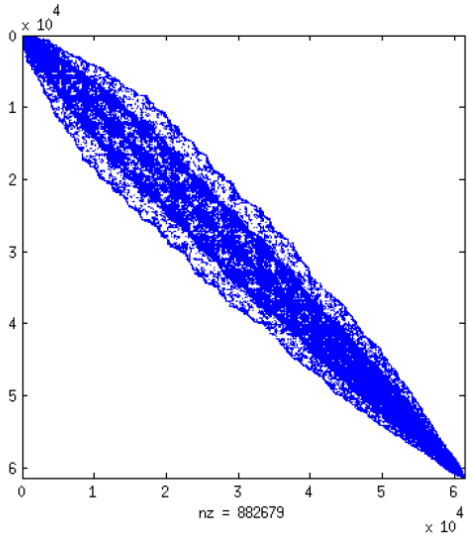

Figure 7. Edges formulation, RCM ordered nodes with S1 numbering scheme.

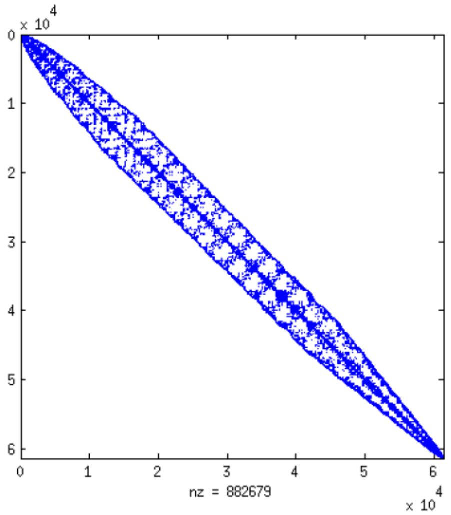

Figure 8. Edges formulation, Graph edges ordered by RCM.

We observed similar results in two and three dimensions for nodes and elements.

Figure 9 shows the sparse structure of the matrix of the element wise finite element formulation for the original numbering of the elements and the RCM reordering of the elements graph.

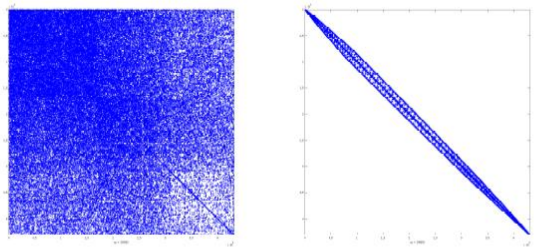

Figure 9. Element formulation, original and RCM renumbered.

In time domain formulation with edge-elements to simulate transient electromagnetic fields in 3d diffusive earth media [20] and high-order Navier-Stokes simulations using a Discontinuous Galerkin Method [17] both of them reduce to a sparse-matrix-times-vector product operation employed by the time solvers. Thus, a simple OpenMp code was implemented to multiply the assembled matrix sparse storage times a vector, and the execution time is reported.

```
void SPARSE::Matrixvector(real vect[],real vectprod[])
{
long long int k;
unsigned int i;
int chunnk=1000;
{
#pragma omp for private(k,i) schedule(static,chunk)
for ( i = 0; i ¡ dim; i++)
{
vectprod[i]=0.0;
for ( k = IA[i]; k ¡ IA[i+1]; k++)
vectprod[i] += AA[k-1]*vect[JA[k-1]-1];
}
}}
```

Figure 10 shows the execution time of the sparse-matrix-times-vector product as a function of the number of iterations. Here original refers to edge-numbering provided by the grid generator whereas reordered refers to the reordered edge graph. For iterations between 200 and 600 the execution time is reduced by an average of 27.5%. The OpenMp code sparse-matrix-times-vector products were performed in an Intel Core CPU 2.90 GHz 8 processors workstation. Figure 11 reports execution times for both the originally numbered and the reordered element-graphs corresponding to the element-wise formulation. It is desirable that grid generators include the RCM renumbering for nodes, edges and elements in order to avoid the post processing of the mesh or the reordering of the assembled matrix in the different finite element formulation. The bandwidth reduction obtained by edge or element reordering reduces the cache misses, improving the execution time of the sparse-matrix-vector product which is the main kernel of iterative solvers as BCG, QMR, etc. [11]

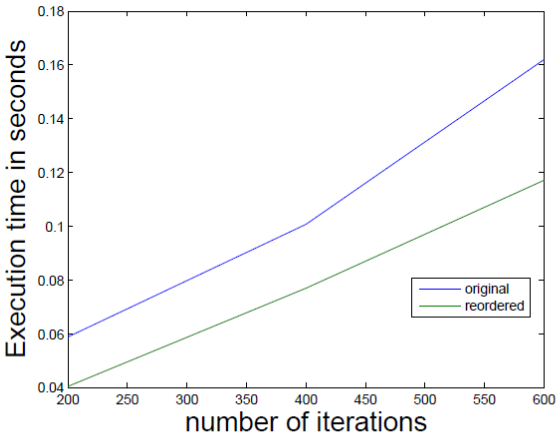

Figure 10. Matrix times vector execution time as a function of the number of iterations, edge elements formulation.

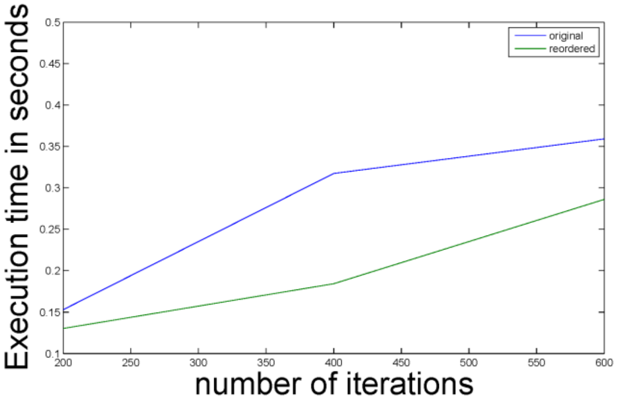

Figure 11. Matrix times vector execution time as a function of the number of iterations, elements formulation.

Conclusions

Bandwidth reduction of matrices produced by nodal edge and element-defined finite element formulations were obtained.

To apply a renumbering scheme like RCM (originally designed for nodal) we have defined the *N* and Element-Graph. The optimal mesh obtained (node-reordered, edge-reordered, and element-reordered) can be very valuable. Some methods, for example those that combine nodal and edge-defined finite element formulations [9] or those that combine nodal finite element and finite volume methods [16] could directly benefit from these reorderings. We suggest that grid generators should incorporate nodal, edge and element renumbering of the produced meshes in order to reduce bandwidth of assembled finite element matrices, avoiding the preprocessing of the mesh or the post processing of the assembled finite element matrix (sometimes quite difficult because the matrix is assembled at execution time or distributed over different processors). Moreover, the bandwidth minimization renumbering scheme that has been discussed reduces the cache misses improving the execution time in the sparse-matrix-times-vector product implemented on OpenMp. Matrix-vector multiplication is one of the main kernels in the iterative solvers. In this work we have discussed first order nodal and (CT/LN) edge finite elements, and a future work is due to investigate high order nodal and edge elements beside extended stencil finite volume formulations (extended neighborhoods for high order finite volume formulations).

References

J. P. Bastos and N. Sadowski. Magnetic Materials and 3D Finite Element Modeling. CRC press, 2014.

D. A. Burgess and M. Giles. Renumbering unstructured grids to improve performance of codes on hierarchical memory machines. Advances in Engineering Software, 28(3):189–201, 1997.

P. Castillo. A review of the local discontinuous Galerkin (ldg) method applied to elliptic problems. Journal Applied Numerical Mathematics, Selected papers from the first Chilean workshop on numerical analysis of partial differential equations, 56(10):1307–1313, 2006.

B. F. Cockburn B., Arnold D.N. and D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. SIAM Journal of Numerical Analysis, 39(2):1749–1779, 2002.

E. Cuthill and J.McKee. Reducing the bandwidth of sparse symmetric matrices. ACM Annual Conference Proceedings of the 24th national conference, pages 157–172, 1969.

D. K. K. D. E. Keyes, W. C. Groop and B. F. Smith. Performance modeling and tunning of an unstructured mesh cfd application. Conference on High Performance Networking and Computing, Proceedings of the 2000 ACM/IEEE conference on Supercomputing, (34), 2000.

S. G. J. Saltz and D. J.Mavriplis. The design and implementation of a parallel unstructured euler solver using software primitives. AIAAA Journal, 1992.

M. Jian. The Finite Element Method in Electromagnetics. IEEE, John Wiley and sons inc, 2014.

V. H.-H. Nguye-Thoi, Phung-Van P. and L. Le-Anh. An edge-based smoothed finite element method (es-fem) for dynamic analysis of 2d fluid-solid interaction problems. KSCE Journal of Civil Engineering, 3(19):641–650, 2015.

T. P. Noreika A. Electromagnetic Field Modeling Using Edge Finite Elements. International Biennial Baltic Electronics Conference, 2008.

D. M. V. P. J. m. C. Octavio Castillo, Josep de la Puente. A parallel tool for numerical approximation of 3d electromagnetic surveys in geophysics. Computación y sistemas, 1(20):29–39, 2016.

R. R. C. M. Ordering. http://people.sc.fsu.edu.

G. Ortigoza. Triangular grid generators for the eigenvalue calculation with edge elements. Revista Mexicana de Física, (2):154–160, 2009.

G. Ortigoza. Tetrahedral grid generators for the eigenvalue calculation with edge elements. Revista Computacion y Sistemas IPN, (14):5–16, 2010.

N. G. P. K. Stockmeyer and W. Poole. An algorithm for reducing the bandwidth and profile of a sparse matrix. SIAM Journal on Numerical Analysis, 13(2):236–250, 1976.

K. Paluszny A. and M. Hohmeyer. Hybrid finite elementfinite volume discretization of complex geologic structures and a new simulation workflow demonstrated on fractured rocks. Geofluids, (7):186–208, 2007.

P.-O. Persson. High-order navier-stokes simulations using a sparse line-based discontinuous galerkin method. 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. Nashville, Tennessee., 2012.

J. R. Schewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. Applied computational geometry, 1148:203–222, 1996.

H. Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. Journal ACMTransactions onMathematical Software, 41(2), 2015.

E. S. Um, J. M. Harris, and D. L. Alumbaugh. An iterative finite element time-domain method for simulating threedimensional electromagnetic diffusion in earth. Geophysical Journal International, 190(2):871–886, 2012.

T. R. Zienkiewicz O. C. and J. Zhu. The Finite Element Method Set. Butterworth-Heinemann Elsevier, 2013.