Excessive Memory Usage of the ELLPACK Sparse Matrix Storage Scheme throughout the Finite Element Computations

Gökay AKINCI[1], *A. Egemen YILMAZ*[1], *Mustafa KUZUOĞLU*[2]

[1] Dept. of Electrical and Electronics Engineering, Ankara University, Ankara, Turkey
[2] Dept. of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey

gakinci@ankara.edu.tr, aeyilmaz@eng.ankara.edu.tr, kuzuoglu@metu.edu.tr

Abstract. *Sparse matrices are occasionally encountered during solution of various problems by means of numerical methods, particularly the finite element method. ELLPACK sparse matrix storage scheme, one of the most widely used methods due to its implementation ease, is investigated in this study. The scheme uses excessive memory due to its definition. For the conventional finite element method, where the node elements are used, the excessive memory caused by redundant entries in the ELLPACK sparse matrix storage scheme becomes negligible for large scale problems. On the other hand, our analyses show that the redundancy is still considerable for the occasions where facet or edge elements have to be used.*

Keywords

Finite element method, sparse matrix, edge elements, computational electromagnetics, ELLPACK.

1. Introduction

The Finite Element Method, which has been originally developed for static problems of structural mechanics and initially used by mechanical and civil engineers, is one of the most widely used numerical methods by the scientific community. It was first formulated in 1940s by Courant after a discussion regarding the versatility of piecewise approximations. In the 1950s, Argyris began putting together the many mathematical ideas (domain partitioning, assembly, boundary conditions, etc.) that form the basis of the Finite Element Method for aircraft structural analysis.

In its original form, the method depends on representation and approximate evaluation of continuous scalar functions at the corners of the subdomains (referred as “element”s) of the whole problem domain. Even though this formulation proves to be sufficient in handling most problems in the structural mechanics, it was compulsory to extend and generalize the method in order to compute vector functions. This yielded the definitions of the so-called edge and facet elements. Thanks to the definition of edge and facet elements, it is possible to solve electromagnetic scattering and radiation problems as well as eddy current problems by means of the Finite Element Method. Tab. 1 enlists which element type is the most suitable one for evaluation of the major functions in electromagnetics.

A major advantage of the Finite Element Method is that it yields sparse matrices throughout the solution process. By means of special storage schemes, it is possible to store and solve very large scale matrices and matrix equations, respectively.

	Node Elements	Edge Elements	Facet Elements	Volume Elements
Types of Represented Functions	Scalar	Vector	Vector	Scalar
Represen-tation Capability of Continuity	Total	Tangential Component	Normal Component	None
Physical Types of Represented Functions	Scalar Potential	Fields, Vector Potentials	Fluxes, Vector Densities	Scalar Densities
Examples from Electromagn etic Theory	Scalar Electric Potential (V or ϕ)	Vector Magnetic Potential **A**, Electric Field Intensity **E**, Magnetic Field Intensity **H**	Magnetic Flux Density **B**, Electric Field Density **D**, Current Density **J**	Charge Density (ρ)

Tab. 1. Element types and their representation capabilities.

ELLPACK sparse matrix storage scheme [1–3] is one of the most widely used and preferred schemes in practice due to its implementation ease. This scheme stores a matrix **A** with size $n \times n$ (1),

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 & a_{13} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ 0 & a_{32} & 0 & a_{34} & a_{35} \\ 0 & 0 & a_{43} & a_{44} & 0 \\ 0 & a_{52} & 0 & a_{54} & a_{55} \end{bmatrix} \tag{1}$$

by means of the following two matrices (2),

$$data = \begin{bmatrix} a_{11} & a_{13} & * \\ a_{21} & a_{22} & * \\ a_{32} & a_{34} & a_{35} \\ a_{43} & a_{44} & * \\ a_{52} & a_{54} & a_{55} \end{bmatrix}, \quad indices = \begin{bmatrix} 1 & 3 & * \\ 1 & 2 & * \\ 2 & 4 & 5 \\ 3 & 4 & * \\ 2 & 4 & 5 \end{bmatrix} \qquad (2)$$

By this definition, both data and indices matrices are of the sizes $n \times m$; where m is the maximum number of nonzero entries in a row of **A**.

The entries shown with symbols "*" are meaningless and redundant values held in the memory. These redundancies are nothing but the aspects referred as the "generosity" of the ELLPACK sparse matrix storage scheme.

We'll carry out an analysis of how generous the ELLPACK sparse matrix storage scheme behaves during the finite element computations [4-6]; in other words, for various element types of various shapes in 2-Dimension (2D) (Fig. 1 and Fig. 2), we'll try to compute the ratio of redundant (or meaningless entries shown with "*") entries in the whole matrix. We'll perform our analysis for node, edge, facet and volume elements for linear triangular [7] and quadrilateral [8] elements.

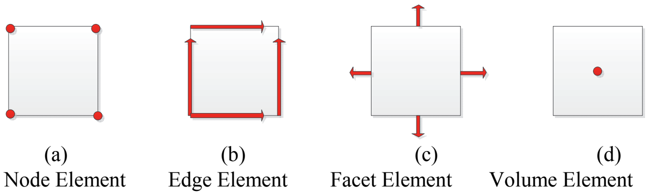

Fig. 1. Quadrilateral elements.

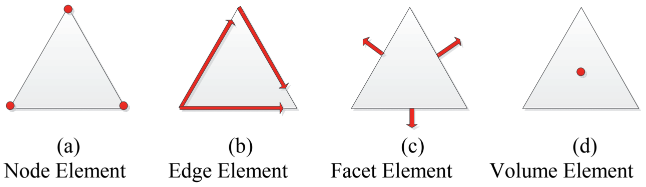

Fig. 2. Triangular elements.

Assume that, we have a problem domain as shown as Fig. 3(a). In case the problem domain is homeomorphic to a rectangle; this domain in the xy-plane can be mapped to a rectangular domain in a uv-plane, as seen in Fig. 3(b). Hence, we'll carry out our further analysis in the uv-plane, without loss of generality.

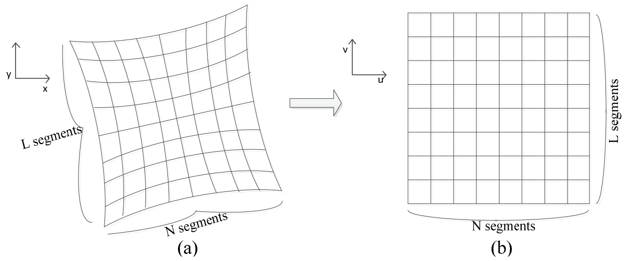

Fig. 3. Problem domain mapped to a regular rectangle in the uv-plane.

2. Quadrilateral Elements

First of all, we'll perform our analysis for quadrilateral node, edge, facet and volume elements, separately.

2.1 Quadrilateral Node Elements

If we consider that this mesh consists of node elements, then the total number of unknowns (i.e. number of nodes) will be $K_{quad\text{-}node} = (N+1) \times (L+1)$, which means that the size of the global system matrix obtained throughout the finite element solution will be $K_{quad\text{-}node} \times K_{quad\text{-}node}$.

Let us consider how much memory the ELLPACK sparse matrix storage scheme will allocate for storing this matrix. For this purpose, first, we have to consider the maximum number of nonzero entries in one row of this matrix.

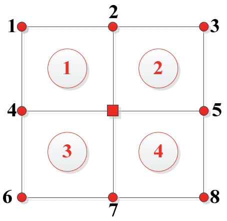

Fig. 4. A node shared by 4 elements.

As seen in Fig. 4, a node can maximally be shared by 4 elements. This means that this node will have 8 neighboring nodes; which means that the row corresponding to this node will have 8 off-diagonal nonzero entries in addition to 1 diagonal nonzero entry. Namely, in this case, the maximum number of nonzero entries in one row will be 9.

Hence data and indices matrices of ELLPACK will be of size $K_{quad\text{-}node} \times 9$. This means that each matrix will require $9 \times (N+1) \times (L+1)$ entries in the memory.

2.2 Quadrilateral Edge Elements

If we consider that this mesh consists of edge elements, then the total number of unknowns (i.e. number of edges) will be $K_{quad\text{-}edge} = 2NL + N + L$, which means that the size of the global system matrix obtained throughout the finite element solution will be $K_{quad\text{-}edge} \times K_{quad\text{-}edge}$.

As seen in Fig. 5, an edge can be shared at most by 2 elements. The row corresponding to this edge will have 6 off-diagonal nonzero entries in addition to 1 diagonal nonzero entry. Namely, in this case, the maximum number of nonzero entries in one row will be 7.

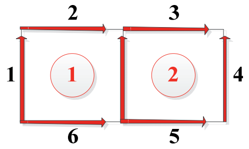

Fig. 5. An edge shared by 2 elements.

Hence data and indices matrices of ELLPACK will be of size $K_{quad\text{-}edge} \times 7$. This means that each matrix will require $7 \times K_{quad\text{-}edge} = 7 \times (2NL + N + L)$ entries in the memory.

2.3 Quadrilateral Facet Elements

All computations regarding the number of unknowns and number of nonzero entries will be equal to that of the edge elements.

Hence data and indices matrices of ELLPACK will be of size $K_{quad\text{-}facet} \times 7$. This means that each matrix will require $7 \times K_{quad\text{-}facet} = 7 \times (2NL + N + L)$ entries in the memory.

2.4 Quadrilateral Volume Elements

If we consider that this mesh consists of volume elements, then the total number of unknowns (i.e. number of centroids) will be $K_{quad\text{-}volume} = N \times L$, which means that the size of the global system matrix obtained throughout the finite element solution will be $K_{quad\text{-}volume} \times K_{quad\text{-}volume}$.

A centroid is owned by one element only; it is not shared. Hence,

- Only diagonal terms in the matrix,
- Only NL entries in the memory.

(ELLPACK or any other sparse matrix storage scheme is unnecessary. It is sufficient to compute and store the diagonal terms in an ordinary array)

3. Triangular Elements

We'll perform our analysis for triangular node, edge, facet and volume elements.

3.1 Triangular Node Elements

We can obtain a triangular element mesh from a quadrilateral mesh in a straightforward manner; which will yield $2NL$ elements. No new nodes are introduced, the total number of unknowns (number of nodes) will be identical to the quadrilateral case; that is $K_{tri\text{-}node} = K_{quad\text{-}node} = (N + 1) \times (L + 1)$.

A node can be shared at most by 6 elements as seen in Fig. 6. The matrix row corresponding to this node will have 6 off-diagonal nonzero entries in addition to 1 diagonal nonzero entry. Namely, in this case, the maximum number of nonzero entries in one row will be 7.

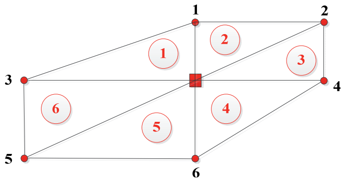

Fig. 6. A node shared by 6 elements.

Hence each matrix will require $7 \times K_{tri\text{-}node} = 7 \times (N + 1) \times (L + 1)$ entries in the memory.

3.2 Triangular Edge Elements

While conversion from the quadrilateral mesh to triangular mesh NL new edges are introduced; hence,

$$\begin{aligned} K_{tri-egde} &= K_{quad-edge} + NL, \\ K_{tri-egde} &= 2NL + N + L + NL = 3NL + N + L. \end{aligned} \tag{3}$$

As seen in Fig. 7, an edge can be shared at most by 2 elements. The matrix row corresponding to this edge will have 4 off-diagonal nonzero entries in addition to 1 diagonal nonzero entry. Namely, in this case, the maximum number of nonzero entries in one row will be 5.

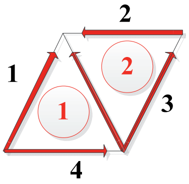

Fig. 7. An edge shared by 2 elements.

Hence each matrix will require $5 \times K_{tri\text{-}edge} = 5 \times (3NL + N + L)$ entries in the memory.

3.3 Triangular Facet Elements

All computations regarding the number of unknowns and number of nonzero entries will be equal to that of the edge elements and each matrix will require $5 \times K_{tri\text{-}facet} = 5 \times (3NL + N + L)$ entries in the memory, too.

3.4 Triangular Volume Elements

If we consider that this mesh consists of triangular volume elements, then the total number of unknowns (i.e. number of centroids) will be $K_{tri\text{-}volume} = 2NL$.

A centroid is owned by one element only; it is not shared. Hence,

- Only diagonal terms in the matrix,
- Only $2NL$ entries in the memory.

4. Exact Number of Nonzero Entries

We'll determine exact number of nonzero entries for each element type.

4.1 Quadrilateral Node Elements

There are 3 types of nodes inside the mesh.

Type 1 node: A node owned by only one element at the corner of the problem domain. There are 4 nodes of this sort. For such nodes, the corresponding row will have only

3 off-diagonal and 1 diagonal nonzero entries; i.e. 4 nonzero entries.

Type 2 node: A node owned by 2 elements at the border of the problem domain. There are ($2N + 2L$ - 4) nodes of this sort. For such nodes, the corresponding row will have 5 off-diagonal and 1 diagonal nonzero entries; i.e. 6 nonzero entries.

Type 3 node: A node owned by 3 elements. There are (N - 1) × (L - 1) nodes of this sort. For such nodes, each row will have 9 nonzero entries. Hence, the actual number of nonzero entries is equal to,

$$\begin{aligned}&[4\times 4]+[6\times(2N+2L-4)]+[9\times(N-1)(L-1)]\\&=9NL+3N+3L+1.\end{aligned} \tag{4}$$

ELLPACK stores 9 × (N + 1) × (L + 1) entries and redundancy rate is

$$\begin{aligned}r&=\frac{\text{number of redundant entries}}{\text{number of stored entries}},\\r&=\frac{6N+6L+8}{9NL+9N+9L+9}.\end{aligned} \tag{5}$$

4.2 Quadrilateral Edge Elements

There are 2 types of edges.

Type 1 edge: An edge shared by only 1 element at the border of the problem domain. There are NL edges of this sort. Rows corresponding to these edges have 3 off-diagonal and 1 diagonal nonzero entries; i.e. 4 nonzero entries.

Type 2 edge: An edge shared by 2 elements. There are $NL + N + L$ edges of this sort. Rows corresponding to these edges have 6 off-diagonal and 1 diagonal nonzero entries; i.e. 7 nonzero entries. Hence, the exact number of nonzero entries will be $11NL + 7N + 7L$.

ELLPACK stores $14NL + 7N + 7L$ entries and redundancy rate is

$$r=\frac{3NL}{14NL+7N+7L}. \tag{6}$$

4.3 Quadrilateral Facet Elements

All computations regarding the number of unknowns, number of nonzero entries and redundancy rate will be equal to that of the edge elements.

4.4 Triangular Node Elements

There are 4 types of nodes inside the mesh.

Type 1 node: A node owned by only one element at the corner of the problem domain. There are 2 nodes of this sort. For such nodes, the corresponding row will have only 2 off-diagonal and 1 diagonal nonzero entries; i.e. 3 nonzero entries.

Type 2 node: A node shared by only 2 elements at the border of the problem domain. There are 2 nodes of this sort. For such nodes, the corresponding row will have only 3 off-diagonal and 1 diagonal nonzero entries; i.e. 4 nonzero entries.

Type 3 node: A node shared by 3 elements. There are ($2N + 2L - 4$) nodes of this sort. For such nodes, the corresponding row will have only 4 off-diagonal and 1 diagonal nonzero entries; i.e. 5 nonzero entries.

Type 4 node: A node shared by 6 elements. There are ($NL - N - L + 1$) nodes of this sort. For such nodes, the corresponding row will have only 6 off-diagonal and 1 diagonal nonzero entries; i.e. 7 nonzero entries. Hence, the exact number of nonzero entries will be $7NL + 3N + 3L + 1$.

ELLPACK stores $7NL + 7N + 7L + 7$ entries and redundancy rate is

$$r=\frac{4N+4L+6}{7NL+7N+7L+7}. \tag{7}$$

4.5 Triangular Edge Elements

There are 2 types of edges inside the mesh.

Type 1 edge: An edge owned by only one element at the border of the problem domain. There are $2NL$ edges of this sort. For such nodes, the corresponding row will have only 2 off-diagonal and 1 diagonal nonzero entries; i.e. 3 nonzero entries.

Type 2 edge: An edge owned by 2 elements. There are $NL + N + L$ edges of this sort. For such nodes, the corresponding row will have only 4 off-diagonal and 1 diagonal nonzero entries; i.e. 5 nonzero entries. Hence, exact number of nonzero entries will be $11NL + 5N + 5L$.

ELLPACK stores $15NL + 5N + 5L$ entries and redundancy rate is

$$r=\frac{4NL}{15NL+5N+5L}. \tag{8}$$

4.6 Triangular Facet Elements

All computations regarding the number of unknowns, number of nonzero entries and redundancy rate will be equal to that of the edge elements.

5. Results and Their Implications for Real-Life Problems

5.1 Wrapping-Up the Results of the Analyses

Summarizing our analysis results in Fig. 8 and Fig. 9, we observe a very interesting aspect of the ELLPACK

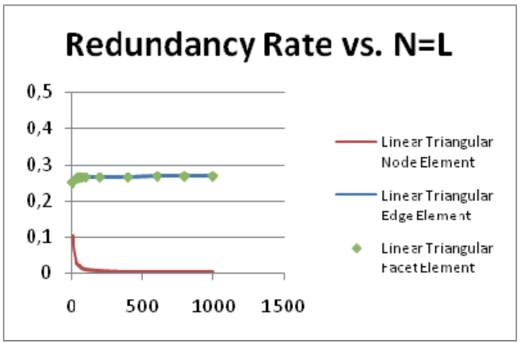

Fig. 8. Redundancy rate vs. different values of $N = L$ (for various types of triangular elements).

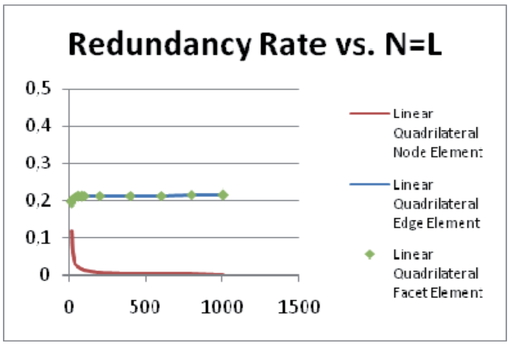

Fig. 9. Redundancy rate vs. different values of $N = L$ (for various types of quadrilateral elements).

sparse storage scheme. For node elements, as a matter fact, the redundancy of the scheme becomes quite negligible (at the order of 0.001%) for very large scale problems. This means that the scheme can be confidently implemented and used in case of node elements. For volume elements, the scheme can also be used safely since there is no redundancy.

On the other hand, for the edge and facet elements, which are quite frequently used in electromagnetic scattering problems, the redundancy rate converges to 26.6% for triangular edge/facet elements; and to 21.5% for quadrilateral edge/facet elements. This means that it should be noted that there will be a considerable amount of redundant resource usage of ELLPACK storage scheme for very large scale electromagnetic scattering problems.

5.2 Real Life Problems

5.2.1 Bistatic Radar Cross Section Computation for Isolated Scatterers

In order to give an idea about the situation in real life electromagnetic scattering problems, let us now consider some conventional problems modeled in two-dimensional space.

First, we will consider an isolated scatterer extending to infinity (or so long in that direction such that this assumption is valid) in one direction. For such problems, the main aim is to obtain the bistatic Radar Cross Section (RCS) of the scatterer. For this purpose, an incident plane wave of an arbitrary direction at a specific frequency (or wavelength λ) is assumed onto the scatterer, and the scattered field caused by the scatterer at all directions (towards any elevation and azimuth angle) is computed by means of the Finite Element Method. As seen in Fig. 10, the scatterer can be thought as occupying a space modeled with $N_1 \times L_1$ elements. Here, it should be noted that it is common practice to choose the element size no more than 0.1λ in case of linear elements for an acceptable level of solution accuracy.

On the other hand, since the FEM is not suitable for direct use in open problem domains, the “Absorbing Boundary Conditions (ABCs)”, particularly the so-called Perfectly Matched Layers (PMLs) [9] shall be applied to such problems in order to terminate the computational domain. The most common approach for PML implementation is the complex-coordinate stretching [10], and this method requires attaining a thickness of at least 3 elements to the PML for the convergence of the solution.

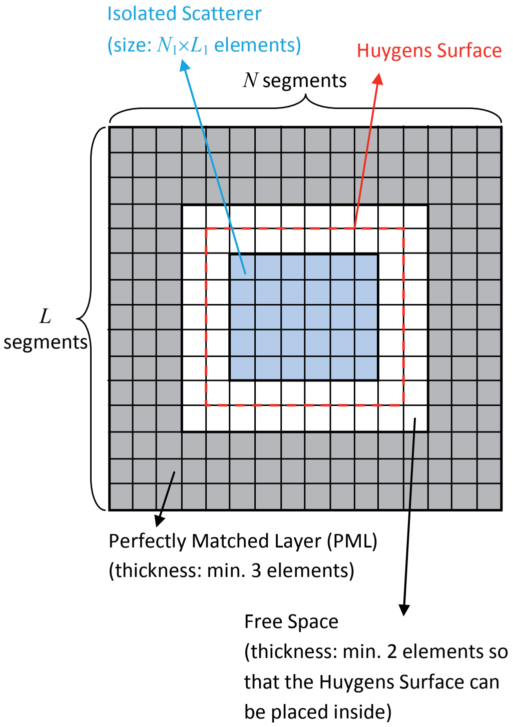
Isolated Scatterer
(size: $N_1 \times L_1$ elements)
Huygens Surface
N segments
L segments
Perfectly Matched Layer (PML)
(thickness: min. 3 elements)
Free Space
(thickness: min. 2 elements so that the Huygens Surface can be placed inside)

Fig. 10. Sample 2-D problem modeled via the Finite Element Method for computation of the RCS of an isolated scatterer.

Moreover, RCS computation requires far-field solutions. In order to preserve the computational resources, far-field is not computed directly. This is rather achieved by applying Huygens’ Surface Equivalence Principle. For this purpose, a closed surface (over which the near field is known) totally inside the free space shall be constructed. By taking the necessary surface integral over this surface [11], [12], it is possible to compute the far-field from the near-field without extending the computational domain dramatically. Certainly, the cost of application of Huygens’

Surface Equivalence Principle is to surround the scatterer with at least 2-element free space in order to be able to choose a closed surface totally residing in free space, as seen in Fig. 10.

Summing up all the issues listed in the previous paragraphs, let us now consider an isolated scatterer with size $3\lambda \times 3\lambda$, i.e. a typical problem for which the Finite Element Method can be applied. First, let us assume that quadrilateral edge elements are used. For such a case, considering an element size of no more than 0.1λ, the scatterer shall be modeled with at least $N_1 \times L_1 = 30 \times 30$ elements. The space occupied by this scatterer shall be surrounded by (i) free space with a minimum thickness of two elements, and (ii) PML with a minimum thickness of two elements. This yields a computational domain of at least $N{\times}L = [30 + (2{\times}2) + (2{\times}3)] \times [30 + (2{\times}2){+} (2{\times}3)] = 40{\times}40$ elements. The analysis held out Sections 3 and 4 as well as the results shown in Fig. 9 show that for this very typical scattering problem (which can be considered as a mid-scale problem), the redundancy rate of the ELLPACK sparse storage scheme is about 20.9 % if modeled by quadrilateral edge elements.

Now, let us assume that the same problem be modeled via linear triangular edge elements. With all considerations listed above, the computational domain shall be divided into at least $2 \times N \times L = 2{\times}[30 + (2 \times 2) + (2 \times 3)] \times [30 + (2 \times 2) + (2{\times}3)] = 2{\times}40{\times}40$ elements. Again, the analysis held out Sections 3 and 4 as well as the results shown in Fig. 9 show that for the same scattering problem, the redundancy rate of the ELLPACK sparse storage scheme is about 26.2 % if modeled by triangular edge elements.

In case the scatterer of interest is a Perfect Electric Conductor (PEC), the volume occupied by the scatterer can be excluded from the computational domain as seen in Fig. 11. The PEC assumption imposes the fact that the total electric and magnetic field inside the scatterer is exactly zero, and it is not required to spend any effort for that particular volume. Hence, exclusion of the relevant volume can be made conveniently in order to decrease the number of elements and the number of unknowns.

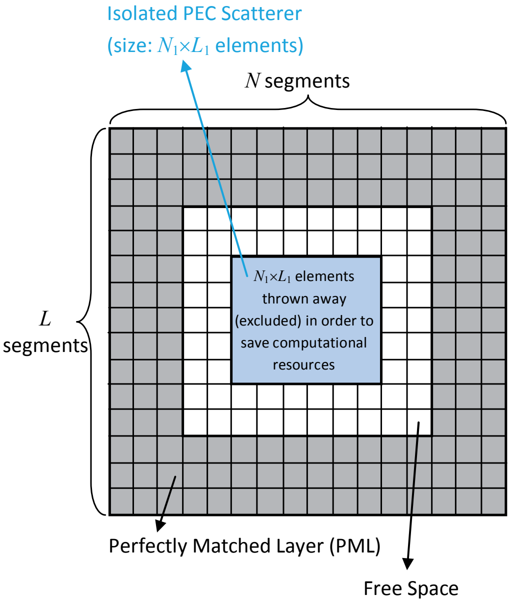
Isolated PEC Scatterer
(size: $N_1{\times}L_1$ elements)
N segments
L segments
$N_1{\times}L_1$ elements thrown away (excluded) in order to save computational resources
Perfectly Matched Layer (PML)
Free Space

Fig. 11. The same 2-D problem for which the volume occupied by the isolated PEC scatterer is excluded.

In this case, the computational domain contains at least $(40 \times 40) - (30 \times 30) = 700$ elements when quadrilateral edge elements are used. Again, the analysis held out Sections 3 and 4 as well as the results shown in Fig. 9 show that for this case, the redundancy rate of the ELLPACK sparse storage scheme is about 20.9 %.

Similarly, if triangular edge elements are used for the PEC scatterer, this will yield at least $2 \times [(40 \times 40) - (30 \times 30)] = 2 \times 700 = 1400$ elements. Eventually, this will yield a redundancy rate of 26.0 % for the ELLPACK sparse storage scheme.

5.2.2 Scattering Parameters' Computation for the Periodic Structures

Another common application of the Finite Element Method as regards the real life electromagnetic scattering problems is nothing but the computation of the scattering parameters for periodic structures. A singly-periodic structure, which can be modeled via 2-D finite element formulation, can be considered as an infinite series of a particular structure cascaded to each other. For such problems, the main aim is to obtain the reflection (the so-called S11 parameter) and the transmission (the so-called S21 parameter) characteristics of the periodic structure. For this purpose, usually a normal incident plane wave at a specific frequency (or wavelength λ) is assumed onto one side of the scatterer, and the scattered fields caused by the scatterer at forward and backward directions (towards any elevation and azimuth angle) are computed by means of the Finite Element Method. Analyses of such structures are quite crucial in real life especially at the design process of Frequency Selective Surfaces (FSSs), Radomes (Radar Domes) and Electromagnetic Band-Gap Structures (EBGs).

As seen in Fig. 12, the periodic structure can be thought as occupying a space modeled with $N_1 \times L_1$ elements. Again, the criterion for the element size to be no more than 0.1λ (in case of linear elements) is still valid for an acceptable level of solution accuracy. PMLs shall be applied to such problems in order to terminate the computational domain, as well. Similarly, attaining a thickness of at least 3 elements to the PML for the convergence of the solution is required. The far field is computed by means of Huygens' Surface Equivalence Principle. Eventually, the cost of application of Huygens' Surface Equivalence Principle is to place at least 2-element free space above and below the free periodic structure in order to be able to choose two surfaces (the former for computation of the scattered field and the latter for the transmitted field) totally residing in free space, as seen in Fig. 12.

The difference in modeling the periodic structures is at the step of imposing the so-called "Periodic Boundary

Conditions” in order to apply Floquet’s Theorem, by which the magnitude of the total fields at the Side Walls are set and forced to be identical, and the phase difference of the total fields are set to be constant [11], [12]. By this manner, it is possible and sufficient to carry out the computations only at a single period of the periodic structure (i.e. it is not necessary to carry out the computations over numerous periods, which would dramatically increase the computational cost).

Again, let us consider a periodic structure of which the cross section can be modeled via $N \times L_1$ quadrilateral elements as seen in Fig. 12. Assume that the thickness of the periodic structure is about 0.2λ (i.e. 2 elements). This is a valid assumption for practical cases since the FSSs, Radomes or EBGs are required to be as thin as possible. Also, it is common practice to adjust the period of such structures more than λ, particularly about 2λ (i.e. 20 elements).

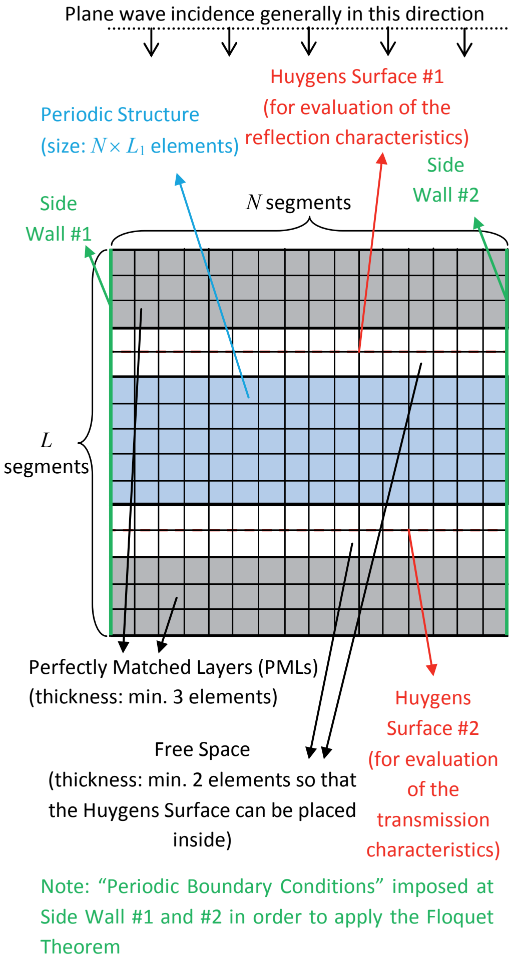
Plane wave incidence generally in this direction
Huygens Surface #1
(for evaluation of the
reflection characteristics)
Periodic Structure
(size: $N \times L_1$ elements)
Side
Wall #2
Side
Wall #1
N segments
L
segments
Perfectly Matched Layers (PMLs)
(thickness: min. 3 elements)
Huygens
Surface #2
(for evaluation
of the
transmission
characteristics)
Free Space
(thickness: min. 2 elements so that
the Huygens Surface can be placed
inside)
Note: “Periodic Boundary Conditions” imposed at
Side Wall #1 and #2 in order to apply the Floquet
Theorem

Fig. 12. Sample 2-D problem modeled via the Finite Element Method for computation of the scattering parameters (transmission and reflection characteristics) for a singly periodic structure.

Hence, the computational domain will contain at least $N \times L = 20 \times [2 + (2 \times 2) + (2 \times 3)] = 20 \times 12$ elements if quadrilateral elements are used. From the results of the previous sections and Fig. 9, it is apparent that the redundancy rate of the ELLPACK storage scheme will be about 19.6 % for such a typical problem.

If the same problem is modeled by means of triangular edge elements, there will be at least $2 \times N \times L = 2 \times 20 \times [2 + (2 \times 2) + (2 \times 3)] = 2 \times 20 \times 12$ elements, which will yield a redundancy rate of 25.9 % for the ELLPACK storage scheme.

6. Conclusions

Due to its implementation ease, ELLPACK sparse matrix storage scheme is one of the most commonly used schemes in practice. On the other hand it has some amount of excessive memory usage. Even though this amount seems to be negligible at first glance, as demonstrated in the previous sections, it might be in the order of 1/5 in case of edge elements.

Particularly, throughout the finite element modeling and solution of the very large-scale electromagnetic scattering problems, ELLPACK sparse storage scheme have a redundancy rate about 26.6 % for triangular edge/facet elements; and about 21.5 % for quadrilateral edge/facet elements.

Moreover, the analyses held out in Section 5.2 demonstrate that for mid-scale real life problems (i.e. in which the electrically mid-size isolated scatterers or periodic structures are in concern) the redundancy rates are still quite significant: about 25.9 to 26.2 % for triangular edge/facet elements; and about 19.6 to 20.9 % for quadrilateral edge/facet elements.

Once again, it should be noted that all our analyses address any 2-D problem domain which is homeomorphic to a rectangle. Hence, the results can be generalized to any structured but non-uniform mesh consisting of triangular or quadrilateral mesh. On the other hand, what happens for the case of unstructured mesh (i.e. mesh with nodal discontinuities requiring the so-called “mesh refinement”), is still an open question. Even though it might not be possible to come up with generalized concrete formulation as in the structured mesh, our ongoing studies are focused on extraction of similar formulas especially for some popular benchmark electromagnetic scattering problems yielding unstructured mesh.

References

[1] VAZQUEZ, F., ORTEGA, G., FERNANDEZ, J. J., GARZON, E. M. Improving the performance of the sparse matrix vector product with GPUs. In *Proceedings of the IEEE 10th International Conference on Computer and Information Technology*. Bradford (United Kingdom), 2010, p. 1146–1151.

[2] BUATOIS, L., CAUMON, G., LEVY, B. Concurrent number cruncher: A GPU implementation of a general sparse linear solver. *International Journal of Parallel, Emergent and Distributed Systems*, 2009, vol. 24, no. 3, p. 205–223.

[3] MONAKOV, A., LOKHMOTOV, A., AVETISYAN, A. Automatically tuning sparse matrix-vector multiplication for GPU architectures. *Lecture Notes in Computer Science (LNCS)*, 2010, vol. 5952, p. 111–125.

[4] CENDES, Z. J. Vector finite elements for electromagnetics field computation. *IEEE Transactions on Magnetics*, 1991, vol. 27, no. 5, p. 3958–3966.

[5] COULOMB, J. L. Finite elements three dimensional magnetic field computation. *IEEE Transactions on Magnetics*, 1981, vol. 17, no. 6, p. 3241–3246.

[6] WANG, R., DEMERDASH, N. A. On the effects of grid ill-conditioning in three dimensional finite element vector potential magnetostatic field computations. *IEEE Transactions on Magnetics*, 1990, vol. 26, no. 5, p. 2190–2192.

[7] WU, J. Y., LEE, R. The advantages of triangular and tetrahedral edge elements for electromagnetic modeling with the finite-element method. *IEEE Transactions on Antennas and Propagation*, 1997, vol. 45, no. 9, p. 1431–1437.

[8] WARREN, G. S., SCOTT, W. R. An investigation of numerical dispersion in the vector finite-element method using quadrilateral elements. *IEEE Transactions on Antennas and Propagation*, 1994, vol. 42, no. 11, p. 1502–1508.

[9] BERENGER, J. P. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 1994, vol. 114, p. 185–200.

[10] CHEW, W. C, WEEDON, W. H. A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates. *Microwave and Optical Technology Letters*, 1994, vol. 7, no. 13, p. 599–604.

[11] VOLAKIS, J. L., CHATTERJEE, A., KEMPEL, L. C. *Finite Element Method for Electromagnetics – Antennas, Microwave Circuits, and Scattering Applications*. 1st ed. New York (NY, USA): IEEE Press, 1998.

[12] JIN, J. *The Finite Element Method in Electromagnetics*. 2nd ed. New York (NY, USA): Wiley Interscience, 2002.

About Authors ...

Gökay AKINCI was born in Kırıkkale, Turkey, in 1987. He received his M.Sc. degree in Electrical and Electronics Engineering from Kırıkkale University in 2011. He is currently working towards a Ph.D. degree at the Department of Electrical and Electronics Engineering in Ankara University. His current research interests are in the areas of parallel programming, sparse matrix and computational electromagnetics.

Asim Egemen YILMAZ received his B.Sc. degrees in Electrical-Electronics Engineering and Mathematics from the Middle East Technical University in 1997. He received his M.Sc. and Ph.D. degrees in Electrical-Electronics Engineering from the same university in 2000 and 2007, respectively. He is now with Ankara University, where he is an Associate Professor.

Mustafa KUZUOGLU received his B.Sc., M.Sc., and Ph.D. degrees in Electrical-Electronics Engineering from the Middle East Technical University in 1979, 1981, and 1986, respectively; where he is currently a Professor. His research interests include computational electromagnetics, inverse problems, and radar.