Received November 2000
Accepted February 2001

Spectral trisection of finite element models

A. Kaveh
Iran University of Science and Technology, Narmak, Tehran, Iran and
H.A. Rahimi Bondarabady
University of Yazd, Yazd, Iran

Keywords *Graph theory, Algebraic model, Laplace transform, Genetic algorithms*

Abstract *In this paper a new method is proposed for finite element domain decomposition. A weighted incidence graph is first constructed for the finite element model. A spectral partitioning heuristic is then applied to the graph using the second and the third eigenvalues of the Laplacian matrix of the graph, to partition it into three subgraphs and correspondingly trisect the finite element model.*

1. Introduction

Parallel processing provides considerable computational power for the analysis of large structural systems. For implementation of a parallel algorithm it is necessary to descretize the given domain into finite element subdomains, and analyze each subdomain by a processor. The results are then assembled for overall solution. An efficient domain decomposition algorithm should satisfy the following properties:

- It should be able to handle any irregular mesh geometry, i.e. it should be based on topological (graph theoretical) properties of the mesh.
- The number of interface nodes or elements between subdomains should be minimized; this helps to reduce the size of the overall problem.
- Subdomains should contain an approximately equal number of elements for balancing the computational load between processors.
- Partitioned subdomains should have good aspect ratios.

Mesh decomposition algorithms using the above criteria have been presented by Kaveh (1997), Farhat (1988), Farhat and Lesoinne (1993), Kaveh and Roosta (1994), Simon (1991), Kaveh and Davaran (1999), Pothen *et al.* (1990), Kaveh and Rahimi Bondarabady (2000), Khan and Topping (1991; 1993), Vanderstraeten *et al.* (1993) and Vanderstraeten and Keunings (1995). Recursive spectral bisection (RSB) performs the decomposition by repeatedly dividing a domain into two subdomains, assigning the elements to each new subdomain according to the components of a vector known as the Fiedler vector (Fiedler, 1993; Mohar, 1993). The Fiedler vector is the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix of the graph model of the mesh (Vanderstraeten *et al.*, 1993). The RSB method is based on the algebraic graph theoretical concepts and it has been shown to produce good results. The computational effort of the RSB method is increased when it operates on very

International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 11 No. 4, 2001, pp. 358-370.

fine meshes with a large number of elements. This difficulty is partially overcome by the RSB method augmented with other graph theoretical (Greedy) approaches (Kaveh and Davaran, 1999). The fine graded meshes, with numerous elements, are usually encountered in adaptive FE analysis, where the density of the mesh is altered after primary FE analysis of the problem on the initial background mesh. Conventional mesh partitioning algorithms operates on the overall final mesh to obtain partitions. Subdomain generation method (SGM) is one of the efficient methods which can approximately handle such variable mesh topologies (Khan and Topping, 1993; Vanderstraeten *et al.*, 1993; Vanderstraeten and Keunings, 1995; Fiedler, 1993). The SGM has been implemented for planar convex finite element subdomains using adaptive triangular meshes.

The SGM operates on initial coarse mesh; therefore, when the number of elements in the initial mesh is large, some difficulties arise from part (a), which dictates the parallel implementation of the SGM (Simon, 1991; Seale and Topping, 1995).

2. Criteria for partitioning

In the process of analysis, it is necessary to have information exchange between adjacent subdomains. As an example, for the finite element mesh shown in Figure 1, the two elements A and B are contained in two different adjacent subdomains. The inter-relation between these elements in the overall stiffness matrix is illustrated in Figure 2. It can easily be observed that the amount of information exchanged between these two elements is equal to three, considering the symmetry of the stiffness matrix. This number is given considering only one unknown per node and it should be multiplied by the degrees of freedom of each node. This number is called the communication number of the two elements. It can easily be proved that, if the number of common nodes of two elements is denoted by n_c, then the communication number will be given as

$$\frac{n_c(n_c+1)}{2}$$

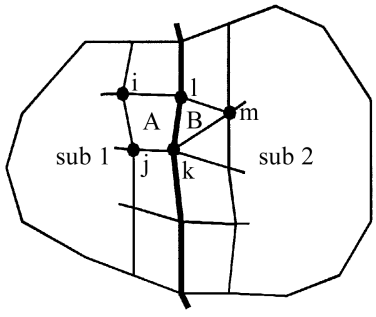


Figure 1.
A finite element mesh, and the two elements A and B contained in two different adjacent subdomains

Obviously for an optimal partitioning it is necessary to minimize the sum of the communication numbers. Such a minimization reduces the interaction of the elements, decreasing the time for computation.

On the other hand, since in the parallel processing the computational time is proportional to the maximum time required for processing each subdomain, the partitioning should be performed in such a way that all subdomains require nearly the same computational time. Since the processing time for each subdomain is proportional to the total number of elements in each subdomain, one should try to partition a model into subdomains with an equal number of elements. Thus for an optimal subdomaining the following considerations are vital:

- Equal number of elements in each subdomain.
- Minimum communication between the elements of each pair of adjacent subdomains.

3. Weighted incidence graphs for finite element models

An incidence graph G of an FEM has its vertices in a one-to-one correspondence with the elements of the considered FEM, and two vertices of G are connected by an edge, if the corresponding elements have at least one common node. Weights are assigned to the edges of G as the communication number for each pair of elements in the FEM. Such a graph is called the weighted incidence graph of the considered FEM.

As an example, a finite element model and its corresponding weighted incidence graph are illustrated in Figure 3.

Using this graph the problem of efficient decomposition of an FEM is transferred into an optimal partitioning of the corresponding graph, with the following properties:

- the numbers of nodes for subgraphs are nearly equal;
- the sum of the weights of the edges for which the two ends are in two different subgraphs is minimum.

A graph partitioning to three parts and the corresponding trisection of the FEM are illustrated in Figure 3.

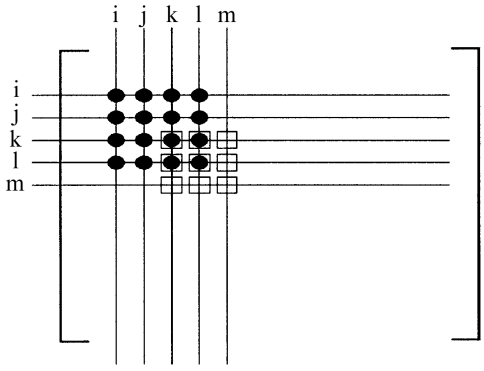


Figure 2.
The inter-relation between the elements in the overall stiffness matrix

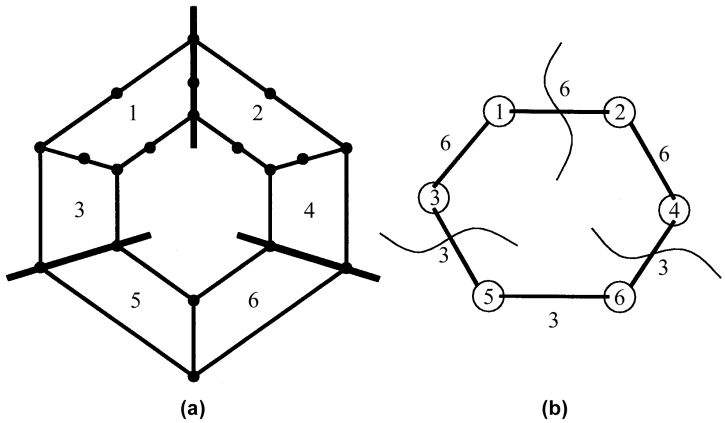


Figure 3.
A finite element model and the corresponding weighted incidence graph

4. Graph partitioning using the genetic algorithm

In this method, with each node i of the graph a number x_i is associated. Depending on the node being included in the first or the second subgraph, this number is +1 or –1.

The Laplacian matrix $L^* = [1^*_{ij}]_{N\times N}$ of a weighted incidence graph is defined as follows:

$$1^*_{ij} = \begin{cases} 1^*_{ii} = \sum_k w_{ik} & \text{if } i = j, \\ 1^*_{ij} = -w_{ij} & \text{if nodes i and j are adjacent,} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where N is the number of nodes of the graph corresponding to the number of the elements of the FEM.

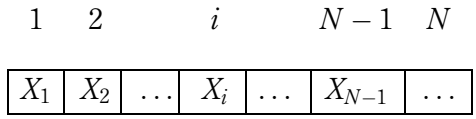
The optimal graph partitioning problem can be formulated as

$$\begin{aligned} &\text{Find } \mathbf{x} = [x_i\}_{N\times 1} \\ &\text{To minimize communication } = \mathbf{x}^t\mathbf{L}^*\mathbf{x} \\ &\text{s.t.} \mathbf{x}^t\mathbf{e} = 0, \mathbf{e}^t = [1, 1, \ldots, 1] \\ &x_i = +1 \text{ or } -1 \end{aligned} \tag{2}$$

The condition $\mathbf{x}^t\mathbf{e} = 0$ means that the number of +1s is the same as the number of –1s. (i.e. equilibrium of the nodes is satisfied). This condition can be fulfilled for graphs with an even number of nodes; however, for those with an odd number of nodes this cannot be fulfilled. Therefore this condition should be replaced by

$$\text{Minimize } \mathbf{x}^t\mathbf{e}. \tag{3}$$

For solution of the above optimization problem the genetic algorithm is employed. One gene is associated with each node of the graph and it is represented by +1 or –1. The chromosome obtained by these genes forms the vector **x** as



x_i being +1 or –1, and n is the number of nodes of the graph.

Fitness function

The objective function has the following form

$$f = \frac{\mathbf{x}^t \mathbf{L}^* \mathbf{x}}{\text{total member weights}} + w\frac{\mathbf{x}^t e}{\text{number of nodes}} \quad (4)$$

where the second term is a penalty term being a function of equilibrium condition, and w is the violation coefficient.

Pairing

Pairing two parents can take place using a competitive approach. Two parents are selected randomly from the selected population and the fittest are chosen for pairing. Repeating a similar process, the second parent is selected.

Cross-over

The process of cross-over is performed. In this step, two new children are produced. Here, the ripple approach is adopted. In this method, for all the genes, two genes are randomly interchanged and the new chromosomes are produced. In these children some of the genes are transformed from their parents.

Replacement

Two children produced in the stage of cross-over are replaced with the two most unsuitable chromosomes in the population (those maximizing the fitness function). In this replacement no attention is paid to the values of the objective function for the chromosomes. Even if the fitness values for these chromosomes are lower than those of the new chromosomes, these replacements are performed.

Mutation

In order to reintroduce genetic diversity to avoid getting trapped in local optima, mutation should be performed. This can be achieved using different techniques. In the first approach, after cross-over and before replacement, mutation is performed on them. In the second approach, after cross-over and replacement mutation is performed on some chromosomes in a random manner. Here the second approach is adopted.

In the process of mutation, irrespective of the method being used, some of the genes of a chromosome are selected for random change. Here since the genes are in the form of +1 and –1, in the process of mutation some genes are changed from +1 to –1 and vice versa.

In order to control the mutation, a parameter called the rate of mutation is used. This parameter is taken as 1/1,000 which shows the probability of the mutation of each gene.

Since the genetic algorithm has an iterative nature, for controlling the convergence, measures should be used. Here the following two measures are considered:

(1) *Total number of generations*. When the number of children becomes the same as the number of population, we say a generation is produced. In the process of genetic operations the number of generations is limited to an appropriate value, e.g. 10,000 in the case of this problem.

(2) *Certainty count.* In the process of iteration if, after some generation, the value of the fitness function does not improve further, the operation is terminated and the result obtained is taken as final. This number of generation is taken as the certainty count, e.g. 100 in the case of this problem.

A result is taken as improved if the value for fitness function in each generation is reduced considerably with respect to the previous one. Thus relative improvement in two consequent generations is taken as a measure of the improvement.

The program developed in this paper is applied to many models and the results show the suitability of the present method for small size problems. For large-scale problems the convergence of the method reduces considerably. In order to increase the applicability of the genetic algorithm, the following method is developed.

5. Graph trisection algorithm

The vector $\mathbf{x}$ will partition the graph into three subgraphs containing equal numbers of vertices (N/3), if the number of z_is are N/3, and it follows that

$$\sum_{k=1}^{N} x_k = 0 \tag{5}$$

where N is assumed to be a multiple of three.

On the other hand, $\mathbf{x}^t.e = \mathbf{x}^t.\phi_1 = 0$; this means that the partitioning vector $\mathbf{x}$ is orthogonal to ϕ_1. It is necessary to assign a measure of how well this vector minimizes the communication between the vertices of each pair of subgraphs.

Consider two arbitrary adjacent vertices i and j of the graph; if these two vertices belong to the same subgraph, then

$$\|x_i - x_j\|^2 = 0, \tag{6}$$

where for each complex number $C = A + B_i, \|C\| = A^2 + B^2$ and, if they belong to different subgraphs, then

$$\|x_i - x_j\|^2 = 3 \tag{7}$$

Thus the resulting communication between them is equal to $\frac{1}{3}\|x_i - x_j\|^2 w_{ij}$. So the total communication between adjacent subgraphs is

$$t_c(x) = \frac{1}{3}\sum_{(i,j)\epsilon E} \|x_i - x_j\|^2 \times w_{ij} \tag{8}$$

On the other hand, according to the fact that $w_{ij} = 0$ for non-adjacent vertices i and j, it can be written:

$$t_c(x) = \frac{1}{2} \times \frac{1}{3}\sum_{i=1}^{N}\sum_{j=1}^{N} \|x_i = x_j\|^2 \times w_{ij} \tag{9}$$

where the coefficient 1/2 is due to the double consideration of edge (i, j).

According to the definition of Laplacian matrix of the weighted graph it can be shown that

$$t_c(x) = \frac{1}{6}\mathbf{x}^t\mathbf{L}^*\mathbf{x}, \tag{10}$$

and the problem of efficient partitioning is converted to the following optimization problem:

$$\begin{aligned} &\text{Minimize } t_c(x) = \frac{1}{6}\mathbf{x}^t\mathbf{L}^*\mathbf{x} \\ &\text{s.t. } x^t.\mathbf{e} = 0 \\ &\text{and } x_k \epsilon \left\{+1, -\frac{1}{2} + \frac{\sqrt{3}}{2}i, -\frac{1}{2} - \frac{\sqrt{3}}{2}i\right\} \end{aligned} \tag{11}$$

Separating real and imaginary parts of vector $\mathbf{x}$, this vector can be written as

$$\mathbf{x} = u + v_i \tag{12}$$

where $u, v \epsilon R^N$. Hence $\sum u_i = \sum v_i = 0$ and $\|x_i - x_j\|^2 = (u_i - u_j)^2 + (v_i - v_j)^2$. This implies that the problem (11) is equivalent to the following problem:

$$\begin{aligned} &\text{Minimize } t_c(\mathbf{x}) = \frac{1}{6}(\mathbf{u}^t\mathbf{L}^*\mathbf{u} + \mathbf{v}^t\mathbf{L}^*\mathbf{v}) \\ &\text{s.t} \quad \mathbf{u}^t.\mathbf{e} = \mathbf{v}^t.\mathbf{e} = 0 \\ &\text{and} \quad (u_i, v_i)\epsilon\left\{(+1, 0), \left(\frac{-1}{2}, \frac{\sqrt{3}}{2}\right), \left(\frac{-1}{2}, \frac{-\sqrt{3}}{2}\right)\right\} \end{aligned} \tag{13}$$

On the other hand, according to $\sum_{i=1}^{N} u_i = \sum_{i=1}^{N} v_i = 0$ it can be written as:

$$\begin{aligned}
\mathbf{u}^t.\mathbf{v} &= \frac{N}{3}\left(1\times 0 + \frac{-1}{2}\times\frac{\sqrt{3}}{2} + \frac{-1}{2}\times\frac{-\sqrt{3}}{2}\right) = 0 \\
\mathbf{u}^t.\mathbf{u} &= \frac{N}{3}\left[(0)^2 + \left(\frac{-1}{2}\right)^2 + \left(\frac{-1}{2}\right)^2\right] = \frac{N}{2} \\
\mathbf{v}^t.\mathbf{v} &= \frac{N}{3}\left[(0)^2 + \left(\frac{\sqrt{3}}{2}\right)^2 + \left(\frac{-\sqrt{3}}{2}\right)^2\right] = \frac{N}{2}
\end{aligned} \tag{14}$$

A property of real, symmetric matrices such as L^* is that their eigenvectors are orthogonal and form a basis in R^N. A set of vectors is said to form a basis in R^N, if all vectors of R^N can be written as a linear combination of these vectors.

Consequently, for an appropriately chosen set of coefficients a_i and b_i, $i = 1, 2, \ldots, N$, the vectors **u** and **v** can be written as

$$\mathbf{u} = \sum_{i=1}^{N} a_i \phi_i$$

and

$$\mathbf{v} = \sum_{i=1}^{N} b_i \phi_i \tag{15}$$

On the other hand, $\phi_1 = \mathbf{e}$ and $\mathbf{u}^t.\mathbf{e} = \mathbf{v}^t.\mathbf{e} = 0$; consequently $a_1 = b_1 = 0$ and

$$\mathbf{u} = \sum_{i=2}^{N} a_i \phi_i$$

and

$$\mathbf{v} = \sum_{i=2}^{N} b_i \phi_i \tag{16}$$

Normalizing the vectors ϕ_i so that $\phi_i^t.\phi_i = N/2$, $i = 2, \ldots, N$, then

$$\mathbf{u}^t.\mathbf{u} = \frac{N}{2}\left(\sum_{i=2}^{N} a_i \phi_i^t\right)\left(\sum_{i=2}^{N} a_i \phi_i\right) = \frac{N}{2}\left(\sum_{i=2}^{N} a_i^2\right) \tag{17}$$

from which

$$\sum_{i=2}^{N} a_i^2 = 1 \tag{18}$$

and similarly

$$\sum_{i=2}^{N} b_i^2 = 1 \tag{19}$$

On the other hand

$$\mathbf{u}.\mathbf{v} = 0 = \left(\sum_{i=2}^{N} a_i \phi_i^t\right)\left(\sum_{i=2}^{N} b_i \phi_i\right) = \frac{N}{2}\left(\sum_{i=2}^{N} a_i b_i\right) \tag{20}$$

from which

$$\sum_{i=2}^{N} a_i b_i = 0 \tag{21}$$

Since $\mathbf{L}^* \phi_i = \lambda_i \phi_i$, then

$$\begin{aligned} \mathbf{u}^t \mathbf{L}^* \mathbf{u} &= \left(\sum_{i=2}^{N} a_i \phi_i^i\right)\left(\sum_{i=2}^{N} a_i L * \phi_i\right) \\ &= \left(\sum_{i=2}^{N} a_i \phi_i^t\right)\left(\sum_{i=2}^{N} a_i \lambda_i \phi_i\right) = \frac{N}{2}\sum_{i=2}^{N} a_i^2 \lambda_i \end{aligned} \tag{22}$$

and similarly

$$\mathbf{v}^t \mathbf{L}^* \mathbf{v} = \frac{N}{2}\sum_{i=2}^{N} b_i^2 \lambda_i \tag{23}$$

In this manner the optimization problem (13) can be written as

$$\left|\begin{array}{ll} \text{Minimize} & t_c(x) = \dfrac{N}{12}\displaystyle\sum_{i=2}^{N}(a_i^2 + b_i^2)\lambda_i \\ \text{s.t.} & \displaystyle\sum_{i=2}^{N} a_i^2 = 1,\ \sum_{i=2}^{N} b_i^2 = 1,\ \sum_{i=2}^{N} a_i b_i = 0 \\ & \mathrm{u_i} = \displaystyle\sum_{i=2}^{N} a_i \phi_i,\ \mathrm{v_i} = \sum_{i=2}^{N} b_i \phi_i \\ & (\mathrm{u_i v_i}) \epsilon \{(\ ), (\ ), (\ )\} \end{array}\right. \tag{24}$$

It is at this point that the heuristic must be applied. If N = 3 is considered, then the conditions (18), (19) and (21) are as follows:

$$a_2^2 + a_3^2 = 1, \quad b_2^2 + b_3^2 = 1, \quad a_2 b_2 + a_3 b_3 = 0 \tag{25}$$

and the other conditions are not applied.

The solution of these equations can be considered as follows:

$$\begin{aligned} |a_2 &= \cos\theta, \quad a_3 = \sin\theta \\ |b_2 &= -\sin\theta, \quad b_3 = \cos\theta \end{aligned} \tag{26}$$

This implies that

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \phi_2 \\ \theta_3 \end{bmatrix} \tag{27}$$

With this approximation the components of (u_i, v_i) are not generally the prescribed discrete values and the vertices of the graph cannot be clustered on the $z_k, k = 1, 2, 3$ basis. Instead the N/3 vertices corresponding to the N/3 closest numbers to $z_k s$ are assigned to one of the subgraphs.

After computation of ϕ_2 and ϕ_3 by changing θ, values of (u_i, v_i) for each vertex can be computed. According to the above criteria the optimum value of θ is the value which results in minimum communication between subgraphs.

For applying the above criteria for finding closest (u_i, v_i) vectors to $z_k s$ it is sufficient to compute and compare angles between this vector and z_k vectors.

Figure 4(a) shows a typical vector (u_i, v_i) which is close to z_1, implying that the vertex i belongs to the subgraph 1.

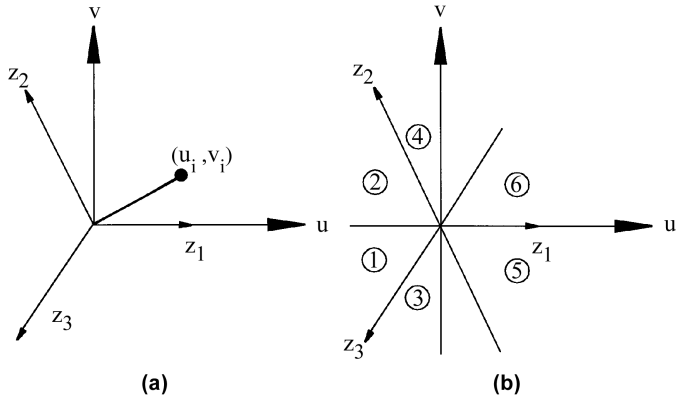


Figure 4.
Trisection in u-v coordinate system

As an example, for the graph shown in Figure 3, and

$$\mathbf{L}^{*} = \begin{bmatrix} 12 & -6 & -6 & & & \\ -6 & 12 & 0 & -6 & & \\ -6 & 0 & 9 & 0 & -3 & \\ & -6 & 0 & 9 & 0 & -3 \\ & & -3 & 0 & 6 & -3 \\ & & & -3 & -3 & 6 \end{bmatrix},$$

the solution of the eigenproblem $\mathbf{L}^{*}\phi_i = \lambda_i\phi_i$, results in

$$\lambda_2 = 3.80 \text{ and } \phi_2 = [-0.41, -0.41, -0.15, -0.15, \ 0.56, 0.56]^t$$

$$\lambda_3 = 4.40 \text{ and } \phi_3 = [-0.25, 0.25, -0.56, 0.56, -0.36, 0.36]^t$$

For $\theta = 0$, we have $u = \phi_2$ and $v = \phi_3$. (u_i, v_i) for $i = 1,\ldots,6$ in the u-v coordinate system are depicted in Figure 4(b). According to this Figure, it can be seen that vertices 5 and 6 belong to subgraph 1, vertices 1 and 4 belong to subgraph 2, and vertices 1 and 3 belong to subgraph 3. The corresponding partitioned graph and FEM are illustrated in Figure 3.

6. Numerical results

Many examples are trisected on a Pentium 2, and the results for a few of the considered models are presented in the following.

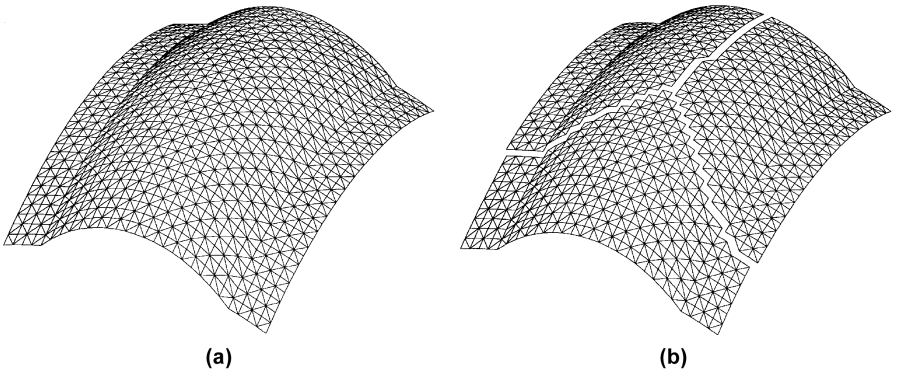


Figure 5.
A shell type FEM

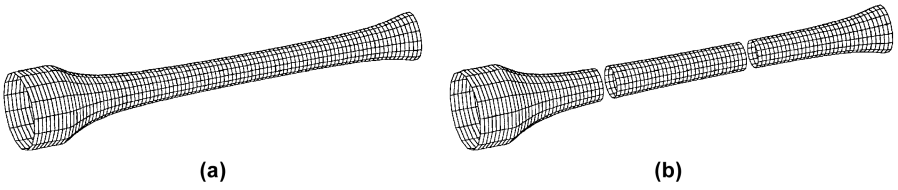


Figure 6.
The FEM of a nuzzle

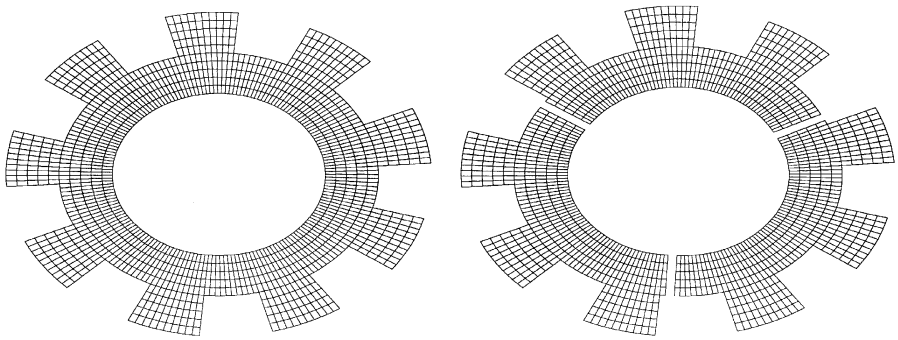


Figure 7.
The FEM of a fan

Example 1: The model of a shell structure is considered as shown in Figure 5(a). This model has 2,016 triangular elements. The model is trisected as illustrated in Figure 5(b). Total computational time has been 26 seconds.

Example 2: The FEM of a nuzzle is considered as shown in Figure 6(a). This model contains 1,485 rectangular elements. The results are illustrated in Figure 6(b). The computer time for this trisection has been 15 seconds.

Example 3: The FEM of a fan with one opening is considered as shown in Figure 7(a). This model contains 1,350 rectangular elements. The trisected model is illustrated in Figure 7(b) with absolutely balanced nodes. The computational time has been 15 seconds.

7. Discussion and concluding remarks

The proposed algorithm for finite element domain decomposition is simple and efficient. A weighted incidence graph is first constructed and a spectral partitioning heuristic is then applied to the graph using the second and the third eigenvalues of the Laplacian matrix of the graph, to partition it into three subgraphs and trisect the corresponding finite element model. This method can recursively be applied for further decomposition of the models.

References

Farhat, C. (1988), "A simple and efficient automatic FEM domain decomposer", *Comput. Struct.*, Vol. 28, pp. 579-602.

Farhat, C. and Lesoinne, M. (1993), "Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics", *Int. J. Num. Meth. Engng.*, Vol. 36, pp. 745-64.

Fiedler, M. (1993), "Algebraic connectivity of graphs", *Czech. Math. J.*, Vol. 23, pp. 298-305.

Kaveh, A. (1997), *Optimal Structural Analysis*, RSP (John Wiley), UK.

Kaveh, A. and Davaran, A. (1999), "Spectral bisection of adaptive finite element meshes for parallel processing", *Comput. Structs.*, Vol. 70, pp. 315-24.

Kaveh, A. and Rahimi Bondarabady, H.A. (2000), "Finite element mesh decompositions using complementary Laplacian matrix", *Communs Numer. Meths Engng*, Vol. 16, pp. 379-89.

Kaveh, A. and Roosta, G.R. (1994), "A graph theoretical method for decomposition in finite element analysis", *Proc. CIVIL-COMP*, Edinburgh, UK, pp. 35-42.

Khan, A.I. and Topping, B.H.V. (1991), "Parallel adaptive mesh generation", *Computing System in Engineering*, Vol. 2 No. 1, pp. 75-102.

Khan, A.I. and Topping, B.H.V. (1993), "Sub-domain generation for parallel finite element analysis", *Proc. Civil Comp*, UK, p. 93.

Mohar, B. (1993), "The Laplacian spectrum of graphs", in Alavi, Y. *et al.* (Eds), *Graph Theory, Combinatorics and Applications*, Vol. 2, John Wiley, New York, NY, pp. 871-98.

Pothen, A., Simon, H.D. and Liou, K.P. (1990), "Partitioning sparse matrices with eigenvectors of graphs", *SIAM J. Matrix Anal. Appl.*, Vol. 11, pp. 430-52.

Seale, C. and Topping, B.H.V. (1995), "Parallel implementation of recursive spectral bisection", *Proc. Civil-Comp 1995*, Edinburgh, UK, pp. 459-73.

Simon, H.D. (1991), "Partitioning of unstructured problems for parallel processing", *Computing Systems in Engineering*, Vol. 2 No. 2/3, pp. 135-48.

Vanderstraeten, D. and Keunings, R. (1995), "Optimized partitioning of unstructured meshes", *Int. J. Num. Meth. Engng.*, Vol. 38, pp. 433-50.

Vanderstraeten, D., Zone, O. and Keunings, R. (1993), "Non-deterministic heuristic for automatic domain decomposition in direct parallel finite element calculation", *Proc. 6th SIAM Conf. on Parallel Processing*, SIAM, pp. 929-32.