Review

Free Mesh Method: fundamental conception, algorithms and accuracy study

By Genki YAGAWA*1,*2,†

(Communicated by Takao INUI, M.I.A.)

Abstract: The finite element method (FEM) has been commonly employed in a variety of fields as a computer simulation method to solve such problems as solid, fluid, electro-magnetic phenomena and so on. However, creation of a quality mesh for the problem domain is a prerequisite when using FEM, which becomes a major part of the cost of a simulation. It is natural that the concept of meshless method has evolved. The free mesh method (FMM) is among the typical meshless methods intended for particle-like finite element analysis of problems that are difficult to handle using global mesh generation, especially on parallel processors. FMM is an efficient node-based finite element method that employs a local mesh generation technique and a node-by-node algorithm for the finite element calculations. In this paper, FMM and its variation are reviewed focusing on their fundamental conception, algorithms and accuracy.

Keywords: finite element method (FEM), free mesh method (FMM), enriched free mesh method (EFMM), meshless method, parallel computing

1. Introduction

The recent progress in computers has enabled a number of complicated natural phenomena to be accurately simulated. Among various computer simulation techniques, the finite element method (FEM) has been most widely used since the invention of the method in late 1950s in a variety of practical fields, such as mechanical, aerospace, nuclear, chemical and civil engineering. It can be applied flexibly to complex boundary shapes using unstructured computational grids, and results, accurate enough for engineering purposes, are obtainable at reasonable cost.¹,2)

When using FEM, one needs to divide a continuum into a number of elements. The individual elements are connected together by a topological map, which is usually called a mesh. The finite element interpolation functions are then built on the mesh. Not only in FEM,¹,2) but in the Finite Difference Method (FDM),³) and the Finite Volume Method (FVM),⁴) the problem spatial domain is discretized into meshes. The mesh must be predefined to provide a certain relationship between the nodes, which becomes the building blocks of the formulation procedure of these conventional numerical methods.

It is well known that the mesh generation process often becomes a serious bottleneck in large-scale computing,⁵–⁷) especially for problems that require frequent mesh refinement, such as the moving boundary problems. The limitations of the mesh-based methods, particularly those of FEM are summarized as follows,

1. The analyst spends much time in creating the mesh, which becomes a major part of the cost of a simulation process. As the cost of central processing unit (CPU) time is drastically decreasing, the concern is more the manpower time, and less the computer time.
2. It is rather difficult to simulate both crack growth with arbitrary and complex paths and phase transformations due to discontinuities that do not coincide with the original nodal lines.
3. Remeshing methods have been proposed for handling these types of problems, where the
problem domain is remeshed at steps during the simulation process to prevent the severe distortion of meshes and to allow the nodal lines to remain coincident with the discontinuity boundaries. For this purpose, complex, robust, and adaptive mesh generation processors have to be developed. However, these processors are only workable for 2D problems.

It is natural that the concept of meshless method has evolved, which is the idea of eliminating or reducing the reliance on the elements and more flexible ways to make use of mesh. In this concept, the domain of the problem is represented, ideally, only by a set of arbitrarily distributed nodes, where the physical field under consideration is approximated by a set of points. Smoothed particle hydrodynamics (SPH), diffuse element method (DEM), gridless Euler/Navier–Stokes solution, element-free Galerkin method (EFGM), reproducing kernel particle method (RKPM), hp-meshless cloud method, finite point method (FPM), moving-particle semi-implicit method (MPS), meshless local boundary integral equation (MLBIE) and meshless local Petrov–Galerkin method (MLPG) are considered as typical meshless methods.

Among these methods, SPH was originally developed for modeling astronomical phenomena using particles, with RKPM and MPS being improvements to this development. These particle type meshless methods require a predefinition of particles for their volumes or masses. The algorithm will then carry out the analyses even if the problem domain undergoes extremely large deformation or separation. This type of method suffers from problems in the imposition of boundary conditions. In addition, predefining the particles still technically requires some kind of mesh. SPH simulates well the overall behaviors of certain class of problems such as highly nonlinear and momentum-driven problems. These particle-based methods mainly aim at numerical simulation of phenomena such as splitting, scattering and unifying of bodies.

In general, methods that do not require a mesh at all are less stable and less accurate. Local point collocation methods and FDMs using irregular grids are among this category. Selection of nodes based on the type of a physical problem can be important for obtaining stable and accurate results. Automation of nodal selection and improving the stability of the solution are still some of the challenges in these kinds of methods. This type of method has a very significant advantage: It is very easy to implement, because no integration is required. There are, however, vital instability issues that require special treatments.

In contrast, DEM, EFGM, MLBIE and MLPG are classified as meshless methods in a narrow sense, and the major aim of these methods is the numerical simulation of continuum mechanics without finite element mesh, while the nature of the approximation is very similar to FEM in that they are based on the Galerkin method. While these meshless methods exhibit excellent performance in some special fields, they have not always succeeded in replacing the conventional FEM. The primary reason is that the meshless methods are not effective in representing the three-dimensional complex shapes required in practical engineering problems.

In this context, a number of FEM-based meshless approaches have been proposed to overcome the difficulty of mesh generation. Manifold method, voxel finite element method, generalized finite element method (GFFEM), partition of unity finite element method (PUFEM), extended finite element method (X-FEM) and finite cover method (FCM) can be classified as FEM-based meshless method. Even though these methods employ elements or a mesh to approximate the physical field, they are still regarded as meshless methods, because an explicit mesh is not required in the input data, or else a flexible mesh, that can be prepared much more easily than a conventional FEM mesh, is available as input data.

The free mesh method (FMM) is one of the earliest FEM-based meshless methods intended for particle-like finite element analysis of problems that are difficult to handle using global mesh generation, such as moving boundary problems, large deformation problems, crack propagation analysis, the separation or unification of bodies, as well as adaptive mesh refinement analysis, especially on parallel processors. FMM is a node-based finite element method that employs a local mesh generation technique and a node-by-node algorithm for the finite element calculations. The method can overcome difficulties arising from the distortion of elements by employing a node-based approach, with maintaining the superior theoretical foundations and the ability to handle complicated boundary shapes in accordance with finite elements. In FMM, both preprocessing and main-processing of analysis can easily be parallelized in terms of nodes, where the preprocessing involves local mesh generation, and the
main-processing the construction and solution of a system of equations. The method is quite suitable for parallel environments.

Reviewed in this paper are FMM as well as the Enriched Free Mesh Method (EFMM), a new version of FMM, focusing on their fundamental conception, algorithms and accuracy. In the following section, the fundamental concept of FMM is reviewed, and the third section deals with EFMM. Finally, the present paper is summarized by some concluding remarks.

2. Basic conception of Free Mesh Method (FMM)

2.1 Outline. FMM is a node-based finite element method, characterized by both the node-based local mesh generation and the node-by-node finite element calculation, which starts with distributing the nodes across the analysis domain. Data to input to the method are surface patches and nodes as shown in Fig. 1, where surface patches are sets of segments in two-dimensional space or planes in three-dimensional space, which surround the analysis domain and define the shape of the domain.

Figure 2 illustrates the basic concept of the method. First, a single node $P_i$ among the distributed nodes is designated as a central node. A local mesh is then generated around $P_i$ using any local mesh generation technique. A group of elements around $P_i$, which consist of several local meshes, are called the satellite elements associated with $P_i$. The satellite elements do not cross each other, and that the analysis domain is covered fully by local meshes. If this condition fails, the satellite elements are judged to be inconsistent (see Fig. 3). After the above local mesh generation, the components of the global coefficient matrix associated with $P_i$ are calculated by integrating all the satellite elements around $P_i$. The above process is repeated for all the nodes in the domain, and a global system of equations is obtained after all the nodes have been integrated. The flowchart of the method is shown in Fig. 4. Several techniques have been proposed for the above local mesh generation.

2.2 Algorithms for local mesh generation. We summarizes three types of local mesh generation algorithms in what follows.

Algorithm I: segment comparison algorithm. The segment comparison algorithm is proposed in the early stage of FMM. Satellite nodes for each central node are found by comparing the lengths of segments formed by nominated nodes existing in a local area called the searching circle, which is defined around the central node with a radius determined by the density of nodes at the central node. The process of searching for satellite nodes is performed using local information about nodal co-ordinates, and can be parallelized in a node-based manner.

Algorithm II: local Delaunay triangulation algorithm with searching circle. A disadvantage of the segment comparison algorithm described above is related to the determination of an appropriate radius of the searching circle. If the radius is too large or too small, the algorithm may fail to determine appropriate satellite nodes, so that inconsistent overlapping of satellite elements occurs as shown in Fig. 3. This problem is referred to as inconsistency in satellite elements. The presence of only a few inconsistencies in satellite elements is known to have little influence on the solution accuracy in the case of linear and stationary problems. However, too many inconsistencies may cause unsatisfactory results in non-linear or time-dependent problems, such as fluid dynamics problems. Therefore, a local Delaunay triangulation algorithm is developed as a substitute for the segment comparison algorithm, where the Delaunay triangulation is performed locally around each node. However, if a normal Delaunay triangulation procedure is used to search for satellite nodes, inconsistency problems can still occur when degeneracy exists in the Delaunay triangulation (see Fig. 5), because satellite nodes are selected independently at each node in a node-by-
node algorithm. To overcome this problem, the Delaunay triangulation algorithm has been modified using nodal identification number information. This algorithm has the advantage that inconsistencies in satellite elements can be avoided in almost any distribution of nodes, as long as the radius of the searching circle is sufficiently large.

Algorithm III: local Delaunay triangulation algorithm with gift-wrapping method. The local Delaunay triangulation algorithm mentioned above employs a searching circle. This sometimes causes a critical failure in the algorithm, because no theoretical basis exists for determining the radius of the searching circle. If the radius is too small, the algorithm cannot provide a proper number of satellite elements. On the other hand, if the radius is too large, too many unnecessary tentative elements are generated, and the algorithm needs extra computer power. In particular, this is a serious problem if the density of nodes varies rapidly over a domain to be solved. To overcome this problem, alternative local Delaunay triangulation
algorithm is developed based on the gift-wrapping method,\textsuperscript{59,60} which is an algorithm that calculates a triangle of Delaunay triangulation (DT) in a one-by-one manner. The algorithm is well suited to parallel computing, because the algorithm is highly localized spatially. Local meshes can always be generated by minimum searching of nodes without searching circles. Furthermore, the algorithm can generate local meshes robustly, even when the object contains knife-edge geometries.
Consider a triangle $abc$ in a two-dimensional space (see Fig. 6). To obtain an adjacent Delaunay triangle on the left side of segment $bc$ (see Fig. 6(a)), the circumcircle of the triangle $abc$ is first drawn as shown in Fig. 6(b). The circle is then moved as shown in Fig. 6(c), until the circle meets a node (see Fig. 6(d)). When the circle meets node $d$, a Delaunay triangle is formed between nodes $b$, $c$ and $d$ because the circle is nothing but the largest empty circle at this moment. Briefly, a new Delaunay triangle is obtained by using segment $bc$ of the settled Delaunay triangle and the nearest node in the outside of the segment $bc$. It is noted that this Delaunay triangle is determined uniquely.

Furthermore, the gift-wrapping method for obtaining Delaunay triangle can easily be extended to the three-dimensional space as shown in Fig. 7. It is noted here that Figs. 7(a) through 7(d) correspond to Figs. 6(a) through 6(d), respectively. However, the Delaunay triangulation described above examines only node co-ordinates data, so that certain boundary shapes, such as reentrant angles, cannot be treated. In order to deal with these shapes, the constrained Delaunay triangulation (CDT), which divides the given domain into Delaunay-like triangulations using the conforming constraint of surface patches, is proposed: a new algorithm for obtaining constrained Delaunay triangulations based on the gift-wrapping in a node-based manner (see Fig. 8). Another difficulty in the three-dimensional mesh generation is related to the so-called slivers, which can be avoided by changing the connectivity of nodes (see Fig. 9). However, this operation reduces the efficiency of
parallel mesh generation considerably. The remedy to avoid slivers will be to regularly arrange nodes as much as possible, which can be done by generating input nodal data using the Centroidal Voronoi Tessellation (CVT).64),65)

2.3 Node-by-node algorithm and parallelization based on nodes.43) The present method consists of a node-by-node procedure for local mesh generation and construction of the global matrix. In other words, the global matrix is constructed in a
row-by-row manner (see Figs. 2 and 4), where the non-zero components of the global matrix in the row of a node correspond to the components derived from the satellite nodes associated with the node.

The final system of equations obtained by the present node-by-node procedure is equivalent to that obtained by the element-by-element procedure of the conventional FEM, while difference between the two procedures appears in the parallel computing. Figure 10(a) shows the element-based domain decomposition for parallel computing, where the global matrix is not constructed explicitly, but parallel computing is performed instead in an element-by-element manner.66,67 On the other hand, the node-based domain decomposition is employed in the proposed method (see Fig. 10(b)), where nodes in the analysis domain are classified into two types: communication-independent nodes and communication-dependent nodes. The former ones are defined as the nodes, which do not require communication between processors when parallel computing is performed, whereas the latter ones are defined as those which require communication between processors. Figure 11 shows both types of nodes regarding PE1 in Fig. 10. Parallel efficiency can be improved by dividing the nodes into these two categories.

The node-based partition of computational loads across multiprocessors shown in Fig. 10(b) relates to row-based division for parallel calculation of the product of a matrix and a vector (see Fig. 12). Communication between processors is required when satellite nodes associated with a central node are located on other processors than that in which the central node is located (see Fig. 13), where the satellite nodes $S_1$ and $S_6$ are handled by a processor different from that handles the central node $P_i$. In other words, inter-processor communication is required for nodes $S_1$ and $S_6$ when the product of a matrix by a vector is calculated. These satellite nodes are referred to as external satellite nodes, whereas other satellite nodes are called as internal satellite nodes. In Fig. 12, the internal satellite nodes correspond to those marked by open circles, whereas the external satellite nodes correspond to those marked by gray triangles and the central nodes to the diagonal components of the matrix. Inter-processor communication is carried out regarding the components marked by gray triangles in Fig. 12. The amount of communication between processors depends on the numbering of nodal identifiers. In this regards, a parallel graph-partitioning library ParMETIS68 is effectively used to renumber nodal identification numbers to minimize the amount of communication.47
2.4 Results of local mesh generation. In Fig. 14, the meshes generated by the above algorithms are compared, where Figs. 14(a)–(c) show the results of Algorithms I, II and III, respectively. Bold lines in Fig. 14(a) depict the inconsistencies in satellite elements. As can be seen from these figures, satisfactory results are achieved except Algorithm I. As another comparison of these three algorithms, meshes with combination of two different nodal densities are shown in Fig. 15, where Algorithm I is seen to fail again to generate appropriate local meshes if the radius of the searching circle is too large or too small. In Algorithm II, the inconsistencies can be avoided if the radius of the searching circle is sufficiently large. However, using too large a searching circle in Algorithm II results in many useless elements, being computationally wasteful. On the other hand, Algorithm III can generate local meshes robustly by a minimal search of neighboring nodes as demonstrated in Figs. 14(c) and 15(c), respectively. Furthermore, Algorithm III is sufficiently robust to treat singular boundary shapes such as cracks: Fig. 16(a) shows input surface patch and nodal data as an example of a knife-edge boundary shape and Fig. 16(b) the mesh generated by Algorithm III.

Figures 17(a)–(f) show the process of mesh generation applied to a three-dimensional crack model using Algorithm III, in which central nodes are selected at random and all the local meshes are generated independently around each node in a robust manner.

Figures 18(a)–(f) and Figs. 19(a)–(f) show the processes of Algorithm III applied to the three dimensional mesh generation for a model of the “Pantheon” building, which are viewed from the outside (see Fig. 18) and from the inside (see Fig. 19), respectively. Figures 20(a)–(f) show the process of parallel mesh generation with eight processors. As can be seen in these figures, the three-dimensional meshes are robustly generated. It
is noted that the present mesh generation technique is favored with parallel environments: Fig. 21 shows a large-scale mesh generated by Algorithm III, where the mesh shown in Fig. 21(a) has approximately $1.2 \times 10^5$ nodes and $4.7 \times 10^5$ elements and that shown in Fig. 21(b) approximately $2.1 \times 10^7$ nodes and $1.2 \times 10^8$ elements, in which virtually parallel technique is employed with single processor.  

3. Enriched Free Mesh Method (EFMM)\(^{(50),(51)}\)

3.1 Outline. “Assumed strain on the clustered local elements” characterizes EFMM (see Fig. 22), where the strain field on the clustered local elements and the displacement field of each local element are assumed independently. Coupling these two independent fields, the following approaches have been proposed: the localized least square method and the method based on the Hellinger–Reissner variational principle.

3.2 EFMM based on localized least square method. EFMM based on the localized least square method (EFMM-LS) assumes the strain field on the clustered local elements as

$$\{\varepsilon(\mathbf{x})\} = [\mathbf{N}'] \{\mathbf{a}\} \quad [1]$$

where $\{\varepsilon(\mathbf{x})\} = \{\varepsilon_{xx}, \varepsilon_{yy}, \gamma_{xy}\}$ is the strain field defined on the clustered local elements and each component of $\{\varepsilon_{xx}, \varepsilon_{yy}, \gamma_{xy}\}$ is assumed independently, and $[\mathbf{N}']$ is a matrix, which consists of arbitrary polynomials as follows,

$$[\mathbf{N}'] = \begin{bmatrix} p'(\mathbf{x}) & 0 & 0 \\ 0 & p'(\mathbf{x}) & 0 \\ 0 & 0 & p'(\mathbf{x}) \end{bmatrix} \quad [2]$$
$p'(x) \text{ is assumed on the clustered local elements as} $

\begin{align*}
\text{linear basis: } \\
&= [1 \ x \ y] \\
\text{quadratic basis: } \\
&= [1 \ x \ y \ x^2 \ xy \ y^2] \\
\text{cubic basis: } \\
&= [1 \ x \ y \ x^2 \ xy \ y^2 \ x^3 \ x^2y \ xy^2 \ y^3]
\end{align*}

\vdots

The coefficients vector $\{a\}$ in Eq. [1] is obtained by minimizing the $L_2$ norm $J$ as follows,

$$J = \sum_{i=1}^{n_e} \sum_{j=1}^{p} [(\varepsilon(x) - \{x_i^j\})^2]$$

where $n_e$ is the number of local elements with $c (= 1, 2, \cdots, n_e)$ being current local element, $p$ the number of points, which are called as the “strain monitoring points” on the clustered local elements with $i (= 1,$

Fig. 18. Process of three-dimensional mesh generation for a Pantheon model by Algorithm III (viewed from the outside).
2, · · · , p) being the current strain monitoring point and \( \{ \varepsilon_i^c \} \) the strain vector of \( i \)-th strain monitoring point on the \( c \)-th local element, which is called as the “mother element”. The stationary condition of Eq. [4] is

\[
\delta J = 2\{a\}^T \sum_{c=1}^{n_c} \sum_{i=1}^{p} [([N_i^c]^T [N_i^c])^{-1} [N_i^c]^T \{ \varepsilon_i^c \}]
\]

\[
= 0
\]

which yields the coefficients vector \( \{a\} \) as follows,

\[
\{a\} = \sum_{c=1}^{n_c} \sum_{i=1}^{p} [([N_i^c]^T [N_i^c])^{-1} [N_i^c]^T \{ \varepsilon_i^c \}]
\]

Let us consider a simple constant strain triangle as the mother element, in which the displacement field of each local element is defined by

\[
\{u\} = \sum_{i=1}^{3} (u_i) \zeta_i
\]
where \( \{ \mathbf{u} \} \) is the displacement field of the local element, \( \{ \mathbf{u}_i \} \) the nodal displacement, and \( \zeta_i \) the area-coordinate.\(^{69}\) Thus, the strain on the strain monitoring points is given by

\[
\{ \varepsilon_j^i \} = \{ \mathbf{B}_j^i \} \{ \mathbf{u}_i \}
\]

where

\[
\{ \mathbf{B}_j^i \} = [ \mathbf{B}_1^i \mathbf{B}_2^i \mathbf{B}_3^i ]
\]

with

\[
[ \mathbf{B}_j^i ] = \begin{bmatrix}
\frac{\partial \zeta_j}{\partial x} & 0 \\
0 & \frac{\partial \zeta_j}{\partial y} \\
\frac{\partial \zeta_j}{\partial y} & \frac{\partial \zeta_j}{\partial x}
\end{bmatrix}, \quad j = 1, 2, 3
\]
By substituting Eq. [8] into Eq. [6], the unknown coefficient \{a\} is determined as

$$a = \sum_{i=1}^{n} \sum_{j=1}^{p} \left( \left[ N_i^T N_j \right]^{-1} \left[ N_i^T B_j^T \{ u_i \} \right] \right)$$  \[10\]

Substituting Eq. [10] into Eq. [1], we obtain

$$\{ \varepsilon (x) \} = \left[ N_i \right] \sum_{i=1}^{n} \sum_{j=1}^{p} \left( \left[ N_i^T N_j \right]^{-1} \left[ N_i^T B_j^T \{ u_i \} \right] \right) = [A] \{ u \}$$ \[11\]

where

$$[A] = \left[ N_i \right] \sum_{i=1}^{n} \sum_{j=1}^{p} \left( \left[ N_i^T N_j \right]^{-1} \left[ N_i^T B_j^T \right] \right)$$ \[12\]

In the elasticity problem, the stress vector \{ \sigma \} and the strain vector \{ \varepsilon \} have the relation as follows,

$$\{ \sigma \} = [D] \{ \varepsilon \}$$ \[13\]

where \([D]\) is the symmetric matrix of material stiffness. With using Eq. [12], the stiffness matrix based on the localized least square method is calculated on the clustered local elements as

$$[k_{LS}] = \int_{\Omega} \{ A \}^T [D] \{ A \} d\Omega$$ \[14\]

where \(\Omega\) is the clustered local elements. It is important to note that the above stiffness matrix is computed in a node-wise manner.

It is also noted that EFMM-LS is closely related to the superconvergent patch recovery proposed by Zienkiewicz and Zhu (Z–Z).\(^{70,71}\) In the adaptive finite element method,\(^{72,73}\) the Z–Z error estimator has been most widely used to estimate the error. The error estimator requires an exact solution, but generally it is impossible to have it. The Z–Z technique then obtains the recovered solution in a post processing stage. The clustered local elements in the present method are equivalent to the superconvergent patch used in the Z–Z technique. The difference between these appears in the fact that the recovering procedure in EFMM-LS is in a main process stage when computing element stiffness matrices. The use of the assumed strain is, in some sense, equivalent to the “post-process” of the Z–Z superconvergent patch recovery.

3.3 EFMM based on Hellinger–Reissner principle. The Hellinger–Reissner (HR) variational principle is employed in EFMM based on the Hellinger–Reissner principle (EFMM-HR)\(^74\) to improve the accuracy of solutions. Let the HR principle of a linear elastic body be defined on the clustered local elements by

$$\Pi(\varepsilon, u) = \int_{\Omega} \{ \varepsilon \}^T \{ D \} \{ \varepsilon \} d\Omega - \frac{1}{2} \int_{\Omega} \{ \varepsilon \}^T \{ D \} \{ \partial u \} d\Omega - \int_{\partial \Omega} \{ u \}^T \{ t \} dS$$ \[15\]
where
\[
\{\dot{\mathbf{u}}\} = \mathbf{B}\{\mathbf{u}\}, \ \{\varepsilon\} = \mathbf{N}^T\{\mathbf{e}\} \tag{16}
\]
with \{\mathbf{b}\} being the applied body force per unit mass, and \{\mathbf{t}\} the applied traction on the boundary \(\mathcal{S}_b\), \{\mathbf{u}\} the unknown nodal displacement and \{\mathbf{e}\} the unknown nodal strain. The unknown values \(\{\mathbf{u}, \varepsilon\}\) of the HR principle satisfy the following equations in a weak manner,
\[
\int_{\Omega} \delta\{\varepsilon\}^T \mathbf{D} (\mathbf{B}\{\mathbf{u}\} - \mathbf{N}\{\mathbf{e}\}) d\Omega = 0 \tag{17}
\]
and
\[
\int_{\Omega} \delta\{\mathbf{u}\}^T \mathbf{B}^T \mathbf{D} \mathbf{N}\{\mathbf{u}\} d\Omega - \int_{\mathcal{S}_e} \delta\{\mathbf{u}\}^T \mathbf{b} dS = 0 \tag{18}
\]
It is noted here that the strain field is defined on the clustered local elements by a node-wise manner, whereas the displacement field is defined on each element by an element-wise manner. Equations [17] and [18] yield the linear matrix equation as follows,
\[
\begin{bmatrix}
-\mathbf{A} & \mathbf{C} \\
\mathbf{C}^T & 0 \\
\end{bmatrix}
\begin{bmatrix}
\{\bar{\varepsilon}\} \\
\{\bar{\mathbf{u}}\} \\
\end{bmatrix}
= \begin{bmatrix}
\{\mathbf{f}_1\} \\
\{\mathbf{f}_2\} \\
\end{bmatrix} \tag{19}
\]
where
\[
\mathbf{A} = \int_{\Omega} \mathbf{N}^T \mathbf{D} \mathbf{N} d\Omega
\]
\[
\mathbf{C} = \int_{\Omega} \mathbf{N}^T \mathbf{D} \mathbf{B} d\Omega
\]
\[
\mathbf{f}_1 = 0
\]
\[
\mathbf{f}_2 = \int_{\Omega} \mathbf{N}^T \mathbf{b} d\Omega + \int_{\mathcal{S}_e} \mathbf{N}^T \mathbf{t} dS
\]
By condensing the coefficient matrix of Eq. [19], we obtain the following equation:
\[
\mathbf{C}^T (\mathbf{A}^{-1} \mathbf{C}) \{\mathbf{u}\} = \{\mathbf{f}_2\} \tag{21}
\]
where the condensation should be performed on the clustered local elements. Thus, the stiffness matrix based on the HR principle is computed on the clustered local elements as follows,
\[
[k_{HR}] = \mathbf{C}^T \mathbf{A}^{-1} \mathbf{C} \tag{22}
\]
It is noted that we can obtain the enriched stiffness matrix without increasing the degrees of freedom of nodes.

3.4 Convergence study. Two kinds of error norms for a beam bending problem as shown in Fig. 23 are taken, which are, respectively, given as
\[
\|E\|_2 = \left[ \int_\Omega (\mathbf{u} - \mathbf{u}^{exact})^T (\mathbf{u} - \mathbf{u}^{exact}) d\Omega \right]^{1/2} \tag{23}
\]
and
\[
\|E\|_e = \left[ \int_\Omega \frac{1}{2} (\varepsilon - \varepsilon^{exact})^T (\sigma - \sigma^{exact}) d\Omega \right]^{1/2} \tag{24}
\]
where \(\|E\|_2\) is the displacement error norm and \(\|E\|_e\) the energy error norm, respectively. \(\mathbf{u}, \varepsilon, \sigma\) are, respectively, the numerical results of displacement, strain and stress, whereas \(\mathbf{u}^{exact}, \varepsilon^{exact}\) and \(\sigma^{exact}\) the exact solutions. Assume that the beam of length \(L = 10\), height \(D = 2\) and thickness \(t = 1\) is subjected to a shear load in plane stress. The material parameters are given as Young’s modulus \(E = 1000\) and Poisson’s ratio \(v = 0.25\). The above displacement and energy convergence norms are plotted against the DOFs in Figs. 24 and 25, respectively, where the meshes employed are \(1 \times 1\) (see Fig. 23), \(2 \times 2, 4 \times 4, 8 \times 8, 16 \times 16, 32 \times 32\) and \(64 \times 64\), respectively. It can be seen from these figures that
1. The error norms of displacement of EFMMs are between those of the linear and the quadratic FEMs (see Fig. 24). However, the convergence rates of EFMMs are almost equal to that of the quadratic FEM.
2. The error norms of energy of the quadratic EFMMs are almost the same as that of the quadratic FEM, whereas those of the linear EFMMs are between the linear and the quadratic FEMs.

The patch test is finally performed using the three models of patch as shown in Fig. 26, where the displacement field:
\[
\begin{align*}
\{\mathbf{u}(x)\} &= \begin{cases} 0.2x \\ -0.6y \end{cases} \tag{25}
\end{align*}
\]
is applied at the boundary. Table 1 shows the results for FEMs and EFMMs. As shown in the table, all of the methods pass the patch test for the Model A,
which is a regular mesh division model. However, for the Models B and C, which are irregular ones, EFMM-LSs fail to pass the test. Here, “Pass” means that the displacement of the internal node (M1 or M2) satisfies Eq. [25]. This implies that EFMM-LSs are of non-conforming type for irregular mesh.

Summarized here are that the accuracy of EFMM are equivalent to that of FEM (quadratic), which is considered to be the most accurate, although the CPU time and the required memory size are found much less compared with those of the latter.

4. Concluding remarks

In the present paper, the fundamental conception, the algorithms and the accuracy of the Free Mesh Method (FMM), are reviewed, which aims at
a particle-like finite element analysis of problems that are difficult to handle with using global mesh generation, such as moving boundaries problems, large deformation problems, crack propagation analysis, the separation or unification of bodies and adaptive mesh refinement analysis, especially on parallel processors. In the method, local finite elements are generated around each node, with local mesh data structures and a system of equations being created based on these nodes. As such, FMM can overcome difficulties associated with the distortion of elements by simply adding or deleting nodes, similar to particle methods. This characteristic is particularly advantageous when a parallel computer is employed. While parallelization of mesh generation is generally difficult, only the distribution of nodes needs to be considered in this method to perform parallel remeshing that favors excellent load balancing between processors. This node-based finite element computation is realized by a robust local mesh generation technique based on the gift-wrapping method. However, the study that still remains is that fast and robust generation of nodes, especially on massively parallel processors. A probabilistic node generation approach with centroidal Voronoi tessellation under a parallel environment will be the most promising.

Next, a new Free Mesh Method called Enriched Free Mesh Method (EFMM) is taken, by which a high accuracy can be obtained without explicitly increasing the degrees of freedom. The central idea of EFMM is that the strain field is assumed on clustered local elements in addition to the usual FEM displacement field on each element. To relate the above two fields, the localized least square method or the Hellinger–Reissner principle are independently employed. The convergence characteristics of the displacement error norms are between that of FEM with the linear displacement field and that with the quadratic one, whereas that of the energy error norms with the quadratic strain field for the clustered elements is equivalent to that of FEM with the quadratic displacement field. EFMM based on the Hellinger–Reissner principle passes the patch test, whereas that based on the localized least square method does not for irregular nodal arrangements. This would be an open question and there is a room for future research.

Acknowledgments

I would like to thank all the researchers and students, who have been involved in the study of FMM with the present author, which has been performed under the sponsorship of University of Tokyo, Toyo University and Sungkyunkwan University in Korea. I am grateful to Professor Tamio Yamakawa, Editor-in-Chief for Proceedings of the Japan Academy, Series B, for his encouragement to prepare this review paper.

References

1) Zienkiewicz, O.C. and Taylor, R.L. (2000) The Finite Element Method. 5th ed. Butterworth Heinemann, Oxford.
2) Liu, G.R. and Quek, S.S. (2002) The Finite Element Method: A Practical Course. Butterworth Heinemann, Oxford.
3) Kleiber, M. and Borkowski, A. (1998) Handbook of Computational Solid Mechanics. Springer-Verlag, Berlin.
4) Leveque, R.J. (2002) Finite Volume Methods for Hyperbolic Problems. Cambridge Univ. Press, New York.
5) Yagawa, G. (1990) Parallel techniques for computational mechanics. Theor. Appl. Mech. 39, 3–9.
6) Papadrakakis, M. (ed.) (1997) Parallel Solution Methods in Computational Mechanics. Wiley, Chichester.
7) Tezduyar, T.E. and Osawa, Y. (1999) Methods for parallel computation of complex flow problems. Parallel Comput. 25, 2039–2066.
8) Peraire, J., Vahdati, M., Morgan, K. and Zienkiewicz, O.C. (1987) Adaptive remeshing for compressible flow computations. J. Comput. Phys. 72, 449–466.
9) Lohner, R. and Baum, J.D. (1992) Adaptive h-refinement on 3D unstructured grids for transient problems. Int. J. Numer. Methods Fluids 14, 1407–1419.
10) de Cougny, H.L., Shephard, M.S. and Ozturan, C. (1994) Parallel three-dimensional mesh generation. Comput. Syst. Eng. 5, 311–323.
11) Lohner, R. (2001) A parallel advancing front grid generation scheme. Int. J. Numer. Methods Eng. 51, 663–678.
12) Belytschko, T., Krongauz, Y., Organ, D., Fleming, M. and Krysl, P. (1996) Meshless methods: an overview and recent developments. Comput. Methods Appl. Mech. Eng. 139, 3–47.
13) Lucy, L.B. (1977) A numerical approach to the testing of the fission hypothesis. Astron. J. 82, 1013–1024.
14) Gingold, R.A. and Monaghan, J.J. (1977) Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon. Not. R. Astron. Soc. 181, 375–389.
15) Monaghan, J.J. (1988) An introduction to SPH. Comput. Phys. Commun. 48, 89–96.
16) Randles, P.W. and Libersky, L.D. (1996) Smoothed particle hydrodynamics: some recent improvements and applications. Comput. Methods Appl. Mech. Eng. 139, 375–408.
17) Nayroles, B., Touzot, G. and Villon, P. (1992) Generalizing the finite element method: diffuse approximation and diffuse elements. Comput. Meth. 10, 297–318.
18) Batina, J.T. (1993) A gridless Euler/Navier–Stokes solution algorithm for complex aircraft applications. AIAA Paper. 93–0333.
19) Morinishi, K. (1995) A gridless type solution for high Reynolds number multielement flow fields. AIAA Paper. 95–1856.
20) Belytschko, T., Lu, Y.Y. and Gu, L. (1994) Element-free Galerkin methods. Int. J. Numer. Methods Eng. 37, 229–256.
21) Lu, Y.Y., Belytschko, T. and Gu, L. (1994) A new implementation of the element free Galerkin method. Comput. Methods Appl. Mech. Eng. 113, 397–414.
22) Noguchi, H., Kawashima, T. and Miyamura, T. (2000) Element free analyses of shell and spatial structures. Int. J. Numer. Methods Eng. 47, 1215–1240.
23) Liu, W.K., Jun, S., Li, S., Adee, J. and Belytschko, T. (1995) Reproducing kernel particle methods for structural dynamics. Int. J. Numer. Methods Eng. 38, 1655–1679.
24) Duarte, C.A. and Oden, J.T. (1996) An h-p adaptive method using clouds. Comput. Methods Appl. Mech. Eng. 139, 257–262.
25) Liszka, T.J., Duarte, C.A.M. and Tworzydło, W.W. (1996) Hp-Meshless cloud method. Comput. Methods Appl. Mech. Eng. 139, 263–288.
26) Onate, E., Idelsohn, S., Zienkiewicz, O.C., Taylor, R.L. and Sacco, C. (1996) A stabilized finite point method for analysis of fluid mechanics problems. Comput. Methods Appl. Mech. Eng. 139, 315–346.
27) Koshizuka, S. and Oka, Y. (1996) Moving-particle semi-implicit method for segmentation of incompressible fluid. Nucl. Sci. Eng. 123, 421–434.
28) Atluri, S.N. and Zhu, T.L. (2000) New concepts in meshless methods. Int. J. Numer. Methods Eng. 47, 537–556.
29) Atluri, S.N. and Zhu, T.L. (1998) A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics. Comput. Mech. 22, 117–127.
30) Liu, G.R. and Ke, B.T. (2006) A stabilized least-squares radial point collocation method (LS-RPCM) for adaptive analysis. Comput. Methods Appl. Mech. Eng. 195, 4843–4861.
31) Ke, B.T., Liu, G.R. and Lu, C. (2007) A regularized least-squares radial point collocation method (RLS-RPCM) for adaptive analysis. Comput. Meth. 40, 837–853.
32) Shi, G.H. (1991) Manifold method of material analysis. Trans. 9th Army Conf. on Appl. Math. Comput., Report No. 92-1, U.S. Army Research Office.
33) Hollister, S.J. and Kikuchi, N. (1994) Homogenization theory and digital imaging: a basis for studying the mechanics and design principles of bone tissue. Biotechnol. Bioeng. 43, 586–596.
34) Melenk, J.M. (1995) On generalized finite element methods. Ph.D. Dissertation. Univ. of Maryland, College Park, MD.
35) Melenk, J.M. and Babuska, I. (1996) The partition of unity finite element method: basic theory and applications. Comput. Methods Appl. Mech. Eng. 139, 289–314.
36) Belytschko, T. and Black, T. (1999) Elastic crack growth in finite elements with minimal remeshing. Int. J. Numer. Methods Eng. 45, 601–620.
37) Belytschko, T., Moes, N., Usui, S. and Parimi, C. (2001) Arbitrary discontinuities in finite elements. Int. J. Numer. Methods Eng. 50, 993–1013.
38) Jin, C. and Suzuki, K. (2000) Methodology and property of cover least square approximation. Trans. Jpn. Soc. Comput. Eng. Sci. 2, 213–218.
39) Yagawa, G. and Yamada, T. (1996) Free mesh method: a new meshless finite element method. Comput. Mech. 8, 383–386.
40) Yagawa, G. and Hosokawa, T. (1997) Application of the free mesh method with Delaunay tessellation in a 3-dimensional problem. Trans. JSEM (A) 63, 2251–2256.
41) Yagawa, G. and Shirazaki, M. (1999) Parallel computing for incompressible flow using a nodal-based method. Comput. Meth. 23, 209–217.
42) Shirazaki, M. and Yagawa, G. (1999) Large-scale parallel flow analysis based on free mesh method: a virtually meshless method. Comput. Methods Appl. Mech. Eng. 174, 419–431.
43) Yagawa, G. and Furukawa, T. (2000) Recent developments of free mesh method. Int. J. Numer. Methods Eng. 47, 1419–1443.
44) Furukawa, T., Yang, C., Yagawa, G. and Wu, C. (2000) Quadrilateral approaches for accurate free mesh method. Int. J. Numer. Methods Eng. 47, 1445–1462.
45) Yagawa, G. and Furukawa, T. (2000) Recent developments of free mesh method. Int. J. Numer. Methods Eng. 47, 1419–1443.
46) Yamashita, D. and Yagawa, G. (2001) Accuracy study due to the connectivity inconsistency in the free mesh method. Proc. 50th National Congress of Theor. Appl. Mech., Tokyo. 155–156.
47) Fujisawa, T., Inaba, M. and Yagawa, G. (2003) Parallel computing of high-speed compressible flows using a node-based finite element method. Int. J. Numer. Methods Eng. 58, 481–511.
48) Yagawa, G. (2004) Node-by-node parallel finite elements. Int. J. Numer. Methods Eng. 60, 69–102.
49) Tian, A. and Yagawa, G. (2007) Non-matching mesh gluing by meshless interpolation—An alternative to Lagrange multipliers. Int. J. Numer. Methods Eng. 71, 473–503.
50) Yagawa, G. and Matsubara, H. (2007) Enriched free mesh method: an accuracy improvement for node-based FEM. Comput. Methods Appl. Sci. 7, 207–219.
51) Matsubara, H. and Yagawa, G. (2009) Convergence studies for enriched free mesh method and its application to fracture mechanics. Int. J. Interact. Multiscale Mechanics 2, 277–293.
Kobayashi, Y. and Yagawa, G. (2009) Compatibility of enriched free mesh method and ultra large scale parallel structural analysis. Theor. Appl. Mech. 57, 377–383.

Nagaoka, S. and Yagawa, G. (2009) Fracture analysis using enriched free mesh method with a new boundary treatment. Theor. Appl. Mech. 57, 411–420.

Bowyer, A. (1981) Computing Dirichlet tessellations. Comput. J. 24, 162–166.

Watson, D.F. (1981) Computing n-dimensional Delaunay tessellation with application to Voronoi polytopes. Comput. J. 24, 167–172.

Cavendish, J.C., Field, D.A. and Frey, W.H. (1985) An approach to automatic three-dimensional finite element mesh generation. Int. J. Numer. Methods Eng. 21, 329–347.

Schroeder, W.J. and Shephard, M.S. (1988) Geometry-based fully automatic mesh generation and the Delaunay triangulation. Int. J. Numer. Methods Eng. 26, 2503–2515.

Inaba, M., Fujisawa, T. and Yagawa, G. (2002) A new algorithm of local mesh generation for the free mesh method. Proc. JSME Dynamics and Design Conference.

Tanemura, M., Ogawa, T. and Ogita, N. (1983) A new algorithm for three-dimensional Voronoi diagrams in linear expected time. Discrete Comput. Geom. 6, 343–367.

Lee, D.T. and Lin, A.K. (1986) Generalized Delaunay triangulations for planar graphs. Discrete Comput. Geom. 1, 201–217.

Crew, L.P. (1989) Constrained Delaunay triangulations. Algorithmica 48, 97–108.

Shewchuk, J.R. (1998) A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations. Proc. 14th Annual Symp. on Comput. Geom., New York, pp. 76–85.

Du, Q., Gunzburger, M. and Ju, L. (2001) Meshfree, probabilistic determination of point sets and regions for meshless computing. Tech. Rep. No. AM248, Department of Mathematics, Pennsylvania State Univ.

Ju, L., Du, Q. and Gunzburger, M. (2001) Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations. Techn. Rep. No. AM250, Department of Mathematics, Pennsylvania State Univ.

Winget, J.M. and Hughes, T.J.R. (1985) Solution algorithms for nonlinear transient heat conduction analysis employing element-by-element iterative strategies. Comput. Methods Appl. Mech. Eng. 52, 711–815.

Nakabayashi, Y., Okuda, H. and Yagawa, G. (1996) Parallel finite element fluid analysis on an element-by-element basis. Comput. Mech. 18, 377–382.

Zienkiewicz, O.C. and Morgan, K. (2006) Finite Element and Approximation. Dover Publications, New York.

Zienkiewicz, O.C. and Zhu, J.Z. (1992) The superconvergent patch recovery and a posteriori error estimates. PART1: The recovery technique. Int. J. Numer. Methods Eng. 33, 1331–1364.

Zienkiewicz, O.C. and Zhu, J.Z. (1992) The superconvergent patch recovery and a posteriori error estimates. PART2: Error estimates and adaptivity. Int. J. Numer. Methods Eng. 33, 1365–1382.

Zienkiewicz, O.C. and Zhu, J.Z. (1992) The superconvergent patch recovery (SPR) and adaptive finite element refinement. Comput. Methods Eng. 101, 207–224.

Babuska, I. and Rheinboldt, W.C. (1978) A-posteriori error estimates for the finite element method. Int. J. Numer. Methods Eng. 12, 1597–1615.

Washizu, K. (1968) Variational Methods in Elasticity and Plasticity. Pergamon Press, New York.

Timoshenko, S.P. and Goodier, J.N. (1987) Theory of Elasticity. McGraw-Hill, India.

(Rceived Dec. 14, 2010; accepted Jan. 28, 2011)
Profile

Genki Yagawa was born in 1942 and started his research career in 1965 with studies on the finite element method superposed on the analytical solution, applying to linear and nonlinear fracture mechanics, after graduating from the Faculty of Engineering at the University of Tokyo. Among the most important works he has accomplished is a massively parallel computing method, where the hierarchical domain decomposition together with the variational method to satisfy the boundary conditions between the sub-domains is employed. It has been proved that the method can solve huge scale problems of billions-degrees of freedom with high accuracy and high performance when applied to computational solid and fluid mechanics problems. He has also created a new method called the Free Mesh Method, where both pre-processing and main-processing of the finite element analysis can be parallelized easily in terms of nodes. These methods, being quite suitable for massively parallel environments and embedded within the internationally well-known codes, have been employed by many researchers and engineers worldwide.

He became Lecturer in 1970, Associate Professor in 1971, Professor in 1984 and Council Member in 1998 at the University of Tokyo. He is currently Emeritus Professor at the University of Tokyo, Professor and Director at the Computational Mechanics Research Center of Toyo University, Visiting Professor at Sungkyunkwan University in Korea, Council Member of Science Council of Japan, and President of International Association for Computational Mechanics.

Among his awards and honors are the Japan Academy Prize (2009), the Toray Science and Technology Prize (2009), the International Association for Computational Mechanics (IACM) Award (2008), the Asia-Pacific Association for Computational Mechanics (APACM) Zienkiewicz Medal (2007), the Japan Association for Computational Mechanics (JACM) Computational Mechanics Award (2004), and Honorary Doctor endowed from Iasi Technical University (Romania) (2010). He is also an Honorary Member of JSME, and a Fellow of IACM, ASME, JSIAM and AESJ.