ELEMENT FREE FORMULATION USED FOR CONNECTING DOMAIN BOUNDARIES

Hsing-Chih Tsai and Chan-Ping Pan*

ABSTRACT

The mapping method is widely used for automatic mesh generation. This method is used in this paper to generate a mesh, which is completely controlled by the program. The user is expected to define the problem in the most simple and natural way. The concept of mesh is not required for the user at all. To accomplish this purpose, several problems need to be solved. One is to combine connecting domains with different numbers of elements. Four methods are developed and are compared in this paper. The element free concept is used by one of the methods. A polynomial boundary is established by the moving least square formulation. Therefore the nodes in neighbor areas are included to get the coefficients of this polynomial. The result is pretty good compared to the traditional methods.

Key Words: element free, moving least square approximation, mapping method, compatibility, penalty method, mixed-type elements.

I. INTRODUCTION

According to the space characteristics of various analysis systems, there are plane geometry, surface geometry and space geometry categories. How to establish elements and nodes for a certain problem is the basic procedure for all analyses. In recent years, many research results have been related to the development of automatic mesh generation. In a two dimensional plane problem, a triangle is the simplest form of element shape used in automatic mesh generation (Cavendish, 1974; Cohen, 1980; Haber *et al*., 1981; Brown, 1981; Ho-Le, 1988). However, convergence behavior of a triangle element is inferior to that of a quadrilateral element. However, intended quadrilateral element meshes will largely increase the difficulty of mesh generation. (Zienkiewicz and Phillips, 1971; Gordon and Hall, 1973; Wordenweber, 1984; Kikuchi, 1985; Talbert and Parkinson, 1990; Chinnaswamy *et al*., 1991; Blacker and Stephenson, 1991; Zhu *et al*., 1991; Rank *et al*, 1993; Lee and Lo, 1994; Cheng and Li, 1996; Sarrate and Huerta, 2000; Kwak *et al*., 2002). Therefore, many scholars have put their efforts into research on mesh generation.

General software packages for commercial use which have automatic mesh generation are largely based on the mapping method (Cohen, 1980; Haber *et al*., 1981; Brown, 1981; Ho-Le, 1988; Zienkiewicz and Phillips, 1971; Gordon and Hall, 1973; Kikuchi, 1985; Chinnaswamy *et al*., 1991; Cheng and Li, 1996). This method is fast, simple and easily controls geometry and mesh density as well as being applicable to highly changeable geometric boundaries and mesh shapes. Its results are of excellent quality.

To get a mesh which is divided by its degree of requirement (Haber *et al*., 1981; Chinnaswamy *et al*., 1991; Cheng and Li, 1996) will affect the speed of convergence and the time consumption for the problem analysis. Moreover, a mesh generation method is judged by its efficiency, convenience and simplicity. The encountered problems for a graded mesh include aspect ratio, distortion, compatibility, etc. Some methods adopt a rigid mesh form and then a mesh smoothing (Cheng and Li, 1996; Kwak *et al*., 2002) procedure to allow for the improvement of aspect ratio and distortion of the elements. If mesh division is based on a multi-region/multi-block mapping method, displacement compatibility between

*Corresponding author. (Tel: 886-2-27376586; Fax: 886-2-27376606; Email: panpe@ms67.hinet.net)

The authors are with the Department of Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan 106, R.O.C.

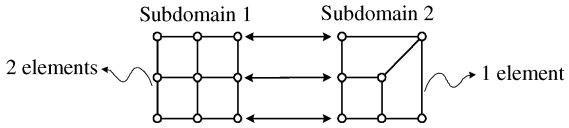

Fig. 1 Unstructured meshes

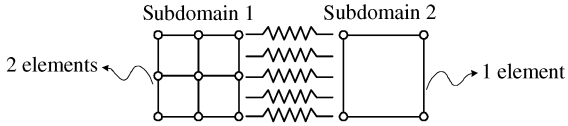

Fig. 2 Structured meshes

sub-domains will become an important factor to be considered during the mesh generation procedure. The adjacent boundary between two sub-domains is usually divided into elements of the same form (Fig. 1). Therefore, each sub-domain will not, independently, set up a mesh division. Some adjustment must be made for the mesh division of adjacent sub-domains. It could distort subsequent element shapes and degrade result's quality. In order to assure compatibility between sub-domains, mesh adjustment will produce the following two undesirable situations:

1. The mesh action will constantly expand outwards, necessitating re-mesh of many sub-domains.
2. In order to prevent the expansion of re-meshing, distorted mesh could appear in a certain sub-domain.

In general, an unstructured mesh could result from the requirement of a fine-tuned graded mesh. Due to the correlation between the number of nodes and the number of elements, an unstructured mesh does not posses the relevant regularity. On the contrary, a structured mesh is simple and easy to construct in a program. Therefore, the main purpose of this paper is to establish the compatibility between two sub-domains, which are free to make their own mesh divisions (Fig. 2).

To maintain the compatibility of the interfaces, some methods are employed for the conjunction of sub-domains. Some examples are the penalty method (Arora *et al*., 1991), the Lagrange multiplier method (Houlsby *et al*., 2000), the master-slave concept (Dohrmann *et al*., 2000), the pseudo-node method (Aminpour *et al*., 1995) etc.

II. FOUR METHODS USED FOR THE DISPLACEMENT COMPATIBILITY BETWEEN SUBDOMAINS

Four methods developed in this paper:

Method I: Penalty Method

Method II: Direct Transformation Method (use 5-point polynomial function to define the common boundary)

Method III: Transformation Method with Modified Elements (use 5-point polynomial function to define the common boundary and modified elements for the adjacent area)

Method IV: MLSA Method (use moving least square approximation to formulate the common boundary)

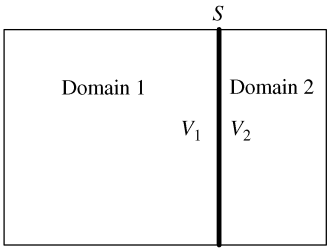

Fig. 3 Sub-domain diagram

1. Penalty Method

The penalty method is used to reduce displacement difference between left and right sides by a relatively strong spring to reach the requirement of displacement compatibility. Strain energy due to the penalty spring (U_P)

$$U_P = \int_s \frac{1}{2}(v_1 - v_2)E_P(v_1 - v_2)ds \tag{1}$$

v_1 and v_2 of the above formula are the displacements of the left and right domains on the adjacent boundary. E_P is the distributed penalty spring constant of this method. This constant is determined by a value relatively to the elastic modulus of the problem (E). The value adopted in this paper is $E_P=E\times10^8$. s represents the adjacent boundary of the two domains (Fig. 3).

For the common boundary shown in Fig. 3.

$$\begin{cases} v_1 = \sum \boldsymbol{N_i u_i} \\ v_2 = \sum \boldsymbol{N_j u_j} \end{cases} \tag{2}$$

v_1 and v_2 represent the displacements simulated in domain 1 and domain 2. $\boldsymbol{N_i}$ and $\boldsymbol{N_j}$ are shape functions used by each element. $\boldsymbol{u_i}$ and $\boldsymbol{u_j}$ are nodal displacements related to each element.

Therefore, (1) can be written as:

$$U_P = \frac{1}{2}\{\boldsymbol{u_i}\ \boldsymbol{u_j}\}\int_s \begin{bmatrix} \boldsymbol{N_i} \\ -\boldsymbol{N_j} \end{bmatrix} E_P[\boldsymbol{N_i}\ -\boldsymbol{N_j}]ds \begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_j} \end{Bmatrix} \tag{3}$$

Penalty stiffness matrix [$\boldsymbol{K_p}$] :

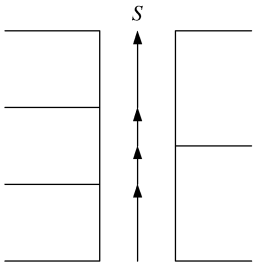

Fig. 4 Gauss quadrature segments for penalty method

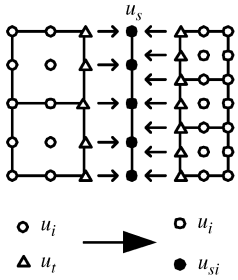

Fig. 5 Transformation of degrees of Freedom

$$[\boldsymbol{K_P}] = \int_s \begin{bmatrix} \boldsymbol{N_i} \\ -\boldsymbol{N_j} \end{bmatrix} E_P [\boldsymbol{N_i} \; -\boldsymbol{N_j}] ds \tag{4}$$

Gauss quadrature is adopted in this paper to do the integration. Each integration segment contains only one element on each side to ensure a systematic programming procedure. The example shown in Fig. 4 contains three elements on the left side and two elements on the right side. Four integration segments are defined to reach the requirement.

Merits of This Method:

This method is simple and easy to understand. It can be applied not only to a problem with the same element type used on both sides, but also to a problem with different element types on each side.

Demerits of This Method:

This method is an approximated method. The displacements on each side are not the same. However, the differences between them are very small. The accuracy of approximation depends on the penalty spring constant chosen by the developer. The bandwidth of stiffness matrix is increased significantly by this method. Therefore, the execution time is increased also.

2. Direct Transformation Method

The nodes and elements on each side determine the corresponding displacement field. The elements and nodes are developed by the automatic mesh procedure. However, proper procedure must be implemented to ensure the compatibility and force transmission of these sides. The third individual displacement field is assumed for the common side of these adjacent elements. Therefore, the corresponding boundary nodes on each side are no longer independent. They must be transformed into the common displacement field. A polynomial function defined by five nodal displacements is assumed in this method. The combination of element types and the displacement field assumed for the common boundary can be very varied freely. The element type used for the example problems is the 9-point plane stress Lagrange element. The displacement field on the element boundary is a quadratic polynomial. In order to improve the accuracy, a fourth degree polynomial is chosen to represent the common boundary.

The strain energy of a typical element connected to the common boundary:

$$U = \frac{1}{2}\{\boldsymbol{u_i} \; \boldsymbol{u_t}\} \int [\boldsymbol{B}]^T [\boldsymbol{D}][\boldsymbol{B}] dV \begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_t} \end{Bmatrix}$$

$$= \frac{1}{2}\{\boldsymbol{u_i} \; \boldsymbol{u_t}\}[\boldsymbol{K}] \begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_t} \end{Bmatrix} \tag{5}$$

$\{\boldsymbol{u}\}$ represents two independent displacements along the $\boldsymbol{X}$ and $\boldsymbol{Y}$ directions $\{\boldsymbol{u} \; \boldsymbol{v}\}^T$. In the above equation, $\boldsymbol{u_i}$ represents independent displacements of non-adjacent boundaries. $\boldsymbol{u_t}$ represents dependent nodal displacements on the adjacent boundaries (Fig. 5). $[\boldsymbol{K}]$ represents the stiffness matrix of the element. Displacement function of the common boundary:

$$u_s = \sum \boldsymbol{N_i}(\varsigma) \boldsymbol{u_{si}} \tag{6}$$

In which, u_s represents displacement on any point of the common boundary interpolated by shape functions $\boldsymbol{N_i}$. $\boldsymbol{u_{si}}$ represents the nodal displacements newly established on the common boundary (Fig. 5).

ς_t is the ς value of corresponding nodes on each adjacent boundary. Therefore, the nodal displacements of these nodes, u_t,, can be represented as follow:

$$u_t = \sum \boldsymbol{N_i}(\varsigma_t) \boldsymbol{u_{si}} \tag{7}$$

Displacement transformation matrix $[\boldsymbol{T}]$

$$\begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_t} \end{Bmatrix} = [\boldsymbol{T}] \begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_{si}} \end{Bmatrix} \tag{8}$$

(8) is substituted into (5), then the strain energy may be expressed as follow:

$$U = \frac{1}{2}\{\boldsymbol{u_i} \; \boldsymbol{u_{si}}\}[\boldsymbol{K_t}] \begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_{si}} \end{Bmatrix} \tag{9}$$

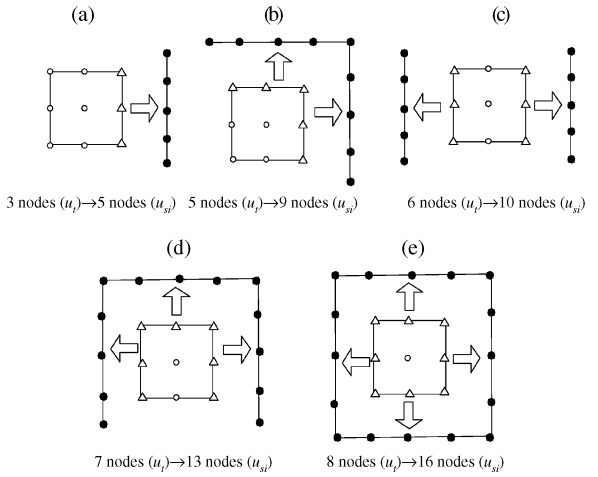

Fig. 6 3_to_5 transform templates for an Element

The corresponding stiffness matrix $[\boldsymbol{K_t}]$ after this transformation is

$$[\boldsymbol{K_t}]=[\boldsymbol{T}]^T[\boldsymbol{K}][\boldsymbol{T}] \tag{10}$$

Considering all possible situations in an element, five transformation templates are categorized in this paper. They include one common boundary, two common boundaries (on adjacent sides and on opposite sides), three common boundaries and four common boundaries (Fig. 6).

In general, the number of elements divided for each sub-domain is very large. Assume the number of elements used for each side of a sub-domain is not less than two. The transformation templates can be simplified to only two types. One is the single common boundary (Fig. 6(a)) and the other is two common boundaries on adjacent sides (Fig. 6(b)).

Merits of This Method:

This method is still an easy and simple way to solve the problem. Also, the total number of degrees of freedom in the global stiffness matrix is less than in the penalty method and the MLSA method. Therefore, the execution time can be cut down.

Demerits of This Method:

Since the polynomial degrees used for adjacent elements and for the common boundary are different, the displacement fields simulated by neighboring elements could be different. Take the formulation shown in this paper as an example. The displacement field used for the adjacent boundary of an element adopts a quadratic polynomial. The displacement field of the common boundary is described by quadratic polynomial. Only the nodal displacements of the adjacent boundaries are consistent with that of the simulated common boundary. There will be a discrepancy between adjacent boundaries and simulated common boundary. However, displacement differences of this kind can be reduced by mesh refinement. The shown examples can verify that the convergence is still very good for this method.

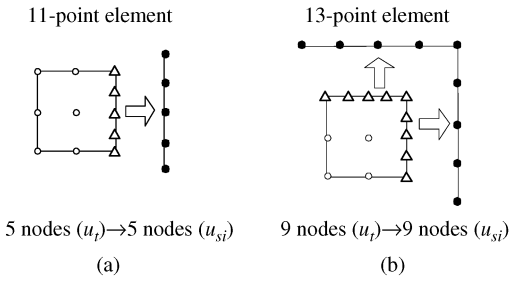

Fig. 7 Two required transformation templates and modified elements

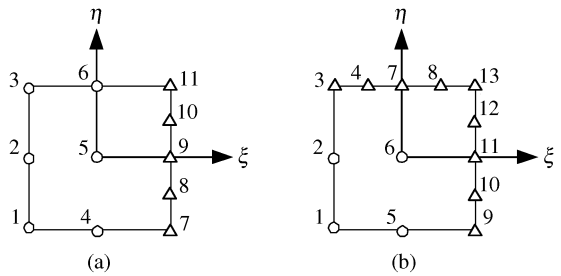

Fig. 8 11-point and 13-point modified elements

3. Transformation Method with Modified Elements

The same as in the previous method, a polynomial function is used to simulate the displacement field of the common boundary. However, the displacement fields used for the adjacent elements are modified to match the polynomial degree used for the common boundary. The plane stress element used in this paper is the 9-point Lagrange element. To reach the requirement of compatibility, two modified elements are derived. The first one is derived for elements with one adjacent boundary. The nodes used for the adjacent boundary will be modified to be five nodes. Therefore, the total number of nodes in an element will be eleven. Also the displacement field on this boundary will be modified to be a quadratic polynomial. The second formulation is for elements with two adjacent boundaries side by side. The total number of nodes will be thirteen (Fig. 7). These two elements are selected because of the assumption that the number of elements used along a sub-domain boundary is not less than two (Fig. 7).

The derived shape functions, $\boldsymbol{N_i}$, for the 11-point element are as follows (Fig. 8(a)):

$$N_1=1/4\xi(\xi-1)\eta(\eta-1) \quad (11a)$$

$$N_2=-1/2\xi(\xi-1)(\eta+1)(\eta-1) \quad (11b)$$

$$N_3=1/4\xi(\xi-1)(\eta+1)\eta \quad (11c)$$

$$N_4=-1/2(\xi+1)(\xi-1)\eta(\eta-1) \quad (11d)$$

$$N_5=(\xi+1)(\xi-1)(\eta+1)(\eta-1) \quad (11e)$$

$$N_6=-1/2(\xi+1)(\xi-1)(\eta+1)\eta \quad (11f)$$

$$N_8=-4/3(\xi+1)\xi(\eta+1)\eta(\eta-1/2)(\eta-1) \quad (11g)$$

$$N_{10}=-4/3(\xi+1)\xi(\eta+1)(\eta+1/2)\eta(\eta-1) \quad (11h)$$

$$N_7=1/4(\xi+1)\xi\eta(\eta-1)-3/8N_8+1/8N_{10} \quad (11i)$$

$$N_{11}=1/4(\xi+1)\xi(\eta+1)\eta+1/8N_8-3/8N_{10} \quad (11j)$$

$$N_9=-1/2(\xi+1)\xi(\eta+1)(\eta-1)-3/4N_8-3/4N_{10} \quad (11k)$$

The derived shape functions, N_i, for the 13-point element are as follows (Fig. 8(b)):

$$N_1=1/4\xi(\xi-1)\eta(\eta-1) \quad (12a)$$

$$N_2=-1/2\xi(\xi-1)(\eta+1)(\eta-1) \quad (12b)$$

$$N_5=-1/2(\xi+1)(\xi-1)\eta(\eta-1) \quad (12c)$$

$$N_6=(\xi+1)(\xi-1)(\eta+1)(\eta-1) \quad (12d)$$

$$N_4=-4/3(\xi+1)\xi(\xi-1/2)(\xi-1)(\eta+1)\eta \quad (12e)$$

$$N_8=-4/3(\xi+1)(\xi+1/2)\xi(\xi-1)(\eta+1)\eta \quad (12f)$$

$$N_{10}=-4/3(\xi+1)\xi(\eta+1)\eta(\eta-1/2)(\eta-1) \quad (12h)$$

$$N_{12}=-4/3(\xi+1)\xi(\eta+1)\eta(\eta+1/2)\eta(\eta-1) \quad (12i)$$

$$N_3=1/4\xi(\xi-1)(\eta+1)\eta-3/8N_4+1/8N_8 \quad (12j)$$

$$N_9=1/4(\xi+1)\xi\eta(\eta-1)-3/8N_{10}+1/8N_{12} \quad (12k)$$

$$N_{13}=1/4(\xi+1)\xi(\eta+1)\eta+1/8(N_4+N_{10}) -3/8(N_8+N_{12}) \quad (12l)$$

$$N_7=-1/2(\xi+1)(\xi-1)(\eta+1)\eta-3/4N_4-3/4N_8 \quad (12m)$$

$$N_{11}=-1/2(\xi+1)\xi(\eta+1)(\eta-1)-3/4N_{10}-3/4N_{12} \quad (12n)$$

Merits of This Method:

Compared to the previous method, full compatibility can be achieved on the adjacent boundaries. Also, the total number of degrees of freedom of the global stiffness matrix is smaller than in the penalty method and MLSA method.

Demerits of This Method:

The derivation of this method is relatively complicated, therefore it is not easy to change the polynomial degree used for the common boundary. Also, any change to the element type or to the simulated model lead to tedious modification.

4. MLSA Method

Since the polynomial degree used to simulate the common boundary in method II and method III is bound to the number of nodes, the polynomial degree is always equal to the number of nodes minus one. In order to get a better approximation, more nodes are required to simulate this common boundary. However, more nodes make the difference of polynomial degrees between the common boundary and adjacent elements get bigger. They make the incompatibility condition in method II get worse. This problem can be solved by a modification of the shape functions of adjacent elements in Method III. However, this derivation is tedious to do. Therefore, we use the element free formulation (Nayroles *et al*., 1992; Belytschko *et al*., 1994; Ginman, 1997) to define the common boundary by a derivation based on the moving least square approximation method (MLSA). The derivation of MLSA can produce a polynomial function, which is defined by a large number of nodes. Therefore the polynomial degree of the common boundary can be set to equal the one used for the adjacent elements. At the same time, the nodes used to describe the common boundary can be increased freely. There is a special characteristic of MLSA, that a simulated displacement field does not pass through the nodal displacements.

According to MLSA, (Gordon and Wixson, 1978; Lancaster and Salkauskas, 1981) the simulated displacement field of the common boundary, u_s, can be expressed as

$$u_s(\xi)=\sum_{j=1}^{m} b_j(\xi)a_j(\xi)\equiv \boldsymbol{b}^T(\xi)\boldsymbol{a}(\xi) \quad (13)$$

ξ is the variable representing the position of a point along the common boundary. $\boldsymbol{b}^T$ is the polynomial function used to simulate this boundary. m is the number of terms used in the polynomial function. In order to match the quadratic polynomial function used within an element, m is set to be three in this paper. Therefore

$$\boldsymbol{b}^T(\xi)=\lfloor 1 \quad \xi \quad \xi^2 \rfloor \quad (14)$$

$\boldsymbol{a}(\xi)$ are the unknown coefficients of the

simulated polynomial. Notice that $\boldsymbol{a}(\xi)$ are function of the position ξ, therefore they are changed continuously along the boundary. The matrix form of $\boldsymbol{a}(\xi)$ is

$$\boldsymbol{a}(\xi)=[a_0(\xi)\ \ a_1(\xi)\ \ a_2(\xi)]^T \tag{15}$$

A value, J, is defined as the weighted sum of the square of errors. The error is the difference between simulated displacement field u_s at nodes and the nodal displacements u_{si}.

$$J=\sum_{i=1}^{n} w(\xi-\xi_i)[\boldsymbol{b}^T(\xi_i)\boldsymbol{a}(\xi)-u_{si}]^2 \tag{16}$$

n is the total number of nodes defined in the common boundary. $w(\xi-\xi_i)$ are the weighting functions defined for each node. The matrix form of (16) is

$$J=(\boldsymbol{B}^T\boldsymbol{a}(\xi)-\boldsymbol{u_{si}})^T\boldsymbol{W}(\xi)(\boldsymbol{B}^T\boldsymbol{a}(\xi)-\boldsymbol{u_{si}}) \tag{17}$$

$$\boldsymbol{u_{si}}=\{u_1\ \ u_2\ \ \cdots\ \ u_n\}^T \tag{18}$$

$u_1, u_2, \cdots, u_n$ in (18) are unknown nodal displacements assumed on the common boundary.

$$\boldsymbol{B}=\{\boldsymbol{b}(\xi_1)\ \ \boldsymbol{b}(\xi_2)\ \ \cdots\ \boldsymbol{b}\ (\xi_n)\} \tag{19}$$

The matrix form of weighting functions is

$$\boldsymbol{W}(\xi)=\begin{bmatrix} w(\xi-\xi_1) & 0 & \cdots & 0 \\ 0 & w(\xi-\xi_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(\xi-\xi_n) \end{bmatrix} \tag{20}$$

In the above equation, $\xi_1, \xi_2, \cdots, \xi_n$ are ξ coordinates corresponding to the assumed nodes. The chosen weighting function is the modified exponential function:

$$w(\xi-\xi_i)=\begin{cases} \dfrac{e^{-(d_I/c)^{2k}}-e^{-(d_{mI}/c)^{2k}}}{1-e^{-(d_{mI}/c)^{2k}}} & d_I \le d_{mI} \\ 0 & d_I > d_{mI} \end{cases} \tag{21}$$

In the above equation, $d_I=\xi-\xi_i$; c is a constant to control the shape of the weighting function. k is a constant related to the order of differentiation required. k=1 is adopted in this paper. d_{mI} determines the influence range of a node. The minimum requirement is that m number of nodes must be included in d_{mI}.

The unknown coefficients $\boldsymbol{a}(\xi)$ are determined by making J to be the minimum. This formulation is the moving least square approximation.

$$\frac{\partial J}{\partial a}=\boldsymbol{H}(\xi)\boldsymbol{a}(\xi)-\boldsymbol{G}(\xi)\boldsymbol{u_{si}}=0 \tag{22}$$

In the above equation,

$$\boldsymbol{H}(\xi)=\boldsymbol{B}\boldsymbol{W}(\xi)\boldsymbol{B}^T \tag{23}$$

$$\boldsymbol{G}(\xi)=\boldsymbol{B}\boldsymbol{W}(\xi) \tag{24}$$

$$\boldsymbol{a}(\xi)=\boldsymbol{H}^{-1}(\xi)\boldsymbol{G}(\xi)\boldsymbol{u_{si}} \tag{25}$$

The simulated displacement $u_s(\xi)$

$$u_s(\xi)=\boldsymbol{b}^T(\xi)\boldsymbol{H}^{-1}(\xi)\boldsymbol{G}(\xi)\boldsymbol{u_{si}} \tag{26}$$

According to the ξ value of the adjacent nodes, $\boldsymbol{u_t}$ are transformed by (26) to those independent nodes, $\boldsymbol{u_{si}}$, of the common boundary. The displacement transformation matrix is as follow:

$$\begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_t} \end{Bmatrix}=[\boldsymbol{T}^{\boldsymbol{mlsa}}]\begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_{si}} \end{Bmatrix} \tag{27}$$

To replace this displacement transformation matrix into (5), the corresponding equation can be written as:

$$U=\frac{1}{2}\{\boldsymbol{u_i}\ \boldsymbol{u_{si}}\}[\boldsymbol{K}_{\boldsymbol{t}}^{\boldsymbol{mlsa}}]\begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_{si}} \end{Bmatrix} \tag{28}$$

The stiffness matrix of the adjacent elements obtained by MLSA, $[\boldsymbol{K}_{\boldsymbol{t}}^{\boldsymbol{mlsa}}]$, can be formulated as

$$[\boldsymbol{K}_{\boldsymbol{t}}^{\boldsymbol{mlsa}}]=[\boldsymbol{T}^{\boldsymbol{mlsa}}]^T[\boldsymbol{K}][\boldsymbol{T}^{\boldsymbol{mlsa}}] \tag{29}$$

The same as in the direct transformation method, a minimum of two element transformation templates should be established.

Independent nodes u_{si} of the common boundary can be selected from a set of nodes at the left and right sides of the common boundary; but repeating nodes should be excluded, otherwise $\boldsymbol{H}^{-1}(\xi)$ cannot be obtained. Therefore, if there are repeated nodes, they should be considered as one single node. The weight function $w(\xi-\xi_i)$ of the repeated nodes can be doubled up to increase the influence of this node.

Nodes on the common boundary can be determined from adjacent elements (Fig. 9), or nodes can be newly defined without any connection to the original nodes. Those newly defined nodes can be selected uniformly (Fig. 10) or non-uniformly (Fig. 11). In order to get a nodal spacing which is gradually changed. The technique used in the finite element

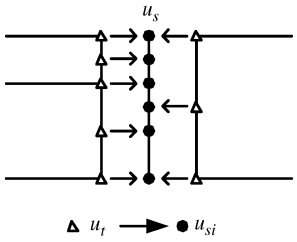

Fig. 9 Nodes obtained from the adjacent elements

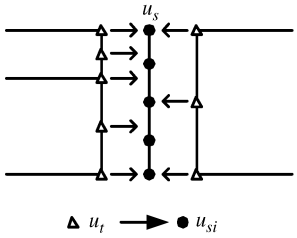

Fig. 10 Nodes newly defined by a uniform distribution

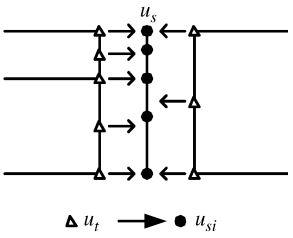

Fig. 11 Nodes newly defined by a non-uniform distribution

method is adopted here. The distribution of defined nodes can be obtained from a mapping of the master element. The density of any interval can be set by the user. The common boundary can be split into several sections. Each section is defined by a mapping from the master element. If quadratic shape functions (or higher degrees) are used to do the mapping, negative values may occur in the mapping. This will make the procedure confused. Therefore a linear transformation is needed to correct these negatives (Fig. 12). This method is relatively simple to the method mentioned by Kwak *et al.*, 2002.

Merits of This Method:

The displacement field defined by MLSA is a polynomial function with continuously changing coefficients. Therefore, the simulation efficiency cannot be judged only by the polynomial degree. A quadratic polynomial function obtained from MLSA is much more flexible than those obtained by interpolation method.

Demerits of This Method:

The displacement functions used by adjacent elements are determined by a single quadratic function. The quadratic function is determined by

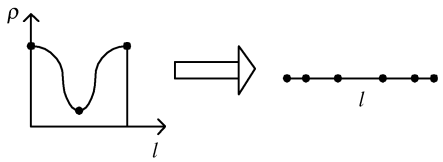

Fig. 12 Nodes defined by mapping to get a smooth distribution

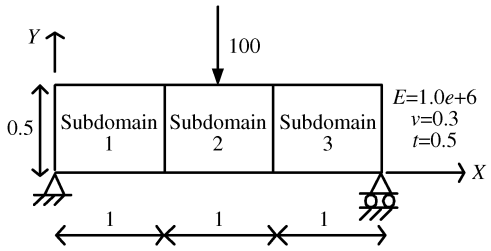

Fig. 13 Simply supported beam

three nodes, which are obtained from the common boundary displacement field. Therefore, gaps still exist between adjacent elements. The displacement field is not fully compatible. However, the convergence of this method can be guaranteed by increasing the number of nodes. To increase the number of nodes in this method is quite easy. The speed of convergence is good also.

The parameters used in MLSA are much more complicated than those used in interpolation methods. Therefore, it is not easy to arrange all these parameters to get the optimum effect for a single problem. Also, the derivation and programming are complicated. It will take time to make the MLSA method more popular.

III. COMPARISON OF NUMERICAL RESULTS

The four methods described in the last section have their merits and demerits in theory and efficiency. For practical applications, execution time and accuracy of analysis are the key points of concern. Whether conditions are fully compatible and the time used in developing the program are considered to be less important. This section will compare the results obtained from these four methods. Two example problems are used for the comparisons. Some parametric studies are executed for the MLSA method to make clear their effects.

1. Displacement Results of Simply Supported Beam and Cantilever Beam

Plane stress models are used to simulate the simply supported beam (Fig. 13) and the cantilever beam (Fig. 14). Three equally divided sub-domains are used

Table 1 Normalized maximum stress neighboring the left interface of simply supported beam with the mesh of Set 3

	The left hand side	The right hand side
(X, Y)	(0.861, 0.014)	(0.861, 0.236)
$\sigma_{max}/\sigma_{exact}$	1.127	1.135
	the left hand side	the right hand side
(X , Y)	(1.070, 0.007)	(1.070 , 0.243)
τ_{max}/τ_{exact}	0.937	0.96

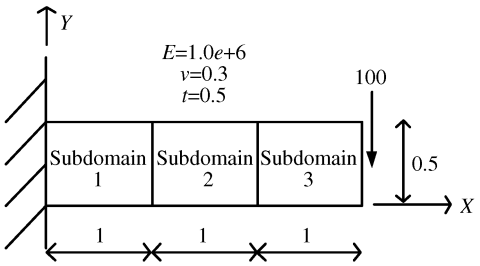

Fig. 14 Cantilever beam

in this comparison. The division of sub-domains is not really required for these two problems. They are chosen for the comparison because of simplicity and easy to get theoretical results. The number of nodes used to simulate the common boundary is five for the direct transformation method and the transformation method with modified elements. All the nodes on the adjacent elements are used for the penalty method and the MLSA method. The parameter k used for the MLSA method is one. d_{mI} is a variable which must include at least three nodal points. c is determined by the maximum distance between nodes.

For the simply supported beam, three sets of mesh are used to simulate these three sub-domains (Set 1:{2×1 , 4×2, 2×1}, Set 2:{4×2, 8×4, 4×2}, Set 3:{8×4, 16×8, 8×4}). The first number shown represents the number of elements used along the X direction, while the second one represents the number of elements along the Y direction. For the cantilever beam, three sets of mesh are (Set 1:{2×1, 4×2, 6×3}, Set 2:{4×2, 8×4, 12×6}, Set 3:{8×4, 16×8, 24×12}). The maximum displacements of each model are chosen to do the comparison. Therefore, the displacement at the middle point is chosen for the simply supported beam, and the end point is chosen for the cantilever beam. Also, the maximum normal stress, σ_{max}, and the maximum shear stress, τ_{max}, of the points for Gauss quadrature are shown in Table 1 to present the convergence of the displacement field differentials.

Comparison of results:

Figure 15 and Fig. 16 show the analysis results

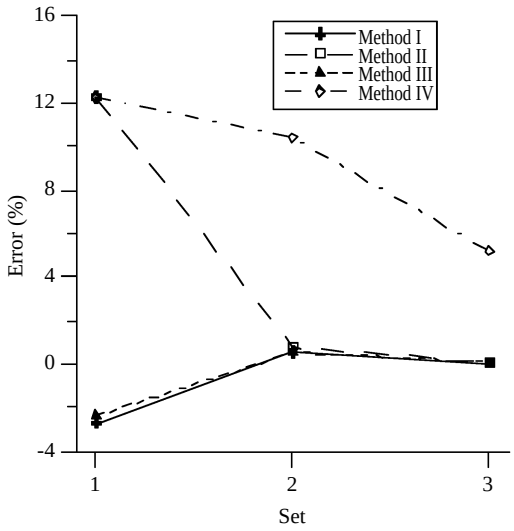

Fig. 15 Comparison of the maximum displacements of the simply supported beam

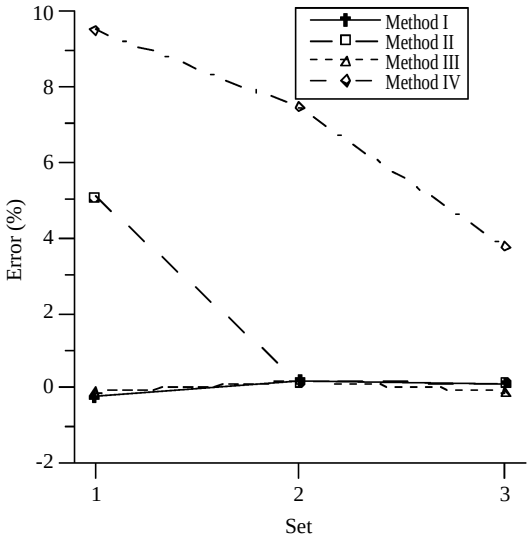

Fig. 16 Comparison of the maximum displacements of the cantilever beam

of the maximum displacements obtained from four methods. These figures demonstrate that all four methods are feasible. For the simply supported beam, the maximum errors are about 12%, which occurred in the MLSA method and the direct transformation method. The maximum error always occurred in the first set of mesh, which is the coarsest mesh. For the cantilever beam, the maximum error is around 9.5%, which occurred in the MLSA method. The results of these four methods converge to the theoretical answers when finer meshes are used.

Sequence of the speed of convergence:

1. Transformation Method with Modified Elements

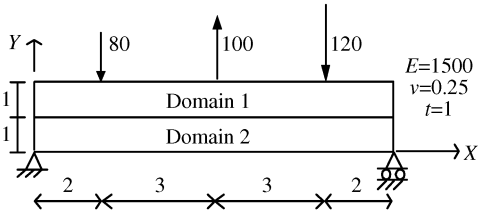

Fig. 17 Division of sub-domains to get a more complicated displacement field

2. Penalty Method
3. Direct Transformation Method
4. MLSA Method (the worst)

Sequence of execution time:

1. Penalty Method
2. MLSA Method
3. Transformation Method with Modified Elements
4. Direct Transformation Method (the shortest)

Because these two example problems are slender beams, therefore the displacement field of the common boundary is close to a linear line. A linear line can be simulated accurately by a polynomial of the first degree. Therefore, the results compared in this section cannot represent the cases with more complicated displacement fields.

2. Cases with Complicated Displacement Field on the Common Boundaries

In order to get a more complicated displacement field for comparison, the simply supported beam shown in the above section is divided horizontally into two sub-domains (Fig. 17). In this case, the theoretical results obtained from ordinary beam theory can still be used for the comparison. Also, the solution of a finite element model without the division of sub-domains is used for comparison. Three concentrated loads with converse directions are applied to the beam to increase the variety of displacement fields.

The number of nodes used for the common boundary is the same as in the previous section. All the nodes on the adjacent elements are used for MLSA method. The parameters of MLSA are the same as the previous section also.

Two sets of mesh (Set 1:{10×2, 5×1}, Set 2:{20×4, 10×2}) are used for the four methods developed in this paper to simulate this problem. A mesh of (80×16) is used for the ordinary finite element model without sub-domain division (A_exact). Discussion is focused on the displacement along the *Y* direction on the common boundary. Analysis results of these two sets of mesh are shown in Fig. 18 and Fig. 19.

Comparison of results:

Results obtained from the penalty method and the MLSA method are close to the exact solution. The results obtained from the direct transformation method and the transformation method with modified elements cannot even get the correct shape of displacement field. The results of the two transformation methods are very close (almost a coincident line in the figures). The results in this problem show that full compatibility is not a key point of solution efficiency. The comparison shows that problems with complicated displacement fields cannot be modeled well by small numbers of nodes. Therefore, some correction must be made to improve these methods. One of the possible improvements is to increase the polynomial degrees to include more nodes in its model. The other is to divide the common boundary into several sections modeled by a quadratic

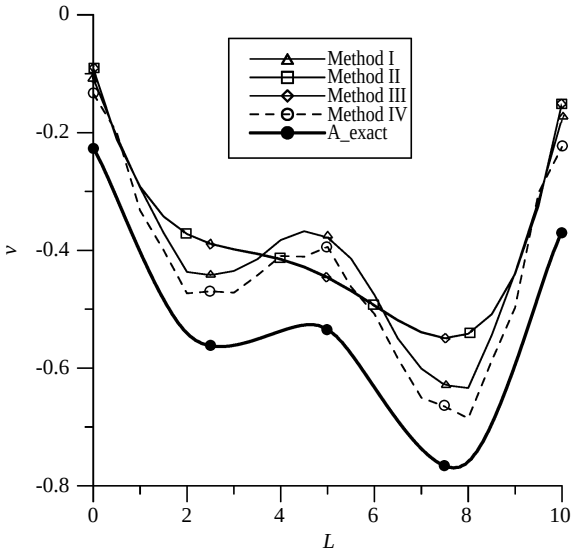

Fig. 18 *Y* displacement along the common boundary {10×2, 5×1}

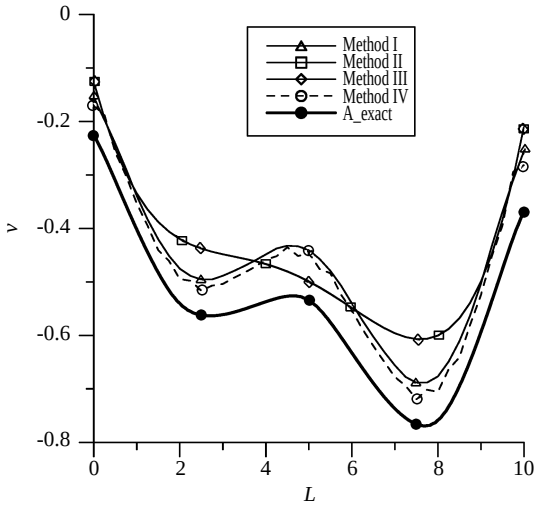

Fig. 19 *Y* displacement along the common boundary {20×4, 10×2}

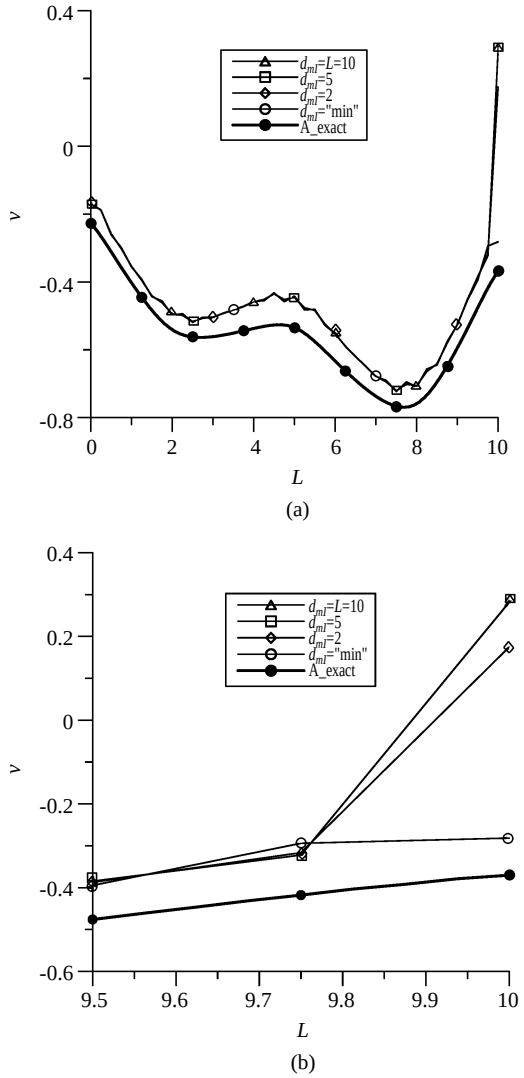

Fig. 20 (a) Parametric study of d_{mi}; (b) Parametric study of d_{mi} (L=9.5~10)

polynomial. However, the two improvements will greatly increase the complexity of the procedure.

The MLSA method shows the best solution accuracy in this problem.

In summary, a good method can be applied to all kinds of problems. Therefore, the penalty method and the MLSA method are better choices than the other two methods.

3. Parametric Study of the MLSA Method

The method used to choose nodes for the common boundary and the size of d_{mI} are two important parameters in the MLSA method. In this section, several sets of the two parameters are compared. The problem in the second section with a mesh {20×4, 10×2} is used for these comparisons.

First, parameter d_{mI} is tested. All nodes on the adjacent boundary are used to build the test model. Four sets of data are selected, which include d_{mI} =L=10, d_{mI}=5, d_{mI}=2 and d_{mI}='min' (the lowest possible d_{mI}). Analysis results are shown in Fig. 20(a). The major difference between these four sets of d_{mI} is focused on L=9.5~10 (Fig. 20(b)), in which, d_{mI}='min' is the most outstanding one, while the results of the other three are very poor between L= 9.5~10. So, it indicates that selection of a larger d_{mI} is not suitable.

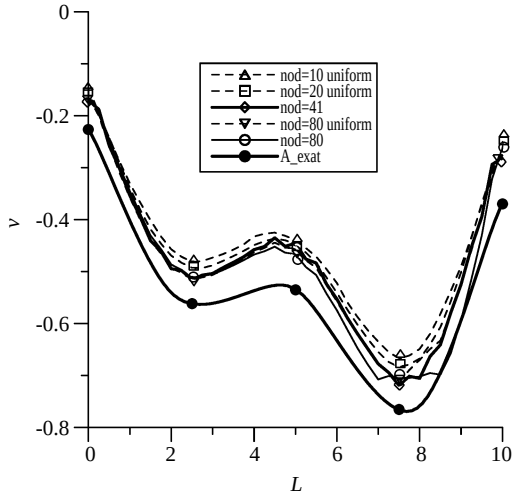

Fig. 21 Selection of nodes for the common boundary

Second, node selection for the common boundary is discussed (choose d_{mI}='min'). Five sets of node distribution include uniform distribution of 10 nodes (nod=10 uniform), uniform distribution of 20 nodes (nod=20 uniform), all nodes of the original adjacent elements (nod=41), uniform distribution of 80 nodes (nod=80 uniform), and 80 nodes distributed with varied density (nod=80). Better results can be obtained by the increase of the number of selected nodes (Fig. 21). However, the best choice is to adopt all nodes of the original adjacent elements.

IV. CONCLUSIONS

The mapping method with graded mesh is suitable to most problems with its changeable conditions. To simplify the procedure used in automatic mesh is also an important concern of program developers. Four methods are developed to solve the compatibility problem between sub-domains. They are the penalty method, the direct transformation method, the transformation method with modified elements, and the MLSA method. All four methods are proved to be feasible for this problem. Comments about these four methods are as follows:

1. The penalty method is the most straightforward, but the time taken for the analysis is the longest.

2. The two transformation Methods are very suitable for problems with simple displacement fields. However, when a complicated displacement is encountered, the results are not acceptable.
3. MLSA has a balanced performance for most problems. The adequate use of all parameters is important for this method.

REFERENCES

Aminpour, M. A., Ransom, J. B., and McCleary, S. L., 1995, "Coupled Analysis Method for Structures with Independently Modeled Finite Element Subdomains," *International Journal for Numerical Methods in Engineering*, Vol. 38, No. 21, pp. 3695-3718.

Arora, J. S., Chahande, A. I., and Paeng, J. K., 1991, "Multiplier Methods for Engineering Optimization," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 7, pp. 1485-1525.

Belytschko, T., Lu, Y. Y., and Gu, L., 1994, "Element-free Galerkin Methods," *International Journal for Numerical Methods in Engineering*, Vol. 37, No. 2, pp. 229-256.

Blacker, T. D., and Stephenson, M. B., 1991, "Paving: a New Approach to Automated Quadrilateral Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 4, pp. 811-847.

Brown, P. R., 1981, "A Non-Interactive Method for Automatic Generation of Finite Element Meshes Using the Schwarz-Christoffel Transformation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 25, No. 1, pp. 101-126.

Cavendish, J. C., 1974, "Automatic Triangulation of Arbitrary Planar Domains for the Finite Element Method," *International Journal for Numerical Methods in Engineering*, Vol. 8, No. 4, pp. 679-697.

Cheng, G., and Li, H., 1996, "New Method for Graded Mesh Generation of Quadrilateral Finite Elements," *Computers and Structures*, Vol. 59, No. 5, pp. 823-829.

Chinnaswamy, C., Amadei, B., and Illangasekare, T. H., 1991, "New Method for Finite Element Transitional Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 31, No. 7, pp. 1253-1270.

Cohen, H. D., 1980, "Method for the Automatic Generation of Triangular Elements on a Surface," *International Journal for Numerical Methods in Engineering*, Vol. 15, No. 3, pp. 470-476.

Dohrmann, C. R., Key, S. W., and Heinstein, M. W., 2000, "Methods for Connecting Dissimilar Three-dimensional Finite Element Meshes," *International Journal for Numerical Methods in Engineering*, Vol. 47, No. 5, pp. 1057-1080.

Ginman, K. M. S., 1997, "Topology Optimization for 2-D Continuum Using Element Free Galerkin Method," *Thesis of The University of Texas at Arlington*.

Gordon, W. J., and Hall, C. A., 1973, "Construction of Curvilinear Co-ordinate Systems and Applications to Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 7, No. 4, pp. 461-477.

Gordon, W. J., and Wixson, J. A., 1978, "Shapard's Method of Metric Interpolation to Bivariate and Multivariate Data," *Mathematics of Computation*, Vol. 32, No. 141, pp. 253-264.

Haber, R., Shephard, M. S., Abel, J. F., Gallagher, R. H., and Greenberg, D. P., 1981, "General Two-Dimensional Graphical Finite Element Preprocessor Utilizing Discrete Transfinite Mapping," *International Journal for Numerical Methods in Engineering*, Vol. 17, No. 7, pp. 1015-1044.

Ho-Le, K., 1988, "Finite Element Mesh Generation Methods: a Review and Classification," *Computer Aided Design*, Vol. 20, No. 1, pp. 27-38.

Houlsby, G. T., Liu, G., and Augarde, C. E., 2000, "A Tying Scheme for Imposing Displacement Constraints in Finite Element Analysis," *Communications in Numerical Methods in Engineering*, Vol. 16, No. 10, pp. 721-732.

Kikuchi, N., 1985, "Adaptive Grid-design Methods for Finite Element Analysis," *Computer Methods in Applied Mechanics and Engineering*, Vol. 55, Nos. 1-2, pp. 129-160.

Kwak, D. Y., Cheon, J. S., and Im, Y. T., 2002, "Remeshing for Metal Forming Simulations-Part I : Two-dimensional Quadrilateral Remeshing," *International Journal for Numerical Methods in Engineering*, Vol. 53, No. 11, pp. 2463-2500.

Lancaster, P., and Salkauskas, K., 1981, "Surfaces Generated by Moving Least Square Method," *Mathematics of Computation*, Vol. 37, No. 155, pp. 141-158.

Lee, C. K., and Lo, S. H., 1994, "A New Scheme for the Generation of a Graded Quadrilateral Mesh," *Computers and Structures*, Vol. 52, No. 5, pp. 847-857.

Nayroles, B., Touzot, G., and Villon, P., 1992, "Generalizing the Finite Element Method : Diffuse Approximation and Diffuse Element," *Computational Mechanics*, Vol. 10, No. 5, pp. 307-318.

Rank, E., Schweingruber, M., and Sommer, M., 1993, "Adaptive Mesh Generation and Transformation of Triangular to Quadrilateral Meshes," *Communications in Numerical Methods in Engineering*, Vol. 9, No. 2, pp. 121-129.

Sarrate, J., and Huerta, A., 2000, "Efficient Unstructured Quadrilateral Mesh Generation," *International Journal for Numerical Methods in Engineering*,

Vol. 49, No. 10, pp. 1327-1350.

Talbert, J. A., and Parkinson, A. R., 1990, "Development of an Automatic Two-dimensional Finite Element Mesh Generator Using Quadrilateral Element and Bezier Curve Boundary Definition, " *International Journal for Numerical Methods in Engineering*, Vol. 29, No. 7, pp. 1551-1567.

Wordenweber, B., 1984, "Finite Element Mesh Generation," *Computer Aided Design*, Vol. 16, No. 5, pp. 285-291.

Zienkiewicz, O. C., and Phillips, D. V., 1971, "Automatic Mesh Generation Scheme for Plane and Curved Surfaces by Isoparametric Co-ordinates, " *International Journal for Numerical Methods in Engineering*, Vol. 3, No. 4, pp. 519-528.

Zhu, J. Z., Zienkiewicz, O. C., Hinton, E., and Wu, J., 1991, "New Approach to the Development of Automatic Quadrilateral Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 4, pp. 849-866.

Manuscript Received: Jul. 03, 2003
Revision Received: Apr. 15, 2004
and Accepted: Apr. 22, 2004