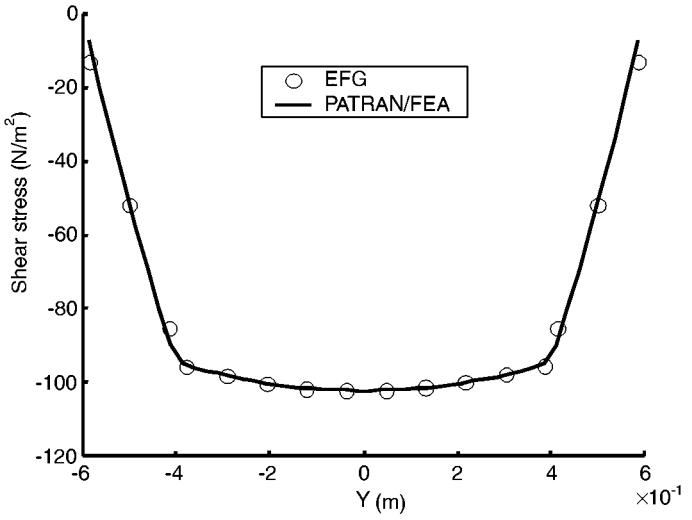


FIGURE 6.23
Numerical solutions for the shear stress at the section $x = L/2$ of the sandwich composite beam.

6.2.5 Remarks

In this section, the penalty method is used to impose the essential boundary and continuity conditions in the EFG method that uses MLS shape functions. It overcomes the drawbacks induced by the use of the method of Lagrange multipliers. The main advantage of the use of the penalty method is that it leads to a positive definite and banded stiffness matrix. The stiffness matrix also has a smaller dimension than those using Lagrangian multipliers, which improves computational efficiency. Numerical examples have demonstrated the performance of the penalty method.

The penalty method was also applied for the treatment of problems with material discontinuity. System equations for multibody problems are derived, and implemented in stress analysis in a composite beam. The numerical results agree well with the analytical and FEM solutions. This demonstrates the potential for application of the penalty method in MFree methods for analyzing structures of composite materials.

The methods for determining the penalty factor were given in Section 4.3.3. More discussion on this is provided in Example 6.12 in Section 6.4.5, when we deal with nonlinear problems.

6.3 Constrained Moving Least Square Method for EFG

As discussed, the root of the difficulty in imposing essential boundary conditions is the use of MLS approximation, which produces shape functions that do not satisfy the Kronecker delta function property. In areas that are far from the essential boundaries, the MLS shape function works just fine. The problems arise only for the nodes on the essential boundaries. If a procedure could be developed that produces MLS shape functions that possess the Kronecker delta function property only for the nodes on the essential boundaries, the

problems would be overcome. The following presents an approach that uses constraints in the process of MLS approximation. The approach was originally proposed by G. R. Liu and Yang in 1999 and is termed the constrained moving least squares (CMLS) method. CMLS was first detailed in Yang's master's.

The basic idea of the CMLS is to impose the constraints of essential boundary conditions in the stage of shape function construction, so that the unconstrained Galerkin weak form can be used to produce a well-behaved equation system. The CMLS enforces the MLS approximation to pass through some desired values of the field variables at nodes at which the essential boundary condition is given. Thus, the approximation functions so obtained have the property of the delta function at the nodes only on the essential boundaries, and the treatment of the boundary conditions at the nodes can be as simple as in FEM. As shown later in this section, the system matrix established is banded, positive definite, and is of the same dimension as the equations produced by FEM.

6.3.1 Formulation

Let $u(\mathbf{x})$ be a function defined in the domain Ω and its approximation be $u^h(\mathbf{x})$. We are to approximate the function at a point of interest $\mathbf{x}$ using n nodes in the support domain of $\mathbf{x}$. The coordinates of the n nodes are defined by $\mathbf{x}_1,\ldots,\mathbf{x}_n$, where $\mathbf{x}_I = (x_I, y_I)$ in two dimensions. The nodal values of the function are denoted as

$$\mathbf{U}_s = \{u_1\ u_2\ \ldots\ u_n\}^T \tag{6.77}$$

The function at point $\mathbf{x}$ is approximated using m terms of monomials, i.e.,

$$u^h(\mathbf{x}) = \sum_{j}^{m} p_j(\mathbf{x}) a_j(\mathbf{x}) \equiv \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{6.78}$$

where $\mathbf{p}(\mathbf{x})$ is a vector of monomial basis and $\mathbf{a}(\mathbf{x})$ is a vector of coefficients, which are the same as those in Equation 5.41.

Assume the approximation function $u^h(\mathbf{x})$ is required to be equal to the nodal values at $k(k \le n, k \le m)$ constrained nodes, which is written as a vector:

$$\mathbf{u}_b = \{\bar{u}_1\ \bar{u}_2\ \ldots\ \bar{u}_k\}^T \tag{6.79}$$

The nodal values of u_I at the remaining unconstrained nodes are written as

$$\mathbf{u}_m = \{u_{k+1}\ u_{k+2}\ \ldots\ u_n\}^T \tag{6.80}$$

Write Equation 6.78 in the form of subvectors as follows:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a} = \{\mathbf{p}_b^T(\mathbf{x})\ \mathbf{p}_m^T(\mathbf{x})\}\begin{Bmatrix}\mathbf{a}_b\\ \mathbf{a}_m\end{Bmatrix} \tag{6.81}$$

where

$$\mathbf{p}_b^T(\mathbf{x}) = \{p_1(\mathbf{x})\ p_2(\mathbf{x})\ \ldots\ p_k(\mathbf{x})\} \tag{6.82}$$

$$\mathbf{p}_m^T(\mathbf{x}) = \{p_{k+1}(\mathbf{x})\ p_{k+2}(\mathbf{x})\ \ldots\ p_m(\mathbf{x})\} \tag{6.83}$$

and

$$\mathbf{a}_b = \{a_1 \;\; a_2 \;\; \ldots \;\; a_k\}^T \tag{6.84}$$

$$\mathbf{a}_m = \{a_{k+1} \;\; a_{k+2} \;\; \ldots \;\; a_m\}^T \tag{6.85}$$

We can then express k coefficients of $\mathbf{a}$, say, $\mathbf{a}_b = [a_1 \; a_2 \; \ldots \; a_k]^T$, in terms of the rest of the coefficients of $\mathbf{a}$, that is, $\mathbf{a}_m = [a_{k+1} \; \ldots \; a_m]^T$, from the constraint equations at k constrained nodes at the essential boundary. These constraint equations can be expressed as

$$\mathbf{u}_b = \{u^h(\mathbf{x}_1) \;\; u^h(\mathbf{x}_2) \;\; \ldots \;\; u^h(\mathbf{x}_k)\}^T \tag{6.86}$$

Substitute Equation 6.78 into Equation 6.86, and form subvectors as follows:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) = \{\mathbf{p}_b^T \;\; \mathbf{p}_m^T\}\begin{Bmatrix}\mathbf{a}_b\\ \mathbf{a}_m\end{Bmatrix} \tag{6.87}$$

we obtain

$$\mathbf{u}_b = \mathbf{P}_b\mathbf{a}_b + \mathbf{P}_m\mathbf{a}_m \tag{6.88}$$

where the moment matrix $\mathbf{P}_b$ is given by

$$\mathbf{P}_b = \begin{bmatrix} p_1(\bar{\mathbf{x}}_1) & p_2(\bar{\mathbf{x}}_1) & \cdots & p_k(\bar{\mathbf{x}}_1)\\ p_1(\bar{\mathbf{x}}_2) & p_2(\bar{\mathbf{x}}_2) & \cdots & p_k(\bar{\mathbf{x}}_2)\\ \vdots & \vdots & \ddots & \vdots\\ p_1(\bar{\mathbf{x}}_k) & p_2(\bar{\mathbf{x}}_k) & \cdots & p_k(\bar{\mathbf{x}}_k)\end{bmatrix} = \begin{bmatrix}\mathbf{p}_b^T(\bar{\mathbf{x}}_1)\\ \mathbf{p}_b^T(\bar{\mathbf{x}}_2)\\ \vdots\\ \mathbf{p}_b^T(\bar{\mathbf{x}}_k)\end{bmatrix} \tag{6.89}$$

$$\mathbf{P}_m = \begin{bmatrix} p_{k+1}(\bar{\mathbf{x}}_1) & \cdots & p_m(\bar{\mathbf{x}}_1)\\ p_{k+1}(\bar{\mathbf{x}}_2) & \cdots & p_m(\bar{\mathbf{x}}_2)\\ \vdots & \ddots & \vdots\\ p_{k+1}(\bar{\mathbf{x}}_k) & \cdots & p_m(\bar{\mathbf{x}}_k)\end{bmatrix} = \begin{bmatrix}\mathbf{p}_m^T(\bar{\mathbf{x}}_1)\\ \mathbf{p}_m^T(\bar{\mathbf{x}}_2)\\ \vdots\\ \mathbf{p}_m^T(\bar{\mathbf{x}}_k)\end{bmatrix} \tag{6.90}$$

We can then express k coefficients of $\mathbf{a}_b$ in terms of the rest of the coefficients of $\mathbf{a}_m$ using Equation 6.88, i.e.,

$$\mathbf{a}_b = \mathbf{C}_1 \cdot \mathbf{u}_b - \mathbf{C}_2 \cdot \mathbf{a}_m \tag{6.91}$$

where

$$\mathbf{C}_1 = \mathbf{P}_b^{-1} \tag{6.92}$$

$$\mathbf{C}_2 = \mathbf{P}_b^{-1} \cdot \mathbf{P}_m \tag{6.93}$$

The vector of coefficients, $\mathbf{a}_m$, is determined by the conventional MLS method over the "free" (unconstrained) nodes. At each free point $\mathbf{x}$ corresponding to $\mathbf{u}_m$, $\mathbf{a}_m$ are chosen to minimize the weighted residual (note that $\mathbf{a}_b$ is determined by Equation 6.91 and should

be considered a constant in the weighted residual):

$$J = \sum_{I=k+1}^{n} \widehat{W}(\mathbf{x} - \mathbf{x}_I)[\mathbf{p}^T(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - u_I]^2 \tag{6.94}$$

where $n - k$ is the number of free nodes, $w(\mathbf{x} - \mathbf{x}_I)$ is a weight function, and u_I is the nodal value at node I ($I = k + 1, k + 2,\ldots,n$). The minimization of J with respect to coefficients $\mathbf{a}_m$ results in the following linear system:

$$\mathbf{A}_1(\mathbf{x})\cdot\mathbf{a}_b + \mathbf{A}_2(\mathbf{x})\cdot\mathbf{a}_m = \mathbf{B}(\mathbf{x})\mathbf{u}_m \tag{6.95}$$

where

$$\mathbf{A}_1(\mathbf{x}) = \sum_{I=k+1}^{n} \widehat{W}_I(\mathbf{x})\mathbf{p}_m(\mathbf{x}_\mathbf{I})\mathbf{p}_b^T(\mathbf{x}_\mathbf{I}) \tag{6.96}$$

$$\mathbf{A}_2(\mathbf{x}) = \sum_{I=k+1}^{n} \widehat{W}_I(\mathbf{x})\mathbf{p}_m(\mathbf{x}_I)\mathbf{p}_m^T(\mathbf{x}_I) \tag{6.97}$$

$$\mathbf{B}(\mathbf{x}) = [\widehat{W}_1(\mathbf{x})\mathbf{p}_m(\mathbf{x}_1), \widehat{W}_2(\mathbf{x})\mathbf{p}_m(\mathbf{x}_2),\ldots,\widehat{W}_n(\mathbf{x})\mathbf{p}_m(\mathbf{x}_{n-k})] \tag{6.98}$$

and

$$\widehat{W}_I(\mathbf{x}) \equiv \widehat{W}(\mathbf{x} - \mathbf{x}_I) \tag{6.99}$$

From Equations 6.91 and 6.95, we can obtain $\mathbf{a}_b$ and $\mathbf{a}_m$

$$\mathbf{a}_b = \mathbf{E}_1\mathbf{u}_b + \mathbf{E}_2\mathbf{u}_m \tag{6.100}$$

and

$$\mathbf{a}_m = \mathbf{D}_1\mathbf{u}_b + \mathbf{D}_2\mathbf{u}_m \tag{6.101}$$

where

$$\mathbf{D}_1 = (\mathbf{A}_1\cdot\mathbf{C}_2 - \mathbf{A}_2)^{-1}\cdot\mathbf{A}_1\cdot\mathbf{C}_1 \tag{6.102}$$

$$\mathbf{D}_2 = -(\mathbf{A}_1\cdot\mathbf{C}_2 - \mathbf{A}_2)^{-1}\cdot\mathbf{B} \tag{6.103}$$

$$\mathbf{E}_1 = \mathbf{C}_1 - \mathbf{C}_2\cdot\mathbf{D}_1 \tag{6.104}$$

$$\mathbf{E}_2 = -\mathbf{C}_2\cdot\mathbf{D}_1 \tag{6.105}$$

The coefficient vector $\mathbf{a}$ can be expressed as

$$\mathbf{a} = \begin{Bmatrix}\mathbf{a}_b\\ \mathbf{a}_m\end{Bmatrix} = \begin{bmatrix}\mathbf{E}_1\\ \mathbf{D}_1\end{bmatrix}\mathbf{u}_b + \begin{bmatrix}\mathbf{E}_2\\ \mathbf{D}_2\end{bmatrix}\mathbf{u}_m = \begin{bmatrix}\mathbf{E}_1 & \mathbf{E}_2\\ \mathbf{D}_1 & \mathbf{D}_2\end{bmatrix}\begin{Bmatrix}\mathbf{u}_b\\ \mathbf{u}_m\end{Bmatrix} \tag{6.106}$$

Hence, we have

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) = [\mathbf{p}_b^T \;\; \mathbf{p}_m^T]\begin{Bmatrix}\mathbf{a}_b\\ \mathbf{a}_m\end{Bmatrix} = [\mathbf{p}_b^T \;\; \mathbf{p}_m^T]\begin{bmatrix}\mathbf{E}_1 & \mathbf{E}_2\\ \mathbf{D}_1 & \mathbf{D}_2\end{bmatrix}\begin{Bmatrix}\mathbf{u}_b\\ \mathbf{u}_m\end{Bmatrix} \tag{6.107}$$

or

$$u^h(\mathbf{x}) = [\boldsymbol{\phi}_b \quad \boldsymbol{\phi}_m]\begin{Bmatrix}\mathbf{u}_b\\ \mathbf{u}_m\end{Bmatrix} = \boldsymbol{\Phi}\mathbf{U}_s \tag{6.108}$$

where

$$\boldsymbol{\phi}_b = \mathbf{p}_b^T\mathbf{E}_1 + \mathbf{p}_m^T\mathbf{D}_1 \tag{6.109}$$

and

$$\boldsymbol{\phi}_m = \mathbf{p}_b^T\mathbf{E}_2 + \mathbf{p}_m^T\mathbf{D}_2 \tag{6.110}$$

To determine the derivatives from the displacement Equation 6.108, it is necessary to obtain the shape function derivatives. The spatial derivatives of the shape functions are obtained by

$$\boldsymbol{\phi}_{,x} = [\boldsymbol{\phi}_{b,x} \quad \boldsymbol{\phi}_{m,x}] \tag{6.111}$$

where

$$\boldsymbol{\phi}_{b,x} = \mathbf{p}_{b,x}^T\mathbf{E}_1 + \mathbf{p}_b^T\mathbf{E}_{1,x} + \mathbf{p}_{m,x}^T\mathbf{D}_1 + \mathbf{p}_m^T\mathbf{D}_{1,x} \tag{6.112}$$

and

$$\boldsymbol{\phi}_{m,x} = \mathbf{p}_{b,x}^T\mathbf{E}_2 + \mathbf{p}_b^T\mathbf{E}_{2,x} + \mathbf{p}_{m,x}^T\mathbf{D}_2 + \mathbf{p}_m^T\mathbf{D}_{2,x} \tag{6.113}$$

6.3.2 Constrained Surfaces Generated by CMLS

To verify CMLS, two examples are presented, the first to show how the CMLS works in surface fitting with or without constraints on the boundary using the formulation developed in the previous subsection.

Example 6.7 Linear Constraint

The first example examines a flat surface fitted before and after a linear constraint is imposed, and the results are shown in Figures 6.24 and 6.25. The surface is parallel to the x–y plane before the application of a linear constraint to the nodes on the boundary (see Figure 6.24). After the imposition of the constraint, the flat surface follows these values at the boundary, as shown in Figure 6.25. This demonstrates that CMLS works well in imposing linear constraints on nodes.

Example 6.8 Parabolic Constraint

The second example examines the same situation, but a parabolic constraint is considered. The results are shown in Figures 6.26 and 6.27. These two examples demonstrate that the surfaces are enforced to pass through the constraint points, while the remaining part of the surface in the position far from the constraint points still possesses the property of a conventional MLS surface. This further confirms that the use of the CMLS algorithm works well in the imposition of constraints on boundary nodes.

It should be noted that the requirement for the order of the monomials in the basis used in CMLS is higher than that of MLS because some of the coefficients are determined by

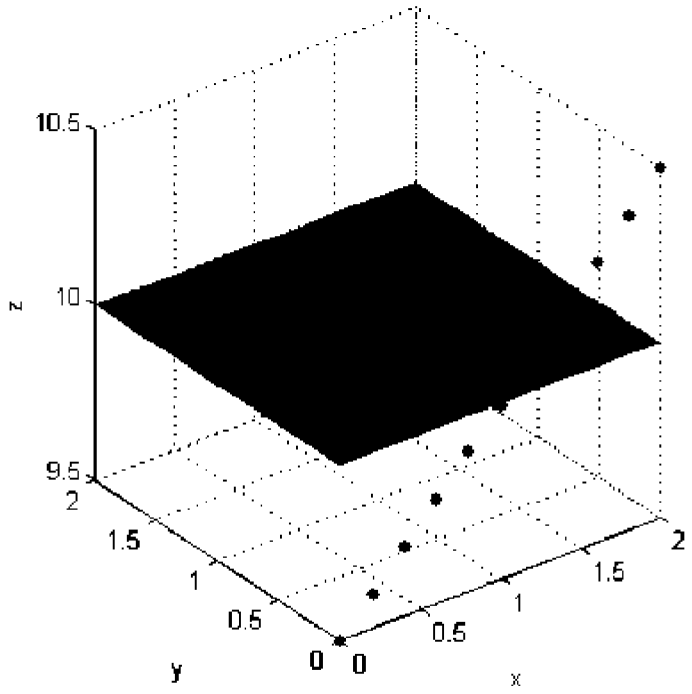


FIGURE 6.24
2D CMLS approximation function before the linear constraint is imposed.

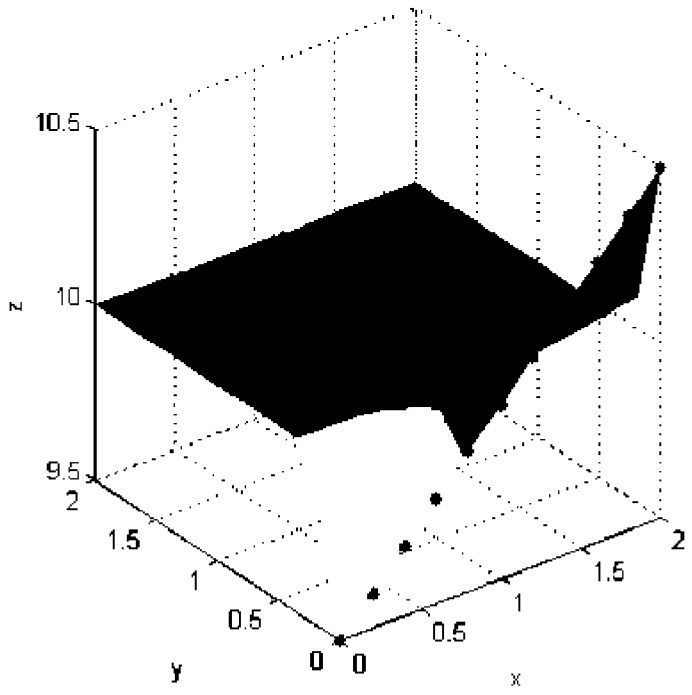


FIGURE 6.25
2D CMLS approximation function after the linear constraint is imposed.

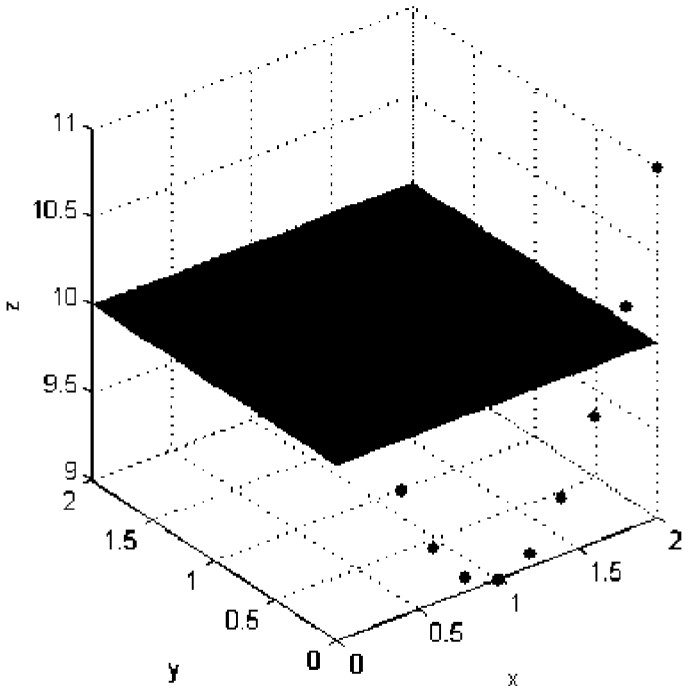


FIGURE 6.26
2D CMLS approximation function before the parabolic constraint is applied.

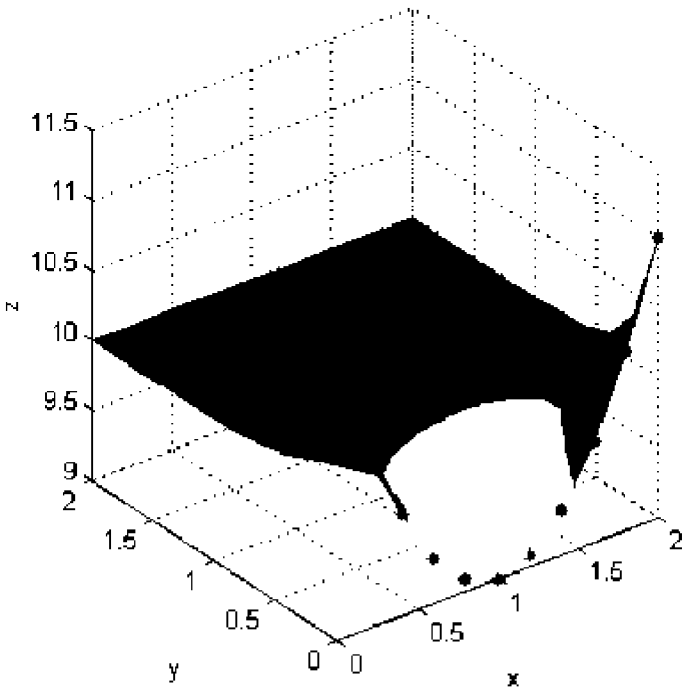


FIGURE 6.27
2D CMLS approximation function after the parabolic constraint is applied.

nodal constraints. In general, the greater the number of constrained points included in the support domain, the greater the number of terms of basis monomials of higher orders that should be used. The number of the polynomial coefficients determined by the nodal enforcement and the number by MLS should be balanced. For example, for a polynomial with six coefficients, it is best that three of the coefficients be determined by nodal enforcement and the rest by MLS. From our experience with 2D problems, the appropriate basis function should be polynomials of order 2 or 3. Including more free nodes can help to reduce the chance of producing singular moment matrices in the process of computing CMLS shape functions.

6.3.3 Weak Form and Discrete Equations

Consider again the mechanics problem stated in Equations 6.1 and 6.2, which was also dealt with in Section 6.2.1. Instead of using the penalty method, we now use CMLS to generate the shape functions that possess the Kronecker delta function property at the nodes on the essential boundaries. The prescribed displacement on the essential boundary can then be imposed directly, as in conventional FEM.

Using CMLS, the Galerkin weak form is as simple as in FEM, because the shape function created by CMLS has the same property as the shape functions of FEM at the essential boundary points. Based on Equation 6.67, we write

$$\int_{\Omega} \delta(\mathbf{Lu})^T(\mathbf{cLu})d\Omega - \int_{\Omega} \delta\mathbf{u}^T\mathbf{b}d\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T\bar{\mathbf{t}}\,d\Gamma = 0 \tag{6.114}$$

Note that the integration on the essential boundary has been removed, because the displacement function $\mathbf{u}$ is to be approximated using the CMLS approximation shown in Equation 6.108. Substituting the approximation of $\mathbf{u}$ into the weak form Equation 6.114 yields the discrete system equations:

$$\mathbf{KU} = \mathbf{F} \tag{6.115}$$

where $\mathbf{U}$ is a vector of nodal parameters of displacements for all the nodes in the problem domain, and $\mathbf{K}$ is the stiffness matrix assembled using the following nodal matrix:

$$\mathbf{K}_{ij} = \int_{\Omega} \mathbf{B}_i^T\mathbf{c}\mathbf{B}_j d\Omega \tag{6.116}$$

where $\mathbf{B}_i$ and $\mathbf{c}$ have the same form as those in Equation 6.38, except that the shape functions used are different. The force vector $\mathbf{F}$ is assembled using the nodal force vector defined in Equation 6.31 but computed using the shape functions defined in this section.

The handling of essential boundary conditions at the boundary nodes is the same as in FEM. All one need do is impose the boundary condition directly to the nodal displacement in the final system equation. Using CMLS is very computationally efficient, especially for large systems, as the banded feature as well as the symmetry of the stiffness matrix is preserved. The drawback of this method is that the possibility of having a singular moment matrix is increased because fewer free nodes are used. Another drawback is that it is difficult to ensure the compatibility in the field function approximation, which leads to difficulty in passing the patch tests as discussed in the next section. Note that CMLS is used only for support domains that have at least one essential boundary point.

The Gauss quadrature scheme is still required to perform the integrations in computing system matrices. Ensuring accurate numerical integration when using CMLS is also more difficult compared with the use of MLS because the order of the CMLS shape functions created is usually higher.

6.3.4 Examples for Mechanics Problems

Example 6.9 Patch Test

The first numerical example of solid mechanics problem is the standard patch test. As square patch of $L_x = 2$ and $L_y = 2$ shown in Figure 6.28 is considered. The displacements are prescribed on all outside boundaries of the patch by a linear function defined by Equation 6.25. The nodal arrangement in this patch is also shown in Figure 6.28. A discrete system equation in the form of Equation 6.115 is established using the CMLS shape functions. Gauss quadrature is used to perform the integration. Table 6.7 shows the numerical results obtained using a background mesh of 40×40 quadrature cells with a 6×6 Gauss point each. The maximum errors of u_x and u_y are of order 10^{-7} and 10^{-6}, respectively.

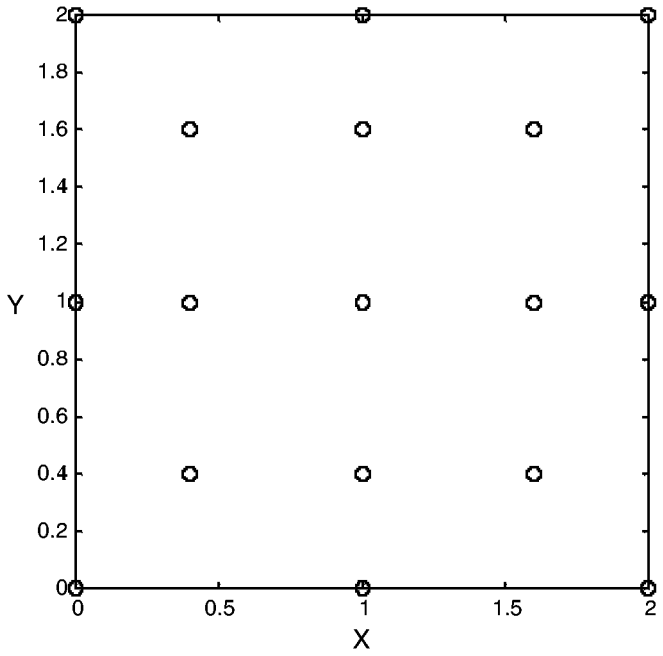


FIGURE 6.28
The nodal arrangement for the patch test.

TABLE 6.7

Numerical Results of the Displacements at Interior Nodes for the Patch Test

Node and Coordinates	u_x	u_y
9, (0.4,0.4)	0.40000024144004	0.40000024434450
10, (1,0.4)	1.00000008122310	0.39999996975363
11, (1.6,0.4)	1.59999956772840	0.39999902865713
12, (0.4,1)	0.40000058611168	1.00000023797684
13, (1,1)	1.00000009865193	0.99999996948019
14, (1.6,1)	1.59999981657464	0.99999978568420
15, (0.4,1.6)	0.39999990454283	1.60000015089939
16, (1,1.6)	0.99999979845197	1.59999992408260
17, (1.6,1.6)	1.59999922275895	1.59999878285197

TABLE 6.8

Maximum Error of the Displacements for a Patch Test for EFG-CMLS

Integration Cells (quadrature points)	Maximum Error of u_x	Maximum Error of u_y
10×10 (4×4)	2.6×10^{-3}	1.8×10^{-3}
20×20 (4×4)	1.4×10^{-4}	5.2×10^{-4}
40×40 (4×4)	1.5×10^{-5}	2.1×10^{-5}
40×40 (6×6)	8×10^{-7}	1.6×10^{-6}
8×8 (16×16)	5.2×10^{-5}	4×10^{-5}

Hence, the patch test is passed to the machine accuracy in this case. Selection of nodes often affects the results of the patch test. We tested a number of other patches of irregular internal nodes, and found that some cases had difficulty passing the patch test. We believe that the reason lies mainly in the compatibility of the field function approximation using CMLS. The nature of this problem is somewhat similar to that using PIM shape functions, which are discussed in detail in Chapter 8.

We found that the accuracy of the numerical results depends heavily on the accuracy of the numerical integration, as shown in Table 6.8. This is because the order of the shape functions created by CMLS is usually higher than those of MLS. Note that the Gauss quadrature is designed for integrating polynomial functions. The quadrature error may increase when it is applied to complex integrands like that given in Equation 6.116 because integrands are fractional functions, which in general cannot be represented exactly by polynomial functions. The Gauss quadrature, therefore, will not be able to produce exact results for the integration. An efficient and accurate numerical integration scheme should be used in EFG-CMLS to pass the patch test.

Example 6.10 Cantilever Beam

For benchmarking purposes, we consider again a beam of characteristic length L and height D subjected to a parabolic traction at the free end, as shown in Figure 6.4. The beam is considered to be of unit thickness, and the plane stress problem is considered. The exact solution is given by Equations 6.51 to 6.56 for displacements and stresses. The parameters used in this section are the same as in Example 6.4.

The arrangement of nodes and quadrature cells is shown in Figure 6.29. In each quadrature cell, 4×4 Gauss points are used. The solutions are obtained using a quadratic basis function with cubic spline weight function, and support domains of $\alpha_s = 2.5$ are used.

Figure 6.30 plots the analytical solution and the numerical solution using the present method for the beam deflection along the x axis. The plot shows excellent agreement between the analytical and present numerical results using CMLS.

Figure 6.31 illustrates the distribution of the normal stresses σ_x on the cross section at $x = L/2$ of the beam. Both the analytical solution and the present EFG-CMLS solution are plotted together for comparison. Very good agreement is observed between the stresses calculated by the analytical formulation and the present EFG-CMLS method. Figures 6.32 and 6.33 show the same comparison for, respectively, the normal stress σ_y and the shear stress τ_{xy} at the section of $x = L/2$ of the beam. Again, very good agreement is observed between the results calculated by the analytical formulation and the present EFG-CMLS method.

Table 6.9 compares the numerical and analytical results for the vertical displacement at point A on the beam (see Figure 6.4). The calculation was performed for models discretized with 18, 24, 55, and 189 nodes. This table shows that the numerical result converges as the number of nodes increases.

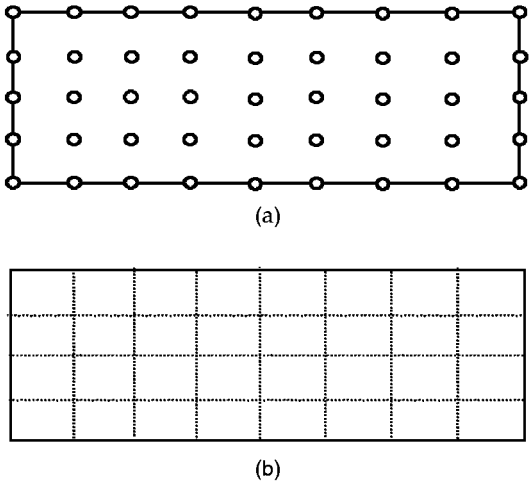


FIGURE 6.29
(a) Nodal arrangement; (b) mesh used for integration.

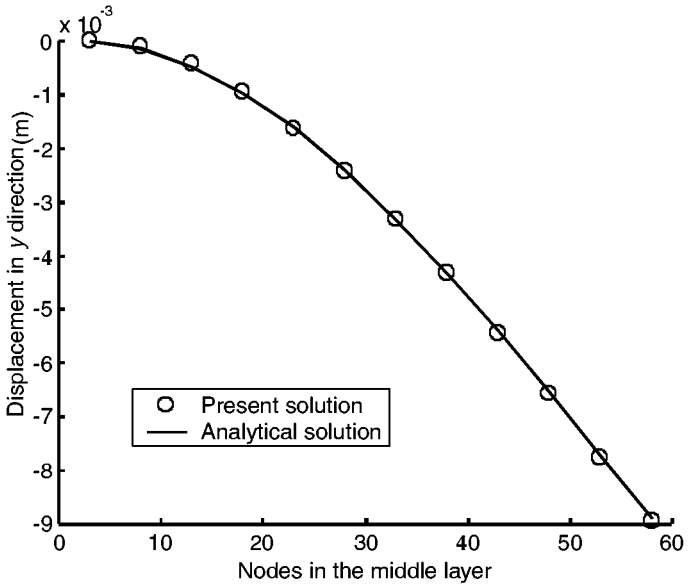


FIGURE 6.30
Analytical and EFG-CMLS numerical solutions for the deflection of the cantilever beam.

Example 6.11 Hole in an Infinite Plate

A plate with a circular hole subjected to a unidirectional tensile load in the x direction is considered, as shown in Figure 6.34. The plane stress condition is assumed. Due to symmetry, only the upper right quarter of the plate is modeled, as shown in Figure 6.35. Corresponding symmetric boundary conditions are applied on $x = 0$ and $y = 0$, i.e.,

$$u_x = 0, \qquad \sigma_{xy} = 0 \quad \text{when} \quad x = 0 \tag{6.117}$$

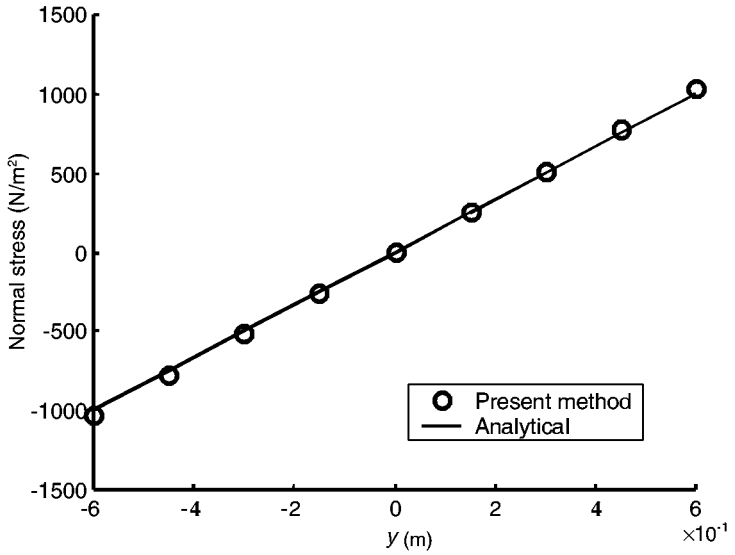


FIGURE 6.31
Analytical and the present EFG-CMLS numerical solutions for σ_x at the section of $x = L/2$ of the cantilever beam.

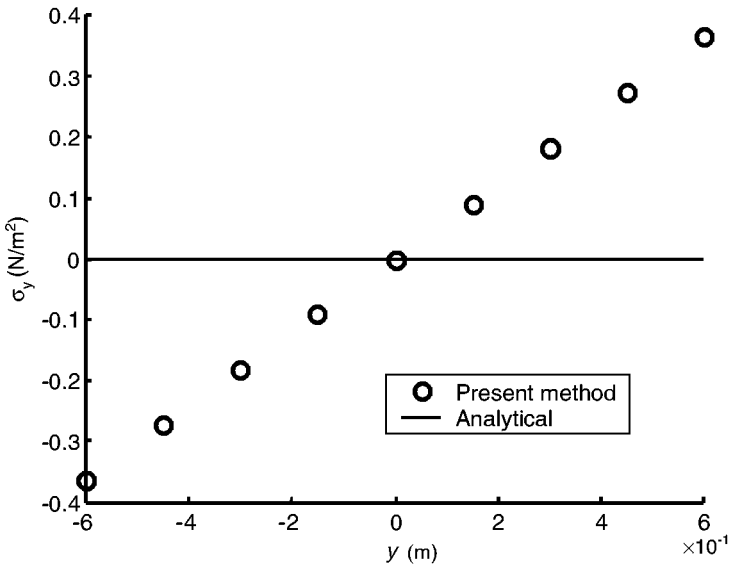


FIGURE 6.32
Analytical and the present EFG-CMLS numerical solutions for σ_y at the section of $x = L/2$ of the cantilever beam.

and

$$u_y = 0, \qquad \sigma_{xy} = 0 \quad \text{when} \quad y = 0 \tag{6.118}$$

The boundary condition at the right edge is

$$\sigma_{xx} = p, \qquad \sigma_{yy} = \sigma_{xy} = 0 \quad \text{when} \quad x = 5 \tag{6.119}$$

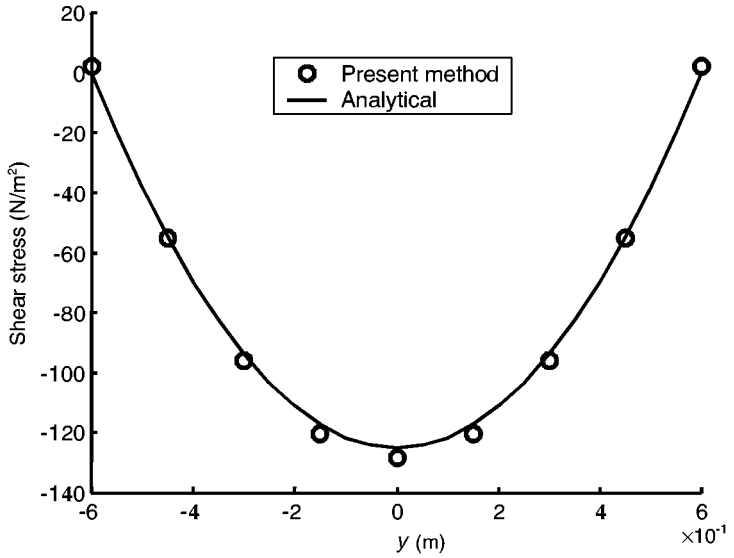


FIGURE 6.33
Analytical and EFG-CMLS numerical solutions for τ_{xy} at the section of $x = L/2$ of the cantilever beam.

TABLE 6.9

Comparison of Vertical Displacement at End of Beam

Number of Nodes	u_y (m) Exact	u_y (m) (EFG-CMLS)	%Error
18	–0.0089	–0.00792	11
24	–0.0089	–0.00837	6
55	–0.0089	–0.00887	0.34
189	–0.0089	–0.00891	0.1

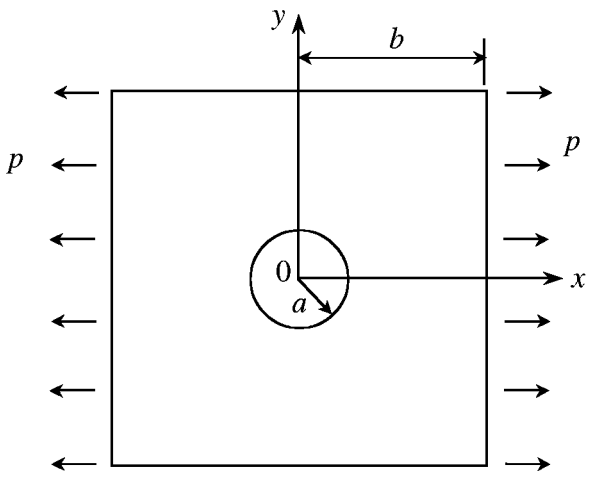


FIGURE 6.34
Plate with a hole subjected to a tensile load in the horizontal direction.

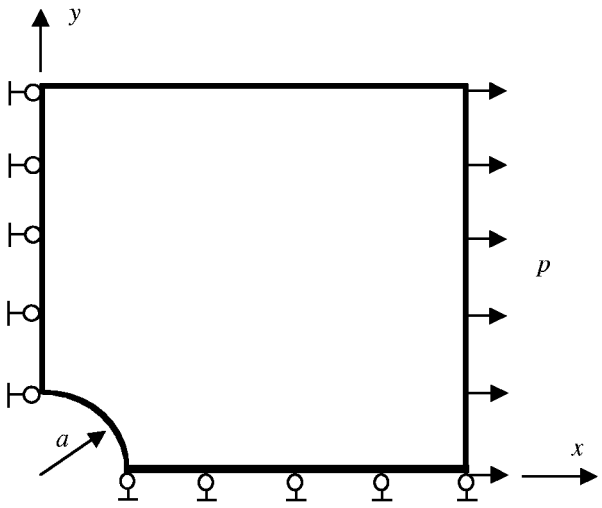


FIGURE 6.35
Quarter model of the plate with a hole subjected to a tensile load in the horizontal direction.

and the boundary condition at the upper edge is

$$\sigma_{xx} = 0, \qquad \sigma_{yy} = \sigma_{xy} = 0 \quad \text{when} \quad y = 5 \tag{6.120}$$

The parameters are listed as follows:

Loading: $p = 1$ N/m
Young's modulus: $E = 1.0 \times 10^3$ N/m^2
Poisson's ratio: $\nu = 0.3$
Height of the beam: $a = 1.0$ m
Length of the beam: $b = 5$ m

Symmetry conditions are imposed on the left and bottom edges, and the inner boundary of the hole is traction free. The tensile load in the x direction is imposed on the right edge. The exact solution for the stresses within the infinite plate is given by the following equations:

The displacement in the radial direction is given by

$$u_r = \frac{\sigma}{4\mu}\left\{r\left[\frac{\kappa-1}{2} + \cos 2\theta\right] + \frac{a^2}{r}[1 + (1+\kappa)\cos 2\theta] - \frac{a^4}{r^3}\cos 2\theta\right\} \tag{6.121}$$

and the displacement in the tangent direction can be calculated using

$$u_\theta = \frac{\sigma}{4\mu}\left[(1-\kappa)\frac{a^2}{r} - r - \frac{a^4}{r^3}\right]\sin 2\theta \tag{6.122}$$

where

$$\mu = \frac{E}{2(1+\nu)} \qquad \kappa = \begin{cases} 3 - 4\nu & \text{plane strain} \\ \dfrac{3-\nu}{1+\nu} & \text{plane stress} \end{cases} \tag{6.123}$$

The normal stress in the x direction can be obtained using

$$\sigma_x(x, y) = 1 - \frac{a^2}{r^2}\left\{\frac{3}{2}\cos 2\theta + \cos 4\theta\right\} + \frac{3a^4}{2r^4}\cos 4\theta \tag{6.124}$$

The normal stress in the y direction is

$$\sigma_y(x, y) = -\frac{a^2}{r^2}\left\{\frac{1}{2}\cos 2\theta - \cos 4\theta\right\} - \frac{3a^4}{2r^4}\cos 4\theta \tag{6.125}$$

and the shear stress is given by

$$\sigma_{xy}(x, y) = -\frac{a^2}{r^2}\left\{\frac{1}{2}\sin 2\theta - \sin 4\theta\right\} + \frac{3a^4}{2r^4}\sin 4\theta \tag{6.126}$$

where (r, θ) are the polar coordinates and θ is measured counterclockwise from the positive x axis. When the condition $b/a > 5$ is satisfied, the solution of a finite plate should be very close to that of an infinite plate. Therefore, the analytical results given in Equations 6.124 to 6.126 are employed as the reference results for comparison.

The CMLS method is used to perform the stress analysis. Two kinds of nodal arrangement are used, as shown in Figure 6.36. The results obtained using the present CMLS method for stress σ_x at $x = 0$ are plotted in Figure 6.37 together with the analytical results for the infinite plate. Figure 6.37 shows that the present CMLS method gives satisfactory results for the problem. The figure also shows that, as the number of the node increases, the results obtained are closer to the analytical solution.

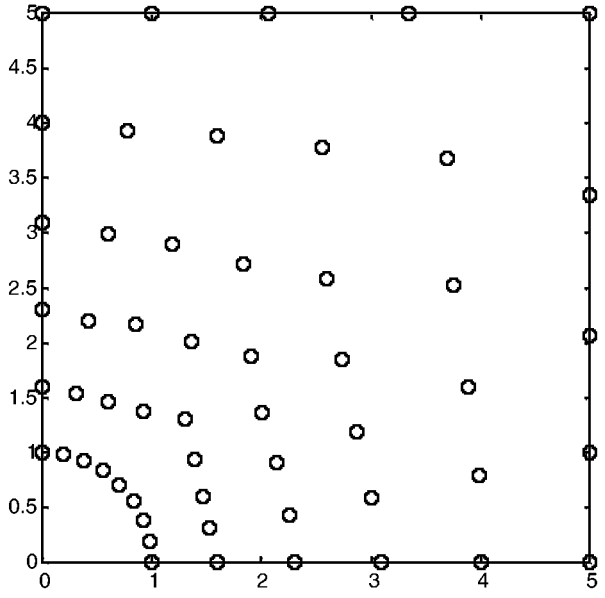
6.3.5 Computational Time

The main advantage of the CMLS method is that it does not increase the number of the unknowns and leads to a banded stiffness matrix. Therefore, it is computationally cheaper than EFG with Lagrange multipliers. Table 6.10 compares the CPU time used in the EFG code, when the present EFG-CMLS method, MLS with Lagrange multipliers, and MLS with penalty method are used for imposing the essential boundary condition in solving the cantilever beam problem. The comparison was done on an HP UNIX workstation. The table shows that the CMLS method saves a significant amount of CPU time compared with MLS with Lagrange multipliers in EFG formulation. However, the CMLS method is

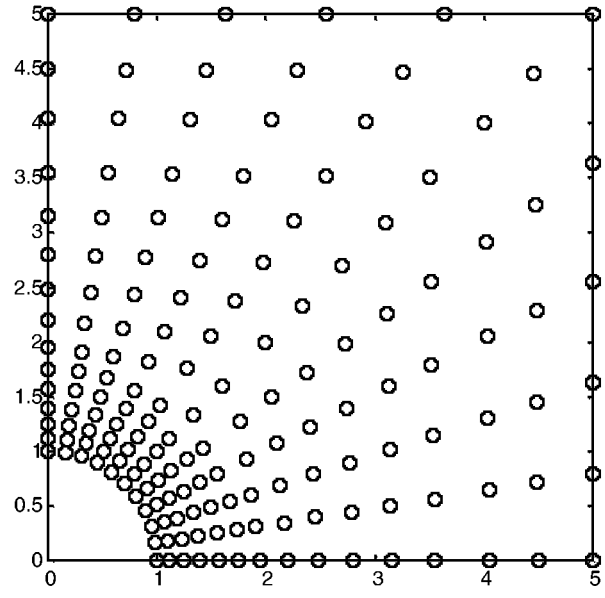
TABLE 6.10

CPU Time for EFG Code Using Different Method of Imposing Essential Boundary Conditions

	CPU Time (s)		
Nodes	EFG (MLS + Lagrange Multiplier)	EFG (Penalty)	EFG (CMLS)
55	1.1	0.6	0.7
189	35.4	3.5	9.3
561	115.2	13.8	30.8



(a)



(b)

FIGURE 6.36
Nodal arrangement for the infinite plate with a central circular hole. (a) 54 nodes; (b) 165 nodes.

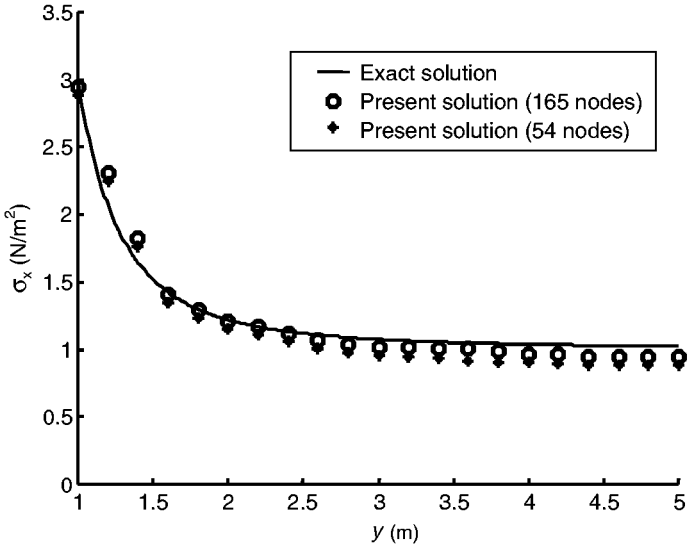


FIGURE 6.37
Comparison between the exact and present EFG-CMLS solution for stresses σ_x at $x = 0$.

less efficient than the penalty method. The reasons are (1) much time is used for computing the CMLS shape function compared with the MLS shape function, and (2) the number of integration points needs to increase because the CMLS shape functions are more complex than those of MLS. The advantage of CMLS over the penalty method may be that there is no need to choose the penalty factor. For large systems, the difference in CPU time between CMLS and the penalty method is expected to decrease as the majority of the CPU time is used for solving the system equation, which should be the same for both cases when CMLS and the penalty method are used.

6.3.6 Remarks

In this section, a technique called the constrained moving least squares method is introduced to construct the shape functions for MFree methods. CMLS enforces the approximation functions to pass through data points wherever necessary, while the usual MLS approximation is applied in other areas. CMLS treats the essential conditions at the stage of constructing shape functions, which simplifies the system equations. Hence, the system equations derived using the CMLS shape functions are positive definite and banded. The treatment of essential boundary conditions is as simple as in FEM. EFG-CMLS has problems passing the patch test, due perhaps to the incompatibility of CMLS approximation. More detailed study may be needed to fully realize the idea of CMLS. A method for solving the incompatibility issue for PIM shape functions is discussed in Chapter 8.

The idea of allowing certain conditions of a problem to be satisfied in the stage of constructing approximation functions should be explored further. MFree procedures for creating shape functions provide a lot of flexibility to meet different kinds of demands, not only on accuracy but also on constraints.

Before moving to the next section, it may be mentioned here that the imposition of essential boundary conditions can also be performed using finite elements attached to the MFree mesh on the portion of the essential boundaries (Krongauz and Belytschko, 1996). Coupling EFG with other numerical methods (see Chapter 13) is another alternative.