



Analyzing Plane-plate Bending with EFGM

Yajing Zhang, Maohui Xia & Yun Zhai

Department of Science, YanShan University

Qinhuangdao 066004, China

E-mail: zhang_yajing@yahoo.cn

Abstract

The moving least-square technique is used to construct shape function in the Element Free Galerkin Method at present, but sometimes the algebra equations system obtained from the moving least-square approximation is ill-conditioned and the shape function needs large quantity of inverse operation. In this paper, the weighted orthogonal functions are used as basis ones, the application in the calculation of plate bending shows that the improved moving least-square approximation is effective and efficient.

Keywords: EFGM, Moving least-square approximation, The weighted orthogonal functions, Plate bending

1. Introduction

The finite element method (Long, 2001, pp. 135-167) has been the main calculation method in the area of mechanics recently, but when it deals with the simulation of large distortion, the grid cells become distorted, then we need to divide the grids again to make equilibriums. So it makes some difficulties. In views of this situation, mesh-free method comes out. This is a new numerical analysis method. Because it need only the information of nodes and thoroughly or partly cancelled the meshes, and also because it has the advantage of excellent accuracy and rapid convergence, the mesh-free method rapidly developed in recent years. Doe (Belytdchko, 1994, pp. 229-256. & Liu, 2005, 64-90) reported that Element Free Galerkin Method (EFGM) is one of this methods. It bases on Moving least-square approximate (MLS) (Belyschko, 1996, pp. 3-47. & Genki, 2000, pp. 1419-1433) approximation and its basic idea is to solve problems with some discrete nodes.

But in the practical calculations, to ever fixed node, the MLS degenerates the classical least-square approximate, and we have to face the ill-conditioned function. This paper aims at the characteristic of MLS, to modify it using the weighted orthogonal functions to be the basic functions. This method is applied to the plate bending which has been studied previously (Fu, 2004, pp. 232-307), and the result shows that this method is effective and efficient comparing with the former one.

2. Moving least-square approximation

EFGM is brought forward by Belytschko, the basic idea is in the domain Ω bounded by Γ , $u(x)$ is the field function, where x is a point in the field. The local approximation of $u(x)$ (Genki, Y., m., 2000) can be defined as

$$u^h(x) = \sum_{j=1}^m P_j(x) a_j(x) = p^T(x) a(x), \quad \forall x \in \Omega \quad (1)$$

Where $p^T(x)$ is built utilizing Pascal's triangle with complete basis polynomial of m-dimensional, and the corresponding coefficient is $a(x)$, which is determined by MLS

$$J = \sum_I^n w(x - x_I) [u^h(x) - u^*(x_I)]^2 = \sum_I^n w(x - x_I) [p^T(x_I) a(x) - u^*(x_I)]^2 \quad (2)$$

Where x_1 is the node around x , which is in the domain of influence. $w(x - x_1)$ is the weight function We use the weight function just as follows

$$\omega_i(r_i) = \begin{cases} \frac{r_{m_i}^2}{r_i^2 + \varepsilon^2 r_{m_i}^2} (1 - \frac{r_i^2}{r_{m_i}^2}) & r_i \leq r_{m_i} \\ 0 & r_i \geq r_{m_i} \end{cases}$$

(Nayroles, 1992, pp. 307-318), this function is simple in form than that brought forward by Belytschko. We can easily get the parameter, Otherwise the approximation solution with either high rand can be got theoryly.

Eq. (4) can be expressed as

$$J(x) = (pa - u)^T \omega(x)(pa - u) \quad (5a)$$

Where

$$p = \begin{bmatrix} p_1(x) & p_2(x_1) & \cdots & p_m(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n) \end{bmatrix} \quad (5b)$$

$$\omega(x) = \begin{bmatrix} \omega(x - x_1) & 0 & \cdots & 0 \\ 0 & \omega(x - x_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega(x - x_1) \end{bmatrix} \quad (5c)$$

Taking the extreme value of Eq. (5), choosing functions $h(x)$, $g(x)$ at random, sign them as follows:

$$(h, g) = \sum_{I=1}^n \omega(x - x_1) h(x_I) g(x_I) \quad (6)$$

Then

$$a_1(x)(p_i, p_1) + a_2(x)(p_i, p_2) + \cdots + a_m(x)(p_i, p_m) = (p_i, u_I), i = 1, 2 \cdots, m \quad (7a)$$

forming it as matrix

$$\begin{bmatrix} (p_1, p_1) & (p_1, p_2) & \cdots & (p_1, p_m) \\ (p_2, p_1) & (p_2, p_2) & \cdots & (p_2, p_m) \\ \vdots & \vdots & \ddots & \vdots \\ (p_m, p_1) & (p_m, p_2) & \cdots & (p_m, p_m) \end{bmatrix} \begin{bmatrix} a_1(x) \\ a_2(x) \\ \vdots \\ a_m(x) \end{bmatrix} = \begin{bmatrix} (p_1, u_1) \\ (p_2, u_1) \\ \vdots \\ (p_m, u_1) \end{bmatrix} \quad (7b)$$

One obtains $a(x)$,

$$a = A^{-1}Bu \quad (8a)$$

Where

$$A = P^T \omega p, B = p^T \omega \quad (8b)$$

Taking the Eq. (8) into Eq. (1),

$$u^h(x) = \sum_{I=1}^n \Phi_I(x) u_I \quad (9a)$$

where

$$\Phi(x) = \sum_{j=1}^m p_j(x)(A^{-1}B)_{jI} \quad (9b)$$

The virtue of Moving least-square approximation is obvious, but sometimes it may make the equation ill-condition. Otherwise, because of large quantity of inverse operations, the calculation time increases. To solve the problems, we bring forward the Moving least-square approximation.

3. The modified Moving least-square approximation

The modified moving least-square approximation employs the weighted orthogonal functions to be basis ones, and overcome the difficulties of MLS. Considering the Congregation of nodes $\{x_i\}$ and the weight $\{\omega_i\}$ ($i = 1, 2, \dots, m$), if a group of function $\varphi_1(x), \varphi_m(x), \dots, \varphi_m(x)$ satisfies the situations just as follows:

$$(\varphi_k, \varphi_j) = \sum_{i=1}^n w_i \varphi_k(x_i) \varphi_j(x_i) = \begin{cases} 0 & k \neq j \\ A_k & k = j \end{cases} \quad (k, j = 1, 2, \dots, m) \quad (10)$$

One calls $\varphi_1(x), \varphi_2(x), \dots, \varphi_m(x)$ the group of orthogonal functions which about Congregation of nodes $\{x_i\}$ with the weight $\{\omega_i\}$.

In the MLS, if the basic function $p_i(x)$, ($i = 1, 2, \dots, m$) is the orthogonal one that about the $\{x_i\}$ with the weight $\{\omega_i\}$ ($i = 1, 2, \dots, m$), and the diagonal elements $(p_i, p_j) = 0$, ($i \neq j$) in the Eq. (7) can be reduced as

$$\begin{bmatrix} (p_1, p_1) & 0 & \cdots & 0 \\ 0 & (p_2, p_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (p_m, p_m) \end{bmatrix} \begin{bmatrix} a_1(x) \\ a_2(x) \\ \vdots \\ a_m(x) \end{bmatrix} = \begin{bmatrix} (p_1, u_1) \\ (p_2, u_1) \\ \vdots \\ (p_m, u_1) \end{bmatrix} \quad (11)$$

Then a_i ($i = 1, 2, \dots, m$) are solved from Eq. (11).

It is easily to see that $a_i(x) = \frac{(p_i, u_1)}{(p_i, p_i)}$, take it to the Eq. (1),

$$u^h(x) = \sum_{i=1}^m p_i(x) \frac{\sum_{l=1}^n \omega_l (x - x_l) p_l(x_l) u_l}{(p_i, p_i)} = \sum_{l=1}^n \Phi_l(x) u_l \quad (12a)$$

The shape function is

$$\Phi_l(x) = \omega_l (x - x_l) \sum_{i=1}^m \frac{p_i(x) p_i(x_l)}{(p_i, p_i)}$$

So in the course of solving a_i ($i = 1, 2, \dots, m$), we avoid the inverse operations and the solution of ill-condition equations (Cheng, 2003, pp. 181-186), and the definition and efficiency are all improved. It can be seen from the above that once a group of basic function which composed by weighted orthogonal functions are obtained, then we can get a_i ($i = 1, 2, \dots, m$), usually we can construct the orthogonal polynomial base using Schmidt orthogonalization.

4. Basic formulas

According to the Mindlin plate theory (Wang, 1996, pp. 154-167), the displacement and corner of a point are the independent field functions, the approximation corner deflection of this point can be imitate from the relational data in the domain, supposing that the nodes in the domain is n , and the array of the nodes of corner and deflection are

$$u^* = [u_1^* \quad u_2^* \cdots u_n^*]^T \quad (13a)$$

Where

$$u_i^* = [\omega_i^* \quad \theta_{x_i}^* \quad \theta_{y_i}^*]^T, \quad (i = 1, 2, \dots, n) \quad (13b)$$

According to MLS, the field function

$$u = [\omega \quad \theta_x \quad \theta_y]^T = \Phi u^* \quad (14a)$$

Where

$$\Phi = [\phi_1 I \quad \phi_2 I \cdots \phi_n I] \quad (14b)$$

is the shape function, and I is 3×3 unit matrix and n is the number of nodes in the field. The generalized strain of the plate is

$$k = \left[-\frac{\partial \theta_x}{\partial x} - \frac{\partial \theta_y}{\partial y} - \left(\frac{\partial \theta_x}{\partial y} + \frac{\partial \theta_y}{\partial x} \right) \right]^T = B u^* \quad (15a)$$

Where θ_x, θ_y are the rotation whose direction agree with $\frac{\partial \omega}{\partial x}, \frac{\partial \omega}{\partial y}$ and B is the strain array.

$$B = \begin{bmatrix} -\phi_{1,x} & 0 & 0 & \cdots & -\phi_{n,x} & 0 & 0 \\ 0 & -\phi_{1,y} & 0 & \cdots & 0 & -\phi_{n,y} & 0 \\ -\phi_{1,y} & -\phi_{1,x} & 0 & \cdots & -\phi_{n,y} & -\phi_{n,x} & 0 \end{bmatrix} \quad (15b)$$

The generalized relation between stress and strain is

$$M = [M_x \quad M_y \quad M_{xy}]^T = DK \quad (16a)$$

Where D is the elastic matrix, to the isotropic material

$$D = \frac{Et^3}{12(1-\mu^2)} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} = D_0 \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} \quad (16b)$$

Where D_0 is the stiffness matrix of the plate and is the Poisson ratio.

In order to easily solve the problem of the curving plate at any condition, penalty function is imposed on the essential boundary in the programming of this paper. To the elastic plate of isotropy, considering the functional of shearing deformed Mindlin plate theory,

$$\begin{aligned} \Pi = & \frac{1}{2} \int_{\Omega} k^T D k d\Omega - \int_{\Omega} q \omega d\Omega + \alpha_1 \left[\int_{\Omega} \left(\frac{\partial \omega}{\partial x} - \theta_x \right)^2 d\Omega + \int_{\Omega} \left(\frac{\partial \omega}{\partial y} - \theta_y \right)^2 d\Omega \right] \\ & + \alpha_2 \int_{s_1} [(\theta_n - \bar{\theta}_n)^2 + (\theta_s - \bar{\theta}_s)^2] ds + \alpha_3 \int_{s_1+s_2} (\omega - \bar{\omega})^2 ds - \int_{s_3} \bar{\theta}_n \omega ds + \int_{s_1+s_2} (\bar{M}_n \theta_n + M_n^- \theta_s) ds \end{aligned} \quad (17)$$

In the above equation, α_1 is the penalty factor of the shearing deformation in the functional, $\alpha_1 = Gt/2k$, G is the shearing mold, t is the thickness of the plate, k is the regulated coefficient which considering the shearing deforming non-proportionate distributed along the orientation of thickness. Take $k = 6/5$, α_2, α_3 is the penalty function which fulfill the essential conditions (physical significance and expression of the other signs can be found in the document (Genki, 2000, pp. 1419-1443). The numerical result shows that, when one takes $\alpha_2 = \alpha_3 = (10^3 - 10^6)E$ (E is the elastic mold), the definition of the result is fairly good, it shows that the essential boundary conditions which caused by this method, make the coefficient matrix symmetry and positive, easily solved.

5. Numerical example

A rectangular plate sustaining even load, one of its subtense ($x=0$ and $x=a$) are freely-supported, the third is free, the forth is built-in ($y=0$) (as fig. 1). The load $q = 2.2 \times 10^4 N/m^2$, the length of side are $a = 10.0m$, $b = 5.5m$, the thickness $t = 0.1m$, Young's modulus $E = 2.0 \times 10^{11} pa$, Poisson ratio $\mu = 0.3$. In this paper, we layout 16×9 even distribution nodes to get 15×8 integral sub-domains, and the Gaussian integration is 4×4 . The deflection of the plate center on the cross-section is as Tab.1, it shows that the solution of this paper is near to the solution obtained by ANSYS. The deflection along the center Line being parallel to x and y axis of the Plate are as Fig.1 and Fig.2.

6. Conclusions

This paper, which aims at the shortcoming of the MLS being easily forming the ill-condition function; the weighted orthogonal functions are used as basis functions to revise the MLS. Compared with the original, it breaks away from the inverse operation, and the amount of calculation is small. The result of the calculation of plate bending shows that the improved MLS is effective and efficient comparing with the result which obtained by the ANSYS software.

References

Belytschko, T., Lu, Y. Y., & Gu, L. (1994). Element-Free Galerkin Methods. *International Journal for Numerical Methods in Engineering*. 37. 229-256.

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., & Krysl, P. (1996). Meshless method: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139. 3-47.

Cheng, Y., m., & Chen, M., j. (2003). Elastic mechanics boundary mesh-free method. *Journal of Mechanical*. 35. 181-186.

Fu, B.L., (2004). Elastic mechanics energy principle and its application. Bei Jing. *Science Press*. pp. 232-307.

Genki, Y., m., & Tomonari, F. (2000). Recent developments of free mesh method. *International Journal for Numerical Methods in Engineering*. pp. 47. 1419-1443.

Liu, G., Liu, T.X., & Xie, Q., (2005). Mesh-free method and its application. *Northwestern Polytechnical Univerdity Press*. 64-90.

Long, Z.F., & Cen, S., (2001). New theory of finite element method-principle. programme. Progress. Bei Jing. *China Water Power Press*. pp. 135-167.

Nayroles, B., et al. (1992). Generalizing the finite element method: diffuse approximation and diffuse elements. *Comput Mech*. 10. 307-318.

Wang, X., c., & Shao, M. (1996). Basic principle and numerical method of finite element method. BeiJing. *Tsinghua University Press*. pp. 154-167.

Table.1 The deflection of the plate center on the cross-section $w (\times 10^{-4} m)$

relative distance	x/a		x/b	
	ANSYS solution	this paper's solution	ANSYS solution	this paper's solution
0	0.000	0.000	0.000	0.000
0.1	1.432	1.432	2.256	2.236
0.2	3.154	3.179	4.221	4.325
0.3	4.788	4.812	5.675	5.690
0.4	5.979	6.032	6.487	6.571
0.5	7.056	7.078	7.065	7.078
0.6	7.450	7.524	6.487	6.571
0.7	8.243	8.267	5.675	5.690
0.8	8.807	8.821	4.221	4.325
0.9	9.454	9.546	2.256	2.236
1.0	10.60	10.72	0.000	0.000

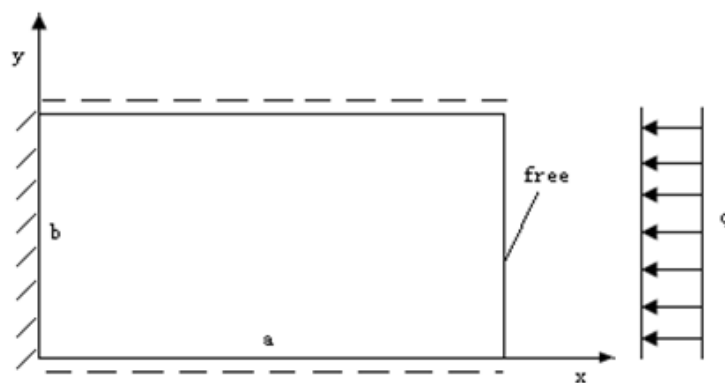


Figure 1. A loaded rectangular plate With complex edges conditions

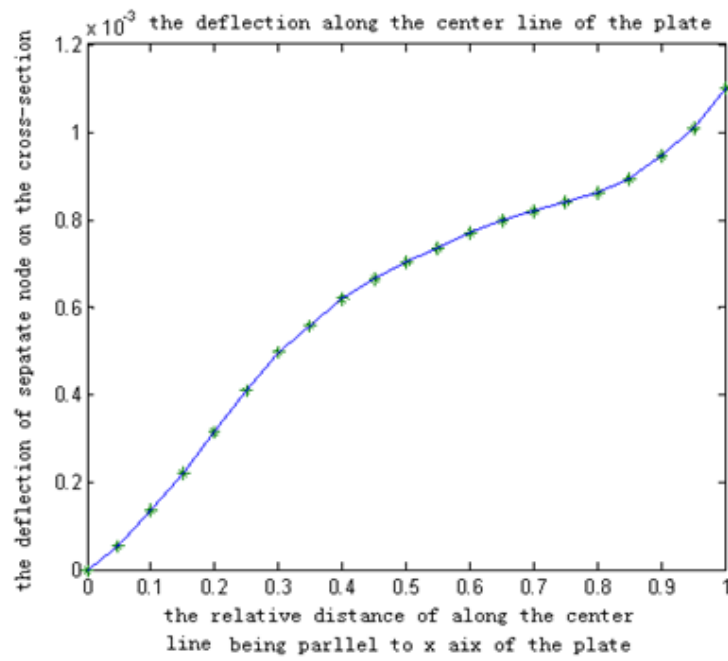


Figure 2. The deflection along the center Line being parallel to x axis of the Plate

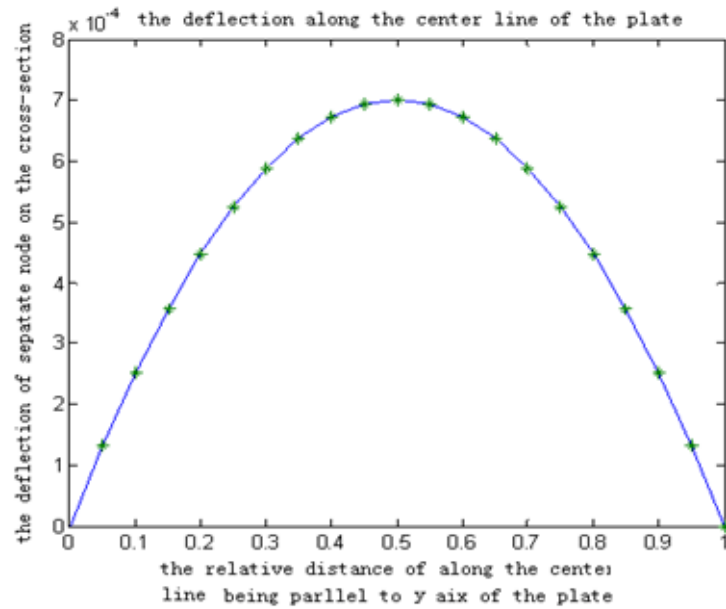


Figure 3. The deflection along the center Line being parallel to y axis of the Plate