



Meshless Methods

Soheil Mohammadi

Professor of Computational Mechanics

School of Civil Engineering

University of Tehran

Tel: +98-21-6111 2258, Fax: +98-21-6640 3808, Cell: +98-912-5030 392

Email: smoham@ut.ac.ir, <http://eng.ut.ac.ir/civ/smohammadi>

Constrained MLS- EFG (CMLS-EFG):

- The aim is to design a MLS shape function procedure that possesses the Kronecker delta function property only for the nodes on the essential boundaries.
- Proposed by Liu/Yang in 1999.
- 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.

CMLS-EFG:

- Let $u(x)$ be a function with nodal values U_s :

$$U_s = \{u_1 \quad u_2 \quad \cdots \quad u_n\}^T$$

- The function is approximated at point x using m terms of monomials:

$$u^h(x) = \sum_{j=1}^m p_j(x) a_j(x) = p^T(x) a(x)$$

- Now, assume the approximation function $u^h(x)$ is required to be equal to the nodal values at k constrained nodes: ($k < n, k < m$)

$$U_b = \{\bar{u}_1 \quad \bar{u}_2 \quad \cdots \quad \bar{u}_k\}^T$$

- And the remaining un-constrained nodes:

$$U_m = \{u_{k+1} \quad u_{k+2} \quad \cdots \quad u_n\}^T$$

CMLS-EFG:

- The function $u^h(x)$ can then be approximated by:

$$u^h(x) = p^T(x)a(x) = \left\{ p_b^T(x) \quad p_m^T(x) \right\} \begin{Bmatrix} a_b \\ a_m \end{Bmatrix}$$

$$p_b^T(x) = \{p_1(x) \quad p_2(x) \quad \cdots \quad p_k(x)\}$$

$$p_m^T(x) = \{p_{k+1}(x) \quad p_{k+2}(x) \quad \cdots \quad p_m(x)\}$$

$$a_b(x) = \{a_1 \quad a_2 \quad \cdots \quad a_k\}^T$$

$$a_m(x) = \{a_{k+1} \quad a_{k+2} \quad \cdots \quad a_m\}^T$$

- The constraint equations can be expressed as:

$$u_b = \{u^h(x_1) \quad u^h(x_2) \quad \cdots \quad u^h(x_k)\}^T$$

CMLS-EFG:

- The constraint equations can finally be expressed as:

$$u_b = P_b a_b + P_m a_m$$

$$P_b = \begin{bmatrix} p_b^T(\bar{x}_1) \\ p_b^T(\bar{x}_2) \\ \vdots \\ p_b^T(\bar{x}_k) \end{bmatrix} = \begin{bmatrix} p_1(\bar{x}_1) & p_2(\bar{x}_1) & \cdots & p_k(\bar{x}_1) \\ p_1(\bar{x}_2) & p_2(\bar{x}_2) & \cdots & p_k(\bar{x}_2) \\ \vdots & \vdots & & \vdots \\ p_1(\bar{x}_k) & p_2(\bar{x}_k) & \cdots & p_k(\bar{x}_k) \end{bmatrix}$$

$$P_m = \begin{bmatrix} p_m^T(\bar{x}_1) \\ p_m^T(\bar{x}_2) \\ \vdots \\ p_m^T(\bar{x}_k) \end{bmatrix} = \begin{bmatrix} p_{k+1}(\bar{x}_1) & p_{k+2}(\bar{x}_1) & \cdots & p_m(\bar{x}_1) \\ p_{k+1}(\bar{x}_2) & p_{k+2}(\bar{x}_2) & \cdots & p_m(\bar{x}_2) \\ \vdots & \vdots & & \vdots \\ p_{k+1}(\bar{x}_k) & p_{k+2}(\bar{x}_k) & \cdots & p_m(\bar{x}_k) \end{bmatrix}$$

CMLS-EFG:

- The k coefficients of a_b can be expressed in terms of the rest of the coefficients:

$$a_b = C_1 u_b - C_2 a_m$$

$$C_1 = P_b^{-1} \quad C_2 = P_b^{-1} P_m$$

- The vector of coefficients a_m is determined by the conventional MLS method over the free (unconstrained) nodes ($n-k$ nodes).
- At each free point x corresponding to u_m , a_m coefficients are chosen to minimize the weighted least square residual:

$$\Pi = \sum_{I=k+1}^n W(x - x_I) \left[p^T(x_I) a(x) - u_I \right]^2$$

CMLS-EFG:

- Minimization of Π with respect to coefficients results in:

$$A_1(x)a_b + A_2(x)a_m = B(x)u_m$$

$$A_1(x) = \sum_{I=k+1}^n W_I(x) p_m(x_I) p_b^T(x_I)$$

$$A_2(x) = \sum_{I=k+1}^n W_I(x) p_m(x_I) p_m^T(x_I)$$

$$B(x) = \begin{bmatrix} W_1(x) p_m(x_1) & W_2(x) p_m(x_2) & \cdots & W_n(x) p_m(x_{n-k}) \end{bmatrix}$$

$$W_I(x) = W(x - x_I)$$

CMLS-EFG:

- Combining:

$$a_b = C_1 u_b - C_2 a_m$$

$$A_1(x)a_b + A_2(x)a_m = B(x)u_m$$

- Results in:

$$a_b = E_1 u_b + E_2 u_m$$

$$a_m = D_1 u_b + D_2 u_m$$

$$D_1 = (A_1 C_2 - A_2)^{-1} A_1 C_1$$

$$D_2 = -(A_1 C_2 - A_2)^{-1} B$$

$$E_1 = C_1 - C_2 D_1$$

$$E_2 = -C_2 D_1$$

CMLS-EFG:

- The coefficient vector a can then be determined:

$$a = \begin{Bmatrix} a_b \\ a_m \end{Bmatrix} = \begin{bmatrix} E_1 \\ D_1 \end{bmatrix} u_b + \begin{bmatrix} E_2 \\ D_2 \end{bmatrix} u_m = \begin{bmatrix} E_1 & E_2 \\ D_1 & D_2 \end{bmatrix} \begin{Bmatrix} u_b \\ u_m \end{Bmatrix}$$

- And the corresponding approximated $u^h(x)$ becomes:

$$\begin{aligned} u^h(x) &= p^T(x) a(x) = \begin{bmatrix} p_b^T & p_m^T \end{bmatrix} \begin{Bmatrix} a_b \\ a_m \end{Bmatrix} \\ &= \begin{bmatrix} p_b^T & p_m^T \end{bmatrix} \begin{bmatrix} E_1 & E_2 \\ D_1 & D_2 \end{bmatrix} \begin{Bmatrix} u_b \\ u_m \end{Bmatrix} \end{aligned}$$

CMLS-EFG:

- CMLS shape functions:

$$u^h(x) = p^T(x)a(x) = \begin{bmatrix} p_b^T & p_m^T \end{bmatrix} \begin{Bmatrix} a_b \\ a_m \end{Bmatrix} = \begin{bmatrix} p_b^T & p_m^T \end{bmatrix} \begin{bmatrix} E_1 & E_2 \\ D_1 & D_2 \end{bmatrix} \begin{Bmatrix} u_b \\ u_m \end{Bmatrix}$$

$$u^h(x) = p^T(x)a(x) = \begin{bmatrix} \phi_b & \phi_m \end{bmatrix} \begin{Bmatrix} u_b \\ u_m \end{Bmatrix} = \Phi U_s$$

$$\phi_b = p_b^T E_1 + p_m^T D_1$$

$$\phi_m = p_b^T E_2 + p_m^T D_2$$

- Derivatives of CMLS shape functions:

$$\phi_{b,x} = p_{b,x}^T E_1 + p_b^T E_{1,x} + p_{m,x}^T D_1 + p_m^T D_{1,x}$$

$$\phi_{m,x} = p_{b,x}^T E_2 + p_b^T E_{2,x} + p_{m,x}^T D_2 + p_m^T D_{2,x}$$

CMLS-EFG:

- The CMLS-EFG Galerkin weak formulation is similar to FEM:

$$\int_{\Omega} \delta \varepsilon^T \sigma \, d\Omega - \int_{\Omega} \delta u^T b \, d\Omega - \int_{\Gamma_t} \delta u^T \bar{t} \, d\Gamma = 0$$

- Results in the discrete set of equations:

$$KU = F$$

$$K = \bigcap_{I,J=1}^{n_t} K_{IJ} \quad K_{IJ} = \int_{\Omega} B_I^T C B_J \, d\Omega$$

$$F = \bigcap_{I=1}^{n_t} f_I \quad f_I = \int_{\Omega} \hat{\Phi}_I^T b \, d\Omega + \int_{\Gamma_t} \hat{\Phi}_I^T \bar{t} \, d\Gamma$$

$$\Phi_I = \begin{bmatrix} \phi_I & 0 \\ 0 & \phi_I \end{bmatrix} \quad B_I = \begin{bmatrix} \phi_{I,x} & 0 \\ 0 & \phi_{I,y} \\ \phi_{I,y} & \phi_{I,x} \end{bmatrix}$$

CMLS-EFG:

1. 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 nodal constraints.
2. 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 should be used.
3. 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.
4. For 2D problems, the appropriate basis function should be polynomials of order 2 or 3.

CMLS-EFG:

5. Including more free nodes can help to reduce the chance of producing singular moment matrices in the process of computing CMLS shape functions. One drawback is the increased possibility of having a singular moment matrix P because fewer free nodes are used.
6. The stiffness matrix K is banded and symmetric.
7. Ensuring accurate numerical integration when using CMLS is also more difficult compared with CMLS because the order of the CMLS shape functions is usually higher.
8. In limit, the method is somewhat similar to that using PIM shape functions.

CMLS-EFG:

9. The Gauss quadrature will not be able to produce exact results for this integration, because it is designed for integrating polynomial functions.
10. The quadrature error may increase when it is applied to complex integrands, especially fractional functions which in general can not be represented exactly by polynomial functions.
11. CMLS approximation is incompatible.
12. CPU time comparison:

$$\begin{matrix} \text{EFG}_{\text{penalty}} < \text{EFG}_{\text{CMLS}} < \text{EFG}_{\text{Lagrange}} \\ (1)(t_1) & (2-3)(t_1) & (10)(t_1) \end{matrix}$$

