MESHFREE SHAPE FUNCTION FROM MOVING LEAST SQUARE

Jeetender Singh Kushawaha

Department of Mechanical Engineering, Indus Institute of Technology and Management,
(Affiliated to Gautam Buddh Technical University),
Kanpur, Pin-209202, India.
E-mail address: k.jitendrasingh@yahoo.com, *Ph*: +91-9450935651

Abstract

The main differentiating point between the meshfree and finite element methods is the shape function. The paper is intended to elaborate the construction of the moving least square approximation shape function and their derivatives in one-dimension, by presenting the related plots of shape function and its derivatives; with different parameters. A tapered bar subjected to tensile load using the element free galerkin method, applying the moving least square approximation shape function, has been modeled and tip displacement with exact and meshfree solution has been found to be in good agreement.

Keywords: Meshfree, weight function, shape function, basis function, matlab

1. Introduction

The development of the approximate methods for the numerical solution of practical problems, re-presentable by partial differential equations has helped engineers, physicists and mathematicians in analyzing the complex phenomena at reduced costs. The finite element method (FEM) is one of the most popular, well-developed and possessing much versatility in analyzing complicated phenomena, whose behavior is governed by increasingly complex partial differential equations.

In recent years, meshfree methods have been developed as an alternative numerical tool in effort to eliminate known drawbacks of the finite element methods. The nature of the various approximation functions used by meshfree methods allows the representation of the problem domains by simply adding or deleting nodes where-ever desired. The prior knowledge of nodal connectivity to form a discrete element as in finite element methods is not necessary, only nodal coordinates and their domain of influence are sufficient to represent the problem domain.

There are several meshfree methods under current development, including the most versatile element free Galerkin (EFG) method proposed by Belytschko et al. [1], Meshless Local Petrov-Galerkin (MLPG) method proposed by Atluri et al. [1, 2] and many other methods. These well-established EFG and MLPG method use the shape functions which are derived from moving least-square approximation. The main purpose of this paper is to elaborate the construction of meshfree shape function using the MLS approximation.

2. Meshfree shape function

The meshfree shape function is the central and most important issue and main differentiating point for the meshfree methods from the finite element methods. There are a number of ways proposed to construct the meshfree shape functions [3]. In this paper the finite series representation, moving least square approximation method is studied and elaborated considering the programming and implementation aspects, the different plots have been included to indicate the different steps and effects of various parameters exclusively.

A good meshfree shape functions needs to satisfy the following conditions as for as efficiently possible:

1) A compulsory condition for the shape function is the satisfaction of partition of unity.
$$\sum_{i=1}^{n} \phi_i(x) = 1 \quad (1)$$
2) It should be able to manage the reasonable randomness of distribution of nodes.
3) The algorithm should be numerically stable.
4) The shape function constructed should have the consistency to enable the convergence of numerical results with increase of nodes.
5) The domain of influence should be compact.
6) The Kronecker-delta property should be satisfied.
7) The computational efficiency should be at par with FEM.
8) Ideally, the field approximation should be compatible throughout the problem domain.

3. Meshfree shape function construction

Moving least square (MLS) was originated by mathematicians for data fitting and surface construction, the procedure for constructing the meshfree shape function using the MLS approximation starts with the assumption that x_1, x_2, x_3, x_4, and x_n are the nodes distributed in the domain **Ω** and the associated field variable or nodal parameter with these node are u_1, u_2, u_3, u_4, and u_n. The different sampling points are represented by **x** which is the locations of different points and x_I represents the nodal points distributed in the domain of the problem **Ω** and the number of these sampling points will dictate the smoothness of the curve plotted for the weight and shape functions.

In meshfree methods approximation of the field variable function *u(x)* without any connectivity information between the nodes is done by considering the approximation as the product of a vector of polynomial basis function and a set of unknown coefficients varying with *x*. This can be also put as the approximation of the function *u(x)* is obtained by assuming the approximate solution as a polynomial function represented as:

$$u(x)_{appx} = a_0(x) + a_1(x)x + a_2(x)x^2 \quad (2)$$

For linear basis in 1D

$$u(x)_{appx} = a_0(x) + a_1(x)x + a_2(x)x^2 \quad (3)$$

For quadratic basis in 1D

Considering the linear polynomial function in one-dimension, equation (2), can be written in matrix form:

$$u(x)_{appx} = \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} a_0(x) \\ a_1(x) \end{bmatrix} \tag{4}$$

$$u(x)_{appx} = \sum_{j=0}^{m} p_j(x) a_j(x) = p^T(x) a(x) \tag{5}$$

Where $p^T(x) = \begin{bmatrix} 1 & x \end{bmatrix}$ (6)

and

$$a^T(x) = \begin{bmatrix} a_0(x) & a_1(x) \end{bmatrix} \tag{7}$$

The value of approximate solution or approximation $u(x)_{appx}$ can be evaluated by determining the unknown coefficients of *x*. The unknown coefficients are evaluated by minimizing the difference between the local approximation at that point or the considered node in the *support domain* and the nodal parameter for the node *I*, i.e.

$$u_I = u(x_I) \tag{8}$$

The approximated value of field variable at the local nodes is given by:

$$u(x, x_I)_{appx} = p^T(x_I) a(x) \tag{9}$$

Again it is emphasized that *a(x)* is arbitrary function of the sampling point's ***x*** and $\boldsymbol{x_I}$ represents the nodal points. The minimization process starts with the construction of a weight residual functional with respect to unknown coefficients, considering the equation (4) and (5), and given by:

$$J = \sum_{I=1}^{n} w(x - x_I)\left(u(x, x_I)_{appx} - u_I\right)^2 \tag{10}$$

The minimization of this functional *J* produces a set of linear equations [1, 3 and 4]:

$$\mathbf{A}(x)\mathbf{a}(x) = \mathbf{B}(x)\mathbf{u} \tag{11}$$

$$\mathbf{a}(x) = \mathbf{A}^{(-1)}(x)\mathbf{B}(x)\mathbf{u} \tag{12}$$

Where **A(x)** is known as weight moment matrix and given by:

$$\mathbf{A}(x) = \sum_{I=1}^{n} w(x - x_I)\, p(x_I)\, p^T(x_I) \tag{13}$$

$$\mathbf{A}(x) = w(x - x_1)\begin{bmatrix} 1 \\ x_1 \end{bmatrix}\begin{bmatrix} 1 & x_1 \end{bmatrix} + w(x - x_2)\begin{bmatrix} 1 \\ x_2 \end{bmatrix}\begin{bmatrix} 1 & x_2 \end{bmatrix} + w(x - x_n)\begin{bmatrix} 1 \\ x_n \end{bmatrix}\begin{bmatrix} 1 & x_n \end{bmatrix} \tag{14}$$

And $$\mathbf{B}(x)=\left[B(x_1)\quad B(x_2)\quad\cdots\quad B(x_n)\right] \tag{15}$$

$$B(x_I)=w(x-x_I)\begin{bmatrix}1\\ x_I\end{bmatrix} \tag{16}$$

The nodal parameters of the field variables are represented by the vector **u**:

$$\mathbf{u}=[u_1\quad u_2\cdots\quad u_n]^{\mathrm{T}} \tag{17}$$

The new variable introduced in the equation (13) is known as the weight function. The weight function considered in this study is cubic spline given by:

$$w(x-x_1)=\begin{cases}\frac{2}{3}-4r^2+4r^3 & if\ \ r\le\frac{1}{2}\\ \frac{4}{3}-4r+4r^2-4r^3 & if\ \ 0.5<r\le 1\\ 0 & if\ \ r>1\end{cases} \tag{18}$$

Another weight function known as Quartic spline, is also used to present the effect of weight function on the shape function, is given as:

$$w(x-x_1)=\begin{cases}1-6r^2+8r^3-3r^4 & if\ \ r\le 1\\ 0 & if\ \ r>1\end{cases} \tag{19}$$

Where r = $|x\text{-}x_I|/d_I$ *and* ,d_I is the radius of influence domain or radius of support domain of the node.

Substitution of equation (12) into the approximate solution equation (5), leads to:

$$u(x)_{appx}=\sum_{(j=0)}^{m}p_j(x)\sum_{(I=1)}^{n}A^{(-1)}(x)B(x)u_I \tag{20}$$

$$u(x)_{appx}=\sum_{I=1}^{n}\phi(\mathrm{x}_\mathrm{I})\mathrm{u}_\mathrm{I}=\mathbf{\Phi}^{\mathbf{T}}(x)\mathbf{u} \tag{21}$$

$$\mathbf{\Phi}(x)=[\phi_1(x)\quad \phi_2(x)\quad \ldots\quad \phi_n(x)] \tag{22}$$

The MLS shape function for i^{th} node is defined by:

$$\phi_I(x)=p^T(x)(A^{(-1)}(x)B(x))_I \tag{23}$$

The MLS shape function for the middle node i.e. I= 3 is expanded to clear the programming aspects and distinguishing between the nodes and sampling points. The domain **Ω**= (0, 1) is represented by five nodes located at points (0 0.25 0.5 0.75 1.0). ϕ_3(x), indicates the shape function associated with the *support node three*, it is a vector having the values corresponding to the number of *support nodes* in the support or influence domain of middle node located at x = 0.5 and is given by:

$$\phi_3(x) = [1 \quad \mathrm{x}_3]\mathbf{A}^{-1}(\mathrm{x})w(x - x_3)\begin{bmatrix} 1 \\ x_3 \end{bmatrix} \tag{24}$$

The influence domain is considered to be equal to d_I = 0.4375, this is the radius of circle; in the area of this circle all the nodes are influencing or contributing to the approximation. The number of nodes in the influence domain of the middle node is three and their location are at *x = 0.25, x = 0.5 & x = 0.75*. The shape function associated with this middle or *star* node at *x* = 0.5, is elaborated to bring out the clarity. The evaluation of weight moment matrix-A; associated with middle node:

$$\mathbf{A}_3(0.25) = w(0.25 - 0.5)\begin{bmatrix} 1 \\ 0.5 \end{bmatrix}[1 \quad 0.5] \tag{25}$$

$$\mathbf{A}_3(0.5) = w(0.5 - 0.5)\begin{bmatrix} 1 \\ 0.5 \end{bmatrix}[1 \quad 0.5] \tag{26}$$

$$\mathbf{A}_3(0.75) = w(0.75 - 0.5)\begin{bmatrix} 1 \\ 0.5 \end{bmatrix}[1 \quad 0.5] \tag{27}$$

The calculations for the shape function are:

$$\phi_3(0.25) = [1 \quad 0.5]\mathbf{A}^{-1}(0.25)w(0.25 - 0.5)\begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \tag{28}$$

$$\phi_3(0.5) = [1 \quad 0.5]\mathbf{A}^{-1}(0.5)w(0.5 - 0.5)\begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \tag{29}$$

$$\phi_3(0.75) = [1 \quad 0.5]\mathbf{A}^{-1}(0.75)w(0.75 - 0.5)\begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \tag{30}$$

After performing the calculation we receive the value of shape function corresponding to $\phi_{3\ (0.25)}$ = 0.1197, $\phi_{3\ (0.50)}$ = 0.7605 & $\phi_{3\ (0.75)}$ = 0.1197. This can be better assimilated by figure-1, representing the shape function, using the linear basis function, cubic spline weight function and the radius of dimension of support domain, d_I = 0.4375 with two different number of sampling points (SP).

3.1. Shape function derivative

The derivatives of shape function can be calculated by applying the product rule to equation (23):

$$\phi_I(x) = p^T(x)(A^{(-1)}(x)B(x))_I \qquad (31)$$

The first derivative is obtained as

$$\phi_{I,x} = (p^T(x)(A^{(-1)}(x)B(x))_I), x \qquad (32)$$

$$\phi_{I,x} = p^T{}_{,x} A^{(-1)}(x)B(x)_I + p^T A^{(-1)}{}_{,x} B(x)_I + p^T A^{(-1)}B(x)_{I,x} \qquad (33)$$

The further expansion for the equation is similar to as described for the shape function.

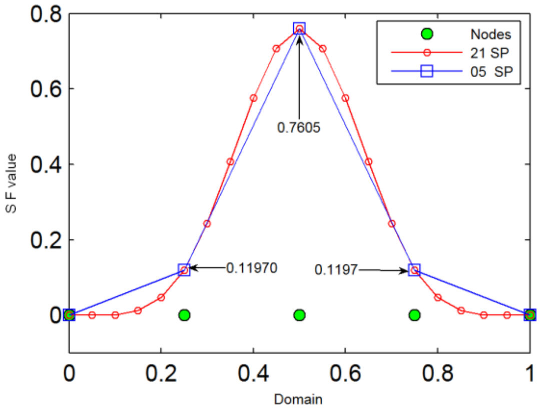


Fig.1. Shape function for middle node

3.2. Nan Error

The critical and erroneous situation occurs whenever the derivative of shape function is to be evaluated at the node which is also a sampling point or point of interest. The derivative of weighting function and shape function cease to exist, this condition if occurs during the meshfree program execution the results will not be logical and accurate. Figure-2 presents the plot of shape function derivatives, in this case the *NaN* error has solved by shifting each node by a distance of 0.0001 from the locations of uniform distribution, within the boundary of domain. The next combined figure-3 presents the plots of weight function, shape function and their derivatives with (above) and without (below) implementing the node shifting. The derivatives of the weighting function are discontinuous and the shape function derivatives cease to exist.

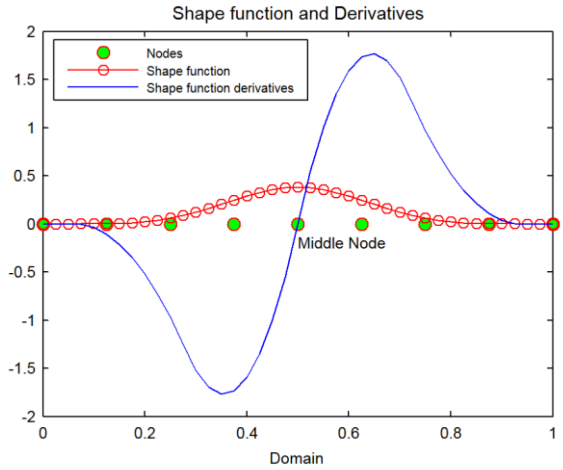


Fig.2. Shape function and derivative for middle node

4. Matlab program implementation

The programming flow chart for the moving least square shape function and derivatives, given by equation (23 and 33), is presented in figure-4. The program steps are elaborated, here under:

1) Enter the domain of the problem and represent the geometry into nodal and sampling points.
2) Initialize the support domain of influence, matrices for the weight function and shape function
3) Initialize the first “for” loop for number of nodes as the shape function needs to be calculated at each node, within this “for” loop other loops are initialized to:
    a. Find the values of weight function and derivatives at the nodal points.

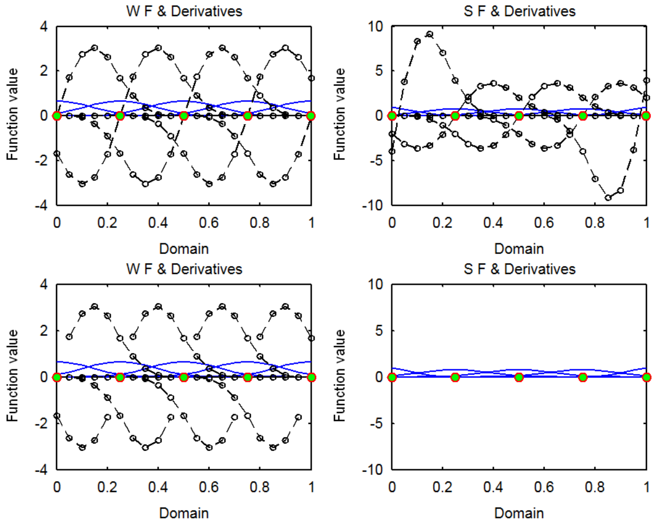


Fig.3. Influence of node shifting

    b. Find the values of weight function and derivatives at the sampling points.
    c. Find A-matrices and their inverse at nodes.

d. Find B-matrices and derivatives at sampling points.
e. Find the value of shape function and derivatives at the nodal and sampling points.
f. End of first "for" loop.

4) Plot the nodal points
5) Plot weight function and derivatives
6) Plot the shape function and derivatives using nodal points and sampling points
7) Add legend, x and y label to the plots

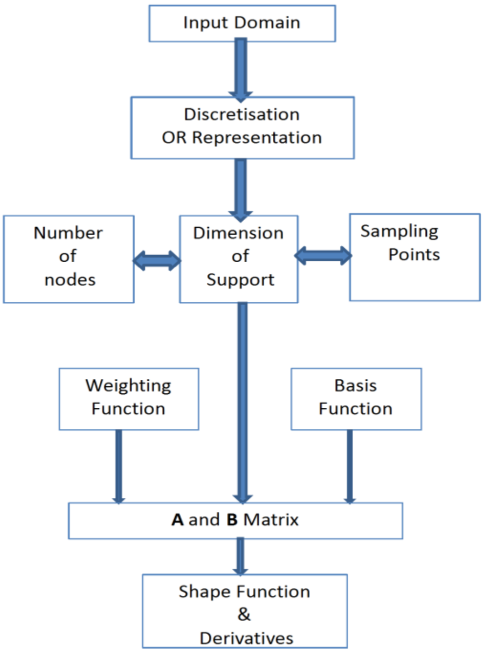


Fig.4. Flow chart for shape function calculation

5. Results and discussion

The various findings related to the properties and effects of various factors on the behavior of the shape function are included along with the plots obtained using the matlab program.

5.1. Partition of unity

The figure-5 presents the shape function for the 9 nodes and the values of the shape function for the middle node, corresponding to other node locations, marked by circles; are presented in the Table-1, to show that the shape function approximates and satisfies partition of unity conditions subject to use of constant terms.

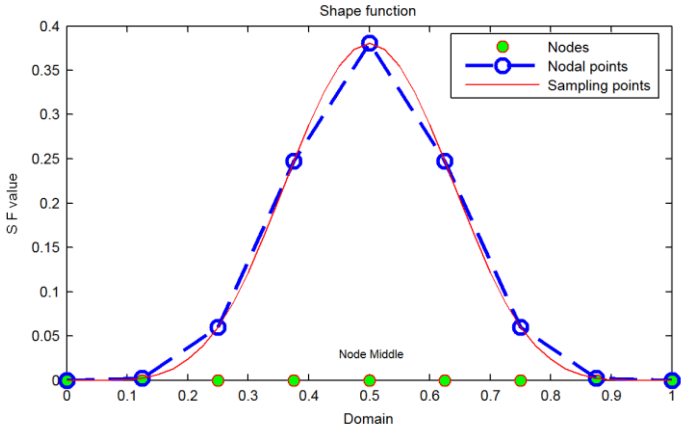


Fig.5. Shape function to confirm partition of unity

Table-1 Partition of unity

Node	1	2	3	4	5	6	7	8	9	Total
$\phi_{(5)}$	0	0.0022	0.0599	0.2475	0.3807	0.2475	0.0599	0.0022	0	0.9999

5.2. Lack of Kronecker delta

The table-1 presents the values of shape function associated with node number five $x = 0.5$, the value of shape function at this node is $\phi_{(5)}(x) = 0.3807 \neq 1.0$. Thus the moving least square shape functions do not satisfy the Kronecker delta condition.

5.3. Compact Support

The shape function provides the compact support and the value is zero outside the support domain of influence of the concerned node.

5.4. Continuity

The shape functions have high order of continuity, although only linear basis has been used, because of the fact that shape functions inherit the continuity of the weighting function. The weighting function used is cubic spline. Figure-6 indicates the effect of the weighting functions, as the two weighting functions have different shapes and order of continuities the shape function inherits these features of the weighting functions.

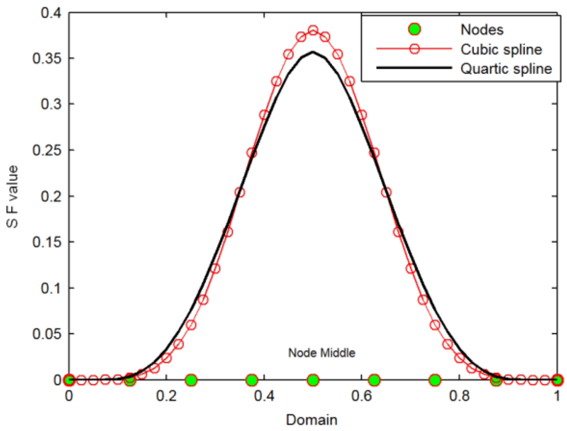


Fig.6. Shape function with cubic and quartic spline using linear basis

5.5. Weighting function

The selection of the weighting function plays a very vital role in the formulation and solution of meshfree methods. The shape functions generated with cubic spline and quartic spline are represented in figure-6. The conclusion from this can be made that the cubic spline weighting function gives the shape function which posses more local character.

5.6. Bell shape

The shape functions possess the *bell shape*, presented in figure-1 through 6, as the number of nodes in the support domain is increased the height of the bell gets lowered and spread gets lengthened increasing the global influence.

5.7. Reduction of peaks

The peak values of the shape functions fall down as the number of the nodes in the support domain is increased as a result the smoothness increases and the local character starts decreasing and the behavior tends to be global. Comparing figure-1 & 5 give a better visualization; note the values on the y-axis.

5.8. Sampling points

The difference between the nodal points and sampling points is very vital and made very clear and unambiguous by figure-7, representing the plots for the first and the second nodes, as the number of sampling points is increased the smoothness of the curve is increased without affecting the peak values of the shape function corresponding to the nodes.

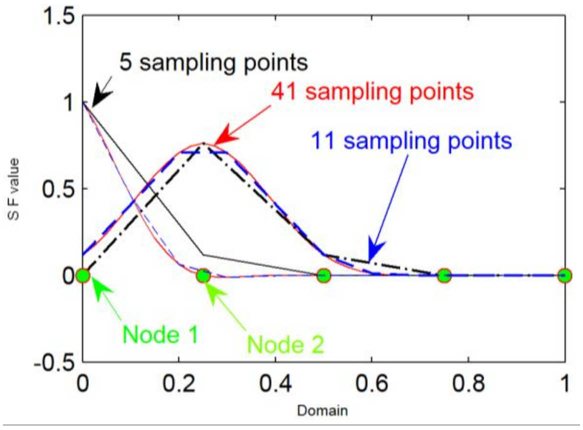
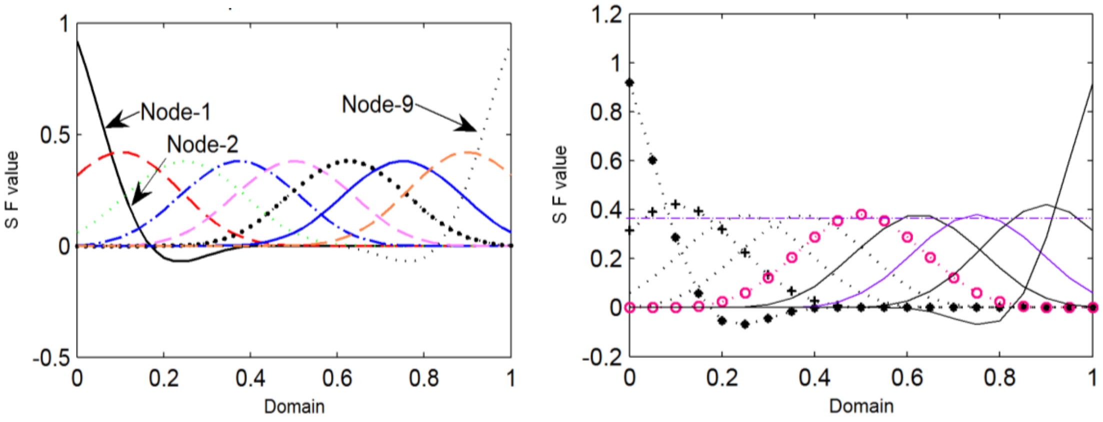


Fig.7. Shape function for the first and second node

5.9. Mirror Image

The shape functions are the mirror image of each other from the central node, presented in figure-8(a). These approach or/and reach unit values near the boundaries of the domain so that the imposition of the boundary condition is simplified. The approach to unit value is reasonably gradual and emphasized by drawing a horizontal line in figure-8(b).



(a) Sampling points-51 (b) Sampling points-21

Fig.8. Shape function for 9 nodes (d_I = 0.4375) Linear basis, Cubic spline

5.10. Influence of support domain

The effect of influence of support domain is represented by figure-9, it can be concluded that with the increase of support domain the local behavior of the shape function diminishes and as the nodes in the support domain are decreased the shape function value approaches to unity. In this condition the shape function will interpolate through the nodal values, if the A-matrix is invertible, however if the number of node in the support domain becomes less than the number of monomials in the basis function, inverse of A-matrix will not exist.

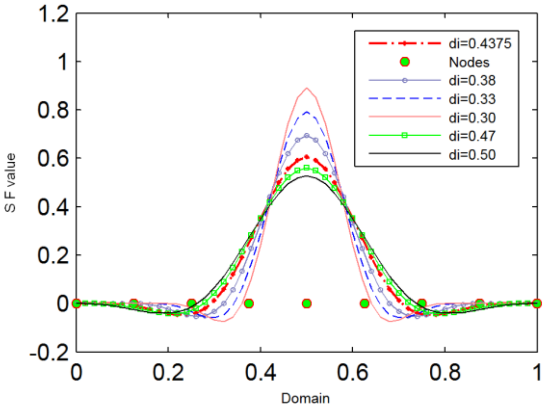


Fig.9. Influence of support domain, d_I

5.11. Basis function

The effect of the basis function on the shape function is presented by figure-10. Moving least square shape functions using linear, quadratic and cubic basis function and cubic spline weighting function are computed and plotted to visualize the effect of basis function on the shape functions. The study concludes that as the order of basis function is increased, the value of $\phi_{(5)}(x)$ increases to maximum and becomes constant as the order of basis and weighting function become equal.

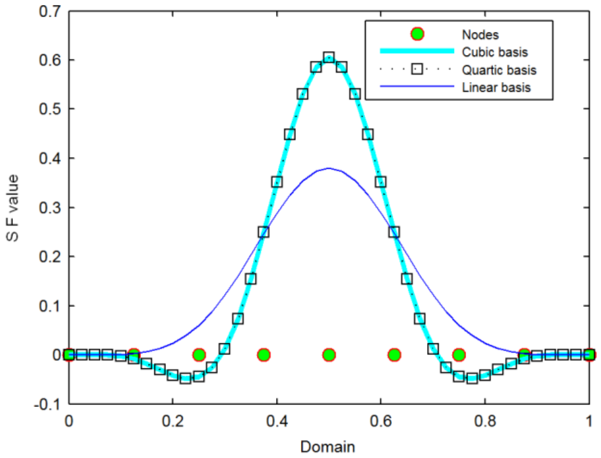


Fig.10. Shape function with different basis functions

5.12. Basis function

The effect on the geometry of the shape function is presented in the figure-11, the quadratic basis function with quartic spline weight function ; 11 & 25 nodes have been used to represent the domain of Ω = (0-1) to get the plots for the first and middle node. The number of sampling points is 51, to get the smooth curves. The shape function loses its local character and adds to the global nature so that the convergence of the solution is achieved with the increase of number of nodes in the domain.

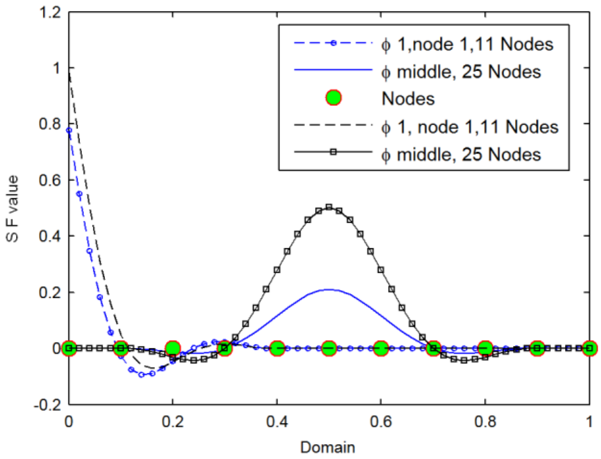


Fig.11. Shape function with coarse and fine nodal distribution

5.12. Validation

The matlab program for the shape function is validated by solving the problem from the elastostatics. The problem of tapered bar has been considered, which posses a mix of complexity and easier exact solution, from [5, 6]. A tapered bar of uniformly varying cross sectional area, A_1 = 1 m^2, A_2 = 0.5 m^2 on each end, subjected to tensile point load, P = 1000 kN, and young's modulus, E = 200 GPa; was modeled and solved using the element free galerkin method and the results are positive and in good agreement to the exact solution, validating that the developed of formulation and matlab program. The figure-12 represents the comparative plots obtained by exact solution and the element free galerkin method with the variation of support domain of influence. The increase in the number of nodes will bring the convergence of the meshfree solution to the exact solution.

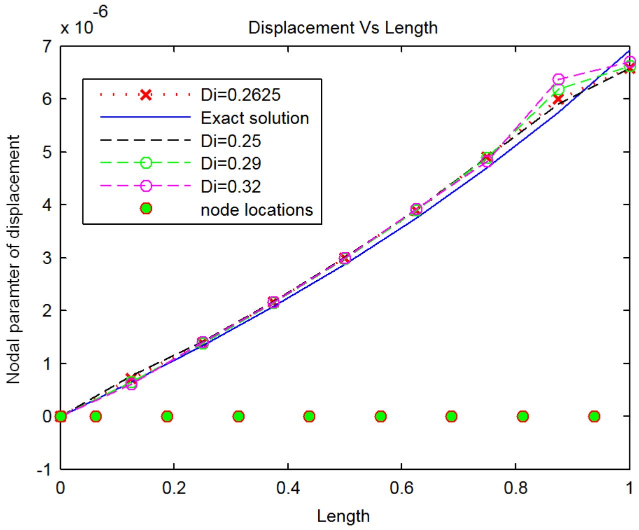


Fig.12. Comparison of EFG and exact solution

6. Conclusion

The moving least square shape function and its derivative is programmed, plotted and studied for the various characteristics of the shape functions, elaborated in context to the parameters affecting the properties of shape function. The danger of *not a number* in contribution to meshfree solution errors is highlighted and prevention methodology has been implemented to eradicate the error generation.

References

[1] Belytschko, T., Lu, Y.Y. and Gu, L. (1994). Element-free Galerkin methods. Int. J. Numer. Methods Eng 37: 229-256.

[2] Atluri, S.N., Kim, H.G. and Cho, J.Y. (1999). A critical assessment of the truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) methods. Comput. Mech. 24: 348-372.

[3] Liu. G. R. (2003). Mesh Free Methods: Moving beyond the finite element methods. CRC Press, Boca Raton, USA.

[4] Belytschko, T. and Dolbow J. (1998). An introduction to programming the meshless element free galerkin method. Archives Comput. Methods Eng. 5: 207-241.

[5] David V. Hutton. (2004). Fundamentals of finite element analysis. Tata Mc-Graw Hill.

[6] Kushawaha, J.S. (2011). Meshfree analysis of elastic bar. Lambert Academic Publishing.