IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 04, Issue 11 (November. 2014), ||V2|| PP 23-35

Construction Of Shape Function By Radial Point Interpolation Method For 1d Case

A. Mjidila[1], S. Jalal[2], L. Bousshine[3], Z. Elmaskaoui[4]
[1,2,3,4]*(Ain Chock Hassan II University, NHSEM, Oasis, Eljadida road, PB8118 Casablanca Laboratory of Technologies of Constructions and the Industrial Systems)*

Abstract: **-** In this paper, is presented a construction of Shape function by Radial Point Interpolation Method for 1D case. The original code 2D, developed by G.R. Lui and coworkers, was modified for 1D case. Treated case is a 1D support Domain built over six nodes. Three basic functions were used to construct the shape functions, Multi-Quadratic (MQ), exponential (Exp) and Thin Plate Spline (TPS) functions.

***Keywords* -** *Mesh-less method, Shape function, RPIM, Basis function.*

I. INTRODUCTION

The method of function interpolation/approximation based on arbitrary nodes is one of the most important issues in an MFree method. MFree methods may be classified according to the MFree interpolation/approximation methods used.

1. *Meshfree methods based on the moving least squares approximation*
2. *Meshfree methods based on the integral representation method for the function approximation*

These two methods can be seen in specialized literature.

3. *Meshfree methods based on the point interpolation method*

The point interpolation method (PIM) is an MFree interpolation technique that was used by GR Liu and his colleagues (GR Liu and Gu, 2001a) to construct shape functions using nodes distributed locally to formulate Mf'ree weak-form methods. Different from the NILS approximation, PIM uses interpolations to construct shape functions that possess Kronecker delta property. Two different types of PIM formulations using the polynomial basis (GR Liu and Gu, 2001c) and the radial function basis (REF) (Wang and GR Liu, 2000) have been developed.

A good method for creating MFree shape functions should satisfy some basic requirements.

- It should be sufficiently robust for reasonably arbitrarily distributed nodes (arbitrary nodal distribution).
- It should be numerically stable (stability).
- It should satisfy up to a certain order of consistency (consistency).
- It should be compactly supported, i.e., it should be regarded as zero outside a bounded region, the support domain.

The approximated unknown function using the shape function should be compatible" (compatibility) throughout the problem domain when a global weak-form is used, or should be compatible within the local quadrature domain when a local weak-form is used.

It is ideal if the shape function possesses the Kronecker delta function property (Delta function property), i.e. the shape function is unit at the node and zero at other nodes in the support domain.

- It should be computationally efficient (efficiency)

II. RADIAL POINT INTERPOLATION SHAPE FUNCTION

In seeking for an approximate solution to a problem governed by PDEs and boundary conditions, one first needs to approximate the unknown field function using trial (shape) functions, before any formulation procedure can be applied to establish the discretized system equations. This paragraph discusses various techniques for Mf'ree shape function constructions. These shape functions are locally supported, because only a set of field nodes in a small local domain are used in the construction and the shape function is not used or regarded as zero outside the local domain. Such a local domain is termed the support domain.

In the finite element method (FEM), the shape functions are created using interpolation techniques based on elements formed by a set of fixed nodes. This type of interpolation is termed stationary element based interpolation. In MFree methods, the problem domain is usually represented by field nodes that are, in general, arbitrarily distributed. The field variables at an arbitrary point in the problem domain are approximated using a group of field nodes in a local support domain. [2]

The point interpolation method (PIM) is one of the series representation methods for the function approximation, and is useful for creating MFree shape functions. In order to avoid the singularity problem in the polynomial PIM, the radial basis function (REF) is used to develop the radial point interpolation method (RPIM) shape functions for MFree weak-form methods (GR Liu and Gu, 2001c; Wang and Liu, 2000; 2002a,c). The RPIM interpolation augmented with polynomials can be written as:

$$u(x) = \sum_{i=1}^{n} R_i(x) a_i + \sum_{j=1}^{m} p_j(x) b_j = R^T(x) a + p^T(x) b \qquad \textbf{(1)}$$

where $R_i(x)$ is a radial basis function (REF), n is the number of REFs, pj(x) is monomial in the space coordinates xT=[x, y], and m is the number of polynomial basis functions. When m=0, pure REFs are used. Otherwise, the REF is augmented with m polynomial basis functions. Coefficients a, and b, are constants to be determined.

In the radial basis function $R_i(x)$, the variable is only the distance between the point of interest x and a node at x_i,

$$\begin{aligned} r &= \sqrt{(x-x_i)^2 + (y-y_i)^2} \quad \text{for 2D problems} \\ \text{and } r &= (x - x_i) \quad \text{for 1D problems} \end{aligned} \qquad \textbf{(2)}$$

There are a number of types of radial basis functions (REF), and the characteristics of RBFs have been widely investigated (Kansa, 1990; Sharan et al.,1997; Franke and Schaback, 1997; etc.). Four often used REFs, the multi-quadrics (MQ) function, the Gaussian (Exp) function, the thin plate spline (TPS) function and the Logarithmic radial basis function are listed in table 1.

Table 1 Typical radial basis functions

	Name	Expression	Shape Parametrs
1.	Multi-Quadratics (MQ)	$R_i(x) = (r_i^2 + (\alpha_c d_c)^2)^q$	$\alpha_c \geq 0, q$
2.	Gaussian (Exp)	$R_i(x) = \exp(-\alpha_c (\frac{r_i}{d_c})^2)$	α_c
3.	Thin Plate Spline	$R_i(x) = r_i^\eta$	η
4.	Logarithmic	$R_i(x) = r_i^\eta \log r_i$	η

In order to determine a_i and b_i in equation (1), a support domain is formed for the point of interest at x, and n field nodes are included in the support domain. Coefficients a_i and b_j in Equation (1) can be determined by enforcing Equation (1) to be satisfied at these n nodes surrounding the point of interest x. This leads to n linear equations, one for each node. The matrix form of these equations can be expressed as

$$U_s = R_0 a + P_m b \qquad \textbf{(3)}$$

where the vector of function values U_s is

$$U_s = \{u_1 \quad u_2 \quad \dots \quad u_n\}^T \qquad \textbf{(4)}$$

The moment matrix of TBFs is

$$R_0 = \begin{bmatrix} R_{1(r1)} & R_{2(r1)} & \dots & R_{n(r1)} \\ R_{1(r2)} & R_{2(r2)} & \dots & R_{1(r2)} \\ \dots & \dots & \dots & \dots \\ R_{1(rn)} & R_{2(rn)} & \dots & R_{n(rn)} \end{bmatrix} \qquad \textbf{(5)}$$

Tne polynomial moment matrix is

$$P_m^T = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_n \\ y_1 & y_2 & y_3 & \dots & y_n \\ . & . & . & \dots & \\ p_m(x_1) & p_m(x_2) & p_m(x_3) & & p_m(x_n) \end{bmatrix}_{(m \times n)} \qquad \textbf{(6)}$$

The vector of coefficients for RBFs is

$$a^T = \{a_1 \quad a_2 \quad \dots \quad a_n\}^T \tag{7}$$

The vector of coefficients for polynomial is

$$b^T = \{b_1 \quad b_2 \quad \dots \quad b_n\}^T \tag{8}$$

In equation (5) r_k in $R_i(r_k)$ is defined as

$$r_k = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \tag{9}$$

However, There are n+m variables in equation (4). The additional m equations can be added using the following m constraint conditions.

$$\sum_{i=1}^{n} p_j(x_i)a_i = P_m^T a = 0, \ \ \mathrm{j} = 1,2,\dots, \ \mathrm{m} \tag{10}$$

Combing Equations (3) et (10) yields the following set of equations in the matrix form

$$U_s = \begin{bmatrix} U_s \\ 0 \end{bmatrix} = \begin{bmatrix} R_0 & P_m \\ P_m^T & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \end{Bmatrix} = Ga_0 \tag{11}$$

Where

$$a_0^T = \{a_1 \quad \mathrm{a}_2 \quad \dots \quad \mathrm{a}_n \quad \mathrm{b}_1 \quad \mathrm{b}_2 \quad \dots \quad \mathrm{b}_m\} \tag{12}$$

$$U_s = \{u_1 \quad u_2 \quad \dots \quad u_n \quad 0 \quad 0 \quad \dots \quad 0\}^T \tag{13}$$

Because the matrix $\mathbf{R}_0$, is symmetric, the matrix **G** will also be symmetric. Solving Equation (11), we obtain

$$a_0 = \begin{Bmatrix} a \\ b \end{Bmatrix} = G^{-1}\tilde{U}_s \tag{14}$$

Equation (14) can be re-written as

$$u(x) = R^T(x)a + p^T(x)b = \{R^T(x) \quad p^T(x)\}\begin{Bmatrix} a \\ b \end{Bmatrix} \tag{15}$$

Using Equation (14) we can obtain

$$u(x) = R^T(x)a + p^T(x)b = \{R^T(x) \quad p^T(x)\}G^{-1}U_s = \tilde{\Phi}^T(x)\tilde{U}_s \tag{16}$$

Where the RPIM shape function can be expressed as

$$\begin{aligned} \tilde{\Phi}^T(x) &= \{R^T(x) \quad p^T(x)\}G^{-1} \\ &= \{\phi_1(x) \quad \phi_2(x) \quad \dots \quad \phi_n(x) \quad \phi_{n+1}(x) \quad \dots \quad \phi_{n+m}(x)\} \end{aligned} \tag{17}$$

Finally, the RPIM shape functions corresponding to the nodal displacements vector Φ(x) are obtained as

$$\tilde{\Phi}^T(x) = \{\phi_1(x) \quad \phi_2(x) \quad \dots \quad \phi_n(x)\} \tag{18}$$

Equation (16) can be re-written as

$$\mathrm{u(x)} = \Phi^T(x)U_s = \sum_{\mathrm{i}=1}^{\mathrm{n}} \phi_{\mathrm{i}} u_i \tag{19}$$

The derivatives of u(x) are easily obtained as

$$\mathrm{u}_{,l}(\mathrm{x}) = \Phi_{,l}^T(x)U_s \tag{20}$$

Where l denotes either the coordinate x or y. A comma designates a partial differentiation with respect to the indicated spatial coordinate that follows.

There are several advantages of using RBFs as a basis in constructing PIM shape functions that use local compact support domains.

- Using REFs can effectively solve the singularity problem of the polynomial PIM.
- RPIM shape functions are stable and hence flexible for arbitrary and irregular nodal distributions.
- RPIM shape functions can be easily created for three-dimensional domains, because the only variable is the distance r in a REF. For three-dimensional interpolation, we simply change the distance expression to

$$r = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \tag{21}$$

- RPIM shape functions are better suited than MLS functions for fluid dynamics problems.

However, RPIM also has some shortcomings, such as

- RPIM shape functions usually give less accurate solutions for solid problems compared to MLS and the polynomial PlM shape functions.
- Some shape parameters must be determined carefully, because they can affect the accuracy and the performance of the RPIM shape functions used in MFree methods.
- RPIM shape functions are usually computationally more expensive than the polynomial PIM because more nodes are required in the local support domain.

III. 1D-RPIM SHAPE FUNCTION CALCULUS

There a 2D-code developed by G.R. Lui and coworkers, was modified for 1D case to calculate the RPIM shape function and theirs first and second derivatives to respect of x.
The case study is a domain formed by six nodes as shown in fig. 1

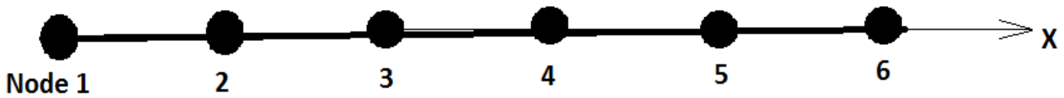

Figure 1: 1D domain formed by six nodes

The table 2 gives the spatial coordinate for each node. A set of six nodes of interest is chosen as is given in table 3.
In this study, the following coefficients are chosen : q=0.5, α_c=2, dc=1, η=2 and ambasis=0 (RPIM is pure).

Table 2, Field nodes and their coordinates

Node	1	2	3	4	5	6
Coordinate	0	1	2	3	4	5

Table 3, Field node of interest

Node	1	2	3	4	5	6
Coordinate	0	1	2	3	4	5

The figure 2 gives the flowchart of modified code.

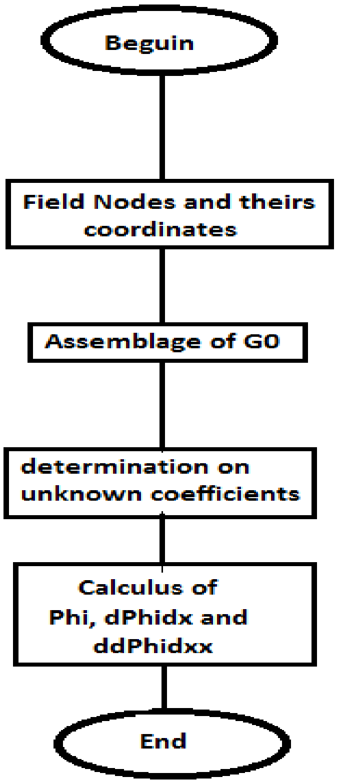

Figure 2, Flowchart of modified code

IV. THE RESULT AND DISCUSSION

The following figures (3), (10) and (17) give the result obtained by the modified code in cases:

- the Quadratic-RBF,
- the exponential-RBF,
- The TPS-RBF.

La Date: 20140724 L Heure: 102502.254 +0100

Node	X	Phi	dPhidx	dPhidxx
1	.000	1.000	-1.383	.243
2	1.000	.000	2.061	-1.003
3	2.000	.000	-1.038	1.262
4	3.000	.000	.498	-.643
5	4.000	.000	-.176	.167
6	5.000	.000	.016	.048
1	.000	.000	-.421	1.298
2	1.000	1.000	-.401	-2.758
3	2.000	.000	1.158	1.713
4	3.000	.000	-.474	-.350
5	4.000	.000	.181	.128
6	5.000	.000	-.035	-.042
1	.000	.000	.138	-.236
2	1.000	.000	-.751	1.836
3	2.000	1.000	-.085	-3.229
4	3.000	.000	.934	1.963
5	4.000	.000	-.307	-.419
6	5.000	.000	.066	.088
1	.000	.000	-.066	.088
2	1.000	.000	.307	-.419
3	2.000	.000	-.934	1.963
4	3.000	1.000	.085	-3.229
5	4.000	.000	.751	1.836
6	5.000	.000	-.138	-.236
1	.000	.000	.035	-.042
2	1.000	.000	-.181	.128
3	2.000	.000	.474	-.350
4	3.000	.000	-1.158	1.713
5	4.000	1.000	.401	-2.758
6	5.000	.000	.421	1.298
1	.000	.000	-.016	.048
2	1.000	.000	.176	.167
3	2.000	.000	-.498	-.643
4	3.000	.000	1.038	1.262
5	4.000	.000	-2.061	-1.003
6	5.000	1.000	1.383	.243

Figure 3, the result by MQ RBF

The following figures (4-9) show shape function Φ and theirs first and second derivatives to respect of x in each node, for MQ-RBF.

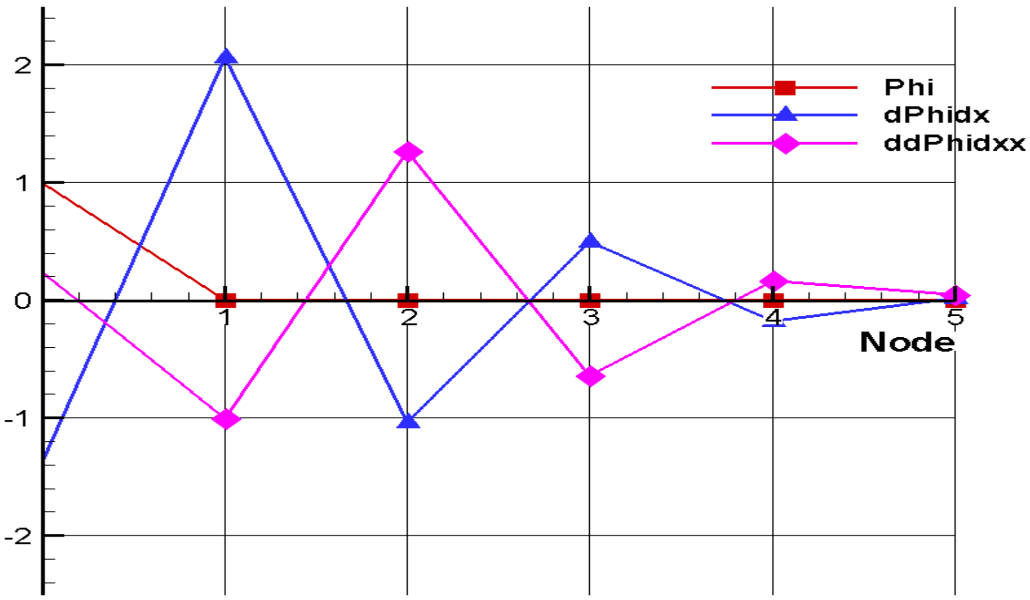

Figure 4, Φ function and and its first and second derivatives to respect of x for node 1 by MQ method

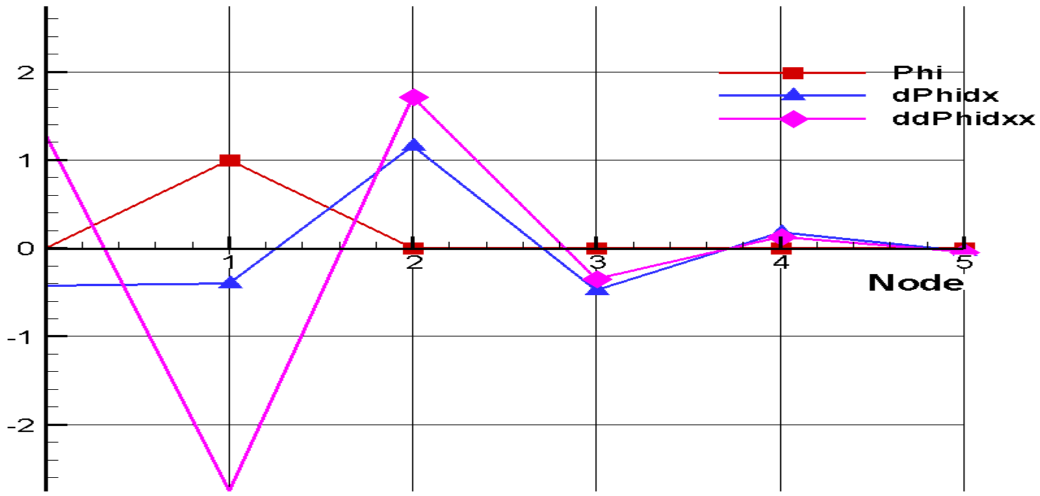

Figure 5, Φ function and and its first and second derivatives to respect of x for node 2 by MQ method

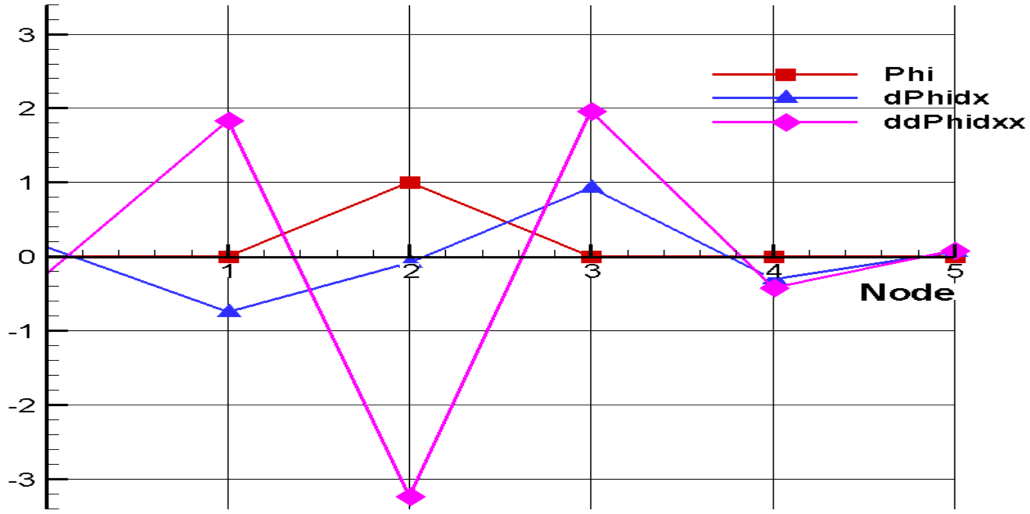

Figure 6, Φ function and and its first and second derivatives to respect of x for node 3 by MQ method

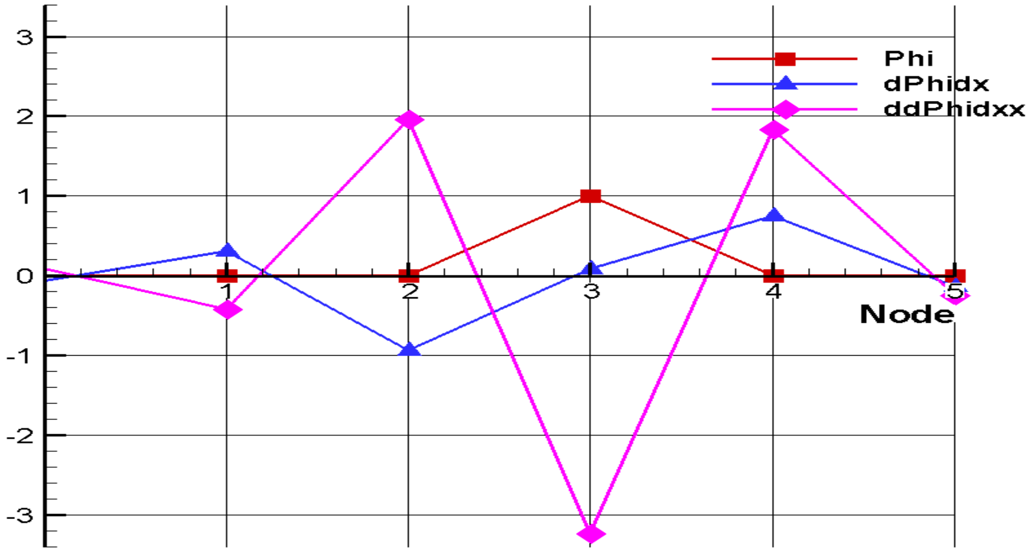

Figure 7, Φ function and and its first and second derivatives to respect of x for node 4 by MQ method

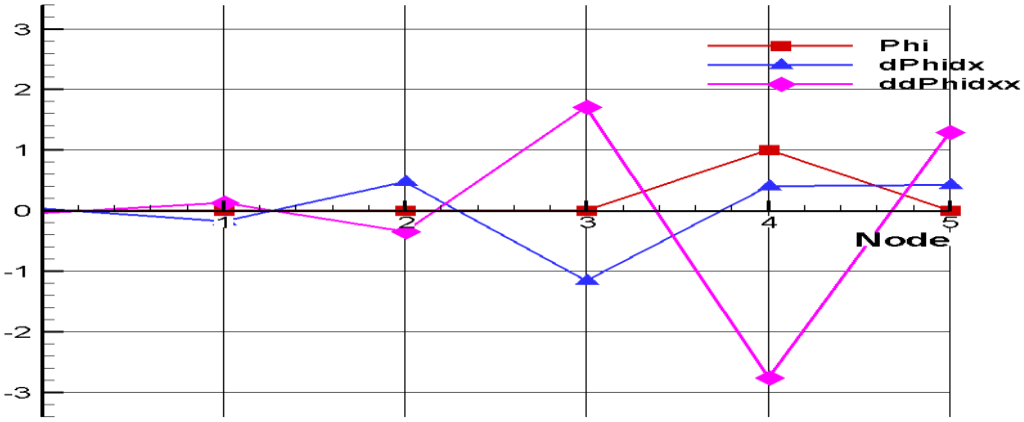

Figure 8, Φ function and and its first and second derivatives to respect of x for node 5 by MQ method

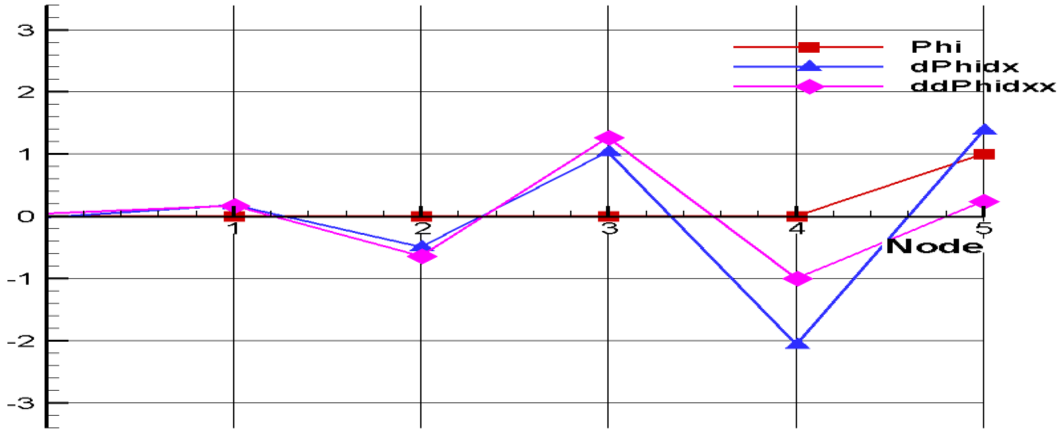

Figure 9, Φ function and and its first and second derivatives to respect of x for node 6 by MQ method

La Date: 20140729 L Heure: 011531.512 +0100

Node	X	Phi	dPhidx	dPhidxx
1	.000	1.000	-.076	-4.304
2	1.000	.000	.562	2.245
3	2.000	.000	-.075	-.288
4	3.000	.000	.010	.039
5	4.000	.000	-.001	-.005
6	5.000	.000	.000	.001
1	.000	.000	-.541	2.248
2	1.000	1.000	-.001	-4.614
3	2.000	.000	.552	2.287
4	3.000	.000	-.073	-.293
5	4.000	.000	.010	.040
6	5.000	.000	-.001	-.005
1	.000	.000	.072	-.288
2	1.000	.000	-.551	2.287
3	2.000	1.000	.000	-4.619
4	3.000	.000	.551	2.288
5	4.000	.000	-.073	-.293
6	5.000	.000	.010	.039
1	.000	.000	-.010	.039
2	1.000	.000	.073	-.293
3	2.000	.000	-.551	2.288
4	3.000	1.000	.000	-4.619
5	4.000	.000	.551	2.287
6	5.000	.000	-.072	-.288
1	.000	.000	.001	-.005
2	1.000	.000	-.010	.040
3	2.000	.000	.073	-.293
4	3.000	.000	-.552	2.287
5	4.000	1.000	.001	-4.614
6	5.000	.000	.541	2.248
1	.000	.000	.000	.001
2	1.000	.000	.001	-.005
3	2.000	.000	-.010	.039
4	3.000	.000	.075	-.288
5	4.000	.000	-.562	2.245
6	5.000	1.000	.076	-4.304

Figure 10, the result by Exp RBF

The following figures (10-15) show shape function Φ and theirs first and second derivatives to respect of x in each node, for EXP-RBF.

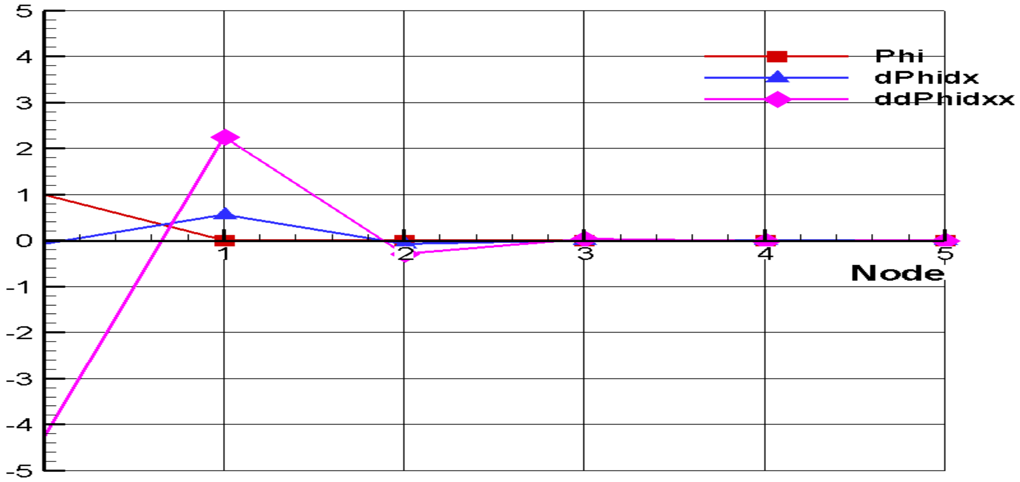

Figure 11, Phi, Φ function and and its first and second derivatives to respect of x for node 1 by EXP method

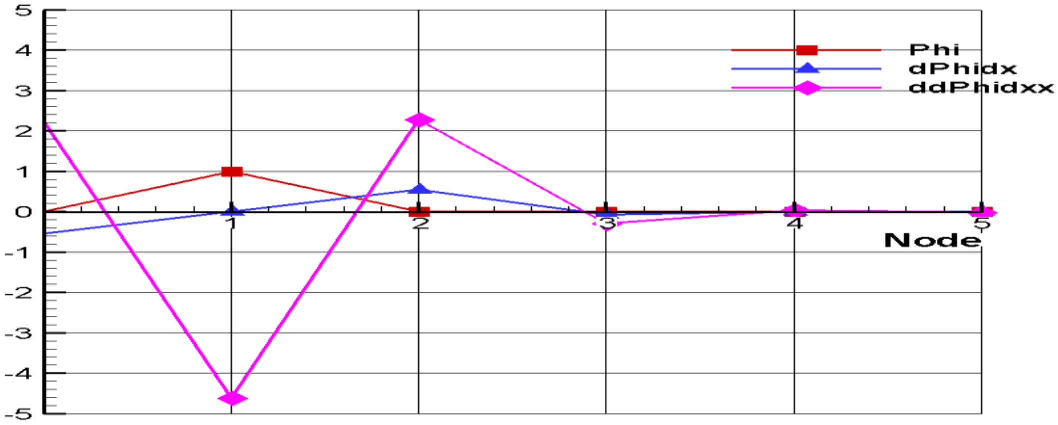

Figure 12,Φ function and and its first and second derivatives to respect of x for node 2 by EXP method

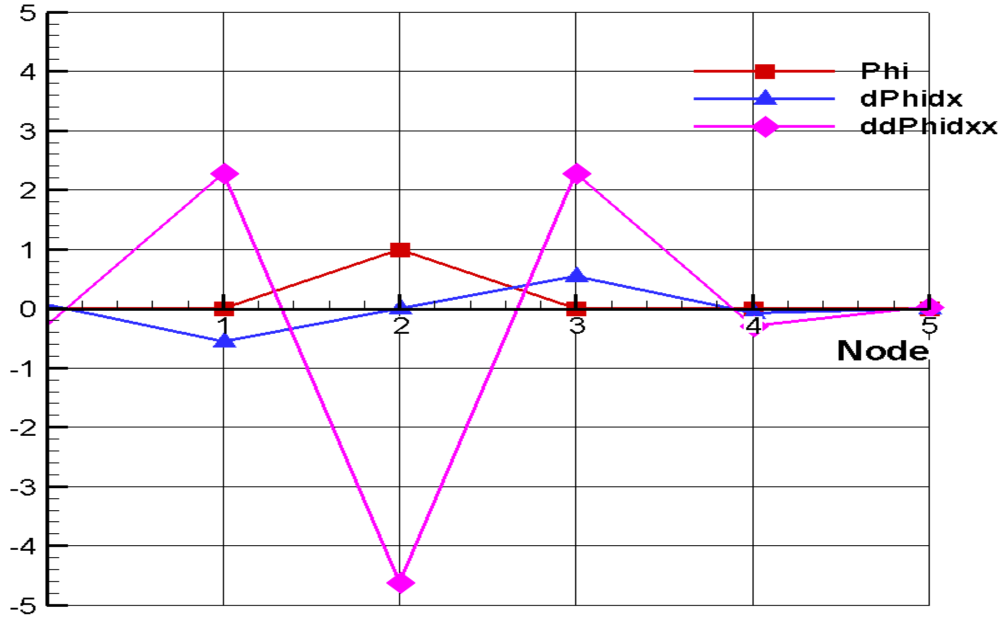

Figure 13, Φ function and and its first and second derivatives to respect of x for node 3 by EXP method

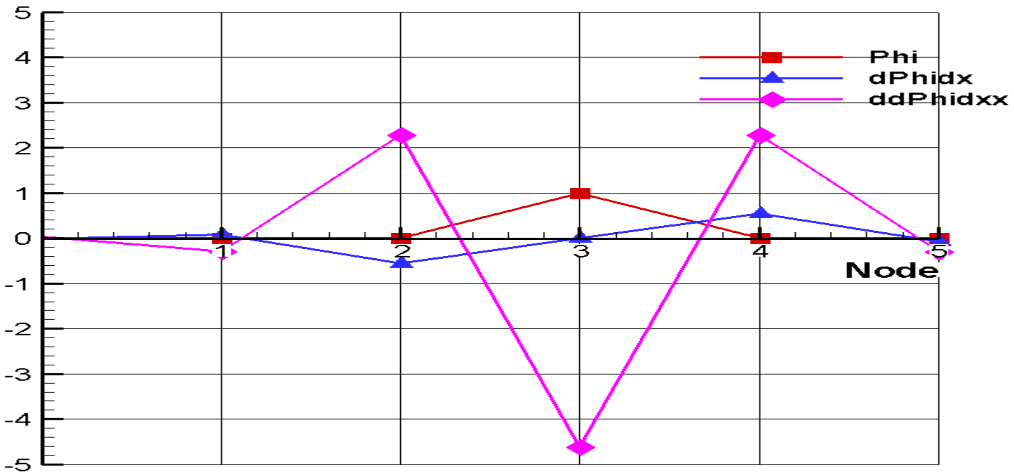

Figure 14, Φ function and and its first and second derivatives to respect of x for node 4 by EXP method

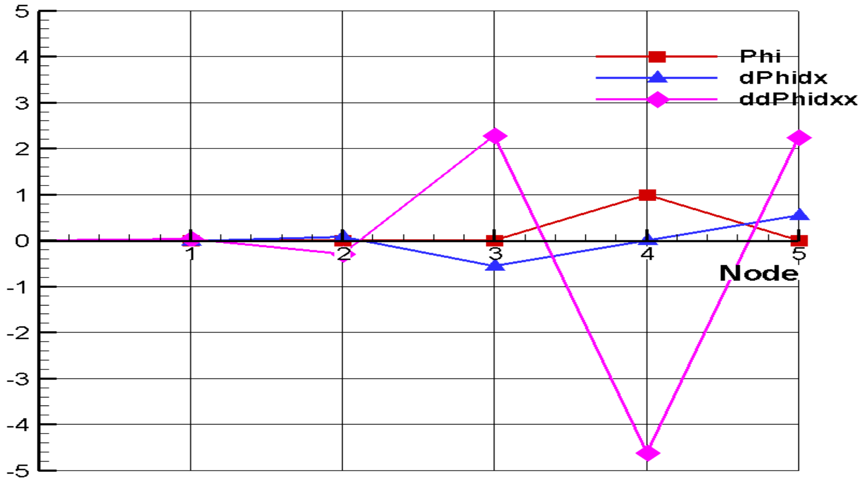

Figure 15, Φ function and and its first and second derivatives to respect of x for node 5 by EXP method

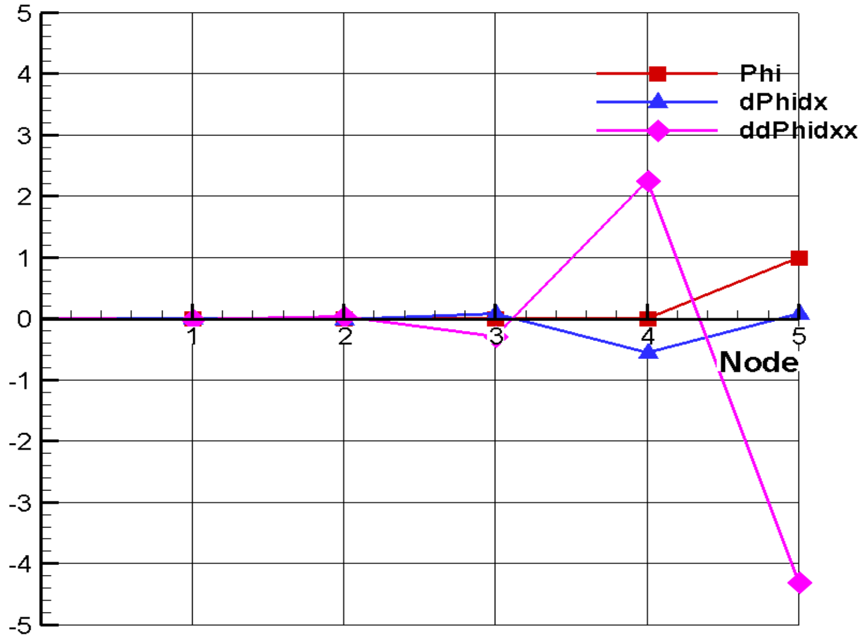

Figure 16, Φ function and and its first and second derivatives to respect of x for node 6 by EXP method

La Date: 20140814 L Heure: 052648.853 +0100

Node	X	Phi	dPhidx	dPhidxx
1	.000	1.000	-1.308	1.867
2	1.000	.000	.125	-3.417
3	2.000	.000	4.750	-1.000
4	3.000	.000	-6.417	5.833
5	4.000	.000	3.625	-4.333
6	5.000	.000	-.775	1.050
1	.000	.000	-.125	.792
2	1.000	1.000	-1.458	-1.042
3	2.000	.000	2.750	-.750
4	3.000	.000	-1.750	1.583
5	4.000	.000	.708	-.708
6	5.000	.000	-.125	.125
1	.000	.000	.046	.042
2	1.000	.000	-.479	.708
3	2.000	1.000	-.375	-1.250
4	3.000	.000	1.042	.083
5	4.000	.000	-.271	.542
6	5.000	.000	.037	-.125
1	.000	.000	.017	-.108
2	1.000	.000	.000	.458
3	2.000	.000	-.500	.250
4	3.000	1.000	-.167	-1.417
5	4.000	.000	.750	.792
6	5.000	.000	-.100	.025
1	.000	.000	.063	.142
2	1.000	.000	-.396	-.792
3	2.000	.000	1.125	1.750
4	3.000	.000	-2.125	-.917
5	4.000	1.000	1.146	-.958
6	5.000	.000	.188	.775
1	.000	.000	.808	1.467
2	1.000	.000	-3.792	-6.417
3	2.000	.000	6.750	10.000
4	3.000	.000	-5.083	-5.167
5	4.000	.000	.042	-1.333
6	5.000	1.000	1.275	1.450

Figure 17, result by TPS-RBF

The following figures (18-23) show shape function Φ and theirs first and second derivatives to respect of x in each node, for TPS-RBF.

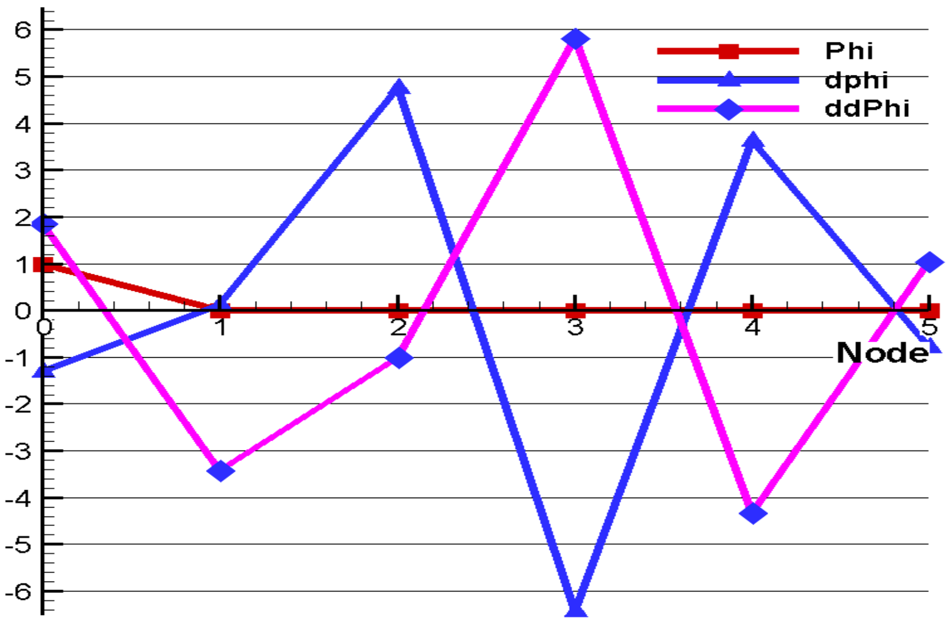

Figure 18,Φ function and and its first and second derivatives to respect of x for node 1 by TPS method

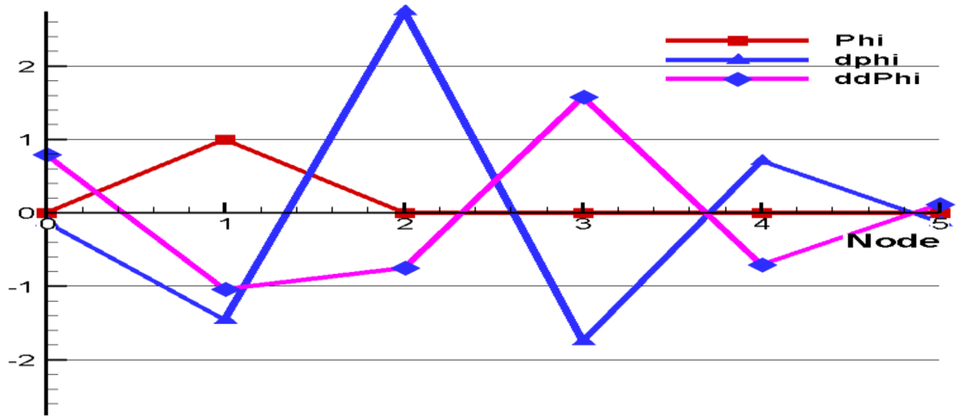

Figure 19,Φ function and and its first and second derivatives to respect of x for node 2 by TPS method

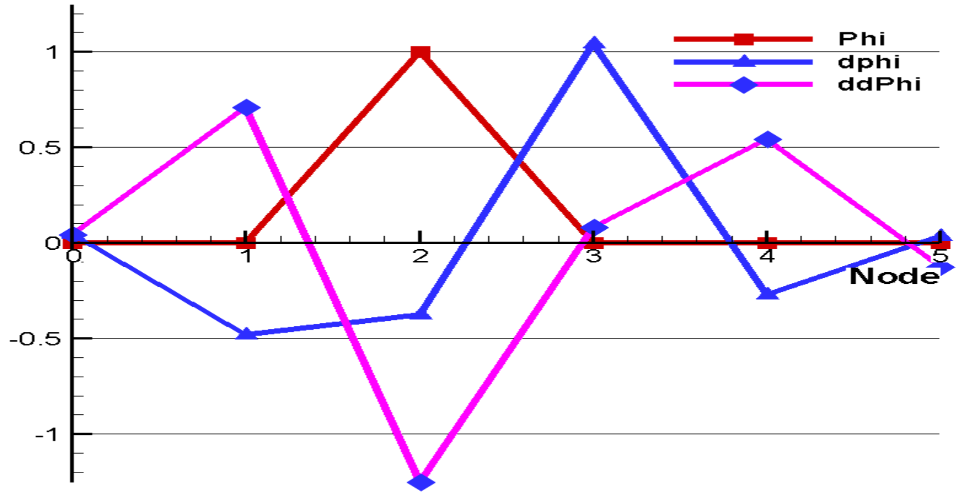

Figure 20,Φ function and and its first and second derivatives to respect of x for node 3 by TPS method

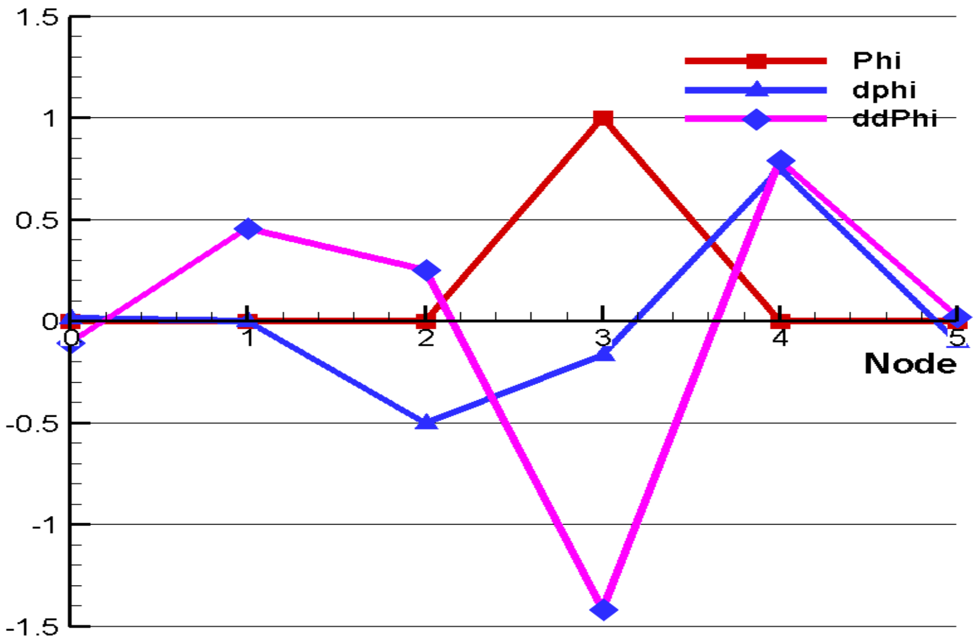

Figure 21,Φ function and and its first and second derivatives to respect of x for node 4 by TPS method

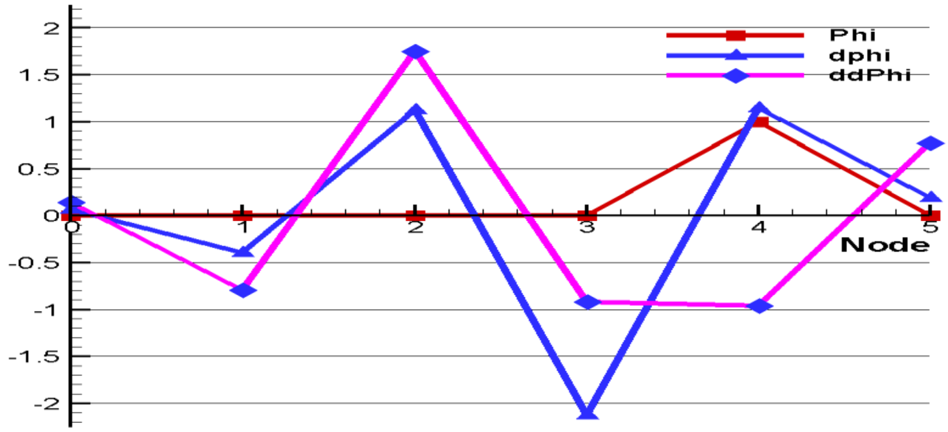

Figure 22,Φ function and and its first and second derivatives to respect of x for node 5 by TPS method

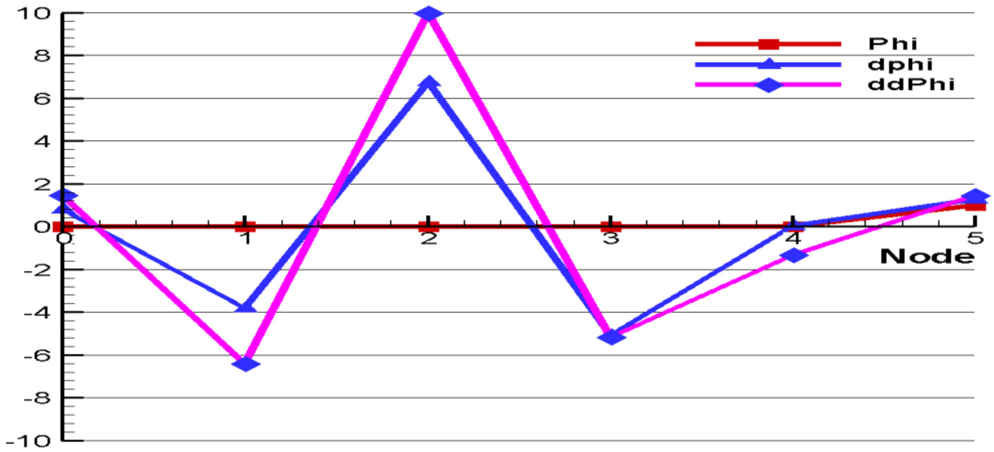

Figure 23,Φ function and and its first and second derivatives to respect of x for node 6 by TPS method

If the points of interest are different from nodes, for example the field (0.5, 1.5, 2.5, 3.5, 4.5, 5.5) is chosen as a set of interest points. In this case the Φ function and its derivatives for the interest point x_v=2.5 are shown in figures 23 to 25 by the three different methods.

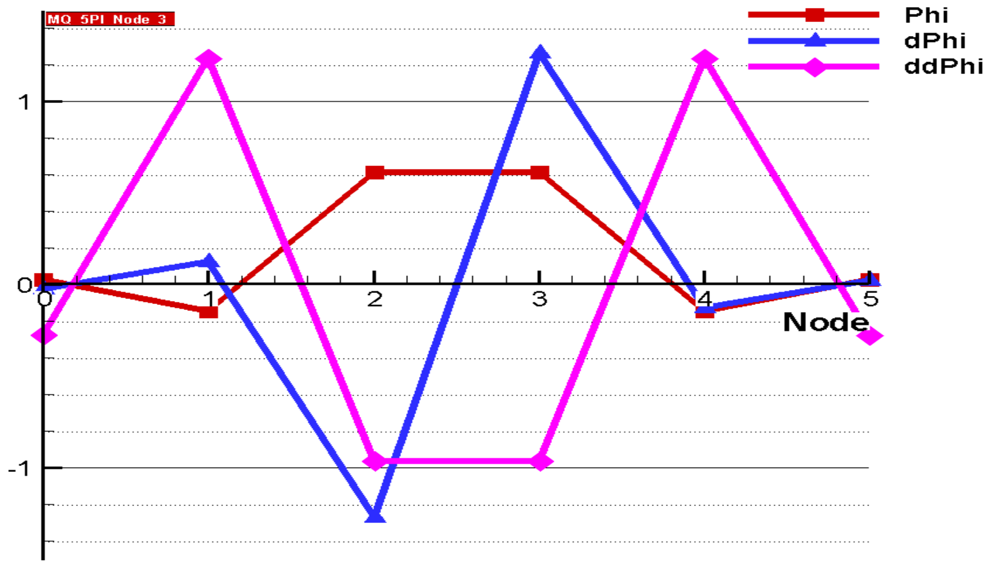

Figure 24, Φ function and and its first and second derivatives for node x_v =2.5 by MQ method.

REFERENCES

Books:

[1] Rong Liu, *Meshfree methods: moving beyond the finite element method* (CRC Press LLC, 2003).
[2] Rong Liu and Y.T. Gu, An Introduction to Meshfree Methods and their programming, (2006 Springer)
[3] Youping Chen, James D. Lee, and Azim Eskandarian, Meshless Methods in Solid Mechanics(2006 Springer Science , Inc)