Solving partial differential equations using Finite Point Method

M.Bitaraf [(1)] and S.Mohammadi [(2)]

Abstract: *At the outset the basis of the finite point method is described and then this method is adopted for solving some partial differential equations. Eventually, the impact of adopted weight functions in MLS approximation on the accuracy is considered and a relation between the number of nodes in support domain and parameter c in exponential weight function is emerged.*

1 Introduction

Designing advanced engineering systems requires the use of computer aided design tools. In such tools computational simulation techniques are often used to model and investigate physical phenomena in an engineering system. The simulation requires solving the complex differential or partial differential equations that govern these phenomena.Traditionally, such complex partial differential equations are solved using numerical methods such as the finite element method (FEM)[1]. In this method the spatial domain, is often discretized into meshes. But mesh generation, especially 3D mesh generation, remains one of the challenges.
Mesh free techniques have become quite popular in computational mechanics. A family of mesh free methods is based on smooth particle hydrodynamic procedures[2]. A second class of mesh free methods is derived from generalized finite difference (GFD) techniques. Among a third class of mesh free techniques we find the so called diffuse element (DE) method, the element free Galerking (EFG) method[3], the reproducing kernel particle (RKP) method, the meshless local Petrov-Galerkin (MLPG) method[4] and the method of finite spheres.
The finite point method (FPM)[5,6] is a truly meshless procedure. The approximation around each point is obtained by using standard moving least square techniques similarly as in DE and EFG methods. The discrete system of equations is obtained by sampling the governing equations at each point as in GFD methods.
The advantages of FPM compared with standard FEM is to avoid the necessity of mesh generation and compared with classical FDM is the facility to handle the boundary conditions and nonstructural distribution of points.

2 Finit Point Method

2.1 Interpolation in Finite Point Method

Moving least square method (MLS)[7] is used for interpolation in FPM. Let $u(x)$ be the function of the field variable defined in domain Ω. The approximation of $u(x)$ at point x is denoted $u^h(x)$. MLS approximation first writes the field function in the form:

(1) Graduate Student, Civil Engineering Dept., University of Tehran, Tehran, Iran, (mrybitaraf@ut.ac.ir).
(2) Associate Professor, Civil Engineering Dept., University of Tehran, Tehran, Iran, (smoham@ut.ac.ir)

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

Where m is the number of terms of monomials (polynomial basis) , and $\mathbf{a}(x)$ is a vector of coefficients and $\mathbf{p}(x)$ is a vector of basis function that consist most often of monomials of the lowest orders to ensure minimum completeness. In 1D space, a complete polynomial basis of order m is given by

$$\mathbf{p}^T(x)=\{p_0(x),p_1(x),\cdots,p_m(x)\}=\{1,x,x^2,\cdots,x^m\} \qquad (2)$$

And in 2D space,

$$\mathbf{p}^T(\mathbf{x})=\mathbf{p}^T(x,y)=\{1,x,y,xy,x^2,y^2,\cdots,x^m,y^m\} \qquad (3)$$

A function of weighted residual is constructed using the approximated values of the field function and the nodal parameters, $u_I=u(x_I)$

$$J=\sum_{1}^{n}W(x-x_I)\left[u^h(x,x_I)-u(x_I)\right]^2=\sum_{1}^{n}W(x-x_I)\left[\mathbf{p}^T(x_I)\mathbf{a}(x)-u_I\right]^2 \qquad (4)$$

Where $W(x-x_I)$ is a weight function, and u_I is the nodal parameter of the field variable at node I .The weight function plays two important roles. The first is to provide weightings for the residuals at different nodes in support domain. The second roles is to ensure that nodes leave or enter the support domain in a gradual (smooth) manner when x moves. it make sure that the MLS satisfy the compatibility condition. $\mathbf{a}(x)$ is chosen to minimize the weighted residual. The minimization condition requires

$$\frac{\partial J}{\partial \mathbf{a}}=0 \qquad (5)$$

Which results in the following linear equation system :

$$\mathbf{a}(x)=\mathbf{A}(\mathbf{x})^{-1}\mathbf{B}(x)\mathbf{U_s} \qquad (6)$$

$$\mathbf{A}(x)=\sum_{1}^{n}W(x-x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I) \qquad (7)$$

$$\begin{aligned}\mathbf{B}(x)&=[\mathbf{B}_1,\mathbf{B}_2,...,\mathbf{B}_n]\\ \mathbf{B}_I&=W(x-x_I)\mathbf{p}(x_I)\end{aligned} \qquad (8)$$

And $\mathbf{U_s}$ is the vector that collects the nodal parameters of the field variables for all the nodes in the support domain. Substituting the equation (6) into equation (1) leads to

$$u^h(x)=\mathbf{p}^T(x)\mathbf{A}^{-1}\mathbf{B}\mathbf{U_s} \qquad (9)$$

To determine the spatial derivatives of the function of the field variable, which are required for deriving the discretized system equations, it is necessary to obtain the derivatives of the MLS shape function.The partial derivative of MLS would be

$$\frac{\partial u(x)}{\partial x}=\left[\frac{\partial \mathbf{p}^T(x)}{\partial x}\mathbf{A}^{-1}\mathbf{B}+\mathbf{p}^T(x)\mathbf{A}^{-1}\left[\frac{\partial \mathbf{A}}{\partial x}\mathbf{A}^{-1}\mathbf{B}+\frac{\partial \mathbf{B}}{\partial x}\right]\right]\mathbf{U}_S \qquad (10)$$

But most often the first term of derivative provides us with adequate accuracy.

2.2 discretization of governing equations

Let us assume a problem governed by the following set of differential equations

$$\begin{aligned}&A(u_j)=0 \quad in\ \Omega\\ &u_j-\bar{u}_j=0 \quad on\ \Gamma_u\\ &B(u_j)=0 \quad on\ \Gamma_t\end{aligned} \qquad (11)$$

The discretized system of equation in the FPM is found by substituting the approximation (12) into equation (15) and collocating the differential equation at each point in the analysis domain. This gives

$$\lfloor A(u_j)\rfloor_p = 0 \quad p = 1,2,...,N_r$$
$$\left[u_j\right]_s - \bar{u}_j = 0 \quad s = 1,2,...,N_u \qquad (12)$$
$$\left[B(u_j)\right]_r = 0 \quad r = 1,2,...,N_t$$

In the above N_u and N_t are the number of points located on the boundaries Γ_u and Γ_t, and N_r is the rest of the point in Ω not belonging to any of the boundaries Γ_u and Γ_t. Equation (16) lead to a system of algebraic equations of the form

$$\mathbf{KU_S} = \mathbf{f} \qquad (13)$$

3 Examples

In this section the efficiency of FPM method is analyzed using this method for solving 1D and 2D partial differential equations. m=2 is chosen for the basis function and the exponential function is used as the weight function:

$$W(x - x_I) = \begin{cases} e^{(-r/cr_m)^2} & r <= r_m \\ 0 & r > r_m \end{cases}$$
$$r = |\mathbf{x} - \mathbf{x}_I| \qquad (14)$$

r_m is the radius of support domain

These problems are solved considering different values of c parameter and radius of support domain and finally the error values are computed. As a consequence, the optimum value of c with respect to r_m would be gained.

3.1 Solving 1D equations using FPM

The 1D partial differential equations considered here, are as follows:

1) $-0.01\dfrac{\partial^2 u}{\partial x^2} + u + 1 = 0 \qquad 0 < x < 1$

$u(0) = u(1) = 0$

2) $\dfrac{\partial^2 u}{\partial x^2} = -\sin(x) \qquad 0 < x < 1$

$\dfrac{\partial u}{\partial x} = \cos(1) \qquad x = 1$

$u = 0 \qquad x = 0$

3) $\dfrac{\partial^2 u}{\partial x^2} = (-100 + (-100x + 250)^2)\, e^{(-50(-x+3)(x-2))} \qquad 0 < x < 5$

$u(0) = u(5) = e^{(-300)}$

At first, 9 nodes are used for the domain (0<x<1) and the equation (1) is solved with different values for radius of support domain (r_m) and c. As the next step, the error values would be computed as shown in Table 1, and it would be found that the greater r_m , the smaller c should be used to achieve better result. The same result would be gained when 11 nodes for domain (0<x<1) are used as shown in table 2. Using smaller c would lead to more accurate result for n=11 compared with n=9 and while the values r_m are the same.

It shows that if the number of nodes in support domain increases, using smaller c would guarantee more accuracy. Enough attention should be paid not to have ill condition or singularity in our solution when we use small c.

Quartic spline weight function is used for MLS approximation instead of exponential weight function and it is found that the former would show more inaccurate results than the latter as shown in Table 3. The obtained results for $n=9, r_m=.25, c=0.5$ have been depicted in Fig.1.

The same results would be achieved for equation (2), as shown in Table 4 and Fig 2.

As an instance of concentration problems like concentrated force, the equation (3) has been chosen which, could simulate these phenomena. FPM again exhibits satisfactory results for solving this equation and the numerical results have been shown in Fig.3.

3.2 Solving 2D equation using FPM

For solving 2D equation, the regular distribution of nodes has been chosen as shown in Fig.4. The 2D equation is:

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0 \qquad 0<x<2 \quad 0<y<2$$

$$u(0,y)=\sin(y) \quad @x=0, u(2,y)=e^2 * \sin(y) \quad @x=2$$

$$u(x,0)=0 \qquad @y=0, \frac{\partial u}{\partial x}=\cos(2) * e^x \qquad @y=2$$

The results are shown in Table 5, Fig.5 and Fig.6. It could be seen that the outcomes are the same as what were gained in 1D. It means that the more number of nodes in support domain, the smaller c is required to achieve better performance.

c	rm=.25	rm=.375	rm=.5
0.2	2.02	1.91	1.71
0.3	1.91	1.56	1.24
0.4	1.68	1.25	2.48
0.5	1.28	2.55	7.24
0.6	1.96	4.72	10.43
0.7	4.48	14.48	8.15

Table 1: *The error percentage of Eq.1 approximation* $(n=9)$

rm=.26	rm=.375	rm=.5
1.7607	3.9	7.2

Table 3: *The error percentage of Eq.1 app. with Quartic Spline weight function*

c	rm=.25	rm=.375	rm=.5
0.2	0.82	0.82	0.91
0.3	0.82	0.98	1.74
0.4	0.92	1.64	4.07
0.5	1.05	2.71	6.83
0.6	1.40	1.52	3.40
0.7	1.21	2.45	1.92

Table 2: *The error percentage of Eq.1 approximation* $(n=11)$

c	rm=.2	rm=.3	rm=.4
0.2	0.31	0.10	0.07
0.3	0.10	0.03	0.18
0.4	0.09	0.19	0.52
0.5	0.07	0.95	0.59
0.6	0.08	6.32	0.63
0.7	0.17	2.42	1.67

Table 4: *The error percentage of Eq.2 approximation*

c	rm=.8	rm=1.2
0.2	1.74	1.45
0.3	1.49	1.19
0.4	1.08	3.13
0.5	1.82	5.55

Table 5: *The error percentage of 2D Eq. approximation*

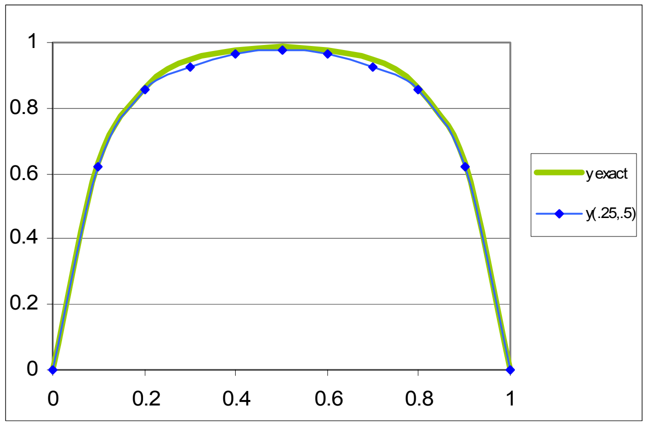


Figure 1: *App. of Eq.1* $(n = 9, r_m = .25, c = 0.5)$

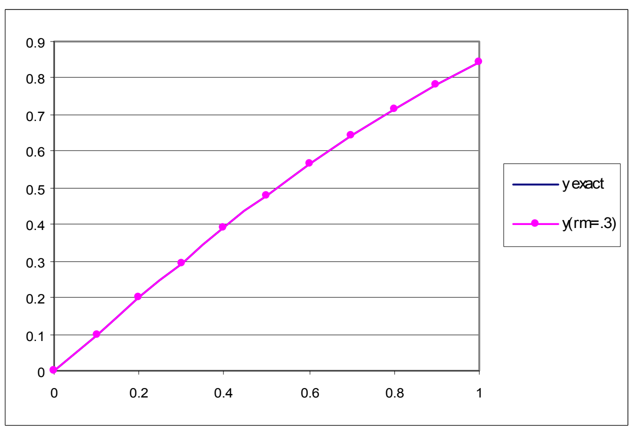


Figure 2: *App. of Eq.2* $(n = 9, r_m = .3, c = 0.3)$

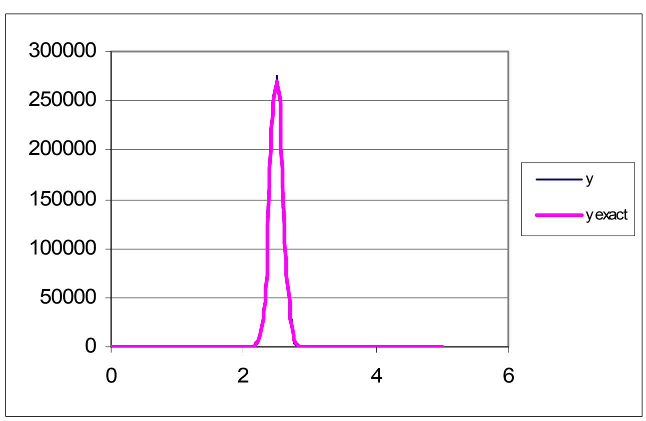


Figure 3: *App. of Eq.3*

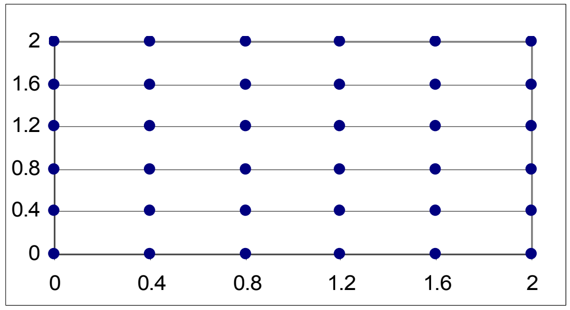
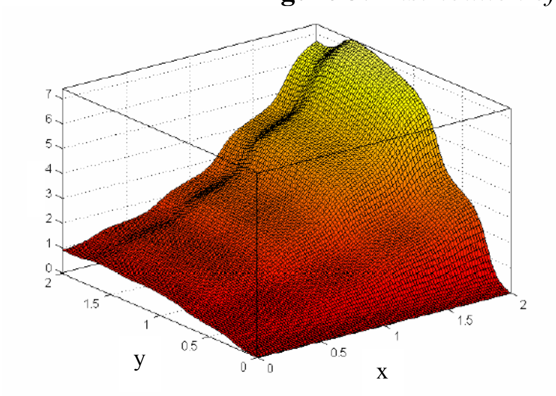


Figure 5: *Distribution of nodes in 2D problem*



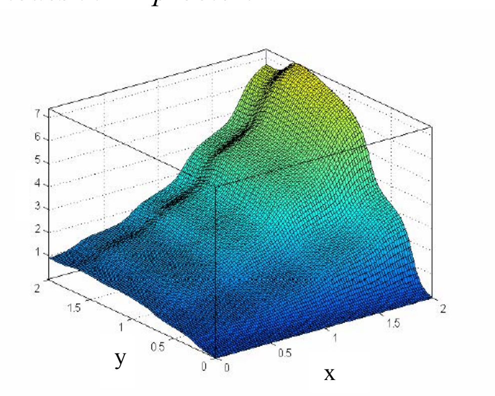


Figure 6: *Exact surface of 2D equation* **Figure 6:** *FPM app. surface of 2D equation*

4 Conclusions

A Finite Point Method has been presented for the simulation of partial differential equations. A Moving Least Square interpolation scheme has been used to derive shape function. Exponential and Quartic spline weight functions are used in MLS. The effect of changing the radius of support domain and weight function on accuracy was considered and a relation between the number of nodes in support domain and parameter c in exponential weight function was emerged.

References

[1] O.C. Zienkiewicz and R.L. Taylor. The finite emlement metoh. *4th Edition, Vol 1, McGraw Hill, (1989)*

[2] J.J Monaghan. Smoothed partivle hydrodynamics: Some recent improvement and applications. *Annu. Rev. Astone. Physics, 30, 543, (1992)*

[3] T. Belytschko, Y. LU and L. GU. Element free Galerkin methods. *Int. J. Num. Meth. Engng.,Vol 37,, 229-56, (1994)*

[4] Atlura SN, Zhu T. A new meshless local Petrov-Galerkin(MLPG) approach in computational mechanics. *Comput Mech,Vol 22, 117-27, (1998)*

[5] E. Onate, F. Perazzo, J. Miquel. A finite Point Method for elasticity problems. *Computer and Sructue, Vol 70, 2151-263, (2000)*

[6] E. Onate, S. Idelson, O.C. Zienkiewicz and R.L. Taylor. A finite Point Method in computational mechanics. Application to convective transport and fluid flow. *Int. J. Num. Meth. Engng., Vol 39, 3839-3866, (1996)*

[7] G. R. Liu, Mesh Free Methods. *CRC, (2003)*