ISSN 0974 - 9373

Vol. 17 No.2 (2013) **Journal of International Academy of Physical Sciences** pp. 123-132

Solving an Unsteady State Flow Problem by Meshless Local Petrov-Galerkin Method (Mlpg)

Amorn Koomsubsiri and Dusadee Sukawat
Department of Mathematics
Kingmongkut's University of Technology
Thonburi Bangkok, 10400, Thailand
Email: amorn@tni.ac.th, dusadee.suk@kmutt.ac.th

Suwon Tangmanee
Centre of Excellence in Mathematics (CEM)
Faculty of Science, Mahidol University, Bangkok, 10400, Thailand
Email: koomsubsiri@gmail.com

(Received December 2, 2012)

Abstract: In this paper, a meshless method called "Meshless Local Petrov-Galerkin (MLPG)" is applied to a simple atmospheric model. The problem domain is represented by a set of arbitrarily distributed nodes. The weight residual in MLPG method is confined to a very small local sub-domain of a node. The numerical integrations are carried out over a local quadrature domain defined for the node, which can also be the local domain where the test (weight) function is defined. The moving least-square (MLS) approximation is employed for constructing shape functions. Results obtained from the MLPG method are compared with the corresponding exact solutions. Effects of boundary condition are also investigated.
Keywords: Meshless Local Petrov-Galerkin, Advection equation

1. Introduction

Numerical method is important for the successful simulation of physical processes as the underlying partial differential equation usually has no analytic solution and has to be approximated .by Conventional numerical methods need a priori definition of the connectivity of the nodes, i.e., they rely on a mesh. Finite Element Method (FEM), Finite Difference Method (FDM) and Finite Volume Method (FVM) may be the most well-known members of these thoroughly developed mesh-based methods. The large deformations in highly nonlinear problems that can deteriorate the accuracy because of mesh or element distortion may cause severe loss of accuracy or even complete failure of computations. The mesh-based methods are

unsuitable for finding solutions of problems with changing domain shape. A new class of numerical methods has been developed which approximates partial differential equations based on only a set of nodes without the need for an additional mesh. This is called mesh free methods (MMs). Mesh free method is different from FEM because a mesh of element is not used in mesh free method, the field variable u at any point $\mathbf{x} = (x, y, z)$ within the problem domain is interpolated using the displacements at its nodes within the support domain of the point at x, i.e.

$$u(\bar{x}) = \sum_{i=1}^{n} \varphi_i(\bar{x}) u_i = \Phi(\bar{x}) U_s \,, \tag{1}$$

for $\bar{x} = (x_1, x_2, x_3, ..., x_n)$, where n is the number of nodes include in a "small local domain" of the point at **x**. The local domain means the interpolation area which is represented by point **x**, u_i is the nodal field variable at the i th node in the small local domain, U_s is the vector that collects all the field variables at these nodes, and $\varphi_i(\bar{x})$ is the shape function of the i th node determined using the nodes that are included in the small domain of **x**. This small local domain is called the support domain of **x**. A support domain of a point **x** determines the number of nodes to be used to support or approximate the function value at **x**.

2. Materials and Methods

2.1 Moving Least-Square (MLS)

In the MLPG method, the problem domain is represented by a set of arbitrarily distributed nodes. The weighted residual method is used to create the discrete system equation. The weighted residual method is in integral form, and a background mesh of cells is still required for the integration. However, the weight residual in MLPG method is confined to a very small local sub-domain of a node. This means that the weak form is satisfied at each node in the problem domain in a local integral sense. Therefore, the weak form is integrated over a "local quadrature domain" that is independent of other domains of other nodes. The moving least-square (MLS) is the basis for many meshless methods because it is generally considered to be one of the best schemes to interpolate data with a reasonable accuracy. MLS approximation is based on three components: a weight function of compact support associated with each node, polynomial basis functions and a set of coefficients that depend on the position x of the

point. The approximation $u^h(\mathbf{x})$ of a function of the field variable $u(\mathbf{x})$ at any point $\mathbf{x}$ in the domain (Ω) of computation is expressed by[2]

$$u^h(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) a_j(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}), \tag{2}$$

where m is the number of terms of monomials (polynomial basis), and a vector of coefficients $\mathbf{a}(\mathbf{x})$ is determined using the function values at a set of nodes that are included in the support domain of $\mathbf{x}$.

$\mathbf{p}(\mathbf{x})$ is a vector of basis function that consists 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

$$\mathbf{p}^{\mathrm{T}}(x) = \{p_0(x) \quad p_1(x) \quad \ldots \quad p_m(x)\} = \{1, x, x^2, \ldots, x^m\}. \tag{2.1}$$

Note that the coefficient vector $\mathbf{a}(\mathbf{x})$ in Eq. (2) is a function of $\mathbf{x}$. A functional of weighted residual is constructed using the approximated values of the field function and the nodal parameters, $u_i = u(\mathbf{x}_i)$

$$J = \sum_{i=1}^{n} \hat{W}(\mathbf{x}-\mathbf{x}_i)\left[\mathbf{p}^{\mathrm{T}}(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - u_i\right]^2, \tag{2.2}$$

where n is the number of nodes in the support domain of $\mathbf{x}$ for which the weight function $\hat{W}(\mathbf{x}-\mathbf{x}_i) \neq 0$. In MLS approximation, $\mathbf{a}(\mathbf{x})$ is chosen to minimize the weighted residual. The minimization condition requires

$$\frac{\partial J}{\partial \mathbf{a}} = 0, \tag{2.3}$$

which leads to the following set of linear relations:

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{U}_s \tag{2.4}$$

$\mathbf{A}(\mathbf{x})$ is called the weighted moment matrix defined by

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^{n} \hat{W}_i(\mathbf{x})\mathbf{p}(\mathbf{x}_i)\mathbf{p}^{\mathrm{T}}(\mathbf{x}_i), \tag{2.5}$$

where $\hat{W}_i(\mathbf{x}) = \hat{W}(\mathbf{x}-\mathbf{x}_i)$

The matrix $\mathbf{B}$ in eq. (2.4) has the form of

(2.6) $$\mathbf{B}(\mathbf{x}) = [\, \hat{W}_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1) \quad \hat{W}_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2) \quad \ldots \quad \hat{W}_n(\mathbf{x})\mathbf{p}(\mathbf{x}_n)\,]$$

and $\mathbf{U}_s$ is the vector that collects the nodal parameters of the field variables for all the nodes in the support domain

(2.7) $$\mathbf{U}_s = \{u_1 \quad u_2 \quad \ldots \quad u_n\}^{\mathrm{T}}.$$

Solving Eq. (2.4) for **a**(**x**),

(2.8) $$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{U}_s$$

Substituting the above equation back into Eq. (2) leads to

(2.9) $$u^h(\mathbf{x}) = \sum_{i=1}^{n} \phi_i(\mathbf{x}) u_i = \mathbf{\Phi}^{\mathrm{T}}(\mathbf{x})\mathbf{U}_s$$

where $\mathbf{\Phi}(\mathbf{x})$ is the vector of MLS shape functions corresponding *n* nodes in the support domain of the point **x**. The shape function $\phi_i(\mathbf{x})$ for the *i*th node is defined by

(2.10) $$\phi_i(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})\left(\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\right)_{ji} = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\left(\mathbf{A}^{-1}\mathbf{B}\right)_i$$

The function obtained by the MLS approximation is a smooth curve and it does not pass through the nodal values. Therefore, the MLS shape functions given in Eq. (2.10) do not, in general, satisfy the Kronecker delta condition. Thus,

(2.11) $$\phi_i(\mathbf{x}_j) \neq \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

The weight function which will be used for this approximation is a quartic spline weight function[2]

(3) $$\hat{W}(S) = \begin{cases} 1 - 6S^2 + 8S^3 - 3S^4 & , \; S \leq 1 \\ 0 & , \; S > 1 \end{cases}$$

where $S = \dfrac{d_c}{d_s}$

d_c is the nodal spacing near the point at **x** and d_s is the size of the support domain for the weight function.

2.2 The Linear Advection Equation

The linear advection equation is an unsteady state flow problem. This is a simple atmospheric model. The one-dimensional linear advection equation is given by the expressions[3].

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x}, \tag{4}$$

where $u = u(x, t)$ is the advection quantity. The initial conditions for (4) are $u = \sin(kx)$, $k = 2\pi/Lx$, $Lx = 1000$m, and the advection speed is $c = 10$m/s. Equation (4) is approximated from $t = 0$s to $t = 10$s with the time step of 0.01s. The exact solution of (4) is $u = \sin(k(x-ct))$. The boundary condition is a periodic boundary.

2.3 The Linear Advection Equation in MLPG form

The partial derivatives of displacement u in Eq. (4) can be transformed to the local weak form using the weight residual with the integral by part and the divergence theorem as

$$\int_{\Omega_s} w\frac{\partial u}{\partial t}d\Omega + \int_{L_s} wcu\mathbf{n}d\Gamma + \int_{\Gamma_{su}} wcu\mathbf{n}d\Gamma - \int_{\Omega_s}\frac{\partial w}{\partial x}cud\Omega = 0, \tag{5}$$

where u is the trial function, w is the test function, and Γ_{su} is a part of the boundary $\partial\Omega_s$ of local sub-domain $\Omega_s\,(\in\Omega)$ over which the essential boundary conditions are specified. In Eq. (5), the continuity requirements on trial (u) and test function (w) are not the same, the formulation is called "local unsymmetric weak formulation" and denote as LUSWF1. That can be generated the discretized system equations in the matrix form as

$$\int_{\Omega_s} \mathbf{W}_I\mathbf{\Phi}\mathbf{u}^t d\Omega + \int_{L_s} \mathbf{W}_I\mathbf{cn}\mathbf{\Phi}\mathbf{u}d\Gamma + \int_{\Gamma_{su}} \mathbf{W}_I\mathbf{cn}\mathbf{\Phi}\mathbf{u}d\Gamma - \int_{\Omega_s} \bar{\mathbf{W}}_I\mathbf{c}\mathbf{\Phi}\mathbf{u}d\Omega = 0. \tag{6}$$

Such that

$$\frac{\partial \mathbf{u}^h}{\partial t} = \begin{bmatrix}\phi_1 & \phi_2 & \cdots & \phi_n\end{bmatrix}\begin{Bmatrix}u_1' \\ u_2' \\ \vdots \\ u_n'\end{Bmatrix} = \mathbf{\Phi}_{1\times n}\mathbf{u}'_{n\times 1}. \tag{7}$$

Rearranging and rewriting Eq. (6) into the new form as

$$\mathbf{C}\mathbf{u}^t = \mathbf{K}\mathbf{u}, \tag{8}$$

where the matrix **C** contains the time derivative terms

(9) $$\mathbf{C} = \int_{\Omega_s} \mathbf{W}_I \mathbf{\Phi} d\Omega .$$

The stiffness matrix **K** contains the terms with spatial derivatives

(10) $$\mathbf{K} = \int_{\Omega_s} \mathbf{c}\bar{\mathbf{W}}_I \mathbf{\Phi} d\Omega - \int_{L_s} \mathbf{c}\mathbf{W}_I \mathbf{n}\mathbf{\Phi} d\Gamma - \int_{\Gamma_{su}} \mathbf{c}\mathbf{W}_I \mathbf{n}\mathbf{\Phi} d\Gamma .$$

Time is best discretized using the Crank-Nicolson scheme, which replaces the time derivative at half-step $u^t\left(t+\frac{\Delta t}{2}\right)$ with the central difference approximation[4]

(11) $$u^t\left(t+\frac{\Delta t}{2}\right) = \frac{u(t+\Delta t)-u(t)}{\Delta t} .$$

For $u\left(t+\frac{\Delta t}{2}\right)$ will be approximated by using the average of $u(t)$ and $u(t+\Delta t)$.

This scheme is unconditionally stable. Therefore, the time-discretized in the matrix form of the linear system equation (Eq. (8)) is obtained

(12) $$\mathbf{C}\left(\frac{\mathbf{u}^{t+\Delta t}-\mathbf{u}}{\Delta t}\right) = \mathbf{K}\left(\frac{\mathbf{u}^{t+\Delta t}+\mathbf{u}}{2}\right),$$

where $\mathbf{u}^{t+\Delta t}$ represent the matrix of $u(t+\Delta t)$ and $\mathbf{u}$ is the matrix of $u(t)$.

The final system that has to be solved at every time-step has the form

(13) $$(2\mathbf{C}-\Delta t\mathbf{K})\mathbf{u}^{t+\Delta t} = (2\mathbf{C}+\Delta t\mathbf{K})\mathbf{u} .$$

If the advection equation in Eq. (4) is transformed to local weak form without using the weight residual and the divergence theorem, then the another local unsymmetric weak formulation and it is denote as LUSWF2. That can be written as

(14) $$\int_{\Omega_s} w\frac{\partial u}{\partial t}d\Omega + \int_{\Omega_s} w\left(c\frac{\partial u}{\partial x}\right)d\Omega = 0 \cdot$$

That can be generated the discretized system equations in a matrix form as

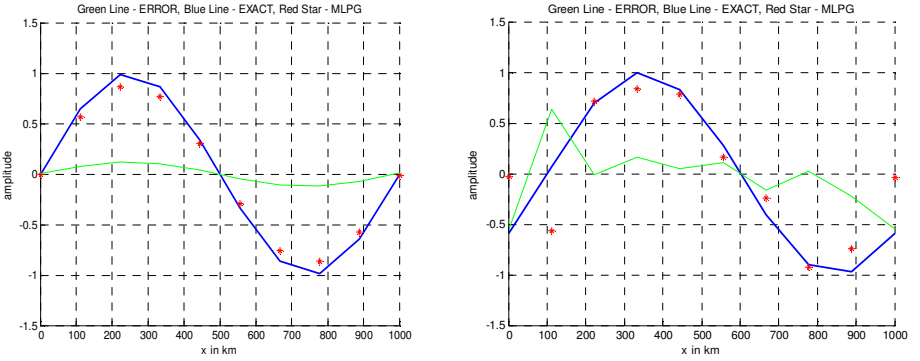
(15) $$\int_{\Omega_s} \mathbf{W}_I\mathbf{\Phi}\mathbf{u}^t d\Omega + \int_{\Omega_s} \mathbf{c}\mathbf{W}_I\bar{\mathbf{\Phi}}\mathbf{u}d\Omega = 0 \cdot$$

Rearranging Eq. (15) into the matrix form of Eq. (8), the new matrix **K** is

(16) $$\mathbf{K} = -\int_{\Omega_s} \mathbf{c}\mathbf{W}_I\bar{\mathbf{\Phi}}d\Omega \cdot$$

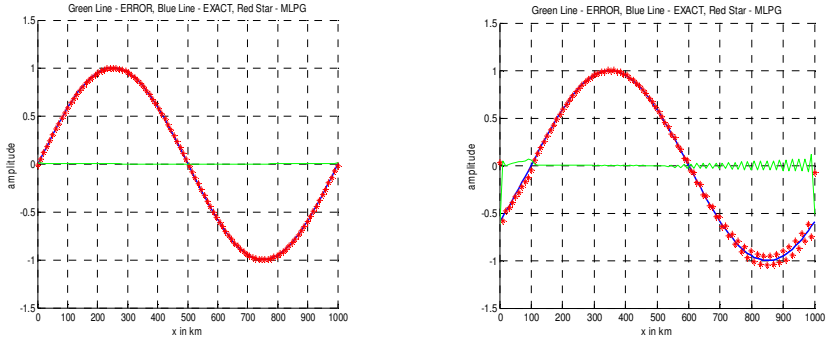
The solution from LUSWF2 will obtained by repeating the procedure in Eq. (13) with using **K** in Eq. (16)

3. Results and Discussion



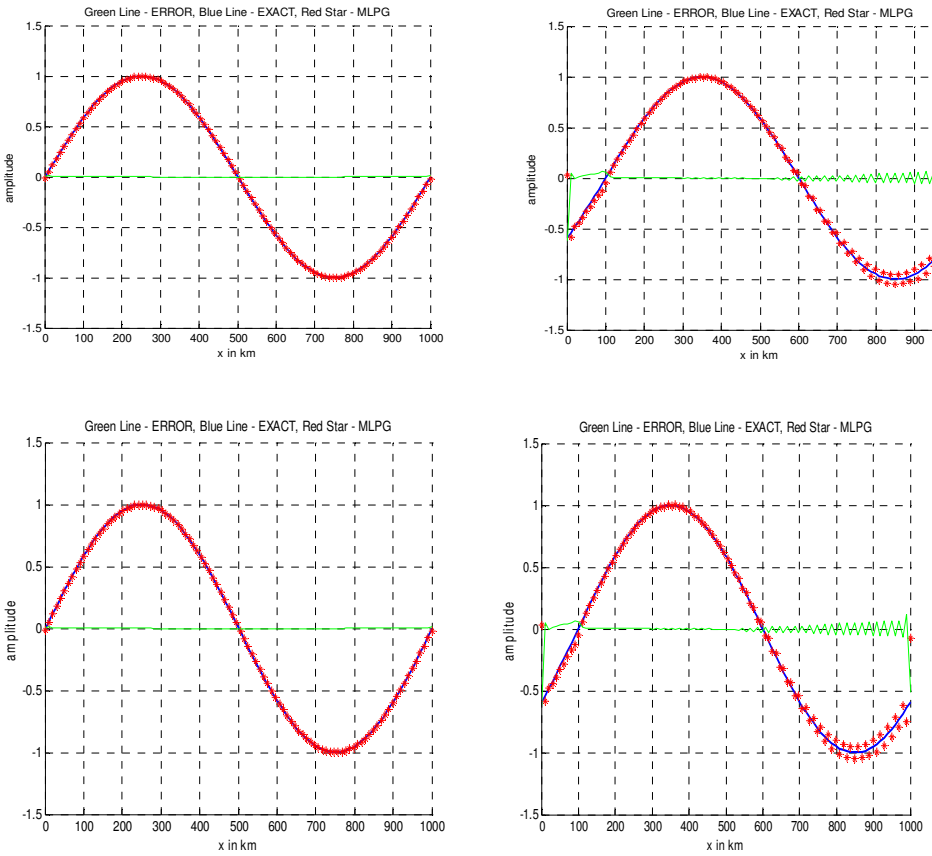
t = 0s t = 10s

Figure 1a Numerical simulations from LUSWF1 with 10 nodes at t = 0s and t = 10s



t=0s t=10s

Fig. 1bNumerical simulations from LUSWF1 with 100 nodes at t = 0s and t = 10s

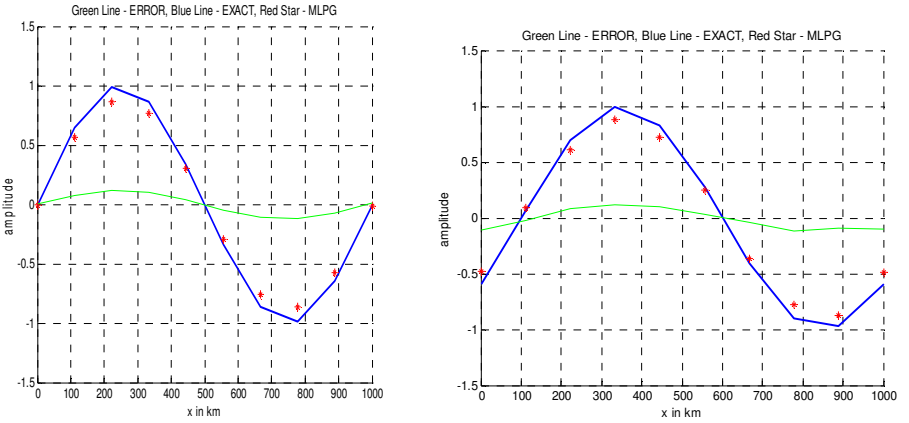


t = 0s t = 10s

Figure 1b Numerical simulations from LUSWF1 with 100 nodes at t = 0s and t = 10s

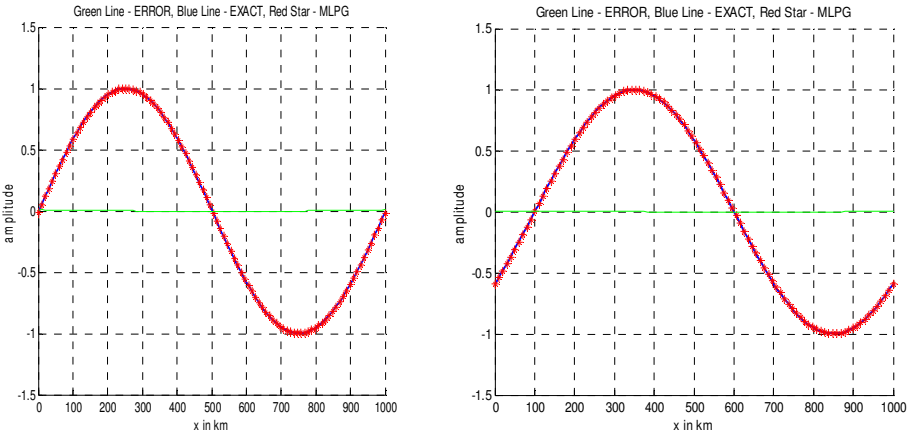
The simulation in Figure 1a shows the moving of sine wave. The red asterisk is the MLPG solution, the blue line is the exact solution and the green line is the error. The wave moves from west to east with the advection speed (c) of 10m/s. From the figure, at the initial time (t = 0s), the result is similar to exact solution. At the later time when t = 10s, the result has large errors. There is no significant difference in amplitude between the MLPG solution and the exact solution. The main cause of error in MLPG method for this case is due to the problem on the boundary. Figure 1b shows the solution of approximation using 100 nodes. Even though the number of node was increased, the error still occur near the boundary when t = 10s. The problem may come from the boundary integral term in Eq. (10), which is not satisfied with the periodic boundary condition.

Repeating the procedure in Eq.(13) using LUSWF2. Figure 2a and 2b show the numerical simulations from the approximation with 10 nodes and 100 nodes, respectively. The efficiency and the accuracy are very good. When t = 10s the sine wave on boundary from MLPG solution are smooth. At the later times, the results still have small error. Thus the periodic boundary condition in this problem is suitable to the LUSWF without the boundary integral term.



t = 0s t = 10s

Figure 2a Numerical simulations from LUSWF2 with 10 nodes at t = 0s and t = 10s



t = 0s t = 10s

Figure. 2b Numerical simulations from LUSWF2 with 100 nodes at t = 0s and t = 10s

4. Conclusion

Meshless Local Petrov-Galerkin (MLPG) is a meshless method, the concept is based on the idea of the local weak form which eliminates the need of the background cell and, consequently, performs the numerical integration in a meshfree sense. In this approach, the global set of equations is derived by writing the weak forms over the small sub-domains defined around the nodes used for the discretization. The size and shape of the local sub-domains may vary from node to node, and they may overlap each other. For each node, it is theoretically possible to define the required size of the local sub-domain by taking into account only the layout of the nodes positioned in the immediate neighbourhood of the node. Consequently, the use of the global mesh of background cells or elements is not necessary throughout the solution procedure, the MLPG represents a truly meshless concepts. Application of the MLPG method to the linear advection equation, by using the cubic spline test function, shows that the main cause of error in approximation for this case is due to the problem on the boundary. The weak point of MLS shape function which do not have Kronecker delta function property. So, the local unsymmetric weak formulation is adjusted without a boundary integral term that is suitable to problems with periodic boundary conditions.

5. Acknowledgments

This research was supported by the Research Fund from the the Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi.

References

1. G. R. Liu, *Meshfree Methods Moving beyond the Finite Element Method, Boca Raton*, CRC Press, 2003.

2. G. R. Liu and Y. T. Gu, *An Introduction to Meshfree Methods and Their Programming,* New York, Springer-Verlag, 2005.

3. R. H. James and R. H. Thomas, *An Introduction to Dynamic Meteorology*, 4th ed., USA, Elsevier Inc, 2004.

4. M. Sterk, B. Robic and R. Trobec, *Mesh Free Method Applied to the Diffusion Equation*, Parallel numerics'05: Theory and Applications, 2005, 57-66