International Journal of
Applied & Experimental Mathematics
Research Article Open Access

The Improved Element-Free Galerkin Method for Two-Dimensional Advection-Diffusion Problems

H. Cheng[1], M. J. Peng[1], Y. M. Cheng[2]*

[1]*Department of Civil Engineering, Shanghai University, Shanghai 200444, China*
[2]*Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University, Shanghai 200072, China*

Abstract

On the basis of the improved moving least-squares (IMLS) approximation, the improved element-free Galerkin (IEFG) method is presented for two-dimensional advection-diffusion problems in this paper, Galerkin weak form of two-dimensional advection-diffusion problems is used to obtain the final discretized equations, the penalty method is used to apply the essential boundary conditions, and difference method for two-point boundary value problems is used for time discretization, then the IEFG method for two-dimensional advection-diffusion problems is presented. Two numerical examples are given to show that the IEFG method has higher computational efficiency.

Publication History:

Received: October 08, 2017
Accepted: December 27, 2018
Published: December 29, 2017

Keywords:

Improved moving least-square approximation, Improved element-free Galerkin method, Two-dimensional advection-diffusion problems, Penalty method, Difference method

Introduction

Meshless method is an important numerical method for science and engineering problems, and has developed rapidly in recent twenty years. Compared with traditional numerical methods based on mesh, the advantage of meshless method only need the information of the nodes in the problem domain, then it can obtain the solution with great precision for some problems, such as the large deformation and dynamic crack growth.

The element-free Galerkin (EFG) method is one of the most important meshless methods, and it has been applied into many science and engineering problems [1-3]. The EFG method is based on the moving least-squares (MLS) approximation, which sometimes forms ill-conditional or singular matrix. In order to overcome the disadvantage of MLS approximation, Cheng, et al. proposed the improved moving least-squares (IMLS) approximation by orthogonalizing the basis function [4]. Using the IMLS approximation to construct shape function, the improved element-free Galerkin (IEFG) method are presented for potential problem [5], transient heat conduction [6], wave equation [7], fracture [8] and elastodynamics [9]. The IEFG method has higher computational efficiency than the EFG method with the same accuracy.

In this paper, we introduce the IEFG method into the two-dimensional advection-diffusion problems. The IMLS approximation is used to obtain the shape functions, Galerkin weak form of two-dimensional advection-diffusion problems is used to obtain the final discretized equations, the penalty method is used to apply the essential boundary conditions, and difference method for two-point boundary value problems is used for time discretization, then the IEFG method for two-dimensional advection-diffusion problems is presented. Two numerical examples are given, and the numerical results are compared with the ones of the EFG method, which shows that the IEFG method in this paper can improve the computational efficiency.

The IEFG Method for Two-Dimensional Advection-Diffusion Problems

The governing equation of two-dimensional advection-diffusion problems is

$$\frac{\partial u}{\partial t}-k_1\frac{\partial^2 u}{\partial x_1^2}-k_2\frac{\partial^2 u}{\partial x_2^2}+v_1\frac{\partial u}{\partial x_1}+v_2\frac{\partial u}{\partial x_2}=f(\boldsymbol{x},t),\ (\boldsymbol{x}=(x_1,x_2)\in\Omega) \tag{1}$$

with the following essential and natural boundary conditions

$$u(\boldsymbol{x},t)=\bar{u}(\boldsymbol{x},t),\ (\boldsymbol{x}\in\Gamma_u), \tag{2}$$

$$q(\boldsymbol{x},t)=k_1\frac{\partial u(\boldsymbol{x},t)}{\partial x_1}n_1+k_2\frac{\partial u(\boldsymbol{x},t)}{\partial x_2}n_2=\bar{q}(\boldsymbol{x},t),\ (\boldsymbol{x}\in\Gamma_q), \tag{3}$$

and the initial condition

$$u(\boldsymbol{x},t)=u_0, \tag{4}$$

where $u(\boldsymbol{x},t)$ is the field function, $\bar{u}(\boldsymbol{x},t)$ is the given field function on the essential boundary Γ_u, $\bar{q}(\boldsymbol{x},t)$ is the given value on the natural boundary Γ_q, $\Gamma=\Gamma_u\cup\Gamma_q$ is the boundary of the problem domain Ω, and $\Gamma_u\cap\Gamma_q=\varnothing$, $f(\boldsymbol{x},t)$ is the source term; k_i is the diffusion efficient in the direction x_i, and v_i is the advection efficient in the direction x_i; u_0 is known function; n_i is the unit outward normal to the boundary Γ in the direction x_i.

The equivalent functional of equations (1) and (3) is

$$\Pi=\int_\Omega\left[u\left(\frac{\partial u}{\partial t}-f\right)\right]\mathrm{d}\Omega+\int_\Omega\frac{1}{2}\left[k_1\left(\frac{\partial u}{\partial x_1}\right)^2+k_2\left(\frac{\partial u}{\partial x_2}\right)^2\right]\mathrm{d}\Omega-\int_{\Gamma_q}u\bar{q}\mathrm{d}\Gamma+\int_\Omega\left(v_1u\frac{\partial u}{\partial x_1}+v_2u\frac{\partial u}{\partial x_2}\right)\mathrm{d}\Omega \tag{5}$$

Imposing essential boundary condition, i.e. equation (5) by the penalty method, we obtain the modified functional as

$$\Pi^*=\Pi+\frac{\alpha}{2}\int_{\Gamma_u}(u-\bar{u})(u-\bar{u})\mathrm{d}\Gamma \tag{6}$$

where α is the penalty factor.

From

$$\delta\Pi^*=0 \tag{7}$$

***Corresponding Author:** Prof. Y. M. Cheng, Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University, Shanghai 200072, China, E-mail: ymcheng@shu.edu.cn

Citation: Cheng YM, Cheng H, Peng MJ (2017) The Improved Element-Free Galerkin Method for Two-Dimensional Advection-Diffusion Problems. Int J Appl Exp Math 2: 122. doi: https://doi.org/10.15344/2456-8155/2017/122

Citation: Cheng YM, Cheng H, Peng MJ (2017) The Improved Element-Free Galerkin Method for Two-Dimensional Advection-Diffusion Problems. Int J Appl Exp Math 2: 122. doi: https://doi.org/10.15344/2456-8155/2017/122

the equivalent integral weak form can be obtained as

$$\delta\Pi^* = \int_\Omega \delta u\cdot\frac{\partial u}{\partial t}\mathrm{d}\Omega - \int_\Omega \delta u\cdot f\mathrm{d}\Omega + \int_\Omega \delta(\boldsymbol{L}u)^{\mathrm{T}}\cdot\tilde{\boldsymbol{k}}\cdot(\boldsymbol{L}u)\mathrm{d}\Omega + \int_\Omega \delta u\cdot v_1\frac{\partial u}{\partial x_1}\mathrm{d}\Omega + \int_\Omega \delta u\cdot v_2\frac{\partial u}{\partial x_2}\mathrm{d}\Omega - \int_{\Gamma_q}\delta u\cdot\bar{q}\mathrm{d}\Gamma + \alpha\int_{\Gamma_u}\delta u\cdot u\mathrm{d}\Gamma - \alpha\int_{\Gamma_u}\delta u\cdot\bar{u}\mathrm{d}\Gamma, \quad (8)$$

where

$$\tilde{\boldsymbol{k}} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \quad (9)$$

$$\boldsymbol{L}(\cdot) = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{bmatrix}(\cdot) \quad (10)$$

We select M nodes $\boldsymbol{x}_I$ (I=1,2,...,M) in plane domain of Ω , $\boldsymbol{x}_I$ are the nodes with domains of influence that cover the point $\boldsymbol{x}$, then we can use the function $u(\boldsymbol{x}_I)$ to approximate the function $u(\boldsymbol{x})$. At the time of t, the function $u(\boldsymbol{x})$ at the node x_I is

$$u_I = u(\boldsymbol{x}_I, t) \quad (11)$$

From the IMLS approximation, the function can be expressed as

$$u(\boldsymbol{x},t) = \Phi^*(\boldsymbol{x})u = \sum_{I=1}^{n}\Phi_I^*(\boldsymbol{x})u_I \quad (12)$$

where n is the nodes number in the compact support domain of $\boldsymbol{x}$.

$$\Phi^*(x) = (\Phi_1^*(x), \Phi_2^*(x), \cdots, \Phi_n^*(x)) = \boldsymbol{p}^{\mathrm{T}}(x)\boldsymbol{A}^*(x)\boldsymbol{B}(x) \quad (13)$$

$\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})=(\boldsymbol{p}_i)$ is the vector of basis function. In general, the linear and the quadratic basis function vectors in the plane domain are given by

$$\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) = (1, x, y) \quad (14)$$

$$\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) = (1, x, y, xy, x^2, y^2) \quad (15)$$

And other matrices and vector are

$$\boldsymbol{A}^*(\boldsymbol{x}) = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{P} \quad (16)$$

$$\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x}) \quad (17)$$

$$\boldsymbol{A}^*(\boldsymbol{x}) = \begin{bmatrix} \frac{1}{(p_1,p_1)} & 0 & \cdots & 0 \\ 0 & \frac{1}{(p_2,p_2)} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{(p_n,p_n)} \end{bmatrix} \quad (18)$$

$$\boldsymbol{P} = \begin{bmatrix} p_1(\boldsymbol{x}_1) & p_2(\boldsymbol{x}_1) & \cdots & p_m(\boldsymbol{x}_1) \\ p_1(\boldsymbol{x}_2) & p_2(\boldsymbol{x}_2) & \cdots & p_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\boldsymbol{x}_n) & p_2(\boldsymbol{x}_n) & \cdots & p_m(\boldsymbol{x}_n) \end{bmatrix} \quad (19)$$

$$\boldsymbol{W}(\boldsymbol{x}) = \begin{bmatrix} w(\boldsymbol{x}-\boldsymbol{x}_1) & 0 & \cdots & 0 \\ 0 & w(\boldsymbol{x}-\boldsymbol{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(\boldsymbol{x}-\boldsymbol{x}_n) \end{bmatrix} \quad (20)$$

$$\boldsymbol{u} = (u_1, u_2, \cdots, u_n)^{\mathrm{T}} \quad (21)$$

where m is the number of basis function, $w(\boldsymbol{x}-\boldsymbol{x}_I)$ is a weighting function with compact support.

From equations (10) and (12), we have

$$\frac{\partial u(\boldsymbol{x},t)}{\partial t} = \frac{\partial}{\partial t}\sum_{I=1}^{n}\Phi_I^*(\boldsymbol{x})u_I = \sum_{I=1}^{n}\Phi_I^*(\boldsymbol{x})\frac{\partial u_I}{\partial t} = \Phi^*(\boldsymbol{x})\dot{\boldsymbol{u}} \quad (22)$$

$$\boldsymbol{L}u(\boldsymbol{x},t) = \sum_{I=1}^{n}\begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{bmatrix}\Phi_I^*(\boldsymbol{x})u_I = \sum_{I=1}^{n}\boldsymbol{B}_I(\boldsymbol{x})u_I = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \quad (23)$$

where

$$\dot{\boldsymbol{u}} = \left(\frac{\partial u(\boldsymbol{x}_1,t)}{\partial t}, \frac{\partial u(\boldsymbol{x}_2,t)}{\partial t}, \cdots, \frac{\partial u(\boldsymbol{x}_n,t)}{\partial t}\right)^{\mathrm{T}} \quad (24)$$

$$\boldsymbol{B}(\boldsymbol{x}) = (\boldsymbol{B}_1(\boldsymbol{x}), \boldsymbol{B}_2(\boldsymbol{x}), \cdots, \boldsymbol{B}_n(\boldsymbol{x})) \quad (25)$$

$$\boldsymbol{B}_I(\boldsymbol{x}) = \begin{bmatrix} \Phi_{I,1}^*(\boldsymbol{x}) \\ \Phi_{I,2}^*(\boldsymbol{x}) \end{bmatrix} \quad (26)$$

Substituting equations (12), (22) and (23) into equation (8) yields

$$\int_\Omega \delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot[\Phi^*(\boldsymbol{x})\dot{\boldsymbol{u}}]\mathrm{d}\Omega - \int_\Omega \delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot f\mathrm{d}\Omega + \int_\Omega \delta[\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}]^{\mathrm{T}}\cdot\tilde{\boldsymbol{k}}\cdot[\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}]\mathrm{d}\Omega + \int_\Omega \delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot v_1\frac{\partial}{\partial x_1}[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\mathrm{d}\Omega + \int_\Omega \delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot v_2\frac{\partial}{\partial x_2}[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\mathrm{d}\Omega - \int_{\Gamma_q}\delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot\bar{q}\mathrm{d}\Gamma + \alpha\int_{\Gamma_u}\delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\mathrm{d}\Gamma - \alpha\int_{\Gamma_u}\delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot\bar{u}\mathrm{d}\Gamma = 0 \quad (27)$$

By analyzing the integral terms in equation (27), we have

$$\int_\Omega \delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot[\Phi^*(\boldsymbol{x})\dot{\boldsymbol{u}}]\mathrm{d}\Omega = \delta\boldsymbol{u}^{\mathrm{T}}\cdot[\int_\Omega \Phi^{*\mathrm{T}}(\boldsymbol{x})\Phi^*(\boldsymbol{x})\mathrm{d}\Omega]\cdot\dot{\boldsymbol{u}} = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\boldsymbol{C}\cdot\dot{\boldsymbol{u}} \quad (28)$$

$$\int_\Omega \delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot f\mathrm{d}\Omega = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\int_\Omega \Phi^{*\mathrm{T}}(\boldsymbol{x})f\mathrm{d}\Omega = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\boldsymbol{F}_1 \quad (29)$$

$$\int_\Omega \delta[\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}]^{\mathrm{T}}\cdot\tilde{\boldsymbol{k}}\cdot[\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}]\mathrm{d}\Omega = \delta\boldsymbol{u}^{\mathrm{T}}\cdot[\int_\Omega \boldsymbol{B}^{\mathrm{T}}(\boldsymbol{x})\tilde{\boldsymbol{k}}\boldsymbol{B}(\boldsymbol{x})\mathrm{d}\Omega]\cdot\boldsymbol{u} = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\boldsymbol{K}\cdot\boldsymbol{u} \quad (30)$$

$$\int_\Omega \delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot v_1\frac{\partial}{\partial x_1}[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\mathrm{d}\Omega = \delta\boldsymbol{u}^{\mathrm{T}}\cdot[\int_\Omega \Phi^{*\mathrm{T}}(\boldsymbol{x})\cdot v_1\frac{\partial}{\partial x_1}\Phi^*(\boldsymbol{x})\mathrm{d}\Omega]\cdot\boldsymbol{u} = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\boldsymbol{G}_1\cdot\boldsymbol{u} \quad (31)$$

$$\int_\Omega \delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot v_2\frac{\partial}{\partial x_2}[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\mathrm{d}\Omega = \delta\boldsymbol{u}^{\mathrm{T}}\cdot[\int_\Omega \Phi^{*\mathrm{T}}(\boldsymbol{x})\cdot v_2\frac{\partial}{\partial x_2}\Phi^*(\boldsymbol{x})\mathrm{d}\Omega]\cdot\boldsymbol{u} = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\boldsymbol{G}_2\cdot\boldsymbol{u} \quad (32)$$

$$\int_{\Gamma_q}\delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot\bar{q}\mathrm{d}\Gamma = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\int_{\Gamma_q}\Phi^{*\mathrm{T}}(\boldsymbol{x})\bar{q}\mathrm{d}\Gamma = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\boldsymbol{F}_2 \quad (33)$$

$$\alpha\int_{\Gamma_u}\delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\mathrm{d}\Gamma = \delta\boldsymbol{u}^{\mathrm{T}}\cdot[\alpha\int_{\Gamma_u}\Phi^{*\mathrm{T}}(\boldsymbol{x})\Phi^*(\boldsymbol{x})\mathrm{d}\Gamma]\cdot\boldsymbol{u} = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\boldsymbol{K}_\alpha\cdot\boldsymbol{u} \quad (34)$$

$$\alpha\int_{\Gamma_u}\delta[\Phi^*(\boldsymbol{x})\boldsymbol{u}]\cdot\bar{u}\mathrm{d}\Gamma = \delta\boldsymbol{u}^{\mathrm{T}}\cdot[\alpha\int_{\Gamma_u}\Phi^{*\mathrm{T}}(\boldsymbol{x})\bar{u}\mathrm{d}\Gamma] = \delta\boldsymbol{u}^{\mathrm{T}}\cdot\boldsymbol{F}_\alpha \quad (35)$$

where

$$\boldsymbol{C} = \int_\Omega \Phi^{*\mathrm{T}}(\boldsymbol{x})\Phi^*(\boldsymbol{x})\mathrm{d}\Omega \quad (36)$$

$$\boldsymbol{F}_1 = \int_\Omega \Phi^{*\mathrm{T}}(\boldsymbol{x})f\mathrm{d}\Omega \quad (37)$$

$$\boldsymbol{K} = \int_\Omega \boldsymbol{B}^{\mathrm{T}}(\boldsymbol{x})\tilde{\boldsymbol{k}}\boldsymbol{B}(\boldsymbol{x})\mathrm{d}\Omega \quad (38)$$

$$\boldsymbol{G}_1 = v_1\int_\Omega \Phi^{*\mathrm{T}}(\boldsymbol{x})\frac{\partial}{\partial x_1}\Phi^*(\boldsymbol{x})\mathrm{d}\Omega \quad (39)$$

$$\boldsymbol{G}_2 = v_2\int_\Omega \Phi^{*\mathrm{T}}(\boldsymbol{x})\frac{\partial}{\partial x_2}\Phi^*(\boldsymbol{x})\mathrm{d}\Omega \quad (40)$$

Citation: Cheng YM, Cheng H, Peng MJ (2017) The Improved Element-Free Galerkin Method for Two-Dimensional Advection-Diffusion Problems. Int J Appl Exp Math 2: 122. doi: https://doi.org/10.15344/2456-8155/2017/122

$$\boldsymbol{F}_2 = \int_{\Gamma_q} \Phi^{*\mathrm{T}}(\boldsymbol{x})\bar{q}\mathrm{d}\Gamma \tag{41}$$

$$\boldsymbol{K}_\alpha = \alpha\int_{\Gamma_u} \Phi^{*\mathrm{T}}(\boldsymbol{x})\Phi^{*}(\boldsymbol{x})\mathrm{d}\Gamma \tag{42}$$

$$\boldsymbol{F}_\alpha = \alpha\int_{\Gamma_u} \Phi^{*\mathrm{T}}(\boldsymbol{x})\bar{u}\mathrm{d}\Gamma \tag{43}$$

Substituting equations (28)-(35) into equation (27), we can obtain

$$\delta\boldsymbol{u}^{\mathrm{T}}\cdot(\boldsymbol{C}\dot{\boldsymbol{u}}+\boldsymbol{K}\boldsymbol{u}+\boldsymbol{K}_\alpha\boldsymbol{u}+\boldsymbol{G}_1\boldsymbol{u}+\boldsymbol{G}_2\boldsymbol{u}-\boldsymbol{F}_1-\boldsymbol{F}_2-\boldsymbol{F}_\alpha)=0 \tag{44}$$

Because the $\delta\boldsymbol{u}^{\mathrm{T}}$ is arbitrary, we can obtain the following ordinary differential equations

$$\boldsymbol{C}\dot{\boldsymbol{u}}+\hat{\boldsymbol{K}}\boldsymbol{u}=\hat{\boldsymbol{F}} \tag{45}$$

$$\hat{\boldsymbol{K}}=\boldsymbol{K}+\boldsymbol{K}_\alpha+\boldsymbol{G}_1+\boldsymbol{G}_2 \tag{46}$$

$$\hat{\boldsymbol{F}}=\boldsymbol{F}_1+\boldsymbol{F}_2+\boldsymbol{F}_\alpha \tag{47}$$

Equation (45) is a linear system of ordinary differential equation, in which time is the only independent variable. Suppose that the time step is Δt, using the traditional difference method for two-point boundary value problems, we can establish the relation of $u_{t+\Delta t}$ and u_t as

$$\theta\left(\frac{\partial\boldsymbol{u}}{\partial t}\right)_{t+\Delta t}+(1-\theta)\left(\frac{\partial\boldsymbol{u}}{\partial t}\right)_t=\frac{\boldsymbol{u}_{t+\Delta t}-\boldsymbol{u}_t}{\Delta t} \tag{48}$$

Solving equation (45) for $(\partial u/\partial t)_{t+\Delta t}$ and $(\partial u/\partial t)$, respectively, and substituting the results into equation (45), as $\boldsymbol{C}$ is independent of time, we obtain

$$\left(\frac{\boldsymbol{C}}{\Delta t}+\theta\hat{\boldsymbol{K}}_{n+1}\right)\boldsymbol{u}_{n+1}=\left[\frac{\boldsymbol{C}}{\Delta t}-(1-\theta)\hat{\boldsymbol{K}}_n\right]\boldsymbol{u}_n+\theta\hat{\boldsymbol{F}}_{n+1}+(1-\theta)\hat{\boldsymbol{F}}_n \tag{49}$$

where θ is a time weighed coefficient, of which the different values correspond to different time difference forms. When θ=0, it is the forward difference scheme. When θ=1/2, it is the C-N (Crank-Nicolson) scheme, and when θ=1, it is the backward difference scheme. In this paper, we use the C-N scheme.

Numerical Examples

In order to verify the advantage of the IEFG method presented in this paper for two-dimensional advection-diffusion problems, we present two numerical examples in this section, and compared the computational accuracy and efficiency of the IEFG method with the ones of the EFG method.

The relative error is defined as

$$\left\|u-u^h\right\|_{L^2(\Omega)}^{rel}=\frac{\left\|u-u^h\right\|_{L^2(\Omega)}}{\left\|u\right\|_{L^2(\Omega)}} \tag{50}$$

where

$$\left\|u-u^h\right\|_{L^2(\Omega)}=\left(\int_\Omega(u-u^h)^2\mathrm{d}\Omega\right)^{1/2} \tag{51}$$

is the L^2 norm of the error.

In this section, the node distribution for each example is regular, and the linear basis function is used. Moreover, 4×4 Gaussian points are used for the Gaussian quadrature in each integration cell.

The first example we considered is the two-dimensional advection-diffusion problem with source term. The governing equation is

$$\frac{\partial u(\boldsymbol{x},t)}{\partial t}=\frac{\partial^2 u(\boldsymbol{x},t)}{\partial x_1^2}+\frac{\partial^2 u(\boldsymbol{x},t)}{\partial x_2^2}-\frac{\partial u(\boldsymbol{x},t)}{\partial x_1}-\frac{\partial u(\boldsymbol{x},t)}{\partial x_2}+f(\boldsymbol{x},t)$$
$$(\boldsymbol{x}\in\Omega,t\in[0,T]) \tag{52}$$

with the initial condition

$$u(\boldsymbol{x},0)=0 \tag{53}$$

and the boundary condition

$$u(\boldsymbol{x},t)=t^{2+\gamma}e^{x_1+x_2},\quad(\boldsymbol{x}\in\Gamma) \tag{54}$$

where

$$f(\boldsymbol{x},t)=0.5\Gamma(3+\gamma)t^2e^{x_1+x_2} \tag{55}$$

The analytical solution of this problem is

$$u(\boldsymbol{x},t)=t^{2+\gamma}e^{x_1+x_2} \tag{56}$$

The problem domain is Ω =[0,1]×[0,1], and T is the total time. In this paper, we select $\gamma=1$.

Using the EFG method to solve this example, 11*11 regularly distributed nodes are selected, the background integral grid is 10*10, Δt=0.01, d_{max} =1.0001, α=6.0×10^{13}, and the quartic spline function is used as the weight function, then the relative errors are 0.0748%, 0.0980%, 0.0972%, 0.0956% and 0.0944% when T are 0.1s, 0.3s, 0.5s, 0.7s and 0.9s, respectively; and the corresponding CPU times are 5.04s, 13.2s, 21.4s, 29.6s, and 39.9s respectvely.

In order to test and verify the effectiveness of the IEFG method, we select Δt=0.01, and 11*11 regularly distributed nodes are selected, the background integral grid is 10*10, d_{max}=1.00001, α=1.2×10^{14} , and the quartic spline function is used as the weight function, the numerical solutions can be obtained with the relative errors of 0.0746%, 0.0955%, 0.0945%, 0.0928% and 0.0916% when T are 0.1s, 0.3s, 0.5s, 0.7s and 0.9s, respectively, and the corresponding CPU times are 4.43s, 11.5s, 18.6s, 25.7s and 32.8s, respectively. The numerical solution and analytical one are in agreement very well (see Figures 1-2).

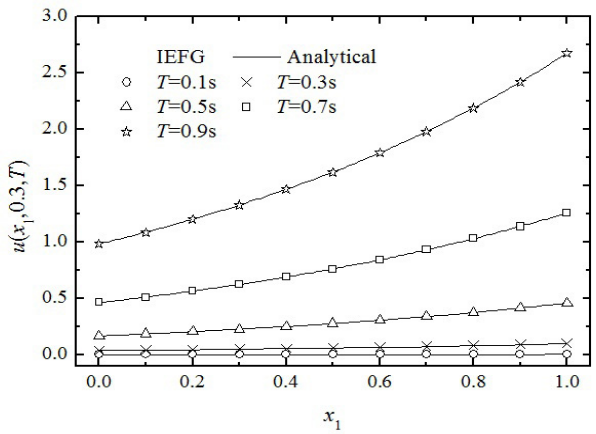

Figure 1: The field function distribution with time along the x_1 axis.

Then we can see that the IEFG method can obtain higher computational efficiency under the condition of same node distribution with similar computational accuracy.

The second example we considered is the following two-dimensional advection-diffusion equation

Citation: Cheng YM, Cheng H, Peng MJ (2017) The Improved Element-Free Galerkin Method for Two-Dimensional Advection-Diffusion Problems. Int J Appl Exp Math 2: 122. doi: https://doi.org/10.15344/2456-8155/2017/122

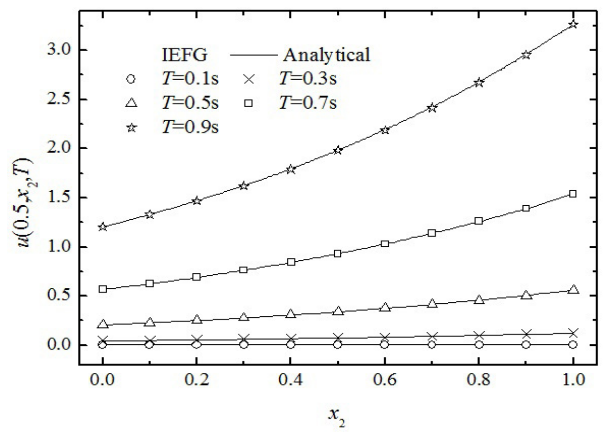

Figure 2: The field function distribution with time along the x_2 axis.

$$\frac{\partial u(\boldsymbol{x},t)}{\partial t}=k_1\frac{\partial^2 u(\boldsymbol{x},t)}{\partial x_1^2}+k_2\frac{\partial^2 u(\boldsymbol{x},t)}{\partial x_2^2}-v_1\frac{\partial u(\boldsymbol{x},t)}{\partial x_1}-v_2\frac{\partial u(\boldsymbol{x},t)}{\partial x_2}$$

$$(\boldsymbol{x}\in\Omega,t\in[0,T]) \tag{57}$$

with the initial condition

$$u(\boldsymbol{x},0)=a(e^{-c_1x_1}+e^{-c_2x_2}) \tag{58}$$

and the boundary conditions

$$u(0,x_2,t)=ae^{bt}(1+e^{-c_2x_2}) \tag{59}$$

$$u(1,x_2,t)=ae^{bt}(e^{-c_1}+e^{-c_2x_2}) \tag{60}$$

$$u(x_1,0,t)=ae^{bt}(e^{-c_1x_1}+1) \tag{61}$$

$$u(x_1,1,t)=ae^{bt}(e^{-c_1x_1}+e^{-c_2}) \tag{62}$$

The problem domain is Ω=[0,1]×[0,1].

The analytical solution of this problem is

$$u(\boldsymbol{x},t)=ae^{bt}(e^{-c_1x_1}+e^{-c_2x_2}) \tag{63}$$

Where

$$\underline{-v_1\pm\sqrt{v_1\ +\ bk_1}} \tag{64}$$

$$c_2=\frac{-v_2\pm\sqrt{v_2^2+4bk_2}}{2k_2} \tag{65}$$

We set $c_1=\left(-v_1+\sqrt{v_1^2+4bk_1}\right)\Big/2k_1$, $c_2=\left(-v_2+\sqrt{v_2^2+4bk_2}\right)\Big/2k_2$ k_1=1.4, k_1=1.7, v_1= v_2=1 and a=b=1, for simplicity. When using the EFG method to solve this example, 11*11 regularly distributed nodes are selected, the background integral grid is 10*10, Δt=0.01, d_{max} =1.16, α=6.3×10[6], and the cubic spline function is used as the weight function, then the relative errors are 0.0064%, 0.0078% and 0.0092% when T are 0.1s, 1s, 3s, respectively; and the corresponding CPU times are 1.3s, 5s and 13s respectively.

When using the IEFG method to solve this example, the same parameters are selected and the cubic spline function is used as the weight function, the numerical solutions can be obtained with the relative errors of 0.0064%, 0.0078% and 0.0092%, when T are 0.1s, 1s and 3s, respectively; and the corresponding CPU times are 1.23s, 4.4s and 11.4s, respectively. The numerical solution and analytical one are in agreement very well (see Figures 3-4).

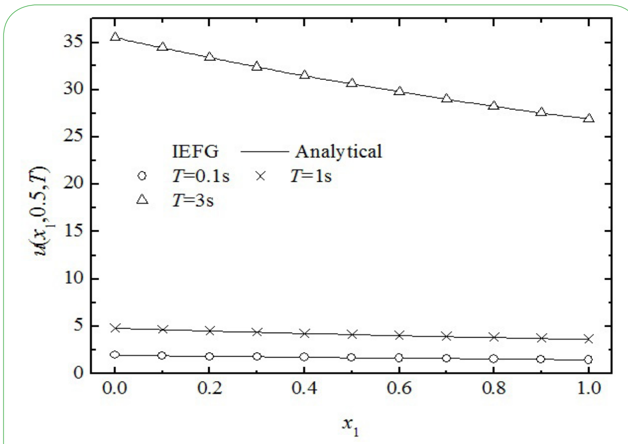

Figure 3: The field function distribution with time along the x_1 axis.

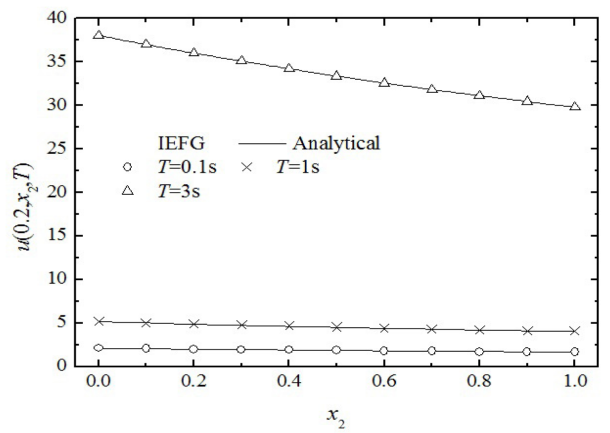

Figure 4: The field function distribution with time along the x_2 axis.

As an extensive investigation of this example, we select k_1=1.4, k_2=1.7, v_1= v_2=1, a=b=1, $c_1=\left(-v_1-\sqrt{v_1^2+4bk_1}\right)\Big/2k_1$ and $c_2=\left(-v_2-\sqrt{v_2^2+4bk_2}\right)\Big/2k_2$.

Using the EFG method to solve this example, 11*11 regularly distributed nodes are selected, the background integral grid is 10*10, Δt=0.01, d_{max} =1.15, α=8.8×10[5], and the cubic spline function is used as the weight function, then the relative errors are 1.4419%, 1.6156% and 1.8028% when T are 0.1s, 1s and 3s, respectively; and the corresponding CPU times are 1.5s, 5.3s and 12.7s respectively.

When using the IEFG method to solve this example, the same parameters are selected and the cubic spline function is used as the weight function, the numerical solutions can be obtained with the relative errors of 1.4419%, 1.6156% and 1.8028% when T are 0.1s, 1s and 3s respectively; and the corresponding CPU times are 1.2s, 4.3s and 11.2s, respectively. The numerical solution and analytical one are in good agreement (see Figures. 5-6).

We can see again that under the condition of same node distribution, similar accuracy can be obtained when using two methods to solve the advection-diffusion problems. However, the IEFG method has higher computational efficiency.

Conclusion

On the basis of the improved moving least-square (IMLS) approximation, the IEFG method for two-dimensional advection-diffusion problems is presented in this paper.

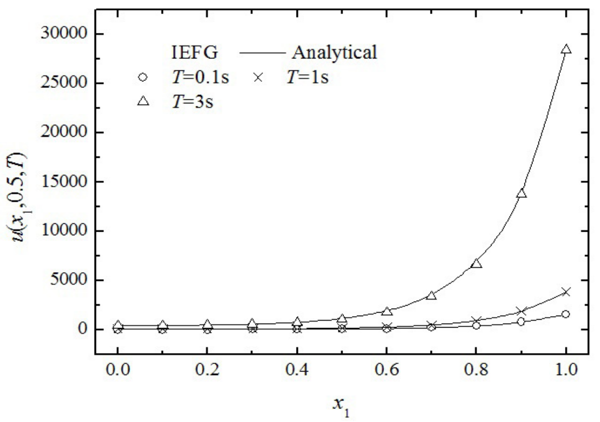

Figure 5: The field function distribution with time along the x_1 axis.

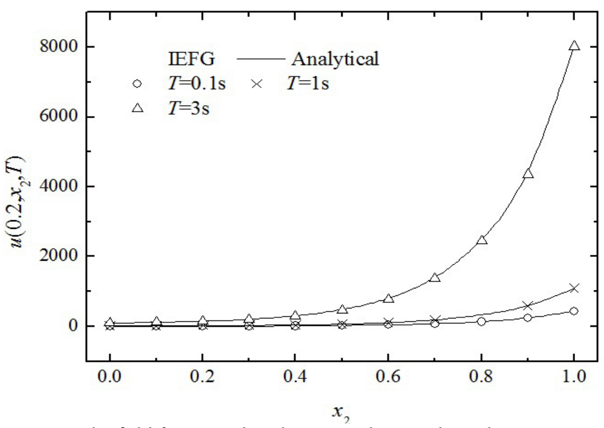

Figure 6: The field function distribution with time along the x_2 axis.

Two numerical examples are given, and the numerical results of the IEFG method are compared with the ones of the EFG method. It is shown that the numerical solutions of the IEFG method are in good agreement with the analytical ones. Moreover, compared with the EFG method, the IEFG method can save the corresponding CPU time under the condition of same node distribution with similar computational accuracy.

Conflict of interest

No authors have a conflict of interest or any financial tie to disclose.

References

1. Gavete L, Falcon S, Ruiz A (2001) An error indicator for the element free Galerkin method. European Journal of Mechanics. A/Solids 20: 327-341.
2. Gavete L, Guesta JL, Ruiz A (2002) A procedure for approximation of the error in the EFG method. International Journal for Numerical Methods in Engineering 53: 677-690.
3. Gavete L, Gavete ML, Alonso B, Martin AJ (2003) A posteriori error approximation in EFG method. International Journal for Numerical Methods in Engineering 58: 2239-2263.
4. Cheng YM, Chen MJ (2003) A boundary element-free method for linear elasticity. Acta Mechanica Sinica 35: 181-186.
5. Zhang Z, Zhao P, Liew KM (2009) Analyzing three-dimensional potential problems with the improved element-free Galerkin method. Computational Mechanics 44: 273-284.
6. Zhang Z, Wang JF, Cheng YM, Liew KM (2013) The improved element-free Galerkin method for three-dimensional transient heat conduction problems. Science China Physics, Mechanics & Astronomy 56: 1568-1580.
7. Zhang Z, Li DM, Cheng YM, Liew KM (2012) The improved element-free Galerkin method for three-dimensional wave equation. Acta Mechanica Sinica 28: 808-818.
8. Zhang Z, Liew KM, Cheng YM, Lee YY (2008) Analyzing 2D fracture problems with the improved element-free Galerkin method. Engineering Analysis with Boundary Elements 32: 241-250.
9. Zhang Z, Hao SY, Liew KM, Cheng YM (2013) The improved element-free Galerkin method for two-dimensional elastodynamics problems. Engineering Analysis with Boundary Elements 37: 1576-1584.