Vol 18 No 10, October 2009
1674-1056/2009/18(10)/4059-06

Chinese Physics B

© 2009 Chin. Phys. Soc.
and IOP Publishing Ltd

Element-free Galerkin (EFG) method for a kind of two-dimensional linear hyperbolic equation*

Cheng Rong-Jun(程荣军)[a)†] and Ge Hong-Xia(葛红霞)[b)]

[a)] *Ningbo Institute of Technology, Zhejiang University, Ningbo* 315100, *China*
[b)] *Faculty of Science, Ningbo University, Ningbo* 315211, *China*

(Received 12 February 2009; revised manuscript received 8 April 2009)

The present paper deals with the numerical solution of a two-dimensional linear hyperbolic equation by using the element-free Galerkin (EFG) method which is based on the moving least-square approximation for the test and trial functions. A variational method is used to obtain the discrete equations, and the essential boundary conditions are enforced by the penalty method. Compared with numerical methods based on mesh, the EFG method for hyperbolic problems needs only the scattered nodes instead of meshing the domain of the problem. It neither requires any element connectivity nor suffers much degradation in accuracy when nodal arrangements are very irregular. The effectiveness of the EFG method for two-dimensional hyperbolic problems is investigated by two numerical examples in this paper.

Keywords: element-free Galerkin (EFG) method, meshless method, hyperbolic problem
PACC: 0200, 0260, 4410

1. Introduction

The finite element method (FEM) and boundary element method (BEM) are well established numerical techniques, which have been used to obtain numerical solutions for various problems in the field of engineering and science.[1−3] Even though these methods are very effective for solving various kinds of partial differential equations, they also have their own limits. For the FEM, the need to produce a body-fitted mesh in two- and three-dimensional problems makes this method time-consuming and difficult to use. As for the BEM, an inhomogeneous equation requires a domain node distribution in addition to a boundary mesh and depends on fundamental solutions. To avoid these problems, in recent years meshless techniques have attracted the attention of researchers. In a meshless (mesh-free) method, a set of scattered nodes is used instead of meshing the domain of the problem. For many engineering problems, such as large deformation and crack growth, it is necessary to deal with extremely large deformations or fractures of the mesh by using the re-meshing technique. The meshless method is a new and interesting numerical technique that can solve many engineering problems that are not solved by conventional numerical methods with a minimum of meshing or no meshing at all.[4]

Some meshless methods have been developed, such as the smooth particle hydrodynamics method (SPH),[5] radial basis function (RBF),[6] element free Galerkin method (EFG),[7,8] meshless local Petrov-Galerkin method (MLPG),[9] finite point method (FPM),[10,11] meshless method based on local boundary integral equation,[12] reproducing kernel particle method (RKPM),[13,14] boundary element-free method (BEFM),[15,16] meshless method with complex variables,[17−20] and so on.

The hyperbolic partial differential equations model the vibrations of structures (e.g. buildings, beams, and machines) and are the basis for fundamental equations of atomic physics. The hyperbolic problems arise in many kinds of engineering problems, which have thus attracted much attention and research recently. It is difficult to find an analytical solution for such problems, so the only choice left is an approximate numerical solution. The popular approach to solving this equation is to discretize time and spatial derivatives by second-order finite difference schemes,[21−23] Mehdi Dehghan has done much work on hyperbolic equations based on the finite difference method,[24−26] M. Kamruzzaman used a meshless collocation method to solve hyperbolic equations,[27] Z. G. Xiong used the FEM to solve a semilinear hyper-

*Project supported by the Natural Science Foundation of Ningbo, China (Grant Nos 2009A610014, 2009A610154, 2008A610020 and 2007A610050).
†Corresponding author. E-mail: chengrongjun76@126.com

bolic equation.[28] To the author's best knowledge, the numerical methods of solving hyperbolic problems are mainly based on the difference method and the finite element method. The meshless methods are seldom applied in this field.

The element-free Galerkin (EFG) method is greatly developed and widely used in various fields. To the best knowledge of the authors, this method has not been used to solve two-dimensional hyperbolic problems. In the present paper, the EFG method is first used to obtain the numerical solution of a two-dimensional hyperbolic problem. In this method, the function over the solution domain needs only a set of nodes, it does not require element connectivity. The variational method is used to obtain the corresponding discrete equations and the essential boundary conditions are enforced by the penalty method. The numerical results whihc are in good agreement with the exact solutions are presented in this paper.

2. Moving least-square approximation[4]

In the EFG method, the moving least-square approximation is used, which is constructed in terms of nodes only. Therefore, there is no need for element and element connectivity data as there is in the FEM. The discretization of the governing equations by the element-free Galerkin method requires moving least-squares approximants, which are made up of three components: a weight function associated with each node, a monomial basis, and a set of coefficients which depends on node position.

2.1. Moving least-square approximation

In the moving least-square approximation,[4] let

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

where m is the number of terms in the basis, $p_j(\boldsymbol{x})$ is monomial basis function, and $\boldsymbol{a}_j(\boldsymbol{x})$ is the coefficient, which as indicated, is the function of the spatial coordinate $\boldsymbol{x}$.

Lancaster and Salkauskas defined a local approximation as

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

where the coefficient $a_j(\boldsymbol{x})$ is obtained by performing a weighted least-square fit in the local approximation. This defines the quadratic form as

$$\begin{aligned} J &= \sum_{i=1}^{n} w(\boldsymbol{x} - \boldsymbol{x}_i) \left[u^h(\boldsymbol{x}, \boldsymbol{x}_i) - u(\boldsymbol{x}_i) \right]^2 \\ &= \sum_{i=1}^{n} w(\boldsymbol{x} - \boldsymbol{x}_i) \left[\sum_{j=1}^{m} p_j(\boldsymbol{x}_i) a_j(\boldsymbol{x}) - u_i \right]^2, \end{aligned} \tag{3}$$

where $w(\boldsymbol{x} - \boldsymbol{x}_i)$ is the weight function with compact support, $\boldsymbol{x}_i$ is the node in the compacted supported domain of $\boldsymbol{x}$. Expression (3) can be rewritten in the vector form

$$J = (\boldsymbol{pa} - \boldsymbol{u})^{\mathrm{T}} \boldsymbol{W}(\boldsymbol{x}) (\boldsymbol{pa} - \boldsymbol{u}), \tag{4}$$

where

$$\boldsymbol{u}^{\mathrm{T}} = (u_1, u_2, ..., u_n), \tag{5}$$

$$\boldsymbol{p} = \begin{bmatrix} p_1(\boldsymbol{x}_1) & p_2(\boldsymbol{x}_1) & \dots & p_m(\boldsymbol{x}_1) \\ p_1(\boldsymbol{x}_2) & p_2(\boldsymbol{x}_2) & \dots & p_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\boldsymbol{x}_n) & p_2(\boldsymbol{x}_n) & \dots & p_m(\boldsymbol{x}_n) \end{bmatrix}, \tag{6}$$

and

$$\boldsymbol{W}(\boldsymbol{x}) = \begin{bmatrix} w(\boldsymbol{x} - \boldsymbol{x}_1) & 0 & \dots & 0 \\ 0 & w(\boldsymbol{x} - \boldsymbol{x}_2) & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w(\boldsymbol{x} - \boldsymbol{x}_n) \end{bmatrix}. \tag{7}$$

To find the coefficient $\boldsymbol{a}(\boldsymbol{x})$, we obtain the extremum of $\boldsymbol{J}$ from

$$\frac{\partial J}{\partial \boldsymbol{a}} = \boldsymbol{A}(\boldsymbol{x}) \boldsymbol{a}(\boldsymbol{x}) - \boldsymbol{B}(\boldsymbol{x}) \boldsymbol{u} = 0, \tag{8}$$

where $\boldsymbol{A}$ is called the moment matrix and is given by

$$\boldsymbol{A} = \boldsymbol{p}^{\mathrm{T}} \boldsymbol{W}(\boldsymbol{x}) \boldsymbol{p}, \tag{9}$$

$$\boldsymbol{B} = \boldsymbol{p}^{\mathrm{T}} \boldsymbol{W}(\boldsymbol{x}). \tag{10}$$

Then we have

$$\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{A}^{-1}(\boldsymbol{x}) \boldsymbol{B}(\boldsymbol{x}) \boldsymbol{u}. \tag{11}$$

The approximation $u^h(\boldsymbol{x})$ can be expressed as

$$u(\boldsymbol{x}) \approx u^h(\boldsymbol{x}) = \sum_{i=1}^{n} \Phi_i^k(\boldsymbol{x}) u_i = \boldsymbol{\Phi}^k(\boldsymbol{x}) \boldsymbol{u}, \tag{12}$$

where the shape function is given by

$$\begin{aligned} \boldsymbol{\Phi}^k(\boldsymbol{x}) &= [\boldsymbol{\Phi}_1^k(\boldsymbol{x}), \dots, \boldsymbol{\Phi}_n^k(\boldsymbol{x})] \\ &= \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{A}^{-1}(\boldsymbol{x}) \boldsymbol{B}(\boldsymbol{x}), \end{aligned} \tag{13}$$

where the superscript k is the order of the polynomial basis. The function corresponding to $k = 0$ is called the Shepard function and given by

$$\Phi_i(\boldsymbol{x}) = \frac{w(\boldsymbol{x} - \boldsymbol{x}_i)}{\sum\limits_{j=1}^{n} w(\boldsymbol{x} - \boldsymbol{x}_j)} . \tag{14}$$

The derivative of the unknown u is approximated by

$$D^\alpha u^h(\boldsymbol{x}) = \sum_{i=1}^{n} D^\alpha \Phi_i^k(\boldsymbol{x}) u_i = D^\alpha \boldsymbol{\Phi}^k(\boldsymbol{x}) \boldsymbol{u}. \tag{15}$$

2.2. Weight function

The weight function used in Eqs.(3)–(13) plays an important role in the EFG method, so the selection of an appropriate weight function is essential. The weight function $w(\boldsymbol{x} - \boldsymbol{x}_i)$ should be non-zero over only a small neighbourhood of $\boldsymbol{x}_i$, called the domain of influence of node $\boldsymbol{x}_i$, to generate a set of sparse discrete equations. The cubic spline weight function used in the present analysis can be written as a function of normalized radius r as follows:

$$w_i = w(\boldsymbol{x} - \boldsymbol{x}_i) \equiv w(r) = \begin{cases} \dfrac{2}{3} - 4r^2 + 4r^3, & r \le \dfrac{1}{2} \\ \dfrac{4}{3} - 4r + 4r^2 - \dfrac{4}{3} r^3, & \dfrac{1}{2} < r \le 1 \\ 0, & r > 1 \end{cases} , \tag{16}$$

where

$$r = \frac{d_i}{d_{m_i}}, \quad d_i = \|\boldsymbol{x} - \boldsymbol{x}_i\|, \quad d_{m_i} = d_{\max} c_i \tag{17}$$

with $d_{\max}$ being a scaling parameter and the distance c_i chosen such that the moment matrix $\boldsymbol{A}$ is not singular.

The spatial derivate of weight function is obtained as

$$\frac{\mathrm{d}w_i}{\mathrm{d}\boldsymbol{x}} = \frac{\mathrm{d}w_i}{\mathrm{d}r} \frac{\mathrm{d}r}{\mathrm{d}\boldsymbol{x}} = \begin{cases} (-8r + 12r^2) \dfrac{\mathrm{sign}(\boldsymbol{x} - \boldsymbol{x}_i)}{d_{m_i}}, & r \le \dfrac{1}{2}, \\ (-4 + 8r - 4r^2) \dfrac{\mathrm{sign}(\boldsymbol{x} - \boldsymbol{x}_i)}{d_{m_i}}, & \dfrac{1}{2} < r \le 1, \\ 0, & r > 1. \end{cases} \tag{18}$$

3. Discretion and numerical implementation

Consider the following two-dimensional linear hyperbolic equation

$$\frac{\partial^2 u}{\partial t^2} + 2a \frac{\partial u}{\partial t} + b^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x, y, t),$$
$$0 \le x, y \le 1,\ 0 \le t < \infty,\ 0 \le b < a \tag{19a}$$

with initial conditions

$$u(x, y, 0) = \varphi_1(x, y), \tag{19b}$$

$$\frac{\partial u(x, y, 0)}{\partial t} = \varphi_2(x, y), \tag{19c}$$

and the boundary conditions

$$u(x, 0, t) = f_0(x, t),$$
$$u(x, 1, t) = f_1(x, t), \tag{20a}$$
$$u(0, y, t) = g_0(y, t),$$
$$u(1, y, t) = g_1(y, t). \tag{20b}$$

The above PDE is an important hyperbolic equation that represents the damped wave equation in two space dimensions, with a source term. The numerical solution of the damped wave equation is of great importance in wave phenomena.

The weak form of Eq.(19a) is

$$\int_\Omega \left[\nabla^{\mathrm{T}} w\nabla u - wf + w\frac{\partial^2 u}{\partial t^2} + 2aw\frac{\partial u}{\partial t} + b^2 wu\right]\mathrm{d}\Omega - \int_S w\left(\frac{\partial u}{\partial x}n_x + \frac{\partial u}{\partial x}n_y\right)\mathrm{d}S = 0. \tag{21}$$

The functional $\Pi(u)$ can be written as

$$\Pi(u) = \frac{1}{2}\int_\Omega \nabla^{\mathrm{T}} u\cdot\nabla u\mathrm{d}\Omega + \int_\Omega \left[u\frac{\partial^2 u}{\partial t^2} + 2au\frac{\partial u}{\partial t} + \frac{1}{2}b^2u^2 - uf\right]\mathrm{d}\Omega. \tag{22}$$

Enforcing essential boundary conditions by using the penalty method, the modified functional $\Pi^*(u)$ can be obtained as

$$\Pi^*(u) = \frac{1}{2}\int_\Omega \nabla^{\mathrm{T}} u\cdot\nabla u\mathrm{d}\Omega + \int_\Omega \left[u\frac{\partial^2 u}{\partial t^2} + 2au\frac{\partial u}{\partial t} + \frac{1}{2}b^2u^2 - uf\right]\mathrm{d}\Omega + \frac{\alpha}{2}\int_S (u-\bar{u})^2\mathrm{d}S. \tag{23}$$

Taking variational calculus, i.e. $\delta\Pi^*(T)$ of Eq.(23), it reduces into

$$\delta\Pi^*(u) = \int_\Omega \nabla^{\mathrm{T}} u\cdot\delta\nabla u\mathrm{d}\Omega + \int_\Omega \left[u\frac{\partial^2 u}{\partial t^2} + 2a\delta u\frac{\partial u}{\partial t} + b^2u\delta u - \delta uf\right]\mathrm{d}\Omega + \alpha\int_S \delta u(u-\bar{u})\mathrm{d}S, \tag{24}$$

where α is the penalty factor.

From Eq.(12), we can obtain the approximation of the temperature function

$$u^h(\boldsymbol{x},t) = \sum_{I=1}^n \Phi_I(\boldsymbol{x})\cdot T_I(t) = \boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T}, \tag{25}$$

$$\frac{\partial u(\boldsymbol{x},t)}{\partial t} = \frac{\partial}{\partial t}\sum_{I=1}^n \Phi_I(\boldsymbol{x})\cdot T_I(t) = \sum_{I=1}^n \Phi_I(\boldsymbol{x})\cdot\frac{\partial T_I(t)}{\partial t} = \boldsymbol{\Phi}(\boldsymbol{x})\dot{\boldsymbol{T}}, \tag{26}$$

$$\frac{\partial^2 u(\boldsymbol{x},t)}{\partial t^2} = \frac{\partial^2}{\partial t^2}\sum_{I=1}^n \Phi_I(\boldsymbol{x})\cdot T_I(t) = \sum_{I=1}^n \Phi_I(\boldsymbol{x})\cdot\frac{\partial^2 T_I(t)}{\partial t^2} = \boldsymbol{\Phi}(\boldsymbol{x})\ddot{\boldsymbol{T}}, \tag{27}$$

where

$$\boldsymbol{\Phi}(\boldsymbol{x}) = (\Phi_1(\boldsymbol{x}),\Phi_2(\boldsymbol{x}),\ldots,\Phi_n(\boldsymbol{x})), \tag{28}$$

$$\boldsymbol{T} = (T_1(t),T_2(t),\ldots,T_n(t))^{\mathrm{T}}, \tag{29}$$

$$\dot{\boldsymbol{T}} = \left(\frac{\partial T_1(t)}{\partial t},\frac{\partial T_2(t)}{\partial t},\ldots,\frac{\partial T_n(t)}{\partial t}\right)^{\mathrm{T}}, \tag{30}$$

$$\ddot{\boldsymbol{T}} = \left(\frac{\partial^2 T_1(t)}{\partial t^2},\frac{\partial^2 T_2(t)}{\partial t^2},\ldots,\frac{\partial^2 T_n(t)}{\partial t^2}\right)^{\mathrm{T}}, \tag{31}$$

$$\boldsymbol{x} = (x,y). \tag{32}$$

Substituting Eqs.(25)–(27) into Eq.(24) yields

$$\begin{aligned}\delta\Pi^*(u) = &\int_\Omega \nabla^{\mathrm{T}}(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})\cdot\delta\nabla(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})\mathrm{d}\Omega\\ &+ \int_\Omega[(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\ddot{\boldsymbol{T}})\\ &+ 2a\delta(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\dot{\boldsymbol{T}})\\ &+ b^2(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})\delta(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})\\ &- \delta(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})f]\mathrm{d}\Omega\\ &+ \alpha\int_S \delta(\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})((\boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{T})-\bar{u})\mathrm{d}S.\end{aligned} \tag{33}$$

Since $\delta\Pi^*(u) = 0$ and $\delta(T)$ is arbitrary in Eq.(33), the following equations can be obtained

$$\boldsymbol{C}\ddot{\boldsymbol{T}} + 2a\boldsymbol{C}\dot{\boldsymbol{T}} + (\boldsymbol{K}+b^2\boldsymbol{C})\boldsymbol{T} = \boldsymbol{F}, \tag{34}$$

where

$$\boldsymbol{C} = \int_\Omega \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\Phi}(\boldsymbol{x})\mathrm{d}\Omega, \tag{35}$$

$$C_{IJ} = \int_\Omega \Phi_I^{\mathrm{T}}(\boldsymbol{x})\Phi_J(\boldsymbol{x})\mathrm{d}\Omega, \tag{36}$$

$$\begin{aligned}\boldsymbol{K} = &\int_\Omega \nabla\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\nabla\boldsymbol{\Phi}(\boldsymbol{x})\mathrm{d}\Omega\\ &+ \int_S \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\alpha\boldsymbol{\Phi}(\boldsymbol{x})\mathrm{d}S,\end{aligned} \tag{37}$$

$$\begin{aligned}K_{IJ} = &\int_\Omega \nabla\Phi_I^{\mathrm{T}}(\boldsymbol{x})\nabla\Phi_J(\boldsymbol{x})\mathrm{d}\Omega\\ &+ \int_S \Phi_I^{\mathrm{T}}(\boldsymbol{x})\alpha\Phi_J(\boldsymbol{x})\mathrm{d}S,\end{aligned} \tag{38}$$

$$F_I = \int_\Omega \Phi_I(x)f\mathrm{d}\Omega + \int_S \alpha\Phi_I(x)\bar{u}\mathrm{d}S. \tag{39}$$

Making the time discretization of Eq.(37) by the

central difference method yields

$$\boldsymbol{C}\frac{\boldsymbol{T}^{(k+2)}-2\boldsymbol{T}^{(k+1)}+\boldsymbol{T}^{(k)}}{\Delta t^2}+2a\boldsymbol{C}\frac{\boldsymbol{T}^{(k+1)}-\boldsymbol{T}^{(k)}}{\Delta t}+(\boldsymbol{K}+b^2\boldsymbol{C})\frac{\boldsymbol{T}^{(k+1)}+\boldsymbol{T}^{(k)}}{2}=\boldsymbol{F}, \tag{40}$$

that is

$$\begin{aligned}&2\boldsymbol{C}\boldsymbol{T}^{(k+2)}\\ =\ &(4\boldsymbol{C}-4a\Delta t\boldsymbol{C}-(\boldsymbol{K}+b^2\boldsymbol{C})\Delta t^2)\boldsymbol{T}^{(k+1)}\\ &+(4a\Delta t\boldsymbol{C}-2\boldsymbol{C}-(\boldsymbol{K}+b^2\boldsymbol{C})\Delta t^2)\boldsymbol{T}^{(k)}\\ &+2\Delta t^2\boldsymbol{F},\end{aligned} \tag{41}$$

where

$$\boldsymbol{T}^{(k)}=\boldsymbol{T}(t_k)=(T_1(t_k),T_2(t_k),\ldots,T_n(t_k)). \tag{42}$$

4. Numerical examples

In this section, the application of the element-free Galerkin (EFG) method to two-dimensional linear hyperbolic problems is demonstrated by several examples which are chosen such that their exact solutions are known.

A regular arrangement of nodes and a background mesh of cells are used to numerically calculate the system equation. In each integration cell, a 4–4 Gauss quadrature scheme is used to evaluate the stiffness matrix. The linear basis and the cubic spline weight function are used in the EFG method.

4.1. Test problem 1

Consider Eq.(19a) with the following conditions

$$\varphi_1(x,y)=\sinh(x)\sinh(y), \tag{43}$$
$$\varphi_2(x,y)=-\sinh(x)\sinh(y), \tag{44}$$
$$f_0(x,t)=0, \tag{45}$$
$$f_1(x,t)=\mathrm{e}^{-t}\sinh(x)\sinh(1), \tag{46}$$
$$g_0(y,t)=0, \tag{47}$$
$$g_1(y,t)=\mathrm{e}^{-t}\sinh(1)\sinh(y), \tag{48}$$
$$\begin{aligned}f(x,y,t)=&-2a\mathrm{e}^{-t}\sinh(x)\sinh(y)\\ &+b^2\mathrm{e}^{-t}\sinh(x)\sinh(y)\\ &-\mathrm{e}^{-t}\sinh(x)\sinh(y),\end{aligned} \tag{49}$$

where the exact solution is

$$u(x,y,t)=\mathrm{e}^{-t}\sinh(x)\sinh(y). \tag{50}$$

4.2. Test problem 2

Consider Eq.(19a) with the following conditions:

$$\varphi_1(x,y)=\sin(x)\sin(y), \tag{51}$$
$$\varphi_2(x,y)=0, \tag{52}$$
$$f_0(x,t)=0, \tag{53}$$
$$f_1(x,t)=\cos(t)\sin(x)\sin(1), \tag{54}$$
$$g_0(y,t)=0, \tag{55}$$
$$g_1(y,t)=\cos(t)\sin(1)\sin(y), \tag{56}$$
$$\begin{aligned}f(x,y,t)=&-2a\sin(x)\sin(y)\sin(t)\\ &+b^2\cos(t)\sin(x)\sin(y)\\ &+\cos(t)\sin(x)\sin(y),\end{aligned} \tag{57}$$

where the exact solution is

$$u(x,y,t)=\cos(t)\sin(x)\sin(y). \tag{58}$$

We use the EFG method to solve the above hyperbolic problem with penalty factor $\alpha=5\times10^3$ and scaling parameter $d_{\max}=1.15$ with time step length $\Delta t=0.001$. For example, Fig. 1 shows node arrangement. The weight function is chosen to be a cubic spline function and the bases are chosen to be linear. In Fig.2, the exact solutions $u(x,y,t)=\mathrm{e}^{-t}\sinh(x)\sinh(y)$ are plotted in comparison with numerical results when $t=0.1$, 0.3, 0.5, 0.7 and 1 at the location $x=0.5$. In Fig.3, exact solutions $u(x,y,t)=\cos(t)\sin(x)\sin(y)$ are plotted in comparison with numerical results when $t=0.1$, 0.3, 0.5, 0.7 and 1 at the location $x=0.5$, respectively. From Figs. 2 and 3, we can conclude that numerical results by using EFG to solve the two-dimensional hyperbolic problems are in good agreement with the exact solutions.

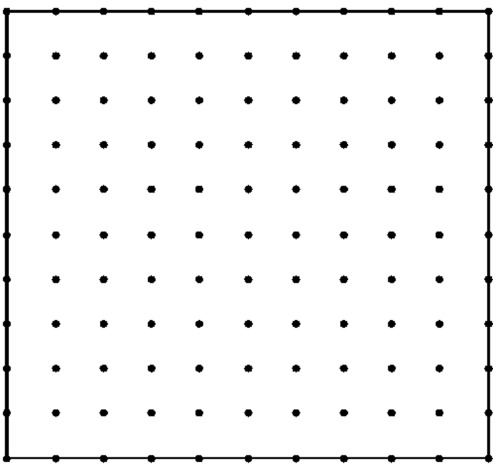


Fig.1. 11×11 nodes arrangement for examples 1 and 2.

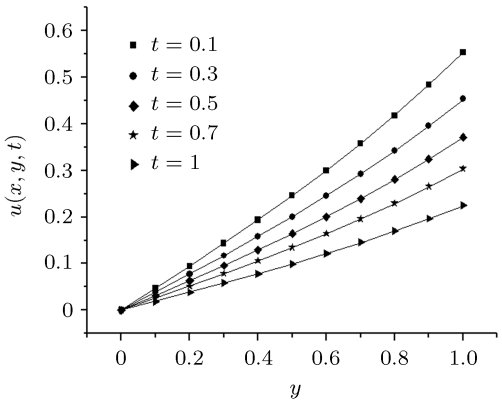


Fig.2. Numerical solution and exact solution of $u(x, y, t)$ when $t = 0.1$, 0.3, 0.5, 0.7, 1 at the location $x = 0.5$ (test example 1).

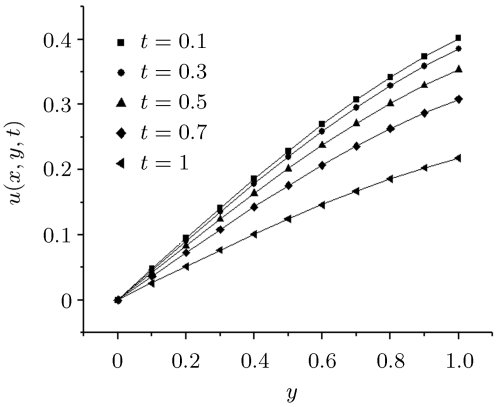


Fig.3. Numerical solution and exact solution of $u(x, y, t)$ when $t = 0.1$, 0.3, 0.5, 0.7, 1 at the location $x = 0.5$ (test example 2).

5. Conclusion

The element-free Galerkin (EFG) method is used for solving the two-dimensional linear hyperbolic problem. The meshless property of EFG is the most important advantage of this scheme over the traditional mesh-dependent techniques such as the finite difference method, the finite element method and the boundary element method. The meshless nature allows us to solve problems with non-regular geometry. The numerical results which are in good agreement with the exact solutions show that the technique is accurate and efficient. The EFG technique can be extended to three-dimensional hyperbolic problems and unsteady nonlinear heat conduction problems.

References

[1] Donea J and Giuliani S 1974 *Nucl. Eng. Des.* **30** 205
[2] Bathe K J and Khoshgoftaar M R 1979 *Nucl. Eng. Des.* **51** 389
[3] Skerget P 1983 *Nucl. Eng. Des.* **76** 47
[4] Belytschko T, Krongauz Y and Organ D 1996 *Comput. Meth. Appl. Mech. Engng.* **139** 3
[5] Monaghan J J 1988 *Comput. Phys. Commun.* **48** 89
[6] Chen W 2000 *Meshfree Methods for Partial Differential Equations* (Berlin: Springer-Verlag)
[7] Belytschko T, Lu Y Y and Gu L 1994 *Int. J. Numer. Meth. Engng.* **37** 229
[8] Cheng R J and Cheng Y M 2008 *Acta Phys. Sin.* **57** 6037 (in Chinese)
[9] Atluri S N and Zhu T 1998 *Comput. Mech.* **22** 117
[10] Cheng R J and Cheng Y M 2008 *Appl. Numer. Math.* **58** 884
[11] Cheng R J and Cheng Y M 2007 *Acta Phys. Sin.* **56** 5569 (in Chinese)
[12] Dai B D and Cheng Y M 2007 *Acta Phys. Sin.* **56** 597 (in Chinese)
[13] Liu W K, Jun S and Zhang Y F 1995 *Int. J. Numer. Meth. Engng.* **20** 1081
[14] Qin Y X and Cheng Y M 2006 *Acta Phys. Sin.* **55** 3215 (in Chinese)
[15] Cheng Y M and Chen M J 2003 *Acta Mech. Sin.* **35** 181 (in Chinese)
[16] Cheng Y M and Peng M J 2005 *Sci. Chin.* G *Phys. Mech. & Astron.* **48** 641
[17] Cheng Y M and Li J H 2005 *Acta Phys. Sin.* **54** 4463 (in Chinese)
[18] Chen Li and Cheng Y M 2008 *Acta Phys. Sin.* **57** 1 (in Chinese)
[19] Chen Li and Cheng Y M 2008 *Acta Phys. Sin.* **57** 5069 (in Chinese)
[20] Peng M J and Cheng Y M 2009 *Engineering Analysis with Boundary Elements* **23** 77
[21] Lapidus L and Pinder G F 1982 *Numerical Solution of Partial Differential Equations in Science and Engineering* (New York: Wiley)
[22] Dehghan M *Math. Comput. Simulation* **71** 16
[23] Dehghan M 2005 *Numer. Methods Partial Differ. Equations* **21** 24
[24] Dehghan M 2006 *Numer. Methods Partial Differ. Equations* **22** 220
[25] Dehghan M 2007 *Chaos, Solitons and Fractals* **32** 661
[26] Dehghan M 2007 *Kybernetes* **36** 791
[27] Kamruzzaman M 2008 *Commun. Numer. Methods in Engineering* **24** 1617
[28] Xiong Z G 2008 *J. Comput. Appl. Math.* **24** 313