International Journal of Heat and Mass Transfer 71 (2014) 720–723

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ijhmt

Application of the finite pointset method to non-stationary heat conduction problems

Edgar O. Reséndiz-Flores [a,*], Irma D. García-Calvillo [b]

[a] Division of Postgraduate Studies and Research, Department of Metal-Mechanical Engineering, The Technological Institute of Saltillo, Blvd. V. Carranza 2400 Col. Tecnológico, C.P. 25280 Saltillo, Coahuila, Mexico

[b] Research Center on Applied Mathematics, Autonomous University of Coahuila, Campus Saltillo, Building S, C.P. 25000 Saltillo, Coahuila, Mexico

ARTICLE INFO

Article history:
Received 21 October 2013
Accepted 30 December 2013
Available online 25 January 2014

Keywords:
Non-stationary heat conduction
Meshless method
Finite pointset method

ABSTRACT

This paper proposes the use of the finite pointset method for the numerical solution of two-dimensional transient heat conduction problems. The strong formulation of the parabolic partial differential equation is directly used instead of the corresponding weak form. Moreover, a numerical comparison between the finite pointset method and the corresponding analytical solutions is reported.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

During the past 30 years different type of meshless methods for solving partial differential equations have been developed with the objective of eliminating part of the difficulties arising when mesh-based methods are used. The starting point of these methods was the smooth particle hydrodynamics (SPH) method proposed by Lucy [1] and Gingold and Monaghan [2] in the area of astrophysics and applied later in other research fields. For a good overview on meshless methods we refer to [3] and references therein.

In the field of heat transfer by conduction different meshless methods have been already applied. Among these scientific works some of the recent publications, to the authors knowledge, are the following works: Liu et al. [4] uses a meshless weighted least squares method for heat conduction. Chen et al. [5] applied a corrective SPH method to solve unsteady heat conduction problems. Cheng and Liew [6,7] applied the reproducing kernel particle method (RKPM) for two and three-dimensional unsteady heat conduction problems, respectively. We refer to [7] for a concise and recent overview on meshless methods on the field of heat conduction problems.

As an alternative to the solution for the unsteady heat conduction equation we propose in this work the application of a slightly different version of the finite point method developed by Oñate [8]. This method is called the finite pointset method (FPM) and to the authors knowledge this version of the method has been developed by Kuhnert [9] in the Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, in Kaiserslautern, Germany, and it has been already applied in the fields of fluid mechanics [10–13] and radiative heat transfer problems [14]. Later on, a very close version of the FPM of Kuhnert has been also developed by Cheng and Liu, [15] in the field of fluid mechanics. In this work we propose the application of the finite pointset method of Kuhnert to the field of heat conduction for unsteady problems being the first time, to the authors knowledge, that this method is applied in this particular research field. In order to get some insight on his performance we compare the numerical solution to some specific problems whose analytical solutions are known.

The structure of the paper is as follows: Sections 2 shortly describe the basic ideas behind FPM. Section 3 presents some issues regarding the numerical implementation of FPM. The test examples are presented in Section 4 with the corresponding results. Finally some conclusions are given in last section.

2. The FPM method

In this section we describe the main ideas of the FPM method proposed by [9]. The FPM is a member of the family of the least square (LS) methods and it is closely related to the finite point method by Oñate et al. [8,16]. Although they are very similar they are not identical. The main difference is that finite point method of Oñate uses polynomial basis and the FPM method uses Taylor series which allow to compute, by an LS approach, the function and its derivatives values that naturally appear as unknown coefficients in the series.

* Corresponding author. Tel./fax: +52 844 438 9500.
E-mail addresses: eresendiz@its.mx (E.O. Reséndiz-Flores), idgarcia@cima.uadec.mx (I.D. García-Calvillo).

0017-9310/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ijheatmasstransfer.2013.12.077

The method is based on the so-called moving least squares procedure which is shortly described next, following [14]:

Let Ω be a given domain with boundary $\partial\Omega$ and suppose that the set of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ is distributed with corresponding function values $f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)$. The problem is to find an approximate value of f at some arbitrary location $f(\mathbf{x})$. Thus, the following procedure can be applied:

Define the approximation to $f(\mathbf{x})$ as

$$\tilde{f}(\mathbf{x}) = \sum_{k=1}^{m} p_k(\mathbf{x}) b_k(\mathbf{x}) = \mathbf{p}^t(\mathbf{x})\mathbf{b}(\mathbf{x}) \tag{1}$$

whose local version reads

$$\tilde{f}(\mathbf{x}, \bar{\mathbf{x}}) = \sum_{k=1}^{m} p_k(\bar{\mathbf{x}}) b_k(\mathbf{x}) = \mathbf{p}^t(\bar{\mathbf{x}})\mathbf{b}(\mathbf{x}) \tag{2}$$

where $p_k(\mathbf{x})$ denotes a set of linear independent functions, in particular, they can be linear monomials.

Now, minimize the quadratic form

$$J = \sum_{j=1}^{n} w(\mathbf{x}, \mathbf{x}_j) e_j^2 \tag{3}$$

$$= \sum_{j=1}^{n} w(\mathbf{x}, \mathbf{x}_j) \left(\sum_{k=1}^{m} p_k(\mathbf{x}_j) b_k(\mathbf{x}) - f(\mathbf{x}_j) \right)^2 \tag{4}$$

in order to get the optimal coefficients

$$\mathbf{b} = A^{-1} B\mathbf{f} = (BP)^{-1} (P^t W)\mathbf{f} \tag{5}$$

where

$$P = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_{n_p}) & p_2(\mathbf{x}_{n_p}) & \cdots & p_m(\mathbf{x}_{n_p}) \end{bmatrix} \tag{6}$$

$$W = \begin{bmatrix} w(\mathbf{x}, \mathbf{x}_1) & 0 & \cdots & 0 \\ 0 & w(\mathbf{x}, \mathbf{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(\mathbf{x}, \mathbf{x}_{n_p}) \end{bmatrix} \tag{7}$$

$$\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_{n_p})]^t = [f_1, f_2, \ldots, f_{n_p}]^t, \tag{8}$$

$$\mathbf{b}(\mathbf{x}) = [b_1(\mathbf{x}), b_2(\mathbf{x}), \ldots, b_m(\mathbf{x})]^t \tag{9}$$

n_p denotes the number of neighbor points $\mathbf{x}_j$ of $\mathbf{x}$ and $w(\mathbf{x}, \mathbf{x}_j)$ denotes a weight function with compact support. Moreover, different weight functions have been used in the literature and the most common functions are the cubic spline and the Gaussian functions, being the last one chosen to be used in this paper.

Once $\mathbf{b}$ is known the function approximation at point $\mathbf{x}$ reads

$$\tilde{f}(\mathbf{x}) = \sum_{k=1}^{m} p_k(\mathbf{x}) b_k(\mathbf{x}) = \mathbf{p}^t(\mathbf{x}) A(\mathbf{x})^{-1} B(\mathbf{x})\mathbf{f} = \Phi(\mathbf{x})\mathbf{f} \tag{10}$$

If the base functions $p_i(\mathbf{x})$ are defined as follows:

$$\mathbf{p}^t = [1, \Delta x_j, \Delta y_j, \ldots], \tag{11}$$

where $\Delta x_j = x_j - x$ and $\Delta y_j = y_j - y$ for $j = 1, \ldots, n_p$, the following equivalent representation is obtained

$$\tilde{f}(\mathbf{x}) \simeq \sum_{k=1}^{m} p_k(\mathbf{x}) b_k(\mathbf{x}) = f(\mathbf{x}_j) + \nabla f(\mathbf{x}_j) \cdot \Delta \mathbf{x}_j + \cdots \tag{12}$$

which implies that under this representation the new vector of unknown coefficients becomes

$$\mathbf{b}(\mathbf{x}) = [f(\mathbf{x}), \partial_x f(\mathbf{x}), \partial_y f(\mathbf{x}), \ldots]^t \tag{13}$$

In this way we automatically get the values of the function and its derivatives at points $\mathbf{x}$. We refer to [10] for a more explicit presentation of the FPM method applied to the Poisson equation.

3. Numerical implementation

Along this section we will describe some issues regarding the numerical implementation for FPM applied to two dimensional problems of the following form

$$\rho c \frac{\partial T}{\partial t} - k\Delta T = \dot{Q}, \quad \text{in } \Omega \tag{14}$$

with the following boundary conditions

$$T = \overline{T} \quad \text{on } \Gamma_1 \tag{15}$$

$$\mathbf{n} \cdot k\nabla T = \bar{q} \quad \text{on } \Gamma_2 \tag{16}$$

$$\mathbf{n} \cdot k\nabla T = h_c(T - T_\infty) \quad \text{on } \Gamma_3 \tag{17}$$

and initial condition

$$T|_{t=0} = T_0 \tag{18}$$

where Ω denotes de domain of interest, $\partial\Omega = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ is the boundary. $T, \dot{Q}, k, c, \rho$ and $\mathbf{n}$ denotes the temperature, source of heat generation per unit volume, thermal conductivity, specific heat of the material, density of the material and the unit outward normal to the boundary, respectively.

3.1. FPM discretization

3.1.1. FPM form for the heat equation

In the FPM representation for the unsteady two dimensional heat equation, the matrices we need to compute by each particle in Ω take the following form:

If $\mathbf{x}_i \in \Omega$, then

$$P = \begin{pmatrix} 1 & \Delta x_1 & \Delta y_1 & \frac{1}{2}(\Delta x_1)^2 & \Delta x_1 \Delta y_1 & \frac{1}{2}(\Delta y_1)^2 \\ 1 & \Delta x_2 & \Delta y_2 & \frac{1}{2}(\Delta y_2)^2 & \Delta x_2 \Delta y_2 & \frac{1}{2}(\Delta y_2)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta x_n & \Delta y_n & \frac{1}{2}(\Delta y_n)^2 & \Delta x_n \Delta y_n & \frac{1}{2}(\Delta y_n)^2 \\ 2 & 0 & 0 & -\Delta t & 0 & -\Delta t \end{pmatrix} \tag{19}$$

$$W = \begin{bmatrix} w(\mathbf{x} - \mathbf{x}_1) & 0 & \cdots & 0 \\ 0 & w(\mathbf{x} - \mathbf{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(\mathbf{x} - \mathbf{x}_n) \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{20}$$

$$\mathbf{f} = \left(T^{\tau,l+1}(\mathbf{x}_1), T^{\tau,l+1}(\mathbf{x}_2), \ldots, T^{\tau,l+1}(\mathbf{x}_{n_p}), 2\Delta t\dot{Q} + 2T^{\tau,l}(\mathbf{x}) + \Delta t \Delta T^{\tau,l} \right)^t \tag{21}$$

$$\mathbf{b}(\mathbf{x}) = \left(T^{\tau,l+1}(\mathbf{x}), \partial_x T^{\tau,l+1}(\mathbf{x}), \partial_y T^{\tau,l+1}(\mathbf{x}), \partial_{xx} T^{\tau,l+1}(\mathbf{x}), \partial_{xy} T^{\tau,l+1}(\mathbf{x}), \partial_{yy} T^{\tau,l+1}(\mathbf{x}) \right)^t \tag{22}$$

where τ and l denotes the iteration and time counters, respectively. Moreover, since the FPM method is an iterative method over each particle in Ω, therefore the stopping criteria used in the implemented algorithm is a relative error of the following form

$$\frac{\sum_{i=1}^{n_p} |T^{\tau+1,l}(\mathbf{x}_i) - T^{\tau,l}(\mathbf{x}_i)|}{\sum_{i=1}^{n_p} |T^{\tau+1,l}(\mathbf{x}_i)|} < \varepsilon \tag{23}$$

were the solution at each time step is obtained once we reach convergence, i.e., $T^{\tau,l+1}(\mathbf{x}_i) = T^{\tau+1}(\mathbf{x}_i)$ as $\tau \to \infty$ and this holds for $i = 1, \ldots, n_p$.

Download English Version:

https://daneshyari.com/en/article/7057783

Download Persian Version:

https://daneshyari.com/article/7057783

Daneshyari.com