Darrell W. Pepper[1]
Professor and Director
NCACM, UNLV,
Las Vegas, NV 89154
e-mail: darrell.pepper@unlv.edu

Xiuling Wang
Assistant Professor
Department of Mechanical Engineering,
Purdue University Calumet,
Hammond, IN 46323
e-mail: wangx@purduecal.edu

D. B. Carrington
Research Scientist
T-3 Fluid Dynamics and
Solid Mechanics Group,
LANL,
Los Alamos, NM 87545
e-mail: dcarring@lanl.gov

A Meshless Method for Modeling Convective Heat Transfer

A meshless method is used in a projection-based approach to solve the primitive equations for fluid flow with heat transfer. The method is easy to implement in a MATLAB *format. Radial basis functions are used to solve two benchmark test cases: natural convection in a square enclosure and flow with forced convection over a backward facing step. The results are compared with two popular and widely used commercial codes:* COMSOL*, a finite element-based model, and* FLUENT*, a finite volume-based model.*
[DOI: 10.1115/1.4007650]

Keywords: numerical heat transfer, meshless method, convection, modeling

Introduction

Numerical solutions for convective heat transfer problems have traditionally been obtained using finite difference (FDM), finite volume (FVM), and finite element methods (FEM). These methods involve using a mesh (or grid) to solve problems, and can become troublesome when discretizing complex domains with irregular boundaries. More recent efforts to reduce the burden of mesh generation include the use of the boundary element method (BEM). While reducing the dimension of a problem, e.g., from 2D to 1D, a mesh is still required in the BEM to discretize the domain boundary.

In the rapidly developing branch of meshless (also known as mesh-free or mesh reduction) numerical methods, there is no need to create a mesh, within the domain or on its boundary, and represents a promising technique to avoid meshing problems [1–3]. A number of mesh reduction techniques such as the dual reciprocity boundary element method [4], meshfree techniques including the dual reciprocity method of fundamental solutions [5], and meshfree local Petrov Galerkin methods [1,6] have been developed for transport phenomena and solution of the Navier–Stokes equations. A popular and simple technique that is applicable to a wide range of problems employs the use of radial basis functions (RBF) [7,8]. This paper focuses on the use of this simplest class of mesh-free methods using the RBF approach.

Meshless methods use a set of nodes scattered within the problem domain and boundaries to represent the domain geometry. These scattered nodes do not form a mesh, thus no information on the geometrical connections among the nodes is required. Generally meshless methods are characterized by the following features: the governing equations can be solved in their strong form, collocation is used to represent fields, complicated physics and geometry can be easily handled, the formulation is almost independent of the problem dimension, no integrations are needed, the method is efficient and accurate with low numerical diffusion, and the method is simple to learn and to code. Meshless methods have been in development for over 10 yr; this paper discusses the application of the RBF meshless method for solving problems involving fluid flow with heat transfer [9–12].

[1]Currently Distinguished Visiting Professor, U.S. Air Force Academy, CO 80840.

Manuscript received October 19, 2010; final manuscript received July 7, 2012; published online December 6, 2012. Assoc. Editor: Akshai Runchal.

Radial Basis Functions

There are many RBFs that have been suggested and applied in various numerical schemes. One of the most popular is the use of multiquadrics (MQ). Other RBFs are thin-plate splines, Gaussian, and Inverse MQ. RBFs are the natural generalization of univariate polynomial splines to a multivariate setting. More detailed discussions on RBFs and various formulations can be found in Kansa [3] and Buhman [7]. The main advantage of this type of approximation is that it works for arbitrary geometry with high dimensions and does not require a mesh. A RBF is a function whose value depends only on the distance from some center point. Using distance functions, RBFs can be easily implemented to reconstruct a plane or surface using scattered data in 2D, 3D or higher dimensional spaces.

MQ were initially proposed by Hardy [13] for analyzing data sets in geoscience. Franke [14] studied RBFs and found that MQs generally perform better than others for the interpolation of scattered data. The exponential convergence of MQ makes it superior to other RBFs [15]. In Kansa's method [3], a function is first approximated by an RBF, and its derivatives are then obtained by differentiating the RBF. Although RBFs were initially developed for multivariate data and function interpolation, their truly meshfree nature has motivated researchers to employ them in solving PDEs. Since multiquadrics are infinitely smooth functions, they are often chosen as the trial function (a common term used in finite element analysis for a variable approximation) for ϕ, i.e.,

$$\phi(\mathbf{r}_j) = \sqrt{r_j^2 + c^2} = \sqrt{(x - x_j)^2 + (y - y_j)^2 + c^2} \qquad (1)$$

where c is a shape parameter provided by the user.

The shape parameter, c, strongly influences the accuracy of the MQ–RBF method. The key factor in obtaining accurate results by the RBF method is the MQ matrix. The choice of the shape parameter c has been a topic of discussion in the community of RBF researchers. A general theoretical analysis of how the shape parameter c is associated with the accuracy of approximation is difficult. Once a critical value of c is reached, the error increases dramatically. The value of c and its association with the condition number are discussed later.

In Kansa's method [3], a variable (assume temperature) is expressed as an approximation in the form

(a) (b) (c)

Fig. 1 Irregular domain discretized using (*a*) three-noded triangular finite elements, (*b*) boundary element, and (*c*) arbitrary interior and boundary points using a meshless method

$$T(\mathbf{x}) = \sum_{j=1}^{N} \phi_j(\mathbf{x}) T_j \tag{2}$$

where $\{T_j\}$ are the unknown temperature values to be determined, $\mathbf{x}$ is the spatial vector denoting x, y, (and z if 3D), and $\phi_j(x) = \phi\|x - x_j\|$ is some form of RBF where $\{x_i\}$, $i = 1, \ldots, N_I$, are interior points and $\{x_i\}$, $i = N_{I+1}, \ldots, N$, are boundary points. Popular choices of RBFs include linear (r), cubic (r^3), MQ ($(r^2 + c^2)^{1/2}$), polyharmonic splines ($r^{2n+1}\log r$ in 2D, r^{2n+1} in 3D), and Gaussian ($\exp(-cr^2)$). The theory of RBF interpolation is discussed in Franke and Schaback [15]; Fasshauer [16] lists many RBFs and their derivatives.

To illustrate the application of the meshless method with RBF, consider the 2D steady-state heat conduction equation with constant properties

$$\begin{aligned} \nabla^2 T &= f(\mathbf{x}), \quad \mathbf{x} \in \Omega \\ T &= g(\mathbf{x}), \quad \mathbf{x} \in \Gamma \end{aligned} \tag{3}$$

where $\mathbf{x} = (x,y)$, $f(\mathbf{x})$ denotes source or sink terms and $g(\mathbf{x})$ denotes boundary values. Now approximate $T(\mathbf{x})$ assuming

$$T(\mathbf{x}) = \sum_{j=1}^{N} \phi(r_j) T_j \tag{4}$$

where r is defined as

$$r_j = \sqrt{(x - x_j)^2 + (y - y_j)^2} \tag{5}$$

To be more specific, we choose MQ as the basis function, i.e., modifying Eq. (5), we obtain Eq. (1), as previously denoted

$$\phi(\mathbf{r}_j) = \sqrt{r_j^2 + c^2} = \sqrt{(x - x_j)^2 + (y - y_j)^2 + c^2}$$

where c is the predetermined shape parameter. The derivatives are expressed as

$$\begin{aligned} \frac{\partial \phi}{\partial x} &= \frac{x - x_j}{\sqrt{r_j^2 + c^2}}, \quad \frac{\partial \phi}{\partial y} = \frac{y - y_j}{\sqrt{r_j^2 + c^2}} \\ \frac{\partial^2 \phi}{\partial x^2} &= \frac{(y - y_j)^2 + c^2}{\left(r_j^2 + c^2\right)^{3/2}}, \quad \frac{\partial^2 \phi}{\partial y^2} = \frac{(x - x_j)^2 + c^2}{\left(r_j^2 + c^2\right)^{3/2}} \end{aligned} \tag{6}$$

Substituting into Eq. (3), the equation set becomes

$$\begin{aligned} \sum_{j=1}^{N_I} \nabla^2 \phi(r_j) T_j &= f(\mathbf{x}) \\ \sum_{j=N_{I+1}}^{N} \phi(r_j) T_j &= g(\mathbf{x}) \end{aligned} \tag{7}$$

For the transient heat conduction equation, an implicit scheme can be used

$$\frac{T^{n+1} - T^n}{\Delta t} - \left(\frac{\partial^2 T^{n+1}}{\partial x^2} + \frac{\partial^2 T^{n+1}}{\partial y^2} \right) = f\left(x, y, T^n, \frac{\partial T^n}{\partial x}, \frac{\partial T^n}{\partial y} \right) \tag{8}$$

where Δt denotes the time step and superscript $n+1$ is the unknown (or new) value to be solved. The approximate solution can be expressed as

$$T(x, y, t^{n+1}) = \sum_{j=1}^{N} \phi_j(x, y) T_j^{n+1}$$

Substituting into Eq. (8), one obtains

$$\begin{aligned} &\sum_{j=1}^{N_I} \left(\frac{\phi_j}{\Delta t} - \left(\frac{\partial^2 \phi_j}{\partial x^2} + \frac{\partial^2 \phi_j}{\partial y^2} \right) \right)(x, y) T_j^{n+1} \\ &\quad = \frac{\phi_j}{\Delta t} T^n(x, y) + f(x, y, t^n, T^n(x, y), T_x^n(x, y), T_y^n(x, y)) \end{aligned} \tag{9a}$$

$$\sum_{j=N_{I+1}}^{N} \phi_j(x, y) T_j^{n+1} = g(x, y, t^{n+1}) \tag{9b}$$

where N_I is the total number of internal nodes and $N_{I+1}, \ldots, N$ are the boundary nodes. An $N \times N$ linear system of equations is generated for the unknown T^{n+1}. Note that

$$\begin{aligned} T^n(x, y) &= \sum_{j=1}^{N} \phi_j(x, y) T_j^n, \quad T_x^n(x, y) = \sum_{j=1}^{N} \frac{\partial \phi_j(x, y)}{\partial x} T_j^n, \\ T_y^n(x, y) &= \sum_{j=1}^{N} \frac{\partial \phi_j(x, y)}{\partial y} T_j^n \end{aligned}$$

Figure 1 shows an arbitrary domain discretized using three-noded triangular elements, boundary elements, and a meshless method. An internal mesh is required in the FEM (Fig. 1(*a*)) and linear elements are used along the boundary in the BEM (Fig. 1(*b*)). Both methods require the use of efficient matrix solvers to obtain values at the prescribed nodes, which can become resource limiting and computationally time consuming if the number of nodes becomes large. The meshless method, with arbitrarily distributed interior and boundary points, requires no mesh, as illustrated in Fig. 1(*c*).

The acceptability of a solution obtained by applying mesh-based numerical schemes largely depends on the quality of the mesh. Constructing a "good-quality" mesh is a difficult problem and the execution time can become high as the number of nodes increase. It turns out that the quality of the solution obtained by using a mesh-free method also depends on the way nodes are placed within the domain. The node placement problem for mesh-free applications is not well explored. A few results addressing node placement are reported in Refs. [17–20].

Mesh generation is an extensively explored problem and a variety of algorithms are available in the literature [19]. For complex

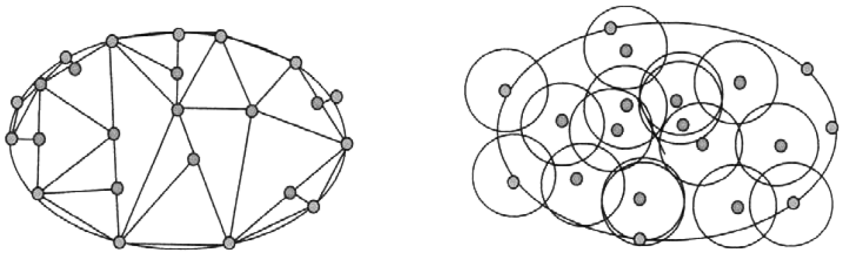

Fig. 2 Nodal placement within (*a*) patches and (*b*) clipping circles

geometrical domains, particularly in three dimensions, the quality of the generated mesh can diminish as the number of nodes increase, ultimately resulting in an inaccurate or failed solution. Although one can arbitrarily assign nodes in a meshless scheme, and eliminate the burdensome task of repeated mesh generation, solution accuracy can similarly deteriorate unless some care is taken to ensure adequate placement. Some of the well-known mesh-free techniques include partition of unity method [21], element-free Galerkin method [17], Voronoi based circle packing, and the biting method [20]. While the quality of a mesh in a grid-based numerical scheme can be measured, it is not very clear how to measure the quality of node placement for mesh-free applications. Li et al. [21] suggested four conditions that can enhance the solution obtained by a mesh-free approach. The conditions are defined in terms of the size of the *patches* (see Fig. 2). Circles, triangles, and rectangles are the simplest examples of patches. The first condition suggested in Ref. [21] is the location of nodes in each patch. A node placed near the center of a patch is considered as a good placement. The other three conditions are (i) the union of the patches must cover the whole domain; (ii) the size of the largest patch should be no more than a constant multiple of the size of the smallest patch; and (iii) the number of patches that cover a node should be small.

The biting method [21] is effective in addressing the above four conditions. In the biting method the node placement is performed by clipping the polygon by circles along the boundary. The clipping is performed outside to inside until the entire polygon is processed. The biting method terminates successfully for polygons with simple shapes. For complicated polygons, the biting method can create more than one residual component. Some techniques from computational geometry for node placement include circle packing and Voronoi diagrams [20]. Figure 2(*b*) shows the placement of nodes within a domain and the clipping using circles [22].

Our interest in this paper is not to discuss the placement of nodes in detail within a problem domain, but to illustrate the concept of the meshless approach utilizing simple placement. We have chosen to keep the problem domains simple and nodal placement aligned with the meshes produced by COMSOL and FLUENT. A more detailed algorithm describing the placement of nodes in a meshless approach is described in Gewali and Pepper [23].

One other issue with meshless methods that needs to be addressed is the condition number of the globally established matrix. It is well-known that a measure of numerical stability for an approximation can be evaluated from the condition number (this number is normally defined as the ratio of the largest and smallest singular values of the matrix, i.e., $\|A\|_2\|A^{-1}\|_2$ where A is the interpolation matrix and subscript 2 denotes the Euclidean norm). As one adds more interpolation points to improve the accuracy of the interpolant, the problem becomes more ill-conditioned. This is due to the decrease in separation distance and not to the increase in the number of data points [16]. Keeping the number of data points fixed (and separation distances), decreasing the value of c leads to an exponential increase in the condition number. Hence, increasing c can lead to an improvement in the condition number—to a point; Franke and Schaback [15] suggest that $c = 0.8(\sqrt{N})/D$, where D is the diameter of the smallest circle containing all the data points and N is the number of points. Preconditioning is another means to assist in limiting such instabilities; we generally found that a value for $c \approx 0.3$ worked well. Research continues in this area.

In order to illustrate the effectiveness of the meshless method, a simple heat conduction problem is examined. A two-dimensional plate is subjected to prescribed temperatures applied along each boundary [24], as shown in Fig. 3. The temperature at the midpoint (1,0.5) is used to compare the numerical solutions with the analytical solution. The analytical solution is given as

$$\theta(x,y) \equiv \frac{T - T_1}{T_2 - T_1} = \frac{2}{\pi}\sum_{n=1}^{\infty} \frac{(-1)^{n+1}+1}{n} \sin\left(\frac{n\pi x}{L}\right) \frac{\sinh(n\pi y/L)}{\sinh(n\pi W/L)}$$

which yields $\theta(1,0.5) = 0.445$, or $T(1,0.5) = 94.5\,^{\circ}\mathrm{C}$. Table 1 lists the final temperatures at the midpoint using a finite element method, a boundary element method, and a meshless method.

The number of elements and number of nodes are listed, along with the value for the temperature at the centroid. A uniform distribution of nodes was employed in this example. Additional examples comparing these numerical schemes are described in Pepper [25]. While all the methods gave reasonably close answers, the meshless method was very close to the exact solution.

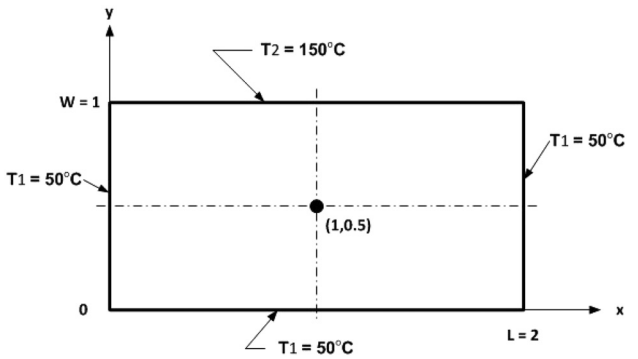

Fig. 3 Steady-state conduction in a two-dimensional plate

Table 1 Comparison of results for FEM, BEM, and meshless methods

Method	Midpoint (°C)	Elements	Nodes
Exact	94.512	0	0
FEM	94.605	256	289
BEM	94.471	64	65
Meshless	94.514	0	325

Method of Approach for Convective Flow

Governing Equations. Assuming incompressible laminar flow with convective heat transfer effects, the following scaling relations are used in the governing equations of momentum and energy

$$\mathbf{x}^* = \frac{\mathbf{x}}{L}, \quad \mathbf{V}^* = \frac{\mathbf{V}}{\alpha/L}, \quad p^* = \frac{p}{\rho \mathbf{V}^2}, \quad t^* = \frac{t}{\mathbf{L}/\mathbf{V}}, \quad T^* = \frac{T - T_c}{T_h - T_c} \tag{10}$$

where * represents nondimensional values. The Reynolds number, Rayleigh number, Prandtl number, and Peclet number are defined as

$$\mathrm{Re} = \frac{\rho V L}{\mu}, \quad \mathrm{Ra} = \frac{g\beta(T_h - T_c)L^3}{\alpha \upsilon}, \quad \mathrm{Pr} = \frac{\upsilon}{\alpha}, \quad \mathrm{Pe} = \frac{VL}{\alpha} \tag{11}$$

The nondimensional forms of the governing equations for conservation of mass, momentum, and energy (dropping the * for convenience) become

$$\nabla \bullet V = 0 \tag{12}$$

$$\frac{\partial V}{\partial t} + (V \bullet \nabla)V = \frac{1}{\rho}\nabla p + C_{\mathrm{vis}}\nabla^2 V + B \tag{13}$$

$$\frac{\partial T}{\partial t} + (V \bullet \nabla)T = C_T \nabla^2 T \tag{14}$$

where the body force in Eq. (13) is defined as $B = \mathrm{PrRa}T$ in the y-direction for natural convection problems. For forced convection cases, $B = 0$. The coefficients in Eqs. (12)–(14) are defined as (1) natural convection: $C_{\mathrm{vis}} = \mathrm{Pr}$, $C_T = 1$, and (2) forced convective flow: $C_{\mathrm{vis}} = 1/\mathrm{Re}$, $C_T = 1/\mathrm{Pe}$.

Projection Methodology. A simple local pressure–velocity coupling (LPVC) algorithm is used based on previous work undertaken by Spalding and his former students [26]. The method represents a local variant of already developed global solutions for coupled heat transfer and fluid flow problems. In order to solve such problems, the time dependent equations are employed. A simple finite difference approximation is adopted to calculate the time derivative. The Navier–Stokes equations are solved iteratively. The LPVC algorithm, where pressure correction is estimated from local mass continuity violation, is used to drive the intermediate velocity toward a divergence-free velocity. A four-step process is employed as follows:

Step 1. In the first step the velocity is estimated from the discretized form of the momentum equation. The calculated velocity, $\hat{\mathbf{V}}$, does not satisfy the mass continuity equation. In order to couple the mass continuity equation with the momentum equation, iteration is used where the first iteration velocity and pressure are set to

$$\mathbf{V}^m = \hat{\mathbf{V}}, \quad p^m = p_{\mathrm{o}}, \quad m = 1 \tag{15}$$

where m denotes the iteration index and p_{o} denotes pressure at time t_{o}. To project the velocity into the divergence-free space, a correction term is added

$$\nabla \cdot (\mathbf{V}^m + \mathbf{V}') = 0 \rightarrow \nabla \cdot \mathbf{V}^m = -\nabla \cdot \mathbf{V}' \tag{16}$$

where $\mathbf{V}'$ stands for velocity correction. Velocity correction is affected only by the pressure correction, i.e.,

$$\mathbf{V}' = -\frac{\Delta t}{\rho}\nabla p' \tag{17}$$

where p' stands for pressure correction.

A pressure correction Poisson equation is constructed by applying the divergence to the velocity correction equation, i.e.,

$$\nabla^2 p' = \frac{\rho}{\Delta t}\nabla \cdot \mathbf{V}^m \tag{18}$$

Step 2. Instead of solving the pressure correction Poisson equation with the proper pressure correction boundary conditions, the pressure correction is assumed to be linearly related to the Laplacian for pressure correction. Therefore, in the second step, the pressure correction is calculated

$$p' \approx L^2 \nabla^2 p' = L^2 \frac{\rho}{\Delta t}\nabla \cdot \mathbf{V}^m \tag{19}$$

where L is the reference length, as previously used in Eq. (10).

Step 3. In the third step, the pressure and velocity are corrected as

$$\begin{aligned} p^{m+1} &= p^m + \gamma p' \\ \mathbf{V}^{m+1} &= \mathbf{V}^m - \gamma \frac{\Delta t}{\rho}\nabla p' \end{aligned} \tag{20}$$

where γ stands for relaxation parameter. If the criterion

$$\nabla \cdot \mathbf{V}^{m+1} < \varepsilon_V \tag{21}$$

where ε_{V} is an error residual, is not met, then the iteration returns to the pressure correction equation; else the pressure–velocity iteration is completed. The calculation proceeds to the final step to ultimately solve for natural convection within a square enclosure or heated flow over a backward facing step.

Step 4. In the fourth step the temperature is calculated using the discretized form of the energy equation. The procedure is then restarted as the solution progresses to the next time step.

The momentum equation is discretized using a linear combination of RBFs and can be expressed in the form

$$\begin{aligned} \sum_{j=1}^{N_I} \hat{\mathbf{V}}_j^{n+1}\phi_j(x,y) = \sum_{j=1}^{N_I} \mathbf{V}_j^n \phi_j(x,y) + \Delta t \Bigg[& C_{\mathrm{vis}} \sum_{j=1}^{N_I} \mathbf{V}_j^n \nabla^2 \phi_j(x,y) \\ & - \sum_{j=1}^{N_I} p_j^n \nabla \phi_j(x,y) \\ & - \sum_{j=1}^{N_I} \mathbf{V}_j^n \phi_j(x,y) \sum_{j=1}^{N_I} \mathbf{V}_j^n \nabla \phi_j(x,y) \\ & + \sum_{j=1}^{N_I} B_j^n \phi_j(x,y) \Bigg] \end{aligned} \tag{22}$$

Discretized forms for the pressure and velocity correction equations are

$$\sum_{j=1}^{N_I} p_j'^{n+1}\phi_j(x,y) = \frac{L^2 \rho}{\Delta t} \sum_{j=1}^{N_I} \hat{\mathbf{V}}_j^n \nabla \phi_j(x,y) \tag{23}$$

$$\sum_{j=1}^{N_I} \mathbf{V}_j'^{n+1}\phi_j(x,y) = \frac{\Delta t}{\rho} \sum_{j=1}^{N_I} p_j'^n \nabla \phi_j(x,y) \tag{24}$$

Intermediate pressure and velocity correction equations can be written as

$$\sum_{j=1}^{N_I} p_j^{n+1}\phi_j(x,y) = \sum_{j=1}^{N} p_j^n \phi_j(x,y) + \beta \sum_{j=1}^{N} p_j'^n \phi_j(x,y) \tag{25}$$

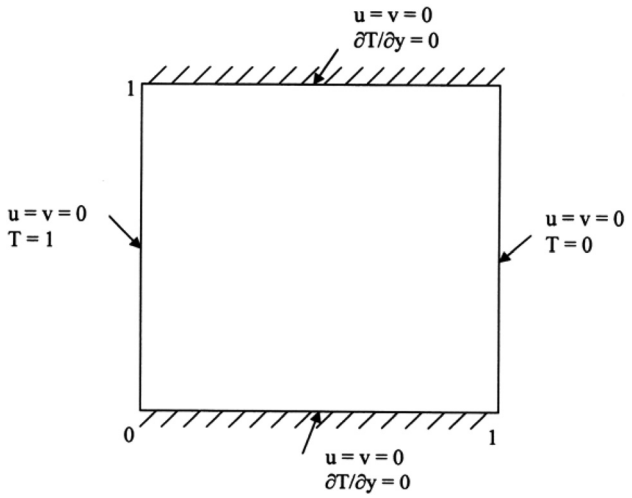

Fig. 4 Natural convection within an enclosed cavity

$$\sum_{j=1}^{N_I} \mathbf{V}_j^{n+1}\phi_j(x,y) = \sum_{j=1}^{N} \hat{\mathbf{V}}_j^{n}\phi_j(x,y) - \gamma\frac{\Delta t}{\rho}\sum_{j=1}^{N} p_j' n\nabla\phi_j(x,y) \quad (26)$$

The energy equation is also discretized using a linear combination of RBFs and is expressed as

$$\sum_{j=1}^{N_I} T_j^{n+1}\phi_j(x,y) = \sum_{j=1}^{N_I} T_j^{n}\phi_j(x,y) + \Delta t\left[C_T\sum_{j=1}^{N_I} T_j^{n}\nabla^2\phi_j(x,y) - \sum_{j=1}^{N_I}\mathbf{V}_j^{n}\phi_j(x,y)\sum_{j=1}^{N_I} T_j^{n}\nabla\phi_j(x,y)\right] \quad (27)$$

where Δt denotes the time step, superscript $n+1$ is the unknown value to be solved, and superscript n is the current known value. Steady-state is achieved if the convergence criteria are satisfied for all variables (generally $\varepsilon_{V,T} = 10^{-4}$). If the criteria are not met, the calculation returns to the first step.

Results

Natural Convection in a Square Enclosure. Natural convection in various geometries is a popular benchmark problem which has been studied extensively for over the past 50 yr. Many papers addressing this problem continue to appear in the literature using various numerical techniques. For natural convection in a differentially heated square cavity, simulation results are readily available in the literature for a wide range of Ra values. Much of the early work in modeling this benchmark problem is discussed in De Vahl Davis [27], who employed a finite difference scheme with a stream function/vorticity formulation.

The domain of the problem is a closed square entity filled with air ($\mathrm{Pr} = 0.71$) with differentially heated isothermal vertical walls and insulated horizontal walls. With constant initial temperature, pressure and velocity set to zero, steady-state is achieved as the solution progresses through time. The configuration of this case is shown in Fig. 4 where $0 \le x \le 1$, $0 \le y \le 1$. Distribution of interior and boundary nodes is shown in Fig. 5.

The boundary conditions are given as

$$\begin{aligned} &u = v = T = p = 0 \ (x, y; t = 0)\\ &u = v = 0 \ (\text{all walls})\\ &T = 0 \ (x = 1, y)\\ &T = 1 \ (x = 0, y)\\ &\partial T/\partial y = 0 \ (x, y = 0; 1) \end{aligned} \quad (28)$$

Results for various Rayleigh numbers (10^3–10^5) were compared with simulations obtained using COMSOL and FLUENT. Figure 6 shows the comparison of velocity vectors in the square cavity for $\mathrm{Ra} = 10^3$ using the meshless method with velocity vectors obtained from COMSOL and FLUENT. The results using the meshless approach compare closely to the expected (and widely published) recirculation patterns obtained with the two commercial codes for this simple case. For the case of $\mathrm{Ra} = 10^4$, velocity profiles on the vertical and horizontal lines through the cavity geometric center are plotted in Figs. 7(*a*) and 7(*b*) and again compare closely with the corresponding results from COMSOL and FLUENT. The recirculation patterns are also similar. In Fig. 8, simulation results of the temperature contours ranging from 0 to 1 with 0.1 as the interval for $\mathrm{Ra} = 10^4$ compare closely with results from COMSOL and FLUENT. Utilizing MATLAB, the meshless code ran quickly on a 32-bit PC notebook and the results instantly displayed using graphical statements embedded within the code structure. While both COMSOL and FLUENT are capable of displaying graphical images, results from all three codes were also saved to TECPLOT files to enable easy comparisons. Close comparisons were also obtained for the case $\mathrm{Ra} = 10^5$ (not shown).

The issue of mesh independent solutions needs to be briefly addressed. This is a particularly well examined test case that has been used for comparison studies by many for several decades, including sessions that have been conducted in various international conferences focused on accuracy utilizing this problem. The solution obtained by De Vahl Davis [27] is the usual reference. The test solutions shown in this first example used both

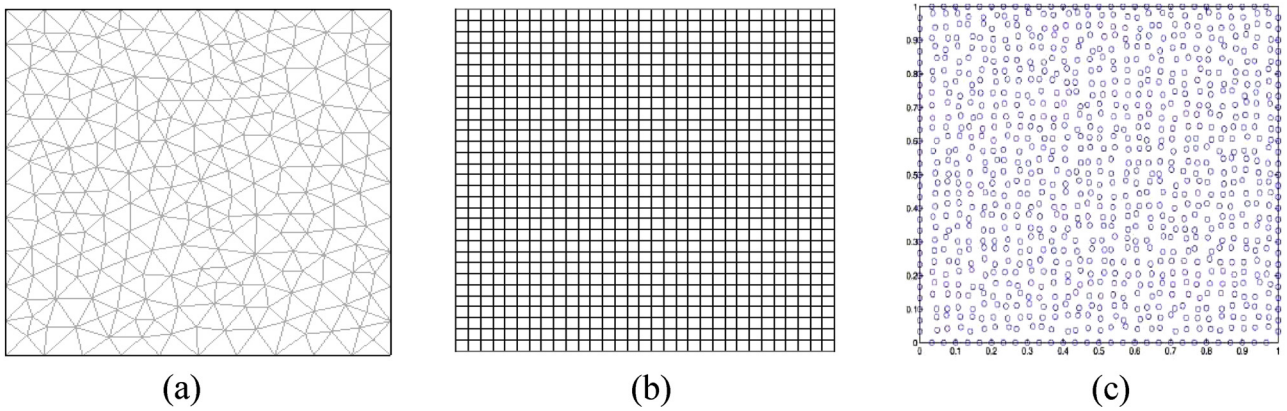

Fig. 5 31 × 31 point distribution for natural convection in a square cavity (*a*) COMSOL mesh, (*b*) FLUENT mesh, and (*c*) meshless node distribution

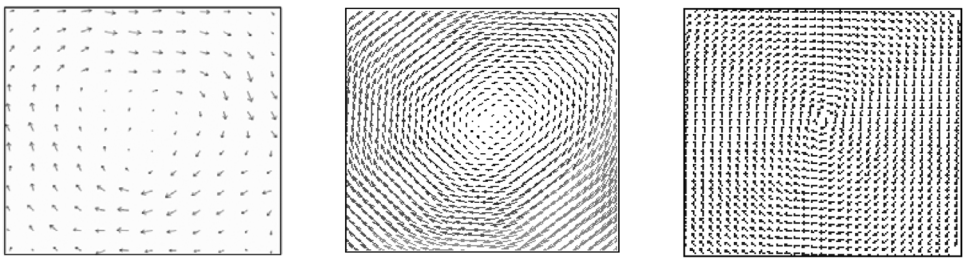

Fig. 6 Velocity vectors for natural convection for Ra = 10^3 in a square cavity using (*a*) COMSOL (*b*) FLUENT, and (*c*) meshless

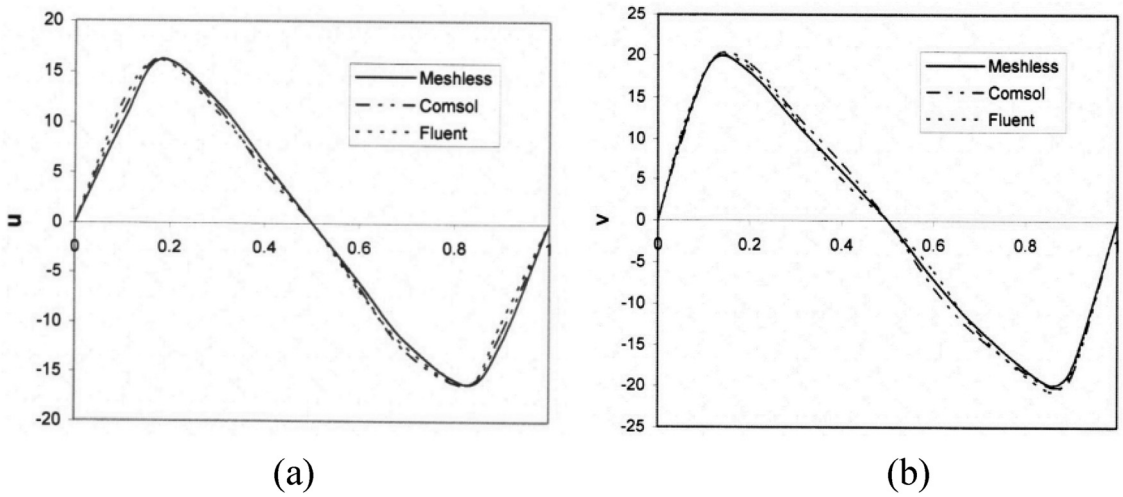

Fig. 7 Velocity profiles for Ra = 10^4 along (*a*) vertical and (*b*) horizontal central lines

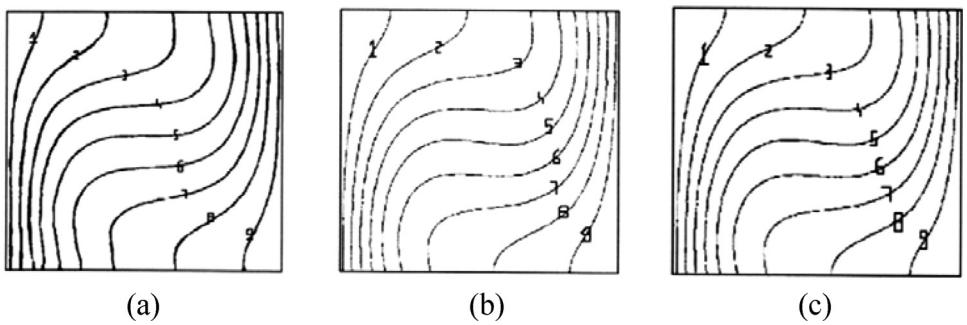

Fig. 8 Isotherms for natural convection in a square cavity for Ra = 10^4 using (*a*) meshless, (*b*) COMSOL, and (*c*) FLUENT (right vertical wall heated)

COMSOL and FLUENT to create solutions for a 31×31 mesh (or equivalent). Results obtained with the 31×31 mesh were subsequently compared with values obtained by Barakos et al. [28], who ultimately used an 80×80 mesh to establish grid independent solutions. The results were essentially the same, indicating that even for this lesser mesh density, the mesh was adequate. In this instance, we did not seek the ultimate, refined solutions that have been done in the past by others using both these commercial codes, but instead opted to create reasonably accurate solutions that could then be compared to the viability of the meshless technique. It is important to remember that the meshless method approximates field functions within an influence domain (patch), in lieu of an element or grid. Likewise, the nodes within each patch can either be uniformly distributed or randomly scattered. We tried both types of configurations and found little difference in the values. However, different RBFs have different accuracies. The MQ functions were found to be very satisfactory. As mentioned earlier, it is not the number of nodes that is critical but the separation distances that must be monitored (leading to ill-conditioning).

Flow With Forced Convection Over a Backward Facing Step. Two-dimensional flow over a backward facing step is also a well-known benchmark case that has been studied extensively over many years—the problem is simple to set up with expected results at various Reynolds numbers. Early research work for this problem focused on the fluid pattern, and many numerical simulations for this case can be found in the literature. Gartling [29] examined this problem for assessing outflow boundary conditions.

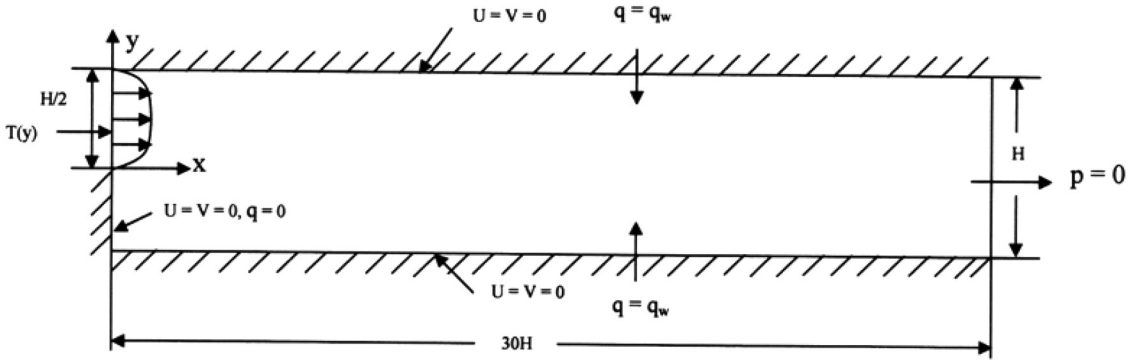

Fig. 9 Problem configuration for forced convection in a backward facing step

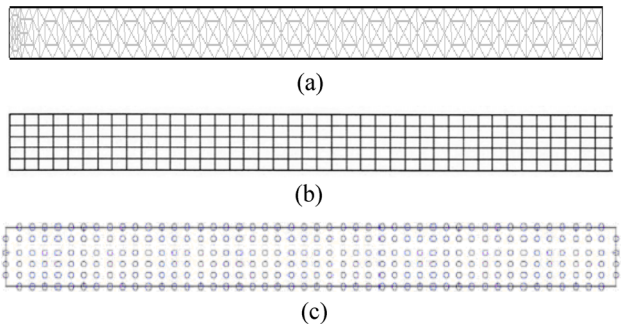

Fig. 10 Typical mesh for backward facing step solution: (*a*) COMSOL mesh of 388 elements, (*b*) FLUENT mesh of 284, and (*c*) 284 nodes for the meshless method

In 1992, Blackwell and Pepper [30] suggested flow over the backward facing step with heat transfer as an ASME benchmark test problem. Numerical simulations from twelve different contributors were presented.

The boundary conditions for this problem are described as

For Inlet Flow

$$u(y) = \begin{cases} 0, & \text{for } 0 \le y \le 0.5 \\ 8y(1-2y) & \text{for } 0.5 \le y \le 1 \end{cases}$$

$$v(y) = 0$$

$$T(y) = \left[1-(4y-1)^2\right]\left[1-\frac{1}{5}(4y-1)^2\right] \quad \text{for } 0 \le y \le 0.5$$

$$\frac{\partial T(y)}{\partial x} = 0 \quad \text{for } 0.5 \le y \le 1 \tag{29}$$

On Upper and Lower Walls

$$u(y) = v(y) = 0$$

$$\nabla T \cdot \mathbf{n} = \frac{32}{5} \tag{30}$$

where $\mathbf{n}$ is the outward unit vector normal to the domain boundary.

For Outlet Flow

$$p = 0 \tag{31}$$

Figure 9 shows the configuration for forced convection over the 2D backward facing step. A constant heat flux is introduced into the upper and lower channel walls immediately downstream of the step. The purpose of this particular set of conditions is to evaluate the change in temperature along the upper and lower surfaces as the fluid at the inlet proceeds down the channel. Ideally, the temperature gradient approaches a constant value with increasing horizontal distance from the step (the temperature gradient, $\partial T/\partial y$, becomes constant throughout the fluid). In this study, $\mathrm{Re} = 800$ and $\mathrm{Pr} = 0.71$, based on the values used in Refs. [29,30]. Distribution of interior nodes and boundary nodes are shown in Fig. 10.

Velocity profiles at $x = 7$ and at $x = 15$ are shown in Fig. 11. The meshless technique produces velocity profiles comparable to values plotted from COMSOL and FLUENT at the same locations. Temperature profiles at $x = 7$ and at $x = 15$ are similarly shown in Fig. 12. The comparisons of temperature also appear to be in good agreement. Temperature contours within the channel downstream of the step are shown in Fig. 13. The isotherm profiles are similar for all three model results [31]. Extensive application of the

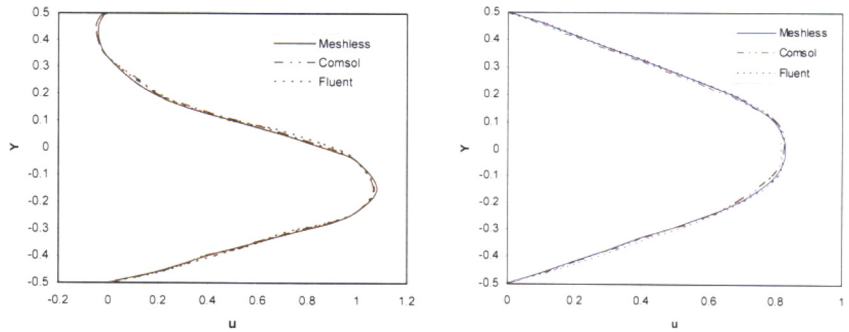

Fig. 11 Velocity profiles for Re = 800

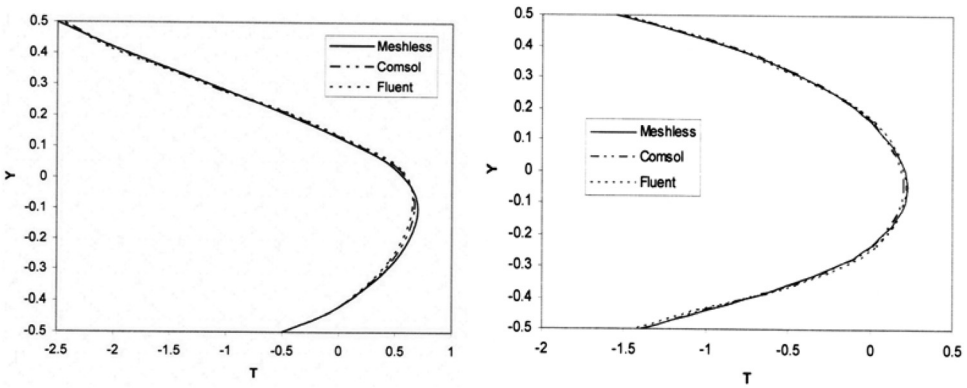

Fig. 12 Temperature profiles for Re = 800

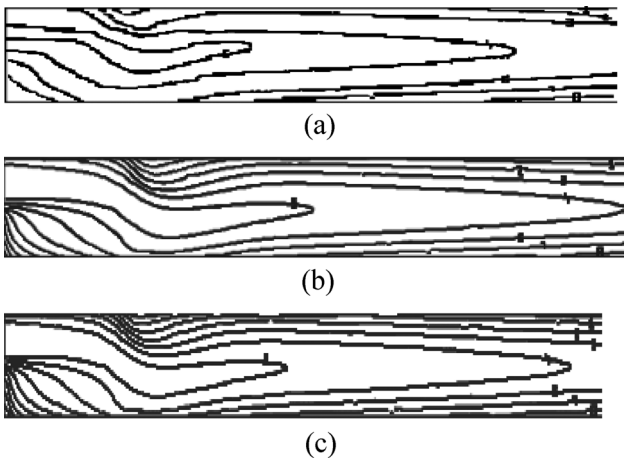

Fig. 13 Isotherms for backward step flow using (a) meshless, (*b*) COMSOL, and (*c*) FLUENT

meshless method for more complex flow configurations can be found in Sarler et al. [22] and Zahab et al. [32].

Conclusions

The governing equations for transient fluid flow with convective heat transfer are solved using RBFs. The use of RBFs allows velocity and temperature variables and their derivatives to be easily established and solved with a general algorithm. To illustrate the application of the technique, natural convection within a square enclosure and forced convection over a backward facing step are both solved. Results are compared with well-known benchmark solutions. Based upon the comparison studies, the meshless method using RBFs appears effective in simulating fluid flow with convection, and has the ability to be applied to a wide range of problems. The number of points required to obtain comparable accuracy is less than mesh-based methods and a formal mesh structure is not required. The meshless method is an economical alternative for solving problems involving fluid flow with or without heat transfer.

Numerical implementation was done in MATLAB. Using a one-step pressure correction, the numerical algorithm needs only a small number of calculations per iteration cycle. The overall numerical procedure yields an algorithm that is fast and robust. Good agreement was achieved when comparing model results from COMSOL and FLUENT for these two simple convective heat transfer problems. Additional effort is underway to examine more complex and much larger problem domains, including 3D simulations, over a wide class of problems. Recent results using the technique to model bioengineering flows and species transport, as well as environmental flows, appear promising.

Nomenclature

$\mathbf{B}$ = 0; PrRaT
C_{vis} = Pr; 1/Re
C_{T} = 1; 1/Pe
c = shape parameter
$\mathbf{g}$ = acceleration due to gravity
L = reference length
N = total number of nodes
N_{I} = number of internal nodes
$N_{\text{I+1}}$ = number of boundary nodes
$\mathbf{n}$ = unit normal vector
p = pressure
p' = pressure correction
p_{o} = pressure at $t = 0$
Pe = Peclet number (Pe = VL/α)
Pr = Prandtl number (Pr = ν/α)
Re = Reynolds number (Re = $\rho VL/\mu$)
Ra = Rayleigh number (Ra = $g\beta[T_{\text{h}} - T_{\text{c}}]L^3/\alpha\nu$)
r_j = radial dimension (x,y)
$\mathbf{r}_j$ = radial basis function (vector)
t = time
T = temperature
T_{c} = cold (or reference) temperature
T_{h} = hot temperature (heated wall)
u = horizontal (x) velocity
v = vertical (y) velocity
$\mathbf{V}$ = velocity vector
$\mathbf{V}^m$ = iterated intermediate value of velocity
$\mathbf{V}'$ = velocity correction
$\hat{\mathbf{V}}$ = calculated velocity (not mass consistent)
$\mathbf{x}$ = spatial vector (x,y,z)
x = horizontal direction
y = vertical direction
α = thermal diffusivity
β = coefficient of thermal expansion
γ = relaxation parameter
Γ = domain boundary
Δt = time step
ε_{T} = error for temperature residual
ε_{V} = error for velocity residual
θ = $(\text{T} - \text{T}_1)/(\text{T}_2 - \text{T}_1)$ in 2D heat conduction test
μ = dynamic viscosity
ν = kinematic viscosity
ρ = density
ϕ = trial function

Ω = problem domain
∇ = gradient operator
∇^2 = Laplacian operator
$*$ = denotes nondimensional

References

[1] Atluri, S. N., and Shen, S., 2002, "The Meshless Local Petrov-Galerkin (MLPG) Method: A Simple & Less-Costly Alternative to the Finite Element and Boundary Element Methods," Comput. Model. Eng. Sci., **3**, pp. 11–52.
[2] Chen, W., 2002, "New RBF Collocation Schemes and Kernel RBFs With Applications," Lect. Notes Comput. Sci. Eng., **26**, pp. 75–86.
[3] Kansa, E. J., 1990, "Multiquadrics—A Scattered Data Approximation Scheme With Application to Computational Fluid Dynamics, Part I," Comput. Math. With Appl., **19**, pp. 127–145.
[4] Sarler, B., and Kuhn, G., 1999, "Primitive Variables Dual Reciprocity Boundary Element Method Solution of Incompressible Navier-Stokes Equations," Eng. Anal. Boundary Elem., **23**, pp. 443–455.
[5] Sarler, B., 2002, "Towards a Mesh-Free Computation of Transport Phenomena," Eng. Anal. Boundary Elem., **26**, pp. 731–738.
[6] Lin, H., and Atluri, S. N., 2001, "Meshless Local Petrov Galerkin Method (MLPG) for Convection-Diffusion Problems," Comput. Model. Eng. Sci., **1**, pp. 45–60.
[7] Buhman, M. D., 2000, *Radial Basis Functions*, Cambridge University Press, Cambridge.
[8] Mai-Duy, N., and Tran-Cong, T., 2005, "An Efficient Indirect RBFN-Based Method for Numerical Solution of PDEs," Numer. Methods Partial Differ. Equ., **21**, pp. 770–790.
[9] Patankar, S. V., and Spalding, D. B., 1972, "Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows," Int. J. Heat Mass Transfer, **15**(10), pp.1787–8061.
[10] Carrington, D. B., and Pepper, D. W., 2002, "Convective Heat Transfer Downstream of a 3-D Backward-Facing Step," Numer. Heat Transfer, Part A, **41**(6–7), pp. 555–578.
[11] Wang, X., and Pepper, D. W., 2007, "Application of an hp-Adaptive FEM for Solving Thermal Flow Problems," J. Thermophys. Heat Transfer, **21**(1), pp. 190–198.
[12] Carrington, D. B., Wang, X., and Pepper, D. W., 2010, "An h-Adaptive Finite Element Method for Turbulent Heat Transfer," Comput. Model. Eng. Sci., **61**(1), pp. 23–44.
[13] Hardy, R. L., 1971, "Multiquadric Equations of Topography and Other Irregular Surfaces," J. Geophys. Res., **76**, pp. 1905–1915.
[14] Franke, R., 1979, "A Critical Comparison of Some Methods for Interpolation of Scattered Data," Naval Postgraduate School, Report No. TR NPS-53-79-003.
[15] Franke, C., and Schaback, R., 1998, "Solving Partial Differential Equations Using Radial Basis Functions," Appl. Math. Comput., **3**, pp. 73–82.
[16] Fasshauer, G. E., 2007, *Meshfree Approximation Methods With MATLAB* (Volume 6 of Interdisciplinary Mathematical Sciences), World Scientific Pub. Co., Singapore.
[17] Atluri, S. N., and Zhu, T., 1998, "A New Mesh-Less Local Petrov-Galerkin Approach in Computational Mechanics," Comput. Mech., **22**, pp. 117–127.
[18] Balachandran, G. R., Rajagopal, A., and Sivakumar, S. M., 2008, "Mesh Free Galerkin Method Based on Natural Neighbors and Conformal Mapping," Comput. Mech., **42**(6), pp. 885–905.
[19] Bern, M., and Eppstein, D., 1992, "Mesh Generation and Optimal Triangulation," *Computing in Euclidean Geometry*, Vol. 1, D. Z. Du and F. K. Hwang, eds., World Scientific Publishing Co., Singapore, pp. 23–90.
[20] Choi, Y., and Kim, S. J., 1999, "Node Generation Scheme for the Mesh-Less Method by Voronoi Diagram and Weighted Bubble Packing," 5th US National Congress on Computational Mechanics, Boulder, CO.
[21] Li, X. Y., Teng, S. H., and Ungor, A., 2000, "Point Placement for Meshless Methods Using Sphere Packing and Advancing Front Methods," ICCES'00, Los Angeles, CA.
[22] Sarler, B., Lorbiecka, A. Z., and Vertnik, R., 2010, "Heat and Mass Transfer Problems in Continuous Casting of Steel: Multiscale Solution by Meshless Method," International Conference on Fluid Dynamics, Cairo, Egypt, Dec. 16–19, Vol. 10.
[23] Gewali, L., and Pepper, D. W., 2010, "Adaptive Node Placement for Mesh-Free Methods," ICCES 10, Las Vegas, NV, Mar. 28–Apr. 1.
[24] Incropera, F. P., and DeWitt, D. P., 2002, *Fundamentals of Heat and Mass Transfer*, 5th ed., J. Wiley & Sons, New York.
[25] Pepper, D. W., 2010, "Meshless Methods for PDEs," Scholarpedia, **5**(5), p. 9838.
[26] Spalding, D. B., 1985, "Numerical Simulation of Natural Convection in Porous Media," Conference on Natural Convection: Fundamentals and Applications, Hemisphere, Washington, DC, pp. 655–673.
[27] De Vahl Davis, G., 1983, "Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution," Int. J. Numer. Methods Fluids, **3**, pp. 249–264.
[28] Barakos, G., Mitsoulis, E., and Assimacopoulos, D., 1994, "Natural Convection Flow in a Square Cavity Revisited: Laminar and Turbulent Models With Wall Functions," Int. J. Numer. Methods Fluids, **18**, pp. 695–719.
[29] Gartling, D. K., 1990, "A Test Problem for Outflow Boundary Conditions—Flow Over a Backward-Facing Step," Int. J. Numer. Methods Fluids, **11**, pp. 953–967.
[30] Blackwell, B. F., and Pepper, D. W., 1992, "Benchmark Problems for Heat Transfer Codes," ASME Winter Annual Meeting (HTD), **21**, pp. 190–198.
[31] Kalla, N. K., 2007, "Solution of Heat Transfer and Fluid Flow Problems Using Meshless Radial Basis Function Method," M.S. thesis, UNLV, Las Vegas, NV.
[32] Zahab, Z. E., Divo, E., and Kassab, A. J., 2009, "A Localized Collocation Meshless Method (LCMM) for Incompressible Flows CFD Modeling With Applications to Transient Hemodynamics," Eng. Anal. Boundary Elem., **33**, pp. 1045–1061.