INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng (2008)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2458

SPH with the multiple boundary tangent method

M. Yildiz[1], R. A. Rook[2] and A. Suleman[2,*,†]

[1]*Faculty of Engineering and Natural Sciences, Sabanci University, 34956 Tuzla, Istanbul, Turkey*
[2]*Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada*

SUMMARY

In this article, we present an improved solid boundary treatment formulation for the smoothed particle hydrodynamics (SPH) method. Benchmark simulations using previously reported boundary treatments can suffer from particle penetration and may produce results that numerically blow up near solid boundaries. As well, current SPH boundary approaches do not properly treat curved boundaries in complicated flow domains. These drawbacks have been remedied in a new boundary treatment method presented in this article, called the *multiple boundary tangent* (MBT) approach. In this article we present two important benchmark problems to validate the developed algorithm and show that the multiple boundary tangent treatment produces results that agree with known numerical and experimental solutions. The two benchmark problems chosen are the lid-driven cavity problem, and flow over a cylinder. The SPH solutions using the MBT approach and the results from literature are in very good agreement. These solutions involved solid boundaries, but the approach presented herein should be extendable to time-evolving, free-surface boundaries. Copyright © 2008 John Wiley & Sons, Ltd.

Received 24 December 2007; Revised 23 July 2008; Accepted 28 July 2008

KEY WORDS: smoothed particle hydrodynamics; incompressible SPH; lid-driven cavity; projective methods; flow over a cylinder

1. INTRODUCTION

Smoothed particle hydrodynamics (SPH) is an adaptive, meshfree, Lagrangian numerical approximation technique used for modelling physical problems. Unlike Eulerian computational techniques such as the finite volume and finite difference methods, SPH does not require a grid, as derivatives are approximated using a kernel function. Each 'particle' in the domain can be associated with

*Correspondence to: A. Suleman, Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada.
†E-mail: suleman@uvic.ca

Contract/grant sponsor: Natural Sciences and Engineering Research Council (NSERC); contract/grant number: CRDPJ 261287-02

one discrete physical object, or it may represent a macroscopic part of the continuum [1]. The continuum is represented by an ensemble of particles each carrying mass, momentum, and other hydrodynamic properties. Although originally proposed to handle cosmological simulations [2, 3] SPH has become increasingly generalized to handle many types of fluid and solid mechanics problems [4–7]. SPH advantages include relatively easy modelling of complex material surface behavior, as well as simple implementation of more complicated physics, such as solidification [8], crystal growth [9], and free-surface flow [10, 11].

A survey of the SPH literature reveals that almost all reported benchmark flow simulations are for extremely low Reynolds numbers [12–14], in the range $0.025 < Re \lesssim 1$. Morris *et al*. [15] have simulated one-dimensional Couette and Poiseuille flows, as well as a two-dimensional flow over a cylinder, all for low Reynolds numbers. Each of the preceding results used the weakly compressible formulation of SPH. Some examples for benchmark problems with higher Reynolds numbers include the works of Chaniotis *et al*. [16] and Takeda *et al*. [17]. Chaniotis *et al*. successfully simulated the lid-driven cavity problem using weakly compressible SPH with $\mathbf{N} = 63\,000$ particles for a Reynolds number $Re = 400$. Takeda *et al*. presented the results of two-dimensional compressible flow over a cylindrical obstacle for Reynold numbers between 6 and 55.

In this article, we simulate two benchmark flow problems for Reynolds numbers from 10 to 100, greater than those found in the literature for benchmark problems solved with incompressible, 'projective' SPH. The selected problems are the two-dimensional lid-driven cavity, and flow over an obstacle (cylinder) simulations. The incompressible SPH approach has also been utilized by Cummins and Rudman [18], and Shao and Lo [19]. Shao and Lo further extended the application of incompressible SPH to deformable free-surface flow by presenting the solution for a dam-breaking problem. Our incompressible SPH approach differs from theirs in terms of a new solid boundary treatment.

The SPH technique computes discrete particle properties using a smoothing, *kernel* distribution function to account for the effects of surrounding particles. It is assumed that the properties characteristic of the particle of interest are influenced by all other particles in the global domain. However, one approximation of SPH is to include only the effects of nearby *neighbor* particles, within a smoothing radius denoted κh, where κ is a coefficient associated with the particular kernel function. The length h defines the *support domain* of the particle of interest (i.e. a localized domain over which the kernel will be non-zero). Throughout the present simulations, the compactly supported two-dimensional quintic spline kernel [20]

$$W(r_{\mathbf{ij}}, h) = \frac{7}{478\pi h^2} \begin{cases} \left(3 - \dfrac{r_{\mathbf{ij}}}{h}\right)^5 - 6\left(2 - \dfrac{r_{\mathbf{ij}}}{h}\right)^5 + 15\left(1 - \dfrac{r_{\mathbf{ij}}}{h}\right)^5 & \text{if } 0 \leqslant r_{\mathbf{ij}} < h \\ \left(3 - \dfrac{r_{\mathbf{ij}}}{h}\right)^5 - 6\left(2 - \dfrac{r_{\mathbf{ij}}}{h}\right)^5 & \text{if } h \leqslant r_{\mathbf{ij}} < 2h \\ \left(3 - \dfrac{r_{\mathbf{ij}}}{h}\right)^5 & \text{if } 2h \leqslant r_{\mathbf{ij}} < 3h \\ 0 & \text{if } r_{\mathbf{ij}} \geqslant 3h \end{cases} \tag{1}$$

was employed, where $r_{\mathbf{ij}}$ is the magnitude of the distance between neighboring particles $\mathbf{i}$ and $\mathbf{j}$. Another common SPH kernel is the cubic spline kernel, which is computationally less expensive due to its lower order.

It was observed that for fluid flow simulations at low Reynolds numbers ($Re = 1$), the cubic spline kernel produces inaccurate velocity fields and in turn inaccurate density fields, while the

DOI: 10.1002/nme

higher-order quintic spline kernel of Equation (1) remains stable. The one-dimensional benchmark Couette and Poiseuille flow simulations did not reveal these instabilities. However, the kernel instabilities did become obvious during the two-dimensional lid-driven cavity simulations. These instability problems were also reported by Morris *et al.*

For clarity, it is worthwhile taking a moment to explicitly state the notational conventions that will be used throughout this article. All vector quantities will be written using the index notation, with latin indices denoting the components. These components will always be written as superscripts to distinguish them from particle identifiers. These superscripts do not represent a contravariant nature, as in tensorial notation. Rather, the superscript placement is convenient to distinguish from particle label indices. As well, throughout this article the Einstein summation convention is employed, where any repeated component index is summed over the range of the index. Latin boldface indices $(\mathbf{i}, \mathbf{j})$ will be used to denote particles and will always be placed as subscripts that are not summed, unless indicated with a summation symbol. For example, the n-dimensional vector denoting the position of particle $\mathbf{i}$ is written as

$$\vec{\mathbf{r}}_{\mathbf{i}} = r_{\mathbf{i}}^{k} \vec{\mathbf{e}}_k \quad \text{for } k = 1, 2, \ldots, n \tag{2}$$

where $\vec{\mathbf{e}}_k$ is a basis vector along the kth component direction. As well, we will employ the concise $()_{\mathbf{ij}}$ difference notation $\vec{\mathbf{r}}_{\mathbf{ij}} \equiv \vec{\mathbf{r}}_{\mathbf{i}} - \vec{\mathbf{r}}_{\mathbf{j}}$ where $r_{\mathbf{ij}} \equiv \|\vec{\mathbf{r}}_{\mathbf{i}} - \vec{\mathbf{r}}_{\mathbf{j}}\|$ denotes the magnitude of the distance between particles $\mathbf{i}$ and $\mathbf{j}$.

2. SMOOTHED PARTICLE HYDRODYNAMICS

The three-dimensional Dirac-delta function $\delta^3(r_{\mathbf{ij}})$ is the starting point for the SPH approximation technique. This function satisfies the identity

$$f(\vec{\mathbf{r}}_{\mathbf{i}}) = \iiint_{\Omega} f(\vec{\mathbf{r}}_{\mathbf{j}}) \delta^3(r_{\mathbf{ij}}) \, \mathrm{d}^3 \vec{\mathbf{r}}_{\mathbf{j}} \tag{3}$$

where $\mathrm{d}^3 \vec{\mathbf{r}}_{\mathbf{j}}$ is a differential volume element and Ω represents the total bounded volume of the domain. The fundamental approximation of SPH is to replace the Dirac-delta function with the even kernel function $W(r_{\mathbf{ij}}, h)$. We then write the fundamental SPH approximation in three dimensions as

$$f(\vec{\mathbf{r}}_{\mathbf{i}}) \approx \langle f(\vec{\mathbf{r}}_{\mathbf{i}}) \rangle \equiv \iiint_{\Omega} f(\vec{\mathbf{r}}_{\mathbf{j}}) W(r_{\mathbf{ij}}, h) \, \mathrm{d}^3 \vec{\mathbf{r}}_{\mathbf{j}} \tag{4}$$

where $\langle f(\vec{\mathbf{r}}_{\mathbf{i}}) \rangle$ is the kernel approximation of the scalar field $f(\vec{\mathbf{r}}_{\mathbf{i}})$ at particle $\mathbf{i}$.

2.1. Spatial derivatives and particle approximation in SPH

In order to determine the SPH approximation for the gradient of a scalar function, we make the substitution $f(\vec{\mathbf{r}}_{\mathbf{j}}) \to \partial f(\vec{\mathbf{r}}_{\mathbf{j}}) / \partial x_{\mathbf{j}}^{l}$ in Equation (4) to produce

$$\left\langle \frac{\partial f(\vec{\mathbf{r}}_{\mathbf{i}})}{\partial x_{\mathbf{i}}^{l}} \right\rangle = \iiint_{\Omega} \frac{\partial f(\vec{\mathbf{r}}_{\mathbf{j}})}{\partial x_{\mathbf{j}}^{l}} W(r_{\mathbf{ij}}, h) \, \mathrm{d}^3 \vec{\mathbf{r}}_{\mathbf{j}}$$

DOI: 10.1002/nme

where the spatial differentiations take place referencing $x_{\mathbf{i}}^l$ coordinates and $x_{\mathbf{j}}^l$ coordinates, respectively. Upon integrating by parts and noting that $\partial W(r_{\mathbf{ij}},h)/\partial x_{\mathbf{i}}^l = -\partial W(r_{\mathbf{ij}},h)/\partial x_{\mathbf{j}}^l$ for a constant smoothing length h, it can be shown that

$$\left\langle \frac{\partial f(\vec{\mathbf{r}}_{\mathbf{i}})}{\partial x_{\mathbf{i}}^l} \right\rangle = \int\int\int_{\Omega} f(\vec{\mathbf{r}}_{\mathbf{j}}) \frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_{\mathbf{i}}^l} \mathrm{d}^3\vec{\mathbf{r}}_{\mathbf{j}} \tag{5}$$

for all interior particles **i**. Using a Taylor series expansion and the properties of a second-rank isotropic tensor, the SPH approximation for the gradient of a scalar function can also be introduced as

$$\left\langle \frac{\partial f(\vec{\mathbf{r}}_{\mathbf{i}})}{\partial x_{\mathbf{i}}^l} \right\rangle = \int\int\int_{\Omega} [f(\vec{\mathbf{r}}_{\mathbf{j}}) - f(\vec{\mathbf{r}}_{\mathbf{i}})] \frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_{\mathbf{i}}^l} \mathrm{d}^3\vec{\mathbf{r}}_{\mathbf{j}} \tag{6}$$

The SPH approximation used for the divergence of a vector function $f^k(\vec{\mathbf{r}}_{\mathbf{i}})$ is an obvious extension of Equation (6) and is obtained by replacing $f \to f^k(\vec{\mathbf{r}}_{\mathbf{i}})$.

The SPH form of the second spatial derivative of a vector field $\partial^2 f^k(\vec{\mathbf{r}}_{\mathbf{i}})/\partial x_{\mathbf{i}}^l \partial x_{\mathbf{i}}^n$ is derived in Appendix A and is given in three dimensions as

$$\left\langle \frac{\partial^2 f^k(\vec{\mathbf{r}}_{\mathbf{i}})}{\partial x_{\mathbf{i}}^l \partial x_{\mathbf{i}}^n} \right\rangle = \int\int\int_{\Omega} [f^k(\vec{\mathbf{r}}_{\mathbf{i}}) - f^k(\vec{\mathbf{r}}_{\mathbf{j}})] \frac{1}{r_{\mathbf{ij}}} \left(\frac{5 r_{\mathbf{ij}}^l r_{\mathbf{ij}}^n}{r_{\mathbf{ij}}^2} - \delta^{ln} \right) \frac{\partial W(r_{\mathbf{ij}},h)}{\partial r_{\mathbf{ij}}} \mathrm{d}^3\vec{\mathbf{r}}_{\mathbf{j}} \tag{7}$$

where δ^{ln} is the Kronecker delta.

The approximation for the Laplacian comes by contracting on indices l and n in Equation (7) to produce

$$\left\langle \frac{\partial^2 f^k(\vec{\mathbf{r}}_{\mathbf{i}})}{\partial x_{\mathbf{i}}^n \partial x_{\mathbf{i}}^n} \right\rangle = 2 \int\int\int_{\Omega} [f^k(\vec{\mathbf{r}}_{\mathbf{i}}) - f^k(\vec{\mathbf{r}}_{\mathbf{j}})] \frac{1}{r_{\mathbf{ij}}} \frac{\partial W(r_{\mathbf{ij}},h)}{\partial r_{\mathbf{ij}}} \mathrm{d}^3\vec{\mathbf{r}}_{\mathbf{j}} \tag{8}$$

which is commonly used in the literature. The Laplacian operator in Equation (8) is reported [15] to be less sensitive to particle disorder compared with the more direct Laplacian, which uses the second spatial derivative of the kernel.

In the above equations, the SPH approximations are written for a continuous distribution. If, however, we recognize that these integrations will be carried out over all **N** discrete particles within the domain, the discrete SPH particle approximation can be obtained by replacing the integration with summation over particle **j** to produce the SPH approximation of a field property $f(\vec{\mathbf{r}})$ at particle **i** in terms of all other interacting particles **j**, and a representative particle volume written in terms of a particle mass $m_{\mathbf{j}}$ and particle density $\rho_{\mathbf{j}}$.

2.2. *Incompressible SPH*

The governing equations used to solve the fluid problems in this article are the mass and linear momentum balance equations, given in component form as

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho \left(\frac{\partial v^k}{\partial x^k} \right), \quad \frac{\mathrm{d}v^k}{\mathrm{d}t} = -\frac{1}{\rho} \frac{\partial p}{\partial x^k} + \mu \nabla^2 v^k + F_B^k \tag{9}$$

DOI: 10.1002/nme

where ρ is the fluid density, p is the absolute pressure, μ is the dynamic fluid viscosity, and v^k and F_B^k are the components of the fluid velocity and the body force terms, respectively. In the present simulations, the fluid is assumed to be incompressible and Newtonian.

There are two popular approaches in SPH for numerically solving the balance of linear momentum equations, namely the weakly compressible assumption and the incompressible SPH assumption. In the former case, the pressure field is computed directly from an artificially imposed state equation. However, this approach requires the use of a sound speed parameter and produces extremely small time steps that are problematic for simulations involving long simulation times. In the latter case, a projective method [18] is used to determine the pressure field. The projection method is based on the principle of Hodge decomposition, which states that any vector field $\bar{f}^k$ can be broken into a divergence-free part f^k plus the gradient of an appropriate scalar potential ϕ so that

$$\bar{f}^k = f^k + \frac{\partial \phi}{\partial x^k} \tag{10}$$

with $\partial f^k/\partial x^k = 0$. By differentiation and contraction on the above equation, we obtain

$$\frac{\partial \bar{f}^k}{\partial x^k} = \frac{\partial^2 \phi}{\partial x^k \partial x^k} \tag{11}$$

since by definition $\partial f^k/\partial x^k = 0$. Equation (11) is the well-known Poisson equation. Furthermore, the two terms on the right-hand side of Equation (10) are orthogonal vectors in an appropriate inner product space, satisfying the orthogonality condition.

The projection method begins by ignoring the pressure gradient in the momentum balance equation given in Equation (9). The solution of Equation (9) without the pressure gradient will produce the intermediate velocity $\bar{v}_{\mathbf{i}}^k$. It is important to note that for the solution of the intermediate velocity, boundary conditions have to be prescribed. Often, the boundary conditions for $v_{\mathbf{i}}^{k,(m+1)}$ are used for $\bar{v}_{\mathbf{i}}^k$. Note that in order to differentiate between spatial and temporal indices, the time index is put within brackets. A detailed discussion on boundary conditions can be found in work by Brown *et al.* [21]. The computed intermediate velocity field $\bar{v}_{\mathbf{i}}^k$ does not, in general, satisfy mass conservation. However, this incorrect velocity field can be projected onto a divergence-free space after solving a pressure Poisson equation, from which the divergence-free part of the velocity field v^k can be extracted. This divergence-free velocity part will, by definition, satisfy the mass conservation equation identically. Specifically, since the pressure term in Equation (9) is troublesome, we instead use the Hodge decomposition of Equation (10) to write

$$\frac{\bar{v}^k}{\Delta t} = \frac{v^k}{\Delta t} + \frac{1}{\rho}\frac{\partial p}{\partial x^k} \tag{12}$$

for the *intermediate velocity* field $\bar{v}^k$. We know that from mass conservation an incompressible fluid is divergence-free with respect to velocity, so that upon taking the divergence of Equation (12) we have

$$\frac{1}{\Delta t}\frac{\partial \bar{v}^k}{\partial x^k} = \frac{\partial}{\partial x^k}\left(\frac{1}{\rho}\frac{\partial p}{\partial x^k}\right) \tag{13}$$

DOI: 10.1002/nme

which is called the *pressure Poisson equation*. The pressure Poisson equation is subjected to Neumann boundary conditions that can be obtained by using the divergence theorem on Equation (13) as

$$\frac{\bar{v}_{\mathbf{i}}^{l} n_{\mathbf{i}}^{l}}{\Delta t} = \frac{n_{\mathbf{i}}^{l}}{\rho_{\mathbf{i}}} \frac{\partial p_{\mathbf{i}}}{\partial x_{\mathbf{i}}^{l}} \tag{14}$$

where the $n_{\mathbf{i}}^{l}$ are the components of the unit normal vector. Once Equation (13) is used to solve for the pressure field, we can use Equation (12) to determine the correct, incompressible velocity field v^{k}.

A main advantage of using incompressible SPH is the elimination of the speed of sound parameter in the time-step conditions. Much larger time steps can be used in this approach, at the computational expense of having to solve the pressure Poisson equation at each time step. The time-step requirement for incompressible SPH is

$$\Delta t \leqslant 0.25 \frac{h_{\mathbf{ij},\min}}{\max \| v_{\mathbf{i}}^{k} \|} \tag{15}$$

where we define

$$h_{\mathbf{ij}} \equiv \frac{h_{\mathbf{i}} + h_{\mathbf{j}}}{2}$$

and $h_{\mathbf{ij},\min}$ is the minimum smoothing length for all $\mathbf{i}-\mathbf{j}$ particle pairs. It has been reported [18] that incompressible SPH is also more accurate and efficient than the weakly compressible approach for flows involving moderate Reynolds numbers. Note that from this point on, we use the acronym SPH with the understanding that a projective approach is used to determine the unknown pressure field of the system.

A first-order Euler time step is used in the simulations for the sake of simplifying the discussion. Yet, it is possible to implement higher-order accurate explicit time integration, such as a leap-frog predictor–corrector scheme, or implicit time integration. The prediction–correction algorithm for the solution of incompressible SPH equations is performed following the projection method originally suggested by Chorin [22, 23]. The general form of the algorithm for incompressible SPH can be found in [18, 19].

2.3. SPH boundary treatment

In most engineering problems, the domain of interest is, in general, bounded. The bounding domain (physical boundary) might be the surfaces of rigid bodies enclosing fluid or enclosed by fluid, fully or partially. The boundary can be stationary or in motion. Recall that the SPH formulations introduced in the preceding sections are valid for all interior particles. However, for particles close to the domain boundary, the kernel will be truncated by the boundary, so that at the boundary, $W(r_{\mathbf{ij}}, h) \neq 0$. Therefore, the application of boundary conditions is problematic in the SPH technique, since SPH approximation no longer produces the $\mathcal{O}(h^2)$ accuracy. Consequently, the proper and correct boundary treatments have been an ongoing concern for an accurate and successful implementation of the SPH approach [24, 25] as well as other meshless methods [26, 27] in the solution of engineering problems with bounded domains. Improper boundary treatment has two important consequences. The first one originates from the penetration of fluid particles into

DOI: 10.1002/nme

boundary walls, which then leave the bounded domain. The second consequence is that kernel truncation at the boundary will produce errors in the solution. In this direction, over the last decade, several different approaches have been suggested to improve boundary treatment. In the following, we will briefly introduce some of these approaches, and comment on their shortcomings. As such, the treatment offered in this presentation will be an improvement to the current status of boundary treatment.

To prevent the penetration of fluid particles across the computational domain, *specular reflections* of fluid particles with the boundary walls were suggested, such that the parallel component of the momentum of the particle is conserved while the normal component is reversed. This approach is perfectly effective and easy to implement for a linear boundary, and sufficient to make sure that particles do not penetrate through the boundary and remain within the flow domain. Nevertheless, for complex geometries with curved or inclined surfaces, or a combination, a complex collision algorithm is required. Another approach is to have a 'bounce-back' procedure in which both components of the velocity are reversed on collision of the particle with a wall. Despite the fact that these approaches prevent boundary penetration, it does not eliminate the error due to the truncation of the kernel by a boundary. As well, these approaches introduce significant distortion in the flow in the vicinity of solid boundaries. Another option is to 'reintroduce' the particles back into the body of the other liquid particles with velocity components taken from a Maxwellian distribution. Alternatively, several boundary repulsive force approaches have been regularly implemented in the SPH literature for the particles trying to penetrate the boundary.

Monaghan suggested placement of boundary particles right on the solid boundary that exert an increasing repulsive force on the particles adjacent to the boundary (particles with kernel truncation). The repulsive force prevents these particles from penetrating through the boundary. Given the similarity between molecular dynamics and SPH, Monaghan suggested a Lennard–Jones potential (LJP)-type force as a repulsive force in the form

$$F_{\mathbf{ij}}^k = \left[\left(\frac{r_0}{r_{\mathbf{ij}}}\right)^{n_1} - \left(\frac{r_0}{r_{\mathbf{ij}}}\right)^{n_2}\right] \frac{K r_{\mathbf{ij}}^k}{r_{\mathbf{ij}}^2} \quad \text{if } \frac{r_{\mathbf{ij}}}{r_0} \leqslant 1 \tag{16}$$

where $F_{\mathbf{ij}}^k$ is the repulsive force per unit mass on fluid particle $\mathbf{i}$ due to the boundary particle $\mathbf{j}$, $n_1 = 12$, and $n_2 = 4$ are constants, r_0 is the cutoff distance, and K is a problem-dependent parameter that is in general chosen as the largest particle velocity in the system squared. An important point to note regarding the aforementioned repulsive force field is that the large variation in the force on a fluid particle moving parallel to the boundary causes a large disturbance to the flow near a boundary. In the simulation of benchmark problems, namely flow in a lid-driven cavity and over a cylindrical obstacle, we observed that utilizing improper LJP-type repulsive forces introduced instability in the flow domain near the boundaries, and in extreme cases corrupted the simulation by blowing up particle positions.

An alternative form of the repulsive boundary force that reduces the disturbance on the flow was also suggested by Monaghan [28] in the form originated from the gradient of a cubic spline kernel

$$F_{\mathbf{ij}}^k = B(r_n, r_t) n_{\mathbf{j}}^k \tag{17}$$

where $F_{\mathbf{ij}}^k$ is the repulsive force per unit mass on a fluid particle $\mathbf{i}$ due to the boundary particle $\mathbf{j}$, and $n_{\mathbf{j}}^k$ is the local unit normal to a boundary particle. Here, $B(r_n, r_t)$ is a function chosen so that its value rapidly increases as r_n goes to zero, meaning that the fluid particle is attempting to penetrate the

DOI: 10.1002/nme

boundary. The quantities r_n and r_t are distances from a fluid particle to a boundary particle in the boundary normal and tangential directions, respectively. Monaghan suggested the function $B(r_n, r_t) = \Gamma(r_n)\chi(r_t)$, where $\Gamma(r_n)$ is defined to imitate the behavior of the gradient of a cubic spline. The gradient of the cubic spline has a maximum at $s \equiv r_{\mathbf{ij}}/h_{\mathbf{ij}} = \frac{2}{3}$ and goes to zero when the distance between two particles becomes zero, $s = 0$.

The boundary force formulation is corrected by replacing the value of the gradient with its maximum for $0 < s < 2/3$ and leads to the following form of the boundary force function:

$$\Gamma(r_n) = \beta \begin{cases} 2/3 & \text{if } 0 \leqslant s < 2/3 \\ 2s - 3s^2/2 & \text{if } 2/3 \leqslant s < 1 \\ (2-s)^2/2 & \text{if } 1 \leqslant s < 2 \\ 0 & \text{if } s \geqslant 2 \end{cases} \tag{18}$$

where $\beta = 0.02c^2/r_n$ is a proportionality constant, and c is the particle speed of sound. Note that for the function $\Gamma(r_n)$ we have that $s = r_n/h_{\mathbf{ij}}$. The other boundary force function $\chi(r_t)$ is defined as

$$\chi(r_t) = \begin{cases} 1 - r_t/\Delta p & \text{if } 0 \leqslant r_t < \Delta p \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where Δp is the spacing between two consecutive boundary particles. The function $\chi(r_t)$ guarantees that a fluid particle moving parallel to the boundary will be subjected to the repulsive force only from two boundary particles between which its position lies. We implemented this boundary treatment on our benchmark problems and observed that it offers slightly better result than LJP, and yet still causes significant disturbance on the flow structure.

Another approach for the boundary treatment in the literature is that proposed by Morris *et al.* [15]. In their approach, they create ghost particles outside of the model domain during their grid or particle generations. Then, for each fluid particle having a kernel truncation by a solid boundary, the unit normal vector to the boundary is calculated in order to determine a tangent line to the boundary. In so doing, they are able to produce field values to these ghost (mirrored) particles through extrapolating the fields of the fluid particle of interest to all ghost particles by using the ratio of shortest distances from fluid particle to the tangent line (d_f) and from tangent line to ghost particles (d_g). For example, the velocity $\vec{\mathbf{v}}_\mathrm{g}$ of a ghost particle is calculated using

$$\vec{\mathbf{v}}_\mathrm{g} = -\frac{d_\mathrm{g}}{d_\mathrm{f}}\vec{\mathbf{v}}_\mathrm{f}$$

where $\vec{\mathbf{v}}_\mathrm{f}$ is the velocity of the associated fluid particle. This relationship requires that the tangential velocity at the boundary is $\vec{\mathbf{v}}_\mathrm{T} = \vec{\mathbf{0}}$, consistent with the no-slip condition. In this presentation, for the sake of discussion, we refer to Morris' treatment as the *single boundary tangent technique*, which is illustrated in Figure 1.

However, as emphasized by other SPH researchers, such as Monaghan, the single boundary tangent technique has significant limitations for treating complex boundary shapes, due to the fact that there might be more than one way to construct a normal vector to the boundary, as illustrated in Figure 2(a). This limitation of the single boundary tangent approach is significant, given that in the literature SPH has been suggested to be advantageous over mesh-dependent techniques in terms of being able to treat flow domains bound by complex boundaries. Even though the current status of

DOI: 10.1002/nme

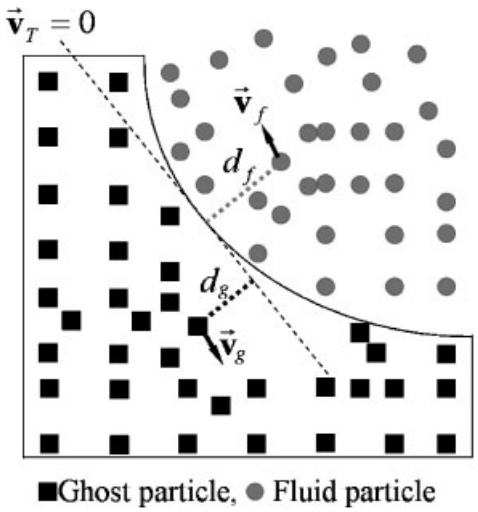

Figure 1. Single boundary tangent technique.

ghost particle implementation is limited to relatively simple geometries, we have observed that the ghost particle approach is the most stable and effective one among all the approaches elaborated on. For SPH to take its rightful place in the family of computational techniques, it is of paramount importance to have a reliable boundary treatment. Therefore, we suggest a new approach that eliminates the aforementioned limitations. We call this approach the *multiple boundary tangent method* [29], which is composed of several steps, as illustrated in Figure 2.

The various steps in the MBT technique, as depicted in Figure 2, are as follows:

1. At each time step, for all boundary particles, tangent lines are computed.
2. Given that each boundary particle has fluid particles in its influence domain as neighbors, these fluid particles are mirrored with respect to the tangent line of the corresponding boundary particle. Neighbors are computed using the standard box-sorting algorithm. Using the cell array structure (the Fortran 90 derived data type), every boundary particle is associated with its corresponding mirrored particles. Spatial coordinates and particle identification numbers of mirrored particles are stored in the cell array. To be more precise, mirrored particles are associated with the particle identification number of the fluid particle from which they are originated (referred to as the 'mother' fluid particle). For example, for a fluid particle indexed with $\mathbf{i}=21$, the ghost particle mirrored about a boundary particle tangent line (for example, boundary particle 11) will also be associated with $\mathbf{i}=21$. Note that fluid and boundary particles have numerical identifications that are permanent, whereas mirrored particles have varying (dummy) indices, throughout the simulation.
3. In the same manner, using the cell array data structure, fluid particles with boundary truncations (near boundary fluid particles) are associated with their neighbor boundary particles. For example, near boundary fluid particle 30 has three boundary particles as neighbors, as illustrated in Figure 2(c). Storing these particles allows computation of the overlapping contributions of mirrored particles from each boundary particle, as well as associating mirrored particles with near boundary fluid particles.

DOI: 10.1002/nme

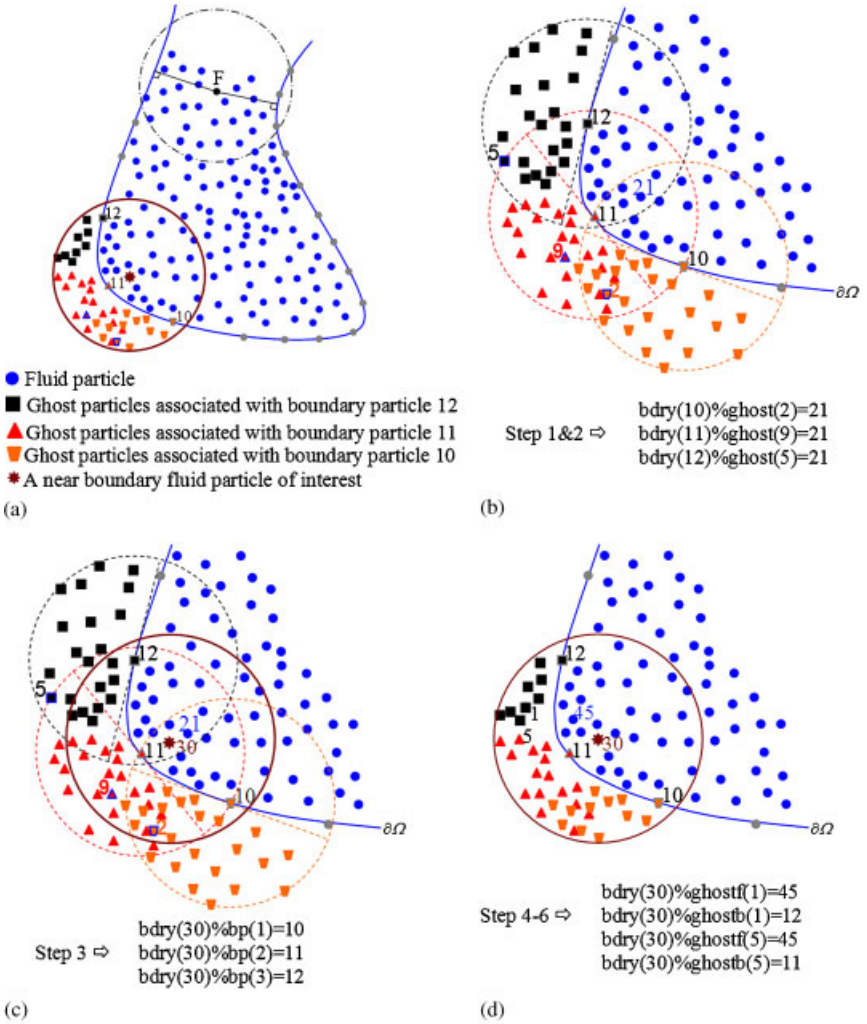

Figure 2. Multiple boundary tangent technique.

4. In a loop over all particles, if a fluid particle has a boundary particle or multiple boundary particles as neighbor(s), then the fluid particle will become a neighbor of all mirrored particles associated with the corresponding boundary particles, on the condition that
   - the mirrored particles are in the influence domain of the fluid particle in question, and
   - for a mirrored particle, its mother particle has to be within the influence domain of the fluid particle in question. This condition is particularly important in dealing with 90° boundary corners. For example, flow over backward-facing step has fluid particles close to the corner at the vertical wall taking high-velocity contributions from particles, which are mirrored from the fluid particles nearby the corner at the horizontal wall. In such

DOI: 10.1002/nme

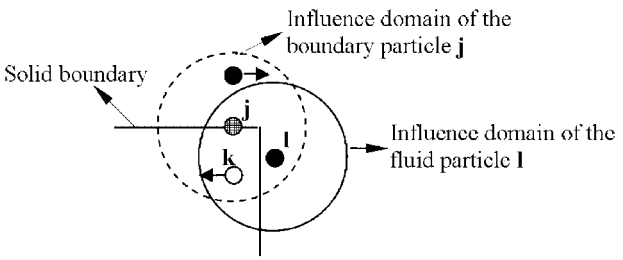

Figure 3. Corner bend problem with mirrored particle averaging.

circumstances, fluid particles will tend to artificially cling towards the vertical wall due to the high-velocity contribution of the mirrored particle here, as illustrated in Figure 3.

5. During the creation of ghost particles, there is an over-creation of ghost particles due to the fact that the influence domain of neighboring boundary particles overlaps. The overlapping contributions of mirrored particles can be eliminated by determining the number of times a given fluid particle is mirrored into the influence domain of the associated fluid particle with respect to a boundary particle's tangent line. Near boundary fluid particles hold the information of spatial coordinates and fluid particle identity numbers, boundary particle identity numbers (i.e. the particle number for a boundary particle to which mirrored particles are associated initially), and over-creation number for mirrored particles in the cell array format. For example, the ghost particles with index 1 and 5 are mirrored from the same fluid particle 45 about the tangent line of the boundary particle 11 and 12, respectively.
6. During the SPH summation over ghost particles for a fluid particle with a boundary truncation, the mass of the ghost particles are divided by the number of corresponding over-creations.

The boundary treatment technique suggested in the present article may have the following advantages over its counterparts in the following aspects:

- Since ghost particles are created for a liquid particle with a missing contribution due to the boundary truncation by using its boundary particles, the approach has the potential to treat complex geometries.
- It does not require initial particle generations outside the boundary as ghost particles.
- It allows the creation of ghost particles totally conforming the shape and the curvature of the boundary, thereby taking into account the effect of boundary curvature.

A ghost particle is given the same mass, density and transport parameters, such as viscosity, as the corresponding fluid particle. As for the field values (i.e. velocities) of a ghost particle, they are obtained depending on the type of boundary condition implemented. For instance, for no-slip boundary conditions, the following relation is implemented:

$$\vec{\mathbf{v}}_{\mathrm{g}} = \vec{\mathbf{v}}_{\mathrm{b}} + \frac{d_{\mathrm{bf}}}{d_{\mathrm{bg}}} \left(\vec{\mathbf{v}}_{\mathrm{b}} - \vec{\mathbf{v}}_{\mathrm{f}} \right) \tag{20}$$

where $\vec{\mathbf{v}}_{\mathrm{g}}$, $\vec{\mathbf{v}}_{\mathrm{b}}$, and $\vec{\mathbf{v}}_{\mathrm{f}}$ are the velocities of the ghost, boundary, and fluid particles, respectively, and d_{bf} and d_{bg} are the distances from the boundary to the fluid and ghost particles, respectively. In this work, the ratio of these two distances for each fluid particle is unity because fluid particles are mirrored with respect to a boundary tangent computed for each particle. As for the implementation

DOI: 10.1002/nme

of a zero-gradient at the boundary, a ghost particle is given the same field values as the corresponding fluid particle. If the boundaries are stationary walls, the ghost particles will have the velocity $\vec{\mathbf{v}}_g = -\vec{\mathbf{v}}_f$ for no-slip, and $\vec{\mathbf{v}}_g = \vec{\mathbf{v}}_f$ for the zero-gradient boundary conditions. The ghost particles are included in the SPH summations for all terms in the solutions of intermediate velocity, pressure, and the projection step.

3. TWO-DIMENSIONAL LID-DRIVEN CAVITY FLOW

The first benchmark problem solved using the MBT SPH approach is a two-dimensional simulation of a lid-driven cavity problem. The no-slip boundary conditions are employed within the cavity for bottom ($y=0$), left ($x=0$), and right ($x=L$) walls. The top of the cavity ($y=H$) has the boundary condition $v_x = v_0 = 10^{-3}\,\mathrm{m/s}$ and $v_y = 0$. The initial condition for all interior fluid particles is taken as $v_x = v_y = 0$. The governing flow equations are the balance of mass and linear momentum as given by Equations (9). The parameters $H = L = 0.1\,\mathrm{m}$, $\rho = 1000\,\mathrm{kg/m^3}$, and $\mu = 10^{-3}\,\mathrm{kg/m\,s}$ were selected for this simulation, producing a Reynolds number $Re = 100$. Figure 4 illustrates

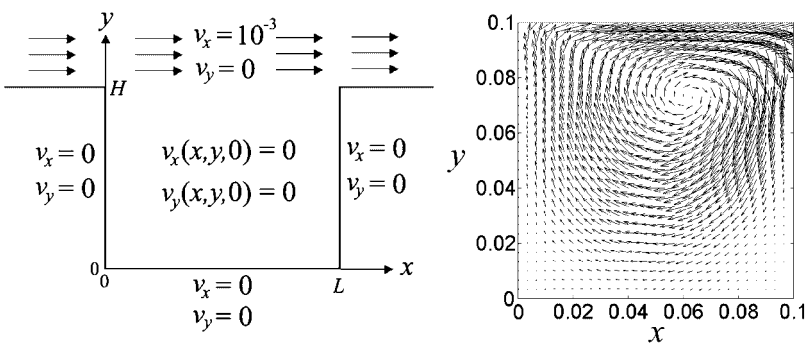

Figure 4. Two-dimensional lid-driven cavity flow domain and velocity vector plot for $Re = 100$.

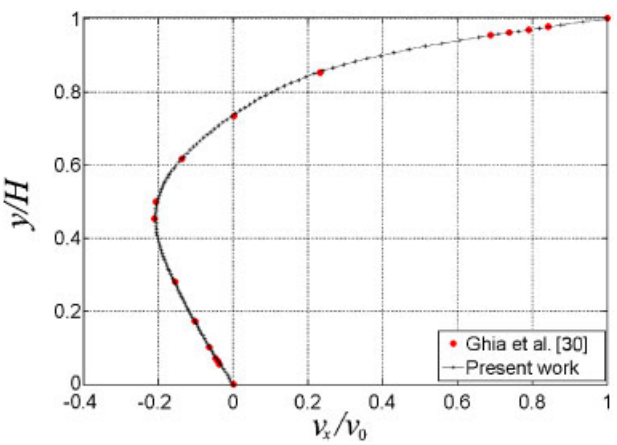

Figure 5. y/H versus v_x/v_0 at $x = L/2$ for $Re = 100$.

DOI: 10.1002/nme

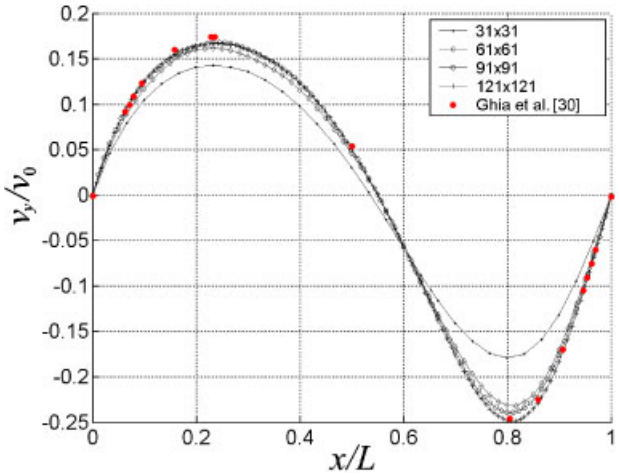

Figure 6. v_y/v_0 versus x/L at $y=H/2$ for $Re=100$.

the problem domain. The computational domain consisted of an array of 121×121 particles, and produced a velocity vector plot (which is plotted using a reduced 31×31 array, for clarity) in Figure 4.

For the lid-driven cavity benchmark problem, the numerical results have been reported by Ghia *et al*. [30] and are referenced for comparison. The normalized horizontal velocity component is plotted below at the vertical domain centerline $x=L/2$ in Figure 5.

The normalized vertical velocity component is plotted at the horizontal domain centerline $y=H/2$ in Figure 6, which also demonstrates the heuristic convergence of the solution as the number of SPH particles is increased. Both SPH results agree very well with the numerical findings of Ghia *et al*.

4. FLOW OVER A CYLINDRICAL OBSTACLE

The flow over a cylindrical obstacle was studied by Morris *et al*. as weakly compressible flow [15] and by Takeda *et al*. as compressible flow [17]. The study by Morris focused on Reynolds number (calculated based on cylinder radius) of 0.03 and 1. The reported results from Takeda *et al*. are for Reynolds numbers between 6 and 55. In this work, two-dimensional flow over a cylindrical obstacle is solved using projective (incompressible) SPH on a rectangular domain with the length of $L=0.9\,\mathrm{m}$, a height of $H=0.6\,\mathrm{m}$, and a cylinder diameter $D=0.04\,\mathrm{m}$. The center of the cylinder is located at Cartesian coordinates $(L/3, H/2)$. The simulation parameters are taken as $\rho=1000\,\mathrm{kg/m^3}$, $\mu=10^{-3}\,\mathrm{kg/m\ s}$, and $F_{Bx}=3\times10^{-6}\,\mathrm{N/kg}$ where F_{Bx} is a body force per unit mass used to model the hydrostatic part of the pressure gradient. Initially, a 200×100 array (in x- and y-directions, respectively) of particles is created in the rectangular domain. The mass of each particle is constant and found through dividing the total mass by the number of particles according to $m_{\mathbf{j}}=\rho HL/\mathbf{N}$ where H and L are the height and length defining the SPH domain area. Employing the geometrical relation for the circle (obstacle), the particles falling into the obstacles are determined and then removed from the initial list (array) of particles. Boundary particles are created on the obstacle with the particle spacing equal to the smallest particle spacing

DOI: 10.1002/nme

among the initial particles. Since particles are removed from inside the obstacle, and boundary particles have been created on the obstacle, the remaining total mass has to be redistributed on the current particles. The particles on the boundaries (on the obstacle, upper and lower boundaries) are labelled so that their positions do not evolve throughout the simulation. The total number of all particles (including boundary and fluid particles) then becomes $\mathbf{N}=19997$. The particles start moving from rest with zero initial velocities, and the adaptive time stepping that satisfies the CFL condition is implemented.

The periodic boundary condition is applied for inlet and outlet particles in the direction of the flow. The assumption behind the periodic boundary condition for the velocity is that the velocity components repeat themselves in space. Particles crossing the outflow boundary are reinserted into the flow domain at the inlet from the same y-coordinate positions with the same field values. However, for pressure, the outlet particles are inserted into the domain with the same pressure gradient as at the outlet. In essence, the periodic boundary approach effectively simulates infinitely long plates. The no-slip boundary condition is implemented for the cylindrical obstacle. For upper and lower walls that bound the simulation domain, the symmetry boundary condition for the velocity is applied, such that $v_y=0$ and $\partial v_x/\partial y=0$ where we use

$$\frac{\partial v_{\mathbf{i}}^x}{\partial y}=\sum_{\mathbf{j}=1}^{\mathbf{N}}\frac{m_{\mathbf{j}}}{\rho_{\mathbf{j}}}v_{\mathbf{ji}}^x\frac{r_{\mathbf{ij}}^y}{r_{\mathbf{ij}}}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial r_{\mathbf{ij}}}=0 \tag{21}$$

for the gradient of the horizontal velocity component in the y-direction.

Figure 7 illustrates the velocity vector plots near the cylinder for Reynolds numbers 10, 20, 30, 40, and 50.

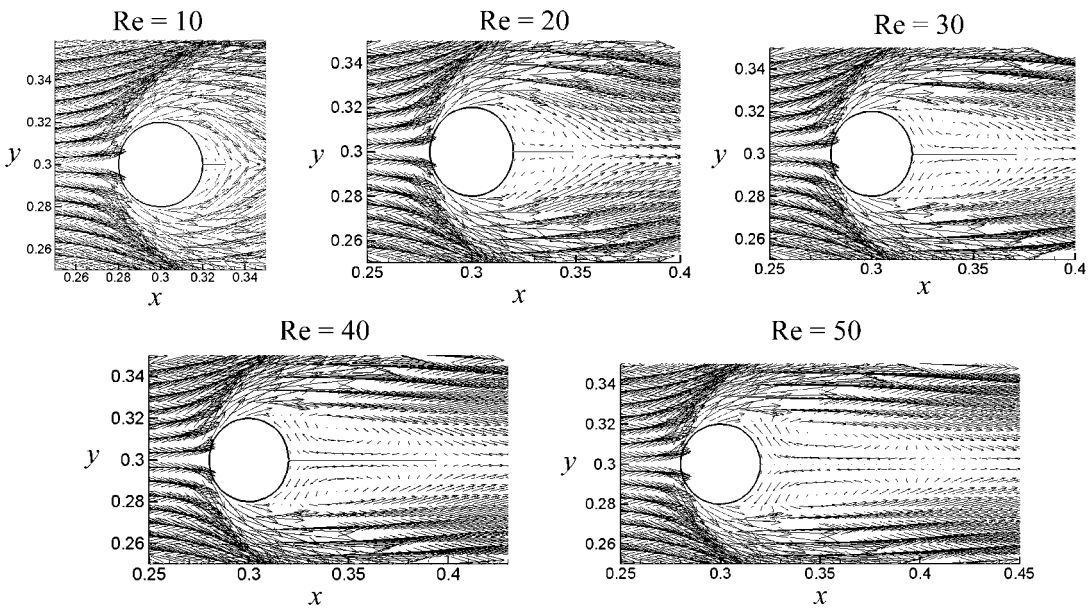

Figure 7. Wake velocity vector plot for $Re=10, 20, 30, 40$, and 50.

DOI: 10.1002/nme

In the following, for the sake of the space, we have provided the magnitude of the velocity (m/s) and the pressure (Pa) fields for $Re = 20, 30$, and 40 only. Figure 8 shows the pressure and velocity magnitude contours found using SPH with the MBT approach for $Re = 20$. The contour plots are obtained by interpolating the field quantities onto the initial mesh, followed by the triangulation of the discrete particles positions. As can be clearly seen from the figure, the contour plot lines are perfectly symmetric with respect to $y = 0$. The minimum and maximum values of pressure in the domain are -0.00147722 and 0.00156577, respectively. In Figure 8, the contour level (1) = -0.00128703, contour level (15) = 0.00137559, and contour level increment = 0.000190187 Pa. In the following figures, the pressure contour plots give an enlarged view of the domain.

Figure 9 shows the pressure and velocity magnitude contours found using SPH with the multiple boundary tangent approach for $Re = 30$. The minimum and maximum values of pressure in the domain are -0.00354169 and 0.00391918 Pa, respectively. The contour level (1) = -0.00307539, contour level (15) = 0.00345287, and contour level increment = 0.000466304409 Pa. Comparison of the pressure contour plot in Figure 9 for $Re = 30$ with the results presented by Takeda *et al*. [17]

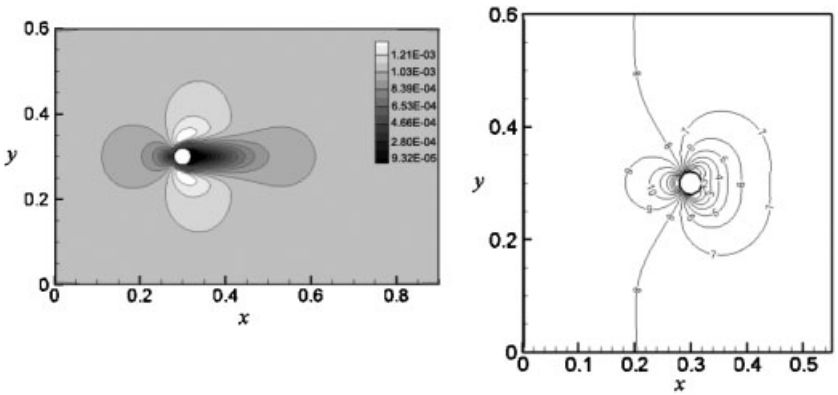

Figure 8. Velocity magnitude and pressure field contours for $Re = 20$.

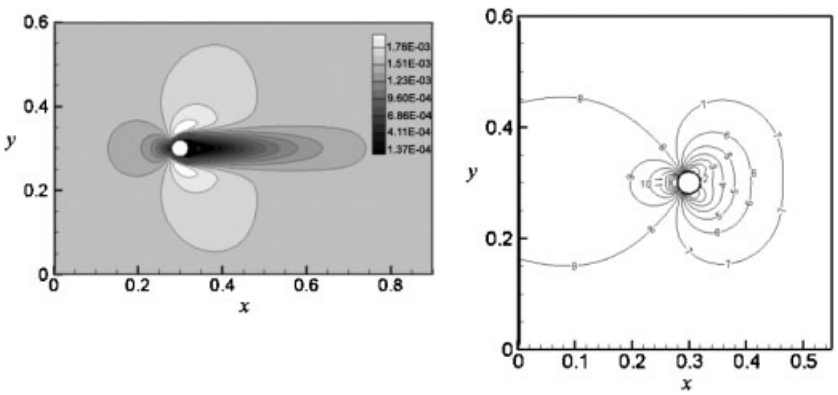

Figure 9. Velocity magnitude and pressure field contours for $Re = 30$ for incompressible SPH.

DOI: 10.1002/nme

reveals that the incompressible solver does not lead to obvious noise in the fields, as compared with a weakly compressible or compressible solver. This is because of the use of an incompressible solver, which was also pointed out by Cummins *et al*. [18]. We also observed noise in the contour field plots when the solution was obtained using the weakly compressible approach, which are given for comparison in Figure 10. Note that in Figure 10, the weakly compressible simulation used an initial array of 200×134 particles, which results in the total number of 26 909 particles including boundary and fluid particle. The pressure was computed from an artificial equation of state.

In order for future authors to compare against the incompressible SPH results provided in this article for the flow around a cylindrical obstacle, we also provide the pressure distribution curve along the path indicated with a dashed line for $Re = 20$ and 30 in Figures 11 and 12.

Figures 13 and 14 present spatial particle positions obtained by incompressible and weakly compressible SPH, respectively, in order to provide a comparison between their Lagrangian natures

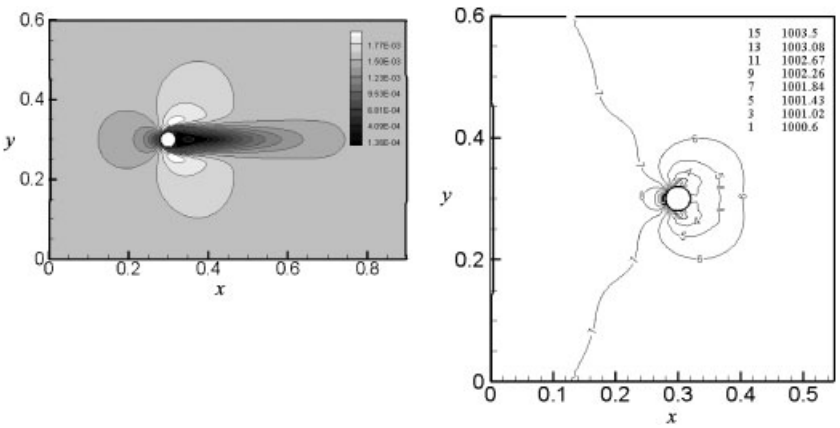

Figure 10. Velocity magnitude and density field contours for $Re = 30$ for weakly compressible SPH.

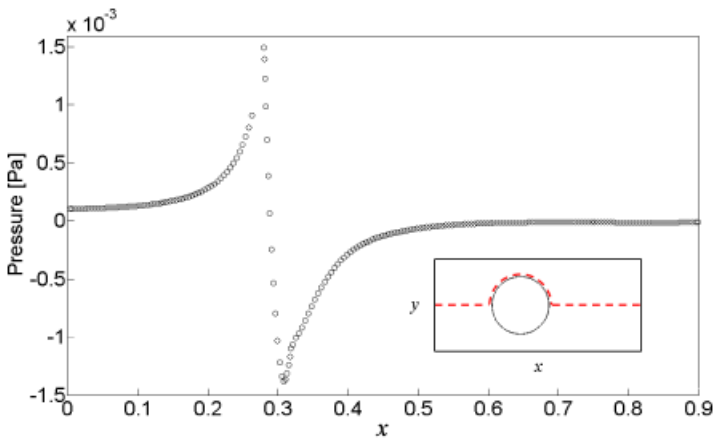

Figure 11. Pressure curves along the path indicated for $Re = 20$.

DOI: 10.1002/nme

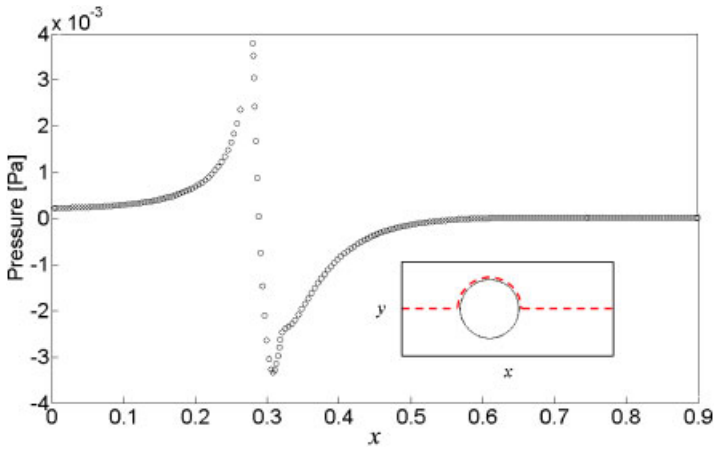

Figure 12. Pressure curves along the path indicated for $Re = 30$.

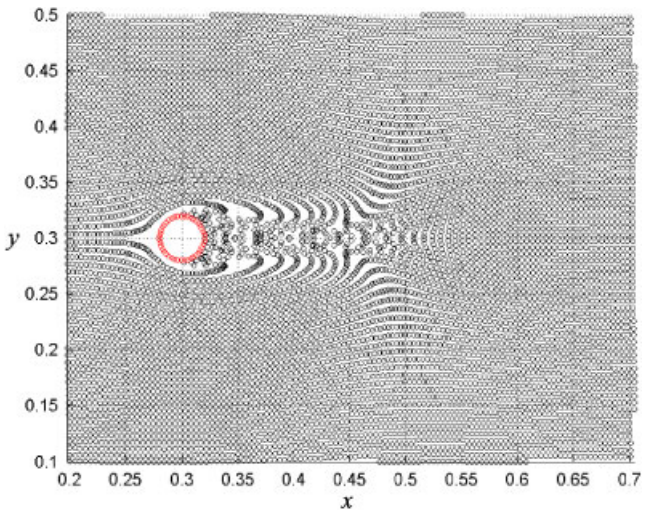

Figure 13. Incompressible SPH particle positions.

of particle tracking. Particle positions obtained by weakly compressible SPH behind the trailing edge of the cylindrical obstacle are highly anisotropic and do not follow flow stream lines closely. As the simulation advances in time, the spatial particle distribution becomes progressively more irregular and hence may lead to the formation of a non-physical cavity behind the trailing edge. Formation of the particle depletion can be attributed to the occurrence of the well-known tensile instability. A similar conclusion has been reported at SPHERIC, the second international workshop, by Colagrossi *et al*. [31] for the solution of viscous flow around a moving square. Unlike weakly compressible SPH, the particle distributions computed by incompressible SPH (refer to Figure 11) are more uniform and closely follow the stream lines. There is no void formation behind the

DOI: 10.1002/nme

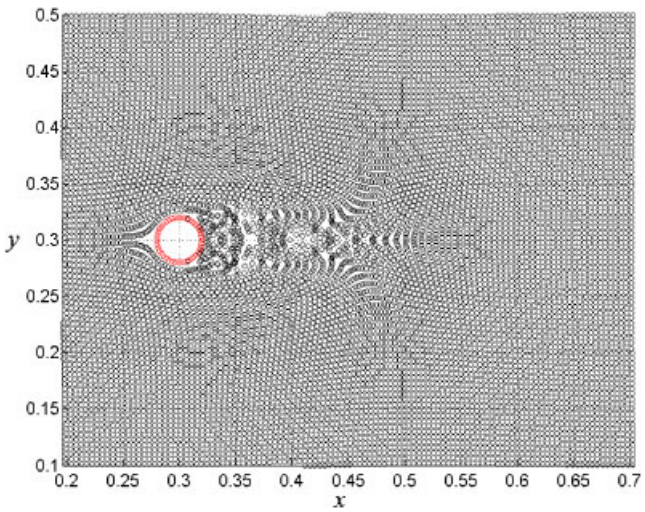

Figure 14. Weakly compressible SPH particle positions.

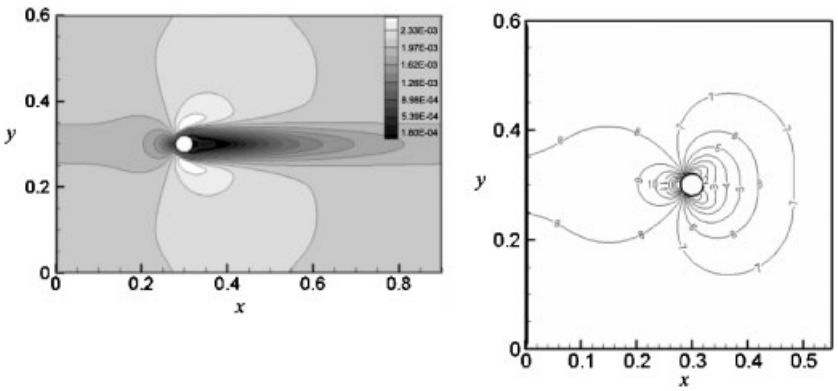

Figure 15. Velocity magnitude and pressure field contours for $Re = 40$.

cylinder that can deteriorate or corrupt the solution. The incompressible SPH solution for flow around a moving square by Lee *et al.* [32] also supports these findings.

Figure 15 shows the pressure and velocity magnitude contours found using SPH with the MBT approach for $Re = 40$. The minimum and maximum values of pressure in the domain are -0.00616261 and 0.00705214 Pa, respectively. The contour level $(1) = -0.00533669086$, contour level $(15) = 0.00622621475486$, and the contour level increment 0.000825921830 Pa.

In order to compare with the experimental results for this benchmark problem, the normalized wake size l/D versus Re is plotted in Figure 16. The length of the wakes for each of the Reynolds numbers are measured from the trailing edge of the cylindrical obstacle to the location in the wake where the x-component of the velocity vector is zero or nearly zero (see velocity vector plots

DOI: 10.1002/nme

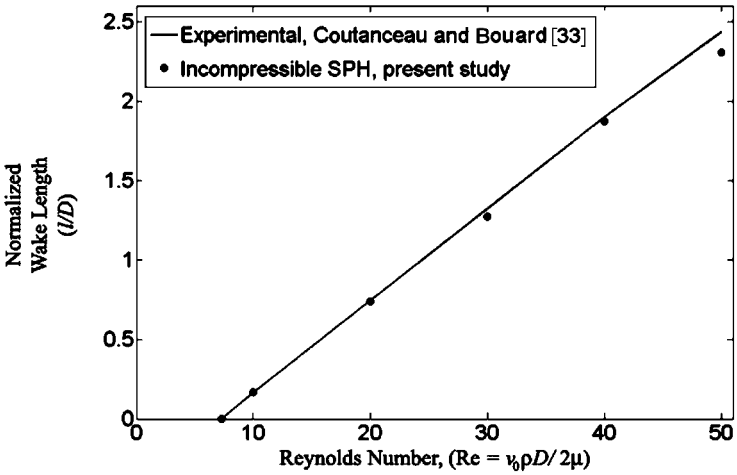

Figure 16. Normalized wake size l/D versus Re.

for the horizontal line used for this measurement). Wake sizes for the Reynolds number ranging from 10 to 50 are compared with the experimental results [33] for $\xi = 0.07$ where ξ is the ratio of spacing between upper and lower bounding walls to diameter of the cylindrical obstacle. The computational results are in very good agreement with the experimental findings, suggesting that incompressible SPH can successfully resolve linear momentum transfer. The Reynolds number is calculated as $Re = v_0 \rho D/2\mu$ where v_0 is the average value of the inlet velocity, and D is the diameter of the cylinder.

5. CONCLUDING REMARKS

In this article we have presented solutions for two important benchmark problems to validate that the proposed MBT method produces results that agree with known numerical and experimental solutions presented in literature. It was found that the SPH results and the results from the literature were in very good agreement. The current presentation is original in three aspects as follows. To our best knowledge, the benchmark problems presented here have not been solved with incompressible SPH. As well, the new MBT approach eliminates many shortcomings of current boundary treatments in the literature.

Presently, the technique proposed herein has been tested only for solid boundaries. Further testing of the MBT method on sharp changes in solid boundary geometries, as in the case of flow over a backward- and forward-facing step, has been initiated. As well, our future work includes flow simulations over more complex solid boundary geometries such as flow over an airfoil and a systematic comparison of the MBT method with other SPH boundary treatment methods. We have not tested the algorithm for treating time-evolving, free-surfaces at this point. Such testing would involve additional challenges in terms of identifying the exact position of the boundary at a given time and computing the normal and tangent lines. Another challenge for very complex free-surface simulations would be to ensure that all mirrored particles remain in the non-fluid domain.

DOI: 10.1002/nme

Finally, as a footnote we have provided a better and more readable derivation for the second-order derivative of a scalar or vector field using the concept of second- and fourth-order isotropic tensors, found in Appendix A.

APPENDIX A

The following section provides a derivation for the SPH approximation for a second-order derivative of a vector function. The derivation is carried out in Cartesian coordinates. Before deriving the SPH vector Laplacian, it is convenient to first construct a more general second-order derivative approximation. To determine this derivative, we begin by rewriting the Taylor series expansion of $f^p(\vec{\mathbf{r}}_\mathbf{j})$ so that

$$f^p(\vec{\mathbf{r}}_\mathbf{j}) = f^p(\vec{\mathbf{r}}_\mathbf{i}) + r^l_{\mathbf{ji}} \left.\frac{\partial f^p(\vec{\mathbf{r}}_\mathbf{j})}{\partial x^l_\mathbf{i}}\right|_{\vec{\mathbf{r}}_\mathbf{j}=\vec{\mathbf{r}}_\mathbf{i}} + \frac{1}{2} r^l_{\mathbf{ji}} r^k_{\mathbf{ji}} \left.\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{j})}{\partial x^l_\mathbf{i} \partial x^k_\mathbf{i}}\right|_{\vec{\mathbf{r}}_\mathbf{j}=\vec{\mathbf{r}}_\mathbf{i}} + \cdots \tag{A1}$$

using the concise $()_{\mathbf{ji}}$ difference notation. Upon multiplying Equation (A1) by the term

$$\frac{r^s_{\mathbf{ij}}}{r^2_{\mathbf{ij}}} \frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{i}}$$

and integrating over the whole space $\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}$, we obtain

$$\begin{aligned}\iiint_\Omega [f^p(\vec{\mathbf{r}}_\mathbf{j}) - f^p(\vec{\mathbf{r}}_\mathbf{i})] \frac{r^s_{\mathbf{ij}}}{r^2_{\mathbf{ij}}} \frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{i}} \mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j} &= \iiint_\Omega r^l_{\mathbf{ji}} \frac{\partial f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x^l_\mathbf{i}} \frac{r^s_{\mathbf{ij}}}{r^2_{\mathbf{ij}}} \frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{i}} \mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j} \\ &+ \iiint_\Omega \frac{1}{2} r^l_{\mathbf{ji}} r^k_{\mathbf{ji}} \frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x^l_\mathbf{i} \partial x^k_\mathbf{i}} \frac{r^s_{\mathbf{ij}}}{r^2_{\mathbf{ij}}} \frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{i}} \mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}\end{aligned} \tag{A2}$$

up to second-order accuracy in the expansion. Note that in Equation (A2) the relationship

$$\frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{j}} = -\frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{i}}$$

has been used. Now the first integral on the right hand side of Equation (A2) is a third-rank tensor that vanished due to spherical symmetry and isotropy. Therefore, Equation (A2) reduces to

$$2 \iiint_\Omega [f^p(\vec{\mathbf{r}}_\mathbf{j}) - f^p(\vec{\mathbf{r}}_\mathbf{i})] \frac{r^s_{\mathbf{ij}}}{r^2_{\mathbf{ij}}} \frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{i}} \mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j} = \frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x^l_\mathbf{i} \partial x^k_\mathbf{i}} \iiint_\Omega r^l_{\mathbf{ji}} r^k_{\mathbf{ji}} \frac{r^s_{\mathbf{ij}}}{r^2_{\mathbf{ij}}} \frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{i}} \mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j} \tag{A3}$$

since the second-order derivative on the right hand side can be pulled outside of the integral, as this derivative has been evaluated at position $\vec{\mathbf{r}}_\mathbf{i}$.

The integral on the right hand side of Equation (A3) can be integrated by parts, which, upon using the Green-Gauss theorem produces the result

$$\iiint_\Omega r^l_{\mathbf{ji}} r^k_{\mathbf{ji}} \frac{r^s_{\mathbf{ij}}}{r^2_{\mathbf{ij}}} \frac{\partial W(r_{\mathbf{ij}}, h)}{\partial x^m_\mathbf{i}} \mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j} = -\iiint_\Omega W(r_{\mathbf{ij}}, h) \frac{\partial}{\partial x^m_\mathbf{j}} \left(\frac{r^l_{\mathbf{ji}} r^k_{\mathbf{ji}} r^s_{\mathbf{ij}}}{r^2_{\mathbf{ij}}} \right) \mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j} \tag{A4}$$

DOI: 10.1002/nme

Using the product rule of calculus, we can obtain the result

$$\frac{\partial}{\partial x_{\mathbf{j}}^m}\left(\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k r_{\mathbf{ij}}^s}{r_{\mathbf{ij}}^2}\right)=\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^s}{r_{\mathbf{ij}}^2}\delta^{km}+\frac{r_{\mathbf{ji}}^k r_{\mathbf{ji}}^s}{r_{\mathbf{ij}}^2}\delta^{lm}+\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k}{r_{\mathbf{ij}}^2}\delta^{sm}-2\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k r_{\mathbf{ji}}^s r_{\mathbf{ji}}^m}{r_{\mathbf{ij}}^4}$$

which, when used in Equation (A4) and then in Equation (A3) yields

$$\begin{aligned}&2\iiint_\Omega [f^p(\vec{\mathbf{r}}_\mathbf{i})-f^p(\vec{\mathbf{r}}_\mathbf{j})]\frac{r_{\mathbf{ij}}^s}{r_{\mathbf{ij}}^2}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_{\mathbf{i}}^m}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}\\&=\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_{\mathbf{i}}^l\partial x_{\mathbf{i}}^k}\iiint_\Omega W(r_{\mathbf{ij}},h)\left[\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^s}{r_{\mathbf{ij}}^2}\delta^{km}+\frac{r_{\mathbf{ji}}^k r_{\mathbf{ji}}^s}{r_{\mathbf{ij}}^2}\delta^{lm}+\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k}{r_{\mathbf{ij}}^2}\delta^{sm}-2\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k r_{\mathbf{ji}}^s r_{\mathbf{ji}}^m}{r_{\mathbf{ij}}^4}\right]\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}\end{aligned}\quad\text{(A5)}$$

To obtain the Laplacian of a vector function, we can contract on indices m and s in Equation (A5) which produces

$$2\iiint_\Omega [f^p(\vec{\mathbf{r}}_\mathbf{i})-f^p(\vec{\mathbf{r}}_\mathbf{j})]\frac{r_{\mathbf{ij}}^m}{r_{\mathbf{ij}}^2}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_{\mathbf{i}}^m}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}=3\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_{\mathbf{i}}^l\partial x_{\mathbf{i}}^k}\iiint_\Omega W(r_{\mathbf{ij}},h)\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k}{r_{\mathbf{ij}}^2}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}\quad\text{(A6)}$$

If we now observe that the integral on the right hand side of Equation (A6) is a second-rank isotropic tensor (since the radially symmetric kernel is multiplied by an even function) we can write for some constant c that

$$\iiint_\Omega W(r_{\mathbf{ij}},h)\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k}{r_{\mathbf{ij}}^2}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}=c\delta^{kl}$$

which, upon contracting on indices k and l gives $c=1/3$ since the kernel is normalized. These results, when used in Equation (A6) produce the SPH Laplacian approximation

$$\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_{\mathbf{i}}^k\partial x_{\mathbf{i}}^k}=2\iiint_\Omega [f^p(\vec{\mathbf{r}}_\mathbf{i})-f^p(\vec{\mathbf{r}}_\mathbf{j})]\frac{r_{\mathbf{ij}}^m}{r_{\mathbf{ij}}^2}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_{\mathbf{i}}^m}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}\quad\text{(A7)}$$

Returning now to Equation (A5) we note that again the fourth-order tensor in the integrand on the right hand side must be an isotropic tensor due to radial symmetry of the kernel. The most general fourth-order isotropic tensor is written as

$$a^{smkl}=c_1\delta^{sm}\delta^{kl}+c_2\delta^{sk}\delta^{ml}+c_3\delta^{sl}\delta^{mk}$$

in terms of constants c_1, c_2, and c_3. As well, the fourth-order tensor in the integrand of Equation (A5) is symmetric in all indices, it can be shown that $c_1=c_2=c_3$ so that

$$\begin{aligned}&\iiint_\Omega W(r_{\mathbf{ij}},h)\left[\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^s}{r_{\mathbf{ij}}^2}\delta^{km}+\frac{r_{\mathbf{ji}}^k r_{\mathbf{ji}}^s}{r_{\mathbf{ij}}^2}\delta^{lm}+\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k}{r_{\mathbf{ij}}^2}\delta^{sm}-2\frac{r_{\mathbf{ji}}^l r_{\mathbf{ji}}^k r_{\mathbf{ji}}^s r_{\mathbf{ji}}^m}{r_{\mathbf{ij}}^4}\right]\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}\\&=c_1(\delta^{sm}\delta^{kl}+\delta^{sk}\delta^{ml}+\delta^{sl}\delta^{mk})\end{aligned}\quad\text{(A8)}$$

DOI: 10.1002/nme

Contracting on all indices produces the result, in three dimensions, $c_1 = 1/5$ so that Equation (A5) simplifies to

$$2\int\int\int_{\Omega}[f^p(\vec{\mathbf{r}}_\mathbf{i})-f^p(\vec{\mathbf{r}}_\mathbf{j})]\frac{r_{\mathbf{ij}}^s}{r_{\mathbf{ij}}^2}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_\mathbf{i}^m}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}=\frac{1}{5}\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_\mathbf{i}^l\partial x_\mathbf{i}^k}(\delta^{sm}\delta^{kl}+\delta^{sk}\delta^{ml}+\delta^{sl}\delta^{mk})$$

or

$$2\int\int\int_{\Omega}[f^p(\vec{\mathbf{r}}_\mathbf{i})-f^p(\vec{\mathbf{r}}_\mathbf{j})]\frac{r_{\mathbf{ij}}^s}{r_{\mathbf{ij}}^2}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_\mathbf{i}^m}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}=\frac{1}{5}\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_\mathbf{i}^l\partial x_\mathbf{i}^k}(\delta^{sm}\delta^{kl}+2\delta^{sk}\delta^{ml}) \tag{A9}$$

Rearranging Equation (A9) yields

$$2\int\int\int_{\Omega}[f^p(\vec{\mathbf{r}}_\mathbf{i})-f^p(\vec{\mathbf{r}}_\mathbf{j})]\frac{r_{\mathbf{ij}}^s}{r_{\mathbf{ij}}^2}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_\mathbf{i}^m}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}-\frac{1}{5}\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_\mathbf{i}^l\partial x_\mathbf{i}^k}\delta^{sm}\delta^{kl}=\frac{2}{5}\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_\mathbf{i}^l\partial x_\mathbf{i}^k}\delta^{sk}\delta^{ml}$$

which simplifies to

$$2\int\int\int_{\Omega}[f^p(\vec{\mathbf{r}}_\mathbf{i})-f^p(\vec{\mathbf{r}}_\mathbf{j})]\frac{r_{\mathbf{ij}}^s}{r_{\mathbf{ij}}^2}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_\mathbf{i}^m}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j}-\frac{1}{5}\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_\mathbf{i}^k\partial x_\mathbf{i}^k}\delta^{sm}=\frac{2}{5}\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_\mathbf{i}^m\partial x_\mathbf{i}^s} \tag{A10}$$

We immediately recognize the second term on the left hand side as the Laplacian, which has already been determined in Equation (A7). Therefore, using the result of Equation (A7) and noting that

$$\frac{\partial W(r_{\mathbf{ij}},h)}{\partial x_\mathbf{i}^m}=\frac{r_{\mathbf{ij}}^m}{r_{\mathbf{ij}}}\frac{\partial W(r_{\mathbf{ij}},h)}{\partial r_{\mathbf{ij}}}$$

produces the desired relation for the SPH second derivative approximation

$$\left\langle\frac{\partial^2 f^p(\vec{\mathbf{r}}_\mathbf{i})}{\partial x_\mathbf{i}^m\partial x_\mathbf{i}^s}\right\rangle=\int\int\int_{\Omega}[f^p(\vec{\mathbf{r}}_\mathbf{i})-f^p(\vec{\mathbf{r}}_\mathbf{j})]\frac{1}{r_{\mathbf{ij}}}\left(\frac{5r_{\mathbf{ij}}^m r_{\mathbf{ij}}^s}{r_{\mathbf{ij}}^2}-\delta^{sm}\right)\frac{\partial W(r_{\mathbf{ij}},h)}{\partial r_{\mathbf{ij}}}\,\mathrm{d}^3\vec{\mathbf{r}}_\mathbf{j} \tag{A11}$$

ACKNOWLEDGEMENTS

Funding provided by the Natural Sciences and Engineering Research Council (NSERC) CRD project no. CRDPJ 261287-02 is gratefully acknowledged.

REFERENCES

1. Liu GR, Liu MB. *Smoothed Particle Hydrodynamics*: *A Meshfree Particle Method*. World Scientific: Singapore, 2003.
2. Lucy LB. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal* 1977; **82**:1013.
3. Gingold RA, Monaghan JJ. Smooth particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**:375.
4. Sigalotti LDG, Klapp J, Sira E, Meleán Y, Hasmy A. SPH simulations of time-dependent Poiseuille flow at low Reynolds numbers. *Journal of Computational Physics* 2003; **191**:622–638.
5. Zhu Y, Fox PJ. Smoothed particle hydrodynamics model for diffusion through porous media. *Transport in Porous Media* 2001; **43**:441–471.

DOI: 10.1002/nme

6. Liu MB, Liu GR, Lam KY. A one-dimensional meshfree particle formulation for simulating shock waves. *Shock Waves* 2003; **13**:201–211.
7. Colagrossi A, Landrini M. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics* 2003; **191**:448–475.
8. Cleary PW, Ha J, Aluine V, Nguyen T. Flow modelling in casting processes. *Applied Mathematical Modelling* 2002; **26**:171–190.
9. Rook RA, Yildiz M, Dost S. Modelling 2D transient heat transfer using SPH and implicit time integration. *Journal of Numerical Heat Transfer B* 2007; **51**:1–23.
10. López H, Sigalotti L, Di G. Oscillation of viscous drops with smoothed particle hydrodynamics. *Physical Review E* 2006; **73**:051201.
11. Meleán Y, Sigalotti L, Di G, Hasmy A. On the SPH tensile instability in forming viscous liquid drops. *Computer Physics Communications* 2004; **157**:191–200.
12. Hu XY, Adams NA. A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics* 2006; **213**:844–861.
13. Ellero M, Kröger M, Hess S. Viscoelastic flows studied by smoothed particle dynamics. *Journal of Non-Newtonian Fluid Mechanics* 2002; **105**:35–51.
14. Ellero M, Tanner RI. SPH simulations of transient viscoelastic flows at low Reynolds number. *Journal of Non-Newtonian Fluid Mechanics* 2005; **132**:61–72.
15. Morris JP, Fox PJ, Zhu Y. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics* 1997; **136**:214–226.
16. Chaniotis AK, Poulikakos D, Koumoutsakos P. Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows. *Journal of Computational Physics* 2002; **182**:67–90.
17. Takeda H, Miyama SM, Sekiya M. Numerical simulation of viscous flow by smoothed particle hydrodynamics. *Progress of Theoretical Physics* 1994; **92**:939–960.
18. Cummins SJ, Rudman M. An SPH projection method. *Journal of Computational Physics* 1999; **152**:584–607.
19. Shao S, Lo EMY. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources* 2003; **26**:787–800.
20. Schoenberg IJ. Contributions to the problem of approximation of equidistant data by analytic functions. *Quarterly of Applied Mathematics* 1946; **4**:45–99.
21. Brown DL, Cortez R, Minion ML. Accurate projection methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2001; **168**:464–499.
22. Chorin AJ. Numerical solutions of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
23. Chorin AJ. On the convergence of discrete approximations to the Navier–Stokes equations. *Mathematics of Computation* 1969; **23**:341–353.
24. Kulasegaram S, Bonet J, Lewis RW, Profit M. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. *Computational Mechanics* 2004; **33**:316–325.
25. Feldman J, Bonet J. Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems. *International Journal for Numerical Methods in Engineering* 2007; **72**:295–324.
26. Krongauz Y, Belytschko T. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering* 1996; **131**:133–145.
27. Alfaro I, Yvonnet F, Chinesta R, Cueto E. A study on the performance of natural neighbour-based Galerkin methods. *International Journal for Numerical Methods in Engineering* 2007; **71**:1436–1465.
28. Monaghan JJ. Smoothed particle hydrodynamics. *Reports on Progress in Physics* 2005; **68**:1703–1759.
29. Yildiz M, Suleman A. SPH with improved ghost particle boundary treatment. *SPHERIC*, *The Second International Workshop*, Spain, 2007; 71–74.
30. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387.
31. Colagrossi A, Colicchio G, Le Touzé D. Enforcing boundary conditions in SPH applications involving bodies with right angles. *SPHERIC*, *The Second International Workshop*, Spain, 2007; 59–62.
32. Lee E-S, Violeau D, Laurence D, Stansby P, Moulinec C. SPHERIC test case 6: 2-D incompressible flow around a moving square inside a rectangular box. *SPHERIC*, *The Second International Workshop*, Spain, 2007; 37–41.
33. Coutanceau M, Bouard R. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. *Journal of Fluid Mechanics* 1977; **79**:231.

DOI: 10.1002/nme