Journal of Computational Physics **182,** 67–90 (2002)
doi:10.1006/jcph.2002.7152

Remeshed Smoothed Particle Hydrodynamics for the Simulation of Viscous and Heat Conducting Flows

A. K. Chaniotis,* D. Poulikakos,* and P. Koumoutsakos†

*Institute of Energy Technology, Laboratory of Thermodynamics in Emerging Technologies, ETH, Zürich, Switzerland; †Institute of Computational Sciences, ETH, Zürich, Switzerland; and Center for Turbulence Research, NASA Ames, Moffett Field, California 94035
E-mail: poulikakos@ltnt.iet.mavt.ethz.ch

Received July 24, 2001; revised May 9, 2002

We present an extension of the classical scheme of smoothed particle hydrodynamics (SPH) for the accurate handling of diffusion terms in the momentum and energy equation of viscous and heat conducting flows. A key aspect of the present SPH approach is the periodic reinitialization (remeshing) of the particle locations, which are being distorted by the flow map. High-order moment conserving kernels are being implemented for this remeshing procedure leading to accurate simulations. The accuracy of the proposed SPH methodology is tested for a number of benchmark problems involving flow and energy transport. The results demonstrate that the proposed SPH methodology is capable of DNS quality simulations while maintaining its robustness and adaptivity.

1. INTRODUCTION

The smooth particle hydrodynamics (SPH) method is a Lagrangian numerical method introduced by Gingold and Monaghan [1], in order to model problems in continuum physics while circumventing some of the limitations of grid-based methods. SPH is a robust numerical technique that has been applied to a wide range of problems, ranging from compressible fluid mechanics to astrophysics simulations and flow structure interactions [1–6]. Although the method enjoys the properties of Lagrangian schemes, such as automatic adaptivity and numerical stability, the extension of the method in order to handle diffusion-type effects has been limited. One of the key difficulties is the handling of diffusion type operators on the Lagrangian mesh, which is usually distorted by the flow map. A methodology to overcome these difficulties is presented in this paper.

The development of the classical SPH methodology for compressible flow fields is described in detail by Monaghan [7]. The computational elements are particles whose location

0021-9991/02 $35.00

is following in a Lagrangian fashion the flow map. The initial flow field quantities are interpolated on the particle locations, and all the flow quantities can be reconstructed by a linear superposition of the flow quantities carried by the particles as weighed by a smooth interpolation kernel [8, 9]. The discrete equations are obtained from continuum equations by expressing the flow quantities as a linear superposition of the physical quantities that are being carried by the particles.

SPH belongs to a class of Lagrangian methods called particle methods. The key advantage of all particle methods is to avoid the explicit discretization of the nonlinear convection term while maintaining an automatic adaptivity for the computational elements. However, particle methods are faced with difficulties when dealing with the approximation of viscous effects. The approximation of diffusion operators in the context of particle methods is a subject that has been extensively addressed in the context of vortex methods in the last decade [10]. Several options have been identified such as the derivation of conservative schemes based on the approximation of the diffusion operator by an integral operator [11, 12], the differentiation of the smoothing kernel [13], and the approximation of the diffusion operator on the distorted Lagrangian grid using some averaging procedures [14].

In this work, we present a numerical scheme to account for diffusion effects in the context of smooth particle hydrodynamics by incorporating a remeshing strategy for the particle locations along with an efficient calculation of the diffusion operators in the distorted particle locations.

The fact that viscosity plays an important role in many physical phenomena of engineering interest underlines the need to improve the modeling of viscous forces while maintaining the adaptivity and robustness of SPH. A commonly employed methodology to account for diffusion effects in SPH is to introduce an artificial viscosity so that conservation of momentum is ensured. However, this scheme usually yields inaccurate results, because the separation of shear and bulk viscosity is not allowed [7, 15].

An alternative approach to remedy this situation is based on a Taylor series expansion of the field quantities in the neighborhood of each particle and combines the standard first-order SPH derivatives with the finite differences method [16]. The method of Brookshaw [16] is computationally efficient, since only the first derivative of the kernel is required and conserves the linear momentum, while the angular momentum is approximately conserved. This type of approximation for the diffusion term has been implemented successfully to simulate heat conduction problems [6, 16] and incompressible viscous flows [5] with solid boundaries but it may lead to inaccurate results when the velocity or the density field is noisy [15].

Another approximation of viscous effects in SPH involves the nested application of the difference approximation thus calculating second-order derivatives from first derivatives [15, 17]. This method can calculate any second derivative in such a way that the formulas are symmetric and they conserve the linear and angular momentum. However, this method is not computationally efficient as it would imply repeated calculations involving all particles and might yield inaccurate results when the density is noisy [15].

Finally, an alternative approximation of the diffusion terms is based on the direct computation of the second derivative of the kernel [13, 18]. However, this method is considered to be very sensitive to disordered particle locations, in particular for low-order kernels [16]. Moreover, it strongly depends on the number of neighbors around each particle. In order to achieve accurate results, a minimum number of neighboring particles is necessary. At the same time as the computational cost is proportional to the number of neighbor particles, it is desirable to keep this number near a lower limit. In order to circumvent this difficulty,

most of the SPH implementations are using variable smoothing length (h), which allows for an efficient balancing the number of neighbor particles near the lower limit.

In the work presented in this paper, the direct differentiation of the kernel function is used to compute second-order derivatives. In order to remedy the drawbacks of the method associated with the particle disorder, the particle locations are periodically reinitialized (*remeshed*) onto a uniform grid. This remeshing process constitutes the key novel feature of the computational scheme presented herein. With this feature, it is possible to construct a computational algorithm based on the direct differentiation of the kern (first and second derivatives) to solve a system of differential equations describing the flow and heat transport in a Lagrangian frame, while accurately accounting for viscous and heat diffusion effects.

The process of remeshing has been introduced in incompressible flow simulations using vortex methods [19] in order to eliminate the creation of spurious vortical structures, resulting from nonoverlapping smooth vortex particles. Earlier relevant studies have addressed in detail issues such as efficient rezoning of particles and particle number control in the particle in cell (PIC) method [20], as well as convergence problems of vortex particle methods using random rezoning [21].

In the present SPH implementation, the remeshing process maintains the particle resolution while ensuring that the computation of the viscous terms involves a constant number of neighboring particles.

The capabilities of the proposed remeshed SPH (rSPH) methodology is tested against finite difference calculations for a number of benchmark problems demonstrating the capabilities of the present SPH methodology as a tool for accurate and robust numerical simulations.

The paper is organized as follows. In Section 2, we outline the governing compressible Navier–Stokes equations for viscous and heat conducting flows. In Section 3, we present the discretization of the governing equations, and in Section 4 we present the validation of the proposed methodology.

2. GOVERNING EQUATIONS

The fundamental system of differential equations governing the motion of a two-dimensional viscous, heat conducting, compressible medium describe the conservation of mass, momentum and energy. The conservation equations for a calorically perfect gas without an energy source are

$$\frac{D\rho}{Dt} = -\rho \frac{\partial u_i}{\partial x_i} \tag{1}$$

$$\rho \frac{Du_i}{Dt} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i \tag{2}$$

with

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right)$$

$$\rho c_v \frac{DT}{Dt} = \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) - p\frac{\partial u_i}{\partial x_i} + \tau_{ij}\frac{\partial u_i}{\partial x_j}, \tag{3}$$

where in order to simplify the equations of motion, index notation is used for vectors and tensors in Cartesian coordinates ($i, j, k = 1, 2$), where Einstein's summation convection

must be taken into account, and x_i are the components of the position vector, u_i the velocity vector components, ρ is the density, p the pressure, g_i the gravitational acceleration component, T the temperature, μ the viscosity, k the thermal conductivity and c_v the specific heat at constant volume.

We employ Sutherland's law to determine the dependence of the viscosity μ on the temperature as $\mu = T^{\frac{2}{3}} \frac{1+S_1}{T+S_1}$. The system of the differential equations (1), (2), and (3) is closed with the equation of state for an ideal gas

$$p = \rho R T, \tag{4}$$

where R is the gas constant.

The nondimensional variables are obtained from the physical variables as

$$\begin{aligned} x_i^* &= \frac{x_i}{L_0}, \quad \rho^* = \frac{\rho}{\rho_0}, \quad t^* = \frac{t U_0}{L_0}, \quad u_i^* = \frac{u_i}{U_0}, \quad T^* = \frac{T}{T_0}, \\ p^* &= \frac{p}{\rho_0 R T_0}, \quad \mu^* = \frac{\mu}{\mu_0}, \quad g_i^* = \frac{g_i}{g_0}, \quad k^* = \frac{k}{k_0} \quad c_v^* = \frac{c_v}{c_{v0}}, \end{aligned} \tag{5}$$

where the superscript $(^*)$ and the subscript $(_0)$ indicate the nondimensional and the reference quantities. The quantities L_0, ρ_0, U_0, T_0, μ_0, $g_0\, k_0$, and c_{v0} denote the characteristic length, density, velocity, temperature, dynamic viscosity, gravity, thermal conductivity, and specific heat at constant volume, respectively. The dimensionless numbers that appear in the equations are

$$\mathrm{Re} = \frac{\rho_0 U_0 L_0}{\mu_0}, \quad M^2 = \frac{U_0^2}{\gamma R T_0}, \quad \mathrm{Pr} = \frac{\mu_0 c_p}{k_0}, \quad Fr = \frac{U_0}{\sqrt{g_0 L_0}}, \tag{6}$$

where γ is the ratio of the specific heat capacities. The symbol Re denotes the Reynolds number, M the Mach number, Pr the Prandtl number, and Fr the Froude number. Other dimensionless numbers can be defined as a function of the reference quantities and these dimensionless numbers

$$Ra = \frac{g_0 L_0^3 \rho_0^2 c_p \Delta T}{\mu_0 k_0 T_0} = \frac{\mathrm{Re}^2 \mathrm{Pr} \Delta T}{Fr^2 T_0}, \quad Gr = \frac{Ra}{\mathrm{Pr}} = \frac{\mathrm{Re}^2 \Delta T}{Fr^2 T_0}, \tag{7}$$

where ΔT is a characteristic variation of temperature of the flow and the symbols Ra denotes the Rayleigh number and Gr the Grashof number.

Finally the nondimensional system of governing equations can be written as

$$\frac{D\rho^*}{Dt^*} = -\rho^* \frac{\partial u_i^*}{\partial x_i^*} \tag{8}$$

$$\rho^* \frac{Du_i^*}{Dt^*} = -\frac{1}{M^2 \gamma} \frac{\partial p^*}{\partial x_i^*} + \frac{1}{\mathrm{Re}} \frac{\partial \tau_{ij}^*}{\partial x_j^*} + \frac{1}{Fr^2} \rho^* g_i^*, \tag{9}$$

with

$$\tau_{ij}^* = \mu^* \left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} - \frac{2}{3} \delta_{ij} \frac{\partial u_k^*}{\partial x_k^*} \right)$$

$$\rho^* c_v^* \frac{DT^*}{Dt^*} = \frac{\gamma}{\mathrm{Re}\,\mathrm{Pr}} \frac{\partial}{\partial x_i^*} \left(k^* \frac{\partial T^*}{\partial x_i^*} \right) - (\gamma - 1) p^* \frac{\partial u_i^*}{\partial x_i^*} + \frac{M^2 \gamma (\gamma - 1)}{\mathrm{Re}} \tau_{ij}^* \frac{\partial u_i^*}{\partial x_j^*} \tag{10}$$

$$p^* = \rho^* T^*. \tag{11}$$

In principle, a particular flow problem may be solved by integrating the mass, momentum, and energy equations, which are described above, and additionally the equation of state. The information, which is necessary for the initial and boundary conditions, is imposed with the system of differential equations in order to have a well-posed problem. The initial conditions are usually prescribed functions that describe the velocity field and the two of the three scalar intensive properties (density, temperature, and pressure) of the flow. For compressible flow, the following boundary conditions can be applied, depending on the particular application:

- Inflow boundaries: Prescribed velocity, temperature, and pressure,
- Solid surface: No-slip conditions for the velocity and prescribed temperature or heat flux,
- Outflow surface: Prescribed pressure and prescribed gradient of velocity and temperature,
- Symmetry plane: zero gradient normal to the boundary for all scalar quantities and the velocity component parallel to the surface and zero velocity for the component that is normal to the symmetry plane,
- Periodic plane: the computational domain is replicated throughout space.

3. THE SPH METHOD

The method is based on the Lagrangian formulation of the governing equations. The flow quantities are discretized into particles. Each particle α is associated with a mass m_a, density ρ_a, velocity u_a, viscosity μ_a, pressure p_a, and position r_a. The initial flow field quantities are interpolated on the particle locations and all the flow quantities can be reconstructed by a linear superposition of the flow quantities carried by the particles as weighed by a smooth interpolation kernel. This interpolation is based on the theory of integral interpolants [8, 9] so that the interpolated value of any function A at position r is expressed as

$$A(r) = \int A(r') W(r - r', h)\, dr', \tag{12}$$

where the integration is over the computational domain, $W$is an interpolation function, and h is a characteristic distance between the particles which is closely related to the domain of influence of the kernel (smoothing length). The choice of the interpolation kernel is the core of the method. Most SPH simulations use splines kernel (cubic, quartic, and quintic) [5, 6, 15]. In this implementation, the quartic spline is used [8, 9]. The quartic spline is constructed from three B-splines requiring the kernel and its first, second, and third derivative be continuous. The kernel is defined as

$$W(r,h) = M_5(r,h) = n_d \begin{cases} \frac{s^4}{4} - \frac{5s^2}{8} + \frac{115}{192} & 0 \le s < \frac{1}{2}, \quad s = \frac{|r|}{h} \\ -\frac{s^4}{6} + \frac{5s^3}{6} - \frac{5s^2}{4} + \frac{5s}{24} + \frac{55}{96} & \frac{1}{2} \le s < \frac{3}{2}, \\ \frac{(2.5-s)^4}{24} & \frac{3}{2} \le s < \frac{5}{2}, \\ 0 & s \ge \frac{5}{2}, \end{cases} \tag{13}$$

where n_d is a normalization constant that depends on the dimensionality of the problem and a function of the smoothing length h. The quartic spline has compact support involving 21 neighboring particles contributing in the interpolation (in 2D). The compact support ensures that the number of neighboring particles is finite, thus minimizing the computational cost. The quintic spline interpolant has the same properties with the quartic spline and additionally continuous fourth derivative but causes an increase in the computational cost [involves 29 neighboring particles (in 2D) contributing in the interpolation]. Both these kernels are second-order accurate.

Numerically, the integral is approximated by a quadrature at the N particle locations

$$A(r) = \sum_b A_b V_b W(r - r_b, h), \tag{14}$$

where the summation index b denotes a particle label and V_b is the volume of the particle $b(V_b = \frac{m_b}{\rho_b})$.

The particles are moved in a Lagrangian fashion using the following formula:

$$\frac{d\vec{x}}{dt} = \vec{u}. \tag{15}$$

In the context of the scheme proposed by Fishelov [13], if the kernel is differentiable, it is possible to calculate the derivative of any function A at position r as

$$\frac{\partial A(r)}{\partial x^i} = \sum_b A_b V_b \frac{\partial W(r - r_b, h)}{\partial x^i}. \tag{16}$$

Equation (16) is not in symmetric form but it is possible to rewrite it in symmetric form, when the differentiation is centered on a particle location $(\partial A(r_a)/\partial x^i = (\nabla A)_a)$, by using the operator

$$\nabla A = \frac{\nabla(FA) - A\nabla F}{F}, \tag{17}$$

where F is unity. When we combine Eqs. (16) and (17), the gradient operator at the location of the α particle reads

$$(\nabla A)_a = \sum_b V_b(A_b - A_a)\nabla W(r_a - r_b, h). \tag{18}$$

An approximation of the second-order derivatives, which describe the diffusion terms, is based on the direct computation of the second derivative of the kernel [13, 18]

$$\frac{\partial^2 A(r)}{\partial x^i x^j} = \sum_b A_b \frac{m_b}{\rho_b} \frac{\partial^2 W(r - r_b, h)}{\partial x^i \partial x^j}. \tag{19}$$

It is possible to rewrite the second derivative (Eq. (19)) in symmetric form by using the operator

$$\frac{\partial^2 A}{\partial x^i x^j} = \frac{\partial^2 A}{\partial x^i x^j} - A\frac{\partial^2 1}{\partial x^i x^j}, \tag{20}$$

and the second-order derivative for the location of the α particle can be written as

$$\left\langle \frac{\partial^2 A}{\partial x^i x^j} \right\rangle_a = \sum_b A_b V_b \frac{\partial^2}{\partial x_i x_j} W(r_a - r_b, h) - A_a \sum_b V_b \frac{\partial^2}{\partial x_i x_j} W(r_a - r_b, h)$$
$$= \sum_b V_b (A_b - A_a) \cdot \frac{\partial^2}{\partial x_i x_j} W(r_a - r_b, h). \tag{21}$$

The use of symmetric differentiation formulas is important as it leads to simulations where the total momentum is conserved exactly [22].

3.1. SPH Formulation of the Governing Equations

3.1.1. *The Continuity Equation*

The continuity equation can be solved by defining the fluid density via an interpolation of the individual masses carried by the particles [23]

$$\rho_a(r) = \sum_b m_b W(r_a - r_b, h). \tag{22}$$

Alternatively, the continuity equation (1) may be expressed as

$$\frac{D\rho_a}{Dt} = -\rho_a \sum_b V_b \vec{u}_b \cdot \nabla_a W(r_a - r_b, h), \tag{23}$$

where V_b is the volume of the particle $b (V_b = \frac{m_b}{\rho_b})$. Equation (23) is not in symmetric form but it is possible to rewrite it in symmetric form by using the concept of Eq. (18). Equations (23) yield

$$\frac{D\rho_a}{Dt} = -\rho_a \sum_b V_b (\vec{u}_b - \vec{u}_a) \cdot \nabla_a W(r_a - r_b, h). \tag{24}$$

Equation (24) has now a symmetric form resulting in more accurate simulations than when Eq. (23) is implemented. This equation has the form of the continuity equation proposed by Monaghan [7]

$$\frac{D\rho_a}{Dt} = -\sum_b m_b (\vec{u}_b - \vec{u}_a) \cdot \nabla_a W(r_a - r_b, h). \tag{25}$$

An important difference is that in Eq. (24), when remeshing at each time step, the volume of the particles remains practically constant. Hence, in our case the weight is the volume of the particles and not the mass. Equation (24) is used for the simulations as it has a computational advantage compared to Eq. (22) since the rate of change for all the physical variables can be computed simultaneously. This implies that all the equations can be computed simultaneously for all the particles. On the other hand, the drawback of this formulation is that the mass conservation is not algebraically guaranteed.

3.1.2. *The Momentum Equation*

The momentum equation (2) for a two-dimensional flow without the gravitational force can be written for each particle a as

$$\left\langle \rho \frac{Du}{Dt} \right\rangle_a = -\left\langle \frac{\partial p}{\partial x} \right\rangle_a + \frac{4}{3}\left\langle \frac{\partial}{\partial x}\mu\frac{\partial u}{\partial x} \right\rangle_a - \frac{2}{3}\left\langle \frac{\partial}{\partial x}\mu\frac{\partial v}{\partial y} \right\rangle_a + \left\langle \frac{\partial}{\partial y}\mu\frac{\partial u}{\partial y} \right\rangle_a + \left\langle \frac{\partial}{\partial y}\mu\frac{\partial v}{\partial x} \right\rangle_a \tag{26}$$

$$\left\langle \rho \frac{Dv}{Dt} \right\rangle_a = -\left\langle \frac{\partial p}{\partial y} \right\rangle_a + \frac{4}{3}\left\langle \frac{\partial}{\partial y}\mu\frac{\partial v}{\partial y} \right\rangle_a - \frac{2}{3}\left\langle \frac{\partial}{\partial y}\mu\frac{\partial u}{\partial x} \right\rangle_a + \left\langle \frac{\partial}{\partial x}\mu\frac{\partial u}{\partial y} \right\rangle_a + \left\langle \frac{\partial}{\partial x}\mu\frac{\partial v}{\partial x} \right\rangle_a, \tag{27}$$

where $\vec{u} = (u, v)$.

Using the standard SPH formulation [7] and the ideas outlined in previous section, the pressure gradient of Eqs. (26) and (27) for particle α can be written as

$$\left\langle \frac{\partial p}{\partial x} \right\rangle_a = \sum_b V_b(p_b - p_a) \cdot \frac{\partial}{\partial x} W(r_a - r_b, h) \tag{28}$$

$$\left\langle \frac{\partial p}{\partial y} \right\rangle_a = \sum_b V_b(p_b - p_a) \cdot \frac{\partial}{\partial y} W(r_a - r_b, h). \tag{29}$$

A general formulation is adopted for the derivation of the viscous terms of the Eqs. (26) and (27), which accounts for variable viscosity μ (the viscosity is a function of temperature). The viscous terms are written out with the help of the chain rule of differentiation:

$$\begin{aligned} \left\langle \frac{\partial}{\partial x_i}\mu\frac{\partial u^k}{\partial x_j} \right\rangle_a &= \left\langle \frac{\partial \mu}{\partial x_i} \right\rangle_a \left\langle \frac{\partial u^k}{\partial x_j} \right\rangle_a + \left\langle \mu \frac{\partial^2 u^k}{\partial x_i x_j} \right\rangle_a \\ &= \left(\sum_b V_b(\mu_b - \mu_a) \cdot \frac{\partial}{\partial x_i} W(r_a - r_b, h) \right) \\ &\quad \left(\sum_b V_b\left(u_b^k - u_a^k\right) \cdot \frac{\partial}{\partial x_j} W(r_a - r_b, h) \right) \\ &\quad + \mu_a \sum_b V_b\left(u_b^k - u_a^k\right) \cdot \frac{\partial^2}{\partial x_i x_j} W(r_a - r_b, h). \end{aligned} \tag{30}$$

The last term in Eq. (30) is constructed by using the concept presented in previous section [Eqs. (19)–(21)]. If the viscosity is constant, the first term in the RHS of the above equation is zero, In this case, the final formulation for the momentum equation is symmetric and it conserves the linear and angular momentum. The SPH formulation of the momentum equation terms [Eqs. (26) and (27)] is obtained in a straightforward manner by combining Eqs. (26)–(30) as needed. Note that Eq. (30) delivers all derivatives of the viscous terms (where $(i, j, k = 1, 2)$ and $(x_1 = x, x_2 = y, u^1 = u, u^2 = v)$).

3.1.3. *The Thermal Energy Equation*

The two-dimensional version of the energy equation (3) reads

$$\left\langle \rho c_v \frac{DT}{Dt} \right\rangle_a = \left\langle \frac{\partial}{\partial x} k \frac{\partial T}{\partial x} \right\rangle_a + \left\langle \frac{\partial}{\partial y} k \frac{\partial T}{\partial y} \right\rangle_a - \left\langle p \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right\rangle_a + \left\langle \frac{2}{3} \mu \left[\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] \right\rangle_a + \left\langle \mu \left[\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)^2 \right] \right\rangle_a . \tag{31}$$

The derivation of the SPH formulation of the heat diffusion term, allowing for a temperature dependent thermal conductivity, is similar to that of the viscous terms of the momentum equation described in the previous section. The final result reads

$$\begin{aligned} \left\langle \frac{\partial}{\partial x_i} \left(k \frac{\partial T}{\partial x_i} \right) \right\rangle_a &= \left\langle \frac{\partial k}{\partial x_i} \right\rangle_a \left\langle \frac{\partial T}{\partial x_i} \right\rangle_a + \left\langle k \frac{\partial^2 T}{\partial x_i^2} \right\rangle_a \\ &= \left(\sum_b V_b (k_b - k_a) \cdot \frac{\partial}{\partial x_i} W(r_a - r_b, h) \right) \\ &\quad \left(\sum_b V_b (T_b - T_a) \cdot \frac{\partial}{\partial x_i} W(r_a - r_b, h) \right) \\ &\quad + k_a \sum_b V_b (T_b - T_a) \cdot \frac{\partial^2}{\partial x_i^2} W(r_a - r_b, h). \end{aligned} \tag{32}$$

The compressibility term and the viscous dissipation term in the energy equation can be easily discretized by using the general formula of Eq. (18). No details are shown for brevity. The energy by compression can be positive or negative, depending upon whether the fluid is expanding or contracting.

3.2. Particle Remeshing

Lagrangian numerical methods enjoy the advantage of automatic adaptivity for their computational elements. However, the flow strain can cluster particles in some regions of the flow field and spread them apart in another. When this occurs, Eq. (19) is not accurately representing diffusion effects resulting in inaccurate simulations. To circumvent this problem, the position of the particles is periodically reinitialized on a uniform grid and the properties of the old particles are interpolated onto the new particle locations. This kind of interpolation has been implemented in a number of calculations involving particle methods [10, 19, 23, 24] but it has not been reported before, to the best of our knowledge, in the context of SPH, in conjunction with the approximation of the diffusion operator.

For the remeshing procedure two types of interpolation are implemented: a second-order ordinary interpolation and a third-order smoothing interpolation, which are further discussed in the following sections.

3.2.1. *Ordinary Interpolation*

The second-order ordinary interpolation formula [10, 24] is conserving the interpolated quantity (zero order moment) as well as its first (impulse) and second moment (angular impulse). This interpolation Λ_2, which is used for the vortex particle simulations [25], can be expressed in one dimension as

$$\Lambda_2(x,h) = \begin{cases} 1-s^2 & 0 \le s < \frac{1}{2}, \quad s = \frac{|x|}{h} \\ \frac{(1-s)(2-s)}{2} & 0 \le s < \frac{3}{2}, \\ 0 & s \ge \frac{3}{2}. \end{cases} \tag{33}$$

The manner in which this interpolation formula is applied, is shown as

$$\Delta\tilde{Q}_i(\tilde{x}_i) = Q_j(x_j)\Lambda_2(\tilde{x}_i - x_j, h), \tag{34}$$

which means that the jth old particle from location x_j with property Q_j contributes in the new ith particle which is in position $\tilde{x}_i$ the interpolating quantity $\Delta\tilde{Q}_i$. The interpolation in higher dimensions is obtained using tensorial products in each coordinate direction. The interpolation 3, 9, and 27 points respectively in one, two, or three dimensions. This formula for two dimensions can be written as

$$\Delta\tilde{Q}_i(\tilde{x}_i, \tilde{y}_i) = Q_j(x_j, y_j)\Lambda_2(\tilde{x}_i - x_j, h)\Lambda_2(\tilde{y}_i - y_j, h). \tag{35}$$

The interpolated quantity Q_j must be an extensive property of the particle that is conserved

$$Q_j = \begin{pmatrix} m_j \\ m_j u_j \\ m_j v_j \\ E_j \end{pmatrix}, \tag{36}$$

where m_j is the mass of the particle, $m_j u_j$ and $m_j v_j$ are the u and v momentum of the particle and E_j the total energy of the particle. Note that the interpolation function Λ is discontinuous at the interpolating node locations. This implies that the larger the fluctuations of the interpolating quantity, the larger the interpolation errors.

3.2.2. *Smoothing Interpolation*

The smoothing interpolation formulas attempt to minimize the error that the ordinary interpolation might produce, providing us with a moment conserving interpolation, which is continuous everywhere inside the interpolation stencil

$$M_4'(x,h) = \begin{cases} 1 - \frac{5s^2}{2} + \frac{3s^3}{2} & 0 \le s < 1, \quad s = \frac{|x|}{h} \\ \frac{(1-s)(2-s)^2}{2} & 1 \le s < 2, \\ 0 & s \ge 2. \end{cases} \tag{37}$$

The interpolation function and also its first and second derivative are continuous and reproduce second-order polynomials. The interpolation uses 4, 16, and 64 points in one, two, and

three dimensions, respectively. This interpolation is used in the present SPH implementation in the main computational domain, away from solid boundaries. For further details on the accuracy and the properties of the interpolating functions, the reader is referred to [10].

3.2.3. *Interpolation Near Solid Boundaries*

The remeshing procedure with ordinary or smoothing interpolating formulas cannot be used near solid boundaries. The interpolating stencil may extend to the interior of the body, which introduces spurious computational elements. Hence, in regions near boundaries we are using a biased ordinary interpolation [10], which is again second order and conserves the same quantities as the ordinary interpolation (conserves the first two moments and the quantity Q_j).

$$\Lambda(x,h) = \begin{cases} 1 - \frac{3}{2}s + \frac{1}{2}s^2 & \text{for 1st cell from the wall,} \quad s = \frac{|x|}{h} \\ s(2-s) & \text{for 2nd cell from the wall,} \\ \frac{s(s-2)}{2} & \text{for 3rd cell from the wall,} \\ 0 & \text{otherwise.} \end{cases} \tag{38}$$

3.3. Numerical Implementation

3.3.1. *Boundary Conditions*

All the boundary conditions are modeled by boundary particles, which have similar physical properties to those of the particles that represent the flow field. These boundary particles interact with the interior particles in a way such that the necessary boundary conditions are satisfied (non-slip, inflow, and outflow for the momentum equation and the temperature or heat flux for the energy equation). More detailed information for the implementation of boundary conditions can be found in [6, 18].

3.3.2. *Time Integration*

We implemented second- and third-order Runge–Kutta schemes for our time integration. Higher-order schemes (fourth-order Runge–Kutta) can be also used. However, due to their larger stability limit, they allow integration using larger time steps. However, this may be detrimental to the overall accuracy of the simulation as remeshing is performed only after each full time step. This choice may led to additional errors caused by particle disorder in the last stage of the integration in flows with a large strain rate.

3.3.3. *Remeshing*

Particles are remeshed with a frequency depending on the strain of the flow map. The number of time steps can vary from one to ten depending upon the flow field and the size of the time step. If the flow field is uniform without recirculation or stagnation regions, the remeshing can be evaluated after 10 or more iterations as particles maintain their uniform distribution. However, for a flow with recirculation (e.g., shear layer), the remeshing procedure is performed at every time step. The additional computational cost of the remeshing is less than 10% of the total computational cost even when remeshing in every time step.

The additional computation time resulting from remeshing is a small penalty to pay compared to the significant advantage of ensuring that the particles are always almost

equidistant, thus resulting in an accurate approximation of the viscous terms. This is a key aspect of the present work compared to ordinary SPH simulations where the distance between the particles can exhibit large variations.

In addition, the remeshing procedure resolves one computational problem that usually occurs in SPH simulations. The pressure force in the momentum equation is proportional to the derivative of the kernel which normally, for B-Splines, reduces to zero when the distance between two particle is small. As a consequence, in such cases the pressure force is attractive and this can introduce large errors in the simulations. The remeshing procedure does not allow the particles to get very close to each other.

On the other hand, the remeshing procedure introduces numerical errors which may be viewed as aliasing errors due to the implementation of remeshing kernels with finite support. The extra diffusion that the remeshing procedure introduces can be quite large (with remeshing at every time step). This is indeed a cause of concern. However, it should be pointed out that the added dissipation induced by remeshing is proportional to the gradients of the flow map, which are induced by the particle distortion. These gradients remain very small when remeshing is performed at each time step.

Using an antidiffusive process is feasible (e.g., remeshing using Beale's method [26]). However, there is no clear way to precisely control the error induced by remeshing since this is strongly coupled with the particular flow map under consideration. Finally, as discussed in [19, 27], the overall effect of remeshing is to act as a subgrid scale model and has been shown to have a negligible effect on the overall accuracy of the simulations [19].

4. TEST CALCULATIONS FOR ISOTHERMAL AND NONISOTHERMAL VISCOUS FLOWS

We examine the validity of our proposed methodology on a series of benchmark two-dimensional problems. These problems are as follows:

- Taylor–Green flow,
- Double shear layer,
- Lid-driven flow in a cavity,
- Natural convection in a differentially heated cavity, and
- Mixed convection in a lid-driven cavity.

4.1. Two-Dimensional Taylor–Green Flow

As a first test of the viscous SPH calculations, we perform a simulation of the Taylor–Green flow. Taylor–Green flow consists of a periodic decaying array of vortices in the $x - y$ plane specified in nondimensional form by

$$u(x, y, t) = -Ue^{bt}\cos(2\pi x)\sin(2\pi y) \tag{39}$$

$$v(x, y, t) = Ue^{bt}\sin(2\pi x)\cos(2\pi y) \tag{40}$$

$$p(x, y, t) = p_{\text{ref}} - \frac{U^2}{4}e^{bt}(\cos(4\pi x) + \cos(4\pi y)), \tag{41}$$

where $b = \frac{-8\pi^2}{\text{Re}}$ and $p_{\text{ref}} = \frac{1}{\gamma M^2}$. The reference Mach number M is set to 0.5 and the

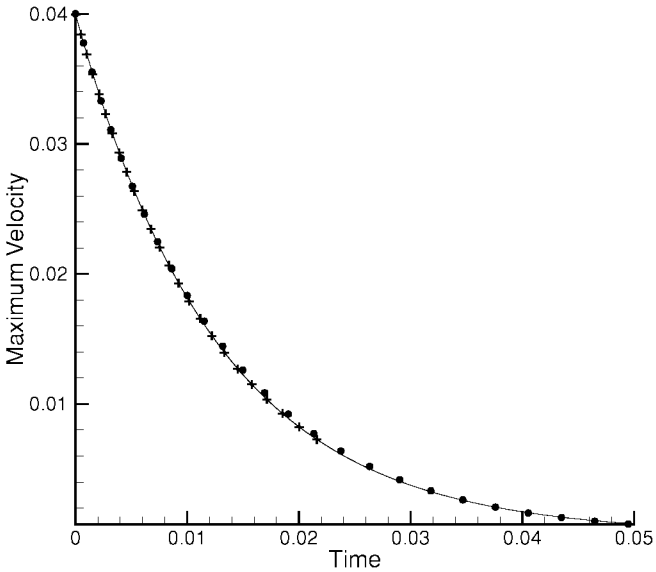


FIG. 1. Decay of total velocity for $\mathrm{Re} = 1$. Comparison of the SPH solution with 15,625 particles (−) with the exact incompressible solution (●) and with a high-order finite differences solution (+) [28].

velocity U is taken as 0.04 and the computational domain was set to $[0, 1] \times [0, 1]$. The simulations are performed for a wide range of the Reynolds number (0.1–1000) to test the accuracy of the method in cases where the viscous effects are either dominant (small Re), comparable (intermediate Re), or minimal (large Re) compared to the inertial forces. In the simulation, the boundary conditions are periodic in all directions. A third-order Runge–Kutta scheme with a constant time step is implemented in all the simulations. The smoothing interpolation formula M_4' (Eq. (37)) was used at every time step for the remeshing. The time-depended behavior of the maximum velocity of the flow for $\mathrm{Re} = 1$ calculated from SPH (with 15,625 particles) is presented in Fig. 1 and is compared to the analytical solution for an incompressible fluid and with numerical results from a compressible finite difference code. The finite difference code is a Navier–Stokes solver in conservative form that uses fifth- and sixth-order compact Pade scheme for the convective and viscous terms respectively and third-order Runge–Kutta for time integration [28]. The compressibility ($M = 0.5$) effects have negligible influence in the Taylor–Green flow so that the incompressible analytical solution can be compared to the computational results. For the error analysis of the SPH simulations, the relative error (L_∞) is used

$$L_\infty = \max_{t=0}^{T_{\max}} \left(\left| \frac{u_{ex}^t - u_{SPH}^t}{u_{ex}^t} \right| \right),$$

where u_{ex}^t denotes the maximum velocity of the exact incompressible solution at time t and u_{SPH}^t the maximum velocity of the SPH simulation at time t. $T_{\max}$ is the time where $u_{ex}^{T_{\max}} = \frac{U}{50}$. The relative error ($L_\infty$) of the SPH calculations is less than 5% for $\mathrm{Re} = 1$ and decreases as the number of particles is increased (Fig. 2). The SPH simulation with 15,625 particles remains accurate ($L_\infty < 4\%$) and grid independent until $\mathrm{Re} = 100$ but it

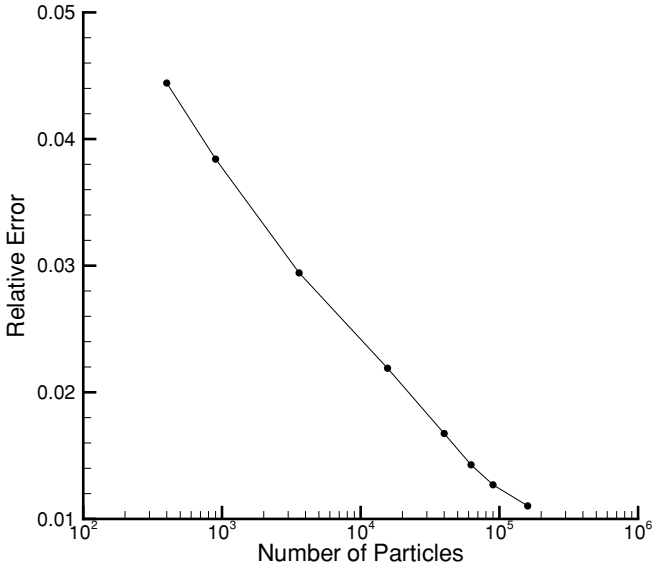


FIG. 2. L_∞ error of SPH simulations of the Taylor–Green flow for different resolutions (Re = 1).

is necessary to increase the particle resolution for larger Reynolds numbers (Re = 1000) (Fig. 3).

4.2. Two-Dimensional Double Shear Layer

The second test for the SPH calculations is the double shear layer geometry, where the initial flow field consists of a horizontal shear layer of finite thickness, perturbed by a small

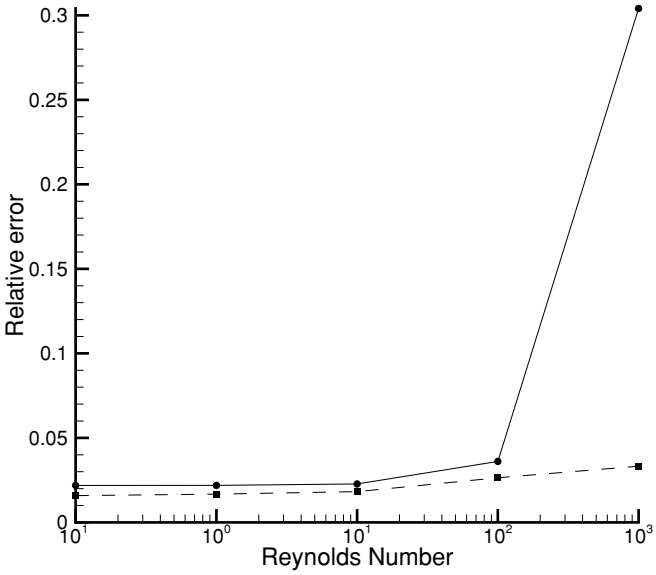


FIG. 3. L_∞ error of SPH simulations of the Taylor–Green flow for different Reynolds numbers. (−) Simulation with 15,625 particles and (- -) with 40,000 particles.

amplitude vertical velocity

$$u(x, y) = U_0 + (1 - U_0)e^{-(10y)^6} \tag{42}$$

$$v(x, y) = \delta \sin(2\pi x) \cos(2\pi y), \tag{43}$$

where $U_0 = -1$ and $\delta = 0.01$. For the initial temperature, the Crocco–Busemann relation is employed, which yields a general relation for the dependence of temperature on velocity, with the assumption of a Prandtl number of unity [29]

$$T(x, y) = M^2 \frac{\gamma - 1}{2}[u(x, y)(1 + U_0) - u(x, y)^2 - U_0] + T_0 \frac{1 - u(x, y)}{(1 - U_0)} + \frac{u(x, y) - U_0}{(1 - U_0)}, \tag{44}$$

where T_0 is set 0.5 and M is the Mach number. For the simulation, the example was considered

$$M = 0.5 \quad \mathrm{Re} = 1000 \quad \mathrm{Pr} = 0.72,$$

and the computational domain was set to $[-0.5, 0.5] \times [-0.5, 0.5]$. For the comparison of the SPH calculations, the same high-order compressible finite difference code [28] as in Section 4.1 above was employed. Both SPH and finite difference simulations must be comparable from a common initial stage that satisfies the Navier–Stokes and the energy equations. The SPH simulation is initialized from the finite difference solution at time $t = 0.3$ by using a seventh-order polynomial interpolation. From the comparison of density snapshots (Fig. 4), one may observe that the two methods are capable of producing

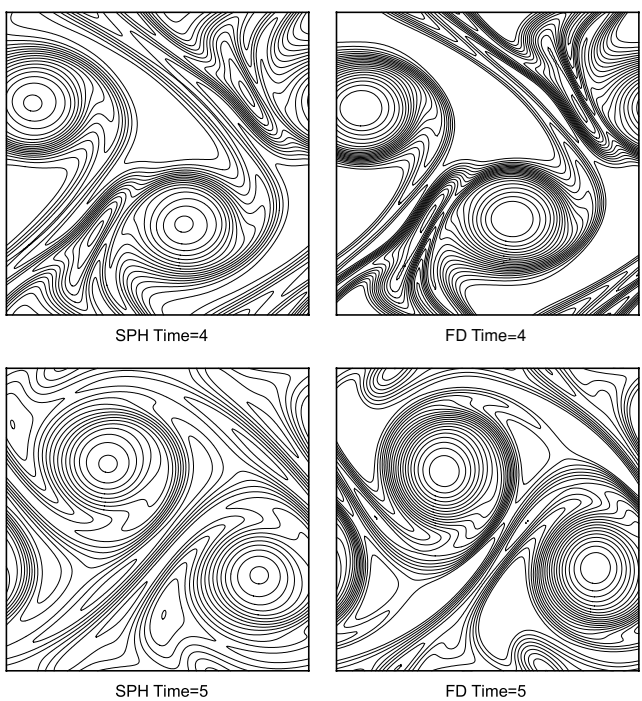
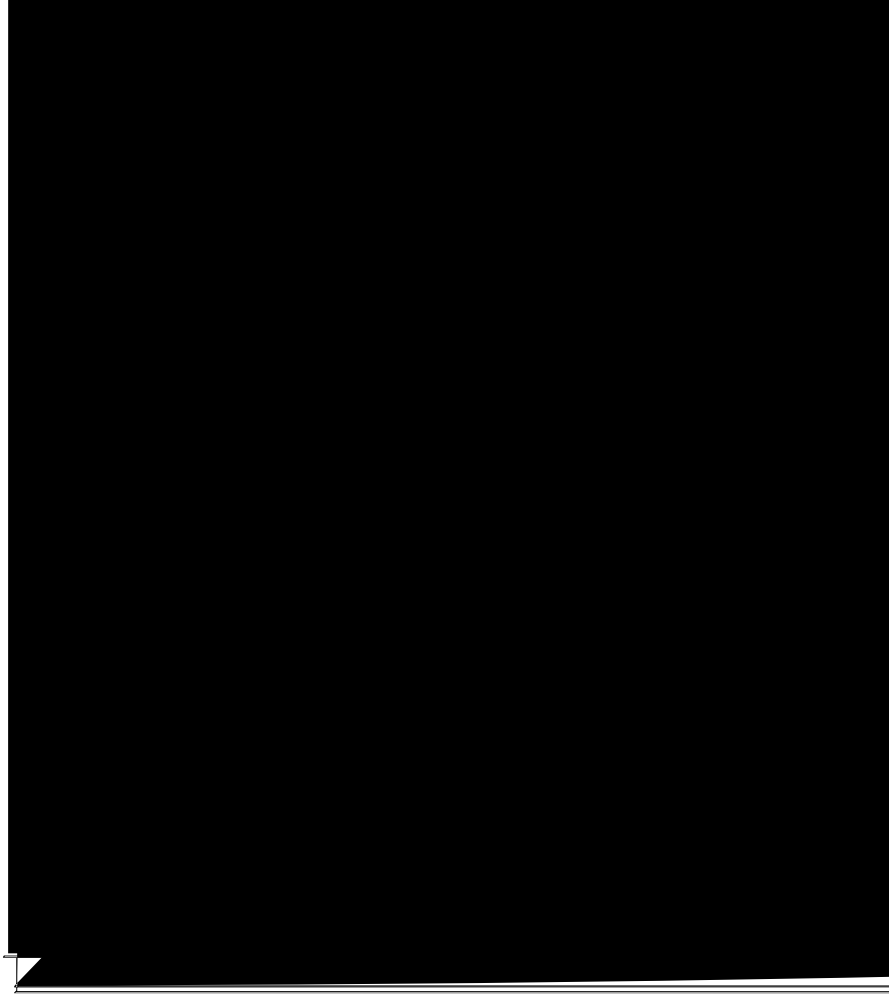


FIG. 4. Density contour for smooth shear layer. Left SPH with 90,000 particles, right finite differences with 150×200 grid [28].



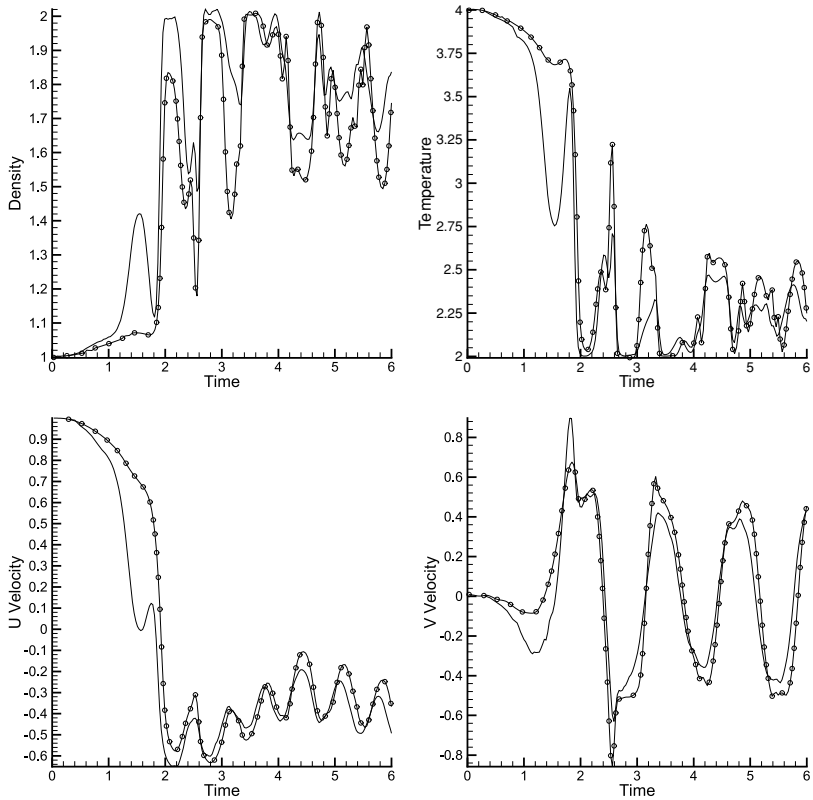


FIG. 8. Time history of density, temperature u and v velocity in point $x = 0$, $y = 0$ for a simulation with 160,000 particles. (−) SPH simulation, (-○-) finite differences [28].

in the main field and the ordinary and the one-side interpolation near the boundaries. For validation of the computations, the benchmark solution of Ghia *et al.* [31] is employed. The comparison of SPH with the numerical results of Ghia *et al*. shows very good agreement for the vorticity along the moving boundary and the boundary layers in the stationary and moving walls (Figs. 9 and 10). In the flow field the central and the secondary recirculations are predicted accurately (Fig. 11).

4.4. Natural Convection in a Differentially Heated Cavity

The SPH implementation is tested for the problem of buoyant laminar flow in a closed cavity with side walls that remain at constant but different temperatures and with insulated bottom and top walls. In order to take into account the buoyancy forces, the gravitational force is included in the momentum equation (2) and the hydrodynamic pressure in the equation of state (4). The characteristic dimensionless numbers of the simulation are

$$M = 0.1 \quad Ra = 10^3, 10^4 \quad \Pr = 0.71.$$

For the SPH simulation, 63,000 particles are used and the solution is compared to the benchmark solution of de Vahl Davis [32] (Table I). The relative error is less than 8% for all quantities tested as shown in Table I and demonstrate good agreement with the

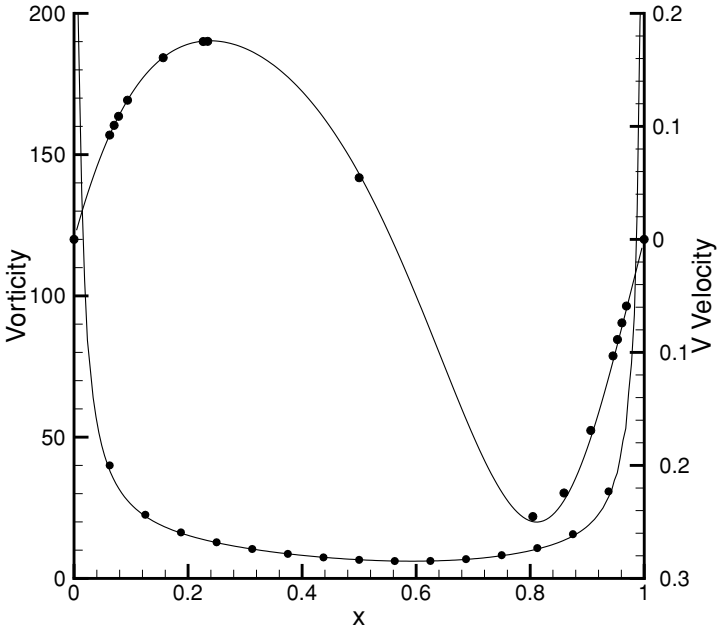


FIG. 9. Vorticity along the moving boundary, and v velocity along horizontal line through geometric center of the cavity for Reynolds 100. (−) SPH simulation, (●) Ghia *et al*. [31].

benchmark solution. Similar accuracy is observed for the flow velocity and temperature fields (Fig. 12).

4.5. Mixed Convection in a Driven Cavity

The last test case of the SPH implementation is the problem of mixed convection in a driven cavity. The top wall of a lid-driven cavity moves with constant speed and it is at a

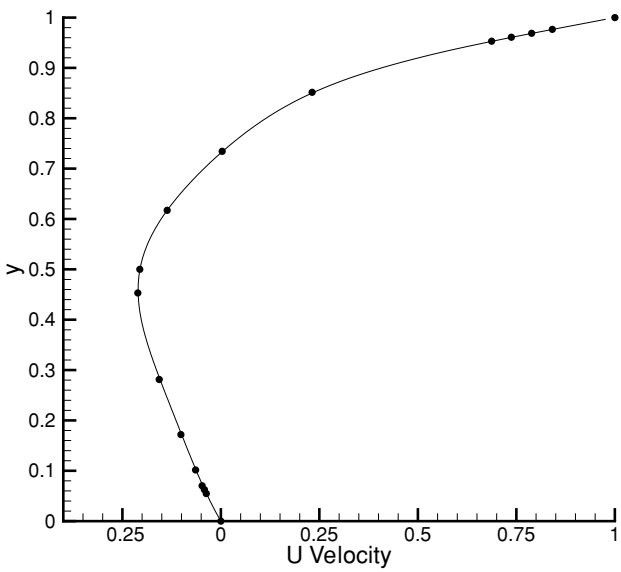


FIG. 10. u velocity along vertical line through geometric center of the cavity for Reynolds 100. (−) SPH simulation, (●) Ghia *et al.* [31].

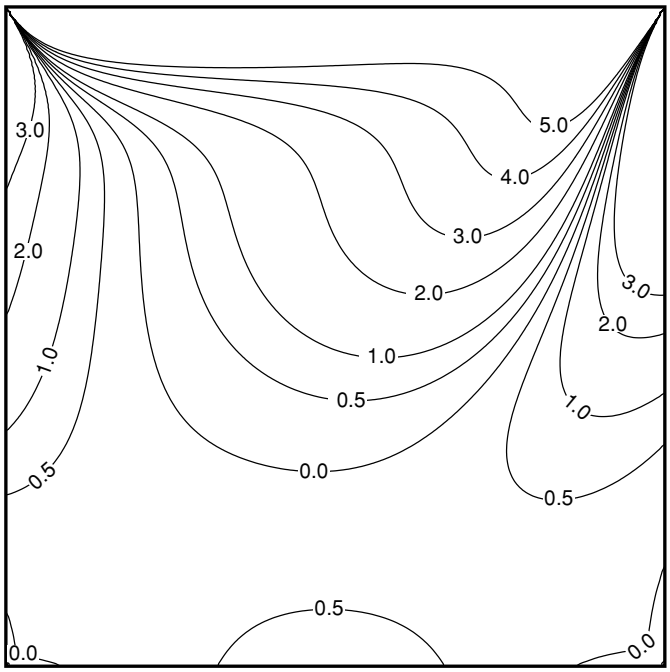


FIG. 11. Vorticity contours for flow in driven cavity (Reynolds 100).

TABLE I
Comparison of the SPH Simulation with the Benchmark Solution of Vahl Devis [32]

	10^3			10^4		
Ra	Benchmark	SPH	error %	Benchmark	SPH	error %
U_{max}	3.649	3.431	5.97	16.178	17.312	7.01
Y	0.813	0.812	0.12	0.823	0.823	0.00
V_{max}	3.697	3.511	5.03	19.617	20.051	2.21
X	0.178	0.176	1.12	0.119	0.112	5.88
Nu_{ave}	1.118	1.037	7.25	2.243	2.117	5.62
$Nu_{1/2}$	1.118	1.039	7.07	2.243	2.117	5.62
Nu_0	1.117	1.033	7.52	2.238	2.081	7.02
Nu_{max}	1.505	1.392	7.51	3.528	3.448	2.27
Y	0.092	0.098	6.52	0.143	0.136	4.90
Nu_{min}	0.692	0.705	1.88	0.586	0.541	7.68
Y	1.000	1.000	0.00	1.000	1.000	0.00

U_{max} = maximum horizontal velocity on the vertical mid-plane of the cavity and its position.

V_{max} = maximum vertical velocity on the horizontal mid-plane of the cavity and its position.

Nu_{ave} = the average Nusselt number throughout the cavity.

$Nu_{1/2}$ = the average Nusselt number on the vertical miplane.

Nu_0 = the average Nusselt number on the vertical boundary of the cavity at $x = 0$.

Nu_{max} = the maximum value of the local Nusselt number on the vertical boundary of the cavity at $x = 0$ and its position.

Nu_{min} = the minimum value of the local Nusselt number on the vertical boundary of the cavity at $x = 0$ and its position.

error = relative error between the benchmark and SPH solution.

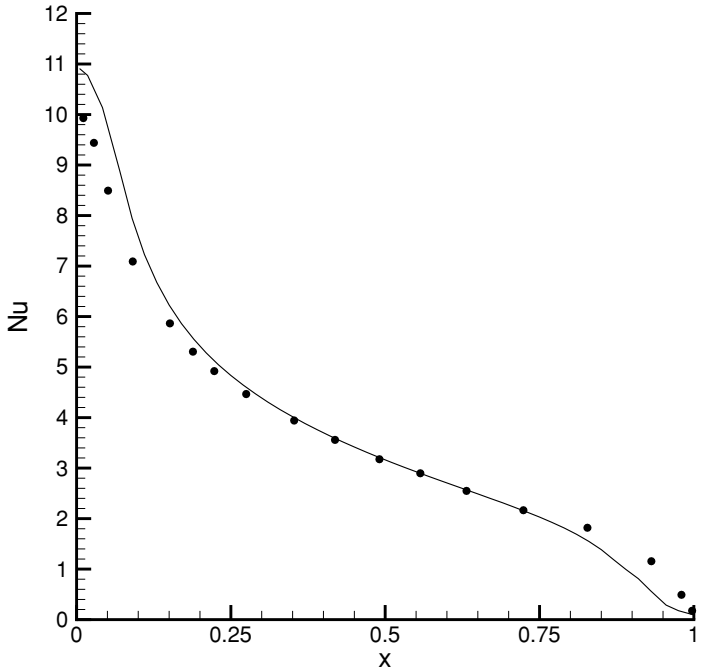


FIG. 14. Local Nusselt profile at the top wall of the cavity for Reynolds 400 and Grashoff 10^2. (−) SPH simulation with 63,000 particles, (●) Iwatsu *et al.*[33].

constant temperature (hot). The other three walls are stationary. The side walls are adiabatic and the bottom wall is isothermal (cold). The nondimensional numbers of the simulation are

$$M = 0.1 \quad \mathrm{Re} = 400 \quad \mathrm{Pr} = 0.71 \quad Gr = 10^2,$$

and 63,000 particles are used. The SPH solution is compared with the numerical results of Iwatsu *et al.* [33]. A schematic representation of the temperature profile along vertical line through geometric center of the cavity is presented in Fig. 13, which shows good agreement near the cold wall. The small difference of the temperature near the hot wall has minimal effect on the heat transfer coefficient (local Nusselt number) (Fig. 14).

5. CONCLUSIONS

The present results indicate that SPH with the implementation of remeshing is an accurate numerical method to resolve low Mach number compressible viscous conductive flows. The accuracy of the method comes with a minimal additional computational cost while maintaining the adaptive character of the method. The implementation of high-order remeshing schemes improves the accuracy of SPH and additionally increases the computational efficiency of the algorithm. The existence of high-frequency acoustic waves, which are resolved in the simulations, places a severe restriction on time-stepping increments. In the present study, it is assumed that a Mach number less than 0.3 corresponds to a practically incompressible flow, which is a reasonable assumption.

The SPH implementation as a compressible flow algorithm can compute natural convection flows with large temperature differences, thus providing us with a computational tool which exceeds the validity and therefore capability of Boussinesq-type models.

Our present work focuses on the implementation of the proposed methodology to three-dimensional flows. The tensorial character of the remeshing schemes makes it easily extendable to three dimensions and no extra algorithmic constraints are imposed by the proposed modeling of the viscous terms in conjunction with remeshing.

However, one difficulty we foresee with the present formulation of remeshing is the simulations of flows involving multiple materials and the possible smearing of shock regions in highly compressible flows. Work is underway to circumvent these difficulties by adopting special remeshing procedures that would accommodate relevant shock-capturing schemes, thus providing a robust and accurate SPH methodology.

ACKNOWLEDGMENT

The authors are grateful to Dr. N. A. Adams and B. Rembold from the Institute of Fluid Dynamics at ETH Zurich for their helpful discussions and for providing the computational simulations (two-dimensional Taylor–Green flow and two-dimensional double shear layer), which are used in the validation of the SPH implementation. We thank the anonymous referees, in particular for the remarks on regrinding versus diffusion that helped us improve of the paper.

REFERENCES

1. R. A. Gingold and J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Mon. Not. Roy. Astron. Soc.* **181**, 375 (1977).
2. J. J. Monaghan and R. A. Gingold, Shock simulation by the particle method Sph, *J. Comput. Phys.* **52**, 374 (1983).
3. J. J. Monaghan, Simulating free-surface flows with Sph, *J. Comput. Phys.* **110**, 399 (1994).
4. J. J. Monaghan and A. Kocharyan, Sph simulation of multiphase flow, *Comput. Phys. Commun.* **87**, 225 (1995).
5. J. P. Morris, P. J. Fox, and Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, *J. Comput. Phys.* **136**, 214 (1997).
6. P. W. Cleary and J. J. Monaghan, Conduction modelling using smoothed particle hydrodynamics, *J. Comput. Phys.* **148**, 227 (1999).
7. J. J. Monaghan, Smoothed particle hydrodynamics, *Annu. Rev. Astron. Astrophys.* **30**, 543 (1992).
8. I. J. Schoenberg, Contribution to the problem of approximation of equidistant data by analytic functions, *Qu. J. Appl. Math.* **4**, 45 (1946).
9. I. J. Schoenberg, Cardinal Spline Interpolation (Soc. for Industr. & Appl. Math., Philadelphia, 1973).
10. G. H. Cottet and P. D. Koumoutsakos, *Vortex Methods: Theory and Practice* (Cambridge Univ. Press, London, 2000).
11. P. Degond and S. Masgallic, The weighted particle method for convection–diffusion equations. 1. The case of an isotropic viscosity, *Math. Comput.* **53**, 485 (1989).
12. P. Degond and S. Masgallic, The weighted particle method for convection–diffusion equations. 2. The anisotropic case, *Math. Comput.* **53**, 509 (1989).
13. D. Fishelov, A new vortex scheme for viscous flows, *J. Comput. Phys.* **86**, 211 (1990).
14. G. Russo, A Lagrangian vortex method for the incompressible Navier–Stokes equations, in *Vortex Dynamics and Vortex Methods*, edited by C. Anderson and C. Greengard (Am. Math. Soc., New York, 1991).
15. S. J. Watkins, A. S. Bhattal, N. Francis, J. A. Turner, and A. P. Whitworth, A new prescription for viscosity in smoothed particle hydrodynamics, *Astron. Astrophys. Suppl. Ser.* **119**, 177 (1996).
16. L. Brookshaw, A method of calculating radiative heat diffusion in particle simulations, *Publ. Astron. Soc. Aust.* **6**, 207 (1985).

17. O. Flebbe, S. Munzel, H. Herold, H. Riffert, and H. Ruder, Smoothed particle hydrodynamics—Physical viscosity and the simulation of accretion disks, *Astrophys. J.* **431**, 754 (1994).
18. H. Takeda, S. M. Miyama, and M. Sekiya, Numerical-simulation of viscous-flow by smoothed particle hydrodynamics, *Prog. Theor. Phys.* **92**, 939 (1994).
19. P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, *J. Comput. Phys.* **138**, 821 (1997).
20. G. Lapenta and J. U. Brackbill, Control of the number of particles in fluid and Mhd particle-in-cell methods, *Comput. Phys. Commun.* **87**, 139 (1995).
21. Y. Brenier and G. H. Cottet, Convergence of particle methods with random rezoning for the 2-dimensional Euler and Navier–Stokes equations, *SIAM J. Numer. Anal.* **32**, 1080 (1995).
22. J. J. Monaghan, An introduction to Sph, *Comput. Phys. Commun.* **48**, 89 (1988).
23. J. J. Monaghan, Particle methods for hydrodynamics, *Comput. Phys. Rep.* **3**, 71 (1985).
24. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (Inst. Phys. Publ. Bristol, 1988).
25. P. Koumoutsakos and A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, *J. Fluid Mech.* **296**, 1 (1995).
26. J. T. Beale, On the accuracy of vortex methods at large times, in *Computational Fluid Dynamics and Reacting Gas Flows*, edited by B. Engquist, M. Luskin, and A. Majda (Springer-Verlag, New York, 1988).
27. G. H. Cottet, Artificial viscosity models for vortex and particle methods, *J. Comput. Phys.* **127**, 299 (1996).
28. B. Rembold and N. A. Adams, Private communication (2001).
29. N. D. Sandham and W. C. Reynolds, *A Numerical Investigation of Compressible Mixing Layer*, Report TF-45, *Thermoscience Division, Mechanical Engineering Department*, Stanford University, 1989.
30. M. L. Minion and D. L. Brown, Performance of under-resolved two-dimensional incompressible flow simulations, II, *J. Comput. Phys.* **138**, 734 (1997).
31. U. Ghia, K. N. Ghia, and C. T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, *J. Comput. Phys.* **48**, 387 (1982).
32. G. D. Davis, Natural-convection of air in a square cavity—A bench-mark numerical-solution, *Int. J. Numer. Meth. Fluids* **3**, 249 (1983).
33. R. Iwatsu, J. M. Hyun, and K. Kuwahara, Mixed convection in a driven cavity with a stable vertical temperature-gradient, *Int. J. Heat Mass Trans.* **36**, 1601 (1993).