Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 1731–1752
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Incompressible smoothed particle hydrodynamics

Marco Ellero [a,b,*], Mar Serrano [c], Pep Español [c]

[a] School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, NSW 2006, Australia
[b] Lehrstuhl für Aerodynamik, Technische Universität München, 85748 Garching, Germany
[c] Departamento de Física Fundamental, UNED, Apartado 60141, 28080 Madrid, Spain

Received 17 May 2005; received in revised form 1 June 2007; accepted 8 June 2007
Available online 29 June 2007

Abstract

We present a smoothed particle hydrodynamic model for incompressible fluids. As opposed to solving a pressure Poisson equation in order to get a divergence-free velocity field, here incompressibility is achieved by requiring as a kinematic constraint that the volume of the fluid particles is constant. We use Lagrangian multipliers to enforce this restriction. These Lagrange multipliers play the role of non-thermodynamic pressures whose actual values are fixed through the kinematic restriction. We use the SHAKE methodology familiar in constrained molecular dynamics as an efficient method for finding the non-thermodynamic pressure satisfying the constraints. The model is tested for several flow configurations.

Keywords: Incompressible flows; Smoothed particle hydrodynamics

1. Introduction

The smoothed particle hydrodynamic (SPH) method for solving the Navier–Stokes equations is a Lagrangian mesh-free model that allows one to solve the continuum hydrodynamic equations with a set of interacting fluid particles [1,2]. The original equations that are discretised are those for a compressible viscous fluid. In many real applications, the time scale of sound propagation is much smaller than the time scale of vorticity diffusion, which is basically dominated by the viscosity of the fluid. If the only relevant scale for the problem of interest is the viscous time scale, it is a waste of computer time to resolve the sonic time scale, which requires very small time steps in order to comply with the Courant condition. For a recent example illustrating these difficulties see Ref. [3]. In addition, the compressibility of the model poses problems with the sound wave reflection at boundaries. One approach in order to simulate incompressible flows with SPH is to run the simulations in the quasi-incompressible limit, that is, by selecting the smallest possible speed of sound which still gives a very low Mach number ensuring density fluctuations within 1% [4,5]. This method is known as the Weakly Compressible Smooth Particle Hydrodynamics (WCSPH). Recently a proposal for constructing an

* Corresponding author. Address: Lehrstuhl für Aerodynamik, Technische Universität München, 85748 Garching, Germany.
E-mail address: marco.ellero@aer.mw.tum.de (M. Ellero).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.06.019

incompressible SPH model has been introduced, which solves a pressure Poisson equation at every time step [6–8]. The Poisson equation arises as a consequence of the requirement that the divergence of the velocity field vanishes. We note that the divergence of the velocity is proportional to the time rate of change of infinitesimal volume elements of the fluid. If the time derivative of an infinitesimal volume element is zero, it means that any infinitesimal element evolves in such a way that it maintains its volume constant. Therefore, in the continuum limit, zeroing the divergence of the velocity field and keeping the volume of infinitesimal elements constant are equivalent approaches. In this paper we follow the last approach and formulate an incompressible SPH model by requiring that the volume of each fluid particle is a constant of motion. This set of additional dynamical invariants are actually a geometrical restriction that can be enforced through the use of Lagrange multipliers. We are following, therefore, a suggestion first made by Monaghan [4]. The advantage of the method presented here over a method that enforces that the velocity has zero divergence is that the latter does not ensure that the density field is constant. In a numerical scheme the requirement that the derivative of a function is zero not always warrants that the function is a constant, due to numerical errors that may accumulate.

The paper is distributed as follows. In Section 2 we show how to enforce the constancy of the volume of the fluid particles through Lagrange multipliers. We review in Section 3 the SHAKE method whose aim is to find an efficient numerical method to solve for the Lagrange multipliers. The implementation of the algorithm to our model is presented in Section 4. Finally, we test the method in Section 5 for the Kolmogorov and Taylor–Green flows, and present results for an elliptical drop. We conclude finally with Section 6.

2. The model

In Ref. [9] we have shown how to formulate a model within the family of smoothed particle hydrodynamics that is thermodynamically consistent. In particular, we have shown that the independent variables of each of the N fluid particles (position $\mathbf{r}_i$, velocity $\mathbf{v}_i$, and entropy S_i), evolve according to two distinct mechanisms. The first one is due to a purely reversible dynamics while the second one is due to a purely irreversible dynamics. All the issues of the incompressibility can be formulated by focusing on the reversible part of the dynamics so we restrict momentarily the discussion to it.

The reversible part of the dynamics of the SPH model can be cast in the following Hamiltonian form

$$\begin{pmatrix} \dot{\mathbf{r}}_i \\ \dot{\mathbf{v}}_i \\ \dot{S}_i \end{pmatrix} = \sum_j \frac{1}{m_i} \begin{pmatrix} \mathbf{0} & \mathbf{1}\delta_{ij} & \mathbf{0} \\ -\mathbf{1}\delta_{ij} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial E}{\partial \mathbf{r}_j} \\ \frac{\partial E}{\partial \mathbf{v}_j} \\ \frac{\partial E}{\partial S_j} \end{pmatrix}, \tag{1}$$

where the energy function is given by

$$E(x) = \sum_i \left[\frac{m_i}{2} \mathbf{v}_i^2 + \mathcal{E}_i \right]. \tag{2}$$

Here, x is a shorthand for the set of independent variables, m_i is the constant mass of particle i (with value m), and $\mathcal{E}_i = \mathcal{E}(m_i, S_i, \mathcal{V}_i)$ is the internal energy of the ith fluid particle as a function of the extensive thermodynamic variables (mass, entropy and volume) of the ith fluid particle. The volume $\mathcal{V}_i$ of the fluid particle is defined in the SPH methodology as the inverse of the density d_i which, in turn, is defined as

$$d_i = \sum_j W(|\mathbf{r}_i - \mathbf{r}_j|) = \frac{1}{\mathcal{V}_i}, \tag{3}$$

where $W(r/h)$ is a bell-shaped weight function with a finite support h and which is normalized to unity

$$\int \mathrm{d}\mathbf{r} W(r) = 1. \tag{4}$$

According to Eq. (3) the volume of each fluid particle is an analytical function of the positions of the neighbouring fluid particles. Note that if there are many neighbouring particles around a given one, the

contribution to the sum in Eq. (3) will be large and so will be the density of the particle. Its associated volume will be thus smaller. For future reference, we introduce also the positive function $F(r)$ through

$$\nabla W(r) = -\mathbf{r}F(r), \quad F(r) \geqslant 0. \tag{5}$$

A usual selection in SPH for $W(r)$ is the Lucy function,

$$W(r) = \frac{5}{\pi h^2}\left(1 + 3\frac{r}{h}\right)\left(1 - \frac{r}{h}\right)^3 \theta(r/h), \tag{6}$$

where $\theta(x)$ is a step function that takes the value 1 if $0 \leqslant x \leqslant 1$ and zero otherwise. The prefactor in Eq. (6) ensures the normalization (4) in two spatial dimensions. The function $F(r)$ follows

$$F(r) = \frac{60}{\pi h^4}\left(1 - \frac{r}{h}\right)^2 \theta(r/h). \tag{7}$$

We can perform explicitly the derivatives of the energy function with respect to the state variables appearing in Eq. (1), with the result

$$\begin{pmatrix} \frac{\partial E}{\partial \mathbf{r}_j} \\ \frac{\partial E}{\partial \mathbf{v}_j} \\ \frac{\partial E}{\partial S_j} \end{pmatrix} = \begin{pmatrix} -\sum_k \frac{\partial \mathcal{V}_k}{\partial \mathbf{r}_j} P_k \\ m\mathbf{v}_j \\ T_j \end{pmatrix}, \tag{8}$$

where we have introduced the pressure P_i and temperature T_i of the fluid particle i, which are defined in the usual thermodynamic sense

$$\begin{aligned} P_i &= -\frac{\partial \mathcal{E}_i}{\partial \mathcal{V}_i}, \\ T_i &= \frac{\partial \mathcal{E}_i}{\partial S_i}. \end{aligned} \tag{9}$$

The equations of motion that result from (1) are then

$$\begin{aligned} \dot{\mathbf{r}}_i &= \mathbf{v}_i, \\ m\dot{\mathbf{v}}_i &= \sum_j \frac{\partial \mathcal{V}_j}{\partial \mathbf{r}_i} P_j, \\ \dot{S}_i &= 0. \end{aligned} \tag{10}$$

These equations conserve the total energy defined in Eq. (2), as a consequence of the fact that the matrix in Eq. (1) is antisymmetric. They also conserve linear momentum, as a consequence of the invariance under translations of the total energy. This translational invariance is due to the corresponding invariance of the volume, i.e.

$$\mathcal{V}_i(\mathbf{r}_1 + \mathbf{a}, \ldots, \mathbf{r}_N + \mathbf{a}) = \mathcal{V}_i(\mathbf{r}_1, \ldots, \mathbf{r}_N), \tag{11}$$

where $\mathbf{a}$ is an arbitrary translation vector. By deriving with respect to $\mathbf{a}$ and setting $\mathbf{a} = 0$ at the end, we find the identity

$$\sum_i \frac{\partial \mathcal{V}_j}{\partial \mathbf{r}_i} = 0. \tag{12}$$

This property ensures that total momentum is conserved by Eq. (10).

In order to describe an incompressible situation, we want to enforce the geometrical restriction that the volume of each fluid particle is a constant of motion. Note that the rate of change of the volume of the particles is, according to the chain rule and the first Eq. (10)

$$\dot{\mathcal{V}}_i = \sum_j \frac{\partial \mathcal{V}_i}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j. \tag{13}$$

This equation allows us to interprete

$$[\nabla \cdot \mathbf{v}]_i \equiv \frac{1}{\mathcal{V}_i} \sum_j \frac{\partial \mathcal{V}_i}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j, \tag{14}$$

as a discrete version of the divergence of the velocity. Instead of enforcing incompressibility by requiring that the divergence of the velocity field is zero, this is, that the time derivative of the volume is zero, we enforce incompressibility by requiring that the volume of every fluid particle is constant, this is,

$$\mathcal{V}_i(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \mathcal{V}_i^0, \quad i = 1, \ldots, N, \tag{15}$$

where $\mathcal{V}_i^0$ is the volume of particle i at time $t = 0$.

The usual way in order to enforce a constraint in classical mechanics is through the use of Lagrange multipliers [10]. In our case, the method of Lagrange multipliers transforms the set of Eq. (10) into the following set

$$\begin{aligned} \dot{\mathbf{r}}_i &= \mathbf{v}_i, \\ m\dot{\mathbf{v}}_i &= \sum_j \frac{\partial \mathcal{V}_j}{\partial \mathbf{r}_i} P_j + \sum_j \frac{\partial \mathcal{V}_j}{\partial \mathbf{r}_i} \lambda_j, \\ \dot{S}_i &= 0, \end{aligned} \tag{16}$$

where λ_j are N Lagrange multipliers, whose actual value is fixed by the fact that the N restrictions (15) are satisfied.

Note for completeness, that the modified equations of motion can still be cast in a Hamiltonian form,

$$\begin{pmatrix} \dot{\mathbf{r}}_i \\ \dot{\mathbf{v}}_i \\ \dot{S}_i \end{pmatrix} = \sum_j \begin{pmatrix} \mathbf{0} & \frac{1}{m}\delta_{ij} & \mathbf{0} \\ -\frac{1}{m}\delta_{ij} & \mathbf{0} & \frac{\partial \mathcal{V}_j}{\partial \mathbf{r}_i} \frac{\lambda_j}{mT_j} \\ \mathbf{0} & -\frac{\partial \mathcal{V}_i}{\partial \mathbf{r}_j}^T \frac{\lambda_i}{mT_i} & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial E}{\partial \mathbf{r}_j} \\ \frac{\partial E}{\partial \mathbf{v}_j} \\ \frac{\partial E}{\partial S_j} \end{pmatrix}. \tag{17}$$

The matrix above is antisymmetric and this automatically ensures that the energy function is conserved by Eq. (17). It is a matter of easy calculation to show that the position and velocity equations reduce to (16), whereas the entropy equation becomes

$$\dot{S}_i = -\frac{\lambda_i}{T_i} \sum_j \frac{\partial \mathcal{V}_i}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j = 0, \tag{18}$$

where we have used Eq. (13) and the fact that, now, $\dot{\mathcal{V}}_i = 0$. Therefore, Eq. (16) represent the Hamiltonian dynamics of a system of fluid particles that move with constant volumes.

By using the definition of the volume (3), we obtain

$$\frac{\partial \mathcal{V}_j}{\partial \mathbf{r}_i} = \frac{1}{d_j^2} \left[\delta_{ij} \sum_k F_{ik} \mathbf{r}_{ik} + F_{ij} \mathbf{r}_{ij} \right], \tag{19}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and Eq. (16) become

$$\begin{aligned} \dot{\mathbf{r}}_i &= \mathbf{v}_i, \\ m\dot{\mathbf{v}}_i &= \sum_j F_{ij} \mathbf{r}_{ij} \left[\frac{P_i + \lambda_i}{d_i^2} + \frac{P_j + \lambda_j}{d_j^2} \right], \\ \dot{S}_i &= 0. \end{aligned} \tag{20}$$

For compressible fluids, when $\lambda_i = 0$, these equations are the SPH symmetrized equations preferred by Monaghan [4]. As time proceeds, the values of λ_j of each fluid particle adjust themselves in such a way

to maintain constant the volume of the fluid particles, despite of the fact that the fluid particles move. The Lagrange multipliers can be interpreted as a non-thermodynamic pressure that adds up to the thermodynamic pressure.

Eq. (20) are a discrete representation of the inviscid Euler equations, which are purely reversible. In order to describe a viscous fluid, viscous forces have to be included in Eq. (20). The full set of equations, in this case, reads [9]

$$\begin{aligned} &\dot{\mathbf{r}}_i = \mathbf{v}_i, \\ &m\dot{\mathbf{v}}_i = \sum_j F_{ij}\mathbf{r}_{ij}\left[\frac{P_i+\lambda_i}{d_i^2}+\frac{P_j+\lambda_j}{d_j^2}\right], -\frac{\eta}{3}\sum_j \frac{F_{ij}}{d_i d_j}[4\mathbf{e}_{ij}\mathbf{e}_{ij}\cdot\mathbf{v}_{ij}+5\mathbf{v}_{ij}], \end{aligned} \tag{21}$$

where η is the shear viscosity (bulk viscosity is neglected for simplicity), $\mathbf{e}_{ij} = \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$ and $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$.

For an incompressible fluid there is no need to introduce an equation of state. As we will be comparing the incompressible algorithm with a compressible one, we need an equation of state for this last one. We adopt the following one

$$P(\rho) = \frac{\rho_0 c^2}{2}\left[\left(\frac{\rho}{\rho_0}\right)^2 - 1\right], \tag{22}$$

where c is the speed of sound and ρ_0 is the equilibrium mass density. With this equation of state which does not depend on the entropy variable, there is no coupling with the dynamic equation for this variable, which will not be considered in the sequel.

3. The Newton–Raphson and SHAKE methodologies

The N Lagrange multipliers λ_i take values such that as time proceeds, the N holonomic constraints (15) are satisfied. The situation here is closely reminiscent to that encountered in the molecular dynamics simulations of molecules with constraints. For this type of problem a powerful methodology, known as SHAKE, has been developed (see [11] for a neat interpretation of the method). The SHAKE method is just a particularly simple case of the Newton–Raphson method. The idea is summarized here.

Let $x = \{x_k, k = 1,\ldots,N\}$ denote the state of the system at time t. We can write formally the equations of motion (21) as

$$\dot{x} = F(x,\lambda), \quad \sigma(x(t,\lambda)) = 0, \tag{23}$$

where $\lambda = \{\lambda_k, k = 1,\ldots,N\}$ denotes the set of N Lagrange multipliers, $x(t,\lambda)$ is the formal solution of the equations of motion and $\sigma(x(t,\lambda)) = 0$ are the set of N constraints. Given an integrator method like, for example, a Runge–Kutta or a Predictor–Corrector (P–C) method, we will have after a time step Δt a new state x' which, in general, will depend on the Lagrange multipliers $x'(\lambda)$. The particular functional dependence on λ depends on the integrator. The new state satisfies the N constraints

$$\sigma(x'(\lambda)) = 0, \tag{24}$$

or in a more compact notation

$$\sigma(\lambda) = 0 \tag{25}$$

for the N-vector λ. This is a set of N non-linear equations for the set of N unknowns, the Lagrange multipliers λ at time t. The solution of a set of non-linear equations is very difficult to obtain in general (see Ref. [12]). However, we have to solve (25) for every time step, and we assume that we have solved the problem in the previous time step, while at the new time step the positions of the particles have changed by small amount (proportional to Δt). This means that the solution of (25) for λ in the previous time step is a good guess for the initial value of an iterative process. In this case, we expect a brute force calculation to work reasonably well. The Newton–Raphson method is such a brute force method.

The Newton–Raphson method is based on the Taylor expansion

$$\sigma(\lambda+\delta\lambda)=\sigma(\lambda)+\mathbf{B}(\lambda)\delta\lambda+\mathcal{O}(\delta\lambda^2), \tag{26}$$

where

$$\mathbf{B}(\lambda)\equiv\frac{\partial\sigma}{\partial\lambda}. \tag{27}$$

Now, assume that λ^n is a good guess for the solution of (25). Then, it can be improved by adding a correction $\delta\lambda=\lambda^{n+1}-\lambda^n$ such that $\sigma(\lambda^n+\delta\lambda)=0$. Then Eq. (26) leads to

$$\delta\lambda=-\mathbf{B}^{-1}(\lambda^n)\sigma(\lambda^n)+\mathcal{O}(\delta\lambda^2). \tag{28}$$

By neglecting higher order terms this can be written as

$$\lambda^{n+1}=\lambda^n-\mathbf{B}^{-1}(\lambda^n)\sigma(\lambda^n), \tag{29}$$

which allows for an iterative solution provided that the iteration converges. In this case, $\lambda^{n+1}\approx\lambda^n$, implying $\sigma(\lambda^n)\approx 0$. Here, n labels the iteration during a given time step.

The Newton–Raphson scheme suffers from the problem of having to invert the $N\times N$ matrix $\mathbf{B}$, which is computationally expensive (i.e. for a LU decomposition we need $\mathcal{O}(N^3)$ operations), despite of the fact that in our particular problem it turns out that $\mathbf{B}$ is a sparse matrix. The SHAKE method is an attempt to avoid this matrix inversion. Let us see how it works. Write Eq. (29) in compact matrix form

$$\mathbf{B}^n(\lambda^{n+1}-\lambda^n)=-\sigma(\lambda^n), \tag{30}$$

where $\mathbf{B}^n=\mathbf{B}(\lambda^n)$. Consider the matrix $\mathbf{S}^n=\mathrm{Diag}[\mathbf{B}^n]^{-1}$, that is, the inverse of the diagonal of the matrix $\mathbf{B}^n$. The matrix $\mathbf{S}^n\mathbf{B}^n$ has the number 1 in the diagonal, so we may write $\mathbf{S}^n\mathbf{B}^n=\mathbf{1}-\mathbf{A}_n$, where $\mathbf{A}_n$ has zero in the diagonal. By multiplying (30) with $\mathbf{S}^n$ we can finally write

$$\lambda^{n+1}-\lambda^n=(\mathbf{1}-\mathbf{A}_n)^{-1}\mathbf{S}^n\sigma(\lambda^n). \tag{31}$$

Now, we can invert $\mathbf{1}-\mathbf{A}_n$ through the series expansion

$$(\mathbf{1}-\mathbf{A}_n)^{-1}=\mathbf{1}+\mathbf{A}_n+\mathbf{A}_n^2+\mathbf{A}_n^3+\cdots \tag{32}$$

If the series can be truncated, say with a few terms, then we only have to perform a few multiplications of a matrix and a vector ($\mathcal{O}(N^2)$ process). In the luckiest case, we can use the maximum truncation of Eq. (32), this is, $(\mathbf{1}-\mathbf{A}_n)^{-1}\approx\mathbf{1}$. This results in the extremely simple iterative algorithm

$$\lambda^{n+1}=\lambda^n-\mathbf{S}^n\sigma(\lambda^n). \tag{33}$$

This algorithm, provided the iteration converges, is very efficient, because the matrix $\mathbf{S}^n$ is diagonal. Note that the error due to truncation in Eq. (32) plays no crucial role provided that the iteration for λ converges. For if it converges, it means $\lambda^{n+1}-\lambda^n=0$ and, therefore in Eq. (31) $\sigma(\lambda^n)=0$, so we get the correct λ that satisfies the constraint. In that respect, the algorithm in Eq. (33) can be further simplified by using instead of $\mathbf{S}^n$, which has to be computed in each iteration, just the value of $\mathbf{S}^0$ evaluated at the first step of the iteration for $\lambda^0=0$. As we show in the Appendix, this is the SHAKE method. As a last remark, let us notice that using SHAKE the iteration requires only a $\mathcal{O}(N)$ number of operations. Additionally, we bypass the problem of storing at every iteration the N^2 elements of the matrix $\mathbf{B}$, as only its N diagonal elements are relevant for the numerical loop.

4. The algorithm

We will solve the differential equations (21) by a second order integrator. The positions are updated according to

$$\mathbf{r}_i'(\lambda)=\mathbf{r}_i^0+\sum_j\mathbf{\Omega}_{ij}\lambda_j, \tag{34}$$

where $\mathbf{r}_i^0$ are the particle positions evaluated after an unconstrained time step by using a second order integrator, i.e. Predictor–Corrector (P–C), while the second term represents the corrections in the positions according to the constraint condition (24). The coefficients in (34) read

$$\mathbf{\Omega}_{ij} = \frac{\Delta t^2}{2d_j^2 m}\left[\delta_{ij}\sum_k F_{ik}\mathbf{r}_{ik} + F_{ij}\mathbf{r}_{ij}\right]. \tag{35}$$

The velocities are updated according to

$$\mathbf{v}_i'(\lambda) = \mathbf{v}_i^0 + \frac{2}{\Delta t}\sum_j \mathbf{\Omega}_{ij}\lambda_j, \tag{36}$$

where $\mathbf{v}^0$ are the unconstrained particle velocities. In Eqs. (34) and (36), $\mathbf{r}_i^0, \mathbf{v}_i^0$ are the result of performing a second order (P–C) time step of the compressible model.

The new positions $\mathbf{r}_i'(\lambda)$ should satisfy the constraint (15). In the present case, the constraint in Eq. (24) can be cast in the following form involving number densities

$$\sigma_i(\lambda) = \sum_j W(r_{ij}'(\lambda)) - d_i^0, \tag{37}$$

where $r_{ij}'(\lambda) = |\mathbf{r}_i'(\lambda) - \mathbf{r}_j'(\lambda)|$ and $d_i^0 = 1/\mathcal{V}_i^0$. The matrix $\mathbf{B}$ defined in Eq. (27) becomes

$$\begin{aligned}\mathbf{B}_{ik} &= \frac{\partial\sigma_i(\lambda)}{\partial\lambda_k} = \frac{\partial}{\partial\lambda_k}\sum_j W(r_{ij}'(\lambda)) = -\sum_j F(r_{ij}'(\lambda))\mathbf{r}_{ij}'(\lambda)\cdot\left[\frac{\partial\mathbf{r}_i'(\lambda)}{\partial\lambda_k} - \frac{\partial\mathbf{r}_j'(\lambda)}{\partial\lambda_k}\right]\\ &= \sum_j F(r_{ij}'(\lambda))\mathbf{r}_{ij}'(\lambda)\cdot\left[\mathbf{\Omega}_{jk}(\lambda) - \mathbf{\Omega}_{ik}(\lambda)\right].\end{aligned} \tag{38}$$

The iteration (33) now becomes

$$\lambda_i^{n+1} = \lambda_i^n - \frac{1}{B_{ii}^0}[d_i^n - d_i^0], \tag{39}$$

where

$$\begin{aligned} d_i^n &\equiv \sum_j W(|\mathbf{r}_i'(\lambda^n) - \mathbf{r}_j'(\lambda^n)|),\\ B_{ii}^0 &= \sum_j F(r_{ij}'(\lambda))\mathbf{r}_{ij}'(\lambda)\cdot[\mathbf{\Omega}_{ji}(\lambda) - \mathbf{\Omega}_{ii}(\lambda)]|_{\lambda=\lambda_0}.\end{aligned} \tag{40}$$

Once Eq. (39) is iterated to the desired convergence, the output of the subroutine is $\mathbf{r}_i'(\lambda)$ and the set of λ_i that fulfill the constraints. In order to finish the time step, we need to compute the new velocities according to Eq. (36).

We should point out that at each iteration we have to compute first the positions $\mathbf{r}_i'(\lambda^n)$, through Eq. (34) and the current value of λ^n. Then we have to compute the actual density of the fluid particles. This means that in each iteration we must loop over all the particles in the system. Although it appears that the computational effort is order N^2 because we have a loop within a loop, the coding can be done in such a way that the effort is of order $2N$: Before entering the loop on the components of λ, compute the densities $\sum_j W(\mathbf{r}_i'(\lambda^n) - \mathbf{r}_j'(\lambda^n))$. In this way, instead of a nested loop, we have two consecutive loops. Actually we have to call two times the linked-list cell routine which has an algorithmic complexity of $\mathcal{O}(N\log N)$ instead of $\mathcal{O}(N)$.

A last remark on the incompressible algorithm described above is here in order. As already mentioned in this section, the strategy adopted consists on splitting the dynamics in two parts which correspond to a first compressible step (that provides $\mathbf{r}_i^0, \mathbf{v}_i^0$) followed by a second one (containing the Lagrange multipliers) able to restore exact incompressibility. A question which arises naturally is therefore, which kind of compressible dynamics (i.e. equation of state) should be used in this intermediate step and, in particular, whether the resulting Courant condition for this intermediate step limits the time step. Note, however that in Eq. (21) we can introduce a new Lagrange multiplier as $\lambda_i' = P_i + \lambda_i$ and use the SHAKE algorithm for computing λ_i' directly.

This bypass the need of an equation of state at the intermediate step or, equivalently, allows to set $P_i = 0$ in the incompressible algorithm.

5. Numerical results

In this section we present the comparison between results of the simulation of the incompressible viscous Eq. (21) and the correspondent weakly compressible equations obtained by neglecting the Lagrange multipliers in Eq. (21).

5.1. Kolmogorov flow

As a benchmark problem, we consider first the so-called Kolmogorov flow, already studied by Posch and Hoover via Smoothed Particle Hydrodynamics simulations [13]. The problem can be described as follows: a Newtonian fluid is defined on a unbounded domain and a body force is applied acting along the x-direction and periodic in the transverse y-direction. The scheme of the body force is shown in Fig. 1. This is an interesting test for an hydrodynamic code because it deals specifically with the transition from a laminar to a turbulent flow. Also the fact that the external forcing is periodic, allows for the use of periodic boundary conditions. As has been shown in [6], different implementations of boundary conditions on solid walls affect strongly the nature of the flow. The Kolmogorov flow, being periodic in space, does not suffer from this problem, thus allowing for a clean comparison between different models.

Let us consider a body force (per unit mass) of the following form:

$$\mathbf{F} = F_0 \sin(k_y y)\widehat{\mathbf{x}}, \tag{41}$$

where $k_y = 2\pi/L_y$ is the wave vector and L_y is the length of the simulation box in the y-direction. For small values of F_0 (in a sense to be clarified later), a theoretical solution can be obtained for the velocity field which has the same functional form as the perturbation

$$\mathbf{v} = v_0 \sin(k_y y)\widehat{\mathbf{x}}, \tag{42}$$

with $v_0 = F_0/\nu k_y^2$ and ν is the kinematic viscosity. Nevertheless, for increasing values of the applied force F_0, the Kolmogorov flow is known to be unstable and a secondary steady flow appears which consists of a periodic configuration of stationary vortices [13,14]. The stability threshold depends upon the Reynolds number which, for this particular flow, reads

$$Re = \frac{F_0}{\nu^2 k_y^3}. \tag{43}$$

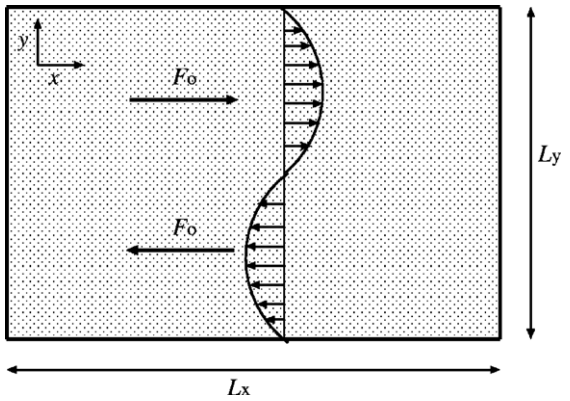

Fig. 1. Sinusoidal body force $\mathbf{F} = F_0 \sin(k_y y)\hat{\mathbf{x}}$ in the Kolmogorov flow. L_x and L_y are the size of the simulation box in the x and y directions, respectively, and the spatial frequency is $k_y = 2\pi/L_y$.

For values of the fluid parameters, corresponding to $Re > Re_c$, the flow develops the instability and secondary flow appears. Here, Re_c is a critical Reynolds number which, in the limiting case of infinite aspect ratio ($\alpha = L_y/L_x = 0$), assumes the value $\sqrt{2}$. For increasing values of the Reynolds number, even the secondary flow starts to exhibit oscillations, and becomes unstable at $Re \approx 2.5 Re_c$. For larger values of Re higher harmonics appear in the flow and eventually, it becomes turbulent [14].

In this section we compare the results of a Weakly Compressible SPH algorithm (WCSPH) with those obtained by using the Incompressible SHAKE-based algorithm (ISPH) as represented in Eq. (21). The comparison is done for values of the fluid parameters giving a Reynolds number sufficiently large to exhibit hydrodynamic instability (i.e. secondary flow): in our case we consider $Re = 5$. Note that for the finite aspect ratio $\alpha = 0.4$ used in this paper, the critical Reynolds number for secondary flow is $Re_c \approx 2.5$, and therefore $Re = 5$ corresponds to a stable steady state for the secondary eddies. We used a total number of $N = 4000$ SPH particles, initially placed on a regular square lattice at distance $d = 0.025$ apart (so all the particles have the same volume). The simulation domain is $[0, 2.5] \times [0, 1]$. The kernel used was the Lucy function in Eq. (6) with a cutoff radius $h = 0.075$, which involves an average number of neighbouring particles approximately equal to 20. For the WCSPH algorithm, the equation of state (22) was used with a Mach number $M = v_0/c = 0.1$ which permits us to reduce the density fluctuations $\Delta\rho/\rho$ to the order of 1%. We encourage the instability by imposing a small initial velocity field given by $\mathbf{V} = (V_x, V_y) = 0, \delta\varpi \sin(2\pi x/L_x)\cos(2\pi y/L_y)$ with $\delta = 0.01$.

As already noticed, the weakly compressible approach needs to resolve the time scales up to the smallest characteristic sonic time. For relative large values of the Reynolds number, such a sonic time scales are much smaller than the relevant viscous ones describing vorticity decay, and therefore, it is waste of computer time to resolve them. This can be seen by looking directly at the stability condition for the time step, based on the viscous terms, which reads [6]

$$\Delta t \leqslant 0.125 \frac{h^2}{v}, \tag{44}$$

where h is meant to be the spatial discretization length. It is well known that, for relative large Reynolds numbers, this condition is not dominant and one has to satisfy the strictest CFL condition based on the speed of sound, which reads

$$\Delta t \leqslant 0.25 \frac{h}{c}. \tag{45}$$

This condition ensures that there is no numerical propagation of signals faster than the speed of sound c. On the other hand, if incompressibility can be "artificially" maintained within the flow (i.e. by solving the Poisson equation for the pressure [6,7] or, as in our case, by using the SHAKE-iteration), the constraint (45) can be relaxed to

$$\Delta t \leqslant 0.25 \frac{h}{v_{\max}}, \tag{46}$$

where $v_{\max}$ is the maximum flow velocity. In the case of Mach number $M = v_{\max}/c \approx 0.1$, the previous choice corresponds to a possible increase in the time step Δt of 10 times respect to the weakly compressible case. However, it should be noticed that a choice of $M = 0.1$ does not always guarantee density fluctuations of 1%. Under particular flow conditions (for example those characterized by stagnation points, i.e. flow around a cylinder) it could be necessary to reduce further the Mach number (and therefore Δt) in order to satisfy $\Delta\rho/\rho < 0.01$. In this case, the advantages of an incompressible flow solver (i.e. Poisson or SHAKE) are likely to be even more pronounced.

Fig. 2 (top) shows the Kolmogorov steady-state velocity flow field evaluated via the WCSPH algorithm with $Re = 5$ and $M = 0.1$. The secondary flow characterized by a regular configuration of stationary vortices is clearly visible. Analogously to the previous case, an ISPH simulation has been carried out for the same value of the Reynolds number. Nevertheless, for the resolution h and the small Reynolds number considered here, the viscous condition (44) was found to be dominant and no pronounced speed-up in the overall calculation was observed. In the best case (i.e. for the coarsest resolution only a factor 2 was gained). This is consistent

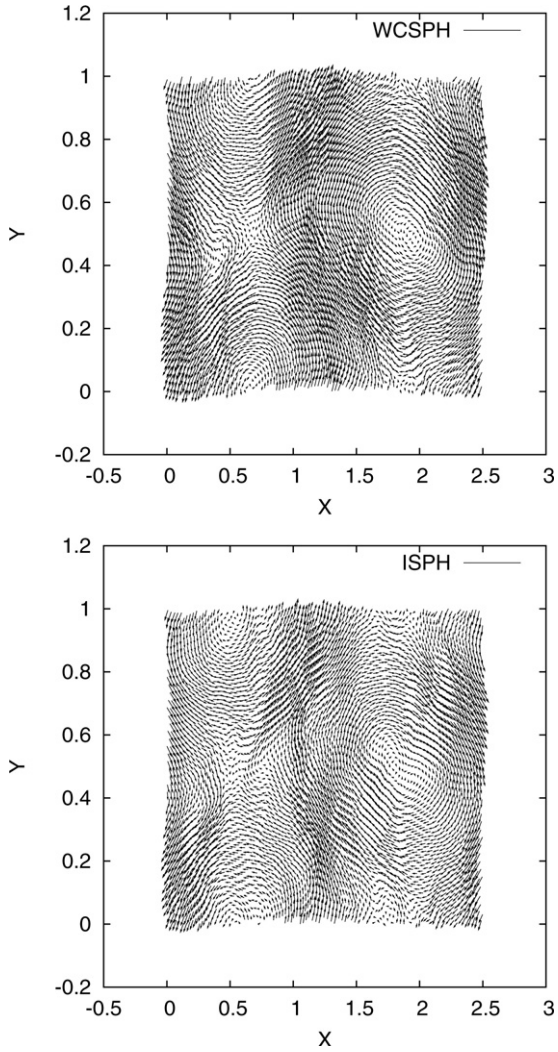

Fig. 2. Steady-state secondary flow evaluated via the WCSPH (top) vs. ISPH (bottom).

with the fact that, for this small Reynolds number, the viscous condition starts to affect only the ISPH time step, therefore limiting its overall performance. Fig. 2 (bottom) shows the steady-state solution evaluated via ISPH. Also in this case, a steady secondary flow is obtained, which is in close agreement with the previous WCSPH simulation shown in Fig. 2 (top).

In order to check the accuracy of the incompressible approximation, we plotted in Fig. 3 (top) the particle-values of the mass density ρ at the steady state (corresponding to the flow fields shown in Fig. 2) as a function of the x-coordinate of the fluid particles. The SHAKE-cycle (39) is iterated n times until the condition

$$\max_i |\rho_i^n - \rho_i^0| < \epsilon, \tag{47}$$

over each particle is satisfied, with $\epsilon = 10^{-2}$. For this tolerance, the ISPH technique should obtain values for density field comparable with WCSPH. In practice, WCSPH shows deviations $\Delta\rho \approx 0.03$ around the averaged density value, while the $\Delta\rho$ for ISPH are consistent with ϵ and correspond to the region in Fig. 3 (top) between the upper and lower lines (the central line corresponds to the initial constant density ρ_0). Notice that both methods seem to achieve a steady-state density field centered around an average value slightly larger than ρ_0, but the ISPH results appear much more confined.

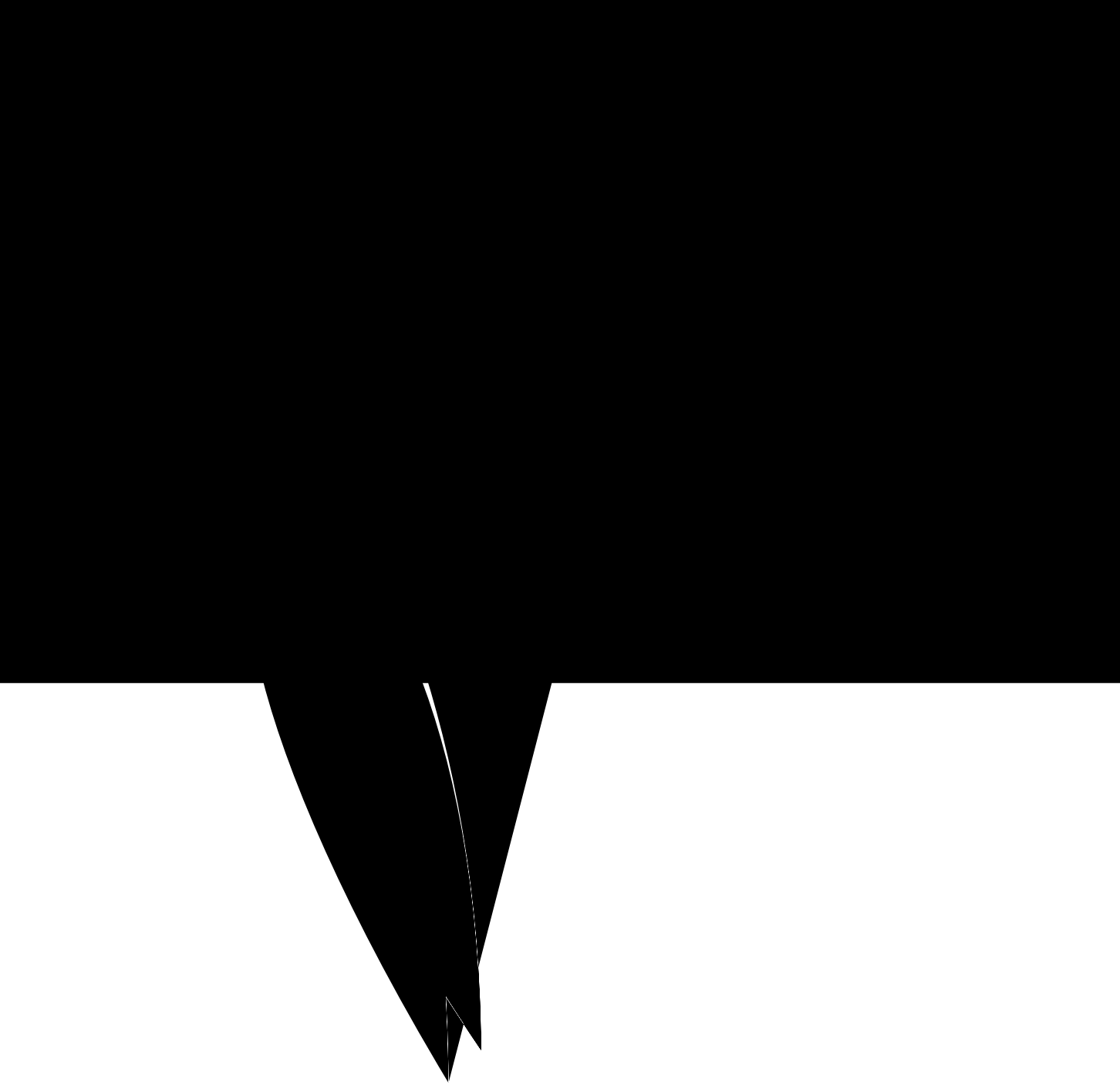

Fig. 4

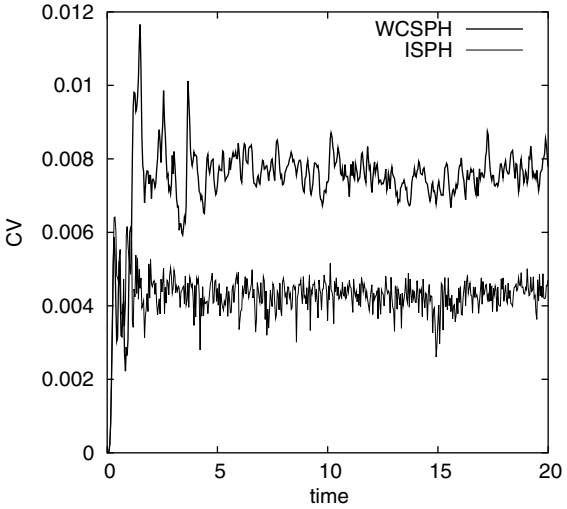

Fig. 4. The spread of the values of the density in Fig. 3 (top) can be quantified with the CV coefficient defined in Eq. (48). Note the improved behaviour of ISPH as compared to WCSPH.

a clear unwanted increasing trend was observed [6]. This is an important difference between the two techniques because it shows that the SHAKE-based algorithm is able to satisfy incompressibility at every time step during the simulation, avoiding the large density variation errors (cumulating in time) which come from the solution of a Poisson equation for the pressure [6]. We will discuss in detail this aspect in the next section.

5.2. Taylor–Green flow

As a further validation for the incompressible SPH flow calculations based on SHAKE, we consider the time decay of a two-dimensional periodic array of vortices. The problem has an analytical solution and it has been already studied via SPH [15]. In 2D, the solution reads:

$$\begin{aligned} v_x(x,y,t) &= -v_0 \mathrm{e}^{-\frac{8\pi^2 t}{Re}} \cos(k_0 x)\cos(k_0 y), \\ v_y(x,y,t) &= +v_0 \mathrm{e}^{-\frac{8\pi^2 t}{Re}} \sin(k_0 x)\cos(k_0 y), \end{aligned} \tag{49}$$

where $v_0 = 1$ and $k_0 = 2\pi/L$. The computational domain is $[0,L]\times[0,L]$ with $L = 1$. According to previous numerical calculation [15], we adopt as an error estimate the relative error L_∞ defined in the following way

$$L_\infty = \max_{t=0}^{t_{\max}} \left(\left| \frac{v^{\mathrm{th}}(t) - v^{\mathrm{sph}}(t)}{v^{\mathrm{th}}(t)} \right| \right), \tag{50}$$

where $v^{\mathrm{th}}(t)$ and $v^{\mathrm{sph}}(t)$ are, respectively, the maximum theoretical and the maximum SPH velocities at the time t. $t_{\max}$ is the final time defined in such a way that the maximum velocity decays below the value $v^{\mathrm{th}}(t_{\max}) = v^{\mathrm{th}}(0)/50$. Note that L_∞ gives an estimate of the maximum relative error during the whole simulation.

A first simulation is run at a small Reynolds number $Re = 0.1$. The particles are initially at the nodes of a regular mesh and the typical particle paths for this flow are shown in Fig. 5. The Mach number is set equal to 0.1 for WCSPH, while for ISPH a tolerance density error $\epsilon = 0.01$ was considered. Fig. 6 shows the typical viscous decay of the maximum velocity for WCSPH and ISPH compared with the analytical solution in Eq. (49). The total number of particles is $N = 10{,}000$. For this small Reynolds number, there are no visible differences between WCSPH, ISPH and the theory.

In order to estimate the error in the two methods, we plot in Fig. 7 the relative error L_∞ of the ISPH vs. WCSPH for different resolutions ranging from 400 to 10,000 particles and $Re = 0.1$. Both methods exhibit

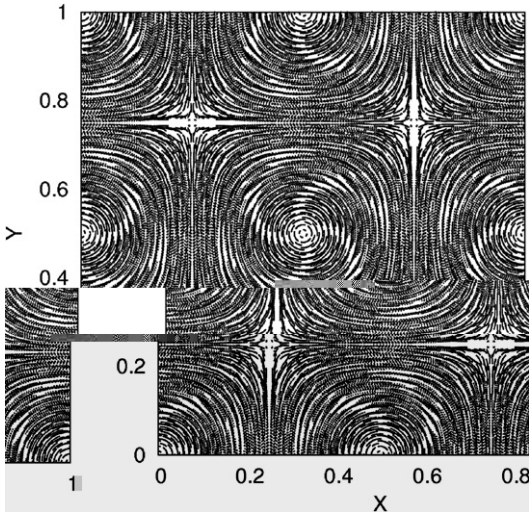

Fig. 5. Particle paths for the Taylor–Green flow for $Re = 0.1$.

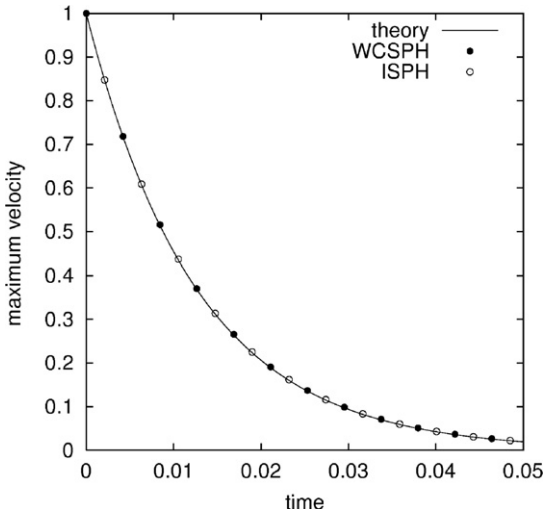

Fig. 6. Decay of the maximum velocity (ISPH vs. WCSPH) for $Re = 0.1$ and 10,000 particles. The line corresponds to the analytical solution.

similar convergence. By increasing the number of particles above 10,000 we are able to reduce the error to a value below 1%, but already a resolution of 1000 particles is sufficient to reproduce this flow accurately ($L_\infty < 6\%$).

Small Reynolds number flows are described very satisfactorily by the two methods. However, Chaniotis et al. [15] pointed out that the SPH accuracy for this flow usually deteriorates for increasing Reynolds numbers. Indeed, in this case, the inertia of the particles becomes relevant and, due to the slow vorticity decay, the particles have the time to change their spatial configuration considerably. This change in the particle's configuration produces notable fluctuations in the hydrodynamic variables, which are not present in the case of small Reynolds number flows. In the simulations at $Re = 100$ that we present below we observe a transition from ordered to disordered meshes which strongly affects the quality of the results. To avoid the transition, the initial condition for the particle locations at $Re = 100$ is the outcome of a previous run of the same Green–Taylor flow until full relaxation of the particle positions. The initial velocity field is then imposed at $t = 0$

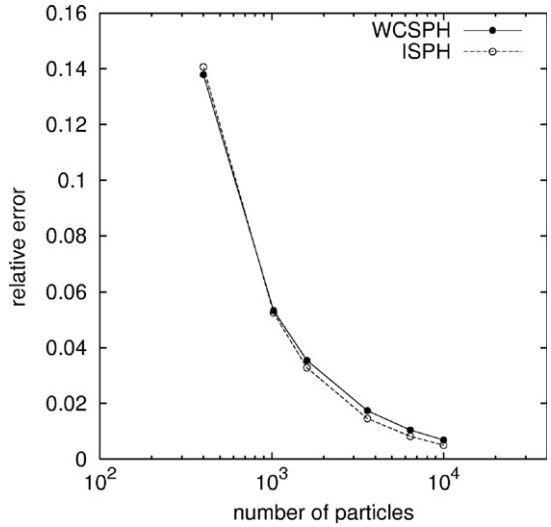

Fig. 7. L_∞ error for the Taylor–Green flow (ISPH vs. WCSPH) for $Re = 0.1$ and different particle resolutions.

by using Eq. (49) on the relaxed particle configuration. An alternative procedure to avoid these features would be to remesh according to the procedure described in [15].

In the next simulations at $Re = 100$ we consider the following notations for the particle resolutions: m2 (40×40), m3 (60×60), m4 (80×80), m5 (100×100). In Fig. 8 and Fig. 9, the velocity decays are plotted for both ISPH and WCSPH with different number of particles. Here, differently to the previous case, deviations between the SPH and theoretical values for the maximum velocity are visible. A number of particles $N = 1600$ (m2) is not sufficient to reproduce accurately the velocity decays for neither ISPH nor WCSPH. However, by increasing the resolution (m2 → m5), the corresponding SPH solution seems to converges to the exact one.

In order to estimate the error made in the ISPH calculations, we examine again the behaviour of L_∞. Fig. 10 shows the final maximum relative errors as a function of the number of particles. The tendency of L_∞ to converge to zero is clear. One thousand and six hundred particles (m2) are not sufficient to describe

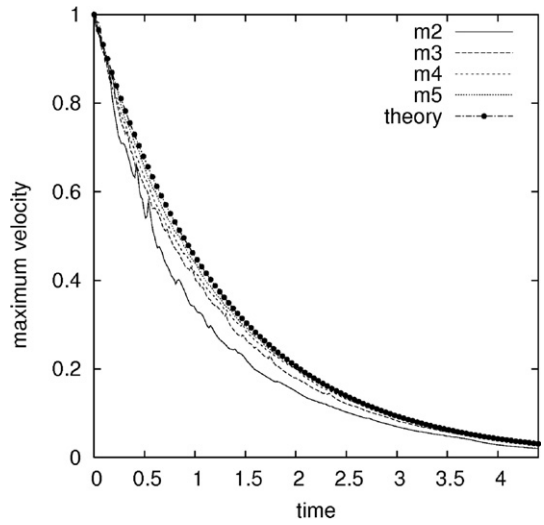

Fig. 8. Evolution of the maximum velocity for the ISPH and for different resolutions ($Re = 100$). The dotted line corresponds to the analytical solution.

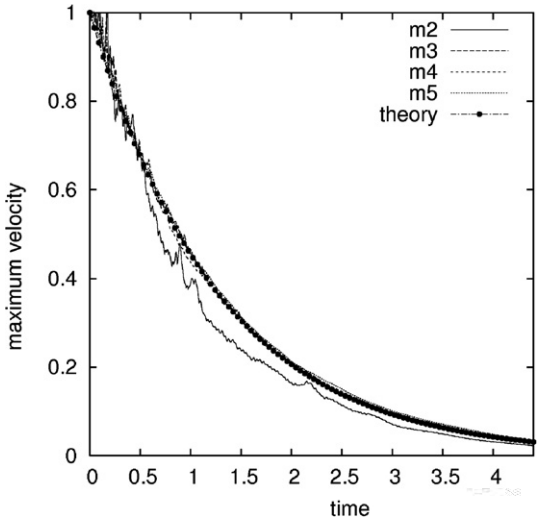

Fig. 9. Evolution of the maximum velocity for the WCSPH and for different resolutions ($Re = 100$). The dotted line corresponds to the analytical solution.

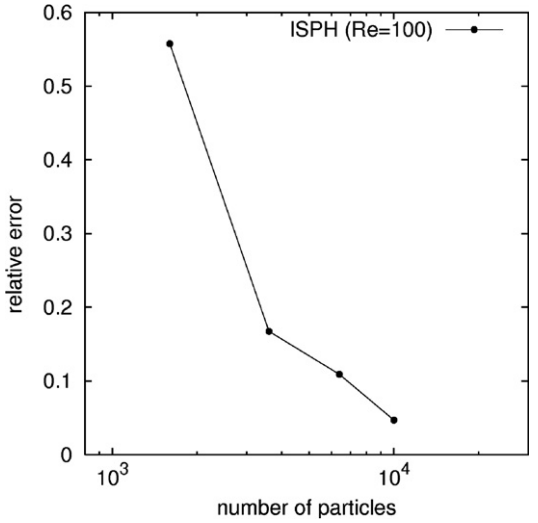

Fig. 10. Convergence of L_∞ for the ISPH ($Re = 100$). A resolution of 10,000 particles (m5) produce a maximum relative error smaller than 5%. In this case the order of convergence is approximately 2, $L_\infty \propto h^2$.

accurately the evolution of this flow (the relative error is of the order of 60%). However, by increasing the number of particles to 10,000 (m5) we can reduce it below 5%.

Cummins and Rudman [6] suggested that a check of the particle positions could be relevant to estimate the overall accuracy of the incompressibility condition. Accordingly, in Fig. 11 and Fig. 12 we plot two snapshots (time $t = 2$) of the particle configurations for the WCSPH and ISPH for $Re = 100$. In both cases the particles are homogeneously distributed over the solution domain. The only difference appears to be in the center of the main vortex where the particles (for ISPH) tend to join together forming lines. This effect has been also observed by Cummins and Rudman in the case of the projection SPH (PSPH) method and the authors suggested that it could be related to the unwanted rise of the CV coefficient. Contrary to PSPH, this specific particle re-arrangement does not have, in our simulations, any effective impact on CV which remains whatsoever constant during all the run. A possible explanation of this behaviour for PSPH could be given in the following

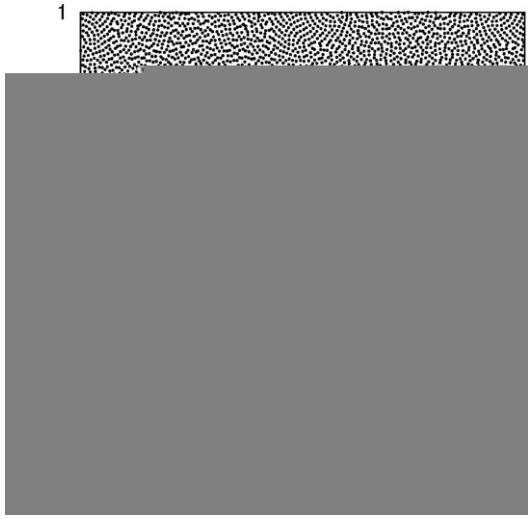

Fig. 11. Taylor–Green particle configuration (m5) for WCSPH at time $t = 2$ ($Re = 100$).

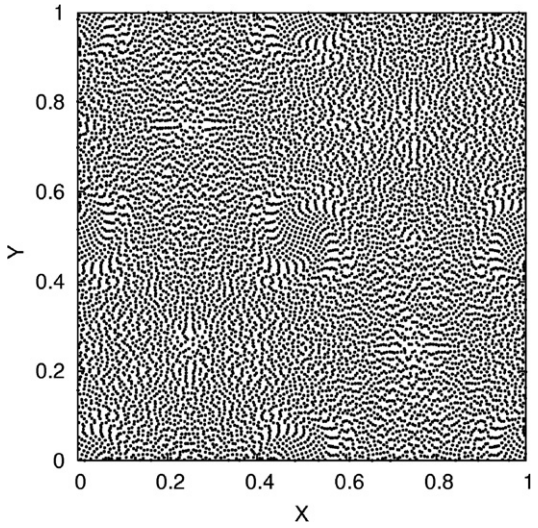

Fig. 12. Taylor–Green particle configuration (m5) for ISPH at time $t = 2$ ($Re = 100$).

terms: as suggested in [6], the new velocities $\mathbf{v}^{n+1}$, obtained by applying an SPH projector to the preliminary velocity field, are effectively divergence-free only within an unavoidable spatial truncation error. In addition, the precision in the determination of the divergence-free vector field is not necessarily the same as that obtained by the solution of the Poisson equation, but errors can arise from a difference between the SPH formulation for the Laplacian and for the gradient of the divergence [16]. Hence, errors in particle positions will be produced (and consequently in the density field) which will cause wrong projections at the next time step and therefore error accumulation. In the SHAKE method, we enforce incompressibility not through a *velocity-dependent condition* but, rather, through a *geometrical constraint* on the particle positions. By construction, in ISPH the accuracy in the density field is not affected by truncation errors but it is controlled uniquely by the tolerance parameter ϵ.

As a last check for spatial convergence, we consider the following L_2 norm for the velocity field:

$$\|\mathbf{v}(t)\|_2 = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(\mathbf{v}_i^{\mathrm{sph}}(t) - \mathbf{v}_i^{\mathrm{th}}(t)\right)^2}. \tag{51}$$

The index i runs over a regular lattice that has a number of nodes equal to the number N of particles. $\mathbf{v}_i^{\mathrm{sph}}(t)$ is the particle velocity interpolated at the lattice nodes and $\mathbf{v}_i^{\mathrm{th}}(t)$ is the analytical solution for the velocity field at the same node.

Table 1 shows the values of L_2 for the ISPH method. Convergence is obtained for increasing number of particles at every time $t = 0.9$, 1.8, 2.7, 3.6, 4.5. Table 2 shows the corresponding L_2-results for the WCSPH. Both methods converge for times smaller than $t = 3.6$ and there is an indication that the convergence of WCSPH is slightly faster than the convergence of ISPH. This is probably due to the more homogeneous distribution of the particles in WCSPH. However, it should also be noticed that for larger times WCSPH does not converge to the analytical incompressible solution for increasing N, (see the row corresponding to $t = 3.6$, 4.5 in Table 2). A detailed analysis shows that the deviations between theoretical and simulation results are due to anisotropies in the distribution of particles near the stagnation points of the flow. These anisotropies also appear in the ISPH simulation but do not seem to be so dangerous as to compromise the convergence of the method. Again, remeshing is a method to avoid these anisotropies. As a final remark, we should point out that, in our convergence analysis, the cutoff radius h has been reduced consistently in order to keep the average number of neighbouring particles constant, $h \propto \Delta x$. As suggested also in [6] higher order of convergence could be obtained by requiring that $h \propto \Delta x^q$ with $q < 1$.

5.3. Efficiency

In this section we evaluate the efficiency of the ISPH method compared with WCSPH. We extract the CPU times required to run the two respective algorithms in the Taylor–Green flow until $t = 2$ for $Re = 100$. The results are presented in Fig. 13, where the advantage of the ISPH is evident over all the range of resolutions simulated.

It should be noticed that the maximum advantage of the ISPH vs. WCSPH is obtained at lower and medium resolutions. Indeed, while WCSPH must comply always with the CFL condition (45), the time step constraint affecting the ISPH method is given by Eq. (46) and therefore a time step 10 times larger can actually be used. However, for increasing number of SPH particles, the viscous condition (44) becomes dominant over (45) and a smaller time step must be adopted. This restriction affects of course earlier the ISPH method, being its maximum allowable time step much larger than WCSPH. The maximum speed-up factor has been found to be 4 for $N = 1600$, and it remains nearly constant until $N = 10^4$ where the ISPH method starts to be influenced

Table 1
Spatial convergence results (ISPH) for the Taylor–Green flow

Time	m2	m3	m4	m5
0.9	6.986	4.707	3.414	2.090
1.8	6.045	2.640	1.646	0.942
2.7	3.051	1.432	0.789	0.331
3.6	1.316	0.757	0.327	0.185
4.5	0.563	0.373	0.149	0.086

In the columns, the L_2 velocity norms ($\times 10^2$) corresponding to different resolutions and for several times are shown ($Re = 100$).

Table 2
Spatial convergence results (WCSPH) for the Taylor–Green flow

Time	m2	m3	m4	m5
0.9	5.012	1.747	1.507	0.531
1.8	2.495	0.807	0.416	0.369
2.7	2.192	0.309	0.185	0.152
3.6	1.776	0.200	0.095	0.107
4.5	1.339	0.195	0.128	0.194

In the columns, the L_2 velocity norms ($\times 10^2$) corresponding to different resolutions and for several times are shown.

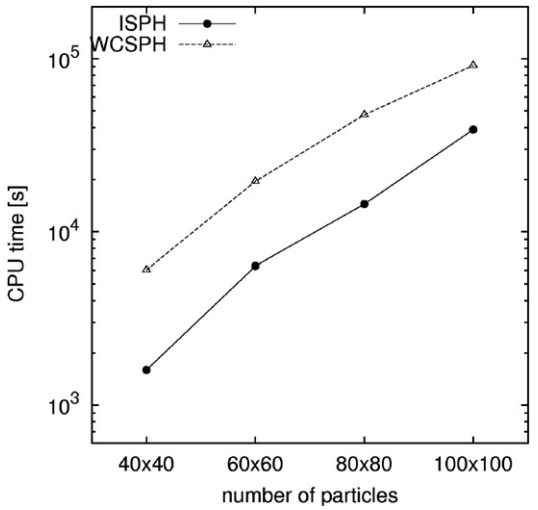

Fig. 13. CPU time as a function of the number of particles: WCSPH vs. ISPH for the Taylor–Green flow.

by the viscous condition (still a factor 3 is gained). The cross-point between the two curves is expected at higher resolutions where the viscous condition is dominant for both methods.

5.4. Deformation of an elliptical drop

Finally, the ability of ISPH to simulate free surfaces has been tested for the two-dimensional flow of an elliptical drop. As illustrated in Ref. [4], if an initial linear velocity field is imposed (and provided that the drop remains elliptical) a simple set of equations can be found for the time-dependent evolution of the axes of the ellipsoid (in the following denoted as a and b). In Ref. [4], Monaghan showed that from the momentum equation it is possible to derive the following equation for A

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \frac{A^2(a^4 - \omega^2)}{(a^4 + \omega^2)}, \tag{52}$$

where A is defined through

$$\frac{\mathrm{d}a}{\mathrm{d}t} = -aA \tag{53}$$

and $\omega = ab$ at time $t = 0$. The previous set of two coupled differential equations has been solved numerically and the results compared with those of ISPH and WCSPH. For the latter, two different density evaluations have been implemented: (A) standard way through the particle summation (3) (WCSPH) and (B) update of the density through the discrete continuity equation as in Ref. [4] (denoted in the following as CONT-SPH). Fig. 14 shows the evolution of the minor axis a for ISPH, WCSPH and CONT-SPH compared with the solution obtained by integrating numerically Eqs. (52) and (53). In all cases a linear velocity field $(V_x, V_y) = 2.3 \times (X - X_c, -Y + Y_c)$ was applied to an initially circular drop (modeled with 1024 particles), being (X_c, Y_c) the center of the drop. The Mach number was set equal to 0.1 for the WCSPH and CONT-SPH while in the ISPH run, the SHAKE routine was iterated at every time step until $\epsilon < 0.01$ as in the previous sections.

From Fig. 14 it results that ISPH and CONT-SPH work better than WCSPH where strong fluctuations during decay take place. This fact has been confirmed by the analysis of the corresponding particle configurations at time $t = 0.4$ (see Fig. 15) where it is clearly visible how the elliptical shape of the drop simulated by WCSPH has been completely lost due to artificial tension effects. Concerning ISPH and CONT-SPH, Fig. 14 suggests that the two methods work equally well: the ISPH solution is closer to the analytical one until $t = 0.4$ when the corresponding curves cross each other; CONT-ISPH, although underestimating more markedly the

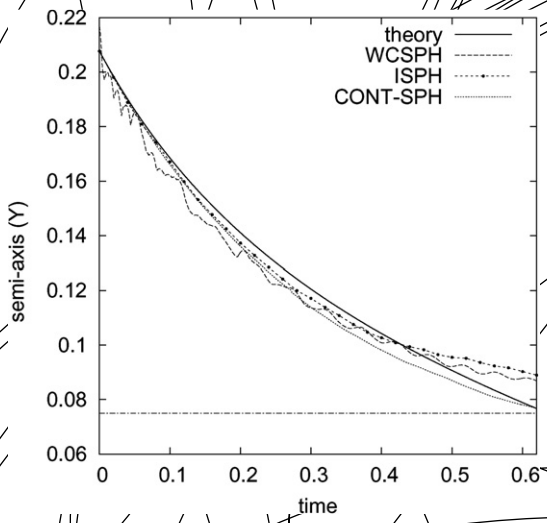

Fig. 14. Evolution of the minor axis a for ISPH, WCSPH (with density summation) and CONT-SPH (with continuity equation) and the solution evaluated numerically from Eqs. (52) and (53). The horizontal dotted line corresponds to the cutoff radius h of the simulation.

analytical solution for initial and intermediate times, remains closer to it for larger times (the crossover between numerical and theoretical curve is delayed to $t \approx 0.62$). All the drops eventually break up after time

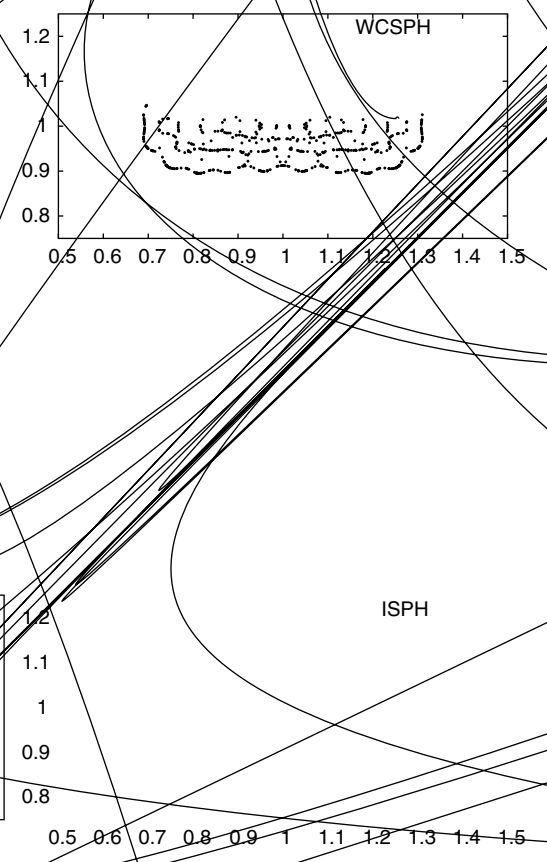

Fig. 15. Snapshots of the drop at $t = 0.4$ for: WCSPH with density summation (top), CONT-SPH obtained by integrating the continuity equation (middle) and ISPH (bottom). Notice the formation of artificial ring structures in the WCSPH drop. Contrary, the inner structure of the ISPH and CONT-SPH drops remains much more homogeneous. In the latter two cases the elliptical shape of the drop is preserved.

$t = 1$. One should notice that, as soon as the transversal dimension of the drop (minor axis a) becomes of the order of the cutoff radius $h = 0.075$ (corresponding to the constant dotted line in Fig. 14), the accuracy of the numerical results should become questionable because SPH is unable to describe physical phenomena below the discretization length h. In order to obtain a better solution, an increasing number of particles should be used in order to reduce h and remove spurious artificial effects.

In order to compare in detail the three methods, three snapshots of the particle configurations at the same instant $t = 0.4$ are shown in Fig. 15. For WCSPH, patterns of alternating rings and voids appear as result of the tensile instability, see Ref. [21]. The elliptical shape is lost. On the contrary, the simulations of ISPH and CONT-SPH do not show such artificial effects and, in addition, preserve the elliptical shape of the drops during all the run.

It is interesting to note that near the boundaries the last two methods present still clumping effects at $t = 0.4$. These results differ from the regular particle configurations shown in Ref. [4] and are probably due to the implementation of the XSPH variant used for the evolution of the particle positions which prevents unphysical particle penetrations and could inhibit instabilities close to the surfaces.

The time steps used in these simulations were, respectively, $\Delta t = 0.02$ for ISPH and $\Delta t = 0.0005$ for WCSPH and CONT-SPH (40 times smaller than ISPH). With a time step $\Delta t = 0.001$, the WCSPH drop blew up shortly after start up.

One of the advantages of taking the density as an independent variable (CONT-SPH) is that the density deficit (see Ref. [19]) near free boundaries is avoided. This produces more regular particle arrangements, free of the tensile instability as observed in our simulations (Fig. 15). The drawback of updating the density through the continuity equation, though, is the lack of exact mass conservation, which may be a problem in those situations where very long times are required. Our incompressible method, by construction, does not make use of the continuity equation and requires the density to be computed from Eq. (3). Therefore, in our method we do have the density deficit problem near boundaries. Actually, we impose that the density of each particle is constant during the simulation and consequently this implies that the particles near the surface will always have a smaller density than those in bulk. Of course, this may induce errors when two free surfaces contact each other. Our algorithm implies that those particles that are no longer at the surface, as a consequence of the merging of the two free surfaces, will still have the lower initial density they had at the beginning when they were at the free surface. We are presently working on a modification along the lines of Ref. [19] in order to correct the density deficit near free surfaces.

The flows presented so far do not require the consideration of boundary conditions on fixed or moving walls. Nevertheless, we have checked that the method works similarly well for planar solid walls. The implementation of boundary conditions on planar walls is done by following Refs. [17,18,5,20]. The main concern, specially in the context of the method presented in this paper, is the problem of the density deficit that occurs if one uses Eq. (3) for the fluid particles near the wall. According to Ref. [18], the density is given by Eq. (3) with a correction factor that depends on the distance of the fluid particle to the wall. The only crucial point where the SHAKE method enters into the formulation of boundary conditions is that one should keep this corrected density fixed with the SHAKE algorithm to its initial constant value. Poiseuille and Couette flows have been performed, with convergence tests similar to the ones presented for the periodic flows above.

6. Conclusions

We have implemented incompressibility in SPH by requiring that the volume of the fluid particles remains constant. As opposed to methods where the divergence of the velocity field is required to vanish, we do not need to solve a Poisson equation. Instead, we have to solve a set of non-linear equations for the constraints that enforce the constancy of the volumes. But this can be done quite efficiently through the SHAKE methodology. The main advantage of the present method over zeroing the divergence of the velocity is that we keep the function (i.e. density) constant rather than its time derivative equal to zero (i.e. $\nabla \cdot \mathbf{v} = 0$). Therefore, we do not observe the accumulation of errors in the density field as in Ref. [6].

We have tested our method in several benchmark problems, analyzing the convergence rate, the efficiency and the treatment of boundary conditions. The potential of the present new method is larger at high Reynolds number simulations. In such simulations, the sonic constraint given by Eq. (45) is the one limiting the time

step, rather than the viscous constraint in Eq. (44). In addition, in our method we do not need to store a $N \times N$ matrix at every time step (as in Poisson solver methods) but only a diagonal one (N elements). For large-scale simulations, as those commonly performed for the DNS of highly turbulent flows, this could also be a notable advantage.

A remark on further possible improvements of the methodology is now in order. As already mentioned, our SHAKE routine starts at any time step with an initial guess $\lambda^0 = 0$ and then iterations are performed until convergence is obtained. It would be interesting to explore the possibility of optimizing the choice for the initial guess by using the information from the previous step (i.e. by considering for every particle its correspondent old λ_i value or some average of it). Further acceleration of the SHAKE loop could be achieved by using also line-searches and backtracking procedures as those commonly used in Newton–Raphson methods to optimize the Newton step and, at the same time preserving its monotonic convergence [12]. Also the use of the higher order terms in Eq. (32) should be considered. It could turn out that this correction, although slowing down one single iterative step, could produce an overall benefit reducing the total number of iterations in the SHAKE loop.

Finally, the Green–Taylor flows analyzed show that particle disorder has a strong influence on the accuracy of results. It is possible to use remeshing techniques in order to have increased accuracy for the same number of particles [15]. The methodology presented here for imposing incompressibility can be directly used in a remeshed solution. Yet, it is an open problem how to formulate suitable remeshing techniques for free boundary problems as in case of a deforming elliptic drop.

Acknowledgments

M.E. is grateful for the financial support provided by the Australian Research Council. Part of the research was funded by the German Research Council (Deutsche Forschungsgemeinschaft). M.S. and P.E. acknowledge financial support from the Ministerio de Educación y Ciencia, FIS2004-01934. M.S. thanks "Programa Propio de la Investigación de la UNED" for financial support.

Appendix A

In this appendix we show that the set of Eq. (33) corresponds to the SHAKE algorithm as presented in Ref. [11]. The essence of the SHAKE method consists on writing the constraint Eq. (24) in the following form

$$\sigma(x'(\lambda)) = a + \mathbf{B}\lambda + c(\lambda) = 0, \tag{A.1}$$

where a is a vector independent of λ, $\mathbf{B}$ is a matrix independent of λ and $c(\lambda)$ is the remaining non-linear function of λ. We can decompose the matrix $\mathbf{B} = \mathbf{B}_\mathrm{D} + \mathbf{B}_\mathrm{O}$ where $\mathbf{B}_\mathrm{D}$ is the diagonal part of $\mathbf{B}$ and $\mathbf{B}_\mathrm{O}$ is the off-diagonal part. Therefore, Eq. (A.1) can be rewritten as

$$\mathbf{B}_\mathrm{D}\lambda = -(a + \mathbf{B}_\mathrm{O}\lambda + c(\lambda)), \tag{A.2}$$

which can be solved iteratively as

$$\lambda^{n+1} = -\mathbf{B}_\mathrm{D}^{-1}(a + \mathbf{B}_\mathrm{O}\lambda^n + c(\lambda^n)). \tag{A.3}$$

Of course, the inverse of a diagonal matrix is trivial to compute and makes the solution of (A.3) easy. The basic assumption in SHAKE is that it is possible to write the constraint in the form (A.1). Depending on the form of the constrain, it is not always easy to identify a linear term. This is precisely the case for the constraint (15). For this reason, we rewrite the SHAKE approach in the following way.

Let us write the constraint equation as

$$\sigma(x'(\lambda)) = \left[\sigma(x'(\lambda)) - \left.\frac{\partial\sigma(x'(\lambda))}{\partial\lambda}\right|_{\lambda=\lambda_0}\lambda\right] + \left.\frac{\partial\sigma(x'(\lambda))}{\partial\lambda}\right|_{\lambda=\lambda_0}\lambda = [a + c(\lambda)] + \mathbf{B}\lambda = 0, \tag{A.4}$$

where we have extracted the "linear" term in λ by defining the matrix $\mathbf{B}$ as

$$\mathbf{B} = \left.\frac{\partial \sigma(x'(\lambda))}{\partial \lambda}\right|_{\lambda=\lambda_0}. \tag{A.5}$$

Here, λ_0 is a reference value of the Lagrange multipliers. In many application in molecular dynamics with constrains, the matrix $\mathbf{B}$ is just a constant and the actual value of λ_0 is irrelevant. In the case that the constraint is such that $\mathbf{B}$ does actually depend on λ one can use for the reference value of λ_0 the one obtained in the previous time step.

By decomposing this matrix $\mathbf{B}$ in its diagonal and off-diagonal parts, Eq. (A.2) will have now the following form

$$\mathbf{B}_{\mathrm{D}}\lambda = -[\sigma(x'(\lambda)) - \mathbf{B}\lambda + \mathbf{B}_0\lambda], \tag{A.6}$$

which can be solved iteratively in the following very simple form

$$\lambda^{n+1} = \lambda^n - \mathbf{B}_{\mathrm{D}}^{-1}\sigma(x'(\lambda^n)). \tag{A.7}$$

It must be emphasized that we have done nothing else than rewriting the SHAKE algorithm in a particular simple form and that Eq. (A.7) is identical to Eq. (A.3). However, the form (A.7) is much simpler than Eq. (A.3) as it appears directly in terms of the constraint.

References

[1] L.B. Lucy, A numerical approach to testing the fission hypothesis, Astron. J. 82 (12) (1977) 1013–1924.
[2] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamic: theory and application to non-spherical stars, Monthly Notices Roy. Astron. Soc. 181 (1977) 375–389.
[3] R. Issa, E.S. Lee, D. Violeau, D.R. Laurence, Int. J. Numer. Methods Fluids 47 (2005) 1101–1106.
[4] J.J. Monaghan, Simulating free surface flows with SPH, J. Comput. Phys. 110 (2) (1994) 399.
[5] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, J. Comput. Phys. 136 (1997) 214–226.
[6] S.J. Cummins, M. Rudman, An SPH projection method, J. Comput. Phys. 152 (2) (1999) 584–607.
[7] J. Pozorski, A. Wawrenczuk, SPH computation of incompressible viscous flows, J. Theor. Appl. Mech. 40 (2002) 917.
[8] X.Y. Hu, N.A. Adams, Angular-momentum conservative smoothed particle dynamics for incompressible viscous flows, Phys. Fluids 18 (2006) 101702.
[9] P. Español, M. Revenga, Smoothed dissipative particle dynamics, Phys. Rev. E 67 (2003) 026705.
[10] H. Goldstein, Classical Mechanics, Addison-Wesley, Massachusetts, 1983.
[11] T. White, G. Cicotti, J.-P. Hansen, Brownian dynamics with constraints, Mol. Phys. 99 (2001) 2023.
[12] W. Press, S. Teukolski, W. Vetterling, B. Flannery, Numerical Recipes in Fortran 77, Cambridge University Press, 1992.
[13] H.A. Posch, W.G. Hoover, Simulation of two-dimensional Kolmogorov flow with smooth particle applied mechanics, Physica A 240 (1997) 286.
[14] G. Sivashinsky, Weak turbulence in periodic flows, Physica D 17 (1985) 243.
[15] A.K. Chaniotis, D. Poulikakos, P. Koumoutsakos, Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows, J. Comput. Phys. 182 (2002) 67.
[16] F. Colin, R. Egli, F.Y. Lin, Computing a null divergence velocity field using smoothed particle hydrodynamics, J. Comput. Phys. 217 (2006) 680–692.
[17] H. Takeda, S.M. Miyama, M. Sekiya, Numerical simulation of viscous flow by smoothed particle hydrodynamics, Prog. Theor. Phys. 92 (5) (1994) 939–960.
[18] S. Kulasegaram, J. Bonet, R. Lewis, M. Profit, A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications, Comput. Mech. 33 (2004) 316.
[19] P.W. Randles, L.D. Libersky, Smoothed particle hydrodynamics: some recent improvements and applications, Comput. Methods Appl. Mech. Eng. 139 (1996) 375–408.
[20] M. Ellero, R.I. Tanner, SPH simulations of transient viscoelastic flows at low Reynolds number, J. Non-Newt. Fluid Mech. 132 (2005) 61–72.
[21] J.J. Monaghan, SPH without tensile instability, J. Comput. Phys. 159 (2) (2000) 290.