INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING, *SERIES B*
Volume 3, Number 2, Pages 126–167

A NEW INCOMPRESSIBLE SMOOTHED PARTICLE HYDRODYNAMICS-IMMERSED BOUNDARY METHOD

RODRIGUE KÉOU NOUTCHEUWA AND ROBERT G. OWENS

Abstract. In this article we develop a new smoothed particle hydrodynamics (SPH) method suitable for solving the incompressible Navier-Stokes equations, even with singular forces. Singular source terms are handled in a manner similar to that in the immersed boundary (IB) method of Peskin (2002). The numerical scheme implements a second-order pressure-free projection method due to Kim and Moin (1985) and completely obviates the difficulties that may be faced in prescribing Neumann pressure boundary conditions. The proposed SPH method is first tested on the planar start-up Poiseuille problem and a detailed error analysis is performed. For this problem, the results are similar whether the SPH particles are free to move or fixed on a regular grid. Our hybrid SPH-IB method is then used to calculate the dynamics of a stretched immersed elastic membrane and the advantages in this case of fixing the SPH particles, rather than allowing them to move with the fluid, are discussed.

Key words. Smoothed particle hydrodynamics, incompressible flow, projection method, singular force, immersed boundary

1. Introduction

The SPH method permits the numerical solution of the equations of fluid dynamics or solid mechanics by representing the fluid/solid as an ensemble of fluid/solid particles. The method is mesh-free and based on a Lagrangian formulation. The method was first described in 1977 by Gingold and Monaghan [19] and independently in the same year by Lucy [37]. The motivation behind the creation of this method was the need to solve complex problems that traditional mesh-based Eulerian methods (such as finite difference methods, finite element methods or finite volume methods) were not able to handle easily. Problems may be encountered when these mesh-based methods are used for the solution of problems involving free surfaces, deformable boundaries and problems involving pronounced deformations. Moreover, when the computational domain is complex it is not always clear how to construct meshes for mesh-based methods that take into account all the irregularities of the domain, even if techniques such as domain decomposition are used. Originally, the SPH method was introduced with the aim of simulating astrophysical phenomena that introduce important variations in the density in complex and non-symmetric geometries [40]. Since its first appearance in the literature the method has been further developed and applied to the solution of many different problems, among which we could cite multi-phase flows [10, 24, 43], free-surface flows [2, 55], impacts and explosions [35, 44], heat conduction modelling [8], viscoelastic flows [16, 17, 47] and solid mechanics [14, 38, 65]. Many other application areas are described in the recent extensive reviews of Cleary et al. [9] and Monaghan [42]. However, there are a great many examples in the scientific literature of SPH methods being tested on problems where, traditionally, mesh-based methods have been successful. For example, SPH methods have been used to solve for flow around right circular cylinders and other bluff bodies [15, 23, 29, 57], steady and

Received by the editors November 14, 2011 and, in revised form, January 29, 2012.
2000 *Mathematics Subject Classification.* 74F10, 76M28.

start-up Poiseuille flow [4, 23, 47, 57] and lid-driven cavity flow [6, 29, 64]. The present paper and a recent article using the SPH method for the simulation of polymeric fluid flows [47] are the only examples in the literature that we know of that use the SPH method to solve the incompressible Navier-Stokes equations with a distribution of singular forces.

When simulating incompressible fluid flows a significant challenge consists of ensuring that the incompressibility condition is satisfied. In order to accomplish this the traditional SPH method, often termed the Weakly Compressible Smoothed Particle Hydrodynamics (WCSPH) method (see, for example, [1, 13, 19, 39, 40, 42, 44, 64]), supposes that the fluid is weakly compressible and the density of the fluid (more particularly, of each fluid particle) is calculated at each time step. The most common technique adopted for approximating the incompressibility of the fluid is to use an equation of state that relates the pressure and the density. Although this equation of state may have several different forms (see the articles cited above) the key element is that the magnitude of the speed of sound used in the equation must be sufficiently small to be practical whilst at the same time being sufficiently large in order to maintain the density approximately constant from one time step to another [4]. The use of a large magnitude speed of sound implies a very severe CFL condition on the time step in the calculations [13, 17, 23, 29, 45, 64]. This approximate enforcement of the incompressibility condition introduces errors that have as their source small fluctuations in the density [13, 55] and may give rise to large pressure oscillations [29, 64].

An alternative approach leading to a truly incompressible SPH method involves the use of a projection (operator splitting) scheme, originally developed in the context of Eulerian mesh-based methods, in order to impose the incompressibility condition in the calculations. The original (first-order) projection scheme was introduced by Chorin [7] in 1968 and has been extensively used since then. The projection method was used for the first time in the context of SPH methods by Cummins and Rudman [13] who gave their scheme the name Projection Smoothed Particle Hydrodynamics (PSPH): an acronym that we will incorporate into the names of our own schemes. In the SPH calculations of Lee et al. [29] involving flow around a square cylinder, lid-driven cavity flow and a dam-breaking problem, their truly incompressible SPH method was found in all cases to yield smoother velocity and pressure profiles than was possible with the WCSPH method. Moreover, the CPU time was anywhere between 2 and 20 times less with the incompressible SPH method than was possible with the WCSPH approach. An important problem with the Neumann pressure boundary condition used in many pressure-correction projection schemes and present, for example, in the work of Cummins and Rudman [13] and Lee et al. [29], has been highlighted by Brown et al. [5], Guermond et al. [21], Hosseini and Feng [23] and others. Typically, zero Neumann boundary conditions are imposed for the Poisson equation satisfied by the pressure. However, in open-boundary problems or when flow is around bluff bodies or through a channel of variable cross section this is not correct and leads to numerical boundary layers. Hosseini and Feng [23] avoided this problem by using the rotational projection scheme of Timmermans et al. [60] and imposing a non-homogeneous Neumann condition on the pressure which was shown to be consistent with the linear momentum equation.

The problems plaguing the accuracy and stability of SPH computations due to particle clustering (anisotropic particle spacing) have been well documented in the scientific literature (see, for example, [18, 25, 64]). Monaghan [41] demonstrated

that the well known tensile instability present in SPH methods will cause particle clustering and showed how this instability could be removed by using an artificial stress. Hu and Adams [25, 26] handled the problem of particle clustering in their SPH projection method by correcting particle positions with a non-linear iterative method. Chaniotis et al. [6] remedied the problem of particle disorder by periodically reinitializing the SPH particles back onto a uniform grid and interpolating the old particle properties onto the new particle positions. In [64], Xu et al. developed an idea of shifting particle positions, originally proposed in the context of the finite volume particle method by Nestor et al. [46]. The authors of [64] shifted particles slightly away from the streamlines and used Taylor series to correct the hydrodynamic variables. Pressure fields were found to be free from spurious oscillations at all Reynolds numbers. Similar results could be obtained with the remeshing strategy of Chaniotis et al. [6].

The motivation for the present work is to propose a modification to the PSPH scheme of Cummins and Rudman [13] suitable for the solution of the incompressible Navier-Stokes equations with singular forces. Rather than perform an explicit calculation of an intermediate velocity in the first step of a projection scheme we will use an implicit velocity scheme which is more stable. Whereas Cummins and Rudman and many other authors solve a Poisson problem for the pressure, the "pressure-free" scheme that we use in the present paper involves the solution only of a scalar function which allows the velocity (and, if required, the pressure) to be determined at the correction step. This approach is based on the projection scheme proposed by Kim and Moin [27] and which Brown et al. [5] called the PMIII projection method. The latter authors showed that the scheme is second order in time. We thereby avoid completely the problem of imposing an unphysical homogeneous Neumann boundary condition on the pressure. A further difference between our SPH projection scheme and those of authors proposing corrections to particle positions in order to avoid clustering [6, 23, 25, 26, 64] is that we introduce the notion of fixed particles. Because the fluid particles may be simply understood to be quadrature points which allow fluid variables to be determined at a given point in space and time, we may treat the governing equations in the Eulerian framework with particles that do not change their position in time. This, at least for a certain class of problems, including those with singular forces, presents a number of advantages over moving particle SPH methods, as we will see later. Certainly, for the problems that we will treat in this paper there seems to be no advantage in regularly remeshing (as in [6], for example). In this paper, when we use a projection SPH method with moving SPH particles we shall refer to a Lagrangian Projection SPH (LPSPH) method. Otherwise, when the quadrature points are kept fixed, we will speak of an Eulerian Projection SPH (EPSPH) method.

The present article is organised as follows. In Section 2 we describe the equations of fluid dynamics and the projection scheme that we will employ. In Section 3 we describe in some detail the SPH discretisation of the equations of the PMIII projection scheme. Section 4 describes the two test problems to which we wish to apply our SPH scheme. The first is start-up planar Poiseuille flow. The second is the problem of the dynamics of a stretched elastic membrane immersed in a viscous fluid. Fluid-structure interaction is handled in a manner analogous to that of the immersed boundary method of Peskin [49]. The results of our computations for these two test problems are presented in Section 5. Results for the start-up Poiseuille problem are compared with the exact solution and the accuracy and convergence of our method compared with that of Bierbrauer et al. [4]. Pressure

and velocity solutions for a stationary immersed membrane may be validated with the known exact solutions for this problem. The results of the problem of an oscillating Hookean membrane are compared with those of Tan et al. [58], who used a second-order accurate immersed interface method to investigate the oscillations of an immersed membrane.

2. Governing equations and projection scheme

2.1. Fluid equations. The Navier-Stokes equations for a viscous incompressible fluid in a $d-$dimensional domain $\Omega \subseteq \mathbb{R}^d$ are

$$\rho \frac{D\boldsymbol{u}}{Dt} = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \boldsymbol{F}, \quad \text{in } \Omega, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \quad \text{in } \Omega, \tag{2}$$

and we consider Dirichlet boundary and initial conditions of the form

$$\boldsymbol{u} = \boldsymbol{u}_b, \quad \text{on } \partial\Omega \text{ (the boundary of } \Omega), \tag{3}$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0. \tag{4}$$

In the above equations $\boldsymbol{u}$ denotes the fluid velocity and p is the pressure. The material derivative D/Dt is defined by

$$\frac{D}{Dt} := \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla).$$

Throughout this paper the fluid density ρ and viscosity μ will be taken as constants everywhere in Ω although allowing these to vary between one material region of the fluid and another is entirely straightforward in the usual SPH framework. The force $\boldsymbol{F}$ in (1) accounts for all external forces that act on the fluid, such as gravity or a singular force distribution due to an immersed boundary, for example. $\boldsymbol{u}_b$ in (3) denotes the boundary value of the fluid velocity and $\boldsymbol{u}_0$ is our notation for the initial value of $\boldsymbol{u}$.

2.2. A second-order projection scheme. In order to overcome various problems relating to the small amount of compressibility of the fluid in WCSPH schemes for incompressible fluids we have chosen to use a suitable projection scheme, many forms of which may be found in the literature (see, for example, [5, 13, 21, 36, 55]). In the present paper we use a variant of the pressure-free projection method proposed by Kim and Moin [27], which Brown et al. [5] refer to as the PMIII projection method. Denoting by $\boldsymbol{F}^n$ the sum of all external forces acting on the fluid at time $t_n = n\Delta t$ the Eulerian PMIII projection scheme for Eqns. (1)-(4) may be written out as follows:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} + ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u})^{n+1/2} = \frac{\nu}{2}\nabla^2(\boldsymbol{u}^n + \boldsymbol{u}^*) + \frac{\boldsymbol{F}^n}{\rho}, \quad \text{in } \Omega, \tag{5}$$

$$\boldsymbol{n} \cdot \boldsymbol{u}^* = \boldsymbol{n} \cdot \boldsymbol{u}_b, \quad \text{on } \partial\Omega, \tag{6}$$

$$\boldsymbol{\tau} \cdot \boldsymbol{u}^* = \boldsymbol{\tau} \cdot \left(\boldsymbol{u}_b + \frac{\Delta t}{\rho} \nabla \varphi^{n+1} \right), \quad \text{on } \partial\Omega, \tag{7}$$

$$\Delta t \nabla \cdot \left(\frac{1}{\rho} \nabla \varphi \right)^{n+1} = \nabla \cdot \boldsymbol{u}^*, \quad \text{in } \Omega, \tag{8}$$

$$\boldsymbol{n} \cdot \boldsymbol{\nabla} \varphi^{n+1} = 0, \quad \text{on } \partial\Omega, \tag{9}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{\Delta t}{\rho} \nabla \varphi^{n+1}, \tag{10}$$

where $\boldsymbol{u}^*$, $\boldsymbol{u}^n$ and $\boldsymbol{u}^{n+1}$ denote, respectively, the intermediate velocity field, and the evaluations of $\boldsymbol{u}$ at times $t_n = n\Delta t$ and $t_{n+1} = (n+1)\Delta t$. The convective derivative appearing on the left-hand side of Eqn. (5) is a second-order (Adams-Bashforth) approximation to this derivative at time step $(n + 1/2)\Delta t$:

$$((\boldsymbol{u} \cdot \nabla)\boldsymbol{u})^{n+1/2} := \frac{3}{2}((\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n) - \frac{1}{2}((\boldsymbol{u}^{n-1} \cdot \nabla)\boldsymbol{u}^{n-1}),$$

involving evaluations at the nth and $(n-1)$th time steps. Although it is known that the second-order Adams-Bashforth method is weakly unstable for pure convection problems this is not expected to be problematic in the present case, where our scheme is for a viscous fluid simulation (see, for example, page 31-5 of [56]). We denote the kinematic viscosity by $\nu = \mu/\rho$. $\boldsymbol{n}$ and $\boldsymbol{\tau}$ signify, respectively, unit normal and tangent vectors on the boundary $\partial\Omega$ of Ω.

The proposed projection scheme described by Eqns. (5)-(10) is a form of temporal discretisation of the Navier-Stokes equations (1)-(2) which allows the pressure and velocity calculations to be decoupled, at the cost of incurring a splitting error. The algorithm employed may be summarised as follows: at the start of each time step an intermediate velocity field $\boldsymbol{u}^*$ is calculated using the equation of conservation of linear momentum but ignoring the incompressibility constraint. Afterwards, once a Poisson equation (8) has been solved for a pressure-like scalar field φ, the intermediate velocity is projected onto the space of solenoidal vector fields in order to find $\boldsymbol{u}^{n+1}$ (Eqn. (10)). We note that φ^{n+1} not yet being available for the prescription (7) of the tangential component of the intermediate velocity boundary condition, an approximation $\widetilde{\varphi}^{n+1}$ is used in its place. According to Kim and Moin [27] it is sufficient for the conservation of second-order accuracy to choose $\widetilde{\varphi}^{n+1} = \varphi^n$. Brown et al. [5] showed that the extrapolation $\widetilde{\varphi}^{n+1} = 2\varphi^n - \varphi^{n-1}$ also leads to second-order accuracy but smaller errors. However, this extrapolation has not been adopted in the present paper.

The pressure at time $(n + 1/2)\Delta t$ need never be calculated explicitly but may be found, if required, from

$$\nabla p^{n+1/2} = \nabla\varphi^{n+1} - \frac{\mu\Delta t}{2}\nabla^2(\frac{1}{\rho}\nabla\varphi^{n+1}). \tag{11}$$

We note that by taking the dot product of each term of Eqn. (11) with a unit normal vector $\boldsymbol{n}$ on the boundary we may derive an expression for $\partial p^{n+1/2}/\partial n$. Using (9)-(10) and then (5) followed by (6) we get

$$\begin{aligned}
\nabla p^{n+1/2} \cdot \boldsymbol{n} &= \nabla\varphi^{n+1} \cdot \boldsymbol{n} - \frac{\mu\Delta t}{2}\nabla^2\left(\frac{1}{\rho}\nabla\varphi^{n+1}\right) \cdot \boldsymbol{n}, \\
&= -\frac{\mu}{2}\nabla^2(\boldsymbol{u}^* - \boldsymbol{u}^{n+1}) \cdot \boldsymbol{n}, \\
&= \left[\frac{\mu}{2}\nabla^2(\boldsymbol{u}^n + \boldsymbol{u}^{n+1}) - \rho\left(\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t}\right) - \rho((\boldsymbol{u} \cdot \nabla)\boldsymbol{u})^{n+1/2} + \boldsymbol{F}^n\right] \cdot \boldsymbol{n}, \\
&= \left[\frac{\mu}{2}\nabla^2(\boldsymbol{u}^n + \boldsymbol{u}^{n+1}) - \rho\left(\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t}\right) - \rho((\boldsymbol{u} \cdot \nabla)\boldsymbol{u})^{n+1/2} + \boldsymbol{F}^n\right] \cdot \boldsymbol{n}, \qquad (12)
\end{aligned}$$

and this is consistent with the time-discretised linear momentum equation. We therefore avoid the numerical boundary layers present in some other projection schemes (see the discussions in [5, 21, 23], for example).

3. Smoothed Particle Hydrodynamics

In the SPH description of a fluid the latter is represented as an ensemble of particles which each possess material properties (density, viscosity) and carry with them their own pressure and velocity. In its usual form the SPH method is a Lagrangian description of the fluid and the particles are transported with the local fluid velocity. The SPH method is based on the notion of integral interpolation. This interpolation begins with the identity

$$f(\boldsymbol{x}) = \int_\Omega f(\boldsymbol{x}')\delta(\boldsymbol{x}-\boldsymbol{x}')d\boldsymbol{x}', \tag{13}$$

for all domains Ω where f is defined and which contain a point having position vector $\boldsymbol{x}'$. Identity (13) cannot be used as it is in the subsequent numerical calculations and is replaced with an approximation to f

$$\langle f(\boldsymbol{x})\rangle := \int_\Omega f(\boldsymbol{x}')W(\boldsymbol{x}-\boldsymbol{x}',h)d\boldsymbol{x}', \tag{14}$$

where the delta function appearing on the right-hand side of (13) has been replaced with a so-called kernel or smoothing function $W(\boldsymbol{x}-\boldsymbol{x}',h)$. In this context, the parameter h is called the smoothing length. The smoothing function is required to satisfy the following properties:

$$W(\boldsymbol{x}-\boldsymbol{x}',h) = 0 \text{ for } |\boldsymbol{x}-\boldsymbol{x}'| \geq kh, \tag{15}$$

$$\int_\Omega W(\boldsymbol{x}-\boldsymbol{x}',h)d\boldsymbol{x}' = 1, \tag{16}$$

$$\lim_{h\to 0} W(\boldsymbol{x}-\boldsymbol{x}',h) = \delta(\boldsymbol{x}-\boldsymbol{x}'), \tag{17}$$

$$W(\boldsymbol{x}-\boldsymbol{x}',h) \geq 0. \tag{18}$$

The kernel function is therefore a function of compact support and any function that satisfies the conditions (15)-(18) may be used as a smoothing function. Several different smoothing functions have been proposed in the literature and in our present work we use a cubic spline function [4, 17, 42]:

$$W(r,h) = w_0 \begin{cases} \frac{2}{3} - q^2 + \frac{1}{2}q^3, & 0 \leq q < 1; \\ \frac{1}{6}(2-q)^3, & 1 \leq q < 2; \\ 0, & q \geq 2, \end{cases} \tag{19}$$

where, for example, $w_0 = \frac{15}{7\pi h^2}$ for two-dimensional problems, $q := \frac{r}{h}$, $r := \|\boldsymbol{x}-\boldsymbol{x}'\|_2$ and k in (15) is equal to 2.

The error in approximating f by the interpolation integral (14) may be calculated using a Taylor series of $f(\boldsymbol{x}')$ about the point $\boldsymbol{x}$. If f is differentiable we get

$$\begin{aligned} \langle f(\boldsymbol{x})\rangle &= \int_\Omega \left[f(\boldsymbol{x}) + \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{x}-\boldsymbol{x}') + O((\boldsymbol{x}-\boldsymbol{x}')^2) \right] W(\boldsymbol{x}-\boldsymbol{x}',h)d\boldsymbol{x}', \\ &= f(\boldsymbol{x})\int_\Omega W(\boldsymbol{x}-\boldsymbol{x}',h)d\boldsymbol{x}' + \frac{\partial f}{\partial \boldsymbol{x}}\int_\Omega (\boldsymbol{x}-\boldsymbol{x}')W(\boldsymbol{x}-\boldsymbol{x}',h)d\boldsymbol{x}' + O(h^2). \end{aligned} \tag{20}$$

Using the property (16) and observing that $(\boldsymbol{x}-\boldsymbol{x}')W(\boldsymbol{x}-\boldsymbol{x}',h)$ is an odd function of $\boldsymbol{x}-\boldsymbol{x}'$ so that

$$\int_\Omega (\boldsymbol{x}-\boldsymbol{x}')W(\boldsymbol{x}-\boldsymbol{x}',h)d\boldsymbol{x}' = 0, \tag{21}$$

we arrive at the conclusion that

$$\langle f(\boldsymbol{x})\rangle = f(\boldsymbol{x}) + O(h^2). \tag{22}$$

That is to say that if the integral in (14) is evaluated exactly, $\langle f \rangle$ is a second-order in h approximation of f.

Let us denote by $\mathcal{B}(\boldsymbol{x}, h)$ a ball of radius $2h$ centred at $\boldsymbol{x}$ and therefore constituting the support of the kernel function $W(\boldsymbol{x} - \boldsymbol{x}', h)$. In order to apply the quadrature rule typically used in SPH methods for the evaluation of the integral on the right-hand side of (14) when $\boldsymbol{x} = \boldsymbol{x}_i$, the position vector of the ith fluid particle, we must first assign to each fluid particle in $\mathcal{B}(\boldsymbol{x}, h)$ an index j, a mass m_j and a density ρ_j (equal to the fluid density) where $j = 1, \ldots, N$ and N denotes the number of particles in $\mathcal{B}(\boldsymbol{x}, h)$. Then, denoting by f_j the SPH approximation of f at $\boldsymbol{x}_j$ for $j = 1, \ldots, N$ we have the numerical approximation

$$f_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j W_{ij}, \tag{23}$$

to $f(\boldsymbol{x}_i)$ where we define

$$W_{ij} = W(\boldsymbol{x}_i - \boldsymbol{x}_j, h).$$

The SPH approximation to the gradient of f may be found by replacing f by $\partial f / \partial \boldsymbol{x}$ in Eqn. (14) and using integration by parts. Supposing that $\mathcal{B}(\boldsymbol{x}, h) \subseteq \Omega$, the vanishing of W on the boundary $\partial\mathcal{B}(\boldsymbol{x}, h)$ (see, for example, [17, 34]) then leads to

$$\left(\frac{\partial f}{\partial \boldsymbol{x}}\right)_i \approx -\sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \frac{\partial W_{ij}}{\partial \boldsymbol{x}_j}, \tag{24}$$

where, for $i \neq j$,

$$\frac{\partial W_{ij}}{\partial \boldsymbol{x}_i} = \left(\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{r}\right) \frac{\partial W_{ij}}{\partial r}, \tag{25}$$

and $r := \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$ is the distance between the ith and jth particles. In the present paper, an alternative approach to the evaluation of the gradient of a function f is adopted (see, for example, [11, 13, 34, 40]): starting with the identity

$$\frac{\partial f}{\partial \boldsymbol{x}} = \rho \left[\frac{\partial}{\partial \boldsymbol{x}} \left(\frac{f(\boldsymbol{x})}{\rho} \right) + \frac{f(\boldsymbol{x})}{\rho^2} \frac{\partial \rho}{\partial \boldsymbol{x}} \right], \tag{26}$$

and using the formula (24) to discretise the gradients appearing in (26), we get

$$\left(\frac{\partial f}{\partial \boldsymbol{x}}\right)_i \approx \rho_i \sum_{j=1}^{N} m_j \left(\frac{f(\boldsymbol{x}_i)}{\rho_i^2} + \frac{f(\boldsymbol{x}_j)}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \boldsymbol{x}_i}, \tag{27}$$

where, in the above, we have used the result

$$\frac{\partial W_{ij}}{\partial \boldsymbol{x}_i} = -\frac{\partial W_{ij}}{\partial \boldsymbol{x}_j}.$$

Note that for points having position vectors $\boldsymbol{x}_i$ within a distance $2h$ of a boundary special care in the above procedures must be taken since part of the boundary of $\mathcal{B}(\boldsymbol{x}, h)$ then lies outside the flow domain Ω. This is remedied by the introduction of ghost particles in a region of depth $2h$ outside the boundary $\partial\Omega$, as will be described later in Section 3.3. In order to discretise the Laplacian of a vector valued function we have used Eqn. (21) of Shao and Lo [55]. In the presentation of the LPSPH and EPSPH methods that follows it has been assumed for simplicity that the fluid density is constant, although this is not necessary in order that the fluid be incompressible. In the event of a non-zero density gradient the SPH discretisation

that we propose below for the Poisson equation (8) could be replaced with Eqn. (16) of Shao and Lo [55], for example.

3.1. The LPSPH method. In the Lagrangian formulation of the SPH method the discretised equation of linear momentum (5) $\boldsymbol{u}^*$ assumes the form

$$\frac{\boldsymbol{u}_i^* - \boldsymbol{u}_i^n}{\Delta t} = \sum_{j\neq i} \frac{2m_j(\mu_i+\mu_j)}{(\rho_i+\rho_j)^2}\left[(\boldsymbol{u}_i^* - \boldsymbol{u}_j^*) + (\boldsymbol{u}_i^n - \boldsymbol{u}_j^n)\right]\frac{1}{r}\frac{\partial W_{ij}}{\partial r} + \frac{\boldsymbol{F}_i^n}{\rho_i}, \tag{28}$$

where μ_i is the viscosity of fluid particle i. The calculation of the intermediate velocity field is followed by the solution of the discretised Poisson equation (8) for the pressure-like variable φ at particle i:

$$\left(\nabla^2\varphi^{n+1}\right)_i = \frac{1}{\Delta t}\left(\rho\nabla\cdot\boldsymbol{u}^*\right)_i. \tag{29}$$

The discrete form of the divergence of $\boldsymbol{u}^*$ at the ith fluid particle position is found in an entirely analogous way to that which led to (27):

$$\begin{aligned}\left(\nabla\cdot\boldsymbol{u}^*\right)_i &= \rho_i\sum_j m_j\left(\frac{\boldsymbol{u}_j^*}{\rho_j^2}+\frac{\boldsymbol{u}_i^*}{\rho_i^2}\right)\cdot\frac{\partial W_{ij}}{\partial \boldsymbol{x}_i},\\ &= \rho_i\sum_j m_j\left(\frac{\boldsymbol{u}_j^*}{\rho_j^2}+\frac{\boldsymbol{u}_i^*}{\rho_i^2}\right)\cdot(\boldsymbol{x}_i-\boldsymbol{x}_j)\frac{1}{r}\frac{\partial W_{ij}}{\partial r}.\end{aligned} \tag{30}$$

The discretised Laplacian of φ at $\boldsymbol{x}_i$ now becomes φ^{n+1}

$$\left(\nabla^2\varphi^{n+1}\right)_i = \sum_j \frac{4m_j(\varphi_i^{n+1}-\varphi_j^{n+1})}{(\rho_i+\rho_j)}\frac{1}{r}\frac{\partial W_{ij}}{\partial r}. \tag{31}$$

Combining the above we have the discrete Poisson equation for φ^{n+1} in the form

$$\sum_{\substack{j\neq i\\ j\in\Omega}} b_{ij}\left(\varphi_i^{n+1}-\varphi_j^{n+1}\right) = \frac{\rho_i^2}{\Delta t}\sum_{j\neq i} m_j\left(\frac{\boldsymbol{u}_j^*}{\rho_j^2}+\frac{\boldsymbol{u}_i^*}{\rho_i^2}\right)\cdot(\boldsymbol{x}_i-\boldsymbol{x}_j)\frac{1}{r}\frac{\partial W_{ij}}{\partial r}, \tag{32}$$

where

$$b_{ij} := \frac{4m_j}{(\rho_i+\rho_j)}\frac{1}{r}\frac{\partial W_{ij}}{\partial r}. \tag{33}$$

After calculating the intermediate velocity and the scalar function φ^{n+1} we find the velocity of each fluid particle at the time step $(n+1)\Delta t$ from Eqn. (10) whose discrete form is the following:

$$\boldsymbol{u}_i^{n+1} = \boldsymbol{u}_i^* - \Delta t\sum_j m_j\left(\frac{\varphi_j^{n+1}}{\rho_j^2}+\frac{\varphi_i^{n+1}}{\rho_i^2}\right)(\boldsymbol{x}_i-\boldsymbol{x}_j)\frac{1}{r}\frac{\partial W_{ij}}{\partial r}. \tag{34}$$

Although in the PMIII projection method it is not necessary to calculate the pressure it may be computed, if so desired, and if the fluid density is constant, from Eqn. (11) at the $(n+1/2)$th step in the form

$$p_i^{n+1/2} = \varphi_i^{n+1} - \frac{\Delta t}{2}\sum_j \frac{4m_j(\mu_i+\mu_j)}{(\rho_i+\rho_j)^2}\left(\varphi_i^{n+1}-\varphi_j^{n+1}\right)\frac{1}{r}\frac{\partial W_{ij}}{\partial r} + c, \tag{35}$$

or, equivalently, as

$$p_i^{n+1/2} = \varphi_i^{n+1} - \frac{\mu_i\rho_i}{2}\sum_j m_j\left(\frac{\boldsymbol{u}_j^*}{\rho_j^2}+\frac{\boldsymbol{u}_i^*}{\rho_i^2}\right)\cdot(\boldsymbol{x}_i-\boldsymbol{x}_j)\frac{1}{r}\frac{\partial W_{ij}}{\partial r} + c, \tag{36}$$

where c is an arbitrary constant.

Eqns. (28), (32) and (34) - (36) are the Lagrangian SPH (LPSPH) form of the discrete governing equations. The SPH particles move with the same velocity as the fluid. The position $\boldsymbol{x}_i^{n+1}$ of each particle i at the end of the $(n+1)$th step is computed from

$$\boldsymbol{x}_i^{n+1} = \boldsymbol{x}_i^n + \frac{1}{2}(\boldsymbol{u}_i^{n+1} + \boldsymbol{u}_i^n)\Delta t, \tag{37}$$

where $\boldsymbol{x}_i^n$ is the position vector of the same particle at the end of the previous (nth) time step.

3.2. The EPSPH method. The standard SPH method is a Lagrangian mesh-free method in which the equations to be solved are discretised using particles that are free to move with the surrounding medium. We are now interested in adopting an Eulerian approach and make the particles assume the role of simple fixed quadrature points. With this in view the time-discretised equation of motion for the intermediate velocity is (see (5))

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} + \left((\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u}\right)^{n+1/2} = \frac{\nu}{2}\nabla^2(\boldsymbol{u}^n + \boldsymbol{u}^*) + \frac{\boldsymbol{F}^n}{\rho}, \tag{38}$$

where the convective derivative term on the left-hand side of Eqn. (38) is evaluated using a second-order Adams-Bashforth scheme, using velocity evaluations at times $t = n\Delta t$ and $(n-1)\Delta t$:

$$\left((\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u}\right)^{n+1/2} = \frac{3}{2}\left(\boldsymbol{u}^n\cdot\nabla\right)\boldsymbol{u}^n - \frac{1}{2}\left(\boldsymbol{u}^{n-1}\cdot\nabla\right)\boldsymbol{u}^{n-1}. \tag{39}$$

The SPH discretisation of the (39) is performed as follows: Starting with the SPH discretisation formula (27) for $(\nabla f)_i$ and calculating its scalar product with a vector $\boldsymbol{u}_i$ leads to

$$(\boldsymbol{u}\cdot\nabla f)_i = \rho_i \sum_j m_j \left(\frac{f_i}{\rho_i^2} + \frac{f_j}{\rho_j^2}\right)\boldsymbol{u}_i\cdot(\boldsymbol{x}_i - \boldsymbol{x}_j)\,\frac{1}{r}\frac{\partial W_{ij}}{\partial r}. \tag{40}$$

If we now replace f by $\boldsymbol{u}$ in (40) and use the scheme (39) we are led to

$$\begin{aligned}\left[(\boldsymbol{u}\cdot\nabla\boldsymbol{u})^{n+1/2}\right]_i = {}& \frac{3\rho_i}{2}\sum_j m_j\left(\frac{\boldsymbol{u}_i^n}{\rho_i^2} + \frac{\boldsymbol{u}_j^n}{\rho_j^2}\right)\boldsymbol{u}_i^n\cdot(\boldsymbol{x}_i - \boldsymbol{x}_j)\,\frac{1}{r}\frac{\partial W_{ij}}{\partial r} \\ & - \frac{\rho_i}{2}\sum_j m_j\left(\frac{\boldsymbol{u}_i^{n-1}}{\rho_i^2} + \frac{\boldsymbol{u}_j^{n-1}}{\rho_j^2}\right)\boldsymbol{u}_i^{n-1}\cdot(\boldsymbol{x}_i - \boldsymbol{x}_j)\,\frac{1}{r}\frac{\partial W_{ij}}{\partial r}.\end{aligned} \tag{41}$$

The remaining terms in the linear momentum equation for $\boldsymbol{u}^*$ are discretised in exactly the same way as in the corresponding equation (28) of the LPSPH method. The SPH discretisation of all other equations in the PMIII projection scheme (see (6)-(10)) is the same as that of the LPSPH formulation of Section 3.1.

A major advantage of using the Eulerian rather than the Lagrangian formulation is the gain in CPU time, as we will see in Section 5.1.1. In the LPSPH method for fluid problems where the velocity is not constant in space, the relative movement of SPH particles makes it necessary to perform a new search of the neighbours of each particle at each time step. For an ensemble of N particles the cost of this search is $O(N^2)$ flops. A further unpleasant consequence of changing particle neighbours is that the coefficient matrices for Eqns. (5) and (8) of the projection scheme must be recomputed at each time step. Much time is therefore saved using the EPSPH

formulation since all particles have the same neighbour list throughout the computations meaning that both the particle neighbour search and construction of the coefficient matrices for the determination of $\boldsymbol{u}^*$ and φ^{n+1} may be performed once only at the start. Unchanging coefficient matrices also means that if a preconditioned iterative method is to be used for the solution of Eqns. (5) and (8) suitable preconditioners, based on incomplete factorisation, for example, need only be calculated once. Another advantage of keeping the SPH particles fixed is that the density of particles in the flow domain remains unchanged. This is not always the case with the LPSPH method where the smoothing length may have to be varied from one particle to another, depending upon the particle number density in each's near neighbourhood. One possibility for overcoming the problem of insufficient particle number density over time is to periodically redistribute the particles as proposed by Chaniotis et al. [6], for example, and this is a strategy that we explore in Section 5.2.1.

3.3. Boundary conditions. It has already been noted that the SPH method requires the quadrature evaluation of integrals of the type (14), defined over a ball $\mathcal{B}$ of radius $2h$. When such a quadrature evaluation is performed for a particle at a distance less than $2h$ from the boundary this requires careful enforcement of boundary conditions.

One of the methods proposed in the literature for the enforcement of the no-penetration velocity boundary condition consists of modelling the boundary with so-called boundary particles that exert a repulsive force on the SPH particles in order to prevent penetration of the boundary by the fluid [40, 44].

Another method used for resolving the problem of how to discretise boundary conditions involves both reference and ghost particles [17, 45]. This approach consists of creating over a distance of $2h$ outside the boundary several rows of so-called ghost particles. Also created within the fluid domain are reference particles situated at a distance in the normal direction from the wall comparable to the initial minimum separation distance of the SPH particles. At each of these reference particles is calculated (using interpolation) a velocity $\boldsymbol{u}_r$ and these reference velocities are used together with that of the boundary $\boldsymbol{u}_b$ to get, by linear extrapolation, the velocities $\boldsymbol{u}_f$ of each of the ghost particles.

The method that we use in our own work is based on image particles [13, 36, 55] defined to be the mirror images of those SPH particles a normal distance of at most $2h$ from the boundary. Each image particle i' carries the same properties (mass, density and viscosity) as the particle i of which it is the image. However, the velocity of an image particle is determined in such a way that the desired velocity boundary conditions are satisfied at each time step. Thus, in order to satisfy the boundary conditions (6)-(7) for the intermediate velocity in the projection scheme PMIII we perform a linear extrapolation of the intermediate velocity value $\boldsymbol{u}_i^*$ of the ith particle and the boundary value

$$\boldsymbol{u}_b + \Delta t \left[\frac{1}{\rho}\nabla\widetilde{\varphi}^{n+1}\right]_{\partial\Omega},$$

to get the intermediate velocity of the image particle i' as

$$\boldsymbol{u}_{i'}^* = 2\left(\boldsymbol{u}_b + \Delta t \left[\frac{1}{\rho}\nabla\widetilde{\varphi}^{n+1}\right]_{\partial\Omega}\right) - \boldsymbol{u}_i^*. \tag{42}$$

As explained at the end of Section 2.2, $\widetilde{\varphi}^{n+1}$ denotes an approximation to the pressure-like scalar field φ^{n+1}.

In order to satisfy the homogeneous Neumann boundary condition

$$\boldsymbol{n} \cdot \boldsymbol{\nabla} \varphi^{n+1} = 0,$$

on φ^{n+1}, the value $\varphi_{i'}$ of the image particle should be defined to be $\varphi_{i'} = \varphi_i$. When the LPSPH method is used the positions of the image particles need to be reassigned at the beginning of each time step since they are the images of moving particles. In contrast, assuming $\boldsymbol{u}_b$ to be constant in time, the image particles are created once and for all at the start of the simulation in the EPSPH method.

Bierbrauer et al. [4] proposed an alternative image particle method for no-slip boundary conditions by approximating the velocity at the image particles using a second order finite difference discretization of the Navier–Stokes equations on the boundary. Their method was shown to enforce the velocity boundary condition with an error that was fourth order in the distance of a fluid particle from the boundary, provided this distance was sufficiently small.

4. Test Problems

4.1. Start-up planar Poiseuille flow. For the first set of tests of our EPSPH and LPSPH methods, we will compute the velocity $\boldsymbol{u}$ of a fluid of constant viscosity μ contained in the two-dimensional channel $\Omega = \{(x, y) : x \in (-\infty, \infty),\ y \in (0, H)\}$ of width H. No-slip conditions are imposed on the channel walls $y = 0$ and H. The fluid, initially at rest, is set in motion by the imposition at all times $t \geq 0$ of a constant force $\boldsymbol{F} = (F, 0)$ for some $F > 0$ (see Eqn. (1)).

Denoting the maximum value as $t \to \infty$ of the streamwise component of the fluid velocity by u_{max} it is easily shown that

$$u_{max} = \frac{FH^2}{8\mu}. \tag{43}$$

If we now use u_{max} as a characteristic speed and H as a characteristic length scale, the non-dimensional exact velocity solution $\hat{\boldsymbol{u}}_{ex} = (\hat{u}_{ex,x}, \hat{u}_{ex,y})$ to the start-up planar Poiseuille problem is given by

$$\begin{aligned} \hat{u}_{ex,x}(\hat{x}, \hat{y}, \hat{t}) &= -4\hat{y}(\hat{y} - 1) \\ &\quad - \frac{32}{\pi^3} \sum_{n=1}^{\infty} \frac{1}{(2n+1)^3} \sin\left((2n+1)\pi\hat{y}\right) \exp\left(-\frac{(2n+1)^2\pi^2}{Re}\hat{t}\right), \end{aligned} \tag{44}$$

$$\hat{u}_{ex,y}(\hat{x}, \hat{y}, \hat{t}) = 0, \tag{45}$$

where we have defined dimensionless variables

$$\hat{\boldsymbol{u}}_{ex} = \boldsymbol{u}/u_{max},\ \hat{x} = x/H,\ \hat{y} = y/H,\ \hat{t} = tu_{max}/H, \tag{46}$$

and where

$$Re := \frac{\rho H u_{max}}{\mu},$$

is the Reynolds number.

In the presentation of the numerical results in Section 5.1 all variables, unless stated otherwise, will be dimensionless and for ease of notation in this section we drop all the hats from these.

4.2. A deformable immersed interface. If one gives a small initial perturbation to a closed elastic immersed membrane that was in equilibrium one expects it to return to its equilibrium state over time. The problem has been used by Tu and Peskin [63] in order to test their Immersed Boundary (IB) method. It has also served as a testbed for the Immersed Interface Method (IIM) for the Stokes

equations [31] and the Navier-Stokes equations (see, for example, [30, 58]). Li and Lai dealt with the steady problem which results with the IIM method [32]. In 2009 Hosseini and Feng [22], following earlier ideas of Tanaka and Takano [59] and Tsubota et al. [61, 62], used an SPH method to model the transport of two-dimensional erythrocytes in capillaries. Beads and springs were used to model the membrane and separate equations of motion were written for the particles on the red cell membrane, these involving bending point forces and elastic extensional forces. The approach proposed in the present paper will be seen to be different, being more akin to the IB method of Peskin [49] where elastic forces generated in the membrane are transmitted to the surrounding fluid.

4.2.1. Modelling of the interface.

Force exerted by a Hookean membrane on the surrounding fluid. Let $\Gamma(t)$ be a simple closed smooth curve representing a Hookean elastic membrane immersed in a two-dimensional viscous fluid. When the membrane is unstretched we will denote its length by L_0 and allow $s \in [0, L_0]$ to denote the arc length measured from some fixed point on the unstretched membrane. s will be used in the paragraphs that follow to label a material point. When the membrane is under tension, the force $\boldsymbol{F}$ on the right-hand side of (1) will include a reaction force of the elastic membrane on the surrounding fluid which, in the absence of any acceleration of the membrane, will be simply the singular elastic force due to the membrane being stretched. Denote by

$$\boldsymbol{X}(s,t) = (X(s,t), Y(s,t)),$$

the position vector at time t of a material point on the interface having label s. Let us call $T = T(s,t)$ the tension in a small element of the membrane at time t. We suppose that the membrane is Hookean, as have many authors before us (see, for example, [28, 30, 63]), so that T is a linear function of the extension of the membrane relative to its unstretched state. That is to say, we write T, according to Hooke's law as

$$T = T_0 \left(\left\| \frac{\partial \boldsymbol{X}}{\partial s} \right\|_2 - 1 \right), \tag{47}$$

where T_0 is Young's modulus. In the unstretched state we note that

$$\left\| \frac{\partial \boldsymbol{X}}{\partial s} \right\|_2 = 1,$$

and that the tension is zero.

A unit tangent vector to $\Gamma(t)$ will be denoted by $\boldsymbol{\tau}(s,t)$ and is given by

$$\boldsymbol{\tau}(s,t) = \frac{\partial \boldsymbol{X}}{\partial s} \Big/ \left\| \frac{\partial \boldsymbol{X}}{\partial s} \right\|_2 . \tag{48}$$

Between two material points $\boldsymbol{X}(s - \Delta s, t)$ and $\boldsymbol{X}(s + \Delta s, t)$, separated in the unstretched state by a small distance $2\Delta s$, the elastic force exerted by the membrane is given approximately by the difference in the tension between these two points. Therefore, at the point $\boldsymbol{X}(s,t)$ we have the approximation

$$2\Delta s \boldsymbol{f}(s,t) \approx T(s + \Delta s, t)\boldsymbol{\tau}(s + \Delta s, t) - T(s - \Delta s, t)\boldsymbol{\tau}(s - \Delta s, t), \tag{49}$$

where $\boldsymbol{f}$ denotes the force density with respect to the arclength in the unstretched state. In the limit as $\Delta s \to 0$ we get

$$\boldsymbol{f}(s,t) = \frac{\partial}{\partial s}(T(s,t)\boldsymbol{\tau}(s,t)). \tag{50}$$

Force experienced by the fluid. The density of the force in the immersed interface is given by (50) and is transmitted to the surrounding fluid. The force $\boldsymbol{F}_m(\boldsymbol{x},t)$ experienced by the fluid due to the membrane that it contains is given by

$$\boldsymbol{F}_m(\boldsymbol{x},t) = \int_0^{L_0} \boldsymbol{f}(s,t)\delta(\boldsymbol{x}-\boldsymbol{X}(s,t))ds. \tag{51}$$

δ here denotes the two-dimensional Dirac delta function. Although (51) means that $\boldsymbol{F}_m(\boldsymbol{x},t) = \boldsymbol{0}$ for all points $\boldsymbol{x}$ not on the membrane itself, in the next section we will replace the Dirac delta function with a kernel function $W(\boldsymbol{x}-\boldsymbol{X}(s,t),h^*)$ for some $h^* > 0$ (see Section 3), thereby distributing the elastic force over some small region of fluid centred at $\boldsymbol{X}(s,t)$.

Discretisation of the interface equations and singular forces. The immersed interface is represented at any point in time $t_n = n\Delta t$ by an ensemble of N_b Lagrangian markers having position vectors $\boldsymbol{X}_k^n = (X_k^n, Y_k^n)$ with $k = 0, 1, \cdots, N_b$. Since the membrane $\Gamma(t)$ is closed, $(X_0^n, Y_0^n) = (X_{N_b}^n, Y_{N_b}^n)$. In all of our simulations these markers are uniformly spaced along the membrane in its unstretched state.

In contrast to Tu et al. [63] who used forward finite difference approximations to the derivatives in Eqns. (47), (48) and (50), we use a cubic spline interpolation of the markers [12, 31, 58] and obtain all these derivatives directly from the interpolating polynomials.

After evaluating the force density $\boldsymbol{f}_k^n$ at each marker point $\boldsymbol{X}_k^n$ on the membrane we should calculate the force $\boldsymbol{F}_m$ to be transmitted to the neighbouring fluid particles. As mentioned above, we replace the delta function in Eqn. (51) with the same form of kernel function W used in Section 3, but with a support radius h^* which may or may not equal the smoothing length h. The integral defining $\boldsymbol{F}_m$ in (51) is therefore approximated as follows:

$$\boldsymbol{F}_m(\boldsymbol{x},t_n) \approx \sum_{k=1}^{N_b} \boldsymbol{f}_k^n W(\boldsymbol{x}-\boldsymbol{X}_k^n,h^*)\Delta s, \tag{52}$$

so that a fluid particle having position vector $\boldsymbol{x}_i$ will experience a force due to the membrane

$$\boldsymbol{F}_m(\boldsymbol{x}_i,t_n) \approx \sum_{k=1}^{N_b} \boldsymbol{f}_k^n W(\boldsymbol{x}_i-\boldsymbol{X}_k^n,h^*)\Delta s. \tag{53}$$

Assembling together all the forces (membrane, imposed pressure gradient etc) that constitute $\boldsymbol{F}^n$ on the right-hand side of Eqn. (5) we may proceed with the calculation of the fluid velocities $\boldsymbol{u}^*$ and $\boldsymbol{u}^{n+1}$.

4.2.2. A semi-implicit scheme for a moving membrane. From the no-slip condition on the membrane surface the latter will move with the local fluid velocity and we can write

$$\frac{\partial \boldsymbol{X}(s,t)}{\partial t} = \boldsymbol{u}(\boldsymbol{X}(s,t)), \tag{54}$$

where the right-hand side is the velocity of the membrane at the material point having position vector $\boldsymbol{u}(\boldsymbol{X}(s,t))$ and satisfies the identity

$$\boldsymbol{u}(\boldsymbol{X}(s,t)) = \int_\Omega \boldsymbol{u}(\boldsymbol{x}(s,t))\delta(\boldsymbol{x}-\boldsymbol{X}(s,t))d\boldsymbol{x}. \tag{55}$$

A completely implicit scheme [31] for the calculation of the update of the Lagrangian markers on the membrane interface would require, at each time step, the computation of the various steps of the projection scheme (5)-(10) at each iteration in an

iterative method until convergence was obtained. This approach, as highlighted by Lee and Leveque [30], is complicated to implement with the Navier-Stokes equations and furthermore is costly in terms of CPU time. We therefore turn to a much simpler semi-implicit scheme described by Lee and Leveque [30] in which iterations at each time step are performed solely in order to calculate the position and velocity of the markers once the velocity $\boldsymbol{u}_i^{n+1} = \boldsymbol{u}^{n+1}(\boldsymbol{x}_i)$ of each of the SPH fluid particles has been computed. The iterations begin with an estimate of the marker velocities which is a discretisation of Eqn. (55):

$$\boldsymbol{u}(\boldsymbol{X}_k^n, t_{n+1})^{[0]} \approx \sum_{j=1}^{N} \boldsymbol{u}^{n+1}(\boldsymbol{x}_j) W(\boldsymbol{X}_k^n - \boldsymbol{x}_j, h) \frac{m_j}{\rho_j}, \tag{56}$$

and we write, in more compact notation,

$$\boldsymbol{u}^{n+1,[0]}(\boldsymbol{X}_k^n) = \sum_{j=1}^{N} \boldsymbol{u}_j^{n+1} W_{kj} \frac{m_j}{\rho_j}, \tag{57}$$

where the notation employed should be obvious. The estimate (57) of the velocity of the marker having position vector $\boldsymbol{X}_k^n$ at time t_n is now used in a forward Euler discretisation over a time step Δt to get a position update

$$\boldsymbol{X}_k^{[0]} = \boldsymbol{X}_k^n + \Delta t \boldsymbol{u}_k^{n+1,[0]}(\boldsymbol{X}_k^n). \tag{58}$$

Now set $I = 0$ and let us use an iterative method to create a sequence of vectors $\boldsymbol{X}_k^{[1]}, \boldsymbol{X}_k^{[2]}, \ldots$ that converge to the root $\boldsymbol{X}$ of the function

$$\boldsymbol{g}(\boldsymbol{X}) = \boldsymbol{X} - \boldsymbol{X}_k^n - \frac{1}{2}\Delta t(\boldsymbol{u}^n(\boldsymbol{X}_k^n) + \boldsymbol{u}^{n+1}(\boldsymbol{X})); \tag{59}$$

where

$$\boldsymbol{u}^{n+1}(\boldsymbol{X}) = \sum_{j=1}^{N} \boldsymbol{u}_j^{n+1} W(\boldsymbol{X} - \boldsymbol{x}_j, h) \frac{m_j}{\rho_j}. \tag{60}$$

The iterations are stopped once $\|g(\boldsymbol{X}_k^I)\| \leq \varepsilon$ for some tolerance ε and some integer I and we then set $\boldsymbol{X}_k^{n+1} = \boldsymbol{X}_k^I$.

Instead of calculating the velocity of the Lagrangian markers as in Eqns. (56) or (60), we have performed the calculations with a normalised form of SPH interpolation [17, 33, 53]:

$$f_i = \frac{\sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j W_{ij}}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} W_{ij}}. \tag{61}$$

This formulation has the advantage that it calculates the exact value of a constant function. In this way, Eqns. (56) and (60) become, respectively,

$$\boldsymbol{u}^{n+1}(\boldsymbol{X}_k^n)^{[0]} = \frac{\sum_{j=1}^{N} \boldsymbol{u}_j^{n+1} W_{kj} \frac{m_j}{\rho_j}}{\sum_{j=1}^{N} W_{kj} \frac{m_j}{\rho_j}}, \tag{62}$$

and

$$\boldsymbol{u}^{n+1}(\boldsymbol{X}) = \frac{\sum_{j=1}^{N} \boldsymbol{u}_j^{n+1} W(\boldsymbol{X} - \boldsymbol{x}_j, h) \frac{m_j}{\rho_j}}{\sum_{j=1}^{N} W(\boldsymbol{X} - \boldsymbol{x}_j, h) \frac{m_j}{\rho_j}}. \tag{63}$$

4.2.3. Membrane in equilibrium. If the membrane Γ is in its unstretched (rest) state and encloses a region of maximum area it will be a circle of radius $r_0 = L_0/2\pi$, provided that it is immersed in a sufficiently large expanse of fluid. See Fig. 1. Let us suppose that the unstretched membrane Γ is centred at the origin. Then if we parametrise Γ by the arc length s measured in an anticlockwise direction from an arbitrary point fixed on Γ the position vector $\boldsymbol{X}$ of a point on the membrane may be written as

$$\boldsymbol{X}(s) = (X(s), Y(s)) = \left(r_0 \cos\left(\frac{s}{r_0}\right), r_0 \sin\left(\frac{s}{r_0}\right)\right), \quad \text{where } s \in [0, L_0]. \tag{64}$$

It is an easy task to verify that in this case

$$\left\| \frac{\partial \boldsymbol{X}}{\partial s} \right\|_2 = 1,$$

and that therefore from (47) the tension $T = 0$. In the rest state the fluid velocity is zero and the pressure is constant everywhere. When the membrane (whether moving with constant velocity or motionless) is under tension then a simple force balance in the normal direction shows that a pressure jump across the membrane is created that satisfies

$$[p] = \frac{\boldsymbol{f} \cdot \boldsymbol{n}}{\|\partial \boldsymbol{X}/\partial s\|_2}, \tag{65}$$

where $[\cdot]$ denotes the jump obtained by subtracting some quantity just to the interior of the membrane from that just to the exterior and $\boldsymbol{n}$ is the outward pointing unit normal vector to Γ. If a closed uniform Hookean membrane is made to assume the form of a circle of radius $r_e > r_0$ (see Fig. 1) and placed in an infinite two-dimensional expanse of quiescent fluid the shape will not change, the fluid velocity will remain zero and in this equilibrium configuration the pressure field will be piecewise constant. The difference between the constant outer pressure value and the constant inner pressure is calculated from (65) and may be shown from Eqns. (47) and (50) to simplify to

$$[p] = -\frac{T_0}{r_e}\left(\frac{r_e}{r_0} - 1\right). \tag{66}$$

5. Numerical Results

In all simulations performed in the present paper, the computational domain Ω_C is two-dimensional and square:

$$\Omega_C = \{(x, y) : a \le x \le b,\ a \le y \le b\}, \tag{67}$$

with no-slip and no-penetration conditions $\boldsymbol{u}_b = \boldsymbol{0}$ on the upper and lower boundaries $y = a$ and b. Periodic boundary conditions are used along $x = a$ and b so that in the LPSPH method particles that leave at one end of the domain re-enter at the other. In this way, the total number of SPH particles in the computational rectangle remains constant. SPH particles at one end of the computational domain interact with those at the other. Initially, a total of $N_x N_y$ particles are placed on a regular grid whose nodes have coordinates (x_i, y_j) $(i = 1, \ldots, N_x,\ j = 1, \ldots, N_y)$ where

$$x_i = a + (i - 1)\Delta x,\ y_j = a + (j - 1/2)\Delta y, \tag{68}$$

and the particle spacings in the two coordinate directions are given by $\Delta x = (b - a)/N_x$ and $\Delta y = (b - a)/N_y$. In the EPSPH method the SPH particles remain in

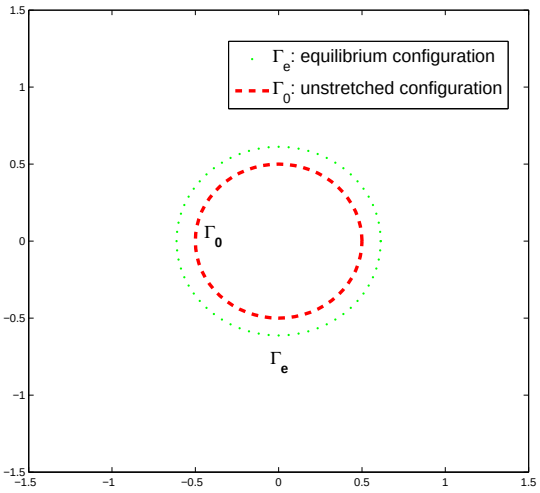


FIGURE 1. Unstretched and equilibrium configurations of a closed simple two-dimensional Hookean elastic membrane

their initial positions at all subsequent times whereas in the LPSPH method they move with the local fluid velocity. Note, however, that despite the movement of the SPH particles in the LPSPH method the inter-particle spacing in the x-direction remains at all times equal to Δx for the test problem of planar Poiseuille flow (Section 4.1).

5.1. Start-up planar Poiseuille flow. The first set of results from our EPSPH and LPSPH methods will be for the start-up Poiseuille problem described in Section 4.1. In order that we might compare our results with those of Bierbrauer et al. [4] we have chosen the following physical parameters: $\mu = 0.001\text{kg m}^{-1}\text{s}^{-1}$, $H = 0.001\text{m}$, $\rho = 1000\text{kg m}^{-3}$ and the maximum streamwise fluid velocity component $u_{max} = 1.25 \times 10^{-5}\text{ms}^{-1}$ This choice of parameters results in a Reynolds number $Re = \rho u_{max} H/\mu = 0.0125$. After non-dimensionalizing all variables as explained in Section 4.1, the problem is solved in the computational domain $\Omega_C = [0,1] \times [0,1]$.

The root mean square (rms) error in the computed velocity $\boldsymbol{u}^{n+1}$ at the $(n+1)$th time step is defined by

$$(\boldsymbol{u}_{ex} - \boldsymbol{u}^{n+1})_{rms} := \frac{1}{\sqrt{N_x N_y}} \left(\sum_{i=1}^{N_x N_y} \left\| \boldsymbol{u}_{ex}(\boldsymbol{x}_i, (n+1)\Delta t) - \boldsymbol{u}_i^{n+1} \right\|_2^2 \right)^{1/2}, \tag{69}$$

where $\boldsymbol{u}_{ex}$ is the exact series solution, whose components are given in (44)-(45). Since all errors in the work of Bierbrauer et al. [4] were evaluated using the rms average (69), this will also be adopted in the presentation of our own results.

In Eqn. (24) the SPH formula for the gradient of a function f at SPH particle i was given as

$$\left(\frac{\partial f}{\partial \boldsymbol{x}} \right)_i \approx - \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \frac{\partial W_{ij}}{\partial \boldsymbol{x}_j}, \tag{70}$$

so that, in one dimension, for a fixed uniform distribution of SPH particles, the corresponding formula for the derivative of a function f at a point x_i becomes

$$\left(\frac{df}{dx}\right)_i \approx -\sum_{j=1}^{N} f_j \frac{\partial W_{ij}}{\partial x_j}\Delta x, \tag{71}$$

where $\Delta x := x_j - x_{j-1}$ $(j = 2, \ldots, N_x)$ is the constant inter-particle spacing. Quinlan et al. [52] derived the following one-dimensional formula in the case of equidistant fixed SPH particles for the error committed by using the SPH approximation (71) for the derivative of a function f at particle i:

$$\begin{aligned}\left(\frac{df}{dx}\right)_i + \sum_{j=1}^{N} f_j \frac{\partial W_{ij}}{\partial x_j}\Delta x = -\frac{h^2}{2}\left(\frac{d^3 f}{dx^3}\right)_i \int s^2 \widehat{W}\, ds + O(h^4) \\ + \left(\frac{\Delta x}{h}\right)^{\beta+2} \frac{B_{\beta+2}}{(\beta+2)!}(1 - 2^{-\beta-1})\left\{\left(\frac{df}{dx}\right)_i [4\widehat{W}_{s=2}^{(\beta+2)} + 2(\beta+1)\widehat{W}_{s=2}^{(\beta+1)}\right] \\ + O(h^2)\} + O\left(\left[\frac{\Delta x}{h}\right]^{\beta+4}\right).\end{aligned} \tag{72}$$

In (72) $W_{ij} = W(x_i - x_j, h)$ is the evaluation at the jth SPH particle of a one-dimensional kernel function centred at x_i and whose support is the interval $[x_i - 2h, x_i + 2h]$ for some $h > 0$. The boundary regularity of W is denoted by β and $\widehat{W} = \widehat{W}(s)$ is a dimensionless kernel function defined by $\widehat{W}(s) = hW(x - x_i, h)$ where $s := (x - x_i)/h$. In (72), superscripts enclosed within parentheses denote derivatives of the order indicated. $B_{\beta+2}$ denotes the $(\beta + 2)$th Bernoulli number. It will be noted from (72) that the SPH method is only consistent in this case if both h and $\Delta x/h$ tend to zero. Equivalently, denoting the number of neighbours within a distance $2h$ of a particle by N_n, we require $N_x \to \infty$ and $N_n \to \infty$ with $N_n/N_x \to 0$ [54].

Although the planar Poiseuille problem is one-dimensional in space it is solved in the present paper, as noted previously, in a two-dimensional domain Ω_C. In order to gain some insight into the behaviour of the rms velocity error for our problem and inspired by (72) we suppose an ansatz for this error, at least for Δx and h sufficiently small, of the form

$$\left(\boldsymbol{u}_{ex} - \boldsymbol{u}^{n+1}\right)_{rms} = Ah^2 + B\left(\frac{\Delta x}{h}\right)^{\beta+2} + \text{h.o.t.}, \tag{73}$$

where A, B are positive constants and "h.o.t." denotes the sum of higher-order terms and the $O(\Delta t^2)$ error terms due to the PMIII projection scheme described in Section 2.2. In the analysis of the errors in our EPSPH scheme in the next section we will assume that all of the terms grouped together in h.o.t. are negligible, at least for Δx, h and Δt sufficiently small. The boundary regularity β of our kernel function W given in (19) is equal to 2.

5.1.1. Results of EPSPH and LPSPH methods. Fig. 2 shows the results of comparing the computed and exact start-up Poiseuille streamwise velocity profiles at dimensionless times varying from 1.25×10^{-5} to 1.25×10^{-2}. The exact solution for u_x is given by Eqn. (44) and the numerical solution has been computed using the EPSPH scheme, although computations with the LPSPH method were nearly identical to those shown in Fig. 2. All the results shown were obtained on a mesh of $N_x \times N_y = 64 \times 64$ SPH particles (corresponding to a particle separation distance of $\Delta x = 1/64$) with a smoothing length of $h = 1.525\Delta x$ and a fixed time step

$\Delta t = 1.25 \times 10^{-6}$. Agreement between the two sets of solutions is excellent at all times.

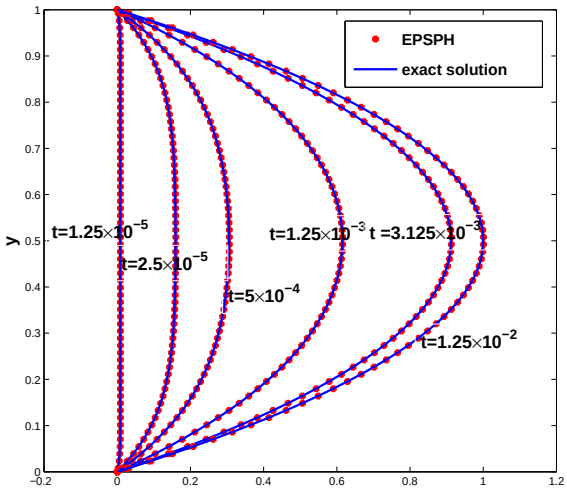


FIGURE 2. Planar Poiseuille problem. Comparison between exact (44) and EPSPH numerical streamwise velocity component profiles. $Re = 1.25 \times 10^{-2}$, $N_x = N_y = 64$, $h = 1.525\Delta x$, $\Delta t = 1.25 \times 10^{-6}$.

Figs. 3 (A) and (B) show, for two different values of the smoothing length h, the evolution in time of the rms velocity errors. In both cases, by the dimensionless time $t = 1.25 \times 10^{-2}$, the errors appear to be essentially independent of time, indicating that the x component of the velocity may be considered to be fully developed at this point. We now turn our attention to the effect of the smoothing length on the rms velocity errors and, in the light of our observations above, calculate these errors at the final time $t = 1.25 \times 10^{-2}$.

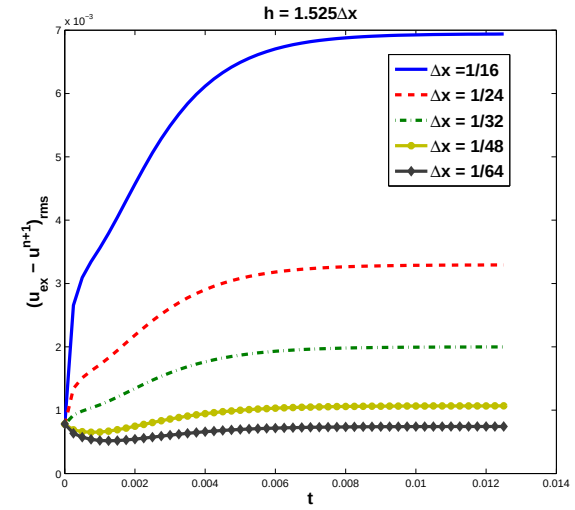
Results of these calculations are shown in Fig. 4 where the log of the rms error for the velocity is plotted, for several different values of the ratio $h/\Delta x$, against the log of Δx. It may be observed that for a given value of Δx the error magnitude is not monotonic in h. Moreover, as Δx decreases, the quadratic convergence, evident for $h/\Delta x \in [1.525, 4]$ and at the largest values of Δx, appears to be lost. Both of these phenomena can be explained, at least qualitatively, on the basis of the ansatz (73). Fixing the ratio of the smoothing length to the inter-particle distance equal to α in (73) and setting $\beta = 2$ leads to

$$\left(\boldsymbol{u}_{ex} - \boldsymbol{u}^{n+1}\right)_{rms} \sim A\alpha^2\Delta x^2 + B\alpha^{-4}, \tag{74}$$

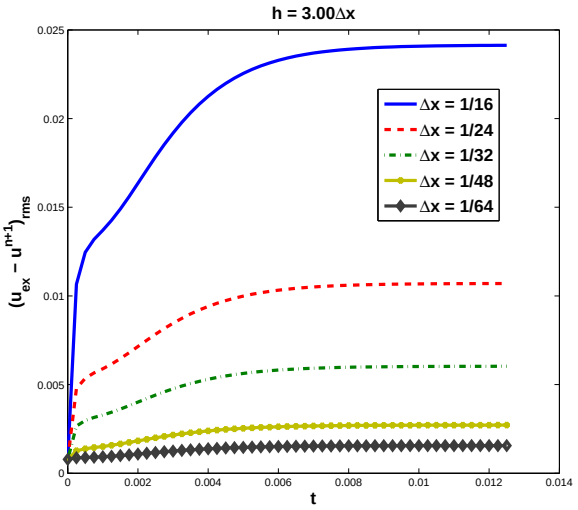
so that differentiating both sides with respect to α and setting the result to zero leads to a minimum at an optimal value

$$\alpha = \alpha_{opt} := \left(\frac{2B}{A\Delta x^2}\right)^{1/6}. \tag{75}$$

The error is therefore expected to decrease with increasing α until the latter equals the optimal value α_{opt} given in (75). Note that the approximate evaluation of α_{opt}



(A)



(B)

FIGURE 3. Planar Poiseuille problem. Evolution with time of rms velocity errors using the EPSPH method. (A) $h = 1.525\Delta x$ and (B) $h = 3\Delta x$. $Re = 1.25 \times 10^{-2}$, $\Delta t = 1.25 \times 10^{-6}$.

in (75) depends on Δx so that as the inter-particle spacing varies the error curves for different values of α might cross one another. A least-squares fit of A and B in (74) to the six data points in Fig. 4 that correspond to $\Delta x = 1/16$ (the largest value of Δx used) yields from (75) a value of α_{opt} equal to 1.43.

For a fixed $\alpha > 1$ and Δx sufficiently large we expect from (74) that the smoothing error $A\alpha^2\Delta x^2$ will dominate over the discretization error $B\alpha^{-4}$ and all the more

so the larger the value of α. Where this is true and taking logs we then see that

$$\log_{10}\left(\left(\boldsymbol{u}_{ex}-\boldsymbol{u}^{n+1}\right)_{rms}\right)\sim\log_{10}(A\alpha^2)-2\log_{10}(1/\Delta x), \tag{76}$$

so that the slope of graphs of the error should be approximately equal to -2, at least for $\log_{10}(1/\Delta x)$ sufficiently small and α sufficiently large. Inspection of Fig. 4 confirms this to be the case.

Finally, in Fig. 4 we see that for Δx sufficiently small we start to lose second-order convergence, particularly for the smaller α value curves. Referring again to (74) this may be seen to be because smoothing error no longer dominates in the truncation error and the discretization error $B\alpha^{-4}$ may no longer be neglected in comparison.

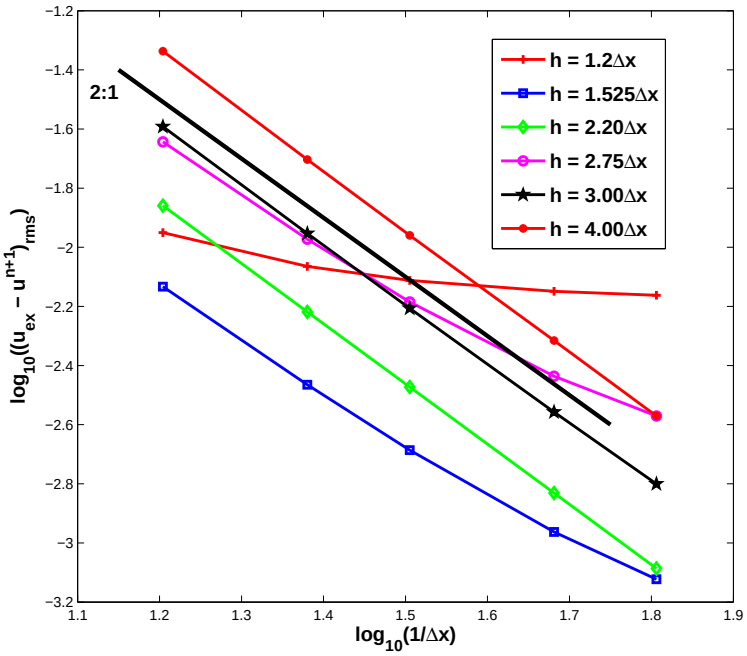


FIGURE 4. Planar Poiseuille problem. Log-log graph of the rms velocity error at $t = 1.25\times10^{-2}$ using the EPSPH method vs. the number of particles in the $x-$direction. $Re = 1.25\times10^{-2}$, $\Delta t = 1.25\times10^{-6}$.

By comparing the smoothing errors in our EPSPH computations of start-up Poiseuille flow in Fig. 4 with those of Bierbrauer et al. [4] in Fig. 5 we note that those in the former case are smaller and that the corresponding order of convergence is higher. The order of convergence of the Bierbrauer et al. computations would seem in some cases to be better than 1 but less than 2. Since the $O(h^2)$ smoothing error predicted in (20) can only be achieved exactly using a set of equidistant quadrature points on a Cartesian grid such as we have in our EPSPH calculations [48], it is tempting to attribute the differences in the two sets of results (those of Bierbrauer et al. and our own) to the non-uniform spacing in the SPH calculations of [4]. However, there are a number of arguments against this. First, and as stated

by the authors in [4], given the small Reynolds number in these calculations, particle motion is limited. We would also add that particles along any line y = constant retain their initial spacing. Secondly, even in the case of particle disordering the order of the dominant smoothing error may be the same, at least for h sufficiently large. The expression corresponding to (72) given by Quinlan et al. [52] in the case of arbitrary particle spacing (see their Eqn. (23)) is rather complicated and precludes easy analysis but predicts leading order terms that are first order in $\Delta x/h$ and order h^{-1} in smoothing length. Despite the divergence as h gets smaller of the smoothing error it is nevertheless dominated by an $O(h^2)$ term for h sufficiently large. Thirdly, we have computed solutions to the same start-up Poiseuille problem using the LPSPH method and show the results in Fig. 6. A quick glance at Fig. 4 shows that particle movement in this case appears to have had very little impact on the smoothing error results.

It is possible that the real reason for the lower-order accuracy in the results of Bierbrauer et al. [4] has something to do with the approach adopted by these authors to the implementation of the boundary conditions (see Section 3.3) [3].

The error analysis of Quinlan et al. [52] in the case of disordered particles may explain why, as Δx decreases (and h with it), there is an upturn in some of the error curves of Bierbrauer et al. [4]. Of course, the increasingly important contribution to the total truncation error of the discretization error (proportional to $\Delta x/h$ if the analysis of Quinlan et al. is applicable) as Δx decreases may also play a role in this upturn.

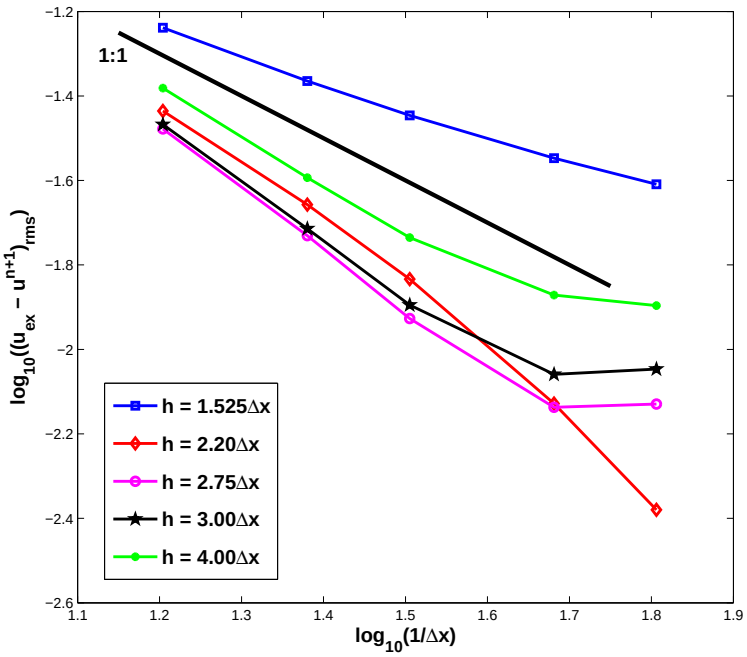


FIGURE 5. Planar Poiseuille problem. Log-log graph of the rms velocity error of Bierbrauer et al. [4] at $t = 1.25 \times 10^{-2}$ vs. the number of particles N_x in the $x-$direction.

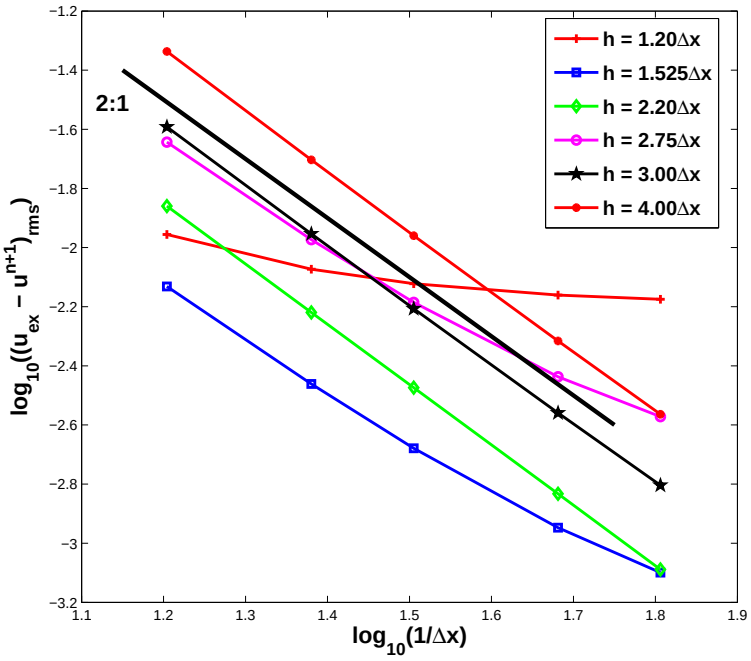


FIGURE 6. Planar Poiseuille problem. Log-log graph of the rms velocity error at $t = 1.25 \times 10^{-2}$ using the LPSPH method vs. the number of particles in the $x-$direction. $Re = 1.25 \times 10^{-2}$, $\Delta t = 1.25 \times 10^{-6}$.

Further evidence of the usefulness of (74) as an aid to understanding the behaviour of the errors incurred with the EPSPH method for the Poiseuille problem may be given with reference to Fig. 7. With Δx fixed, the smoothing error proportional to h^2 dominates over the discretization error which is proportional to $(\Delta x/h)^4$ for h sufficiently large. Exactly *how* large depends on Δx, of course. In this regime we will have

$$\log_{10}\left(\left(\boldsymbol{u}_{ex} - \boldsymbol{u}^{n+1}\right)_{rms}\right) \sim \log_{10}(A) + 2\log_{10}(h), \tag{77}$$

and this explains the slope of 2 for h sufficiently large in all the log-log error curves in Fig. 7. When h is sufficiently small (again, precisely *how small* sufficiently small is depends on Δx) the discretization error term in (74) will dominate and we get

$$\log_{10}\left(\left(\boldsymbol{u}_{ex} - \boldsymbol{u}^{n+1}\right)_{rms}\right) \sim \log_{10}(B\Delta x^4) - 4\log_{10}(h), \tag{78}$$

so that in this case the slope of the error curve on a log-log plot should be equal to -4. This should be most obvious for larger Δx values in Fig. 7. With $\Delta x = 1/24$, for example, the line joining the first and second data points has slope -3.85. It will be noticed, however, that as we further increase Δx in Fig. 7, the slope of the line joining the data points corresponding to the two smallest h values actually decreases. This is because, as stated previously,

$$\left(\boldsymbol{u}_{ex} - \boldsymbol{u}^{n+1}\right)_{rms} \sim Ah^2 + B\left(\frac{\Delta x}{h}\right)^{\beta+2}, \tag{79}$$

is expected to be valid only for Δx and h sufficiently small. As the value of either Δx or h increase the higher-order terms (in the "h.o.t." of (74)) become increasingly important and, in addition, A and B may no longer be justifiably regarded as constants.

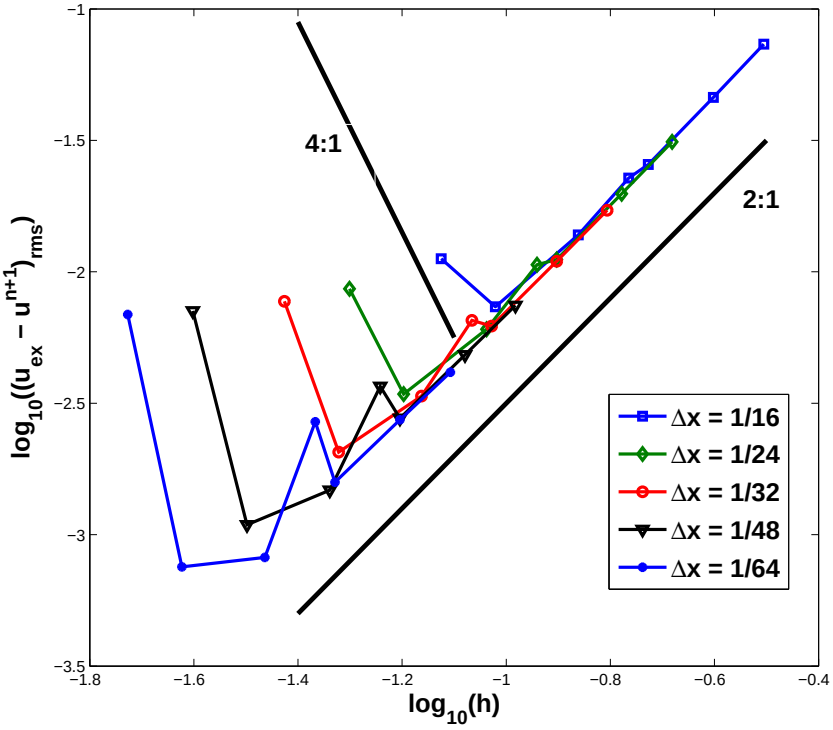


FIGURE 7. Planar Poiseuille problem. Log-log graph of the rms velocity error at $t = 1.25 \times 10^{-2}$ using the EPSPH method vs. the smoothing length h. $Re = 1.25 \times 10^{-2}$, $\Delta t = 1.25 \times 10^{-6}$.

A careful study of the data underlying the graphs shown in Figs. 4 and 6 reveals that the errors incurred with the EPSPH method are only slightly smaller than those of the LPSPH method. In the present context the great gain in fixing the SPH particles lies not in the accuracy of the results, therefore, but rather as has been remarked already in Section 3.2, in the lower computational cost. Tables 1 and 2 show the CPU times (divided by 10^5) in seconds for the solution of the start-up Poiseuille problem using both the EPSPH and LPSPH methods on a 2.66GHz machine with 4GB Ram. The results shown are for a number of different x-direction particle spacings Δx and values of the smoothing parameter h and give the total execution times for calculations performed over the non-dimensional time interval $[0, 1.25 \times 10^{-2}]$ with a dimensionless time step $\Delta t = 1.25 \times 10^{-6}$. For equivalent calculations the EPSPH method is between 2.83 and 5.25 times faster than the LPSPH method. There are two reasons for this. First, when the SPH particles are fixed a search for the neighbours within a distance $2h$ of a particle (an $O(N_x^2 N_y^2)$ flops operation when done for all the particles) need only be done once at the start

since the neighbour list for all the particles does not change. Secondly, when the EPSPH method is used the coefficient matrices in Eqns. (28) and (32) for $\boldsymbol{u}^*$ and φ^{n+1}, respectively, remain the same at all time steps and are therefore constructed just once. All this contrasts sharply with the need in the LPSPH method of both performing neighbour searches and constructing coefficient matrices for the $\boldsymbol{u}^*$ and φ^{n+1} problems at each time step.

TABLE 1. Planar Poiseuille problem. CPU times (divided by 10^5) in seconds as a function of h and Δx for calculations over the dimensionless time interval $[0, 1.25 \times 10^{-2}]$. $\Delta t = 1.25 \times 10^{-6}$. EPSPH method.

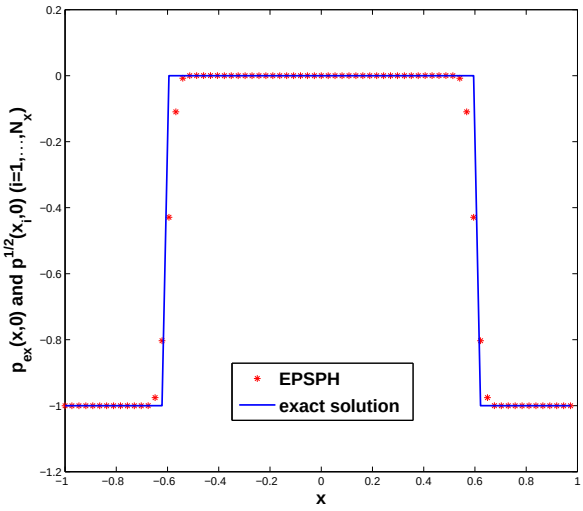
	$\Delta x = 1/16$	$\Delta x = 1/24$	$\Delta x = 1/32$	$\Delta x = 1/48$	$\Delta x = 1/64$
$h = 1.20\Delta x$	0.035022	0.067669	0.113006	0.239269	0.415746
$h = 1.525\Delta x$	0.046215	0.089396	0.147856	0.311473	0.540184
$h = 2.20\Delta x$	0.090997	0.178537	0.294559	0.617233	1.060960
$h = 2.75\Delta x$	0.147899	0.287261	0.472361	0.982132	1.677688
$h = 3.00\Delta x$	0.203369	0.391393	0.640025	1.306324	2.274285
$h = 4.00\Delta x$	0.333689	0.645198	1.042638	2.120318	3.552049
$h = 5.00\Delta x$	0.669753	1.283128	2.119406	4.022646	7.023340

TABLE 2. Planar Poiseuille problem. CPU times (divided by 10^5) in seconds as a function of h and Δx for calculations over the dimensionless time interval $[0, 1.25 \times 10^{-2}]$. $\Delta t = 1.25 \times 10^{-6}$. LPSPH method.

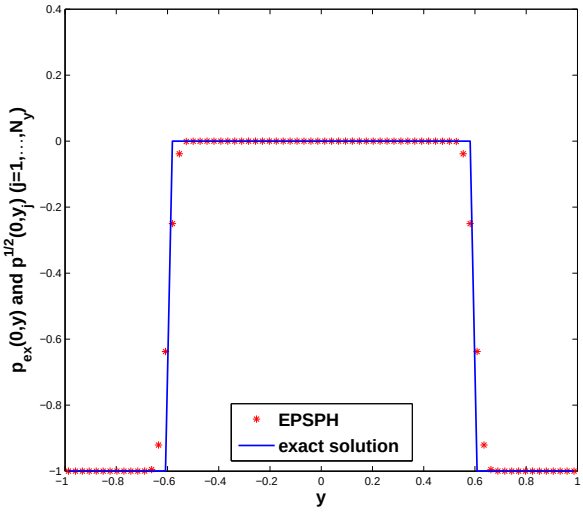
	$\Delta x = 1/16$	$\Delta x = 1/24$	$\Delta x = 1/32$	$\Delta x = 1/48$	$\Delta x = 1/64$
$h = 1.20\Delta x$	0.099005	0.187561	0.319504	0.714037	1.323536
$h = 1.525\Delta x$	0.127923	0.252388	0.424687	0.948320	1.760337
$h = 2.20\Delta x$	0.274136	0.538272	0.908807	2.010848	3.731087
$h = 2.75\Delta x$	0.475463	0.930296	1.573539	3.561883	6.672354
$h = 3.00\Delta x$	0.571796	1.116098	1.891615	4.573303	10.035217
$h = 4.00\Delta x$	1.197621	2.303141	3.975691	9.391495	18.396347
$h = 5.00\Delta x$	2.277747	5.864076	10.2052380	21.126515	-

5.2. Immersed membrane problem.

5.2.1. Steady immersed membrane problem. In order to compute the pressure field created by a Hookean membrane in equilibrium we choose a square computational domain $\Omega_C = [-1.5, 1.5] \times [-1.5, 1.5]$ and choose the rest and equilibrium radii to be $r_0 = 0.5$ and $r_e = 0.6$, respectively. The circular stretched membrane is placed such that its centre is at $(0, 0)$. No-slip boundary conditions are imposed on $y = \pm 1.5$ and periodic boundary conditions in the $x-$direction. Admittedly, wall effects will mean that the equilibrium shape of the membrane in our numerical experiment will not remain perfectly circular of radius r_e but deviations have not been observed to be significant. Choosing $T_0 = 3$ and assuming a circular shape leads to the normal and tangential components of the force density being equal to -1 and 0, respectively and Eqn. (66) predicts that the pressure jump $[p]$ should be equal to -1.



(A)



(B)

FIGURE 8. Hookean membrane in equilibrium. Cross section of $p^{1/2}$ calculated with the EPSPH method along (A) $y = 0$, (B) $x = 0$, compared with exact values. $T_0 = 3$, $\mu = 0.1$, $\rho = 1$, $N_x = N_y = 70$, $N_b = 140$, $\Delta t = 0.1\Delta x$, $2h = 1.2(\Delta x + \Delta y)$, $h^* = 1.4h$.

Figs. 8 (A) and (B) show, respectively, cross sections of the steady $p^{1/2}$ profile along $y = 0$ and along $x = 0$ as calculated with the EPSPH method on a mesh with $N_x = N_y = 70$, $N_b = 140$ and an initial velocity $\boldsymbol{u}^0 = \mathbf{0}$. The viscosity

value $\mu = 0.1$. Due to the use of kernel functions W rather than delta functions in the elastic force evaluation (51) the discontinuity in the pressure across $r = 0.6$ is smoothed.

We begin our study of the accuracy of the LPSPH and EPSPH methods by investigating how the errors in the pressure develop over time from a zero initial velocity $\boldsymbol{u}^0$ if the membrane is fixed in its equilibrium position (a circle of radius $r_e = 0.6$). In Fig. 9 (A)-(F) we show the mean absolute errors $(p_{ex} - p^{n+1/2})_{mad}$ for both methods, where p_{ex} denotes the exact pressure and the mean absolute difference (mad) is defined as follows:

$$(p_{ex} - p^{n+1/2})_{mad} := \frac{1}{N_x N_y} \sum_{i=1}^{N_x N_y} \left| p_{ex}(\boldsymbol{x}_i, (n+1/2)\Delta t) - p_i^{n+1/2} \right|. \tag{80}$$

The graphs (A), (C) and (E) were computed with 1600 particles ($N_x = N_y = 40$) that were initially regularly spaced, whereas those in the right-hand column were computed on a finer mesh of 6400 particles (initially $N_x = N_y = 80$). For all calculations the viscosity μ was set equal to 0.25. In all cases shown, the errors are smaller at a given time when the finer mesh is used, as is to be expected. Figs. 9 (A) and (B) show the growth of the pressure error with time when the LPSPH method is employed with no reinitialisation of the fluid particles. It may be seen that the error suddenly grows significantly at a certain moment in time and this is due to inadequate density of the SPH particles in the vicinity of the membrane. Periodic reinitialisation of the SPH particles (shown in Figs. 9 (C) and (D)) reduces the pressure error almost to its initial value, a conclusion being that that after the first time step most of the growth in error in Figs. 9 (A) and (B) is attributable to movement of the particles. This movement takes place because the pressure jump across the membrane is not exactly balanced by the normal component of the elastic force, thus creating a flow away from the membrane. Further evidence of SPH particle migration near the membrane surface will be presented in our calculations of the membrane position for the time-dependent problem, in Section 5.2.2. In contrast to the LPSPH results, the results of computations on both meshes with the EPSPH method, shown in Figs. 9 (E) and (F), show that the pressure error stabilises rapidly as both the pressure and velocity components converge to their respective steady values.

It may be seen from Figs. 9 (E) and (F) that the pressure error decreases in magnitude with mesh refinement. The EPSPH method will now be used to study in more detail convergence of the steady pressure and velocity. The computed pressure and velocity will be considered to have attained their respective final steady values p^∞ and $\boldsymbol{u}^\infty$ when differences in the pressure and velocity at successive time steps and over all mesh points become, in some sense, sufficiently small. The rms value of the difference of the evaluation at two successive time steps of the pressure at all SPH points is given by

$$(p^{n+1/2} - p^{n-1/2})_{rms} := \frac{1}{\sqrt{N_x N_y}} \left(\sum_{i=1}^{N_x N_y} \left(p_i^{n+1/2} - p_i^{n-1/2} \right)^2 \right)^{1/2}. \tag{81}$$

Similarly, the definition of the rms value of $\boldsymbol{u}^{n+1} - \boldsymbol{u}^n$ is:

$$(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n)_{rms} := \frac{1}{\sqrt{N_x N_y}} \left(\sum_{i=1}^{N_x N_y} \left\| \boldsymbol{u}_i^{n+1} - \boldsymbol{u}_i^n \right\|_2^2 \right)^{1/2}. \tag{82}$$

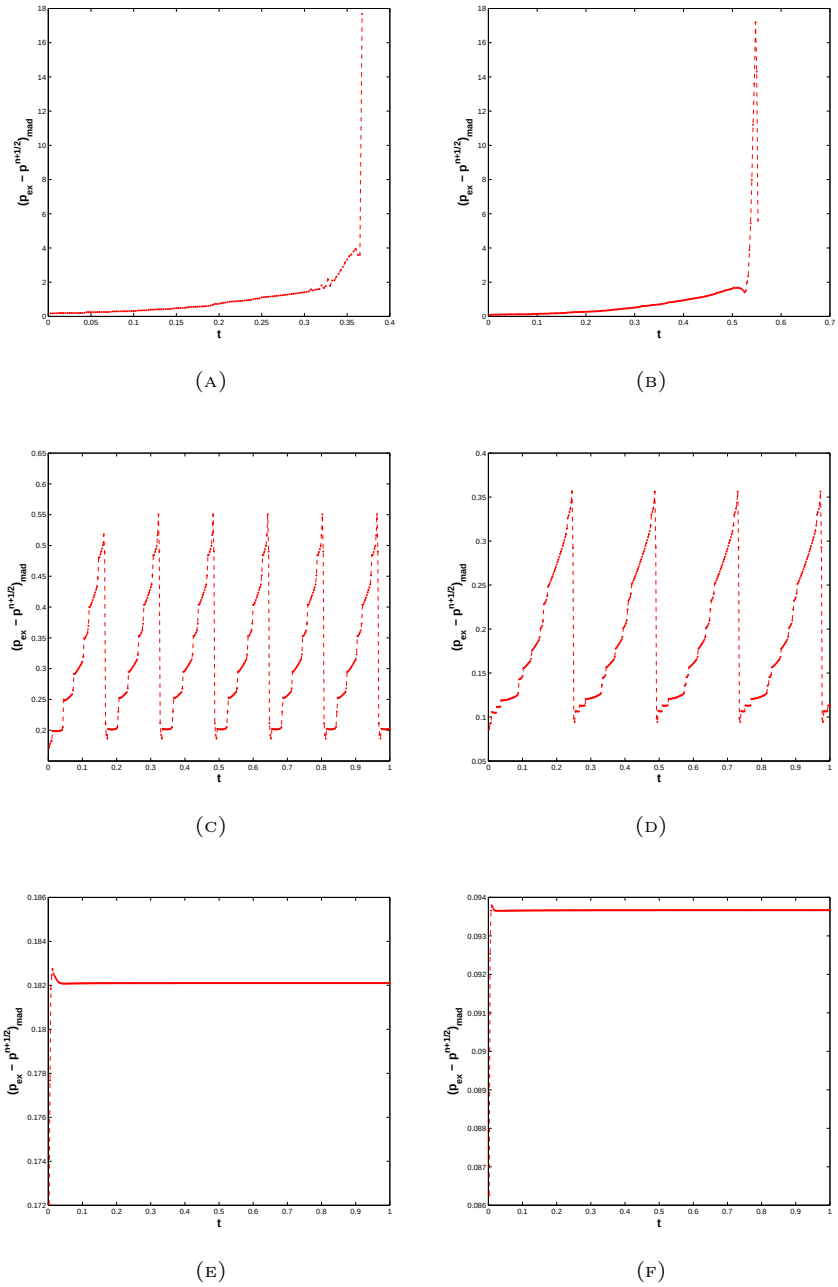
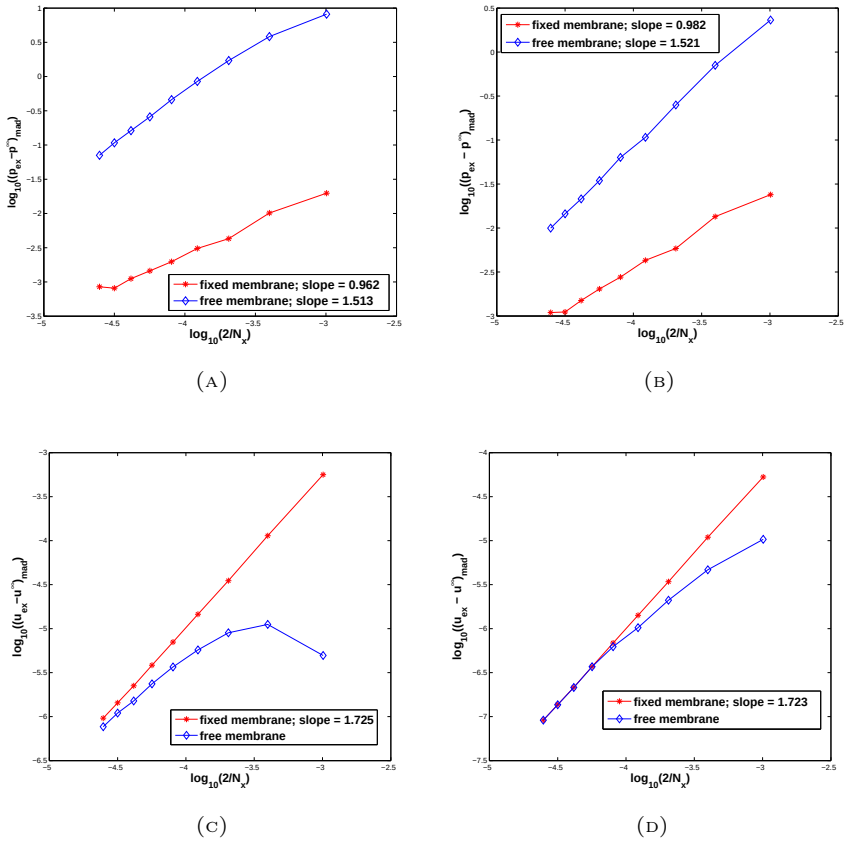


FIGURE 9. Hookean membrane in equilibrium. Evolution of mean absolute pressure error calculated using LPSPH without reinitialisation of the fluid particles ((A)-(B)), LPSPH with reinitialisation of the fluid particles ((C)-(D)) and EPSPH ((E)-(F)). The results shown in figures (A), (C) and (E) were calculated with $N_x = N_y = 40$ initially whereas those in (B), (D) and (F) were calculated with $N_x = N_y = 80$ initially. In all cases $T_0 = 3$, $\mu = 0.25$, $\rho = 1$, $N_b = 360$, $\Delta t = 0.1\Delta x$, $2h = 2.25(\Delta x + \Delta y)$, $h^* = 0.5h$.

The stopping criterion for our time-stepping is that the maximum of $(p^{n+1/2} - p^{n-1/2})_{rms}$ and $(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n)_{rms}$ should be less than $\varepsilon \Delta t$ where Δt is the time step and ε has been set equal to 10^{-3}.

In Figs. 10 (A)-(D) we show the mean absolute errors $(p_{ex} - p^{\infty})_{mad}$ and $(\boldsymbol{u}_{ex} - \boldsymbol{u}^{\infty})_{mad}$ plotted against mesh size for both freely-moving and fixed membranes, where $\boldsymbol{u}_{ex}$ is the exact velocity (in this case equal to $\mathbf{0}$ everywhere) and the mean absolute difference of the steady velocity from the exact value is given by

$$(\boldsymbol{u}_{ex} - \boldsymbol{u}^{\infty})_{mad} := \frac{1}{N_x N_y} \sum_{i=1}^{N_x N_y} \|\boldsymbol{u}_{ex}(\boldsymbol{x}_i, \infty) - \boldsymbol{u}_i^{\infty}\|_1 \,. \tag{83}$$



(A) (B)

(C) (D)

FIGURE 10. Hookean membrane in equilibrium. Mean absolute errors of (A)-(B): the pressure and (C)-(D): the velocity, calculated using the EPSPH method with $N_x = N_y$ for fixed and freely moving membranes. In (A) and (C) $\mu = 0.25$ and in (B) and (D) $\mu = 1$. In all cases $T_0 = 3$, $\rho = 1$, $N_b = 360$, $\Delta t = 0.1\Delta x$, $2h = 2.25(\Delta x + \Delta y)$ and $h^* = 0.5h$.

The results shown in Figs. 10 (A)-(D) were computed on a sequence of meshes with $N_x = N_y = 40, 60, 80, 100, 120, 140, 160$ and 200. The smoothing length h was chosen as $h = 1.125(\Delta x + \Delta y)$ where Δx and Δy are the interparticle spacings and

the radius of the smoothed delta functions placed along the membrane was set as $2h^* = 1.125(\Delta x + \Delta y)$. A constant ratio was maintained between the time step and Δx by setting $\Delta t = 0.1\Delta x$. The choice of viscosity distinguishes Figs. 10 (A) and (C) from Figs. 10 (B) and (D): in the former μ is equal to 0.25 and in the latter $\mu = 1$. The orders of convergence for the pressure calculations in Figs. 10 (A) and (B) are based on the seven largest values of N_x for the free membrane and on the second to sixth largest values of N_x for the fixed membrane.

The results for the pressure show that the errors in the case of the freely moving membrane are significantly higher than in the fixed membrane case. This may be explained, at least partially, by the fact that in the fixed membrane case the tension in the membrane is constant whereas with the freely moving membrane, because of loss of mass, the area enclosed by the membrane gets smaller and so therefore does the tension in the membrane and hence the pressure jump across it. Of course, the exact solution with which we are comparing the computed pressure in all cases is for a fixed circular membrane. In the case of a freely moving membrane an SPH particle might be crossed by the computed membrane position and the pressure be incorrectly evaluated as a consequence at that point. Comparing Figs. 10 (A) and (B) we note that the difference in the pressure errors for fixed and freely moving membranes is smaller in Fig. 10 (B) than in Fig. 10 (A) and that the values of the errors for the moving membrane in Fig. 10 (B) are also smaller. This is because of the higher fluid viscosity for the results shown in Fig. 10 (B): our method is more accurate and the greater fluid viscosity leads to less deformation of the membrane when it is free to move. The order of convergence of the pressure for the freely moving membranes is close to optimal (1.5 [21]) and in the fixed membrane cases is approximately 1 irrespective of the viscosity, similar to IB methods, and therefore much as we might expect [12, 20].

The computation of the velocity is much less sensitive to the issue of whether the membrane moves or not: the exact solution is zero everywhere so that little changes in the computed values when SPH points are crossed by a moving membrane and the error curves almost perfectly coalesce for sufficiently refined meshes. Since the EPSPH method is more accurate with a larger viscosity, the errors in Fig. 10 (D) are smaller than those in Fig. 10 (C) but the rates of convergence are almost quadratic, and therefore optimal, in both cases.

5.2.2. Unsteady immersed membrane problem. The results of this section are for a Hookean membrane initially stretched into the form of an ellipse with semi-major and semi-minor axes of length $r_x(0) = 0.75$ and $r_y(0) = 0.5$, respectively, where $r_x(t)$ and $r_y(t)$ will henceforth denote the half-length of the ellipse axes in the x and y directions at time t. In an unbounded domain the shape of the membrane would therefore be expected to evolve in time to that of a circle of radius $r_e = \sqrt{r_x(0)r_y(0)} \approx 0.61237$. As in our computations in the previous section of a membrane in equilibrium, we choose a square computational domain $\Omega = [-1.5, 1.5] \times [-1.5, 1.5]$ and choose the rest radius to be $r_0 = 0.5$. As previously, the influence of the domain boundaries means that the final equilibrium shape will not be exactly circular but the difference has been found to be very small. For all the results shown in this section, except for those in Table 3 and Fig. 19, we have chosen $2h = 1.2(\Delta x + \Delta y$ and $h^* = 1.4h$.

Although all results presented in this section have been computed with SPH particles having the same density and viscosity throughout Ω, fluids having different properties inside and outside the membrane could be treated even with the EPSPH method by determining at each time step whether a certain number of the SPH

particles lie inside or outside the membrane (we would not have to perform the query for all the SPH particles). If the membrane is considered to be a convex polygon having N_b vertices this would require an initial $O(N_b)$ CPU time pre-processing step followed by an $O(\log N_b)$ CPU time calculation for each query [51].

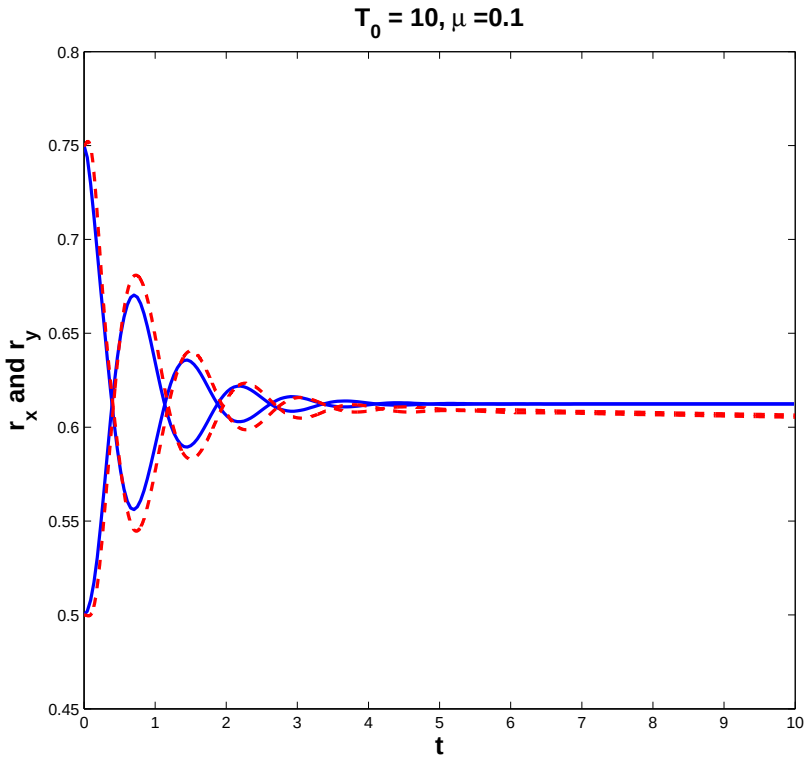
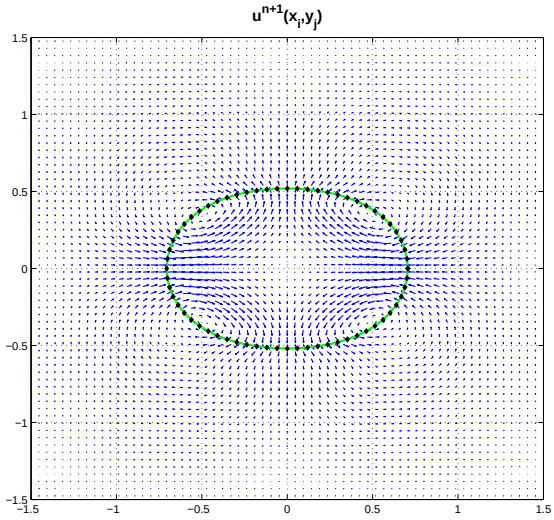


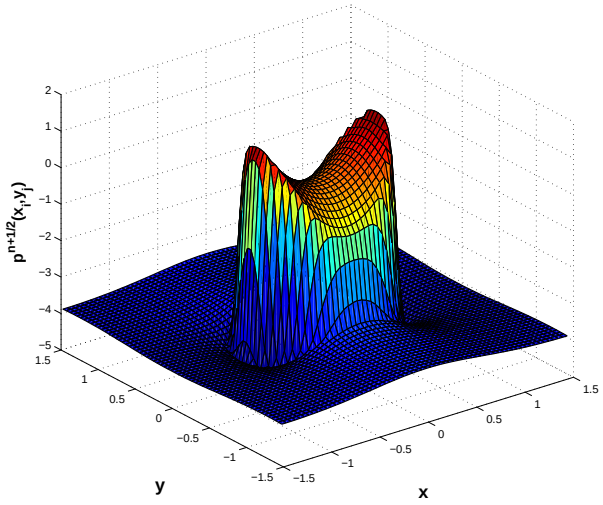
FIGURE 11. Oscillations of a Hookean membrane. Evolution of the initial semi-major axis $r_x(t)$ and semi-minor axis $r_y(t)$. $r_x(0) = 0.75$, $r_y(0) = 0.5$, $T_0 = 10$, $\mu = 0.1$, $\rho = 1$, $N_x = N_y = 64$, $N_b = 128$, $\Delta t = 1 \times 10^{-4}$. —: Tan et al. [58], - -: EPSPH method

In Fig. 11 we show the convergence of $r_x(t)$ and $r_y(t)$ when the equilibrium value of the membrane tension T_0 is set equal to 10 and the fluid viscosity μ both inside and outside the membrane is equal to 0.1. The results of our EPSPH method are compared with those of Tan et al. [58], who used a second-order accurate immersed interface method to investigate the oscillations of an immersed membrane under a variety of choices of viscosities for the fluid inside and outside the membrane. The oscillations seen in both sets of results are due to the fact that the fluid viscosity is insufficient to diffuse the elastic energy quickly enough to make the convergence to steady state monotone. In both sets of computations shown in Fig. 11 the fluid equations were discretised using a 64×64 grid. Although there is good qualitative agreement between our numerical predictions and those of Tan et al. [58] it may be seen that the amplitude of the oscillations in our results is greater than in theirs. We also observe in our case that over there is a progressive loss of mass from the theoretical value πr_e^2 over time. This is unfortunately typical of IB methods [50, 63]

and the use of SPH does not resolve this issue. However, as we show later in this section, the error in the area of fluid enclosed by the membrane at any point in time converges to zero with grid refinement.

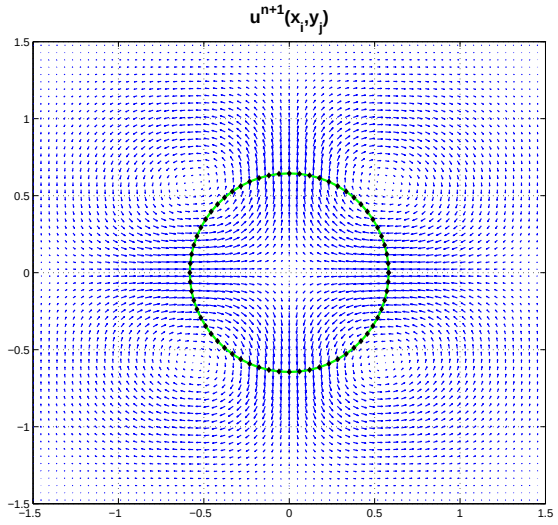


(A)

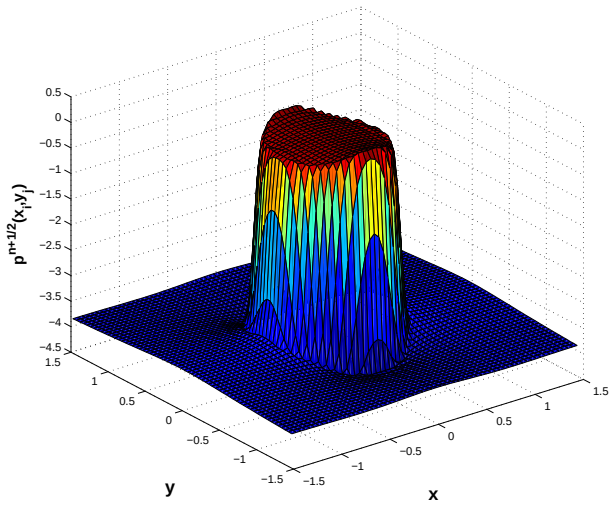


(B)

FIGURE 12. Oscillations of a Hookean membrane. (A) Velocity $\boldsymbol{u}(\boldsymbol{x},t)$ and (B) pressure $p(\boldsymbol{x},t)$ calculated with the EPSPH method. $T_0 = 10$, $\mu = 0.1$, $\rho = 1$, $t = 0.2$, $N_x = N_y = 64$, $N_b = 128$, $\Delta t = 1 \times 10^{-4}$.



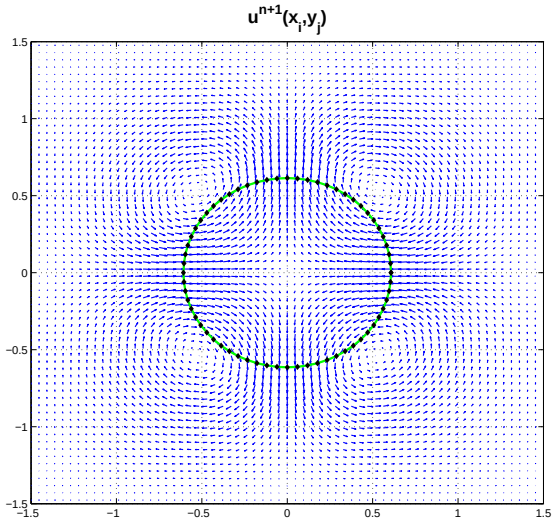
(A)



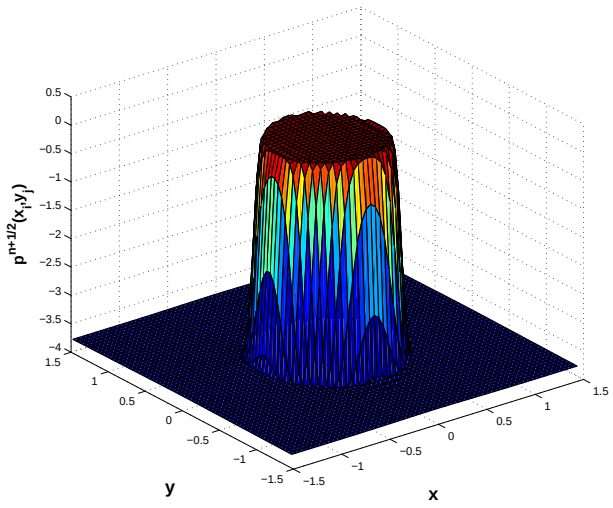
(B)

FIGURE 13. Oscillations of a Hookean membrane. (A) Velocity $\boldsymbol{u}(\boldsymbol{x},t)$ and (B) pressure $p(\boldsymbol{x},t)$ calculated with the EPSPH method. $T_0 = 10$, $\mu = 0.1$, $\rho = 1$, $t = 0.5$, $N_x = N_y = 64$, $N_b = 128$, $\Delta t = 1 \times 10^{-4}$.

In Figs. 12-14 we show the velocity and profiles of the pressure field at $t = 0.2$, $t = 0.5$ and $t = 2.0$. Comparison with the equivalent Fig. 9 of [58] again shows convincing qualitative agreement in our results although, as might be expected with an IB-type method, the regularisation (52)-(53) of the delta functions along



(A)



(B)

FIGURE 14. Oscillations of a Hookean membrane. (A) Velocity $\boldsymbol{u}(\boldsymbol{x},t)$ and (B) pressure $p(\boldsymbol{x},t)$ calculated with the EPSPH method. $T_0 = 10$, $\mu = 0.1$, $\rho = 1$, $t = 2$, $N_x = 64$, $N_y = 64$, $N_b = 128$, $\Delta t = 1 \times 10^{-4}$.

the membrane interface has, as a consequence, a smoothing of the pressure profiles compared to those in [58].

Assuming all other parameters (physical and numerical) are fixed, a decrease in the fluid viscosity has been seen to lead to oscillations of greater amplitude and

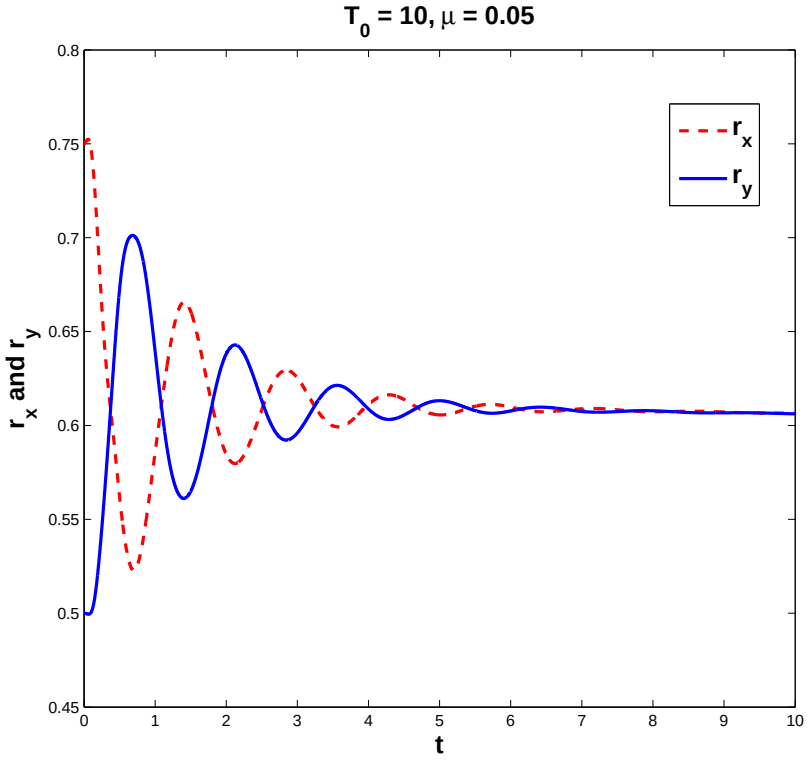


FIGURE 15. Oscillations of a Hookean membrane. Evolution of the initial semi-major and semi-minor axes. - -: $r_x(t)$, —: $r_y(t)$ calculated with the EPSPH method. $r_x(0) = 0.75$, $r_y(0) = 0.5$, $T_0 = 10$, $\mu = 0.05$, $\rho = 1$, $N_x = N_y = 64$, $N_b = 128$, $\Delta t = 1 \times 10^{-4}$.

frequency over the duration of the simulation ($t \in [0, 10]$). This is evidenced in Figs. 15 and 16 where $\mu = 0.05$ and $\mu = 0.025$, respectively. A sufficiently large viscosity value will, on the other hand, lead to monotonic convergence to the equilibrium shape and this is shown, for $\mu = 1$, in Fig. 17.

We have also attempted to solve the time-dependent immersed membrane problem with the LPSPH method and show in Fig. 18 (A)-(D) the membrane and distribution of the SPH particles at times $t = 0.200$, 0.250, 0.275 and 0.285 when the simulation starts from the same initial configuration as the previous EPSPH calculations and the equilibrium membrane tension and viscosity are chosen equal to 10 and 0.1, respectively. It is clear that over time the combined effect of membrane motion and the strong pressure gradients near the membrane surface drive SPH particles away from the membrane, whether these particles are those enclosed by the membrane (shown in red) or outside it (shown in blue). Necessarily this leads to errors since integrals are performed in the SPH method using quadrature rules based on these points. We have observed that the membrane will no longer be elliptic, as may be seen in Fig. 18 (D), for example, and leads, ultimately, to complete breakdown of the simulation.

A number of avenues may be explored in an attempt to correct the problems with the LPSPH method elucidated above. Simply reducing the size of the time

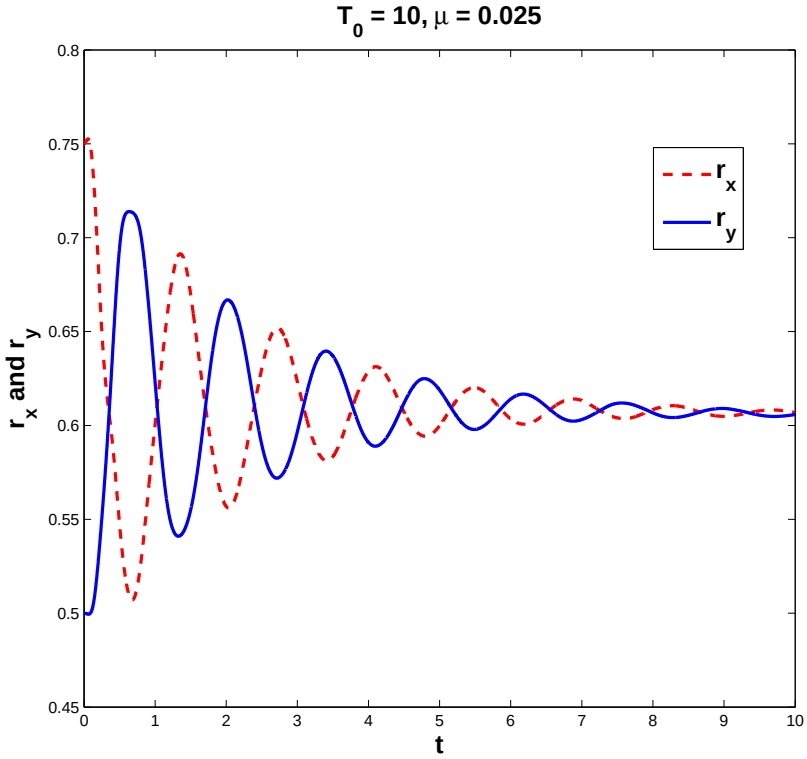


FIGURE 16. Oscillations of a Hookean membrane. Evolution of the initial semi-major and semi-minor axes. - -: $r_x(t)$, —: $r_y(t)$ calculated with the EPSPH method. $r_x(0) = 0.75$, $r_y(0) = 0.5$, $T_0 = 10$, $\mu = 0.025$, $\rho = 1$, $N_x = N_y = 64$, $N_b = 128$, $\Delta t = 1 \times 10^{-4}$.

step Δt or increasing the number of SPH particles increases the cost of the calculations at any time t and only delays the onset of the perturbations to the elliptic form of the membrane. Merely increasing the smoothing length h or the radius $2h^*$ of the regularised delta functions on the membrane does not resolve the issue of particle repulsion. The recent method of Xu et al. [64] shifts SPH particles slightly away from the streamlines and uses Taylor series to calculate the hydrodynamic characteristics at the new particle positions. This would be unlikely to solve the problem of particle deficiency in the present case, however, since the observed particle void is not between particle strings but in the immediate vicinity of the membrane, primarily due to the high pressure gradient created by numerical errors in the computed pressure jump. Another approach consists of redistributing the SPH particles once the number of neighbouring particles of at least one SPH particle drops below a certain tolerance. However, although this restores accuracy to the quadrature rules for the evaluation of integrals it cannot undo the errors in the calculations of the membrane position that have accumulated up to this point. Moreover, the process of reinitialisation necessitates the interpolation of particle velocity and pressure values from the old positions, thus introducing further error.

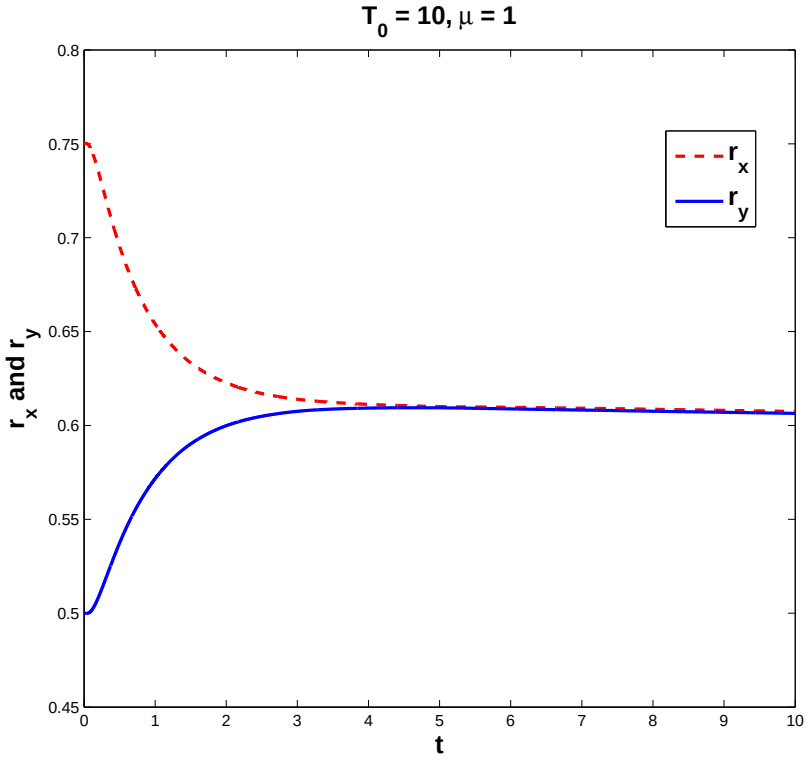
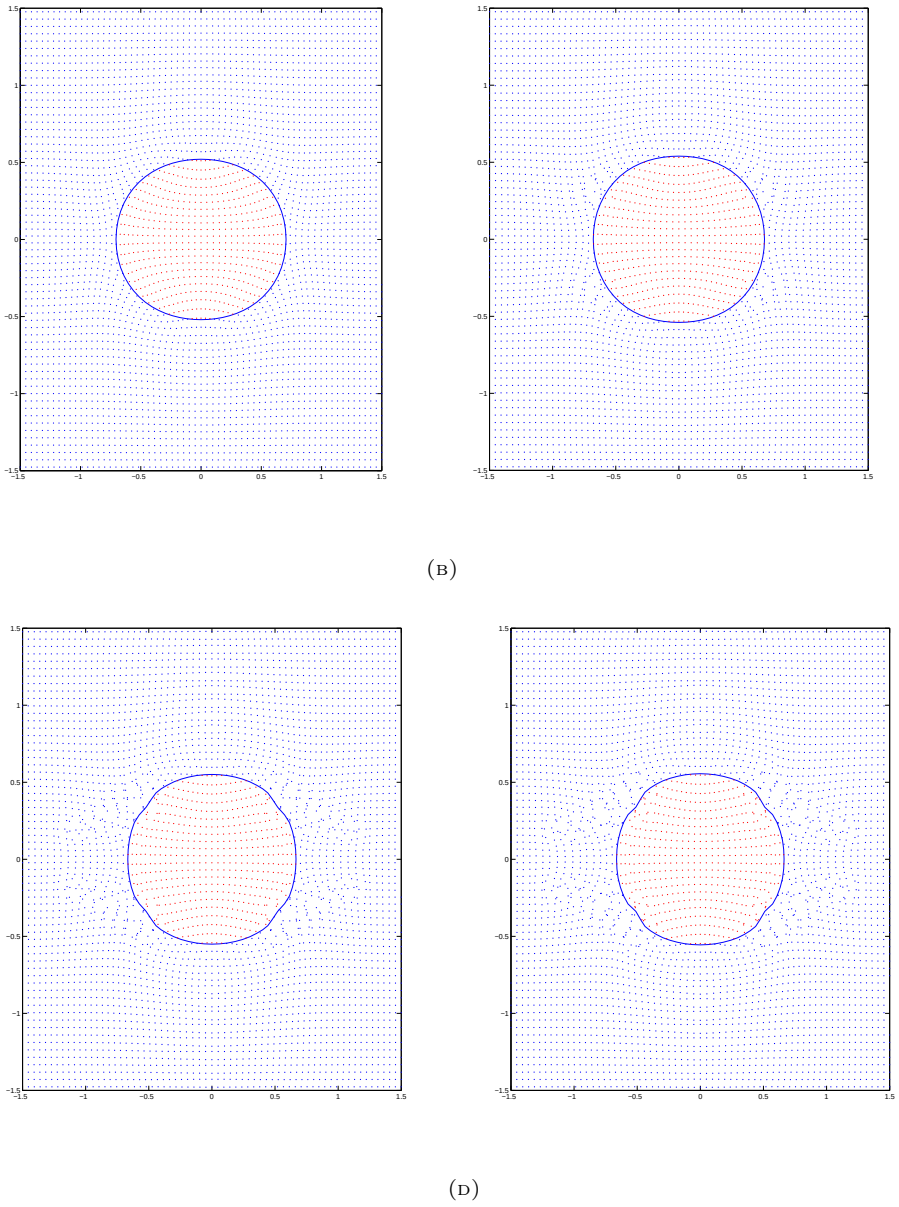


FIGURE 17. Oscillations of a Hookean membrane. Evolution of the initial semi-major and semi-minor axes. - -: $r_x(t)$, —: $r_y(t)$ calculated with the EPSPH method. $r_x(0) = 0.75$, $r_y(0) = 0.5$, $T_0 = 10$, $\mu = 1$, $\rho = 1$, $N_x = N_y = 64$, $N_b = 128$, $\Delta t = 1 \times 10^{-4}$.

Convergence results. In this final part of the present paper we investigate convergence of our EPSPH method for a regularised moving membrane where h^* is held constant. Let us first study the errors committed in calculating the pressure and the x and y components u and v of the velocity of the time-dependent membrane problem at a fixed point in time as we refine the mesh. Since an exact solution is not available we will calculate the values of these variables using three different meshes ($N_x = N_y = 32$, 64 and 128) and compare them with the corresponding values obtained on a refined mesh with $N_x = N_y = 256$. The smoothing length is kept in a fixed ratio (1.2) to the inter-particle distance of separation and the radius of the support of the regularised delta functions on the membrane is fixed equal to 0.085. Let us denote by $\psi_{N_x}^{n+p}$ a function ψ computed on an $N_x \times N_x$ SPH grid at time $(n+p)\Delta t$. Then, for a qth order accurate method we expect [31]

$$\frac{\left|\psi_{N_x}^{n+1/2} - \psi_{256}^{n+1/2}\right|_{max,\Omega}}{\left|\psi_{2N_x}^{n+1/2} - \psi_{256}^{n+1/2}\right|_{max,\Omega}} \approx \frac{C(1/N_x)^q - C(1/256)^q}{C(1/2N_x)^q - C(1/256)^q} = \begin{cases} \dfrac{8^q - 1}{4^q - 1}, & \text{if } N_x = 32, \\ \dfrac{4^q - 1}{2^q - 1}, & \text{if } N_x = 64, \end{cases}$$

where C is a constant and we define in the present context the maximum absolute value $|\cdot|_{max,\Omega}$ of any scalar function defined over the computational domain Ω_C



(A) (B)

(C) (D)

FIGURE 18. Oscillations of a Hookean membrane. Distribution of fluid particles and of the membrane calculated with the LPSPH method at (A) $t = 0.200$, (B) $t = 0.250$, (C) $t = 0.275$ and (D) $t = 0.285$. $T_0 = 10$, $\mu = 0.1$, $\rho = 1$, $N_x = N_y = 64$, $N_b = 128$, $\Delta t = 1 \times 10^{-4}$.

to be the maximum of its absolute value at all the SPH particle positions. Table 3 shows the results of our computations at time $t = 0.0635$ when $T_0 = 1$ and $\mu = 0.05$. The maximum order of convergence appears to be approximately 3 although this has reduced to first order convergence when (**??**) is used with $N_x = 64$. Nevertheless,

the orders of convergence are better than those of the IB method presented in Table 2 of [31], even if the errors themselves are somewhat larger.

TABLE 3. Oscillations of a Hookean membrane. Maximum absolute errors of $p_{N_x}^{n+1/2}$, $u_{N_x}^{n+1}$ and $v_{N_x}^{n+1}$ at time $t = 0.0625$ calculated with the EPSPH method. $T_0 = 1$, $\mu = 0.05$, $\Delta t = \Delta x$, $N_b = 512$, $2h = 1.2(\Delta x + \Delta y)$, $2h^* = 0.085$.

$N_x = N_y$	$\left\|p_{N_x}^{n+1/2} - p_{256}^{n+1/2}\right\|_{max,\Omega}$	q	$\left\|u_{N_x}^{n+1} - u_{256}^{n+1}\right\|_{max,\Omega}$	q	$\left\|v_{N_x}^{n+1} - v_{256}^{n+1}\right\|_{max,\Omega}$	q
32	2.8390×10^{-1}	3.56	5.7788×10^{-1}	3.35	5.8963×10^{-1}	4.23
64	2.3799×10^{-2}	0.77	5.6064×10^{-2}	1.12	3.1248×10^{-2}	1.25
128	8.8049×10^{-3}		1.7687×10^{-2}		9.2301×10^{-3}	

Given that conservation of mass requires that the area of the fluid enclosed remain constant we may calculate the error in the computed enclosed area at any time if the initial value of the enclosed area is known. We have performed such calculations with the EPSPH method at times $t = 1$ and 2 for the same membrane as in the previous experiment described above. The exact area is $0.75 \times 0.5 \times \pi = 0.375\pi$. The results are shown in Fig. 19. These plots were produced from calculations with $N_x = N_y = 40$, 60, 80, 100 and 120 while the number of material points N_b on the membrane was chosen equal to 360 and the time step $\Delta t = 0.1\Delta x$. Although continuous leaking from the inside to the outside takes place throughout the computations, giving rise to a larger error at $t = 2$ than at $t = 1$, we observe third-order convergence with mesh refinement at any time.

In all cases above, allowing h^* to decrease with mesh refinement still led to convergence of both the pressure and velocity, as well as of the area contained of the region enclosed by the membrane, but at lower orders than those reported here for the regularised membrane.

6. Conclusions

In this paper we have developed a truly incompressible SPH method which may be used even for fluid problems having singular sources. Singular force terms in the Navier-Stokes equations are handled in a manner similar to that of the IB method of Peskin [49] via the replacement of delta functions with smooth kernel functions. The incompressible Navier-Stokes equations are discretised in time using a pressure-free projection scheme, originally due to Kim and Moin [27]. In addition to enforcing the incompressibility condition on the fluid velocity a feature of the projection scheme has been shown to be that the normal derivative of the pressure on the boundary, if calculated, is consistent with the time-discretised linear momentum equation, unlike some other SPH projection schemes [13, 29] where inconsistent homogeneous Neumann pressure boundary conditions are imposed. We thereby completely avoid the numerical boundary layers that might be seen when alternative projection schemes are used. In Section 5.1 numerical evidence was presented to show that our scheme has a truncation error of $O(\Delta t^2, h^2, (\Delta x/h)^4)$.

In Section 5.2 it has been shown that the particular problem of particle deficiency observed in the case of a stationary or moving elastic membrane (see Figs. 9 (A)-(D) and 18 (A)-(D), for example) cannot be remedied by using a simple reinitialisation of SPH particles. Since the problem is due to the large pressure gradient created at the membrane interface and not to tensile instability and the formation of particle strings, the particle position reassignment strategy of Xu et al. [64] is unlikely to

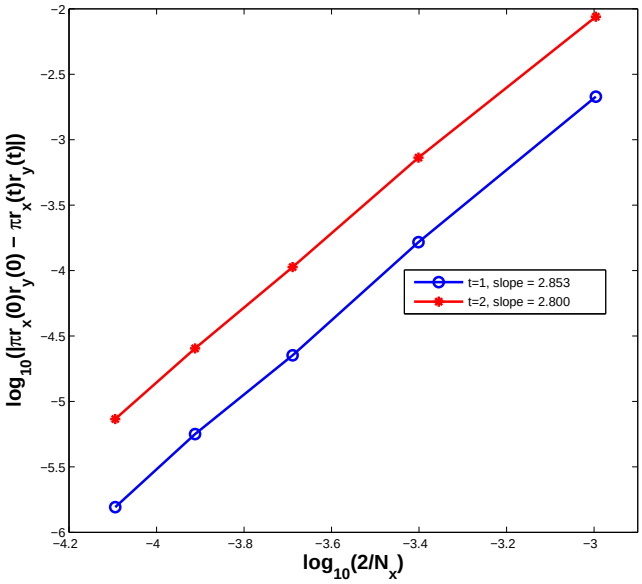


FIGURE 19. Oscillations of a Hookean membrane. Errors in the area enclosed by the membrane vs. $N_x(= N_y)$ at times $t = 1$ and 2, calculated with the EPSPH method. $r_x(0) = 0.75$, $r_y(0) = 0.5$, $T_0 = 1$, $\mu = 0.05$, $\rho = 1$, $\Delta t = 0.1\Delta x$, $N_b = 360$, $2h = 1.2(\Delta x + \Delta y)$, $2h^* = 0.085$.

work either. Using fixed SPH particles in such a situation has been shown to be simple and effective and the results obtained are in very good agreement with both exact solutions and those of Tan et al. [58].

By fixing the SPH particles there is undeniably a loss of some of the flexibility associated with mesh-free methods and a limitation placed on the classes of problems that may therefore be solved. On the positive side, it has been shown in Section 5.1.1 that there are significant computational cost savings to be made in using a fixed SPH grid where the problem being solved allows for this. As evidenced in the same section there are also accuracy advantages in using fixed uniform particle spacings [48, 52]. Moreover, problems may still be solved in complicated geometries since nothing in the method constrains the fixed SPH particles to lie on a regular grid. In the case of a moving closed membrane immersed in a fluid it has been pointed out in Section 5.2.2 that one potential drawback of fixing the SPH particles is that it is more difficult to ascribe different properties (density, viscosity) to the fluids inside and outside the membrane. This may be rectified, however, by ascertaining at each time step whether an SPH particle is within or outside the membrane: a classical problem in computational geometry that, for a convex or star-shaped polygon of N_b vertices, can be solved in $O(\log N_b)$ time per particle once an $O(N_b)$ pre-processing step has been performed [51].

Acknowledgments

The authors wish to thank Frank Bierbrauer for making available the data in Fig. 5. This work was supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.

References

[1] R. Ata and A. Soulaïmani. A stabilized SPH method for inviscid shallow water flows. *Int. J. Numer. Meth. Fl.*, 47:139–159, 2005.

[2] B. Ataie-Ashtiani, G. Shobeyri, and L. Farhadi. Modified incompressible SPH method for simulating free surface problems. *Fluid Dyn. Res.*, 40:637–661, 2008.

[3] F. Bierbrauer. Private communication. 2011.

[4] F. Bierbrauer, P. C. Bollada, and T. N. Phillips. A consistent reflected image particle approach to the treatment of boundary conditions in smoothed particle hydrodynamics. *Comput. Meth. Appl. Mech.*, 198:3400–3410, 2009.

[5] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for incompressible Navier-Stokes equations. *J. Comput. Phys.*, 168:464–499, 2001.

[6] A. K. Chaniotis, D. Poulikakos, and P. Koumoutsakos. Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows. *J. Comput. Phys.*, 182:67–90, 2002.

[7] A. J. Chorin. Numerical solution of the Navier-Stokes squations. *Math. Comput.*, 22:745–762, 1968.

[8] P. W. Cleary and J. J. Monaghan. Conduction modelling using smoothed particle hydrodynamics. *J. Comput. Phys.*, 148:227–264, 1999.

[9] P. W. Cleary, M. Prakash, J. Ha, N. Stokes, and C. Scott. Smooth particle hydrodynamics: status and future potential. *Prog. Comput. Fluid Dyn.*, 7:70–90, 2007.

[10] A. Colagrossi and M. Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *J. Comput. Phys.*, 191:448–475, 2003.

[11] F. Colin, R. Egli, and F. Y. Lin. Computing a null divergence velocity field using smoothed particle hydrodynamics. *J. Comput. Phys.*, 217:580–692, 2006.

[12] M. Conti. Une nouvelle implémentation de la méthode IIM pour les équations de Navier-Stokes avec la présence d'une force singulière. Master's thesis, Université de Montréal, Montréal, Canada, 2008.

[13] S. J. Cummins and M. Rudman. An SPH projection method. *J. Comput. Phys.*, 152:584–607, 1999.

[14] R. Das and P. W. Cleary. Effect of rock shapes on brittle fracture using smoothed particle hydrodynamics. *Theor. Appl. Fract. Mech.*, 53:47–60, 2010.

[15] M. Ellero and N. A. Adams. SPH simulations of flow around a periodic array of cylinders confined in a channel. *Int. J. Numer. Meth. Eng.*, 86:1027–1040, 2011.

[16] M. Ellero, M. Kröger, and S. Hess. Viscoelastic flows studied by smoothed particle hydrodynamics. *J. Non-Newtonian Fluid Mech.*, 105:35–51, 2002.

[17] J. Fang, R. G. Owens, L. Tacher, and A. Parriaux. A numerical study of the SPH method for simulating transient viscoelastic free surface flows. *J. Non-Newtonian Fluid Mech.*, 139:68–84, 2006.

[18] J. Fang and A. Parriaux. A regularized Lagrangian finite point method for simulation of incompressible viscous flows. *J. Comput. Phys.*, 227:8894–8908, 2008.

[19] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.*, 181:375–380, 1977.

[20] Z.-X. Gong, C.-J. Lu, and H.-X. Huang. Accuracy analysis of immersed boundary method using method of manufactured solutions. *Appl. Math. Mech.*, 31:1197–1208, 2010.

[21] J. L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Comput. Meth. Appl. Mech.*, 195:6011–6045, 2006.

[22] S. M. Hosseini and J. J. Feng. A particle-based model for the transport of eythrocytes in capillaries. *Chem. Eng. Sci.*, 64:4488–4497, 2009.

[23] S. M. Hosseini and J. J. Feng. Pressure boundary conditions for computing incompressible flows with SPH. *J. Comput. Phys.*, 230:7473–7487, 2011.

[24] X. Y. Hu and N. A. Adams. A multi-phase SPH method for macroscopic and mesoscopic flows. *J. Comput. Phys.*, 213:844–861, 2006.

[25] X. Y. Hu and N. A. Adams. An incompressible multi-phase SPH method. *J. Comput. Phys.*, 227:264–278, 2007.

[26] X. Y. Hu and N. A. Adams. A constant-density approach for incompressible multi-phase SPH. *J. Comput. Phys.*, 228:2082–2091, 2009.
[27] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comput. Phys.*, 59:308–323, 1985.
[28] A. T. Layton. An efficient numerical method for the two-fluid Stokes equations with moving immersed boundary. *Comput. Meth. Appl. Mech.*, 197:2147–2155, 2008.
[29] E.-S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *J. Comput. Phys.*, 227:8417–8436, 2008.
[30] L. Lee and R. J. Leveque. An immersed interface method for incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 25:832–856, 2003.
[31] R. J. Leveque and Z. Li. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 25(3):709–735, 1997.
[32] Z. Li and M.-C. Lai. The immersed interface method for the Navier-Stokes equations with singular forces. *J. Comput. Phys.*, 171:822–842, 2001.
[33] G. R. Liu and M. B. Liu. *Smoothed Particle Hydrodynamics. A Meshfree Particle Method*. World Scientific, Singapore, 2003.
[34] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Arch. Comput. Method. Eng.*, 17:25–76, 2010.
[35] M. B. Liu, G. R. Liu, K. Y. Lam, and Z. Zong. Smoothed particle hydrodynamics for numerical simulation of underwater explosion. *Comput. Mech.*, 30:106–118, 2003.
[36] E. Y. M. Lo and S. Shao. Simulation of near-shore solitary wave mechanics by an incompressible SPH method. *Appl. Ocean Res.*, 24:275–286, 2002.
[37] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astron. J.*, 88:1013–1024, 1977.
[38] G. W. Ma, X. J. Wang, and Q. M. Li. Modeling strain rate effect of heterogeneous materials using SPH method. *Rock Mech. Rock Eng.*, 43:763–776, 2010.
[39] J. J. Monaghan. Particle methods for hydrodynamics. *Comput. Phys. Rep.*, 3:71–124, 1985.
[40] J. J. Monaghan. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astr.*, 30:543–574, 1992.
[41] J. J. Monaghan. Sph without a tensile instability. *J. Comput. Phys.*, 159:290–311, 2000.
[42] J. J. Monaghan. Smoothed particle hydrodynamics. *Rep. Prog. Phys.*, 68:1703–1759, 2005.
[43] J. J. Monaghan and A. Kocharyan. SPH simulation of multi-phase flow. *Comput. Phys. Commun.*, 87:225–235, 1995.
[44] J. J. Monaghan, A. Kos, and N. Issa. Fluid motion generated by impact. *J. Waterw. Port. C.-ASCE*, 6:250–259, 2003.
[45] J. P. Morris, P. J. Fox, and Yi Zhu. Modeling low Reynolds number incompressible flows using SPH. *J. Comput. Phys.*, 136:214–226, 1997.
[46] R. Nestor, M. Basa, and N. Quinlan. Moving boundary problems in the finite volume particle method. In Pierre Maruzewski, editor, *Proceedings of the ERCOFTAC SIG SPHERIC IIIrd International Workshop*, pages 109–114. Lausanne, Switzerland, 2008.
[47] R. K. Noutcheuwa and R. G. Owens. A mixed Brownian dynamics - SPH method for the simulation of flows of suspensions of bead-spring chains in confined geometries with hydrodynamic interaction. *J. Non-Newtonian Fluid Mech.*, 166:1327–1346, 2011.
[48] G. Oger, M. Doring, B. Alessandrini, and P. Ferrant. An improved SPH method: Towards higher order convergence. *J. Comput. Phys.*, 225:1472–1492, 2007.
[49] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.
[50] C. S. Peskin and B. F. Printz. Improved volume conservation in the computation of flows with immersed elastic boundaries. *J. Comput. Phys.*, 105:33–46, 1993.
[51] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
[52] N. J. Quinlan, M. Basa, and M. Lastiwka. Truncation error in mesh-free methods. *Int. J. Numer. Meth. Eng.*, 66:2064–2085, 2006.
[53] P. W. Randles and L. D. Libersky. Smoothed particle hydrodynamics some recent improvements and applications. *Comput. Meth. Appl. Mech.*, 138:375–408, 1996.
[54] F. A. Rasio. Particle methods in astrophysical fluid dynamics. *Prog. Theor. Phys. Supp.*, 138:609–621, 2000.
[55] S. Shao and E. Y. M. Lo. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with free a surface. *Adv. Water Resour.*, 26:787–800, 2003.

[56] W. Shyy and R. Mittal. Solution methods for the incompressible Navier-Stokes equations. In Richard W. Johnson, editor, *The Handbook of Fluid Dynamics*, pages 31–1 – 31–33. CRC Press LLC, Boca Raton, 1998.

[57] H. Takeda, S. M. Miyama, and M. Sekiya. Numerical simulation of viscous flow by smoothed particle hydrodynamics. *Prog. Theor. Phys.*, 92:939–960, 1994.

[58] Z. Tan, D.V. Le, Z. Li, K. M. Lim, and B. C. Khoo. An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane. *J. Comput. Phys.*, 227:9955–9983, 2008.

[59] N. Tanaka and T. Takano. Microscopic-scale simulation of blood flow using SPH method. *Int. J. Comput. Meth.*, 2:558–568, 2005.

[60] L. J. P. Timmermans, P. D. Minev, and F. N. van de Vosse. An approximate projection scheme for incompressible flow using spectral elements. *Int. J. Numer. Meth. Fl.*, 22:673–688, 1996.

[61] K. Tsubota, S. Wada, and T. Yamaguchi. Particle method for computer simulation of red blood cell motion in blood flow. *Comput. Meth. Prog. Bio.*, 83:139–146, 2006.

[62] K. Tsubota, S. Wada, and T. Yamaguchi. Simulation study on effects of hematocrit on blood flow properties using particle method. *J. Biomech. Sci. Eng.*, 1:159–170, 2006.

[63] C. Tu and C. Peskin. Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods. *SIAM J. Sci. Comput.*, 13:1361–1376, 1992.

[64] R. Xu, P. Stansby, and D. Laurence. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *J. Comput. Phys.*, 228:6703–6725, 2009.

[65] G. M. Zhang and R. C. Batra. Symmetric smoothed particle hydrodynamics (SSPH) method and its application to elastic problems. *Comput. Mech.*, 43:321–340, 2009.

Département de physique, Université de Montréal C.P. 6128, succ. centre-ville Montréal QC H3C 3J7, Canada.

E-mail: keourodrigue@yahoo.fr

Département de mathématiques et de statistique, Université de Montréal C.P. 6128, succ. centre-ville Montréal QC H3C 3J7, Canada.

E-mail: owens@dms.umontreal.ca

URL: http://www.dms.umontreal.ca/Professeurs/owens