Computers & Fluids 91 (2014) 47–56

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Moving least squares simulation of free surface flows

C.L. Felter [a,*], J.H. Walther [b,c], C. Henriksen [d]

[a] MAN Diesel & Turbo, Teglholmsgade 41, 2450 Copenhagen SV, Denmark
[b] Department of Mechanical Engineering, Technical University of Denmark, Building 403, 2800 Kgs. Lyngby, Denmark
[c] Computational Science and Engineering Laboratory, ETH Zürich, Clasiusstrasse 33, CH-8092 Zürich, Switzerland
[d] Department of Mathematics, Technical University of Denmark, Building 303, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 7 June 2012
Received in revised form 15 August 2013
Accepted 3 December 2013
Available online 16 December 2013

Keywords:
Free surface
Meshfree
Moving least squares

ABSTRACT

In this paper a Moving Least Squares method (MLS) for the simulation of 2D free surface flows is presented. The emphasis is on the governing equations, the boundary conditions, and the numerical implementation. The compressible viscous isothermal Navier–Stokes equations are taken as the starting point. Then a boundary condition for pressure (or density) is developed. This condition is applicable at interfaces between different media such as fluid–solid or fluid–void. The effect of surface tension is included. The equations are discretized by a moving least squares method for the spatial derivatives and a Runge–Kutta method for the time derivatives. The computational frame is Lagrangian, which means that the computational nodes are convected with the flow. The method proposed here is benchmarked using the standard lid driven cavity problem, a rotating free surface problem, and the simulation of drop oscillations. A new exact solution to the unsteady incompressible Navier–Stokes equations is introduced for the rotating free surface problem.

1. Introduction

Modelling flows with a free surface is a non-trivial task because the fluid domain changes during the simulation. In the VOF method [1] the liquid and the gas phase is modelled on a fixed grid of nodes (Eulerian approach). This means that the fluid–gas interface is *captured* in an average sense by some of the grid cells. In the Level-Set Method [2] the boundary is given implicitly as a level curve of a scalar field defined over the entire solution domain. This is also a Eulerian approach with approximate representation of the free surface. Another class of methods exists in which Lagrangian coordinates are used. One example is the SPH method [3] in which particles interact with each other in a pair-wise fashion and convect with the flow.

The method proposed here is based on the Reproducing Kernel Method [4] which is a kind of gridless finite difference method. It is also a Lagrangian approach and allows for explicit *tracking* of the free surface using special surface nodes. A downside is the non-uniform distribution of the nodes as the simulation progresses. Therefore, redistribution of the computational nodes is necessary at certain intervals. The basis functions used for the spatial derivatives can also be used for the interpolation of the current field values onto the new node set. MLS is basically a *centered* scheme, which has a drawback regarding decoupling modes. However, the redistribution of nodes has a smoothening effect, which in many cases helps to keep the unwanted modes under control.

The main benefits of the proposed method are:

(i) In a two-phase flow problem where one phase can be considered void (with a hydrostatic pressure) the computational domain needs only include the other phase. In such cases the simulation problem becomes smaller than e.g. VOF and Level-Set Method simulations.
(ii) No global system of equations needs to be solved.
(iii) The method can handle large deformations without the need to redefine a computational mesh.

2. Governing equations

The Navier–Stokes equations express the rate of change of momentum for a fluid particle. Using tensor notation and Cartesian coordinates they may be stated as [5]

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \frac{1}{\rho}\frac{\partial \sigma_{ij}}{\partial x_j} \tag{1}$$

where $\mathrm{d}(\cdot)/\mathrm{d}t$ denotes the material derivative with respect to time, and

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} \tag{2}$$

is the stress tensor. The symbol δ_{ij} is the Kronecker delta, and the shear stress τ_{ij} is given by

* Corresponding author. Tel.: +45 33852697.
E-mail addresses: ChristianL.Felter@man.eu (C.L. Felter), walther@mavt.ethz.ch, jhw@mek.dtu.dk (J.H. Walther), Christian.Henriksen@mat.dtu.dk (C. Henriksen).

0045-7930/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compfluid.2013.12.006

$$\tau_{ij} = \mu \varepsilon_{ij} \tag{3}$$

where μ is the dynamic viscosity, and ε_{ij} is the deformation rate tensor

$$\varepsilon_{ij} = \frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} - \frac{2}{3}\frac{\partial v_k}{\partial x_k}\delta_{ij} \tag{4}$$

For a 2D problem equations Eqs. (1)–(4) give the Navier–Stokes equations for a compressible fluid with constant viscosity

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= \frac{1}{\rho}\left[-\frac{\partial p}{\partial x} + \mu\left(\frac{4}{3}\frac{\partial^2 u}{\partial x^2} + \frac{1}{3}\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2}\right)\right] \\ \frac{\mathrm{d}v}{\mathrm{d}t} &= \frac{1}{\rho}\left[-\frac{\partial p}{\partial y} + \mu\left(\frac{4}{3}\frac{\partial^2 v}{\partial y^2} + \frac{1}{3}\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 v}{\partial x^2}\right)\right] \end{aligned} \tag{5}$$

In Eqs. (1) and (5) the left hand side of the momentum equations is written using the material derivative. This notation indicates that Lagrangian coordinates are used. Lagrangian coordinates are associated with points that follow the deformation of the media which is expressed by the following equations

$$\frac{\mathrm{d}x}{\mathrm{d}t} = u \quad \frac{\mathrm{d}y}{\mathrm{d}t} = v \tag{6}$$

The continuity equation expresses the conservation of mass, and may be written in Lagrangian form as [5]

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \tag{7}$$

The present model assumes isothermal flow and the system is closed using the ideal gas law

$$p = C_0 \rho \tag{8}$$

where C_0 is a constant. The speed of sound a relates to the pressure–density function in the following way

$$a^2 = \frac{\partial p}{\partial \rho} = C_0 \tag{9}$$

Thus the speed of sound is controlled directly by the gas constant.

3. Boundary conditions

For the problems considered two types of boundary conditions are needed, namely a solid wall and a free surface.

3.1. Solid wall

On solid bodies the no-slip condition is used. The physical meaning is that there can be no relative velocity between particles on the solid and fluid particles touching the solid. Therefore, the velocity field of the fluid must equal the velocity on the solid at the solid/fluid interface

$$u_{\text{fluid}} = u_{\text{wall}} \quad v_{\text{fluid}} = v_{\text{wall}} \tag{10}$$

In order to evaluate stresses and forces on solid walls the pressure on the solid boundary is needed (see Eq. (2)).

A boundary condition for pressure (or equivalently density) is derived from the momentum equations Eq. (5) and the normal vector on the solid $\mathbf{n} = (n_x\ n_y)^T$. Taking the scalar product between these vectors gives

$$\begin{aligned} n_x\frac{\mathrm{d}u}{\mathrm{d}t} + n_y\frac{\mathrm{d}v}{\mathrm{d}t} &= n_x\frac{1}{\rho}\left[-\frac{\partial p}{\partial x} + \mu\left(\frac{4}{3}\frac{\partial^2 u}{\partial x^2} + \frac{1}{3}\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2}\right)\right] \\ &\quad + n_y\frac{1}{\rho}\left[-\frac{\partial p}{\partial y} + \mu\left(\frac{4}{3}\frac{\partial^2 v}{\partial y^2} + \frac{1}{3}\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 v}{\partial x^2}\right)\right] \end{aligned} \tag{11}$$

Using the notation $\partial p/\partial \mathbf{n} = n_x \partial p/\partial x + n_y \partial p/\partial y$ Eq. (11) may be rearranged in the following way

$$\begin{aligned} \frac{\partial p}{\partial \mathbf{n}} &= -\rho\left(n_x\frac{\mathrm{d}u}{\mathrm{d}t} + n_y\frac{\mathrm{d}v}{\mathrm{d}t}\right) + n_x\left[\mu\left(\frac{4}{3}\frac{\partial^2 u}{\partial x^2} + \frac{1}{3}\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2}\right)\right] \\ &\quad + n_y\left[\mu\left(\frac{4}{3}\frac{\partial^2 v}{\partial y^2} + \frac{1}{3}\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 v}{\partial x^2}\right)\right] \end{aligned} \tag{12}$$

which constitutes a Neumann boundary condition for pressure p.

The partial derivatives of pressure in the left hand side of Eq. (12) can be replaced by expressions obtained from the equation of state Eq. (12)

$$\frac{\partial p}{\partial x} = C_0\frac{\partial \rho}{\partial x} \quad \frac{\partial p}{\partial y} = C_0\frac{\partial \rho}{\partial y} \tag{13}$$

With this substitution Eq. (12) becomes a single inhomogeneous Neumann condition for the single unknown ρ.

3.2. Free surface

The condition which must be satisfied at a free surface is the balance of normal and shear stresses. A free surface is sketched in Fig. 1. The traction at the fluid side of the free surface interface is composed of two terms. One term stems from the stress tensor associated with the fluid, the other term is due to surface tension. Surface tension gives rise to a jump in normal stress f, which is proportional to the curvature κ of the free surface

$$f = \kappa\sigma = \frac{x'y'' - y'x''}{\left((x')^2 + (y')^2\right)^{3/2}}\sigma \tag{14}$$

where σ is the *surface tension* and x and y describes the free surface expressed in terms of the arc length parameter τ.

A boundary condition for the free surface is obtained by requiring a balance of the normal and shear stress at either side of the interface

$$\begin{aligned} \sigma^{\mathrm{A}}_{ij} n_i n_j + \kappa\sigma &= \sigma^{\mathrm{B}}_{ij} n_i n_j \\ \sigma^{\mathrm{A}}_{ij} n_i t_j &= \sigma^{\mathrm{B}}_{ij} n_i t_j \end{aligned} \tag{15}$$

where

σ_{ij}	is the stress tensor
n_i	is the normal vector
t_i	is the tangent vector
A, B	denote media A and media B, respectively

Let medium A be a compressible fluid; the stress tensor then becomes (from Eq. (12)

$$\sigma^{\mathrm{A}}_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} - \frac{2}{3}\frac{\partial v_k}{\partial x_k}\delta_{ij}\right)$$

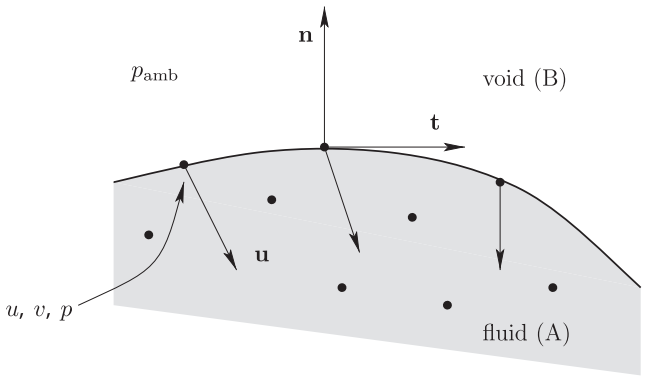

Fig. 1. Sketch of a free surface. Computational nodes carry velocity components u, v, and density ρ and are shown as dots. The surrounding media is considered void with a constant pressure p_{amb}.

Medium B is considered as a void with zero viscosity. In this case the stress tensor associated with medium B reduces to

$$\sigma_{ij}^{B} = -p_{amb}\delta_{ij}$$

where p_{amb} is the ambient pressure. For 2D problems the following relationship between normal vector and tangent vector exists

$$n_x = -t_y \text{ and } n_y = t_x$$

Substituting these expressions in Eq. (15) the following boundary conditions for the free surface interface are obtained

$$-p - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right) + 2\mu\left(\frac{\partial u}{\partial x}n_x^2 + \frac{\partial v}{\partial y}n_y^2 + \left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)n_x n_y\right) \quad + \kappa\sigma = -p_{amb} \tag{16}$$

$$2\left(\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}\right)n_x n_y + \left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\left(n_y^2 - n_x^2\right) = 0 \tag{17}$$

where all terms are evaluated at the free surface. It is seen that the two equations have three unknowns, namely u, v, and p. Therefore the density boundary condition Eq. (12) is also enforced on the free surface. This gives three simultaneous equations for three unknowns. The system is linear due to the linearity of the ideal gas law.

4. Discretization

The continuous system of equations is discretized using the *semi-discretization* method. The idea of the semi-discretization method is to separate the discretization of time from the discretization of space.

A very useful feature of the procedure is that a partial differential equation, upon discretization, can be treated as a set of ordinary differential equations.

4.1. Spatial derivatives

Spatial derivatives are obtained using the method of moving least squares [6]. This method belongs to the family of meshfree methods and can be derived from different starting points. For example, the method coincides with the Reproducing Kernel Method [7] which is inspired by the earlier Smoothed Particle Hydrodynamics (SPH) [8]. On the other hand one can interpret MLS as a generalization of the finite difference method [9].

The idea is to approximate a continuous field by patches of polynomial surfaces. Each surface is fitted to the data points in the sense of least squares, using only local information. An approximating surface will be associated with each computational point, see Fig. 2.

The field variable u is approximated by

$$u^h(\mathbf{x},\mathbf{x}^*) = \sum_{i=1}^{N} p_i(\mathbf{x}-\mathbf{x}^*)a_i(\mathbf{x}^*) \tag{18}$$

where $\mathbf{x}$ denotes the point of evaluation, $\mathbf{x}^*$ is the point of expansion, $p_i(\mathbf{x})$ and $a_i(\mathbf{x}^*)$ are basis functions and coefficients,

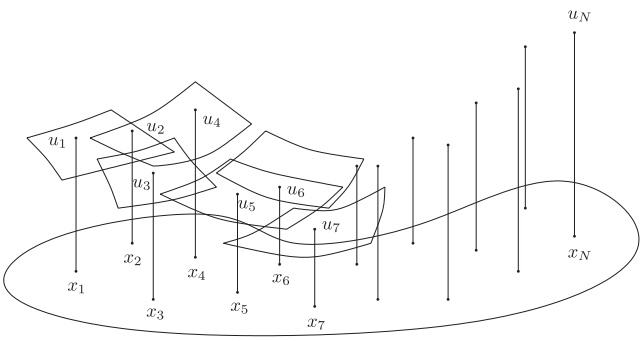

Fig. 2. Polynomial surfaces approximating discrete values from the true field u.

respectively, and N is the number of basis functions. This expression allows the computation of an approximation u^h at the point $\mathbf{x}$. In the following it is assumed that the point of evaluation $\mathbf{x}$ and the point of expansion $\mathbf{x}^*$ coincide at some nodal point $\mathbf{x}_k$. The basis functions form a nomial basis, which is complete up to second order (excluding the constant term)

$$\begin{aligned} &p_1(\mathbf{x}) = x \quad p_2(\mathbf{x}) = y \\ &p_3(\mathbf{x}) = x^2 \quad p_4(\mathbf{x}) = xy \quad p_5(\mathbf{x}) = y^2 \end{aligned} \tag{19}$$

In order to find the coefficients $\mathbf{a}$ associated with nodal point $\mathbf{x}_k$ the following functional is defined

$$\begin{aligned} J(\mathbf{x}_k) &= \sum_{j=1}^{M} W(h,\mathbf{x}_j-\mathbf{x}_k)\left[u^h(\mathbf{x}_j,\mathbf{x}_k) - (u_j-u_k)\right]^2 \\ &= \sum_{j=1}^{M} W(h,\mathbf{x}_j-\mathbf{x}_k)\left[\sum_{i=1}^{N} p_i(\mathbf{x}_j-\mathbf{x}_k)a_i(\mathbf{x}_k) - (u_j-u_k)\right]^2 \end{aligned} \tag{20}$$

where W is the kernel (weight function), h is the smoothing length, and M is the number of data points. Note that the known data values u_j are shifted by the amount given by u_k. This transformation gives $u^h(\mathbf{x}_k,\mathbf{x}_k) \equiv 0$ which is consistent with $p_0(\mathbf{x}) = \text{const} = 0$ (in Eq. (19) the constant term is implicitly zero).

The functional J expresses the squared error between the known field values u_j and the values produced by the approximation Eq. (18). Each term in the sum is scaled by the factor W, which is known as the *kernel*. The bi-cubic spline kernel developed by Monaghan for SPH simulations [10] is used

$$W(h,\mathbf{x}-\mathbf{x}^*) = W(h,r) = \begin{cases} \frac{40}{7\pi h^2}(1-3/2q^2+3/4q^3) & \text{for } 0\leqslant q<1 \\ \frac{40}{28\pi h^2}(2-q)^3 & \text{for } 1\leqslant q<2 \\ 0 & \text{for } 2\leqslant q \end{cases} \tag{21}$$

where

$$r = \sqrt{(x-x^*)^2+(y-y^*)^2} \tag{22}$$

$$q = \frac{2r}{h} \tag{23}$$

The *smoothing length* h denotes the radius of the circular support for W, and is a constant for all data points $\mathbf{x}_i$. In Fig. 3 a plot of the kernel is shown. The functionality of the kernel is to adjust the significance of each term in Eq. (20). Since W has compact support a point outside the smoothing length will not contribute in Eq. (20). Because W has zero slope on its boundary a data point can enter or leave the support smoothly. Finally, points that are close to $\mathbf{x}_k$ have a larger weight than points farther away.

The coefficients $\mathbf{a}$ are found by minimizing the functional J. This is achieved by satisfying the condition of stationarity

$$\begin{aligned} &\frac{\partial J}{\partial a_m} = 0 \\ &\sum_{j=1}^{M} 2W(h,\mathbf{x}_j-\mathbf{x}_k)\left[\sum_{i=1}^{N} p_i(\mathbf{x}_j-\mathbf{x}_k)a_i(\mathbf{x}_k) - (u_j-u_k)\right] \\ &p_m(\mathbf{x}_j-\mathbf{x}_k) = 0 \end{aligned} \tag{24}$$

for the $m = 1,2,\ldots,5$ basis functions. Eq. (24) can be rearranged into the matrix form

$$\mathbf{Ma} = \mathbf{b} \tag{25}$$

where, using the notation $W^j = W(h,\mathbf{x}_j-\mathbf{x}_k)$ and $p_i^j = p_i(\mathbf{x}_j-\mathbf{x}_k)$

$$\mathbf{M} = \begin{bmatrix} \sum W^j p_1^j p_1^j & \sum W^j p_2^j p_1^j & \sum W^j p_3^j p_1^j & \cdots \\ \sum W^j p_1^j p_2^j & \sum W^j p_2^j p_2^j & \sum W^j p_3^j p_2^j & \cdots \\ \sum W^j p_1^j p_3^j & \sum W^j p_2^j p_3^j & \sum W^j p_3^j p_3^j & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{26}$$

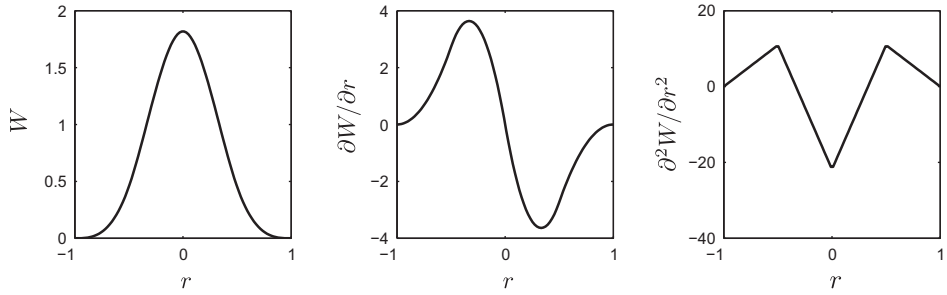

Fig. 3. The bi-cubic kernel Eq. (21) as a function of x, for fixed values of the smoothing length $h = 1$ and $y = 0$. The expansion point $\mathbf{x}^*$ is set to $(0, 0)$.

$$\mathbf{a} = (\, a_1 \quad a_2 \quad a_3 \quad \ldots)^{\mathrm{T}} \tag{27}$$

$$\mathbf{b} = \left(\sum W^j p_1^j (u_j - u_k) \quad \sum W^j p_2^j (u_j - u_k) \quad \ldots \right)^{\mathrm{T}} \tag{28}$$

where the summation index is $j = 1, 2, \ldots, M$, with M being the number of data points. The matrix $\mathbf{M}$ is called the *matrix of moments* and has size $N \times N$, with N being the number of basis functions (here $N = 5$).

Once the coefficients have been calculated approximations of the spatial derivatives can be obtained by differentiating Eq. (18). Taking the derivative with respect to x gives

$$\frac{\partial u^h(\mathbf{x}, \mathbf{x}_k)}{\partial x} = \sum_{i=1}^{N} \frac{\partial p_i(\mathbf{x} - \mathbf{x}_k)}{\partial x} a_i(\mathbf{x}_k) \tag{29}$$

Substituting the expressions for the basis functions Eq. (19) gives

$$\frac{\partial u^h}{\partial x} = a_1 + 2(x - x_k)a_3 + (y - y_k)a_4 \tag{30}$$

Finally, evaluating at $\mathbf{x}_k$ gives

$$\frac{\partial u^h}{\partial x} = a_1 \tag{31}$$

Higher order derivatives and partial derivatives with respect to y are found similarly. Below is listed the relevant derivatives needed for the discretization of Eq. (5)

$$\begin{aligned}
&\frac{\partial u}{\partial x} = \mathbf{e}_1 \mathbf{M}^{-1} \mathbf{b}^u \quad \frac{\partial u}{\partial y} = \mathbf{e}_2 \mathbf{M}^{-1} \mathbf{b}^u \\
&\frac{\partial^2 u}{\partial x^2} = 2\mathbf{e}_3 \mathbf{M}^{-1} \mathbf{b}^u \quad \frac{\partial^2 u}{\partial x \partial y} = \mathbf{e}_4 \mathbf{M}^{-1} \mathbf{b}^u \quad \frac{\partial^2 u}{\partial y^2} = 2\mathbf{e}_5 \mathbf{M}^{-1} \mathbf{b}^u \\
&\frac{\partial v}{\partial x} = \mathbf{e}_1 \mathbf{M}^{-1} \mathbf{b}^v \quad \frac{\partial v}{\partial y} = \mathbf{e}_2 \mathbf{M}^{-1} \mathbf{b}^v \\
&\frac{\partial^2 v}{\partial x^2} = 2\mathbf{e}_3 \mathbf{M}^{-1} \mathbf{b}^v \quad \frac{\partial^2 v}{\partial x \partial y} = \mathbf{e}_4 \mathbf{M}^{-1} \mathbf{b}^v \quad \frac{\partial^2 v}{\partial y^2} = 2\mathbf{e}_5 \mathbf{M}^{-1} \mathbf{b}^v \\
&\frac{\partial \rho}{\partial x} = \mathbf{e}_1 \mathbf{M}^{-1} \mathbf{b}^\rho \quad \frac{\partial \rho}{\partial y} = \mathbf{e}_2 \mathbf{M}^{-1} \mathbf{b}^\rho
\end{aligned} \tag{32}$$

where $\mathbf{e}_j$ denotes the jth standard basis vector and $\mathbf{M}$ is defined as in Eq. (26). Note that $\mathbf{M}^{-1}$ is common for all derivatives at a specific node, so only a single matrix factorization is needed. Since $\mathbf{M}$ is symmetric and positive definite the Cholesky factorization can be used. The right hand sides $\mathbf{b}^u$, $\mathbf{b}^v$, and $\mathbf{b}^\rho$ are defined as in Eq. (28) based on the field variables u, v, and ρ respectively.

4.2. Neighbour search

The objective of the neighbor search is to determine which nodes interact with each other, i.e. are within smoothing length. This way summation over all nodes in Eq. (25) which is an $O(N^2)$ operation can be avoided. The present work utilizes a cell list [11] to ensure an $O(N)$ algorithm when searching for neighboring particles.

4.3. Curvature of the free surface

The free surface normal and curvature are calculated by fitting a curve through neighboring nodes on the free surface, see Fig. 4. The curve is assumed parabolic in x and y and parametrized by τ, which denotes the arc length of a piece-wise linear curve along the nodes on the free surface. For example the x-coordinate is defined by

$$\begin{aligned}
x_{i-1} &= c_0 + c_1 \tau_W + c_2 \tau_W^2 \\
x_i &= c_0 \\
x_{i+1} &= c_0 + c_1 \tau_E + c_3 \tau_E^2
\end{aligned} \tag{33}$$

where $\tau_W = \tau_{i-1} - \tau_i$ and $\tau_E = \tau_{i+1} - \tau_i$ (i enumerating the nodes along the free surface). Similar expressions are used for the y-coordinate. Differentiating the resulting polynomials enables the computation of the surface normals and curvatures as needed in Eqs. (16) and (17).

4.4. Remeshing

The term remeshing is used here to mean the redistribution of computational nodes. After some time a new set of computational nodes are generated. These nodes are positioned evenly in the current domain, and then the field values at these positions are found by interpolation from the current state variables. Because of the Lagrangian framework nodes are convected with the flow, which will potentially lead to clustering or spreading of nodes in different parts of the domain. This effect can be undesirable, however, applying the remeshing procedure at certain intervals resolves the problem. Another reason for introducing remeshing is the fact that MLS is a centered scheme, which can suffer from uncontrolled modes that grow slowly during the simulation. The interpolation of state variables has a beneficial effect on stability because it also acts as a filter.

The position of the new computational nodes is found from the current set by connecting neighboring nodes with linear springs. Particles i and j interact with each other by a force given by

$$F_{ij} = K(L_0 - L) \tag{34}$$

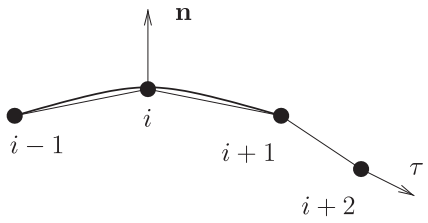

Fig. 4. Curve fit through points on the free surface.

where K is the spring stiffness, L_0 is the undeformed spring length, and L is the distance between the nodes. A cut-off distance is introduced in a way similar to the smoothing length in MLS. Thus particles farther away than $1.1L_0$ are not connected by springs. The value of L_0 is chosen to match an average spacing of the particles, given the number of particles and the surface area of the domain.

The position of the new particles is found by considering one particle at a time while all other particles remain fixed. A position of the particle where all spring forces balance each other is found by Newton iteration. After one global iteration the position of all new particles (except those of free surfaces) have been updated. The process ends when the maximum change in particle position from on global iteration to the next is smaller than a specified tolerance. See [12].

Particles on the free surface might also spread or cluster (however, by continuity a particle on the free surface remains on the free surface). If this happens remeshing of the free surface is also needed. This case is more simple since for a 2D problem the free surface is 1D and the particles can be positioned explicitly in an even fashion.

4.5. Time stepping

Each nodes carries of following quantities: x, y, u, v, and ρ. In order to advance the solution from time level $k-1$ to k the following steps are taken. For interior nodes Eq. (5)–(7) are time integrated using an explicit third order Runge–Kutta scheme [13].

Nodes on solid walls are updated in the following way: Position and velocity of nodes on a solid are computed from the prescribed motion of the solid body. Density (or conversely pressure) is computed from Eq. (12) and (13). Spatial derivatives are computed as described in Section 4, but nodes on the solid body do not couple with each other. In Eq. (12) the time derivatives are approximated as follows

$$\frac{\mathrm{d}u}{\mathrm{d}t} \approx \frac{u^k - u^{k-1}}{t^k - t^{k-1}} \quad \frac{\mathrm{d}v}{\mathrm{d}t} \approx \frac{v^k - v^{k-1}}{t^k - t^{k-1}} \tag{35}$$

where k denotes the time level. The very first time step is handled by assuming that the initial condition at time $t = 0$ is also valid for $t < 0$.

Nodes on the free surface are updated as follows: The position of a free surface node is updated using the Runge–Kutta method already mentioned, i.e. time-integration of Eq. (6). The velocity components (and density) are computed at any sub-step of the Runge–Kutta method by solving Eq. (12), (13), (16), and (17) for u, v, and ρ.

Summarizing the Runge–Kutta method is used to update position, velocity and density of interior nodes and position of free surface nodes. Sets of equations are used to update velocity and density of free surface nodes and density of solid body nodes. The time step is limited by the CFL-condition [14,15] and the Fourier-limit [16], that is

$$\Delta t = f \cdot \mathrm{MIN}\left(\frac{h}{a + \sqrt{u^2 + v^2}}, \frac{\rho h^2}{4\mu} \right) \tag{36}$$

where $f < 1$ is a safety factor. Eq. (36) should hold for all particles individually and is used as a guide when selecting the time step; which is then fixed throughout the simulation.

5. Results

5.1. Lid driven cavity

The lid driven cavity problem is a standard benchmark in computational fluid mechanics. The simulation domain consists of a square region confined by solid walls. The walls are stationary, with exception of the top wall which is moving at a constant horizontal speed. As a consequence a singularity develops at the top left and top right corner, where the x-component of velocity is discontinuous.

The Mach number is defined by the ratio between the maximum fluid speed and the speed of sound. Selecting a speed of sound (Eq. (9)) of 350 m/s and lid speed 1 m/s gives $\mathrm{Ma} = 0.0029$. The Reynolds number is defined by the ratio between the inertial forces and viscous forces. By appropriate choice of cavity dimensions and fluid properties the Reynolds number becomes $\mathrm{Re} = 1$. With the Ma significantly smaller than 0.3 and Re equal to 1 the problem can be regarded as incompressible and in the creeping flow limit. Therefore it is possible to compare the present results with a high resolution second order central difference solution of the problem based on a incompressible, creeping flow formulation.

The remeshing procedure is engaged after every 10 time steps. In order to facilitate the comparison of results the nodes are remeshed to a regular grid layout. The number of nodes along each dimension is denoted by N. In Fig. 5 (left) the steady-state solution is shown for $N = 31$.

Fig. 5 (right) shows the x-component of the velocity along the line $x = L/2$, together with a solution obtained from an incompressible creeping flow model. In both cases $N = 201$. It is seen that the two models give very similar results as expected. Table 1 shows the simulation results for various discretizations $\Delta x = L/(N-1)$. In order to establish the spatial convergence rate the error in u at the center node is measured. The exact value u_{ref} is estimated using the incompressible creeping flow model with $N = 301$. The error is then defined by

$$E = \left| u^N_{\mathrm{center}} - u_{\mathrm{ref}} \right| \tag{37}$$

which is evaluated for $N = 11, 31, 51, 71, 91$. Another error measure is obtained by integrating u from $y = 0$ to $y = L$ along the center line $x = L/2$. At steady state this integral I_u should be zero in order to respect the continuity of flow. In Fig. 6 I_u and E are plotted against the spatial resolution Δx. Computing the slope of a line through the linear data gives the following results $\alpha_I = 1.0001$ and $\alpha_u = 0.9204$. These figures indicate that the spatial discretization has first order convergence. This result could be anticipated since the basis is second order complete, but the nodes are unevenly distributed (except right after remeshing).

In Fig. 7 (left) the maximum allowed time step Δt is plotted against the spatial resolution Δx. The step sizes were determined with two digits of precision by trial and error. It is seen that the step size depends linearly on the spatial resolution, corresponding to a CFL-like stability limit.

In Fig. 7 (right) the average time spent per time step is plotted against the number of nodes. A linear relationship between these quantities can be observed. This result is expected, since a fully explicit formulation is used. The cost per node is constant, although, different for interior nodes and nodes on the boundary. The neighbor search algorithm also has complexity $O(N)$, where N is the total number of nodes.

5.2. Rotating free surface

The simulation problem consists of a circular shaft with radius $r_1 = 0.005$, rotating at a constant speed, and fully immersed in fluid. The fluid domain is shaped as an annulus with boundaries consisting of the shaft and a free surface. The length scale of the problem is the radius $r_2 = 0.012$ at the free surface and the velocity scale is defined by $U = r_2\omega$, with $\omega = 100$ being the rotational speed of the shaft. The surface tension is set to $\sigma = 0$. The model

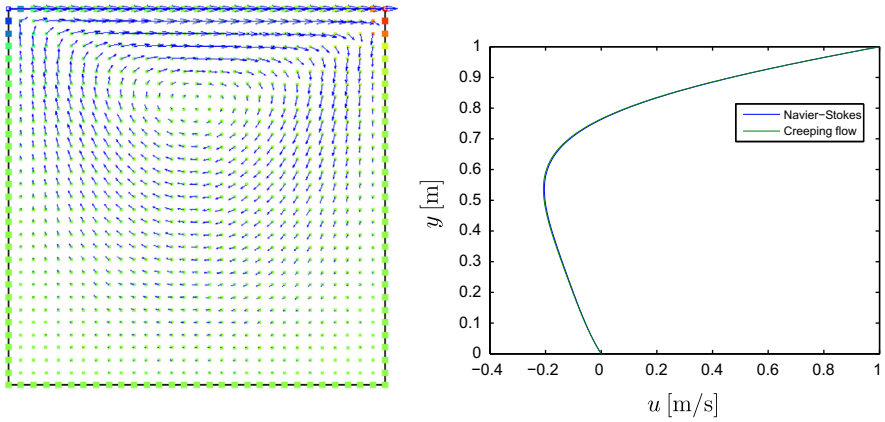

Fig. 5. Left: Steady state configuration of the lid driven cavity. Red color means high density, while blue color denotes low density. The velocity vectors are indicated by arrows. Right: Solutions of the lid driven cavity problem obtained from (1) the Navier–Stokes equations and (2) an incompressible creeping flow model. The number of discretization points along each axis is $N = 201$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Simulation results for the lid driven cavity problem. N is the number of discretization points in each coordinate direction. Δx and Δt denote the discretization size of space and time, respectively. I_u is the integral of u from $y = 0$ to $y = L$ at $x = L/2$. E as defined in Eq. (37). t_{elapsed} is the CPU time spent for each simulation.

N	Δx (m)	Δt (s)	I_u (m^2/s)	E (m/s)	t_{elapsed} (s)
11	1.000×10^{-1}	6.8×10^{-4}	5.29×10^{-2}	5.45×10^{-2}	1.937
31	3.333×10^{-2}	2.4×10^{-4}	1.87×10^{-2}	2.20×10^{-2}	23.969
51	2.000×10^{-2}	1.5×10^{-4}	1.11×10^{-2}	1.35×10^{-2}	94.266
71	1.143×10^{-2}	1.1×10^{-4}	7.68×10^{-3}	9.56×10^{-3}	244.047
91	1.111×10^{-2}	9.1×10^{-5}	6.07×10^{-3}	7.38×10^{-3}	472.500
201	5.000×10^{-3}	2.1×10^{-5}	2.66×10^{-3}	3.33×10^{-3}	719.060

parameters are chosen such that: Re = 10, Ma = 0.0028. The remeshing algorithm is engaged after every 20 time steps.

This simulation problem is a test for the free surface boundary condition. After the initial transient the fluid should rotate as a rigid body with the same angular velocity as the shaft, since the shear stress on the free surface is zero. This property can be tested by comparing the speed at a given radius with the exact value for a rigid body motion.

Fig. 8 shows the speed of the particles as a function of radius at the final time step of the simulation. For comparison the same value for a rotating rigid disc (given by $U = R\omega$) is also shown. Good agreement is observed, which indicates that there is in fact no shear stress between in free surface and the surrounding void.

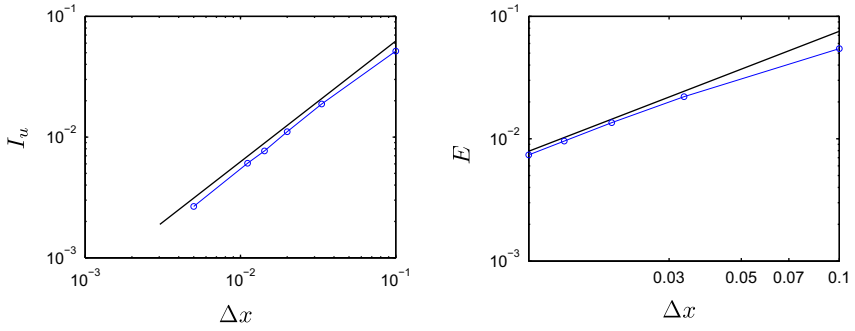

Fig. 6. Convergence rate of the lid driven cavity problem. Lines without marker have slope 1. Left: I_u plotted against Δx, Right: E as defined in Eq. (37) plotted against Δx.

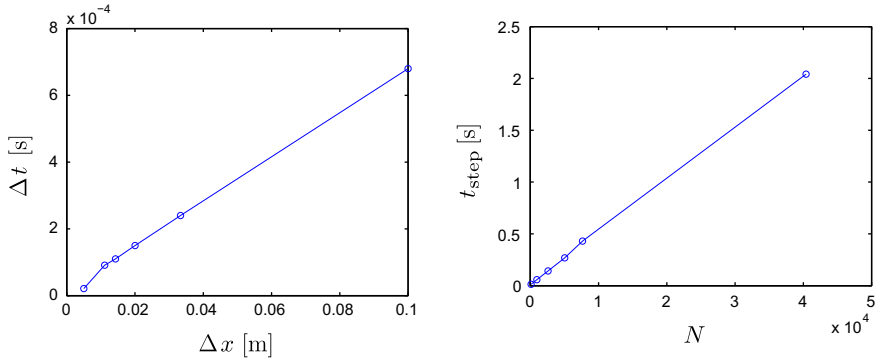

Fig. 7. Left: The maximum allowed step size Δt as a function of the spatial resolution Δx. Right: Average time spent per time step (wall clock) as a function of the total number of nodes.

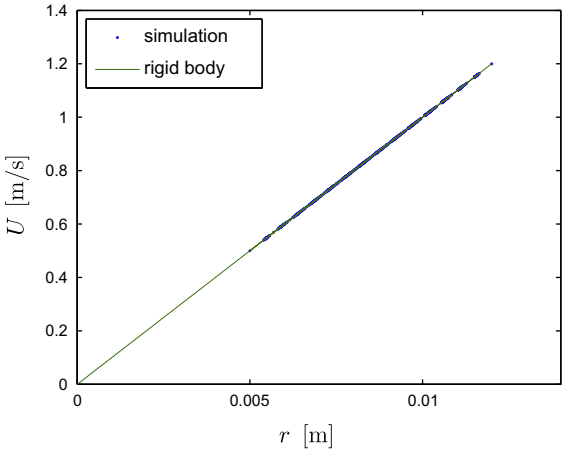

Fig. 8. Comparison of particle speed with exact rigid body motion.

The transient evolution of the system is shown graphically in Fig. 9 for three moments in time. Initially the density is constant everywhere. After the transient density increases with increasing radius due to the centrifugal force.

In order the compare the transient solution with an exact solution a simplified model assuming incompressibility has been derived

$$\frac{\rho}{\mu}\frac{\partial \omega}{\partial t} = \frac{1}{r^3}\frac{\partial}{\partial r}\left(r^3\frac{\partial \omega}{\partial r}\right) \tag{38}$$

The solution can be expressed as an infinite set of eigenvalues and eigenvectors. See Appendix A for details.

For the particular problem considered here the first eigenvalues and values integration constants a and b are given in Table 2. The solution $R(r)$ is plotted for the first 3 eigenvectors in Fig. 10. Clearly, it would require infinitely many eigenvectors and values to represent the initial condition given in Eq. (A.8). As an alternative the MLS solution is compared with the transient analytical solution when the initial condition is given by one eigenvector. The exact solution is then a single term, namely, the product of Eqs. (A.12) and (A.11) with constants from one row of Table 2 substituted in. Graphical results are shown in Fig. 11. The figure shows that the first eigenmode is well resolved, while the third eigenmode does not follow the temporal evolution exactly.

All in all, in the special case of incompressible fluid, one can separate variables to obtain an analytical solution. The MLS simulation agree well with the analytical solution in this case.

5.3. Oscillating droplet

The oscillating droplet problem is concerned with simulation of small vibrations in a drop of fluid. The initial configuration is given as a circular shape which has been perturbed into an ellipse. When the simulation begins the drop will deform due to the action of the surface tension which results in an oscillatory motion. The initial geometry is defined by the semi-axes $L_x = 1.00$ and $L_y = 1.005$, and the flow properties are given by the density $\rho_0 = 900$, dynamic viscosity $\mu_0 = 10^{-6}$, and surface tension $\sigma_0 = 1$.

Table 2
Eigenvalues and vectors of Eq. (A.10).

Order	κ	a	b
1	123.0748818	−1	−4.204658615
2	655.0174015	−1	1.662983702
3	1111.447665	−1	0.1267057127

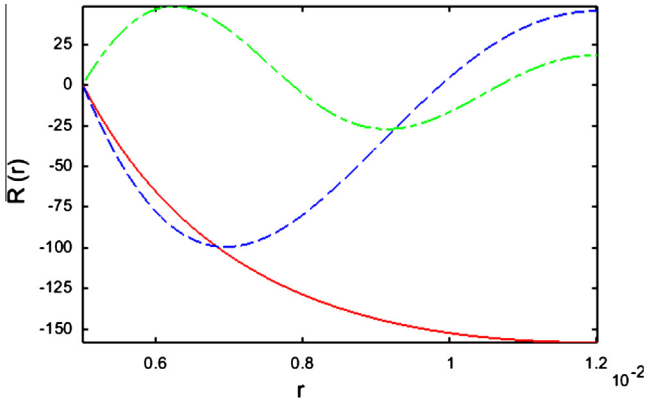

Fig. 10. Eigenvectors of Eq. A.10). Full line: 1st eigenvector, Dash-dash: 2nd eigenvector, Dash-dot: 3rd eigenvector.

The motion of liquid drops in general is very complex. However, analytical results are available for special limiting cases. Lord Rayleigh [17] has derived an expression for the frequency of oscillation of two-dimensional drops that are perturbed slightly from a circular shape and that have zero viscosity

$$\omega_n^2 = \frac{\sigma_0}{\rho_0 R^3} n(n^2-1) \tag{39}$$

In the formula $n = 1$ corresponds to rigid body motion and $n = 2$ corresponds to the first mode of oscillation.

In Fig. 12 three snapshots from the simulation are shown. As can be seen the fluid is initially at rest, except for points on the boundary. Imposition of the boundary conditions on the free surface results in non-zero velocity components due to the surface tension. Because the fluid is compressible the fluid motion due to surface tension is superposed with a (small) net contraction of the drop. The simulation is performed with a fixed time step, and the remeshing algorithm is engaged after every 100 time steps. Instead of eliminating viscosity completely, a small value of $\mu = 10^{-6}$ Pa s is used. This is necessary because the free surface boundary condition Eq. (17) is not valid for $\mu = 0$.

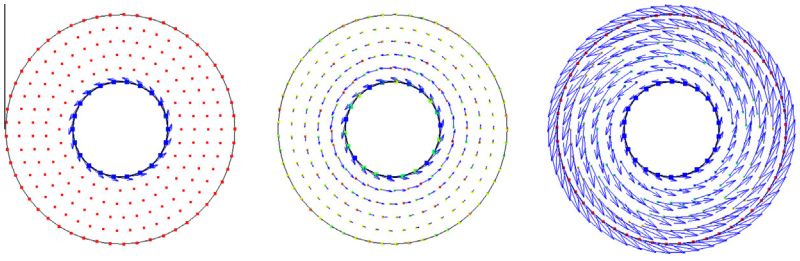

Fig. 9. Evolution of the rotating free surface test problem. Note that the color scale is adjusted to the instantaneous data range in each time step. Left: $t = 0$ s, Middle: $t = 7.8 \times 10^{-5}$ s, Right: $t = 5 \times 10^{-2}$ s (final time).

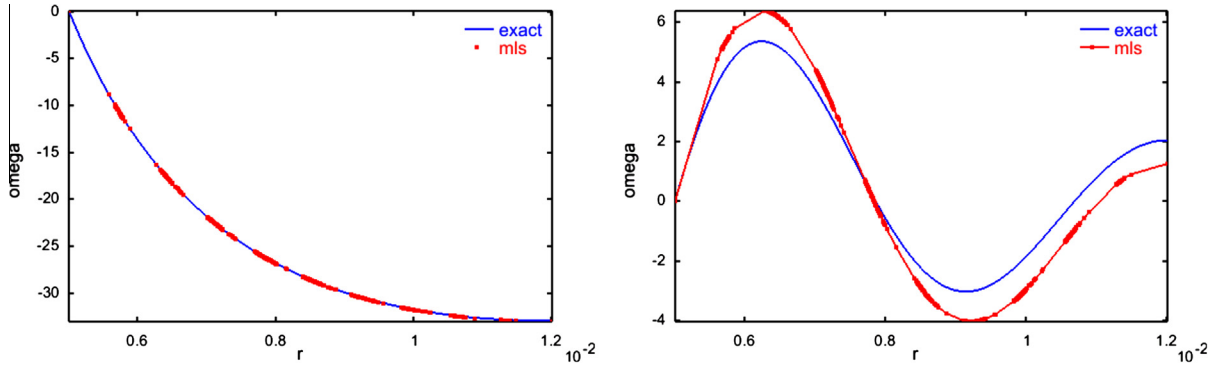

Fig. 11. Left: 1st eigenvector solution at time $t = 8e-4$. Right: 3rd eigenvector solution at time $t = 1.6e-4$.

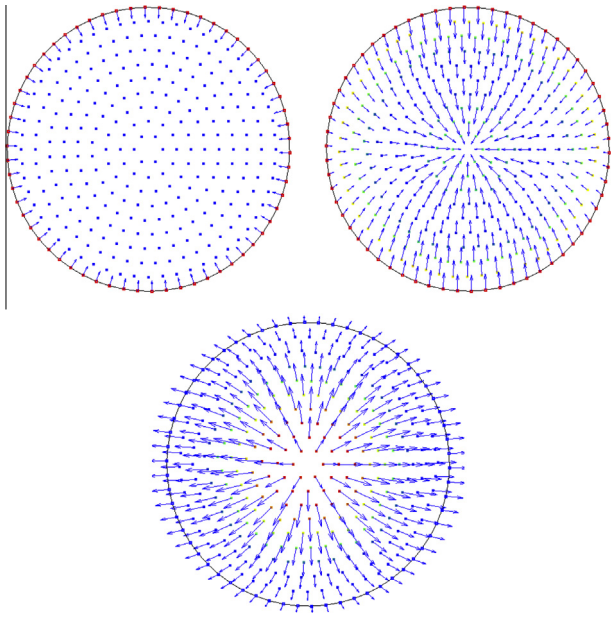

Fig. 12. The oscillating drop test problem. Top left: $t = 0$ s, top right: $t = 2.75 \times 10^{-2}$ s, bottom $t = 3.05 \times 10^{-2}$ s.

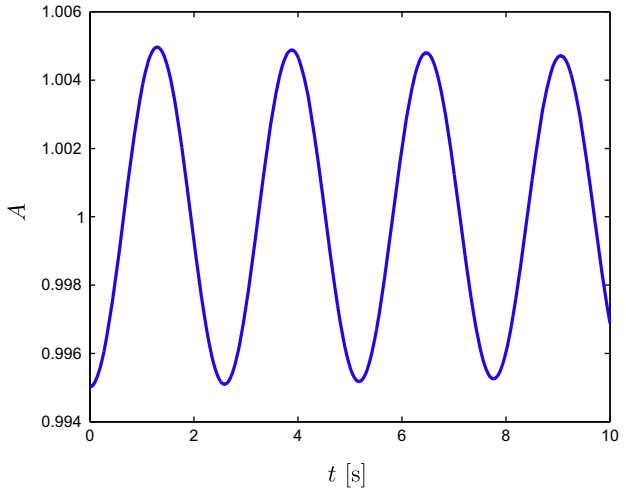

Fig. 13. The ratio between the main axis of the oscillating drop plotted against time.

Table 3
Dependency of frequency and damping on remeshing rate. R is remeshing rate, ω the oscillation angular frequency, ΔA drop in amplitude. Physical viscosity $\mu = 1e-6$.

R	ω	ΔA
12	2.43534	8.03462e−4
25	2.43220	5.16400e−4
50	2.42984	3.50514e−4
100	2.42985	2.52400e−4

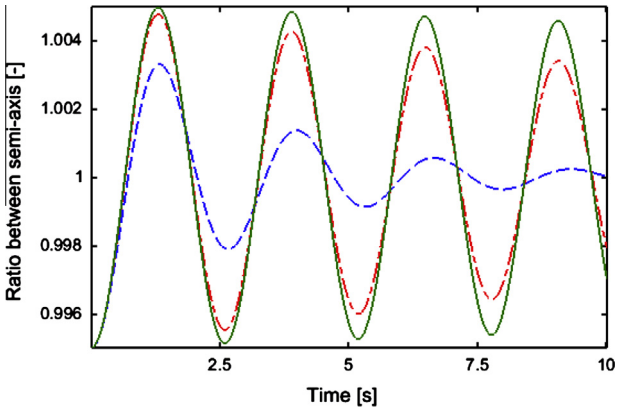

Fig. 14. Droplet oscillations. Full line: $\mu = 1e-3$, Dash-dot: $\mu = 1e-2$, Dash-dash: $\mu = 1e-1$.

Table 4
Dependency of frequency and damping on physical viscosity. μ is viscosity, ω the oscillation angular frequency, ΔA drop in amplitude. Remesh rate is 100.

μ	ω	ΔA
1e−1	2.36136	3.05537e−3
1e−2	2.42828	1.35787e−3
1e−3	2.42985	3.80574e−4
1e−4	2.42985	2.65101e−4
1e−5	2.42985	2.53553e−4
1e−6	2.42985	2.52399e−4
1e−7	2.42985	2.52284e−4

The drop motion is monitored by the ratio of the main axes of the ellipsoid throughout the simulation

$$A = \frac{L_x}{L_y} \tag{40}$$

The transient response is shown in Fig. 13. Extracting the frequency gives $\omega_{\text{num}} = 2.42985$, which can be compared with the analytical value for the 1st mode of oscillation $\omega_{\text{exact}} = \sqrt{6} = 2.44949$. The

agreement is seen to be very good, the relative deviation being less than 1%.

In order to test the dependency of the frequency on remeshing rate the simulation is repeated using different remeshing rates. The results are given in Table 3. Damping is quantified by measuring the change in amplitude between the first positive peak and the fourth positive peak, see Fig. 14. It can be seen that remeshing more often increases the damping of the oscillation. This means that remeshing adds false diffusion to the numerical scheme. Similarly the oscillation frequency grows as remeshing is carried out more often. This result is less intuitive and not investigated further in this paper.

In Table 4 the oscillation frequency and damping is given as a function of physical viscosity. It can be seen that once physical viscosity is 1e−3 or smaller then the oscillation frequency remains constant within the first 6 digits. Similarly the damping tends to a fixed value as viscosity is decreased. Keeping Table 3 in mind the figures suggest that a remeshing rate of 100 becomes the dominant factor once physical viscosities are smaller than 1e−3 or 1e−4.

6. Conclusion

A moving least squares method for the discretization of the compressible Navier–Stokes equations has been presented. The method has been applied for confined as well as free surface flows including surface tension. The benchmark cases demonstrate a first order spatial convergence and it has been shown that the time required for simulation depends linearly on the number of nodes in the model. A detailed description of the boundary conditions for a free surface has been given.

Three test cases have been considered, all of them being solved with good accuracy. Furthermore, an analytical solution of the rotating free surface problem was found.

The proposed method has the following advantages: simple translation from partial differential equation to algebraic system, good scalability with respect to number of nodes, explicit formulation with limited coupling throughout the domain (parallelizing of the code should not be hard).

Appendix A. Derivation and solution of rotating disc problem

Considering the friction forces between concentric tubes the following equations apply

$$\begin{aligned} m &= \rho 2\pi r \mathrm{d}r \\ I &= mr^2 \\ \tau &= \mu r \frac{\mathrm{d}\omega}{\mathrm{d}r} \\ A &= 2\pi r \\ F &= A\tau \\ M &= Fr \\ I\frac{\mathrm{d}\omega}{\mathrm{d}t} &= \Sigma M \end{aligned} \tag{A.1}$$

where m is the mass of a tube at radius r with wall thickness $\mathrm{d}r$, I is the moment of inertia, τ is the traction between tubes, A is the surface area between tubes (assuming tube length equal to 1), F is the force generated at the interface between tubes, M is the torque between tubes, and ω is the angular velocity (rotation speed) of a tube. Rearranging gives

$$\frac{\rho}{\mu}\frac{\partial\omega}{\partial t} = \frac{1}{r^3}\frac{\partial}{\partial r}\left(r^3\frac{\partial\omega}{\partial r}\right) \tag{A.2}$$

with the boundary conditions

$$\omega(r_1,t) = \omega_{\text{shaft}} \quad \text{for } t > 0 \tag{A.3}$$

$$\frac{\partial\omega(r_2,t)}{\partial r} = 0 \quad \text{for } t > 0 \tag{A.4}$$

and initial condition

$$\omega(r,0) = 0 \text{ for } r_1 < r \leqslant r_2 \tag{A.5}$$

Clearly, $\omega(r,t) = \omega_{\text{shaft}}$ is a constant solution. Writing a solution as the sum of the constant solution and a transient, $\omega(r,t) = \omega_{\text{shaft}} + \omega_T(r,t)$, and inserting into Eq. (A.2), imply that the transient must satisfy,

$$\frac{\rho}{\mu}\frac{\partial\omega_T}{\partial t} = \frac{1}{r^3}\frac{\partial}{\partial r}\left(r^3\frac{\partial\omega_T}{\partial r}\right) \tag{A.6}$$

the boundary conditions

$$\omega_T(r_1,t) = \frac{\partial\omega_T(r_2,t)}{\partial r} = 0 \text{ for } t > 0, \tag{A.7}$$

and the initial condition

$$\omega_T(r,0) = 0 \text{ for } r_1 < r \leqslant r_2 \tag{A.8}$$

Following the lines of [18], Eqs. (A.6), (A.7) and (A.8) can be solved by separation of variables. Letting $\omega_T(r,t) = R(r)T(t)$, leads to

$$\frac{1}{T(t)}\frac{\rho}{\mu}\frac{\mathrm{d}T(t)}{\mathrm{d}t} = D \tag{A.9}$$

$$\frac{1}{R(r)}\frac{1}{r^3}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^3\frac{\mathrm{d}R}{\mathrm{d}r}\right) = D \tag{A.10}$$

where D is the separation constant. Since it is known that the transient tends to zero, when t tends to infinity, we must have $D < 0$, so we can set $D = -\kappa^2$, with $\kappa > 0$. Then the solutions are

$$T(t) = c\exp\left(-\frac{\mu\kappa^2 t}{\rho}\right) \tag{A.11}$$

$$R(r) = \frac{a\mathrm{J}_1(\kappa r)}{r} + \frac{b\mathrm{Y}_1(\kappa r)}{r} \tag{A.12}$$

Eq. (A.10) is a eigenvalue problem, $D = -\kappa^2$ is an eigenvalue, and an eigenvector $R(r)$ is only determined up to a multiplicative constant. Hence we can arbitrarily fix the integration constant $a = -1$, and solve for κ and b.

References

[1] Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. J Comput Phys 1981;39:201–25.
[2] Grooss J, Hesthaven JS. A level set discontinuous Galerkin method for free surface flows. Comput Methods Appl Mech Eng 2005;195:3406–29. 2006.
[3] Shao S, Lo EYM. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. Adv Water Resour 2003;26:787–800.
[4] Liu GR. Meshfree methods: moving beyond the finite element method. CRC Press; 2009.
[5] Liu GR, Liu MB. Smoothed particle hydrodynamics, a meshfree particle method. World Scientific; 2003. p. 105–24.
[6] Belytschko T, Krongauz Y, Organ D, Flemming M, Krysl P. Meshless methods: an overview and recent developments. Comput Method Appl Mech Eng 1996;139(1–4):3–47.
[7] Liu GR. Mesh free methods. Moving beyond the finite element method. CRC Press; 2003. p. 77–8.
[8] Monaghan JJ. Simulating free surface flows with SPH. J Comput Phys 1994;110:399–406.
[9] Armando Duarte C. A review of some meshless methods to solve partial differential equations. TICAM Report 95-06; 1995.
[10] Monaghan JJ. SPH without a tensile instability. J Comput Phys 2000;159:290–311.
[11] Hockney RW, Eastwood JW. Computer simulation using particles. New York: MCGraw-Hill; 1981.
[12] Reboux S, Schrader B, Sbalzarini IF. A self-organizing lagrangian particle method for adaptive-resolution advection-diffusion simulations. J Comput Phys 2012;231:3623–46.
[13] Iserles Arieh. A first course in the numerical analysis of differential equations. Cambridge University Press; 2003. p. 40.

[14] Courant R, Friedrichs K, Lewy H. On the partial difference equations of mathematical physics. IBM J Res Develop 1928;11(2):215–34.
[15] Anderson Lohn David. Computational fluid dynamics: the basics with applications. McGraw-Hill Inc.; 1995.
[16] Incropera Frank P, DeWitt David P. Fundamentals of heat and mass transfer. 5th ed. Wiley; 2002.
[17] Rayleigh Lord. On the capillary phenomena of jets. Proc Roy Soc Lond 1879;29:71–97.
[18] Mendiburu AA, Carrocci LR, Carvalho JA. Analytical solution for transient one-dimensional Couette flow considering constant and time-dependent pressure gradients. Therm Eng 2009;8(2):92–8.