A Meshfree Method For Incompressible Fluid Flows with Incorporated Surface Tension

Sudarshan Tiwari — Jörg Kuhnert

Fraunhofer Institut Techno- und Wirtschaftmathematik
Gottlieb-Daimler-Strasse
Gebäude 49
D-67663 Kaiserslautern, Germany
tiwari@itwm.fhg.de, kuhnert@itwm.fhg.de

ABSTRACT. *A meshfree particle method is used to simulate free surface flows. This is a Lagrangian method. Flows are modeled by the incompressible Navier-Stokes equations. The particle projection method is used to solve the Navier-Stokes equations. The spatial derivatives are approximated by the weighted least squares method (WLS). The pressure Poisson equation is solved by a local iterative procedure with the help of WLS. Numerical experiments are presented for two dimensional cases. In the case of breaking dam problem the numerical result is compared with the experimental result. The surface tension effects are studied in different shapes of drops and Laplace's law is verified. Finally, the collisions of two drops are simulated.*

RÉSUMÉ. *Une méthode sans maillage est utilisée pour simuler les écoulements à surface libre. Il s'agit d'une méthode Lagrangienne. Les écoulements sont modélisés par les équations incompressibles de Navier-Stokes. La méthode de projection des particules est utilisée pour résoudre les équations de Navier-Stokes. Les dérivées spatiales sont approchées par la méthode des moindres carrés pondérés (WLS). L'équation de pression de Poisson est résolue par une procédure itérative locale à l'aide de WLS. Les expériences numériques sont présentées pour deux cas 2D. Dans le cas du problème de rupture de barrage, les résultats numériques sont comparés avec l'expérience. Les effets de tension surfacique sont étudiés pour différentes formes de gouttes et la loi de Laplace est vérifiée. Finalement, les collisions entre deux gouttes sont simulées.*

KEYWORDS: *Meshfree method, incompressible Navier-Stokes equations, projection method, free surface flow, least squares (LSQ) approximation.*

MOTS-CLÉS : *Méthode sans maillage, équations incompressibles de Navier-Stokes, méthode de projection, écoulement à surface libre, approximation par moindres carrés.*

REEF – 11/2002. Meshfree Computational Mechanics, pages 965 to 987

1. Introduction

In this paper we present a meshfree particle method for simulations of free surface flows. This is a Lagrangian method. A fluid domain is first replaced by a discrete number of points, which are referred to as particles. Each particle carries all fluid information, like density, velocity, temperature etc. and moves with fluid velocity. Therefore, particles themselves can be considered as geometrical grids of the fluid domain. This method has some advantages over grid based techniques, for example, it can handle fluid domains, which change naturally, whereas grid based techniques require additional computational effort.

Numerical simulations of free surface flows have many industrial applications like casting, tank filling and others. Many methods have been developed to simulate free surface flows (Hansbo 1992, Harlow *et al*. 1965, Hirt *et al*. 1981, Kelecy *et al.* 1997, Kothe *et al.* 1992, Maronnier *et al.* 1999, Tiwari *et al.* 2000). A classical grid free Lagrangian method is Smoothed Particle Hydrodynamics (SPH), which was originally introduced to solve problems in astrophysics (Lucy 1977, Gingold *et al.* 1977). It has since been extended to simulate the compressible Euler equations in fluid dynamics and applied to a wide range of problems, see (Monaghan 92, Monaghan *et al.* 1983, Morris *et al.* 1997). The method has also been extended to simulate inviscid incompressible free surface flows (Monaghan 94). The implementation of the boundary conditions is the main problem of the SPH method.

Another approach for solving fluid dynamic equations in a grid free framework is the moving least squares or least squares method (Belytschko *et al.* 1996, Dilts 1996, Kuhnert 99, Kuhnert 2000, Tiwari *et al.* 2001 and 2000). With this approach boundary conditions can be implemented in a natural way just by placing the particles on boundaries and prescribing boundary conditions on them (Kuhnert 99). The robustness of this method is shown by the simulation results in the field of airbag deployment in car industry. Here, the membrane (or boundary) of the airbag changes very rapidly in time and takes a quite complicated shape (Kuhnert *et al.* 2000).

In (Tiwari *et al.* 2000) we have performed simulations of incompressible flows as the limit of the compressible Navier-Stokes equations with some stiff equation of state. This approach was first used in (Monaghan 92) to simulate incompressible free surface flows by SPH. The incompressible limit is obtained by choosing a very large speed of sound in the equation of state such that the Mach number becomes small. However the large value of the speed of sound restricts the time step to be very small due to the CFL-condition.

The projection method of Chorin (Chorin 68) is a widely used approach to solve problems goverbed by the incompressible Navier-Stokes equation in a grid based structure. In (Tiwari *et al.* 2001), this method has been applied to a grid free framework with the help of the weighted least squares method. The scheme gives accurate results for the incompressible Navier-Stokes equations. The occurring

Poisson equation for the pressure field is solved by a grid free method. In (Tiwari *et al.* 2001), it has been shown that the Poisson equation can be solved accurately by this approach for any boundary conditions. The Poisson solver can be adopted to the weighted least squares approximation procedure with the condition that the Poisson equation and the boundary condition must be satisfied on each particle. This is a local iteration procedure.

In this paper, we further extend the scheme, presented in (Tiwari *et al.* 2001), to free surface flows. Numerical experiments are obtained with and without surface tension forces. The broken dam problem is solved without surface tension forces. The Laplace's law (Landau *et al.* 1959) has been tested for different shapes of bubbles. The numerical scheme, presented here, reproduces the Laplace's law exactly. Finally, the binary drop collision of liquid drops shows that the scheme is suitable for simulations of free surface flows.

The paper is organized as follows. In section 0 we present the mathematical model and boundary conditions. In section 0 the numerical scheme is described. In section 0, the weighted least squares method and its application to the Finite Pointset Method (FPM) is presented. The algorithm of determination of the free surface particles is presented in section 0. Finally, some numerical tests are presented in section 0.

2. Mathematical model and boundary conditions

We consider the incompressible Navier-Stokes equations in the Lagrangian form.

$$\frac{D\vec{v}}{Dt} = -\tfrac{1}{\rho}\nabla p + \nu\Delta\vec{v} + \vec{g}, \tag{1}$$

$$\nabla\cdot\vec{v} = 0. \tag{2}$$

Here, ρ is the mass density, $\vec{v}$ is the velocity vector, $\vec{g}$ is the body force acceleration vector, ν is the kinematic viscosity and p the dynamic pressure.

In addition to equations (1) and (2), appropriate initial and boundary conditions have to be provided.

For the discussion within this paper, we will consider various types of boundary conditions: solid wall, inflow, outflow, and free surface boundaries. However, we will emphasize on free surface boundaries, since these are indeed a delicate problem and have many applications in industry and sciences.

For a solid wall, one can use either free slip or no slip boundary conditions. If the viscosity is too low, a free slip condition seems to be appropriate, coupled with some

model for the boundary layer. For inflow boundaries, all velocity components need to be prescribed. The surface stress boundary condition on the interface between two fluids or free surfaces is given by (Landau *et al.* 1959) as

$$\left[\tau \cdot \vec{n} - p\vec{n}\right] = \sigma\kappa\vec{n}, \tag{3}$$

where

- σ is the surface tension of the fluid, which is assumed to be constant,
- κ is the curvature on the interface,
- $\vec{n}$ is the unit normal vector on the interface, and
- τ is the viscous stress tensor is given by $\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$.

The symbol $[.]$ denotes the jump across the free surface boundary between two fluids. Suppose that the viscosity of one fluid adjacent to the free surface is negligible and has the pressure p_0, then the normal and tangential components of the other fluid from equation (3)

$$p - \vec{n} \cdot \tau \cdot \vec{n} = p_0 + \sigma\kappa \tag{4}$$

$$\vec{t} \cdot \tau \cdot \vec{n} = 0. \tag{5}$$

Here, $\vec{t}$ denotes the unit tangent vector on the interface.

The implementation of boundary conditions (4) and (5) requires a sufficiently good approximation of the first and second spatial derivatives of the velocity as well as of the curvature of the free surface. Please find a detailed discussion about the numerical scheme in section 0. The approximation techniques employed are described in section 0.

3. Numerical scheme

The numerical idea we present here is a generalized Finite Difference Method for the Navier-Stokes-equations for incompressible problems with free surfaces. In fact, we call this method Finite Pointset Method (FPM) for the reasons we will find below.

The idea is to fill the flow domain with (numerical) points. These points are carriers of all relevant physical information (*i.e.* velocity, density, pressure etc.). In this context, we might call these points also particles, meaning they are not representing physical particles, rather they are representing a certain, finite piece of the fluid considered. The important point is, that, as time evolves, the particles are

moved with fluid velocity such that the numerical points (particles) move in the same fashion as a point of the fluid would move.

On the path of some numerical particle, the relevant quantities need to be updated, *i.e.* the velocity, pressure etc. will change. In order to describe these changes, we use the Navier-Stokes-equations as stated above and discretize them directly on each particle. Hence, no weak formulation is used. Since in the Navier-Stokes equations, spatial derivatives of the velocity and the pressure appear, we will have to find a way of giving good approximations for these terms based on the knowledge of the discrete pressure and velocity values. The method employed here is the so-called weighted least squares method. Please turn to section 0 for a detailed introduction.

Summarizing, the whole idea is to fill the flow domain with numerical points (particles), each of which being carrier of relevant physical information. We let the particles move with fluid velocity. The mean interaction radius between the particles for approximating derivatives is given by the symbol h. h is also referred to as smoothing length. As particles move on their path, physical quantities will have to be updated, governed by the Navier-Stokes-equations.

Now let us turn to the numerical method in detail. We consider the projection method described in (Chorin 68). This is an explicit method being of first order accuracy in time. It consists of two fractional steps. At the first step we explicitly compute the new particle positions and the intermediate velocity $\vec{v}^*$ by

$$\vec{x}^{n+1} = \vec{x}^n + \delta t \vec{v}^n \tag{6}$$

$$\vec{v}^* = \vec{v}^n + \delta t \nu \Delta \vec{v}^n + \delta t \vec{g}^n \tag{7}$$

Then, at the second step, we correct $\vec{v}^*$ by solving the equation

$$\vec{v}^{n+1} = \vec{v}^* - \delta t \nabla p^{n+1} \tag{8}$$

with the incompressibility constraint

$$\nabla \cdot \vec{v}^{n+1} = 0. \tag{9}$$

Here, for simplicity, we have considered ρ to be 1. By taking the divergence of equation (8) and by making use of (9), which is the constraint that $\vec{v}^{n+1}$ must be a divergence free vector field, we come up with the Poisson equation for the pressure

$$\Delta p^{n+1} = \frac{\nabla \cdot \vec{v}^*}{\delta t} \tag{10}$$

The boundary condition for solid walls as well as for inflow boundaries is obtained by projecting equation (8) on the outward unit normal vector $\vec{n}$ to the boundary Γ. Thus, we obtain the Neumann boundary condition

$$\left(\frac{\partial p}{\partial \vec{n}}\right)^{n+1} = -\frac{1}{\delta t}\left(\vec{v}_\Gamma^{\,n+1} - \vec{v}_\Gamma^{\,*}\right)\cdot \vec{n},$$

where $\vec{v}_\Gamma$ is the value of $\vec{v}$ on Γ. Assuming $\vec{v}\cdot\vec{n} = 0$ on Γ, we obtain

$$\left(\frac{\partial p}{\partial \vec{n}}\right)^{n+1} = 0$$

on Γ. Moreover, the Dirichlet boundary condition

$$p = p_0 + \sigma\kappa + \vec{n}\cdot\tau\cdot\vec{n}$$

applies for free surface as well as for outflow particles in the context of the pressure Poisson equation (10).

We note that particle positions change only in the first step. The intermediate velocity $\vec{v}^*$ is obtained for each particle on its new location. Finally, the pressure and the divergence free velocity fields are computed also on exactly the same new particle positions.

We approximate the spatial derivatives appearing in (7) and (8) by the weighted least squares method. Furthermore, the pressure Poisson equation (10) is also solved in the least squares sense. In the following section, we describe the method of approximation of spatial derivatives. The Poisson solver, and the approximation of the curvature of free surfaces by the weighted least squares method are presented as well.

4. Weighted least squares method (WLS) and its application for FPM

In general, we would like to approximate spatial derivatives of some function. The problem is that we only know the discrete function values exactly at the particle positions. To approximate a derivative of some function at some given point, the discrete function values of the neighbor particles being in a ball about the point considered are taken into account. The WLS, which is employed for that purpose, does not require a regular grid structure. This is of big advantage for FPM.

Let $f(t,\vec{x})$ be a scalar function and $f_i(t)$ its discrete values at the particle positions $\vec{x}_i$ for $i = 1,2,\cdots,N$ and time t. Consider the problem to approximate spatial derivatives of that particular function $f(t,\vec{x})$ at some particular particle position $\vec{x}$ based on the discrete function values of its neighbor points.

In order to restrict the number of points we introduce a weight function $w = w(\vec{x}_i - \vec{x}, h)$ with small compact support, where h determines the size of the support and represents the smoothing length. The weight function can be arbitrary, however it makes sense to choose a Gaussian weight function of the form

$$w(\vec{x}_i - \vec{x}, h) = \begin{cases} \exp\left(-\alpha \cdot \frac{\|\vec{x}_i - \vec{x}\|^2}{h^2} \right), if \ \frac{\|\vec{x}_i - \vec{x}\|}{h} \le 1 \\ 0, else, \end{cases}$$

where α is a positive constant and is considered to be in the range of 6. The size of h defines a set of neighbor particles around $\vec{x}$. So far, in our implementation, we allow user given h as a function in space and time. However, no adaptive choice of h is realized yet. Working with user given h implies that new particles will have to be brought into play as the particle distribution becomes too sparse or, logically, particles will have to be removed from the computation domain as they become too dense.

Let $P(\vec{x}, h) = \{\vec{x}_i : i = 1,2,\cdots n\}$ be the set of n neighbor points of $\vec{x}$ in a ball of radius h. For consistency reasons, some obvious restrictions are required, for example, in 2D there should be at least 5 neighbor particles and they should neither be on the same line nor on the same circle.

The determination of derivatives of a function can be computed easily and accurately by using the Taylor series expansion and the least squares approximation. We write Taylor's expansion about the point $\vec{x}$ with unknown coefficients and then compute these coefficients by minimizing a weighted error over the neighbor points.

Hence, consider Taylor's expansion of $f(t, \vec{x}_i)$ about $\vec{x}$

$$f(t, \vec{x}_i) = f(t, x) + \sum_{k=1}^{3} f_k(t, \vec{x})\left(x_i^{(k)} - x^{(k)}\right) + \frac{1}{2}\sum_{k,l=1}^{3} f_{kl}(t, \vec{x})\left(x_i^{(k)} - x^{(k)}\right)\left(x_i^{(l)} - x^{(l)}\right) + e_i,$$

where e_i is the error in Taylor's expansion at the point $\vec{x}_i$. The symbol $\vec{x}_i^{(k)}$, represents the k-th component of the particle position $\vec{x}_i$. The unknowns f_k and $f_{kl} (= f_{lk})$ for $k, l = 1, 2, 3$ represent the approximations of the first and second derivatives of f and are computed by minimizing the error e_i for $i = 1, 2, \cdots n$, where $f(t, \vec{x}) = f$ is the known discrete function value at the particle position $\vec{x}$. The system of equations can be written as

$$\vec{e} = M\vec{a} - \vec{b},$$

where

$$M = \begin{pmatrix} \Delta x1_1 & \Delta x2_1 & \Delta x3_1 & \Delta x11_1 & \Delta x12_1 & \Delta x13_1 & \Delta x22_1 & \Delta x23_1 & \Delta x33_1 \\ \Delta x1_2 & \Delta x2_2 & \Delta x3_2 & \Delta x11_2 & \Delta x12_2 & \Delta x13_2 & \Delta x22_2 & \Delta x23_1 & \Delta x33_1 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ \Delta x1_n & \Delta x2_n & \Delta x3_n & \Delta x11_n & \Delta x12_n & \Delta x13_n & \Delta x22_n & \Delta x23_n & \Delta x33_n \end{pmatrix},$$

$$\vec{a} = \left[f_1, f_2, f_3, f_{11}, f_{12}, f_{13}, f_{22}, f_{23}, f_{33} \right]^t,$$

$$\vec{b} = \left[f_1 - f, f_2 - f, \cdots, f_n - f \right]^t \text{ and } \vec{e} = \left[e_1, e_2, \cdots, e_n \right]^t.$$

The symbols are defined as

$$\Delta xk_i = x_i^{(k)} - x^{(k)}, \; \Delta xkl_i = \left(x_i^{(k)} - x^{(k)} \right)\left(x_i^{(l)} - x^{(l)} \right) \text{ with } (k \neq l) \text{ and}$$

$$\Delta xkk_i = \frac{1}{2}\left(x_i^{(k)} - x^{(k)} \right)^2 \text{ for } k,l = 1, 2, 3 \text{ and } i = 1, 2, \cdots, n.$$

For $n \geq 10$, this system is over-determined for the nine unknowns f_k and $f_{kl} (= f_{lk})$ for $k,l = 1,2,3$.

The unknowns in the vector $\vec{a}$ are obtained from a weighted least squares method by minimizing the quadratic form

$$J = \sum_{i=1}^{n} w_i e_i^2.$$

The above equations can be expressed as

$$J = \left(M\vec{a} - \vec{b} \right)^t W \left(M\vec{a} - \vec{b} \right)^{-1},$$

with

$$W = \begin{pmatrix} w_1 & 0 & . & . & . & 0 \\ 0 & w_2 & . & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & . & w_n \end{pmatrix},$$

where $w_i = w(\vec{x}_i - \vec{x}, h)$. The minimization of J formally yields

$$\vec{a} = \left(M^t W M\right)^{-1}\left(M^t W\right)\vec{b} \tag{11}$$

The Taylor expansion may include high order expansion. The employment of particular weight functions can force the least square approximations to recover the finite difference discretization in the special case that all particles are placed in a regular grid structure.

4.1. *Weighted least squares approach for the Poisson equation*

As we have seen in the description of the numerical scheme in section 0, we need to solve the pressure Poisson equation

$$\Delta p = \frac{\nabla \cdot \vec{v}^*}{\delta t} \tag{12}$$

with the boundary conditions

$$\frac{\partial p}{\partial \vec{n}} = 0$$

for solid walls as well as for inflow boundaries and

$$p = p_0 + \sigma\kappa + \vec{n} \cdot \tau \cdot \vec{n}$$

for free surface as well as for outflow boundaries. Here, the symbol p denotes p^{n+1} for the sake of simplicity.

Since we are in a grid free structure, it is not obvious to apply the classical methods like finite difference or finite element methods for a numerical scheme solving the above Poisson equation. Of course, one could construct a regular grid and solve the Poisson equation by some classical finite difference method and then interpolate the results of pressure back to the original particle distribution. However, this will give smearing effects and is possibly of high computational effort especially if geometries become complex.

Therefore, we use a local iteration approach on the basis of the least squares approximation, where the Poisson equation is forced to strictly satisfy. The main advantage is that this procedure can be applied directly to the given particle distribution. This method is stable and gives accurate results for all boundary value problems of the Poisson equation, see (Tiwari *et al.* 2001) for details.

In the beginning of this section, we have presented the least squares method to approximate derivatives of a function at an arbitrary point from its neighbor values.

Now we have a slightly different situation. Based on the function values of the neighbor particles, we would like to compute an approximate function value under the condition that some determined value of the approximate Laplacien is fulfilled. The pressure values at the new particle positions are not yet known. Therefore, the least squares approach cannot be applied directly as described in the previous subsection. Hence, we prescribe an initial guess $p^{(0)}$ for the pressure p. Now, we consider the problem of determining p at an arbitrary particle position $\vec{x}$ from its neighbor points $\vec{x}_i, i = 1, \cdots, n$. As in the previous section, we again consider a Taylor expansion of p about some point $\vec{x}$

$$p^{(j)}(\vec{x}_i) = p^{(j+1)}(\vec{x}) + \sum_{k=1}^{3} p_k^{(j+1)}(\vec{x})\left(x_i^{(k)} - x^{(k)}\right) +$$

$$\frac{1}{2}\sum_{k,l=1}^{3} p_{kl}^{(j+1)}(\vec{x})\left(x_i^{(k)} - x^{(k)}\right)\left(x_i^{(l)} - x^{(l)}\right) + e_i^{(j+1)} \tag{13}$$

for $j = 0, 1, 2, \cdots$, where $p^{(0)}(\vec{x}_i)$ are the given initial discrete particle values. We require that the Poisson equation (12) be satisfied at $\vec{x}$. Hence, we have to add the following equation to the set of n equations in (13)

$$\frac{\nabla \cdot \vec{v}^*}{\delta t} = p_{11}^{(j+1)}(\vec{x}) + p_{22}^{(j+1)}(\vec{x}) + p_{33}^{(j+1)}(\vec{x}) \cdot$$

If $\vec{x}$ is a particle of some solid wall or inflow boundary, we also have to enforce the Neumann boundary condition to strictly satisfy by adding the equation

$$0 = p_1^{(j+1)}(\vec{x})n_x + p_2^{(j+1)}(\vec{x})n_y + p_3^{(j+1)}(\vec{x})n_z$$

to the given system of $n+1$ equations. Here, n_x, n_y, n_z are the respective components of the unit normal vector $\vec{n}$.

If the particle belongs to a free surface or outflow boundary, we have the Dirichlet condition satisfy strictly by adding the equation

$$p_\Gamma(\vec{x}) = p^{(j+1)}(\vec{x}) \cdot$$

Here, $p_\Gamma(\vec{x})$ is a user given boundary value for the pressure. Summing up, for boundary particles, we have a total of $n+2$ equations for 10 unknowns. In general, the number of neighbors is greater than 10.

The coefficients we obtain by minimizing the residuals e_i are

$$p^{(j+1)}, p_1^{(j+1)}, p_2^{(j+1)}, p_3^{(j+1)}, p_{11}^{(j+1)}, p_{12}^{(j+1)}, p_{13}^{(j+1)}, p_{22}^{(j+1)}, p_{23}^{(j+1)}, p_{33}^{(j+1)}$$

for $j = 0, 1, 2, \cdots$ at a particular location $\vec{x}$. For example, the functional to be minimized for a boundary particle with Neumann boundary condition reads as

$$J = \sum_{i=1}^{n} w_i \left(e_i^{(j+1)}\right)^2 + \left(\Delta p^{(j+1)} - \frac{\nabla \cdot \vec{v}^*}{\delta t}\right)^2 + \left(\frac{\partial p^{(j+1)}}{\partial \vec{n}} - 0\right)^2 \cdot$$

Similarly to (11), the minimization of J is given by

$$\vec{a}^{(j+1)} = \left(M^t W M\right)^{-1} \left(M^t W\right) \vec{b}^{(j)}, \; j = 0, 1, 2, \cdots,$$

where the matrices and the vectors differ slightly from (11) and are given by

$$M = \begin{pmatrix} 1 & \Delta x1_1 & \Delta x2_1 & \Delta x3_1 & \Delta x11_1 & \Delta x12_1 & \Delta x13_1 & \Delta x22_1 & \Delta x23_1 & \Delta x33_1 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ 1 & \Delta x1_n & \Delta x2_n & \Delta x3_n & \Delta x11_n & \Delta x12_n & \Delta x13_n & \Delta x22_n & \Delta x23_n & \Delta x33_n \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & n_x & n_y & n_z & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$W = \begin{pmatrix} w_1 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & w_2 & . & . & . & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & w_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\vec{a}^{(j+1)} = \left[p^{(j+1)}, p_1^{(j+1)}, p_2^{(j+1)}, p_3^{(j+1)}, p_{11}^{(j+1)}, p_{12}^{(j+1)}, p_{13}^{(j+1)}, p_{22}^{(j+1)}, p_{23}^{(j+1)}, p_{33}^{(j+1)}\right],$$

$$\vec{b}^{(j)} = \left[p_1^{(j)}, p_2^{(j)}, \cdots, p_n^{(j)}, \frac{\nabla \cdot \vec{v}^*}{\delta t}, 0\right]^t \cdot$$

The scheme (13) is clearly an iterative process. The iteration is stopped if the local error satisfies

$$\frac{\sum_{i=1}^{N}\left|p_i^{(j+1)}-p_i^{(j)}\right|}{\sum_{i=1}^{N}\left|p_i^{(j+1)}\right|}\le\varepsilon .$$

Finally, the solution is defined by $p(\vec{x}_i) := p^{(j+1)}(\vec{x}_i)$ as j tends to infinity. The parameter ε is a very small positive constant and can differ according to the size of h. The convergence rate is faster if h is taken larger. Therefore, multigrid approaches can indeed be useful in order to reduce the computational effort.

Of course, it is necessary to prescribe the initial value of the pressure at time $t=0$. For the pressure iteration, the initial guess of the pressure for time level $n+1$ is taken as the pressure from time level n.

4.2. *Approximation of derivatives of velocities on the free surface*

For free surface particles we are required to include the boundary conditions (4) and (5) into the approximation of spatial derivatives in (7). For the sake of simplicity we consider the case of two spatial dimensions. The boundary conditions (4) and (5) can be explicitly written as

$$2\mu\left(\frac{\partial u}{\partial x}n_x^2+\frac{\partial u}{\partial y}n_xn_y+\frac{\partial v}{\partial x}n_xn_y+\frac{\partial v}{\partial y}n_y^2\right)=p-p_0+\sigma\kappa \tag{14}$$

$$2\frac{\partial u}{\partial x}n_xn_y+\frac{\partial u}{\partial y}\left(n_y^2-n_x^2\right)+\frac{\partial v}{\partial x}\left(n_y^2-n_x^2\right)+2\frac{\partial v}{\partial y}n_xn_y=0. \tag{15}$$

Here, u,v denote the respective components of the velocity vector $\vec{v}$ and n_x,n_y denote the components of the unit normal vector and the tangent vector is defined by $\vec{t}=\left(-n_y,n_x\right)$.

The incorporation of (14) and (15) into the approximation of derivatives of velocities on the free surface is not straightforward. (14) and (15) both contain the first derivatives of both u and v. On the other hand, for example, the derivatives of u are obtained by Taylor's expansion, which contains only the derivatives of u but not of v. Therefore, we have to compute derivatives of u and v together. The method is again an extension of the occurring least squares matrix. Suppose that we want to approximate the derivatives of u and v on the free surface particle located at (x,y) from its neighbor values. Let $(u,v)=(u,v)(x,y)$, $(u,v)_i=(u,v)(x_i,y_i)$, $dx_i=x_i-x$, $dy_i=y_i-y$ for $i=1,\cdots,n$. Consider Taylor's expansion of u and v around (x,y)

$$u_i = u + \frac{\partial u}{\partial x} dx_i + \frac{\partial u}{\partial y} dy_i + \frac{1}{2}\frac{\partial^2 u}{\partial x^2} dx_i^2 + \frac{\partial^2 u}{\partial x \partial y} dx_i dy_i + \frac{1}{2}\frac{\partial^2 u}{\partial y^2} dy_i^2 + e_{ui} \tag{16}$$

and

$$v_i = v + \frac{\partial v}{\partial x} dx_i + \frac{\partial v}{\partial y} dy_i + \frac{1}{2}\frac{\partial^2 v}{\partial x^2} dx_i^2 + \frac{\partial^2 v}{\partial x \partial y} dx_i dy_i + \frac{1}{2}\frac{\partial^2 v}{\partial y^2} dy_i^2 + e_{vi} \tag{17}$$

As in the previous cases, we can rewrite equations (16) and (17) in matrix form as

$$\vec{e} = M\vec{a} - \vec{b},$$

where

$$M = \begin{pmatrix} dx_1 & dy_1 & \frac{1}{2}dx_1^2 & dx_1 dy_1 & \frac{1}{2}dy_1^2 & 0 & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ dx_n & dy_n & \frac{1}{2}dx_n^2 & dx_n dy_n & \frac{1}{2}dy_n^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & dx_1 & dy_1 & \frac{1}{2}dx_1^2 & dx_1 dy_1 & \frac{1}{2}dy_1^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & dx_n & dy_n & \frac{1}{2}dx_n^2 & dx_n dy_n & \frac{1}{2}dy_n^2 \end{pmatrix} \tag{18}$$

$\vec{a} = \left[u_x, u_y, u_{xx}, u_{xy}, u_{yy}, v_x, v_y, v_{xx}, v_{xy}, v_{yy}\right]^t$ and

$\vec{b} = \left[u_1 - u, \cdots, u_n - u, v_1 - v, \cdots, v_n - v\right]^t$.

The above system has $2n$ equations with 10 unknowns. The minimization of the error gives the derivatives of both velocity components together. However, one has to invert a 10-by-10 matrix instead of a 5-by-5 matrix. Therefore, we use this larger system only for the few free surface particles. For the incorporation of the boundary

conditions (14) and (15), we have to add these equations to the system (16), (17), where the matrix (18) is enhanced by two lines in the sense

$$\begin{matrix} 2\mu n_x^2 & 2\mu n_x n_y & 0 & 0 & 0 & 2\mu n_x n_y & 2\mu n_y^2 & 0 & 0 & 0 \\ 2n_x n_y & \left(n_y^2 - n_x^2\right) & 0 & 0 & 0 & \left(n_y^2 - n_x^2\right) & 2n_x n_y & 0 & 0 & 0 \end{matrix}$$

and the right hand side vector $\vec{b}$ is given by

$$\vec{b} = \left[u_1 - u, \cdots, u_n - u, v_1 - v, \cdots v_n - v, p - p_0 + \sigma\kappa, 0\right]^t .$$

The minimization process is the same as above.

4.3. *Approximation of the local curvature on free surfaces*

Boundary condition (4) requires the knowledge of the curvature of the free surface. In this section, we describe the approximation of the curvature in the two dimensional case. We approximate a circle of radius R with the center $\left(x_c, y_c\right)$ running trough the free surface particle located at $\left(x, y\right)$ such that it fits, locally, all the neighbor-surface-points in a least squares sense. The curvature κ and the unit normal vector on free surface at $\left(x, y\right)$ are given by

$$\kappa = \frac{1}{R},$$

$$n_x = \left(n - x_c\right)\kappa, \; n_y = \left(y - y_c\right)\kappa .$$

If the center of the circle lies outside of the fluid considered, the sign of curvature is taken negative.

A circle is represented by a general second order equation

$$x^2 + y^2 + Dx + Ey + F = 0,$$

where D, E, F are to be determined. The radius and the center of the circle running through $\left(x, y\right)$ is given by

$$x_c = \frac{-D}{2}, y_c = -\frac{E}{2}, R = \frac{1}{2}\sqrt{D^2 + E^2 - 4F}, \left(D^2 + E^2 \succ 4F\right)$$

Now, we have to determine the coefficients D, E, F at every free surface particle $\left(x, y\right)$ from its free surface neighbor particles $\left(x_i, y_i\right), i = 1, \cdots, n$. Here $\left(x, y\right)$ is one of

the (x_i, y_i). In general, there are more than three neighbor points, therefore these coefficients are approximated by the weighted least squares method, as described above. In order to avoid the least squares approximation, one can choose two nearest neighbors (one left and one right) of (x, y) such that the circle can be fitted more accurately. Singularities may occur, if all free surface particles lie on the same straight line. In this case the curvature is considered to be zero.

5. Determination of the free surface particles

In this section we would like to give a brief description of the strategy how to indicate particles belonging to a free surface. We would like to remind the reader, that these particles are not known a priori, however it is important to have a very accurate selection of them, otherwise the whole numerical procedure and application of boundary conditions is likely to fail. For the determination of the free surface particles, we come up with a definition. We say that a particle at the position $\vec{x}_i$ belongs to some free surface, if we can place a sphere in the neighborhood of the particle such that

(i) $\vec{x}_i$ belongs to the surface of the sphere (*i.e.* it is not the center)

(ii) the radius of the sphere is $r_S = \alpha h$ where h is the smoothing length and α is a constant, preferably in the range $\alpha \in [0.7,1.0]$

(iii) no other particle lies inside of the sphere

This definition is rather theoretical, however it makes sense. If a particle is really at the free surface, then there will be indeed such a sphere, because one half-space is more or less empty for surface particles. An interior particle, however, should not find such a sphere, or, in other words, if it would find a sphere meeting the above conditions, then this would mean there is a big hole in the interior of the flow domain, and this is not acceptable from the point of view of computational accuracy. Consequently, this means that interior holes have to be stuffed with particles before their radius tends to reach the magnitude of r_S.

Obeying these rules, we have a unique description of particles at the free surface. More problematic is the implementation of the whole idea. To search for appropriate holes for one particle (for instance for the particle at position $\vec{x}_i$), it takes about $25M^2$ floating point operations, where M is the number of relevant neighbor particles related to the position $\vec{x}_i$. However, M is usually in the range of $M \in [20,50]$ for 2D-applications and $M \in [40,90]$ for 3D-applications, depending on the particle configuration. Hence, the effort of searching surface particles is huge and can take 10 percent of the over-all-computation-time. The idea to reduce that effort is to

(i) consider only those particles as candidates for being at the free surface at time level t_n, if they were in the neighborhood of a free surface particle at time level t_{n-1} (this reduces the number of particles to be checked)

(ii) do the search for the free surface particles not for each time step.

Both methods mentioned above have shown excellent applicability.

6. Numerical Tests

6.1. *Breaking dam problem*

The breaking dam problem is a very popular and simple test case, which helps to validate numerical schemes for the simulation of free surface flows. It consists of a simple initial configurations and simple initial and boundary conditions. In (Martin *et al.* 1952) the experimental results are reported and several authors have reported their numerical results (Hansbo 92, Hirt *et al.* 1981, Kelecy *et al.* 1997, Maronnier *et al.* 1999, Monaghan 94).

Consider a rectangular column of water with a width of $a = 0.1m$ and a height of $0.2m$. The lines $x = 0, y = 0$, and $x = 0.6$ consist of the solid wall. In the simulation, the upper and the right boundary of the water columns are considered as the free surface boundary. Initially, 1136 particles are distributed randomly. The size of the smoothing length is $h = 0.01$. The gravity is $g = 9.81m/s^2$ and acts downwards. The initial velocity is set to zero. The initial pressure p_0 is also considered to be zero. The air pressure is assumed to be zero and surface tension forces are neglected. When the right wall (dam) is removed, the water column collapses under the influence of the gravity. The density and the viscosity of the fluid are $\rho = 1kg/m^3, \mu = 0.0004kg/(ms)$. No slip boundary condition is used on the solid walls. The particles, plotted successively in time, are shown in Figure 1

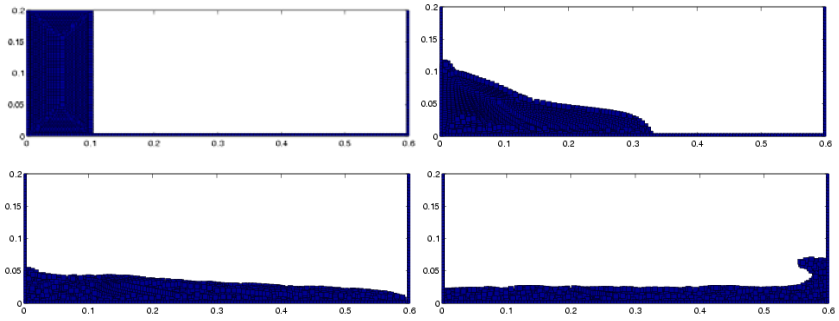

Figure 1. *Particle positions at successive times*

In Figure 2 the position of the leading fluid front versus time is compared with experimental results (Martin *et al.* 1952). The figure shows a good agreement between the numerical and experimental results.

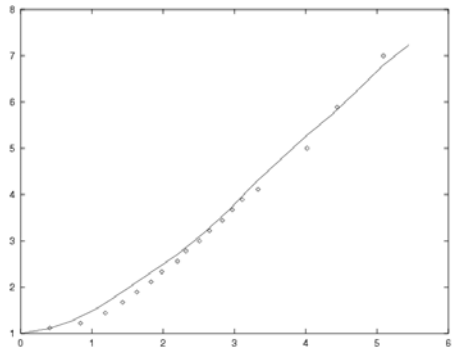

Figure 2. *Dimensionless front position* x/a *versus dimensionless time* $t/\sqrt{2g/a}$

6.2. *Laplace Law*

It is well known that a drop of arbitrary shape becomes spherical due to the surface tension forces on the free surface boundary. In the equilibrium state a bubble should satisfy the Laplace's law

$$p_l - p_g = \sigma\kappa.$$

Here, p_l is the pressure inside of the liquid drop and p_g is the background pressure, which is considered to be zero. In the following, we have considered three types of drops, the exact circular drop, the octagonal drop and the square shaped drop. In all cases we consider the fluid parameters $\rho = 1kg/m^3$, $\mu = 0.1kg(ms)$, $\sigma = 1dynes/m$. The initial pressure, velocity and the body force are set to zero in all cases. The drop pressure is considered as the average pressure of all particles. Hence, in the equilibrium, the following relation must hold

$$p_l = \sigma\kappa.$$

(a) Exact circular drop: We consider the exact circle of radius 1 on the free surface and we generate particles inside as shown in Figure 3. The particles are placed in the distance of 1/10. In this case we obtain a curvature of 1 along the free

surface boundary. The initial pressure of the drop is considered to be zero. After time $t = 0.002s$ the pressure reaches 1. The maximum velocity is 2.9952e-7.

Figure 3. *Exact circular drop*

(b) Octagonal drop: Here, we approximate a circle by an octagon. In this case there are 8 corner points, where the curvatures naturally are higher. As many authors (Lafaurie *et al*. 1994,Ginzburg *et al*. 2001) reported the ``spurious" or ``anomalous" currents around the free surface. In practice we always obtain a n-gon for finer grids. Hence, the large curvatures on the corners of the free surface boundary produce such currents as shown in Figure 4. The drop reaches equilibrium and becomes circular in the steady state, see Figure 5. In the steady state, the maximum velocity is equal to 3.88069e-4 and the drop pressure is equal to 1.06943992. Figure 6 shows that the relation between surface curvature and drop pressure justifies the Laplace law.

(c) Square shaped drop: As a final test of the Laplace Law, we consider a square shaped drop. In this case there are four corners representing naturally large curvature. The value of the curvatures on the other free surface particles is zero. In Figure 7, we have plotted the time evolution of the drop. After short time, it shows some oscillating behavior but finally it reaches the state of equilibrium. For larger viscosity, the drop reaches equilibrium without oscillation. In the equilibrium state the maximum velocity is equal to 1.47952134e-3 and the pressure is equal to 1.77552119. Hence, the value of the pressure and the curvature of the drop in equilibrium justify the Laplace law (see Figure 8).

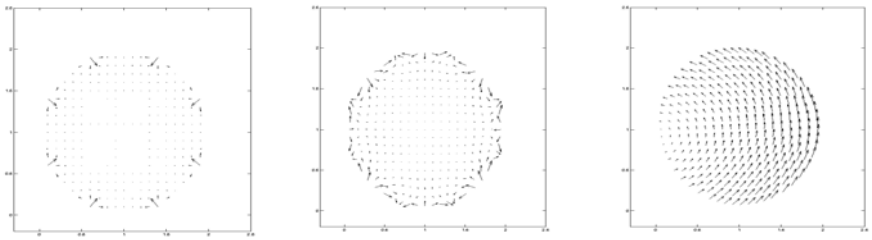

Figure 4. *Velocity profile of octagonal drop at time* $t = 0.001s$ *(left),* $t = 0.301s$ *(middle) and* $t = 5.001s$ *(right)}*

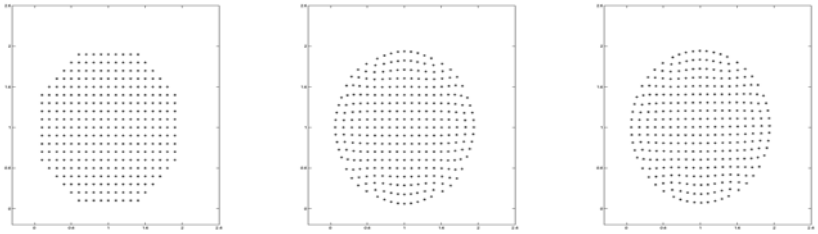

Figure 5. *Positions of particles octagonal drop at time* $t = 0.0s$ *(left),* $t = 0.301s$ *(middle) and* $t = 5.001s$ *(right)*

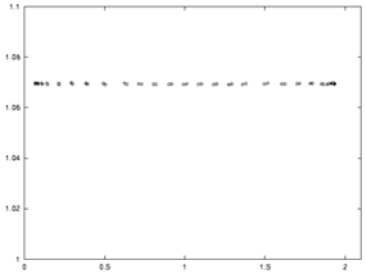

Figure 6. *The curvature on free surface of the octagonal drop at* $t = 5.001s$

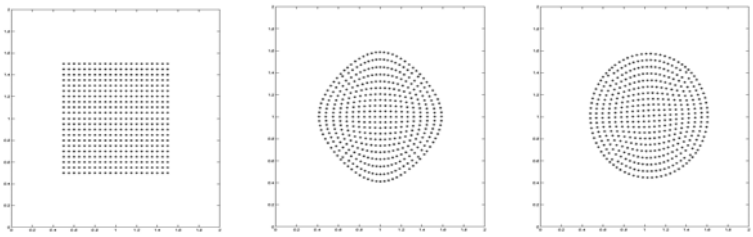

Figure 7. *Square drop at time* $t = 0.001s$ *(left),* $t = 0.301s$ *(middle) and* $t = 5.001s$ *(right)*

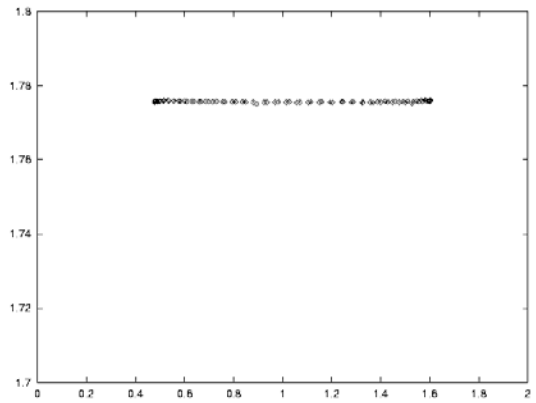

Figure 8. *The curvature on free surface of the square drop at* $t = 5.001s$

6.3. *Drop Collisions*

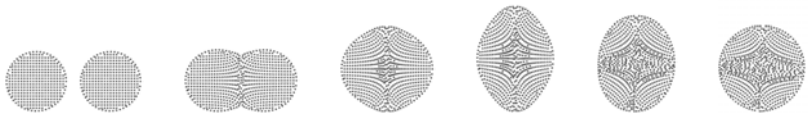

Figure 9. *Head on collision at time* $t = 0.001s$, $t = 0.101s$, $t = 0.201s$, $t = 0.301s$, $t = 3.001s$ *and* $t = 5.001s$ *(from left to right)*

We consider two drops of the same size moving with the same magnitude of velocities in opposite directions. The magnitude of the initial velocity is $4m/s$. The body force of the drops is considered to be zero. The radii of the drops equal to $1m$ and the initial spacing of the particles is 1/10. The density and viscosity are $\rho = 1kg/m^3$ and $\mu = 1kg(ms)$. Hence the Reynolds number is $\mathrm{Re} = \rho UD/\mu = 16$, where $U = 8m/s$ is the relative velocity, $D = 2$ is the diameter of drop. The surface tension coefficient is set to $\sigma = 1dynes/m$, such that the Weber number becomes $We = \rho DU^2/\sigma = 128$. The numerical results are very close to the experimental and other numerical results, presented in (Ash *et al.* 1990, Kothe *et al.* 1992,Lafaurie *et al.* 1994).

Two types of collisions are considered. The first one is the head on collision. In the parameters mentioned above both drops are merging into a single drop after collision. As we see in Figure 9, the drop becomes elliptical at time $t = 0.301s$. Due

to the higher surface tension on the top and bottom of the drop, it starts to shrink the top and bottom and stretching the left and right side. The results are comparable with those presented in (Ash *et al*. 1990). Finally, it reaches the equilibrium state and the shape remains unchanged. We simply wanted to test whether the presented method works for simulations of free surface flows and therefore have not tested the collisions for higher surface tension force and larger Reynolds number.

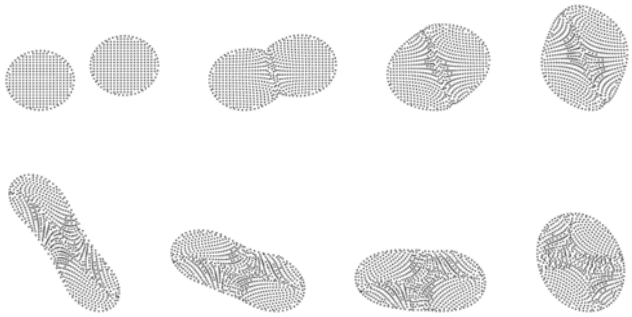

Figure 10. *Non-central collision with impact parameter 0.25 collision at time* $t = 0.001s$, $t = 0.101s$, $t = 0.201s$, $t = 0.301s$, $t = 0.701s$, $t = 1.251s$, $t = 3.001s$ *and* $t = 5.001s$ *(from left to right and top to bottom)*

As a second example of drop collision we consider the non-central collision with impact parameter $B = 0.25$. Other input parameters are same as in the case of head on collision. In contrast to the first case, the drop is rotating after collision. Shrinking and stretching of drop due to the effect of surface tension is similar to the head on collision. The time evolution of the drop is presented in Figure 10.

7. Conclusion

A meshfree method is used to simulate free surface flows. The incompressible Navier-Stokes equations are used as a mathematical model. The numerical experiments are performed with and without surface tension force on free surface. The spatial derivatives of the Navier-Stokes equations are approximated by the weighted least squares method. The pressure Poisson equation is solved by the least squares method. Free surface boundary conditions can be directly included in the least squares approximation. Locations of free surface particles are determined by a very simple approach. Close agreements between numerical and experimental results show the robustness of the scheme. Future work will be the extension of the method in 3D.

Acknowledgments

The authors would like to thank Deutsche Forschungsgemeinschaft for the financial support.

8. References

Ash N., Poo J. Y., "Coalescence and separation in binary collisions of liquid drops", *J. Fluid Mech*., Vol. 221, 1990, pp. 183-204.

Belytschko T., Krongauz Y., Flemming M., "Organ D., Liu W.K.S., Smoothing and accelerated computations in the element free Galerkin method", *J. Comp. Appl. Maths*,. Vol. 74, 1996, pp. 111-126.

Chorin A., "Numerical solution of the Navier-Stokes equations", *J. Math. Comput*,. Vol. 22, 1968, pp. 745-762.

Dilts G. A., "Moving least squares particle hydrodynamics. I: consistency and stability", *Hydrodynamics methods group report*, Los Alamos National Laboratory, 1996.

Gingold R. A., Monaghan J. J., "Smoothed particle hydrodynamics: theory and application to non-spherical stars", *Mon. Not. Roy. Astron. Soc.*, Vol. 181, 1977, pp. 375-389.

Ginzburg I., Wittum G., "Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants", *J. Comp. Phys.*,. Vol. 166, 2001, pp. 302-335.

Hansbo P., "The characteristic streamline diffusion method for the time dependent incompressible Navier-Stokes equations", *Comp. Meth. Appl. Mech. Eng*., Vol. 99, 1992, pp. 171-186.

Harlow F. H., Welch J. E., "Numerical study of large amplitude free surface motions", *Phys. Fluids*, Vol.8, 1965, p. 2182.

Hirt C. W., Nichols B. D., "Volume of fluid (VOF) method for dynamic of free boundaries", *J. Comput. Phys.*, Vol. 39, 1981, p. 201.

Kelecy F. J., Pletcher R. H., "The development of free surface capturing approach for multi dimensional free surface flows in closed containers", *J. Comput. Phys*., Vol. 138, 1997, p. 939.

Kothe D. B., Mjolsness R. C., "RIPPLE: A new model for incompressible flows with free surfaces", *AIAA Journal*, Vol. 30, No 11, 1992, pp. 2694-2700.

Kuhnert J., General smoothed particle hydrodynamics, Ph.D. Thesis, Kaiserslautern University, Germany, 1999.

Kuhnert J., "An upwind finite pointset method for compressible Euler and Navier-Stokes equations", *preprint, ITWM, Kaiserslautern*, Germany, 2000.

Kuhnert J., Tramecon A., Ullrich P., "Advanced Air Bag Fluid Structure Coupled Simulations applied to out-of Position Cases", *EUROPAM Conference Proceedings 2000*, ESI group, Paris, France.

Landau L. D., Lifshitz E. M., *Fluid Mechanics*, Pergamon, New York, 1959.

Lafaurie B., Nardone C., Scardovelli R., Zaleski S., Zanetti G., "Modelling Merging and Fragmentation in Multiphase Flows with SURFER", *J. Comput. Phys.*, Vol. 113, 1994, pp. 134 - 147.

Lucy L. B., "A numerical approach to the testing of the fission hypothesis", *Astron, J.*, Vol. 82, 1977, p. 1013.

Maronnier V., Picasso M., Rappaz J., "Numerical simulation of free surface flows", *J. Comput. Phys*. Vol. 155, 1999, p. 439.

Martin J. C., Moyce M. J., "An experimental study of the collapse of liquid columns on a liquid horizontal plate", *Philos. Trans. Roy. Soc. London*, Ser. A 244, 1952, p. 312.

Monaghan J. J., "Smoothed particle hydrodynamics", *Annu. Rev. Astron. Astrop*, Vol. 30, 1992, pp. 543-574.

Monaghan J. J., "Simulating free surface flows with SPH", *J. Comput. Phys.*, Vol. 110, 1994, p. 399.

Monaghan J. J., Gingold R. A., "Shock Simulation by particle method SPH", *J. Comp. Phys.*, Vol. 52, 1983, pp. 374-389.

Morris J. P., Fox P. J., Zhu Y., "Modeling Low Reynolds Number Incompressible Flows Using SPH", *J. Comput. Phys*., Vol. 136, 1997, pp. 214-226.

Tiwari S., Kuhnert J., "Grid free method for solving Poisson equation", *Berichte des Fraunhofer ITWM*, Kaiserslautern, Germany, Nr. 25, 2001.

Tiwari S., Kuhnert J., "Finite pointset method based on the projection method for simulations of the incompressible Navier-Stokes equations", to appear in M. Griebel, M. A. Schweitzer (Eds.), Springer LNCSE*: Meshfree Methods for Partial Differential Equations*, Springer-Verlag, 2003.

Tiwari S., Kuhnert J., *Particle method for simulations of free surface flows*, preprint Fraunhofer ITWM, Kaiserslautern, Germany, 2000.

Tiwari S., "A LSQ-SPH approach for compressible viscous flows", to appear in *Proceedings of the 8th International Conference on Hyperbolic Problems Hyp2000*.

Tiwari S., Manservisi S., *Modeling incompressible Navier-Stokes flows by LSQ-SPH*, Berichte des Fraunhofer ITWM, Kaiserslautern, Germany, 2000.