A Numerical Scheme for Solving Incompressible and Low Mach Number Flows by the Finite Pointset Method

Sudarshan Tiwari* and Jörg Kuhnert**

Fraunhofer Institut für Techno- und Wirtschaftmathematik, Gottlieb Daimler Strasse, Geb. 49, D–67663 Kaiserslautern, Germany.

Abstract A meshfree projection method for compressible as well as incompressible flows and the coupling of two phase flows with high density and viscosity ratios is presented. The Navier-Stokes equations are considered as the basic mathematical model and are solved by the implicit projection method. The implicit projection method yields the linear second order partial differential equations. These equations are solved by the weighted least squares method and are compared with the exact solutions. A one dimensional shock tube problem is exhibited for compressible flows. Finally, two phase incompressible and quasi compressible flows are used to simulate a two phase cavity filling problem.

1 Introduction

The Finite Pointset Method (FPM) is a meshfree method to solve partial differential equations. The computational domain is represented by a finite number of particles (pointset), also referred to as numerical points. These points can be arbitrarily distributed, however they have to provide a neighborhood relationship governed by the smoothing length, i.e. each point needs to find sufficiently many neighbor points within a ball of certain radius. Considering the equation of fluid dynamics, the numerical points move with fluid velocity and carry all information which completely describes the flow problem concerned. Of course, this is a fully Lagrangian method being appropriate for flow simulations with complicated as well as rapidly changing geometry [KTU00], involving free surfaces [TK202, TK03] or phase boundaries [HJKT03].

The classical meshfree Lagrangian method to handle problems in fluid dynamics is the Smoothed Particle Hydrodynamics (SPH). SPH was initially developed to study phenomena in astrophysics [GM97, 15]. Later, it was extended to flow cases even on earth [CR99, 16, Mor00, MFZ97]. Unfortunately,

* tiwari@itwm.fhg.de
** kuhnert@itwm.fhg.de

SPH has poor approximation properties, especially of the second order derivatives, required to model the Navier-Stokes equations. Moreover, it is difficult to incorporate boundary conditions of certain types. In SPH, incompressible flows are approximated by using the compressible approach together with a very stiff equation of state.

The FPM is based on least squares approximations, where the higher order derivatives can be approximated very accurately and the boundary conditions can be treated in a classical sense [Ku99]. Several computations of flow problems using the method of least squares or moving least squares are reported by different authors, see [Dil96, Ku99, Ku02, TK01, TK102, TK202, TK03, TM03, Tiw00] and other references therein.

The numerical scheme for incompressible and slightly compressible flow phenomena, presented in this article, is based on the classical projection idea of Chorin [Cho68, TK102]. Due to that, the solutions of Poisson as well as Helmholtz differential equations, in particular, form a central task of FPM. These equations can be solved directly in the given meshfree structure with Dirichlet, Neumann or Cauchy boundary conditions in a very accurate way [TK01]. Also, see section 3.1 and 3.2. Moreover, free surfaces can be incorporated very efficiently [TK202].

For some industrial applications, such as simulations of car tank refueling, several fluid phases like fuel, air and foam might be involved. Not all phases can be assumed to be incompressible, as for instance the air inside of the tank might be compressed during the filling process. This is a rather slow compression, with the Mach number tending to zero. However, the compression plays a big role as it partially governs the filling process. Thus, we would like to incorporate compressibility effects into the classical re-projection idea and finally come up with an implicit scheme for compressible as well as incompressible flows.

Therefore, we are going to present an idea to simulate low Mach number and incompressible flows with exactly the same procedure, i.e. the incompressible case turns out to be a special case of the compressible regime. We consider the Navier-Stokes equations as the mathematical model. We solve these equations by the projection method implicitly. The implicit scheme results in linear second order partial differential equations (Poisson, Helmholtz). We solve them using the constraint least squares method suggested in [TK01], see also section 4.

Most of the methods for solving multi phase flows are based on meshgrid techniques [BKZ92, GW01, HW65, HN81, KP97], where additional computational effort has to be put in order to model the dynamics of interphase boundaries. The advantage of using the particle method is that phases can be distinguished by simply assigning flags to the fluid particles which identify their proper phase. The phase-flags are carried in the same fashion as all other physical data.

Since the particles move with fluid velocity, they may scatter or accumulate together. If they scatter and create holes in the computational domain,

singularities may arise. Hence, holes have to be detected and new particles have to be added. Similarly, any two particles being too close to each other, have to be replaced by a single one.

In this paper we have excluded surface tension effects. The CSF model [BKZ92] can easily be extended by using the approach proposed in [Mor00]. The work is in progress.

We have obtained results from convergence studies for general second order linear partial differential equations. If the coefficients are constant, the scheme has second order convergence. If the coefficients are discontinuous, which occur for solving multi phase flows, the proposed scheme is of first order convergence. The implicit projection method is tested for compressible flows by solving a 1D shock tube problem and the results are compared with the exact solutions. Finally, we present a two phase flow case for cavity filling, where the air is considered to be compressible.

The paper is organized as follows. In section 2, we introduce the mathematical model and the numerical scheme. In section 3, we present the FPM for solving general elliptic partial differential equations. The numerical results are presented in section 4.

2 Governing Equations

We consider two immiscible fluids, for example, liquid and gas. We distinguish the liquid and gas particles by assigning appropriate flags on them. We assume that the viscosity μ and the density ρ jump are discontinuous in the phase boundary. These discontinuities can cause numerical instabilities around the interfaces. To avoid them, in every time step we consider the smoothed densities and viscosities on and around the interface. This means for the discretization of the momentum equations, we consider the smooth density and viscosity and then reassign these values to the original (non smoothed) ones. The interface region can be detected by checking the flags of particles in the neighborhood. We update the smoothed density $\tilde{\rho}$ and the smoothed viscosity $\tilde{\mu}$ in each time step at each particle position $\boldsymbol{x}$ near the interface by using the Shepard interpolation

$$\tilde{\rho}(\boldsymbol{x}) = \frac{\sum_{i=1}^{m} w_i \rho_i}{\sum_{i=1}^{m} w_i}, \qquad \tilde{\mu}(\boldsymbol{x}) = \frac{\sum_{i=1}^{m} w_i \mu_i}{\sum_{i=1}^{m} w_i}, \tag{2.1}$$

where m is the total number of neighbor particles related to $\boldsymbol{x}$ (i.e. all numerical points being within a circle of radius h around $\boldsymbol{x}$, h is called smoothing length). The neighbor particles in the interface region are taken from the liquid as well as from the gas phases. Far from the interface we have $\tilde{\rho} = \rho$ and $\tilde{\mu} = \mu$. We consider a truncated Gaussian as weight function, in general this can be any compactly supported smooth function.

2.1 Navier–Stokes Equations

Let Ω be an open bounded domain in $\mathbb{R}^d$ $(d = 1, 2, 3)$ with boundary Γ. Let $\boldsymbol{v}$, T and p be the velocity, temperature and pressure fields representing the state variables. The compressible Navier-Stokes equations in the Lagrangian form can be written as

$$\frac{D\rho}{Dt} = -\rho\nabla\cdot\boldsymbol{v} \tag{2.2}$$

$$\rho\frac{D\boldsymbol{v}}{Dt} = -\nabla p + \nabla\cdot\sigma(\boldsymbol{v}) + \rho\boldsymbol{g} \tag{2.3}$$

$$\rho c_v\frac{DT}{Dt} = -p\nabla\cdot\boldsymbol{v} + (\sigma\cdot\nabla)\cdot\boldsymbol{v} + \nabla\cdot(\kappa\nabla T). \tag{2.4}$$

Here, κ denotes the heat conduction coefficient, $\boldsymbol{g}$ the body force, c_v the specific heat capacity. By D/Dt we denote the Lagrangian derivative. The stress tensor is

$$\sigma_{ij}(\boldsymbol{v}) = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\nabla\cdot\boldsymbol{v}\delta_{ij}\right),$$

where $\delta_{ij} = 0$ is the Kronecker delta. We close the system (2.2-2.4) by the equation of state

$$\rho = \rho(p, T). \tag{2.5}$$

Then, from the continuity equation (2.2) we obtain

$$\nabla\cdot\boldsymbol{v} = -\frac{1}{\rho}\frac{D\rho}{Dt} = -\frac{1}{\rho}(\frac{\partial\rho}{\partial p}\frac{Dp}{Dt} + \frac{\partial\rho}{\partial T}\frac{DT}{Dt}). \tag{2.6}$$

Equation (2.6) is a very important relation which later allows us to derive a projection idea for compressible flow phenomena.

Since the density and the viscosity are smoothed according to (2.1) near the interface, we can rewrite the momentum equations (2.3) whose spatial components are given by

$$\frac{D\boldsymbol{v}}{Dt} = \boldsymbol{g} - \frac{1}{\tilde{\rho}}\left[\nabla p + (\nabla\tilde{\mu}\cdot\nabla)\boldsymbol{v} + \tilde{\mu}\Delta\boldsymbol{v} + \nabla\boldsymbol{v}\cdot\nabla\tilde{\mu} - \frac{2}{3}\nabla\tilde{\mu}(\nabla\cdot\boldsymbol{v}) + \frac{1}{3}\tilde{\mu}\nabla(\nabla\cdot\boldsymbol{v})\right]$$

The above presented equations are to be solved with appropriate initial and boundary conditions which are specified in the section where numerical tests are performed.

3 Numerical Scheme

We consider Chorin's projection method [Cho68] implicitly for both compressible as well as incompressible flows. It consists of two fractional steps. We first compute the new particle positions at time level t^{n+1} by

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta t \boldsymbol{v}^n. \tag{3.1}$$

Then, for each particle, we compute the smoothed density $\tilde{\rho}^{n+1}$ and viscosity $\tilde{\mu}^{n+1}$ according to (2.1) and then compute the intermediate velocity $\boldsymbol{v}^*$ by

$$\begin{aligned} \boldsymbol{v}^* + \frac{\Delta t}{\tilde{\rho}} \left[(\nabla \tilde{\mu} \cdot \nabla) \boldsymbol{v}^* + \tilde{\mu} \Delta \boldsymbol{v}^* \right] = \\ \boldsymbol{v}^n - \frac{\Delta t}{\tilde{\rho}} \left[\nabla \boldsymbol{v}^n \cdot \nabla \tilde{\mu} - \frac{2}{3} \nabla \tilde{\mu} (\nabla \cdot \boldsymbol{v}^n) + \frac{1}{3} \tilde{\mu} \nabla (\nabla \cdot \boldsymbol{v}^n) \right] + \Delta t \boldsymbol{g}. \end{aligned} \tag{3.2}$$

Here, Δt represents the time step, $\Delta \boldsymbol{v}$ is the Laplacian of $\boldsymbol{v}$, $\tilde{\rho} = \tilde{\rho}^{n+1}$ and $\tilde{\mu} = \tilde{\mu}^{n+1}$.

The second step consists in establishing the new velocity $\boldsymbol{v}^{n+1}$ by correcting the intermediate velocity $\boldsymbol{v}^*$. For this, we need to solve the equation

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^* - \frac{\Delta t}{\tilde{\rho}} \nabla p^{n+1} \tag{3.3}$$

with the constraints

$$\nabla \cdot \boldsymbol{v}^{n+1} = -\frac{1}{\tilde{\rho}} \frac{D\tilde{\rho}}{Dt} \tag{3.4}$$

with respect to p^{n+1}. By applying the divergence operator to equation (3.3), and using the relation (2.6), we obtain

$$-\frac{1}{\tilde{\rho}} \left(\frac{\partial \tilde{\rho}}{\partial p} \frac{Dp}{Dt} + \frac{\partial \tilde{\rho}}{\partial T} \frac{DT}{Dt} \right) = \nabla \cdot \boldsymbol{v}^* - \Delta t \nabla \cdot \left(\frac{\nabla p^{n+1}}{\tilde{\rho}} \right). \tag{3.5}$$

Now, using $\frac{Dp}{Dt} = \frac{p^{n+1} - p^n}{\Delta t}$, equation (3.5) can be expressed in the form

$$\frac{-1}{\tilde{\rho} \Delta t^2} \frac{\partial \tilde{\rho}}{\partial p} p^{n+1} + \nabla \cdot \left(\frac{\nabla p^{n+1}}{\tilde{\rho}} \right) = \frac{1}{\Delta t} \left(\frac{-1}{\tilde{\rho}} \frac{\partial \tilde{\rho}}{\partial p} p^n + \frac{\partial \tilde{\rho}^n}{\partial T^n} \frac{DT^n}{Dt} + \nabla \cdot \boldsymbol{v}^* \right). \tag{3.6}$$

Using the quotient rule for the second term on the left hand side of (3.6), we obtain

$$\frac{-1}{\Delta t^2} \frac{\partial \tilde{\rho}}{\partial p} p^{n+1} - \frac{1}{\tilde{\rho}} \nabla \tilde{\rho} \cdot \nabla p^{n+1} + \Delta p^{n+1} = \frac{\tilde{\rho}}{\Delta t} \left(\frac{-1}{\tilde{\rho}} \frac{\partial \tilde{\rho}}{\partial p} p^n + \frac{\partial \tilde{\rho}^n}{\partial T^n} \frac{DT}{Dt} + \nabla \cdot \boldsymbol{v}^* \right). \tag{3.7}$$

On the right hand side of (3.7), it is obvious to replace $\frac{DT}{Dt}$ by equation (2.4).

The boundary condition for p^{n+1} is obtained by projecting the equation (3.3) on the unit normal vector $\boldsymbol{n}$ to the boundary Γ. Thus, we obtain the Neumann boundary condition

$$\frac{\partial p^{n+1}}{\partial \boldsymbol{n}} = -\frac{1}{\Delta t} (\boldsymbol{v}_\Gamma^{n+1} - \boldsymbol{v}_\Gamma^*) \cdot \boldsymbol{n}, \tag{3.8}$$

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

$$\frac{\partial p^{n+1}}{\partial \boldsymbol{n}} = 0 \tag{3.9}$$

on Γ. If no boundary velocity $\boldsymbol{v}_\Gamma$ is known a priori (i.e. if the velocity at the boundary is a result of the computations itself, such as for free surfaces or outflow boundaries), the Dirichlet boundary conditions are appropriate for p^{n+1}.

In the case of incompressible flow, the first term on the left hand side and the first and second terms of the right hand side of equations (3.7) vanish, which results in the classical projection idea of Chorin.

Furthermore, for the temperature we have to solve the following equation

$$T^{n+1} - \frac{\Delta t}{c_v \rho^n} \nabla \cdot \left(\kappa \nabla T^{n+1} \right) = T^n + \frac{\Delta t}{c_v \rho^n} \left[-p^n \nabla \cdot \boldsymbol{v}^n + \left(\sigma(\boldsymbol{v}^n) \cdot \nabla \right) \boldsymbol{v}^n \right]. \tag{3.10}$$

Finally, we update the density for the compressible flow by

$$\rho^{n+1} = \rho(p^{n+1}, T^{n+1}). \tag{3.11}$$

The remaining task is the discretization and solution of the equations (3.2, 3.7, 3.10) on the given (meshfree) point cloud. For this, we establish big linear systems of equations, where the matrix represents the discrete approximation of the differential operators involved, and the right hand side reflects the source terms. In order to establish the mentioned discrete operators with respect to the point cloud, we employ the weighted least squares method, presented in section 3.1.

3.1 Least Squares Method for Approximation of Derivatives

Let $\psi : \Omega \longrightarrow \mathbb{R}$ be a scalar function and ψ_i its discrete values at the particle positions $\boldsymbol{x}_i$ for $i = 1, 2, \ldots, N$. Consider the problem to approximate spatial derivatives of that particular function $\psi(\boldsymbol{x})$ at some particle position $\boldsymbol{x}$ based on the discrete function values of its neighbor points. In order to restrict the number of points we introduce a weight function $w = w(\boldsymbol{x}_i - \boldsymbol{x}; h)$ with small compact support, where h determines the size of the support.

The weight function can be quite arbitrary, however it makes sense to choose a Gaussian weight function of the form

$$w(\boldsymbol{x}_i - \boldsymbol{x}; h) = \begin{cases} exp(-\alpha \frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|^2}{h^2}), & \text{if } \frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|}{h} \leq 1 \\ 0, & \text{else,} \end{cases}$$

where α is a positive constant and is considered to be in the range of 6. 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 as they become too dense.

Let $P(\boldsymbol{x}, h) = \{\boldsymbol{x}_i : i = 1, 2, \ldots, m\}$ be the set of m neighbor points of $\boldsymbol{x} = (x, y, z)$ in a ball of radius h. We note that the central particle $\boldsymbol{x}$ is one element of the neighbor set $P(\boldsymbol{x}, h)$. For consistency reasons, some obvious restrictions are required, for example, in $3D$ there should be at least 9 particles in addition to the central point and they should neither be on the same line nor on the same circle. In the following, we derive the Least Squares Method for three dimensional problems.

We determine the derivatives of a function by using the Taylor series expansion and the least squares approximation. Hence, consider m Taylor expansions of $\psi(\boldsymbol{x_i})$ about $\boldsymbol{x}$

$$\psi(\boldsymbol{x}_i) = \psi(\boldsymbol{x}) + \sum_{j=1}^{m} \frac{\partial \psi^{|j|}}{\partial x^{j_1} \partial y^{j_2} \partial z^{j_3}} \frac{1}{j!} (x_i - x)^{j_1} (y_i - y)^{j_2} (z_i - z)^{j_3} + e_i, \quad (3.12)$$

for $i = 1, \ldots, m$, where e_i is the error in Taylor's expansion at the point $\boldsymbol{x}_i$.

Denote the coefficients

$$a_1 = \tfrac{\partial \psi}{\partial x},\ a_2 = \tfrac{\partial \psi}{\partial y},\ a_3 = \tfrac{\partial \psi}{\partial z},\ a_4 = \tfrac{\partial^2 \psi}{\partial x^2},\ a_5 = \tfrac{\partial^2 \psi}{\partial x \partial y},$$
$$a_6 = \tfrac{\partial^2 \psi}{\partial x \partial z},\ a_7 = \tfrac{\partial^2 \psi}{\partial y^2},\ a_8 = \tfrac{\partial^2 \psi}{\partial y \partial z},\ a_9 = \tfrac{\partial^2 \psi}{\partial z^2}.$$

Let us assume that $\psi(\boldsymbol{x}) = \psi$ is the known discrete function value at the particle position $\boldsymbol{x}$. For $m > 9$, this system is overdetermined with respect to the unknowns a_i and can be rewritten as

$$\boldsymbol{e} = M\boldsymbol{a} - \boldsymbol{b}, \quad (3.13)$$

where

$$M = \begin{pmatrix} dx_1 & dy_1 & dz_1 & \frac{1}{2}dx_1^2 & dx_1 dy_1 & dx_1 dz_1 & \frac{1}{2}dy_1^2 & dy_1 dz_1 & \frac{1}{2}dz_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ dx_m & dy_m & dz_m & \frac{1}{2}dx_m^2 & dx_m dy_m & dx_m dz_m & \frac{1}{2}dy_m^2 & dy_m dz_m & \frac{1}{2}dz_m^2 \end{pmatrix},$$

$\boldsymbol{a} = (a_1, a_2, \ldots a_9)^T$, $\boldsymbol{b} = (\psi_1 - \psi, \ldots, \psi_m - \psi)^T$, $\boldsymbol{e} = (e_1, \ldots, e_m)^T$ and $dx_i = x_i - x,\ dy_i = y_i - y, dz_i = z_i - z$.

The unknowns a_i are computed by minimizing a weighted error over the neighboring points. Thus, we have to minimize the following quadratic form

$$J = \sum_{i=1}^{m} w_i e_i^2 = (M\boldsymbol{a} - \boldsymbol{b})^T W (M\boldsymbol{a} - \boldsymbol{b}) \quad (3.14)$$

with $W = \mathrm{diag}\,(w_1, \ldots, w_m)$, where $w_i = w(\boldsymbol{x}_i - \boldsymbol{x}; h)$. The minimization of J with respect to $\boldsymbol{a}$ formally yields (if $M^T W M$ is nonsingular)

$$\boldsymbol{a} = (M^T W M)^{-1} (M^T W) \boldsymbol{b}. \quad (3.15)$$

3.2 Least Squares Method for Solving Elliptic Equations

We now consider the following linear second order differential model equation, which represents all equations in the above presented projection scheme

$$A\psi + \boldsymbol{B} \cdot \nabla\psi + C\Delta\psi = f, \tag{3.16}$$

where the coefficients $A, \boldsymbol{B}, C$ are given and real and $f = f(\boldsymbol{x})$ is a given real valued function. We solve this equation with Dirichlet $\psi = \phi$ or Neumann boundary conditions

$$\frac{\partial\psi}{\partial\boldsymbol{n}} = \phi \quad \text{on} \quad \Gamma. \tag{3.17}$$

In the following, we demonstrate the method to solve (3.16-3.17). To our knowledge, there are two types of methods of directly solving elliptic equations in a given meshfree configuration. The first one is presented in [LO80], which can be directly derived from the equation (3.15). The second one is presented in [TK01], where equations (3.16) and (3.17) are added as constraints in the least squares approximation. The comparisons of both methods are presented in [IT02]. It is found that the method presented in [TK01] is more stable and the Neumann boundary condition can be easily included in the approximation. In this paper, we give a short overview about the method presented in [TK01].

We consider $\boldsymbol{x}$ as a central particle and its set of neighbors $P(\boldsymbol{x}, h) = \{\boldsymbol{x}_i : i = 1, 2, \ldots, m\}$. Furthermore, we consider the above Taylor's expansions (3.12). In (3.12) we have assumed that $\psi(\boldsymbol{x}) = \psi$ is a known discrete function value at $\boldsymbol{x}$. Now, let us assume that ψ is not known and denote it by a_0.

We add equations (3.16) and (3.17) as constraints into the m Taylor's expansions (3.12) . These two additional equations is rewritten in the following forms

$$Aa_0 + B_1a_1 + B_2a_2 + B_3a_3 + C(a_4 + a_7 + a_9) = f \tag{3.18}$$

$$n_1a_1 + n_2a_2 + n_3a_3 = \phi, \tag{3.19}$$

where $\boldsymbol{B} = (B_1, B_2, B_3), \boldsymbol{n} = (n_1, n_2, n_3)$. Note that, for the Dirichlet boundary condition, we have only the $m + 1$ equations, where we directly prescribe the boundary conditions on the boundary particles. The matrix M and vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{e}$ are slightly different from above. They are given by

$$\tilde{M} = \begin{pmatrix} 1 & dx_1 & dy_1 & dz_1 & \frac{1}{2}dx_1^2 & dx_1dy_1 & dx_1dz_1 & \frac{1}{2}dy_1^2 & dy_1dz_1 & \frac{1}{2}dz_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & dx_m & dy_m & dz_m & \frac{1}{2}dx_m^2 & dx_mdy_m & dx_mdz_m & \frac{1}{2}dy_m^2 & dy_mdz_m & \frac{1}{2}dz_m^2 \\ A & B_1 & B_2 & B_3 & C & 0 & 0 & C & 0 & C \\ 0 & n_1 & n_2 & n_3 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and by

$$\tilde{\boldsymbol{a}} = (a_0, a_1, a_2, \ldots, a_9)^T, \ \tilde{\boldsymbol{b}} = (\psi_1, \ldots, \psi_m, f, \phi)^T, \\ \text{and} \qquad \tilde{\boldsymbol{e}} = (e_1, \ldots, e_m, e_{m+1}, e_{m+2})^T.$$

Now, we minimize the functional

$$\tilde{J} = \sum_{i=1}^{m+2} w_i e_i^2, \tag{3.20}$$

where $e_{m+1} = (A\psi + \boldsymbol{B} \cdot \nabla\psi + C\Delta\psi - f)$, $e_{m+2} = \left(\frac{\partial\psi}{\partial \boldsymbol{n}} - \phi\right)$ and $w_{m+1} = w_{m+2} = 1$.

Similarly, the minimization of $\tilde{J}$ yields

$$\tilde{\boldsymbol{a}} = (\tilde{M}^T\tilde{W}\tilde{M})^{-1}(\tilde{M}^T\tilde{W})\tilde{\boldsymbol{b}} \tag{3.21}$$

with $\tilde{W} = \text{diag}\,(w_1, \ldots, w_m, 1, 1)$.

The vector $(\tilde{M}^T\tilde{W})\tilde{\boldsymbol{b}}$ is explicitely given by

$$\begin{aligned}(\tilde{M}^T\tilde{W})\tilde{\boldsymbol{b}} = \Bigg(&\sum_{i=1}^{m} w_i\psi_i + Af,\ \sum_{i=1}^{m} w_i dx_i\psi_i + B_1 f + n_1\phi,\\ &\sum_{i=1}^{m} w_i dy_i\psi_i + B_2 f + n_2\phi,\ \sum_{i=1}^{m} w_i dz_i\psi_i + B_3 f + n_3\phi,\\ &\frac{1}{2}\sum_{i=1}^{m} w_i dx_i^2\psi_i + Cf,\ \sum_{i=1}^{m} w_i dx_i dy_i\psi_i,\ \sum_{i=1}^{m} w_i dx_i dz_i\psi_i,\\ &\frac{1}{2}\sum_{i=1}^{m} w_i dy_i^2\psi_i + Cf,\ \sum_{i=1}^{m} w_i dy_i dz_i\psi_i,\ \frac{1}{2}\sum_{i=1}^{m} w_i dz_i^2\psi_i + Cf\Bigg)^T.\end{aligned}$$

Let $\beta_0, \beta_1, \ldots, \beta_9$ be the first row of the matrix $(\tilde{M}^T\tilde{W}\tilde{M})^{-1}$. We are looking for the function $\psi = a_0$, therefore, equating the first components of vectors on both sides of (3.21), we obtain

$$\begin{aligned}\psi = \beta_0&\left(\sum_{i=1}^{m} w_i\psi_i + Af\right) + \beta_1\left(\sum_{i=1}^{m} w_i dx_i\psi_i + B_1 f + n_1\phi\right) +\\ \beta_2&\left(\sum_{i=1}^{m} w_i dy_i\psi_i + B_2 f + n_2\phi\right) + \beta_3\left(\sum_{i=1}^{m} w_i dz_i\psi_i + B_3 f + n_3\phi\right) +\\ &\beta_4\left(\frac{1}{2}\sum_{i=1}^{m} w_i dx_i^2\psi_i + Cf\right) + \beta_5\left(\sum_{i=1}^{m} w_i dx_i dy_i\psi_i\right) +\\ &\beta_6\left(\sum_{i=1}^{m} w_i dx_i dz_i\psi_i\right) + \beta_7\left(\frac{1}{2}\sum_{i=1}^{m} w_i dy_i^2\psi_i + Cf\right) +\\ &\beta_8\left(\sum_{i=1}^{m} w_i dy_i dz_i\psi_i\right) + \beta_9\left(\frac{1}{2}\sum_{i=1}^{m} w_i dz_i^2\psi_i + Cf\right).\end{aligned}$$

Rearranging the terms, we have

$$\psi - \sum_{i=1}^{m} w_i \left(\beta_0 + \beta_1 dx_i + \beta_2 dy_i + \beta_3 dz_i + \beta_4 \frac{dx_i^2}{2} + \beta_5 dx_i dy_i + \right.$$
$$\left. + \beta_6 dx_i dz_i + \beta_7 \frac{dy_i^2}{2} + \beta_8 dy_i dz_i + \beta_9 \frac{dz_i^2}{2} \right) \psi_i =$$
$$(\beta_0 A + \beta_1 B_1 + \beta_2 B_2 + \beta_3 B_3 + \beta_4 C + \beta_7 C + \beta_9 C)\, f +$$
$$+ (\beta_1 n_1 + \beta_2 n_2 + \beta_3 n_3)\, \phi.$$

Hence, if we consider $\boldsymbol{x}_j$ an arbitrary particle and $\boldsymbol{x}_{j_i}$ its neighbors of number $m(j)$, then we have the following sparse system of equations for the unknowns $\psi_j, j = 1, \ldots, N$

$$\psi_j - \sum_{i=1}^{m(j)} w_{j_i} \left(\beta_0 + \beta_1 dx_{j_i} + \beta_2 dy_{j_i} + \beta_3 dz_{j_i} + \beta_4 \frac{dx_{j_i}^2}{2} + \beta_5 dx_{j_i} dy_{j_i} + \right.$$
$$\left. \beta_6 dx_{j_i} dz_{j_i} + \beta_7 \frac{dy_{j_i}^2}{2} + \beta_8 dy_{j_i} dz_{j_i} + \beta_9 \frac{dz_{j_i}^2}{2} \right) \psi_{j_i} =$$
$$[\beta_0 A + \beta_1 B_1 + \beta_2 B_2 + \beta_3 B_3 + (\beta_4 + \beta_7 + \beta_9) C]\, f_j +$$
$$(\beta_1 n_1 + \beta_2 n_2 + \beta_3 n_3)\, \phi_j. \qquad (3.22)$$

We can represent the above sparse system in compact matrix-vector-form as

$$A\boldsymbol{\Psi} = \boldsymbol{b}. \qquad (3.23)$$

Hence, (3.23) is a big sparse linear system of equations and can be solved using iterative methods. In this paper we have used SOR method.

Now back to the projection method for the Navier-Stokes equations, we have $d + 2$ such iterative systems, where d is the number of space dimension. As an initial guess for the iterative solvers at time level $n + 1$, we assume the corresponding value at the time level n. This saves a lot of iteration steps.

4 Numerical Tests

4.1 Solutions of Second Order Linear PDEs

Example 1

Consider the second order partial differential equation of the type

$$\psi + \psi_x + \psi_y + \psi_{xx} + \psi_{yy} = f \qquad \text{in} \qquad [0, 1] \times [0, 1]. \qquad (4.1)$$

The exact solution is given by

$$\psi = (x - \frac{1}{2})(y - \frac{1}{2})(1 - \frac{x^2}{2} - \frac{y^2}{2}). \qquad (4.2)$$

We consider the Dirichlet boundary value problems, where the boundary conditions can be directly obtained from the exact solution.

Table 4.1 shows the maximum error between the exact and the numerical solutions and shows second order convergence.

Table 4.1. Convergence results for Example 1.

N	Smoothing length	Maximum Error
676	0.1	0.00011
2601	0.05	2.7867e-5
10201	0.025	6.9778e-6

Example 2

Consider the following equation with discontinuous coefficient

$$\psi + \nabla \cdot (k \nabla \psi) = f \qquad \text{in} \qquad [0,1] \times [0,1], \tag{4.3}$$

where

$$k = \begin{cases} 1000, & \text{if } y \geq 0.5 \\ 1, & \text{else.} \end{cases}$$

Consider the exact solution

$$\psi = \frac{1}{k}(x - \frac{1}{2})(y - \frac{1}{2})(1 - \frac{x^2}{2} - \frac{y^2}{2}). \tag{4.4}$$

We again consider the Dirichlet boundary conditions. In this example, the source is given by

$$f = -(3x - \frac{1}{2})(y - \frac{1}{2}) - (x - \frac{1}{2})(3y - \frac{1}{2}) \tag{4.5}$$

and we have smoothed three times the coefficient k in the vicinity of the interface $y = 0.5$. The smooth coefficient k is denoted by $\tilde{k}$. Hence the above equation (4.3) is given by

$$\psi + \nabla \tilde{k} \cdot \nabla \psi + \tilde{k} \Delta \psi = f. \tag{4.6}$$

In Table 4.2 we have presented the maximum error between the exact and numerical solutions and we see that the numerical solution converges with of order one. If we use the interface conditions, we get the second order convergence [IT02].

Table 4.2. Convergence results for Example 2.

N	Smoothing length	Maximum Error
676	0.1	0.0691
2601	0.05	0.0377
10201	0.025	0.0203

4.2 Compressible Flows

Consider the 1d compressible flow. We note that, we do not consider smoothed density $\tilde{\rho}$ here. The viscosity is considered to be a constant number. Suppose the flow is ideal gas where the equation of state is given by

$$p = \rho RT = \frac{R}{c_v}\rho e,$$

where R is the gas constant and e the internal energy. Hence, we have

$$\nabla \cdot \boldsymbol{v} = \frac{1}{T}\frac{dT}{dt} - \frac{1}{p}\frac{dp}{dt}. \tag{4.7}$$

We have further assumed the dynamic viscosity μ and the heat conductivity coefficient κ to be constant. The scheme is tested by solving the Sod problem [Sod78], where we let the heat conductivity and viscosity tend to zero such that the solution of the Navier-Stokes equations converges to those of the Euler equations.

In this case, equation (3.2) is given by

$$v^* - \Delta t\frac{4}{3}\frac{\mu}{\rho}\frac{\partial^2 v^*}{\partial x^2} = v^n \tag{4.8}$$

with boundary condition $\boldsymbol{v}^* = 0$. Equation (3.7) is given by

$$-\frac{\rho^n}{\Delta t^2 p^n}p^{n+1} - \frac{1}{\rho^n}\frac{\partial \rho^n}{\partial x}\frac{\partial p^{n+1}}{\partial x} + \frac{\partial^2 p^{n+1}}{\partial x^2} = -\frac{\rho^n}{\Delta t^2} + \rho^n\frac{\nabla \cdot v^*}{\Delta t} - \frac{\rho^n}{T\Delta t}\frac{dT}{dt}, \tag{4.9}$$

where $\frac{dT}{dt}$ is replaced by the energy equation (2.4). Equation (4.9) is solved together with the Neumann boundary condition (3.9). After obtaining the pressure, we correct the velocity $\boldsymbol{v}^*$ according to (3.3).

Equation (3.10) is given by

$$T^{n+1} - \frac{\Delta t}{c_v}\frac{\kappa}{\rho^n}\frac{\partial^2 T^{n+1}}{\partial x^2} = T^n + \frac{\Delta t}{\rho^n c_v}\left(-p^n\frac{\partial v^n}{\partial x} + \frac{4}{3}\mu(\frac{\partial v^n}{\partial x})^2\right) \tag{4.10}$$

with boundary conditions $T = T_0$ on Γ. Finally, we update the density according to (2.5).

Let the domain $\Omega = (0, 1)$ with boundary points 0 and 1. We consider the discretization over $[0, 1]$ with N particles at x_i for $i = 1, 2, \ldots, N$ and constant time step Δt.

The initial conditions for Sod's problem are [Sod78]

$$\rho_i^0 = 1, \;\; v_i^0 = 0, \;\; e_i^0 = 2.5 \;\text{ for }\; 0 \le x_i < 0.5$$

$$\rho_i^0 = 0.125, \;\; v_i^0 = 0, \;\; e_i^0 = 2.0 \;\text{ for }\; 0.5 \le x_i \le 1$$

completed by the following boundary conditions

$$v(t) = 0,\ e(t) = 2.5 \ \text{ at } \ x = 0 \ \text{ and } \ v(t) = 0,\ e(t) = 2 \ \text{ at } \ x = 1.$$

We note that the initial and boundary conditions for the temperature T is obtained from the relation

$$T = \frac{1}{c_v} e.$$

The initial spacing of the particles is given by $dx = 1/N$, where N is the total number of particles considered and the size of the support h is equal to 3 times the initial spacing of the particles.

Since the scheme is of central difference type, there are some oscillations for small viscosity. Therefore, we choose $\mu = \mu(N),\ \kappa = \kappa(N)$. For example, for $N = 100$, we considered $\mu = 0.001$ and $\kappa = 0.001$. The heat coefficient does not play big role for the stability of the scheme. It can be set to zero.

Since we solve the conservation equations implicitly, we need the restriction of time step only for the motion of the particles. The time step is should be chosen such that the particles cannot move more than a partition of h in each time step.

The numerical solutions are obtained for 100, 400 and 2000 particles and are compared with the exact solutions of the compressible Euler equation at the fixed time $t = 0.2$. The values for *mu* and *kappa* are for 400 and 2000 particles are four and twenty times smaller than thoose for 100 particles. The time step for 100 particles is chosen 0.002. Similarly, for 400 and 2000 particles the time smaller step is taken by corresponding factors.

In figures 4.1 we plot the exact and numerical results, like the density, velocity and the pressure. It is clear that the scheme is stable and the solutions of the Navier-Stokes equations tend to the Euler solutions when the number of particles tends to infinity and the viscosity and the heat conduction coefficient tend to zero.

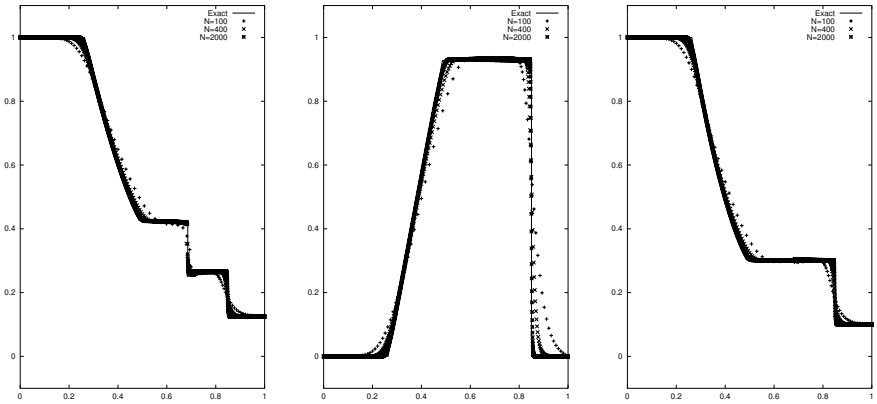

Figure 4.1. Density(left), Velocity(center) and Pressure (right) at $t = 0.2$. Solid lines represent the exact solutions, dots represent the numerical solutions.

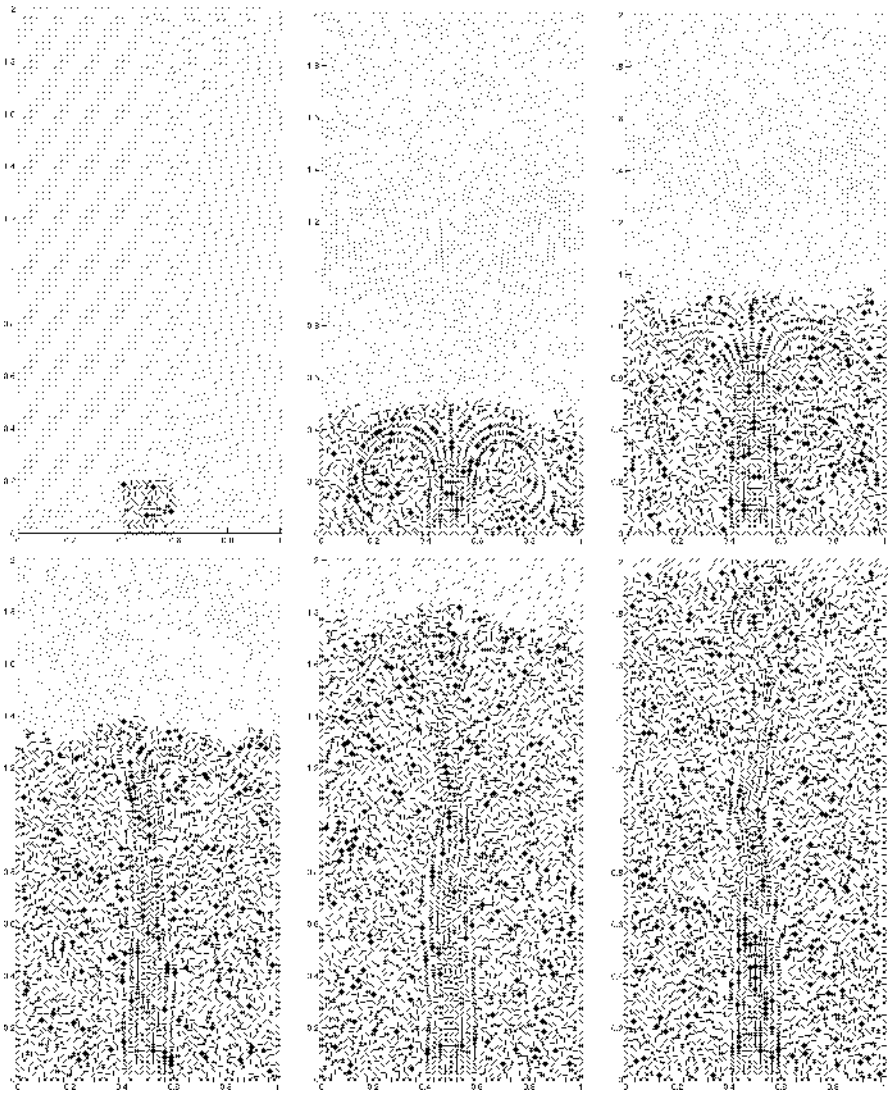

Figure 4.2. Level of water at $t = 0.0, 1.0, 2.0$ (top row, left to write) and at $t = 3.0, 4.0, 4.4$ (bottom row), star are water particles and dots are gas particles

4.3 Two Phase Flows

In this case we express all quantities in dimensionless form. So, they can be interpreted as being in SI-units. Consider a cavity of $[0, 1] \times [0, 2]$ initially filled with air. On the center of the lower boundary we place a hole as inflow

boundary. Similarly, there is an outflow hole in the top boundary. The width of the inflow and outflow holes is 0.2. The rest of the boundaries are solid walls with no slip conditions. The inflow velocity is 2. There is gravity acting downwards with $g = 9.81$. The densities of air and water are 1 and 1000 respectively. The dynamic viscosity of air is $1.81 * 10^{-5}$ and $1.005 * 10^{-3}$ for water. We consider the air as compressible and the equation of state is given by

$$\rho = \rho_0 + \frac{1}{c^2}(p - p_0) \tag{4.11}$$

with speed of sound $c = 5$ and the reference density ρ_0 and reference pressure p_0 are initial density and pressure, equal to 1 and 0, respectively. The time step $\Delta t = 0.002$ is considered. The theoretical fill time for this particular cavity is 5. Since fill time for initially replaced liquid is 0.1, therefore, the fill time for this case is 4.9.

In Figure 4.2 we have plotted the filling process for different times. At time 4.4 we stopped the simulation since the liquid particles started to leave the outflow boundary. The numerical result shows close approximation of the theoretical fill time.

References

[BKZ92] Brackbill J. U., Kothe D. B. and Zemach C.:: A continuum method for modeling surface tension, J. Comput. Phys., 100, 335–354 (1992)

[Cho68] Chorin A.: Numerical solution of the Navier-Stokes equations, J. Math. Comput. vol. 22, 745–762 (1968)

[CR99] Cummins S. J. and Rudmann M.: An SPH Projection Method, J. Comput. Phys., 152, 284–607 (1999)

[Dil96] Dilts G. A.: Moving least squares particle hydrodynamics I, consistency and stability, Hydrodynamics methods group report, Los Alamos National Laboratory (1996)

[GM97] Gingold R. A., Monaghan J. J.: Smoothed particle hydrodynamics: theory and application to non-spherical stars, Mon. Not. Roy. Astron. Soc., 181, 375–389 (1997)

[HJKT03] Hietel D., Junk M., Kuhnert J. and Tiwari S.: Meshless methods for Conservation Laws, preprint ITWM (2003)

[GW01] Ginzburg I., Wittum G.: Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolant, J. Comp. Phys., 166, 302–335 (2001)

[HN81] Hirt C. W., Nichols B. D.: Volume of fluid (VOF) method for dynamic of free boundaries, J. Comput. Phys., 39, 201 (1981)

[HW65] Harlow F. H., Welch J. E.: Numerical study of large amplitude free surface motions, Phys. Fluids, 8 (1965)

[IT02] Iliev O., Tiwari S.: A generalized (meshfree) finite difference discretization for elliptic interface problems, Numerical Methods and Applications, Lecture notes in Computer Sciences, I. Dimov, I. Lirkov, S. Margenov, Z. Zlatev (Eds), Springer Verlag, 480–489 (2002)

[KP97] 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., 138, 939 (1997)

[Ku99] Kuhnert J.: General smoothed particle hydrodynamics, Ph.D. thesis, Kaiserslautern University, Germany, (1999)

[Ku02] Kuhnert J.: An upwind finite pointset method for compressible Euler and Navier-Stokes equations, Springer LNCSE: Meshfree methods for Partial Differential Equations, Vol. 26, M. Griebel, M. A. Schweitzer (Eds) (2002)

[KTU00] Kuhnert J., Tramecon A., Ullrich P.: Advanced Air Bag Fluid Structure Coupled Simulations applied to out-of Position Cases, EUROPAM Conference Proceedings, ESI group, Paris, France (2000)

[LO80] Liszka T., Orkisz J.: The finite difference method on arbitrary irregular grid and its application in applied mechanics, Computers & Structures, 11, 83–95 (1980)

[Luc77] Lucy L. B.: A numerical approach to the testing of the fission hypothesis, Astron. J., 82, 1013 (1977)

[Mon94] Monaghan J. J.: Simulating free surface flows with SPH, J. Comput. Phys., 110, 399 (1994)

[Mor00] Morris J. P.: Simulating Surface Tension with Smoothed Particle Hydrodynamics, Int. J. Numer. Methods Fluids, 33, 333–353 (2000)

[MFZ97] Morris J. P., Fox P. J., Zhu Y.: Modeling Low Reynolds Number Incompressible Flows Using SPH, J. Comput. Phys., 136, 214–226 (1997)

[Sod78] Sod G. A: A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comp. Phys. 27, 1–31, (1978)

[TK01] Tiwari S., Kuhnert J.: Grid free method for solving Poisson equation, preprint, Berichte des Fraunhofer ITWM, Kaiserslautern, Germany, Nr. 25 (2001)

[TK102] Tiwari S., Kuhnert J.: Finite pointset method based on the projection method for simulations of the incompressible Navier-Stokes equations, Springer LNCSE: Meshfree methods for Partial Differential Equations, Vol. 26, M. Griebel, M. A. Schweitzer (Eds) (2002)

[TK202] Tiwari S., Kuhnert J.: A meshfree method for incompressible fluid flows with incorporated surface tension , revue aurope'enne des elements finis, Volume 11–nř 7–8, (Meshfree and Particle Based approaches in Computational Mechanics) (2002)

[TK03] Tiwari S., Kuhnert J.: Particle method for simulations of free surface flows, Hyperbolic Problems: Theory, Numerics, Application, Proceedings of the Ninth International Conference on Hyperbolic Problems, T. Y. Hou and E. Tadmor (Eds), Springer Verlag (2003)

[Tiw00] Tiwari S.: A LSQ-SPH approach for compressible viscous flows, International series of numerical mathematics Vol. 141, H. Freistuehler, G. Warnecke (Eds), Birkhaueser (2000)

[TM03] Tiwari S., Manservisi S.: Modeling incompressible Navier-Stokes flows by LSQ-SPH, Nepal Mathematical Sciences Report Vol 20 No 1 & 2 (2003)