JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
40, 4, pp. 917-937, Warsaw 2002

SPH COMPUTATION OF INCOMPRESSIBLE VISCOUS FLOWS

JACEK POZORSKI
ARKADIUSZ WAWREŃCZUK

Institute of Fluid Flow Machinery, Polish Academy of Sciences, Gdańsk
e-mail: jp@imp.gda.pl

The paper is concerned with the numerical flow computation using the method of Smoothed Particle Hydrodynamics (SPH). Fundamental concepts of the method are briefly recalled. Physical aspects of the incompressibility and its mathematical formulation are described. A new proposal for the incompressibility constraint is put forward in the SPH context. Numerical implementation of the method is described and main parts of the algorithm are presented, including the issue of boundary conditions. Some validation cases and examples of results for viscous flows are presented.

Key words: viscous flow, particle method, incompressibility

1. Introduction

Smoothed Particle Hydrodynamics (SPH) is a particle simulation method developed in astrophysics (Monaghan, 1992) where the classical computational fluid dynamics (Eulerian) methods are not sufficiently effective. Being a Lagrangian approach (and basically grid-free), the SPH is able to naturally treat highly-varying density, deformable boundaries and free surfaces, propagation of discontinuities, multiphase flows and other physically complex flow situations.

In the SPH, the flow continuum is discretised by using particles. The particles, which can be thought of as Lagrangian fluid elements, carry all the necessary information about flow variables; in this feature lies the main strength of the method. For a more detailed description of the SPH, see Monaghan (1992), Morris (1996), and Morris et al. (1997). From the very beginning, the

method finds more and more application areas, such as its native astrophysics, materials science (Libersky et al., 1993), and fluid mechanics, including free surface flows (Monaghan, 1994), creeping flows (Morris et al., 1997), shock waves (Monaghan and Gingold, 1983; Morris and Monaghan, 1997), and, partly (for computation of statistics only), also turbulent flows in the PDF approach (Welton, 1998).

The SPH has originally been formulated for compressible flows, with the pressure field linked to the fluid density through an equation of state. Incompressible flows have been computed with this compressible formulation, assuring that the flow velocity scale be a small fraction of the sound velocity (often using an artificial equation of state, cf. Morris et al. (1997)). An alternative, "truly" incompressible approach has first been proposed only recently by Cummins and Rudman (1999). In the authors' opinion, however, an important aspect of the incompressibility has to be accounted for in the governing equations of the SPH approach. Hence, a new formulation of the incompressible SPH is proposed in the present paper.

This work considers the aspects of implementation of the incompressibility in the context of the SPH method for viscous flows. The paper is structured as follows. The SPH method is introduced in Section 2. Then, the existing formulation of the incompressibility is recalled and a new proposal is detailed (Section 3). Next, we describe a numerical algorithm of the SPH method we have developed (Section 4). Some results for test cases illustrate the validation of parts of the algorithm (viscosity, incompressibility, boundary conditions); first computational studies of viscous incompressible flows are also reported (Section 5). Conclusions and perspectives for further works are contained in Section 6.

2. Basics of the formalism

Now we present some fundamental concepts of the SPH method. This will help to understand the "philosophy" of this approach and to get some flavour of its technical aspects; however, the review paper by Monaghan (1992) and references therein are recommended for a full coverage.

2.1. Summation interpolants

Formally any function $A(\boldsymbol{r})$ can be approximated by an integral interpolant $A_I(\boldsymbol{r})$

$$A_I(\boldsymbol{r}) = \int A(\boldsymbol{r}')W(\boldsymbol{r}-\boldsymbol{r}',h)\,d\boldsymbol{r}' \tag{2.1}$$

where $A(\boldsymbol{r}')$ are values of the function at points $\boldsymbol{r}'$ and W is a weighting function (so called kernel) with a parameter h that can be thought of as a smoothing length or a linear dimension of the kernel; the integration is over the entire space (or over the kernel support, if compact). Obviously, the approximation error is kernel-dependent, with $A_I(\boldsymbol{r}) - A(\boldsymbol{r}) = \mathcal{O}(h^2\nabla^2 A)$. Eq. (2.1) can be thought of as a coarse-graining formula for any hydrodynamic variable (e.g.density, velocity components, or energy). The kernel is a function that must satisfy the normalisation condition $\int W(\boldsymbol{r}-\boldsymbol{r}',h)\,d\boldsymbol{r}' = 1$ and, secondly, in the limit of small smoothing length, $h \to 0$, it must tend to the Dirac delta $W(\boldsymbol{r},h) \to \delta(\boldsymbol{r})$.

In particle methods, instead of the field $A(\boldsymbol{r})$, with its argument $\boldsymbol{r}$ being continuous in space, a fine-grained (particle) representation of A is introduced (cf. Gingold and Monaghan, 1982). The integral interpolant (2.1) becomes then a discrete, or summation, interpolant $A_S(\boldsymbol{r})$ defined by a sum over all particles b in the system, with the volume element replaced by the specific volume (the inverse of the particle number density)

$$A_S(\boldsymbol{r}) = \sum_b m_b \frac{A_b}{\rho_b} W(\boldsymbol{r}-\boldsymbol{r}_b,h) \tag{2.2}$$

where the index b denotes the values at the locations $\boldsymbol{r}_b$, m_b is the particle mass, ρ is density. In the following, we will skip the subscript "S" in the summation formulae. It is to be remembered that the variables $\boldsymbol{r}_b$, m_b are attached to the particles while the density ρ_b and A_b (which is a generic name for any variable describing the flow field: velocity, internal energy, etc.) should be understood as field values that are computable at *any* point $\boldsymbol{r}_b$ (in particular: at the particle location).

Now, the crucial point is that a differentiable approximation of a function can be constructed by using interpolants with the differentiable kernel. Such a concept allows one to avoid the use of a discrete mesh and finite difference formulae, which is otherwise unavoidable in many methods for solving partial and/or ordinary differential equations (finite volume, particle-mesh, particle in cell, to name only a few). For example, the gradient of A in Eq. (2.2) is

$$\nabla A_S(\boldsymbol{r}) = \sum_b m_b \frac{A_b}{\rho_b} \nabla W(\boldsymbol{r}-\boldsymbol{r}_b,h) \tag{2.3}$$

Any differential operator can be constructed in a similar way. Consequently, no finite difference approximations are needed to compute spatial derivatives because this operation is shifted to the kernel which is a known function.

Obviously, the kernel must be quite easily differentiable to make the method computationally efficient. The efficiency constraint also applies to the summation process. Formally, the sum is done over all particles, for the integration in (2.1) is over the whole flow domain. For a large number of particles, this can become prohibitively time-consuming. Therefore, the kernel is usually chosen as a function with a compact support; this limits the computation of sums to particles from some part of the whole domain only.

Remembering these constraints we can, in principle, construct a kernel of our choice, which adds to the flexibility of the method. In practical applications of the SPH, kernels are most often based on polynomial functions like the cubic spline (also used in the present work)

$$W(\boldsymbol{r},h) = C\begin{cases} 1-\dfrac{3}{2}q^2+\dfrac{3}{4}q^3 & \text{if} \quad 0 \leqslant q \leqslant 1 \\ \dfrac{1}{4}(2-q)^3 & \text{if} \quad 1 \leqslant q \leqslant 2 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

where $q = |\boldsymbol{r}|/h$ and C is a normalisation constant. This choice is convenient for numerical purposes: the kernel is sufficiently smooth for most applications, easy to compute, and has a compact support. Another possible choice is the Gaussian kernel $W(r,h) = C\exp(-r^2/h^2)$. The undesirable feature of this kernel is that the support is not compact, though values of W are very small for r greater than $3h$, say. The modified Gaussian kernel with some cut-off distance, quartic or quintic spline kernels are also used; generally, the choice often depends on the computor. Recently, a thorough stability analysis of the SPH equations with various kernels has been performed (Morris, 1996).

2.2. Governing equations

The SPH is a particle method; so the physical system is treated as built of particles (real or notional) and the whole set of governing evolution equations should be expressed in the SPH approach using the above-presented formalism. In other words, the dynamics of the system results from a certain form of particle interaction, responsible for any process occurring in the system we wish to describe, like momentum and energy transfer. Such an approach involves the calculation of the sums as (2.2) or (2.3). For viscous flows, the governing equations are as follows.

First, the continuity equation

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\boldsymbol{U} \tag{2.5}$$

can be discretely represented as

$$\frac{d\rho_a}{dt} = \sum_b m_b \boldsymbol{U}_{ab} \cdot \nabla_a W_{ab} \tag{2.6}$$

Alternatively, the fluid density can be computed from its definition formula

$$\rho_a = \sum_b m_b W_{ab} \tag{2.7}$$

In the above expressions, some shorthand symbols have been introduced: $W_{ab} = W(\boldsymbol{r}_{ab}, h)$, $\boldsymbol{r}_{ab} = \boldsymbol{r}_a - \boldsymbol{r}_b$ and $\boldsymbol{U}_{ab} = \boldsymbol{U}_a - \boldsymbol{U}_b$; moreover, ∇_a denotes the gradient operator with respect to $\boldsymbol{r}_a$.

Next, the momentum equation for the incompressible fluid (the Navier-Stokes eq.) is

$$\rho \frac{d\boldsymbol{U}}{dt} = \rho \boldsymbol{g} - \nabla P + \mu \nabla^2 \boldsymbol{U} \tag{2.8}$$

where P is the pressure, $\boldsymbol{g}$ is the mass forces, and μ is the dynamic viscosity. In the SPH formalism, it takes the form

$$\frac{d\boldsymbol{U}_a}{dt} = \boldsymbol{g}_a - \sum_b \Big[m_b \Big(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2} \Big) \nabla_a W_{ab} - \Pi_{ab} \Big] \tag{2.9}$$

where $\boldsymbol{g}_a = \boldsymbol{g}(\boldsymbol{r}_a)$. Moreover, Π_{ab} is the viscosity term equal to (Morris et al., 1997)

$$\Pi_{ab} = m_b \frac{(\mu_a + \mu_b) \boldsymbol{r}_{ab} \cdot \nabla_a W_{ab}}{\rho_a \rho_b (\boldsymbol{r}_{ab}^2 + \eta^2)} \boldsymbol{U}_{ab} \tag{2.10}$$

where a small correction η in the denominator (usually $\eta \sim 0.1h$) is meant to avoid the singularity (if needed for a given W). It should be noted that the above expression approximates the true (physical) viscosity term. It has been a common practice to use an artificial viscosity term in compressible SPH formulations, in order to obtain a better resolution of shock waves; this will not be considered here. Moreover, let $\boldsymbol{F}_{ab}$ denote the expression in the square bracket in Eq. (2.9). It is thus easily seen that particle momenta change due to the inter-particle interactions $\boldsymbol{F}_{ab}$ and external influences $\boldsymbol{g}_a$ (like mass force or imposed mean pressure gradient driving the flow).

In the following, we will assume that fluid properties (density and viscosity) are temperature-independent, so the equation for the internal (thermal) energy will not be needed here.

The particle equation of motion

$$\frac{d\boldsymbol{r}_a}{dt} = \boldsymbol{U}_a \tag{2.11}$$

completes the system. Although very simple, it implies that the convection is treated exactly in the SPH.

It is worth saying a few words about the smoothing length h. It determines the local spatial resolution and also specifies the number of particles treated as near neighbours. The choice of this quantity is determined by two opposing aspects. If h is too small, due to a small number of contributing particles the approximations tend to be noisy. On the other hand, if h is too large, the resulting fields become over-smoothed, and the computation of the interpolation sums over many particles is less efficient. A compromise usually consists in taking h equal to several times the average inter-particle distance.

3. Incompressibility in SPH

A simple way to present the concept of incompressibility is to imagine the fluid as a set of hard spheres (which represent the particles). Suppose these spheres are contained in some volume; by applying a force, we can then decrease the volume to some extent, limited by the size of the spheres. When the spheres touch each other, the system is in its most compact configuration. Now, if one particle is displaced, all other will react instantly in the whole system, immediately changing their positions. This implicates the infinite velocity of the transfer of information and the elliptic character of the governing equations.

The condition of constant density, $\rho = \text{const}$, put to continuity equation (2.5) results in

$$\nabla \cdot \boldsymbol{U} = 0 \tag{3.1}$$

The condition $\rho = \text{const}$ and Eq. (3.1) are a mathematical formulation of the fact that the system is incompressible. Basically, they can be numerically implemented in two ways. The first one uses the high sound speed which is the measure of the propagation of perturbations in the system. Roughly speaking: a non-uniform density will relax to its equilibrium value through the propagation of pressure waves that will drive the particle motion. The liquid equation of state is usually modified to avoid prohibitively small time steps due to the CFL constraint. This approach is close to the classical artificial compressibility method used in Eulerian solvers. The second approach (detailed below) is to enforce condition (3.1) through the projection method, as first proposed by Cummins and Rudman (1999).

3.1. Projection theory – zero-divergence constraint

The projection method enables introducing the incompressibility by casting the velocity field on the divergence-free space; it is possible because every vector field $\boldsymbol{A}$ can be written as the sum of a potential field and a solenoidal one

$$\boldsymbol{A} = \nabla\phi + \boldsymbol{A}_d \tag{3.2}$$

where $\nabla \cdot \boldsymbol{A}_d = 0$. So, the velocity field computed in the SPH at a current time step can be decomposed in the above way by finding the curl-free part of the field and leaving only the divergence-free part.

Practically, the decomposition is performed by finding the curl-free component from the momentum equation and subtracting it from the original vector field

$$\boldsymbol{U}_d = \boldsymbol{U} - \frac{\Delta t}{\rho}\nabla\delta P \tag{3.3}$$

As seen in Eq. (2.8), the velocity is linked with the pressure. We want basic incompressibility condition (3.1) to be fulfilled during all calculation. However, this condition is generally not satisfied at the beginning of each time step Δt. By local corrections to the pressure field $\delta P^{(1)}$, obtained from the equation

$$\nabla \cdot \frac{1}{\rho}\left(\nabla\delta P^{(1)}\right) = \frac{1}{\Delta t}\nabla \cdot \boldsymbol{U} \tag{3.4}$$

condition (3.1) can be fulfilled. The projection method is akin to the so-called pressure correction widely used in Eulerian solvers, too.

3.2. Constant density constraint

At the same time we must also remember that condition (3.1) comes from the assumption that $\rho =$ const in the whole domain. This equation is at the origin of the second correction term. The pressure field is related to the density, which should be constant. The density is calculated on the basis of the masses and positions of the particles. The masses are constant, so we have to change the positions to make the density uniform in the domain. The second correction, used also in the PDF computation of turbulent flows (Minier and Pozorski, 1999), relates to the deviations of density from the mean value (cf. Pope, 1985). In the context of the SPH, an analogous formula, Eq. (3.12) below, can be proven as follows. Consider a flow of an incompressible fluid with a constant density ρ_0 imposed at the beginning of the computation. At

any time instant t, the fluid density at any point $\boldsymbol{r}$ in the flow domain is computed out of the particles (we suppose they all are of equal mass m_0)

$$\rho(\boldsymbol{r},t) = m_0 \sum_b W(\boldsymbol{r} - \boldsymbol{r}_b(t)) \tag{3.5}$$

At the time step t^n, the fluid density $\rho^n(\boldsymbol{r})$ in the domain (possibly close to ρ_0) is known. Now, at $t^{n+1} = t^n + \Delta t$, a numerical integration scheme yields a set of particle velocities $\boldsymbol{U}_b^{n+1}$ and locations $\widetilde{\boldsymbol{r}}_b^{n+1}$. The flow density computed out of these locations is generally no longer uniform

$$\widetilde{\rho}^{n+1}(\boldsymbol{r}) = m_0 \sum_b W(\boldsymbol{r} - \widetilde{\boldsymbol{r}}_b^{n+1}) \neq \text{ const} \tag{3.6}$$

In order to satisfy the constant density constraint, the pressure gradient field $\nabla \delta P^{(2)}$ is computed and used to correct the particle locations at t^{n+1}

$$\boldsymbol{r}_b^{n+1} = \widetilde{\boldsymbol{r}}_b^{n+1} - \frac{1}{\rho_0} \nabla \delta P_b^{(2)} \frac{\Delta t^2}{2} \tag{3.7}$$

This pressure gradient is meant to produce the constant density at t^{n+1} again

$$\rho^{n+1}(\boldsymbol{r}) = m_0 \sum_b W(\boldsymbol{r} - \boldsymbol{r}_b^{n+1}) = \rho_0 = \text{ const} \tag{3.8}$$

Substituting (3.7) and expanding into the Taylor series around $\boldsymbol{r} - \widetilde{\boldsymbol{r}}_b^{n+1}$ yields

$$\begin{aligned} \rho_0 &= m_0 \sum_b W\Big(\boldsymbol{r} - \widetilde{\boldsymbol{r}}_b^{n+1} + \frac{\Delta t^2}{2\rho_0} \nabla \delta P_b^{(2)}\Big) = \\ &= m_0 \sum_b W(\boldsymbol{r} - \widetilde{\boldsymbol{r}}_b^{n+1}) + \frac{\Delta t^2}{2\rho_0} m_0 \sum_b \nabla \delta P_b^{(2)} \cdot \nabla W(\boldsymbol{r} - \widetilde{\boldsymbol{r}}_b^{n+1}) \end{aligned} \tag{3.9}$$

Consider now the SPH formula (cf. Eq. (2.3)) for the following derivative

$$\nabla \cdot \Big(\frac{\rho^n}{\rho_0} \nabla \delta P^{(2)}\Big) = m_0 \sum_b \frac{\rho_b^n \nabla \delta P_b^{(2)}}{\rho_0 \rho_b^n} \cdot \nabla W(\boldsymbol{r} - \widetilde{\boldsymbol{r}}_b^{n+1}) + \mathcal{O}(\Delta t) \tag{3.10}$$

it is exactly the expression appearing in the second RHS term of (3.9). Hence, using (3.6)

$$\rho_0 = \widetilde{\rho}^{n+1} + \frac{\Delta t^2}{2} \nabla \cdot \Big(\frac{\rho^n}{\rho_0} \nabla \delta P^{(2)}\Big) \tag{3.11}$$

The above formula directly yields the (generalised) Poisson equation for the pressure correction

$$\frac{1}{\rho_0}\nabla\cdot\Big(\frac{\rho^n}{\rho_0}\nabla\delta P^{(2)}\Big)=\frac{2}{\Delta t^2}\Big(1-\frac{\widetilde{\rho}^{n+1}}{\rho_0}\Big) \tag{3.12}$$

To sum up: the introduction of the incompressibility involves two Poisson's equations. The first one is the consequence of incompressibility condition (3.1) and the second one is related to the perturbations of the density. Given appropriate boundary conditions, the solution of these equations is unique. One only has to remember that the boundary-value problem consisting of the Poisson eq. $\nabla^2\delta P = q$ with homogeneous (zero) Neumann's condition is well posed when the additional compatibility constraint $\int q\,dV = 0$ is satisfied.

4. Numerical algorithm of the method

4.1. Generalities

Using the basic SPH equations recalled in Section 2, the zero-divergence constraint and the new proposal to preserve a constant particle density (Section 3), we constructed an algorithm to compute both incompressible and compressible flows. The algorithm can be written as follows:

- initialize the particles in the computational domain; create an auxiliary mesh
- for each time step:
  - update particle-mesh data (linked lists)
  - compute the densities, Eq. (2.7)
  - compute the pressures from the equation of state (*for compressible flows only*)
  - compute the right-hand sides of the equations of motion
  - advance the equations of motion by a time step, Eqs. (2.11) and (2.9)
  - solve the correction related to the density, Eq. (3.12), update particle positions
  - solve the correction related to the velocity divergence, Eq. (3.4), update particle velocities
- save and/or process the results.

The corrections to particle velocities and positions are computed only in the incompressible case which is of main interest for us at this time. The smoothing length h is chosen so that the number of neighbours within the kernel support be roughly in the range of 10-20.

Initially, the particles are located uniformly in the computational domain, either in a regular rectangular array or using the pseudorandom generator with the uniform distribution. The particle velocity components are assigned from given initial profiles.

4.2. Calculation of the interpolant sums

One of the important features of the SPH are the summation formulae obtained from the interpolation procedure. Each sum appearing in the governing equations must be calculated for every particle. Therefore, the summation procedure takes a considerable part of the actual simulation time. For this reason, a great amount of effort has been devoted to the increase of the effectiveness of this part of the algorithm. To this aim, an auxiliary mesh has been introduced and symmetry of the summation terms has been used.

As stated before, the SPH is basically a grid-free method. However, we use the auxiliary mesh to find the nearest neighbours (particles). With choice (2.4) of the kernel formula, the "nearest neighbour" should be meant as one for which the interparticle distance satisfies $r_{ab} \leqslant 2h$. For a regular mesh of size $2h$ superimposed onto the simulation area, it means that for a given particle only the particles from the same cell and the nearest neighbouring cells contribute to the sum.

To make the task easier, we use the so-called linked lists to store particle-mesh data (Hockney and Eastwood, 1981) using two integer arrays. The first one (of size equal to the number of cells) is called `fpc`, ie. *first-particle-in-cell.* In `fpc`(i) we store the index of the first particle found in the cell i; `fpc`(i) is equal to zero only if the cell i is empty. The second array (of size equal to the overall number of particles used in simulation), is called `spc`, ie. *subsequent-particles-in-cell.* For a particle n, the array element `spc`(n) stores the pointer to the particle (in the same cell) next to the particle n (if any). For example: suppose that particles 3 and 5 (out of five) are in the cell number 1, the remaining are in the cell number 2, and the cell number 3 is empty. We perform a loop over the particles to find their respective cell numbers, so `fpc`$(1) = 3$. The indices of the subsequent particles found in same cell are stored in `spc`, so in our example `spc`$(3) = 5$ (particle 3 "points" to particle 5 in the same cell) and `spc`$(5) = 0$ (particle 5 is the last in the cell). After assigning all

particles to the cells, the arrays in our example will look like `fpc`$= (3, 1, 0)$, `spc`$= (2, 4, 5, 0, 0)$.

The computational effort can also be considerably reduced by making use of the intrinsic symmetry of the terms of the interaction force between the particles. Because $F_{ab} = -F_{ba}$ for any pair of particles a and b, the two contributions can be computed at a time. We only have to assure that the summation over all particles is made in such a way that the force is computed only once for any two interacting particles. In order to do this, the summation is organised with a loop over the cells and not over the particles. This part of the algorithm is constructed as follows:

→ loop over cells

→ loop over particles in a given cell

- compute and store self-interaction term F_{aa} for a given particle a
- compute interactions with *subsequent* particles b in cell; store F_{ab} and F_{ba}
- compute interactions with particles b in *subsequent* cells; store F_{ab} and F_{ba}.

In practice, the above algorithm becomes somewhat more complicated because boundary conditions have to be accounted for (cf. below).

4.3. Time evolution

In the current approach, the first order explicit (Euler) scheme is used for solving ordinary differential equations (2.11) and (2.9) for time advancement. The time-discrete form of the evolution equation for velocity is

$$\boldsymbol{U}_a^{n+1} = \boldsymbol{U}_a^n - \Delta t \sum_b F_{ab} - \Delta t \nabla \delta P_a^{(1)} \tag{4.1}$$

time evolution of positions

$$\boldsymbol{r}_a^{n+1} = \boldsymbol{r}_a^n + \boldsymbol{U}_a^{n+1} \Delta t - \frac{\Delta t^2}{2} \nabla \delta P_a^{(2)} \tag{4.2}$$

with the pressure corrections $\delta P_a^{(1)}$ and $\delta P_a^{(2)}$ given by (3.4) and (3.12), respectively. For the explicit integration scheme, like the one applied, a constraint for the maximum allowed time step applies, and this should be checked in all simulations. The advantage of this kind of integration scheme as applied

to the particle method is that the problems with detecting particle collisions with walls and double application of boundary conditions (a debatable issue in predictor-corrector type schemes) are avoided. However, in unbounded domains or for flows with periodic boundary conditions, second-order schemes are definitely recommended. The integration scheme, however important, is not considered crucial at the actual stage of the development, but it should be improved to increase the computational efficiency.

4.4. Boundary conditions

Boundary conditions are vitally important in any physical problem. For a given geometry and governing equations, they determine the solution. Their numerical implementation, however, is often not so obvious and straightforward. Boundary conditions supplement equations of time evolution, computation of physical quantities of a system and they also appear in the problem of solving the Poisson equation. The summation takes place over particles in the system. Such expressions can create asymmetries near the boundaries. For example, a particle placed far from walls in a homogeneous medium can see (statistically) the same number of neighbours in every direction. But if it is located near the wall, the asymmetry will appear, and the number of neighbours will be different in various directions.

One of the ways to cope with this problem is to add a number of immobile particles that simulate the presence and influence of the walls. Here, however, we decided to use a more flexible approach. We applied the idea of mirror particles, already known in the SPH context (Morris et al., 1997) and in the probability density function method for turbulent flows (Minier and Pozorski, 1999). Their idea is based on the fact that all physical variables in the SPH are calculated by using sum interpolants. The problem is illustrated in Fig. 1. There, to any particle located at $\boldsymbol{r}_b$, at a distance not greater than h from the flat boundary (let $\boldsymbol{r}_w$ stand for the boundary point closest to particle b), we introduce its mirror particle located at $\boldsymbol{r}_{b'} = 2\boldsymbol{r}_w - \boldsymbol{r}_b$, i.e. symmetrically with respect to the boundary; hence $W(\boldsymbol{r}_w - \boldsymbol{r}_b) = W(\boldsymbol{r}_w - \boldsymbol{r}_{b'})$. Mirror particles are not a part of the system, i.e. their dynamics is not computed, but they possess features like mass, velocity etc. Their role is limited to "simulate" the vicinity of the flow domain boundaries and to enforce the boundary conditions for velocity and other quantities (if present in computations). For example, to ensure the no-slip boundary conditions for velocity, the density at b' is taken as $\rho_{b'} = \rho_b$, the mass of the mirror particle is taken as $m_{b'} = m_b$, and its velocity $\boldsymbol{U}_{b'} = 2\boldsymbol{U}_w - \boldsymbol{U}_b$. Then, Eq. (2.2) results in

$$\boldsymbol{U}(\boldsymbol{r}_w) = \sum_b m_b \frac{\boldsymbol{U}_b}{\rho_b} W(\boldsymbol{r}_w - \boldsymbol{r}_b) + \sum_{b'} m_{b'} \frac{2\boldsymbol{U}_w - \boldsymbol{U}_b}{\rho_{b'}} W(\boldsymbol{r}_w - \boldsymbol{r}_{b'}) = \tag{4.3}$$

$$= 2\boldsymbol{U}_w \sum_b \frac{m_b}{\rho_b} W(\boldsymbol{r}_w - \boldsymbol{r}_b) = \boldsymbol{U}_w$$

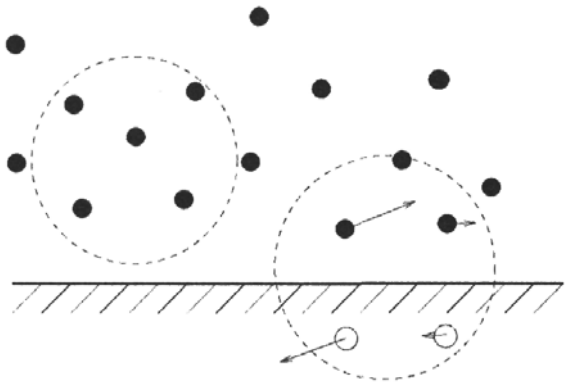

Fig. 1. Influence of boundaries in summation formulae: particles (•) and corresponding mirror particles (○), introduced in the boundary zone. Here, the velocities of the mirror particles are chosen to satisfy the no-slip wall boundary condition

The issue of boundary conditions shows up in a different way in the Poisson equations for pressure correction (e.g. normal derivatives equal to zero on the walls, given pressures at flow inlet/outlet). In the simulation we also consider the fluid flow with inlet/outlet. In that case, new particles are generated at the flow inlets to simulate the flux of mass into the domain. The new particles get randomly generated positions in the inlet area and velocities from a prescribed profile. The particles leaving the computational domain through the outlet are removed.

5. Computation results

The results can be divided into two categories: first, the tests which perform the basic checks of the algorithm and of the Poisson module; and second, the simulations of two-dimensional viscous flows.

5.1. 1D test of viscous damping

To test the method, we chose first a one dimensional problem with periodic boundary conditions to allow the density (and pressure waves) to propagate

freely. For that simulation, 500 particles are pseudorandomly distributed on the interval. The initial velocities are set to zero and the system is let to damp the initial perturbation in the density (created by leaving one of the cells empty). For such tests, a compressible version of the algorithm was used, with an equation of state in the form $p/\rho = \text{const}$. We calculated the density and its rms value (or standard deviation) over the interval. A temporal rms density record is presented in Fig. 2. It is readily seen that the initial perturbation is damped; the remaining residual density fluctuations, mainly due to somewhat disordered arrangement of particle locations, are an inherent feature of the SPH method. For the same initial configuration, Fig. 3 shows velocity records of a chosen particle in both inviscid (plot a) and viscous case (plot b). As expected, the initial perturbation repeats infinitely in the inviscid periodic system while it gets damped in the presence of viscosity. The plots also show the classic feature of developing numerical instability if the time step of the simulation is too large (cf. discussion in Section 4).

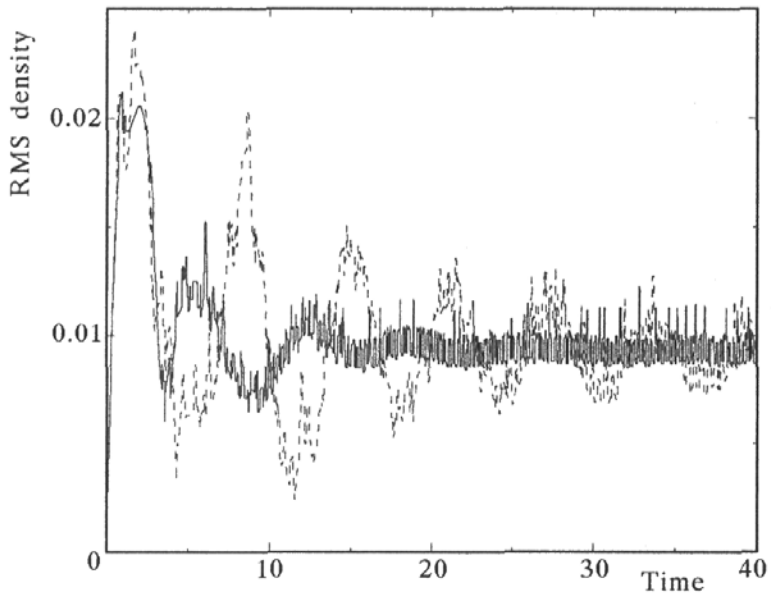

Fig. 2. RMS fluid density – response to initial perturbation; two different viscosity coefficients

5.2. Tests of the pressure correction algorithm

Next, we performed a test of the incompressibility constraint to check the performance of Poisson equation (3.12). A wall-driven cavity flow (cf. below) at $\text{Re} = 1$ with particles initially at rest was taken as the test case. Fig. 4 shows temporal evolution of non-uniformity of the density in the whole domain, computed out of the densities at the centres of the auxiliary mesh used to

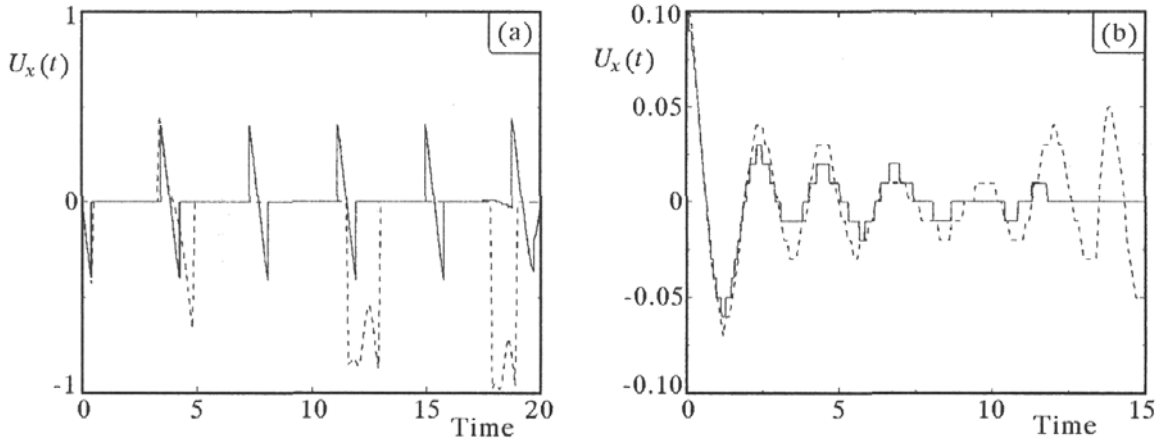

Fig. 3. Velocity record of a chosen particle – response to initial perturbation in the density field: (a) inviscid flow, (b) viscous flow. Solid lines: sufficiently small time step; dashed lines: too large time step (unstable computation)

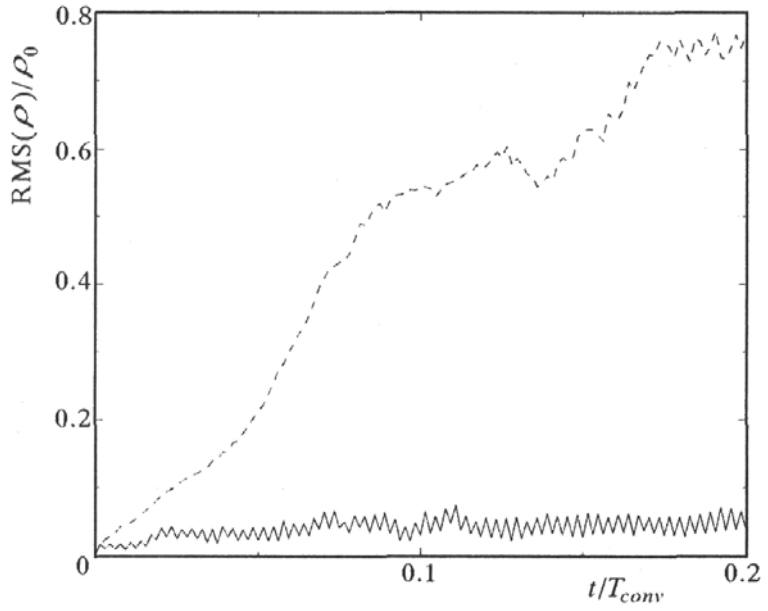

Fig. 4. Evolution of the density rms in the whole flow domain (lid driven cavity at Re = 1) with applied correction of particle locations (solid line) and without it (dashed line)

solve the Poisson equation. The density rms on the plot is normalised by the constant flow density, and the time is normalised by the convective time scale for the cavity. It is readily seen that the incompressibility correction does a good job: after the initial transient due to flow acceleration, the density rms attains a plateau; its level decreases with the number of particles used in the computation. On the other hand, in the case when no correction of particle

locations is applied, apart from the increasing non-uniformity of the density, also the results of the flow velocity field (not shown here) become degraded.

5.3. The Couette and Poiseuille flows

Two different cases of unsteady, initially at rest, two-dimensional viscous flows were considered: an impulsively started Couette's flow and Poiseuille's flow with suddenly applied streamwise pressure gradient. SPH results were obtained with only 20 particles in the cross-stream direction; the time step was $10^{-3}T_0$, with a characteristic time scale of the flow being equal to $T_0 = h^2/\nu$.

Temporal evolution of the velocity profile in the wall-driven Couette flow (with the upper wall velocity U_w) in the channel $0 \leqslant y \leqslant h$ was compared with the analytical formula after Schlichting and Gersten (1997)

$$U(\eta) = U_w \sum_{k=0}^{\infty} \operatorname{erfc}\,(2k\eta_h + \eta) + \operatorname{erfc}\,(2(k+1)\eta_h - \eta) \tag{5.1}$$

with $\eta = 0.5y/\sqrt{\nu t}$ and $\eta_h = 0.5h/\sqrt{\nu t}$. The computed and analytical velocity profiles are in good agreement for several time instants, as shown in Fig. 5a.

In the case of Poiseuille's flow, the developing velocity profile is known from a theoretical formula (cf. Morris et al., 1997)

$$\begin{aligned} U(y,t) &= \frac{G}{2\nu} y(y-h) + \\ &+ \sum_{k=0}^{\infty} \frac{4Gh^2}{\nu\pi^3(2k+1)^3} \sin\Big[\frac{\pi y}{h}(2k+1)\Big] \exp\Big[-\frac{(2k+1)^2\pi^2\nu t}{h^2}\Big] \end{aligned} \tag{5.2}$$

where G is the driving kinematic pressure gradient (or body force). As shown in Fig. 5b, the accordance of the computations with theory is satisfactory again.

5.4. The lid-driven cavity flow

This is one of the standard test cases used for validation of numerical algorithms for viscous flows. The geometry is very simple, but the flow structure becomes fairly complicated with the increasing Reynolds number, with more and more vortical regions appearing. It is also a genuine 2D flow where the implementation of incompressibility constraints is essential in order to obtain satisfying results. Another fairly difficult point in the SPH implementation is the correct statement of boundary conditions and the correct management of the mirror particles in summation formulae.

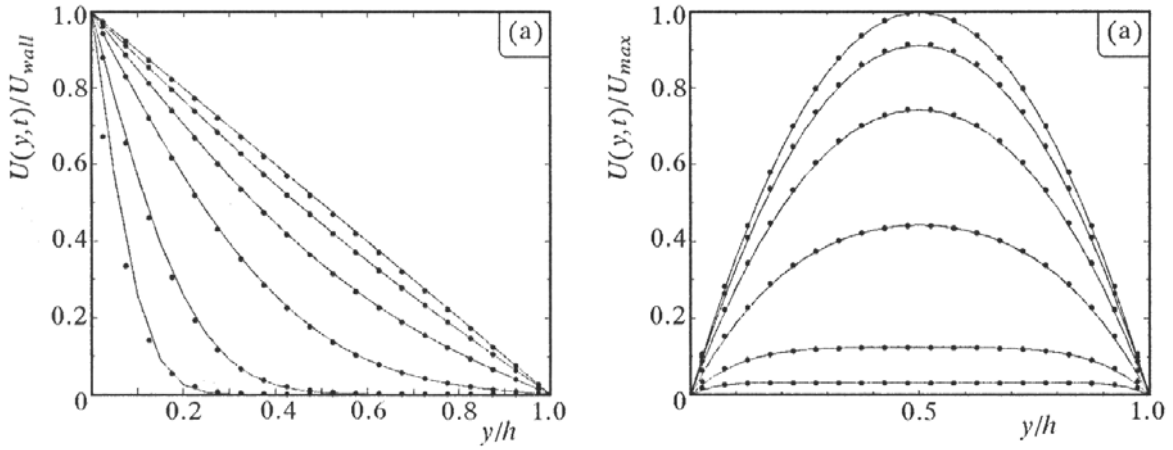

Fig. 5. Velocity profiles in developing (unsteady) viscous flows: (a) Couette flow; (b) Poiseuille flow. Results are plotted for several values of non-dimensional time $4\sqrt{\nu t}/h = 4\sqrt{t/T_0}$, equal to 0.25, 0.5, 1.0, 1.5, 2.0, and ∞, respectively. Solid line: analytical formula, circles: SPH computation

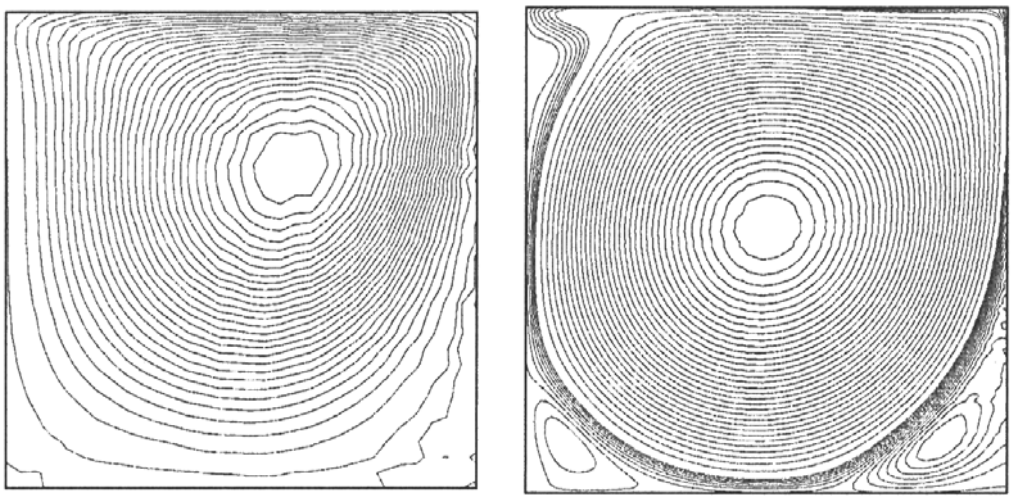

Fig. 6. Lid-driven cavity flow: computed streamlines at $\mathrm{Re} = 100$ and $\mathrm{Re} = 1000$; streamlines not equidistant for enhancing the visibility of secondary vortices

We simulated a flow in the rectangular domain $0 \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1$. The flow was driven by motion of the upper boundary $(y = 1)$. Results have been obtained for the Reynolds numbers up to $\mathrm{Re} = 1000$ using up to 80×80 particles at $\mathrm{Re} = 100$ and up to 240×240 particles at $\mathrm{Re} = 1000$. Computed streamlines in the cavity flow for two different Reynolds numbers are presented in Fig. 6. Profiles of velocity components: U_x for $x = 0.5$ and U_y for $y = 0.5$ are shown in Fig. 7. They coincide reasonably well with Eulerian computation results on a fine grid (Ghia et al., 1982) which are taken here as the reference data.

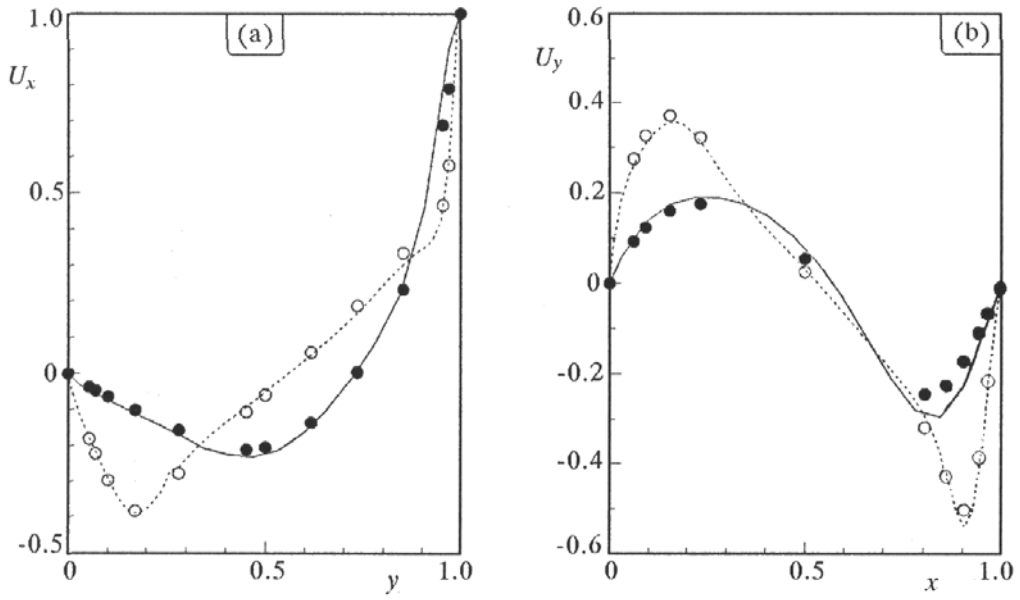

Fig. 7. Velocity profiles for the lid-driven cavity flow: (a) $U_x(y)$ at $x = 0.5$; (b) $U_y(x)$ at $y = 0.5$. Solid line: SPH computation at Re = 100; dotted line: SPH computation at Re = 1000. Reference results of Ghia et al. (1982): • (at Re = 100) and ∘ (at Re = 1000)

5.5. The Rayleigh-Taylor instability

The R-T instability-induced unsteady flow (its initial development at least) was numerically simulated in a square box of the size L, filled in the upper half with the heavier fluid of the density ρ_2, and in the lower half with the lighter fluid of the density ρ_1, with $\rho_2/\rho_1 = 2$, placed in the gravity field g. The linear stability analysis predicts this configuration to be unstable when (cf. Faber, 1995)

$$\frac{\pi}{L}\sqrt{\frac{\sigma}{(\rho_2 - \rho_1)g}} < 1 \tag{5.3}$$

where σ stands for the surface tension coefficient. We note the first known attempt to model the surface tension in the SPH formalism by Morris (2000). We do not account for the surface tension here ($\sigma = 0$), so instability criterion (5.3) is always met. However, to simulate the R-T instability numerically, we disturbed the initial configuration slightly in a kind of cosine wave (but initial particle velocities were still kept zero) in order to trigger the motion. The results show particle locations (Fig. 8a,b) and velocity field (Fig. 8c) in the initial phase of the developing instability. In the course of time, they become physically incorrect as the flow gets disordered with important diffusion of two fluids across the internal material interface which loses its identity pretty

quickly. It is difficult to comment on this behaviour now; however, the question arises whether the account for the surface tension would be sufficient to prevent this. Therefore the presented R-T results are of a qualitative character for the time being, suggesting directions for further improvement.

Although Fig. 8 looks like presenting typical results obtained with molecular simulation methods (as MD or DSMC), it should be emphasised here that the SPH approach is based on a Lagrangian discretisation of a *macroscopic continuum*, so the simulated particle system should behave as a continuum when the time goes on and not just as an ensemble of molecules, so the mixing phenomenon (mutual penetration of two fluids) should not occur when the molecular transport of mass is switched off (cf. Monaghan, 1989).

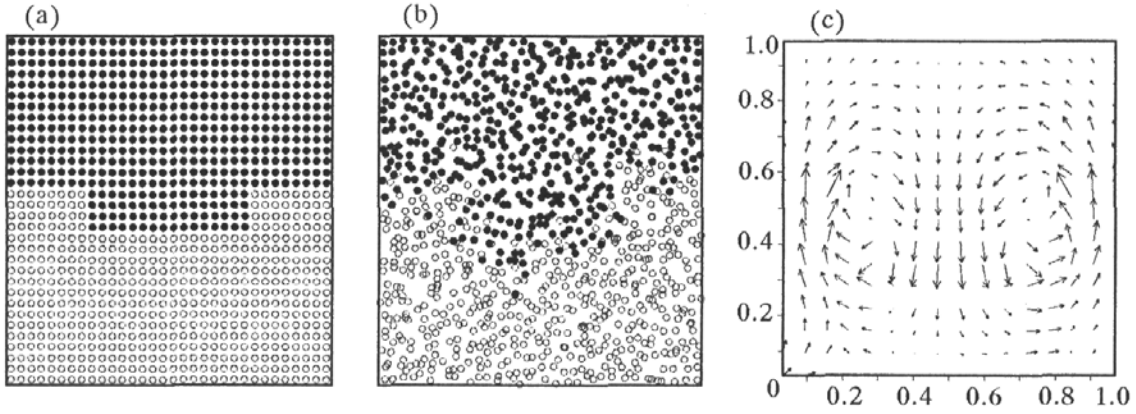

Fig. 8. Rayleigh-Taylor instability: (a) initial particle locations; (b) later particle locations; (c) velocity field at a later instant. Light fluid (•) and heavy fluid (○)

6. Conclusion and perspectives

The aim of the paper is to report the application of the SPH method in the area of incompressible viscous flows. The algorithm has been developed for incompressible case, scrutinized in a selection of validation tests and applied to two non-trivial 2D flow cases. Although there are other well-developed methods in numerical fluid dynamics (including finite element and finite volume methods) that are able to deal in a more efficient way with the type of flows tested to date, the point here is to validate the SPH algorithm before going further into more physically-complex flows where the potential advantages of the method are expected to show its full value. The incompressible formulation proposed here is currently being developed and, as a next step, will be used for cases of free-surface flows and the near-wall turbulence.

Acknowledgement

The activities reported in the present paper make part of a greater project supported by the State Committee for Scientific Research under grant KBN No. 955/T07/98/15. One of the authors (A.W.) kindly acknowledges the opportunity to carry out some of the computations at the TASK Computer Centre in Gdańsk.

References

1. CUMMINS S.J., RUDMAN M., 1999, An SPH projection method, *J. Comput. Phys.*, **152**, 584-607
2. FABER T.E., 1995, *Fluid Dynamics for Physicists,* Cambridge University Press, Cambridge
3. GHIA U., GHIA K.N., SHIN C.T., 1982, High Re-solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, *J. Comput. Phys.*, **48**, 387-411
4. GINGOLD R.A., MONAGHAN J.J., 1982, Kernel estimates as a basis for general particle methods in hydrodynamics, *J. Comput. Phys.*, **46**, 429-453
5. HOCKNEY R.W., EASTWOOD J.W., 1981, *Computer Simulations Using Particles,* McGraw-Hill, New York
6. LIBERSKY L.D., PETSCHEK A.G., CARNEY T.C., HIPP J.R., ALLAHDADI F.A., 1993, High strain Lagrangian hydrodynamics. A three-dimensional SPH code for dynamic material response, *J. Comput. Phys.*, **109**, 67-75
7. MINIER J.P., POZORSKI J., 1999, Wall boundary conditions in PDF methods and application to a turbulent channel flow, *Phys. Fluids*, **11**, 2632-2644
8. MONAGHAN J.J., 1989, On the problem of penetration in particle methods, *J. Comput. Phys.*, **82**, 1-15
9. MONAGHAN J.J., 1992, Smoothed particle hydrodynamics, *Annu. Rev. Astron. Astrophys.*, **30**, 543-574
10. MONAGHAN J.J., 1994, Simulating free surface flows with SPH, *J. Comput. Phys.*, **110**, 399-406
11. MONAGHAN J.J., GINGOLD R.A., 1983, Shock simulation by the particle method SPH, *J. Comput. Phys.*, **52**, 374-389
12. MORRIS J.P., 1996, Analysis of Smoothed Particle Hydrodynamics with applications, Ph.D. thesis, Monash University, Clayton Victoria, Australia
13. MORRIS J.P., 2000, Simulating surface tension with Smoothed Particle Hydrodynamics, *Int. J. Num. Meth. Fl.*, **33**, 333-353

14. MORRIS J.P., FOX P.J., ZHU Y., 1997, Modeling low Reynolds number incompressible flows using SPH, *J. Comput. Phys.*, **136**, 214-226

15. MORRIS J.P., MONAGHAN J.J., 1997, A switch to reduce SPH viscosity, *J. Comput. Phys.*, **136**, 41-50

16. POPE S.B., 1985, PDF methods for turbulent reactive flows, *Prog. Energy Combust. Sci.*, **11**, 119-192

17. SCHLICHTING H., GERSTEN K., 1997, *Grenzschicht-Theorie* (9. Auflage), Springer, Berlin

18. WELTON W.C., 1998, Two-dimensional PDF/SPH simulations of compressible turbulent flows, *J. Comput. Phys.*, **139**, 410–443

Obliczenia nieściśliwych przepływów lepkich metodą cząstek rozmytych (SPH)

Streszczenie

Praca dotyczy numerycznych obliczeń przepływów z wykorzystaniem metody cząstek rozmytych (ang. SPH – *Smoothed Particle Hydrodynamics*). Przypomniano pokrótce podstawy metody, a także fizyczne i matematyczne aspekty nieściśliwości. Przedstawiono nową propzycję spełnienia warunku nieściśliwości w podejściu SPH. Opisano numeryczną implementację metody i podstawowe elementy algorytmu, w tym spsób stawiania warunków brzegowych. Przedstawiono wyniki testów metody oraz przykłady obliczeń przepływów lepkich.

Manuscript received December 19, 2001; accepted for print January 23, 2002