

APPLICATION OF SMOOTHED PARTICLE HYDRODYNAMICS METHOD FOR SIMULATING INCOMPRESSIBLE LAMINAR FLOW

UDC 519.6:532.5

Summary

The paper presents the smoothed particle hydrodynamics (SPH) method, a numerical method for simulating fluid dynamics phenomena, based on particle systems. The SPH method approximates continuum with a finite number of particles which carry physical properties and serve as approximation points for spatial functions. Integral approximations of field functions and their derivatives are described using smoothing kernel functions which have to satisfy a number of conditions to ensure consistency with a given order. Particle approximations are also shown. This method can be used for simulating either compressible or incompressible flows if the special equation of state, proposed by Morris et al., is applied. A computer algorithm is developed for two standard benchmarking cases, the Poiseuille and the Couette flow. It is shown that the simulation results agree fairly well with the analytical series solution. Nevertheless, some combinations of time step for numerical integration and sound speed can lead to non-physical phenomena. Like any other numerical method, SPH has its advantages and disadvantages noted through practical use and theoretical considerations, which are briefly described in this paper.

Key words: computational fluid dynamics, smoothed particle hydrodynamics, kernel smoothing functions, Poiseuille flow, Couette flow

1. Introduction

Numerical simulations have become a very important approach for solving complex problems in engineering and science. With continuously increasing computer power, they have become a tool of scientific investigation used instead of expensive and time-consuming experiments. Computational fluid dynamics (CFD) comprises grid- or mesh-based numerical methods that solve basic field equations subject to boundary conditions by approaches involving a large number of elements leading to a large number of unknowns. Basic CFD techniques are the boundary element method (BEM), the finite element method (FEM), the finite difference method (FDM) and the finite volume method (FVM). Based on a properly pre-defined mesh, the governing equations can be converted to a set of algebraic equations with nodal unknowns for the field variables. Currently, these methods are dominant in numerical simulations for solving practical problems in engineering and science. Despite the great success, they suffer from difficulties in dealing with problems with a free surface, the deformable boundary, a moving interface, and extremely large deformation and crack

propagation. Moreover, for problems with complicated geometry, the generation of quality mesh has become a difficult, time-consuming and costly process [1, 2].

Solutions to the Navier – Stokes equations for various low Reynolds laminar flows are of great importance for both theoretical considerations and practical use. Since there are few analytical solutions to such equations, one often seeks help within numerical methods. Numerical methods such as the finite difference method, the finite volume method and the finite element method are often used in order to simulate fluid dynamics. Even though they are well established and have strong practical foundations, these methods have their disadvantages along with advantages. For example, fluid phenomena with the free surface effect as well as with moving boundary conditions are very hard to handle with such methods. The most often used finite volume method numerically solves the Navier-Stokes equations that are derived for the Eulerian, i.e. a fixed coordinate system. These equations are nonlinear due to convective terms [3]. To overcome these problems, the Smoothed Particle Hydrodynamics (SPH) method was introduced, which is Lagrangian in nature. Governing equations in Lagrangian form are linear, which facilitates implementation. SPH approximates continuum with a finite number of arbitrarily distributed particles which carry physical properties such as density, velocity, etc. Since the method does not need predefined connectivity between those particles, it is considered meshless. For instance, in the VOF (volume of fluid) method, the flow is defined with a non-linear conservative system of Navier-Stokes equations, and the computational domain is presented with a structured grid. Such a process of structured grid generation is time-consuming and requires a lot of user attention. In meshless methods, such as SPH, there is no need to worry about the quality of a mesh in regards to orthogonality, smoothness, skewness of volumes, etc. This quality of the SPH method results in an increase in the required CPU time, that is to say the simulation is slower. Even though it is considered that meshless methods are to become superior to the above-mentioned mesh-based methods [2], they are still in their development stage and are immature. As already mentioned, mesh-based methods demand nodal connectivity, which requires tedious pre-processing often done by trained engineers. In the SPH method, or any other meshless method, the time reserved for pre-processing is replaced by the CPU time which becomes cheaper on daily basis. One should bear in mind that this also extends the simulation time.

Since the fluid motion is described with a continuity equation and the Navier-Stokes equations, their SPH formulation is presented. These equations are used in a computer algorithm developed for the simulation of the time-dependant plane Poiseuille and Couette flow. It will be shown that obtained numerical results are in good agreement with analytical series solution provided by Morris et al. [12]. The method is still young, and has to overcome numerous problems in practical use, such as particle approximation consistency, faster and more robust simulation of incompressible flows using special equations of state, easier implementation of boundary conditions, etc.

2. Formulation of SPH equations of motion

The formulation of SPH is often divided into two key steps. The first step is the integral representation or the so-called kernel approximation of field functions. The second one is the particle approximation. As mentioned, continuum is discretized by a set of particles that need not to be connected. The SPH formulation is based on the integral representation of a field function using kernel functions instead of the Dirac delta function [4] as follows:

$$f(x_i) = \int_{\Omega} f(x'_i) W(x_i - x'_i, h) dx'_i \quad (1)$$

where $f(x_i)$ is the field function and Ω is the volume of integration which includes the position vector x_i . $W(x_i - x_i', h)$ is the kernel function which depends on the distance between two nodes and on the smoothing length, denoted h . Since the kernel function is used instead of the Dirac delta function, equation (1) is an approximation of a field function. Some conditions that the kernel function must satisfy in order to ensure consistency up to a given order will be discussed in the next section. If the kernel function is even, it can be easily shown that such approximation is of second-order accuracy.

Approximation for the field derivative of a function (1) is expressed by:

$$\frac{\partial f(x_i)}{\partial x_i} = \int_{\Omega} \frac{\partial f(x_i')}{\partial x_i'} W(x_i - x_i', h) dx_i' \quad (2)$$

Using the chain rule, one obtains:

$$\frac{\partial f(x_i)}{\partial x_i} = \int_{\Omega} \frac{\partial}{\partial x_i'} \left[f(x_i') W(x_i - x_i', h) \right] dx_i' - \int_{\Omega} f(x_i') \frac{\partial W(x_i - x_i', h)}{\partial x_i'} dx_i' \quad (3)$$

If the divergence theorem [5] is applied to the first integral in equation (3), and since the kernel function vanishes on the boundaries of integration, the first integral equals zero if the kernel function is not truncated by the boundary. This truncation problem arises for particles that are near the boundary. Various virtual particles can be placed near the boundary to overcome this problem. In summary, if the distance from a particle to the boundary is at least h , equation (3) becomes:

$$\frac{\partial f(x_i)}{\partial x_i} = - \int_{\Omega} f(x_i') \frac{\partial W(x_i - x_i', h)}{\partial x_i'} dx_i' \quad (4)$$

The above equation shows that the derivation of field function is transferred to the derivation of kernel function. This formulation is similar to the Galerkin weak formulation in the finite element method [6].

Since dx_i' in the above equations represents the differential fluid volume, it is replaced by the finite volume ΔV_{β} of particle β , which is related to the particle mass m_{β} by a well-known expression:

$$m_{\beta} = \Delta V_{\beta} \rho_{\beta} \quad (5)$$

If the integral in equation (2) is replaced by summation over N particles that are inside the smoothing length of position x , the particle approximation of a spatial function is obtained as:

$$f(x) = \sum_{\beta=1}^N \frac{m_{\beta}}{\rho_{\beta}} f(x_{\beta}) W(x - x_{\beta}, h) \quad (6)$$

Note that x still refers to the spatial vector. At particle α , the spatial function is calculated as:

$$f(x_{\alpha}) = \sum_{\beta=1}^N \frac{m_{\beta}}{\rho_{\beta}} f(x_{\beta}) W_{\alpha\beta} \quad (7)$$

where the term $W_{\alpha\beta}$ is the kernel function of particle α , evaluated at particle β , i.e. $W_{\alpha\beta} = W(x_\alpha - x_\beta, h)$. Equation (7) states that the value of a function at some particle α is obtained using the weighted average of N particles that are near the particle α . By applying mathematical identities, Monaghan [4] derived the following approximations for the derivative of spatial function:

$$\frac{\partial f(x_\alpha)}{\partial x_\alpha} = \frac{1}{\rho_\alpha} \left[\sum_{\beta=1}^N m_\beta \left[f(x_\beta) - f(x_\alpha) \right] \frac{\partial W_{\alpha\beta}}{\partial x_\alpha} \right] \quad (8)$$

$$\frac{\partial f(x_\alpha)}{\partial x_\alpha} = \rho_\alpha \left[\sum_{\beta=1}^N m_\beta \left[\frac{f(x_\beta)}{\rho_\beta^2} + \frac{f(x_\alpha)}{\rho_\alpha^2} \right] \frac{\partial W_{\alpha\beta}}{\partial x_\alpha} \right] \quad (9)$$

Expressions (8) and (9) are symmetrised since the spatial function is evaluated in the sense of interacting pairs of particles. They are considered to increase the accuracy and stability of simulation, and are therefore often used. The spatial derivative of kernel function is given by:

$$\frac{\partial W_{\alpha\beta}}{\partial x_\beta} = \frac{x_\alpha - x_\beta}{r_{\alpha\beta}} \frac{\partial W_{\alpha\beta}}{\partial r_{\alpha\beta}} = \frac{x_{\alpha\beta}}{r_{\alpha\beta}} \frac{\partial W_{\alpha\beta}}{\partial r_{\alpha\beta}} \quad (10)$$

With the previously mentioned consideration, SPH approximations for the continuity equation and the Navier-Stokes equations can be derived. There are generally two approaches to approximating density: a summation density and a continuity density approach. In this paper, only the summation density approach is considered. It can be obtained straightforwardly by replacing the spatial function with density in equation (6):

$$\rho_\alpha = \sum_{\beta=1}^N m_\beta W_{\alpha\beta} \quad (11)$$

This approach is intuitive because it simply states that the density of particle α is obtained by the weighted average of the mass of particles that are inside of the support domain of particle α .

The symmetrised Navier-Stokes equation is used, while others can be found in literature [4]:

$$\begin{aligned} \frac{Dv_{i(\alpha)}}{Dt} = & - \sum_{\beta=1}^N m_{(\beta)} \left(\frac{p_{(\alpha)}}{\rho_{(\alpha)}^2} + \frac{p_{(\beta)}}{\rho_{(\beta)}^2} \right) \frac{\partial W_{(\alpha\beta)}}{\partial x_{i(\alpha)}} + \\ & + \sum_{\beta=1}^N m_{(\beta)} \left(\frac{\mu_{(\alpha)} \varepsilon_{ij(\alpha)}}{\rho_{(\alpha)}^2} + \frac{\mu_{(\beta)} \varepsilon_{ij(\beta)}}{\rho_{(\beta)}^2} \right) \frac{\partial W_{(\alpha\beta)}}{\partial x_{j(\alpha)}} \end{aligned} \quad (12)$$

where $v_{i(\alpha)}$ is the velocity, $p_{(\alpha)}$ is the pressure, $\mu_{(\alpha)}$ is the dynamic viscosity and $\varepsilon_{ij(\alpha)}$ is the strain rate tensor of particle α . Note that indices for the Cartesian coordinates (i, j) are written outside the brackets, while particle indices (α, β) are written inside the brackets for simplicity. Approximation for the strain rate tensor is given by [3] as:

$$\begin{aligned} \varepsilon_{ij(\alpha)} = & \sum_{\beta=1}^N \frac{m_{(\beta)}}{\rho_{(\beta)}} v_{j(\beta\alpha)} \frac{\partial W_{(\alpha\beta)}}{\partial x_{i(\alpha)}} + \sum_{\beta=1}^N \frac{m_{(\beta)}}{\rho_{(\beta)}} v_{i(\beta\alpha)} \frac{\partial W_{(\alpha\beta)}}{\partial x_{j(\alpha)}} - \\ & - \left(\frac{2}{3} \sum_{\beta=1}^N \frac{m_{(\beta)}}{\rho_{(\beta)}} v_{i(\beta\alpha)} \frac{\partial W_{(\alpha\beta)}}{\partial x_{i(\alpha)}} \right) \delta_{ij} \end{aligned} \quad (13)$$

where $v_{i(\beta\alpha)}$ is the velocity difference between particles α and β , i.e. $v_{i(\beta\alpha)} = v_{i(\beta)} - v_{i(\alpha)}$, and the Kronecker delta symbol [7] is denoted by δ_{ij} . Note that instead of using approximations that include the second derivative of kernel function in order to represent viscous fluid forces, two first derivatives are used, since the strain rate tensor at particle α (13) is evaluated before the acceleration of particle α (12). This leads to the possibility of using lower order kernel functions.

3. Kernel functions

In order to ensure consistency up to a given order and to approximate the spatial function, as well as its first and second derivatives, the kernel function must satisfy a number of conditions [4]:

$$\begin{aligned} M_0 &= \int_{\Omega} W(x_i - x_i', h) dx_i' = 1 \\ M_1 &= \int_{\Omega} (x_i - x_i') W(x_i - x_i', h) dx_i' = 0 \\ M_2 &= \int_{\Omega} (x_i - x_i')^2 W(x_i - x_i', h) dx_i' = 0 \\ &\cdot \\ &\cdot \\ &\cdot \\ M_n &= \int_{\Omega} (x_i - x_i')^n W(x_i - x_i', h) dx_i' = 0 \end{aligned} \quad (14)$$

and

$$\begin{aligned} W(x_i - x_i', h) \Big|_{\partial\Omega} &= 0 \\ W'(x_i - x_i', h) \Big|_{\partial\Omega} &= 0 \end{aligned}$$

where W' denotes the gradient of kernel function and M_n stands for the n -th momentum of kernel function. These conditions can be easily derived using the Taylor expansion of $f(x_i')$ about x_i , assuming that $f(x_i')$ is sufficiently smooth.

It should be noted that these are conditions that the kernel function has to satisfy in order to approximate spatial functions and their derivatives by means of integral representation. Consistency conditions for particle approximations are generally not satisfied, especially for

particles near the boundary and irregularly distributed particles. Some techniques have been developed in order to satisfy those conditions, such as Reproducing Kernel Particle Method [4]. Many different kernel functions can be easily found in literature [4]. In this paper, B – spline, the cubic kernel function, see Fig. 1, is used:

$$W(R, h) = \begin{cases} \alpha_d \left(\frac{2}{3} - R^2 + \frac{1}{2} R^3 \right), & \text{for } 0 \leq R < 1 \\ \frac{\alpha_d}{6} (2 - R)^3, & \text{for } 1 \leq R < 2 \\ 0, & \text{for } R \geq 2 \end{cases} \quad (15)$$

where R is the relative distance between interacting pairs of particles, defined as:

$$R = \frac{r}{h} = \frac{|x_i - x_i'|}{h} \quad (16)$$

Coefficient α_d is determined from first equation in (14), which ensures zero order consistency. For two dimensional cases, α_d is $15/7\pi h^2$, and the kernel function can be visualized as shown in Fig. 2. This kernel function is often used in the applied SPH method due to its resemblance with the Gauss bell curve. Although, it should be noted, that its second derivative is a piecewise linear function, which can cause some stability issues.

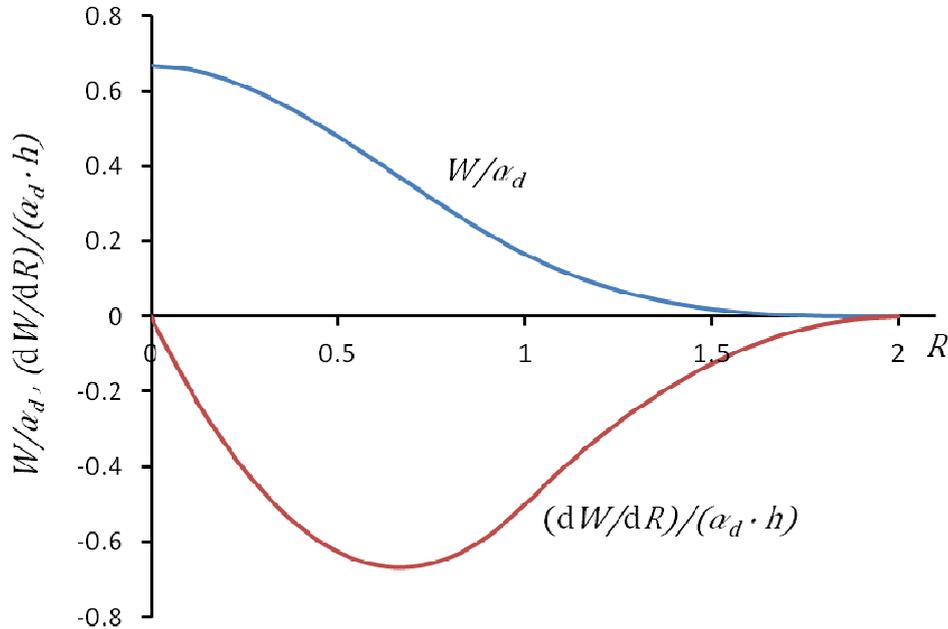


Fig. 1 B – Spline kernel function (15) and its derivative

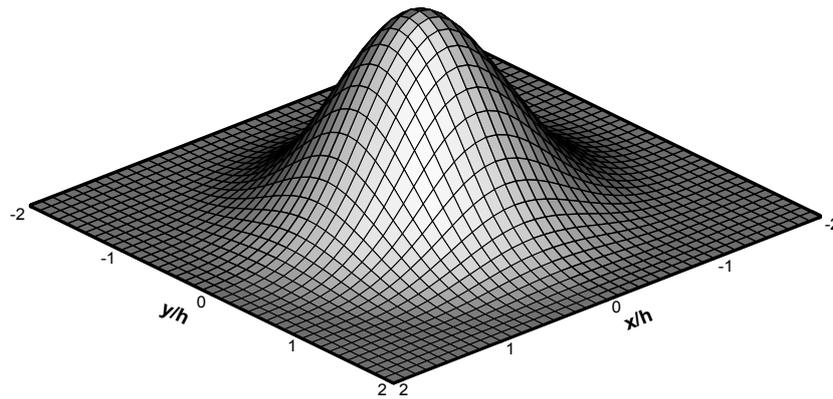


Fig. 2 Graphical representation of two dimensional kernel function

4. Numerical examples

The following section deals with the applications of the SPH method to incompressible flows to examine its effectiveness and to provide foundations for more complex problems. Two standard benchmarking cases, the Poiseuille and the Couette flow are studied in this paper.

4.1 Poiseuille flow

The Poiseuille flow is a fundamental problem of modern fluid dynamics. Fluid flows between two fixed parallel plates at $x_2 = 0$ and $x_2 = l$, where x_2 is the Cartesian coordinate perpendicular to plates, Fig. 3. The body force F_1 , which represents the longitudinal pressure gradient, starts to act on stationary fluid, and the flow reaches the steady state after a while.

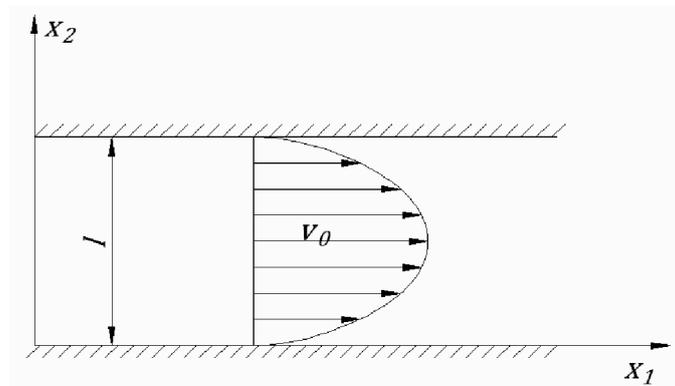


Fig. 3 Geometry and steady state velocity profile of the Poiseuille flow

The Navier-Stokes equation that describes such a flow at low Reynolds number is given by [8]:

$$\rho \frac{dv_1}{dt} = -\frac{dp}{dx_1} + \mu \frac{d^2 v_1}{dx_2^2} \quad (17)$$

Steady velocity profile can be easily obtained if one inserts $v_1 \neq v_1(t)$ into (17):

$$v_1(x_2) = -\frac{1}{2\nu} \frac{dp}{dx_1} \left(\frac{l^2}{4} - x_2^2 \right) \quad (18)$$

Integration constant $l^2/4$ is obtained using standard boundary conditions for viscous flows, i.e. the fluid velocity on fixed walls is equal to zero. The gradient pressure term can be replaced by the body force.

A solution to the unsteady flow problem is provided by Morris et al. [4]:

$$v_1(x_2, t) = \frac{F}{2\nu} x_2(x_2 - l) + \sum_{n=0}^{\infty} \frac{4Fl^2}{\nu\pi^3(2n+1)^3} \sin\left[\frac{\pi x_2}{l}(2n+1)\right] \exp\left[-\frac{(2n+1)^2 \pi^2 \nu}{l^2} t\right] \quad (19)$$

The following parameters are taken in order to simulate the considered flow:

$$l = 10^{-3} \text{ m} \quad - \text{ distance between fixed plates,}$$

$$\rho = 10^3 \text{ kg/m}^3 \quad - \text{ density of water,}$$

$$\nu = 10^{-6} \text{ m}^2/\text{s} \quad - \text{ kinematic viscosity of water,}$$

$$F = 2 \cdot 10^{-4} \text{ m/s}^2 \quad - \text{ body force in } x_1 \text{ direction,}$$

$$v_0 = 2.5 \cdot 10^{-5} \text{ m/s} \quad - \text{ peak velocity.}$$

The code was written in C++ programming language. The fluid domain is rectangular, from $x_1 = x_2 = 0$, to $x_1 = 0.0005 \text{ m}$ and $x_2 = l = 0.001 \text{ m}$. The domain is represented by a total of 861 particles, 41 in vertical direction, and 21 in longitudinal. The number of particles in vertical direction is greater than that in longitudinal since the velocity profile against the vertical axis is observed. Also, the use of periodic boundary condition in longitudinal direction justifies a lesser number of particles in longitudinal direction. The smoothing length h is 2.2 times greater than the initial distance of particles in x_2 direction. Boundary condition is implemented by the so-called virtual particles [4] presented by solid circles in Fig. 4. Velocity is zero for those particles at all time steps, and their position is not updated, which means they do not actually move. Periodic boundary condition is also used in order to shrink the modelled domain. It basically means that if some particles exceed a predetermined value of x_1 (leave the fluid domain in that direction), they are moved back to the inlet boundary. Also, particles near the inlet and outlet boundaries do not have enough neighbours to ensure the consistency of second order. To overcome this problem, particles on the inlet boundary are influenced by some particles on the outlet boundary and vice versa. Hatched circles shown in Fig. 4 represent the neighbouring particles of the crossed particle at the inlet.

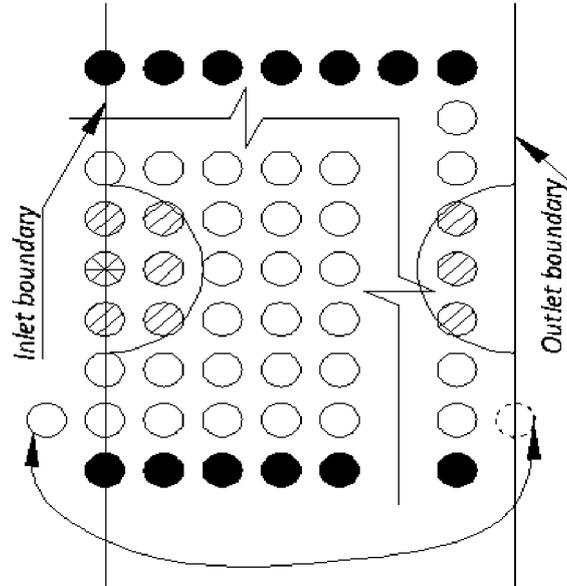


Fig. 4 Virtual particles and periodic boundary condition

In the standard SPH method for compressible flows, particles are moving due to the pressure gradient, while the pressure is calculated using the equation of state. Using the equation of state for incompressible flows leads to prohibitively small time steps. In reality, every fluid is slightly compressible, which leads to the artificial compressibility concept. Morris et al. [12] used the following equation of state in order to evaluate particle pressure:

$$p = c^2 \rho \quad (20)$$

where c represents the speed of sound. Nevertheless, if one uses the real speed of sound for water, $c \cong 1480 \text{ m/s}$, density variation becomes insignificant. Therefore, much lower values of speed of sound are admissible. The value of $c^2 = 5 \cdot 10^{-12} \text{ m}^2/\text{s}^2$ is taken for this simulation. This coefficient depends on the balance of pressure, viscous and body forces [12], and is related to the Mach number and thus to the compressibility of a fluid.

Particle motion is determined by the XSPH model [4] which takes the velocity of neighbouring particles into account:

$$\frac{dx_{i(\alpha)}}{dt} = v_{i(\alpha)} - \varepsilon \sum_{\beta=1}^N \frac{m(\beta)}{\rho(\beta)} v_{i(\alpha\beta)} W_{(\alpha\beta)} \quad (21)$$

where ε represents the constant with values from 0 to 1. The use of this method causes more coherent motion of particles. Constant ε has a value of 0.3 for this simulation.

Time integration is performed by a leapfrog scheme [9] which is simple, second-order accurate and easy to implement. The particle position is updated twice in each time step, while the velocity is evaluated only once as follows:

$$\begin{aligned} x_{n+\frac{1}{2}} &= x_n + \frac{1}{2} v_n \Delta t \\ v_{n+1} &= v_n + a_{n+\frac{1}{2}} \Delta t \\ x_{n+1} &= x_{n+\frac{1}{2}} + \frac{1}{2} v_{n+1} \Delta t \end{aligned} \quad (22)$$

The number of iterations is denoted by n , while a represents the particle acceleration.

Using a time step Δt of 10^{-6} s, error can be very roughly estimated with Δt^2 . Since the program operates with very small numbers, the accuracy of computer arithmetic also comes into question [10].

4.2 Couette flow

The Couette flow is similar to the Poiseuille flow. The geometry is the same, with the same parameters from the previous chapters. Fluid is initially stationary and the fluid motion is caused by a sudden movement of the upper plate with a constant velocity of $v_0 = 2.5 \cdot 10^{-5}$ m/s. At about $t = 1$ s, the fluid reaches a steady state, Fig. 5. The time-dependant series solution is also provided by Morris et al. [4] as:

$$v_1(x_2, t) = \frac{v_0}{l} x_2 + \sum_{n=1}^{\infty} \frac{2v_0}{n\pi} (-1)^n \sin\left[\frac{n\pi x_2}{l}\right] \exp\left[-\nu \frac{n^2 \pi^2}{l^2} t\right] \quad (23)$$

Simulation parameters and procedures are the same as the ones used in the simulation of the Poiseuille flow.

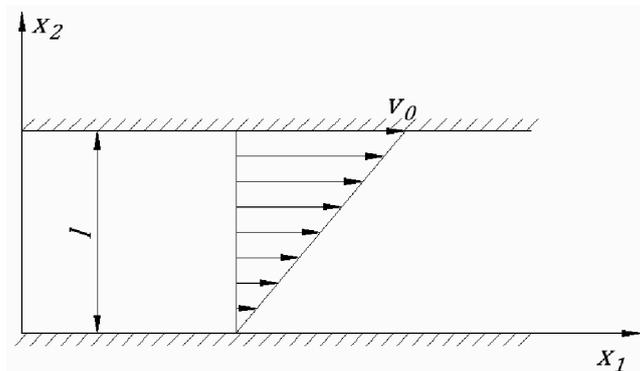


Fig. 5 Geometry and steady state velocity profile of the Couette flow

5. Results

Numerical results are in good agreement with analytical solutions (19) and (23), especially for the steady state flow, as can be seen in Figs. 6 and 7. The results are also in good agreement with those provided in literature [4]. By decreasing the time step, the simulation becomes extremely long, but the solution becomes more accurate and stable, as expected. It should be noted that there are small oscillations of velocity profile for both the Poiseuille and the Couette flow. This is probably caused by the inconsistency of particle approximations as well as by using the special equation of state (20). Better results should be obtained if one used a truly incompressible model for the simulation of incompressible flows [11]. These equations are obtained by forcing the solenoid velocity field in each time step and they are a bit more difficult to implement. Since the mentioned equations are implicit, the resulting linear system needs to be solved in every time step, which extremely prolongs the simulation. It can be seen that the obtained numerical solutions for the Poiseuille and the Couette flow are more erroneous for smaller values of t , as well as near the boundary, due to the particle deficiency or the inconsistency of integral representation via kernel functions. Steady state is reached roughly at $t = 1$ s as shown in Figs. 6 and 7. Also, it seems that the accuracy of simulation is shown to be better for higher values of t , with the above mentioned oscillations of velocity profile. One can remedy that problem by the least-squares polynomial fitting, if necessary.

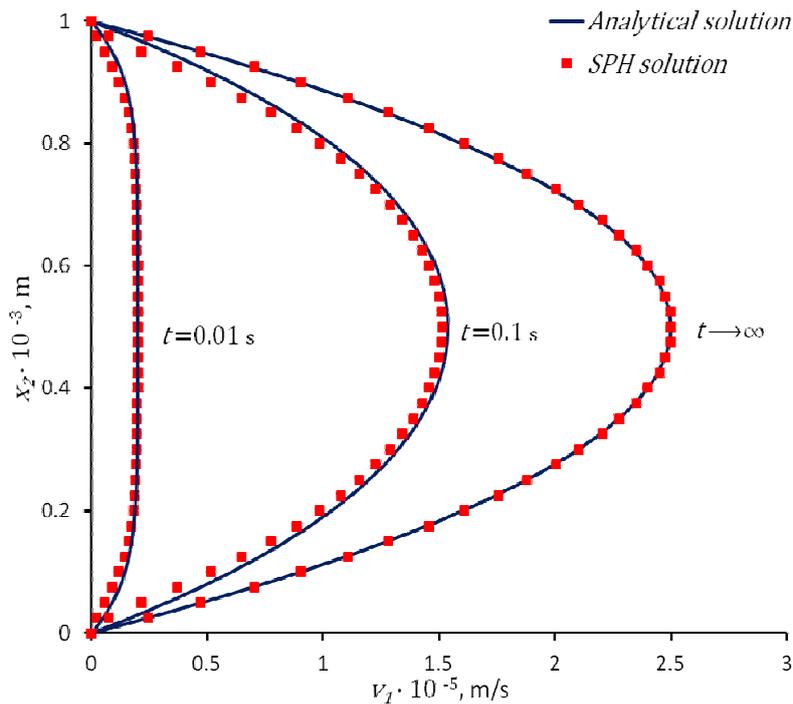


Fig. 6 Comparison between the SPH solution and the analytical solution provided by Morris et al. [12] for the Poiseuille flow

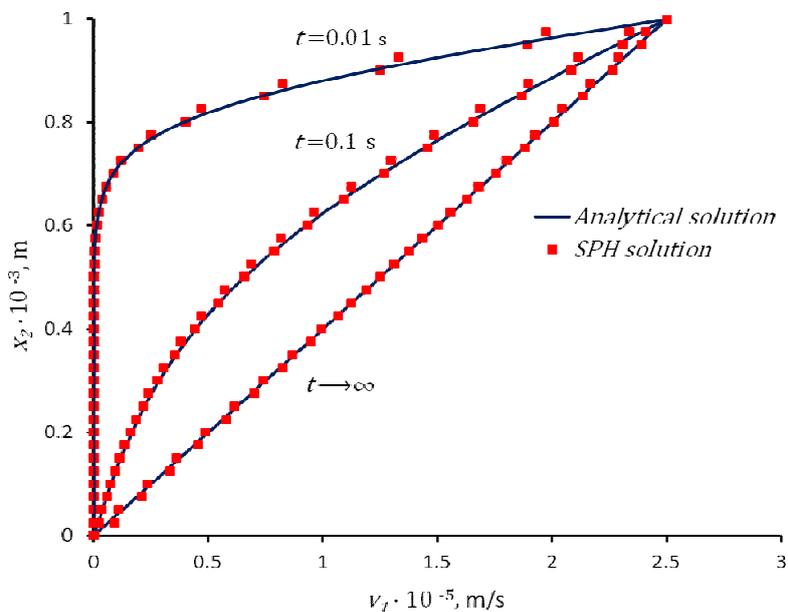


Fig. 7 Comparison between the SPH solution and the analytical solution provided by Morris et al. [12] for the Couette flow

6. Concluding remarks and ideas for future work

This paper presents the essentials of the SPH method, such as the integral representation of a spatial function and particle approximations. The consistency conditions for the integral representation via kernel function are shown, while the consistency of particle approximation is only mentioned. An interested reader is advised to see more details and other simulations in [4], which is the first book on the SPH method, simple and easy to read. The code for simulating two dimensional flows for low Reynolds numbers is developed and tested on the Poiseuille and the Couette flow.

As mentioned, the SPH method is a Lagrangian meshless method, i.e. the fluid domain is represented by particles. These attributes ensured the use of SPH in computer graphics. Since the accuracy in that area is not of vital importance, SPH is becoming more and more popular.

In future, the authors would like to develop models for turbulence based on a pseudo-random number generator and to test them with analytical solutions for simple flows or the finite volume method for arbitrary geometry. Although it is just an idea at this stage, a kind of model for impenetrability of rigid boundaries is considered. This model should be based on collisions modelling for solid mechanics with restitution coefficients.

It is known that meshless methods, such as SPH, use a lot more CPU time. Even though they are immature in comparison with the finite element method or the finite volume method, it can be expected that such methods will be used more widely in computational physics, especially for engineering problems. The time that engineers have to spend in order to generate a mesh in the well established grid based methods is becoming more and more expensive in comparison with the relatively cheap CPU time. The CPU time can be shortened by using parallel processors and corresponding algorithms. The SPH method is expected to mature over time through both theoretical considerations and practical implementation. Much more work has to be done to provide methods and algorithms which ensure effective particle approximation consistency up to a given order. Also, virtual particles for generating the boundary condition can sometimes be cumbersome to deal with, so other methods that could offset boundary effects should be taken into consideration.

REFERENCES

- [1] Liu, M. B., Liu, G. R., Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments, *Arch Comput Methods Eng* (2010) 17, pp. 25–76.
- [2] Liu, G. R., *Mesh Free Methods, Moving Beyond the Finite Element Method*, CRC Press LLC, Boca Raton, Florida, 2003.
- [3] Ferziger J. H., Perić, M., *Computational Methods for Fluid Dynamics*, Springer, Berlin, 1996.
- [4] Liu, G. R., Liu, M. B., *Smoothed Particle Hydrodynamics: a Meshfree Particle Method*, World Scientific, Singapore, 2003.
- [5] Kreyszig, E.: *Advanced Engineering Mathematics*, John Wiley & Sons, Inc., New York, 2006.
- [6] Sorić, J., *Metoda konačnih elemenata*, Golden marketing, Tehnička knjiga, Zagreb, 2004.
- [7] Degiuli, N., Werner, A., *Mehanika Fluida IB - podloge za nastavu*, <http://www.fsb.hr/zbrodo/>.
- [8] Werner, A., *Odabrana poglavlja iz mehanike fluida*, zbirka zadataka, Fakultet strojarstva i brodogradnje, Zagreb, 2002.
- [9] Romeel, D., *Leapfrog Integration*, http://ursa.as.arizona.edu/~rad/phys305/ODE_III/node11.html.
- [10] Singer, S., *Numerička matematika, predavanja*, Zagreb, 2009.
- [11] Colin, F., Egli, R., Lin, F. Y., Computing a null divergence velocity field using smoothed particle hydrodynamics, *Journal of Computational Physics* 217 (2006), pp. 680-692.
- [12] Morris, J. P., Fox, P. J., Zhu, J. , Modelling low Reynolds number incompressible flows using SPH, *Journal of Computational Physics*, 136:214 - 226 , 1997.

Submitted: 09.5.2012

Accepted: 15.11.2012

Vuko Vukčević, wukcevic@gmail.com
Andreja Werner, andreja.werner@fsb.hr
Nastia Degiuli, nastia.degiuli@fsb.hr
University of Zagreb
Faculty of Mechanical Engineering and
Naval Architecture,
Ivana Lučića 5, 10000 Zagreb, Croatia