A Second Order Semi-Lagrangian Particle Finite Element Method for Fluid Flows

Jonathan Colom Cobb[1], Julio García-Espinosa[1,2], Borja Servan Camas[1], Prashanth Nadukandi[3]

1 International Centre for Numerical Methods in Engineering

2 Universitat Politècnica de Catalunya - BarcelonaTech

3 University of Manchester

Abstract

In this paper, a second order SL-PFEM scheme for solving the incompressible Navier-Stokes equations is presented. This scheme is based on the second order velocity Verlet algorithm, which uses an explicit integration for the particle's trajectory and an implicit integration for the velocity. The algorithm is completed with a predictor-multicorrector scheme for the integration of the velocity correction using the Finite Element Method. A second order projector based on least squares is used to transfer the intrinsic variables information from the particles onto the background mesh, while a second order interpolation scheme is used to transfer the accelerations from the mesh to the particles. Convergence analyses are carried out to assess the second order convergence.

OPEN ACCESS

Published: 13/07/2019

DOI: 10.1007/s40571-019-00258-9

Keywords:
Semi-Lagrangian
Particles
PFEM
FEM
Second-order
Niceship project

1. Introduction

Solving the convective terms in transport equations using Lagrangian particles avoids the instabilities introduced by the standard approaches in the Eulerian framework. Instead, it leads to the problem of solving the particles trajectories, which offers minimum numerical erosion. However, Lagrangian and semi-Lagrangian methods are usually based on first order in time integration schemes, which limits its rate of convergence. As an exception, a recent work [4] within the semi-Lagrangian Particle Finite Element Method (SL-PFEM) framework, has proposed a second order scheme. It is based on Strang´s symmetric operator splitting, a third order projector to transfer data from the particle to the mesh, and on estimating the particle´s trajectories using a linear approximation to the instantaneous streamline determined by the velocity field.

Until [4], the SL-PFEM framework has been developed based on explicit integrators that compute streamlines to approximate the particle´s trajectories. This approach has been successfully applied to many different problems [1, 2, 3, 4]. In fact, the more convection dominates the flow, the better the streamlines approximate the pathlines and the larger the time step that can be used. However, in many problems, convection does not dominate and streamlines computed explicitly at the beginning of the time step are not a good approximation of pathlines. In these cases, small time steps are required to obtain accurate results. An example of this was presented in [5], where a linear wave problem was used as an example. Figure 1 has been extracted from [5], and it shows how pathlines can be approximated by explicit streamlines and by an explicit acceleration method. It can be observed that the explicitly streamline is not such a good approximation and therefore it requires to use very small time steps. On the other hand, including information of the acceleration, even in an explicit way, can considerably improve the approximation. This is the motivation of this work, where an algorithm based on the second order velocity Verlet integrator [6, 7, 8] is proposed, since it has some desirable properties such as being second order accurate over time, time reversible, and symplectic when applied to Hamiltonian systems [8].

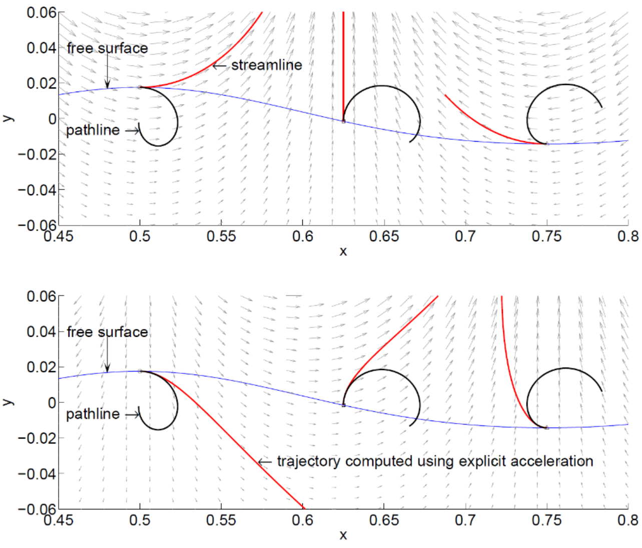


Figure 1: Approximation of pathlines of fluid particles under an acceleration field induced by a linear wave. Top; using explicit streamlines. Down: computing trajectories with explicit

Correspondence: Jonathan Colom Cobb (jcolom@cimne.upc.edu), Julio García-Espinosa (julio@cimne.upc.edu), Borja Servan Camas (bservan@cimne.upc.edu), Prashanth Nadukandi (prashanth.nadukandi@manchester.ac.uk).

acceleration. Figure extracted from [5]

This paper is organized as follows: first, the problem of particles moving with a velocity field is introduced. Second, the velocity Verlet scheme is presented to integrate the particle´s equations of motion. Third, the incompressible Navier-Stokes equations are introduced in a semi-Lagrangian framework, where the divergence free condition is enforced using a predictor-multicorrector scheme inspired in the fractional step method. Fourth, a global least-square projector is analysed. And fifth, a number of convergence verification, validation and demonstration cases are carried out.

2. Particles moving with a velocity field

Along the paper, Eulerian variables (mesh values) are written with lower case letters, particle´s variables are written with upper case letters and vectors are written with bold letters. Let $\boldsymbol{u}(\boldsymbol{x},t)$ be a velocity field, let $\{\lambda\}$ be a set of particles each of them identified with a label λ . The position occupied by each particle at time t is denoted by $\boldsymbol{X}_\lambda(t)$, and the velocity of the particle by $\boldsymbol{U}_\lambda(t)$. If the particle velocity is given by the velocity field $\boldsymbol{u}$, then $\boldsymbol{U}_\lambda(t) = \boldsymbol{u}(\boldsymbol{X}_\lambda(t),t)$, and the particles's trajectories are the pathlines defined by the velocity field $\boldsymbol{u}$. The particle's equations of motion are:

$$\frac{d\boldsymbol{X}_\lambda(t)}{dt} = \boldsymbol{U}_\lambda(t) = \boldsymbol{u}(\boldsymbol{X}_\lambda(t),t) \tag{1}$$

$$\frac{d\boldsymbol{U}_\lambda(t)}{dt} = \boldsymbol{A}_\lambda(t) \tag{2}$$

where the particle's acceleration $\boldsymbol{A}_\lambda(t)$ is the rate of change of $\boldsymbol{U}_\lambda(t)$. An acceleration field $\boldsymbol{a}(\boldsymbol{x},t)$ can be derived from the velocity field as:

$$\boldsymbol{a}(\boldsymbol{x},t) = d_t\boldsymbol{u}(\boldsymbol{x},t) \tag{3}$$

where d_t is the total derivative

$$d_t\boldsymbol{u} = \partial_t\boldsymbol{u} + \boldsymbol{u}\nabla\boldsymbol{u}. \tag{4}$$

Given an acceleration field, $\boldsymbol{a}$, derived from a velocity field $\boldsymbol{u}$ as in Eq. (4), the particles' velocities and positions can be obtained integrating the equations of motion with initial conditions:

$$\boldsymbol{U}_\lambda(0) = \boldsymbol{u}(\boldsymbol{X}_\lambda(0),0) \tag{5}$$

3. Numerical integration of particles' equations of motion

Let $\boldsymbol{a}(\boldsymbol{x},t) = d_t\boldsymbol{u}(\boldsymbol{x},t)$ be an acceleration field derived from an unknown velocity field $\boldsymbol{u}(\boldsymbol{x},t)$. Given a set of particles $\{\lambda\}$ with initial position and velocity $\boldsymbol{X}_\lambda(0)$ and $\boldsymbol{U}_\lambda(0) = \boldsymbol{u}(\boldsymbol{X}_\lambda(0),0)$. Then the position and velocity at time t is obtained integrating the equations of motion:

$$\boldsymbol{X}_\lambda(t^{n+1}) = \boldsymbol{X}_\lambda(t^n) + \int_{t^n}^{t^{n+1}} \boldsymbol{U}_\lambda(t)\,dt \tag{6}$$

$$\boldsymbol{U}_\lambda(t^{n+1}) = \boldsymbol{U}_\lambda(t^n) + \int_{t^n}^{t^{n+1}} \boldsymbol{A}_\lambda(t)\,dt \tag{7}$$

In molecular dynamics, a well-known numerical integration scheme for the equations of motion is the velocity Verlet integrator [6,7], which reads:

$$\boldsymbol{X}_\lambda(t^{n+1}) = \boldsymbol{X}_\lambda(t^n) + \Delta t\,\boldsymbol{U}_\lambda(t^n) + \frac{\Delta t^2}{2}\boldsymbol{A}_\lambda(t^n) + O(\Delta t^3) \tag{8}$$

$$\boldsymbol{U}_\lambda(t^{n+1}) = \boldsymbol{U}_\lambda(t^n) + \frac{\Delta t}{2}\left(\boldsymbol{A}_\lambda(t^n) + \boldsymbol{A}_\lambda(t^{n+1})\right) + O(\Delta t^3) \tag{9}$$

where eq. (8) is explicit in time and provides the particle position. We can rewrite eqs. (8) and (9) as:

$$\boldsymbol{X}_\lambda^{n+1} = \boldsymbol{X}_\lambda^n + \Delta t\,\boldsymbol{U}_\lambda^n + \frac{\Delta t^2}{2}\boldsymbol{a}^n(\boldsymbol{X}_\lambda^n) \tag{10}$$

$$\boldsymbol{U}_\lambda^{n+1} = \boldsymbol{U}_\lambda^n + \frac{\Delta t}{2}\left(\boldsymbol{a}^n(\boldsymbol{X}_\lambda^n) + \boldsymbol{a}^{n+1}(\boldsymbol{X}_\lambda^{n+1})\right) \tag{11}$$

where $\boldsymbol{X}_\lambda^n = \boldsymbol{X}_\lambda(t^n)$, $\boldsymbol{U}_\lambda^n = \boldsymbol{U}_\lambda(t^n)$, $\boldsymbol{a}^n(\boldsymbol{X}_\lambda^n) = \boldsymbol{a}(\boldsymbol{X}_\lambda^n, t^n)$.

4. A Semi-Lagrangian approach for solving the incompressible Navier-Stokes equations

1.1 Lagrangian formulation of the incompressible Navier-Stokes equations

The incompressible Navier Stokes equations can be written as:

$$d_t\boldsymbol{u} = -\nabla\left(\frac{P}{\rho}\right) + \nu\Delta\boldsymbol{u} + \boldsymbol{f} \tag{12}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{13}$$

Where P is the fluid pressure, ρ is the fluid density, ν is the kinematic viscosity and $\boldsymbol{f}$ are mass forces. Eq. (12) can be expressed as $d_t\boldsymbol{u} = \boldsymbol{a}$, where the acceleration field is given by:

$$\boldsymbol{a} = -\nabla\left(\frac{P}{\rho}\right) + \nu\Delta\boldsymbol{u} + \boldsymbol{f} \tag{14}$$

Solving the Navier Stokes' equation in a Lagrangian framework by integrating the fluid particles dynamics is not as simple, since the acceleration field depends on the fluid velocity and the pressure. In order to ensure the divergence free condition of the flow field, the pressure must fulfil an integral equation on the domain. This proves to be very complicated when imposing the continuity condition in a Lagrangian framework.

1.2 Semi-Lagrangian integration of the incompressible Navier-Stokes equations

Below, a Semi-Lagrangian (SL) approach is introduced to overcome the difficulties of estimating the pressure gradient and viscosity terms in a Lagrangian framework. To do so the acceleration field will be obtained in an Eulerian framework by projecting the particle´s velocity field onto the Eulerian space.

Let $\{\lambda\}$ be a set of fluid particles. Then the equation of motions can be integrated using the velocity Verlet integrator:

$$\boldsymbol{X}_\lambda^{n+1} = \boldsymbol{X}_\lambda^n + \Delta t\,\boldsymbol{U}_\lambda^n + \frac{\Delta t^2}{2}\boldsymbol{a}^n(\boldsymbol{X}_\lambda^n) \tag{15}$$

$$\frac{\boldsymbol{U}_\lambda^{n+1} - \boldsymbol{U}_\lambda^n}{\Delta t} = \frac{1}{2}\left(\boldsymbol{a}^n(\boldsymbol{X}_\lambda^n) + \boldsymbol{a}^{n+1}(\boldsymbol{X}_\lambda^{n+1})\right) \tag{16}$$

As stated above, this scheme is second order accurate over time. Let´s assume that all variables are known at time t^n. Then the new position of the fluid particles at time t^{n+1} can be easily calculated using Eq.(15). Reordering Eq. (16):

$$\boldsymbol{U}_{\lambda}^{n+1} = \boldsymbol{U}_{\lambda}^{n+1/2} + \frac{\Delta t}{2}\boldsymbol{a}^{n+1}(\boldsymbol{X}_{\lambda}^{n+1}) \tag{17}$$

where $\boldsymbol{U}_{\lambda}^{n+1/2} = \boldsymbol{U}_{\lambda}^{n} + \frac{\Delta t}{2}\boldsymbol{a}^{n}(\boldsymbol{X}_{\lambda}^{n})$. In order to solve eq. (17), the term $\boldsymbol{a}^{n+1}$ will be solved in an Eulerian framework. Then an equivalent Eulerian equation is imposed:

$$\boldsymbol{u}^{n+1}(\boldsymbol{x}) = \boldsymbol{u}^{n+1/2}(\boldsymbol{x}) + \frac{\Delta t}{2}\boldsymbol{a}^{n+1}(\boldsymbol{x}) \tag{18}$$

Where $\boldsymbol{u}^{n+1/2}$ is obtained via projection:

$$\boldsymbol{u}^{n+1/2}(\boldsymbol{x}) = P_{\{\lambda\}}^{n+1}\left[\{\boldsymbol{U}_{\lambda}^{n+1/2}\}\right] \tag{19}$$

$P_{\{\lambda\}}^{n+1}$ is a projector operator that transfer the values of the variables from the set of particle with positions $\{\boldsymbol{X}_{\lambda}^{n+1}\}$ onto the mesh.

We now introduce what we call the coherence condition for the projector:

$$\psi^{n+1}(\boldsymbol{x}) = P_{\{\lambda\}}^{n+1}\left[\{\psi(\boldsymbol{X}_{\lambda}^{n+1})\}\right], \tag{20}$$

where ψ is an arbitrary variable defined on the Eulerian framework. This condition means that the projection of the nodal values interpolated at the particle´s return the original nodal values. If this condition is fulfilled, then from Eq. (18) and (20):

$$\boldsymbol{u}^{n+1}(\boldsymbol{x}) = P_{\{\lambda\}}^{n+1}\left[\{\boldsymbol{U}_{\lambda}^{n+1/2}\}\right] + \frac{\Delta t}{2}P_{\{\lambda\}}^{n+1}\left[\{\boldsymbol{a}^{n+1}(\boldsymbol{X}_{\lambda}^{n+1})\}\right] \tag{21}$$

which is equivalent to say that $\boldsymbol{u}^{n+1}(\boldsymbol{x}) = P_{\{\lambda\}}^{n+1}\left[\{\boldsymbol{U}_{\lambda}^{n+1}\}\right]$. There is another outcome from the fulfilment of the coherence condition. Figure 2 shows the conceptual implementation of a generic SL-PFEM. In general two iterative loops are required to avoid any sort of splitting error. However, if the coherence condition is fulfilled, there is no need of iterating at the outer loop. This is because the correction of the particles´ velocities, given by $\frac{\Delta t}{2}\boldsymbol{a}^{n+1}(\boldsymbol{X}_{\lambda}^{n+1})$, will be projected back to the mesh as it is, with no further need of correction.

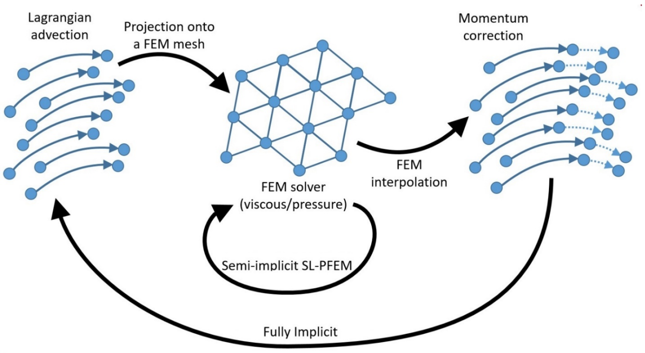


Figure 2: Conceptual description of the SL-PFEM

Eq. (18) can be solved iteratively by further splitting into:

$$\hat{\boldsymbol{u}}^{n+1,i+1} = \boldsymbol{u}^{n+1/2} + \frac{\Delta t}{2}\left(-\nabla\left(\frac{P^{n+1,i}}{\rho}\right) + \nu\Delta\hat{\boldsymbol{u}}^{n+1,i+1} + \boldsymbol{f}^{n+1}\right) \tag{22}$$

$$\boldsymbol{u}^{n+1,i+1} - \hat{\boldsymbol{u}}^{n+1,i+1} = \frac{\Delta t}{2}\left(-\nabla\left(\frac{P^{n+1,i+1}}{\rho}\right) + \nabla\left(\frac{P^{n+1,i}}{\rho}\right)\right) \tag{23}$$

Where i is the iteration index. Now in order to obtain the pressure $P^{n+1,i+1}$, the continuity condition is imposed. Introducing the continuity equation ($\nabla\cdot\boldsymbol{u}^{n+1,i+1} = 0$) into Eq.(23) and reordering

$$\Delta\left(\frac{P^{n+1,i+1}}{\rho}\right) = 2\frac{\nabla\cdot\hat{\boldsymbol{u}}^{n+1,i+1}}{\Delta t} + \Delta\left(\frac{P^{n+1,i}}{\rho}\right) \tag{24}$$

Once the iterative process between Eqs. (22)-(23) has converged, from eq. (18) we obtain:

$$\boldsymbol{a}^{n+1}(\boldsymbol{x}) = 2\frac{\boldsymbol{u}^{n+1}(\boldsymbol{x}) - \boldsymbol{u}^{n+1/2}(\boldsymbol{x})}{\Delta t} \tag{25}$$

1.3 The SL-PFEM method for solving the incompressible Navier-Stokes equations

In the previous section, a projector operator was introduced to project the particle´s variables onto the Eulerian space. Then, a strategy to solve the acceleration due to the pressure gradient and viscous terms can be formulated in an Eulerian framework.

The finite element method (FEM) is introduced to solve Eq. (22) and (24). First, we need to discretize it in space using a finite element mesh. Then, any intrinsic dependent variable ψ can be expressed as:

$$\psi_h(\boldsymbol{x}) = \sum_b N^b(\boldsymbol{x})\psi_b \tag{26}$$

where $\{b\}$ is the set of mesh nodes, $N^b(\boldsymbol{x})$ are the classic Finite Element (FE) linear shape functions, and ψ_b are the nodal values. And let $\boldsymbol{\psi}_h$ be the vector of nodal values ψ_b.

Once the spatial discretization is set with a FE mesh, the projector operator must transfer values from particles onto the background mesh following Eq. (21):

$$\boldsymbol{u}_h^{n+1} = P_{\{\lambda\}}^{n+1}\left[\{\boldsymbol{U}_{\lambda}^{n+1}\}\right] \tag{27}$$

Where $P_{h,\{\lambda\}}$ must fulfil the condition given in Eq. (20) leading to:

$$\boldsymbol{a}_h^{n+1}(\boldsymbol{x}) = P_{h,\{\lambda\}}^{n+1}\left[\{\boldsymbol{a}_h^{n+1}(\boldsymbol{X}_{\lambda}^{n})\}\right] \tag{28}$$

Using the FE Galerkin method as in [9,10] to solve the variational formulation of Eqs. (22) and (24) we get:

$$\left(\bar{\bar{\boldsymbol{M}}} - \frac{\Delta t\nu}{2}\bar{\bar{\boldsymbol{L}}}\right)\hat{\boldsymbol{u}}_h^{n+1,i+1} = \bar{\bar{\boldsymbol{M}}}\boldsymbol{u}_h^{n+1/2} - \frac{\Delta t}{2\rho}\bar{\bar{\boldsymbol{G}}}\boldsymbol{P}_h^{n+1,i} + \frac{\Delta t}{2}\bar{\bar{\boldsymbol{M}}}\boldsymbol{F}_h^{n+1,i} \tag{29}$$

$$\bar{\bar{\boldsymbol{L}}}\boldsymbol{P}_h^{n+1,i+1} = \frac{2\rho}{\Delta t}\bar{\bar{\boldsymbol{D}}}\hat{\boldsymbol{u}}_h^{n+1,i+1} + \bar{\bar{\boldsymbol{D}}}\bar{\bar{\boldsymbol{M}}}^{-1}\bar{\bar{\boldsymbol{G}}}\boldsymbol{P}_h^{n+1,i} \tag{30}$$

Where $\bar{\bar{\boldsymbol{M}}}$ is the FE mass matrix, $\bar{\bar{\boldsymbol{L}}}$ is the FE Laplacian matrix, $\bar{\bar{\boldsymbol{G}}}$ is the FE gradient matrix, and $\bar{\bar{\boldsymbol{D}}}$ is the FE divergence matrix. Once the iterative loop over i converges, then the new acceleration field is obtained from Eq. (18). The convergence criteria used for the pressure and velocity are:

$$\frac{\|\boldsymbol{P}_h^{n+1,i+1} - \boldsymbol{P}_h^{n+1,i}\|}{\|P_{max}^{n+1,i+1} - P_{min}^{n+1,i+1}\|} < 0.001 \tag{31}$$

$$\frac{\| \, |\boldsymbol{U}_h|^{n+1,i+1} - |\boldsymbol{U}_h|^{n+1,i} \, \|}{|\boldsymbol{U}|_{max}} < 0.001 \tag{32}$$

5. Global least square projector

Let $\{\Psi_\lambda\}$ be a set of values carried by a set of particles and $\{\psi_b^*\}$ be the set of projected values on the mesh nodes. The projector operator used in this work is based on the minimization of the global least square error. The global least square error is given by:

$$\epsilon_\psi = \sum_\lambda (\psi_h(\boldsymbol{X}_\lambda) - \Psi_\lambda)^2 \tag{33}$$

where $\psi_h(\boldsymbol{X}_\lambda) = \sum_b N^b(\boldsymbol{X}_\lambda)\psi_b^*$

$$\epsilon_\psi = \sum_\lambda \left(\sum_b N^b(\boldsymbol{X}_\lambda)\psi_b^* - \Psi_\lambda \right)^2 \tag{34}$$

Minimizing the least square error via $\partial\epsilon_\psi / \partial\psi_b^* = 0$:

$$\sum_\lambda \left(\sum_c N^b(\boldsymbol{X}_\lambda) N^c(\boldsymbol{X}_\lambda)\psi_c^* \right) = \sum_\lambda N^b(\boldsymbol{X}_\lambda)\Psi_\lambda \tag{35}$$

Eq. (35) represents a linear system of equation with as many unknowns as mesh nodes. This system can be seen as a discrete version of the classic FE projection. Introducing an interpolated field on the particles $\Psi_\lambda = \sum_c N^c(\boldsymbol{X}_\lambda)\psi_c$:

$$\sum_\lambda \left(\sum_c N^b(\boldsymbol{X}_\lambda) N^c(\boldsymbol{X}_\lambda)\psi_c^* \right) = \sum_\lambda \left(\sum_c N^b(\boldsymbol{X}_\lambda) N^c(\boldsymbol{X}_\lambda)\psi_c \right), \tag{36}$$

leading to $\psi_c^* = \psi_c$ and the fulfilment of the coherence condition given in eq. (20).

In order to analyse the order of convergence of the projector, let's assume that the particles values are given by a continuous and smooth function ψ . Then the equalities $\Psi_\lambda = \psi(\boldsymbol{X}_\lambda)$ and $\psi(\boldsymbol{X}_\lambda) = \sum_c N^c(\boldsymbol{X}_\lambda)\psi(\boldsymbol{x}_c) + O(h_e^2)$ hold, where h_e is the characteristic length of the element where particle λ is at. Using Eq. (35):

$$\sum_\lambda N^b(\boldsymbol{X}_\lambda) \left(\sum_c N^c(\boldsymbol{X}_\lambda)\psi_c^* \right) = \sum_\lambda N^b(\boldsymbol{X}_\lambda) \left(\sum_c N^c(\boldsymbol{X}_\lambda)\psi(\boldsymbol{x}_c) \right) + \sum_\lambda N^b(\boldsymbol{X}_\lambda) O(h_e^2) \tag{37}$$

And, the solution to eq. (37) can be expressed as:

$$\psi_c^* = \psi(\boldsymbol{x}_c) + \delta\psi_c \tag{38}$$

Where $\delta\psi_c$ is the solution of:

$$\sum_\lambda N^b(\boldsymbol{X}_\lambda) \left(\sum_c N^c(\boldsymbol{X}_\lambda)\delta\psi_c \right) = \sum_\lambda N^b(\boldsymbol{X}_\lambda) O(h_e^2) \tag{39}$$

Then, the projected value converges with second order accuracy in space:

$$\psi_c^* = \psi(\boldsymbol{x}_c) + O(h_e^2) \tag{40}$$

6. Error accumulation in the SL-PFEM

SL-PFEM methods are prone to accumulate errors as information is passed from the Lagrangian particles onto the background mesh and vice-versa. Those errors are originated in the projection and interpolation steps. And the accumulation of those errors could lead to several undesired effects such as the degradation of the rate of convergence of the numerical scheme, and the incapability to reach a steady state solution. This section aims at demonstrating that the rate of convergence of the proposed scheme is not degraded due to this phenomenon and that second order convergence can be achieved.

Given a velocity field $\boldsymbol{u}(\boldsymbol{x}, t^n) = \boldsymbol{u}^n(\boldsymbol{x})$ and the corresponding acceleration field $\boldsymbol{a}(\boldsymbol{x}, t^n) = \boldsymbol{a}^n(\boldsymbol{x}) = d_t\boldsymbol{u}^n(\boldsymbol{x})$, and a set of particles with initial condition $\boldsymbol{U}_\lambda^0 = \boldsymbol{u}^0(\boldsymbol{X}_\lambda^0)$, the velocity Verlet algorithm estimates the velocity and position at $t^n = n\Delta t$ with second order convergence:

$$\hat{\boldsymbol{X}}_\lambda^{n+1} = \hat{\boldsymbol{X}}_\lambda^n + \Delta t \, \hat{\boldsymbol{U}}_\lambda^n + \frac{\Delta t^2}{2} \boldsymbol{a}^n\left(\hat{\boldsymbol{X}}_\lambda^n\right) = \boldsymbol{X}_\lambda^{n+1} + O(t^n \Delta t^2) \tag{41}$$

$$\hat{\boldsymbol{U}}_\lambda^{n+1} = \hat{\boldsymbol{U}}_\lambda^n + \frac{\Delta t^2}{2} \left(\boldsymbol{a}^n\left(\hat{\boldsymbol{X}}_\lambda^n\right) + \boldsymbol{a}^{n+1}\left(\hat{\boldsymbol{X}}_\lambda^{n+1}\right) \right) = \boldsymbol{u}^{n+1}(\boldsymbol{X}_\lambda^{n+1}) + O(t^n \Delta t^2) \tag{42}$$

Now, we will evaluate the error accumulation over time, by estimating the error in the Lagrangian integration of a given acceleration field and subsequent mesh projection-interpolation. Let us consider a space discretization using a FE mesh, in which the given acceleration field is approximated by an interpolated field, as follows:

$$\boldsymbol{a}_h^n(\boldsymbol{x}) = \sum_b N^b(\boldsymbol{x})\boldsymbol{a}(x_b, t^n) = \sum_b N^b(\boldsymbol{x})\boldsymbol{a}_b^n \tag{43}$$

The velocity at the mesh nodes is initiated from the initial particle velocity $\boldsymbol{U}_\lambda^0$ through the projection:

$$\left\{\hat{\boldsymbol{u}}_b^0\right\} = P^0_{\{\lambda\}}\left[\{\boldsymbol{U}_\lambda^0\}\right] = P^0_{\{\lambda\}}\left[\{\boldsymbol{u}^0(\boldsymbol{X}_\lambda^0)\}\right] = \left\{\boldsymbol{u}_b^0 + \boldsymbol{\epsilon}^{P,0}\right\} \tag{44}$$

Where $\boldsymbol{\epsilon}^{P,0}$ is the projection error. The projected velocity is then interpolated back to the particles using the projected values:

$$\hat{\boldsymbol{u}}^0(\boldsymbol{X}_\lambda^0) = \sum_b N^b(\boldsymbol{x})\hat{\boldsymbol{u}}_b^0 = \sum_b N^b(\boldsymbol{x})\left(\boldsymbol{u}_b^0 + \boldsymbol{\epsilon}^{P,0}\right) = \boldsymbol{u}_h^0(\boldsymbol{X}_\lambda^0) + \boldsymbol{\epsilon}_h^{P,0}(\boldsymbol{X}_\lambda^0) \tag{45}$$

Then the interpolation error has to be taken into account (both for the velocity and acceleration):

$$\boldsymbol{u}_h^0(\boldsymbol{X}_\lambda^0) = \boldsymbol{u}^0(\boldsymbol{X}_\lambda^0) + \boldsymbol{\epsilon}_h^{u,0}(\boldsymbol{X}_\lambda^0)$$
$$\boldsymbol{a}_h^0(\boldsymbol{X}_\lambda^0) = \boldsymbol{a}^0(\boldsymbol{X}_\lambda^0) + \boldsymbol{\epsilon}_h^{a,0}(\boldsymbol{X}_\lambda^0) \tag{46}$$

Finally, the Verlet integrator is used to obtain the particles' position and velocities:

$$\tilde{\boldsymbol{X}}_\lambda^1 = \boldsymbol{X}_\lambda^0 + \Delta t\,\boldsymbol{u}^0\left(\boldsymbol{X}_\lambda^0\right) + \Delta t\,\epsilon^{\boldsymbol{u},0}\left(\boldsymbol{X}_\lambda^0\right) + \Delta t\,\boldsymbol{\epsilon}_h^{P,0}\left(\boldsymbol{X}_\lambda^0\right) + \frac{\Delta t^2}{2}\boldsymbol{a}^0\left(\boldsymbol{X}_\lambda^0\right) + \frac{\Delta t^2}{2}\epsilon^{\boldsymbol{a},0}\left(\boldsymbol{X}_\lambda^0\right) \tag{47}$$

$$\tilde{\boldsymbol{U}}_\lambda^1 = \boldsymbol{U}_\lambda^0 + \frac{\Delta t}{2}\left(\boldsymbol{a}^0\left(\boldsymbol{X}_\lambda^0\right) + \boldsymbol{a}^1\left(\tilde{\boldsymbol{X}}_\lambda^1\right)\right) + \frac{\Delta t}{2}\left(\epsilon^{\boldsymbol{a},0}\left(\boldsymbol{X}_\lambda^0\right) + \epsilon^{\boldsymbol{a},1}\left(\tilde{\boldsymbol{X}}_\lambda^1\right)\right) \tag{48}$$

Considering that the projector and the interpolator are second order accurate, and using Eq. (8)-(9):

$$\tilde{\boldsymbol{X}}_\lambda^1 = \boldsymbol{X}_\lambda^0 + \Delta t\,\boldsymbol{u}^0\underbrace{\left(\boldsymbol{X}_\lambda^0\right) + \frac{\Delta t^2}{2}\boldsymbol{a}^0\left(\boldsymbol{X}_\lambda^0\right)}_{\hat{\boldsymbol{X}}_\lambda^1 = \boldsymbol{X}_\lambda^1 + O(\Delta t^3)} + \Delta t\underbrace{\epsilon^{\boldsymbol{u},0}\left(\boldsymbol{X}_\lambda^0\right)}_{O(\Delta t\, h_e^2)} + \Delta t\underbrace{\boldsymbol{\epsilon}_h^{P,0}\left(\boldsymbol{X}_\lambda^0\right)}_{O(\Delta t\, h_e^2)} + \underbrace{\frac{\Delta t^2}{2}\epsilon^{\boldsymbol{a},0}\left(\boldsymbol{X}_\lambda^0\right)}_{O(\Delta t^2 h_e^2)} \tag{49}$$

$$\tilde{\boldsymbol{U}}_\lambda^1 = \boldsymbol{U}_\lambda^0 + \underbrace{\frac{\Delta t}{2}\left(\boldsymbol{a}^0\left(\boldsymbol{X}_\lambda^0\right) + \boldsymbol{a}^1\left(\hat{\boldsymbol{X}}_\lambda^1\right)\right)}_{\hat{\boldsymbol{U}}_\lambda^1 = \boldsymbol{U}_\lambda^1 + O(\Delta t^3)} + \underbrace{\frac{\Delta t}{2}\left(\boldsymbol{a}^1\left(\tilde{\boldsymbol{X}}_\lambda^1\right) - \boldsymbol{a}^1\left(\hat{\boldsymbol{X}}_\lambda^1\right)\right)}_{\sim\frac{\Delta t}{2}(h_e^2\Delta t)\nabla\boldsymbol{a}^1} + \underbrace{\frac{\Delta t}{2}\left(\boldsymbol{\epsilon}^{\boldsymbol{a},0}\left(\boldsymbol{X}_\lambda^0\right) + \boldsymbol{\epsilon}^{\boldsymbol{a},1}\left(\tilde{\boldsymbol{X}}_\lambda^1\right)\right)}_{O(\Delta t\, h_e^2)} \tag{50}$$

Which reads:

$$\tilde{\boldsymbol{X}}_\lambda^1 = \hat{\boldsymbol{X}}_\lambda^1 + \Delta t O\left(h_e^2\right) = \boldsymbol{X}_\lambda^1 + O\left(\Delta t^3\right) + O\left(h_e^2\Delta t\right) \tag{51}$$

$$\tilde{\boldsymbol{U}}_\lambda^1 = \hat{\boldsymbol{U}}_\lambda^1 + \Delta t\,\boldsymbol{O}\left(h_e^2\right) = \boldsymbol{U}_\lambda^1 + O\left(\Delta t^3\right) + O\left(h_e^2\Delta t\right) \tag{52}$$

Eqs. (51)-(52) show that the error for the first time step is $O\left(\Delta t^3\right) + O\left(h_e^2\Delta t\right)$. It is straightforward to show that the resulting errors for the following time steps are those related to the accumulation, which reads:

$$\tilde{\boldsymbol{X}}_\lambda^{n+1} = \boldsymbol{X}_\lambda^n + (n+1)O\left(\Delta t^3\right) + (n+1)O\left(h_e^2\Delta t\right) = \boldsymbol{X}_\lambda^n + t^{n+1}\left(O\left(\Delta t^2\right) + O\left(h_e^2\right)\right) \tag{53}$$

$$\tilde{\boldsymbol{U}}_\lambda^{n+1} = \boldsymbol{U}_\lambda^n + (n+1)O\left(\Delta t^3\right) + (n+1)O\left(h_e^2\Delta t\right) = \boldsymbol{U}_\lambda^n + t^{n+1}\left(O\left(\Delta t^2\right) + O\left(h_e^2\right)\right) \tag{54}$$

resulting in a second order scheme in space and time.

It is obvious that, when solving the Navier-Stokes equations, the exact acceleration field is not known. On the contrary, it must be estimated. In this work, a FE Galerkin method is used to solve the pressure and viscous terms, which is a second order accurate method [9]. Then, this error is also interpolated to the particles and has to be added to the interpolation error considered previously. However, this error does not degrades the rate of convergence of the method since it is second order, too.

As a conclusion, the accumulated error of the projected velocity is:

$$\left\{\hat{\boldsymbol{u}}_b^{n+1}\right\} = P_{\{\lambda\}}^n\left[\left\{\tilde{\boldsymbol{U}}_\lambda^{n+1}\right\}\right] = P_{\{\lambda\}}^0\left[\left\{\boldsymbol{U}_\lambda^{n+1} + t^n\left(O\left(\Delta t^2\right) + O\left(h_e^2\right)\right)\right\}\right] \tag{55}$$

Then:

$$\left\{\hat{\boldsymbol{u}}_b^{n+1}\right\} = \left\{\boldsymbol{u}_b^{n+1}\right\} + \left\{\underbrace{\boldsymbol{\epsilon}_b^{P,n+1}}_{O\left(h_e^2\right)}\right\} + \left\{\underbrace{\boldsymbol{e}_b^{n+1}}_{t^{n+1}\left(O\left(\Delta t^2\right)+O\left(h_e^2\right)\right)}\right\} \tag{56}$$

7. Convergence analyses

This section presents different examples showing the rate of convergence of the SL-PFEM presented.

1.4 Linear wave trajectories

As it was pointed out in the introduction, for non-dominant convective flows the explicit streamlines computed at the beginning of the time step are not good approximants to pathlines and therefore small time steps are required to achieve accurate results [5]. The flow field induced by a linear wave is a good example used to demonstrate this point. And, if information regarding the acceleration is introduced, the required time step can be increased (see Figure 1).

In this section, the velocity Verlet algorithm is used to estimate the trajectory of a particle moving in the acceleration field induced by the velocities of a linear wave. The 2D velocity potential of an Airy wave with mean free surface located at $y = 0$ is given by:

$$\phi(x,y,t) = \frac{Ag}{\omega}\frac{\cosh\left(K(H+y)\right)}{\cosh\left(KH\right)}\sin\left(Kx - \omega t\right) \tag{57}$$

Where ϕ is the velocity potential, A is the wave amplitude, $K = 2\pi/L$ is the wave number and L is the wave length, H is the water depth, $\omega = 2\pi/T$ is the wave angular frequency and T is the wave period, x and y are the vertical and horizontal spatial coordinates, t is time. The velocity field is obtained as the gradient of the velocity potential $(u_x, u_y) = (\phi_x, \phi_y)$. Then, the corresponding acceleration field is obtained via $(a_x, a_y) = d_t(u_x, u_y)$. Figure 1 shows the velocity field induced by a linear wave. Using the velocity Verlet integrator we obtain the following equations for the position of the particle and its velocity components:

$$X_\lambda^{n+1} = X_\lambda^n + \Delta t\,\phi_x(\boldsymbol{X}_\lambda^n, t^n) + \frac{\Delta t^2}{2}\left(\phi_{xt}\left(\boldsymbol{X}_\lambda^n, t^n\right) + \phi_x\left(\boldsymbol{X}_\lambda^n, t^n\right)\phi_{xx}\left(\boldsymbol{X}_\lambda^n, t^n\right) + \phi_y(\boldsymbol{X}_\lambda^n, t^n)\phi_{xy}\left(\boldsymbol{X}_\lambda^n, t^n\right)\right) \tag{58}$$

$$Y_\lambda^{n+1} = Y_\lambda^n + \Delta t\,\phi_y(\boldsymbol{X}_\lambda^n, t^n) + \frac{\Delta t^2}{2}\left(\phi_{yt}\left(\boldsymbol{X}_\lambda^n, t^n\right) + \phi_x\left(\boldsymbol{X}_\lambda^n, t^n\right)\phi_{yx}\left(\boldsymbol{X}_\lambda^n, t^n\right) + \phi_y(\boldsymbol{X}_\lambda^n, t^n)\phi_{yy}\left(\boldsymbol{X}_\lambda^n, t^n\right)\right) \tag{59}$$

$$\frac{U_\lambda^{n+1} - U_\lambda^n}{\Delta t} = \frac{1}{2}\left(\phi_{xt}\left(\boldsymbol{X}_\lambda^n, t^n\right) + \phi_x\left(\boldsymbol{X}_\lambda^n, t^n\right)\phi_{xx}\left(\boldsymbol{X}_\lambda^n, t^n\right) + \phi_y(\boldsymbol{X}_\lambda^n, t^n)\phi_{xy}\left(\boldsymbol{X}_\lambda^n, t^n\right)\right) + \frac{1}{2}\left(\phi_{xt}\left(\boldsymbol{X}_\lambda^{n+1}, t^{n+1}\right) + \phi_x\left(\boldsymbol{X}_\lambda^{n+1}, t^{n+1}\right)\phi_{xx}\left(\boldsymbol{X}_\lambda^{n+1}, t^{n+1}\right) + \phi_y(\boldsymbol{X}_\lambda^{n+1}, t^{n+1})\phi_{xy}\left(\boldsymbol{X}_\lambda^{n+1}, t^{n+1}\right)\right) \tag{60}$$

$$\frac{V_\lambda^{n+1} - V_\lambda^n}{\Delta t} = \frac{1}{2}\left(\phi_{yt}\left(\boldsymbol{X}_\lambda^n, t^n\right) + \phi_y\left(\boldsymbol{X}_\lambda^n, t^n\right)\phi_{yx}\left(\boldsymbol{X}_\lambda^n, t^n\right) + \phi_y(\boldsymbol{X}_\lambda^n, t^n)\phi_{yy}\left(\boldsymbol{X}_\lambda^n, t^n\right)\right) + \frac{1}{2}\left(\phi_{yt}\left(\boldsymbol{X}_\lambda^{n+1}, t^{n+1}\right) + \phi_y\left(\boldsymbol{X}_\lambda^{n+1}, t^{n+1}\right)\phi_{yx}\left(\boldsymbol{X}_\lambda^{n+1}, t^{n+1}\right) + \phi_y(\boldsymbol{X}_\lambda^{n+1}, t^{n+1})\phi_{yy}\left(\boldsymbol{X}_\lambda^{n+1}, t^{n+1}\right)\right) \tag{61}$$

The velocity Verlet has been used to estimate the particle trajectory due to a linear wave with $A = 0.01m$, $H = 0.1m$, $L = 1m$, and $T = 1.0726s$ and with initial position $x_0 = 0.5m$ and $y_0 = -0.02m$. The trajectory has been estimated for several time steps ($\Delta t = T/10$, $\Delta t = T/20$, $\Delta t = T/40$, and $\Delta t = T/80$), using the analytical solution for the initial velocity. Figure 3 compares the analytical solution of the trajectory for one wave period with the numerical ones. Comparing the final position of the analytical solution and the numerical ones, an error estimate is obtained. We use this error to carry out a convergence analysis (Table **1**).

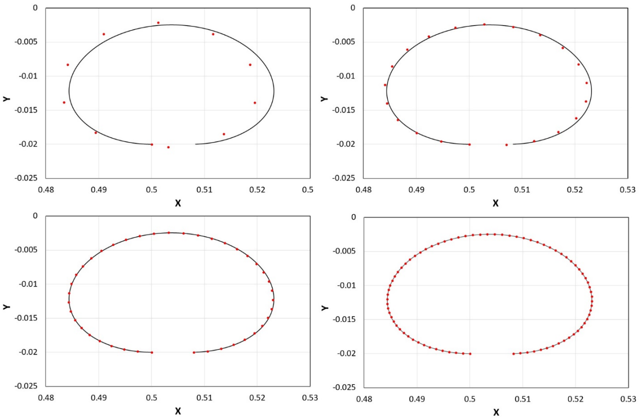


Figure 3: Particle trajectory integration in the wave problem using the velocity Verlet algorithm. Black solid line represents the analytical solution. Red dots represent de numerical solution for $\mathbf{\Delta t}=\boldsymbol{T}/5$, $\mathbf{\Delta t}=\boldsymbol{T}/10$, $\mathbf{\Delta t}=\boldsymbol{T}/20$, $\mathbf{\Delta t}=\boldsymbol{T}/40$, and $\mathbf{\Delta t}=\boldsymbol{T}/80$.

Table 1: Values and errors for the position and velocity of a particle at $=\boldsymbol{T}$.

	$x(T)$	$y(T)$	$Error$ X	vx(T)	vy(T)	Error V
Analytical	0.508308	-0.020000		-0.09850702	-0.00238937	
$\Delta t=T/10$	0.503153	-0.020397	0.00517037	-0.09850997	-0.00178868	0.00060070
$\Delta t=T/20$	0.507039	-0.020091	0.00127241	-0.09851727	-0.00222848	0.00016122
$\Delta t=T/40$	0.507992	-0.020015	0.00031649	-0.09851019	-0.00234849	0.00004100
$\Delta t=T/80$	0.508229	-0.019997	7.9029E-05	-0.09850785	-0.00237911	0.00001029

Now, we will integrate Eqs. (1)-(2) using the semi-Lagrangian approach given the acceleration field induced by a linear wave. The acceleration and velocities at the mesh nodes ($\boldsymbol{a}_h^0$ and $\boldsymbol{u}_h^0$) are initialized with the analytical values. The particle's initial velocities and accelerations are interpolated from the mesh initial values ($\boldsymbol{U}_\lambda^0=\boldsymbol{u}_h^0(\boldsymbol{X}_\lambda^0)$ and $\boldsymbol{A}_\lambda^0=\boldsymbol{a}_h^0(\boldsymbol{X}_\lambda^0)$). Afterwards, during the simulation the acceleration field is imposed on the mesh nodes and interpolated at the particles, while the velocities are obtained via the semi-Lagrangian approach. The algorithm reads as follows:

Step 1: Move particles from $t=n\Delta t$ to $t=(n+1)\Delta t$:

$$\boldsymbol{X}_\lambda^{n+1}=\boldsymbol{X}_\lambda^n+\Delta t\,\boldsymbol{u}_h^n(\boldsymbol{X}_\lambda^n)+\frac{\Delta t^2}{2}\boldsymbol{a}_h^n(\boldsymbol{X}_\lambda^n)$$

Step 2: Interpolate the particle's acceleration at the new position:

$$\boldsymbol{A}_\lambda^{n+1}=\boldsymbol{a}_h^{n+1}(\boldsymbol{X}_\lambda^{n+1})$$

Step 3: The new particle´s velocity is updated:

$$\boldsymbol{U}_\lambda^{n+1}=\boldsymbol{U}_\lambda^n+\frac{1}{2}\Delta t\left(\boldsymbol{A}_\lambda^n+\boldsymbol{A}_\lambda^{n+1}\right)$$

Step 4: The particle´s velocities are projected onto the mesh to obtain the new velocity field on the background mesh:

$$\boldsymbol{u}_h^{n+1}=P_{\{\lambda\}}^{n+1}\left[\{\boldsymbol{U}_\lambda^{n+1}\}\right]$$

In order to achieve second order convergence, each step must be at least second order in time and space. Step 1 has already been demonstrated that is second order. Step 2 interpolates from the mesh values to the particles using the FE linear shape functions. This interpolation is second order as demonstrated in [3]. Step 3 integrates the velocity in time with a Crank-Nicolson scheme, which is known to be second order in time. And finally, Step 4 is the projection step using the global least square projector which, in the previous section has been demonstrated to be second order in space and time.

The time step has been setup such that $h/\Delta t=4.661$, being h the characteristic mesh size. Table 2 provides the values of the mean quadratic error (MQE) of the velocity on the mesh and Figure 4 shows that the rate of convergence is of second order, as expected.

Table 2: Mean quadratic error of the velocity on the mesh at $\boldsymbol{t}=\boldsymbol{T}$

	$RMSE=\sqrt{\sum_{i=1}^{N}(\boldsymbol{u}_h(\boldsymbol{x}_i)-\boldsymbol{u}_a(\boldsymbol{x}_i))^2/N}$
$h=H/2$	1.226E-03
$h=H/4$	2.345E-04
$h=H/6$	1.176E-04
$h=H/8$	6.951E-05
$h=H/10$	4.742E-05

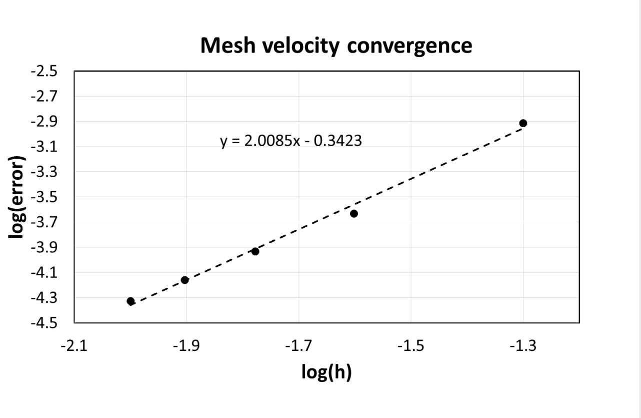


Figure 4: Convergence analysis for the velocity on the mesh for a linear wave using the semi-Lagrangian approach.

1.5 Lid driven cavity flow.

The particulars for this case study are given in Table 3, and Figure 5 shows the discretization used and the boundary conditions applied. An average of 2 iterations is only required for convergence.

Although the steady state solution is achieved after one thousand time steps, the simulation time has been set to $T = 10^5 \Delta t$ to verify whether the steady state solution shows symptoms of error accumulation.

The results obtained with the present second-order Verlet-SLPFEM are compared to those obtained using a high resolution spectral method in [11]. Also results are shown for a first-order SL-PFEM based on the Backward Euler integrator for the velocity equation. Figure 6 compares the streamlines and Figure 7 compares the horizontal velocity and pressure at the mid-section. While the second-order SL-PFEM based on the Verlet algorithm provides very similar results to those of [11], the first-order SL-PFEM based on the Euler scheme produces a highly diffusive solution on the pressure and velocity. Despite being a steady-state problem the first-order method is incapable of achieving the right solution with the current time step. It would require a significant reduction of the time step, which leads to a significant increase of CPU time. On the other hand, the second-order scheme produces an accurate solution for the current time step.

Figure 8 shows the evolution of the horizontal velocity at the node located at (x,y)=(0.5,0.175). It is observed that an average of 0.392 meters per second is obtained with a variance of 0.001 m/s approximately. No symptoms of error accumulation are observed.

Table 3: particular of the lid driven cavity flow

Lid velocity V	$-1m/s$
Reynolds number Re	1000
Domain size	$1m\ x\ 1m$
Domain discretization	$80x80$
Number of triangle elements	25600
Number of nodes	12961
Average number of particles per element	3
Mesh size $\Delta x = \Delta y$	0.0125
Time step Δt	0.1
Courant number	8
Simulation time	$10^5 \Delta t$
Sampling time	$10^2 \Delta t$

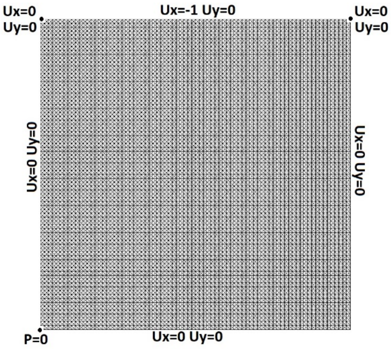
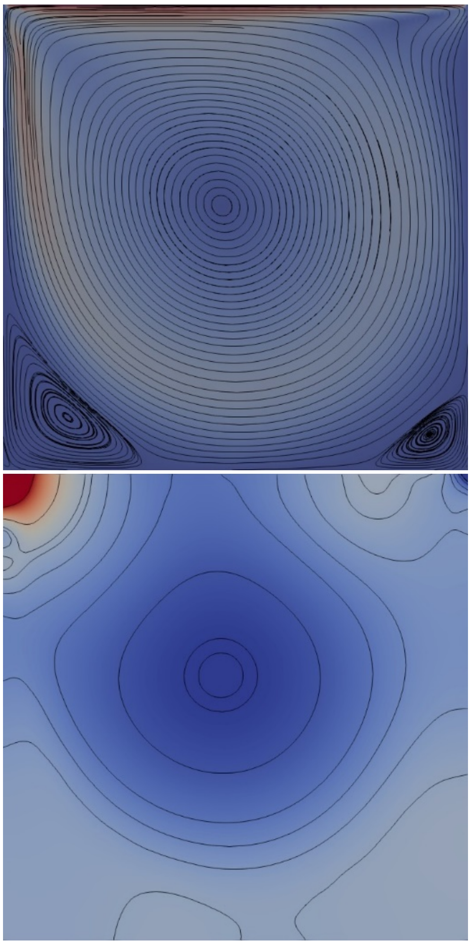


Figure 5: Structured mesh for lid driven cavity flow and boundary conditions.



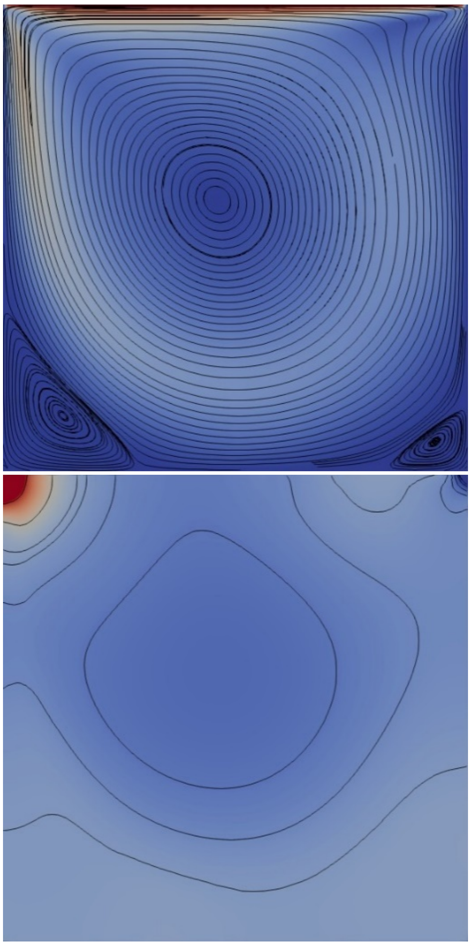


Figure 6: Streamlines comparison. Up: Second order Verlet-SLPFEM. Down: First Order Euler SL-PFEM.

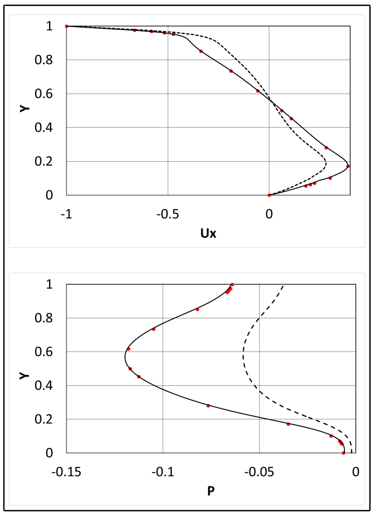


Figure 7: Horizontal velocity and pressure profiles at the mid-section x=0.5. Red dots: spectral method [11]. Solid line: Second Order Verlet-SLPFEM. Dash line: First Order Euler-SLPFEM.

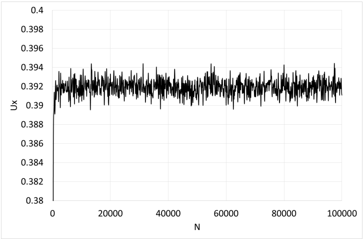


Figure 8: Horizontal velocity evolution at (x,y)=(0.5,0.175) for the second-order Verlet -SLPFEM

1.6 Two-dimensional Taylor-Green vortex.

The two-dimensional Taylor-Green vortex problem is used to evaluate the convergence order of the numerical strategy solving an incompressible flow problem for which an analytical solution is known. The analytical solution is given by:

$$u_x(x,y,t) = -\sin(x)\cos(y)e^{-2\nu t} \qquad (62)$$

$$u_y(x,y,t) = \cos(x)\sin(y)e^{-2\nu t}$$

$$P(x,y,t) = \frac{1}{4}[\cos(2x) + \cos(2y)]e^{-4\nu t}$$

where $\nu = 0.001m^2/s$ is the kinematic viscosity. The problem is solved using a square domain $[0,\pi]x[0,\pi]$ and setting slip boundary conditions for the velocity. Also, to prevent the pressure to be undetermined, the following condition is imposed at $(x,y) = (0,0)$:

$$P(0,0,t) = \frac{1}{2}e^{-4\nu t} \qquad (63)$$

The velocity and pressure field are initialized at $t = 0$ with the analytical solution. The simulation is carried out until a final time of $t = 10s$ is reached. Table 4 provides the values of the mesh size, time step, and the number of time steps computed. Figure 9 shows the mesh used for case 2.

Table 4: Numerical parameters use in the 2D Taylor-Green Vortex

Case	Mesh size	Time step	Number of time steps
1	$\Delta x = \pi/8$	$\Delta t = 0.2s$	50
2	$\Delta x = \pi/16$	$\Delta t = 0.1s$	100
3	$\Delta x = \pi/24$	$\Delta t = 0.0667s$	150
4	$\Delta x = \pi/32$	$\Delta t = 0.05s$	200
5	$\Delta x =$	$\Delta t =$	300

	$\pi/48$	$0.0333s$	
6	Δx = $\pi/64$	Δt = $0.025s$	400

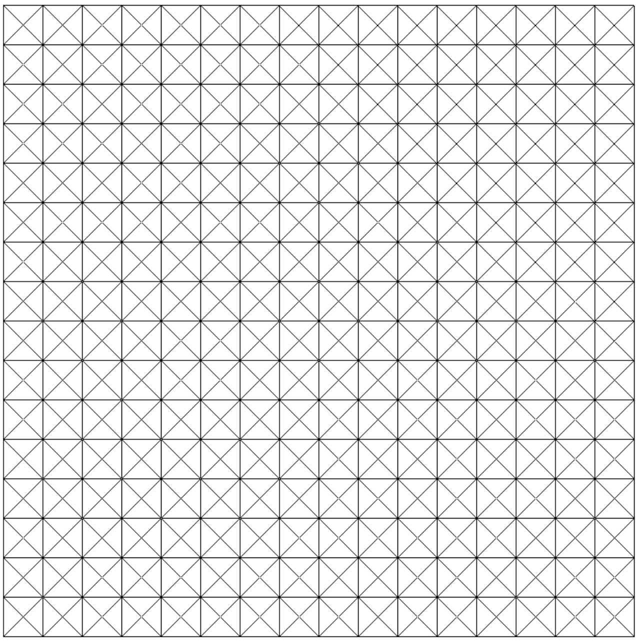


Figure 9: Structured mesh with $\mathbf{\Delta x} = \boldsymbol{\pi}/16$.

Figure 10 shows the convergence of the proposed algorithm. It can be seen that both, pressure and velocity converge with second order convergence rate.

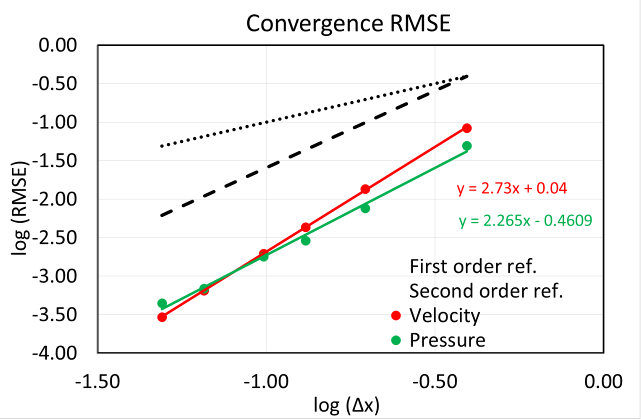


Figure 10: Convergence analysis for the 2D Taylor-Green Vortex. Black dashed line represents first order convergence. Black dotted line represents second order convergence. Green: pressure errors. Red: velocity errors.

1.7 Two-dimensional Steady Taylor-Green vortex.

A modified version of the two-dimensional Taylor-Green vortex problem is used to evaluate the convergence order of the numerical strategy solving an incompressible flow problem for which an analytical solution is known. In this version, a mass force is introduced to replace the energy dissipated by the viscous terms, so that a steady solution is obtained. The mass force to be introduced is:

$$f_x(x,y) = -2\nu\sin(x)\cos(y) \qquad (64)$$
$$f_y(x,y) = 2\nu\cos(x)\sin(y)$$

and the analytical solution to the problem is:

$$u_x(x,y) = -\sin(x)\cos(y) \qquad (65)$$
$$u_y(x,y) = \cos(x)\sin(y)$$
$$P(x,y) = \frac{1}{4}[\cos(2x) + \cos(2y)]$$

The problem is solved with $\nu = 0.01m^2/s$ ($Re = 314$). The fluid domain is a square of $[0,\pi]x[0,\pi]$. Slip boundary conditions are used for the velocity. Also, to prevent the pressure to be undetermined, the following Dirichlet condition is imposed:

$$P(0,0) = 0.5 \qquad (66)$$

The velocity and pressure field are initialized with the analytical solution. And average of 2 iterations is only required for convergence. The simulation is carried out for $t = 400s$. Based on the maximum velocity ($u_{max} = 1$), the Courant number used is $Cr = \Delta t/\Delta x = 2.04$. The case is initiated with 3 particles per element, and particles are removed when it exceeds 6 particles per element. At the end of the simulation the average number of particles per element is slightly lower than 3.

Table 5 provides the values of the mesh size, time step, and the number of time steps computed. Figure 11 shows an intermediate mesh used in the analysis. Figure 12 how the pressure on the mesh and the particle's velocity obtained for case 64. Both results are quite close to the analytical solution. For this case, the root mean square error is in the order of 0.003 for both, velocity and pressure.

Table 5: Numerical parameters use in the 2D Steady Taylor-Green Vortex

Case	Mesh size	Time step	Number of time steps
16	Δx = $\pi/16$	$\Delta t = 0.4s$	1000
24	Δx = $\pi/24$	Δt = $0.26667s$	1500
32	Δx = $\pi/32$	$\Delta t = 0.2s$	2000
48	Δx = $\pi/48$	Δt = $0.13333s$	3000
64	Δx = $\pi/64$	$\Delta t = 0.1s$	4000

96	$\Delta x = \pi/96$	$\Delta t = 0.06667s$	6000
128	$\Delta x = \pi/128$	$\Delta t = 0.05s$	8000
192	$\Delta x = \pi/192$	$\Delta t = 0.03333s$	12000

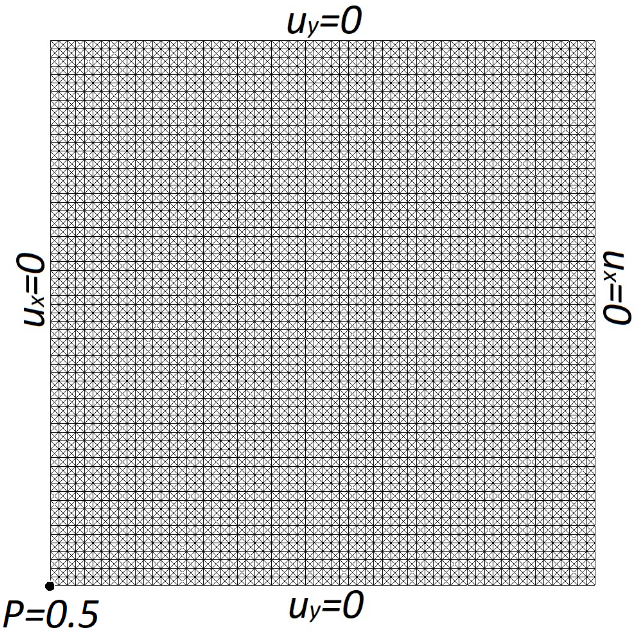
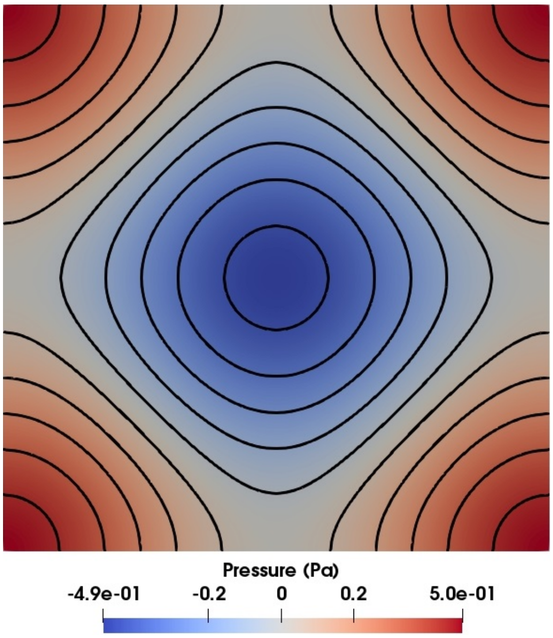


Figure 11: Structured mesh with $\Delta x = \pi/64$ and boundary conditions applied



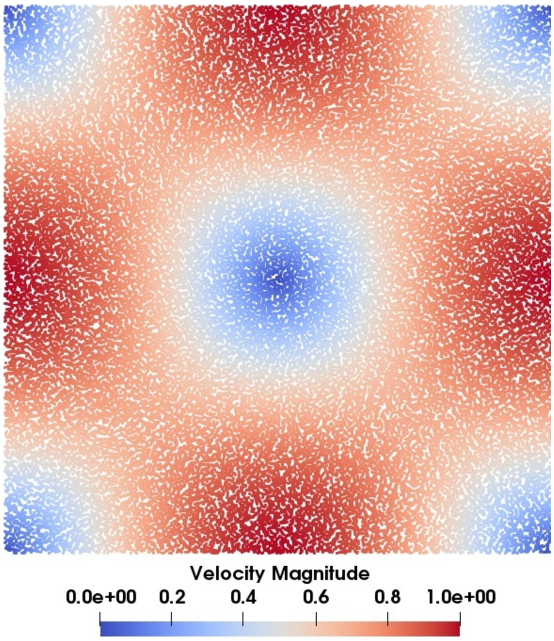


Figure 12: Case 64 results. Pressure (left) and velocity (right)

Figure 13 shows the evolution of the errors along the simulation. It is observed that the errors achieved a constant value. Hence no symptoms of error accumulation over time is appreciated. Figure 14 provides the rate of convergence obtained using an average of three particles per element. The root mean square error (RMSE) has been used for the analysis. Table 6 shows the convergence of the proposed algorithm for pressure and velocity.

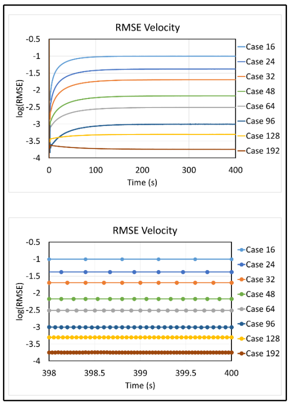


Figure 13: Error evolution along time.

Table 6: Rate of convergence. Root mean square error (RMSE) and maximum error (MaxE)

Convergence Rate	
Velocity	Pressure
2.59	2.66

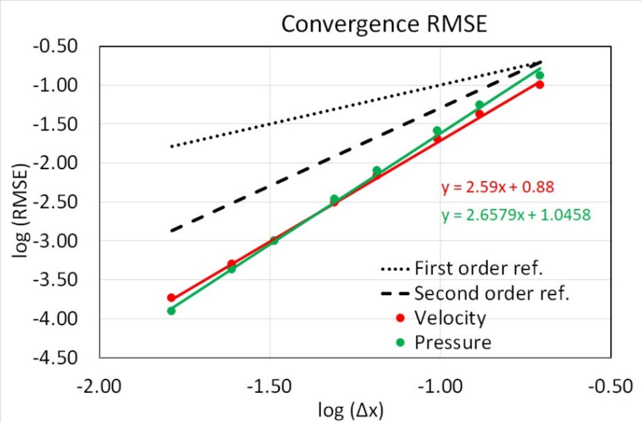


Figure 14: Convergence analysis for the 2D Taylor-Green Vortex. Black dashed line represents first order convergence. Black dotted line represents second order convergence.

Blue: pressure errors. Red: velocity errors.

While in an Eulerian method all dependent variables remain unchanged once the steady state solution is reached, this is not the case in the SL-PFEM method. From the point of view of the particles, the flow is never steady since the intrinsic variables they transport are continually changing.

If a SL-PFEM is not properly tailored, then the error accumulation will make the method to be incapable of computing a steady-state solution. This is because the accumulation of errors along the simulation time will eventually cause a numerical blow up. The present SL-PFEM formulation is capable of computing steady state solutions without errors concerns.

1.8 Three-dimensional flow past a cylinder.

The three dimensional flow around a cylinder is simulated to assess the suitability of the present method for solving three dimensional problems. The particulars of the problem are given in Table 7. An average of 2-3 iterations are only required for convergence.

For a Reynolds number of 200 and an aspect ratio of the cylinder height (H) to the cylinder diameter (D) of H/D=1, the flow behaves as a two dimensional flow [12]. A three dimensional structured mesh with five levels of refinement has been used, keeping the Courant number constant. Figure 15 shows the computational domain used, and Figure 16 shows the corresponding meshes used in this analysis. The particulars for each mesh are provided in Table 8, as well as the total number of time steps simulated. In average, two particles per elements are used.

Table 7: Flow past a cylinder particulars

Cylinder diameter (D)	1 *m*
Cylinder height (H)	1 *m*
Inlet velocity (V)	1 *m/s*
Viscosity (ν)	0.005
Re	200

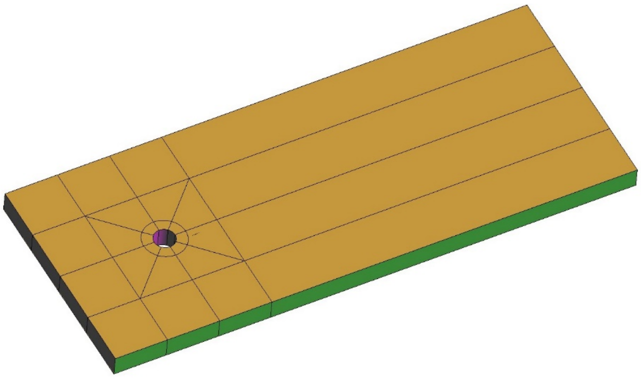


Figure 15: Computational domain used to generate the structured mesh.

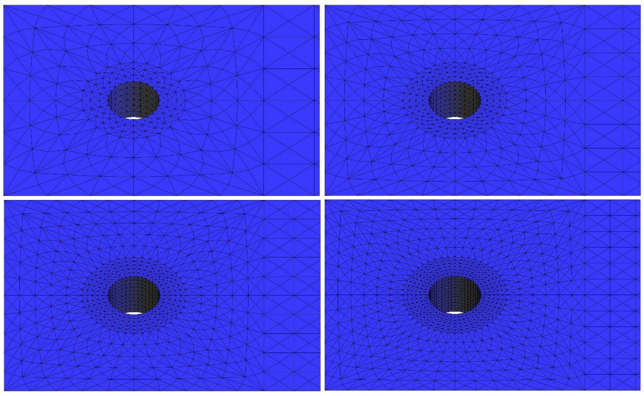


Figure 16: Computational mesh refinement for the 3D flow past a cylinder.

Table 8: Numerical data used in the 3D flow past a cylinder

Mesh	Δx	Δt	*N tetras*	*N Nodes*	*N* Δt
1	0.1333	0.0667	62208	14487	1500
2	0.1	0.05	147456	33412	2000
3	0.08	0.04	288000	64185	2500
4	0.0667	0.0333	497664	109686	3000

Table 9 provides the Strouhal numbers obtained in each case study. It is observed how the lift force amplitude and the Strouhal number converge towards values of 0.27 and 0.196, respectively. Figure 17 plots the logarithmic errors for the Strouhal number versus the logarithmic mesh size for each case study. Comparing with the second order reference line, it is observed that the convergence rate of the Strouhal number is close to second order. Figure 18 and Figure 19 provides a snapshot of the pressure field, particles velocities, streamlines and vorticity for Case 2.

Table 9: Numerical results for the 3D flow past a cylinder

Case	*St*	*Error St*	C_L
1	0.133	0.0630	0.177
2	0.162	0.0339	0.255
3	0.178	0.0184	0.246
4	0.186	0.0101	0.258
Ref. [	0.1		

12]	96		

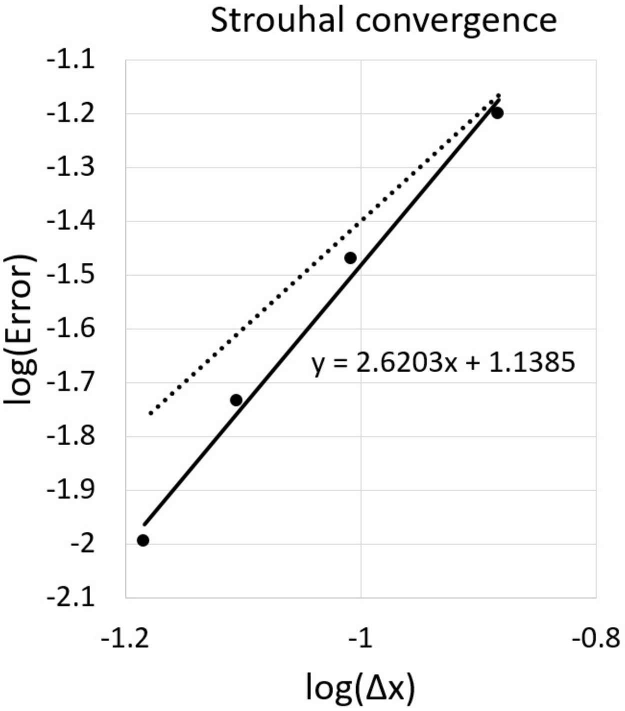


Figure 17: Rate of convergence for the Strouhal number. Circles represent the log (error) for each case study. Solid line represents the trend of the convergence. Dot line represents second order convergence.

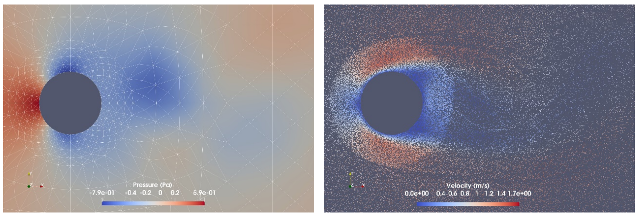

Pressure (Pa)
-7.9e-01 -0.4 -0.2 0 0.2 5.9e-01
Velocity (m/s)
0.0e+00 0.4 0.6 0.8 1 1.2 1.4 1.7e+00

Figure 18: Case 2 snapshots: top view of pressure map on mesh (left) and particles velocities (right).

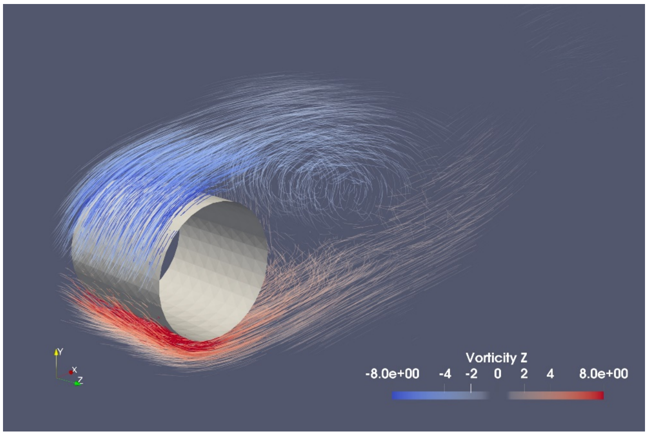


Figure 19: Case 2 snapshots: oblique view of vorticity on particles.

Conclusions

In this paper, a second order semi-Lagrangian (SL) method for solving the incompressible Navier-Stokes equations has been presented within the framework of the Particle Finite Element Method (PFEM). The method is based on the velocity Verlet integrator, which offers an explicit formulation for particle tracking with good numerical properties. Therefore, particles only need to be transported once per time step, which makes this integrator very attractive for the semi-Lagrangian PFEM (SL-PFEM). The algorithm is completed with a predictor-multicorrector scheme for integrating the velocity correction using the Finite Element Method. A second order global least square projector is used to make sure that the coherence condition is fulfilled, which avoids the need of iterating and projecting more than once per time step. The proposed method has been tested for different types of flows in two and three dimensions, and the second order rate of convergence has been achieved. Moreover, it has been shown that there are no symptoms of error accumulation along time, which is a major concern for semi-Lagrangian schemes because it could compromise the rate of convergence and its capability to compute steady-state problems.

Acknowledgements

This work relates to Department of the Navy Grant N62909-16-1-2236 issued by Office of Naval Research Global. The United States Government has a royalty-free licence throughout the world in all copyrightable material contained herein.

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest

References

[1] S.R. Idelsohn, J. Marti, P. Becker, E. Oñate: Analysis of multifluid flows with large time steps using the particle finite element method. International Journal for Numerical Methods in Fluids 75(9), 621–644 (2014). DOI 10.1002/fld.3908.

[2] S. R. Idelsohn, N. Nigro, A. Limache, E. Oñate: Large time-step explicit integration method for solving problems with dominant convection. Computer Methods in Applied Mechanics and Engineering 217-220, 168–185 (2012). DOI 10.1016/j.cma.2011.12. 008.

[3] P. Becker, S. R. Idelsohn, E. Oñate. A unified monolithic approach for multi-fluid flows and fluidâASstructure interaction using the Particle Finite Element Method with fixed mesh. Computational Mechanics (2014). DOI 10.1007/s00466-014-1107-0

[4] J. M. Gimenez, H. J. Aguerrea, S. R. Idelsohnd, N. M. Nigro. A space-time second-order particle-based method to solve ow problems on arbitrary meshes. Accepted for publication in Journal of Computational Physics. DOI 10.1016/j.jcp.2018.11.034

[5] P. Nadukandi, B. Servan-Camas, P. A. Becker, and J. Garcia-Espinosa. Seakeeping with the semi-Lagrangian Particle Finite Element Method. Comp. Part. Mech. (2017) 4: 321. DOI 10.1007/s40571-016-0127-2

[6] L. Verlet. Computer "Experiments" on Classical Fluids. I. Thermodynamical properties of Lenard-Jones Molecules. Phys. Rev. E. 59(5), 98-103. (1967)

[7] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. J. Chem. Phys. 76, 637 (1982). DOI 10.1063/1.442716.

[8] E. Hairer, C. Lubich and G. Wanner (2003). Geometric numerical integration illustrated by the Störmer–Verlet method. Acta Numerica, 12, pp 399-450. DOI 10.1017/S0962492902000144

[9] J. Garcia-Espinosa, A. Valls, and E. Oñate. ODDLS: A new unstructured mesh finite element method for the analysis of free surface flow problems Int. J. Numer. Meth. Engng 76(9) 1297-1470 (2008).

[10] E. Oñate , J. García-Espinosa, S.R. Idelsohn, F. Del Pin. Finite calculus formulations for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches. Comput. Methods Appl. Mech. Engrg. 195 3001–3037 (2006).

[11] O. Botella and R. Peyret. Benchmark spectral results on the lid-driven cavity flow. Computers & Fluids Vol. 27, No. 4, pp. 421-433, 1998.

[12] H. Jiang and L. Cheng. Strouhal Reynolds number relationship for flow past a circular cylinder. J. Fluid Mech. (2017), vol. 832, pp. 170-188.