A second-order in time and space particle-based method to solve flow problems on arbitrary meshes

Article *in* Journal of Computational Physics · November 2018

DOI: 10.1016/j.jcp.2018.11.034

4 authors, including:

Juan M. Gimenez
National Scientific and Technical Research Council
34 PUBLICATIONS **92** CITATIONS

SEE PROFILE

Sergio Rodolfo Idelsohn
Catalan Institution for Research and Advanced Studies
407 PUBLICATIONS **5,753** CITATIONS

SEE PROFILE

Norberto Nigro
National Scientific and Technical Research Council
208 PUBLICATIONS **959** CITATIONS

SEE PROFILE

A second-order in time and space particle-based method to solve flow problems on arbitrary meshes

The final publication is available at Elsevier via https://doi.org/10.1016/j.jcp.2018.11.034

Juan M. Gimenez[a,b], Horacio J. Aguerre[a,c], Sergio R. Idelsohn[d,e], Norberto M. Nigro[a,b]

[a] *Centro de Investigación de Métodos Computacionales (CIMEC), UNL-CONICET, Santa Fe, Argentina*
[b] *Facultad de Ingeniería y Ciencias Hídricas - Universidad Nacional del Litoral, Ciudad Universitaria, Santa Fe, Argentina*
[c] *GIMCE, Facultad Regional Concepción del Uruguay, Universidad Tecnológica Nacional, CdU, Argentina*
[d] *International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain*
[e] *Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain*

Abstract

This work presents a novel proposal of a second-order accurate (time and space) method for solving transport equations including incompressible flows problems within a mixed Lagrangian-Eulerian formulation. This methodology a symmetrical operator splitting, the use of operators to transfer data between particles and the background mesh, and an improved version of the eXplicit Integration Following the Streamlines (X-IVS) method. In the case of incompressible flows, .

New interpolation and projection operators are evaluated and quadratically accurate solutions of scalar transport tests are presented. Then, incompressible flow problems are solved where the rate of convergence of the method is assessed using both structured and unstructured background grids. The method is implemented in the open source platform OpenFOAM® allowing arbitrary meshes and obtaining reliable computing comparisons with standardized solvers. obtained reveal that the current method is able to obtain a lower level of error than a fast Eulerian alternative, without increasing the total computing time.

Keywords: second-order, particle methods, incompressible flow, operator splitting, OpenFOAM®, PFVM

1. Introduction

Over decades, computer simulation of incompressible fluid flows has been mainly based on the Eulerian formulation of the fluid mechanics equations [1]. The original Eulerian methods lacked the ability to solve some of these new models. For example, in the case of simulation of heterogeneous flows, Eulerian methods had to be enhanced and adapted to capture and manage new features as the free-surface behavior without convincing results at the beginning.

Methods based on the Lagrangian formulation were born as an alternative to traditional Eulerian methods [2–5]. Researchers found that the Lagrangian approach allowed . However, these methods presented drawbacks due to, the inaccurate treatment of boundary conditions, and of the solution fields when elliptic problems, as diffusion, were solved on the irregular set of particles. These facts, along with the expensive processing required to update the particle vicinity discouraged the massive use of pure Lagrangian strategies [6].

In this context, mixed Lagrangian-Eulerian (LE) methods, where a particle-based method is combined with a fixed or reconstructed grid to support part of the time-step computation, naturally appeared [7–9]. Here, using the Lagrangian approach, the convective term is handled by the solution of the trajectory equation for the particles, avoiding the instabilities or the numerical diffusion introduced when grid-based solutions are used. , because of its implicit nature, the parabolic and elliptic problems are solved using Eulerian approaches. This avoids the strong numerical restrictions (small time-steps, weak compressibility) required to not destabilize particle-based solutions.

A popular branch of LE methods is the Particle Finite Element Method (PFEM) [10]. However, it was not until the development of X-IVS (eXplicit Integration following the Velocity and Streamlines) [11] that both computing times and accuracy were comparable with pure Eulerian methods when solving convective-dominant problems. The strategy of integrating the convection of fluid particles, based on following the streamlines of the flow in the current time-step instead of the particle trajectories, is an alternative approach to solve the non-linearities of the flow equations. Adding this strategy to the original PFEM method, a new methodology called Particle Finite Element Method Second Generation (PFEM-2) [12] appeared. X-IVS gave the possibility of solving complex flows with large time-steps, the presence of the mesh allowed accurate solutions of the fractional step method.

Despite impressive results in particular applications, some limitations of LE methods currently its massive use replacing classical Eulerian methodologies. For example, LE methods are restricted to certain kind of spatial discretization of the domain of interest. Some of them were developed to work only with Cartesian or structured grids [9, 13, 14] or simplices [10, 12], their use for industrial problems particularly when boundary layers are required. In this context, a LE method with the capability of employing arbitrary (general polyhedral) meshes is a desirable feature not covered yet.

An important drawback of LE methodologies, is the unavailability of general and open implementations of these strategies. Usually, research groups develop and maintain their own codes with their own specifications, data formats, policies, which for an external user to test or employ the implementation of the numerical method.

Perhaps the most important limitation of LE methodologies is their low global rate of convergence (ROC). Although some recent works in LE field were devoted the spatial order of the algorithms for data transference between the particles and the mesh [15, 16], the poor accuracy (first-order) of temporal approximations limits the quality of the global results even using high order spatial operators. , these issues are due to the low order integration of the temporal derivative in transport equations. The origin of the drawback varies according to the LE method analyzed, for example, some of them update the particle location and state with just a first-order approximation [8, 16], while others employ a first-order splitting for the treatment of convection in particles and diffusion on the mesh. If is improved, there are several second-order time pressure-velocity , such as the fractional step based on the block LU decomposition [17] and the Pressure-Implicit with Splitting of Operators (PISO) [18], able to be employed to obtain a global second-order method. Recently, Maljaars [19] proposes a global second-order method on triangular meshes, but the maximum CFL to employ, the large number of

particles required for the least squares-based projection increases largely the computational cost.

The aim of this work is overcoming the mentioned limitations of current LE methods. In this context, a second-order, in both time and space, LE method for solving flow problems over general polyhedral meshes, is presented. The methodology consists of solving the transport equation performing a symmetric operator splitting based on the Strang's proposal [20] and includes operators to transfer data between mesh and particles. In the case of incompressible flows, this work solves the governing equations using a combination of the SIMPLE [21] and PISO method called the PIMPLE algorithm [22]. In this algorithm, a set of non-linear (PIMPLE) iterations are performed per time-step (outer correction loop). For each iteration, the split momentum equation is used as velocity predictor followed by multiple velocity correctors (PISO iterations). In this sense, the pressure is updated multiple times per one momentum equation-update.

Several features of the current proposal are taken from PFEM-2 as the capability of using large time-steps due to the particle integration approach. However, an updated version of the X-IVS integration method is developed and analyzed. Also, in order to favor the broadcast of the method, the implementation is carried out in the popular OpenFOAM®suite which is a free and open-source code based on a cell-centered version of the Finite Volume Method (FVM). A major advantage of OpenFOAM®is its structured implementation which simplifies of reliable and efficient computational continuum-mechanics algorithms [23].

The work is structured as follows: Section 2 presents the general transport problem equation and the methodology employed to solve it, and concludes on the presentation of the algorithm and its extension to incompressible flow problems. In Section 3, tests to evaluate the interpolation and projection operators on structured and unstructured grids are performed. Also, the solver is tested solving a scalar transport problem with the analytical solution to assess the global ROC of the methodology. Then, in Section 4, including a preliminary analysis of the algorithm efficiency. Finally, the computationally-demanding three-dimensional Taylor-Green vortex problem is evaluated. This work ends discussing the main conclusions in Section 5.

2. Mathematical and numerical modeling

Equations governing various physical phenomena can be gathered into a generic form. Let Ω be a bounded domain, the so-called transport equation is written generically as,

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (\mathbf{\Gamma} \nabla \phi) - \nabla \cdot (\mathbf{u}\phi) + S_\phi \quad \text{in } \Omega, \tag{1}$$

where ϕ is an arbitrary intensive scalar or vectorial field, $\mathbf{\Gamma}$ is the diffusivity tensor, $\mathbf{u}$ is the velocity field that the quantity is moving with, and S_ϕ is a source term, which can depend on the solution ϕ or not. To solve the numerical problem, conditions over the boundary Γ of Ω, and initial conditions $\phi^0 = \phi(\mathbf{x}, t = 0)$ are imposed.

The momentum balance can be obtained from Equation 1 with $\phi = \rho\mathbf{u}$, with ρ the fluid density, and selecting $S = -\nabla p$ being p the pressure and $\mathbf{\Gamma} = \mu$ the dynamic viscosity of a Newtonian fluid. Because the transport velocity is unknown, . Moreover, in order to model the behavior of a fluid flow, the momentum equation is coupled with the continuity (or mass conservation) equation which can be seen as another transport equation

where $\phi = \rho$, $\mathbf{\Gamma} = 0$ and $S_\phi = 0$. In the case of incompressible flows, the density ρ is constant and then the continuity equation is reduced to the incompressibility constraint conforming the so-called Navier-Stokes equations for incompressible flows:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = \nabla \cdot (\nu \nabla \mathbf{u}) - \nabla p' \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad \text{in } \Omega, \tag{2}$$

where $\nu = \mu/\rho$ is the kinematic viscosity and $p' = p/\rho$ is the .

The quality of a numerical method is related to the order of convergence of its error. Finite Elements (FE) with linear shape functions, the Finite Volume (FV) considering linear variation in a cell, Finite Differences (FD) with first-order Taylor expansion are second-order accurate in space. In addition, when a second-order scheme, as Crank-Nicholson or preferentially the bounded backward differencing [24], is used , an overall second-order method for solving the transport equation (1) is obtained.

The use of Eulerian alternatives for the treatment of convective terms is limited while the Lagrangian approach several . With the solution of particle trajectory problems the convective term is naturally solved . a particle-based convection, caution must be taken the overall convergence order, mainly because an operator splitting to solve convection and diffusion and/or pressure problems separately is required to perform the time integration of the transport problem.

Next subsections deal with the presentation of the methodology proposed in this work, starting from the operator splitting for time advancing, and continuing with the employed. Subsequently, the operators for transferring states between particles and mesh are presented, continuing with the description of the particle time-integration.

2.1. Operators Splitting

The initial value problem 1 may be rewritten as,

$$\frac{\partial \phi}{\partial t} = (\mathbf{A} + \mathbf{B})\phi \quad \text{in } \Omega, \tag{3}$$

where $\mathbf{A}$ and $\mathbf{B}$ are linear operators. In the case of the discretization of Eq. (1), to the matrices approximating the operators of diffusion and convection respectively (the source term can be also included in the first operator). Given the solution at time t^n, the solution at next time-step $t^{n+1} = t^n + \Delta t$, i.e. ϕ^{n+1} can be found ,

$$\phi^{n+1} = e^{\Delta t(\mathbf{A}+\mathbf{B})}\phi^n \quad \text{in } \Omega. \tag{4}$$

Computing directly $e^{\Delta t(\mathbf{A}+\mathbf{B})}$ means solving exactly the equation but sometimes it is useful to calculate $e^{\Delta t\mathbf{A}}$ and $e^{\Delta t\mathbf{B}}$, separately. A simple option is the standard scheme known as *Godunov operator splitting*[25] which reads:

$$e^{\Delta t(\mathbf{A}+\mathbf{B})} \approx e^{\Delta t\mathbf{A}} e^{\Delta t\mathbf{B}} \quad \text{in } \Omega. \tag{5}$$

It can be shown (5) [26]. To compute the solution at the final time T, it takes $T/\Delta t$ steps. Then, because of error accumulation, the method is first-order accurate. On the other hand, among the proposals for second-order splittings, one of the most popular is the Strang (or symmetrical) splitting [20]:

$$e^{\Delta t(\mathbf{A}+\mathbf{B})} \approx e^{\frac{1}{2}\Delta t\mathbf{A}} e^{\Delta t\mathbf{B}} e^{\frac{1}{2}\Delta t\mathbf{A}} \quad \text{in } \Omega. \tag{6}$$

Replacing each term of last equation with a second-order Taylor expansion, the error of the approximation decays like Δt^3 [26]:

$$\mathbf{I}+\Delta t(\mathbf{A}+\mathbf{B})+\frac{1}{2}\Delta t^2(\mathbf{A}+\mathbf{B})^2 = (\mathbf{I}+\frac{1}{2}\Delta t\mathbf{A}+\frac{1}{8}\Delta t^2\mathbf{A}^2)(\mathbf{I}+\Delta t\mathbf{B}+\frac{1}{2}\Delta t^2\mathbf{B}^2)(\mathbf{I}+\frac{1}{2}\Delta t\mathbf{A}+\frac{1}{8}\Delta t^2\mathbf{A}^2)+\mathcal{O}(\Delta t^3). \tag{7}$$

Integrating in a second-order temporal scheme. Equation (7) reveals the need for second-order approximation to each exponential of the right-hand side of Equation (6), which means the need of using second-order time operators to obtain a global second-order time splitting. Therefore, the solution of the general transport equation (1) at time t^n, i.e. ϕ^n, the solution at next time-step ϕ^{n+1} can be found using the splitting (6) which in the practice requires performing three stages:

$$\frac{\partial\hat{\phi}}{\partial t} = \nabla\cdot\left(\mathbf{\Gamma}\nabla\hat{\phi}\right)+S_\phi \qquad \text{with } t^n < t <= t^{n+1/2} \text{and } \hat{\phi}^n = \phi^n \tag{8}$$

$$\frac{\partial\hat{\hat{\phi}}}{\partial t} = -\nabla\cdot\left(\mathbf{u}\hat{\hat{\phi}}\right) \qquad \text{with } t^n < t <= t^{n+1} \text{and } \hat{\hat{\phi}}^n = \hat{\phi}^{n+1/2} \tag{9}$$

$$\frac{\partial\hat{\hat{\phi}}}{\partial t} = \nabla\cdot\left(\mathbf{\Gamma}\nabla\hat{\hat{\phi}}\right)+S_\phi \qquad \text{with } t^{n+1/2} < t <= t^{n+1} \text{and } \hat{\hat{\phi}}^{n+1/2} = \hat{\hat{\phi}}^{n+1} \tag{10}$$

and finally $\phi^{n+1} = \hat{\hat{\phi}}^{n+1}$.

The methodology proposed in this work is based on the Strang's splitting. The problems which involve diffusion (and a source term), i.e. stages (8) and (10), are solved in a Eulerian fashion using second-order discretization schemes for time and space. On the other hand, the convective problem (9) is solved with a particle-based calculation supported with a mesh field for the transport velocity. The initial conditions of each sub-problem require the final state of the previous one, therefore data transfer between the two discretizations is required. This procedure must be accurate enough to preserve the global ROC.

2.2. Spatial discretization

The LE methodology presented in this work two spatial discretizations: a background mesh Eulerian calculations and the particle cloud to calculate convective terms.

In this context, a background mesh that partitions Ω into a number of subregions Ω_c $\Omega = \bigcup_c \Omega_c$ with $\emptyset = \bigcap \Omega_c$ is employed. Particularly, the cell-collocated finite volume discretization is used here. In FV, each subregion is named cell, the unknowns are placed at cell centers, and the cell balancing is done according to the incoming/outcoming flux over cell boundaries f (faces).

On the other hand, . A point p, known as *particle*, . The particles used in this work are PFEM-like: the particles are massless and the value carried ϕ_p an average of the data around , i.e. a representation of the continuous surrounding media. Also, according to its position $\mathbf{x}_p$, which is function of time t, each particle is contained by (or belongs to) cell c if $\mathbf{x}_p \in \Omega_c$.

The solution procedure involves data transfer between the mesh and the particles. In this context, *interpolation* is the action of approximating the value of any mesh field ϕ . While *projection* is the inverse operation, i.e. the field values from the particles to the mesh.

The errors from interpolation E_i and projection E_p are independent of the time spacing [27]. The total accumulated errors E_{it} and E_{pt}, for interpolation and projection respectively, at time T after N time-steps will be

$$E_{it} = E_i N = E_i \frac{T}{\Delta t} \quad \text{and} \quad E_{pt} = E_p N = E_p \frac{T}{\Delta t} \tag{11}$$

which reveals the requirement of a third-order interpolation/projection operators to obtain a quadratic global ROC.

Interpolation from cells to particles. The mesh-field values on arbitrary points which are not coincident with the placement of unknowns must be determined. When these points are $\mathbf{x}_p$, this task is named interpolation and symbolized with $\mathbf{\Lambda}$. The FV cell-collocated discretization places the field values on cell c centroid $\mathbf{x}_c$, defining $\phi_c = \phi(\mathbf{x}_c)$. Considering the Taylor series expansion of any field ϕ around the point $\mathbf{x}_c$, and truncating terms, the mesh field at particle position, $\phi(\mathbf{x}_p)$, :

$$\mathbf{\Lambda}(\phi) = \phi(\mathbf{x}_p) = \phi_c + (\mathbf{x}_p - \mathbf{x}_c) \cdot (\nabla\phi)_c + \frac{1}{2}(\mathbf{x}_p - \mathbf{x}_c) \cdot (\nabla\nabla\phi)_c \cdot (\mathbf{x}_p - \mathbf{x}_c) + \mathcal{O}[(\mathbf{x}_p - \mathbf{x}_c)^3] \tag{12}$$

where $(\nabla\phi)_c$ is the cell-centered gradient of the field, which may be explicitly approximated using the Gauss formula as:

$$(\nabla\phi)_c \approx \frac{1}{V_c} \sum_f \phi_f \mathbf{S}_f, \tag{13}$$

the cell volume, $\mathbf{S}_f$, and ϕ_f is the interpolated value of the unknown at the face center f. Similarly, $(\nabla\nabla\phi)_c$ is the Hessian matrix of ϕ at cell c where each column is calculated from each gradient component, i.e.

$$(\nabla\nabla\phi)_c \approx \frac{1}{V_c} \sum_f \left[\mathbf{S}_f \otimes (\nabla\phi)_f\right]. \tag{14}$$

The cell-to-face interpolation considers a linear variation of the field between the centroids of the cells which share the face. In non-structured meshes, this interpolation could lead to first-order errors since the face centroid may not be placed on the line joining the cell centers. This issue is tackled using an iterative corrector procedure named as *skewness corrections* [24].

As shown in (12), the error of the proposed interpolation procedure is $E_i = C_i(\mathbf{x}_p - \mathbf{x}_c)^3 = \tilde{C}\Delta x^3$, where $\tilde{C}_i$ depends on a third derivative of ϕ. In this way, considering CFL=U$\Delta x/\Delta t$ constant, U a reference velocity, the accumulated error due to interpolation after N time-steps is at most $E_{it} = \tilde{C}_i \dfrac{\mathrm{CFL}}{\mathrm{U}} T\Delta x^2 = \mathcal{O}(\Delta x^2)$.

Projection from particles to cells. Since the beginning, LE methods have dealt with several strategies to project particle values into a background mesh. The first approach comes from the SPH method and employs a smoother or averaging function known as *kernel*. Here, the physical quantity at any position can be obtained by summing the relevant properties of all the particles which lie within the range of the kernel. The particles contributions are weighted according to their distance to the point of interest. However, as shown by Sigalotti et. al. [28], these SPH-like projection schemes including corrective strategies, which were thought to converge to second or even higher order, are actually first-order accurate, or at best close to second-order. A higher order projection strategy, employed by other particle-based methods is on radial basis functions (RBF). However, this approach requires inverting a matrix with a size proportional to the number of particles. Moreover, on an user-defined shape parameter [29].

In the context of particles and FE meshes, strategies based on the FE shape functions are discussed in [15]. In particular, a bounded and minimally diffusive second-order accurate projection is described. This operation is based on solving a linear system with the size of mesh nodes whose solution is limited to avoid spurious oscillations, which particularly useful for sharp functions. Even its promissory results, this strategy cannot be directly employed in the current work because of the requirement of the nodal-based approximation of FE. Other [19] propose employing minimization procedures as least squares at the cost of inverting an equation system per cell.

In this work, a third-order projection from particles to a general FV mesh is presented where averaging functions (W) as the W4 Wendland or the quintic RBF [30] kernels are used. To improve the convergence rate of the standard SPH projection, mesh field derivatives are included exploiting the geometrical data provided by the FV discretization. Similarly to the interpolation procedure, a second-order Taylor expansion around the cell centroid is used. Then, the proposal consists of solving the following weighted average:

$$\mathbf{\Pi}(\phi) = \phi_c = \frac{\displaystyle\sum_{p\in P_c} \left[\phi_p + (\nabla\phi)_c \cdot (\mathbf{x}_c - \mathbf{x}_p) - \frac{1}{2}(\mathbf{x}_c - \mathbf{x}_p)\cdot(\nabla\nabla\phi)_c \cdot (\mathbf{x}_c - \mathbf{x}_p)\right] W(|\mathbf{x}_c - \mathbf{x}_p|)}{\displaystyle\sum_{p=1}^{P} W(|\mathbf{x}_c - \mathbf{x}_p|)}, \tag{15}$$

where the set of surrounding particles P_c is composed by those that belong to the cell c or to a cell sharing at least one vertex with the cell c (extended stencil). the derivatives depend on the unknown ϕ_c, the proposal of Equation (15) leads to a global equation system which could be solved implicitly. The alternative approach

used in this work consists of an iterative procedure which reads:

$$\phi_c^{k+1} = \mathbf{\Pi}(\phi) = \frac{\sum_{p \in P_c} \left[\phi_p + (\nabla\phi)_c^k \cdot (\mathbf{x}_c - \mathbf{x}_p) - \frac{1}{2} (\mathbf{x}_c - \mathbf{x}_p) \cdot (\nabla\nabla\phi)_c^k \cdot (\mathbf{x}_c - \mathbf{x}_p) \right] W(|\mathbf{x}_c - \mathbf{x}_p|)}{\sum_{p=1}^{P} W(|\mathbf{x}_c - \mathbf{x}_p|)}, \tag{16}$$

where k is the iteration number, the first iteration considers $\nabla\phi_c^0 = \mathbf{0}$ and along the remaining iterations the gradient and the Hessian matrix are calculated with the Gauss formula (13).

Remark. The lack of particles on the neighborhood of a cell center, i.e. $P_c = \{\}$, overcame via the so-called preventive and palliative strategies. In the former, particles are seeded or removed after each time-step in order to keep the particle distribution balanced. In the latter, particles are seeded in those cells where $P_c = \{\}$ is searched with a backward integration. More details about the management of particle inventory can be found in [15].

2.3. Particle time integration

Given the solution on the particle p at time t^n, i.e. $\phi_p(t^n) = \phi_p^n$, the solution at time t^{n+1} consists on solving the system:

$$\begin{cases} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{u}^\tau(\mathbf{x}_p^\tau)\, d\tau \\ \phi_p^{n+1} = \phi_p^n \end{cases}, \tag{17}$$

which is equivalent to solve the Equation 9 for each particle in the Lagrangian formulation, i.e.:

$$\frac{D\mathbf{x}}{Dt} = \mathbf{u}, \qquad \frac{D\phi}{Dt} = 0. \tag{18}$$

Due to computational performance, explicit strategies are preferable to integrate the trajectory equation given in (17). While most of Lagrangian methodologies solve this integral using basic operators which usually are limited by the CFL < 1 restriction [10, 31], some LE methodologies take advantage of mesh velocity field to improve the approximation of the trajectories overcoming the CFL limit and still obtaining stable and accurate solutions [13, 14]. In this context, the eXplicit Integration following Velocity Streamlines method (X-IVS) [11] . In its original proposal, the velocity field obtained on the mesh at time t^n is used for the time integration, i.e.:

$$\mathbf{u}^\tau(\mathbf{x}_p^\tau) \approx \mathbf{u}^n(\mathbf{x}_p^\tau) \tag{19}$$

using the same spatial order approximation at any particle position $\mathbf{x}_p$.

Some improvements were proposed in recent works in the context of two-phase flows [32, 33], but none of deal with the intrinsic low order of the temporal discretization, in particular when transport velocity has high

temporal variation. In this context, this work introduces changes on the X-IVS formulation to improve the temporal rate of convergence.

In the case of scalar transport problems the velocity field is known. Then, at time $t^n \leq t^\tau \leq t^{n+1}$, this field is approximated as:

$$\mathbf{u}^\tau(\mathbf{x}_p^\tau) \approx (1-\alpha^\tau)\mathbf{u}^n(\mathbf{x}_p^\tau) + \alpha^\tau \mathbf{u}^{n+1}(\mathbf{x}_p^\tau) \qquad \text{with} \qquad \alpha^\tau = (t^\tau - t^n)/(t^{n+1} - t^n). \tag{20}$$

On the contrary, in flow problems, the velocity at time t^{n+1} is unknown. Originally, X-IVS used the streamlines (velocity at time t^n) to update the particle position being first-order accurate. In this work, making use of the partial solutions provided by the iterative procedure (PIMPLE), couple the pressure and velocity and solve the incompressible flow problem, a second-order temporal approximation for trajectory is proposed. Similarly to Equation (20), the particles follow the pathlines determined by:

$$\mathbf{u}^\tau(\mathbf{x}_p^\tau) = \mathbf{u}^{\tau,i}(\mathbf{x}_p^{\tau,i}) \approx (1-\alpha^\tau)\,\mathbf{u}^n(\mathbf{x}_p^{\tau,i}) + \alpha^\tau \mathbf{u}^{n+1,i-1}(\mathbf{x}_p^{\tau,i}) \qquad \text{with} \qquad \alpha^\tau = (t^\tau - t^n)/(t^{n+1} - t^n) \tag{21}$$

where i is the PIMPLE iteration number, $\mathbf{u}^{n+1,i-1}$ is the velocity solution at time t^{n+1} of the previous PIMPLE iteration, while $\mathbf{u}^{n+1,0} = \mathbf{u}^n$ is the solution of the previous time-step. Note that (21) implies recalculating the particle trajectory at every PIMPLE iteration.

Once determined the assumptions for the transport velocity field at intermediate times $t^n \leq t^\tau \leq t^{n+1}$, the value problem (17) can be solved using either analytic [11, 34] or numerical strategies [15]. For a compromise between computational cost and accuracy, the numerical strategy adopted in this work consists of a sub-stepping procedure, which reads:

$$\begin{cases} \mathbf{x}_p^{n+\frac{k}{\mathrm{K}}} = \mathbf{x}_p^{n+\frac{k-1}{\mathrm{K}}} + \dfrac{(\mathbf{k}_1 + \mathbf{k}_2)}{2}\delta t \\ \hat{\phi}_p^{n+\frac{k}{\mathrm{K}}} = \phi_p^{n+\frac{k-1}{\mathrm{K}}} \end{cases}, \tag{22}$$

where the sub-step number k is in the range $[1, \mathrm{K}]$ with $\mathrm{K}\delta t = t^{n+1} - t^n = \Delta t$, being K defined to limit the elemental CFL of each particle $\mathrm{CFL}_p = |\mathbf{u}(\mathbf{x}_p)|\delta t/\Delta x$. The vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ are the intermediate velocity approximations from the second-order Runge-Kutta integrator (RK2) which reads:

$$\mathbf{k}_1 = \mathbf{u}^{n+\frac{k-1}{\mathrm{K}}}(\mathbf{x}_p^{n+\frac{k-1}{\mathrm{K}}}) \quad \text{and} \quad \mathbf{k}_2 = \mathbf{u}^{n+\frac{k}{\mathrm{K}}}(\mathbf{x}_p^{n+\frac{k-1}{\mathrm{K}}} + \mathbf{k}_1\delta t). \tag{23}$$

The estimation of the velocity field at is done following the interpolation strategy for a general polyhedron presented in Equation (12). Also, taken into account that the container cell of any particle p must be updated at the end of each sub-step according to its intermediate position $\mathbf{x}_p^{n+\frac{k}{\mathrm{K}}}$. The particle-face intersections and the modification of the container cell performed using a barycentric-based tracking, similar to the one presented in [35], which works robustly irrespective of mesh quality.

2.4. Algorithms

The proposed LE algorithm which solves the on mesh and convection on particles by means of the Strang's splitting and considers the interchange of data between particles and mesh, is shown in Algorithm (1). Here, it is assumed that all fluid variables are known at time t^n for both, the particles and the mesh cell centers. Subindexes $()_c$ y $()_p$ represent a generic mesh cell c and a generic particle p respectively.

Algorithm 1 - Time-Step for general scalar transport equation.

1. On mesh, calculate half diffusion step (8) to obtain $\hat{\phi}_c^{n+\frac{1}{2}}$ on each cell center.
2. Update the particles state $\hat{\phi}_p^{n+\frac{1}{2}} = \phi_p^n + \mathbf{\Lambda}(\hat{\phi}_c^{n+\frac{1}{2}} - \phi_c^n)$ with (12).
3. For each particle, calculate the full convection step solving (22) to obtain $\mathbf{x}_p^{n+1}$ and $\hat{\hat{\phi}}_p^{n+1}$
4. Update the cell centers state $\hat{\hat{\phi}}_c^{n+1} = \mathbf{\Pi}(\hat{\hat{\phi}}_p^{n+1})$ with (16).
5. On mesh, calculate half diffusion step (10) to obtain ϕ_c^{n+1}.
6. Update the particles state $\phi_p^{n+1} = \hat{\hat{\phi}}_p{}^{n+1} + \mathbf{\Lambda}(\phi_c^{n+1} - \hat{\hat{\phi}}_c^{n+1})$ with (12)

In the case of incompressible flows, this work uses . This procedure is able to be used with large CFL numbers since the algorithm searches the convergence of the non-linear convective term through an iterative process called *outer correction loop*, analogous to the SIMPLE method. Within each *outer correction loop* (PIMPLE) iteration, the Strang's splitting is used to solve the momentum equation of Equation (2) defining a new velocity field which is not divergence free. After that, PISO iterations are computed solving a pressure equation on the mesh to impose the incompressibility constraint. Note that, as mentioned previously, during the PIMPLE i-th iteration the X-IVS integration is also recalculated employing a new velocity field for transporting the particles: the temporal interpolation between the previous time-step field $\mathbf{u}_c^n$ and the latest velocity field of the previous iteration $\mathbf{u}_c^{n+1,i-1}$. Finally, a time-step for the solver of incompressible flows is presented in Algorithm 2

Henceforth, the presented methodology is referred as the Particle Finite Volume Method (PFVM). The Algorithms 1 and 2 are implemented in the library OpenFOAM®inheriting most of the beneficial features: the large set of tools along with the models, boundary conditions and linear equation system solvers.

In order to improve the implementation of the Strang splitting, Strang [20] and Leveque [25] mention that the combination of the half-step operators at the end of one step and at the beginning of the next one can be replaced by the single-step operator. However, in the context of the Algorithm 2, the Strang splitting for momentum equation is followed by the PISO loop, a sequence which is iterated to manage the nonlinearity of the convective term. This latter turns not straighforward to combine the second diffusion operator of the current timestep and the first of the next one into a unique computation.

Algorithm 2 - Time-Step for incompressible flows.

1. do PIMPLE iteration (outer correction loop):
   - 1.1. On mesh, calculate half momentum step (without convection) (8) to obtain $\hat{\mathbf{u}}_c^{n+\frac{1}{2}}$ on each cell center
   - 1.2. Update the particles state $\hat{\mathbf{u}}_p^{n+\frac{1}{2}} = \mathbf{u}_p^n + \mathbf{\Lambda}(\hat{\mathbf{u}}_c^{n+\frac{1}{2}} - \mathbf{u}_c^n)$ with (12).
   - 1.3. For each particle, calculate the full convection step solving (22) (X-IVS iterated) to obtain $\mathbf{x}_p^{n+1}$ and $\hat{\mathbf{u}}_p^{n+1}$
   - 1.4. Update the cell centers state $\hat{\mathbf{u}}_c^{n+1} = \mathbf{\Pi}(\hat{\mathbf{u}}_p^{n+1})$ with (16)
   - 1.5. On mesh, calculate half momentum step (without convection) (10) to obtain the velocity prediction $\tilde{\mathbf{u}}_c^{n+1}$
   - 1.6. do PISO iteration:
     - 1.6.1. On mesh, calculate the Poisson pressure equation $\nabla^2 p = \frac{1}{\Delta t}\nabla \cdot \tilde{\mathbf{u}}_c^{n+1}$ to obtain p^{n+1}
     - 1.6.2. On mesh, correct the velocity prediction to obtain a divergence-free $\tilde{\mathbf{u}}_c^{n+1}$ field.
   - 1.7. Go to 1.6 until complete PISO corrections required. If done, set $\mathbf{u}_c^{n+1} = \tilde{\mathbf{u}}_c^{n+1}$.
2. Go to 1 until convergence of pressure residuals and particle positions.
3. Update the particles state $\mathbf{u}_p^{n+1} = \hat{\mathbf{u}_\mathbf{p}}^{n+1} + \mathbf{\Lambda}(\mathbf{u}_c^{n+1} - \hat{\mathbf{u}}_c^{n+1})$ with (12)

3. Scalar Tests

3.1. Accuracy of interpolations and projections

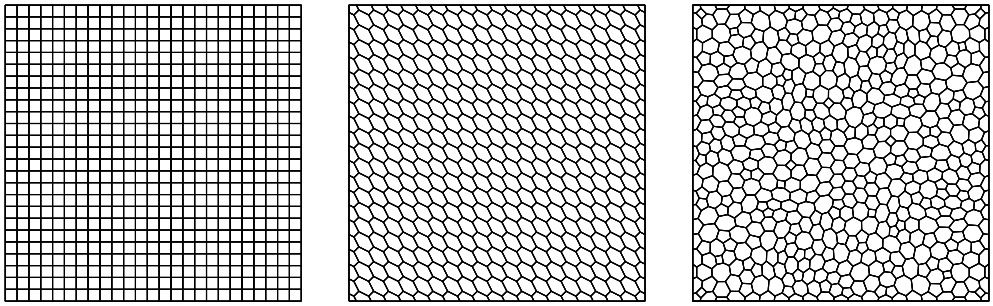
In order to evaluate the mesh convergence of the interpolation (12) and projection (16) strategies presented above, a 2D test is designed. The domain is a square of length $[-1,-1] \times [1,1]$ and discretized with: I) a structured Cartesian grid (Figure 1a), II) a structured polyhedral mesh (Figure 1b) and III) an unstructured polyhedral mesh (Figure 1c). The polyhedral meshes are constructed as the dual Voronoi grid of a structured mesh of triangles and a Delaunay triangulation respectively. The coarsest refinement splits each direction into 20 elements which leads to a mesh size of $H = 0.1$, where three particles per cell are seeded at random positions. The successive grid refinements are achieved in I) splitting each quad into four quads, in II) splitting each triangle of the dual mesh into four triangles and using the Voronoi correspondence, and in III) generating a Delaunay triangulation with $h = H/2^l$ the reference size and using the corresponding Voronoi grid, where $0 <= l <= 5$ is the level of refinement.

The analytic function to be interpolated/projected is the bi-dimensional sinusoidal profile:

$$\phi^{ex}(x, y) = \sin(\pi x)\sin(\pi y), \tag{24}$$

where neither its values nor its derivatives vanish at domain boundaries.

In the interpolation test, the analytic function is evaluated on cell centers and interpolated to the particles. On the other hand, in the projection test, the function is imposed on the particles and projected to the cell centers. For projection, the W4 Wendland kernel is used using five iterations to compute the derivatives in (16).



(a) I: Quads structured. (b) II: Polyhedral structured. (c) III: Polyhedral unstructured.

Figure 1. Interpolation and projection tests. Coarsest meshes employed are shown.

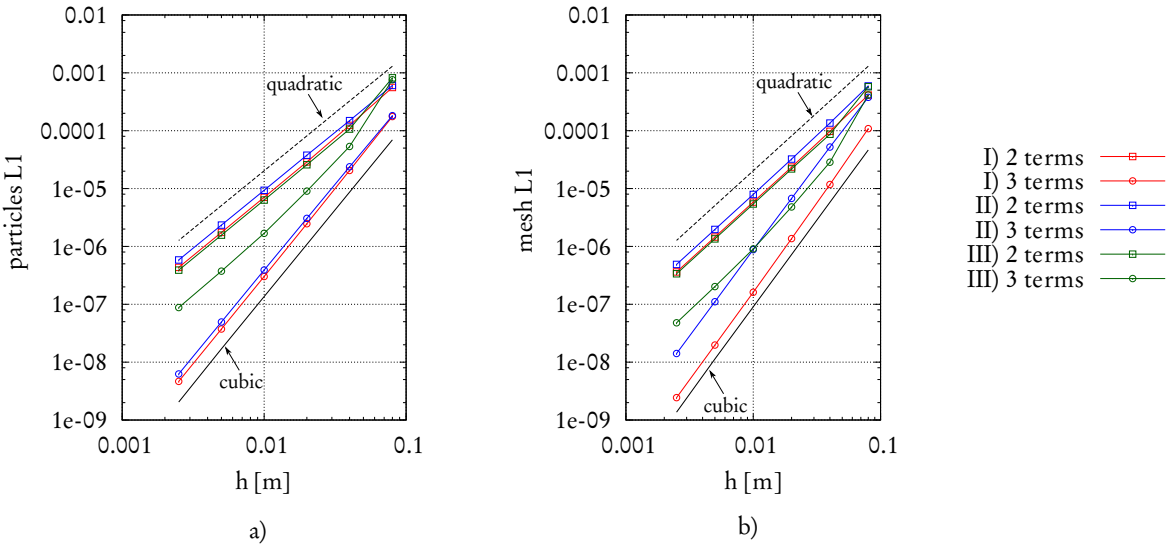


Figure 2. Interpolation (a) and projection (b) tests. Error convergence with three kind of grids: I) Cartesian structured, II) polyhedral structured and III) polyhedral unstructured. The legend indicates the number of terms used for the Taylor expansion.

In order to measure the numerical error, the discrete weighted L_P-norm is used, which reads:

$$L_P = \left(\frac{1}{V} \sum_{c=1}^{C} V_c |\phi_c^{ex} - \phi_c|^P \right)^{1/P} \tag{25}$$

where C is the number of cells, V_c is the volume of the cell c, V is the total volume of the mesh, ϕ^{ex} is the analytic solution and ϕ is the numerical solution. Particularly, when $P = 2$ is known as the weighted root mean square error (RMSE).

In Figure 2, the grid convergence of the interpolation and projection operators are presented. For each case, two Taylor expansions are used to calculate (12) and (16): *2-terms* refers to use only up to the first derivative of the function (the gradient), while *3-terms* refers the term with the second-derivative (Hessian matrix). The proposed algorithms, for both interpolation and projection operations, obtain up to third-order convergence on structured meshes. In the case of polyhedral meshes, skewness schemes must be used for not losing convergence

order due to poor estimation of the cell-centered gradients.

3.2. Rotating Gaussian with diffusion

The target of this test is to evaluate the convergence order of the proposed LE methodology (Algorithm 1) solving a scalar transport equation with diffusion. The problem has analytic solution where the unknown field $\phi(\mathbf{x}, t)$ is a rotating Gaussian described by:

$$\phi(x, y, t) = \frac{b}{b + 4\Gamma t} \exp\left(-\frac{|\mathbf{x} - \mathbf{r}(t)|^2}{b + 4\Gamma t}\right), \tag{26}$$

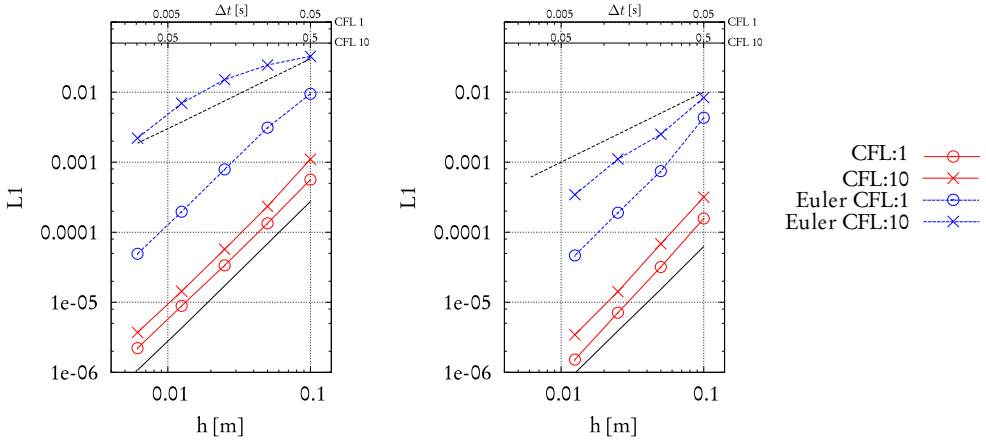
with $b = 0.025$ and $\mathbf{r} = [x_r(t), y_r(t)]$ the radius vector to the Gaussian center with $x_r(t) = \frac{1}{2}\cos(t)$, $y_r(t) = \frac{1}{2}\sin(t)$. The value of the diffusivity Γ is selected according to the kind of evolution desired. The velocity field is a pure rotation with $\mathbf{u} = (-y, x)$ being x and y the geometrical coordinates. The domain is a square of sides [-1,-1] $\times$ [1,1], the initial condition is the solution field at $t = 0$, i.e. $\phi(x, y, 0)$, while time-dependent Dirichlet conditions are imposed on all boundaries.

Two mesh topologies are employed to evaluate the accuracy of the proposed methodology. On the one hand, structured grids are selected being the coarsest mesh composed of 20×20 cells, leading to a mesh-size $\Delta x = H = 0.1$. To obtain finer meshes, each quadrilateral is split into four quads giving a mesh size factor of two ($h = H/2^l$). On the other hand, unstructured polyhedral grids are used. These meshes are the dual Voronoi grid of a Delaunay triangulation with decreasing mesh sizes.

At the beginning three particles are randomly seeded per element with the analytic field, i.e. $\phi_p = \phi(\mathbf{x}_p, 0)$ at $t = 0$, and five projection iterations are chosen to transfer data from particles to mesh. Second order accurate discretization is used for temporal and spatial operators. In the case of the unstructured mesh, skew-corrected schemes are selected for gradient and interpolation operators. Moreover, to compare the behavior of the solutions with large time-steps, per mesh with U=$\sqrt{2}$, up to reach the final simulation time $T_f = 2\pi$ s, i.e. one full rotation. As a numerical reference, the problem is also solved with a OpenFOAM®implementation of a standard scalar transport solver on FV (named Euler in next comparisons).

Figure 3 presents the error obtained per numerical solution. As a first conclusion, both numerical strategies reach second-order convergence for both meshes even using large time-steps. However, large differences on the accuracy are also observed. For example, on structured meshes, comparing the solutions with the finest mesh and the largest time-step, the error of PFVM is almost two orders of magnitude lower than the Eulerian one. Regarding the computing times, the PFVM is more efficient than the Eulerian alternative: comparing solutions with the same accuracy, in particular the Eulerian with CFL=1 and finest structured mesh ($h = H/16$) and PFVM with CFL=10 and $h = H/8$, the computing time required by the former (55.2 seconds) the required by the latter (28.1 seconds).

Similar conclusions polyhedral meshes (Figure 3b). Again, using large time-steps deteriorates the Eulerian solution while particle-based solutions look insensitive . This example shows for the first time a robust particle-based method capable of employing and second-order on unstructured polyhedral meshes.



(a) Structured mesh. (b) Unstructured mesh.

Figure 3. Case Rotating Gaussian with diffusion, $\Gamma = 0.001$. L1 norm error. Black-dashed and black-filled lines are the linear and quadratic references respectively.

4. Incompressible Flow Tests

In this Section, several incompressible flow problems are solved with the numerical strategy proposed in this work. Two tests with analytical solution are selected from literature to evaluate the convergence order of the Algorithm 2: the Taylor-Green vortex problem and the advection of a viscous vortex. Structured and unstructured grids are employed, and moderate and large time-steps are also evaluated. Moreover, comparisons with a pure Eulerian numerical reference which also uses the PIMPLE algorithm are carried out for the advected vortex case. The configuration of each simulation is similar for both solvers: temporal and spatial operators, linear solvers and tolerances criteria are shared between every simulation. Finally, a complex three-dimensional case, which requires parallel processing, is solved.

4.1. Two-dimensional Taylor-Green vortex

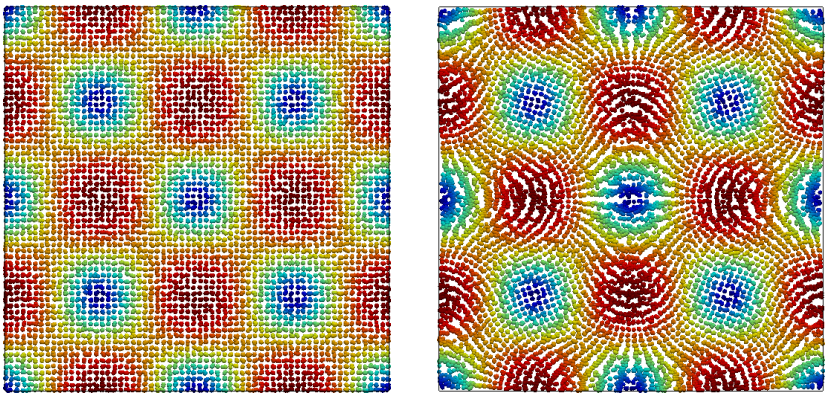
An incompressible flow problem, , The problem configuration has the following analytic solution:

$$\begin{aligned} u\left(x,y,t\right) &= -\sin(x)\cos(y)\,e^{-2\nu t}, \\ v\left(x,y,t\right) &= \cos(x)\sin(y)\,e^{-2\nu t}, \\ p(x,y,t) &= \frac{1}{4}\left[\cos(2x)+\cos(2y)\right]\,e^{-4\nu t}, \end{aligned} \tag{27}$$

where ν is the kinematic viscosity. The problem is solved using a square domain with a side length of 2π and setting cyclic conditions at boundaries. Kinematic viscosity is set in $0.001\,\mathrm{m}^2/\mathrm{s}$ defining a Re number based on vortex size of 1570. Two types of meshes are used: a structured mesh of quads and an unstructured mesh of polyhedra. Second-order spatial mesh operators are used for using also skewness and non-orthogonal corrections in the unstructured case. Temporal derivatives are discretized with the Crank-Nicholson scheme. The initial

condition for velocity and pressure is the analytic solution at time $t = 0$ s. The time-step is set up in order to obtain a maximum CFL number of approximately 3 (which takes as reference velocity U = max $|\mathbf{u}|$) and each simulation is run until reaching the total time of $T_f = 1.28\,s$. In each time-step, convergence is obtained when pressure residuals falls three orders. Two particles are seeded initially each cell, while five iterations are employed for the projection operator.

Figure 4 shows the location of the particles at initial and final times .



(a) $t = 0$ (b) $t = 1.28$

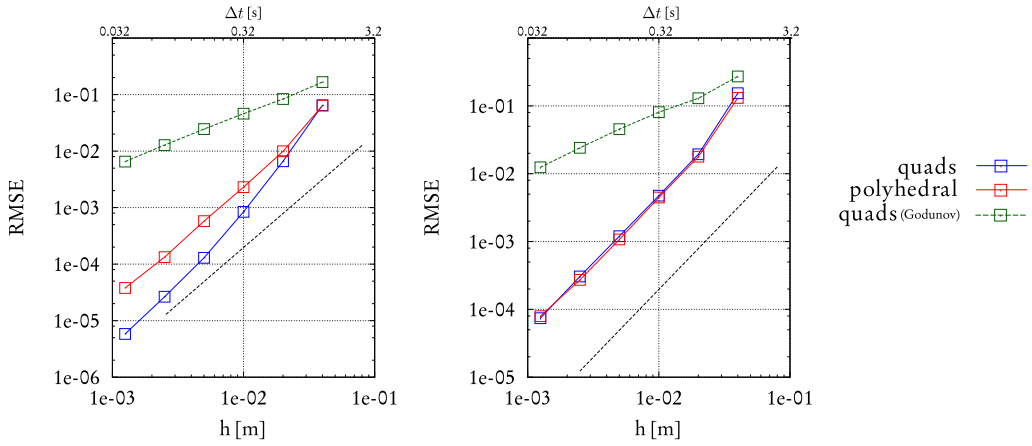
Figure 4. Case two-dimensional Taylor Green Vortex. Particles location at initial and final times for the case $h = \pi/32$. Velocity magnitude goes from 0 (blue) to 1 (red).

Starting from a reference size of $H = \pi/8$, a grid convergence study is performed varying the cell size of the domain discretizations $h = H/2^l$, with a refinement process similar to the discussed in Section 3.1 up to $l = 6$ levels. The results are presented in Fig. 5 showing the root mean square error of the velocity (the error of u_x and u_y are similar since the problem is symmetric) and the pressure as a function of the characteristic grid length. At least second-order convergence of the error is achieved with both grid types for both velocity and pressure. In the case of velocity, the ROC between the two finest grids is 2.12 for the structured case and 2.06 for the unstructured case, while for pressure is 2.05 and 1.93 respectively.

Figure 5 also presents the error convergence using a modification of the PFVM algorithm [32]). first-order convergence and the absolute error for the mesh is two orders greater. This reinforces the need for the current development in order to PFVM to be competitive with Eulerian alternatives.

Computational efficiency. The computing time required each stage of the Algorithm 2 is evaluated solving the bi-dimensional Taylor-Green case with a mesh of 64^2 cells ($h = \pi/32$). Simulations are carried out using different time-steps to analyze the dependence on CFL of each particle stage.

Table 1 reports the time consumed by each stage, where the columns are related to the following steps: momentum (1 and 5), pressure (6), convection (3), projection (4), interpolation (2 and 9). From the results, it is shown that the projection step, which does not depend on time-step, is computational demanding in



(a) velocity (b) pressure

Figure 5. . Dashed-line is second-order reference.

comparison with the remaining stages. This is due to the iterations required by the explicit operator employed. An implicit version should be soon implemented to improve the global algorithm efficiency. On the other hand, the X-IVS (convection) cost depends on the time-step the choice of CFL_p=1 as limit to update the particle position. This setting was taken after a selection process to obtain a good compromise between accuracy of particle trajectories and computing speed.

max CFL	momentum	pressure	convection	projection	interpolation	main	nOuterCorrectors	RMSE u
0.75	0.17	0.51	0.17	0.79	0.15	1.98	4	0.00049
3	0.07	0.17	0.21	0.29	0.06	0.93	5	0.00089
6	0.05	0.12	0.28	0.2	0.03	0.81	7	0.003

Table 1. CPU time [s] for different algorithm stages. Case Taylor-Green 2d flow with 64^2 cells. In each case, the average of nOuterCorrectors (PIMPLE iterations) required per time-step to reach the desired tolerance is shown.

Finally, note that the computing cost of the entire algorithm does not decrease proportionally when CFL is increased. The variable cost of convection and the larger number of nOuterCorrectors -required by PIMPLE to reach the desired tolerance when CFL increases- justify this fact.

Furthermore, the efficiency of this implementation is better than another trendy LE method: Liu [16] reports computing times for the Taylor-Green 2D vortex with $Re = 100$ analyzed in this section, but solved with the

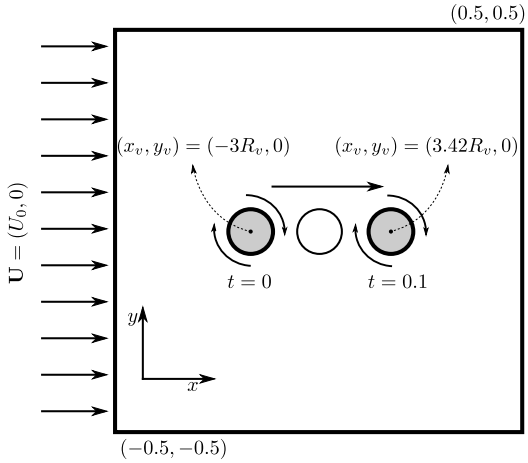


Figure 6. Convection of a vortex by a uniform flow. The starting and final positions of the vortex (x_v, y_v) are indicated.

MPPM. This method requires more than 9 s to reach T_f=5 s in a grid of 41^2 cells. The accuracy report shows an error of the maximum velocity $\max(|\mathbf{u}|)$ at $t = 5$ s of 0.02 (RMSE is not reported). The same case solved with PFVM only spends 1.5 s and obtains maximum velocity error of 0.0019.

4.2. Advected viscous vortex problem

An unsteady two-dimensional incompressible flow problem with analytic solution is presented [36]. It consists in the transport of a vortex structure by a uniform mean flow. The initial condition of the velocity $\mathbf{u} = (u, v)$ is:

$$u(x, y, 0) = U_0 - \frac{\Gamma}{R_v^2}\,(y - y_v)\exp\left(\frac{-r^2}{2R_v^2}\right), \qquad v(x, y, 0) = \frac{\Gamma}{R_v^2}\,(x - x_v)\exp\left(\frac{-r^2}{2R_v^2}\right), \tag{28}$$

where x_v and y_v are the vortex center coordinates, $r = \sqrt[2]{(x - x_v)^2 + (y - y_v)^2}$ is the distance to the vortex center, Γ is the vortex strength, R_v the radius of the vortex and U_0 is the mean flow velocity magnitude. The problem has analytical solution given by:

$$u(x, y, t) = U_0 - \frac{\Gamma}{R_v^2}\,\frac{(y - y_v)}{\alpha^2}\exp\left(\frac{-r^2}{2\,\alpha\,R_v^2}\right), \qquad v(x, y, t) = \frac{\Gamma}{R_v^2}\,\frac{(x - x_v)}{\alpha^2}\exp\left(\frac{-r^2}{2\,\alpha\,R_v^2}\right), \tag{29}$$

where α is defined by:

$$\alpha = 1 + 2\,\nu\,t, \tag{30}$$

being ν the dynamic viscosity. In the case of pressure, an analytic solution only available for the inviscid case [36].

The configuration of the problem is similar to those employed in the work of Aguerre *et al.* [37]: the vortex strength is set up to $\Gamma = 0.036$, the vortex radius is $R_v = 0.01556$ m and the mean flow velocity is $U_0 = 1$ m/s. A graphical scheme of the domain and problem description is shown in Figure 6. The vortex is positioned at the time $t = 0$ s in $(x_v, y_v) = (-3R_v, 0)$ and is convected to the position $(x_v, y_v) \approx (3.42R_v, 0)$ at time $t = 0.1$ s.

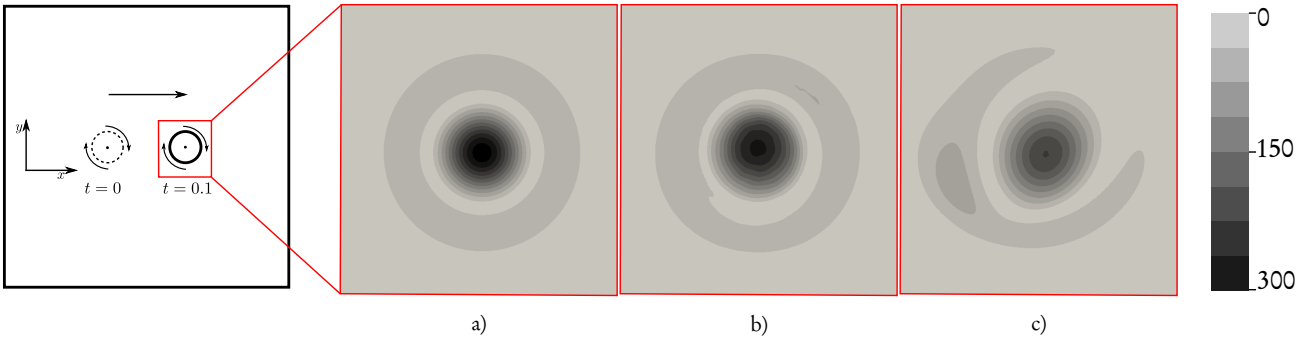
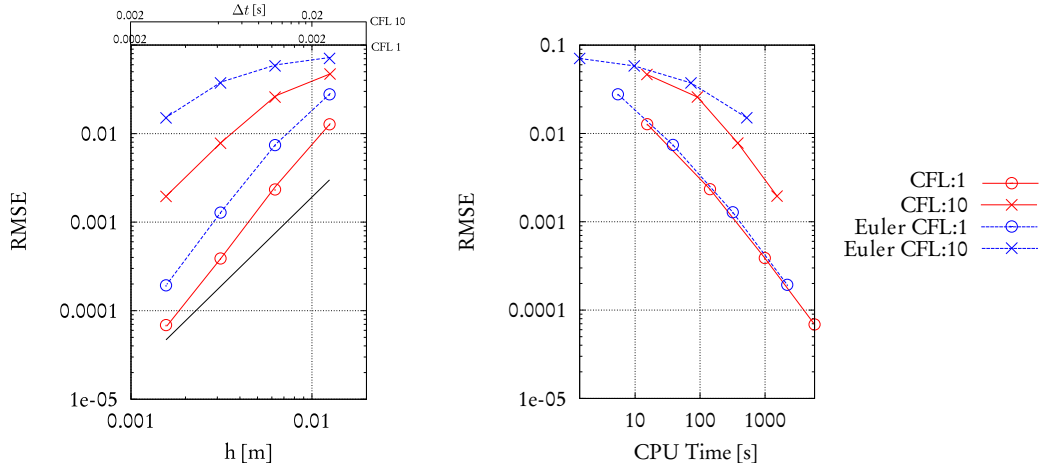


Figure 7. Advecting an inviscid vortex. Vorticity magnitude at $t = 0.1$. a) analytic solution, b) PIMPLE with Lagrangian convection (PFVM) and c) PIMPLE with Eulerian convection. Numerical solutions with CFL= 10 and a grid size of $h = H/8$.

The pressure-velocity coupling is solved with the PIMPLE algorithm iterating each time-step without relaxation up to reach five orders convergence of velocity residuals. Second-order operators are used for both spatial and temporal discretizations. The domain boundaries are located from the vortex so that the analytic solution there falls below machine precision (10^{-16}). At boundaries, the velocity is set up to the uniform mean flow U_0 and the pressure is fixed to $p = 0$ at the outlet. On the other hand, a null normal gradient is configured for p at the rest of the boundary. The problem is solved using structured meshes with different grid refinements (from $N_y = 40$ to $N_y = 640$), varying the time-step length accordingly to maximum CFL desired.



(a) RMSE u component (b) Computing Time vs. RMSE

Figure 8. Results of the advected inviscid vortex. Comparison between the current proposal with the full Eulerian pimpleFOAM solver. Black-Filled line is second-order reference.

Firstly, the inviscid case ($\nu = 0$) is analyzed. Figure 8 presents the results obtained with the current proposal simulated with CFL=1 and 10, where a comparison against the Eulerian solver pimpleFOAM (named Euler in comparisons) simulating the same CFL numbers is carried out. The grid convergence obtained with our proposal is at least second-order independently the CFL number (Figure 8a), while the Eulerian approach decreases its

ROC when large CFLs are used. Table 2 resumes the convergence order obtained for the different cases analyzed.

Convection type	ν	CFL	ROC u	ROC v	ROC p
Lagrangian	0.0	1	2.56	2.51	1.63
Lagrangian	0.0	10	2.13	2.08	2.06
Eulerian	0.0	1	2.7	2.77	2.98
Eulerian	0.0	10	1.31	1.61	1.46
Lagrangian	0.001	1	1.9	1.89	-
Lagrangian	0.001	10	1.93	1.97	-
Eulerian	0.001	1	1.7	1.75	-
Eulerian	0.001	10	1.58	1.77	-

Table 2. Error analysis in the advected vortex problem. The ROC presented is the ratio between the grids.

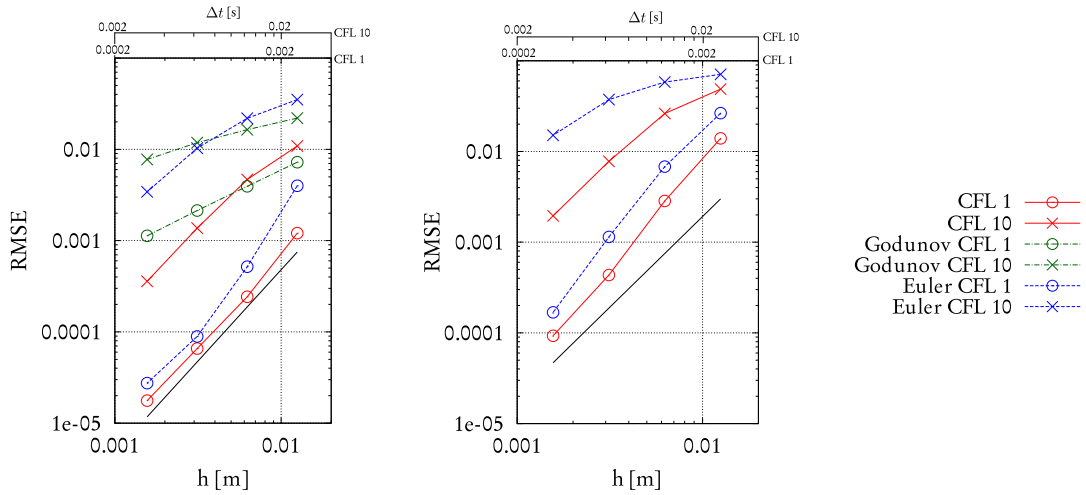
Computational efficiency. Figure 8b presents a comparison of the execution times and its respective RMSE between the current proposal and the standard pimpleFOAM solver. Each simulation was carried out in a single processor of an Intel i7-7700K with 32Gb RAM. Due to the particle calculations, the computational effort per PIMPLE iteration is 200% to 250% greater using our algorithm (moderate and large CFL respectively). The fact that the red line is below the blue one means that the total computing time required to reach certain error level is lower using the current particle-based approach, i.e. PFVM is more efficient than the standard PIMPLE this case.

The case is also solved including diffusion: the viscous case with $\nu = 0.001$ is selected. Similarly to the inviscid case, second-order accuracy is reached with PFVM. Figure 9a also presents the error convergence using the splitting of operators. , the first-order convergence obtained is a strong limitation for the accuracy of the method. In the case of the Eulerian solutions, the ROC does not reach a complete second-order (see Table 2). Additionally, the inviscid case is solved on unstructured polyhedral meshes. Figure 9b shows the grid convergence for that case, where at least second-order convergence is obtained.

4.3. Three-dimensional Taylor-Green vortex

Finally, the three-dimensional Taylor-Green vortex problem [38] is solved. This test starts with an initial vortex flow which until resting, transitioning through turbulence and energy dissipation phases. The aim of this problem is to evaluate the capability of the PFVM approach to reproduce the physics of the flow in a complex three-dimensional case. As a contrast tool, a pseudo-spectral solution is used [31].

The domain of the problem is a cube of a side length of 2π. The problem is defined by a periodic flow which



(a) RMSE u with $\nu = 0.001$. Cartesian grids

(b) RMSE u with $\nu = 0.0$. Polyhedral meshes

Figure 9. Results of the advected vortex. Comparison between the current proposal with Strang splitting, an alternative using the first-order splitting Godunov, and the full Eulerian pimpleFOAM solver. Black-Filled line is second-order reference.

starts with the following initial condition for the velocity components and the pressure:

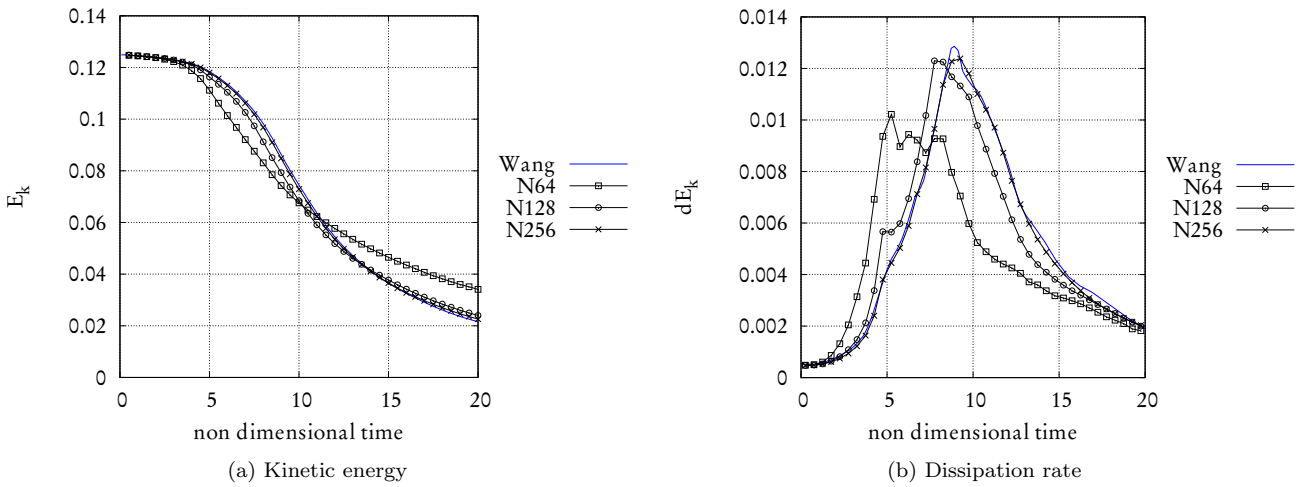
$$
\begin{aligned}
u(x,y,z,t=0) &= V_0 \sin\left(\frac{x}{L}\right)\cos\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right), \\
v(x,y,z,t=0) &= -V_0 \cos\left(\frac{x}{L}\right)\sin\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right), \\
z(x,y,z,t=0) &= 0, \\
p(x,y,z,t=0) &= p_0 + \frac{\rho_0 V_0^2}{16}\left[\cos\left(\frac{2x}{L}\right)+\cos\left(\frac{2y}{L}\right)\right]\left[\cos\left(\frac{2z}{L}\right)+2\right],
\end{aligned}
\tag{31}
$$

where the velocity V_0, the characteristic length L and the reference density ρ_0 are set up to 1 and the reference pressure p_0 is set up to 0. The viscosity is defined in order a Reynolds number of 1600 ($\mathrm{Re} = V_0 L/\nu$). The problem is studied computing the temporal evolution of the kinetic energy E_k and its derivative $-dE_k/dt$ which represents the kinetic energy dissipation rate. The kinetic energy is calculated as:

$$
E_k = \frac{1}{\rho_0 \Omega}\int_\Omega \frac{1}{2}\left(\boldsymbol{U}\cdot\boldsymbol{U}\right) d\Omega, \tag{32}
$$

being $\boldsymbol{U} = (u, v, z)$ the velocity vector.

A mesh convergence study is performed. The domain is discretized with a structured hexahedral grid of $[N \times N \times N]$ cells per side. The set up of the solver is configured as done in the viscous vortex problem with second-order discretization schemes for the diffusive and temporal terms, solving the convection with particles and interpolating and projecting with third-order schemes. The Δt value is set up to define a maximum CFL number of approximately 6. Three different mesh resolutions are employed for the convergence study: N=64,



(a) Kinetic energy

(b) Dissipation rate

Figure 10. Temporal evolution of kinetic energy and its derivative. Solution for different mesh refinements using the proposed algorithm compared with Wang's [31] solution.

N=128, and N=256. Note that theses cases require huge computational resources (the finest mesh involves more than 16 cells), therefore the test also evaluates the parallel implementation of the methodology. Particularly, the cases with $N = 128$ and $N = 256$ are parallelized with 16 and 40 processors respectively.

The temporal evolutions of E_k and $-dE_k/dt$ are presented respectively in Fig 10a and Fig. 10b for each mesh resolution analyzed. Firstly, the grid convergence is assessed qualitatively where the solution with N=256 with the energy progress on time and particularly with the evolution of the dissipation rate. Then, a quantitative evaluation of the results is achieved measuring the differences on E_k between the numerical solutions obtained and the pseudo-spectral reference. The RMSE are 0.154, 0.0496 and 0.0129 for the cases with $N = 64$, 128 and 256 respectively. For the last pair of solutions the error falls with a second-order rate.

Parallel Performance. The case with $N = 256$ the parallel performance. This implementation does not impose any restriction in the domain decomposition procedure. using the OpenFOAM®parallel capabilities based on the MPI protocol. However, the Lagrangian step of the PFVM requires implementing additional parallel functionalities. During X-IVS integration, the particles cross inter-domain boundaries [39]. The projection operator also requires interaction between subdomains: a particle projects its fields to the cell . The latest cells could belong to other subdomains, therefore particle data must be transferred in order to properly calculate the weighted average on cell centers.

The parallel performance is evaluated performing a strong scalability test where the number of processors involved in the computation is varied. The metric employed is the speed-up regarding the run with $n = 10$ processors, leading to $S_n = T_{10}/T_n$ with n the number of processors involved. The cluster employed is composed by Intel Xeon CPU E5-2640 (2CPU x 8 cores - 96Gb RAM) connected via Infiniband QDR 40 Gbps.

The problem is solved varying the number of processors by a factor of two up to 160 processors. The results are presented in Table 3 which shows an efficiency of 70% when 160 processors are employed.

n	10	20	40	80	160
S_n	1	1.84	3.51	6.52	11.2

Table 3. Speed-up of the incompressible flow solver implementation.

5. Conclusions

A mixed Lagrangian-Eulerian (LE) methodology which is second-order accurate in time and space was presented. The new strategy, here named Particle Finite Volume Method (PFVM), uses a symmetrical splitting of operators where the convection is solved with particles and the remaining terms are computed with a standard Finite Volume approach. the global rate of convergence, interpolation and projection operators, which are based on Taylor expansion, were developed. Including the first three terms of the expansion, these operators achieve third-order on structured meshes, while second-order convergence is always obtained using two terms. The LE solver was firstly tested in an advection-diffusion problem obtaining quadratic convergence of the error in structured and unstructured meshes. The present approach has an absolute error which is two order of magnitude lower than a full Eulerian solution.

In the case of incompressible flows, the symmetrical splitting of the momentum equation is used as a predictor in the segregated pressure-velocity scheme PIMPLE. Several tests varying Reynolds number, boundary conditions, mesh topology, and CFL numbers were carried out confirming that the particle-based strategy proposed leads to lower errors than a full Eulerian alternative in every case analyzed. Particularly, when large time-steps are employed, PFVM enlarges its accuracy advantage the full Eulerian solver PIMPLE, due to preserving the quadratic convergence.

It is important to remark the relevance of the operator splitting in order to obtain global quadratic convergence. Tests showed that if a first-order splitting is employed, even using second-order solutions per algorithm stage, the global convergence is first-order. Thus, global second-order convergence is only obtained when a proper splitting strategy is employed, example the symmetrical Strang approach, in combination with high-order spatial and time discretization schemes.

Due to employing the OpenFOAM®library, the implementation naturally inherits most of the features for solving PDE's, including pre- and post-processing capabilities. Also, good parallel efficiency was obtained with more than one hundred processors. In this context, reliable comparisons against other flow solvers can be performed as follows: comparing the curve of CPU time versus RMSE, the current PFVM implementation is more efficient, reaching the same error level than a Eulerian alternative employing slightly less computing time.

The promissory results shown in this paper support future research, particularly the development of an implicit projection operator which should lead to lower computing times, increasing differences between the LE and the Eulerian efficiency. In addition, a natural extension of this methodology is its application to two-phase or free-surface problems.

6. Acknowledgments

The authors would thank to Dr. Pedro Morin because of his valuable contribution on technical discussion meetings. Also, the authors would like to thank Universidad Nacional del Litoral (CAI+D 2016 PIC 50420150100067LI and CAI+D 2016 PJ 50020150100018LI) and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2016-2908).

The final publication is available at Elsevier via https://doi.org/10.1016/j.jcp.2018.11.034.

7. Bibliography

[1] J. Donea, A. Huerta, Finite Element Method for Flow Problems, Wiley, Chichester England, 2003.

[2] J. Monaghan, An introduction to SPH, Computational Physics Communications 48 (1988) 89–96.

[3] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, Nuclear Science and Engineering 123 (3) (1996) 421–434. doi:10.13182/NSE96-A24205.

[4] R. M. Nestor, M. Basa, M. Lastiwka, N. J. Quinlan, Extension of the finite volume particle method to viscous flow, Journal of Computational Physics 228 (5) (2009) 1733 – 1749. doi:10.1016/j.jcp.2008.11.003.

[5] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, M. Teschner, Implicit incompressible sph, IEEE Transactions on Visualization and Computer Graphics 20 (3) (2014) 426–435. doi:10.1109/TVCG.2013.105.

[6] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, Computer Methods in Applied Mechanics and Engineering 139 (1) (1996) 3 – 47. doi:10.1016/S0045-7825(96)01078-X.

[7] F. H. Harlow, Pic and its progeny, Computer Physics Communications 48 (1) (1988) 1 – 10. doi:10.1016/0010-4655(88)90017-3.

[8] S. Idelsohn, E. Oñate, N. Calvo, F. Del Pin, The meshless finite element method, Int. J. Num. Meth. Engng. 58,6 (2003) 893–912.

[9] Y.-H. Hwang, A moving particle method with embedded pressure mesh (mppm) for incompressible flow calculations, Numerical Heat Transfer, Part B: Fundamentals 60 (5) (2011) 370–398. doi:10.1080/10407790.2011.601178.

[10] S. Idelsohn, E. Oñate, F. Del Pin, The particle finite element method a powerful tool to solve incompressible flows with free-surfaces and breaking waves., International Journal of Numerical Methods 61 (2004) 964–989.

[11] S. Idelsohn, N. Nigro, A. Limache, E. Oñate., Large time-step explicit integration method for solving problems with dominant convection, Comp. Meth. in Applied Mechanics and Engineering 217-220 (2012) 168–185. doi:10.1016/j.cma.2011.12.008.

[12] S. Idelsohn, N. Nigro, J. Gimenez, R. Rossi, J. Marti., A fast and accurate method to solve the incompressible navier-stokes equations, Engineering Computations 30-Iss:2 (2013) 197–222. doi:10.1108/02644401311304854.

[13] J. Stam, Stable fluids, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1999, pp. 121–128. doi:10.1145/311535.311548.

[14] A. Allievi, R. Bermejo, Finite element modified method of characteristics for Navier-Stokes equations, Int. J. Numer. Meth. Fluids 32 (2000) 439–464.

[15] J. Gimenez, Enlarging time steps for solving one and two phase flows using the particle finite element method, Ph.D. thesis, Facultad de Ingeniería y Ciencias Hídricas - Centro de Investigaciones en Mecanica Computacional. Santa Fe, Argentina (2015).

[16] K.-S. Liu, T. W.-H. Sheu, Y.-H. Hwang, K.-C. Ng, High-order particle method for solving incompressible navier–stokes equations within a mixed lagrangian–eulerian framework, Computer Methods in Applied Mechanics and Engineering 325 (2017) 77 – 101. doi:10.1016/j.cma.2017.07.001.

[17] J. Perot, An analysis of the fractional step method, Journal of Computational Physics 108 (1) (1993) 51 – 58. doi:10.1006/jcph.1993.1162.

[18] R. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational Physics 62 (1) (1986) 40 – 65. doi:10.1016/0021-9991(86)90099-9.

[19] J. M. Maljaars, R. J. Labeur, M. Möller, A hybridized discontinuous galerkin framework for high-order particle–mesh operator splitting of the incompressible navier–stokes equations, Journal of Computational Physics 358 (2018) 150 – 172. doi:10.1016/j.jcp.2017.12.036.

[20] G. Strang, On the construction and comparison of difference schemes, SIAM j. Numer. Anal. 5(3) (1968) 506 – 517. doi:10.1137/0705041.

[21] L. S. Caretto, A. D. Gosman, S. V. Patankar, D. B. Spalding, Two calculation procedures for steady, three-dimensional flows with recirculation, in: H. Cabannes, R. Temam (Eds.), Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Springer Berlin Heidelberg, Berlin, Heidelberg, 1973, pp. 60–68.

[22] I. Gatin, V. Vukčević, H. Jasak, H. Rusche, Enhanced coupling of solid body motion and fluid flow in finite volume framework, Ocean Engineering 143 (2017) 295 – 304. doi:10.1016/j.oceaneng.2017.08.009.

[23] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (6) (1998) 620–631. doi:10.1063/1.168744.

[24] H. Jasak, Error analysis and estimation for finite volume method with applications to fluid flow, Ph.D. thesis, Imperial College of Science, Technology and Medicine (1996).

[25] R. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
URL https://books.google.com.ar/books?id=0_ZjpMSZiwoC

[26] S. MacNamara, G. Strang, Operator splitting, in: R. Glowinski, S. Osher, W. Yin (Eds.), Splitting Methods in Communication, Imaging, Science, and Engineering, Springer, 2017.

[27] S. Idelsohn, E. Oñate, N. Nigro, P. Becker, J. Gimenez, Lagrangian versus eulerian integration errors, Computer Methods in Applied Mechanics and Engineering 293(0) (2015) 191–206. doi:10.1016/j.cma.2015.04.003.

[28] L. D. G. Sigalotti, J. Klapp, O. Rendón, C. A. Vargas, F. Peña-Polo, On the kernel and particle consistency in smoothed particle hydrodynamics, Applied Numerical Mathematics 108 (Supplement C) (2016) 242 – 255. doi:10.1016/j.apnum.2016.05.007.

[29] S. Sarra, E. Kanse, Multiquadric Radial Basis Function Approximation Methods for the Numerical Solution of Partial Differential Equations, in: S. Atluri (Ed.), Advances in Computational Mechanics, Vol. 2, Tech Science Press, 2009.

[30] B. Fornberg, T. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, Computers & Mathematics with Applications 43 (3) (2002) 473 – 490. doi:10.1016/S0898-1221(01)00299-1.

[31] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P. Persson, B. Leer, M. Visbal, Highorder cfd methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (8) (2013) 811–845. doi:10.1002/fld.3767.

[32] J. Gimenez, L. González, An extended validation of the last generation of particle finite element method for free surface flows, Journal of Computational Physics 284 (0) (2015) 186 – 205. doi:10.1016/j.jcp.2014.12.025.

[33] P. Nadukandi, B. Servan-Camas, P. A. Becker, J. Garcia-Espinosa, Seakeeping with the semi-lagrangian particle finite element method, Computational Particle Mechanics 4 (3) (2017) 321–329. doi:10.1007/s40571-016-0127-2.

[34] P. Nadukandi, Numerically stable formulas for a particle-based explicit exponential integrator, Computational Mechanics 55 (5) (2015) 903–920. doi:10.1007/s00466-015-1142-5.

[35] R. Löhner and J. Ambrosiano, A vectorized particle tracer for unstructured grids, Journal of Computational Physics 91 (1) (1990) 22 – 31. doi:10.1016/0021-9991(90)90002-I.

[36] G. Wang, F. Duchaine, D. Papadogiannis, I. Duran, S. Moreau, L. Y. Gicquel, An overset grid method for large eddy simulation of turbomachinery stages, Journal of Computational Physics 274 (2014) 333 – 355. doi:10.1016/j.jcp.2014.06.006.

[37] H. Aguerre, S. M. Damián, J. Gimenez, N. Nigro, Conservative handling of arbitrary non-conformal interfaces using an efficient supermesh, Journal of Computational Physics 335 (2017) 21 – 49. doi:10.1016/j.jcp.2017.01.018.

[38] G. Taylor, A. Green, Mechanism of the production of small eddies from large ones, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 158 (895) (1937) 499–521.

[39] J. Gimenez, N. Nigro, S. Idelsohn, Evaluating the performance of the particle finite element method in parallel architectures, Computational Particle Mechanics 1 (1) (2014) 103–116. doi:10.1007/s40571-014-0009-4.