



Lagrangian versus Eulerian integration errors

Sergio Idelsohn^{a,b,c,*}, Eugenio Oñate^a, Norberto Nigro^c, Pablo Becker^a, Juan Gimenez^c

^aInternational Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain

^bInstitució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain

^cCentro de Investigaciones en Mecánica Computacional (CIMEC), CONICET-UNL, Santa Fe, Argentina

Received 8 July 2014; received in revised form 4 April 2015; accepted 10 April 2015

Available online 7 May 2015

Highlights

- The errors between the use of an Eulerian or a Lagrangian frame have been studied.
- For low Reynolds numbers the Eulerian frame is more competitive.
- Moderated-high Reynolds numbers are the cases that Lagrangian frames are competitive
- In a Lagrangian frame, a strategy to improve the projection errors is essential.
- For multi-fluid or free-surface flows the Lagrangian frames are more suitable.

Abstract

The possibility to use a Lagrangian frame to solve problems with large time-steps was successfully explored previously by the authors for the solution of homogeneous incompressible fluids and also for solving multi-fluid problems (Idelsohn et al. 2012; 2014; 2013). The strategy used by the authors was named Particle Finite Element Method second generation (PFEM-2).

The objective of this paper is to demonstrate in which circumstances the use of a Lagrangian frame with particles is more accurate than a classical Eulerian finite element method, and when large time-steps and/or coarse meshes may be used.

© 2015 Elsevier B.V. All rights reserved.

Keywords: Numerical integration errors; Particle methods; Multi-fluids flows; Lagrange formulations; Incompressible Navier–Stokes equations

1. Introduction

Over the last decades, computer simulation of incompressible fluid flow has been mainly based on the Eulerian formulation of the fluid mechanics equations on fixed domains [1]. During this period, hardware has evolved considerably increasing the speed performance of computations and allowing better facilities for data entry and the display of results. However in these decades there have been no substantial improvements on the numerical methods used concerning the efficiency of the algorithm. In most practical engineering problems, very fine mesh and very small time-steps are needed to reach acceptable results. This handicap exceeds most time the efficiency of current powerful computers.

* Corresponding author at: International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain.

E-mail addresses: sergio@cimne.upc.edu (S. Idelsohn), onate@cimne.upc.edu (E. Oñate), nnigro@santafe-conicet.gov.ar (N. Nigro), pablobecker@gmail.com (P. Becker), jmarcelogimenez@gmail.com (J. Gimenez).

More recently, particle-based methods in which each particle is followed in a Lagrangian manner have been used for fluid flow problems [2–5]. Monaghan [2] proposed the first ideas for the treatment of astrophysical hydrodynamic problems with the so-called Smoothed Particle Hydrodynamics Method (SPH) which was later, generalized to fluid mechanics problems [2–5]. Koshizuka and coworkers [6,7] developed a similar method to SPH, named Moving Particle Simulation (MPS). SPH and MPS belong to the family of the so-called meshless methods, as well as the Finite Point Method [8–10]. Lately, the meshless ideas were generalized to take into account the finite element type approximations in order to obtain more accurate solutions [11]. This method was called the Meshless Finite Element Method (MFEM) and uses the Extended Delaunay Tessellation [12] to build the mesh in a computing time, which is linear with the number of nodal points.

A natural evolution of the last work was the Particle Finite Element Method (PFEM) [13,14]. The PFEM combines the particle precept with the Finite Element Method (FEM) shape functions using a background finite element mesh. This mesh may be quickly rebuilt at each time-step (PFEM with moving mesh) or may be a fixed mesh (PFEM with fixed mesh). In the last case, the results from the Lagrangian particles are projected on a fixed mesh at each time-step. The idea of combining fixed meshes with moving particles is not new. It was introduced for convection diffusion problems in [15] and was used in the so-called Particle in Cell method (PIC) [16] and later in its extension called the Material Point Method (MPM) [17]. All these methods use a Finite Element (FE) background mesh. Despite that both the PFEM and the MPM use a fixed FE mesh and a set of Lagrangian particles, there are important differences in the way the particles are employed: thus, while in the MPM all computations are performed on the mesh, in the PFEM the aim is to calculate as much as possible on the particles, leaving small corrections to be performed on the mesh. However, the most important difference is that in the PFEM the particles do not represent a fixed amount of mass, but rather material points that transport only intrinsic properties of the fluid. This allows to use a variable number of particles and therefore simplifying refinement.

The PFEM has been successfully used to solve the Navier–Stokes equations [18–21] and fluid–structure interaction problems [22–25] as well as solid mechanics problems [26]. The advantages of the PFEM concerning the tracking of internal interfaces have also been explored and used to solve fluid mechanics problems including multi-fluid flows [27].

The possibility to use the PFEM to solve non-linear problems with large time-steps in order to obtain an accurate and fast solution was successfully explored by the authors for the solution of the homogeneous incompressible Navier–Stokes equations [28] and multi-fluid problems [29]. This new strategy was named PFEM-2. The objective of this paper is to demonstrate why PFEM-2 is more accurate than a classical Eulerian FEM in certain particular cases when large time-step and/or coarse meshes are used. The authors claim that nowadays, the best way to improve the efficiency of the algorithms in order to take advantages of the increasing computer power is the use of this particle-based method.

The layout of the paper is the following. After a general description in Section 2 of the particular time integration used to allow large time-steps, the PFEM-2 is described in Section 3. The analysis of the space and time integration errors for both the Eulerian and Lagrangian formulations is introduced in Section 4 for the convection–diffusion equation. Section 5 describes the generalization to the Navier–Stokes equations. Finally, prior to the conclusions, some numerical tests are presented to validate the method and the possibilities of using the PFEM-2 for solving convection dominant problem with large time-steps.

2. General description

Let t be the time variable. The initial value of the time-step will be called t^n , the final value t^{n+1} , the difference will be the time-step $\Delta t = t^{n+1} - t^n$ and some internal time value between t^{n+1} and t^n will be named $t^{n+\theta}$.

Let \mathbf{x} be the vector defining the position of a spatial point in a three-dimensional (3D) space and let \mathbf{x}_p be the vector defining the position of a material point (particle). This particle position may be described on a fixed frame as a function of time. For simplicity $\mathbf{x}_p(t)$ will be written as \mathbf{x}_p^t . The particles move according with the velocity field \mathbf{V} , which is a function of space and time $\mathbf{V} = \mathbf{V}(\mathbf{x}, t) = \mathbf{V}(\mathbf{x}_p^t, t)$. For simplicity it will be written as $\mathbf{V}^t(\mathbf{x})$ or $\mathbf{V}^t(\mathbf{x}_p^t)$.

The unknown function will be described by $T(\mathbf{x}, t) = T^t(\mathbf{x})$ for the case of an Eulerian frame and by $T(\mathbf{x}_p^t, t) = T^t(\mathbf{x}_p^t)$ for a Lagrangian frame. The unknown function may be a scalar function (temperature, pressure, level set position, etc.) or a vector function. The particular case in which the unknown function is the velocity field itself (as in the case of the Navier–Stokes equations) will be considered separately.

In a general way, all convective equations may be written in an Eulerian frame as:

$$\left(\frac{\partial T}{\partial t} + \mathbf{V}\nabla T\right) = Q \tag{2.1}$$

where Q includes the source term $q^t(\mathbf{x})$, the diffusion term $\nabla \cdot (\kappa \nabla T)$, the reaction term cT , etc.

$$Q = Q^t(\mathbf{x}) = q^t(\mathbf{x}) + \nabla \cdot (\kappa \nabla T) + cT + \dots \tag{2.2}$$

The same convective equation may be written in a Lagrangian frame as:

$$\begin{cases} \frac{DT}{Dt} = Q \\ \frac{D\mathbf{x}_p}{Dt} = \mathbf{V} \end{cases} \tag{2.3}$$

where Q is

$$Q = Q^t(\mathbf{x}_p^t) = q^t(\mathbf{x}_p^t) + \nabla \cdot [\kappa \nabla T^t(\mathbf{x}_p^t)] - cT^t(\mathbf{x}_p^t) + \dots \tag{2.4}$$

Time integration.

Eqs. (2.1) and (2.3) will be integrated in time in an Eulerian frame using the simplest linear in time θ -method:

$$T^{n+1}(\mathbf{x}) = T^n(\mathbf{x}) + \int_n^{n+1} (Q^t - \mathbf{V}^t \nabla T^t) dt \approx T^n(\mathbf{x}) + (Q - \mathbf{V} \nabla T)^{n+\theta} \Delta t \tag{2.5}$$

with the definition of any function $f^{n+\theta} = (1 - \theta)f^n + \theta f^{n+1}$.

In the same way, in the Lagrangian frame the θ -method reads:

$$\begin{cases} T^{n+1}(\mathbf{x}_p^{n+1}) = T^n(\mathbf{x}_p^n) + \int_n^{n+1} Q^t dt \approx T^n(\mathbf{x}_p^n) + Q^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \\ \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^t dt \approx \mathbf{x}_p^n + \mathbf{V}^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \end{cases} \tag{2.6}$$

with the definition $f^{n+\theta}(\mathbf{x}_p^{n+\theta}) = (1 - \theta)f^n(\mathbf{x}_p^n) + \theta f^{n+1}(\mathbf{x}_p^{n+1})$.

The time integration error of a function f^t for $\theta = 1/2$ is proportional to the second derivative of the function and Δt^2 , i.e.:

$$\int_n^{n+1} f^t dt \approx f^{n+\theta} \Delta t \pm \varepsilon \Delta t \quad \text{with } \varepsilon = o\left\langle \frac{\partial^2 f}{\partial t^2} \Delta t^2 \right\rangle. \tag{2.7}$$

Other form of time integration for the Lagrangian frame:

It is interesting to note that using the same linear θ -method, the Lagrangian frame allows different ways to evaluate the integral of a function. For instance, the first equation of (2.6) may be also written as:

$$T^{n+1}(\mathbf{x}_p^{n+1}) \approx T^n(\mathbf{x}_p^n) + (1 - \theta) \int_n^{n+1} Q^n(\mathbf{x}_p^t) dt + \theta \int_n^{n+1} Q^{n+1}(\mathbf{x}_p^t) dt \tag{2.8}$$

in which the functions evaluated at time t^n and t^{n+1} are integrated with moving particles following the streamlines at the respective time-steps. Eq. (2.8) seems to be more accurate than (2.6) because it takes into account all the intermediate positions of the particles. Another possibility is a mix between (2.6) and (2.8) as:

$$T^{n+1}(\mathbf{x}_p^{n+1}) \approx T^n(\mathbf{x}_p^n) + (1 - \theta) \int_n^{n+1} Q^n(\mathbf{x}_p^t) dt + \theta Q^{n+1}(\mathbf{x}_p^{n+1}) \Delta t. \tag{2.9}$$

The advantage of (2.9) over (2.8) is that the integration over the streamlines is performed in the explicit part of the equation and the remaining implicit part is simpler. To perform the first integral in (2.9) a possible numerical strategy is to divide the time step into m time subintervals. The analytical solution to solve this integral for linear unknown approximation was also presented in [30].

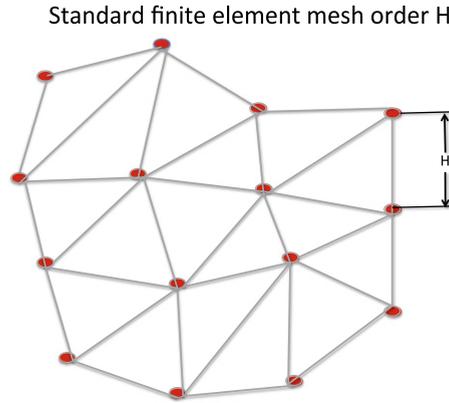


Fig. 3.1. Fixed finite element mesh of order H .

The use of (2.9) for the unknown with $\theta \neq 0$ and the use of (2.9) for the integration of the velocity – second equation of (2.6) – with $\theta = 0$, has been named *X-IVAS+implicit correction* in [30].

$$\mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + (1 - \theta) \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t) dt + \theta \mathbf{V}^{n+1}(\mathbf{x}_p^{n+1}) \Delta t. \quad (2.10)$$

This integration will be used in all the examples presented using a Lagrangian frame. However, in the present study for the error analysis, the standard θ -method will be considered, assuming that the X-IVAS integration improves the results and decreases the integration errors, but without evaluating this difference.

3. The Particle Finite Element Method, second generation (PFEM-2)

The Particle Finite Element Method, second generation (PFEM-2) has been described in [28,29]. Only a summary will be presented here for completeness. Detail can be found in [28,29].

While PFEM-2 can be used either with fixed or moving mesh, however, in this paper only PFEM-2 with fixed mesh will be described. The reason is that the goal of this work is to compare the errors between Lagrangian and Eulerian formulations, and therefore it is more consistent to work with a fixed mesh in both frameworks.

One of the main characteristics of PFEM-2 with a fixed mesh is the existence of two meshes: one is a fixed mesh which is the standard FE mesh, while the other mesh, which is a fictitious mesh, is the particle distribution. This last mesh may be considered as a moving mesh without connectivities. On the fixed mesh, all the derivatives of the unknown functions will be evaluated and also will be used to interpolate any variable from the FE mesh to the particle mesh. On the particle mesh only the unknown function will be evaluated but not its derivatives. For this reason, the particle mesh does not need connectivities, only the particle positions are enough to represent the unknown function.

For reasons to be justified later in the error analysis, the particle mesh will be finer than the FE mesh. Let us assume that the FE mesh is of order H and the particle mesh is of order h with $H \gg h$. See Figs. 3.1 and 3.2.

The steps to solve a problem using PFEM-2 are the following:

- (0) As a prerequisite, the value of the unknown $T^n(\mathbf{x})$, the source $Q^n(\mathbf{x})$ and the velocity field $\mathbf{V}^n(\mathbf{x})$ in all the nodes of the FE mesh must be known from the previous time-step. Also the unknown function is stored on the particles $T^n(\mathbf{x}_p^n)$.
- (1) Evaluate the new particle position \mathbf{x}_p^{n+1} using (2.10) with $\theta = 0$.
- (2) Evaluate the explicit part of (2.9) to give an initial prediction of $T^{n+1}(\mathbf{x}_p^{n+1})$. Let us call it $\hat{T}^{n+1}(\mathbf{x}_p^{n+1})$.
- (3) Project from the particle to the FE mesh $\hat{T}^{n+1}(\mathbf{x}_p^{n+1}) \implies \hat{T}^{n+1}(\mathbf{x})$.
- (4) Solve implicitly on the FE mesh, using the standard FEM and the second part of (2.9) to obtain $\delta T = T^{n+1}(\mathbf{x}) - \hat{T}^{n+1}(\mathbf{x})$.
- (5) Update the solution on the particles $T^{n+1}(\mathbf{x}_p^{n+1}) = \hat{T}^{n+1}(\mathbf{x}_p^{n+1}) + \delta T(\mathbf{x}_p^{n+1})$.
- (6) If necessary return to Step 1 with $\theta \neq 0$, otherwise go to the next time-step.

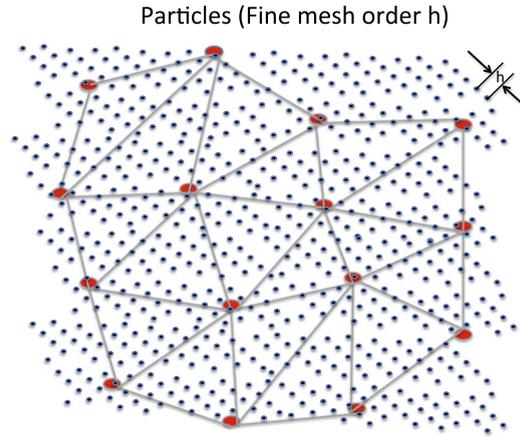


Fig. 3.2. Moving particle mesh of order h .

It must be noted that the iterative process described in Step 6 is not needed for the case of convection–diffusion equation where the velocity field is known at all times. In this case $\theta \neq 0$ may be used from the initial step. This situation is different for the Navier–Stokes equations, where the velocity field is the unknown. In this case, the problem is non-linear and the only way to obtain the exact solution is iteratively using at each iteration an approximation of the velocity at time t^{n+1} . However, the linearization introduced by imposing $\theta = 0$ on the first step has been enough to obtain accurate results in all the examples tested, making the iteration process unnecessary.

It is important to note also than in the Step 5 the interpolation from the coarse mesh to the particles is performed on the increments of the unknown function δT and not on the unknown function itself. This is imperative in order to preserve the arbitrary shape of the unknown function on the particles.

4. Lagrangian versus Eulerian integration errors

Eulerian frame:

The time integration using any numerical formula introduces some errors. In particular, the *linear θ -method* proposed in (2.5) introduces a minimum error for $\theta = 1/2$ proportional to the second time derivative of the integrated function and the square of the time-step, i.e.:

$$T^{n+1} = T^n + \int_n^{n+1} (Q^t - \mathbf{V}^t \nabla T^t) dt \approx T^n + [(Q - \mathbf{V} \nabla T)^{n+\theta} \pm \varepsilon_t] \Delta t \tag{4.1}$$

with the time integration error

$$\varepsilon_t = o \left\langle \frac{\partial^2}{\partial t^2} [Q - \mathbf{V} \nabla T] \Delta t^2 \right\rangle = o \left\langle [Q - \mathbf{V} \nabla T]^{\bullet\bullet} \Delta t^2 \right\rangle \tag{4.2}$$

where $[f]^{\bullet\bullet}$ means $\frac{\partial^2 f}{\partial t^2}$.

In the same way the FEM approximation of the functions and the space derivatives of the function introduces spatial error. This means that the sum of the function and some spatial integration errors ε_x is integrated in time by:

$$T^{n+1} = T^n + \int_n^{n+1} (Q^t - \mathbf{V}^t \nabla T^t \pm \varepsilon_x) dt \approx T^n + [(Q - \mathbf{V} \nabla T \pm \varepsilon_x)^{n+\theta} \pm \varepsilon_t] \Delta t. \tag{4.3}$$

In this paper only linear finite elements approximations of the unknown are considered. Then, the spatial errors are proportional to the second derivative of the functions and the square of the mesh size, i.e.:

$$\varepsilon_x = o \left\langle [Q - \mathbf{V} \nabla T]'' H^2 \right\rangle \tag{4.4}$$

where $[f]''$ means $\frac{\partial^2 f}{\partial \mathbf{x}^2}$.

Avoiding higher order terms the unknown function after a time-step will be:

$$T^{n+1} = T^n + (Q - \mathbf{V}\nabla T)^{n+\theta} \Delta t \pm \varepsilon^E.$$

Hence, the Eulerian error is:

$$\varepsilon^E = o\left\langle (Q'' - [\mathbf{V}\nabla T]'' H^2) \right\rangle \Delta t \pm o\left\langle (Q^{\bullet\bullet} - [\mathbf{V}\nabla T]^{\bullet\bullet} \Delta t^2) \right\rangle \Delta t. \quad (4.5)$$

Lagrangian frame:

In the same way, the errors in the numerical evaluation of the unknown function and the particle position for the Lagrangian frame will be:

$$\begin{cases} T^{n+1}(\mathbf{x}_p^{n+1}) = T^n(\mathbf{x}_p^n) + Q^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \pm o\left\langle Q'' H^2 \right\rangle \Delta t \pm o\left\langle Q^{\bullet\bullet} \Delta t^2 \right\rangle \Delta t \\ \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{V}^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \pm \underbrace{o\left\langle \mathbf{V}'' H^2 \right\rangle \Delta t \pm o\left\langle \mathbf{V}^{\bullet\bullet} \Delta t^2 \right\rangle \Delta t}_{\varepsilon_x^p}. \end{cases} \quad (4.6)$$

It must be noted that the error in the evaluation of the particle position ε_x^p introduces an error in the evaluation of the unknown function. Expanding in series around \mathbf{x}_p^{n+1}

$$T^{n+1}(\mathbf{x}_p^{n+1} + \varepsilon_x^p) = T^{n+1}(\mathbf{x}_p^{n+1}) + \nabla T \varepsilon_x^p + o\left\langle T'' (\varepsilon_x^p)^2 \right\rangle. \quad (4.7)$$

The first equation of (4.6) remains:

$$T^{n+1}(\mathbf{x}_p^{n+1}) = T^n(\mathbf{x}_p^n) + Q^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \pm o\left\langle Q'' H^2 \right\rangle \Delta t \pm o\left\langle Q^{\bullet\bullet} \Delta t^2 \right\rangle \Delta t \pm \nabla T \varepsilon_x^p \quad (4.8)$$

or also calling ε^L all the Lagrangian errors:

$$T^{n+1} = T^n + Q^{n+\theta} \Delta t \pm \varepsilon^L.$$

Introducing the value of ε_x^p from (4.6) into (4.8) the Lagrangian errors ε^L remains:

$$\varepsilon^L = \pm o\left\langle (Q'' \pm \mathbf{V}'' \nabla T) H^2 \right\rangle \Delta t \pm o\left\langle (Q^{\bullet\bullet} \pm \mathbf{V}^{\bullet\bullet} \nabla T) \Delta t^2 \right\rangle \Delta t. \quad (4.9)$$

Comparing the Eulerian errors in (4.5) with the Lagrangian ones in (4.9), the main differences are in the terms:

$$\underbrace{[\mathbf{V}\nabla T]'' H^2 \pm [\mathbf{V}\nabla T]^{\bullet\bullet} \Delta t^2}_{\text{Eulerian}} \neq \underbrace{\mathbf{V}'' \nabla T H^2 \pm \mathbf{V}^{\bullet\bullet} \nabla T \Delta t^2}_{\text{Lagrangian}}. \quad (4.10)$$

It is interesting to note that, despite the difference between the Eulerian and the Lagrangian errors are problem dependent, in general the errors in the Eulerian frame are higher than those in the Lagrangian one. For instance, in the standard convection–diffusion problem, where the convective field is known and has a constant or nearly constant velocity (see Fig. 4.1), Eq. (4.10) reads:

$$\underbrace{\mathbf{V}'' \nabla T H^2 \pm \mathbf{V}^{\bullet\bullet} \nabla T \Delta t^2}_{\text{Lagrangian}} = 0; \quad \underbrace{[\mathbf{V}\nabla T]'' H^2 \pm [\mathbf{V}\nabla T]^{\bullet\bullet} \Delta t^2}_{\text{Eulerian}} \neq 0.$$

This is so because in this case:

$$\mathbf{V}'' = \mathbf{V}^{\bullet\bullet} = 0, \quad \text{but } [\mathbf{V}\nabla T]'' \pm [\mathbf{V}\nabla T]^{\bullet\bullet} = [\mathbf{V}\nabla T]'' \pm \mathbf{V}[\nabla T]^{\bullet\bullet} \neq 0. \quad (4.11)$$

Projection errors:

The Lagrangian frame must add the projection errors to the previous integration ones. Projection errors are introduced while mapping the information from the particle mesh to the FE mesh.

Considering a minimum number of particles surrounding a fixed node (Fig. 4.2) a linear interpolation may be used to project the unknown function from the neighboring particles around a node to the node itself. In this case, the error

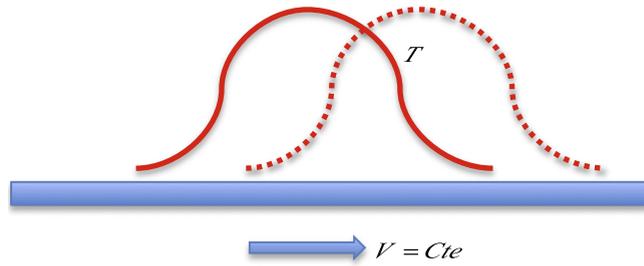


Fig. 4.1. Convection of a unknown function in a constant velocity field.

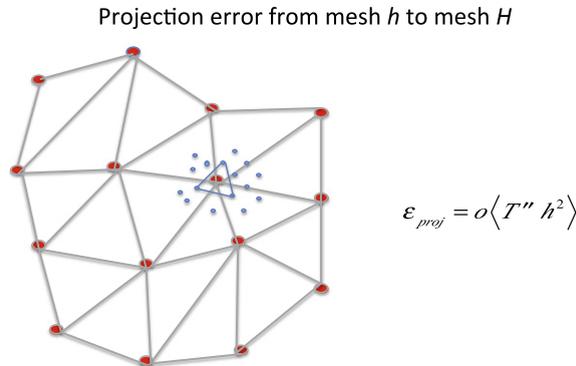


Fig. 4.2. Projection errors from mesh h to mesh H .

will be proportional to the second derivative of the unknown and the size of the particle mesh h .

$$\epsilon_P = o\langle T'' h^2 \rangle. \tag{4.12}$$

This projection error is introduced at each projection step independently of the time-step size. Then, in a total period of time $(t^f - t^0)$, the Lagrangian error becomes:

$$\epsilon_L = o\langle (Q'' \pm \mathbf{V}'' \nabla T) H^2 \rangle (t^f - t^0) \pm o\langle (Q'' \pm \mathbf{V}'' \nabla T) \Delta t^2 \rangle (t^f - t^0) \pm o\langle T'' h^2 \rangle \frac{(t^f - t^0)}{\Delta t}. \tag{4.13}$$

The importance to use a fine particle mesh with $h \ll H$ becomes now obvious for decreasing projection errors. Several existing Lagrangian methods, as for instance the Characteristic Galerkin method [31], use a single mesh. In these methods the projection errors are of order H^2 , the same order as the spatial diffusive error, and therefore they lead to very diffusive methods.

Errors due to adding and removing particles:

Another handicap of PFEM-2 is the permanent need of adding and removing particles from the fine mesh. Effectively, the particle mesh is a moving mesh. This means that particles move accordingly to the velocity field. In such cases in which the velocity field or the fixed mesh size H are not constant, keeping the mesh size h sufficiently small in order to obtain the projection errors to acceptable values is not always possible. Adding and sometimes removing particles becomes a need to preserve the accuracy and the computational cost acceptable.

Starting from Eq. (4.12), it is possible to obtain guidelines to define the number of particles. Particles must be added or removed accordingly to the following two criteria:

- (1) when the local h size increases or decreases.
- (2) when the second derivative of the unknown function increases or decreases.

By adding/removing particles it is possible to have a complete control over the h size in order to obtain the desired error value.

During the process of adding a new particle, the unknown function must be interpolated from the neighboring particles. In some cases, for simplicity the unknown function is interpolated from the fixed mesh. In this case, the

error introduced is of order

$$\varepsilon_I = o\left(T'' H^2\right). \tag{4.14}$$

Otherwise, when the surrounding particles are used for computing the new unknown value, the error is of order

$$\varepsilon_I = o\left(T'' h^2\right). \tag{4.15}$$

This is also the error introduced when a particle is removed.

Adding or removing particles must be performed only in such cases where it is strictly necessary due to the diffusive error introduced at each adding/removing step.

On the other hand, it is important to control the local value of h to avoid important projection errors in subsequent time-steps. A good balance between adding particles along time and removing as few as possible is a good strategy.

Advantages and disadvantages of the Eulerian/Lagrangian frames:

Comparing the equations for the errors in the Eulerian and Lagrangian frames the following conclusions may be drawn:

- (1) Eulerian frames are better for diffusion dominant problems. In these cases, the errors between the Lagrangian and Eulerian approaches are of the same order, but in the Lagrangian frames the projection errors must be added.
- (2) Lagrangian frames are better for convective dominant problems when the convective flow is constant or nearly constant. The remaining cases, when the convective flow presents high variations the Lagrangian or the Eulerian frame will be better or worse depending on the projection errors.
- (3) Eulerian frames are better for stationary problems. In these cases, $[\nabla T]^{**} = 0$ and there are not projection errors.

Generalization to the Navier–Stokes equations:

It is interesting to analyze it separately the particular case in which the unknown function is the convective term itself as for the Navier–Stokes equations. The errors equations are the same, but the conclusions are fundamentally different.

Generalizing Eq. (4.10), the difference between the errors in the Eulerian and Lagrangian frames for convective dominant problems (high Reynolds number) are:

$$\underbrace{[\nabla\nabla\nabla]'' H^2 \pm [\nabla\nabla\nabla]^{**}\Delta t^2}_{Eulerian} \neq \underbrace{\nabla''\nabla\nabla H^2 \pm \nabla^{**}\nabla\nabla\Delta t^2}_{Lagrangian}. \tag{4.16}$$

Adding the projection errors for a period $(t^f - t^0)$ reads:

$$\underbrace{([\nabla\nabla\nabla]'' H^2 \pm [\nabla\nabla\nabla]^{**}\Delta t^2)(t^f - t^0)}_{Eulerian} \neq \underbrace{(\nabla''\nabla\nabla H^2 \pm \nabla^{**}\nabla\nabla\Delta t^2)(t^f - t^0) \pm \nabla''h^2 \frac{(t^f - t^0)}{\Delta t}}_{Lagrangian}. \tag{4.17}$$

It is not anymore possible to separate the case in which the convective field is constant from the case in which $\nabla\nabla$ is not constant. Nevertheless, some particular cases may be analyzed in order to draw some conclusions:

1. Eulerian frames are better for low Reynolds number. In these cases, the errors between the Lagrangian and the Eulerian approaches are of the same order, but on Lagrangian frames the projection errors must be added.
2. Lagrangian frames are better for convective dominant problems when the velocity has a smooth variation in time but the gradient of the velocity has high spatial variations. This case is very common in fluid mechanics problems. See for instance Fig. 4.3. The remaining cases are better or worse depending on the projection errors.
3. Eulerian frames are better for stationary problems. In these cases, $[\nabla\nabla]^{**} = 0$ and there are no projection errors.
4. Lagrangian frames are better for multi-fluid flows. This is because Eulerian frames need to solve a level set equation to know the position of the interface. The level set equation [32,33] is a convection equation that requires small time-steps to yield accurate results due to the considerations concluded in the previous section, i.e.:

$$\frac{\partial^2 \mathbf{V}}{\partial t^2} \nabla \mathbf{V} < 1 \quad \text{and} \quad \frac{\partial^2 [\nabla \nabla \mathbf{V}]}{\partial t^2} \approx \mathbf{V} \frac{\partial^2 [\nabla \mathbf{V}]}{\partial t^2} \gg 1.$$

$$\frac{\partial^2 \mathbf{V}}{\partial t^2} \nabla \mathbf{V} < 1 \quad \text{and} \quad \frac{\partial^2 [\mathbf{V} \nabla \mathbf{V}]}{\partial t^2} \approx \mathbf{V} \frac{\partial^2 [\nabla \mathbf{V}]}{\partial t^2} \gg 1$$

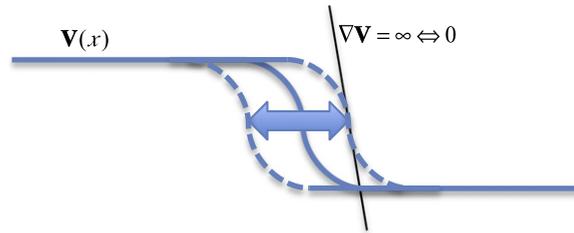


Fig. 4.3. Case with a small variation of the velocity field and a large variation of the velocity gradient.

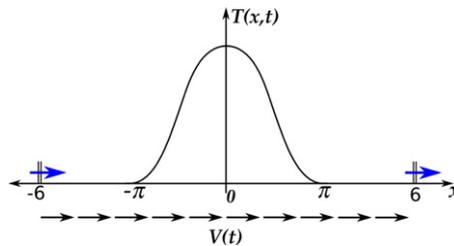


Fig. 5.1. Manufactured 1D problem.

5. Numerical examples

Manufactured, one-dimensional problem:

In order to test the validity of the expressions presented in Section 3, a particular 1D problem was tailored to have an analytical solution in order to compare the results.

The velocity field is constant in space but has a variation in time with a constant coefficient ω_V such that:

$$V^t(x) = V_0 + V_t \sin(\omega_V t) \tag{5.1}$$

with $V_0 = 5$ and $V_t = 2.5$.

The target solution is a trigonometric function that does not move in the space x , but rather changes its amplitude in time, affected by the parameter ω :

$$T^t(x) = [\cos(x) + 1] \sin(\omega t) \quad \forall -\pi \leq x \leq \pi. \tag{5.2}$$

An initial value of $T^0 = 0$ and periodic boundary conditions $T(-6) = T(6)$ were used. For these conditions, the source term $Q^t(x)$ becomes (5.3). It is interesting to note that despite the target $T(x, t)$ must not be affected by the velocity, the load actually does depend on V^t in order to obtain the desired solution.

$$Q^t(x) = \begin{cases} [\cos(x) + 1] \omega \cos(\omega t) - V^t \sin(x) \sin(\omega t) + \kappa \cos(x) \sin(\omega t) & \forall -\pi \leq x \leq \pi \\ 0 & \forall -\pi > x \text{ or } x > \pi. \end{cases} \tag{5.3}$$

The 1D space was split in equal elements of length $H = 0.2$. The conductivity coefficient was set to $\kappa = 0.01$. This gives a Peclet number based on the source length from 1570 to 4700. Fig. 5.1 shows a graphic representation of the problem.

The problem was solved with different time-steps from $\Delta t = 0.01$ to $\Delta t = 20$. Defining the Courant number as the number of elements that a material point passes by at each time step ($Co = V \Delta t / H$), the range of this value is 0.125 to 7.5 for this problem. The frequency ω of the amplitude was also variable from $\omega = 1$ for the nearly stationary case, to $\omega = 10$ for the most transient case. The number of particle used in the Lagrangian frames was set to around 10 particles by element in order to reduce the projection errors to a minimum, leading to $h = 1/10H$.

Table 5.1

Manufactured 1D problem. Average of the L^2 norm $\times 10^{-2}$ of the error for different frequencies.

Average of the L^2 norm by 10^2 during the first 50 s of simulation						
Dt	W = 1		W = 5		W = 10	
	Lagrangian	Eulerian	Lagrangian	Eulerian	Lagrangian	Eulerian
0.20	0.30	3.7	0.31	$\gg 1$	0.35	$\gg 1$
0.10	0.30	0.96	0.31	$\gg 1$	0.35	$\gg 1$
0.05	0.30	0.26	0.30	3.3	0.35	3.87
0.01	0.30	0.05	0.30	0.09	0.35	0.14

Table 5.2

Manufactured 1-D problem. Average of the L^2 norm of the error for different mesh size.

Frame Courant Dt	Fine mesh $h = 0.1$		Standard mesh $h = 0.2$		Coarse mesh $h = 0.4$	
	Lagrangian	Eulerian	Lagrangian	Eulerian	Lagrangian	Eulerian
	Co from 0.25 to 15		Co from 0.125 to 7.5		Co from 0.0625 to 3.5	
0.20	0.13	$\gg 1$	0.31	$\gg 1$	0.94	$\gg 1$
0.10	0.13	$\gg 1$	0.31	$\gg 1$	0.94	$\gg 1$
0.05	0.13	3.3	0.30	3.3	0.94	3.4
0.01	0.13	0.11	0.30	0.09	0.94	0.21

This very simple 1D problem is interesting because it represents, to some extent, the problem of a stationary shock wave in a fluid that moves at high speed. As the shock wave can be identified on a fixed part of the domain, this problem is usually solved with an Eulerian method.

Table 5.1 presents the results of the L^2 norm of the error averaged for a period of simulation of 50 s. In this problem, the velocity field in all the different cases tested was the same, with $V'' = 0$ and $V^{\bullet\bullet} \neq 0$. The unknown function $T^t(x)$ has a variation in time depending of the frequency ω . For small values of ω , small values of the variable $[V \nabla T]^{\bullet\bullet}$ will be obtained. Contrary, for high frequencies, this value will be high. Table 5.1 confirms the expected results presented in Section 3. High frequencies introduce high errors in the Eulerian frame. In these cases very small time-steps are needed to obtain acceptable results. However, the Lagrangian frame is independent from the frequency and the time-step size in this problem, because the error depends only on $V^{\bullet\bullet}$ instead of $[V \nabla T]^{\bullet\bullet}$. The Eulerian frame has acceptable results only in the near stationary case ($\omega = 1$) and even in this case, small time-steps are needed. For the other cases ($\omega = 5$ and $\omega = 10$), time-steps 20 times smaller are needed to have acceptable results in the Eulerian frame. Nevertheless, for very small time-steps, the Eulerian frame presents a better convergence to the exact results while in the Lagrangian frame, the error results do not decrease to lower values due to the projection errors.

Table 5.2 shows the same problem changing the mesh size. The results show a similar behavior. This means that the previous conclusions are independent from the mesh size. The solution in the Eulerian frame depends on the time-step and cannot be improved by decreasing the mesh size.

Rigid body rotation of Zalesak's disk:

This test consists in the pure convection of a region composed of a circle with a slot [34]. The computational domain is $R^2 : [0 : 100] \times [0 : 100]$. The convected region is a circle centered at (50; 75) with a radius of 15 and a slot of width 5 and height 25. The velocity field is a rigid body rotation around the center of the domain with a period of 628 time units:

$$\begin{aligned} V_x &= (\pi/314)(50 - y) \\ V_y &= (\pi/314)(x - 50). \end{aligned} \quad (5.4)$$

The FE mesh used was 100 points in each direction, conforming a Cartesian mesh with 20,000 triangles and a mesh size $H = 1$. Large time-steps were used in the simulation corresponding to a Courant number approximately to 4.5.

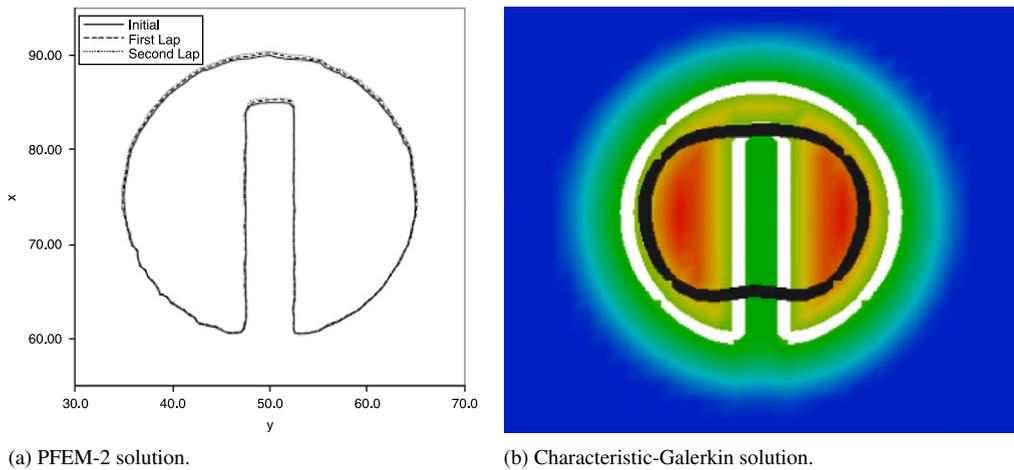


Fig. 5.2. Zalesak's disk results after one and two full revolutions with 100 grid points per direction and $Co = 4.5$. (a) PFEM-2 interface location after one and two revolutions; (b) Characteristic-Galerkin solution. White line: Initial interface location. Black line: interface location after one revolution. The interface is located where the distance function is valued zero, plotted with contour fill.

The number of particles used in the fine mesh of PFEM-2 was around six particles within each triangular element, leading to an average h of about $h \approx H/\sqrt{12}$.

This is an interesting problem for validating the possibility of a numerical method that perfectly convects a shape function. This is important not only for the permanent need to convect functions in practice, but also because in multi-fluid flows it is always necessary to convect the internal interfaces between two fluids. The accuracy of the results depends on a great extent in the accuracy of the convection of the internal interfaces. The particular case of convection without changes in the shapes must be solved exactly; otherwise large errors must be expected in other, more difficult to validate, cases.

The field at the initial step and after one and two revolutions are shown in Fig. 5.2 for two different methods based on two Lagrangian approaches: (a) PFEM-2; (b) Characteristic-Galerkin method.

In this example, the term V^{**} is zero, but the term $[V\nabla T]^{**}$ is not. This explains why this problem solved with the Eulerian frame only gives acceptable results for very small time-steps. However, despite that the two methods used for solving this problem of Fig. 5.2 are Lagrangian and therefore should have null convective errors, the Characteristic Galerkin Method lies a much more diffusive behavior. This difference is due to projection errors since the Characteristic-Galerkin Method [31] moves the nodes of the mesh in a Lagrangian way to obtain the unknown in the previous position. This is equivalent to use only one particle per node of the mesh. In this case, as expressed in Eq. (4.14) the error is the order $\epsilon_I = o\langle T'' H^2 \rangle$ instead of the order $\epsilon = o\langle T'' h^2 \rangle$ as for PFEM-2. Consequently, not all the Lagrangian methods are able to convect correctly a function. The importance to use several particles to avoid large projection errors is clearly evidenced in this example.

Convection–diffusion transport of a Gaussian hill:

The problem consists of transport a Gaussian hill signal with physical diffusion. The velocity field is a flow rotating around the center of a square domain. The initial Gaussian signal is displaced from the center of the domain at a certain radius and its shape makes the transported signal have a non-zero value in a limited region of the domain initially. Fig. 5.3 shows the problem definition. Since the velocity field corresponds to a rigid body motion, the solution can be easily decomposed into two parts: The conductivity of the material diffuses the Gaussian hill while the solution is convected; Using the Lagrangian (material) field as a reference system, the problem simply becomes a pure diffusion problem and therefore the analytical solution is trivial.

The initial solution is a Gaussian function with

$$T(x, 0) = \begin{cases} 1/4(1 + \cos(\pi X))(1 + \cos(\pi Y)) & \forall X^2 + Y^2 \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

with $\mathbf{X} = (\mathbf{x} - \mathbf{x}_0)/\sigma$. In the example tested $\mathbf{x}_0 = (\frac{1}{6}, \frac{1}{6})$ and $\sigma = 0.2$ were used. The rotating field had an angular rotation of $\omega = 2$ and the diffusive coefficient was set to $\kappa = 0.0001$.

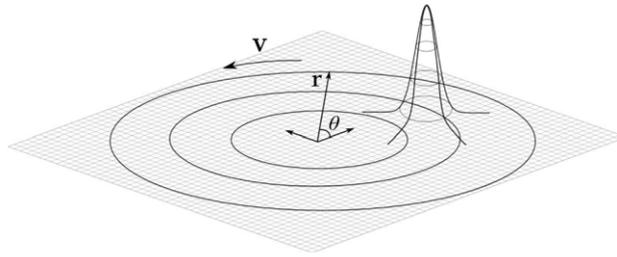


Fig. 5.3. Convection–diffusion transport of a Gaussian hill. Problem definition.

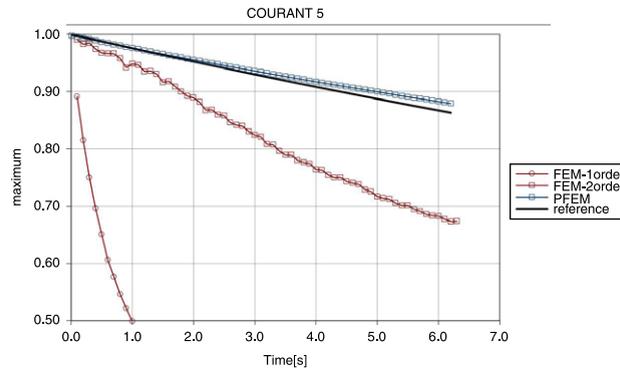


Fig. 5.4. Convection–diffusion transport of a Gaussian hill. Top value of the Gaussian hill at different time-steps with a constant Courant number = 5 for different methods.

The fixed mesh used was a 30×30 triangular mesh and the time-step was fixed to have a Courant number of 5. Note that this represents a very large time-step for this coarse mesh. The same problem was solved with a very fine mesh and a very small time-step to have a reference solution.

Fig. 5.4 shows the results of the top of the Gaussian hill during the period studied of seven seconds. The black full line is the reference solution. The two very diffusive curves correspond to the standard FEM using an Eulerian frame. The more diffusive results are obtained using explicit integration ($\theta = 0$). The other curves correspond to the more accurate implicit integration ($\theta = 1/2$). Both Eulerian solutions are very diffusive for this large time-step. The Eulerian and the Lagrangian solutions coincide if the time-step is reduced to a Courant number of 0.5.

In this example V^{**} and V'' are null, however $[V\nabla T]^{**}$ and $[V\nabla T]''$ are not. This explains, accordingly with (4.10), the big difference in the error between the Lagrangian and the Eulerian approaches. In PFEM-2 the projection error was decreased using nine particles in each element.

Rayleigh–Taylor instability:

This problem simulates the evolution of two fluids initially at rest in a gravity field. The density of the upper fluid is three times larger than the one placed underneath. Due to an initial disturbance the denser fluid moves down and the lighter one does the opposite. A mixture of fluids is created, which is lately segregated. The final state reaches a stable equilibrium with the denser fluid at the bottom layer and the lighter one at the top layer. The evolution of the instability has been investigated by Guermond and Quartapelle [35] among others.

This problem falls in the category of the cases discussed in Section 4, in which the unknown function is the actual velocity field. The main differences in the evaluation of the errors are those included in Eqs. (4.16)–(4.17). Furthermore it is a multi-fluid problem. This means that method based on Eulerian frames must make use of the concept of Level-Set (or Volume of Fluid) to determine the internal interface position. Hence, the evaluation of the interface position is dominated by a convective equation, where the errors are described in Eq. (4.10).

The computational domain was $[-d/2, d/2] \times [-2d, 2d]$. Other physical parameters were selected to give $Re = 1000$. A fixed finite element mesh was generated formed by 80,000 structured triangles ($H = 0.01$). Slip boundary conditions were set on each wall. The time-step was $\Delta t = 0.01$, which allows $Courant \approx 8$. Between five and eight particles per element were used which suppose an h between ($h \approx H/\sqrt{10}$) and $h \approx H/\sqrt{16}$.

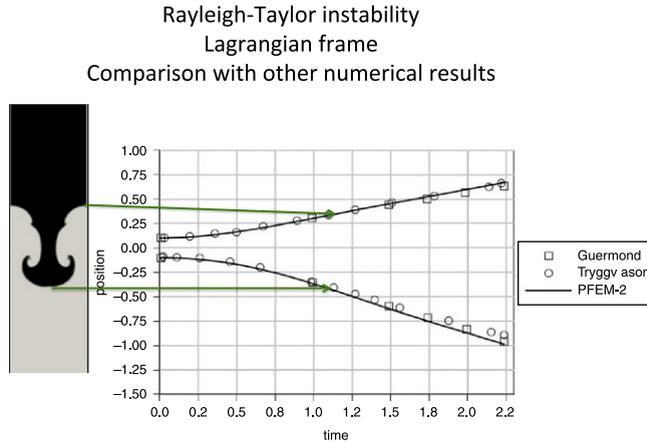


Fig. 5.5. Rayleigh–Taylor instability. Vertical position of the tip of the falling and rising fluid.

Results for the vertical position of the tip of the falling and rising fluid are shown in Fig. 5.5. It can be observed that the current solution is in good agreement with the reference results [35].

In order to emphasize the differences between the errors introduced in the space–time integration for large time-steps using a Lagrangian or an Eulerian frame, the current case was also simulated with a large range of time-steps using PFEM-2 and comparing with results obtained by the widely known OpenFOAM code. The solver InterFOAM was chosen, which implements a Volume of Fluid (VoF) algorithm for multi-fluid flows [36,37].

Fig. 5.6 presents the comparison of the solutions with PFEM-2 and OpenFOAM.

From snapshots it can be concluded that PFEM-2 yields approximately the same solution for the different time-steps. However, OpenFOAM only gives accurate results for $\Delta t < 0.001$. For larger time-steps the evolution of the mushroom-like interface differs from the reference results and this divergence is increased when the time-steps are increased. Moreover, simulations with OpenFOAM diverge when the Courant number 0.5 is reached. This happens at different times depending on the selected time-step and also on the solver used. For instance, for an implicit solver the solution is more stable, but the results are more diffusive.

Another relevant feature to take into account when comparing both algorithms is that similar CPU times are required to solve a time-step.

Comparing the CPU times required to complete one second of actual time in the current case, the results show that using the same time-step both solvers have similar performance, being OpenFOAM slightly faster. However, due to the capability of increasing the time-step with PFEM-2, much smaller CPU times are achieved with similar accuracy, being the solution with PFEM-2 one order of magnitude faster than that with OpenFOAM.

6. Conclusions

The errors between the use of an Eulerian or a Lagrangian frame have been studied for the cases of linear interpolation in time and in space.

For a Lagrangian frame, there are projection errors that must be added to the space–time integration. Projection errors are of the order $o(V'' h^2)$ and must be taken into account at each time-step.

The main difference in the evaluation of the errors between the Eulerian and Lagrangian frames are for the cases of convective dominant problems (high Peclet or Reynolds number). For these cases the main conclusion is that the integration errors are proportional to:

$$\underbrace{[\mathbf{V}\nabla\mathbf{V}]'' H^2 \pm [\mathbf{V}\nabla\mathbf{V}]'' \Delta t^2}_{Eulerian} \neq \underbrace{\mathbf{V}'' \nabla\mathbf{V} H^2 \pm \mathbf{V}'' \nabla\mathbf{V} \Delta t^2 \pm \mathbf{V}'' h^2 / \Delta t}_{Lagrangian}.$$

The following highlights can be drawn from these expressions:

- Low Reynolds numbers (or low Peclet numbers) have the same integration errors in both frames. Taking into account that Lagrangian errors have also projection errors, the Eulerian frame is more competitive in these cases.

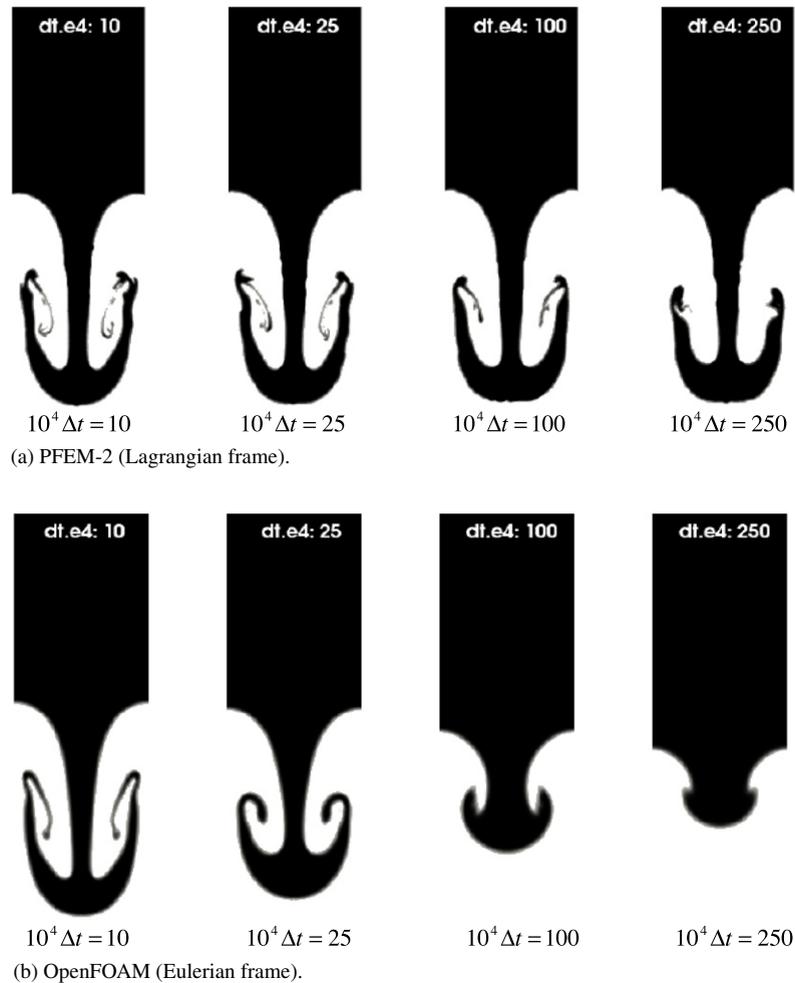


Fig. 5.6. Rayleigh–Taylor Instability. Different shape of the internal interface at $t = 2.25$ for different time-step sizes.

- For moderated or high Reynolds numbers are the cases that a Lagrangian frame may be competitive. In these cases, having big variations of the velocity gradients (in time or in space), but small variations of the velocity itself, improves the competitiveness of the Lagrangian frame. Contrarily, stationary problems or near constant in space velocity fields make an Eulerian frame more suitable.
- In all Lagrangian frame-based methods, the use of a strategy to improve the projection errors is essential. Otherwise projection errors overweight the advantages of the accuracy in the Lagrangian integration of the convective terms.
- For heterogeneous fluids (multi-fluid, multi-phase or free-surface flows), the Lagrangian frame is more suitable because the position of the internal interfaces or the free-surface is free from the Eulerian convective errors. In such cases, a reduction of more than one order of magnitude in the total computing time has been obtained using a Lagrangian frame.
- In cases where the physics of the problem imposes very small time-steps, an Eulerian frame is more convenient because the error does not depend on $1/\Delta t$.

Acknowledgments

This research was supported by the HFLUIDS project of the National RTD Plan of the Spanish Ministry of Science and Innovation I+D BIA2010-15880; by the ERC Advanced Grant projects “REALTIME”, AdG-2009325 and “SAFECON”, AdG-267521; by the F7 PEOPLE-ITN project Multi-scale Modeling of Landslides and Debris

Flows “MuMoLaDe”, REA Grant Agreement 289911; by the CONICET (Argentina) and the Universidad Nacional del Litoral (CAI+D Nro 50120110100435LI, (2011)) and by ANPCyT-FONCyT (Grants PICT 2013 0830). J. Gimenez and N. Nigro gratefully acknowledge also the support of CONICET and the ANPCyT, both from Argentina, through a doctoral grant from the FONCyT program.

References

- [1] J. Donea, A. Huerta, *Finite Element Method for Flow Problems*, John Wiley, 2003.
- [2] J.J. Monaghan, An introduction to SPH, *Comput. Phys. Comm.* 48 (1988) 89–96.
- [3] R.A. Gingold, J.J. Monaghan, Kernel estimates as a basis for general particle methods in hydrodynamics, *J. Comput. Phys.* 46 (1981) 429–453.
- [4] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics, theory and application to non-spherical stars, *Mon. Not. R. Astron. Soc.* 181 (1997) 375–389.
- [5] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, *Internat. J. Numer. Methods Fluids* 20 (1995) 1081–1106.
- [6] S. Koshizuka, H. Tamako, Y. Oka, A particle method for incompressible viscous flow with fluid fragmentation, *Comput. Fluid Dyn. J.* 113 (1995) 134–147.
- [7] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, *Nucl. Sci. Eng.* 123 (1996) 421–434.
- [8] E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, A finite point method in computational mechanics. Applications to convective transport and fluid flow, *Internat. J. Numer. Methods Engrg.* 39 (1996) 3839–3866.
- [9] E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, C. Sacco, A stabilized finite point method for analysis of fluid mechanics problems, *Comput. Methods Appl. Mech. Engrg.* 139 (1–4) (1996) 315–346.
- [10] E. Oñate, C. Sacco, S.R. Idelsohn, A finite point method for incompressible flow problems, *Comput. Vis. Sci.* 2 (2000) 67–75.
- [11] S.R. Idelsohn, E. Oñate, N. Calvo, F. Del Pin, The meshless finite element method, *Internat. J. Numer. Methods Engrg.* 58 (6) (2003) 893–912.
- [12] S.R. Idelsohn, N. Calvo, E. Oñate, Polyhedrization of an arbitrary point set, *Comput. Methods Appl. Mech. Engrg.* 192 (22–24) (2003) 2649–2668.
- [13] S.R. Idelsohn, E. Oñate, F. Del Pin, The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves, *Internat. J. Numer. Methods Engrg.* 61 (7) (2004) 964–984.
- [14] S.R. Idelsohn, E. Oñate, F. Del Pin, The particle finite element method a powerful tool to solve incompressible flows with free-surfaces and breaking waves, *Internat. J. Numer. Methods Engrg.* 61 (2004) 964–989.
- [15] S.P. Neuman, Adaptive Eulerian–Lagrangian finite element method for advection–dispersion, *Internat. J. Numer. Methods Engrg.* 20 (1984) 321–337.
- [16] Brackbill, Ruppel, FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, *J. Comput. Phys.* 65 (1986) 314–343.
- [17] Z. Więckowski, The material point method in large strain engineering problems, *Comput. Methods Appl. Mech. Engrg.* 193 (39) (2004) 4417–4438.
- [18] R. Aubry, S.R. Idelsohn, E. Oñate, Particle finite element method in fluid mechanics including thermal convection–diffusion, *Comput. Struct.* 83 (17–18) (2005) 1459–1475.
- [19] E. Oñate, J. Garcia, S.R. Idelsohn, Ship hydrodynamics, in: E. Stein, R. De Borst, T.J.R. Hughes (Eds.), *Encyclopedia of Computational Mechanics*, J. Wiley, 2004.
- [20] S.R. Idelsohn, E. Oñate, F. Del Pin, N. Calvo, Lagrangian formulation: the only way to solve some free-surface fluid mechanics problems, in: H.A. Mang, F.G. Rammerstorfer, J. Eberhardsteiner (Eds.), *Fifth World Congress on Computational Mechanics*, July 7–12, Vienna, Austria, 2002.
- [21] A. Larese, R. Rossi, E. Oñate, S.R. Idelsohn, Validation of the particle finite element method (PFEM) for simulation of the free-surface flows, *Eng. Comput.* 25 (4) (2008) 385–425.
- [22] S.R. Idelsohn, E. Oñate, F. Del Pin, N. Calvo, Fluid–structure interaction using the particle finite element method, *Comput. Methods Appl. Mech. Engrg.* 195 (2006) 2100–2113.
- [23] E. Oñate, S.R. Idelsohn, M.A. Celigueta, R. Rossi, Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free-surface flows, *Comput. Methods Appl. Mech. Engrg.* 197 (2008) 1777–1800.
- [24] S.R. Idelsohn, J. Marti, A. Limache, E. Oñate, Unified Lagrangian formulation for elastic solids and incompressible fluids. Application to fluid–structure interaction problems via the PFEM, *Comput. Methods Appl. Mech. Engrg.* 197 (2008) 1762–1776.
- [25] F. Del Pin, S.R. Idelsohn, E. Oñate, R. Aubry, The ALE/Lagrangian particle finite element method: a new approach to computation of free-surface flows and fluid-object interactions, *Comput. Fluids* 36 (2007) 27–38.
- [26] J. Oliver, J.C. Cante, R. Weyler, C. Gonzalez, J. Hernandez, Particle finite element methods in solid mechanics problems, in: E. Oñate, R. Owen (Eds.), *Computational Plasticity. Vol. 7*, Springer, Netherlands, 2007, pp. 87–103.
- [27] S.R. Idelsohn, M. Mier-Torrecilla, E. Oñate, Multi-fluid flows with the particle finite element method, *Comput. Methods Appl. Mech. Engrg.* 198 (2009) 2750–2767.
- [28] S.R. Idelsohn, N.M. Nigro, A. Limache, E. Oñate, Large time-step explicit integration method for solving problems with dominant convection, *Comput. Methods Appl. Mech. Engrg.* 217–220 (2012) 168–185.
- [29] S.R. Idelsohn, J. Marti, P. Becker, E. Oñate, Analysis of multi-fluid flows with large time-steps using the particle finite element method, *Internat. J. Numer. Methods Fluids* 75 (2014) 621–644. <http://dx.doi.org/10.1002/flid.3908>.
- [30] S.R. Idelsohn, N.M. Nigro, J.M. Gimenez, R. Rossi, J. Marti, A fast and accurate method to solve the incompressible Navier–Stokes equations, *Eng. Comput.* 30 (2) (2013) 197–222.
- [31] O. Pironneau, M. Tabata, Stability and convergence of a Galerkin characteristics finite element scheme of lumped mass type, *Internat. J. Numer. Methods Fluids* 64 (2010) 1240–1253.

- [32] S.J. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, *J. Comput. Phys.* 169 (2001) 463–502.
- [33] Douglas Enright, et al., A hybrid particle level set method for improved interface capturing, *J. Comput. Phys.* 183 (1) (2002) 83–116.
- [34] S. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* 31 (3) (1979) 335–362.
- [35] J. Guermond, L. Quartapelle, A projection FEM for variable density incompressible flows, *J. Comput. Phys.* 165 (2000) 167–188.
- [36] E. Berberovich, N.P. van Hinsberg, S. Jakirli, I.V. Roisman, C. Tropea, Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution, *Phys. Rev. E* 79 (2009) 036306. <http://dx.doi.org/10.1103/PhysRevE.79.036306>.
- [37] S. Marquez Damian, N. Nigro, An extended mixture model for the simultaneous treatment of small-scale and large-scale interfaces, *Internat. J. Numer. Methods Fluids* 75 (8) (2014) 547–574. <http://dx.doi.org/10.1002/flid.3906>.