

Implementing moving object capability in a two-phase Eulerian model for sediment transport applications

Eduard Puig Montellà^{1,*}, Cyrille Bonamy¹, Julien Chauchat¹, and Tian-Jian Hsu²

¹University of Grenoble Alpes, LEGI, G-INP, CNRS, 38000 Grenoble, France
Email address: eduard.puig.montella@gmail.com

²Civil and Environmental Engineering, Center for Applied Coastal Research, University of Delaware, Newark, DE 19711, USA

DOI: <https://doi.org/10.51560/ofj.v4.119>

Results with version(s): OpenFOAM® v2312

Repository: <https://github.com/SedFoam/sedfoam>

Abstract. This article presents the validation of a modified 6DoF solver which includes the contribution of particle forces on rigid objects. The solver is an extension of `sedFoam`, employed for a wide range of sediment transport applications. The first validation consists in a simple case of a sphere freely falling in a pure fluid. Then, the same sphere is submerged in a fluid with suspended particles. The next step involves placing a granular bed under the sphere to study its arrest due to particle-particle interactions. Lastly, the trajectory of a cylinder immersed in a uniform granular flow is studied to emphasize the lift and drag forces exerted by the particle phase on the solid object. Results show that `overSedDyMFoam` accurately reproduces experimental/numerical measurements successfully capturing the behavior of solid objects subjected to granular and fluid forces. This implementation is a significant step towards understanding complex interactions between fluid, particles, and structures.

1. Introduction

Human beings have been shaping nature for thousands of years, indeed, the anthropogenic transformation of the use of land and water resources has allowed human societies to thrive. Dams, ports, sea pipelines, dykes and other hydraulic structures have been built to ensure social and economic development. However, the structures placed on the seabed frequently contribute to the destruction of important ecosystems and disturb the equilibrium of the sediment transport. In addition to the environmental impact, human kind objects or structures may also be weakened or damaged due to intensive erosion around the foundations compromising their integrity. Erosion and sedimentation processes have been studied for decades but many open questions still remain. Unsurprisingly, fluid-particle-structure interactions (FPSI) make the problem of sediment transport even more challenging.

A classical FPSI worth highlighting is the scour in the vicinities of a solid object placed on the seabed induced by waves or currents. Scour around horizontal and vertical cylinders has been largely studied experimentally under wave and steady currents conditions [1–5]. From the numerical point of view, several works [6–8] have studied the flow and scour around a cylindrical pile since the early work of [9]. Single-phase models [6, 10–12] were proposed based on the coupling between a hydrodynamic (where RANS equations are solved) and a morphological model. In these works, the evolution of the scour is treated as a two-way coupling phenomenon between the deformable bed boundary and the flow field. Although these works provided invaluable insight to have a better understanding of the interaction between vortex structures and the sediment erosion, they relied on empirical sediment transport formulas obtained for steady and uniform flow conditions [13, 14] which hinders the study of hydraulic structures where the flow conditions are usually highly unsteady. In order to overcome this constraint, a few multiphase flow models have been developed to study the scour process and have a better understanding of the erosion dynamics.

* Corresponding author

Multiphase models that treat both fluid and solid phases as a continuum are referred to as Eulerian-Eulerian (E-E) models. These mesh-based approaches, employing methods like the finite element method (FEM) and the finite volume method (FVM), have proven to be successful in several sediment transport applications [7,8,15–17]. Although E-E models are computationally efficient, some physical phenomena at the grain scale may not be captured without proper closure forms. Therefore, Lagrangian-Lagrangian (L-L) methods, such as the smooth particle hydrodynamics (SPH-SPH) and SPH-Discrete element method (SPH-DEM) [18,19] couplings, have been proposed due to their capability in handling large deformations and deformable boundaries. L-L models are highly accurate due to their ability to track individual particles within a fluid flow. L-L approaches offer a detailed understanding of particle behavior and interactions, especially in complex scenarios like sediment transport or fluid-structure interactions (FSI). However, the accuracy of L-L models often comes at the cost of computational time. A midway alternative consists in using Eulerian-Lagrangian (E-L) methods. In these approaches, individual sediment particles are tracked within the fluid flow. Popular models like DEM coupled with Computational Fluid Dynamics (CFD) fall into this category and are frequently used in engineering problems such as soil erosion, [20], landslides [21], fluid-solid fluidization [22] or capillarity in porous media [23].

The aforementioned methods have been widely used to study FSI, often setting up solid mechanics problems using imported stress boundary conditions derived from CFD solutions. Regarding the single-phase and E-E models, many efforts have been done in coastal and ocean engineering problems without sediments to study the motion of ships and floating objects [24–27]. Additionally, some Eulerian approaches are commonly coupled with software for structural dynamics in order to simulate the fluid flow, the structural forces and the potential structural deformations [28,29]. From the Lagrangian perspective, similar FSI have been investigated to study the deformation of elastic structures [30–32]. Unfortunately, very few studies have focused on FPSI where the structure can move or deform. Current approaches heavily rely on computationally expensive methods using E-L or L-L approaches [33–35], with limited (none to our knowledge) exploration using more cost-effective methods like E-E or single-phase models. Indeed, the important progress of E-E and single-phase approaches to study the impact of structures on the erodible sediment bed [7,8,15–17] is not sufficient because the structural objects are assumed to be completely static and rigid, which can lead to unrealistic results when analyzing structural stability. Therefore, it is essential to shift the focus towards understanding the dynamics of structures influenced by complex sediment transport processes.

OpenFOAM® [36] is an open-source CFD toolbox that can be used to conduct coupled multi-physics simulations. Despite the large number of physics solvers present and accessible in OpenFOAM®, the fluid-sediment coupling is scarce in the official release. However, due to full access to the source code, OpenFOAM® can be easily customized. For sediment transport applications, several solvers have been developed based on OpenFOAM® such as SediFoam [37], scourFoam [38], sediDriftFoam [39] or sedFoam [16]. In this work, we will focus on sedFoam which originally stems from the twoPhaseEulerFoam solver [40]. sedFoam models the sediment phase as a continuum, requiring the specification of constitutive laws for sediment stresses. We opted for sedFoam as it is actively maintained by the community and it is the only two-phase continuum approach dedicated to sediment transport applications. Thanks to the two-phase formulation, physics are captured more naturally with fewer closure forms compared to single-phase models. Plus, the $\mu(I)$ rheology already implemented in sedFoam makes the solver very suitable to study dense granular flows. This conference article aims to introduce and lay the groundwork for future studies on FPSI within an Eulerian framework, thus, take the first steps to model the scour phenomenon and subsequent motion of immersed structures with acceptable computational efficiency. The sedFoam solver, as it was written originally, is not capable of dealing with dynamic objects. In this article, we extend the capability of sedFoam to allocate moving objects and set a proof of concept for further studies on FPSI. Hence, the originality of the present article lies in the implementation of equations of motion that take into account the sediment and fluid forces imposed by the environment acting on the solid objects. By accurately modeling the six degrees of freedom (6DoF), simulations using the new solver called overSedDyMFoam can provide valuable insights into the behavior of structures subjected to fluid and particle forces. Thus, it can be used to improve the design and operation of hydraulic structures, bridges or offshore platforms.

Section 2 of the article is dedicated to the mathematical formulation of the two-phase approach. The implementation of the model and the 6DoF solver is detailed in section 3. Then, a numerical validation and a few proof of concept examples are presented in section 4. Finally, section 5 gives the conclusions of the manuscript and draws the perspectives of the work.

2. Methodology

2.1. Two-phase flow governing equations.

The original formulation of `sedFoam` proposed by [16] has been modified to account for the mesh motion. Following [41], the integral forms of the mass continuity equations for the particle and fluid phase over a moving volume V bounded by the surface S are written as

$$\frac{\partial}{\partial t} \int_V \alpha dV + \int_S \alpha (\mathbf{u}^s - \mathbf{u}_g) \cdot \mathbf{n} dS = 0, \quad (1)$$

$$\frac{\partial}{\partial t} \int_V (1 - \alpha) dV + \int_S (1 - \alpha) (\mathbf{u}^f - \mathbf{u}_g) \cdot \mathbf{n} dS = 0. \quad (2)$$

Here, α , $\mathbf{u}^s$ and $\mathbf{u}^f$ are the particle volume fraction, the particle phase velocity and the fluid phase velocity, respectively. $\mathbf{n}$ is the outward normal vector on the surface S , whereas $\mathbf{u}_g$ stands for the mesh velocity.

The momentum conservation equations for the particle phase and fluid phase are given by

$$\begin{aligned} & \frac{\partial}{\partial t} \int_V \rho^s \alpha \mathbf{u}^s dV + \int_S \rho^s \alpha ((\mathbf{u}^s - \mathbf{u}_g) \mathbf{u}^s) \cdot \mathbf{n} dS = \int_V \alpha (\rho^s - \rho^f) \mathbf{g} dV \\ & + \int_V \frac{(1 - \alpha) \rho^f \nu^f}{K} (\mathbf{u}^f - \mathbf{u}^s) dV - \int_V \nabla p^s dV + \int_V \nabla \cdot \boldsymbol{\tau}^s dV - \int_V \alpha \nabla p^f dV, \end{aligned} \quad (3)$$

$$\begin{aligned} & \frac{\partial}{\partial t} \int_V \rho^f (1 - \alpha) \mathbf{u}^f dV + \int_S \rho^f (1 - \alpha) ((\mathbf{u}^f - \mathbf{u}_g) \mathbf{u}^f) \cdot \mathbf{n} dS = \int_V \frac{(1 - \alpha) \rho^f \nu^f}{K} (\mathbf{u}^s - \mathbf{u}^f) dV \\ & + \int_V \nabla \cdot \boldsymbol{\tau}^f dV - \int_V (1 - \alpha) \nabla p^f dV, \end{aligned} \quad (4)$$

where ρ^s is the particle density, ρ^f is the fluid density, p^f is the excess of pore pressure defined as the difference between the pore pressure and the hydrostatic pressure, p^s is the particle pressure, $\boldsymbol{\tau}^s$ is the granular shear stress, $\boldsymbol{\tau}^f$ is the fluid shear stress and K is the permeability of the porous medium. In this work, Ergun's approach [42] is used to model the permeability for dense packings while Gidaspow-Schiller-Naumann model [43] is adopted to deal with particle suspensions.

Although Eqn. 1-Eqn. 4 have a strong similarity to the two-phase mathematical formulation proposed by [16] to deal with non-moving meshes, this set of equations has to be treated differently due to the relative advection velocities $(\mathbf{u}^f - \mathbf{u}_g)$ and $(\mathbf{u}^s - \mathbf{u}_g)$, which, after the discretization of Eqn. 1-Eqn. 4, are reformulated as cell face relative fluxes. In section 3 we will discuss the computation of absolute and relative fluxes under topological changes.

2.2. Closure forms.

The fluid phase shear stress is expressed as

$$\boldsymbol{\tau}^f = 2\rho^f \nu^{eff} \mathbf{S}^f, \quad (5)$$

where

$$\mathbf{S}^k = \frac{1}{2} (\nabla \mathbf{u}^k + (\nabla \mathbf{u}^k)^T) - \frac{1}{3} tr(\nabla \mathbf{u}^k), \quad (6)$$

is the deviatoric and symmetric part of the velocity gradient for the phase k ($k = f$ for the fluid phase and $k = s$ for the particle phase) and ν^{eff} is the effective viscosity, which according to [44], can be computed with the phenomenological expression given by

$$\nu^{eff} = \nu^f \left[1 + 2.5\alpha \left(1 - \frac{\alpha}{\alpha_{max}} \right)^{-1} \right], \quad (7)$$

where α_{max} is the maximum volume concentration set to 0.625 in this work.

Following [16, 45, 46], the total particle phase pressure p^s is defined as the sum of a viscous shear rate-dependent contribution p_s^s and the contribution of enduring contacts p_c^s , as

$$p^s = p_s^s + p_c^s, \quad (8)$$

where p_c^s is modeled as

$$p_c^s = \begin{cases} 0 & \alpha < \alpha_{rlp} \\ E \frac{(\alpha - \alpha_{rlp})^3}{(\alpha_{rcp} - \alpha)^5} & \alpha \geq \alpha_{rlp}, \end{cases} \quad (9)$$

where E is the elastic modulus (chosen to be $E = 0.1Pa$), α_{rcp} is the random close packing and α_{rlp} is the random loose packing. We adopt the value for sphere packings ($\alpha_{rcp} = 0.625$ and $\alpha_{rlp} = 0.56$).

Depending on the granular flow regime of the numerical simulation different expressions for the shear rate-dependent pressure can be used [47]. In the grain inertia regime, the inertial number ($I = d||\mathbf{S}^s||/\sqrt{p^s/\rho^s}$) is used to obtain the expression for the shear rate-dependent pressure given by

$$p_\infty^s = \rho^f d^2 ||\mathbf{S}^s||^2 \left(\frac{0.31\alpha}{\alpha_{max} - \alpha} \right)^2, \quad (10)$$

whereas in the viscous regime, the viscous number ($I_v = \rho^f \nu^f ||\mathbf{S}^s||/p^s$) is considered to get the shear rate-dependent pressure expression written as

$$p_\infty^s = \rho^f \nu^f ||\mathbf{S}^s|| \left(\frac{\alpha}{\alpha_{max} - \alpha} \right)^2. \quad (11)$$

However, as suggested by [48, 49], p_∞^s is consistently defined to be the stationary shear-induced pressure whereas the actual pressure is supposed to converge asymptotically to that value with accumulated strain, therefore, the equation that governs the space-time development of p_s^s is given by

$$\frac{\partial p_s^s}{\partial t} + \mathbf{u}^s \cdot \nabla p_s^s = -K_2(p_s^s - p_\infty^s) ||\mathbf{S}^s||, \quad (12)$$

where K_2 is a calibration parameter set to 1.

According to [50, 51], the ratio between shear stress and pressure can be scaled by the inertial (or viscous) number, as

$$\tau^s = \mu(I) p^s \frac{\mathbf{S}^s}{||\mathbf{S}^s||}, \quad (13)$$

where $\mu(I)$ is the friction coefficient for a certain shear state described in [44] as

$$\mu(I) = \mu_1 + \frac{\mu_2 - \mu_1}{I_o/I + 1}, \quad (14)$$

where the empirical material constants correspond to the static friction coefficient μ_1 , the saturation friction coefficient μ_2 , and the reference inertial (or viscous) number I_o . In this work, we assume $\mu_1 = 0.4$, $\mu_2 = 0.7$ and $I_o = 0.3$ for the inertial regime whereas the set $\mu_1 = 0.24$, $\mu_2 = 0.39$ and $I_o = 0.01$ is adopted for the viscous regime. In order to have an expression for τ^s resembling the definition for the fluid shear stress, the shear stress due to frictional contacts can be rewritten as

$$\tau^s = 2\rho^s \nu^s \mathbf{S}^s, \quad (15)$$

where ν^s is the frictional shear viscosity given by

$$\nu^s = \frac{\mu(I_v) p^s}{\rho^s (||\mathbf{S}^s||^2 + \lambda_r^2)^{1/2}}, \quad (16)$$

where λ_r is a regularization parameter from [52] taken equal to $\lambda_r = 10^{-6} \text{ s}^{-1}$.

2.3. 6DoF.

The position and orientation of a rigid body in space can be described using the six degrees of freedom (6DoF): three for translation and three for rotation. By determining the forces acting on the body, we can calculate its motion. OpenFOAM[®] incorporates this process by computing forces on a rigid body and solving for its motion, thus, mesh changes are required. The 6DoF motion module implemented in OpenFOAM[®] has been redesigned to take into account the frictional and particle stresses exerted on a solid object. The forces acting on a body can be categorized into three distinct types: fluid, particle, and external forces. Fluid forces refer to forces that the fluid imposes, such as the excess of pore pressure ($\mathbf{f}_f$), buoyancy ($\mathbf{f}_b$), and viscous ($\mathbf{f}_v$) forces. Particle forces involve frictional shear forces ($\mathbf{f}_{fric}$) and normal stresses ($\mathbf{f}_p$) arising from particle contacts. External forces, on the other hand, include forces like gravity ($\mathbf{f}_g$). These forces are given by

$$\mathbf{f}_f = \oint_S p^f dS, \quad (17)$$

$$\mathbf{f}_p = \oint_S p^s dS, \quad (18)$$

$$\mathbf{f}_\nu = \oint_S 2\rho^f \nu^{eff} \mathbf{S}^f \cdot \mathbf{n} dS, \quad (19)$$

$$\mathbf{f}_{fric} = \oint_S 2\rho^s \nu^s \mathbf{S}^s \cdot \mathbf{n} dS, \quad (20)$$

$$\mathbf{f}_b = - \int_V \rho^f \mathbf{g} dV, \quad (21)$$

$$\mathbf{f}_g = \int_V \rho^{obj} \mathbf{g} dV. \quad (22)$$

These forces are collectively denoted as $\mathbf{F}$, while the combined moments are referred to as $\mathbf{M}$. To determine the body's motion, we must calculate both the acceleration $\mathbf{a}$ and the angular acceleration γ . Accelerations are related to the moments and forces, as

$$\mathbf{F} = \mathbf{f}_f + \mathbf{f}_p + \mathbf{f}_\nu + \mathbf{f}_{fric} + \mathbf{f}_b + \mathbf{f}_g = m\mathbf{a}, \quad (23)$$

$$\mathbf{M} = \gamma \mathbf{I}, \quad (24)$$

where m and $\mathbf{I}$ are the mass and inertia of the object, respectively. When we integrate the translational acceleration over a specific time interval, we can obtain the current angular velocity $\omega_{t+\Delta t}$ and linear velocity $\mathbf{v}_{t+\Delta t}$ given by

$$\omega_{t+\Delta t} = \int_t^{t+\Delta t} \alpha dt = \omega_t + \gamma \Delta t, \quad (25)$$

$$\mathbf{v}_{t+\Delta t} = \int_t^{t+\Delta t} \mathbf{a} dt = \mathbf{v}_t + \mathbf{a} \Delta t. \quad (26)$$

Using $\omega_{t+\Delta t}$ and $\mathbf{v}_{t+\Delta t}$, we can calculate the rotation and the distance covered since the previous time interval.

3. Implementation

The PIMPLE algorithm, which combines the PISO with the SIMPLE algorithm, is used for `overSed-DyMFoam`. The numerical steps required for the PIMPLE algorithm are outlined in Fig. 1. Initially, the mesh, configuration, and properties of fluid and solid phase are read and initialized. In each time step, the mesh is modified based on the solution of the 6DoF, followed by the resolution of the sediment transport equation and the closure forms for the sediment and fluid stresses. The next step includes solving the pressure equation. The final stage of the PIMPLE algorithm incorporates the turbulent models. More details on the turbulent models can be found in [16], the publication associated with the initial release of `sedFoam`. While various turbulent models are available in `sedFoam` and can influence the stresses acting on a given solid object, this article primarily focus on integrating granular stresses within the 6DoF framework and emphasize the impact of the granular phase on solid bodies or structures, thus, in this benchmark we focus on simple cases under the laminar regime. Finally, this loop is repeated `nOuterCorrectors` times.

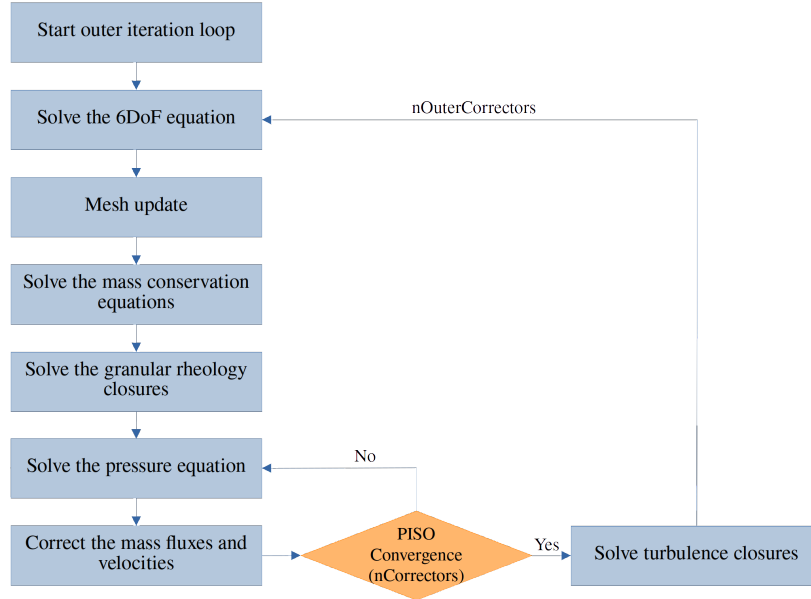


Figure 1. Flowchart of the overSedDyMFoam solver.

As mentioned before, overSedDyMFoam is an extension of the solver sedFoam introduced by [16]. We encourage the reader to have a look at the code and documentation of sedFoam available for download from ¹GitHub. The scope of the present article is not to discuss the closure forms and algorithms used in sedFoam, instead, we will focus on the implementation of libraries and code modifications that allow overSedDyMFoam to deal with moving objects. Before jumping into the details, it is worth presenting the two types of dynamic meshes considered for this work:

- **Overset mesh:** The basic principle of overset method is to construct two disconnected meshes, the background mesh, which remains static, and the overset mesh, which displaces relative to the background mesh (see Fig. 2a). The overset method is suitable for objects with a large range of motion. Essentially, it tackles the governing equations independently on both meshes. Within the background grid, elements that correspond to the interior of the overset domain are labeled as *holes* and excluded from calculations. Instead, boundary values of the overset mesh are *interpolated* into the background mesh. The same process is done for the overset domain. Consequently, at each time step, cells in both the background and overset meshes are categorized as *calculated* (solved through equations), *interpolated* (values are computed by interpolation from nearby elements in the opposite domain), or *holes* (computation is excluded from these cells). Fig. 2 illustrates in gray the *calculated* cells while pale green is denoting the *hole* region. Each mesh includes *interpolated* cells: in Fig. 2b we observe that those neighboring the *hole* of the overset mesh belong to the background mesh and their donors are from overset mesh. While the *interpolated* cells belonging to the overset mesh are found at the boundary of the overset mesh and their donors are from the background mesh as shown in Fig. 2c. These meshes are independent and the coupling is achieved through interpolation. Various interpolation methods are available in OpenFOAM[®] (`cellVolumeWeight`, `inverseDistance`, `leastSquares`, and `trackingInverseDistance`). In this work, we will adopt the `inverseDistance` approach. It is worth noting that implementation of the overset framework in OpenFOAM[®] and a comprehensive explanation of the `inverseDistance` method can be found in [53].
- **Morphing mesh:** This type of mesh can deform but the connectivity between the cells remains unchanged. The mesh deformation is performed by applying a different displacement to each mesh point. Mesh deformation will introduce relative motion between mesh points, which will in turn distort mesh cells and modify the mesh in an inhomogeneous way. This usually leads to high distortions, especially for large motions, which reduces the mesh quality.

The new file `overSedDyMFoam.C`, located at `$/solvers/overSedDyMFoam/overSedDyMFoam.C`, includes several functions and classes that are worth highlighting before breaking down the details of the code:

¹<https://github.com/SedFoam/SedFoam>

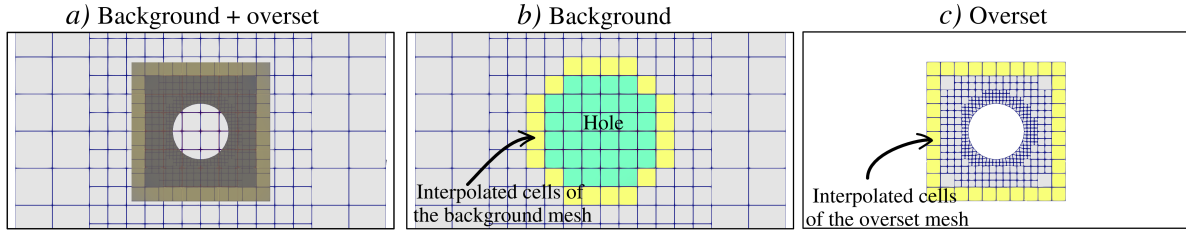


Figure 2. a) Background and overset mesh overlapping, b) *hole* cells in pale green, *calculated* cells in gray and *interpolated* cells in light yellow defined for the background mesh, and c) *calculated* cells in gray and *interpolated* cells in light yellow defined for the overset mesh.

- The `mesh.update()` calls the mesh motion library to calculate the new position of points and updates the mesh. The mesh update may need the quantities or dependent variables to be remapped on the new mesh or the mesh created after moving the node points of the mesh.
- `correctPhiSedFaceMask.H`, found at `$/solvers/overSedDymFoam/correctPhiSedFaceMask.H`, updates the surface velocity field and fluxes:

```

1 // Zero Uf on old faceMask (Hole-Interpolated faces)
2 Ufa *= faceMaskOld;
3 Ufb *= faceMaskOld;
4 // Update Uf and phi on new Calculated-Interpolated faces
5 Ufa += (1-faceMaskOld)*fvc::interpolate(Ua);
6 Ufb += (1-faceMaskOld)*fvc::interpolate(Ub);
7 phia = mesh.Sf() & Ufa;
8 phib = mesh.Sf() & Ufb;
9 phia *= faceMask;
10 phib *= faceMask;

```

here, a mask for the faces (`faceMask`) is created by interpolating `cellMask`. `cellMask` is a field that assumes 0 for *hole* cells and 1 elsewhere (see Fig. 2). Likewise, `faceMaskOld` is generated from the `faceMask` corresponding to the previous time step. Both `faceMaskOld` and `faceMask` are used to set zero velocity at the faces of *hole* cells and update the surface velocities at the *calculated-interpolated* faces. `interpolate` function is used to transform cell-center quantities, such as the particle $\mathbf{u}^s$ (Ua in the code) and fluid velocities $\mathbf{u}^f$ (Ub in the code), to face centers quantities (Ufa and Ufb) respectively. It is worth mentioning that `correctPhiSedFaceMask.H` is only used for the overset mesh approach.

- `fvc::makeAbsolute(phia, Ua)` and `fvc::makeAbsolute(phib, Ub)`. This function adds the flux caused by the movement of the mesh to the flux across the mesh control volume boundaries. This gives the absolute flux relative to a fixed reference frame.
- `fvc::makeRelative(phia, Ua)` and `fvc::makeRelative(phib, Ub)` routines perform the opposite operation of `fvc::makeAbsolute()`. Hence, this function subtracts the flux of the mesh at every cell face, leaving the flux at the cell faces that is relative to the moving reference frame.
- `correctPhiSed.H` is found at `$/solvers/overSedDymFoam/correctPhiSed.H` and it is called when `correctPhi` is set to `True` in the `fvSolution` dictionary. If the mesh is moving, the fields must be mapped from the old to the new mesh, which may introduce interpolation errors. Therefore, `correctPhiSed.H` is included to mitigate the errors and satisfy the mass conservation. In the `correctPhiSed.H` file, the PIMPLE algorithm's pressure corrector equation is solved a specified number of times indicated by the variable `nNonOrthogonalCorrectors`, defined in `fvSolution`. The objective of this process is to derive a pressure corrector, which is subsequently utilized to ensure that the fluxes across the mesh cells satisfy the principle of continuity. This correction is carried out before the main PIMPLE loop. `correctPhiSed.H` is only used for the overset mesh approach.

All the previous functions are crucial to handle dynamic meshes. Although we are not going to describe line by line `overSedDymFoam.C`'s file, it is worth highlighting the part where fluxes are recalculated after mesh changes:

```

1  bool changed = mesh.update();
2  if (changed)
3  {
4      Info << "MESH CHANGED" << endl;
5      #include "setCellMask.H"
6      #include "setInterpolatedCells.H"
7      #include "correctPhiSedFaceMask.H"
8      gh = (g & mesh.C()) - ghRef;
9      ghf = (g & mesh.Cf()) - ghRef;
10
11     fvc::makeRelative(phia, Ua);
12     fvc::makeRelative(phib, Ub);
13
14     surfaceScalarField alphaf = fvc::interpolate(alpha);
15     surfaceScalarField betaf = scalar(1.0) - alphaf;
16     phi = alphaf*phia + betaf*phib;
17
18     // Calculate absolute flux from the mapped surface velocity
19     if (correctPhi)
20     {
21         #include "correctPhiSed.H"
22     }
23 }

```

First of all, `mesh.update()` function is called. If the mesh is moving, the scalar field `cellMask` is applied to the cells. Zero values correspond to holes and 1 elsewhere. Then, `interpolatedCells` assigns 0 values to the interpolated cells, otherwise cells are set to 1. Right after, `correctPhiSedFaceMask.H` is called in order to correct the fluxes in the solver using the new velocity values. `fvc::makeRelative()` is used to make the sediment Φ^a (`phia` in the code) and fluid Φ^b (`phib` in the code) fluxes relative to the mesh motion. Then, the mixture flux (`phi`) is computed before the flux correction function (`correctPhiSed.H`) is called to ensure continuity. Here, the function `interpolate` is called to transform the solid volume fraction field `alpha` (stored as cell-center values) to `alphaf` (stored as face-center values).

Regarding the resolution of the mass conservation (Eqn. 1), terms are rewritten as function of the relative velocity between the sediment and the fluid phase to obtain an hyperbolic equation with higher numerical robustness. Thus, Eqn. 1 becomes

$$\frac{\partial}{\partial t} \int_V \alpha dV + \int_S \alpha (\mathbf{u}^m - \mathbf{u}_g) \cdot \mathbf{n} dS - \int_S -\mathbf{u}^r \alpha (1 - \alpha) \cdot \mathbf{n} dS = 0, \quad (27)$$

where $\mathbf{u}^m$ is the mixture velocity that can be determined as

$$\mathbf{u}^m = \alpha \mathbf{u}^s + (1 - \alpha) \mathbf{u}^f, \quad (28)$$

and the relative velocity $\mathbf{u}^r$ is defined as the difference between the sediment and fluid phase, as

$$\mathbf{u}^r = \mathbf{u}^s - \mathbf{u}^f. \quad (29)$$

It is worth noticing that the first convective term in Eqn. 27 is written relative to the mesh velocity. Thus, the relative motion flux for the mixture velocity is considered as

$$\Phi = (\mathbf{u}^m \cdot \mathbf{n}) dS - (\mathbf{u}_g \cdot \mathbf{n}) dS. \quad (30)$$

On the contrary, the second convective term in Eqn. 27 is the absolute flux corresponding to relative velocity between the two phases given by

$$\Phi^r = \Phi^a - \Phi^b, \quad (31)$$

where Φ^a is the solid phase flux computed as $\Phi^a = (\mathbf{u}^s \cdot \mathbf{n}) dS$ and the Φ^b is the fluid phase flux computed as $\Phi^b = (\mathbf{u}^f \cdot \mathbf{n}) dS$.

The resolution of the mass conservation equation is performed in `alphaEqn.H`, located at `$/solver-s/overSedDymFoam/alphaEqn.H`, and is written as follows:

```

1 word scheme("div(phi, alpha)");
2 word schemer("div(phir, alpha)");
3
4 surfaceScalarField phic("phic", phi); // phi is relative
5 fvc::makeAbsolute(phia, Ua);
6 fvc::makeAbsolute(phib, Ub);
7 surfaceScalarField phir("phir", phia - phib);
8
9 for (int acorr=0; acorr<nAlphaCorr; acorr++)
10 {
11     fvScalarMatrix alphaEqn
12     (
13         fvm::ddt(alpha)
14         + fvm::div(phic, alpha, scheme)
15         + fvm::div(-fvc::flux(-phir, (1.0-alpha), schemer), alpha, schemer)
16     );
17
18     alphaEqn.relax();
19     alphaEqn.solve();
20
21     alpha.min(alphaMax);
22     alpha.max(0);
23
24     alpha *= cellMask;
25     alpha.correctBoundaryConditions();
26
27     beta = (scalar(1.0) - alpha);
28 }
29
30 rho = alpha*rhoa + beta*rhob;

```

The next step of the PIMPLE loop consists in solving the pressure equation and correcting the velocities. Files `UEqn.H` and `pEqn.H`, found at `$/solvers/overSedDymFoam/pU/`, contain the part of the code for the momentum and the pressure equation, respectively. The formulation of `UEqn.H` and `pEqn.H` for the `overSedDymFoam` solver is almost the same as for a fixed grid. The main changes are listed below:

- The function `interpolatedFaces.H` is called if `massFluxInterpolation` found in `fvSolution` is set to `yes`. This function contributes to correct the mass flux imbalance produced by the overset interpolation. `interpolatedFaces.H` is only used for the overset mesh approach.
- The resolved velocity field is multiplied by the field `cellMask` so the velocity is set to zero if there is *hole* present. This only applies for the overset mesh approach.

3.1. 6DoF implementation.

In OpenFOAM®, forces on a rigid object need to be extracted so the velocity and the displacement for a time step can be calculated. In particular, `$/FOAM_SRC/functionObjects/forces/forces/-forces.C` function is used to compute the forces and moments by integrating the pressure and viscous forces over a given list of patches. In order to adapt the 6DoF to account for the particle stresses exerted on a rigid body, `forces.C` file is renamed as `forcesSed.C` and modified as:

```

1 vectorField fN(rho(p)*Sfb[patchi]*(p_rbg.boundaryField()[patchi]- pRef));
2
3 vectorField fNsolid(rho(p)*Sfb[patchi]*(pS.boundaryField()[patchi] - pRef)
4
5 vectorField fT(Sfb[patchi] & devRhoReffb[patchi]);

```

where

```

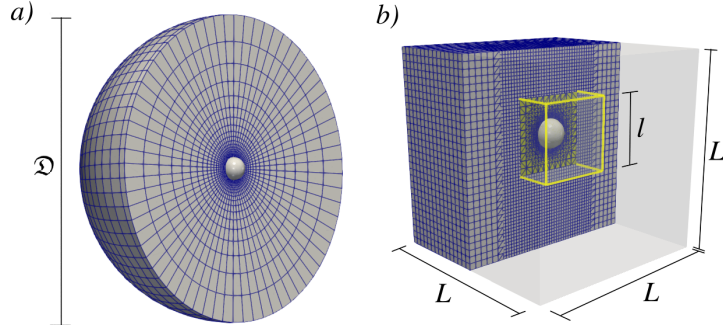
1
2 devRhoReff = -muEff*dev(twoSymm(fvc::grad(Ub)))-muFra*dev(twoSymm(fvc::grad(Ua)));
3 const volSymmTensorField::Boundary& devRhoReffb = devRhoReff().boundaryField();

```

so the following contributions are computed for the chosen patches:

Table 1. Geometrical and physical properties used to reproduce the falling sphere experiment [54].

Parameter	Value
ρ^{obj}	$1120 kg/m^3$
ρ^f	$970 kg/m^3$
ν^f	$3.845 \cdot 10^{-4} m^2/s$
Sphere diameter (D)	$0.015m$
Domain diameter for the morphing mesh ($\mathfrak{D}$)	$6.66D$
Box dimensions for the overset mesh ($l \times l \times l$)	$2.5D \times 2.5D \times 2.5D$
Background mesh dimensions of the overset approach ($L \times L \times L$)	$6.66D \times 6.66D \times 6.66D$

**Figure 3.** Geometrical domain for the *a*) morphing and *b*) overset meshes.

- the excess of fluid pressure p^f (p_rbg in the code)
- the particle pressure p^s (pS in the code)
- the fluid viscous contribution $\boldsymbol{\tau}^f$ ($-\mu_{\text{Eff}} \cdot \text{dev}(\text{twoSymm}(\text{fvc}::\text{grad}(\text{Ub})))$ in the code)
- the frictional contribution $\boldsymbol{\tau}^s$ ($-\mu_{\text{Fra}} \cdot \text{dev}(\text{twoSymm}(\text{fvc}::\text{grad}(\text{Ua})))$ in the code)

Similarly, a new library for the motion solver is created. In our case, we name it `$FOAM_SRC/sixDoFRigidBodyMotion` to `sixDoFRigidBodyMotionSedFoam`. The only difference between the original and the new library lies on the list of forces that are called to compute the mesh motion. Thus, `sixDoFRigidBodyMotionSedFoam` uses the forces retrieved from `forcesSed.C`. It is important noting that for a numerical simulation we need to specify the motion solver (`sixDoFRigidBodyMotion`) and import the library for the motion solver (`sixDoFRigidBodyMotionSedFoam`). This piece of information must be included in the `dynamicMeshDict` dictionary such as:

```

1 motionSolverLibs (sixDoFRigidBodyMotionSedFoam);
2
3 motionSolver sixDoFRigidBodyMotion;
```

4. 6DoF numerical cases

The scope of this section is to validate the modified 6DoF solver that includes the contribution of sediment forces on rigid objects. Although the `overSedDymFoam` intends to be a solver capable to reproduce complex fluid-sediment-structure interactions such as deep burial of objects in the seabed, the motion of intruders under bedload transport or the self-burial process of rigid objects under wave-induced conditions, this section starts presenting a very simple case and more ingredients are added throughout the article to illustrate the potential capabilities of `overSedDymFoam`. The starting point of this section is the free fall of a sphere immersed in a pure Newtonian fluid. Then, the same configuration is considered except that the sphere is submerged in a fluid with particles in suspension. The next step consists of placing a compacted sediment bed under the sphere to bring the object to a halt due to the particle-particle interactions. The last part of this section is restricted to a 2D configuration to reduce the computational cost. In particular, the trajectory of a cylinder subjected to a uniform granular flow is studied. These numerical cases and the scripts for post-processing can be found in `$/tutorialsDyM`.

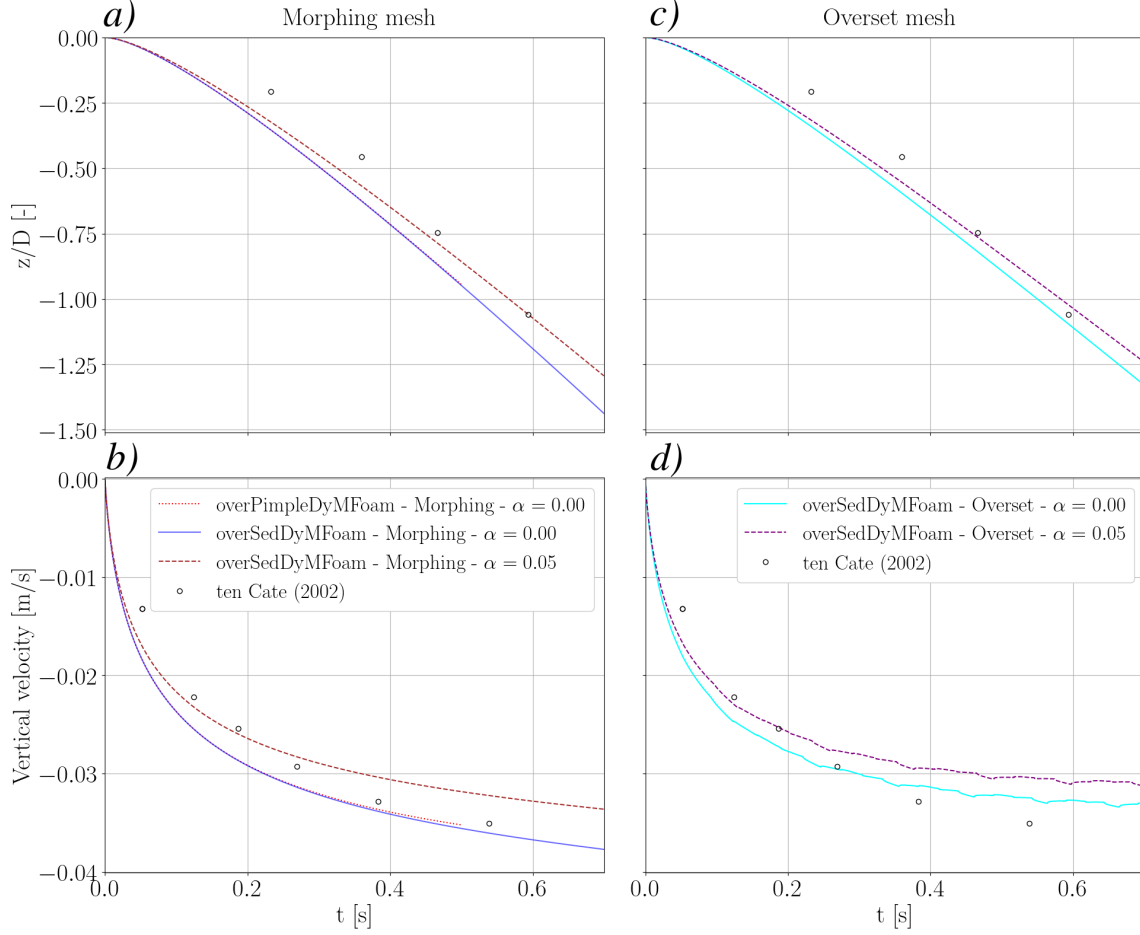


Figure 4. Sphere *a)* trajectory and *b)* velocity evolution using different solvers and concentrations using a morphing mesh. Sphere *c)* trajectory and *d)* velocity evolution using different concentrations using an overset mesh.

4.1. Falling sphere in pure fluid.

Our first numerical simulation to verify the implementation of the 6DoF consists of a free fall of a sphere with a diameter $D = 15\text{mm}$ settling in silicon oil. The experimental setting consists in a rectangular box of $100 \times 100 \times 160\text{mm}$. In the numerical domain we adopt two different meshes depending on the dynamic mesh approach. For the morphing mesh, a spherical mesh is considered whereas a box mesh is assumed for the overset method. The input parameters have been chosen to reproduce the experiments of a settling sphere at Reynolds number equal to $Re = W^{obj} D / \nu^f = 1.5$ [54], where W^{obj} is the terminal velocity of the sphere. The geometrical and physical parameters are summarized in Tab. 1 and the numerical set-up is displayed in Fig. 3. The numerical cases are located at `$/tutorialsDyM/FallingSphereMorphing` and `$/tutorialsDyM/FallingSphereOverset`. This benchmark is inspired by the ²online tutorial of a settling sphere of OpenFOAM®.

In Fig. 4 we observe the temporal evolution of the sphere velocity which tends towards its terminal settling value. The evolution of the vertical position and velocity using `overSedDyMFoam` is in very good agreement with the experimental measurements [54]. Additionally, a simulation using `overPimpleDyMFoam` solver from the ESI version of OpenFOAM® is also included in Fig. 4. The lack of discrepancies between `overPimpleDyMFoam` and `overSedDyMFoam`, in addition to the agreement with the experimental data, support the correct implementation of the 6DoF capability in `overSedDyMFoam`. Fig. 4 shows mild differences between the overset and the morphing mesh results suggesting that the overset approach needs to be further refined or/and improve the interpolation method to smooth the path displayed in Fig. 4. On the other hand, the morphing mesh is not able to predict a constant terminal

²https://wiki.openfoam.com/Settling_Sphere_by_Michael_Alletto

Table 2. Geometrical and physical properties used to reproduce the impact of a spherical intruder on a granular bed.

Parameter	Value
ρ^s	$2650kg/m^3$
ρ^f	$1000kg/m^3$
ρ^{obj}	$1141kg/m^3$
ν^f	$1 \cdot 10^{-6}m^2/s$
Grain diameter (d)	$10mm$
Sphere diameter (D)	$0.015m$
Domain diameter for the morphing mesh ($\mathfrak{D}$)	$13.3D$
Granular flow regime	Viscous

velocity because large displacements compromise the accuracy of the results. It is worth mentioning that shifting the curves a certain amount of time leads to a better fitting with the experimental data, these results and discussion are found in appendix B.1. Furthermore, a mesh convergence analysis is also found in appendix B.1.

4.2. Falling sphere in a fluid with a sediment suspension.

In this scenario we want to extend the results by reproducing, numerically, the falling sphere in a fluid with the presence of sediments in suspension. For this numerical set-up, the content of sediments is set to 5%, the particles are neutrally buoyant (i.e. same density as the fluid) with a mean diameter of $d = 0.29mm$ and the Gidaspow-Schiller-Naumann drag model is considered. The numerical cases are located at `$/tutorialsDyM/FallingSphereSuspensionMorphing` and `$/tutorialsDyM/FallingSphereSuspensionOverset`. For a sphere suspended in a mixture of solid particles, the settling velocity is lower than its terminal velocity in pure fluid [55, 56]. This is due to the relative position of particles and their hydrodynamic interactions that create a hindering effect. This effect can be simulated by increasing the effective viscosity. Indeed, according to Eqn. 7, the effective viscosity is proportional to the solid volume fraction, therefore, the sphere falls with lower acceleration and reaches a lower terminal velocity as shown in Fig. 4. Similar to the case of the falling sphere in a pure fluid, the trajectory curves for the overset and morphing mesh follow slightly different paths. Nonetheless, the gap between the $\alpha = 0$ and $\alpha = 0.05$ trajectory curves remains similar regardless of the dynamic mesh approach.

4.3. Resting sphere.

This section tests the ability of `overSedDymFoam` to capture the particle pressure and frictional stresses on a rigid body. Granular material at equilibrium acts as a solid-like state. However, according to [57], the impact of a solid object may cause the soil to behave like a fluid. Unlike impacting a liquid, the object must overcome a minimum yield stress to penetrate the granular bed. The value of this yield stress depends on how the material is packed. Granular materials are also able to dissipate energy effectively through grain-grain interactions, a process slightly different from the way liquids dissipate energy through viscosity. Numerous studies [57–59] have investigated the underlying physics involved in objects penetrating granular beds. Certain studies [60–62] have examined the upward drag force that an object experiences while it penetrates a granular bed. Whereas other works [63–65] geared the effort to gain insight into force-chains networks that are originated during the impact of the object. Indeed, the intensity, orientation, evolution and other features of particle-particle and particle-object interactions inside the granular bed are crucial to understand the behavior of dense granular systems that undergo jamming, shear banding or cratering [58, 66, 67]. Although previous works provide fundamental explanations to understand the microscopic processes responsible for dissipating kinetic energy, and the dynamics of a rigid object penetrating a granular bed, the scope of the present article is limited to illustrate the ability of the model to numerically reproduce the impact and arrest of a sphere falling on a granular bed.

In this simulation, only the morphing mesh is considered. The computational domain is assumed to be a spherical domain having a diameter $\mathfrak{D} = 13.3D$. Ergun drag model is adopted to model the permeability. As shown in Fig. 6a, a horizontal sediment bed is included in the numerical simulation.

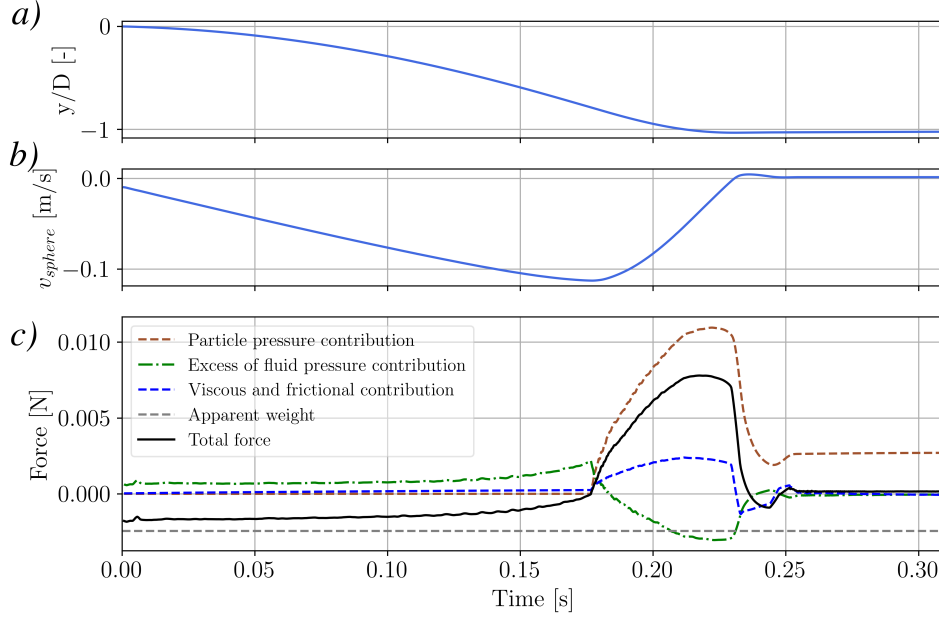


Figure 5. Sphere a) trajectory and b) velocity obtained with the morphing mesh approach. c) Plot of the total forces and each contribution exerted on the sphere from the morphing mesh approach.

The bed interface is located at a distance of $0.84D$ from the lowest point of the sphere. The rest of the physical and geometrical parameters are summarized in Tab. 2. The numerical case is located at `$/tutorialsDyM/RestingSphereMorphing` and a mesh convergence test is presented in appendix B.3.

From the starting point ($t = 0$ s), the object speeds up and the fluid pressure and viscosity forces act against gravity (see fluid pressure and viscous contribution build up in Fig. 5c). At $t \approx 0.19$ s the spherical intruder touches the sediment bed, thus, contact and frictional forces start developing to counterbalance the weight of the intruder slowing down the velocity of the intruder (see the vertical velocity reduction in Fig. 5b and the changes in the velocity field of Fig. 6b). The particle and frictional stresses fields illustrated in Fig. 6c and Fig. 6d at $t = 0.22$ s evidence the fact that the particle forces developing beneath the intruder are strong enough to bring the spherical object to arrest. Although no data has been used to validate this numerical case, it is worth mentioning that a similar behavior has been reported in experimental and numerical approaches with similar configurations in terms of networks of contact forces [61, 65, 67]. The object stops when the particle forces are in equilibrium with the gravitational force. Indeed, as shown in Fig. 5b and Fig. 5c, after $t > 0.26$ s, the net force and the intruder velocity are zero, thus, the intruder remains static and partially buried as shown in the last row ($t = 0.31$ s) of Fig. 6.

4.4. Uniform granular flow around a cylinder.

Previous numerical cases focused on the hydrodynamic forces and frictional/particle forces for sediments at rest. In this section we evaluate the response of an object subjected to a granular flow. This configuration is particularly important for studying grain segregation. When a bidisperse granular material is submitted to a shear flow, it often exhibits segregation phenomena [68, 69]: large and small particles have a tendency to migrate in different regions. This tendency to segregate is a major issue in many industrial applications involving mixing processes, and in many geophysical flows such as deposits of rock avalanches, landslides or grain size sorting in sediment transport. Another challenging situation is the burial of large objects in the seafloor such as unexploded munitions which are a threat for human beings, coastal wildlife and flora. An archetypal situation to understand this problem is the dense granular flow of small particles around a large solid body such as a cylinder or a sphere. Lift and drag forces are generated on rigid bodies immersed in moving granular media. However, these forces are not as well understood as in Newtonian fluids. Drag force induced on objects in granular media has been studied largely. On the contrary, little progress has been made to understand the lift force. Both [70] and [71]

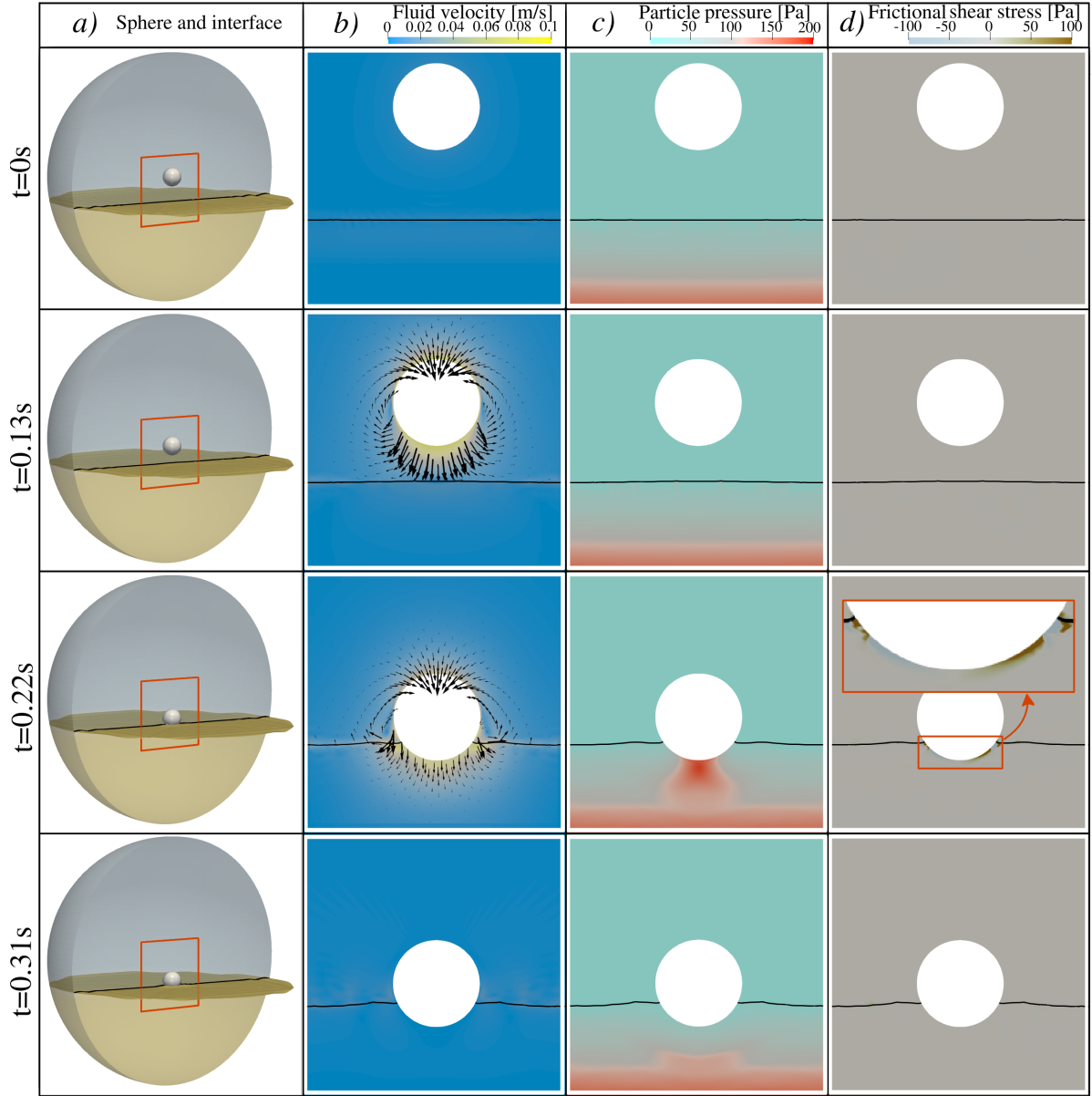


Figure 6. *a)* Sphere position and interface. The orange rectangle indicates the detailed area of the *b)* fluid velocity field, *c)* particle pressure field and *d)* frictional shear stress field around the sphere displayed for $t = 0.0s$, $t = 0.13s$, $t = 0.22s$ and $t = 0.31s$. The black isoline indicates the interface at $\alpha = 0.3$.

reported that symmetrical objects, such as a cylinder, may experience lift. Moreover, the authors pointed out the pressure gradient induced by gravity could be the source of this asymmetrical force.

In this subsection we analyze the stresses and the trajectory of an intruder immersed in a uniform granular flow. In order to reduce the computational cost, a cylinder (a 2D object in OpenFOAM®) is used instead of the spherical intruder employed in previous numerical cases. Inspired by the work of [71], where lift and drag forces on a cylinder were measured under a granular flow, we adopt a similar set-up as depicted in Fig. 7. More specifically, a granular packing is considered, then, the bottom plate is set to a prescribed constant velocity that induces a uniform granular flow above the bottom plate. A cylinder is placed in the granular medium distorting the flow patterns. As illustrated in Fig. 7, a constant horizontal velocity of $u_o = 0.01m/s$ is prescribed at the bottom to generate the uniform granular flow. The initial velocity field is set to $u_o = 0.01m/s$ as well. DEM simulations performed using the open-source code YADE [72] are also presented to test the accuracy of overSedDymFoam. In YADE, the solid particles are spherical and monodisperse. The interactions between them are elastic-plastic, with normal and tangential stiffness k_n and k_s , and Coulomb friction coefficient μ_s . Newton's second law of motion is integrated explicitly through iterative time stepping (implementation details can be found in [72]). The

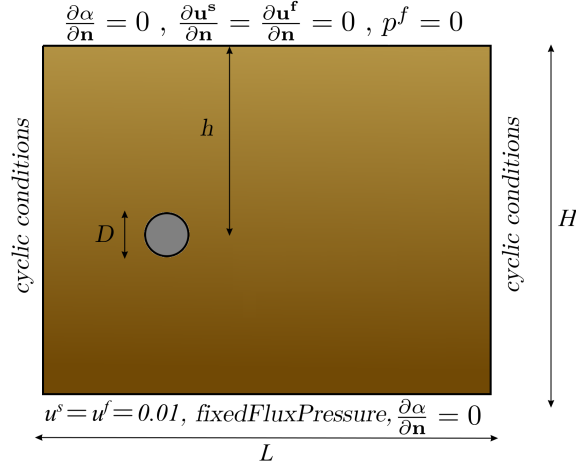


Figure 7. Numerical set-up to model the segregation of a large intruder immersed in a granular flow.

Table 3. Geometrical and physical properties used to reproduce the granular flow around a cylinder with `sedFoam` (and `overSedDymFoam`) and `YADE`.

Parameter	<code>sedFoam</code>	<code>YADE</code>
ρ^s	2500 kg/m^3	2500 kg/m^3
ρ^f	50 kg/m^3	-
ν^f	$1.5 \cdot 10^{-5} \text{ m}^2/\text{s}$	-
Grain diameter (d)	0.0015 m	0.0015 m
Cylinder diameter (D)	$3.3d$	$3.3d$
Cylinder depth (h)	$5.8D$	$5.8D$
Domain depth (H)	$39d$	$39d$
Domain length (L)	$85d$	$100d$
Elastic modulus (E)	0.1 Pa	50000 Pa
Poisson ratio (ν)	-	0.4
Friction coefficient (μ_1)	0.4	0.4
Drag model (K)	Ergun	-
Granular flow regime	Inertial	-

parameters that have been employed are summarized in Tab. 3. It is worth noting that DEM simulations assume a dry granular material, however, the density ratio using `overSedDymFoam` is slightly different to avoid numerical errors. Plus, in `YADE` simulations, a rigid horizontal layer of cohesive particles is placed at the top to avoid deformations above the granular packing induced by the intruder. The numerical case is located at `$/tutorialsDyM/2DCylinderUniformGranularFlow`.

Before testing the capability of dynamic meshes, a first numerical comparison with a static cylinder is presented to compare the stress and velocity fields between DEM and `sedFoam` simulations. In this subsection we are interested in visualizing how the granular flow (uniform far from the object) is modified by the presence of a static cylinder and what is the stress distribution around the cylinder that induce both drag and lift forces. Fig. 8 illustrates the velocity and stress fields around the cylinder obtained with `sedFoam` and `YADE`. The words *Slip* and *No slip* in Fig. 8 refer to the boundary conditions applied to the solid velocity at the cylinder surface. *No slip* condition assumes $\mathbf{u}^s = \mathbf{0}$ meaning that the particles on the cylinder are sticking to the wall and have zero relative velocity at the interface. The *Slip* condition, on the other hand, assumes sliding particles, allowing for a non-zero tangential velocity at the interface. The choice between *Slip* and *No slip* conditions depends on the physical characteristics of the problem. In most cases, the *No slip* condition is appropriate for simulating pure fluid flows over solid surfaces. However, in granular flow applications, particles may roll or be dragged on a solid boundary. Therefore, we explore the consequences of these two conditions for the solid velocity applied on the surface of the

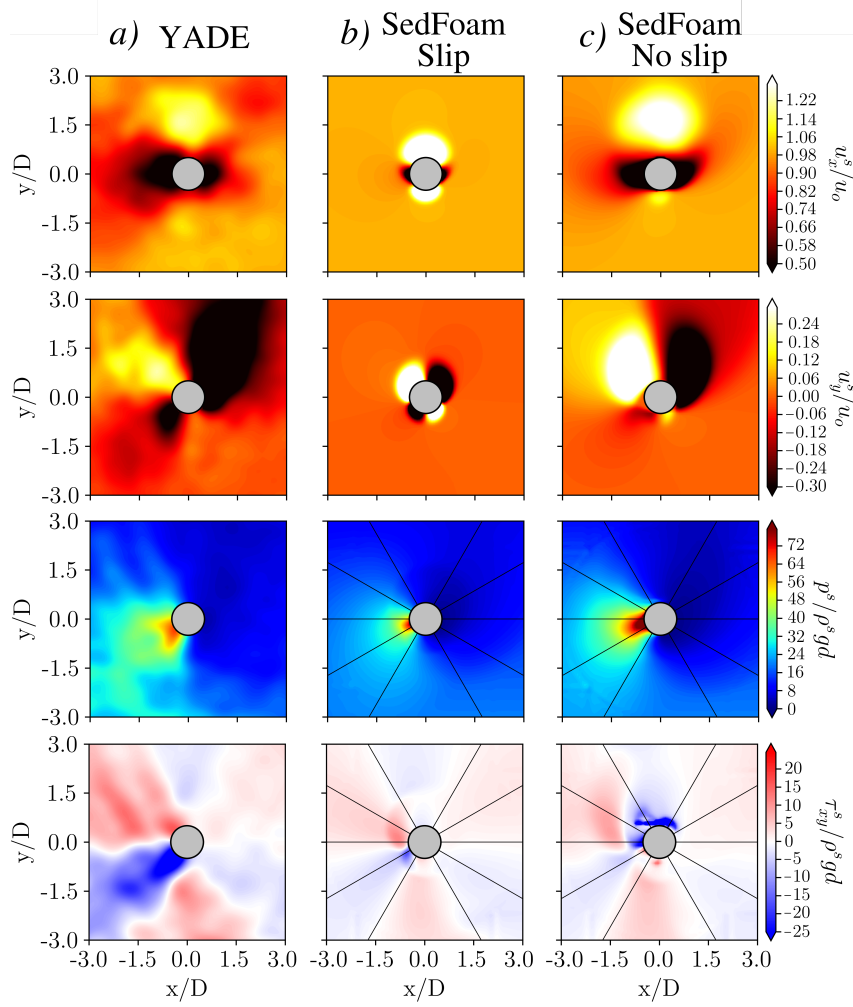


Figure 8. Sediment velocity, particle pressure and frictional shear stress fields around the cylinder modeled with a) YADE, b) *sedFoam* with *Slip* condition around the cylinder and c) *sedFoam* with *No slip* condition around the cylinder.

cylinder. In order to display the averaged fields resulting from the discrete YADE simulations in a structured grid, each quantity Ψ has been spatially averaged at point i as

$$\bar{\Psi}_i = \sum_{j=1}^N e^{-|r_j/d|^2} \cdot \Psi_j, \quad (32)$$

where r_j is the distance between the element j and the grid point i . N is the number of total discrete elements. Then, $\bar{\Psi}_i$ is averaged temporally as

$$\hat{\Psi}_i = \frac{1}{T} \sum_{t=1}^T \bar{\Psi}_i. \quad (33)$$

Fig. 8 evidences a high pressure region in front of the object and a low pressure region at the downstream side. The distribution of pressure is not completely symmetrical with respect to the horizontal line. Indeed, Fig. 8 shows the high pressure area is slightly shifted downward and the low pressure is localized slightly upward. According to [71], this asymmetry is the origin of the lift force on the cylinder. The frictional shear stresses in Fig. 8 evidence the sheared regions are oriented around 30 degrees in good agreement with [71] results. Additionally, the asymmetric patterns of the frictional shear stress are also manifested in Fig. 8: the shear stress regions located upstream are much higher than downstream. The radial stress distribution around the cylinder is illustrated in Fig. 9 where the normal stress (σ_N^s) becomes higher in the bottom front quarter (peak between 20° and 30°) and lower at the upper back part

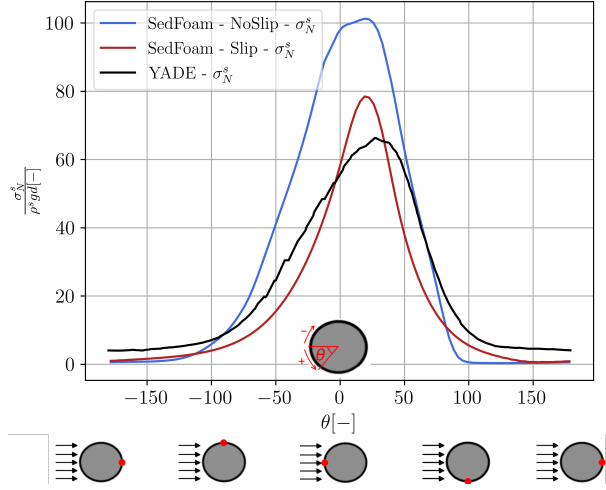


Figure 9. Radial distribution of normal stresses on the surface of the cylinder as a function of the angle θ .

of the cylinder ($-180^\circ < \theta < -90^\circ$ in Fig. 9). Both Fig. 8 and Fig. 9 suggest that qualitative agreement is found between [71], YADE and *sedFoam* results.

The presence of a pressure gradient significantly affects the movement of grains around the cylinder as depicted in the first two rows of Fig. 8. According to [71], in the absence of a pressure gradient, when the grains would reach the front of the cylinder, they would shift either upwards or downwards, accelerate above and below the cylinder, and eventually return to the cylinder's wake downstream exhibiting symmetry in the upward and downward directions. However, in the presence of a pressure gradient, the displacement of grains is significantly disrupted, with grains preferring to pass above the cylinder, where the pressure is lower, rather than below it (see the vertical velocities being lower below the cylinder compared to above it in the second row of Fig. 8). Consequently, the presence of a pressure gradient not only alters the local pressure but also appears to have a significant effect on the flow of grains. Overall, Fig. 8 show reasonable agreement between the DEM and *sedFoam* results. Fig. 8 shows that the choice of *Slip/No-slip* has an important effect on *sedFoam* results. According to Fig. 8, the *No slip* condition on the cylinder provides a better description of the velocity field, however the solid pressure is overpredicted in front of the cylinder. Indeed, Fig. 9 shows that the DEM normal stress peak ($\frac{\sigma_N^s}{\rho^s g d} = 66$) is better modeled with the *Slip* condition with a normal stress peak of $\frac{\sigma_N^s}{\rho^s g d} = 78$. Regarding the asymmetry of the normal stresses (σ_N^s) around the cylinder, Fig. 9 shows that *sedFoam* peaks are shifted around 20° compared to the 30° shift of the YADE peak.

The frictional shear stress field displayed in Fig. 8c suggests that the *No slip* condition is probably not appropriate to study this problem as it is remarkably different from the YADE solution. On the contrary, the *Slip* condition, where particles can roll and move around the cylindrical boundary, provides a similar distribution of the shear stress field (see Fig. 8b) but with lower magnitude than the YADE results. Previous discrepancies, such as higher shear stress values in the DEM, the degree of asymmetry or the minor differences in flow patterns, could be attributed to the inherent averaging process of DEM simulations, the choice of the particle velocity boundary condition of the cylinder surface, and the rheological model adopted within *sedFoam*. The sensitivity of the continuum model results to the choice of the boundary condition for the granular phase (*Slip/No slip*) confirms that more work needs to be done to accurately predict granular flows in this configuration. Beyond the boundary condition, the rheological model also needs improvements in order to better predict the granular flow around a static cylinder. This question is beyond the scope of the present study, our main goal here is to verify the qualitative behavior of the FPSI model in the two extreme situations when the fluid-particle forces acting on the object are dominated by the fluid forces or by the granular forces.

At this point we can remove the motion constraint of the object and track the vertical trajectory of the cylinder induced by the lift force generated from the asymmetrical particle stress field discussed previously. Fig. 10 shows the evolution of the large intruder during the *overSedDymFoam* and YADE simulations. Four series of samples prepared with YADE are plotted in Fig. 10a to take into account the stochastic nature due to local grain and/or anisotropic effects. An average YADE curve of the four series is also included in Fig. 10a. The main difference between YADE and *overSedDymFoam* is the fluctuating

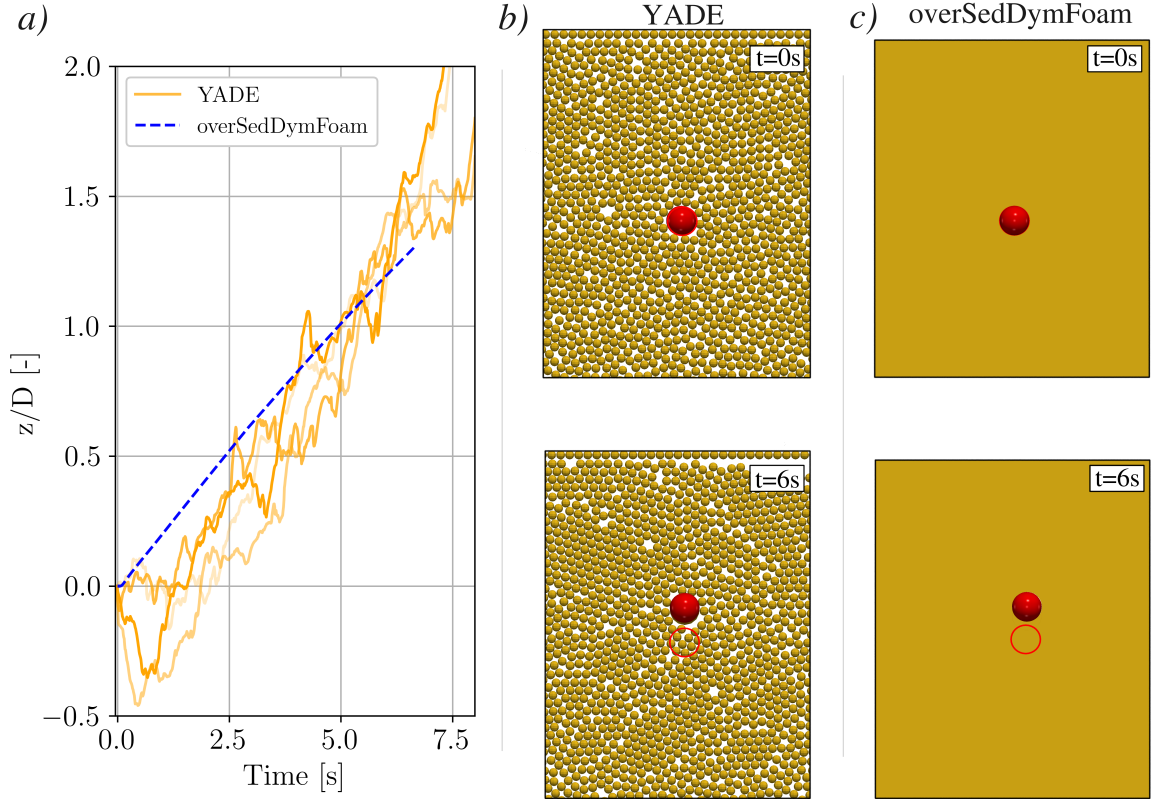


Figure 10. a) Vertical position of the intruder as a function of time. b) YADE and c) overSedDymFoam cylinder position at the initial time ($t=0s$) and $t=6s$. The red circle indicates the original position of the intruder.

path of the YADE series compared to the monotonic trajectory predicted by overSedDymFoam. This behavior is not unexpected due to the nature of the DEM simulations where local effects may modify the trajectory. We also observe that the slope of the trajectory obtained with overSedDymFoam is slightly lower than the DEM one. Although this difference could probably be attributed to the chosen granular rheology and the particle velocity condition applied on the cylinder, it is worth highlighting the remarkable similar path in terms of time scale and average upward motion shown in Fig. 10b and Fig. 10c.

5. Conclusion

The implementation of the 6DoF library into the sedFoam code has been successfully accomplished. As far as OpenFOAM® development is concerned, the main novelty of this work is that, in addition to the fluid stresses, the contribution of granular stresses acting on a solid object is also considered to determine the velocity and motion of a solid object. First, the main modifications to the code to enable dynamic meshes were presented. Then, the specificities of the 6DoF library were discussed. Following the code modifications, the numerical model was tested using a validation case where no particles were present. Additionally, overSedDymFoam was utilized to investigate the effects of sediment stresses in three distinct scenarios. These numerical cases encompassed a dilute suspension (involving a falling sphere immersed in a dilute suspension), a dense granular medium (studying the lift force on a cylinder immersed in a uniform dense granular flow), and a situation where the sphere interacted with the interface between the granular medium and the pure fluid phase.

The results obtained from these tests demonstrated that our proof-of-concept implementation successfully replicates the dynamics of solid objects subjected to both granular and fluid forces. This is a very important contribution to further address major scientific open-questions such as scour erosion around pipelines and communication cables, piping or burial of unexploded ordnances. Naturally, these are research perspectives that are beyond the scope of the present contribution.

6. Data reproducibility

Numerical cases presented in the article and post-processing scripts are located at `$/tutorialsDyM`. In addition to the *Case files* and *source code* found in the file system of the OpenFOAM® Journal, the

version of the `SedFoam` solver used to reproduce the main figures of the article is available on ³Zenodo with the following DOI: <https://doi.org/10.5281/zenodo.5095242>. Moreover, results, the cases for the merge convergence study and post-process scripts are available for download on ⁴Zenodo.

Furthermore, the user can also follow the tutorials at ⁵<https://sedfoam.github.io/sedfoam/> to better understand the main commands and files used to generate the numerical cases of the present article.

Acknowledgements

This research is supported by the Strategic Environmental Research and Development Program (MR20-1478). J. Chauchat and C. Bonamy would also like to acknowledge the financial support from Agence Nationale de la Recherche (ANR) through project SheetFlow (ANR-18-CE01-0003). Part of the computations presented in this paper were also performed using the GENCI infrastructure under Allocations A0140107567 and the GRICAD infrastructure. We are grateful to the developers involved in OpenFOAM, who lay the foundation of the `SedFoam` model.

Author Contributions: Conceptualisation, E.P.M. and C.B.; methodology, E.P.M. and C.B.; software, E.P.M. and C.B.; validation, E.P.M.; formal analysis, E.P.M., C.B. and J.C.; investigation, E.P.M.; resources, E.P.M., C.B. and J.C.; data curation, E.P.M., C.B. and J.C.; writing—original draft preparation, E.P.M.; writing—review and editing, E.P.M., C.B., J.C. and T.J.H.; visualisation, E.P.M.; supervision, C.B. and J.C.; project administration, J.C. and T.J.H.; funding acquisition, T.J.H. All authors have read and agreed to the published version of the manuscript.

Appendix A. Summary of numerical parameters

The following table summarizes the numerical schemes, correctors and other numerical parameters used in the previous numerical simulations.

Table 4. Numerical inputs, loops and schemes used in the numerical simulations.

Parameter	Falling sphere		Resting sphere	Cylinder
	Overset	Morphing	Morphing	Morphing
correctPhi	false	true	true	true
nOuterCorrectors	11	4	6	2
nCorrectors	1	7	3	4
nNonOrthogonalCorrectors	0	3	4	6
Time discretization	Euler			
Gradient term discretization	Gauss linear			
Divergence operators	Gauss linear			
Laplacian operator	Gauss linear corrected			
oversetInterpolation	inverseDistance	-	inverseDistance	-
6DoF solver	Newmark			

Appendix B. Merge convergence studies

Ensuring accuracy and efficiency in computational simulations relies significantly on selecting an appropriate mesh size. The convergence tests presented in this appendix explore the impact of different grid sizes, aiming to find a balance between computational resources and numerical precision.

The time step, Δt , is adjusted automatically based on two Courant numbers, one related to the local flow velocity and the local grid size, written as

$$Co = \frac{1}{2V_c} \sum_f \Phi_f \Delta t, \quad (34)$$

and one related to the relative velocity, written as

$$Co = \frac{1}{2V_c} \sum_f |\Phi_f^a - \Phi_f^b| \Delta t, \quad (35)$$

³<https://zenodo.org/records/5095242>

⁴<https://zenodo.org/records/10967984>

⁵<https://sedfoam.github.io/sedfoam/>

where the subscript f denotes the cell faces and V_c corresponds to the volume of the cell. The most limiting time step is used as the criterion for setting the adjustable time step. In this work, the Courant number is set to 0.01.

B.1. Falling sphere in pure fluid - Morphing approach.

This section is centered on the mesh convergence analysis of the falling sphere scenario using the morphing mesh. Before we start the mesh study, it is worth mentioning that a more accurate fit can be achieved by shifting the curves. As illustrated in Fig. 11, a mere 0.035-second delay aligns the numerical curve more accurately with the experimental data. Although this delay could be originated from experimental inaccuracies, for the main content of the manuscript and subsection B.2, we maintain the uncorrected solution.

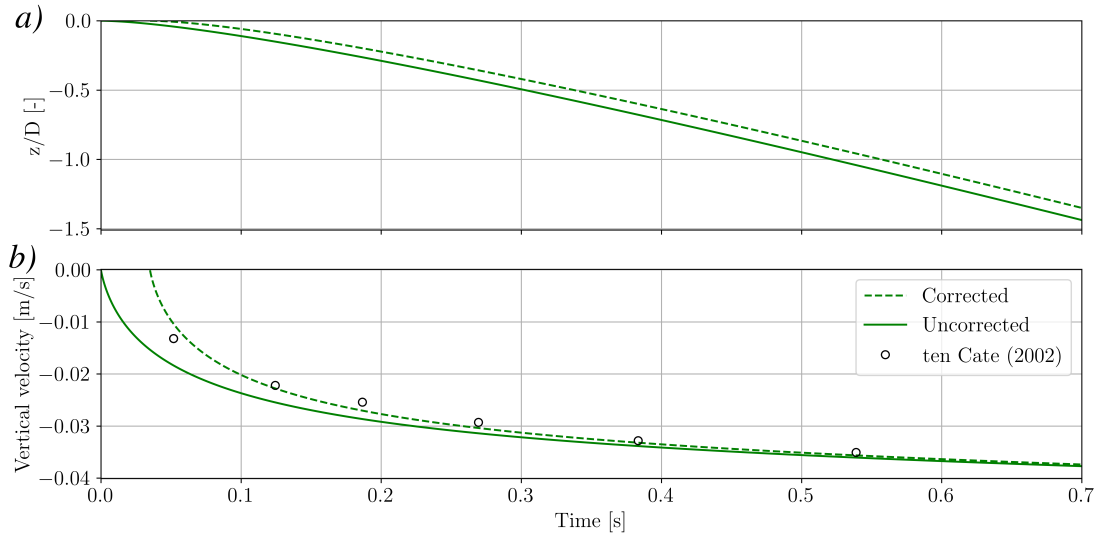


Figure 11. Sphere *a)* trajectory and *b)* velocity evolution using a morphing mesh. Continuous line corresponds to the curves without any modification, whereas the dashed lines refer to the same curves shifted by 0.035 seconds.

Regarding the mesh convergence study, both Fig. 12 and Tab. 5 depict a considerable decrease in error as the mesh is refined. The error displayed in Fig. 12*b* and Fig. 12*d* is computed as

$$Error = \frac{|v_t^{exp} - v_t^{sim}|}{|v_t^{exp}|} \times 100, \quad (36)$$

where v_t^{exp} and v_t^{sim} denote the vertical velocities of the sphere measured by [54] and predicted by overSeddyMFoam, respectively, evaluated at three distinct time instances: A - $t = 0.125s$, B - $t = 0.27s$ and C - $t = 0.54s$.

Yet, this refinement notably escalates the computational time without yielding substantial benefits for meshes more refined than M2 as evidenced by the slight reduction of root-mean-square error (RMSE) reported in Tab. 5.

Table 5. Number of cells, CPU times and root-mean-square error (RMSE) for different mesh resolutions to model the falling sphere in pure fluid with the morphing approach. The RMSE is computed for both corrected and uncorrected initial times.

	Number of cells	CPU time [h]	RMSE	RMSE corrected
M0	768	1.8	0.0028	0.0013
M1	3840	5.3	0.0026	0.0010
M2	34680	36.5	0.0023	0.0007
M3	277440	977.0	0.0022	0.0006

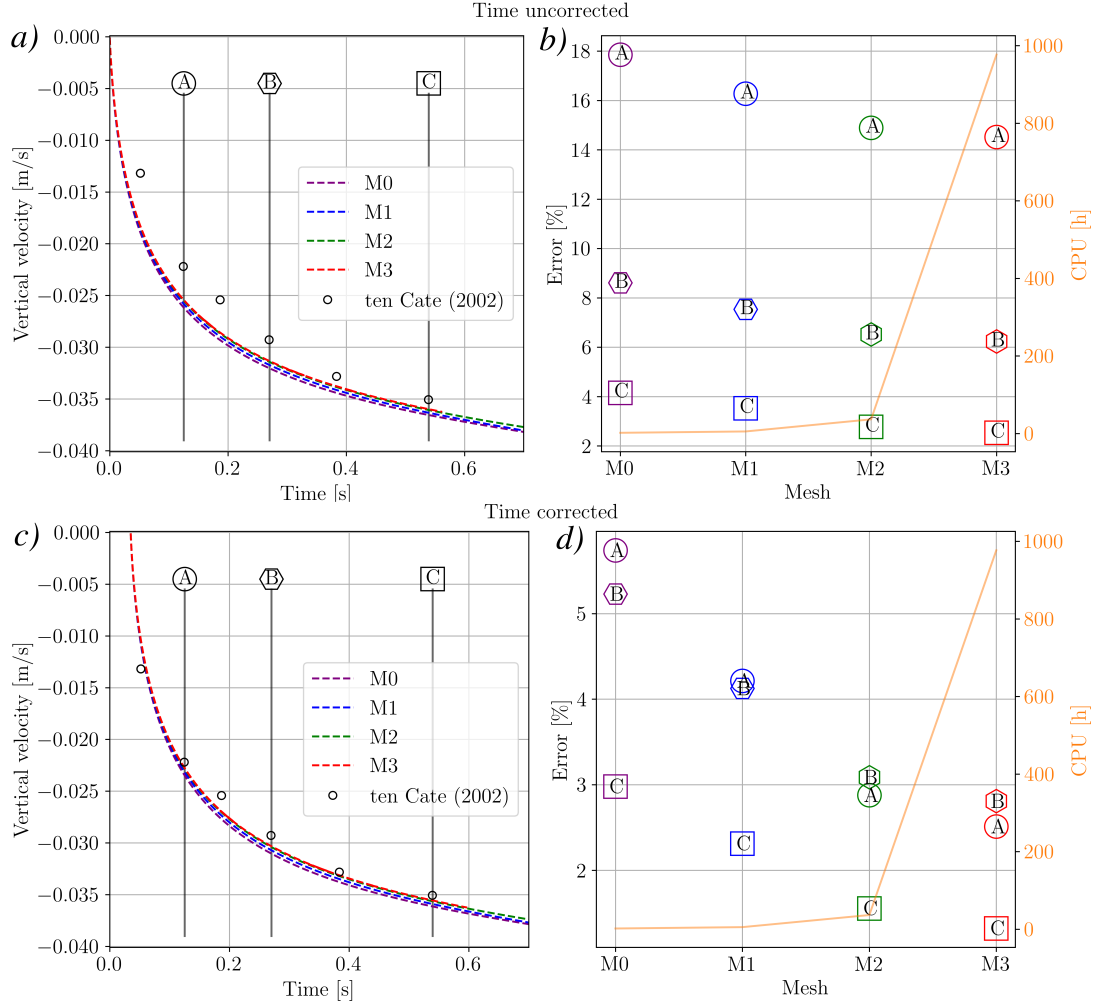


Figure 12. Without initial time correction: *a*) vertical velocity of the sphere as a function of time and *b*) error for different mesh resolutions at different time moments (A - $t = 0.125s$, B - $t = 0.27s$ and C - $t = 0.54s$). The orange continuous line indicates the CPU time needed for each mesh. With initial time correction: *c*) vertical velocity of the sphere as a function of time and *d*) error for different mesh resolutions at different time moments.

B.2. Falling sphere in pure fluid - Overset approach.

We conducted a sensitivity analysis to assess the convergence of our meshes within the overset approach. In this scenario we evaluate the influence of the mesh sizes of the background and overset mesh independently, without considering any time correction.

First, the sensitivity to the background mesh has been tested and results are displayed and summarized in Fig. 13 and Tab. 6. Improvements were observed with mesh M1 compared to the coarsest mesh, M0. In addition to the reduced RMSE, the computational time remained relatively similar. Conversely, as depicted in Fig. 13b and detailed in Tab. 6, the finest mesh, M2, evidenced marginal improvement compared to M1.

Table 6. Number of cells of the background mesh, CPU times and RMSE for different mesh resolutions of the background mesh to model the falling sphere in pure fluid with the overset approach.

	Number of cells	CPU time [h]	RMSE
M0	5059	91	0.00221
M1	9720	106	0.00210
M2	23040	120	0.00208

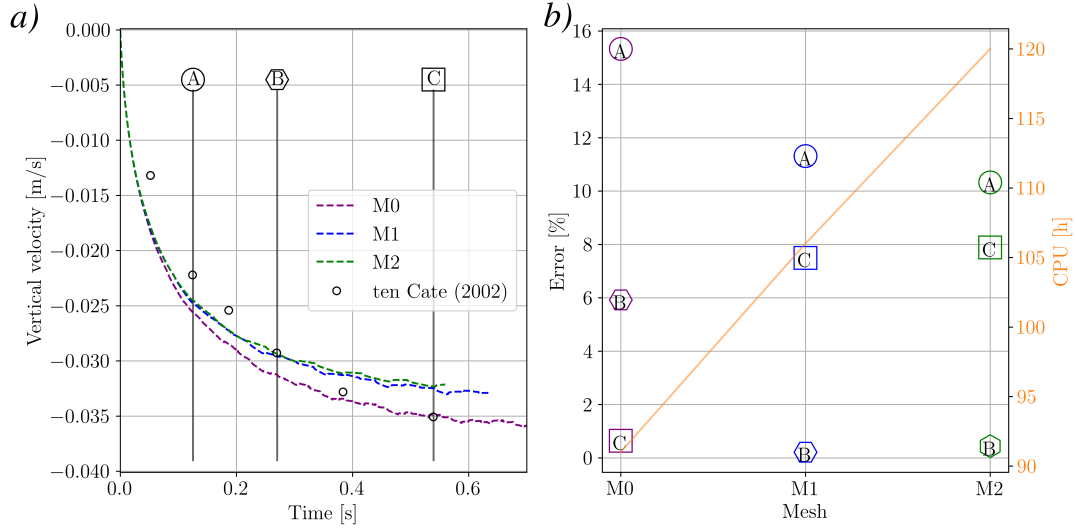


Figure 13. a) Vertical velocity of the sphere as a function of time and b) error for different background grid sizes at different time moments (A - $t = 0.125s$, B - $t = 0.27s$ and C - $t = 0.54s$). The orange continuous line indicates the CPU time needed for each mesh.

Regarding the sensitivity to the mesh resolution of the overset region, as presented in Fig. 14 and Tab. 7, we notice a lower variability compared to the influence of the background mesh quality. Both Fig. 14 and Tab. 7 show that M1 and M2 exhibit low RMSE, suggesting that further refinements would not provide much better results while they would demand substantially higher computational costs.

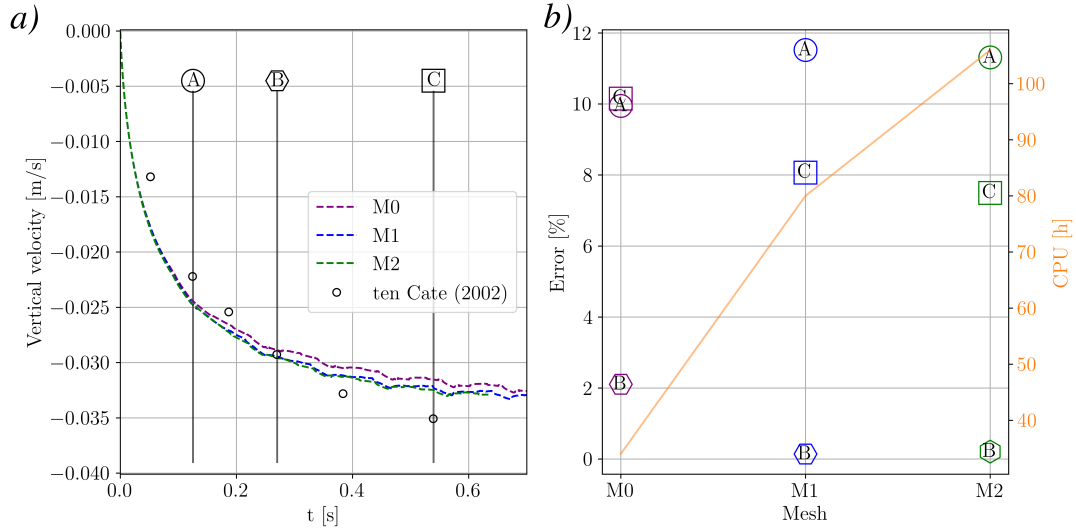


Figure 14. a) Vertical velocity of the sphere as a function of time and b) error for different overset grid sizes at different time moments (A - $t = 0.125s$, B - $t = 0.27s$ and C - $t = 0.54s$). The orange continuous line indicates the CPU time needed for each mesh.

Table 7. Number of cells of the background mesh, CPU times and RMSE for different mesh resolutions of the overset mesh to model the falling sphere in pure fluid with the overset approach.

	Number of cells	CPU time [h]	RMSE
M0	16904	34	0.00245
M1	38903	80	0.00220
M2	58592	106	0.00210

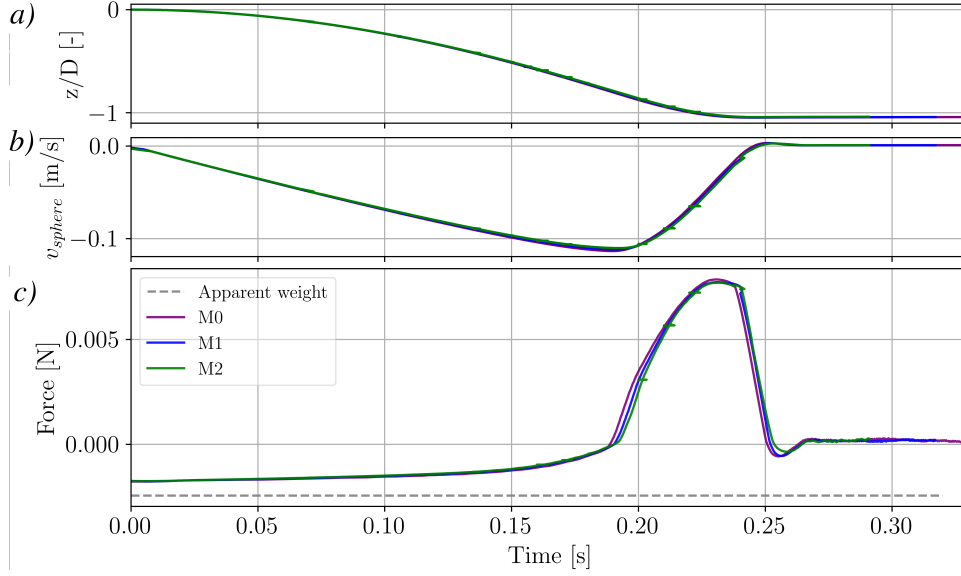


Figure 15. Sphere *a)* trajectory and *b)* velocity obtained with the morphing mesh approach. *c)* Plot of the total forces and each contribution exerted on the sphere from the morphing mesh approach. Resting sphere numerical case.

B.3. Resting sphere.

This section focuses on the mesh convergence study of the resting sphere on a granular bed using the morphing mesh. Three meshes have been considered as reported in Fig. 15 and Tab. 8.

The different curves displayed in Fig. 15 indicate minimal differences among the three meshes. Finer meshes were omitted due to their high computational expense, while coarser meshes than M0, were excluded due to numerical problems. In fact, according to Tab. 8 and Fig. 15, using M0 is sufficient to replicate the numerical case, showing no advantage in employing a finer mesh.

Table 8. Number of cells of the mesh and CPU times for different grid sizes of the morphing mesh to model the sphere falling on a dense granular bed.

	Number of cells	CPU time [h]
M0	291600	211
M1	550338	1210
M2	984150	6240

B.4. Uniform granular flow around a cylinder.

The final section of the mesh convergence study focuses on examining the impact of grid resolution on the numerical simulation of the uniform granular flow around a cylindrical obstacle. The results depicted in Fig. 16 indicate that the intruder is lifted more rapidly with coarser meshes. As a matter of fact, the slopes of the intruder trajectories obtained with YADE closely resemble those of M0. However, the initial delay in YADE simulations brings the curves obtained with finer meshes in *overSedDymFoam* closer to the trajectories reported by YADE, resulting in a lower RMSE as detailed in Tab. 9. Moreover, it is noteworthy that very similar trajectories are observed between M2 and M3, implying that further refinements are unlikely to yield different results.

Table 9. Number of cells of the mesh, CPU times and RMSE for different grid sizes of the morphing mesh to model the uniform granular flow around a cylinder.

	Number of cells	CPU time [h]	RMSE
M0	3652	7.5	0.40
M1	13832	43.1	0.28
M2	31700	226.2	0.20
M3	70864	1029.0	0.18

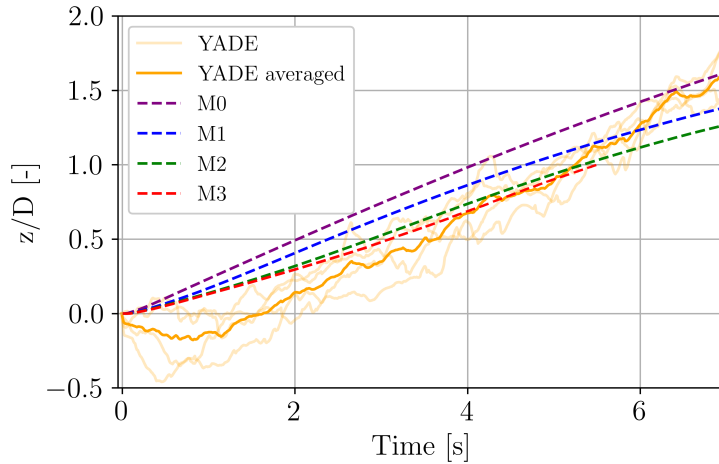


Figure 16. Vertical position of the intruder as a function of time using different grid sizes.

References

- [1] B. M. Sumer, H. R. Jensen, Y. Mao, and J. Fredsøe, “Effect of lee-wake on scour below pipelines in current,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 114, no. 5, pp. 599–614, 1988.
- [2] R. Whitehouse, *Scour at marine structures: A manual for practical applications*. Thomas Telford, 1998.
- [3] B. W. Melville and A. J. Raudkivi, “Effects of foundation geometry on bridge pier scour,” *Journal of Hydraulic Engineering*, vol. 122, no. 4, pp. 203–209, 1996.
- [4] B. M. Sumer *et al.*, *The mechanics of scour in the marine environment*. World Scientific, 2002.
- [5] A. Schendel, A. Hildebrandt, N. Goseberg, and T. Schlurmann, “Processes and evolution of scour around a monopile induced by tidal currents,” *Coastal Engineering*, vol. 139, pp. 65–84, 2018.
- [6] C. Baykal, B. M. Sumer, D. R. Fuhrman, N. G. Jacobsen, and J. Fredsøe, “Numerical investigation of flow and scour around a vertical circular cylinder,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2033, p. 20140104, 2015.
- [7] A. Mathieu, J. Chauchat, C. Bonamy, and T. Nagel, “Two-phase flow simulation of tunnel and lee-wake erosion of scour below a submarine pipeline,” *Water*, vol. 11, no. 8, p. 1727, 2019.
- [8] T. Nagel, J. Chauchat, C. Bonamy, X. Liu, Z. Cheng, and T.-J. Hsu, “Three-dimensional scour simulations with a two-phase flow model,” *Advances in Water Resources*, vol. 138, p. 103544, 2020.
- [9] A. Roulund, B. M. Sumer, J. Fredsøe, and J. Michelsen, “Numerical and experimental investigation of flow and scour around a circular pile,” *Journal of Fluid mechanics*, vol. 534, pp. 351–401, 2005.
- [10] B. Brørs, “Numerical modeling of flow and scour at pipelines,” *Journal of hydraulic Engineering*, vol. 125, no. 5, pp. 511–523, 1999.
- [11] W. Zhang, M. U. Zapata, X. Bai, D. Pham-Van-Bang, and K. D. Nguyen, “Three-dimensional simulation of horseshoe vortex and local scour around a vertical cylinder using an unstructured finite-volume technique,” *International Journal of Sediment Research*, vol. 35, no. 3, pp. 295–306, 2020.
- [12] S. Gautam, D. Dutta, H. Bihs, and M. S. Afzal, “Three-dimensional computational fluid dynamics modelling of scour around a single pile due to combined action of the waves and current using level-set method,” *Coastal Engineering*, vol. 170, p. 104002, 2021.
- [13] E. Meyer-Peter and R. Müller, “Formulas for bed-load transport,” in *IAHSR 2nd meeting, Stockholm, appendix 2*. IAHR, 1948.
- [14] F. Engelund and J. Fredsøe, “A sediment transport model for straight alluvial channels,” *Hydrology Research*, vol. 7, no. 5, pp. 293–306, 1976.
- [15] L. Amoudry and P.-F. Liu, “Two-dimensional, two-phase granular sediment transport model with applications to scouring downstream of an apron,” *Coastal Engineering*, vol. 56, no. 7, pp. 693–702, 2009.
- [16] J. Chauchat, Z. Cheng, T. Nagel, C. Bonamy, and T.-J. Hsu, “Sedfoam-2.0: a 3-d two-phase flow numerical model for sediment transport,” *Geoscientific Model Development*, vol. 10, no. 12, 2017.
- [17] B. Tsai, A. Mathieu, E. P. Montellà, T.-J. Hsu, and J. Chauchat, “An eulerian two-phase flow model investigation on scour onset and backfill of a 2d pipeline,” *European Journal of Mechanics-B/Fluids*, vol. 91, pp. 10–26, 2022.
- [18] M. Robinson, M. Ramaioli, and S. Luding, “Fluid–particle flow simulations using two-way-coupled mesoscale sph–dem and validation,” *International journal of multiphase flow*, vol. 59, pp. 121–134, 2014.
- [19] G. Pahar and A. Dhar, “On information coupling in hybrid isph framework for fluidized granular systems,” *Advances in Water Resources*, vol. 131, p. 103366, 2019.
- [20] Y. Guo and X. B. Yu, “Comparison of the implementation of three common types of coupled cfd-dem model for simulating soil surface erosion,” *International Journal of Multiphase Flow*, vol. 91, pp. 89–100, 2017.
- [21] T. Nian, D. Li, Q. Liang, H. Wu, and X. Guo, “Multi-phase flow simulation of landslide dam formation process based on extended coupled dem-cfd method,” *Computers and Geotechnics*, vol. 140, p. 104438, 2021.
- [22] P. Gupta, J. Sun, and J. Ooi, “Dem-cfd simulation of a dense fluidized bed: Wall boundary and particle size effects,” *Powder technology*, vol. 293, pp. 37–47, 2016.

- [23] E. P. Montellà, C. Yuan, B. Chareyre, and A. Gens, “Hybrid multi-scale model for partially saturated media based on a pore network approach and lattice boltzmann method,” *Advances in Water Resources*, vol. 144, p. 103709, 2020.
- [24] H. Jasak, “Openfoam: open source cfd in research and industry,” *International Journal of Naval Architecture and Ocean Engineering*, vol. 1, no. 2, pp. 89–94, 2009.
- [25] Y.-l. Li, R.-c. Zhu, G.-p. Miao, and F. Ju, “Simulation of tank sloshing based on openfoam and coupling with ship motions in time domain,” *Journal of Hydrodynamics, Ser. B*, vol. 24, no. 3, pp. 450–457, 2012.
- [26] P. J. Martínez-Ferrer, L. Qian, Z. Ma, D. M. Causon, and C. G. Mingham, “Improved numerical wave generation for modelling ocean and coastal engineering problems,” *Ocean Engineering*, vol. 152, pp. 257–272, 2018.
- [27] D. P. Guillen and A. W. Abboud, “Heat transfer enhancement due to cold cap motion from bubbling in a waste glass melter,” Idaho National Laboratory (INL), Idaho Falls, ID (United States), Tech. Rep., 2023.
- [28] O. E. Myrli and H. Khawaja, “Fluid-structure interaction (fsi) modelling of aquaculture net cage,” 2019.
- [29] X. Nan, X. Liu, L. Chen, Q. Yan, and J. Li, “Study of the bridge damage during flooding based on a coupled vof-fsi method,” *Journal of Engineering Research*, p. 100081, 2023.
- [30] C. Antoci, M. Gallati, and S. Sibilla, “Numerical simulation of fluid–structure interaction by sph,” *Computers & structures*, vol. 85, no. 11–14, pp. 879–890, 2007.
- [31] Y. Tang, Q. Jiang, and C. Zhou, “A lagrangian-based sph-dem model for fluid–solid interaction with free surface flow in two dimensions,” *Applied Mathematical Modelling*, vol. 62, pp. 436–460, 2018.
- [32] F. Xie, G. Zhang, and D. Wan, “Mps-dem coupled method for fluid-structure interaction with free surface flows,” in *ISOPE Pacific/Asia Offshore Mechanics Symposium*. ISOPE, 2020, pp. ISOPE-P.
- [33] A. Leonardi, F. K. Wittel, M. Mendoza, R. Vetter, and H. J. Herrmann, “Particle–fluid–structure interaction for debris flow impact on flexible barriers,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 31, no. 5, pp. 323–333, 2016.
- [34] K. Wu, D. Yang, N. Wright, and A. Khan, “An integrated particle model for fluid–particle–structure interaction problems with free-surface flow and structural failure,” *Journal of fluids and structures*, vol. 76, pp. 166–184, 2018.
- [35] Y. He, A. E. Bayly, and A. Hassanpour, “Coupling cfd-dem with dynamic meshing: A new approach for fluid-structure interaction in particle-fluid flows,” *Powder Technology*, vol. 325, pp. 620–631, 2018.
- [36] H. Jasak and T. Uroić, “Practical computational fluid dynamics with the finite volume method,” *Modeling in Engineering Using Innovative Numerical Methods for Solids and Fluids*, pp. 103–161, 2020.
- [37] R. Sun and H. Xiao, “Sedifoam: A general-purpose, open-source cfd–dem solver for particle-laden flow with emphasis on sediment transport,” *Computers & Geosciences*, vol. 89, pp. 207–219, 2016.
- [38] Y. Li, *Development of a new Euler-Lagrange model for the prediction of scour around offshore structures*. The University of Liverpool (United Kingdom), 2014.
- [39] N. R. Olsen, S. Kadia, E. Pummer, and G. Hillebrand, “An openfoam solver for computing suspended particles in water currents,” *Journal of Hydroinformatics*, 2023.
- [40] H. Rusche, “Computational fluid dynamics of dispersed two-phase flows at high phase fractions,” Ph.D. dissertation, Imperial College London (University of London), 2003.
- [41] H. Jasak and Z. Tukovic, “Automatic mesh motion for the unstructured finite volume method,” *Transactions of FAMENA*, vol. 30, no. 2, pp. 1–20, 2006.
- [42] S. Ergun, “Fluid flow through packed columns,” *Chemical engineering progress*, vol. 48, no. 2, p. 89, 1952.
- [43] H. Enwald, E. Peirano, and A.-E. Almstedt, “Eulerian two-phase flow theory applied to fluidization,” *International Journal of Multiphase Flow*, vol. 22, pp. 21–66, 1996.
- [44] F. Boyer, É. Guazzelli, and O. Pouliquen, “Unifying suspension and granular rheology,” *Physical Review Letters*, vol. 107, no. 18, p. 188301, 2011.
- [45] P. C. Johnson and R. Jackson, “Frictional–collisional constitutive relations for granular materials, with application to plane shearing,” *Journal of fluid Mechanics*, vol. 176, pp. 67–93, 1987.
- [46] Z. Cheng, T.-J. Hsu, and J. Calantoni, “Sedfoam: A multi-dimensional eulerian two-phase model for sediment transport and its application to momentary bed failure,” *Coastal Engineering*, vol. 119, pp. 32–50, 2017.
- [47] Y. Forterre and O. Pouliquen, “Flows of dense granular media,” *Annu. Rev. Fluid Mech.*, vol. 40, pp. 1–24, 2008.
- [48] E. Montellà, J. Chauchat, B. Chareyre, C. Bonamy, and T. Hsu, “A two-fluid model for immersed granular avalanches with dilatancy effects,” *Journal of Fluid Mechanics*, vol. 925, 2021.
- [49] E. Montellà, J. Chauchat, C. Bonamy, D. Weij, G. Keetels, and T. Hsu, “Numerical investigation of mode failures in submerged granular columns,” *Flow*, vol. 3, p. E28, 2023.
- [50] G. MiDi, “On dense granular flows,” *The European Physical Journal E*, vol. 14, no. 4, pp. 341–365, 2004.
- [51] C. Cassar, M. Nicolas, and O. Pouliquen, “Submarine granular flows down inclined planes,” *Physics of fluids*, vol. 17, no. 10, p. 103301, 2005.
- [52] J. Chauchat and M. Médale, “A three-dimensional numerical model for dense granular flows based on the μ (i) rheology,” *Journal of Computational Physics*, vol. 256, pp. 696–712, 2014.
- [53] P. Tisovska, “Description of the overset mesh approach in esi version of openfoam,” *Proceedings of CFD with Open-Source Software*, 2019.
- [54] A. Ten Cate, C. Nieuwstad, J. Derksen, and H. Van den Akker, “Particle imaging velocimetry experiments and lattice-boltzmann simulations on a single sphere settling under gravity,” *Physics of Fluids*, vol. 14, no. 11, pp. 4012–4025, 2002.
- [55] G. Batchelor, “Sedimentation in a dilute dispersion of spheres,” *Journal of fluid mechanics*, vol. 52, no. 2, pp. 245–268, 1972.
- [56] R. H. Davis and A. Acrivos, “Sedimentation of noncolloidal particles at low reynolds numbers,” *Annual Review of Fluid Mechanics*, vol. 17, no. 1, pp. 91–118, 1985.
- [57] D. Van Der Meer, “Impact on granular beds,” *Annual review of fluid mechanics*, vol. 49, pp. 463–484, 2017.
- [58] H. Katsuragi et al., *Physics of soft impact and cratering*. Springer, 2016.
- [59] J. Ruiz-Suárez, “Penetration of projectiles into granular targets,” *Reports on Progress in Physics*, vol. 76, no. 6, p. 066601, 2013.

- [60] D. Lohse, R. Bergmann, R. Mikkelsen, C. Zeilstra, D. Van Der Meer, M. Versluis, K. Van Der Weele, M. van der Hoef, and H. Kuipers, “Impact on soft sand: void collapse and jet formation,” *Physical review letters*, vol. 93, no. 19, p. 198003, 2004.
- [61] H. Katsuragi and D. J. Durian, “Unified force law for granular impact cratering,” *Nature physics*, vol. 3, no. 6, pp. 420–423, 2007.
- [62] —, “Drag force scaling for penetration into granular media,” *Physical review E*, vol. 87, no. 5, p. 052208, 2013.
- [63] A. H. Clark, L. Kondic, and R. P. Behringer, “Particle scale dynamics in granular impact,” *Physical review letters*, vol. 109, no. 23, p. 238302, 2012.
- [64] L. Kondic, X. Fang, W. Losert, C. O’Hern, and R. Behringer, “Microstructure evolution during impact on granular matter,” *Physical Review E*, vol. 85, no. 1, p. 011305, 2012.
- [65] Y. Xu, J. Padding, M. van der Hoef, and J. Kuipers, “Detailed numerical simulation of an intruder impacting on a granular bed using a hybrid discrete particle and immersed boundary (dp-ib) method,” *Chemical engineering science*, vol. 104, pp. 201–207, 2013.
- [66] A. J. Liu and S. R. Nagel, *Jamming and rheology: constrained dynamics on microscopic and macroscopic scales*. CRC Press, 2001.
- [67] D. I. Goldman and P. Umbanhowar, “Scaling and dynamics of sphere and disk impact into granular media,” *Physical Review E*, vol. 77, no. 2, p. 021308, 2008.
- [68] H. Rousseau, R. Chassagne, J. Chauchat, R. Maurin, and P. Frey, “Bridging the gap between particle-scale forces and continuum modelling of size segregation: application to bedload transport,” *Journal of Fluid Mechanics*, vol. 916, p. A26, 2021.
- [69] R. Chassagne, R. Maurin, J. Chauchat, J. Gray, and P. Frey, “Discrete and continuum modelling of grain size segregation during bedload transport,” *Journal of Fluid Mechanics*, vol. 895, p. A30, 2020.
- [70] Y. Ding, N. Gravish, and D. I. Goldman, “Drag induced lift in granular media,” *Physical review letters*, vol. 106, no. 2, p. 028001, 2011.
- [71] F. Guillard, Y. Forterre, and O. Pouliquen, “Lift forces in granular media,” *Physics of Fluids*, vol. 26, no. 4, p. 043301, 2014.
- [72] V. Smilauer, V. Angelidakis, E. Catalano, R. Caulk, B. Chareyre, W. Chevrement, S. Dorofeenko, J. Duriez, N. Dyck, J. Elias *et al.*, “Yade documentation,” *arXiv preprint arXiv:2301.00611*, 2023.