

Non-Inertial Lagrangian Particle Tracking on Arbitrarily Moving Grids

F. Caccia* and A. Guardone*

Corresponding author: alberto.guardone@polimi.it

* Politecnico di Milano, Milan, Italy.

Abstract: This paper presents a local non-inertial Lagrangian particle tracking (LPT) formulation for one-way coupled particle transport on moving grids, with the present study focused on rigid grid motion. The method advances particle velocity and acceleration in the inertial frame while updating particle position on a grid frozen over each carrying phase time interval. To recover frame consistency in the presence of grid rotation, the grid rotation vector is reconstructed from the grid velocity field and used to rotate particle displacements consistently within the frozen frame. In this way, the method avoids intermediate moving-mesh reconstruction during particle substepping while preserving the correct kinematics for rigid-body translation and rotation. The formulation is verified on prescribed rigid-motion test cases, including steady rotation and time-dependent pitching motion, and is shown to recover the expected convergence behaviour. It is then validated on a pitching-airfoil test case. The proposed approach provides a low-cost moving-grid treatment and can be directly extended to deforming-grid scenarios through a more general local kinematic description.

1. Introduction

Lagrangian particle tracking (LPT) is a widely used approach for modelling dilute multiphase flows in which a dispersed phase is transported by a carrier fluid. Such flows arise in a broad range of engineering and scientific applications, including atmospheric and environmental flows, and propulsion and energy systems. The interaction can lead to a wide range of phenomena, including surface fouling, erosion, and ice accretion. In many of these problems, the fluid domain is not stationary but undergoes a motion, such as translation or rotation, which is typically handled in computational fluid dynamics (CFD) through moving-grid formulations.

When the carrier flow is computed on a moving mesh, most commonly within an Arbitrary Lagrangian–Eulerian (ALE) framework, the tracking of particles becomes non-trivial. Particle trajectories must remain physically invariant with respect to the chosen reference frame, while accounting for grid motion in a numerically robust manner. This challenge is particularly relevant in unsteady aeronautical applications, such as rotor and propeller flows, where recent advances in high-performance computing have enabled fully three-dimensional, time-accurate simulations using Eulerian formulations for the dispersed phase [1, 2, 3, 4].

Conventional LPT strategies advance particles in an inertial frame and require continuous interpolation of grid motion, which can introduce additional computational cost and algorithmic complexity [5]. This led to limited use in aerospace applications, requiring alternative formulations based on non-inertial descriptions on frozen grids [6]. Such formulations reduce algorithmic complexity and computational overhead related to grid motion interpolation, but limitations remain in terms of generality, frame consistency, or time-resolved accuracy of particle statistics.

The present work is framed within this context. A local non-inertial LPT formulation is introduced and verified for grids undergoing rigid motion. The particle velocity and acceleration are updated in the inertial frame, while the particle position is advanced on a grid frozen over the current carrying phase time interval. To recover frame consistency under rigid rotation, the grid rotation vector is reconstructed

from the grid velocity field and then used to rotate the particle displacement prior to the position update in the frozen frame. The application is to one-way coupled particle-laden flows. The proposed approach aims to provide an accurate and efficient particle-tracking treatment for moving-grid flow problems, while reducing algorithmic complexity and computational overhead. General deforming-grid motion is not addressed here, but the present formulation can be directly extended through a more general kinematic treatment based on the local deformation map.

The remainder of the paper is organised as follows. Section 2 presents the proposed local non-inertial formulation. Section 3 reports the verification results for rigidly moving grids. Final remarks are given in Section 5.

2. Methodology

The present formulation is restricted to rigid grid motion over each carrying phase time interval. The particle dynamics are integrated in the inertial frame, while particle positions are updated on a grid that is kept frozen during the particle substeps. A local rotational correction is then introduced to preserve frame consistency when the grid undergoes rigid-body rotation. Particles are tracked with the in-house code PoliDrop [7].

2.1 Time discretization

Let the carrier-phase solution be available in the inertial frame at the carrying phase times t_k and t_{k+1} , with

$$\Delta t = t_{k+1} - t_k. \quad (1)$$

Over the same interval, the dispersed phase is integrated with a smaller time step

$$\Delta \tau = \frac{\Delta t}{N}, \quad (2)$$

where N is the number of particle substeps. The substep times are

$$t_{k,n} = t_k + n\Delta \tau, \quad n = 0, \dots, N. \quad (3)$$

At time t_k , the nodal grid coordinates, the grid velocity field, and the carrier-flow variables are assumed known from the CFD solver. During the interval $[t_k, t_{k+1}]$, the mesh topology is frozen at time t_k for tracking particles. The quantities required by the particle force model are interpolated in time between the two CFD states.

2.2 Grid kinematics

The present paper considers the case in which the grid motion is locally rigid over the CFD interval. The grid velocity field can then be written as

$$\mathbf{v}_G(\mathbf{x}, t) = \mathbf{V}_G(t) + \boldsymbol{\Omega}_G(t) \times (\mathbf{x} - \mathbf{x}_c(t)), \quad (4)$$

where $\mathbf{V}_G(t)$ is the translational velocity, $\boldsymbol{\Omega}_G(t)$ is the angular velocity vector, and $\mathbf{x}_c(t)$ is a reference point. With this grid kinematics, the skew-symmetric part of the grid-velocity gradient uniquely identifies the angular velocity vector. Equivalently,

$$\boldsymbol{\Omega}_G(t) = \frac{1}{2} \nabla \times \mathbf{v}_G(t). \quad (5)$$

In the implementation, a piecewise-constant angular velocity is used over each CFD interval:

$$\mathbf{\Omega}_G^k = \frac{1}{2} [\mathbf{\Omega}_G(t_k) + \mathbf{\Omega}_G(t_{k+1})]. \quad (6)$$

2.3 Local non-inertial particle update

The particle acceleration and velocity are updated in the inertial frame. Denoting by $\mathbf{v}_f^{k,n}$ the carrier velocity seen by the particle at substep (k, n) , the particle dynamics can be computed as

$$\mathbf{a}_p^{k,n} = f(\mathbf{v}_p^{k,n}, \mathbf{v}_f^{k,n}, \dots), \quad (7)$$

$$\mathbf{v}_p^{k,n+1} = \mathbf{v}_p^{k,n} + \mathbf{a}_p^{k,n} \Delta\tau, \quad (8)$$

where $f(\cdot)$ denotes the adopted particle-force model.

The corresponding inertial displacement over the particle substep is

$$\Delta\mathbf{x}_p^{k,n+1} = \mathbf{v}_p^{k,n+1} \Delta\tau. \quad (9)$$

The particle position is not updated directly on the moving mesh. Instead, the grid is kept frozen at the configuration corresponding to t_k , and the particle position is advanced in that frozen frame. Let $\tilde{\mathbf{x}}_p^{k,n}$ denote the particle position in the frozen grid associated with the interval $[t_k, t_{k+1}]$, with

$$\tilde{\mathbf{x}}_p^{k,0} = \mathbf{x}_p(t_k), \quad (10)$$

i.e., at $t_{k,n} = t_k$, the particle is in the inertial position.

To account for the frame rotation, the inertial displacement is rotated into the frozen frame using the angular motion accumulated from t_k to $t_{k,n}$:

$$\Delta\mathbf{a}^{k,n} = (t_{k,n} - t_k) \mathbf{\Omega}_G^k. \quad (11)$$

Let $\mathcal{R}(\mathbf{a}, \mathbf{b})$ denote the rotation of the vector $\mathbf{a}$ by the finite rotation vector $\mathbf{b}$, for instance through Rodrigues' formula. The particle displacement expressed in the frozen frame is then

$$\Delta\tilde{\mathbf{x}}_p^{k,n+1} = \mathcal{R}(\Delta\mathbf{x}_p^{k,n+1}, \Delta\mathbf{a}^{k,n}). \quad (12)$$

The grid translation over the same substep is accounted for through the local grid velocity, expressed in the frozen frame:

$$\tilde{\mathbf{v}}_G^{k,n} = \mathcal{R}(\mathbf{v}_G^{k,n}, \Delta\mathbf{a}^{k,n}). \quad (13)$$

The particle position update on the frozen grid is finally written as

$$\tilde{\mathbf{x}}_p^{k,n+1} = \tilde{\mathbf{x}}_p^{k,n} + \Delta\tilde{\mathbf{x}}_p^{k,n+1} - \tilde{\mathbf{v}}_G^{k,n} \Delta\tau, \quad (14)$$

or, equivalently, as

$$\tilde{\mathbf{x}}_p^{k,n+1} = \tilde{\mathbf{x}}_p^{k,n} + \mathcal{R}(\Delta\mathbf{x}_p^{k,n+1} - \mathbf{v}_G^{k,n} \Delta\tau, \Delta\mathbf{a}^{k,n}). \quad (15)$$

Because the mesh is frozen over the interval $[t_k, t_{k+1}]$, particle localisation can be performed with the same static-grid search procedures used in stationary meshes.

2.4 Resynchronization at the carrying phase time

At the end of the particle subcycling, the particle position $\tilde{\mathbf{x}}_p^{k,N}$ is still expressed on the grid frozen at time t_k . Before advancing to the next CFD interval, the particle must be mapped back to the inertial grid configuration at t_{k+1} .

Let

$$\Delta \mathbf{x}_G = \mathbf{x}_G(t_{k+1}) - \mathbf{x}_G(t_k) \quad (16)$$

be the nodal displacement over the CFD interval. For rigid grid motion, the inertial particle position at the beginning of the next interval is recovered by applying the same displacement map evaluated at the particle location:

$$\mathbf{x}_p^{k+1,0} = \tilde{\mathbf{x}}_p^{k,N} + \Delta \mathbf{x}_G(\tilde{\mathbf{x}}_p^{k,N}). \quad (17)$$

The algorithm then proceeds to the next CFD interval, freezing the mesh at t_{k+1} and repeating the same sequence of particle substeps.

3. Verification

3.1 Verification setup

A cubic domain with coordinates $[-32, 32]^3$ is considered. The mesh is composed of a uniform Cartesian discretization of $16 \times 16 \times 16$ hexahedra, subsequently subdivided into tetrahedra for particle localisation. The domain is populated with N_p non-interacting particles governed by the Stokes drag model

$$\frac{d\mathbf{v}_p}{dt} = \frac{\mathbf{v}_f - \mathbf{v}_p}{\tau_p}, \quad (18)$$

where τ_p is the particle relaxation time.

The simulations are performed over the time interval $t \in [0, 1]$. The carrying phase time step is fixed to

$$\Delta t_c = \frac{1}{800}, \quad (19)$$

and every m -th carrying phase solution and grid position is considered known to the particle tracking solver, such that the time interval $\Delta t = t_{k+1} - t_k$ is

$$\Delta t = m \Delta t_c = \frac{m}{800} \quad (20)$$

with

$$m \in \{1, 2, 4, 8, 16, 32\}. \quad (21)$$

Two prescribed rigid grid motions are considered, namely: a steady rotation, where the domain rotates at constant angular velocity

$$\boldsymbol{\Omega}_G(t) = 2\pi \hat{\mathbf{z}}, \quad (22)$$

so that one full revolution is completed over the simulation time; and a pitching motion, where the domain undergoes a time-dependent rigid rotation with angular velocity

$$\boldsymbol{\Omega}_G(t) = 2\pi \cdot 10 \frac{\pi}{180} \cos(2\pi t) \hat{\mathbf{z}}, \quad (23)$$

corresponding to a pitching amplitude of 10° , so that one full cycle is completed over the simulation time.

The trajectory error is measured through the average particle position error

$$\|\mathbf{x}_p - \hat{\mathbf{x}}_p\|_2 = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\mathbf{x}_{p,i} - \hat{\mathbf{x}}_{p,i}\|_2, \quad (24)$$

where $\hat{\mathbf{x}}_{p,i}$ denotes the exact particle position.

3.2 Freestream preservation

A basic consistency requirement is that a particle initially in equilibrium with a uniform carrier flow should remain on the corresponding straight inertial trajectory independently of the rigid motion of the underlying grid. To assess this property, a uniform flow field

$$\mathbf{v}_f = 30 \hat{\mathbf{x}} \quad (25)$$

is prescribed. Particles are initially released in kinematic equilibrium,

$$\mathbf{v}_p(0) = \mathbf{v}_f, \quad (26)$$

from the upstream plane $x = -30$. The exact solution is therefore

$$\mathbf{x}_p(t) = \mathbf{x}_p(0) + \mathbf{v}_f t. \quad (27)$$

The convergence results for the steady-rotation and pitching cases are reported in Figure 1 and Figure 2, respectively. The methodology proposed is compared with results computed by setting $\boldsymbol{\Omega}_G^k = \mathbf{0}$ during the particle position update in the frozen grid, resulting in the algorithm proposed in [6].

For both cases, the translational-only frozen-grid update ($\boldsymbol{\Omega}_G^k = \mathbf{0}$) exhibits a systematic error because the orientation change of the grid-attached frame is not accounted for during the particle substeps. As a result, refining only the particle substep does not remove the dominant kinematic inconsistency associated with the frozen-frame approximation over the carrier phase time interval.

When the rotational correction is included, the expected convergence behaviour is recovered. In the steady rotation case, the proposed method restores the correct first-order behaviour with respect to the particle time step. In the pitching case, the use of an interval-wise averaged angular velocity introduces an additional error associated with the finite carrying phase time step, which is reduced by decreasing m or Δt , i.e., providing a better approximation of $\mathbf{v}_G(t)$ and $\boldsymbol{\Omega}_G^k(t)$.

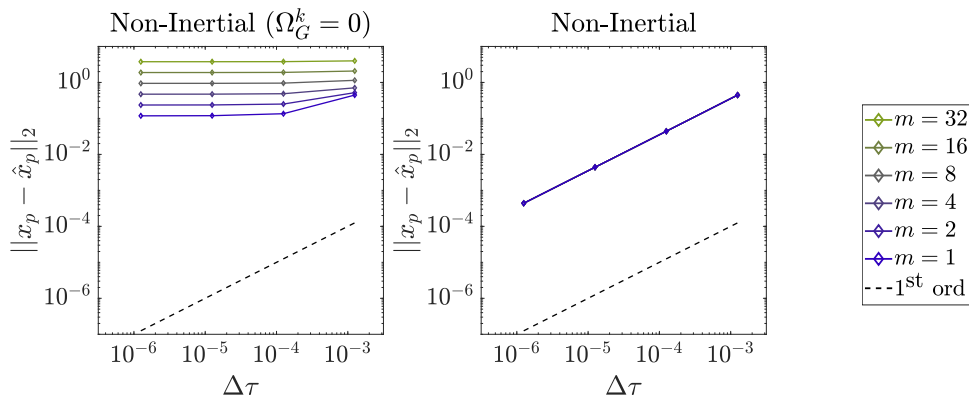


Figure 1: Freestream-preservation error for steady rigid rotation.

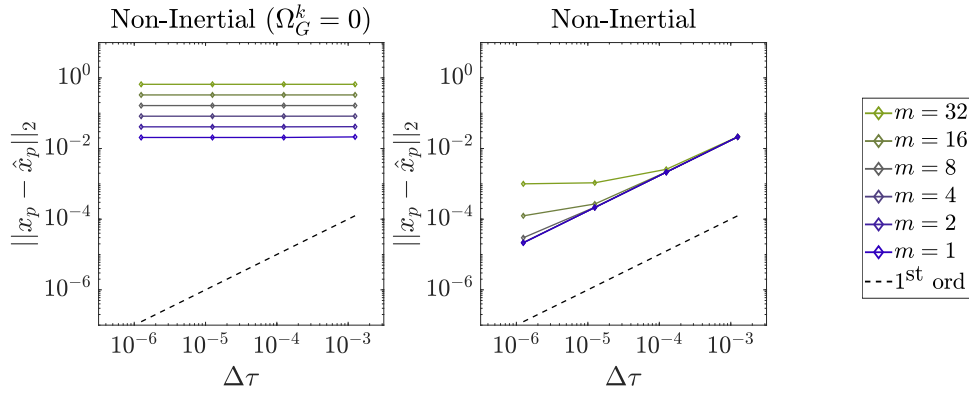


Figure 2: Freestream-preservation error for time-dependent pitching motion.

3.3 Non-uniform, steady flow field

As a second verification case, a non-uniform but time-invariant carrying flow field is prescribed:

$$\mathbf{v}_f(\mathbf{x}) = \mathbf{x} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}}. \quad (28)$$

The particle dynamics are still governed by the Stokes drag model introduced above. In this case, the exact particle trajectory can be obtained analytically and is therefore used to evaluate the tracking error.

For a particle released at $\mathbf{x}_p(0) = \mathbf{x}_{p,0}$ with initial velocity $\mathbf{v}_p(0) = \mathbf{v}_{p,0}$, the exact solution reads

$$\hat{\mathbf{x}}_p(t) = \mathbf{x}_{p,0} \frac{\lambda_1 e^{\lambda_2 t} - \lambda_2 e^{\lambda_1 t}}{\lambda_1 - \lambda_2} + \mathbf{v}_{p,0} \frac{e^{\lambda_1 t} - e^{\lambda_2 t}}{\lambda_1 - \lambda_2}, \quad (29)$$

where

$$\lambda_{1,2} = \frac{-1 \pm \sqrt{1 + 4\tau_p}}{2\tau_p}. \quad (30)$$

In the present tests, particles are initially released in kinematic equilibrium,

$$\mathbf{v}_{p,0} = \mathbf{v}_f(\mathbf{x}_{p,0}) = \mathbf{x}_{p,0}, \quad (31)$$

and their initial positions are distributed along the y axis.

Two frozen-grid non-inertial solutions are compared. The first one uses the exact prescribed frame rotation during particle substepping. In this case, the grid is still frozen over each carrying-phase interval, but the particle position vector is updated using the exact rotation of the non-inertial frame at every $\Delta\tau$, without reconstructing the frame motion from the grid velocity field. The second solution is the proposed local formulation, in which the frame motion is reconstructed from the local grid velocity and angular velocity evaluated at the carrying-phase times. The corresponding errors for steady rigid rotation and pitching motion are reported in Figures 3 and 4, respectively.

For the steady-rotation case, both solutions exhibit the expected first-order behaviour with respect to $\Delta\tau$, and the error is essentially independent of m . The prescribed-rotation solution gives the lower error. In this case, the gap between the two solutions directly measures the discretisation error associated with reconstructing the frame motion through the grid velocity field instead of using the exact rotation of the reference frame.

For the pitching case, the error depends on m also for the prescribed-rotation solution. This is associated with the time interpolation of the carrying-flow velocity during particle substepping. While the carrying flow is steady in the moving frame, it becomes time-dependent in the inertial frame because of the pitching motion. An interpolation assuming constant rotation rate is used, and this introduces an error

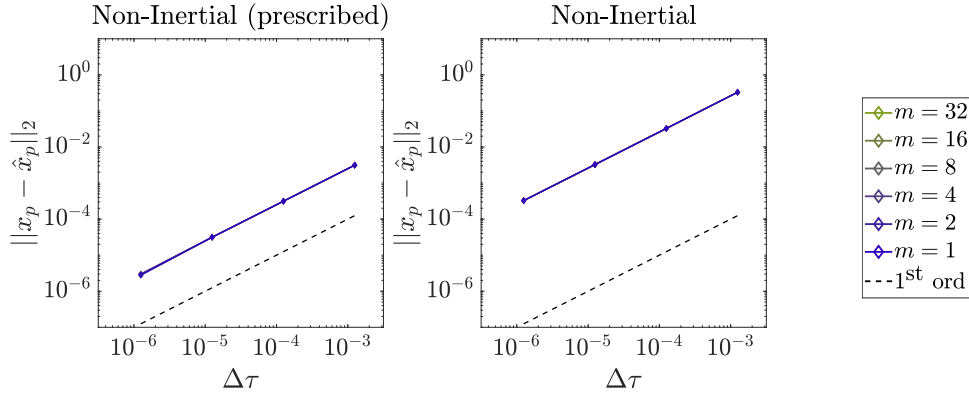


Figure 3: Verification with a steady non-uniform carrying flow field. The domain rotates at constant angular velocity.

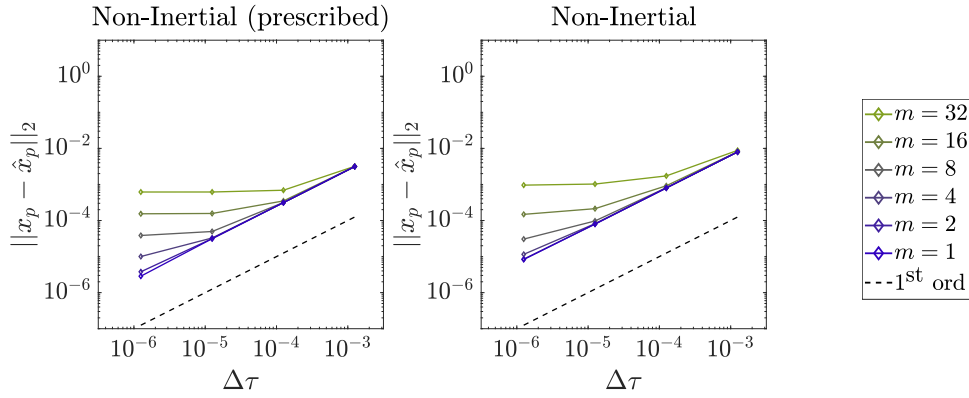


Figure 4: Verification with a steady non-uniform carrying flow field. The domain undergoes a pitching motion.

that depends on $\Delta t = m \Delta t_c$. Since this interpolation is common to both formulations, the same trend with m is observed in both solutions.

4. Validation on a pitching airfoil

As a validation case, the proposed formulation is applied to the pitching SC2110 airfoil test case considered by [8] and later studied numerically by [9]. The reference conditions correspond to Run 36 of the experimental campaign by [8]. The airfoil chord is $c = 0.381$ m, the freestream velocity is $V_\infty = 77.167$ m s⁻¹, and the angle of attack varies sinusoidally as $5^\circ \pm 6^\circ$ at a frequency of 2.8 Hz. The corresponding reduced frequency is

$$k_\Omega = \frac{c}{2} \frac{\Omega}{V_\infty} \approx 0.043. \quad (32)$$

A monodispersed cloud of water droplets with diameter $d_p = 22$ μ m is considered. The corresponding section-based Stokes number is

$$\text{Stk}_0 = \frac{\rho_p d_p^2}{18 \mu_f} \frac{V_\infty}{c} \approx 0.3 \quad (33)$$

resulting in particles affected by the underlying carrying flow field.

The carrying phase is computed by solving unsteady RANS equations on a structured O-grid. The Lagrangian particle tracking solution is obtained with the local non-inertial formulation presented in this paper, using the available moving-grid CFD solution directly, with no prescribed analytical motion. The

carrying-phase solution is sampled every other CFD state, corresponding to 100 carrying-phase inputs per pitching cycle ($m = 2$), and the particle integration time step is set to $\Delta\tau = 1 \times 10^{-6}$ s, after assessing the independence of the solution on the parameters.

The results are compared in terms of collection efficiency β , i.e., the fraction of the dispersed phase freestream mass impacting at a given location on the surface. The collection efficiency is computed iteratively by solving the particle tracking problem multiple times and progressively refining the seeded cloud until statistical convergence is achieved. Convergence is assumed when the ℓ^2 norm of the difference between the collection-efficiency distributions obtained in two consecutive iterations falls below 10^{-6} . This iterative strategy, coupled with the present non-inertial formulation, leads to an efficient solution procedure, since the computational cost scales only with the number of particles effectively introduced to reach convergence of the target quantity. By contrast, a fully-inertial approach based on grid tracking would also require interpolation of the grid motion during the particle substeps, introducing an additional computational overhead at each problem iteration to compute the collection efficiency that does not carry information for statistical convergence. For the present test case, the resulting time to solution is approximately one order of magnitude lower than that of the corresponding fully-inertial approach with continuous grid tracking during the dispersed phase sub-stepping.

The collection efficiency at the minimum, mean, and maximum angle of attack, computed in the Lagrangian solution between t_k and t_{k+1} , is reported in Figure 5, and compared with the unsteady Eulerian results computed by [9]. The comparison shows good agreement over the full impingement region at all reported phases of the pitching cycle. The location of the impingement limits and the peak collection efficiency are reproduced consistently, with only minor local differences in the maximum values, and the hysteresis at 5° is correctly captured during the upstroke and downstroke motion.

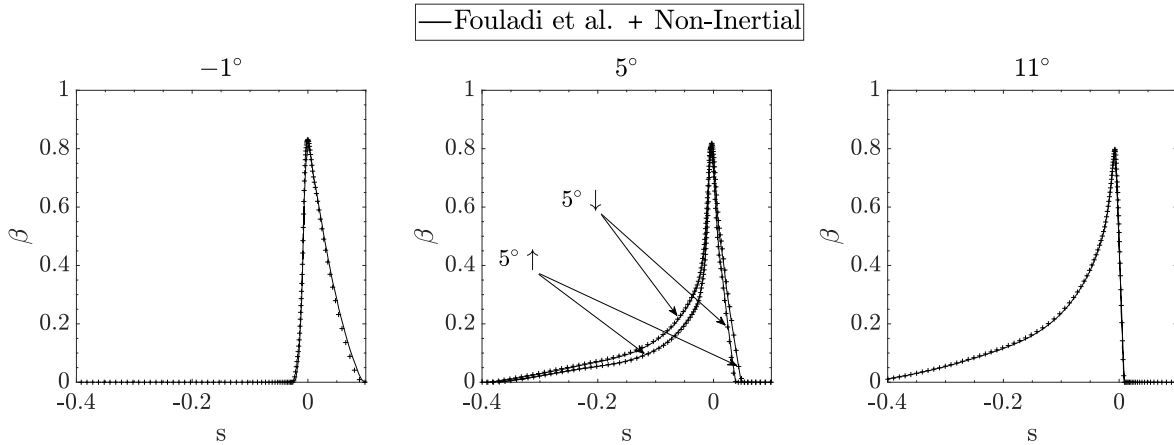


Figure 5: Collection efficiency on the pitching SC2110 airfoil at the minimum, average, and maximum angles of attack, compared with the unsteady Eulerian results of [9].

5. Conclusions

A local non-inertial Lagrangian particle tracking formulation has been presented for one-way coupled dispersed-phase transport on moving grids. The method advances the particle dynamics in the inertial frame while updating particle positions on a mesh frozen over each carrying-phase time interval. In this way, intermediate interpolation of the grid motion during particle substepping is avoided, and particle localisation can be performed with the same algorithms adopted for steady meshes.

The proposed formulation has been verified on prescribed test cases representative of moving-grid applications. The results show that the method recovers the expected convergence behaviour and provides accurate particle trajectories while retaining the computational advantages of a frozen-grid treatment. This is achieved without requiring the grid motion to be prescribed analytically at the particle-substep

level.

The method was then validated on the pitching SC2110 airfoil test case through the computation of a time-resolved collection-efficiency distribution in a moving-grid setting, showing that the present formulation can be applied to unsteady impingement problems of practical interest. The predicted collection-efficiency distributions at representative phases of the oscillation show good agreement with the reference results from the literature, correctly reproducing the impingement limits, the peak values, and the hysteresis between upstroke and downstroke conditions. This result is of particular interest because unsteady collection-efficiency predictions on moving grids are typically addressed with Eulerian formulations, which are preferred, when possible, in time-dependent applications for their computational efficiency. In combination with the iterative computation of the collection efficiency, the proposed formulation also provides a significant reduction in time to a statistically converged solution with respect to a fully-inertial tracking approach.

These features make the approach attractive for particle tracking on moving meshes, where reducing the cost and complexity associated with mesh motion is a primary requirement. Although the present paper has focused on rigid grid motion, the formulation can be naturally extended to deforming-grid configurations through a more general local kinematic description.

Acknowledgements

This study was carried out within the Space It Up project funded by the Italian Space Agency, ASI, and the Ministry of University and Research, MUR, under contract n. 2024-5-E.0 - CUP n. I53D24000060005.

References

- [1] Cristhian N. Aliaga, Martin S. Aubé, Guido S. Baruzzi, and Wagdi G. Habashi. Fensap-ice-unsteady: Unified in-flight icing simulation methodology for aircraft, rotorcraft, and jet engines. *Journal of Aircraft*, 48(1):119–126, 2011.
- [2] Chen Xi and Zhao Qi-Jun. Numerical simulations for ice accretion on rotors using new three-dimensional icing model. *Journal of Aircraft*, 54(4):1428–1442, 2017.
- [3] Xi Chen, Qijun Zhao, and George Barakos. Numerical analysis of aerodynamic characteristics of iced rotor in forward flight. *AIAA Journal*, 57(4):1523–1537, 2019.
- [4] Soonho Shon, Yoonpyo Hong, Younghyo Kim, Rho Shin Myong, and Kwanjung Yee. Icing characteristics of a wing behind a propeller wake. *AIAA Journal*, 63(8):3321–3339, 2025.
- [5] Graham B. Macpherson, Niklas Nordin, and Henry G. Weller. Particle tracking in unstructured, arbitrary polyhedral meshes for use in cfd and molecular dynamics. *Communications in Numerical Methods in Engineering*, 25(3):263–273, 2009.
- [6] Myles Morelli, Tommaso Bellosta, and Alberto Guardone. Lagrangian particle tracking in sliding mesh applicable for rotorcraft icing applications. In *45th European Rotorcraft Forum, Warsaw*, pages 7–20, 2019.
- [7] Tommaso Bellosta, Giacomo Baldan, Giuseppe Sirianni, and Alberto Guardone. Lagrangian and eulerian algorithms for water droplets in in-flight ice accretion. *Journal of Computational and Applied Mathematics*, 429:115230, 2023.
- [8] Tonja Reinert, Robert J. Flemming, Robert Narducci, and Roger J. Aubert. Oscillating airfoil icing tests in the nasa glenn research center icing research tunnel. In *SAE 2011 International Conference on Aircraft and Engine Icing and Ground Deicing*. SAE International, June 2011.
- [9] Habibollah Fouladi, Cristhian N. Aliaga, and Wagdi G. Habashi. Quasi-unsteady icing simulation of an oscillating airfoil. In *7th AIAA Atmospheric and Space Environments Conference*, 2015.