Stefano Oliani[1]
Department of Engineering,
University of Ferrara,
Via Saragat 1,
Ferrara 44122, Italy
e-mail: stefano.oliani@unife.it

Riccardo Friso
Department of Engineering,
University of Ferrara,
Via Saragat 1,
Ferrara 44122, Italy
e-mail: riccardo.friso@unife.it

Nicola Casari
Department of Engineering,
University of Ferrara,
Via Saragat 1,
Ferrara 44122, Italy
e-mail: nicola.casari@unife.it

Michele Pinelli
Department of Engineering,
University of Ferrara,
Via Saragat 1,
Ferrara 44122, Italy
e-mail: michele.pinelli@unife.it

Alessio Suman
Department of Engineering,
University of Ferrara,
Via Saragat 1,
Ferrara 44122, Italy
e-mail: alessio.suman@unife.it

Mauro Carnevale
Department of Mechanical Engineering,
University of Bath,
Bath BA2 7AY, UK
e-mail: mc2497@bath.ac.uk

Progresses in Particle-Laden Flows Simulations in Multistage Turbomachinery With OpenFOAM

Numerical simulations of particle-laden flows have received growing attention in the last decade, due to the broad spectrum of industrial applications in which discrete phases prediction is of interest. Among these, ingestion of particles by turbomachinery is an area that is seeing vivid research and studies. The most common technique to tackle this kind of problem is the Eulerian–Lagrangian method, in which individual particles are tracked inside the domain. At the same time, in multistage turbomachinery simulation interfaces are needed to couple the flow solution in adjacent domains in relative motion. In this work, an open-source extension for Lagrangian simulations in multistage rotating machines is presented in the foam-extend environment. First, a thorough discussion of the implementation is presented, with particular emphasis on particle passage through general grid interfaces (GGI) and mixing planes. Moreover, a mass-conservative particle redistribution technique is described, as such a property is requested at interfaces between multiple reference frame (MRF). The peculiarities of the algorithm are then shown in a relevant test case. Eventually, three turbomachinery applications are presented, with growing complexity, to show the capabilities of the numerical code in real-life applications. Simulation results in terms of erosion and impacts on aerodynamic surfaces are also presented as examples of possible parameters of interest in particle-laden flow computations.
[DOI: 10.1115/1.4054076]

Keyword: computational fluid dynamics (CFD)

1 Introduction

Liquid and gaseous flows are often dispersed with particles, arousing the interest of the research community toward multiphase and multispecies flows in the last decades [1,2]. Among these, due to its diffusion and engineering relevance, numerical modeling of particle-laden flows in multistage turbomachinery applications plays an important role. In this track, active research fields concern, to cite a few, droplet trajectory analysis during online washing of compressors [3] or wet steam turbines operations [4], and performance degradation due to erosion and deposition of dust and soot flowing into the gas path [5–7]. To tackle this kind of problem, Lagrangian tracking algorithms are undoubtedly the most commonly employed due to their physical soundness and easiness of use. The major shortcoming of this technique is that it is computationally demanding since a large number of particles have to be tracked. In the case of simulation of large and complex rotating machinery domains, this could translate into overwhelming computational cost and unreasonable deployment of resources. To overcome the issue of high computational times in single-phase flow computations, in the last few decades, several techniques have been developed to simplify calculations without missing the fundamental aspects. Noteworthy examples are multiple reference frame (MRF), mixing plane interfaces, sliding mesh simulations with reduced periodicity and harmonic balance methods, just to mention a few. It is therefore clear that a modern Lagrangian tracking algorithm must be able to accommodate such features and should aim to completely fit into the computational frameworks in which turbomachinery simulations are recently carried out. Besides, it should be computationally efficient and able to run in parallel in order not to overly burden the computation. Ideally, the Lagrangian tracking routine should not be a stand-alone part of the solver, being instead completely integrated into the employed code. On this purpose, *foam-extend*, the community-driven branch of the opensource software OPENFOAM (OF) [8], represents a suitable framework for delivering such numerical tools. Moreover, foam-extend includes new turbo-tools capabilities, the most important ones being the fully implicit version of general grid interfaces (GGI) and mixing planes implemented by Jasak and Beaudoin [9]. The implementation of these tools allows for steady-state simulations of rotating machines stages with the

[1] Corresponding author.

Contributed by the International Gas Turbine Institute (IGTI) of ASME for publication in the JOURNAL OF TURBOMACHINERY. Manuscript received August 23, 2021; final manuscript received March 10, 2022; published online April 19, 2022. Tech. Editor: David G. Bogard.

frozen rotor or mixing plane [10] technique. In this scenario, a new procedure needs to be defined in order to handle particle crossing based on the well-established algorithm for interpolation across nonconformal interfaces.

This subject is not novel in the literature, as several studies have been proposed exploiting similar numerical tools embedded in other commercial or in-house softwares. Ghenaiet [11] and Hamed et al. [12] studied erosion in axial flow turbine using a frozen-rotor simulation. In this regard, particles passed through the stator/rotor interface maintaining their position, while the velocity was transformed in the corresponding reference frame. The main drawback of this method is that unsteady effects are not captured, and the erosion pattern as well as the flow field depend on the clocking between the blade rows. Within the same steady-state framework, Tabakoff et al. [13] studied erosion effects in a turbine stage using a different method for interface-crossing. Although they did not employ a mixing plane for the flow, particles were circumferentially redistributed in a random manner at the stator/rotor interface. This was done to simulate the evening-out effect of reciprocal motion on the particle distribution at the stator-exit plane. More recently, Bidwell [14] used a similar technique combined with a mixing plane interface and an in-house code to study ice accretion in the E^3 turbofan engine. Similarly, Mustafa [15] simulated droplets trajectory during online washing of a multistage compressor using the commercial code CFX-TASCflow and mixing plane interfaces. Yang and Boulanger [16] performed the numerical simulation in both steady and unsteady manner of the full annulus of an axial fan and compared the results in terms of erosion, showing significant differences. In all these cases, either separate in-house codes were used to perform the flow simulation and the Lagrangian tracking or the exploited software was not capable to handle all possible types of particle–interface interactions. Zagnoli et al. [17] and Prenter et al. [18] used ANSYS FLUENT to compare steady and unsteady deposition patterns on an axial turbine stage. The authors implemented a user-defined function (UDF) to account for particle circumferential redistribution at the mixing plane interface. From these observations, it is clear that, although some relevant support for interfaces-crossing in particle-laden flows is available, there is still room for improvements in this area.

In the present work, making use of the turbo-tools utilities of foam-extend, particle-laden flows simulations are extended to multistage turbomachinery in an opensource environment. A methodology is proposed that allows for particle tracking through interfaces between zones with rotating motion and/or in multiple reference frames exploiting the GGI support. To summarize, this work aims to push forward the state of the art of particle tracking in multistage turbomachinery by developing as follows:

(1) A computationally efficient, automatic treatment for any kind of interface a particle crosses during the simulation, unified in a single simulation software. This method will be completely integrated in the numerical routine, making the best use of the already available interfaces treatments and MRF support for the continuous phase.
(2) A simple and intuitive treatment for particle circumferential redistribution at mixing plane interfaces, without any need of additional UDFs and able to ensure conservation of discrete-phase mass flowrate while handling reverse flows without any special requirement.
(3) An implementation of the above features in an opensource environment, thus making the code completely shareable, usable, and modifiable by the community, in a fully opensource philosophy.

The outline of the paper is as follows. In Sec. 2, an overview of the particle tracking algorithm is given, and the attention is focused on the trajectory computation in multiple reference frame and moving meshes cases. Section 3 is devoted to the description of the numerical technique for particles treatment at various kinds of interfaces met by particles during the simulation of a single/multistage rotating machine. In Sec. 4, results for a very simple axial annulus test-case are presented to illustrate and benchmark various features of different interface treatments. Afterward, in Sec. 5, three geometries of growing complexity are simulated to assess the code performance and reliability in complex flow fields.

Finally, it is important to underline that the methodology will be presented with specific addressing to turbomachinery applications. Nevertheless, it is quite general and suitable for any kind of simulation in which support for particles moving through interfaces between different cell zones and reference frame is needed.

2 Lagrangian Tracking Algorithm in Multiple Reference Frame and Moving Meshes

A brief summary and the most important aspects of the Lagrangian particle tracking algorithm in OF are outlined in this section, while for a thorough review the reader is referred to [19,20]. Eulerian–Lagrangian methods are sometimes referred to as noncontinuum models, since the dispersed phase is treated in a discrete way in a frame of reference that moves with the particle (Lagrangian), while the fluid phase is treated as a continuous phase in a fixed (Eulerian) frame. The Lagrangian frame is a natural way to treat particles in dilutes flow, thus making the use of these models very popular in applications like erosion and deposition in turbomachinery, aerosol transport, and spray combustion. The main drawbacks of the Lagrangian approach are the high turnaround times and the fact that it poses severe challenges in the coupling back to the fluid phase, as pointed out by Balachandar and Eton [21]. The equations of motion for the discrete phase are here reported for the sake of clarity

$$\frac{d\mathbf{x_p}}{dt} = \mathbf{v_p} \tag{1}$$

$$m_p \frac{d\mathbf{v_p}}{dt} = \mathbf{F_I} + \mathbf{F_{NI}} \tag{2}$$

where $\mathbf{F_I}$ and $\mathbf{F_{NI}}$ are the inertial and noninertial forces acting on the particle, respectively, this last one being present only if the simulation is carried out in MRF. Inertial forces can include several different actions on the particle such as drag, gravity, Saffman lift, and virtual mass. In the case of rotating machinery, $\mathbf{F_{NI}}$ incorporates centrifugal and Coriolis forces, respectively, defined as

$$\mathbf{F_{centr}} = -m_p \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{x_p} \tag{3}$$

$$\mathbf{F_{Cor}} = -2m_p \boldsymbol{\omega} \times \mathbf{v_p} \tag{4}$$

where $\boldsymbol{\omega}$ is the angular velocity of the rotating frame, which will be zero in the stator zone and equal to the shaft angular speed in rotor zone. Particles are always tracked in their actual frame, so that their trajectory is computed based on their relative velocity that corresponds to the absolute velocity if they are passing through a vane row. Equations (1) and (2) are solved iteratively either in a steady or unsteady manner, depending on whether the fluid flow field is updated or not after each Lagrangian step. To effectively track a discrete particle into the domain, it is also necessary to know the actual cell of the mesh where it resides. This is important to interpolate the values of the continuous phase at particle position to be used in the equation of motion (e.g., the fluid velocity to calculate the drag force). At each step, the Lagrangian algorithm updates the particle cell using a computationally efficient search routine, as detailed, for example, in Ref. [22].

3 Interface Treatment for Lagrangian Tracking in Multistage Simulations

In multirow turbomachinery simulations, interfaces are needed to separate flow zones in reciprocal motion. Usually, steady-state simulations are performed when fast computations are needed and

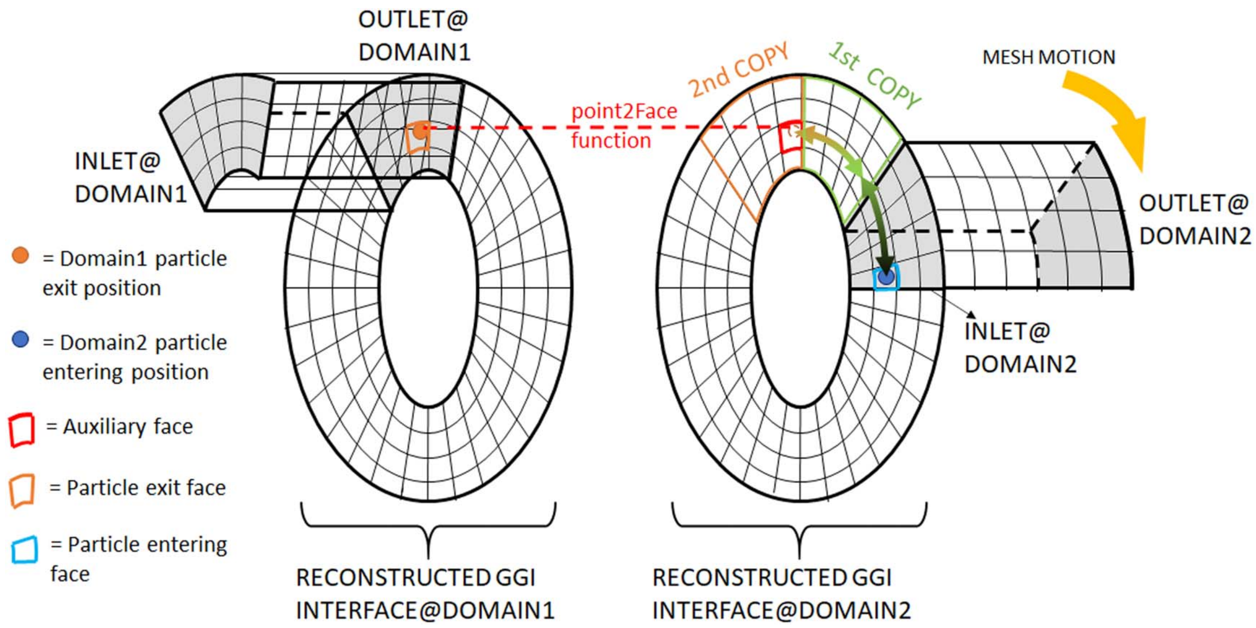

Fig. 1 Sketch of the method for particle crossing through overlapGGI interface

stator–rotor interaction effects are not important. In this case, interfaces in the fluid domain denote the passage from one reference frame to another, and particle velocity has to be updated accordingly. Conversely, when transient effects must be taken into account, dynamic meshes with sliding interfaces are often employed. GGIs were introduced to extend OpenFOAM capabilities to turbomachinery simulations [9] and are able to tackle efficiently all these cases. GGIs are used to couple different zones of the domain at the discretization level, joining multiple mesh regions into a single contiguous domain. This is done by an area-weighted interpolation through conformal or nonconformal patches, without any need for topological changes of the mesh (in opposition to attach-detach methods). Specialized versions of GGIs are used to couple different types of interface encountered in turbomachinery simulations, such as periodic boundaries, nonoverlapping sliding interfaces, and mixing planes. To perform numerical simulations of particle-laden flows in foam-extend, the support for particle tracking all the way through this kind of patches has been introduced, as described in the next paragraphs. This implementation is suitable also for parallel computations, provided that the couple of patches are kept in the same processor.

3.1 OverlapGGI Interface for Transient Computations.

OverlapGGIs are suitable for transient computations with dynamic mesh, in which one interface slides on the other. Typically, periodicity is exploited to avoid modeling the whole 360-deg domain, and inevitably nonoverlapping areas appear during the mesh motion. The outline of the method implemented for particle crossing overlapGGI is sketched in Fig. 1 for a single axial stage. In this example, DOMAIN1 is fixed while DOMAIN2 rotates and the coupled overlapGGI patches join the two domains, so that particles exiting OUTLET@DOMAIN1 must re-enter at INLET@DOMAIN2. Notice that the two interfaces (in gray) do not overlap at all, but in this case foam-extend reconstructs the whole 360-deg patches by copying the original patch an integer number of times based on the user-specified transformation. In this way, the continuous-phase fluxes interpolation can be easily done since each face addressing on the other side of the interface can be retrieved. In Fig. 1, the two domains are distanced for the sake of clarity, but the two sides of the interface share the same axial coordinate in the actual computation. Suppose that a particle leaves OUTLET@DOMAIN1 at a specific position and face. At such occurrence, the newly implemented point2Face function finds the auxiliary face on the other side of the reconstructed patch that shares the exit position with the overlapGGI interface of DOMAIN1. This function is computationally efficient since it exploits the GGI infrastructure, searching for the auxiliary face only among the few faces included into the addressing list of the exit face at DOMAIN1. Once this is done, the final particle position and cell at INLET@DOMAIN2 must be computed using the periodicity hypothesis. This is easily achieved since the face numbering is corresponding between the original interface and its copies. Ultimately, the entering position is obtained by rotating the exit position around the machine axis by an angle ϕ. All these things being considered, the new particle entering face at DOMAIN2 and the rotation angle ϕ are

$$f_{ent} = f_{aux} - \lfloor f_{aux}/nFaces \rfloor * nFaces \tag{5}$$

$$\phi = \lfloor f_{aux}/nFaces \rfloor * \theta_r \tag{6}$$

where f_{ent} and f_{aux} are respectively the entering and auxiliary face's number, $nFaces$ is the size of the receiving overlapGGI interface, while θ_r is the angular pitchwise extension of the receiving patch. The operator "$\lfloor \quad \rfloor$" denotes the floor function that rounds its argument to the nearest integer less than or equal to the argument itself. The final cell is simply the cell that owns the particle entering face (shown in blue in Fig. 1). To preserve the radial and tangential components of the velocity of the particle, the velocity itself is transformed according to the same rotation. Now, the particle is accurately located on the right face at INLET@DOMAIN2 and the tracking can proceed as usual in the second domain. It is emphasized that in the special case in which the two sides of the interface are perfectly overlapping (basic GGI implementation, as in frozen-rotor simulations), the simple application of the point2face function suffices to correctly assign the new face and cell to the particle. Moreover, if the GGI interfaces are used to couple two nonconforming periodic patches (cyclicGGI), then the particle position and velocity must be transformed according to the periodicity before applying the point2Face function. A final remark concerns the robustness of the algorithm to any kind of flow field at the interface. Indeed, the described technique is able to handle reverse flow without any particular treatment, since the roles of the sending and receiving sides are perfectly interchangeable with respect to particle motion.

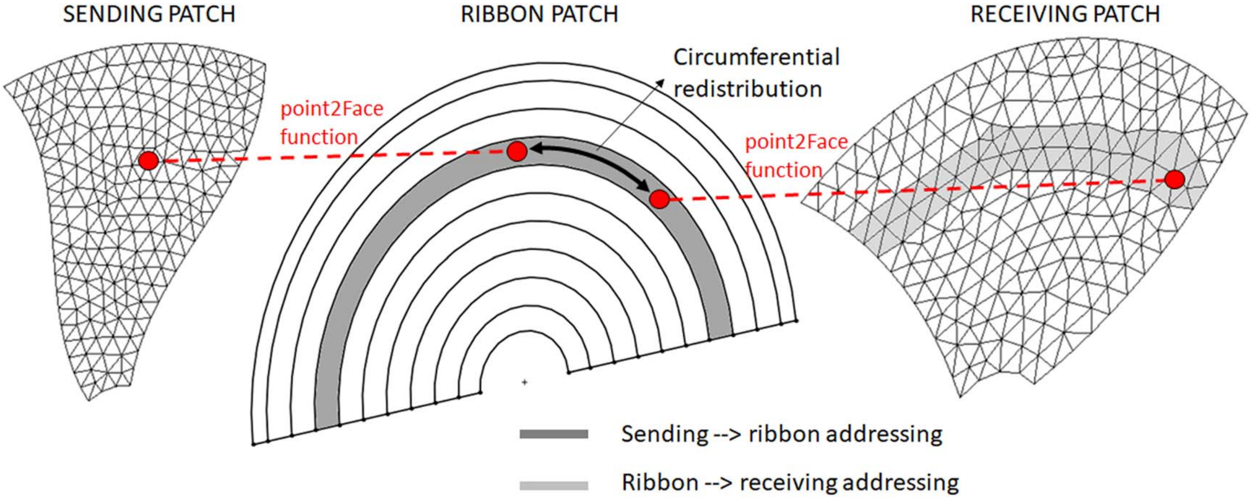
SENDING PATCH
RIBBON PATCH
RECEIVING PATCH
point2Face function
Circumferential redistribution
point2Face function
Sending --> ribbon addressing
Ribbon --> receiving addressing

Fig. 2 Sketch of the method for particle crossing through mixing plane interface

3.2 Mixing Plane Interface for Steady Simulations. The implementation of the mixing plane in foam-extend relies on two consecutive GGI interpolators. The first interpolation is done from the upstream patch to a *ribbon patch*, where the flow quantities are circumferentially averaged to remove unsteadiness due to blade wakes and passage-to-passage flow variation. To this end, the ribbon patch is divided into a 1D profile and then expanded in the circumferential direction. In this way, the averaging of the flow quantities at a specific radial location is performed. Similarly, the particles are randomly redistributed in a circumferential sense to simulate the evening-out effect of the reciprocal motion of stator and rotor. At the mixing plane, meshes do not usually offer full tangential coverage, see, for example, the unstructured and nonconformal sending and receiving patches in Fig. 2. When a particle crosses the sending patch, the point2Face function is used to find on which ribbon of the ribbon patch the particle must be located. This searching algorithm is very efficient since each face of the sending patch addresses very few ribbons on the ribbon patch. On the ribbon patch, particle position is converted into a cylindrical coordinates system and its circumferential coordinate is randomly reassigned in the pitchwise range spanned by the addressing faces of the actual ribbon, as indicated by the black arrow in Fig. 2. This is done to ensure that the particles are relocated inside the receiving domain. Notice also that, since the tangential boundaries of the interface can have an arbitrary shape provided they span a fixed angular sector, the pitchwise bounds must be recalculated for each ribbon, because they can vary from one radial location to another. Once that the particle has been redistributed, a second call of the point2Face function allows to find the entering face in the receiving domain. Particle position is then projected back into the original coordinate system and its velocity is rotated accordingly to the rotational redistribution angle to preserve the radial and tangential components. Usually, the mixing plane interface denotes also the passage from one frame of reference to another, so particle velocity is eventually updated accordingly. It should be mentioned that this method is not conservative with respect to the mass flowrate of the discrete phase because the sending and receiving domains typically have different angular extensions. Indeed, for

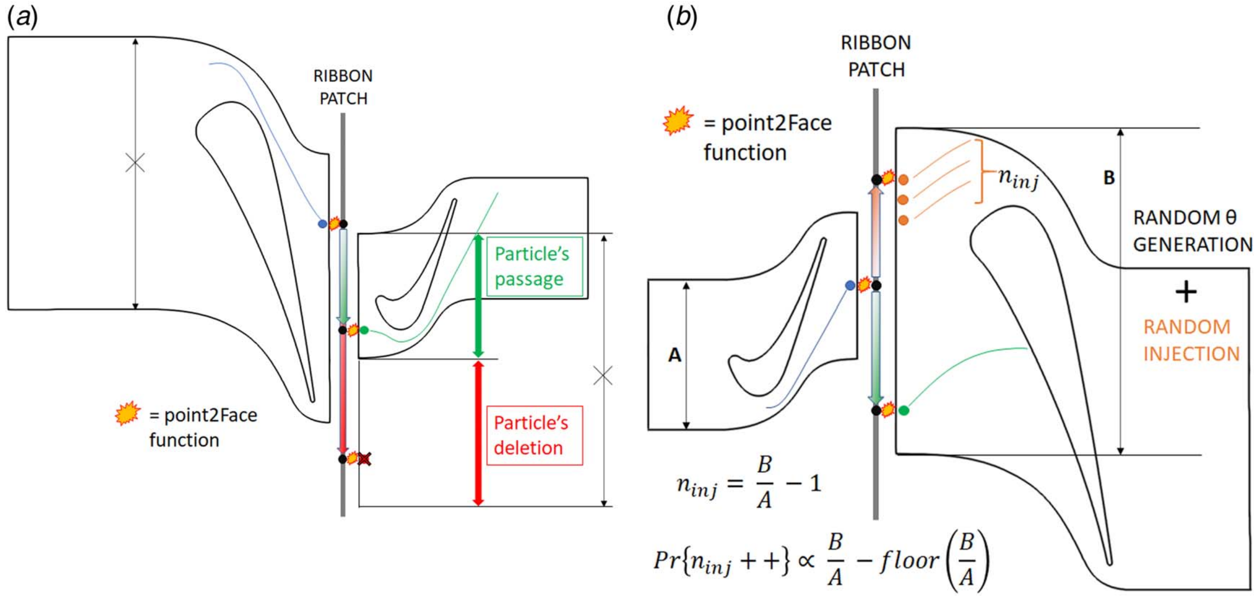

Fig. 3 Mixing plane treatment to preserve the full-annulus mass flowrate of the discrete phase across the interface: (*a*) the receiving side has a pitchwise extent smaller than the sending one and (*b*) the receiving side has a pitchwise extent greater than the upstream one

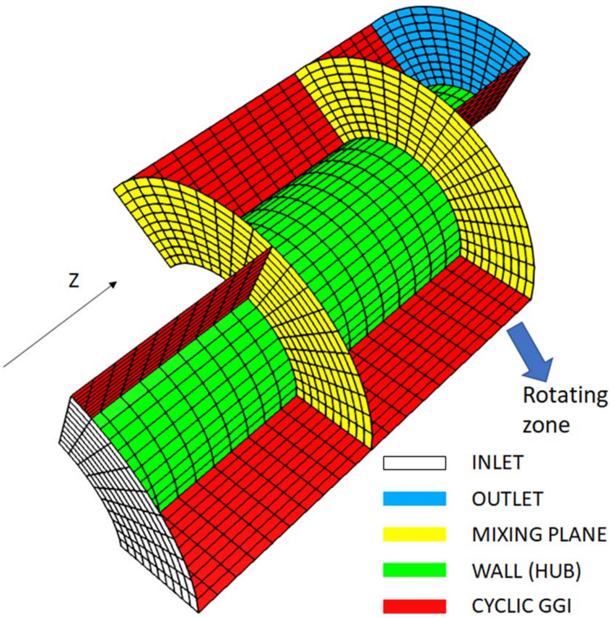

Fig. 4 Test-case domain and relative patches

the specific case of a receiving domain that spans twice the angular sector of the sending domain: in this case for each particle crossing the sending patch, two should be injected into the receiving domain. To ensure mass conservation in this full-annulus sense, the passage algorithm has been modified adding or deleting particles based on the pitchwise coverage of the two domains. Figure 3(*a*) shows two cylindrical sections at a fixed radius of an axial gas turbine unwrapped onto a plane. Focusing on Fig. 3(*a*), the case where the particle exits from the bigger side of the interface and enters into the smaller side. In this case, denoting with $\Delta\theta$ the angular extension of a patch and with θ_{min} and θ_{max} the bounds of its angular sector, the random circumferential position on the ribbon patch is generated into the following range:

$$\theta_{min,r} \le \phi_{random} \le \theta_{min,r} + \Delta\theta_s \tag{7}$$

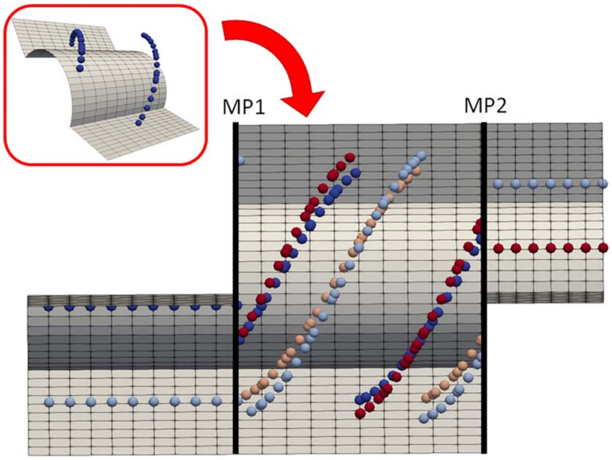

Fig. 5 Top and side views of particles trajectory showing injection, deletion, and redistribution at the mixing plane interfaces

where the subscripts s and r stand for sending and receiving, respectively. Then, the following condition is applied: if ϕ_{random} is between $\theta_{min,s}$ and $\theta_{max,s}$, the particle crosses the interface; while if $\phi_{random} \ge \theta_{max,s}$, the particle is deleted.

Figure 3(*a*) shows schematically the procedure. Conversely, in the case the particle passes from the rotor domain to the stator of the successive stage of an axial turbine, the situation is that depicted in Fig. 3(*b*). The first thing is then evaluating how many particles must be injected, n_{inj}, depending on the ratio B/A (see Fig. 3(*b*)). Considering that the particle will necessarily cross the interface, being randomly redistributed and continuing on its trajectory, one has simply $n_{inj} = B/A - 1$. Nevertheless, the ratio B/A will not, in general, be an integer, so a Metropolis-Hastings rejection algorithm is introduced. First, the decimal part of the ratio is computed simply as $\epsilon = B/A - floor(B/A)$, and then a random number is generated in the interval [0, 1]. If the random number is lower than ϵ, another particle is added to the injection at the interface so that, statistically, mass conservation is achieved. Every injected particle is a copy of the original one, but the injection is randomized in the circumferential direction and the velocity is rotated consequently, exactly as in the previous case. The algorithm is completely automatic, meaning

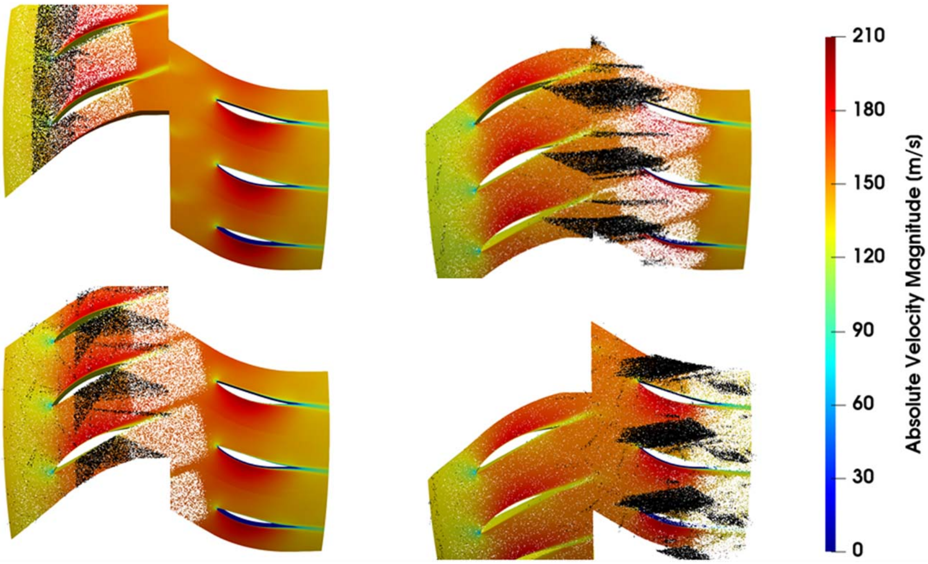

Fig. 6 Evolution of particles inside the domain (three passages are reported for the sake of clarity). White particles have a diameter of 2 μm, while black ones have a diameter of 165 μm. The fluid slice is colored by velocity magnitude in the absolute frame.

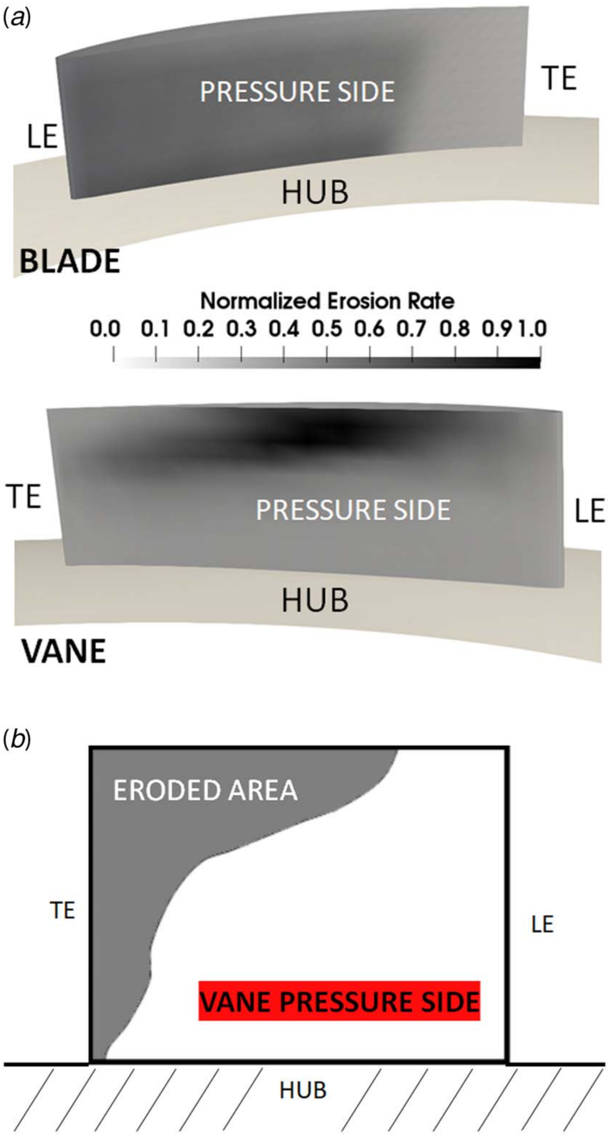

Fig. 7 **(*a*) Normalized erosion rates on compressor blade and vane pressure sides and (*b*) Erosion on a stator vane adapted from Ref. [29]**

that it switches between the two cases by computing each time the pitchwise range of the sending and receiving patches and evaluating which one is bigger. In this manner, there is not a preferential crossing direction of the interface and eventual reverse flows are easily handled. In the next section, a proof of concept will be presented in order to distill down and benchmark the various aspects of the method.

4 Proof of Concept: Axial Annulus Domain

To test the correct implementation of the aforementioned features of the code and to show how they work in practice, a simulation in a very simple geometry has been run. The domain is shown in Fig. 4 and is composed of three annular sections with varying periodicity, separated by two mixing plane interfaces, while the only walls present are the hub and shroud. The central section of the domain rotates about the z-axis and the flow field herein is solved in a moving reference frame. At the inlet, air is injected with an absolute velocity of 1 m/s aligned with the rotation axis and the flow is considered to be laminar. The angular extension of the rotating region is twice the one of the inlet and outlet. Two particles with a diameter of 10 μm are injected from the inlet at the equilibrium velocity with the fluid. The only forces acting on the particles are sphere drag and noninertial forces in the relative frame of reference (Coriolis and centrifugal). Finally, a *cyclicGGI* boundary condition has been imposed on the periodic patches of the three zones. Particle trajectories are reported in Fig. 5. The two particles are circumferentially redistributed across the upstream mixing plane and transferred into the rotating annulus zone. Additionally, another two particles are injected at a random position, because the rotating zone covers twice the angular span with respect to the stationary ones.

The trajectory becomes discontinuous at the interface, since the velocity is switched from absolute to relative. In the rotating frame, particles exhibit helical trajectories, as can be also seen in the close-up on the left, representing the trajectory of an individual particle. This simple test benchmarks the good implementation of MRF support. The cyclicGGIs correctly transfer particles between the coupled periodic patches, also rotating the velocity by the transformation angle. Finally, at the second mixing plane interface, particles are once again redistributed and two of them are deleted, ensuring (statistically) the conservation of the discrete-phase mass flowrate. Eventually, particle velocity is transformed back into the absolute reference frame when crossing the downstream mixing plane.

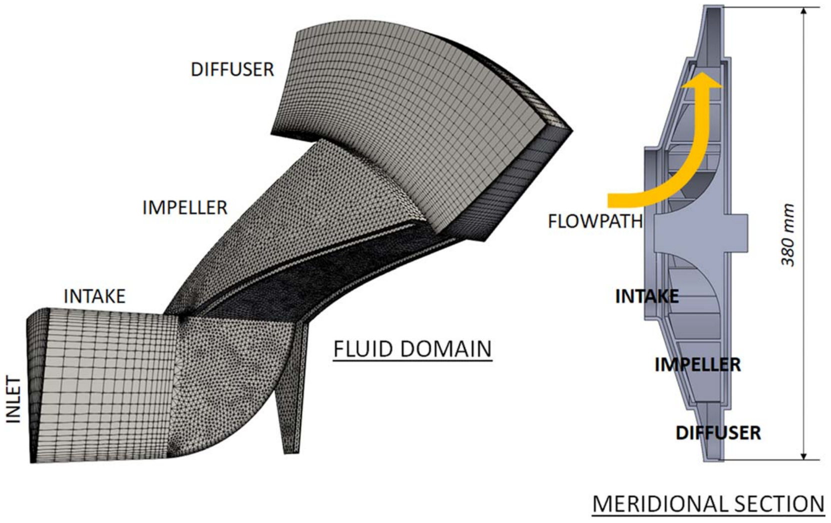

Fig. 8 **Radial fan computational domain**

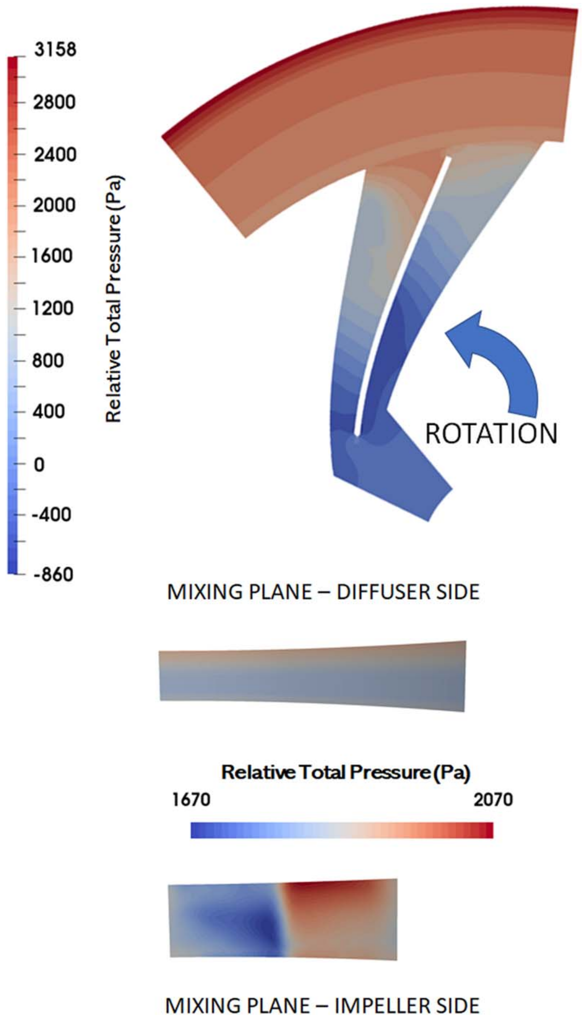
3158
2800
2400
2000
1600
1200
800
400
0
-400
-860
Relative Total Pressure (Pa)
ROTATION
MIXING PLANE – DIFFUSER SIDE
Relative Total Pressure (Pa)
1670
2070
MIXING PLANE – IMPELLER SIDE

Fig. 9 Pressure distribution on the fan blade and illustration of the mixing plane average

5 Case-Studies Results

In this section, three turbomachinery applications with growing geometrical complexity are presented. It is underlined that the aim is just to show the correct behavior of the particles when they cross different types of interfaces, while providing possible real applications of the techniques developed in this paper. For each simulation, the convergence of the fluid flow has been achieved with second-order schemes in space (and also in time for transient computations). The employed solvers have been largely validated in previous works [23–25]. Additionally, the accuracy and efficiency of the proposed method are not impaired in any way by the continuous-phase solution. Therefore, validation against experimental results are not presented here not to bother the reader with overabundant information. To remedy this lack, references to detailed results from other works are given where possible. To model turbulence, the $k-\omega$ SST model has been used with an automatic wall treatment with a blending function between low Reynolds and wall function treatments depending on the local y^+. The stochastic behavior of particle trajectories due to turbulence is accounted for by using the discrete random walk model of Gosman and Ioannides [26]. Throughout the three simulations, one-way coupling with the continuous phase is assumed. This simplification is acceptable in most turbomachinery applications although, strictly speaking, one-way coupling is on a sound basis only in cases of low volume fractions of the dispersed phase ($<10^{-6}$), as shown in the diagram by Elghobashi [27]. No thermal, chemical, collision, or surface modeling has been used in the simulations. Furthermore, the only force acting on the particle is a spherical drag, calculated as

$$\mathbf{F_D} = \frac{18\mu}{\rho_p d_p^2}\frac{C_D \mathrm{Re}_p}{24}(\mathbf{v}-\mathbf{v_p}) \tag{8}$$

$$\mathrm{Re}_p = \frac{\rho d_p |\mathbf{v}-\mathbf{v_p}|}{\mu} \tag{9}$$

where μ is the dynamic viscosity, C_D is the drag coefficient calculated using the formula of Schiller and Naumann [28], and Re_p is the particle Reynolds number.

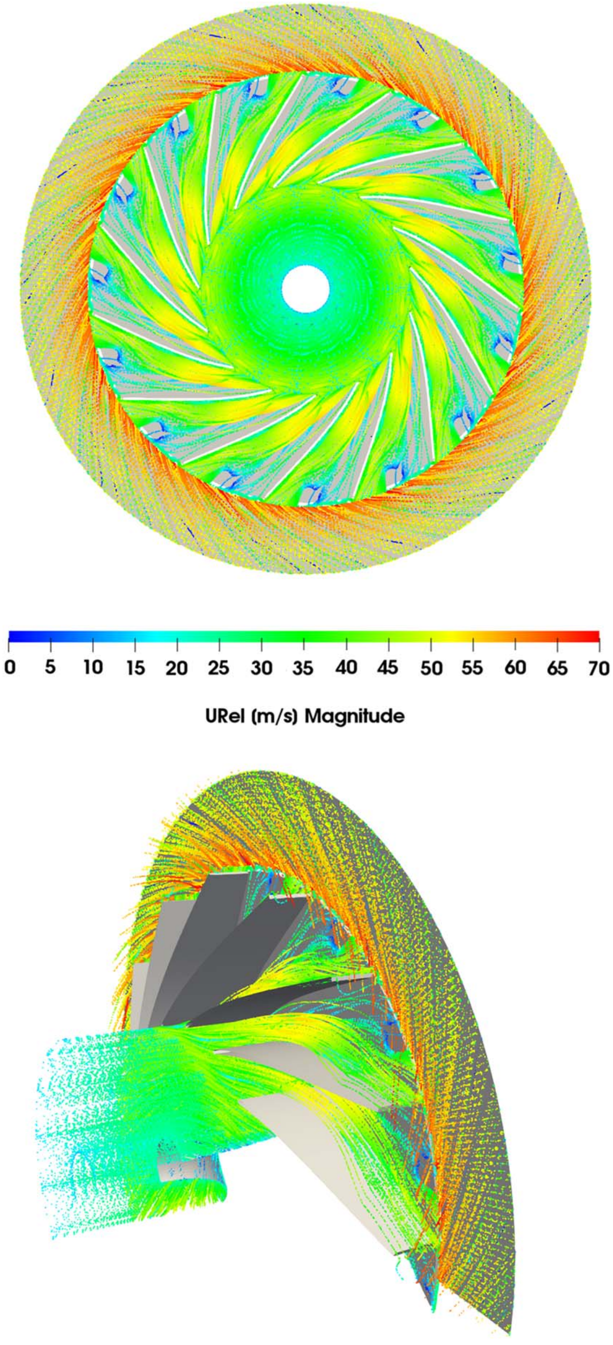

Fig. 10 Frontal and side views of particles trajectory inside the radial fan

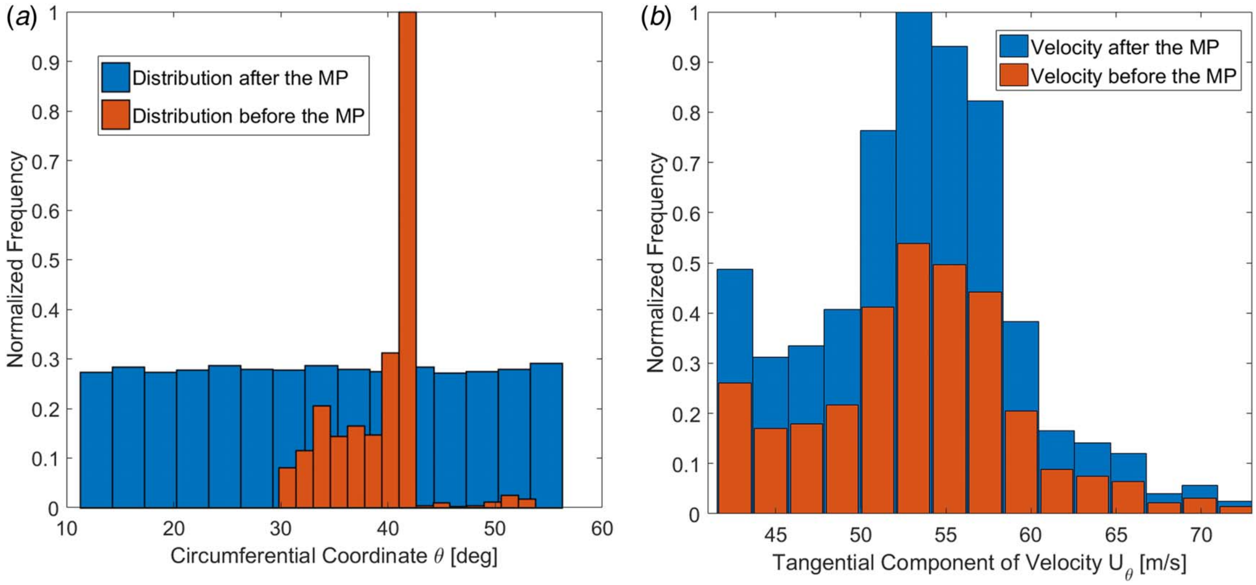

Fig. 11 Statistical analysis of particle redistribution through the mixing plane interface between the impeller and the diffuser. Particle velocity is preserved, as shown by the tangential velocity distribution in the second figure: (*a*) Circumferential redistribution and (*b*) Tangential velocity preservation.

5.1 Erosion in a Reduced-Span Axial Compressor. The first application presented shows the unsteady erosion of a reduced-span axial compressor. One blade passage per row has been simulated exploiting the periodicity, while the mesh is composed of 42,000 elements, allowing to perform an unsteady simulation in a small turnaround time. The density-based library of solvers by Borm et al. [23] has been endowed with the Lagrangian tracking support and used for the simulation. At the inlet, ambient conditions have been set, while at the outlet a static pressure of 1 atm has been imposed. The shaft rotational speed is 8140 rpm. The aim here is to show the correct behavior of the overlapGGI interface treatment, as reported in Fig. 1, in a dynamic mesh transient simulation. Sand ingestion has been simulated, injecting at domain inlet 40,000 particles with diameters of 2 μm and 165 μm (20,000 each). These two diameters have been chosen to show the great difference in particle trajectory with changing diameter. Moreover, an equivalent diameter of 165 μm was used in Balan and Tabakoff experiment [29], so the results can be compared. Evolution of the discrete phase in time is reported in Fig. 6, where the instantaneous position of particles is depicted at four instants, equally spaced by one blade passing period ΔT. It is evident at a first glance that 2 μm particles (in white) are able to follow the flow path much better than larger-sized particles (in black). Three blade passages are reported for the sake of clarity, to highlight that the strongly nonuniform discrete-phase distribution at the exit of the first blade row is maintained across the interface. Moreover, particles are correctly transferred also in nonoverlapping areas of the GGI, accommodating arbitrary reciprocal positions of the two cascades. Particles are tracked in the absolute reference frame, while wall relative velocity is used to

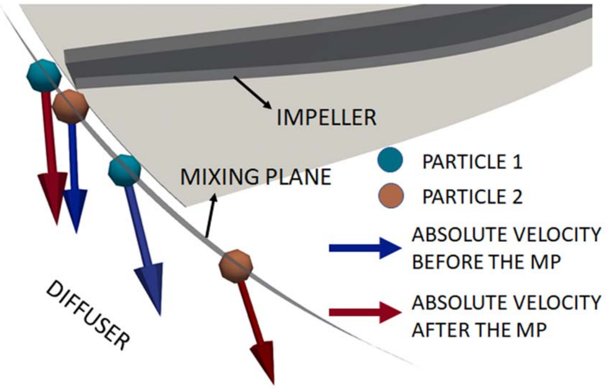
IMPELLER
MIXING PLANE
DIFFUSER
PARTICLE 1
PARTICLE 2
ABSOLUTE VELOCITY BEFORE THE MP
ABSOLUTE VELOCITY AFTER THE MP

Fig. 12 Detail of two particles crossing the mixing plane with respective velocity vector rotation

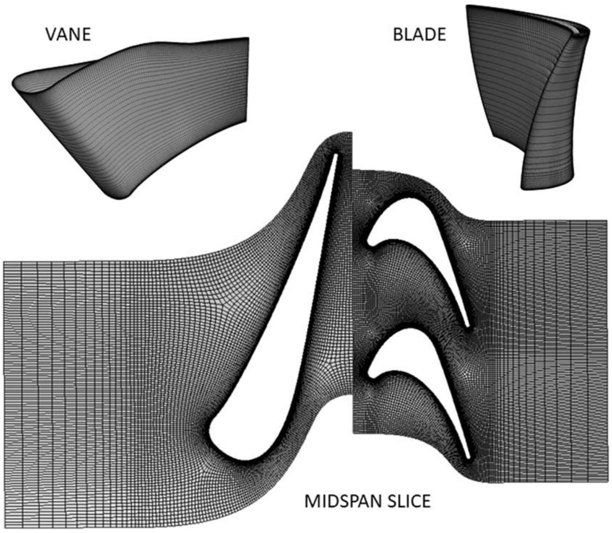

Fig. 13 E^3 axial turbine computational domain

Table 1 Boundary conditions for the turbine impacts case

	Quantity	Value
Inlet	Total pressure	13.24 bar
	Total temperature	1633 K
Walls	Temperature	1100 K
Outlet	Pressure	2.754 bar
Rotor	Rotational velocity	12,624 rpm

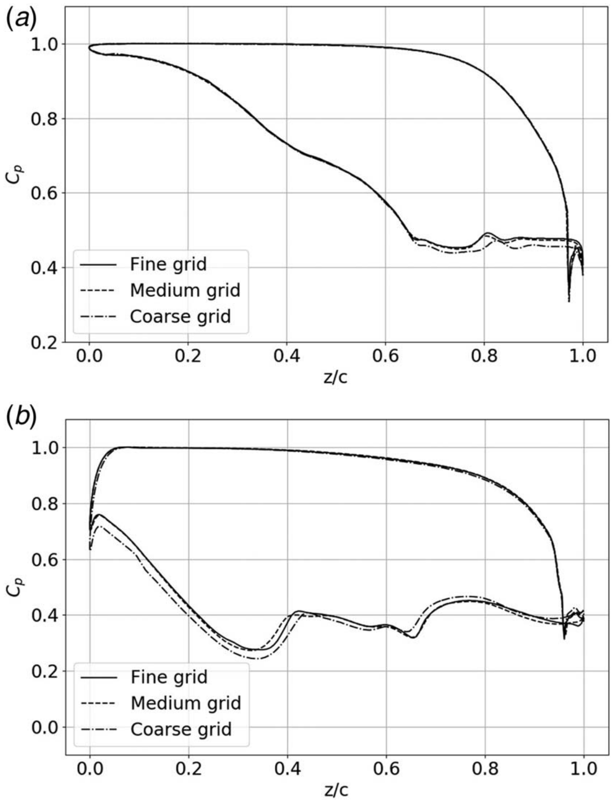

Fig. 14 Grid sensitivity results by means of C_p evolution at the mid-span surface: (*a*) vane C_p evolution at the mid-span and (*b*) blade C_p evolution at the mid-span

account for particle–wall interaction. Rebound has a prominent effect on larger-sized particles, since their impact on blade pressure side changes the angle at which they approach the vane row (see Fig. 6(*b*)). For simplicity, an elastic rebound has been hypothesized, while the Finnie model [30] has been used to estimate the erosion rate of the aerodynamic surfaces. The obtained erosion rate on blade and vane pressure sides, normalized by its maximum value, is shown in Fig. 7(*a*). It can be noticed that blade pressure side is uniformly eroded, apart from the trailing edge zone, where large-sized particles do not impact. A small amount of particles rebound on the pressure side toward the suction side of the successive blade, so that little erosion has been found on the surface of the latter. On the other hand, particles tend to migrate toward vane tip around mid-chord, where the erosion peak is located. No significant impacts have been found on vane suction side. The aforementioned results agree with Balan experimental observations [29], as illustrated by the comparison with Fig. 7(*b*). In the figure, the erosion on the pressure side of a stator vane is reported after the compressor ingested a total of 25 kg over a period of 605 s. It can be noticed that the erosion peak is located on the pressure side, approximately at midchord near the vane tip. At midspan and on the suction side of the blade no significant erosion was found, similarly to the results obtained in the numerical simulation.

5.2 Trajectory Analysis in a Radial Fan. The second application presented pertains to the trajectory analysis in a radial fan processing ambient air. On the right of Fig. 8, a meridional plane section of the CAD geometry is shown, while on the left the computational fluid domain is represented. The domain is composed of three zones: the intake, the impeller, and the vaneless diffuser, for a total of 325,000 mesh elements. The simulation has been run out with the solver *MRFSimpleFoam* that allows for steady-state simulations of incompressible flows with MRF support. The impeller is

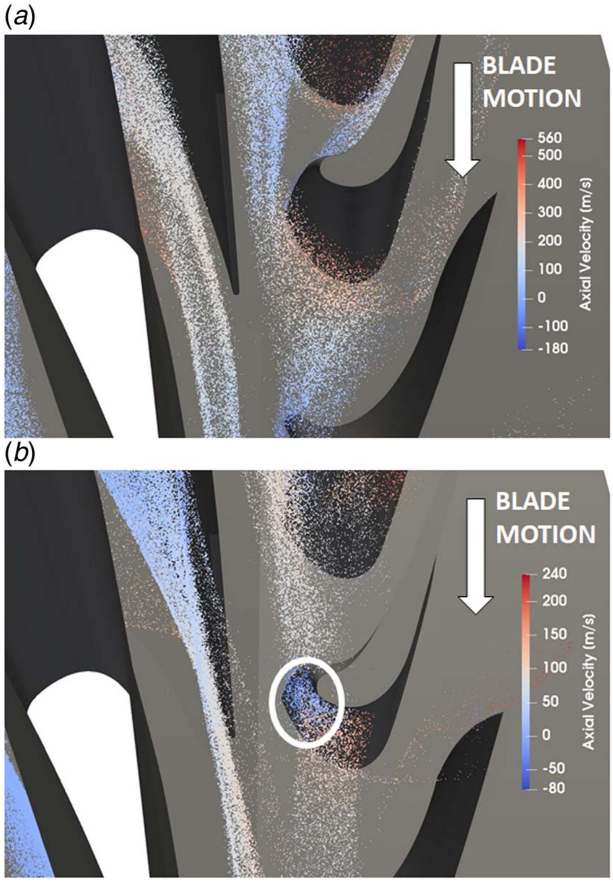

Fig. 15 Detail of particles trajectory near blade leading edge for a couple of diameters. The two axial velocity scales highlight the stronger acceleration undergone by smaller particles in the statoric row. The white circle indicates particles that are flowing back toward the vane: (*a*) 1 μm particles and (*b*) 32 μm particles.

solved in a relative frame rotating with shaft angular speed, whereas two mixing plane interfaces are used to couple different zones in relative motion. No volute is present after the diffuser since in the original geometry the total pressure rise is split between two stages, so the choice here is to simulate only the first stage. This also simplifies the simulation because domain periodicity can be exploited, whereas the presence of a variable-section volute after the diffuser would not allow such simplification. Specifically, the intake and the impeller have the same periodicity of 24 deg (corresponding to a single-passage of an impeller with a blade count of 15). Due to the presence of the mixing plane, and since the diffuser is vaneless, the flow field inside it is axisymmetric and the periodicity could have been arbitrarily chosen (e.g., equal to the other two zones). Instead, it has been imposed equal to 45 deg, with the specific aim of showing injection of particles at a mixing plane in a situation analogous to the one of Fig. 3(*b*). The mesh is not conformal at the interfaces, since a structured mesh has been used for the intake and

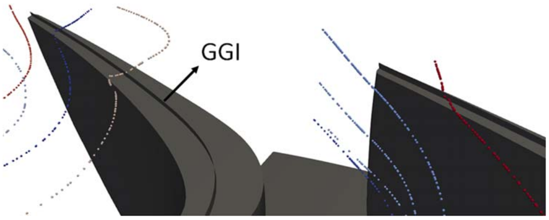

Fig. 16 Detail of particles crossing blade tip GGI

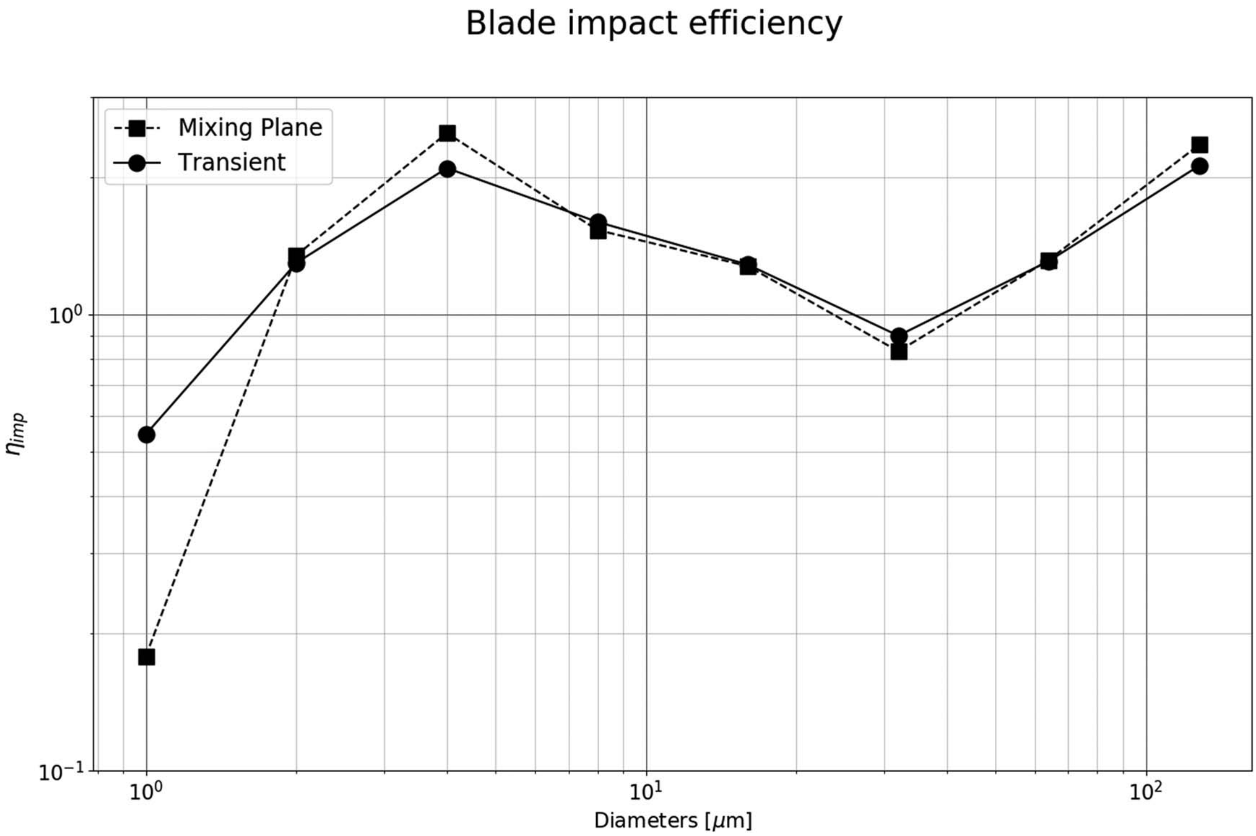

Fig. 17 Impact efficiency for various diameters on the turbine blade surface

the diffuser, while a hybrid, unstructured tetrahedral mesh with inflation at the walls has been used for the impeller. The mesh was generated using ANSYS MESHING module. The fan has been simulated at his design point, corresponding to ambient conditions at the inlet, a processed mass flowrate of 0.0486 m^3/s and an impeller rotational speed of 5140 rpm. The resulting relative total pressure is shown in Fig. 9 at the midspan of the blade and the diffuser, and also on the two sides of the mixing plane interface. Here, the strongly nonuniform pressure distribution at the impeller exit is smoothed out due to the circumferential average operated by the mixing plane to provide inflow conditions for the diffuser.

Moving to the discrete-phase calculation, particles having ARD properties with a diameter of 10 μm were injected at the inlet of the domain once the flow field achieved convergence. Trajectories were recorded for one hundred particles and reported in Fig. 10. The trajectories are computed by integrating the absolute velocity in the intake and diffuser zones, and the relative velocity in the impeller frame, so their derivative is not continuous through interfaces. The presence of a separation bubble on the suction side of the blade induces a strongly nonuniform particle concentration at the impeller exit, as can be clearly seen in the low-velocity area of the trajectories in the frontal view of Fig. 10. Nonetheless, the mixing plane correctly redistributes the particles and a uniform concentration appears immediately after the interface, while individual particle velocities are maintained and transformed in the absolute frame. The total particles count after the interface was of 185, while the angular span ratio of the diffuser and the impeller is 1.875, thus meaning that the random injection of particles at the interface to preserve the mass flowrate is correctly implemented. A statistical analysis of particle transfer across the diffuser-impeller mixing plane is also reported to further highlight the methodology. Figure 11(*a*) shows the circumferential position of particles before and after the interface. To obtain it, the circumferential coordinate of particles position was registered and the pitchwise extension on the two sides of the interface was divided into 15 bins. The number of particles pertaining to each bin was then calculated and divided by the maximum value to obtain the normalized frequency plotted in the figure. The distribution before the mixing plane is strongly nonuniform along the impeller pitch, exhibiting a marked peak at a location corresponding to the blade suction side (see also Fig. 10). On the other hand, a uniform circumferential distribution appears immediately after the mixing plane due to the random redistribution. The analysis has been pushed forward focusing also on the velocity of the particles. Since the absolute velocity vector is rotated by an angle corresponding to the circumferential coordinate reassignment, the magnitude of the vector does not change across the interface. This can be easily seen by calculating, for example, the circumferential component of particle velocity and post-processing it in an analogous way to Fig. 11(*a*). The results are reported in Fig. 11(*b*) and show that the tangential velocity distribution is unchanged across the interface. Nevertheless, the velocity vector is rotated, as illustrated in Fig. 12 for two different particles crossing the interface. In the figure, the spheres denote different particles, whereas the vector represents the absolute velocity. Velocity vectors before and after the mixing plane are reported. At last, the correct implementation of the injection mechanism at the interface can be verified by observing Fig. 11(*b*). Indeed, for each tangential velocity bin, the ratio of the frequencies between the bars is equal to the ratio of the angular spans of the impeller and diffuser sides of the interface (1.875).

5.3 Impact Analysis on E^3 HPT First Stage.

The last application concerns the impact analysis in the first stage of the high pressure turbine (HPT) of a E^3 aeroderivative engine. A midspan slice of the mesh, alongside with the vane and blade surfaces, is shown in Fig. 13, while for the rotational velocity and the boundary conditions the reader is referred to Table 1. A more detailed analysis of this case can be found in Oliani et al. [24], where the authors presented also the influence of different mixing plane average types on blade impacts. The grid is composed of nearly 3.3 million elements and has been selected after a grid sensitivity analysis. The results of the grid sensitivity in terms of the pressure coefficient on the vane and the blade are reported in Fig. 14. The grid employed in the calculations is the "medium" one. This example is representative of a rather comprehensive application of the proposed methodology, featuring most of interface types encountered in turbomachinery CFD. A discrete set of diameters has been injected uniformly at the domain inlet. Specifically, 100,000 particles of eight different diameters were injected, ranging from 1 to 128 μm growing with the power of 2. The results were collected for a steady (mixing plane) and an unsteady (overlapGGI) simulation and compared in terms of impact efficiency on the vane and blades surfaces. As expected, the pressure side of the two rows was more subject to discrete-phase impacts. Nonetheless, some impacts were also found on the suction side of the vane for larger diameters, meaning that some particles undergo a back and forth motion across the interface. This can be seen in Fig. 15 in the case of the unsteady interaction between the two rows, where 1 μm particles are able to follow fluid streamlines in proximity of the blade leading edge, while 32 μm particles impinge on the blade and are then pushed back toward the vane. This is also highlighted by the white circle in Fig. 15(*b*) showing particles exhibiting a negative axial velocity. Since no particles were lost across the interfaces in the steady as well as in transient simulations, the described approach is able to properly handle interface-crossing in the two directions. In this case, the blade count of the two cascades was reduced to 1 : 2 in order to have equal pitches and thus perform the unsteady computation without modeling the full annulus. The same geometry has been used for the mixing plane simulation, so there was no need to inject or delete any particles at the interface. Moreover, in the tip gap of the rotor domain, a GGI has been used to accommodate the meshing requirements. As illustrated in Fig. 16, particles transported by tip leakage flows are perfectly able to cross this interface without any trajectory modifications. At last, we can consider the global impact efficiency (η_{imp}) on the blade surface, defined as the ratio between the number of particle impacting on the blade and the total number of particles. This parameter was calculated for each diameter and is reported in Fig. 17. It can be noticed that η_{imp} can take values greater than one because particles can rebound multiple times on surfaces. To focus only on the impacts characteristics, no deposition models were used in the simulations. As can be seen from the figure, the circumferential redistribution at the mixing plane interface is able to capture, on average, the unsteady impact efficiency, except for the smallest diameter. In this latter case, indeed, the small inertia of particles makes them more sensitive to the unsteady flow field caused by rotor–stator interaction. Since the impact pattern on the vane was very similar between the steady and transient cases, it has not been reported here.

6 Conclusions

In the present work, the current capabilities of the foam-extend CFD software have been extended to particle-laden flows in multistage turbomachinery. Specifically, new techniques for particles transfer across domain interfaces have been presented with the aim of allowing Eulerian–Lagrangian simulations in a more general set of turbomachinery environments. This was not possible so far due to the lacking of a correct support for Lagrangian tracking through GGI and mixing plane interfaces, which are typically encountered in multistage rotating machinery computations. The presented methodology is conceptually simple and has been totally integrated and automatized in the CFD code, so that the users do not have to bother specifying the correct type of interaction between the particles and the interfaces. Moreover, the method is based on exploiting the existing features of the numerical code needed for the continuous-phase calculation, like face-to-face addressing and interfaces reconstruction/interpolation. This has allowed to achieve robustness and efficiency during the particle tracking step of the algorithm, making the implementation also suitable for parallel computations. With specific reference to mixing planes, a method has been proposed capable of circumferentially redistributing the particles on the aft side of the interface in a random manner inside the ribbon stripes, while correctly preserving the mass flowrate of the discrete phase. This is necessary since usually a difference in the pitchwise extension of the two domains connected by a mixing plane is present.

Moving to the computational results, first, a straightforward test-case has been provided to benchmark the correct implementation of the methodology; then, three case-studies of growing geometrical complexity have been presented. Each of these simulation aims at testing specific features of the code, while showing possible applications of interest in the opensource multiphase flow community. Specifically, the applications presented featured different combination of GGI and mixing plane interfaces, focusing on erosion in an axial compressor, trajectory analysis in a centrifugal fan, and impact analysis in an axial turbine. The numerical results have shown the extended code capabilities and are in agreement with other results previously reported in the literature.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The authors attest that all data for this study are included in the paper.

Nomenclature

m	=	mass
$\mathbf{v}$	=	velocity
$\mathbf{x}$	=	position
$\mathbf{F}$	=	force
d_p	=	particle diameter
C_D	=	drag coefficient
y^+	=	nondimensional wall distance
Re	=	Reynolds number

Greek Symbols

η	=	impact efficiency
θ	=	patch angular bound
μ	=	dynamic viscosity
ρ	=	density
ϕ_{random}	=	random angular coordinate
$\boldsymbol{\omega}$	=	angular velocity vector

Subscripts and Superscripts

p	=	particle
D	=	drag
I	=	inertial
NI	=	noninertial
Cor	=	coriolis
$centr$	=	centrifugal

References

[1] Crowe, C. T., Schwarzkopf, J. D., Sommerfeld, M., and Tsuji, Y., 2012, *Multiphase Flows With Droplets and Particles*, CRC Press, Boca Raton, FL.

[2] Guha, A., 2008, "Transport and Deposition of Particles in Turbulent and Laminar Flow," Annu. Rev. Fluid Mech., **40**(1), pp. 311–341.

[3] Stalder, J.-P., 2001, "Gas Turbine Compressor Washing State of the Art: Field Experiences," ASME J. Eng. Gas Turbines Power, **123**(2), pp. 363–370.

[4] Crane, R., 2004, "Droplet Deposition in Steam Turbines," Proc. Inst. Mech. Eng. Part C – J. Mech. Eng. Sci., **218**(8), pp. 859–870.

[5] Suman, A., Casari, N., Fabbri, E., Di Mare, L., Montomoli, F., and Pinelli, M., 2019, "Generalization of Particle Impact Behavior in Gas Turbine Via Non-Dimensional Grouping," Prog. Energy Combust. Sci., **74**(1), pp. 103–151.

[6] Suman, A., Casari, N., Fabbri, E., Pinelli, M., Di Mare, L., and Montomoli, F., 2019, "Gas Turbine Fouling Tests: Review, Critical Analysis, and Particle Impact Behavior Map," ASME J. Eng. Gas Turbines Power., **141**(3), p. 032601.

[7] Dunn, M. G., 2012, "Operation of Gas Turbine Engines in an Environment Contaminated With Volcanic Ash," ASME J. Turbomach., **134**(5), p. 051001.

[8] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C., 1998, "A Tensorial Approach to Computational Continuum Mechanics Using Object-Oriented Techniques," Comput. Phys., **12**(6), pp. 620–631.

[9] Jasak, H., and Beaudoin, M., 2011, "OpenFOAM Turbo Tools: From General Purpose CFD to Turbomachinery Simulations," ASME-JSME-KSME 2011 Joint Fluids Engineering Conference: Symposia – Parts A, B, C, and D, Fluids Engineering Division Summer Meeting, Hamamatsu, Shizuoka, Japan, July 24–29, Vol. 1, pp. 1801–1812.

[10] Beaudoin, M., Nilsson, H., Page, M., Magnan, R., and Jasak, H., 2014, "Evaluation of an Improved Mixing Plane Interface for OpenFOAM," Conf. Ser.: Earth Environ. Sci., **22**(2), p. 022004.

[11] Ghenaiet, A., 2019, "Prediction of Erosion in an Axial Turbine With Initial Position of Blade," Proceedings of the 13th European Turbomachinery Conference, Paper No. ETC2019-111.

[12] Hamed, A. A., Tabakoff, W., Rivir, R. B., Das, K., and Arora, P., 2004, "Turbine Blade Surface Deterioration by Erosion," ASME J. Turbomach., **127**(3), pp. 445–452.

[13] Tabakoff, W., Hamed, A., and Metwally, M., 1991, "Effect of Particle Size Distribution on Particle Dynamics and Blade Erosion in Axial Flow Turbines," ASME J. Eng. Gas Turbines Power, **113**(4), pp. 607–615.

[14] Bidwell, C. S., 2012, "Ice Particle Transport Analysis With Phase Change for the e^3 Turbofan Engine Using lewice3d Version 3.2," Technical Report ™-217700, NASA STI Report.

[15] Mustafa, Z., Pilidis, P., Amaral Teixeira, J., and Ahmad, K., 2006, "CFD Aerodynamic Investigation of Air–Water Trajectories on Rotor-Stator Blade of an Axial Compressor for Online Washing," Turbo Expo: Power for Land, Sea, and Air (Turbomachinery, Parts A and B, Vol. 6), Paper No. GT2006-90745, pp. 1385–1394.

[16] Yang, H., and Boulanger, J., 2013, "The Whole Annulus Computations of Particulate Flow and Erosion in an Axial Fan," ASME J. Turbomach., **135**(1), p. 011040.

[17] Zagnoli, D., Prenter, R., Ameri, A., and Bons, J. P., 2015, "Numerical Study of Deposition in a Full Turbine Stage Using Steady and Unsteady Methods," Turbo Expo: Power for Land, Sea, and Air (Turbomachinery, Vol. 2C), GT2015-43613, Montreal, Quebec, Canada, June 15–19.

[18] Prenter, R., Ameri, A., and Bons, J. P., 2017, "Computational Simulation of Deposition in a Cooled High-Pressure Turbine Stage With Hot Streaks," ASME J. Turbomach., **139**(9), p. 091005.

[19] Shirolkar, J., Coimbra, C., and McQuay, M., 1996, "Fundamental Aspects of Modeling Turbulent Particle Dispersion in Dilute Flows," Prog. Energy Combust. Sci., **22**(4), pp. 363–399.

[20] Mashayek, F., and Pandya, R., 2003, "Analytical Description of Particle/Droplet-Laden Turbulent Flows," Prog. Energy Combust. Sci., **29**(4), pp. 329–378.

[21] Balachandar, S., and Eton, J. K., 2010, "Turbulent Dispersed Multiphase Flows," Annu. Rev. Fluid Mech., **42**(1), pp. 11–133.

[22] Macpherson, G. B., Nordin, N., and Weller, H. G., 2009, "Particle Tracking in Unstructured, Arbitrary Polyhedral Meshes for Use in CFD and Molecular Dynamics," Commun. Numer. Methods Eng., **25**(3), pp. 263–273.

[23] Borm, O., Jemcov, A., and Kau, H., 2011, "Density Based Navier Stokes Solver for Transonic Flows," Sixth OpenFOAM Workshop, State College, PA, June 13–16, pp. 1–30.

[24] Oliani, S., Friso, R., Casari, N., Pinelli, M., Suman, A., and Carnevale, M., 2021, "A Comparative Analysis of Particle-Mixing Plane Interaction in Multistage Turbomachinery Simulations," European Turbomachinery Conference 14, ETC2021-544, Virtual Conference, Apr. 12–16.

[25] Greifzu, F., Kratzsch, C., Forgber, T., Lindner, F., and Schwarze, R., 2016, "Assessment of Particle-Tracking Models for Dispersed Particle-Laden Flows Implemented in Openfoam and Ansys Fluent," Eng. Appl. Comput. Fluid Mech., **10**(1), pp. 30–43.

[26] Gosman, A. D, and Loannides, E., 1983, "Aspects of Computer Simulation of Liquid-Fueled Combustors," J. Energy, **7**(6), pp. 482–490.

[27] Elghobashi, S., 1994, "On Predicting Particle-Laden Turbulent Flows," Appl. Sci. Res., **52**(1), pp. 309–329.

[28] Shiller, L., and Naumann, A., 1933, "A Drag Coefficient Correlation," Z. des Vereines Deutscher Ingenieure, **77**(12), pp. 318–320.

[29] Balan, C., and Tabakoff, W., 1984, "Axial Flow Compressor Performance Deterioration," 20th Joint Propulsion Conference, AIAA Paper 84-1208.

[30] Finnie, I., 1960, "Erosion of Surfaces by Solid Particles," Wear, **3**(2), pp. 87–103.