Prepared for *Computational Mechanics*

Alessio Castorrini · **Alessandro Corsini** · **Franco Rispoli** · **Paolo Venturini** · **Kenji Takizawa** · **Tayfun E. Tezduyar**

Computational analysis of performance deterioration of a wind turbine blade strip subjected to environmental erosion

Abstract Wind-turbine blade rain and sand erosion, over long periods of time, can degrade the aerodynamic performance and therefore the power production. Computational analysis of the erosion can help engineers have a better understanding of the maintenance and protection requirements. We present an integrated method for this class of computational analysis. The main components of the method are the Streamline-Upwind/Petrov-Galerkin (SUPG) and Pressure-Stabilizing/Petrov-Galerkin (PSPG) stabilizations, a finite element particle-cloud tracking method, an erosion model based on two time scales, and the Solid-Extension Mesh Moving Technique (SEMMT). The turbulent-flow nature of the analysis is handled with a Reynolds-Averaged Navier–Stokes (RANS) model and SUPG/PSPG stabilization, the particle-cloud trajectories are calculated based on the computed flow field and closure models defined for the turbulent dispersion of particles, and one-way dependence is assumed between the flow and particle dynamics. Because the geometry update due to the erosion has a very long time scale compared to the fluid–particle dynamics, the update takes place in a sequence of "evolution steps" representing the impact of the erosion. A scale-up factor, calculated in different ways depending on the update threshold criterion, relates the erosions and particle counts in the evolution steps to those in the fluid–particle simulation. As the blade geometry evolves, the mesh is updated with the SEMMT. We present computational analysis of rain and sand erosion for a wind-turbine blade strip, including a case with actual rainfall data and experimental aerodynamic data for eroded airfoil geometries.

Keywords Wind turbine · Blades · Rain erosion · Sand erosion · SUPG and PSPG methods · Particle-cloud tracking model · Erosion scale-up

1 Introduction

Wind-turbine blades are quite often subjected to environmental erosion. Heavy rainfall and frequent sandstorms can wear the blade coating. If the blade does not have sufficient surface protection, it can be seriously damaged by the erosion. This changes the blade aerodynamics, degrading the wind-turbine performance [1]. A deeply-eroded leading edge could result in up to 20% loss in power generation [2].

With the increasing size of wind-turbine blades, computational analysis is becoming inevitable in reducing the prototyping cost. Being able to predict how the erosion evolves as time passes could lead to improvement of the wind-turbine technology. Such a predictive tool can help the designers find new blade protection strategies. It can also help them plan effective maintenance schedules. Earlier, closely-related studies [3; 4] focused on the prediction of rain erosion patterns for a full-span wind-turbine blade. In this work we are focusing on the effect of the erosion on the aerodynamic performance of the blade, highlighting the correlation between the erosion patterns and geometry evolution.

It is clear that the aerodynamics, erosion, and the blade shape, interacting with each other, have much influence on the wind-turbine performance. It was shown in [5] that different blade geometries, at the same operational point and for the same power output, lead to different local aerodynamic fields with very different erosion patterns. Therefore we expect that even a minor modification on the critical parts of the blade section can make a noticeable difference in the flow field. The blade shape is composed of airfoils optimized to obtain the best performance. If the geometry of the critical parts of the airfoil changes, the flow field will be influenced and the aerodynamic performance and power generation of

---

Alessio Castorrini, Alessandro Corsini, Franco Rispoli, Paolo Venturini
Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome
Via Eudossiana, 18, I-00184 Rome, Italy
E-mail: alessio.castorrini@uniroma1.it, franco.rispoli@uniroma1.it

Kenji Takizawa
Department of Modern Mechanical Engineering, Waseda University
3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan
E-mail: Kenji.Takizawa@tafsm.org

Tayfun E. Tezduyar
Mechanical Engineering, Rice University – MS 321
6100 Main Street, Houston, TX 77005, USA
Faculty of Science and Engineering, Waseda University
3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan
E-mail: tezduyar@tafsm.org

the turbine will be diminished. Furthermore, the aerodynamics affects the particle motion and thus, the erosion itself. To increase the fidelity of the wind-turbine computational analysis, we have developed an integrated method for predicting the time evolution of the interaction between the fluid–particle dynamics and blade erosion and geometry change.

The typical approach we see in the literature does not account for the change in the flow field due to the geometry change. In most studies the flow simulation is performed only on the original geometry, and then the computed flow field is used to predict the particle transport and the particle erosion/deposit on the blade surface. This class of applications are characterized by small particle concentration. Therefore one-way dependence is assumed between the flow and particle dynamics, that is, particle (and cloud) motion is driven by the flow but the flow is not influenced by the particles. Tabakoff and coworkers [6; 7; 8] were among the earliest researchers working on numerical prediction of erosion in turbomachinery. The methods and experiments of Tabakoff and coworkers served as a foundation for other studies in the field. For example, Ghenaiet [9] simulated the erosion in radial compressors and ventilating systems, and Suzuki and Yamamoto [10] in axial compressors.

The main components of the integrated method we have developed for computational analysis of this class of problems are the Streamline-Upwind/Petrov-Galerkin (SUPG) [11] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [12] stabilizations, a finite element particle-cloud tracking (PCT) method with one-way dependence, an erosion model based on two time scales, and the Solid-Extension Mesh Moving Technique (SEMMT). The PCT method was first formulated by Baxter and Smith [13], and then further developed [14; 15] and improved to obtain statistically-independent results [16]. The trajectory of the particle-cloud center is calculated with finite element discretization. Of the elements of that "particle mesh," we use the ones inside the cloud, which has a trajectory-dependent radius. The tracking method accounts for the drift-velocity gradient in the near-wall regions [14; 17].

The turbulent-flow nature of the analysis is handled with a Reynolds-Averaged Navier–Stokes (RANS) model and the SUPG and PSPG stabilizations. These are complemented with the "DRDJ" stabilization [18; 19; 20; 21; 22]. The acronym "DRD" stands for the "Diffusion for Reaction-Dominated" formulation, introduced in [18], and the DRDJ is the version [19] that takes into account the local "jump" in the solution. There are various ways of calculating the stabilization and discontinuity-capturing (DC) parameters to be used with the SUPG, PSPG and discontinuity-capturing methods (see, for example, [23; 24; 18; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 19; 36; 37; 38; 39; 20; 40; 21; 41; 22; 42; 43; 44; 45; 46; 28; 35; 47; 48; 49; 50; 51; 52]). Here we use the ones given in [28]. The particle-cloud trajectories are calculated based on the flow field and a closure model for the turbulent dispersion of particles. The closure model can be based on the turbulence closure variables or the scale-separation feature of the variational multiscale (VMS) method [53].

Several empirical and semi-empirical models are available for erosion. The model used in [4] for rain erosion was a modified version of the Keegan model [54]. In this article, we use models defined in terms of the mass eroded per unit mass of the particles impacting on the blade surface. For the rain erosion, we use the model from Springer et al. [55], and for the sand erosion, the model from Oka et al. [56].

The geometry update due to the erosion has a very long time scale compared to the fluid–particle dynamics [57; 58]. Therefore a single time-marching simulation with the typical time-step size of the flow computation not practical. Instead, we use a sequence of "evolution steps" to represent the impact of the erosion. We time-discretize the evolution of the geometry not in terms of a standard time step, but in terms of a threshold erosion value that we expect to alter the blade aerodynamics from its current operation pattern or a threshold operation period that we expect to be long enough to alter the blade aerodynamics.

The computation associated with an evolution step gives us the erosion distribution for a specific particle size, for a specific number of particles injected to the computational domain. We have two approaches for scaling-up the erosion distribution at each evolution step. A. We scale-up the erosion distribution by a factor that raises its maximum value to the threshold erosion value. We use the same factor to scale-up the number of particles to the actual number of particles for that evolution step. B. We scale-up the number of particles by a factor that raises it to the actual number of particles for the threshold operation period. We use the same factor to scale-up the erosion distribution.

As the blade geometry evolves, we update the mesh with the SEMMT [59; 60; 61; 62; 63]. The core mesh moving method in the SEMMT is the Jacobian-based stiffening method [64; 65; 66; 67]. In the core mesh moving method, the motion of the internal nodes is determined by solving the equations of linear elasticity. The mesh deformation is dealt with selectively based on the sizes of the elements. The selective treatment is attained by altering the way we account for the Jacobian of the transformation from the element domain to the physical domain. The objective is to stiffen the smaller elements, which are typically placed near solid surfaces, more than the larger ones. When the method was introduced in [64; 65; 66], it consisted of simply dropping the Jacobian from the finite element formulation of the mesh moving (elasticity) equations. This results in the smaller elements being stiffened more than the larger ones. In the SEMMT, the thin layers of elements placed near solid surfaces are treated almost like an extension of the solid structure. In solving the equations of elasticity governing the motion of the fluid nodes, higher stiffness is assigned to the thin layers of elements compared to the other fluid elements. Two ways of accomplishing this were proposed in [59]: solving the elasticity equations for the nodes connected to the thin layers of elements separately from the elasticity equations for the other nodes, or together.

To show how the integrated method works, we perform computational analysis of rain and sand erosion for a wind-turbine blade strip, including a case with actual rainfall data and experimental aerodynamic data for eroded airfoil geometries.

In Section 2 we provide an overview of the integrated method. Section 3 is an overview of the mathematical model, including the RANS and PCT models. The SUPG/PSPG/DRDJ method for the Navier–Stokes and RANS closure equations is given in Section 4. In Section 5, we describe the discretized particle equations, including the turbulent dispersion of particles. The erosion models and erosion thickness computation, including how the scale-up factors are calculated, are described in Section 6. The SEMMT is described in Section 7. The computations are presented in Section 8, and the concluding remarks are given in Section 9.

2 An integrated method

We present an integrated method to simulate the long-term erosion of a wind-turbine blade. The method is made of a multiphase flow solver coupled with a geometry update method.

The time scales associated with the unsteady aerodynamics and turbulence, particle transport and dynamics, erosion of the target material and the change in the geometry that produces a significant variation in the average flow field and particle trajectories are different. The geometry update due to the erosion has a very long time scale compared to the fluid–particle dynamics, making a single time-marching simulation with the typical time-step size of the flow computation not practical. The basic idea is to have a sequence of evolution steps to represent the impact of the erosion. We time-discretize the evolution of the geometry not in terms of a standard time step, but in terms of a threshold erosion value that we expect to alter the blade aerodynamics from its current operation pattern or a threshold operation period that we expect to be long enough to alter the blade aerodynamics. The alteration of the flow patterns leads to the alteration of the particle dynamics, which in turn alters the erosion patterns.

An evolution step is composed of the sub-steps listed below in the order they are taken.

1. Compute the flow field with the SUPG/PSPG/DRDJ method [18; 12; 28; 19; 20; 21; 22].
2. Compute the particle-cloud trajectories with the PCT method described in [4].
3. Compute the erosion distribution over the blade surface.
4. Scale-up the computed erosion distribution using the threshold erosion value or operation period that triggers a blade geometry alteration.
5. Update the blade geometry and fluid mechanics mesh to the next configuration. The mesh update is done with the SEMMT [59; 60; 61; 62; 63].

The computation associated with an evolution step gives us the erosion distribution e for a specific particle size, for a specific number of particles injected to the computational domain. Depending on the application, we have two approaches for scaling-up e at each evolution step.

A. We scale-up e by a factor that raises its maximum value e_{max} to the threshold erosion value e_{thr}. We use the same factor to scale-up the number of particles to the actual number of particles for that evolution step. At the end of all the evolution steps, we obtain a correlation map from the damaged configurations to the amount of particles needed to produce those configurations. This map can later be used to estimate by interpolation the geometrical configuration resulting from the amount of particles in a specific application.
B. We scale-up the number of particles by a factor that raises it to the actual number of particles for the threshold operation period. We use the same factor to scale-up e. At the end of all the evolution steps, we obtain a correlation map from the actual number of particles to the damaged configurations. This map can later be used to directly estimate the damaged configurations from a specific rate of rainfall or sandstorm during a long, specific period (e.g., 25 years) that we want to know the damage for.

3 Mathematical model

3.1 Fluid-phase RANS model for incompressible turbulent flows

Let $\Omega \subset R^{nsd}$ be the spatial domain with boundary Γ, and $(0,T)$ be the time domain. The unsteady RANS equations of incompressible turbulent flows can be written on Ω and $\forall t \in (0,T)$ as

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\mathscr{F}}\right) - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = \boldsymbol{0}, \tag{1}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \tag{2}$$

$$\rho\left(\frac{\partial \boldsymbol{\phi}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{\phi} + \boldsymbol{B}_{k\varepsilon}\boldsymbol{\phi} - \boldsymbol{\mathscr{F}}_{k\varepsilon}\right) - \boldsymbol{\nabla}\cdot\left(\rho\left(\boldsymbol{\nabla}\boldsymbol{\phi}\right)\boldsymbol{\nu}_{k\varepsilon}\right) = \boldsymbol{0}, \tag{3}$$

where ρ, $\boldsymbol{u}$, and $\boldsymbol{\phi} = (k, \tilde{\varepsilon})^T$ are the density, velocity, and turbulence closure variables, and k and $\tilde{\varepsilon}$ are the turbulent kinetic energy and homogeneous dissipation. The symbols $\boldsymbol{\mathscr{F}}$ and $\boldsymbol{\mathscr{F}}_{k\varepsilon}$ represent the vector of external forces and the source vector of the turbulence closure equations.

As given in [68], $\boldsymbol{\mathscr{F}}$ represents the volume sources related to the second- and third-order terms in the non-isotropic stress–strain relation [69]. It is expressed as

$$\boldsymbol{\mathscr{F}} = \boldsymbol{\nabla}\cdot\left(-0.1\nu_t\tau\left(\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})\frac{1}{3}\boldsymbol{I}\right)\right.$$

$$\begin{aligned}
&+0.1\nu_t\tau\left(\overline{\boldsymbol{\omega}}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u})+(\overline{\boldsymbol{\omega}}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u}))^T\right)\\
&+0.26\nu_t\tau\left(\overline{\boldsymbol{\omega}}(\boldsymbol{u})\cdot\overline{\boldsymbol{\omega}}(\boldsymbol{u})-\overline{\boldsymbol{\omega}}(\boldsymbol{u}):\overline{\boldsymbol{\omega}}(\boldsymbol{u})\frac{1}{3}\boldsymbol{I}\right)\\
&-10c_\mu^2\nu_t\tau^2\left(\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\overline{\boldsymbol{\omega}}(\boldsymbol{u})\right.\\
&\qquad\qquad\left.+(\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\overline{\boldsymbol{\omega}}(\boldsymbol{u}))^T\right)\\
&-5c_\mu^2\nu_t\tau^2\boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})\boldsymbol{\varepsilon}(\boldsymbol{u})\\
&\left.+5c_\mu^2\nu_t\tau^2\overline{\boldsymbol{\omega}}(\boldsymbol{u}):\overline{\boldsymbol{\omega}}(\boldsymbol{u})\boldsymbol{\varepsilon}(\boldsymbol{u})\right).
\end{aligned} \tag{4}$$

Here $\boldsymbol{I}$ is the identity tensor, $\boldsymbol{\varepsilon}(\boldsymbol{u})=\boldsymbol{\nabla u}+(\boldsymbol{\nabla u})^T$ is twice the strain-rate tensor, $\overline{\boldsymbol{\omega}}(\boldsymbol{u})=\boldsymbol{\nabla u}-(\boldsymbol{\nabla u})^T$ is twice the vorticity tensor, ν_t is the turbulent kinematic viscosity defined as $\nu_t=c_\mu f_\mu\tau k$, and $\tau=k/\tilde{\varepsilon}$ is the turbulence time scale, with c_μ and f_μ and other closure coefficients for the turbulence model [69] given in Table 1. In the table, $Re_t=k^2/(\nu\tilde{\varepsilon})$ is the turbulence Reynolds number, with ν being the molecular viscosity, and $\hat{e}$ and $\hat{\overline{\omega}}$ are the strain-rate and vorticity invariants defined as $\hat{e}=\tau\sqrt{0.5\boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})}$ and $\hat{\overline{\omega}}=\tau\sqrt{0.5\overline{\boldsymbol{\omega}}(\boldsymbol{u}):\overline{\boldsymbol{\omega}}(\boldsymbol{u})}$.

c_μ	$\frac{0.3\left(1-\exp\left(-0.36/\exp\left(-0.75max\left(\hat{e},\hat{\overline{\omega}}\right)\right)\right)\right)}{1+0.35\left(max\left(\hat{e},\hat{\overline{\omega}}\right)\right)^{1.5}}$
f_μ	$1-\exp\left(-(Re_t/90)^{0.5}-(Re_t/400)^2\right)$
$c_{\varepsilon 1}$	1.44
$c_{\varepsilon 2}$	1.92
$f_{\varepsilon 2}$	$1-0.3\exp\left(-Re_t^2\right)$
σ_ε	1.3
σ_k	1.0

Table 1 Turbulence closure coefficients

The source vector $\mathscr{F}_{k\varepsilon}$ is expressed as

$$\mathscr{F}_{k\varepsilon}=\begin{bmatrix} P_k-D \\ c_{\varepsilon 1}P_k\frac{\tilde{\varepsilon}}{k}+E \end{bmatrix}, \tag{5}$$

where $P_k=\boldsymbol{R}:\boldsymbol{\nabla u}$ is the production of turbulent kinetic energy, with $\boldsymbol{R}$ being the Reynolds stress tensor, $D=2\nu\|\boldsymbol{\nabla}\sqrt{k}\|^2$, and $E=0.0022\hat{e}k\tau\nu_t\|\boldsymbol{\nabla}\cdot(\boldsymbol{\nabla u})\|^2$.

The stress tensor is expressed as

$$\boldsymbol{\sigma}=-\left(p+\frac{2}{3}\rho k\right)\boldsymbol{I}+\rho\nu_u\boldsymbol{\varepsilon}(\boldsymbol{u}), \tag{6}$$

where p is the pressure, and $\nu_u=\nu+\nu_t$.

The diffusion terms in the turbulence closure equations are represented with the diffusivity matrix defined as

$$\boldsymbol{\nu}_{k\varepsilon}=\begin{bmatrix} \nu+\frac{\nu_t}{\sigma_k} & 0 \\ 0 & \nu+\frac{\nu_t}{\sigma_\varepsilon} \end{bmatrix}, \tag{7}$$

with the values of the coefficients σ_k and σ_ε given in Table 1.

The reaction terms, absorption-like in Eq. (3), are represented with the matrix

$$\boldsymbol{B}_{k\varepsilon}=\begin{bmatrix} B_k & 0 \\ 0 & B_\varepsilon \end{bmatrix}, \tag{8}$$

with

$$B_k=\frac{\tilde{\varepsilon}}{k}, \qquad B_\varepsilon=c_{\varepsilon 2}f_{\varepsilon 2}\frac{\tilde{\varepsilon}}{k}, \tag{9}$$

and the coefficients $c_{\varepsilon 2}$ and $f_{\varepsilon 2}$ are given in Table 1.

The essential and natural boundary conditions corresponding to Eqs. (1) and (3) are

$$\boldsymbol{u}=\boldsymbol{g}\text{ on }\Gamma_g \quad\text{and}\quad \boldsymbol{\phi}=\boldsymbol{g}_{k\varepsilon}\text{ on }\Gamma_{gk\varepsilon}, \tag{10}$$

$$\boldsymbol{n}\cdot\boldsymbol{\sigma}=\boldsymbol{h}\text{ on }\Gamma_h \quad\text{and}\quad \boldsymbol{n}\cdot(\rho(\boldsymbol{\nabla\phi})\boldsymbol{\nu}_{k\varepsilon}=\boldsymbol{0}\text{ on }\Gamma_{hk\varepsilon}, \tag{11}$$

where Γ_g, $\Gamma_{gk\varepsilon}$, Γ_h and $\Gamma_{hk\varepsilon}$ are the subsets of the boundary Γ, $\boldsymbol{n}$ is the unit normal vector, and $\boldsymbol{g}$, $\boldsymbol{g}_{k\varepsilon}$ and $\boldsymbol{h}$ are given functions.

3.2 Dispersed-phase model

Particle trajectories are simulated in a Lagrangian reference frame. Since particle concentration in this class of applications is very small (less than 10^{-6} in the particle volume fraction), a one-way dependence approach can be used [70]. That is, the flow influences the particle motion but the particles do not influence the flow. The concept of one-way dependence has been used in other computational engineering analyses. For example, in [71], the concept is used for computing the aerodynamic forces acting on the suspension lines of spacecraft parachutes, where the suspension lines are assumed to have no influence on the flow. In [46], the same assumption is used to study the particle–shock interaction. In [72; 73; 74], the assumption is used in flow-driven string dynamics in turbomachinery, where the strings are assumed to have no influence on the flow. We use the PCT model [75] to simulate a large number of particles without tracking them individually. The PCT approach was used in turbulent particle dispersion [13; 16; 76; 77; 78] and validated in turbomachinery and biomass furnaces [79; 80]. In the PCT model, each trajectory is not for a particle, but for a group ("cloud") of particles, thus represent the evolution of the cloud position as a function of time:

$$\boldsymbol{x}_c=\int_0^t \boldsymbol{v}_c dt'+(\boldsymbol{x}_c)_0. \tag{12}$$

Here, subscript c refers to the cloud, $\boldsymbol{v}_c$ is the velocity of the cloud, and $(\boldsymbol{x}_c)_0$ is the initial position of the cloud, which is the inflow boundary in our computations.

The equation of motion for the cloud is given by the Basset–Boussinesque–Oseen formulation, which, with one-way dependence hypothesis according to Armenio and Fiorotto [81], reads as

$$\frac{d\boldsymbol{v}_c}{dt}=\tau_R^{-1}\left(\langle\boldsymbol{u}\rangle-\boldsymbol{v}_c\right)+\langle\boldsymbol{f}\rangle+\left(1-\frac{\rho}{\rho_p}\right)\mathbf{a}_{\mathrm{GRAV}}, \tag{13}$$

where $\langle\ \rangle$ denotes ensemble average (defined later), $\boldsymbol{f}$ is the centrifugal and Coriolis forces, ρ_p is the particle material density, $\mathbf{a}_{\mathrm{GRAV}}$ is the gravitational acceleration, and τ_R is the particle relaxation time, which, for spherical particles, is

$$\tau_R^{-1} = \frac{3}{4d_p} C_D \frac{\rho}{\rho_p} \|\langle \boldsymbol{u} \rangle - \boldsymbol{v}_c\|. \tag{14}$$

Here, d_p is the particle diameter and C_D is the drag coefficient based on the particle Reynolds number $Re_p = \frac{\|\langle \boldsymbol{u} \rangle - \boldsymbol{v}_c\| d_p}{\nu}$, first introduced in [82]. The Stokes number is defined as

$$Stk = \frac{\tau_R U}{L}, \tag{15}$$

where U is the free-stream velocity and L is the flow length scale, which, in this context, is the chord length of the blade strip. The initial value for $\boldsymbol{v}_c$ is given as $\boldsymbol{v}_c(0) = \langle \boldsymbol{u} \rangle|_{t=0}$.

The ensemble average for the dispersed phase within the cloud is defined according to the hypothesis of independent statistical events, and for any ensemble-averaged quantity θ it reads as

$$\langle \theta \rangle = \frac{\int_{\Omega_c} \theta PDF(\boldsymbol{x},t) d\Omega}{\int_{\Omega_c} PDF(\boldsymbol{x},t) d\Omega}. \tag{16}$$

Here, Ω_c is the cloud domain and $PDF(\boldsymbol{x},t)$ is the probability density function.

In the PCT approach, the particle position distribution within a cloud is assumed to be Gaussian, and the cloud size varies in time, depending on the flow behavior. The PDF of the particle distribution within the cloud is

$$PDF(\boldsymbol{x},t) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left(-\frac{1}{2} \left(\frac{\|\boldsymbol{x} - \boldsymbol{x_c}\|}{\sigma} \right)^2 \right). \tag{17}$$

Here, σ is the square root of the variance of particle position, which accounts for the turbulent dispersion of particles. We will define it in Section 5. The cloud radius is 3σ, and that gives us Ω_c. Each cloud is assumed to consist of spherical particles with the same physical characteristics.

Combining Eqs. (13) and (14), we obtain

$$\frac{d\boldsymbol{v}_c}{dt} = C_D' \|\langle \boldsymbol{u} \rangle - \boldsymbol{v}_c\| \left(\langle \boldsymbol{u} \rangle - \boldsymbol{v}_c \right) + \langle \boldsymbol{f} \rangle + \left(1 - \frac{\rho}{\rho_p} \right) \mathbf{a}_{\mathrm{GRAV}}, \tag{18}$$

where

$$C_D' = \frac{3}{4d_p} C_D \frac{\rho}{\rho_p}. \tag{19}$$

4 SUPG/PSPG/DRDJ formulation of the fluid mechanics equations

For completeness, we include from [3; 4] the SUPG/PSPG/DRDJ method.

4.1 Stabilized formulations

In describing the SUPG/PSPG/DRDJ formulation of Eqs. (1), (2) and (3), we assume that we have some suitably-defined finite-dimensional trial solution and test function spaces $\mathcal{S}_u^h$, $\mathcal{S}_p^h$, $\mathcal{S}_\phi^h$ and $\mathcal{V}_u^h$, $\mathcal{V}_p^h$, $\mathcal{V}_\phi^h$. The SUPG/PSPG/DRDJ formulation is written as follows: find $\boldsymbol{u}^h \in \mathcal{S}_u^h$, $p^h \in \mathcal{S}_p^h$ and $\boldsymbol{\phi}^h \in \mathcal{S}_\phi^h$, such that $\forall \boldsymbol{w}^h \in \mathcal{V}_u^h$, $\forall q^h \in \mathcal{V}_p^h$ and $\forall \boldsymbol{\psi}^h \in \mathcal{V}_\phi^h$:

$$\begin{aligned}
&\int_\Omega \boldsymbol{w}^h \cdot \rho \left(\frac{\partial \boldsymbol{u}^h}{\partial t} + \boldsymbol{u}^h \cdot \boldsymbol{\nabla} \boldsymbol{u}^h - \boldsymbol{\mathscr{F}}^h \right) d\Omega \\
&+ \int_\Omega \frac{1}{2} \boldsymbol{\varepsilon}(\boldsymbol{w}^h) : \boldsymbol{\sigma}^h d\Omega - \int_{\Gamma_h} \boldsymbol{w}^h \cdot \boldsymbol{h}^h d\Gamma \\
&+ \int_\Omega q^h \boldsymbol{\nabla} \cdot \boldsymbol{u}^h d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{P}^{stab}(\boldsymbol{w}^h, q^h) \cdot \left(\text{Ł}(\boldsymbol{u}^h, p^h) - \rho \boldsymbol{\mathscr{F}}^h \right) d\Omega = 0,
\end{aligned} \tag{20}$$

where

$$\text{Ł}(\boldsymbol{w}^h, q^h) = \rho \left(\frac{\partial \boldsymbol{w}^h}{\partial t} + \boldsymbol{u}^{\boldsymbol{h}} \cdot \boldsymbol{\nabla} \boldsymbol{w}^{\boldsymbol{h}} \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\boldsymbol{w}^h, q^h), \tag{21}$$

and

$$\begin{aligned}
&\int_\Omega \boldsymbol{\psi}^h \cdot \rho \left(\frac{\partial \boldsymbol{\phi}^h}{\partial t} + \boldsymbol{u}^h \cdot \boldsymbol{\nabla} \boldsymbol{\phi}^h + \boldsymbol{B}_{k\varepsilon}^h \boldsymbol{\phi}^h - \boldsymbol{\mathscr{F}}_{k\varepsilon}^h \right) d\Omega \\
&+ \int_\Omega \boldsymbol{\nabla} \boldsymbol{\psi}^h : \left(\rho (\boldsymbol{\nabla} \boldsymbol{\phi}^h) \boldsymbol{v}_{k\varepsilon}^h \right) d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{P}_{k\varepsilon}^{stab}(\boldsymbol{\psi}^h) \cdot \left(\text{Ł}_{k\varepsilon}(\boldsymbol{\phi}^h) - \rho \boldsymbol{\mathscr{F}}_{k\varepsilon}^h \right) d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{K}_{k\varepsilon}^{DC} \rho \boldsymbol{\nabla} \boldsymbol{\psi}^h : \boldsymbol{\nabla} \boldsymbol{\phi}^h d\Omega = 0,
\end{aligned} \tag{22}$$

where

$$\begin{aligned}
\text{Ł}_{k\varepsilon}(\boldsymbol{\phi}^h) = \rho &\left(\frac{\partial \boldsymbol{\phi}^h}{\partial t} + \boldsymbol{u}^h \cdot \boldsymbol{\nabla} \boldsymbol{\phi}^h + \boldsymbol{B}_{k\varepsilon}^h \boldsymbol{\phi}^h \right) \\
&- \boldsymbol{\nabla} \cdot \left(\rho (\boldsymbol{\nabla} \boldsymbol{\phi}^h) \boldsymbol{v}_{k\varepsilon}^h \right).
\end{aligned} \tag{23}$$

We calculate $\boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} \boldsymbol{u}^h)$ in the E term of $\boldsymbol{\mathscr{F}}_{k\varepsilon}^h$ by first calculating the nodal values of $\boldsymbol{\nabla} \boldsymbol{u}^h$ by least-squares projection and then taking the divergence of the interpolated value of $\boldsymbol{\nabla} \boldsymbol{u}^h$.

In Eqs. (20)–(23), $\boldsymbol{P}^{stab}$, $\boldsymbol{P}_{k\varepsilon}^{stab}$ and $\boldsymbol{K}_{k\varepsilon}^{DC}$ are the SUPG/PSPG stabilization operators and the DC matrix of the DRDJ stabilization. The vectors $\boldsymbol{P}^{stab}$ and $\boldsymbol{P}_{k\varepsilon}^{stab}$ take the forms

$$\boldsymbol{P}^{stab}(\boldsymbol{w}^h, q^h) = \tau_{SUPG} (\boldsymbol{u}^h \cdot \boldsymbol{\nabla}) \boldsymbol{w}^h + \frac{\tau_{PSPG}}{\rho} \boldsymbol{\nabla} q^h, \tag{24}$$

$$\boldsymbol{P}_{k\varepsilon}^{stab}(\boldsymbol{\psi}^h) = \begin{bmatrix} \tau_{SUPG-k} & 0 \\ 0 & \tau_{SUPG-\varepsilon} \end{bmatrix} (\boldsymbol{u}^{\boldsymbol{h}} \cdot \boldsymbol{\nabla}) \boldsymbol{\psi}^h. \tag{25}$$

Here τ_{SUPG} and τ_{PSPG} are the SUPG and PSPG stabilization parameters. These are defined in Section 4.2.

The DC matrix is defined as

$$\boldsymbol{K}^{DC}_{k\varepsilon} = \begin{bmatrix} \kappa_{DRDJ-k} & 0 \\ 0 & \kappa_{DRDJ-\varepsilon} \end{bmatrix}, \tag{26}$$

where κ_{DRDJ-k} and $\kappa_{DRDJ-\varepsilon}$ are the DRDJ diffusivities (see [18; 19; 20; 21; 22]).

4.2 Stabilization parameters

We first define the element length [18] in the advection-dominated limit:

$$h_{UGN} = 2\left(\sum_{a=1}^{n_{en}} |\boldsymbol{s}\cdot\boldsymbol{\nabla}N_a|\right)^{-1}, \tag{27}$$

where $\boldsymbol{s} = \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}$, n_{en} is the number of element nodes, and N_a is the interpolation function associated with node a.

In the diffusion-dominated limit, the element lengths [28] are defined as

$$h_{RGN} = 2\left(\sum_{a=1}^{n_{en}} |\boldsymbol{r}\cdot\boldsymbol{\nabla}N_a|\right)^{-1}, \tag{28}$$

$$h_{RGN-k} = 2\left(\sum_{a=1}^{n_{en}} |\boldsymbol{r}_{\boldsymbol{k}}\cdot\boldsymbol{\nabla}N_a|\right)^{-1}, \tag{29}$$

$$h_{RGN-\varepsilon} = 2\left(\sum_{a=1}^{n_{en}} |\boldsymbol{r}_{\boldsymbol{\varepsilon}}\cdot\boldsymbol{\nabla}N_a|\right)^{-1}, \tag{30}$$

where $\boldsymbol{r}$, $\boldsymbol{r}_k$ and $\boldsymbol{r}_\varepsilon$ are the unit vectors in the direction of the solution gradient:

$$\boldsymbol{r} = \frac{\boldsymbol{\nabla}\|\boldsymbol{u}\|}{\|\boldsymbol{\nabla}\|\boldsymbol{u}\|\|}, \qquad \boldsymbol{r}_k = \frac{\boldsymbol{\nabla}|k|}{\|\boldsymbol{\nabla}|k|\|}, \qquad \boldsymbol{r}_\varepsilon = \frac{\boldsymbol{\nabla}|\tilde{\varepsilon}|}{\|\boldsymbol{\nabla}|\tilde{\varepsilon}|\|}. \tag{31}$$

The components of τ_{SUPG} corresponding to the advection-, transient-, and diffusion-dominated limits were defined in [28; 36] as

$$\tau_{SUGN1} = \left(\sum_{a=1}^{n_{en}} |\boldsymbol{u}\cdot\boldsymbol{\nabla}N_a|\right)^{-1} = \frac{h_{UGN}}{2\|\boldsymbol{u}\|}, \tag{32}$$

$$\tau_{SUGN2} = \frac{\Delta t}{2}, \tag{33}$$

$$\tau_{SUGN3} = \frac{h_{RGN}^2}{4\nu}, \tag{34}$$

$$\tau_{SUGN3-k} = \frac{h_{RGN-k}^2}{4\nu_k}, \tag{35}$$

$$\tau_{SUGN3-\varepsilon} = \frac{h_{RGN-\varepsilon}^2}{4\nu_\varepsilon}. \tag{36}$$

From these, the stabilization parameters are defined as

$$\tau_{SUPG} = \left(\frac{1}{\tau_{SUGN1}^2} + \frac{1}{\tau_{SUGN2}^2} + \frac{1}{\tau_{SUGN3}^2}\right)^{-\frac{1}{2}}, \tag{37}$$

$$\tau_{PSPG} = \tau_{SUPG}, \tag{38}$$

$$\tau_{SUPG-k} = \left(\frac{1}{\tau_{SUGN1}^2} + \frac{1}{\tau_{SUGN2}^2} + \frac{1}{\tau_{SUGN3-k}^2}\right)^{-\frac{1}{2}}, \tag{39}$$

$$\tau_{SUPG-\varepsilon} = \left(\frac{1}{\tau_{SUGN1}^2} + \frac{1}{\tau_{SUGN2}^2} + \frac{1}{\tau_{SUGN3-\varepsilon}^2}\right)^{-\frac{1}{2}}. \tag{40}$$

5 Discretized particle equations

For completeness, we include from [3; 4] the discretized particle equations. In these equations, ensemble averaging is carried out over the discretized cloud domain $\Omega_c = \bigcup_{e=1}^{n_{elc}} \Omega_c^e$, where Ω_c^e is the cloud element and n_{elc} is the number of elements. The cloud elements come from a fixed mesh, which we call "particle mesh," and consist of the elements of the fixed mesh within a radius of 3σ. With that, the discretized version of ensemble averaging is written as

$$\langle\theta\rangle^h = \frac{\sum_{e=1}^{n_{elc}} \int_{\Omega_c^e} \theta PDF(\boldsymbol{x},t)d\Omega}{\sum_{e=1}^{n_{elc}} \int_{\Omega_c^e} PDF(\boldsymbol{x},t)d\Omega}, \tag{41}$$

where the element-level integration is performed by Gaussian quadrature.

5.1 Trajectory calculation

Spatially-discretized version of Eq. (18) is written as

$$\frac{d\boldsymbol{v}_{\boldsymbol{c}}^h}{dt} = \boldsymbol{a}_c^h, \tag{42}$$

where

$$\boldsymbol{a}_c^h = C_D'\|\langle\boldsymbol{u}\rangle^h - \boldsymbol{v}_c^h\|\left(\langle\boldsymbol{u}\rangle^h - \boldsymbol{v}_c^h\right) + \langle\boldsymbol{f}\rangle^h + \left(1 - \frac{\rho}{\rho_p}\right)\mathbf{a}_{\mathrm{GRAV}}. \tag{43}$$

Time discretization of Eq. (42) is done with a predictor–multicorrector algorithm.

Predictor stage:

$$(\boldsymbol{v}_c^h)_{n+1}^0 = (\boldsymbol{v}_c^h)_n + (\boldsymbol{a}_c^h)_n\Delta t. \tag{44}$$

Multicorrector stage:

$$(\boldsymbol{v}_c^h)_{n+1}^{i+1} = (\boldsymbol{v}_c^h)_n + \left((\boldsymbol{a}_c^h)_n + (\boldsymbol{a}_c^h)_{n+1}^i\right)\frac{\Delta t}{2}. \tag{45}$$

Here the subscript n is the time level, and the superscript i is the counter for the multiple corrections. We stop the corrections when

$$\frac{(\mathbf{v}_c^h)_{n+1}^{i+1} - (\mathbf{v}_c^h)_{n+1}^{i}}{(\mathbf{v}_c^h)_{n+1}^{i+1}} \leq 2\times10^{-2}. \tag{46}$$

At each time step, the PCT model requires the computation of the cloud mean position and radius, and identification of the elements contained within the cloud volume. This is done with the search algorithm described in [17].

5.2 Parameters of the turbulent dispersion of particles

The variance is taken to be dependent upon the Lagrangian time scale of the particle-laden flow, τ_L, and, according to Baxter [75], its Markovian approximation reads as

$$\sigma^2 = 2(v')_c^2\tau_L^2\left(\frac{t}{\tau_L} - (1-e^{-t/\tau_L})\right) + \sigma_0^2, \tag{47}$$

where τ_L is defined as

$$\tau_L = max(\tau, \tau_p) = max(\tau, \tau_R), \tag{48}$$

with τ_p given as $\tau_p = \tau_R$ and τ defined below. The fluctuating component of the particle velocity for the cloud, driven by the turbulent dispersion of particles [83], reads as

$$(v')_c^2 = (u')_c^2\left(1 - e^{-\tau/\tau_p}\right). \tag{49}$$

We consider two sets of definitions (see [45]) for $(u')_c^2$ and τ. The first one is based on the eddy viscosity model (EVM) through the turbulence closure variables [75]:

$$(u')_{c-EVM}^2 = \frac{2}{3}\left\langle k\right\rangle^h, \tag{50}$$

$$\tau = \frac{\left(\left\langle c_\mu\right\rangle^h\right)^{3/4}\left\langle k\right\rangle^h}{0.817\left\langle\tilde{\varepsilon}\right\rangle^h}. \tag{51}$$

The second set is based on the VMS approach, first proposed in [53] and further developed for RANS computations [20; 36; 84]. In this case $(u')_c^2$ is based on the VMS scale separation $\mathbf{u} = \mathbf{u}^h + \mathbf{u}'$, where $\mathbf{u}^h$ is the resolved flow velocity and $\mathbf{u}'$ is the fine-scale flow velocity modeled as

$$\mathbf{u}' = -\frac{1}{\rho}\tau_{SUPG}\left(\text{Ł}(\mathbf{u}^h, p^h) - \rho\mathscr{F}^h\right). \tag{52}$$

Then the definitions of the parameters for the VMS turbulent dispersion of particles become

$$(u')_{c-VMS}^2 = \left\langle\|\mathbf{u}'\|^2\right\rangle^h, \tag{53}$$

$$\tau = \tau_{SUPG}. \tag{54}$$

In the computations reported here, we use the first set.

6 Erosion models and erosion thickness computation

In Section 2 we described two scale-up approaches that drive the sequence of evolution steps. In the “threshold erosion value” approach, we specify e_{thr}, and we assume that when the scaled-up e_{max} reaches e_{thr}, the erosion is at a level to alter the blade aerodynamics from its current operation pattern. The nominal blade geometry plays a role in determining e_{thr}. The e computed in a simulation associated with an evolution step depends on the current blade geometry and the size and spatial distribution of the particles. We assume that all these remain in effect during an evolution step, justifying the scale-up of e to the erosion distribution for that evolution step. In the “threshold operation period” approach, we specify an operation period that we expect to be long enough to alter the blade aerodynamics, and that becomes the duration associated with each evolution step. We again assume that the current blade geometry and the size and spatial distribution of the particles remain in effect during an evolution step. The scale-up factor becomes the ratio of the number of particles in that duration to the number of particles used in the simulation. More details on the scale-up factors will be given in Section 6.5.

6.1 Erosion thickness computation

The erosion thickness e on the surface of the target, calculated at the element level, is expressed as

$$e = \frac{m_e}{\rho_t}, \tag{55}$$

where m_e is the eroded mass per unit area, computed in the simulation associated with the evolution step, and ρ_t is the density of the target material. Following the notation in [4], we obtain the eroded mass by summing up the mass eroded in each time step Δt:

$$m_e = \sum \Delta m_e, \tag{56}$$

where, after a threshold particle impact counted is reached,

$$\Delta m_e = E\Delta n_p m_p. \tag{57}$$

Here E is the erosion rate, Δn_p is the particle impact count per unit area in Δt, and m_p is the particle mass. In the finite element PCT computation, Δn_p is calculated as

$$\Delta n_p = C_{elem} v_{i,n,elem}\Delta t, \tag{58}$$

where C_{elem} is the particle concentration in the element and $v_{i,n,elem}$ is the normal component of the particle impact velocity. Both are evaluated at the element center.

We selected two suitable models to determine E, one for rain erosion [55] and one for sand erosion [56].

6.2 Rain erosion

We use the model proposed by Springer et al. [55]. It is a model based on the experiments. It is assumed that the erosion is zero until a threshold impact count $(n_p)_{thr}$ is reached, and grows linearly with the impact count after that. An empirical expression is used for $(n_p)_{thr}$:

$$A_{pp}(n_p)_{thr} = a_1 \left(\frac{S_{eff}}{\overline{\sigma}_0} \right)^{a_2}, \tag{59}$$

where A_{pp} is the projected area of the particle, S_{eff} is a parameter which characterizes the strength of the coating material, $\overline{\sigma}_0$ is the stress produced by the impinging droplet, and $a_1 = 7.1 \times 10^{-6}$ and $a_2 = 5.7$ are model constants, all defined in [55]. In our computations, $(n_p)_{thr}$ is a time-averaged quantity, obtained by integration over the particle simulation.

The model is assumed to be valid until an upper-limit impact count $(n_p)_f$ is reached. The limit has been obtained from experimental observation, and it is given as

$$A_{pp}(n_p)_f = a_3 \left(\frac{S_{eff}}{\overline{\sigma}_0} \right)^{a_2}, \tag{60}$$

where $a_3 = 21.3 \times 10^{-6}$. As pointed out in [55], in most practical situations the usefulness of the material does not extend beyond $(n_p)_f$.

After the impact count raches $(n_p)_{thr}$, the eroded mass is calculated as

$$\Delta m_e = \alpha \Delta n_p, \tag{61}$$

where α represents the mass eroded per impact. It is obtained in [55] for homogeneous materials as

$$\alpha = a_4 \frac{1}{(A_{pp}(n_p)_{thr})^{a_5}} m_p, \tag{62}$$

where $a_4 = 0.023$ and $a_5 = 0.7$ are empirical constants, given in [55]. From that we extract $\overline{E}$:

$$\overline{E} = a_4 \frac{1}{(A_{pp}(n_p)_{thr})^{a_5}}, \tag{63}$$

which, by definition, is also a time-averaged quantity, and that is what we use in Eq. (57).

6.3 Sand erosion

For E used in Eq. (57), we adopt the expression given in [85]:

$$E = K(v_{i,elem})^n, \tag{64}$$

where K and n are empirical coefficients that depend on the impact angle (see Figure 1) and coating material. They are provided in a tabulated form (see Section 8). We note that this model assumes a double-dependency on the impact angle, in determining the value of K and n, and in calculating the impact count, through the parameter $v_{i,n,elem}$.

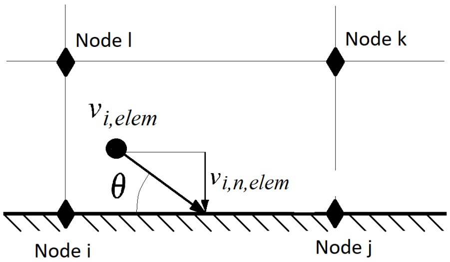

Fig. 1 Impact angle in the sand erosion model

6.4 Erosion scale-up

The erosion growth in the rain erosion model is shown in Figure 2. Keeping in mind that n_p, $(n_p)_{thr}$ and e vary on the blade surface S, we define on S a "banked" impact count B and an "incubation" function δ_B, both element-level quantities. Here B is the banked scaled-up impact count at the beginning of the evolution step i, which will be nonzero for the elements where the scaled-up impact count did not reach $(n_p)_{thr}$ in the previous evolution step and are still incubating, and $\delta_B = 1$ will mark those elements. For the elements where the scaled-up impact count reached $(n_p)_{thr}$ in the previous evolution step, $B = 0$ and $\delta_B = 0$.

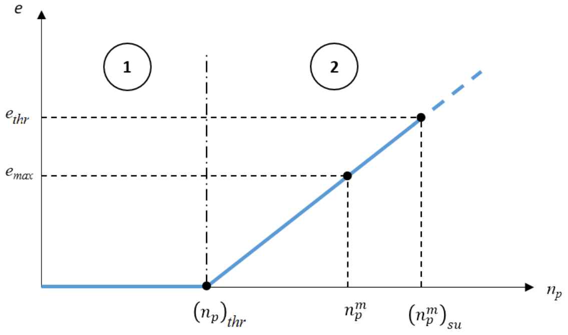

Fig. 2 Erosion growth in the rain erosion model

Consider the first evolution step. We initialize $B^{(0)} = 0$ and $\delta_B^{(0)} = 1$ everywhere on S. The scaled-up erosion is calculated as

$$\eta^{(1)} = \frac{\overline{E}^{(i)} m_p}{\rho_t} \left(F_{ACT}^{(1)} n_p^{(1)} - (n_p)_{thr}^{(i)} \right), \tag{65}$$

where $F_{ACT}^{(1)}$ is the actual scale-up factor for Step 1, obtained with the scale-up approach selected (see Section 6.5). With $F_{ACT}^{(1)}$, we scale-up e at the end of Step 1 and update B and δ_B for all elements:

$$\begin{cases} e_{SU}^{(1)} = \eta^{(1)},\ \delta_B^{(1)} = 0,\ B^{(1)} = 0, & \text{if } \eta^{(1)} > 0, \\ e_{SU}^{(1)} = 0,\ \delta_B^{(1)} = 1,\ B^{(1)} = F_{ACT}^{(1)} n_p^{(1)}, & \text{if } \eta^{(1)} \le 0. \end{cases} \tag{66}$$

For evolution steps with $i > 1$, Eqs. (65) and (66) are generalized as

$$\eta^{(i)} = \frac{\overline{E}^{(i)} m_p}{\rho_t} \left(B^{(i-1)} + F_{ACT}^{(i)} n_p^{(i)} - \delta_B^{(i-1)} (n_p)_{thr}^{(i)} \right), \quad (67)$$

$$\begin{cases} e_{SU}^{(i)} = \eta^{(i)}, \ \delta_B^{(i)} = 0, B^{(i)} = 0, \ \text{if } \eta^{(i)} > 0, \\ e_{SU}^{(i)} = 0, \ \delta_B^{(i)} = 1, B^{(i)} = B^{(i-1)} + F_{ACT}^{(i)} n_p^{(i)}, \ \text{if } \eta^{(i)} \leq 0. \end{cases} \quad (68)$$

It is clear that these expressions extend to erosion models with zero threshold impact count, i.e. models with $(n_p)_{thr} = 0$, and become simpler in such cases.

For the sand erosion model the scale-up process is similar, but we keep in mind that it is a model with zero threshold impact count, and E is a function of the impact angle and impact velocity. Consequently, the scale-up becomes

$$e_{SU}^{(i)} = F_{ACT}^{(i)} e^{(i)}, \quad (69)$$

where e comes from Eq. (55).

6.5 Scale-up factors

6.5.1 Specified threshold erosion value

For the rain erosion model, for each element on S we solve the following equation for $F^{(i)}$:

$$e_{thr} = \frac{\overline{E}^{(i)} m_p}{\rho_t} \left(B^{(i-1)} + F^{(i)} n_p^{(i)} - \delta_B^{(i-1)} (n_p)_{thr}^{(1)} \right). \quad (70)$$

We set $F_{ACT}^{(i)}$ equal to the minimum of $F^{(i)}$ over all the elements on S and use that in Section 6.4.

For the sand erosion model, for each element on S we use the following expression for $F^{(i)}$:

$$F^{(i)} = \frac{e_{thr}}{e^{(i)}}. \quad (71)$$

We again set $F_{ACT}^{(i)}$ equal to the minimum of $F^{(i)}$ over all the elements on S and use that in Section 6.4. This is equivalent to setting

$$F_{ACT}^{(i)} = \frac{e_{thr}}{e_{max}^{(i)}}. \quad (72)$$

The scale-up factor $F_{ACT}^{(i)}$, as described in Section 6.4, is used in obtaining $e_{SU}^{(i)}$, the scaled-up erosion distribution. The same factor is used in obtaining $(n_{TOT})_{SU}^{(i)}$, the total number of particles needed to enter the PCT domain at the evolution step i to produce that erosion distribution:

$$(n_{TOT})_{SU}^{(i)} = F_{ACT}^{(i)} n_{TOT}, \quad (73)$$

where $n_{TOT} = \frac{1}{A_{PINF}} \sum_{C=1}^{N_C} n_C$ is the total number of particles per unit area entering the PCT simulation domain. Here N_C is the number of particle clouds entering the PCT simulation domain, n_C is the number of particles per cloud, and A_{PINF} is the simulation inflow area. From $(n_{TOT})_{SU}^{(i)}$, we can calculate the operation period associated with the evolution step i.

6.5.2 Specified threshold operation period

In a specific application, given the actual threshold operation period $T^{(i)}$ for an evolution step, and the number of particles per unit area and per unit time, $\dot{n}_R$, we know the total number of particles entering the PCT domain as $n_R^{(i)} = \dot{n}_R T^{(i)}$. Usually, $\dot{n}_R$ is a known quantity from field data, e.g. the rainfall in a geographical region during a year or the particle flow into a gas-turbine duct in a certain operation period. We can then calculate the scale-up factor as

$$F_{ACT}^{(i)} = \frac{n_R^{(i)}}{n_{TOT}} \quad (74)$$

and use that in Section 6.4.

We will show a specific application of this approach in Section 8.3, extracting $\dot{n}_R$ from the measured rain data in a specific region.

7 Mesh update with the SEMMT

As the blade geometry evolves based on the scaled-up erosion distribution, we update the mesh with the SEMMT. The core mesh moving method in the SEMMT is the Jacobian-based stiffening method. We include, from [86], a brief description of these methods.

The Jacobian-based stiffening method was introduced in [64; 65; 66]. The motion of the internal nodes is determined by solving the equations of linear elasticity. At the boundaries, the normal velocity of the mesh matches the normal velocity of the fluid. The mesh deformation is dealt with selectively based on the sizes of the elements. Selective treatment based on element sizes is attained by altering the way we account for the Jacobian of the transformation from the element domain to the physical domain. The objective is to stiffen the smaller elements, which are typically placed near solid surfaces, more than the larger ones. When the method was introduced in [64; 65; 66], it consisted of simply dropping the Jacobian from the finite element formulation of the mesh moving (elasticity) equations. This results in the smaller elements being stiffened more than the larger ones. The method was named “Jacobian-based stiffening” in [67]. It was also augmented in [67] to a more extensive kind by introducing a stiffening power that determines the degree by which the smaller elements are rendered stiffer than the larger ones. This approach, when the stiffening power is set to 1.0, would be identical to the one introduced in [64].

The SEMMT was proposed in 2001 [59]. In the SEMMT [59; 60; 61; 62; 63], the thin layers of elements placed near solid surfaces are treated almost like an extension of the solid structure. In solving the equations of elasticity governing the motion of the fluid nodes, higher stiffness is assigned to the thin layers of elements compared to the other fluid elements. Two ways of accomplishing this were proposed in [59]: solving the elasticity equations for the nodes connected to the thin layers of elements separately from the elasticity equations for the other nodes, or together. If they are solved separately, for the thin layers of elements, as boundary conditions at the interface with the other elements, traction-free conditions would be used. The separate-solution option is referred to as "SEMMT – Multiple Domain (SEMMT-MD)," and the unified-solution option as "SEMMT – Single Domain (SEMMT-SD)." The test computations presented in [87; 61; 62; 63] show that the SEMMT is very effective in protecting the thin layers of elements near solid surfaces from deformation.

8 Computations

8.1 Computational conditions

The computational analysis is for a blade strip that has a DU-96-W-180 airfoil section with chord length 1.0 m. This airfoil profile is commonly used for the midspan blade section. The computations are for 6° angle-of-attack (AoA), which corresponds to a reasonable operation-average value for the blade section we have. The computational domain, shown in Figure 3, is for a wind tunnel section. The domain is 30 m long, 3.2 m high, and extends 0.2 m in the spanwise direction. The inflow velocity is normal to the inlet plane, with $Re = 1.2\times10^6$. The turbulence quantities are set to obtain a turbulence intensity of 5% at the inlet. The boundary conditions are no-slip, $k = 0$ and $\varepsilon = 0$ on the blade surface, slip and zero-flux on the upper and lower boundaries, zero-stress and zero-flux at the outflow boundary, and periodicity in the spanwise direction. The mesh, shown in Figure 4, is structured and has 3×10^5 hexahedral elements, with y^+ less than 1.0 everywhere on the blade surface.

Table 2 shows the properties of the fluid and particle phases. Table 3 shows, for the sand erosion, the empirical coefficients of the erosion model from [85].

Table 2 Properties of the fluid and particle phases

Air density	1.23 kg/m^3
Air velocity at the inlet	18 m/s
Reynolds number	1.2×10^6
Water density	1×10^3 kg/m^3
Diameter of the rain drops	1×10^{-3} m
Stokes number for the rain erosion	55.6
Density of the sand particles (SiO_2)	2.3×10^3 kg/m^3
Diameter of the sand particles	4×10^{-5} m
Stokes number for the sand erosion	0.361

Table 3 Empirical coefficients of the sand erosion model (for UHMWPE coating material) [85]

θ [°]	K [(m/s)$^{-n}$]	n
0	0	2.8000
15	1.366×10^{-9}	2.8065
30	3.337×10^{-9}	2.6056
60	1.490×10^{-9}	2.6500
90	2.350×10^{-11}	2.6500

The time-step size in both rain erosion and sand erosion cases is 1×10^{-6} s. In both PCT simulations we use 6 particle clouds, with 0.2 m diameter and with 50 million particles in each cloud.

8.2 Results for specified threshold erosion value

In both rain and sand erosion, we specify a threshold erosion value as the scale-up approach in the evolution steps. We show how the blade geometry evolves during the erosion. We set $e_{thr} = 2.5\times10^{-5}$ m, and the number of evolution steps is 24. With these values, at the end of the evolution, the maximum erosion thickness cannot exceed 6 mm. In this section we do not associate a time period to this blade evolution. If needed, we can of course calculate the total number of particles needed to enter the PCT domain at every evolution step, and given a rainfall or sand data in a specific application, we can map the blade evolution to elapsed time.

8.2.1 Rain erosion

To simplify the calculations, we assume that beyond the first evolution step every element on the blade surface have already reached $(n_p)_{thr}$ and we set $B^{(i-1)} = 0$ and $\delta_B^{(i-1)} = 0$. We make this assumption thinking that the elements that have not actually reached $(n_p)_{thr}$ are likely to have an erosion rate that will be negligible compared to the elements with $F_{ACT} n_p > (n_p)_{thr}$. Therefore assuming that beyond the first evolution step they also have reached $(n_p)_{thr}$ would not make much difference.

Figure 5 shows the geometry evolution and the initial and final meshes. We see how the erosion patterns vary at different evolution steps because of the variation in the geometry, aerodynamic field, and the particle transport. As the mesh deforms in response to the change in the blade geometry, the quality of the thin layers of elements near the blade surface is retained and this maintains the flow resolution in the boundary layer. Figure 6 shows the distribution of the impact count and erosion for the first and last evolution steps. The erosion pattern is symmetric at the beginning but loses symmetry with the variation of the geometry. Figure 7 shows the pressure and turbulent kinematic viscosity for the first and last evolution steps.

Figure 8 shows, for the first evolution step, the mass center trajectories for the particle clouds at different AoA. Be-

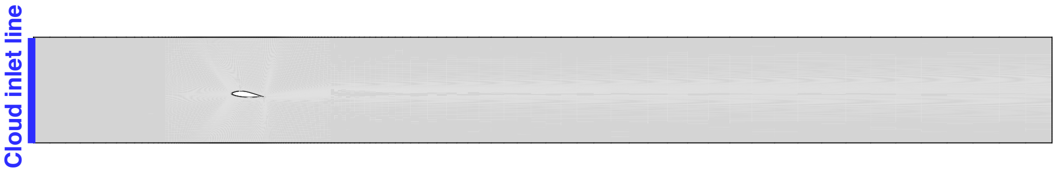

Fig. 3 Computational domain

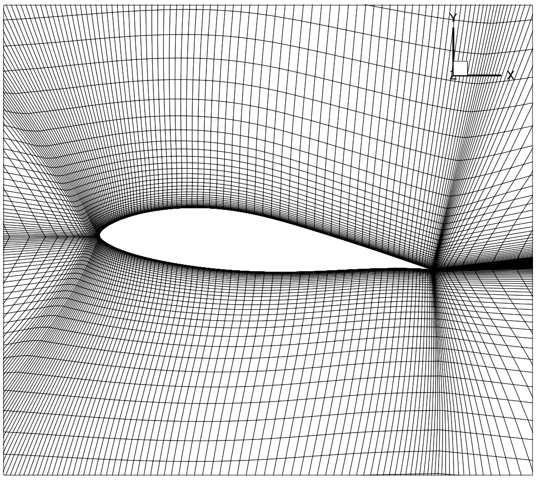

Fig. 4 Mesh cross-section

cause the Stokes number is quite high due to the inertia of the rain drops (1 mm diameter), the particles are not that affected by the drag. Figure 9 shows, for the first evolution step, the erosion patterns at different AoA. At the zero-lift AoA ($-2.2°$) we see a smooth variation in the erosion distribution on the suction and pressure sides. At 2° there is more erosion on the suction side, while the desirable erosion condition (balance between the two sides) is seen at 6°. At higher AoA (10°) the max erosion is on the pressure side.

8.2.2 Sand erosion

Because the sand particles are much smaller, the sand flow will have less kinetic energy and consequently less erosive power than the rain flow. Figure 10 shows the geometry evolution, initial and final meshes, and the pressure for the first and last evolution steps. Figure 11 shows the distribution of the impact count and erosion for the first and last evolution steps. Figure 12 shows, for the first evolution step, the mass center trajectories for the particle clouds at different AoA. The particles are following the flow more than in the rain erosion; this is due to the lower Stokes number. The deviation is greater at the leading edge. Figure 13 shows, for the first evolution step, the erosion patterns at different AoA. At the zero-lift AoA ($-2.2°$) the erosion is mostly at the leading edge. At 2° the erosions are comparable on the two sides. At 6° and 10° the erosion peaks and most of the erosion is on the pressure side.

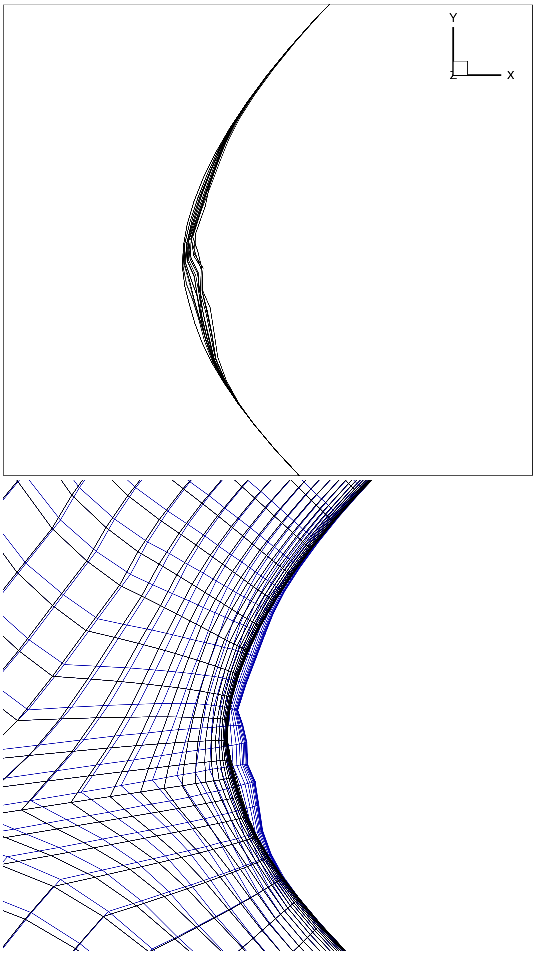

Fig. 5 Rain erosion. Geometry evolution and initial (*black*) and final (*blue*) meshes at the middle section of the blade strip

8.3 Results for specified threshold operation period: rain erosion field application

Given the rainfall data in this specific application, we specify a threshold operation period for the evolution steps. From that we extract the total number of particles entering the PCT domain in an evolution step and calculate the scale-up factor as described in Section 6.5.2. We compare the performance of the eroded airfoil with the experimental data from Sareen et al. [88].

As reported in [4], rain erosion is a common problem for the wind farms in Northern Europe. In Norther Scotland, the

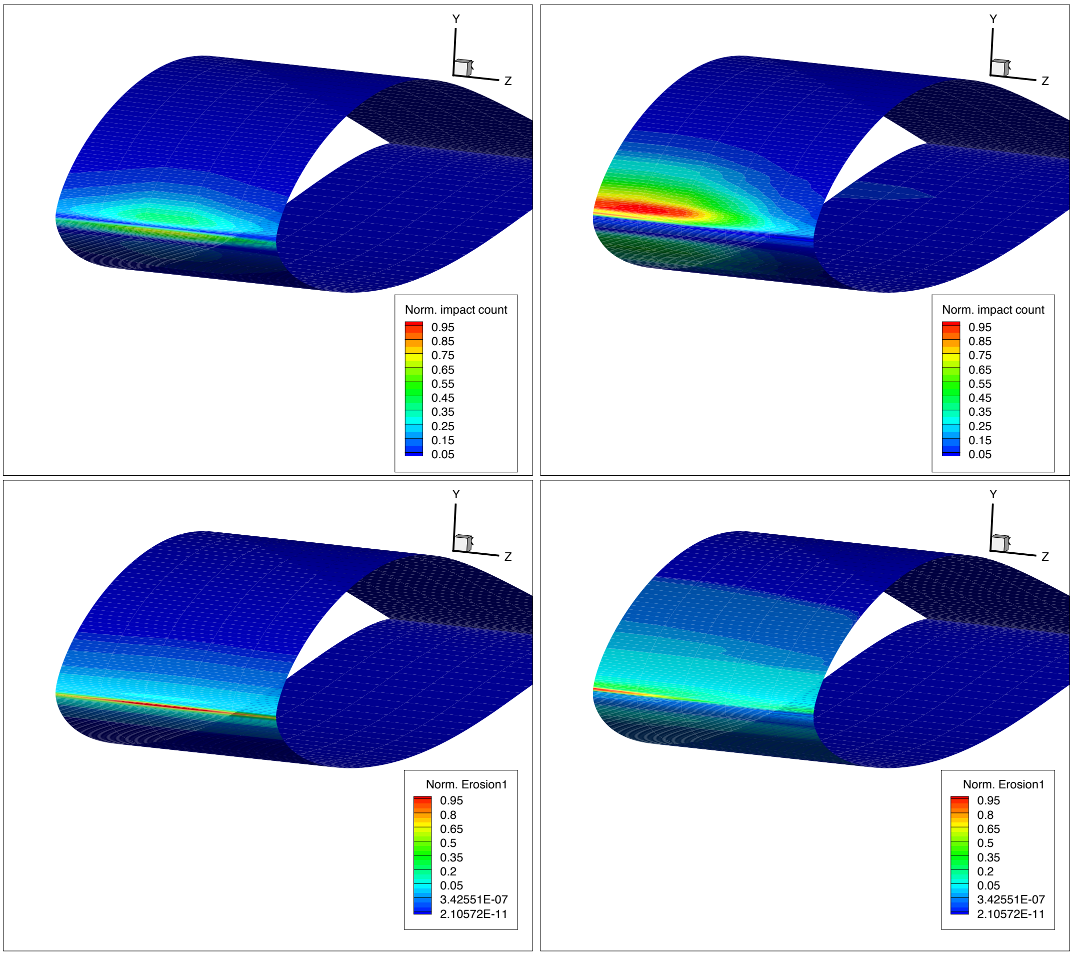

Fig. 6 Rain erosion. Normalized impact count (*top*) and erosion (*bottom*), at the end of the first (*left*) and last (*right*) evolution steps

average annual rainfall, H^Y_{RF}, is 2,000 mm, and that can also be represented as a rainfall rate of V_{RF} = 2,000 mm/yr. We define the rainwater fraction as

$$\beta_R = \frac{V_{RF}}{V_R}, \tag{75}$$

where V_R is the raindrop velocity, given in units consistent with the units of V_{RF}. The rainfall rate the blade sees is

$$V_{BRF} = \beta_R V_B = \frac{V_{RF}}{V_R} V_B, \tag{76}$$

where V_B is the blade velocity. The annual rainfall for the blade will then be

$$H^Y_{BRF} = \frac{V_B}{V_R} H^Y_{RF}. \tag{77}$$

In the PCT simulation, assuming the same number of particles in each cloud, we can calculate the simulation blade rainfall as

$$H^Y_{SBRF} = \frac{N_C n_C V_d}{A_{PINF}}, \tag{78}$$

where V_d is the raindrop volume.

Remark 1 *The expressions given by Eqs. (75) and (76) are based on the assumption that the rainfall is vertical, the blade velocity is horizontal, and the inflow plane is perpendicular to the blade velocity vector. In Appendix A we provide expressions for the general case, without these assumptions.*

In calculating F_{ACT} from Eq. (74), we first calculate n_R and n_{TOT}:

$$n_R = \frac{H^Y_{BRF}}{V_d}, \tag{79}$$

$$n_{TOT} = \frac{N_C n_C}{A_{PINF}} = \frac{H^Y_{SBRF}}{V_d}, \tag{80}$$

and then obtain

$$F_{ACT} = \frac{n_R}{n_{TOT}} = \frac{H^Y_{BRF}}{H^Y_{SBRF}}. \tag{81}$$

We could have of course reached the same point by simply recognizing that the ratio of the actual and simulation

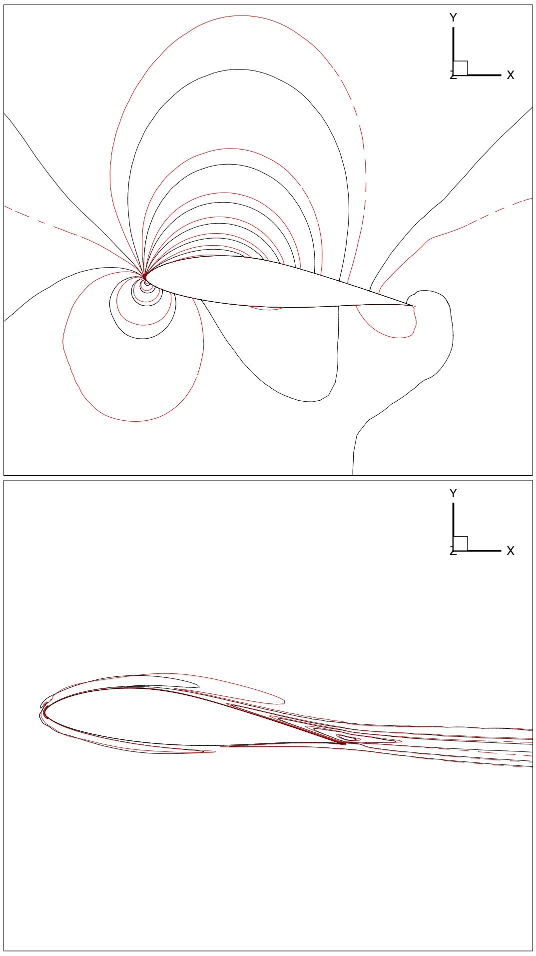

Fig. 7 Rain erosion. Pressure (*top*) and turbulent kinematic viscosity (*bottom*) at the middle section of the blade strip, for the first (*black*) and last (*red*) evolution steps

rainfalls the blade sees, $\frac{H^Y_{BRF}}{H^Y_{SBRF}}$, is the scale-up factor we are looking for.

The threshold operation period for the evolution steps is 1.0 year, and the number of evolution steps is 20, representing 20 years of service. Using the calculated value of F_{ACT}, at each evolution step we scale-up the computed erosion, obtaining the erosion in that year. Figure 14 shows the blade geometry at the beginning and end of the evolution steps. The maximum erosion thickness is 0.0012 m after 20 years.

We use the airfoil geometry from the middle section of the eroded blade strip to perform 2D computations at different AoA and obtain the lift and drag coefficients C_L and C_D. We use the software XFOIL [89] for the computations. Despite the non-smooth nature of the eroded geometry, partly because of the limited size of the significant-erosion area, we were able to obtain converged and reasonable solutions for AoA values from 1° to 15°. Figure 15 shows the comparison of C_L and C_D for the original and eroded airfoil geometries with the experimental data [88]. The eroded airfoil has lower lift coefficient, higher drag, and thus, lower aerodynamic efficiency. Figure 15 also shows a reasonably good agreement between the computed and experimental data.

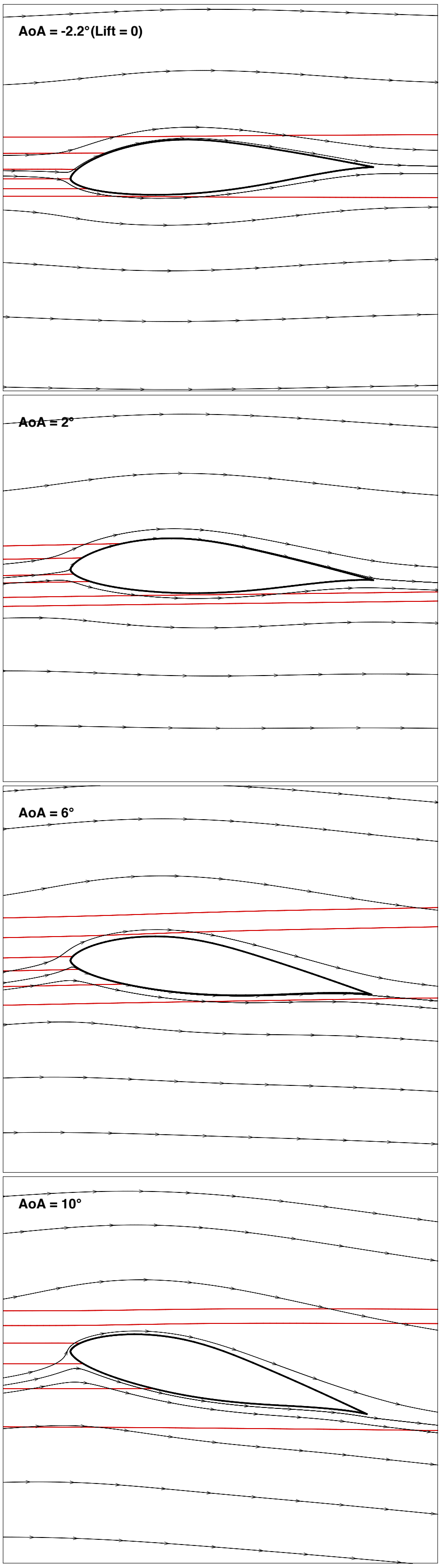

Fig. 8 Rain erosion. Streamlines (*black*) and mass center trajectories for the particle clouds (*red*) at the middle section of the blade strip at different AoA, for the first evolution step (with the initial geometry)

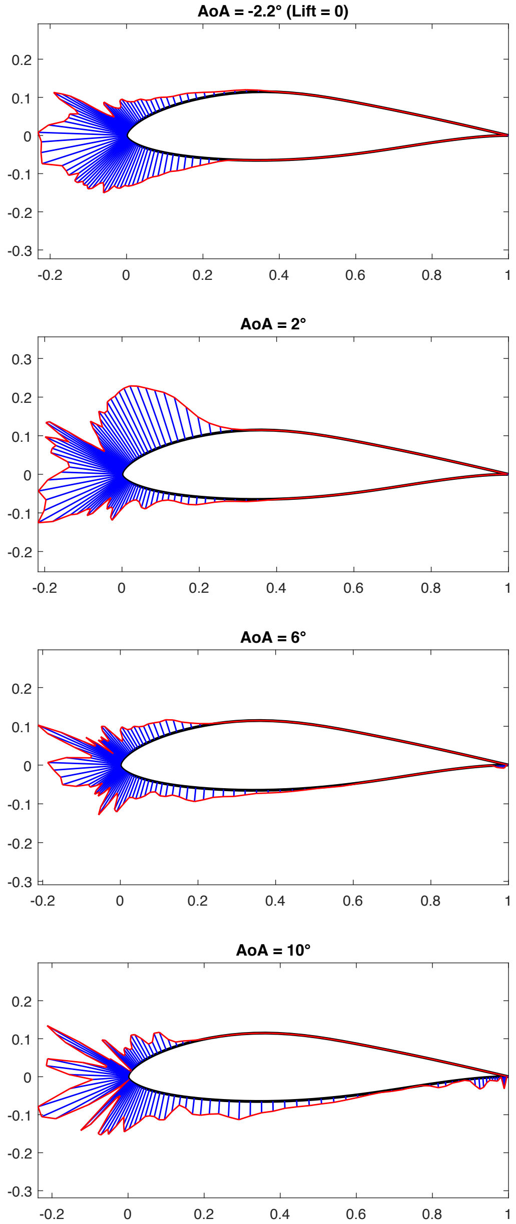

Fig. 9 Rain erosion. Erosion patterns at the middle section of the blade strip at different AoA, for the first evolution step

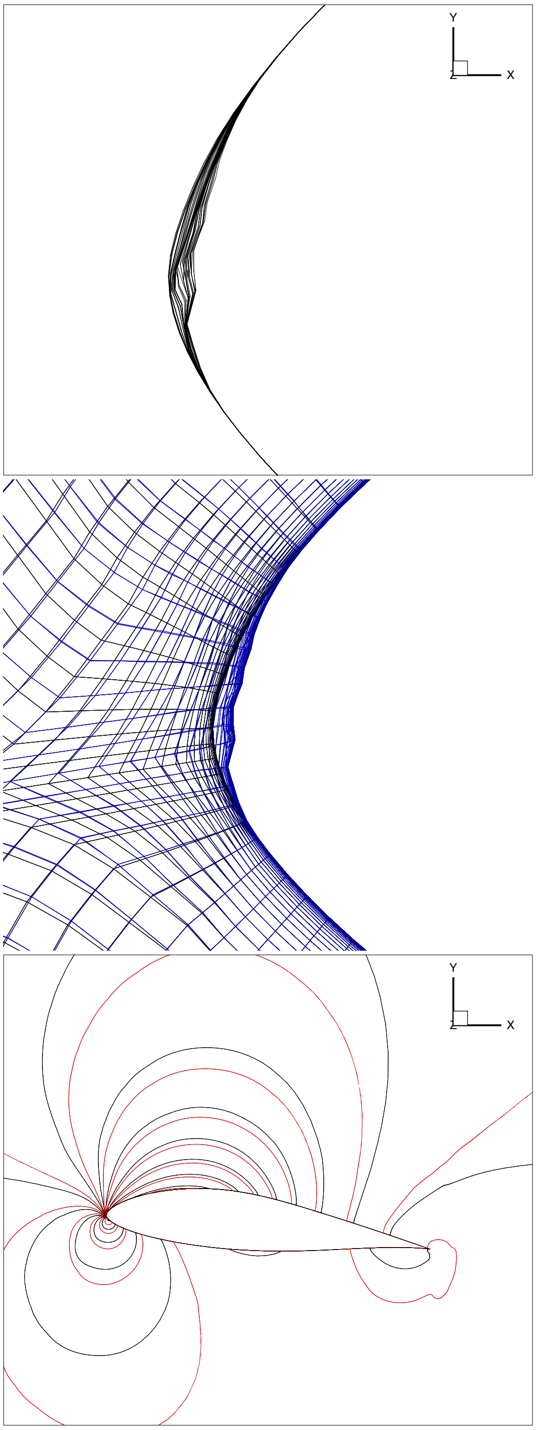

Fig. 10 Sand erosion. Geometry evolution, initial and final meshes, and pressure for the first (*black*) and last (*red*) evolution steps, at the middle section of the blade strip

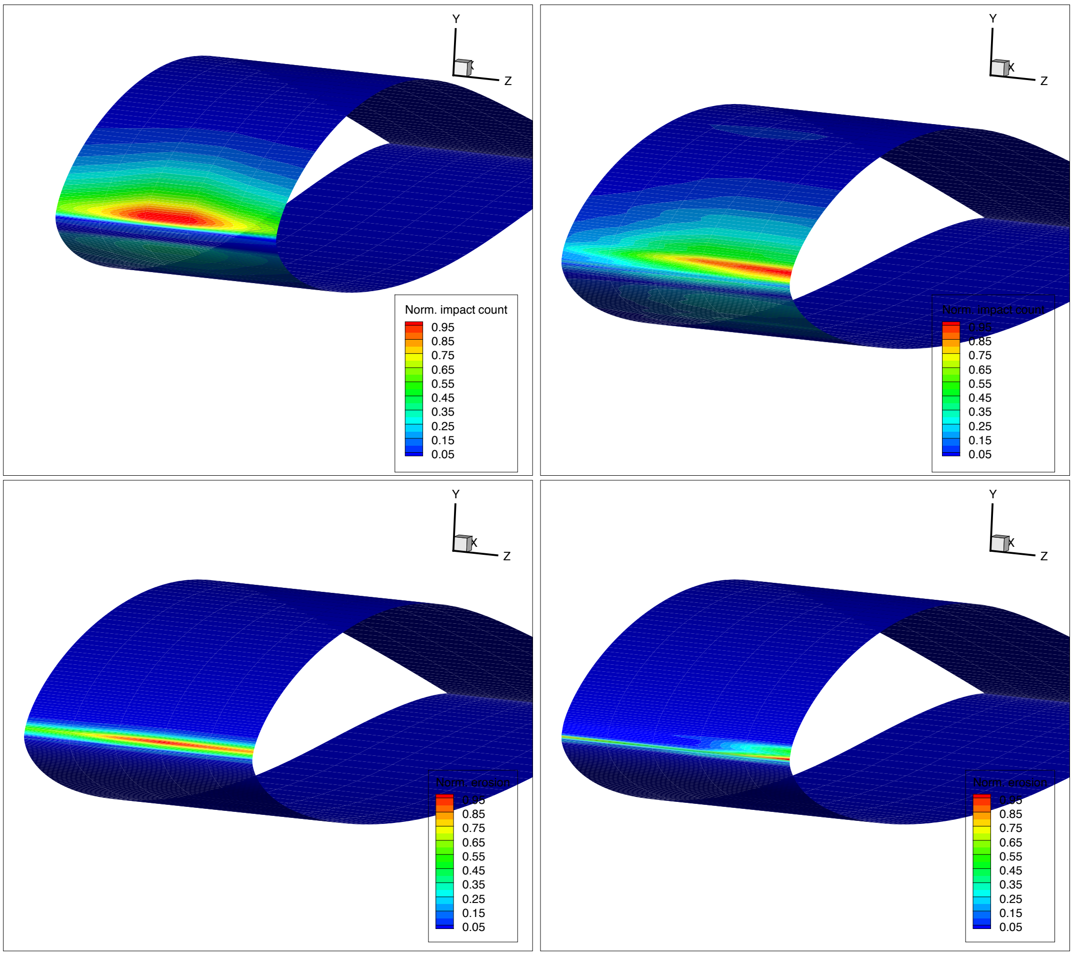

Fig. 11 Sand erosion. Normalized impact count (*top*) and erosion (*bottom*), at the end of the first (*left*) and last (*right*) evolution steps

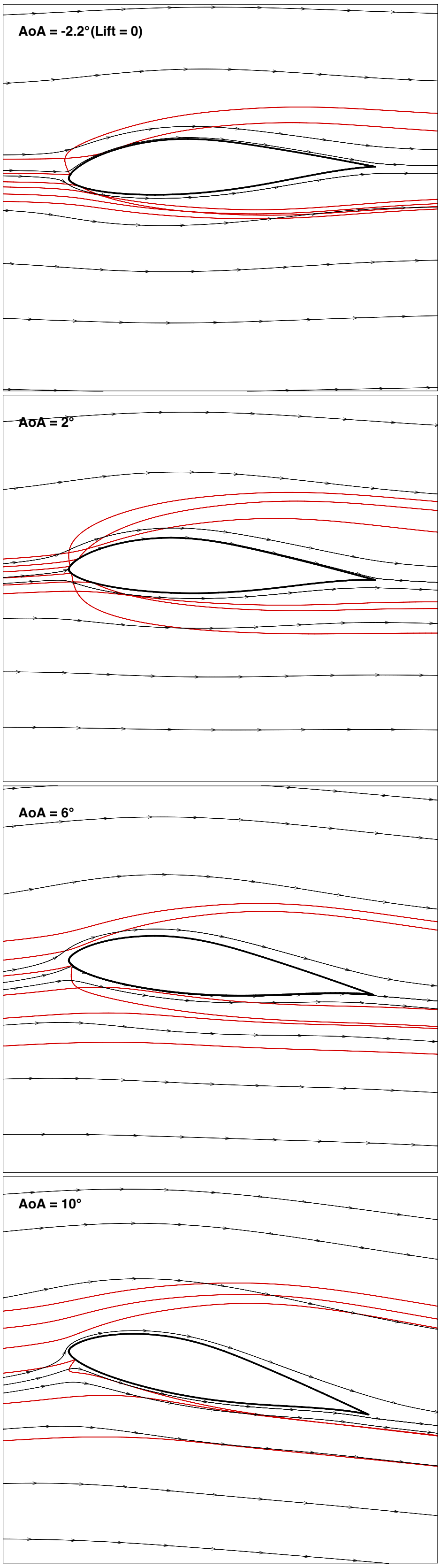

Fig. 12 Sand erosion. Streamlines (*black*) and mass center trajectories for the particle clouds (*red*) at the middle section of the blade strip at different AoA, for the first evolution step (with the initial geometry)

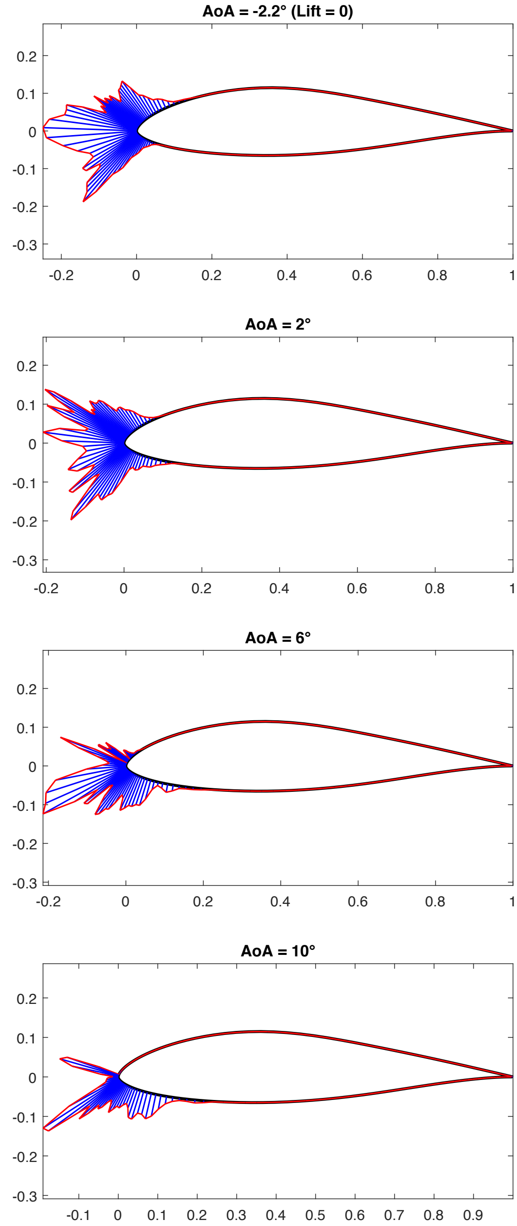

Fig. 13 Sand erosion. Erosion patterns at the middle section of the blade strip at different AoA, for the first evolution step

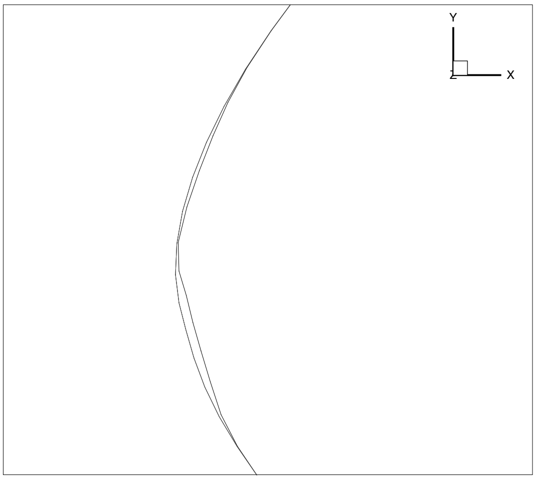

Fig. 14 Rain erosion field application. Geometry at the middle section of the blade strip at the beginning and end of the evolution steps

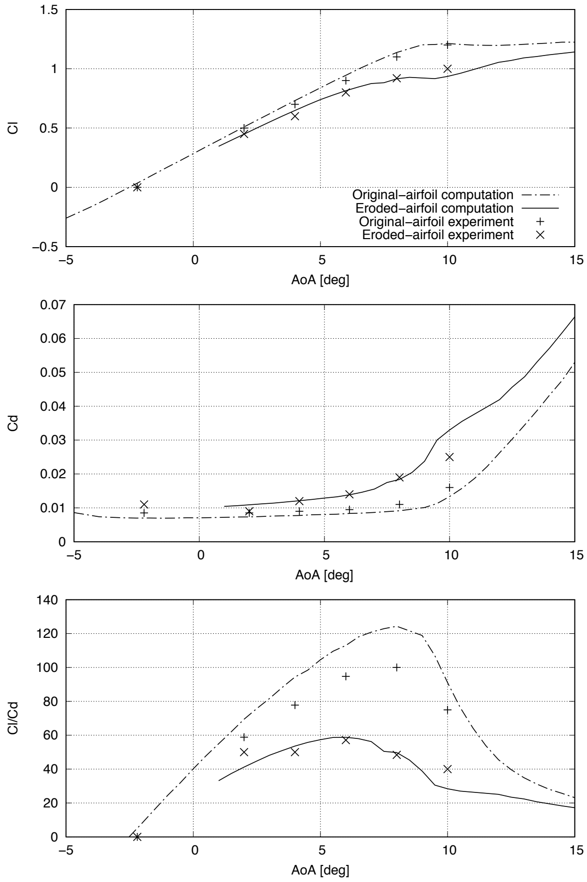

Fig. 15 Computed and experimental aerodynamic coefficients for the original and eroded airfoil geometries

9 Concluding remarks

We have presented an integrated method for computational analysis of wind-turbine blade rain and sand erosion. The analysis is valuable because rain and sand erosion, over long periods of time, can degrade the aerodynamic performance and therefore the power production. Being able to predict how the erosion evolves as time passes could lead to improvement of the wind-turbine technology and help engineers have a better understanding of the maintenance and protection requirements. The analysis is challenging because it involves rotating turbulent flows, large number of particles carried by the flow, turbulent dispersion of particles, and the blade geometry change due to the erosion has a very long time scale compared to the fluid–particle dynamics.

The main components of the integrated method are the SUPG and PSPG stabilizations, a finite element PCT method, an erosion model based on two time scales, and the SEMMT. The turbulent-flow nature of the analysis is handled with a RANS model and the SUPG/PSPG stabilization, complemented with the DRDJ stabilization. The particle-cloud trajectories are calculated based on the computed flow field and closure models defined for the turbulent dispersion of particles. One-way dependence is assumed between the flow and particle dynamics. Because a single time-marching simulation with the typical time-step size of the flow computation is not practical, we use a sequence of evolution steps to represent the impact of the erosion. A scale-up factor relates the erosions and particle counts in the evolution steps to those in the fluid–particle simulation. The factor is calculated based on a threshold erosion value that we expect to alter the blade aerodynamics from its current operation pattern or a threshold operation period that we expect to be long enough to alter the blade aerodynamics. As the blade geometry evolves due to the erosion, the mesh is updated with the SEMMT, which not only protects the smaller elements from mesh deformation, but also protects the thin layers of elements near the blade surface.

We presented computational analysis of rain and sand erosion for a wind-turbine blade strip, including a case with actual rainfall data and experimental aerodynamic data for eroded airfoil geometries. We showed that the analysis can provide valuable information, such as how the AoA influences the particle trajectories and erosion patterns, what part of the blade section is more vulnerable to erosion, and how long it takes for the erosion to reach a specified level. We also showed, with experimental verification, how the analysis can predict the aerodynamic-performance degradation due to the erosion.

Acknowledgment

This work was supported in part by Sapienza University of Rome “Progetti Grandi 2017” grant “Development of advanced modeling techniques for coupled multiphysics in open and ducted rotor fluid machines” - n. prot. RG11715C81D7D03A. The mathematical model and computational method parts of the work were also supported in part by Grant-in-Aid for Challenging Exploratory Research 16K13779 from JSPS and Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) (for the 5th author) and ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University (for the last author).

A Blade rainfall for the general case

In Section 8.3, the expressions given for the rainwater fraction and blade rainfall were based on the assumption that the rainfall is vertical, the blade velocity is horizontal, and the inflow plane is perpendicular to the blade velocity vector. Here we provide expressions for the general case, without these assumptions.

We define the raindrop velocity as a vector, $\mathbf{V}_R$, with the rain falling at an angle γ_R, measured from the vertical axis, as seen in Figure 16. The rainfall rate can be written as

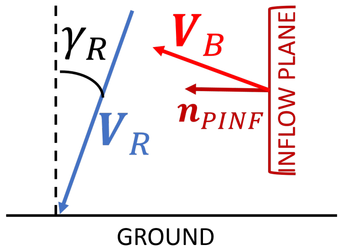

Fig. 16 Vectors of the blade rainfall for the general case

$$V_{RF} = \beta_R V_R \cos\gamma_R, \tag{82}$$

where $V_R = \|\mathbf{V}_R\|$. From that we obtain the rainwater fraction:

$$\beta_R = \frac{V_{RF}}{V_R \cos\gamma_R}. \tag{83}$$

We define the blade velocity also as a vector, $\mathbf{V}_B$, and the unit normal vector of the inflow plane is $\mathbf{n}_{PINF}$, as seen in Figure 16. The blade rainfall rate can then be written as

$$V_{BRF} = \beta_R (\mathbf{V}_B - \mathbf{V}_R) \cdot \mathbf{n}_{PINF} = \frac{V_{RF}}{V_R \cos\gamma_R} (\mathbf{V}_B - \mathbf{V}_R) \cdot \mathbf{n}_{PINF}, \tag{84}$$

where $V_B = \|\mathbf{V}_B\|$. With that, the annual rainfall for the blade is

$$H^Y_{BRF} = \frac{H^Y_{RF}}{V_R \cos\gamma_R} (\mathbf{V}_B - \mathbf{V}_R) \cdot \mathbf{n}_{PINF}. \tag{85}$$

If the inflow plane is perpendicular to the blade velocity vector, $\mathbf{n}_{PINF} = \frac{\mathbf{V}_B}{V_B}$, which is typically the case, then the annual rainfall for the blade becomes

$$H^Y_{BRF} = \frac{H^Y_{RF}}{V_R \cos\gamma_R} \left(V_B - \frac{\mathbf{V}_B}{V_B} \cdot \mathbf{V}_R \right) = \frac{H^Y_{RF}}{\cos\gamma_R} \left(\frac{V_B}{V_R} - \frac{\mathbf{V}_B \cdot \mathbf{V}_R}{V_B V_R} \right). \tag{86}$$

References

1. K. Wood, "Blade repair: Closing the maintenance gap", *Composites Technology*, (2011).
2. 3M. "A 3m study is the first to show the effects of erosion on wind turbine efficiency. [online], 2011 www.pressebox.com/pressrelease/3m- deutschland- gmbh/a-3m- study- is- the- first- to- show- the- effects- of- erosion- on-wind- turbine- efficiency/boxid/445007", 2011.
3. A. Castorrini, A. Corsini, F. Rispoli, P. Venturini, K. Takizawa, and T.E. Tezduyar, "SUPG/PSPG computational analysis of rain erosion in wind-turbine blades", in Y. Bazilevs and K. Takizawa, editors, *Advances in Computational Fluid–Structure Interaction and Flow Simulation: New Methods and Challenging Computations*, Modeling and Simulation in Science, Engineering and Technology, 77–96, Springer, 2016, ISBN 978-3-319-40825-5.
4. A. Castorrini, A. Corsini, F. Rispoli, P. Venturini, K. Takizawa, and T.E. Tezduyar, "Computational analysis of wind-turbine blade rain erosion", *Computers & Fluids*, **141** (2016) 175–183, doi: 10.1016/j.compfluid.2016.08.013.
5. A. Corsini, A. Castorrini, E. Morei, F. Rispoli, F. Sciulli, and P. Venturini, "Modeling of rain drop erosion in a multi-MW wind turbine", in *ASME Turbo Expo*, Montreal, Canada, (2015).
6. M.F. Hussein and W. Tabakoff, "Computation and plotting of solid particle flow in rotating cascades", *Computers & Fluids*, **2** (1974) 1–15.
7. A.A. Hamed, W. Tabakoff, R.B. Rivir, K. Das, and P. Arora, "Turbine blade surface deterioration by erosion", *Journal of Turbomachinery*, **127** (2005) 445–452.
8. A. Hamed, W. Tabakoff, R. Swar, D. Shin, N. Woggon, and R. Miller, "Combined experimental and numerical simulations of thermal barrier coated turbine blades erosion", (2013).
9. A. Ghenaiet, "Numerical study of sand ingestion through a ventilating system", in *Proceedings of the World Congress on Engineering*, volume 2, (2009) 1–3.
10. M. Suzuki and M. Yamamoto, "Numerical simulation of sand erosion phenomena in a single-stage axial compressor", *Journal of Fluid Science and Technology*, **6** (2011) 98–113.
11. A.N. Brooks and T.J.R. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations", *Computer Methods in Applied Mechanics and Engineering*, **32** (1982) 199–259.
12. T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", *Advances in Applied Mechanics*, **28** (1992) 1–44, doi: 10.1016/S0065-2156(08)70153-4.
13. L.L. Baxter and P.J. Smith, "Turbulent dispersion of particles: the stp model", *Energy and Fuels*, **7** (1993) 852–859.
14. P. Venturini, *Modelling of particle-wall deposition in two-phase gas-solid flows*, Ph.D. thesis, Sapienza University of Rome, 2010.
15. L. Cardillo, A. Corsini, G. Delibra, F. Rispoli, A.G. Sheard, and P. Venturini, "Simulation of particle-laden flows in a large centrifugal fan for erosion prediction", in *58th American Society of Mechanical Engineers Turbine and Aeroengine Congress*, Düsseldorf, Germany, (2015).
16. S.K. Kaer, *Numerical investigation of ash deposition in straw-fired furnaces*, Ph.D. thesis, Aalborg University, Denmark, 2001.
17. A. Corsini, A. Marchegiani, F. Rispoli, and P. Venturini, "Predicting blade leading edge erosion in an axial induced draft fan", *ASME Journal of Engineering for Gas Turbines and Power*, **134** (1993).
18. T.E. Tezduyar and Y.J. Park, "Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 307–325, doi: 10.1016/0045-7825(86)90003-4.
19. A. Corsini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "Improved discontinuity-capturing finite element techniques for reaction effects in turbulence computation", *Computational Mechanics*, **38** (2006) 356–364, doi: 10.1007/s00466-006-0045-x.
20. A. Corsini, C. Menichini, F. Rispoli, A. Santoriello, and T.E. Tezduyar, "A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms", *Journal of Applied Mechanics*, **76** (2009) 021211, doi: 10.1115/1.3062967.
21. A. Corsini, C. Iossa, F. Rispoli, and T.E. Tezduyar, "A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors", *Computational Mechanics*, **46** (2010) 159–167, doi: 10.1007/s00466-009-0441-0.
22. A. Corsini, F. Rispoli, and T.E. Tezduyar, "Stabilized finite element computation of NOx emission in aero-engine combustors", *International Journal for Numerical Methods in Fluids*, **65** (2011) 254–270, doi: 10.1002/fld.2451.
23. T.E. Tezduyar and T.J.R. Hughes, "Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations", in *Proceedings of AIAA 21st Aerospace Sciences Meeting*, AIAA Paper 83-0125, Reno, Nevada, (1983).
24. T.J.R. Hughes and T.E. Tezduyar, "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations", *Computer Methods in Applied Mechanics and Engineering*, **45** (1984) 217–284, doi: 10.1016/0045-7825(84)90157-9.
25. T.E. Tezduyar and D.K. Ganjoo, "Petrov-Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: Applications to transient convection-diffusion problems", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 49–71, doi: 10.1016/0045-7825(86)90023-X.
26. G.J. Le Beau, S.E. Ray, S.K. Aliabadi, and T.E. Tezduyar, "SUPG finite element computation of compressible flows with the entropy and conservation variables formulations", *Computer Methods in Applied Mechanics and Engineering*, **104** (1993) 397–422, doi: 10.1016/0045-7825(93)90033-T.
27. T.E. Tezduyar and Y. Osawa, "Finite element stabilization parameters computed from element matrices and vectors", *Computer Methods in Applied Mechanics and Engineering*, **190** (2000) 411–430, doi: 10.1016/S0045-7825(00)00211-5.
28. T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575, doi: 10.1002/fld.505.
29. T.E. Tezduyar, "Finite element methods for fluid dynamics with moving boundaries and interfaces", in E. Stein, R.D. Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, Chapter 17, Wiley, 2004, ISBN 978-0-470-84699-5.
30. T.E. Tezduyar, "Finite elements in fluids: Stabilized formulations and moving boundaries and interfaces", *Computers & Fluids*, **36** (2007) 191–206, doi: 10.1016/j.compfluid.2005.02.011.
31. T.E. Tezduyar and M. Senga, "Stabilization and shock-capturing parameters in SUPG formulation of compressible flows", *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 1621–1632, doi: 10.1016/j.cma.2005.05.032.
32. T.E. Tezduyar and M. Senga, "SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing", *Computers & Fluids*, **36** (2007) 147–159, doi: 10.1016/j.compfluid.2005.07.009.
33. T.E. Tezduyar, M. Senga, and D. Vicker, "Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing", *Computational Mechanics*, **38** (2006) 469–481, doi: 10.1007/s00466-005-0025-6.
34. T.E. Tezduyar and S. Sathe, "Enhanced-discretization selective stabilization procedure (EDSSP)", *Computational Mechanics*, **38** (2006) 456–468, doi: 10.1007/s00466-006-0056-7.
35. T.E. Tezduyar and S. Sathe, "Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques", *International Journal for Numerical Methods in Fluids*, **54** (2007) 855–900, doi: 10.1002/fld.1430.
36. F. Rispoli, A. Corsini, and T.E. Tezduyar, "Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (DCDD)", *Computers & Fluids*, **36** (2007) 121–

126, doi: 10.1016/j.compfluid.2005.07.004.

37. T.E. Tezduyar, S. Ramakrishnan, and S. Sathe, “Stabilized formulations for incompressible flows with thermal coupling”, *International Journal for Numerical Methods in Fluids*, **57** (2008) 1189–1209, doi: 10.1002/fld.1743.
38. F. Rispoli, R. Saavedra, A. Corsini, and T.E. Tezduyar, “Computation of inviscid compressible flows with the V-SGS stabilization and YZβ shock-capturing”, *International Journal for Numerical Methods in Fluids*, **54** (2007) 695–706, doi: 10.1002/fld.1447.
39. Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, “YZβ discontinuity-capturing for advection-dominated processes with application to arterial drug delivery”, *International Journal for Numerical Methods in Fluids*, **54** (2007) 593–608, doi: 10.1002/fld.1484.
40. F. Rispoli, R. Saavedra, F. Menichini, and T.E. Tezduyar, “Computation of inviscid supersonic flows around cylinders and spheres with the V-SGS stabilization and YZβ shock-capturing”, *Journal of Applied Mechanics*, **76** (2009) 021209, doi: 10.1115/1.3057496.
41. M.-C. Hsu, Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, “Improving stability of stabilized and multiscale formulations in flow simulations at small time steps”, *Computer Methods in Applied Mechanics and Engineering*, **199** (2010) 828–840, doi: 10.1016/j.cma.2009.06.019.
42. A. Corsini, F. Rispoli, and T.E. Tezduyar, “Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique”, *Journal of Applied Mechanics*, **79** (2012) 010910, doi: 10.1115/1.4005060.
43. A. Corsini, F. Rispoli, A.G. Sheard, and T.E. Tezduyar, “Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations”, *Computational Mechanics*, **50** (2012) 695–705, doi: 10.1007/s00466-012-0789-4.
44. P.A. Kler, L.D. Dalcin, R.R. Paz, and T.E. Tezduyar, “SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems”, *Computational Mechanics*, **51** (2013) 171–185, doi: 10.1007/s00466-012-0712-z.
45. A. Corsini, F. Rispoli, A.G. Sheard, K. Takizawa, T.E. Tezduyar, and P. Venturini, “A variational multiscale method for particle-cloud tracking in turbomachinery flows”, *Computational Mechanics*, **54** (2014) 1191–1202, doi: 10.1007/s00466-014-1050-0.
46. F. Rispoli, G. Delibra, P. Venturini, A. Corsini, R. Saavedra, and T.E. Tezduyar, “Particle tracking and particle–shock interaction in compressible-flow computations with the V-SGS stabilization and YZβ shock-capturing”, *Computational Mechanics*, **55** (2015) 1201–1209, doi: 10.1007/s00466-015-1160-3.
47. K. Takizawa, T.E. Tezduyar, S. McIntyre, N. Kostov, R. Kolesar, and C. Habluetzel, “Space–time VMS computation of wind-turbine rotor and tower aerodynamics”, *Computational Mechanics*, **53** (2014) 1–15, doi: 10.1007/s00466-013-0888-x.
48. K. Takizawa, T.E. Tezduyar, and T. Kuraishi, “Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires”, *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2227–2255, doi: 10.1142/S0218202515400072.
49. K. Takizawa, T.E. Tezduyar, H. Mochizuki, H. Hattori, S. Mei, L. Pan, and K. Montel, “Space–time VMS method for flow computations with slip interfaces (ST-SI)”, *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2377–2406, doi: 10.1142/S0218202515400126.
50. K. Takizawa, T.E. Tezduyar, and Y. Otoguro, “Stabilization and discontinuity-capturing parameters for space–time flow computations with finite element and isogeometric discretizations”, *Computational Mechanics*, **62** (2018) 1169–1186, doi: 10.1007/s00466-018-1557-x.
51. Y. Otoguro, K. Takizawa, T.E. Tezduyar, K. Nagaoka, and S. Mei, “Turbocharger turbine and exhaust manifold flow computation with the Space–Time Variational Multiscale Method and Isogeometric Analysis”, *Computers & Fluids*, published online, DOI: 10.1016/j.compfluid.2018.05.019, May 2018, doi: 10.1016/j.compfluid.2018.05.019.
52. T. Kuraishi, K. Takizawa, and T.E. Tezduyar, “Tire aerodynamics with actual tire geometry, road contact and tire deformation”, *Computational Mechanics*, published online, DOI: 10.1007/s00466-018-1642-1, October 2018, doi: 10.1007/s00466-018-1642-1.
53. T.J.R. Hughes, “Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods”, *Computer Methods in Applied Mechanics and Engineering*, **127** (1995) 387–401.
54. M.H. Keegan, D. Nash, and M. Stack, “On erosion issues associated with the leading edge of wind turbine blades”, *Journal of Physics D: Applied Physics*, **46** (2013) 383001.
55. G.S. Springer, C.-I. Yang, and P.S. Larsen, “Analysis of rain erosion of coated materials”, *Journal of Composite Materials*, **8** (1974) 229–252.
56. Y.I. Oka, K. Okamura, and T. Yoshida, “Practical estimation of erosion damage caused by solid particle impact: Part 1: Effects of impact parameters on a predictive equation”, *Wear*, **259** (2005) 95–101.
57. A. Castorrini, A. Corsini, F. Morabito, F. Rispoli, and P. Venturini, “Numerical simulation with adaptive boundary method for predicting time evolution of erosion processes”, in *ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition*. American Society of Mechanical Engineers, (2017) V02DT48A019–V02DT48A019.
58. A. Castorrini, P. Venturini, A. Corsini, and F. Rispoli, “Numerical simulation of the blade aging process in an induced draft fan due to long time exposition to fly ash particles”, *Journal of Engineering for Gas Turbines and Power*, **141** (2019) 011025.
59. T. Tezduyar, “Finite element interface-tracking and interface-capturing techniques for flows with moving boundaries and interfaces”, in *Proceedings of the ASME Symposium on Fluid-Physics and Heat Transfer for Macro- and Micro-Scale Gas-Liquid and Phase-Change Flows (CD-ROM)*, ASME Paper IMECE2001/HTD-24206, ASME, New York, New York, (2001).
60. T.E. Tezduyar, “Stabilized finite element formulations and interface-tracking and interface-capturing techniques for incompressible flows”, in M.M. Hafez, editor, *Numerical Simulations of Incompressible Flows*, World Scientific, New Jersey, (2003) 221–239.
61. K. Stein, T.E. Tezduyar, and R. Benney, “Automatic mesh update with the solid-extension mesh moving technique”, *Computer Methods in Applied Mechanics and Engineering*, **193** (2004) 2019–2032, doi: 10.1016/j.cma.2003.12.046.
62. T.E. Tezduyar, S. Sathe, R. Keedy, and K. Stein, “Space–time finite element techniques for computation of fluid–structure interactions”, *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 2002–2027, doi: 10.1016/j.cma.2004.09.014.
63. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, February 2013, ISBN 978-0470978771.
64. T.E. Tezduyar, M. Behr, S. Mittal, and A.A. Johnson, “Computation of unsteady incompressible flows with the finite element methods: Space–time formulations, iterative strategies and massively parallel implementations”, in *New Methods in Transient Analysis*, PVP-Vol.246/AMD-Vol.143, ASME, New York, (1992) 7–24.
65. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal, “Parallel finite-element computation of 3D flows”, *Computer*, **26** (1993) 27–36, doi: 10.1109/2.237441.
66. A.A. Johnson and T.E. Tezduyar, “Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces”, *Computer Methods in Applied Mechanics and Engineering*, **119** (1994) 73–94, doi: 10.1016/0045-7825(94)00077-8.
67. K. Stein, T. Tezduyar, and R. Benney, “Mesh moving techniques for fluid–structure interactions with large displacements”, *Journal of Applied Mechanics*, **70** (2003) 58–63, doi: 10.1115/1.1530635.
68. A. Corsini and F. Rispoli, “Flow analyses in a high-pressure axial ventilation fan with a non-linear eddy viscosity closure”, *Interna-*

tional Journal of Heat and Fluid Flow, **17** (2005) 108–155.

69. T.J. Craft, B.E. Launder, and K. Suga, "Development and application of a cubic eddy-viscosity model of turbulence", *International Journal of Heat and Fluid Flow*, **17** (1996) 108–155.
70. S. Lain and M. Sommerfeld, "Turbulence modulation in dispersed two-phase flow laden with solids from a lagrangian perspective", *International Journal of Heat and Fluid Flow*, **24** (2003) 616–625.
71. T.E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher, "Space–time finite element computation of complex fluid–structure interactions", *International Journal for Numerical Methods in Fluids*, **64** (2010) 1201–1218, doi: 10.1002/fld.2221.
72. K. Takizawa, T.E. Tezduyar, and H. Hattori, "Computational analysis of flow-driven string dynamics in turbomachinery", *Computers & Fluids*, **142** (2017) 109–117, doi: 10.1016/j.compfluid.2016.02.019.
73. K. Komiya, T. Kanai, Y. Otoguro, M. Kaneko, K. Hirota, Y. Zhang, K. Takizawa, T.E. Tezduyar, M. Nohmi, T. Tsuneda, M. Kawai, and M. Isono, "Computational analysis of flow-driven string dynamics in a pump and residence time calculation", in *Proceedings of the 29th IAHR Symposium on Hydraulic Machinery and Systems*, Kyoto, Japan, (2018).
74. T. Kanai, K. Takizawa, T.E. Tezduyar, K. Komiya, M. Kaneko, K. Hirota, M. Nohmi, T. Tsuneda, M. Kawai, and M. Isono, "Methods for computation of flow-driven string dynamics in a pump and residence time", *Mathematical Models and Methods in Applied Sciences*, to appear, 2019.
75. L.L. Baxter, *Turbulent transport of particles*, Ph.D. thesis, Brigham Young University, 1989.
76. L.P. Wang, *On the dispersion of heavy particles by turbulent motion*, Ph.D. thesis, Washington State University, 1990.
77. L.J. Litchford and S.M. Jeng, "Efficient statistical transport model for turbulent particle dispersion in sprays", *AIAA Journal*, **29** (1991) 1443–1451.
78. S. Jain, *Three-dimensional simulation of turbulent particle dispersion*, Ph.D. thesis, University of Utah, 1995.
79. D. Borello, P. Venturini, F. Rispoli, and G.Z.R. Saavedra, "Prediction of multiphase combustion and ash deposition within a biomass furnace", *Applied Energy*, **101** (2013) 413–422.
80. P. Venturini, D. Borello, C.V. Iossa, D. Lentini, and F. Rispoli, "Modelling of multiphase combustion and deposit formation and deposit formation in a biomass-fed boiler", *Energy*, **35** (2010) 3008–3021.
81. V. Armenio and V. Fiorotto, "The importance of the forces acting on particles in turbulent flows", *Physics of Fluids*, **13** (2001) 2437–2440.
82. L. Schiller and A. Naumann, "Uber die grundlegenden berechnungen bei der schwekraftaubereitung", *Zeitschrift des Vereines Deutscher Ingenieure*, **77** (1933) 318–320.
83. P.J. Smith, "3-D turbulent particle dispersion submodel development", Quarterly progress report, Department of Energy, Pittsburgh Energy Technology Center, 1991.
84. A. Corsini, F. Rispoli, and A. Santoriello, "A variational multiscale high-order finite element formulation for turbomachinery flow computations", *Computer Methods in Applied Mechanics and Engineering*, **194** (2005) 4797–4823.
85. S. Arjula, A. Harsha, and M. Ghosh, "Solid-particle erosion behavior of high-performance thermoplastic polymers", *Journal of Materials Science*, **43** (2008) 1757–1768.
86. T.E. Tezduyar, K. Takizawa, and Y. Bazilevs, "Fluid–structure interaction and flows with moving boundaries and interfaces", in E. Stein, R.D. Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics Second Edition*, Part 2 Fluids, Wiley, published online, December 2017, ISBN 9781119003793.
87. K. Stein and T. Tezduyar, "Advanced mesh update techniques for problems involving large displacements", in *Proceedings of the Fifth World Congress on Computational Mechanics*, On-line publication: Paper-ID: 81489, http://www.researchgate.net/publication/303737884/, Vienna, Austria, (2002).
88. A. Sareen, C.A. Sapre, and M.S. Selig, "Effects of leading edge erosion on wind turbine blade performance", *Wind Energy*, **17** (2014) 1531–1542.
89. M. Drela and H. Youngren, "Xfoil subsonic airfoil development system", *Software Package, available online at http://web. mit. edu/drela/Public/web/xfoil [retrieved Feb. 2011]*, (2008).