Computers and Fluids 301 (2025) 106775

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

# A dynamic immersed boundary method for simulating an adaptive nozzle generating discrete wind gusts

K. Boulbrachene, M. Breuer *

*Professur für Strömungsmechanik, Helmut-Schmidt-Universität Hamburg, D-22043 Hamburg, Germany*

ARTICLE INFO

*Keywords:*
Horizontal wind gusts
Numerical wind tunnel
Immersed boundary method
Moving-least-squares
Large-eddy simulation
Fluid–structure interaction

ABSTRACT

The recently developed wind gust generator, the *adaptive nozzle* (Wood and Breuer, 2025) , features a nozzle with a fully rotatable upper contour, enabling a smooth gust generation with low unwanted flow disturbances. While preserving the underlying gust-generation principle of its predecessor, the new design significantly reduces pressure losses caused by flow blockage and preserves the original horizontal trajectory of the flow along the streamwise direction. While experiments have validated the improved design, a comprehensive numerical analysis is crucial to resolve the three-dimensional flow fields across the entire computational domain. This shall also facilitate capturing the resulting transient aerodynamic loads on a wind tunnel specimen — quantities difficult to measure experimentally. To accurately capture the complex flow dynamics, high-fidelity large-eddy simulations are conducted, modeling the nozzle's upper contour as a dynamic immersed boundary (IB). A curvilinear Eulerian grid is employed to ensure both efficient and precise spatial resolution of the problem. The moving least-squares (MLS) version of the direct forcing IB approach (Vanella and Balaras, 2009) is used to construct an IB kernel for each Lagrangian marker. Additionally, the MLS approach is also applied to construct one-sided kernel functions for Lagrangian points near the boundaries of the computational domain. Challenges related to the efficient IB simulation on curvilinear grids are addressed, and a solution is proposed within the MLS framework. The predicted results are analyzed in detail and validated against the experimental data by Wood and Breuer (2025) , providing insights into the effectiveness of the new design in generating controlled wind gusts.

## 1. Introduction

Wind gusts, a common yet powerful meteorological phenomenon, pose a significant challenge to engineers due to their potentially devastating impact on structures. In particular, lightweight constructions made of thin membranes such as large umbrellas, tents, and stadium roofs are highly susceptible to sudden wind loads. Short-term gusts can critically affect their stability and fatigue strength. To gain deeper insight into the interaction between strong wind gusts and such constructions, direct measurements on real structures may seem like the most straightforward approach (see, e.g., [1]). However, conducting large-scale experiments presents considerable challenges: simultaneous measurements of fluid flow and structural deformation at real scales are not only complex and costly but also depend on unpredictable natural wind conditions. One must set up the measurement equipment and wait for suitable weather conditions, making controlled investigations nearly impossible.

As an alternative, wind tunnel experiments at reduced scales provide a viable and practical solution. However, this requires the ability to generate wind gusts artificially under controlled conditions. Developing reliable gust-generation methods in laboratory settings is therefore essential for advancing the understanding of fluid–structure interactions (FSI) and improving the resilience of structures subjected to extreme wind events. Li et al. [2] conducted a comprehensive review on experimental methods for investigating the impact of downburst winds on civil infrastructures. Gust fronts generated by downbursts, characterized by strong horizontal velocities, are closely related to horizontal wind gusts. The study categorized the experimental methods into three types: (*i*) standalone simulators (gravity-current-based; see [3] or impinging-jet-based; see [4]), (*ii*) modified boundary layer wind tunnels with active elements (e.g., rotatable plates or movable gates; see [5,6]), and (*iii*) multi-fan wind tunnels with either active [7] or hybrid active–passive [8] turbulence control.

Among these, modified boundary layer wind tunnels provide a cost-effective approach for studying the effect of wind gusts on structures. Recently, Wood et al. [9] developed a novel artificial wind gust generator (WGG) falling within this category, known as the *paddle*. By

* Corresponding author.
*E-mail address:* breuer@hsu-hh.de (M. Breuer).

https://doi.org/10.1016/j.compfluid.2025.106775
Received 21 May 2025; Received in revised form 8 July 2025; Accepted 27 July 2025
Available online 6 August 2025

rapidly and partially obstructing the outlet of the wind tunnel nozzle, the paddle induces a significant acceleration of the flow, generating a controlled horizontal gust. This gust generator has already been successfully applied in a follow-up experimental study by Breuer and Neumann [10], where synchronous high-speed measurements were conducted on a flexible T-structure subjected to wind gust loading.

Both experimental studies have been numerically reproduced in Boulbrachene and Breuer [11] and Breuer et al. [12], respectively. In Boulbrachene and Breuer [11], the immersed boundary (IB) method with a direct forcing approach was implemented into a finite-volume flow solver to model the translation motion of the paddle. The numerical results demonstrated excellent agreement with the measurements of Wood et al. [9] at key monitoring points within the flow field, validating the robustness and accuracy of the simulation approach. Similarly, in Breuer et al. [12], a numerical counterpart of the experimental investigation on the T-structure under wind gust loading was developed. This was achieved by combining the IB-based gust generation methodology with a partitioned FSI approach using the Arbitrary Lagrangian–Eulerian (ALE) framework. The study successfully replicated both the gust generation process in the wind tunnel and the complex FSI problem of the thin flexible structure interacting with the highly unsteady turbulent flow.

In the meantime, Wood and Breuer [13] proposed an enhanced WGG design, known as the *adaptive nozzle*. The improved design addresses the key drawbacks of the paddle setup, particularly the high pressure loss in the nozzle due to flow blockage and the pronounced detached shear layer caused by flow separation at the bottom edge of the plate. Compared to the paddle design, the adaptive nozzle significantly reduced velocity fluctuations on the falling flank of the generated gust. Additionally, it achieved a shorter restitution time – defined as the period during which the wake flow recovers from the velocity undershoot to its initial state – enhancing the overall efficiency and stability of the generated gust. A detailed description of the setup and the measuring techniques employed is provided in Section 2.

While wind tunnel experiments offer valuable insights, they are often constrained by the spatial and temporal resolution of the measurement techniques applied, limiting the ability to capture complex three-dimensional flow structures and transient aerodynamic loads across the entire test section. A numerical study can provide a more detailed understanding of the fundamental mechanisms governing the adaptive nozzle's performance. Simulations enable a comprehensive analysis of the entire flow field, including regions that are challenging to access experimentally, such as the interior of the nozzle. By leveraging high-fidelity numerical methods, it becomes possible to quantify velocity fluctuations, pressure distributions, and turbulence characteristics in great detail.

Since the experimental gust generator relies on a nozzle with a movable upper contour, accurately capturing this feature in numerical flow simulations is essential. A viable approach is the ALE method, which allows the fluid mesh to adapt to the contour's motion while enforcing the no-slip boundary condition at the FSI interface at each time step. While the ALE method effectively resolves flow dynamics along moving interfaces, it leads to significant challenges in FSI problems involving large deformations. One major drawback is the computational expense associated with frequent grid adaptations [14], which can substantially increase the simulation time. Moreover, excessive grid adaptation can degrade the grid quality – compromising properties such as orthogonality and smoothness – which may ultimately affect the accuracy and stability of the numerical results, especially when these are based on large-eddy simulations.

An alternative approach is the immersed boundary method (IBM), originally introduced by Peskin [15], which employs overlapping discretizations for the fluid and structural domains. The no-slip boundary condition at the fluid–solid interface is enforced by introducing an appropriate boundary forcing term. Among the various IBM formulations available in the literature (see, e.g., Mittal and Iaccarino [16]), the present study adopts the continuous forcing IBM (diffuse-interface IB) with the direct forcing approach of Uhlmann [17], which is particularly well-suited for handling moving immersed boundaries. In this method, boundary forces are computed on Lagrangian markers rather than Eulerian grid points. These forces are then distributed over a few neighboring Eulerian cells using a regularized delta function kernel (IB kernel), ensuring smoother hydrodynamic force distributions.

In his foundational work, Peskin [15] analytically derived a four-point IB kernel based on a set of postulates designed to ensure several essential characteristics. Others [18–20] adopted a similar approach to derive IB kernels with different support sizes (e.g., three-, five-, and six-point kernels). The postulates considered include polynomial reproduction conditions (also known as discrete moment conditions), grid translational invariance, and compact support. However, a key limitation of this analytical approach is its reliance on a uniform Cartesian Eulerian grid. In FSI problems, it is often beneficial to employ stretched or even curvilinear Eulerian grids to enhance computational efficiency and avoid unnecessary high spatial resolution in regions far from the object of interest. For instance, in this study it is preferable to cluster grid points in the vicinity of the upper contour and around the region corresponding to the motion range of it. This necessitates embedding the Lagrangian markers within a non-uniform curvilinear structured Eulerian grid, which renders the direct application of IB kernels designed for uniform grids unsuitable, as it would violate the integral conservation of force and moment.

To address this issue, the current study employs a numerical approach to derive the IB kernel, specifically the moving least-squares (MLS) method [21]. The MLS version of the direct forcing IB method was first introduced by Vanella and Balaras [22]. This approach constructs an interpolation kernel (also referred to as the MLS generating function) for each Lagrangian marker by solving a weighted least-squares problem defined over a set of neighboring Eulerian points. The method supports any non-uniform distribution of Eulerian points, and the resulting interpolation kernels satisfy the polynomial reproduction constraints. This, in turn, ensures the integral conservation of force and moment while preserving numerical accuracy and stability. Although Vanella and Balaras [22] applied the IB-MLS method to a uniform Cartesian grid, the approach is general and has been extended to unstructured grids (see, e.g., [23,24]).

In previous IB-MLS studies [22,23,25], a cubic spline weight function is transformed into a regularized kernel function that satisfies the zeroth- and the first-order discrete moment conditions. However, it is important to note that if the conditions imposed by the MLS construction are already satisfied by the weight function (e.g., when an IB kernel is used as a weight function on a uniform grid), the generating function obtained through the MLS procedure remains identical to the weight function [26,27]. However, Bale et al. [27] have highlighted that when the MLS construction is used to reduce the support of the weighting function, the resulting generating function generally differs from the original kernel. The authors leveraged the MLS construction in uniform Cartesian grids to generate one-sided (asymmetric) IB kernels by modifying the support of a standard two-sided (symmetric) weight function. The primary motivation for introducing one-sided IB kernels was to avoid spurious flows inside closed geometries in the diffuse-interface IB method and to address interactions between immersed structures and the boundaries of the computational domain. The latter objective holds particular significance for the current study, as the nozzle's upper contour is designed to traverse between two side walls, thereby restricting the applicability of symmetric kernels in the near-wall regions. While this study builds upon previous MLS work, it highlights the challenges associated with its direct application and instead proposes a new method for efficient and robust IB simulations on a curvilinear block-structured grid, as described in Section 3.3.2.

The structure of this paper is as follows: Section 2 presents the experimental setup of the second-generation wind gust generator. Section 3 details the numerical methodologies used to develop the computational framework, beginning with the fluid flow simulation approach

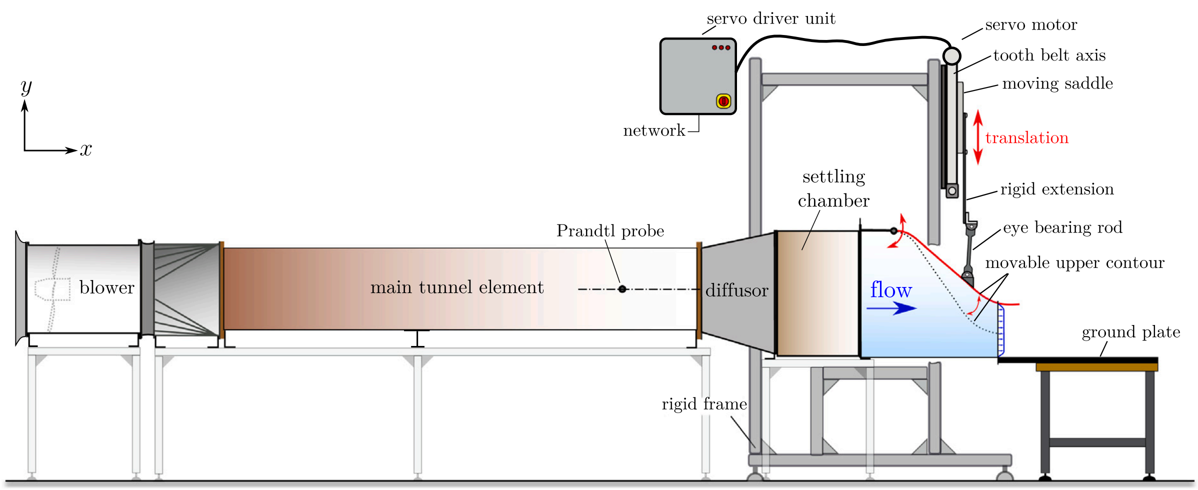



**Fig. 1.** Sketch of the Eiffel-type wind tunnel with the adaptive nozzle contour in red (Two sketches from Wood and Breuer [13] merged).

in Section 3.1, followed by the scheme for efficient tracking of Lagrangian markers in Section 3.2. The immersed boundary method, specifically adapted for simulating the moving nozzle on a curvilinear block-structured grid, is introduced in Section 3.3. Section 4 provides a comprehensive overview of the computational setup, including the kinematics of the nozzle's upper contour. Finally, Section 5 presents an analysis of the instantaneous flow field induced by the adaptive nozzle, along with a detailed comparison of phase-averaged experimental and numerical results for velocity fields and surface pressure signals.

## 2. Test case configuration and measuring techniques

The current numerical study builds upon the aforementioned experimental investigations conducted by Wood and Breuer [13,28], which serve as the foundational basis for validating the numerical simulations presented here. Therefore, it is essential to briefly outline the experimental setup and highlight the relevant physical parameters.

Fig. 1 shows a sketch of the Eiffel-type wind tunnel designed to study the adaptive nozzle concept. The key components of the wind tunnel are the blower, the main tunnel element, the diffusor, the settling chamber equipped with four honeycomb flow straighteners, and the adaptive nozzle with its movable upper contour. The nozzle outlet, in its initial position, features a cross-section of 0.25 m × 0.5 m, with a nozzle contraction ratio of 2:1. The flow exiting the nozzle enters the open laboratory space. A ground plate is attached to the underside of the nozzle's outlet, allowing the generated horizontal wind gusts to flow across its surface. By adjusting the rotation rate of the blower, the free-stream velocity in the open test section can range from 1.5 m/s to 20 m/s. The streamwise turbulence intensity at the nozzle's outlet has been determined to be 1.5 % [13].

Fig. 1 also depicts the operating principle of the gust generator, which relies on dynamically adjusting the outlet area of the wind tunnel nozzle. According to the principle of mass conservation, a reduction of the cross-section leads to an increase of the velocity at the nozzle's outlet. If this reduction of the cross-section occurs within a short time interval, a sharp rise of the velocity, resembling a wind gust, has to be expected. The general working principle was previously proved by Wood et al. [9] using the first-generation wind gust generator (the *paddle*). However, the paddle-based WGG presented notable drawbacks, including significant pressure losses inside the nozzle and a pronounced detached shear layer caused by flow separation at the paddle. Consequently, improved variants warrant consideration.

The key ingredients of the second-generation WGG are the movable upper contour attached to a rotatable bearing and the tooth belt axis driven by a servo motor. The saddle of the tooth belt axis moves vertically, both downward and upward, with a stroke length of $l_s$. All kinematic quantities (length, velocity, acceleration) are controlled using the FESTO automation software tool. An eye bearing rod, coupled with a rigid extension of the saddle, connects the tooth belt axis to the upper contour of the nozzle. Thus, this setup converts the saddle's vertical translational motion into the rotational motion of the upper contour around the bearing. By combining a downward stroke and an upward stroke of the saddle, the nozzle contour is dynamically adapted within a short time interval, leading to a temporary reduction followed by a re-enlargement of the outlet area. In Wood and Breuer [13], an outlet area reduction ratio of 37.2 % was chosen, offering a balanced compromise between the gust velocity achieved and the size of the effective area within the test section of the wind tunnel where the gust remains reproducible.

To comprehensively analyze the flow field during the gust event, Wood and Breuer [13] employed three different measurement techniques.

Particle-image velocimetry (PIV) was employed for synchronized measurements of the flow field in the $x$-$y$-symmetry plane of the test section. The synchronization procedure relied on a master trigger signal to align the operation of the gust generator with the flow field measurements, ensuring a common time frame. The measurement system utilized a classical mono-PIV configuration, comprising a Litron Nano L PIV laser (532 nm, 200 mJ) and a TSI Powerview CCD camera (6400 × 4400 pixels, 12-bit). DEHS tracer droplets, with a most penetrating particle size of 0.3 μm, were used for seeding on the suction side of the wind tunnel. To enable phase-averaging of the flow field, a total of 100 individual gust generations were recorded. Each gust event was resolved by 33 instants in time, ensuring an adequate temporal resolution of the phase-averaged data. With a temporal resolution of 25 ms, the total time span covered was 800 ms, corresponding to a dimensionless time of $t/\Delta t_{\rm gust} = 1.6$, where $\Delta t_{\rm gust} = 0.5$ s is the gust generation time defined in Section 4.3.

The second measuring device utilized a constant-temperature anemometer (CTA). This technique was applied to capture time-resolved velocity data for the generated gusts at 11 specific monitoring points within the wind tunnel depicted in Fig. 2. These monitoring locations were distributed along the streamwise direction at normalized coordinates $x/l = 0., 0.1, 0.2 \ldots 1.0$, at a fixed height of $y/l = 0.15$ above the ground plate. Note that the origin of the coordinate system is defined on the surface of the ground plate at a distance of $\Delta x = 0.05$ m from the nozzle outlet. The reference length was set to $l = 0.5$ m, corresponding to the width of the nozzle. Here phase-averaging of the velocity magnitude was conducted based on 46 gust events. The data were recorded to capture the characteristic development of the gust

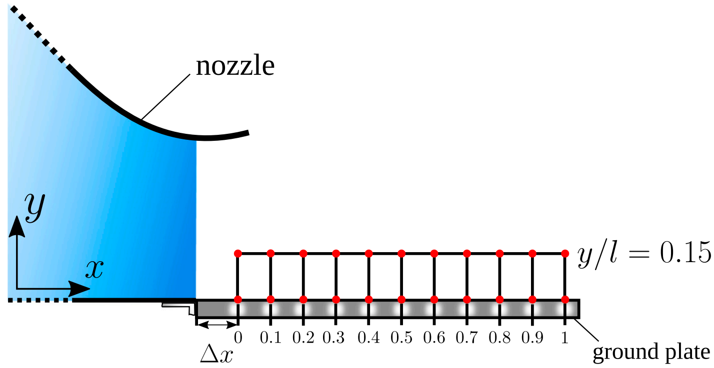



**Fig. 2.** Definition of the monitoring points on and above the ground plate.

at the defined monitoring points and to validate the conducted PIV measurements.

In addition to PIV and CTA, a pressure transducer (Scanivalve MSP4232 pressure scanner, 32 channels) was employed to record time histories of the surface pressure signals. The positions of the 11 monitoring points correspond to those of the CTA measurements but were now located directly on the ground plate ($y/l = 0$). Surface pressures at all monitoring points were recorded synchronously at a sampling rate of 1000 Hz. Similar to the CTA measurements, phase-averaging was performed using data from 46 wind gust events per measurement point.

## 3. Numerical framework

### 3.1. Flow solver

In this study, the incompressible Navier–Stokes equations are solved using the in-house flow solver FASTEST-3D [29]. The equations are spatially discretized via the finite-volume method on a curvilinear, block-structured, body-fitted mesh using a collocated cell-centered variable arrangement. Parallel computation is enabled through domain decomposition and explicit message passing with MPI. Surface and volume integrals are approximated using the midpoint rule, while face-centered flow variables are linearly interpolated from adjacent cell-centered values, yielding second-order accuracy in space. Convective fluxes are computed using the flux-blending technique [30,31], combining 3 % first-order upwind and 97 % second-order central schemes. To ensure a stable pressure-velocity coupling on non-staggered grids and avoid pressure oscillations, the momentum interpolation scheme of Rhie and Chow [32] is used.

Time integration is performed using a second-order accurate semi-implicit predictor–corrector scheme [33]. In the predictor step, a low-storage multi-stage Runge–Kutta method is used to advance the momentum equations and compute an intermediate velocity field, which is not divergence-free. To account for the presence of an immersed boundary, the direct forcing IB method (detailed in Section 3.3) is applied. In this approach, the intermediate velocity is first interpolated to the discrete Lagrangian locations. Based on the difference between the interpolated velocity and the prescribed desired velocity, the fluid–solid boundary force is computed. This force is then spread as a volume force to the surrounding Eulerian grid cells, yielding a modified intermediate velocity that (nearly) satisfies the no-slip condition at the fluid–solid interface. The corrector step then solves a Poisson equation for the pressure correction using an incomplete LU factorization [34], projecting the velocity field onto a solenoidal space and ensuring mass conservation. This approach achieves second-order accuracy in both space and time.

The large-eddy simulation (LES) technique is employed where the large-scale turbulent structures are directly resolved and the effects of the unresolved smaller scales are modeled using a subgrid-scale (SGS) model. Specifically, the standard Smagorinsky model [35] is applied with a commonly used model constant $C_S = 0.1$ and van-Driest damping in the vicinity of solid walls. With sufficient near-wall grid resolution (as demonstrated in Section 4.1), wall-resolved LES is performed and no wall functions are required. The flow solver was validated based on a variety of test cases, see, e.g., [36–42]. The ingredients, which are required for the incorporation of the IB method, are described next.

### 3.2. Lagrangian marker tracking

Lagrangian markers are tracked in the computational space (c-space), which is obtained by mapping the structured grid connecting the cell centers (i.e., the **z**-grid shown in Fig. 3(a)) to an orthonormal grid [43]. This approach circumvents the need for the time-intensive search in the physical space (p-space) that would otherwise be required to locate the new computational cell containing each Lagrangian marker. In c-space, the position of each Lagrangian marker, denoted as $\xi_l$ (see Fig. 3(b)), is expressed as the sum of the cell's index triple and a fractional offset:

$$\boldsymbol{\xi}_l = \left[\xi_l, \eta_l, \zeta_l\right] = \left[i_l, j_l, k_l\right] + \left[\Delta\xi_l^+, \Delta\eta_l^+, \Delta\zeta_l^+\right]. \tag{1}$$

Tracking the Lagrangian markers in this manner also serves in efficiently identifying the nearest cell center to each Lagrangian marker (defined here as $\mathbf{x}_{\hat{i},\hat{j},\hat{k}}$), necessary for setting the bounds of the regularized delta function used in the IB method. The index triples $\left[i_l, j_l, k_l\right]$ and $\left[\hat{i}_l, \hat{j}_l, \hat{k}_l\right]$ can be directly obtained by applying "int" and "nint" operators (available in programming languages) to the computational coordinates $\left[\xi_l, \eta_l, \zeta_l\right]$. The fractional offset can then be evaluated using Eq. (1), and its complement then reads:

$$\Delta\xi_l^- = 1 - \Delta\xi_l^+, \qquad \Delta\eta_l^- = 1 - \Delta\eta_l^+, \qquad \Delta\zeta_l^- = 1 - \Delta\zeta_l^+. \tag{2}$$

The fractional offset and its complement can be employed to write the transformation $T$ of the marker's position $\boldsymbol{\xi}_l$ to the p-space by means of a trilinear interpolation using the cell centered values $\mathbf{x}_{i,j,k}$ [43]:

$$\begin{aligned}\mathbf{X}_l = T(\boldsymbol{\xi}_l) = {} & \mathbf{x}_{i,j,k} \left[\Delta\xi_l^- \Delta\eta_l^- \Delta\zeta_l^-\right] + \mathbf{x}_{i+1,j,k} \left[\Delta\xi_l^+ \Delta\eta_l^- \Delta\zeta_l^-\right] + \\ & \mathbf{x}_{i,j+1,k} \left[\Delta\xi_l^- \Delta\eta_l^+ \Delta\zeta_l^-\right] + \mathbf{x}_{i,j,k+1} \left[\Delta\xi_l^- \Delta\eta_l^- \Delta\zeta_l^+\right] + \\ & \mathbf{x}_{i+1,j+1,k} \left[\Delta\xi_l^+ \Delta\eta_l^+ \Delta\zeta_l^-\right] + \mathbf{x}_{i+1,j,k+1} \left[\Delta\xi_l^+ \Delta\eta_l^- \Delta\zeta_l^+\right] + \\ & \mathbf{x}_{i,j+1,k+1} \left[\Delta\xi_l^- \Delta\eta_l^+ \Delta\zeta_l^+\right] + \mathbf{x}_{i+1,j+1,k+1} \left[\Delta\xi_l^+ \Delta\eta_l^+ \Delta\zeta_l^+\right].\end{aligned} \tag{3}$$

This interpolation is used to determine the physical coordinates of the Lagrangian markers, which are required by the immersed boundary solver (as discussed in Section 3.3) and to facilitate visualization of the immersed boundary during the post-processing stage.

The nozzle's upper contour numerically modeled as an immersed boundary is set into motion by prescribing the desired velocity $\mathbf{U}_l^d(t) = \left[U_l^d, V_l^d, W_l^d\right]$ in the physical space for each Lagrangian marker. To update each marker's position within the c-space, the prescribed velocity components must first be transformed into the c-space as follows:

$$\frac{d\xi_l}{dt} = \left(\frac{\partial\xi}{\partial x}\right)_l U_l^d + \left(\frac{\partial\xi}{\partial y}\right)_l V_l^d + \left(\frac{\partial\xi}{\partial z}\right)_l W_l^d, \tag{4}$$

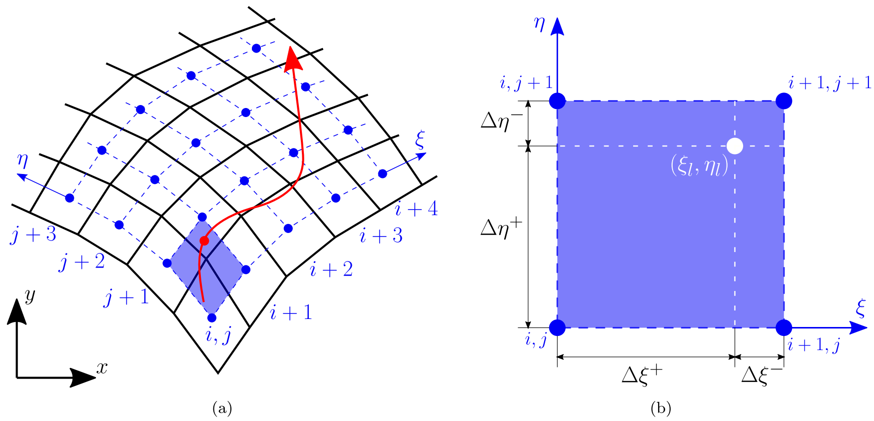



**Fig. 3.** (a) Two-dimensional p-space schematic sketch showing a computational grid (black solid lines), cell centers (blue dots), **z**-grid (blue dashed lines), and a Lagrangian marker's trajectory (red solid line). (b) Corresponding c-space representation showing the coefficients of the trilinear interpolation.

$$\frac{d\eta_l}{dt} = \left(\frac{\partial \eta}{\partial x}\right)_l U_l^d + \left(\frac{\partial \eta}{\partial y}\right)_l V_l^d + \left(\frac{\partial \eta}{\partial z}\right)_l W_l^d, \tag{5}$$

$$\frac{d\zeta_l}{dt} = \left(\frac{\partial \zeta}{\partial x}\right)_l U_l^d + \left(\frac{\partial \zeta}{\partial y}\right)_l V_l^d + \left(\frac{\partial \zeta}{\partial z}\right)_l W_l^d. \tag{6}$$

Hence, the location of a Lagrangian marker is updated by carrying out the temporal integration:

$$\xi_l^{n+1} = \xi_l^n + \left(\frac{\partial \xi}{\partial x}\right)_l \int_{t^n}^{t^{n+1}} U_l^d\, dt + \left(\frac{\partial \xi}{\partial y}\right)_l \int_{t^n}^{t^{n+1}} V_l^d\, dt + \left(\frac{\partial \xi}{\partial z}\right)_l \int_{t^n}^{t^{n+1}} W_l^d\, dt, \tag{7}$$

$$\eta_l^{n+1} = \eta_l^n + \left(\frac{\partial \eta}{\partial x}\right)_l \int_{t^n}^{t^{n+1}} U_l^d\, dt + \left(\frac{\partial \eta}{\partial y}\right)_l \int_{t^n}^{t^{n+1}} V_l^d\, dt + \left(\frac{\partial \eta}{\partial z}\right)_l \int_{t^n}^{t^{n+1}} W_l^d\, dt, \tag{8}$$

$$\zeta_l^{n+1} = \zeta_l^n + \left(\frac{\partial \zeta}{\partial x}\right)_l \int_{t^n}^{t^{n+1}} U_l^d\, dt + \left(\frac{\partial \zeta}{\partial y}\right)_l \int_{t^n}^{t^{n+1}} V_l^d\, dt + \left(\frac{\partial \zeta}{\partial z}\right)_l \int_{t^n}^{t^{n+1}} W_l^d\, dt. \tag{9}$$

For a fixed grid as used in the present case, the spatial metric coefficients appearing in Eqs. (7) to (9) are time independent. They represent the entries of the Jacobian of the p-space to the c-space coordinate transformation $\mathbf{J}$ and are evaluated at the Lagrangian location. Employing the $CZ+$ tracking scheme proposed by Schäfer and Breuer [43], the Jacobian of the inverse coordinate transformation $\mathbf{J}^{-1}$ (i.e., c-space to p-space) is analytically evaluated by differentiating Eq. (3). The first column of $\mathbf{J}^{-1}$ at a position $\xi_l$ reads:

$$\begin{aligned}\mathbf{J}_1^{-1}(\xi_l) = \frac{\partial \mathbf{X}_l}{\partial \xi_1} = \frac{\partial T(\xi_l)}{\partial \xi_1} = {} & \Delta\eta_l^- \Delta\zeta_l^- (\mathbf{x}_{i+1,j,k} - \mathbf{x}_{i,j,k}) + \\ & \Delta\eta_l^+ \Delta\zeta_l^- (\mathbf{x}_{i+1,j+1,k} - \mathbf{x}_{i,j+1,k}) + \\ & \Delta\eta_l^- \Delta\zeta_l^+ (\mathbf{x}_{i+1,j,k+1} - \mathbf{x}_{i,j,k+1}) + \\ & \Delta\eta_l^+ \Delta\zeta_l^+ (\mathbf{x}_{i+1,j+1,k+1} - \mathbf{x}_{i,j+1,k+1}).\end{aligned} \tag{10}$$

Following the evaluation of $\mathbf{J}_m^{-1}(\xi_l)$ where $m = 1, 2, 3$ and inverting it, the spatial metrics in Eqs. (7) to (9) are obtained.

### 3.3. Immersed boundary method with direct forcing approach

The nozzle's upper contour is modeled as a rigid boundary moving in an incompressible viscous fluid by the immersed boundary (IB) method with the direct forcing approach proposed by Uhlmann [17]. At the continuous level, the impact of the immersed boundary ($\Gamma$) on the surrounding fluid is incorporated into the governing equations by adding a forcing term $\mathbf{f}$ to their right-hand side. The Navier–Stokes equations for an incompressible fluid on the domain $\Omega$ then read:

$$\left.\begin{aligned} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \nabla \cdot (\mathbf{u}\mathbf{u}) &= \mu \Delta \mathbf{u} - \nabla p + \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned}\right\} \qquad \mathbf{x} \in \Omega, \tag{11}$$

where $\mathbf{f}$ is a volume force arising from the presence of the immersed boundary. At the discrete level, variables are differentiated by their locations, with lower-case letters indicating Eulerian (fluid grid) quantities and upper-case letters referring to Lagrangian (solid boundary) quantities. In the collocated grid arrangement employed by the finite-volume solver described in Section 3.1, Eulerian locations correspond to the cell centers $\mathbf{x}_e$ (i.e., $\mathbf{x}_e \in \Omega\ \forall\ 1 \le e \le N_E$, where $N_E$ denotes the total number of cell centers). The Lagrangian locations $\mathbf{X}_l$ represent the distributed Lagrangian force points defining the immersed boundary $\Gamma$ (i.e., $\mathbf{X}_l \in \Gamma\ \forall\ 1 \le l \le N_L$, where $N_L$ denotes the total number of Lagrangian markers).

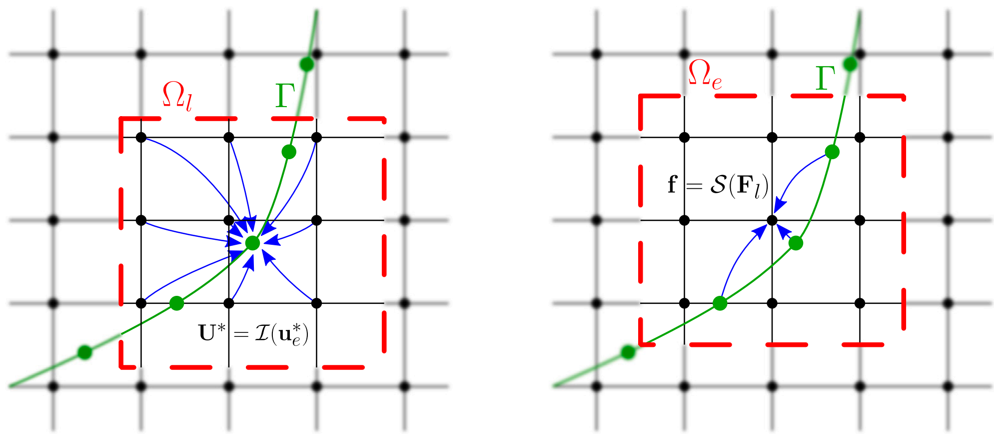
The methodology of the direct forcing method involves evaluating the force at the fluid–solid interface (i.e., within the Lagrangian frame of reference). The force is formulated in such a manner that it enforces (to a certain extent) the no-slip boundary condition at each Lagrangian marker. In the context of the predictor–corrector method, the Lagrangian volume force at a time step $t^n$ reads [17]:

$$\mathbf{F}_l^n = \rho\, \frac{\mathbf{U}_l^d - \mathbf{U}_l^*}{\Delta t} \quad . \tag{12}$$

Here, $\mathbf{U}_l^d$ and $\mathbf{U}_l^*$ are the desired and the intermediate velocity after the predictor step at a Lagrangian location $\mathbf{X}_l$, respectively. The desired velocity $\mathbf{U}_l^d$ is explicitly prescribed by the rigid-body motion of the nozzle's upper contour defined in Section 4.3. In the context of time-dependent immersed boundaries, the Lagrangian markers are typically not coinciding with the Eulerian cell centers. This scenario is illustrated in Fig. 4, which shows a representative case of the spatial relationship between the Lagrangian markers and the Eulerian grid. Since the fluid solver predicts the intermediate velocity only at the Eulerian locations $\mathbf{x}_e$, $\mathbf{U}_l^*$ has to be evaluated by interpolating the Eulerian flow variables to the corresponding Lagrangian locations $\mathbf{X}_l$. This process of transferring information between the two discrete frames of reference is referred to as the interpolation operation. On a uniform one-dimensional Eulerian grid of grid spacing $h$, this operation is performed by employing a regularized Dirac delta function $\delta_h$ [15]:

$$\delta_d(x_e - X_l) \ = \frac{1}{d}\, \varphi(r), \tag{13}$$

where $d$ is the dilation parameter ($d = h$ for uniform grids) and $\varphi$ is a continuous kernel function of symmetric compact support. The dilation parameter defines the extension of the kernel's support $\Omega_l$, and the parameter $r$ is the normalized distance between an Eulerian and a Lagrangian coordinate ($r = (x - X)/d$). On a three-dimensional uniform Cartesian grid $g_h$ of grid size $h$, a three-dimensional regularized delta function is formed as the tensor product of three one-dimensional functions (one for each coordinate direction) and employed to perform




(a) (b)

**Fig. 4.** (a) Interpolation of Eulerian velocities to a Lagrangian location. (b) Spreading of Lagrangian forces to an Eulerian location.

the interpolation operation $\mathcal{I}(\mathbf{u}_e^*)$:

$$\mathbf{U}_l^* = \sum_{e\in N_{E,l}} \mathbf{u}_e^*\, \delta_h\left(\mathbf{x}_e - \mathbf{X}_l\right)\, \Delta v_e, \tag{14}$$

where $N_{E,l}$ denotes the set of Eulerian cells whose centroids $\mathbf{x}_e$ lie within the support of the kernel associated with the $l$th Lagrangian marker, and $\Delta v_e$ is the volume of the $e$th Eulerian cell. An illustration of the interpolation operation, which evaluates a discrete Lagrangian quantity as a weighted average of neighboring discrete Eulerian quantities, is depicted in Fig. 4(a).

Following the interpolation step, the Lagrangian boundary force $\mathbf{F}_l$ is directly computed at each Lagrangian force point using Eq. (12). The presence of the immersed boundary is then accounted for in the momentum equations governing the fluid by spreading the Lagrangian volume forces onto the Eulerian grid and incorporating the spread force as a volumetric source term on the right-hand side. The spreading operation $\mathcal{S}(\mathbf{F}_l)$, illustrated in Fig. 4(b), utilizes the same regularized delta function as the interpolation operation to evaluate the Eulerian volume force:

$$\mathbf{f}_e = \sum_{l\in N_{L,e}} \mathbf{F}_l\, \delta_h\left(\mathbf{x}_e - \mathbf{X}_l\right)\, \Delta V_l\,, \tag{15}$$

where $N_{L,e}$ denotes the set of Lagrangian markers whose kernel support encompasses the $e$th Eulerian cell, and $\Delta V_l$ is the volume associated with the $l$th Lagrangian marker. The Lagrangian volume can be expressed as $\Delta V_l = \Delta S_l\, \epsilon_l$, where $\Delta S_l$ represents the surface area associated with the $l$th marker and $\epsilon_l$ denotes its characteristic thickness. In uniform Eulerian grids, this volume is assumed to be equal to the volume of the fluid cell, provided that the Lagrangian markers are uniformly distributed such that $\Delta S_l \approx \Delta v_e^{2/3}$ (i.e., $\epsilon_l \approx h$ [17]). An alternative approach for determining the characteristic thickness is to set it such that when the force is spread from the Lagrangian marker and interpolated back to it, the exact force is recovered [44]. This method necessitates the solution of a linear system of equations that enforces the duality of the interpolation and the spreading operators. Zhou and Balachandar [45], however, have shown that the Lagrangian volume serves as a relaxation factor governing how fast the no-slip condition is satisfied. The authors have also found that the optimal choice of the Lagrangian volume corresponds to the largest value permitted by a stability condition. Eventually, the choice of the Lagrangian volume is somewhat arbitrary, as it primarily influences the magnitude of the feedback forcing [46]. In this work, the real thickness of the nozzle's upper contour is specifically used to define the third dimension for the Lagrangian grid as discussed in Section 4.2.

The steps of the predictor–corrector method, modified to account for the immersed boundary, can now be summarized by the following equations:

1. $\mathbf{u}_e^* = \mathbf{u}_e^n + \dfrac{\Delta t}{\rho}\mathbf{rhs}^n \qquad \forall\ \mathbf{x}_e \in g_h, \qquad (16)$
2. $\mathbf{U}_l^* = \mathcal{I}(\mathbf{u}_e^*) \qquad \forall\ \mathbf{X}_l \in \Gamma, \qquad (17)$
3. $\mathbf{F}_l = \rho\, \dfrac{\mathbf{U}_l^d - \mathbf{U}_l^*}{\Delta t} \qquad \forall\ \mathbf{X}_l \in \Gamma, \qquad (18)$
4. $\mathbf{f}_e = \mathcal{S}(\mathbf{F}_l) \qquad \forall\ \mathbf{x}_e \in g_h, \qquad (19)$
5. $\mathbf{u}_e^{**} = \mathbf{u}_e^* + \dfrac{\Delta t}{\rho}\,\mathbf{f}_e \qquad \forall\ \mathbf{x}_e \in g_h, \qquad (20)$
6. $\Delta p_e' = \dfrac{\rho}{\Delta t}\nabla\cdot\mathbf{u}_e^{**} \qquad \forall\ \mathbf{x}_e \in g_h, \qquad (21)$
7. $\mathbf{u}_e^{n+1} = \mathbf{u}_e^n - \dfrac{\Delta t}{\rho}\nabla p_e' \qquad \forall\ \mathbf{x}_e \in g_h. \qquad (22)$

Compared to the standard predictor–corrector scheme, steps 2 to 5 have been added and lead to a second intermediate velocity $\mathbf{u}_e^{**}$, that satisfies (with a decreasing error at each time step) the no-slip boundary condition at the fluid–solid interface. This velocity serves as the foundation for constructing the right-hand side of the Poisson Eq. (21) used to compute the pressure correction, $p_e'$ (step 6), and subsequently determine the updated velocity, $\mathbf{u}_e^{n+1}$ (step 7).

Several regularized delta function kernels, effectively smearing the boundary forces over Eulerian grid cells within their compact support, have been derived in the literature. The selection of a particular kernel often depends on the required level of accuracy, the smoothing properties, and the desired numerical efficiency. Nevertheless, all smoothing kernels are designed to satisfy, at a minimum, the zeroth- and the first-order discrete moment conditions on uniform grids:

$$\sum_{e\in N_{E,l}} \delta_h\left(\mathbf{x}_e - \mathbf{X}_l\right)\, \Delta v_e = 1 \qquad \forall\, l, \tag{23}$$

$$\sum_{e\in N_{E,l}} \left(\mathbf{x}_e - \mathbf{X}_l\right)\, \delta_h\left(\mathbf{x}_e - \mathbf{X}_l\right)\, \Delta v_e = 0 \qquad \forall\, l. \tag{24}$$

These discrete analogues of the kernel's fundamental properties ensure the integral conservation of the force and torque in the spreading operation:

$$\begin{aligned}\sum_{e\in N_E} \mathbf{f}_e\, \Delta v_e &= \sum_{e\in N_E}\sum_{l\in N_{L,e}} \mathbf{F}_l\, \delta_h\left(\mathbf{x}_e - \mathbf{X}_l\right)\, \Delta V_l\, \Delta v_e \\ &= \sum_{l\in N_L} \mathbf{F}_l \left(\cancelto{1}{\sum_{e\in N_{E,l}} \delta_h\left(\mathbf{x}_e - \mathbf{X}_l\right)\, \Delta v_e}\right) \Delta V_l = \sum_{l\in N_L} \mathbf{F}_l\, \Delta V_l,\end{aligned} \tag{25}$$

$$\sum_{e\in N_E} \mathbf{x}_e \times \mathbf{f}_e\, \Delta v_e = \sum_{e\in N_E} \mathbf{x}_e \times \left(\sum_{l\in N_{L,e}} \mathbf{F}_l\, \delta_h\left(\mathbf{x}_e - \mathbf{X}_l\right)\, \Delta V_l\right) \Delta v_e$$

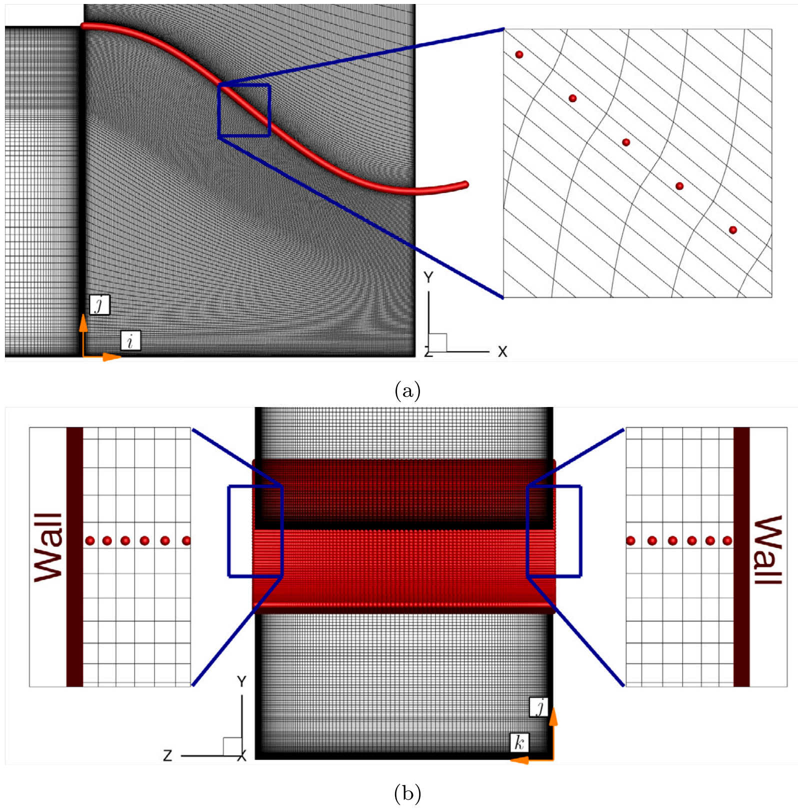



**Fig. 5.** Schematic showing the distribution of Lagrangian markers (red spheres) (a) in the $z$-plane of the curvilinear grid. (b) Near-wall boundaries in an $i$-plane of the grid within the nozzle region.

$$= \sum_{l\in N_L} \left( \cancelto{\mathbf{X}_l}{\sum_{e\in N_{E,l}} \mathbf{x}_e\, \delta_h \left(\mathbf{x}_e - \mathbf{X}_l\right)\, \Delta v_e} \right) \times \mathbf{F}_l\, \Delta V_l = \sum_{l\in N_L} \mathbf{X}_l \times \mathbf{F}_l\, \Delta V_l. \tag{26}$$

Unlike most IBM applications that rely on a Cartesian Eulerian grid, this work employs a curvilinear stretched grid to discretize the nozzle region (as shown in Fig. 5(a) and detailed in Section 4.1). This approach clusters grid points near the walls, enabling an accurate resolution of the boundary layer. Furthermore, employing a curvilinear grid avoids extending the refinement to flow regions where it is not necessary, leading to significant computational savings.

Since the nozzle's upper contour moves between two side walls (see Fig. 5(b)), Lagrangian points must inevitably be positioned near the boundaries of the computational domain to ensure a perfect seal at the corners. This proximity limits the applicability of kernels with symmetric support for these specific Lagrangian markers, necessitating the use of non-symmetric kernel's support that accounts for the boundaries of the computational domain.

Remedies to the two aforementioned deviations from the classical IBM (i.e., markers on curvilinear stretched grids and markers with one-sided kernels) have been previously proposed in the literature. This work incorporates the moving least squares version of the direct forcing immersed boundary method (MLS-IBM) introduced by Vanella and Balaras [22], beside its recent one-sided extension proposed by Bale et al. [27]. A key advantage of the MLS method is that it does not impose a specific arrangement of Eulerian interpolation points around a Lagrangian marker (e.g., uniform distribution). This characteristic makes the MLS approach effective for handling immersed boundaries on curvilinear grids.

The work of Bale et al. [27] extends the MLS technique to construct one-sided (non-symmetric) IB kernels from standard two-sided (symmetric) ones. This advancement enables velocity interpolation and force spreading exclusively on one side of the fluid–solid interface. This specific ability is particularly beneficial for avoiding spurious flows within enclosed areas and for facilitating the handling of immersed structures near the boundaries of the computational domain.

The following provides a summary of the MLS method, along with a new proposition for its application to curvilinear block-structured grids. A detailed explanation of how the general framework is applied to different Lagrangian markers is also presented.

### 3.3.1. Standard MLS formulation

The basic idea of the moving least squares version of the direct forcing immersed boundary method is to compute interpolation weights that transfer a quantity to a Lagrangian marker from arbitrarily distributed Eulerian cell centers in its vicinity by solving a weighted least squares optimization problem. The MLS method approximates the value of the Eulerian variable $g$ at the discrete location of the $l$th Lagrangian marker by:

$$g_l = \sum_{i=1}^{m} p_i(\mathbf{X}_l)\, c_i^g(\mathbf{X}_l) = \mathbf{p}(\mathbf{X}_l)^\intercal\, \mathbf{c}^g(\mathbf{X}_l). \tag{27}$$

Here, $\mathbf{p}(\mathbf{X}_l) \in \mathbb{R}^m$ is the basis function vector evaluated at the Lagrangian location $\mathbf{X}_l$ and $\mathbf{c}^g(\mathbf{X}_l) \in \mathbb{R}^m$ is the vector of unknown coefficients depending on the Eulerian quantity $g_e$. For a second-order accurate three-dimensional spatial discretization scheme, it is sufficient to define $\mathbf{p}$ as a linear basis function vector (i.e., $m = 4$ and $\mathbf{p}(\mathbf{x}) = [1 \quad x \quad y \quad z]$). The unknown coefficient vector $\mathbf{c}^g(\mathbf{X}_l)$ is found by minimizing the weighted $L^2$-norm of the error function:

$$\min_{\mathbf{c}^g(\mathbf{X}_l)} \quad J_l = \frac{1}{2} \sum_{e\in N_{E,l}} \phi_{e,l} \left( \mathbf{p}(\mathbf{x}_e)^\intercal\, \mathbf{c}^g(\mathbf{X}_l) - g_e \right)^2, \tag{28}$$

where $\phi$ denotes the MLS weighting function and $\phi_{e,l} = \phi(\mathbf{x}_e - \mathbf{X}_l)$ represents the discrete interpolation weight for the $e$th Eulerian cell. This discrete weight is equivalent to the product of the kernel function normalized by the dilation coefficient and the volume of the fluid cell

$\Delta v_e$ (i.e., $\delta_h(\mathbf{x}_e - \mathbf{X}_l)\,\Delta v_e$ appearing in Eq. (14)). Eq. (28) can also be written in a matrix form as:

$$\min_{\mathbf{c}^g(\mathbf{X}_l)} \quad J_l = \frac{1}{2}\left(\mathcal{A}^\intercal \mathbf{c}^g(\mathbf{X}_l) - \mathbf{g}\right)^\intercal \operatorname{diag}(\boldsymbol{\phi}_l)\left(\mathcal{A}^\intercal \mathbf{c}^g(\mathbf{X}_l) - \mathbf{g}\right), \tag{29}$$

where the entries of the polynomial matrix $\mathcal{A} \in \mathbb{R}^{m \times N_{E,l}}$ are the values of the basis functions at the Eulerian interpolation stencil (i.e., $\mathcal{A}_{ij} = p_i(\mathbf{x}_j)$, with $i = 1, \ldots, m$, $j = 1, \ldots, N_{E,l}$). The minimization of the error $J_l$ with respect to the vector of unknown coefficients (i.e., $\frac{\partial J_l}{\partial \mathbf{c}^g(\mathbf{X}_l)} = 0$) yields (see Appendix A.1 for details):

$$\mathbf{c}^g(\mathbf{X}_l) = \mathcal{G}^{-1}\,\mathcal{B}\,\mathbf{g}, \tag{30}$$

where $\mathcal{G} = \mathcal{A}\operatorname{diag}(\boldsymbol{\phi}_l)\mathcal{A}^\intercal$ is the positive-definite Gram matrix and $\mathcal{B} = \mathcal{A}\operatorname{diag}(\boldsymbol{\phi}_l)$. Substituting the solution of $\mathbf{c}^g(\mathbf{X}_l)$ in Eq. (27) yields:

$$g_l = \mathbf{p}(\mathbf{X}_l)^\intercal\,\mathcal{G}^{-1}\,\mathcal{B}\,\mathbf{g} = \tilde{\boldsymbol{\phi}}_l^\intercal\,\mathbf{g} = \sum_{e \in N_{E,l}} \tilde{\phi}_{e,l}\,g_e, \tag{31}$$

where $\tilde{\boldsymbol{\phi}}_l = \mathcal{B}^\intercal\,\mathcal{G}^{-1}\,\mathbf{p}(\mathbf{X}_l) \in \mathbb{R}^{N_{E,l}}$ is the MLS kernel function vector (also known as the *generating function*). Unlike the expression given in Eq. (27), where the coefficient vector (i.e., $\mathbf{c}^g(\mathbf{X}_l)$) is dependent on both the Lagrangian marker's position and the Eulerian variable being interpolated, the form in Eq. (31) depends only on the coordinates of the Lagrangian marker. In the context of the immersed boundary method, it is more practical to directly use the latter form as this enables the use of a single kernel function for the interpolation of velocity components and the spreading of the forces.

It is worth mentioning that, although not explicitly imposed, the generating functions evaluated by the MLS method satisfy the discrete moment conditions (also known as the polynomial reproduction constraints):

$$\sum_{e \in N_{E,l}} p_i(\mathbf{x}_e)\,\tilde{\phi}_{e,l} = p_i(\mathbf{X}_l), \qquad \forall i = 1, \ldots, m, \tag{32}$$

with the matrix form written as:

$$\mathcal{A}\,\tilde{\boldsymbol{\phi}}_l = \mathbf{p}(\mathbf{X}_l). \tag{33}$$

A proof can be found in Appendix A.2.

*3.3.2. New proposition for curvilinear block-structured grids*

In diffuse-interface IB methods, the dilation parameter (appearing in Eq. (13)) defines the interpolation window $\Omega_l$ and, consequently, the number of Eulerian cells included in the stencil of the Lagrangian marker ($N_{E,l}$). In addition, this parameter plays a role in computing the initial interpolation weights ($\phi_{e,l}$) required for the MLS procedure.

In most classical IBM applications, the structure of interest is immersed in a Cartesian uniform grid. In such cases, the dilation parameter for each coordinate direction is straightforward, and it is equal to the uniform grid spacing along that direction. In non-uniform grids, however, local dilation parameters are estimated in the literature for each Lagrangian marker using a criterion that considers the nearest Eulerian cell ($\mathbf{x}_{\hat{e}}$) and its immediate neighbors (see, e.g., [23,44,47]). Then, based on the evaluated parameters, the set of Eulerian cell centers within the interpolation window is sought. However, some Eulerian centroids may lie at the boundary of the interpolation window, resulting in near-zero weight values, which can render the moment matrix ($\mathcal{G}$ in Eq. (30)) singular. In addition, this approach introduces an extra computational overhead due to the need for a larger search radius around the Lagrangian marker. That would also result in systems of varying sizes for different Lagrangian markers, requiring dynamic memory allocation at each time step (for moving immersed boundaries) for the vectors and matrices needed for the MLS method. The issue becomes more pronounced when solving the problem in parallel, particularly when the interpolation window of a Lagrangian marker, located on a connectivity patch between two parallel blocks of a block-structured grid, extends to cover multiple ghost cells. In second-order accurate finite-volume codes, however, only one ghost cell is typically employed.

The proposition made here makes the determination of the interpolation weights independent of the dilation parameter. Consequently, there is no need to estimate the dilation parameter, and the risk of obtaining zero interpolation weights is eliminated. The interpolation window is also fixed to encompass the exact number of Eulerian cells that the kernel $\varphi$ is intended to cover. In other words, a $n$-point kernel incorporates $n$ Eulerian cell centers. This is achieved by evaluating the interpolation weights with respect to the computational space (explained in Section 3.2) instead of the physical space as:

$$\phi^c_{e,l} = \frac{1}{\Delta v_{\hat{e}}}\,\varphi(\xi_e - \xi_l)\,\varphi(\eta_e - \eta_l)\,\varphi(\zeta_e - \zeta_l)\,\Delta v_e\,, \tag{34}$$

where the superscript $c$ indicates that the value is computed with respect to the computational space and $\Delta v_{\hat{e}}$ is the volume of the closest Eulerian cell. Note that the normalized parameter $r$ has been replaced by the difference between the Eulerian and the Lagrangian computational coordinates, which are already normalized. For a uniform grid, $\phi^c_{e,l}$ will immediately satisfy the moment conditions in Eqs. (23) and (24) provided that the full support of the kernel $\varphi$, which satisfies these conditions, is used. For a non-uniform distribution of Eulerian cells around a Lagrangian marker, the kernel's weights are evaluated without accounting for the non-uniformity of the grid. However, they are scaled by the ratio $\frac{\Delta v_e}{\Delta v_{\hat{e}}}$ for each cell center. This scaling causes $\phi^c_{e,l}$ to deviate from the moment conditions as $\frac{\Delta v_e}{\Delta v_{\hat{e}}} \neq 1$ for $e \neq \hat{e}$.

A thorough validation study of the proposed immersed boundary approach on a block-structured curvilinear grid is performed in Appendix B based on the flow past a stationary and an oscillating cylinder.

*3.3.3. General framework*

The MLS procedure detailed in Section 3.3.1 is used in conjunction with the approach proposed in Section 3.3.2. If a Lagrangian marker is far from the boundaries of the computational domain, the 3-point kernel proposed by Roma et al. [18] is applied in all local directions leading in total to 27 points. Contrarily, when a Lagrangian marker is near the domain boundary, the 5-point kernel by Bao et al. [20] is employed only along the constrained directions. Fig. 6 depicts the selection of kernels based on the marker's proximity to the boundaries of the computational domain. This selection is motivated by the findings of Bale et al. [27], which demonstrate that using a wider kernel helps to prevent the Gram matrix in the one-sided MLS from becoming ill-conditioned due to insufficient support.

The general procedure for each Lagrangian marker can be summarized as follows:

1. The nearest Eulerian cell center in the computational space and its direct neighbors in each local block direction (i.e., $i$, $j$ and $k$) are identified.
2. A check is carried out whether any of the Eulerian neighbors belong to the boundaries of the computational domain and the affected local directions are constrained accordingly to exclude those neighbors (e.g., wall). If a local direction is constrained, it is extended from its other side to include one extra Eulerian interpolation point.
3. Based on the absolute difference between the Lagrangian computational coordinates and its Eulerian counterparts along each local block direction, a kernel function is assigned. For instance, if $\max\left(|\xi_e - \xi_l|\right) \le 1.5\;\forall\, e \in N_{E,l}$, the 3-point kernel is employed. This will be the case (in all local block directions) for all Lagrangian markers far from the boundaries of the computational domain. However, if $\max\left(|\xi_e - \xi_l|\right) > 1.5$, this indicates that the interpolation window was extended in step 2 and the 5-point kernel is assigned to the $i$th direction.

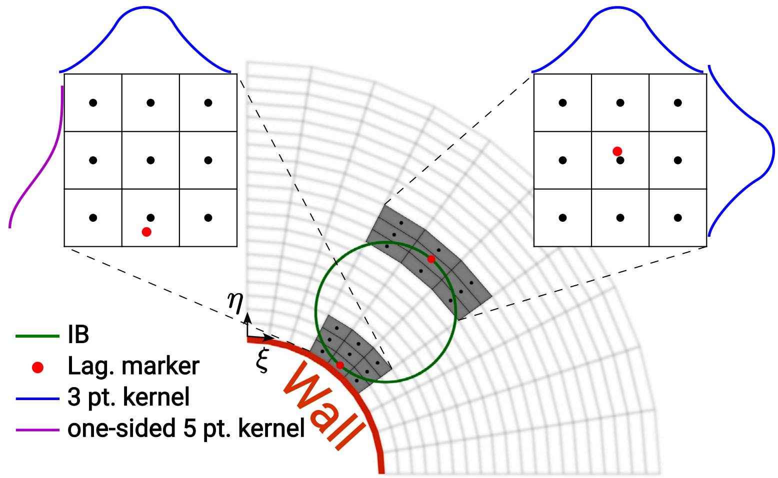



**Fig. 6.** Schematic sketch showing two Lagrangian markers within a curvilinear block-structured grid. The sketch highlights the interpolation kernels employed along each local coordinate direction for markers near and far from the boundary of the computational domain.

4. The MLS weights are evaluated based on Eq. (34). These weights do not satisfy the discrete moment conditions due to the curvilinear Eulerian grid, the reduced (one-sided) kernel support, or both.
5. The MLS kernel function vector is found using Eq. (A.15). These modified weights satisfy the polynomial reproduction constraints needed for the conservation of force and torque.

## 4. Description of the setup

### 4.1. Computational domain, boundary conditions and numerical grid

The purpose of the current setup is to develop a numerical model of the experimental wind gust generator described by Wood and Breuer [13,28], i.e., the adaptive nozzle. This approach enables a more detailed analysis of the complex flow generated by the nozzle's moving upper contour, as all flow field variables, including velocities and pressure, are computed across the entire computational domain. Fig. 7 illustrates the numerical setup designed to replicate its experimental counterpart shown in Fig. 1. A uniform horizontal velocity field of $u_{inlet}^{steady} = 2.53$ m/s is defined at the inlet patch, i.e., the left cross-section of the square duct. Within the duct and the nozzle, the boundary layers develop along all walls, reaching a free-stream velocity of $u_\infty = 5.14$ m/s at the nozzle's outlet, thus matching the conditions of the experimental setup.

Note that the inflow turbulence intensity found in the experiment (see Section 2) has been neglected in the present predictions due to the following two reasons. First, the reported value of 1.5 % refers only to the bulk flow in the wind tunnel. However, the turbulence intensity is expected to vary across the nozzle outlet cross-section, particularly near the walls. Second, the knowledge about the turbulence intensity alone is not sufficient to artificially generate inflow turbulence. At the very least, additional information regarding the characteristic length scales of the approaching inflow turbulence is required. Since none of these quantities is currently available, it is more reasonable to neglect the low-level inflow turbulence entirely rather than to represent it based on potentially incorrect assumptions.

No-slip and impermeability boundary conditions are imposed on the walls of the duct, the nozzle, the far-field walls and the ground plate. A convective boundary condition is applied at the outlet patch with a convective velocity of $u_{conv} = u_\infty$.

As depicted in Fig. 7, the real geometry in front of the nozzle is simplified, retaining only the square duct directly ahead of the nozzle with a newly defined length of $l_{duct} = 1$ m and a cross-sectional side length of $w = 0.5$ m. This duct replaces the flow straightener, the diffusor, and the square duct of the wind tunnel. The nozzle's geometry with a contraction ratio $A_{inlet}/A_{outlet}$ of 2:1 is exactly replicated to ensure that identical boundary conditions as in the experiment are applied. The computer-aided design (CAD) sketch used to define the nozzle's upper contour, which is fabricated and used in the experiments of Wood and Breuer [13,28] is depicted in Fig. 8. A precise definition of this curve is crucial for generating the curvilinear Eulerian grid and for providing an accurate distribution of Lagrangian markers along the curve.

Based on the coordinates and boundary conditions depicted in Fig. 8, the geometry of the upper contour is defined by the following piecewise cubic spline:

$$y_{\text{noz}} = \begin{cases} -1.662\,x^3 - 1.884\,x^2 + 0.5, & 0 \le x < 0.12 \\ 6.855\,(x-0.12)^3 - 2.482\,(x-0.12)^2 - 0.524\,(x-0.12) + 0.47, & 0.12 \le x < 0.25 \\ 4.581\,(x-0.25)^3 + 0.191\,(x-0.25)^2 - 0.821\,(x-0.25) + 0.375, & 0.25 \le x < 0.43 \\ -4.461\,(x-0.43)^3 + 2.665\,(x-0.43)^2 - 0.307\,(x-0.43) + 0.26, & 0.43 \le x < 0.5 \\ -21.045\,(x-0.5)^3 + 2.926\,(x-0.5)^2 + 0.25, & 0.5 \le x < 0.525 \\ 2.63\,(x-0.525)^3 + 1.347\,(x-0.525)^2 + 0.106\,(x-0.525) + 0.251, & 0.525 \le x < 0.575. \end{cases} \tag{35}$$

Note that for the sake of brevity the unit [m] is left out here. The spline in the range 0 m $\le$x$\le$ 0.5 m is used to generate a structured curvilinear grid around the nozzle. However, the entire spline including the tail of the nozzle compromising the range 0.5 m $\le x \le$ 0.575 m is employed to define the coordinates of the Lagrangian markers. The Eulerian grid depicted as a zoomed excerpt in Fig. 9 is a block-structured curvilinear grid composed of $105.6 \times 10^6$ control volumes. Note that this number of control volumes and their distribution in the computational domain are the outcome of an optimization process carried out prior to the final simulations. This preliminary study is not described here for the sake of brevity.

The resulting geometric structure, made up of 12 blocks, is subdivided into 786 parallel blocks. This setup enables the parallel execution across 786 cores, achieving a load balancing efficiency according to the pure CFD prediction without Lagrangian markers of 100 %.

To accurately capture the significant viscous effects that give rise to thin boundary layers, grid refinement near rigid walls including duct walls, nozzle walls and the ground plate is necessary. This refinement ensures that the first cell centers are positioned within the viscous sublayer ($y^+ < 5$). The hydraulic diameter at the nozzle's outlet, defined as $D_h = 4\,w\,h/(2(w+h)) = 0.333$ m (with $h = 0.25$ m), is used as the characteristic length for determining the Reynolds number Re

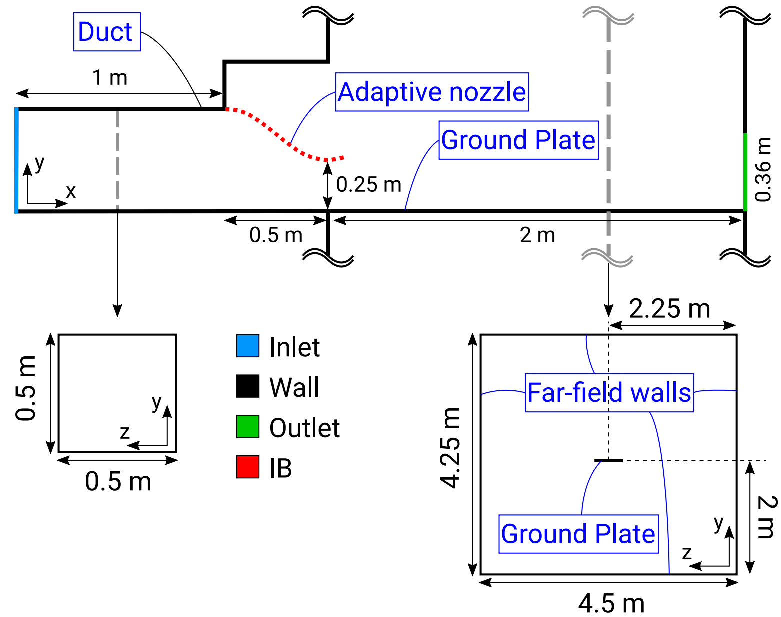



**Fig. 7.** Geometry of the test case and definition of the boundary conditions.

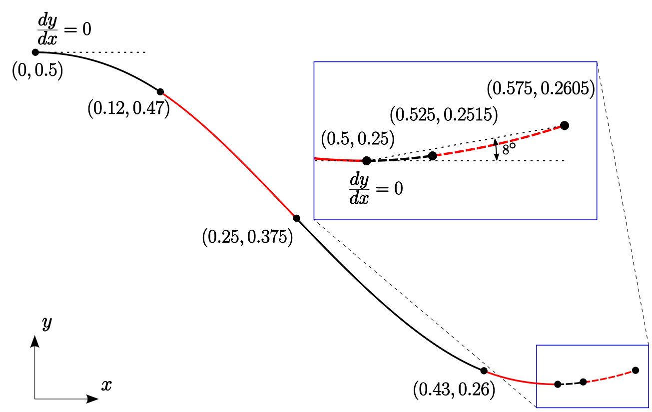



**Fig. 8.** Nozzle's upper contour spline. All coordinates are given in meter.

$= \rho\, D_h\, u_\infty/\mu = 114{,}878$. Based on this Reynolds number, the Darcy friction factor as reported by Moody [48] is:

$$f = 0.0055\left[1 + \left(2\times 10^{-4}\,\frac{k}{D_h} + \frac{10^6}{\mathrm{Re}}\right)^{\frac{1}{3}}\right], \tag{36}$$

where $k$ is the pipe's effective roughness height. Assuming a smooth pipe (i.e., $k = 0$) the skin friction coefficient is evaluated by $c_f = f/4 = 4.2 \times 10^{-3}$.

To accurately resolve the near-wall region, the maximum height of the first grid cell off the wall, denoted as $\Delta y$, can be estimated such that a dimensionless wall distance at the cell center of $y^+_{cc} = 2$ is achieved. This results in the relationship $\Delta y/D_h = (4/\mathrm{Re}) \cdot \sqrt{2/c_f} \approx 7.6 \times 10^{-4}$, yielding $\Delta y \approx 2.5 \times 10^{-4}$ m. To optimize computational efficiency, the grid is mildly stretched in the off-wall directions, with a maximum expansion factor of 1.04. A maximum grid spacing of $10^{-3}$ m is applied in the $x$-$y$ plane within the region that will be passed by the nozzle's contour during its movement. This refinement ensures an accurate fluid-IB interaction and a precise representation of the moving shear layer induced by the nozzle's upper contour. Note that Van-Driest damping of the eddy viscosity near the immersed boundary is presently not considered. This omission stems from the fact that implementing a wall-distance-based damping function within a diffuse-interface immersed boundary framework is non-trivial and particularly challenging on curvilinear grids. Alternatively, subgrid-scale models that do not rely on such damping functions, such as the WALE model [49] or the dynamic SGS model [50], can be used in future studies.

On the ground plate, where the numerical results are to be compared with experimental measurements, the grid is uniformly spaced in the streamwise direction with a spacing of $5 \times 10^{-3}$ m. This grid definition is applied to avoid excessive coarseness due to stretching, ensuring that the generated gust is properly resolved.

Outside the comparison region, the grid is stretched with a stretching factor of 1.04 in the downstream direction extending to the far right boundary of the computational domain. In the spanwise direction, the same grid stretching factor is applied without any imposed maximum grid spacing constraints. To maintain the stability of the explicit time integration scheme and to appropriately resolve the turbulent flow in time, a time step size $\Delta t = 2 \times 10^{-5}$ s is chosen, ensuring a CFL number of less than 0.4.

### *4.2. Distribution of Lagrangian markers describing the adaptive nozzle*

The immersed boundary given by the nozzle's upper contour is defined by a set of Lagrangian markers distributed along the continuous spline defined by Eq. (35). These markers are positioned such that their spacing in each local coordinate direction is set by the minimum local Eulerian grid spacing (i.e., $\Delta x_{i,min}$ for $i = 1, 2, 3$) encountered by the immersed boundary during the nozzle's closing and opening motion.

Within the nozzle's block (i.e., $0\ \mathrm{m} \le x \le 0.5$ m) the distribution of Lagrangian markers is defined such that one marker is assigned to each cell center. At the contour's free end (i.e., the tail of the nozzle in the range $0.5\ \mathrm{m} \le x \le 0.575$ m), a uniform distribution of Lagrangian markers with a spacing equal to the minimum Eulerian grid spacing is employed. This distribution is motivated by the fact that, during the downstroke, these Lagrangian points overlap with the region where the finest Eulerian grid spacing is defined. In this way, any potential fluid penetration through the immersed boundary surface caused by insufficient coverage of Eulerian cells by the Lagrangian markers window function is prevented. In the spanwise direction (i.e., along the $z$-axis), the distribution is defined such that one marker is assigned to each cell center. Consequently, the markers are set to follow the grid stretching applied along the $z$-axis. This decision is driven by the fact that, during

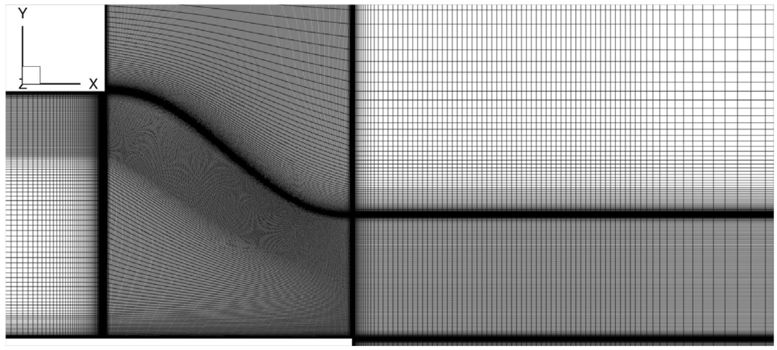



**Fig. 9.** Zoom of the computational grid around the nozzle's contour in the $x$-$y$ plane. Only every second grid line in each direction is shown.

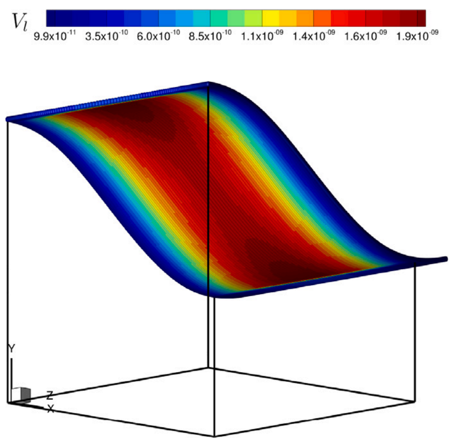



**Fig. 10.** Volumes associated with the non-uniformly distributed Lagrangian markers.

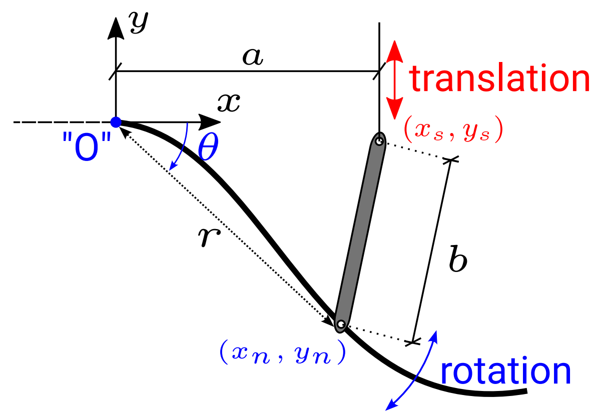



**Fig. 11.** Schematic sketch of the nozzle's upper contour connected to the saddle via an eye-bearing rod.

the nozzle's closing and opening motion, the distribution of Eulerian points along the $z$-axis remains fixed. A similar approach was used by Boulbrachene and Breuer [11], resulting in a significant reduction in the number of Lagrangian markers compared to a uniform distribution without compromising the accuracy of the results. The fabricated metal sheet used as the nozzle's upper contour in the wind tunnel has a thickness of $2 \times 10^{-3}$ m. This is accounted for numerically by assigning the same thickness to the immersed boundary. A uniform Lagrangian grid spacing equal to the minimum Eulerian cell spacing of $2.5 \times 10^{-4}$ m is set in the direction normal to the IB surface. As a result, 8 layers of Lagrangian markers are defined along this direction.

Based on the description above, the number of Lagrangian markers in each IB local direction is $N_1 = 892$, $N_2 = 160$ and $N_3 = 8$, i.e., $N_L = N_1\, N_2\, N_3 = 1,141,760$. Such a non-uniform distribution of Lagrangian markers within a non-uniform curvilinear Eulerian grid is efficient, as it places markers more densely only where necessary. Fewer markers help to reduce the additional computational load associated with the interpolation, spreading and tracking processes. It is also worth mentioning that due to the non-uniform distribution of Lagrangian markers, the Lagrangian volumes are also spatially non-uniform as depicted in Fig. 10.

### 4.3. Definition of the rotational velocity of the adaptive nozzle

While the saddle of the artificial wind gust generator follows a prescribed vertical translational motion, the adaptive nozzle undergoes a rotational motion as depicted in Fig. 11. This conversion from translational to rotational motion is achieved through an eye-bearing rod of length $b = 0.15$ m. Since the velocity of each Lagrangian marker is required to compute the Lagrangian volume force in Eq. (12), it is essential to deduce the corresponding rotational velocity of the adaptive nozzle. Based on the pure rotational motion around the origin of the coordinate system (point "O" in Fig. 11), the velocity components of the Lagrangian markers can be determined.

A detailed description of the transformation is provided in Appendix C. The following two geometric conditions hold throughout the nozzle's motion:

$$b^2 = (x_s - x_n)^2 + (y_s - y_n)^2, \tag{37}$$

$$r^2 = x_n^2 + y_n^2, \quad \text{where } r = 0.336 \text{ m}. \tag{38}$$

By substituting $x_s = a = 0.295$ m, $x_n = r\cos\theta$ and $y_n = r\sin\theta$ in Eq. (37), one can find the relation between the coordinate $y_s$ and the angle of rotation $\theta$:

$$y_s = r\sin\theta + \sqrt{b^2 - (a - r\cos\theta)^2} \tag{39}$$

Differentiating Eq. (39) with respect to time and solving for $\omega_z$ yields:

$$\omega_z = \frac{v_s}{x_n - \frac{(a-x_n)\, y_n}{\sqrt{b^2-(a-x_n)^2}}}. \tag{40}$$

The unknown coordinates of the point on the nozzle (i.e., $x_n(t)$ and $y_n(t)$) are determined by expanding Eq. (37) and replacing $x_n^2 + y_n^2$ by $r^2$ according to Eq. (38), leading to the following system of equations to be solved:

$$x_n a + y_n y_s = k, \quad \text{where:} \quad k = \frac{r^2 + a^2 + y_s^2 - b^2}{2}$$
$$x_n^2 + y_n^2 = r^2,$$

whose solutions are provided in Appendix C.

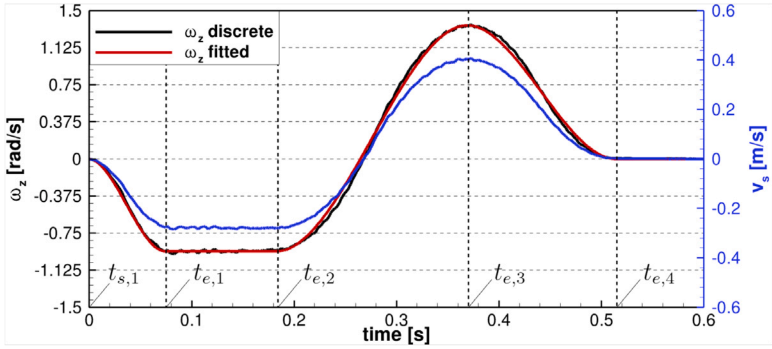



**Fig. 12.** Kinematic characteristics of the nozzle's rotational velocity and the saddle's vertical velocity.

The time-dependent vertical displacement $y_s(t)$ defined in the experiment by the saddle's vertical translational motion is used to evaluate the coordinates of the point on the nozzle $(x_n, y_n)$. These coordinates, along with the vertical velocity curve of the saddle (i.e., $v_s(t)$), are used to determine a discrete $\omega_z(t)$ curve as depicted in Fig. 12.

A smooth (differentiable) analytical function is fitted to the discrete one as shown in Fig. 12. The fitted rotational velocity function found reads:

$$\omega_{z,\mathrm{fit}}(t) = \begin{cases} -A_1\left(3\phi_1^2 - 2\phi_1^3\right), & \text{if } t_{s,1} = 0\,\mathrm{s} \le t < t_{e,1} = 0.075\,\mathrm{s} \\ -A_1, & \text{if } t_{s,2} = t_{e,1} \le t < t_{e,2} = 0.184\,\mathrm{s} \\ A_2\left(3\phi_3^2 - 2\phi_3^3\right) - A_1, & \text{if } t_{s,3} = t_{e,2} \le t < t_{e,3} = 0.37\,\mathrm{s} \\ (A_2 - A_1) - (A_2 - A_1)\left(3\phi_4^2 - 2\phi_4^3\right), & \text{if } t_{s,4} = t_{e,3} \le t < t_{e,4} = 0.515\,\mathrm{s} \\ 0, & \text{otherwise,} \end{cases} \tag{41}$$

where $\phi_i = \left(t - t_{mid,i}\right)/t_{w,i}$, $t_{mid,i} = \left(t_{e,i} + t_{s,i}\right)/2$, $t_{w,i} = t_{e,i} - t_{s,i}$, $A_1 = 0.934$, and $A_2 = 2.284$.

The translational velocities of each Lagrangian force point are then evaluated by:

$$\mathbf{U}_l^d(\mathbf{X}_l) = \boldsymbol{\omega}_{\mathrm{fit}} \times \left(\mathbf{X}_l - \mathbf{x}_c\right), \tag{42}$$

where $\mathbf{x}_c$ is the center of rotation of the immersed boundary and $\boldsymbol{\omega}_{\mathrm{fit}} = \left[0,\, 0,\, \omega_{z,\mathrm{fit}}\right]$. As visible in Fig. 12, the entire gust generation time is presently set to $\Delta t_{\mathrm{gust}} = 0.5$ s. This characteristic time scale is also used in Section 5 to normalize the time axis.

As shown in Wood and Breuer [13], the presently applied motion pattern of the saddle and the resulting rotation of the upper contour are set to resemble the so-called extreme coherent gust (ECG) with a (1 - cos)-type signal as defined in the IEC-Standard [51].

### 4.4. Definition of the inlet velocity variation

In Wood and Breuer [13], an undershooting of the streamwise velocity below the free-stream velocity was observed in the wake of the generated gust. A similar behavior was observed in the first-generation gust generator [9], though it was much more pronounced and suffered from a significantly longer restitution phase. In both gust generators, the authors attributed this phenomenon to pressure losses caused by the sudden partial closure of the nozzle's outlet. This effect combined with the fact that the blower is not operated in a mode that compensates for these increased pressure losses, results in less air being blown through the wind tunnel leading to an undershoot below the free-stream velocity. This was further confirmed in the experiments by placing a Prandtl probe inside the main tunnel element (see Fig. 1) during the gust generation and calculating the dynamic pressure as the difference between the measured total and static pressures. Based on this dynamic pressure, the reduction of the inlet velocity was determined as shown in Fig. 13(a). However, the amplitudes of the velocity drop applied in the simulation are not directly based on the measured velocity drop, as the relevant parts of the geometry (i.e., the main tunnel element, the diffuser, and the settling chamber) are not included in the simulation. Instead, they are determined numerically by analyzing the difference between the experimental and simulated streamwise velocity signal at the first monitoring point $\Delta u_{mon,1} = u_{mon,1}^{num} - u_{mon,1}^{exp}$ for the case that the preliminary simulation is carried out with a constant inlet velocity at the inlet patch (i.e., $u_{inlet} = u_{inlet}^{steady} = 2.53$ m/s). The continuity equation applied to the velocity difference is used to determine the transient inlet velocity:

$$\begin{aligned} \rho\, \Delta u_{inlet}(t)\, A_{inlet} &= \rho\, \Delta u_{mon,1}\, A_{noz,out}(t)\,, \\ \Delta u_{inlet}(t) &= \Delta u_{mon,1}\, \frac{A_{noz,out}(t)}{A_{inlet}} = \Delta u_{mon,1}\, \frac{A_{outlet}\,(1 - BR)}{A_{inlet}}\,, \\ u_{inlet}^{trans}(t) &= u_{inlet}^{steady} - \Delta u_{mon,1}\, \frac{(1 - BR)}{2}\,. \end{aligned} \tag{43}$$

Since the motion of the nozzle's upper contour is prescribed, the blocking ratio can be determined as a function of time, given by $BR(t) = \left(A_{outlet} - A_{noz,out}(t)\right)/A_{outlet}$. The inlet velocity function found is depicted in Fig. 13(b) and reads:

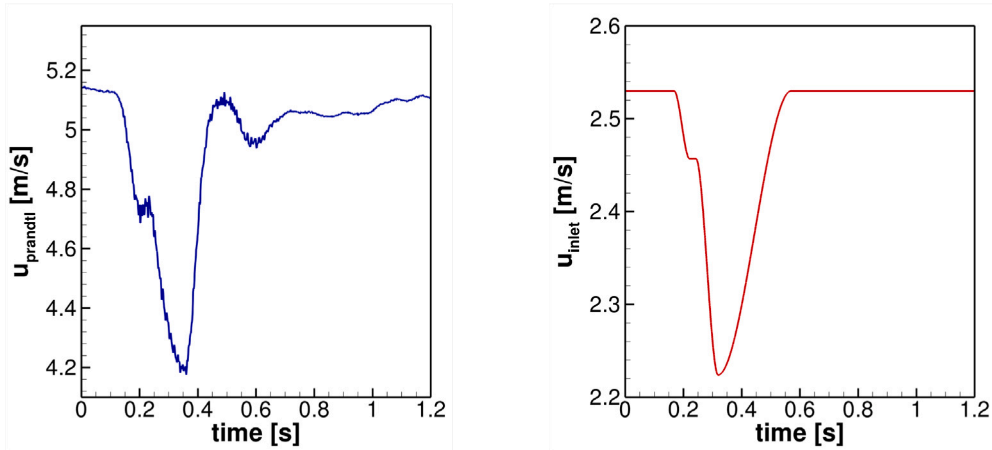
$$u_{inlet}^{trans}(t) = \begin{cases} u_{inlet}^{steady}, & \text{if } 0\,\mathrm{s} \le t < t_{s,1}^{in} = 0.168\,\mathrm{s} \\ u_{inlet}^{steady} - A_3\left(3\phi_1^2 - 2\phi_1^3\right), & \text{if } t_{s,1}^{in} \le t < t_{e,1}^{in} = 0.221\,\mathrm{s} \\ u_{inlet}^{steady} - A_3, & \text{if } t_{s,2}^{in} = t_{e,1}^{in} \le t < t_{e,2}^{in} = 0.243\,\mathrm{s} \\ u_{inlet}^{steady} - A_3 - A_4\left(3\phi_3^2 - 2\phi_3^3\right), & \text{if } t_{s,3}^{in} = t_{e,2}^{in} \le t < t_{e,3}^{in} = 0.319\,\mathrm{s} \\ (u_{inlet}^{steady} - A_3 - A_4) + (A_3 + A_4)\left(3\phi_4^2 - 2\phi_4^3\right), & \text{if } t_{s,4}^{in} = t_{e,3}^{in} \le t < t_{e,4}^{in} = 0.57\,\mathrm{s} \\ u_{inlet}^{steady} & \text{otherwise,} \end{cases} \tag{44}$$

where $\phi_i$ is defined in a similar manner as in Eq. (41) with $A_3 = 0.073$, and $A_4 = 0.233$. Interestingly, the trend of the numerically obtained transient inlet velocity function closely matches that measured by the Prandtl probe. However, a direct comparison of the amplitudes between the prescribed inlet velocity function and the measured data is not meaningful, as the latter is obtained within the main tunnel element.

## 5. Results

### 5.1. Instantaneous flow field induced by the wind gust generator

The initial analysis focuses on demonstrating the time-dependent flow characteristics of the gust induced by the rotational motion of the nozzle's upper contour. Fig. 14 annotates five distinct time instants, at which the streamwise velocity component and the pressure fields in the $x$-$y$ symmetry plane are illustrated in Fig. 15. The monitoring points, labeled 1 to 11, are those defined in Fig. 2 and denote specific positions within the wind tunnel's test section selected for comparison with




(a) (b)

**Fig. 13.** Time-dependent inlet boundary condition: (a) Measured inlet velocity based on Prandtl probe [13] and (b) the prescribed transient inlet velocity applied in the simulation.

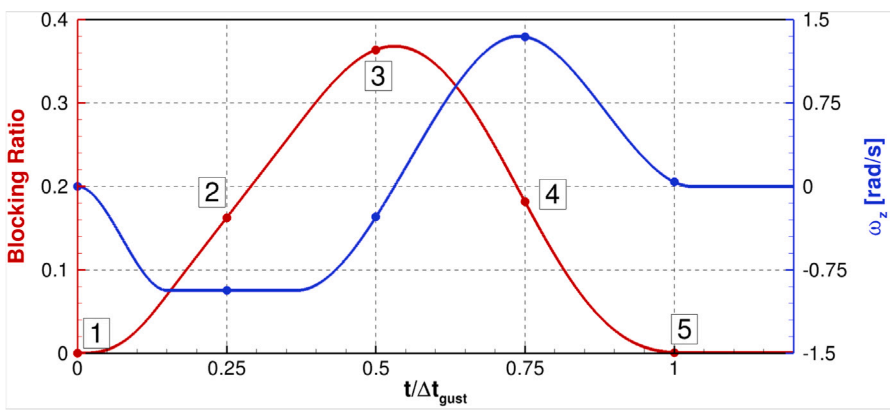



**Fig. 14.** Nozzle's blocking ratio annotated with five characteristic time instants at which the flow is visualized in Fig. 15; rotational velocity of the upper contour during the downward and upward strokes.

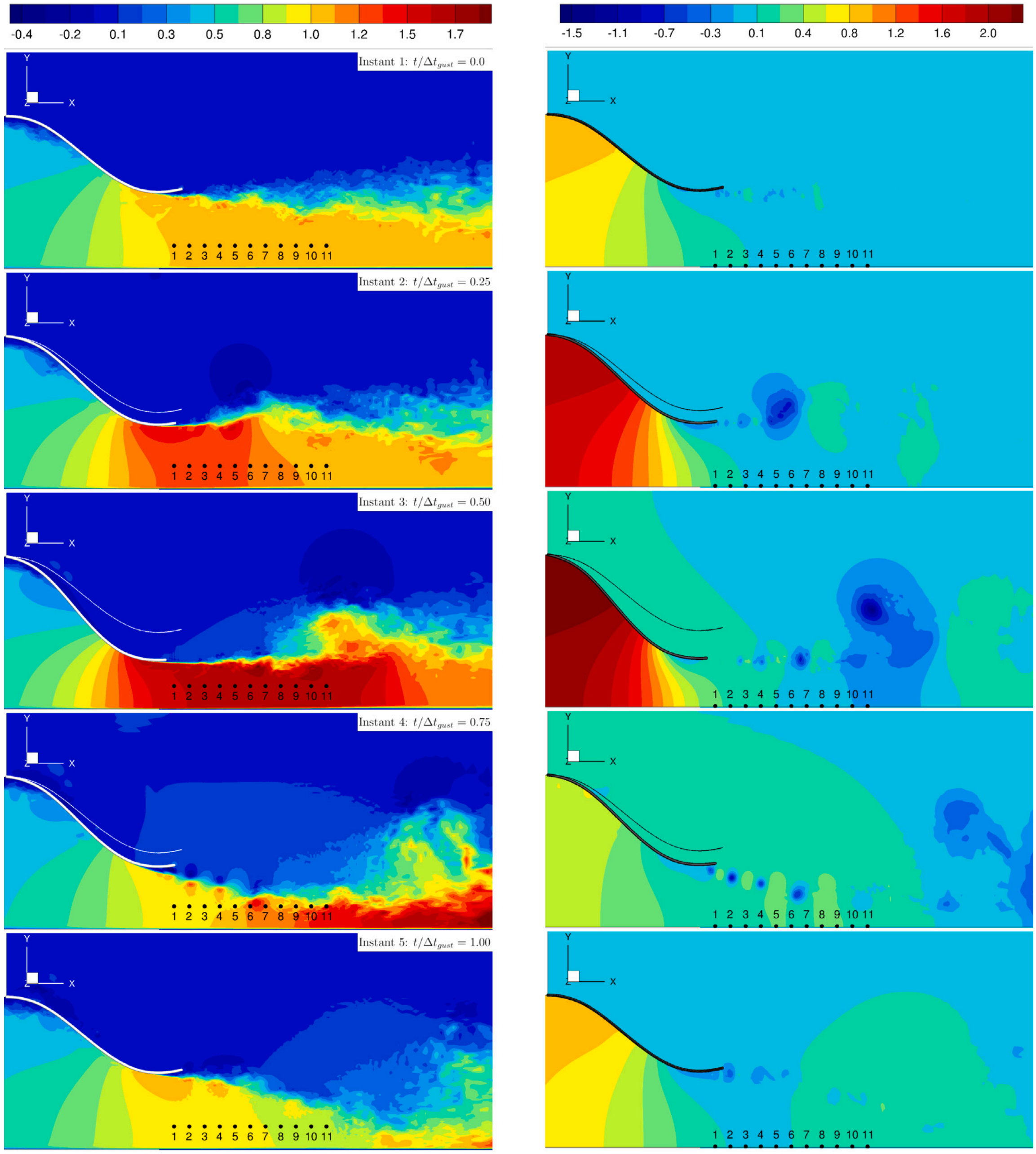
the experimental data by Wood and Breuer [13,28]. These positions correspond to the typical area, in which a wind tunnel model is placed.

At instant 1 (i.e., at $t/\Delta t_{\text{gust}} = 0$), the nozzle's upper contour is in its idle position, and the flow fields define the initial conditions for the numerical gust generator. Prior to this instant, the flow is allowed to fully develop while the immersed boundary remains stationary, ensuring a stable reference condition before any dynamic motion begins. Two distinct flow phenomena can be observed at this stage. First, boundary layers form along the top and bottom walls of the nozzle, with the latter continuing to develop along the ground plate. Second, a free shear layer emerges, separating the high-velocity core flow from the surrounding slower-moving fluid. Obviously, this detached shear layer results from the boundary layer separating from the nozzle's upper contour slightly before its end.

At instant 2 (i.e., at $t/\Delta t_{\text{gust}} = 0.25$), the upper contour of the nozzle moves downward from its idle position, partially obstructing approximately 16 % of the nozzle's original outlet area. This reduction in area accelerates the streamwise velocity above the ground plate. However, unlike the flow dynamics generated by the paddle gust generator, the flow exiting the nozzle maintains its original trajectory pointing in the main flow direction, highlighting a key advantage of the new design. Additionally, the clockwise rotation of the upper contour results in an increase in the static pressure within the nozzle, as the contour pushes against the flow direction, causing a pressure buildup upstream. A small counter-clockwise rotating vortex forms near the tip of the nozzle's upper contour due to the local flow separation caused by the sudden downward movement.

At instant 3 (i.e., at $t/\Delta t_{\text{gust}} = 0.5$), the upper contour's downstroke is nearly complete, reaching a blocking ratio of 36 %. The recorded high streamwise velocity values along the ground plate characterize the strong horizontal wind gust generated. The initially thick free shear layer observed near the tip of the upper contour at instant 1 has evolved into a thin sharp shear layer, adjusting its altitude dynamically in response to the contour's motion. Moreover, higher pressure values are recorded inside the nozzle compared to instant 2, resulting from a stronger negative pressure gradient developing near the nozzle's outlet and further accelerating the flow beneath the nozzle's upper contour. The vortical structure, formed by the interaction between the moving contour's tip and the flow, has grown in size and been convected further downstream.

Instant 4 (i.e., $t/\Delta t_{\text{gust}} = 0.75$) captures a moment during the upward return of the contour toward its idle position. In this phase, the strength of the generated gust decreases as the flow expands within the outlet. Simultaneously, the free shear layer destabilizes, shedding small vortices from the upper contour's tip, resulting in a streamwise velocity field with strong fluctuations. Moreover, the free shear layer tilts and




(a) $u/u_\infty$

(b) $p/\left(\frac{1}{2}\rho u_\infty^2\right)$

**Fig. 15.** Snapshots of the streamwise velocity (a) and the pressure (b) fields during the nozzle's downward and upward strokes. Instants correspond to those depicted in Fig. 14. For the instants 2 to 4, a line indicating the upper contour's idle position is depicted for reference.

no longer remains horizontal, as it fails to adjust quickly enough to the rapid upward movement of the nozzle's contour. The lowest pressure values inside the nozzle, compared to all previous instants, are recorded. This is attributed to the counter-clockwise rotation of the nozzle's contour, which moves in favor of the main flow direction, reducing the flow redirection and consequently lowering the pressure.

At instant 5 (i.e., at $t/\Delta t_{\text{gust}} = 1$), the nozzle's upper contour has returned to its idle position, marking the end of the dynamic motion. The flow gradually recovers, returning to its initial state.

### 5.2. Comparison of numerical and experimental results

The proposed numerical framework is validated through a direct comparison of the numerical predictions with the experimental measurements of Wood and Breuer [13]. In the experiments, phase-averaging over 46 gusts was performed on the measured quantities to capture the characteristic development of the gust. This was crucial for two primary reasons. First, a free shear layer exists at the top surface of the jet exiting the nozzle, where small vortical structures are irregularly generated and convected downstream. This makes the flow generated by the upper contour's motion dependent on the flow state at the onset of movement. Second, phase averaging is required due to the presence of a streamwise turbulence intensity of about 1.5 % measured at the nozzle's outlet of the Eiffel-type wind tunnel. Although no inflow turbulence is considered in the simulation for simplicity, the free shear layer and its subsequent effects are present in the simulations as shown at instant 1 in Fig. 15. Therefore, simulation results are phase-averaged in the same manner as in the experiment to enable a consistent comparison. The phase-averaged numerical results are based on 46 initial condition states separated by an interval of 0.2 s. The choice is motivated by a detailed examination of the flow variables at the shear layer, which revealed that for the chosen time interval the flow states exhibit statistical independence.

#### 5.2.1. Pointwise and field velocity

The comparison is based on the time histories of the velocity magnitude at the aforementioned 11 streamwise positions shown in Fig. 15(a) falling within the range $0 \leq x/l \leq 1.0$. The phase-averaged CTA measurements of the velocity magnitude and their numerical counterpart are depicted in Fig. 16.

The time histories at individual monitoring points generally exhibit a bell-shaped pattern, reflecting the evolution of the nozzle's blocking ratio. In accordance with the principle of mass conservation, the flow beneath the upper contour accelerates as the nozzle's outlet contracts further, reaching its peak velocity at the maximum blocking ratio. This phenomenon can also be clearly seen in Fig. 17, which shows a comparison between the measured and predicted phase-averaged streamwise velocity contours in the symmetry plane at different instants of the gust generation. Both experiment and simulation depict the highest streamwise velocity at $t/\Delta t_{gust} = 0.5$. As the nozzle's contour returns to its idle position, the streamwise velocity decreases accordingly. The subsequent drop below the free-stream velocity (see Fig. 16, Fig. 17(f) and Fig. 17(g)) is at least partially attributed to the prescribed transient inlet velocity defined in Eq. (44), which accounts for the reduced airflow through the wind tunnel's blower. This velocity undershoot becomes more pronounced as the gust propagates downstream.

The velocity magnitude curves for the monitoring points within $0 \leq x/l \leq 0.2$ (i.e., Figs. 16(a) to 16(c)) closely align with those observed in the experimental data. However, for the monitoring points within $0.3 \leq x/l \leq 1$ (i.e., Figs. 16(d) to 16(k)), the descending slope of these curves includes an additional peak, becoming more pronounced as one moves downstream. This fluctuation stems from the flow separation near the tip of the nozzle's upper contour, triggered by its rapid upward motion. This separation generates a series of small vortices in the detached, inclined shear layer, which destabilize and ultimately disrupt the flow at the downstream monitoring points, as illustrated at instant 4 in Fig. 15(a). This series of small vortices can even be found in the phase-averaged numerical data in Figs 17(e) and 17(f). While this additional peak is absent in the phase-averaged experimental data, the instantaneous data of Wood and Breuer [13,28] (not shown here) reveal similar velocity oscillations to those predicted by LES. The persistence of this velocity peak in the phase-averaged numerical results is attributed to the absence of inflow turbulence. When turbulence intensity is accounted for, flow separation occurs at varying times during the nozzle's upstroke. Consequently, vortex shedding becomes desynchronized across different gust realizations, causing the peak to be averaged out. That explains the differences in the shear layer observed between the measurements and the predictions.

A notable feature in the velocity signals is the presence of two distinct local maxima of the velocity. In addition to the maximum velocity observed during the gust generation period at $t/\Delta t_{\mathrm{gust}} \approx 0.5$, another maximum emerges after the gust has formed (i.e., at $t/\Delta t_{\mathrm{gust}} > 1$), particularly at the monitoring points 5 to 9. Obviously, a second, smaller local velocity maximum is formed in the far wake of the generated gust. This overshoot, much like the preceding undershoot, becomes more pronounced further downstream. Overall, this quantitative comparison demonstrates that the simulation methodology suggested effectively captures the gust generation process in the wind tunnel. With the exception of the free shear layer, the predicted velocity distributions in Fig. 17 are in close agreement with the PIV measurements by Wood and Breuer [13].

Note that Fig. D.24 in Appendix D also depicts a comparison between the measured and predicted phase-averaged vertical velocity contours at different instants of the gust generation process. Overall, the agreement is again good, but the deviations in the free shear layer are also present there, as expected.

#### 5.2.2. Surface pressure at monitoring points

For a more detailed analysis of the numerical predictions, a surface pressure based comparison is carried out. The time history of the predicted phase-averaged pressure coefficient $\langle c_p \rangle = 2\,\langle p \rangle/(\rho\, u_\infty^2)$ and its measured counterpart at the 11 streamwise monitoring points shown in Fig. 15(b) is depicted in Fig. 18. It is worth mentioning that the predicted phase-averaged pressure coefficients are uniformly shifted upward by a constant factor of 0.0383. This shift is applied because, for the simulation of incompressible fluids, a pressure reference point must be defined, anchoring the entire pressure field. As the simulation progresses over a long period of time, the flow not exiting the computational domain through the outlet eventually reaches the pressure reference point. Consequently, the pressure at the monitoring point decreases, necessitating the upward shift to align the numerical predictions with the experimental measurements.

For all monitoring points, the pressure coefficient is initially greater than zero, with its value decreasing downstream. This trend is clearly depicted in Fig. 18(l) showing the recorded pressure coefficients at the monitoring points for the case without gust. The higher values of the pressure coefficient at the upstream points are attributed to the elevated static pressure inside the nozzle, which exceeds the atmospheric pressure $p_\infty$ (i.e., here the pressure at the pressure reference point). This negative pressure gradient is also visible at $t/\Delta t_{gust} = 0$ in Fig. 19, showing the phase-averaged pressure coefficient within the domain of interest.

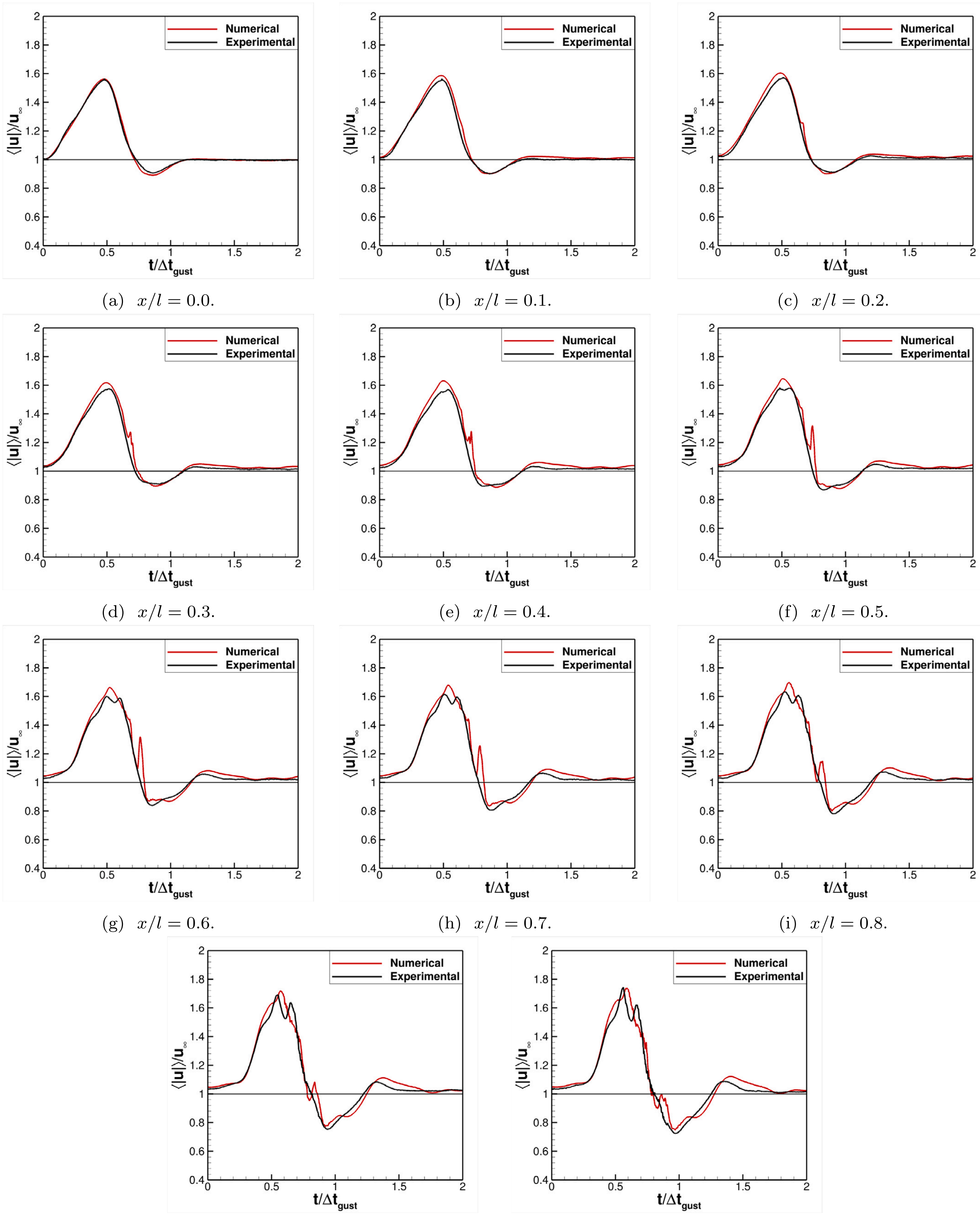
The temporal evolution of $\langle c_p \rangle$ follows a similar pattern across all monitoring points. During the initial phase of the nozzle's downward stroke, the surface pressure increases due to the compression effect caused by the contraction of the nozzle's outlet as shown at $t/\Delta t_{gust} = 0.1$ in Fig. 19. This pressure rise is observed at all monitoring points, though its magnitude decreases with increasing downstream distance.

Following the initial rise, the pressure coefficient undergoes a sharp drop, reaching negative values for $x/l > 0.4$ (see Figs. 18(f) to 18(k)). The contour plots at $0.2 \leq t/\Delta t_{gust} \leq 0.5$ in Fig. 19 clearly indicate that this drop is caused by the large vortical structure forming at the tip of the nozzle's upper contour due to the sudden downward motion. While this vortex influences the upstream points only slightly, its impact becomes more pronounced downstream as it expands and gains intensity.

As the large vortical structure is convected downstream, positive pressure values are recovered due to the continued pressure buildup inside the nozzle as depicted at $0.3 \leq t/\Delta t_{gust} \leq 0.5$ in Fig. 19. Obviously, points near the nozzle's outlet (i.e., $0 \leq x/l \leq 0.1$) experience this pressure rise first, reaching a peak at the end of the downstroke, significantly earlier than those further downstream. For points within $0.2 \leq x/l \leq 1.0$, the majority of the pressure rise leading to a peak occurs during the upstroke phase, aligning with the velocity drop caused by the expansion of the nozzle's outlet (see $0.7 \leq t/\Delta t_{gust} \leq 0.8$ in Fig. 19). Unlike the nearly constant velocity peaks across the monitoring points, the pressure coefficient peaks increase significantly in downstream direction, eventually reaching a nearly constant maximum for $x/l > 0.7$.

Subsequently, a sharp pressure drop occurs during the upward stroke of the nozzle as the velocity begins to rise again, recovering from its undershoot.

Overall, the comparison between the numerical and the experimental surface pressure data demonstrates strong agreement in terms of both trend and magnitude. The observed discrepancies, particularly at the first two points, can be attributed to geometric inaccuracies associated with the fabricated nozzle. This surface pressure analysis reinforces the validity of the proposed numerical framework in capturing the fundamental mechanisms of the gust generation.



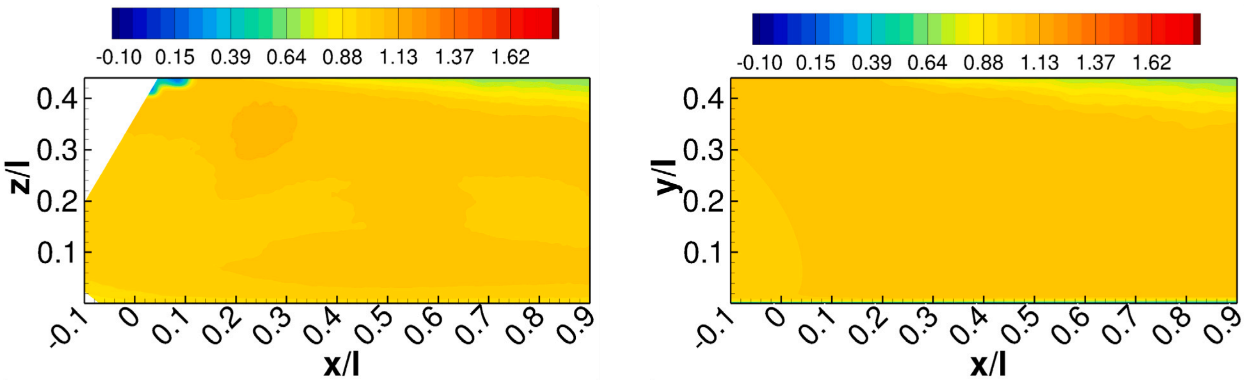

(a) $x/l = 0.0$. (b) $x/l = 0.1$. (c) $x/l = 0.2$.

(d) $x/l = 0.3$. (e) $x/l = 0.4$. (f) $x/l = 0.5$.

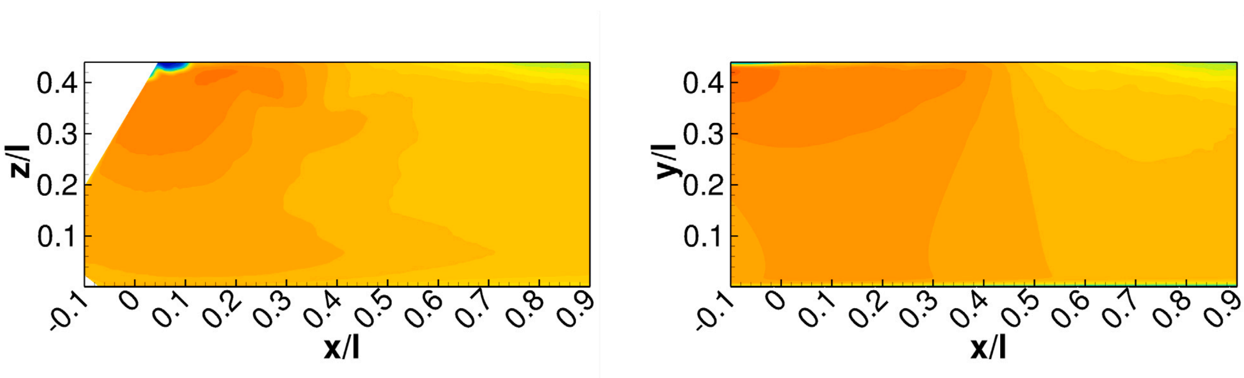
(g) $x/l = 0.6$. (h) $x/l = 0.7$. (i) $x/l = 0.8$.

(j) $x/l = 0.9$. (k) $x/l = 1.0$.

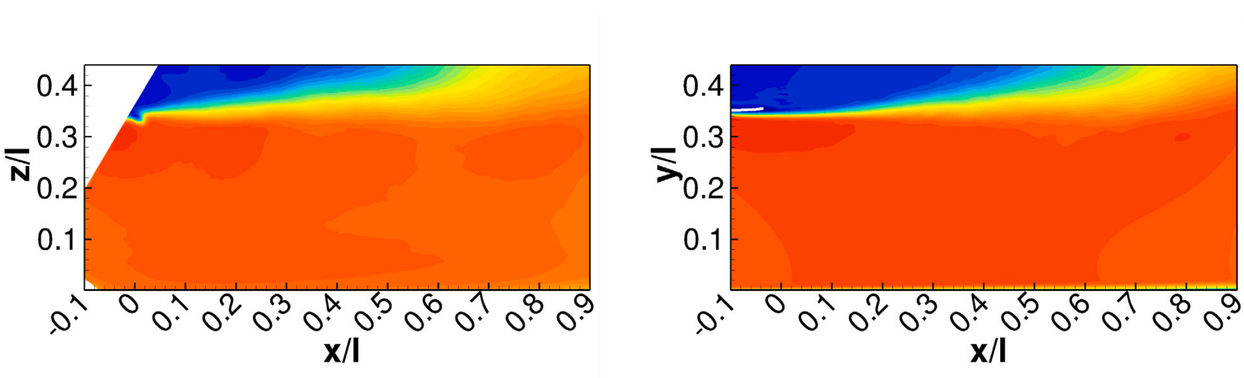
**Fig. 16.** Time histories of the phase-averaged experimental velocity magnitude [13,28] and its numerical counterpart at the 11 monitoring points depicted in Fig. 15(a).



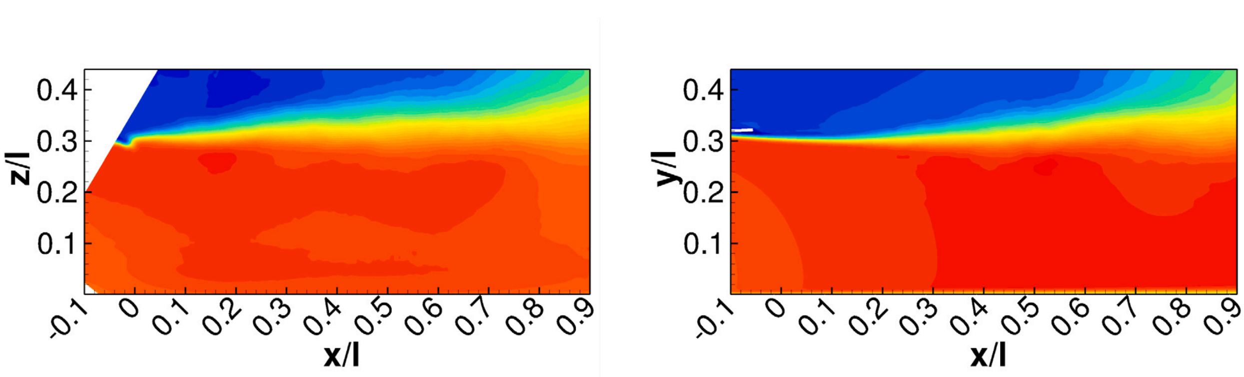

(a) $t/\Delta_{\text{gust}} = 0$: Experiment (left) vs. Simulation (right).




(b) $t/\Delta_{\text{gust}} = 0.2$: Experiment (left) vs. Simulation (right).

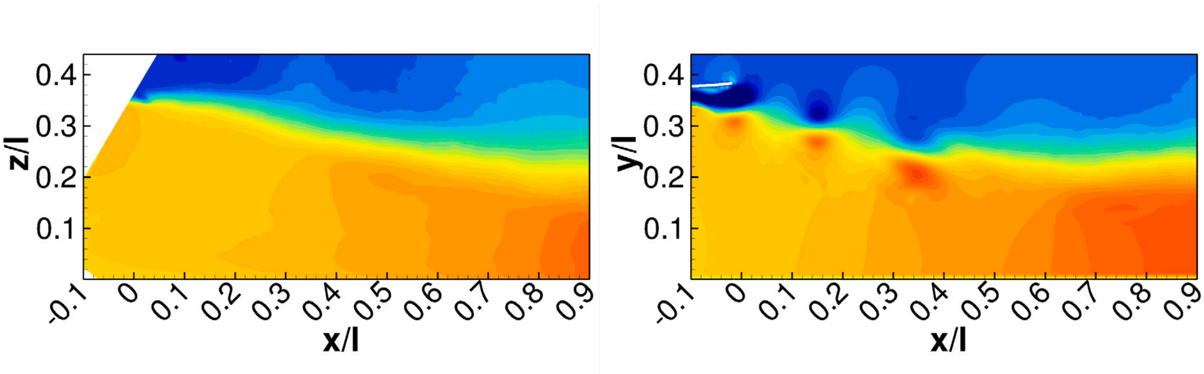



(c) $t/\Delta_{\text{gust}} = 0.4$: Experiment (left) vs. Simulation (right).

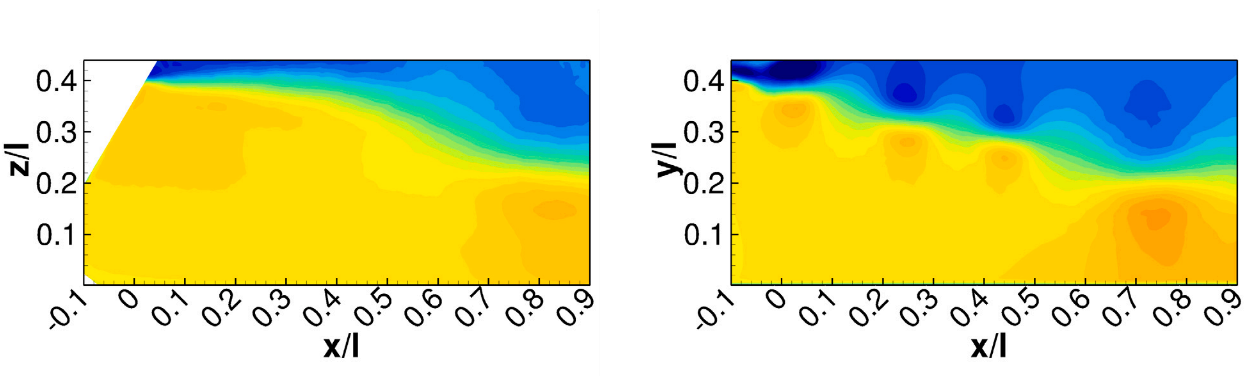



(d) $t/\Delta_{\text{gust}} = 0.5$: Experiment (left) vs. Simulation (right).

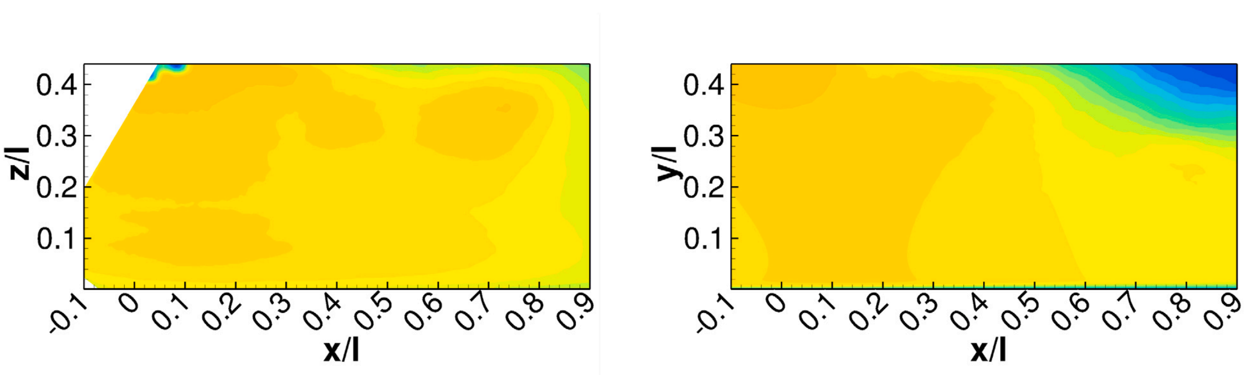
**Fig. 17.** Comparison between measured and predicted data of the phase-averaged streamwise velocity in the symmetry plane at different time instants of the gust generation. Experimental data taken from Wood and Breuer [13].



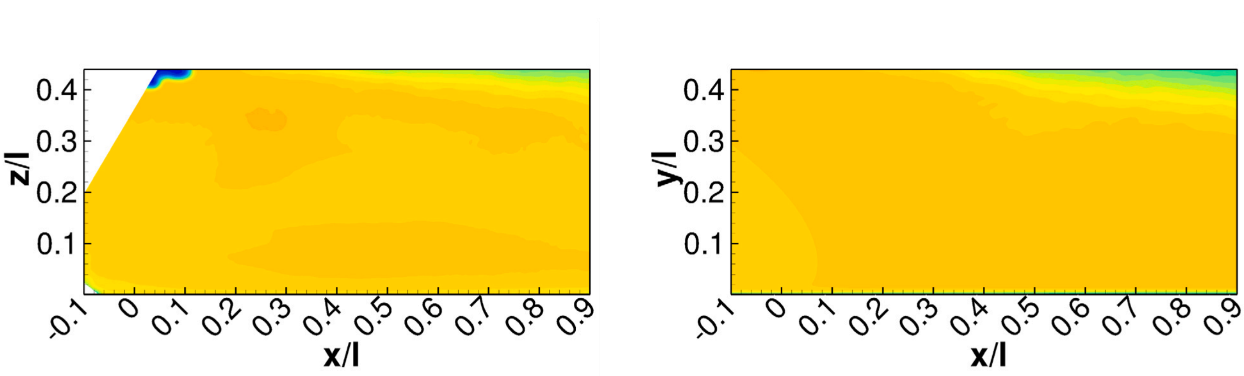
(e) $t/\Delta_{\mathrm{gust}} = 0.7$: Experiment (left) vs. Simulation (right).



(f) $t/\Delta_{\mathrm{gust}} = 0.8$: Experiment (left) vs. Simulation (right).

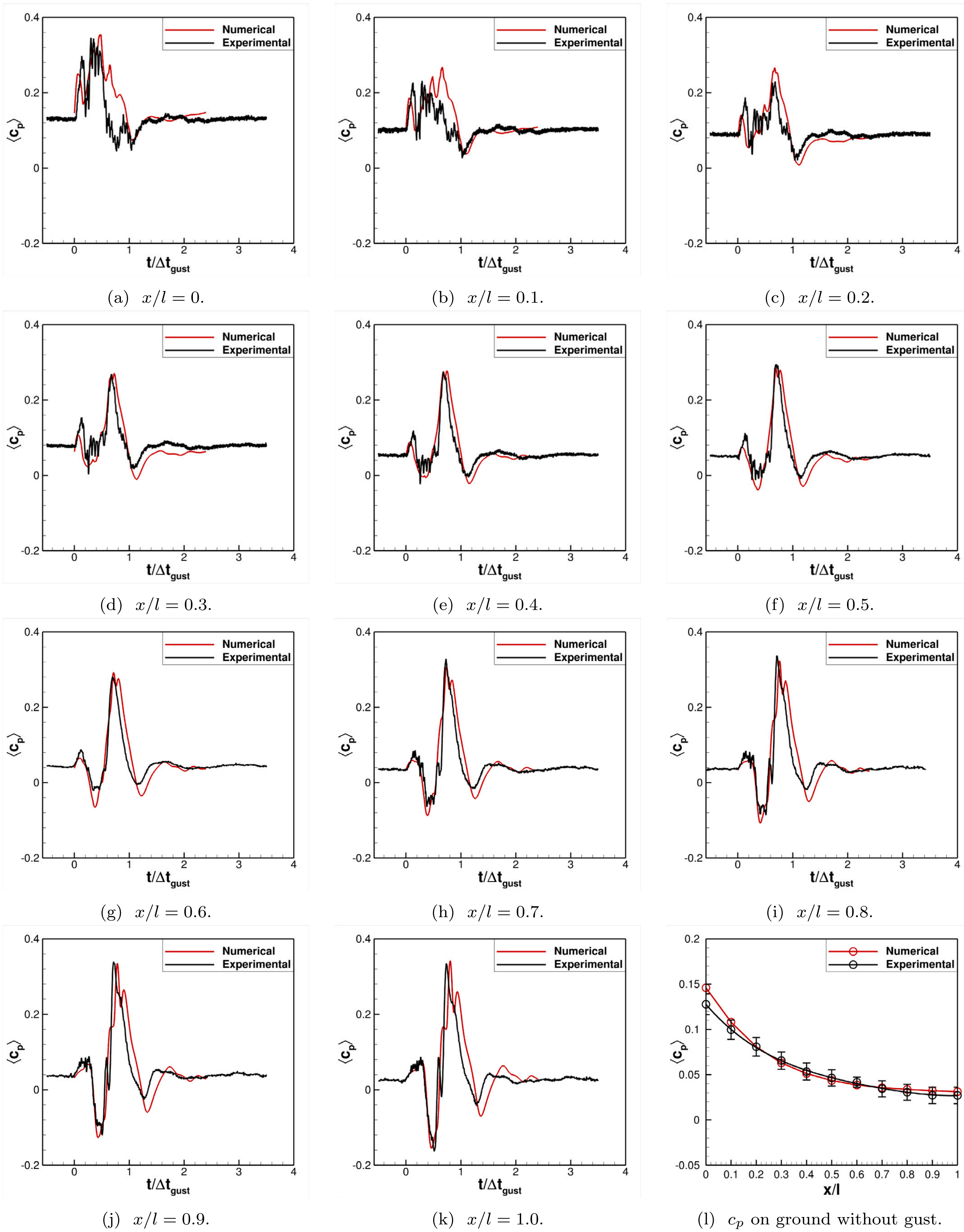


(g) $t/\Delta_{\mathrm{gust}} = 1.0$: Experiment (left) vs. Simulation (right).



(h) $t/\Delta_{\mathrm{gust}} = 1.6$: Experiment (left) vs. Simulation (right).

**Fig. 17.** (*continued*).




(a) $x/l = 0$.
(b) $x/l = 0.1$.
(c) $x/l = 0.2$.
(d) $x/l = 0.3$.
(e) $x/l = 0.4$.
(f) $x/l = 0.5$.
(g) $x/l = 0.6$.
(h) $x/l = 0.7$.
(i) $x/l = 0.8$.
(j) $x/l = 0.9$.
(k) $x/l = 1.0$.
(l) $c_p$ on ground without gust.

**Fig. 18.** Time histories of the phase-averaged experimental surface pressure coefficient [13] and its numerical counterpart at the 11 monitoring points depicted in Fig. 15(b).

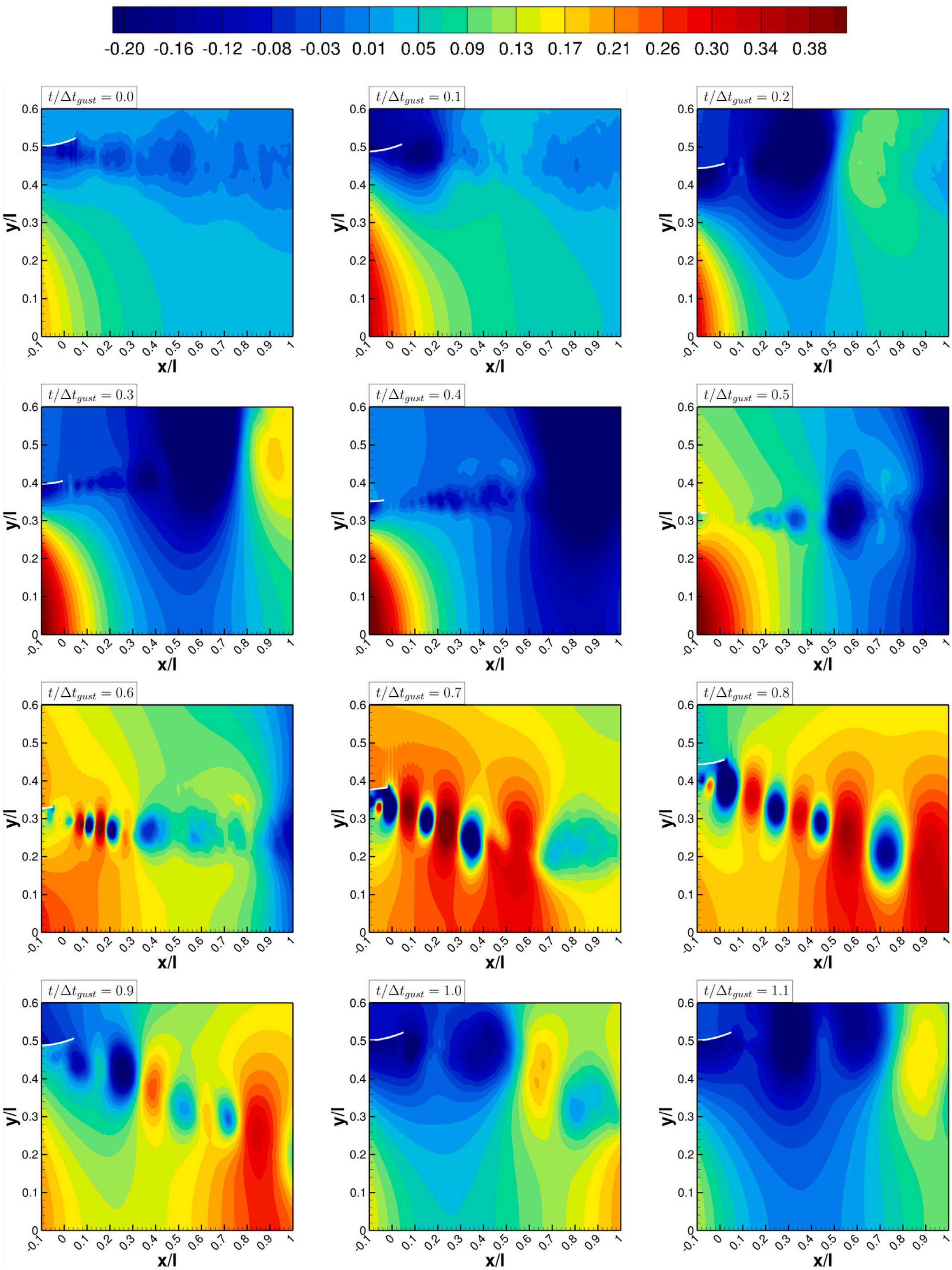



**Fig. 19.** Predicted phase-averaged pressure coefficient in the symmetry plane at different time instants of the gust generation.

## 6. Conclusions

This contribution presents a dynamic immersed boundary method designed for the application on curvilinear block-structured grids, specifically intended for high-fidelity large-eddy simulations of complex flow dynamics. Building on the direct forcing diffuse-interface IB approach of Uhlmann [17], the method employs the moving least square approach of Vanella and Balaras [22] to construct an IB kernel for the Lagrangian markers. In the present study, a new variant tailored for curvilinear block-structured grids is proposed, enabling the interpolation weights for transferring Eulerian flow variables to Lagrangian locations to be determined independently of the dilation parameter. By utilizing a fixed interpolation window, the interpolation weights are evaluated with respect to the computational space rather than the physical space. To construct one-sided kernel functions for Lagrangian points near boundaries, the MLS approach is also applied. This is particularly relevant in the present case, where the nozzle's upper contour traverses between two side walls, preventing the use of symmetric kernels.

The newly devised IB methodology is applied to numerically replicate the highly instantaneous turbulent flow field induced by a recently developed wind gust generator [13]. This generator utilizes a wind tunnel nozzle with a rotatable upper contour, enabling a rapid reduction of the nozzle's outlet area within a short time interval to produce a smooth horizontal wind gust. The motion of the nozzle's upper contour is captured by approximately 1.1 million non-uniformly distributed Lagrangian markers, which move through the flow field following the motion pattern of the experimental setup. Currently, a (1 - cos)-type signal is considered to generate extreme coherent gusts (ECG) as defined in the IEC-Standard [51].

First, the instantaneous flow field induced by the gust generator is analyzed based on the velocity and pressure fields. Unlike the measurements, the predicted data provide all flow quantities across the entire computational domain – both inside the nozzle and within the test section – at a high temporal resolution, allowing for deeper insight into the gust generation process. For a direct comparison with the measurements, the flow data are averaged over 46 gust cycles. The predictions of the phase-averaged data align well the measurements, except for a series of small vortices in the detached shear layer at a later stage of the gust generation. These vortices appear only in the simulations, likely due to the absence of inflow turbulence, which is currently not accounted for in the predictions. In the experiments, a streamwise turbulence intensity of approximately 1.5 % was observed, which is assumed to desynchronize the flow separation during the nozzle's upstroke and the subsequent vortex formation. As a result, these vortical structures are averaged out in the phase-averaged experimental data, whereas they persist in the simulations due to the lack of inflow turbulence. Nevertheless, comparisons with the PIV and CTA data of Wood and Breuer [13,28] reveal that the proposed simulation methodology effectively captures the gust generation process in the wind tunnel. In addition to velocity field comparisons, the analysis of the surface pressure at multiple monitoring points further validated the numerical approach. The predicted phase-averaged pressure coefficients showed good agreement with the measured values in both trend and magnitude, confirming the model's capability to reproduce the unsteady pressure dynamics induced by the gust. Minor deviations observed near the nozzle's outlet were attributed to geometric inaccuracies in the physical setup. Thus, an efficient IB-based method for the application on curvilinear grids has been developed, providing a valuable tool for future studies on the effect of wind gusts on flexible structures and supporting complementary experimental investigations. The IB method proposed can also be combined with the ALE method, as recently demonstrated in [12], offering unprecedented possibilities in the simulation of FSI problems and enabling the selective use of the respective strengths of both approaches. Furthermore, extending the present study beyond the discrete gust scenario to include periodic gust cycles or continuously modulated velocity disturbances offers a natural avenue for future investigations.

## CRediT authorship contribution statement

**K. Boulbrachene:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **M. Breuer:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.


## Acknowledgments

The initial part of the project was financially supported by the German Research Foundation (Deutsche Forschungsgemeinschaft) under the contract number BR 1847/17-1, which ended already. Presently, funding computational resources (HPC cluster HSUper) have been provided by the project hpc.bw, funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr. dtec.bw is funded by the European Union – NextGenerationEU. Last but not least we would like to thank Dr. J.N. Wood (HSU Hamburg) for the fruitful discussions on the topic.


## Appendix A. Minimizing error functions

This section derives the solutions of the minimization problem for the standard MLS method and the Backus–Gilbert formulation of the MLS method.

### A.1. Standard MLS

The standard MLS formulation seeks the vector of unknown coefficients $\mathbf{c}^g(\mathbf{X}_l)$ by minimizing the error function, which is expressed in a matrix form as:

$$\min_{\mathbf{c}^g(\mathbf{X}_l)} \quad J_l = \frac{1}{2}\left(\mathcal{A}^\intercal\,\mathbf{c}^g(\mathbf{X}_l) - \mathbf{g}\right)^\intercal \mathrm{diag}(\boldsymbol{\phi}_l)\,\left(\mathcal{A}^\intercal\,\mathbf{c}^g(\mathbf{X}_l) - \mathbf{g}\right). \tag{A.1}$$

Expanding the terms yields:

$$\begin{aligned} J_l &= \frac{1}{2}\,\left((\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\mathcal{A} - \mathbf{g}^\intercal\right)\mathrm{diag}(\boldsymbol{\phi}_l)\,\left(\mathcal{A}^\intercal\,\mathbf{c}^g(\mathbf{X}_l) - \mathbf{g}\right), \\ &= \frac{1}{2}\,\Big[(\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\,\mathbf{c}^g(\mathbf{X}_l) - (\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathbf{g} \\ &\quad -\mathbf{g}^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\,\mathbf{c}^g(\mathbf{X}_l) + \mathbf{g}^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathbf{g}\Big]. \end{aligned} \tag{A.2}$$

Rewriting the third term in Eq. (A.2) leads to:

$$\begin{aligned} \mathbf{g}^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\,\mathbf{c}^g(\mathbf{X}_l) &= \left[(\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\left(\mathbf{g}^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\right)^\intercal\right]^\intercal \\ &= (\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\left(\mathbf{g}^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\right)^\intercal \\ &= (\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\left(\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\right)^\intercal\,\mathbf{g} \\ &= (\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathbf{g}. \end{aligned} \tag{A.3}$$

Hence Eq. (A.2) simplifies to:

$$J_l = \frac{1}{2}\,\left[(\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\,\mathbf{c}^g(\mathbf{X}_l) - 2\,(\mathbf{c}^g(\mathbf{X}_l))^\intercal\,\mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathbf{g} + \mathbf{g}^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathbf{g}\right]. \tag{A.4}$$

Differentiating $J_l$ with respect to the vector of unknown coefficients $\mathbf{c}^g(\mathbf{X}_l)$ yields:

$$\frac{\partial J_l}{\partial \mathbf{c}^g(\mathbf{X}_l)} = \mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\,\mathbf{c}^g(\mathbf{X}_l) - \mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathbf{g} \tag{A.5}$$

Setting $\dfrac{\partial J_l}{\partial \mathbf{c}^g(\mathbf{X}_l)} = 0$ and solving $\mathbf{c}^g(\mathbf{X}_l)$ leads to:

$$\mathbf{c}^g(\mathbf{X}_l) = \mathcal{G}^{-1}\,\mathcal{B}\,\mathbf{g}, \tag{A.6}$$

where:

$$\mathcal{G} = \mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal, \quad \mathcal{B} = \mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l). \tag{A.7}$$

### A.2. Proof of the MLS method's satisfaction of discrete moment conditions

The proof relies on considering the Backus–Gilbert formulation of the moving least squares method [26,27], which incorporates the polynomial reproduction conditions into the definition of the quadratic minimization problem:

$$\min_{\tilde{\boldsymbol{\phi}}_l} \quad J_l = \frac{1}{2}\,\tilde{\boldsymbol{\phi}}_l^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)^{-1}\,\tilde{\boldsymbol{\phi}}_l,$$
$$\text{subject to:} \quad \mathcal{A}\,\tilde{\boldsymbol{\phi}}_l = \mathbf{p}(\mathbf{X}_l). \tag{A.8}$$

By the use of the Lagrange multiplier $\lambda \in \mathbb{R}^m$, the constrained problem in Eq. (A.8) is reformulated to an unconstrained minimization as:

$$\min_{\tilde{\boldsymbol{\phi}}_l,\lambda_l} \quad \mathcal{L}_l = \frac{1}{2}\tilde{\boldsymbol{\phi}}_l^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)^{-1}\,\tilde{\boldsymbol{\phi}}_l - \lambda_l^\intercal\,\left(\mathcal{A}\,\tilde{\boldsymbol{\phi}}_l - \mathbf{p}(\mathbf{X}_l)\right). \tag{A.9}$$

The solution to the problem defined in Eq. (A.8) is obtained by minimizing the Lagrangian $\mathcal{L}_l$ by imposing the conditions $\dfrac{\partial \mathcal{L}_l}{\partial \tilde{\boldsymbol{\phi}}_l} = 0$ and $\dfrac{\partial \mathcal{L}_l}{\partial \lambda_l} = 0$. First, expanding Eq. (A.9) yields:

$$\min_{\tilde{\boldsymbol{\phi}}_l,\lambda_l} \quad \mathcal{L}_l = \frac{1}{2}\tilde{\boldsymbol{\phi}}_l^\intercal\,\mathrm{diag}(\boldsymbol{\phi}_l)^{-1}\,\tilde{\boldsymbol{\phi}}_l - \lambda_l^\intercal\,\mathcal{A}\,\tilde{\boldsymbol{\phi}}_l + \lambda_l^\intercal\,\mathbf{p}(\mathbf{X}_l). \tag{A.10}$$

The second term appearing in Eq. (A.10) can be written as:

$$\lambda_l^\intercal\,\mathcal{A}\,\tilde{\boldsymbol{\phi}}_l = \left[\tilde{\boldsymbol{\phi}}_l^\intercal\,\left(\lambda_l^\intercal\,\mathcal{A}\right)^\intercal\right]^\intercal = \tilde{\boldsymbol{\phi}}_l^\intercal\,\left(\lambda_l^\intercal\,\mathcal{A}\right)^\intercal = \tilde{\boldsymbol{\phi}}_l^\intercal\,\mathcal{A}^\intercal\,\lambda_l. \tag{A.11}$$

Substituting Eq. (A.11) in Eq. (A.10) and minimizing the Lagrangian with respect to the vector unknown weights (i.e., $\dfrac{\partial \mathcal{L}_l}{\partial \tilde{\boldsymbol{\phi}}_l} = 0$) yields the first equation:

$$\mathrm{diag}(\boldsymbol{\phi}_l)^{-1}\,\tilde{\boldsymbol{\phi}}_l - \mathcal{A}^\intercal\,\lambda_l = 0. \tag{A.12}$$

Similarly, the Lagrangian in Eq. (A.10) is minimized with respect to the Lagrange multiplier (i.e., $\dfrac{\partial \mathcal{L}_l}{\partial \lambda_l} = 0$) to deduce the second equation:

$$\mathcal{A}\,\tilde{\boldsymbol{\phi}}_l - \mathbf{p}(\mathbf{X}_l) = 0. \tag{A.13}$$

The solution to the two equations above is determined by first solving for $\tilde{\boldsymbol{\phi}}_l$ in Eq. (A.12) and substituting the result into Eq. (A.13). This yields the Lagrange multiplier:

$$\lambda_l = \left(\mathcal{A}\,\mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\right)^{-1}\,\mathbf{p}(\mathbf{X}_l) = \mathcal{G}^{-1}\,\mathbf{p}(\mathbf{X}_l), \tag{A.14}$$

where $\mathcal{G}$ is the positive-definite Gram matrix. Finally, the vector of unknown weights reads:

$$\tilde{\boldsymbol{\phi}}_l = \mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\,\lambda_l = \mathrm{diag}(\boldsymbol{\phi}_l)\,\mathcal{A}^\intercal\,\mathcal{G}^{-1}\,\mathbf{p}(\mathbf{X}_l) = \mathcal{B}^\intercal\,\mathcal{G}^{-1}\,\mathbf{p}(\mathbf{X}_l). \tag{A.15}$$

By examining Eqs. (31) and (A.15), it becomes evident that both MLS approaches yield identical weight vectors. Due to the fact that the polynomial reproduction constraints are inherently embedded into the Backus–Gilbert formulation, the equivalence of the two methods indicates that the standard moving least squares method also reproduces the polynomial basis functions.

## Appendix B. Validation

### B.1. Flow past a stationary cylinder

In this test case, the accuracy of the proposed IBM is assessed based on the prediction of the flow past a stationary cylinder. This benchmark is aligned to previous experimental [52,53] and numerical studies [17,44,54]. A circular computational domain of a diameter 30 $D$ is adopted, where $D$ corresponds to the diameter of the cylinder. Assuming that the flow is two-dimensional, the domain size in $z$-direction is set to 1 control volume. The center of the cylinder ($x_{cyl}$, $y_{cyl}$) is located at the center of the computational domain. The domain is discretized using an O-grid with an angular spacing of $\Delta\theta = 2\pi/240 = 0.0262$. In the vicinity of the cylinder, i.e., for $0.25\,D < r < 0.75\,D$, two different resolutions with uniform radial spacings of $\Delta r = 0.0025\,D$ (200 points) and $\Delta r = 0.005\,D$ (400 points) are used. This ensures a similar spatial resolution for the subsequent oscillating cylinder case. Within the inner region (i.e., for $r < 0.25\,D$), the domain is described by a specific O-grid with a square block filling the center, while in the outer region (i.e., for $r > 0.75\,D$), the grid is stretched geometrically with a stretching factor of 1.025. A total of 240 Lagrangian markers are employed, distributed uniformly along the cylinder circumference according to the angular resolution of the Eulerian grid. The thickness of the immersed boundary is set to the radial grid spacing of each respective grid resolution. The computational domain along with a zoomed view of the grid near the cylinder are shown in Fig. B.20.

A uniform inlet velocity $u_\infty$ is set at the front part of the domain, while a convective outflow boundary condition is used at the rear part. Due to the assumption of two-dimensionality, the application of periodic boundary conditions in $z$-direction is permitted. To validate the IBM framework for flow regimes dominated by either viscous or inertial effects, simulations are carried out at Reynolds numbers $Re = u_\infty D/\nu = 20$, 100, and 185. A constant time step is used, corresponding to a CFL number of 0.2.

Fig. B.21 shows snapshots of the $z$-vorticity field of the flow for the different Reynolds numbers along with the corresponding temporal evolution of the drag and lift coefficients for both grid resolutions (i.e., $D/\Delta r = 400$ and $D/\Delta r = 200$) mentioned above. It is worth mentioning that the angular resolution and the number of Lagrangian markers is identical on both grids. At the lowest $Re$ number the prediction converges to a steady-state solution, while the two higher $Re$ numbers show the development of the classical von Kármán vortex street in the wake leading to periodic oscillations of $C_L$ and $C_D$. The computed drag coefficients and Strouhal numbers obtained using the present IB formulation, as well as those from boundary-fitted simulations, are compared in Table B.1 with reference values from the literature. The IB method yields slightly higher drag coefficients than found in experimental data and boundary-fitted results. This is typical for diffuse-interface IB methods, where the fluid–solid interface is smeared over several grid cells, effectively increasing the apparent body size and, consequently, the computed drag. A similar effect explains the slightly higher drag observed for the coarse grid compared to the fine one (see Fig. B.21(b)). If the Lagrangian volume is not explicitly prescribed and the thickness of the immersed boundary is numerically computed, this would generally lead to lower drag coefficients, bringing the results closer to those obtained with boundary-fitted methods [17,47]. Nonetheless, the results obtained with both resolutions are nearly identical, confirming the grid independence of the solution. Overall, good agreement with the reference data is found. To complement the quantitative validation and to provide a direct visual assessment of the boundary condition enforcement, Fig. B.22 shows streamline plots around the stationary cylinder at the three Reynolds numbers. In all cases, the streamlines remain tangential to the surface, clearly demonstrating that the proposed immersed boundary

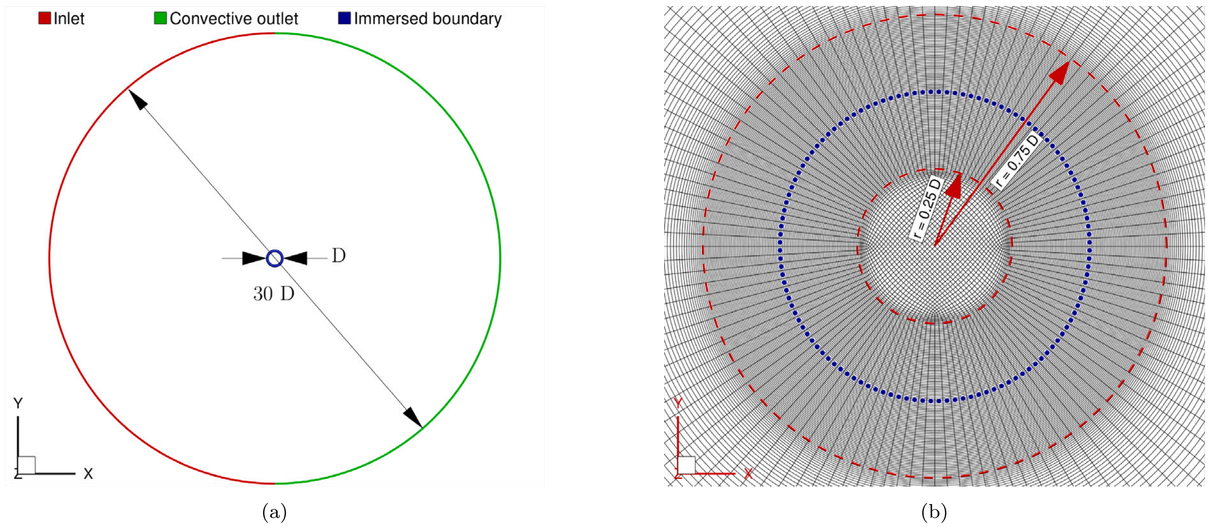



**Fig. B.20.** Test case setup: (a) Computational domain and boundary conditions. (b) Zoom of the computational grid around the cylinder in the $x$–$y$ plane. Only every second grid line in each direction is displayed and the Lagrangian markers are indicated by blue circles.

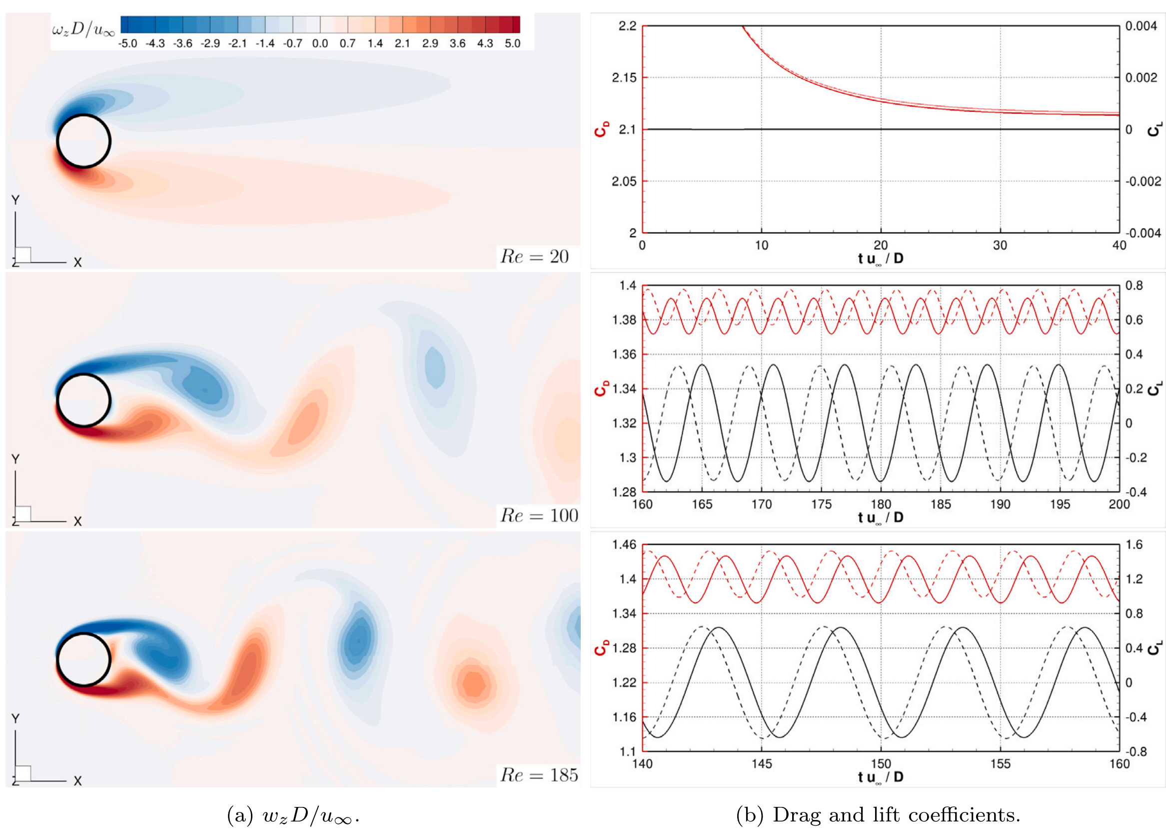



**Fig. B.21.** Snapshot of (a) the $z$-vorticity field and (b) the time history of the drag and lift coefficients of the fine grid (continuous lines) and the coarse grid (dashed lines).

formulation accurately enforces the no-slip condition at the fluid–solid interface.

### B.2. Flow past an oscillating cylinder

In this section, the case of a transversely oscillating cylinder in a cross-flow is considered as a benchmark case to assess the accuracy of the IBM in scenarios involving forced motion of the immersed boundary. The computational domain, boundary conditions, and grid setup used here are the same as those described in Appendix B.1. The cylinder undergoes a time-periodic motion perpendicular to the main flow direction, with its vertical position defined by:

$$y_{cyl}(t) = A_e \sin(2\pi f_e t) \quad \text{with} \quad \frac{A_e}{D} = 0.2, \quad \frac{f_e}{f_0} = 0.8. \tag{B.1}$$

Here, $f_0$ represents the natural shedding frequency of the flow past a stationary cylinder determined in Appendix B.1. The Reynolds number is set to $Re = 185$, resulting in the same flow conditions as those considered in Uhlmann [17]. This configuration yields a vortex shedding frequency that synchronizes with the imposed oscillation

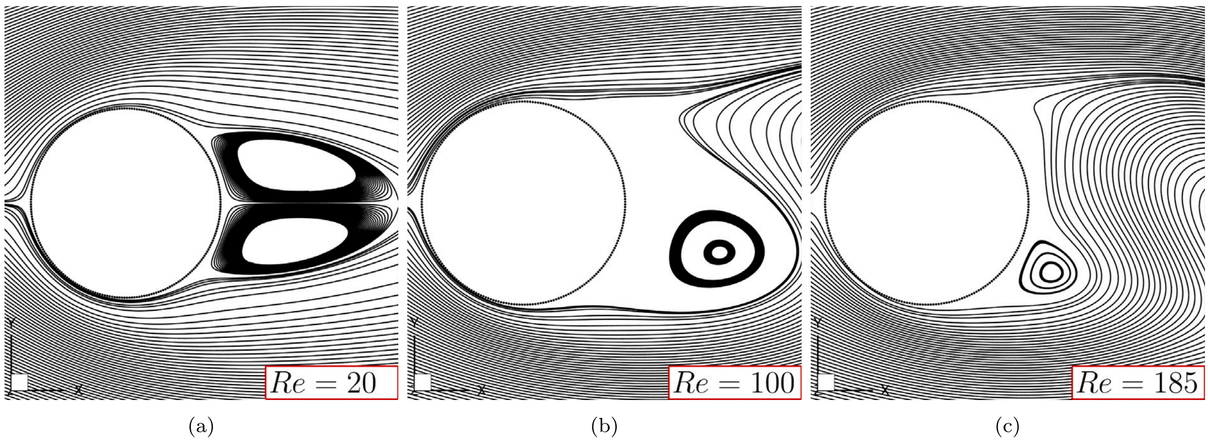



**Fig. B.22.** Snapshots of the streamlines for the flow past a stationary cylinder at (a) $Re = 20$, (b) $Re = 100$, and (c) $Re = 185$..

**Table B.1**
Mean drag coefficient $\overline{C}_D$ and Strouhal number $St$ for the flow past a stationary cylinder at different Reynolds numbers. The corresponding lateral domain sizes $H_y/D$ and grid resolutions $D/\Delta x$ used in the simulations are also listed.

| | Method | $H_y/D$ | $D/\Delta x$ | $\overline{C}_D$ | $St$ |
|---|---|---|---|---|---|
| Re = 20 | | | | | |
| Tritton [52] | Experiment | – | – | 2.09 | – |
| Taira and Colonius [55] | Diffuse-interface IBM | 60 | 50 | 2.06 | – |
| Yi et al. [56] | Diffuse-interface IBM | 16 | 120 | 2.066 | – |
| Present | Diffuse-interface IBM | 30 | 200 | 2.115 | – |
| Present | Diffuse-interface IBM | 30 | 400 | 2.113 | – |
| Present | Boundary-fitted | 30 | 400 | 2.111 | – |
| Re = 100 | | | | | |
| Liu et al. [57] | Boundary-fitted | – | – | 1.350 | 0.165 |
| Mittal et al. [58] | Sharp-interface IBM | 40 | 66.7 | 1.350 | 0.166 |
| Uhlmann [17] | Diffuse-interface IBM | 40 | 38.4 | 1.453 | 0.169 |
| Yi et al. [56] | Diffuse-interface IBM | 16 | 120 | 1.354 | 0.165 |
| Present | Diffuse-interface IBM | 30 | 200 | 1.387 | 0.168 |
| Present | Diffuse-interface IBM | 30 | 400 | 1.382 | 0.167 |
| Present | Boundary-fitted | 30 | 400 | 1.364 | 0.168 |
| Re = 185 | | | | | |
| Lu and Dalton [53] | Experiment | – | – | 1.28 | 0.195 |
| Guilmineau and Queutey [54] | Boundary-fitted | 51 | 100 | 1.287 | 0.195 |
| Yang and Balaras [59] | Sharp-interface IBM | 30 | – | 1.366 | 0.197 |
| Pinelli et al. [44] | Diffuse-interface IBM | 34 | 200 | 1.430 | 0.196 |
| Yi et al. [56] | Diffuse-interface IBM | 16 | 120 | 1.352 | 0.193 |
| Present | Diffuse-interface IBM | 30 | 200 | 1.408 | 0.196 |
| Present | Diffuse-interface IBM | 30 | 400 | 1.399 | 0.195 |
| Present | Boundary-fitted | 30 | 400 | 1.347 | 0.195 |

frequency of the cylinder, indicating a lock-in condition. Fig. B.23 shows the drag coefficient as a function of the instantaneous vertical position of the cylinder once a quasi-steady state is reached. The results obtained with the present IB method at the two different grid resolutions are compared against the boundary-fitted ALE simulation and previous reference studies [17,56,60]. Overall, the IB simulations reproduce the general trend of the drag coefficient with respect to the cylinder's vertical displacement and the computed curves fall within the range bounded by the reference data. The minor spurious oscillations observed in the coarse grid simulation are significantly reduced upon grid refinement, reflecting the improvement in interface representation. These oscillations are attributed to the use of a 3-point kernel, which is set by the MLS construction to satisfy only up to the first moment condition and has a weak translational invariance. Table B.2 summarizes the time-averaged drag coefficients ($\overline{C}_D$), the mean oscillation amplitudes of both drag ($C'_D$) and lift ($C'_L$) coefficients, and the root-mean-square

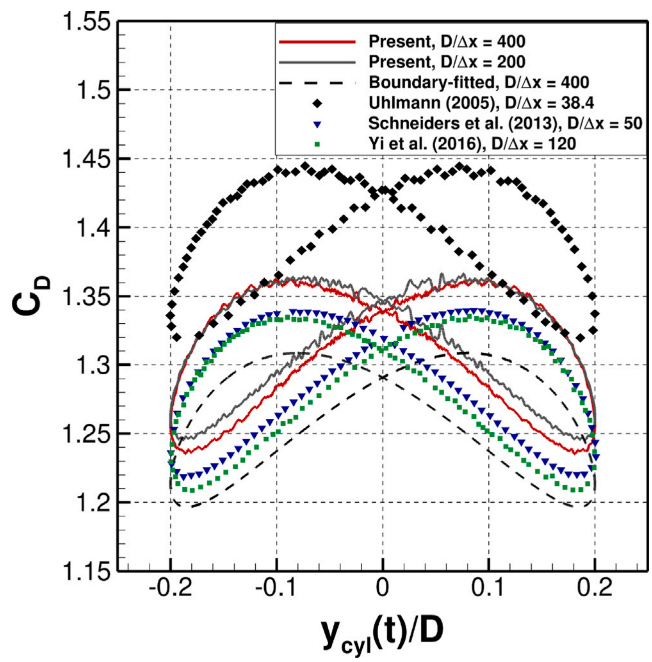
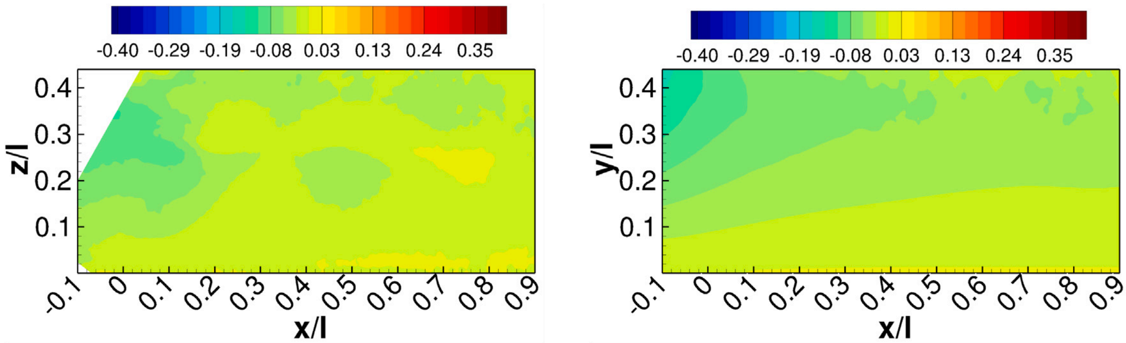



**Fig. B.23.** Variation of the drag coefficient for a transversely oscillating circular cylinder in a uniform cross-flow at $Re = 185$ as a function of the cylinder's vertical position.

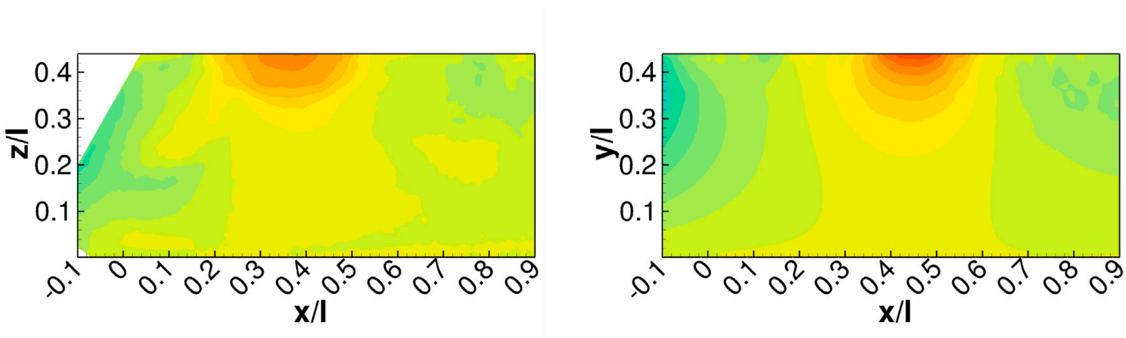
**Table B.2**
Drag and lift coefficients and associated quantities for the flow past an oscillating cylinder at $Re = 185$.

| | $\overline{C}_D$ | $C'_D$ | $C'_L$ | $C_{L,\mathrm{RMS}}$ |
|---|---|---|---|---|
| Lu and Dalton [53] | 1.25 | – | – | 0.18 |
| Uhlmann [17] | 1.354 | ±0.065 | – | 0.166 |
| Schneiders et al. [60] | 1.279 | – | – | 0.082 ∼ 0.086 |
| Yi et al. [56] | 1.273 | ±0.062 | ±0.124 | 0.083 |
| Present ($D/\Delta x = 200$) | 1.303 | ±0.060 | ±0.132 | 0.089 |
| Present ($D/\Delta x = 400$) | 1.297 | ±0.063 | ±0.127 | 0.085 |
| Present boundary-fitted ($D/\Delta x = 400$) | 1.253 | ±0.056 | ±0.110 | 0.076 |

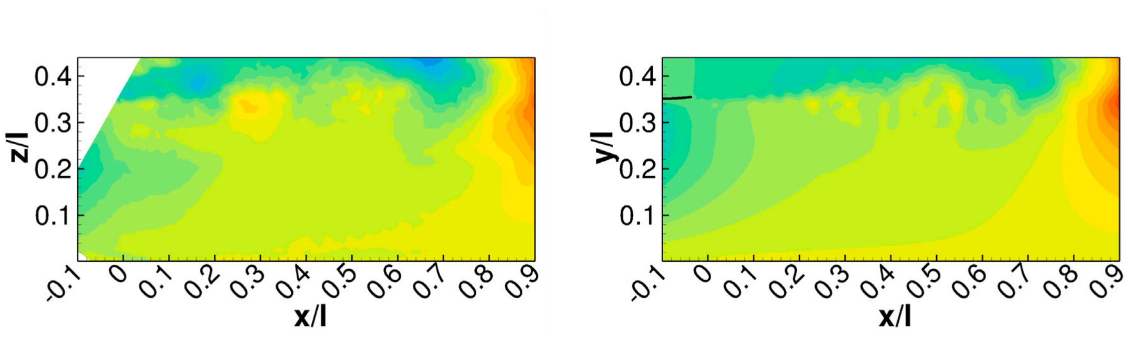
value of the lift coefficient ($C_{L,RMS}$). To accurately evaluate the lift force, the inertial contribution of the "pseudo-fluid" enclosed by the cylinder is subtracted from the total Lagrangian force summation [17]. The obtained results show very good agreement with the reference data confirming reproducibility and reliability of the predictions.



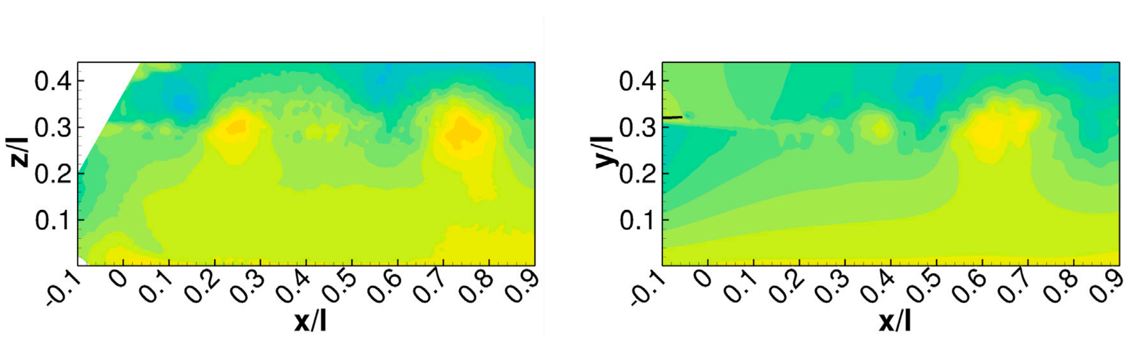

(a) $t/\Delta_{\mathrm{gust}} = 0$: Experiment (left) vs. Simulation (right).




(b) $t/\Delta_{\mathrm{gust}} = 0.2$: Experiment (left) vs. Simulation (right).

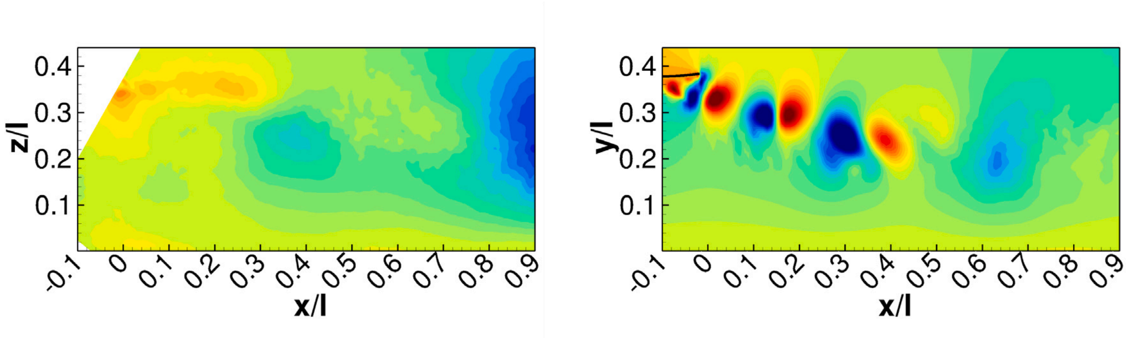



(c) $t/\Delta_{\mathrm{gust}} = 0.4$: Experiment (left) vs. Simulation (right).

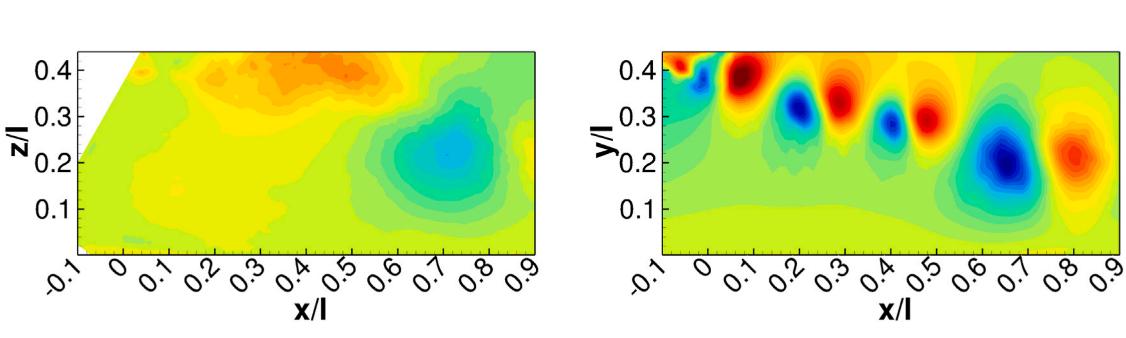



(d) $t/\Delta_{\mathrm{gust}} = 0.5$: Experiment (left) vs. Simulation (right).

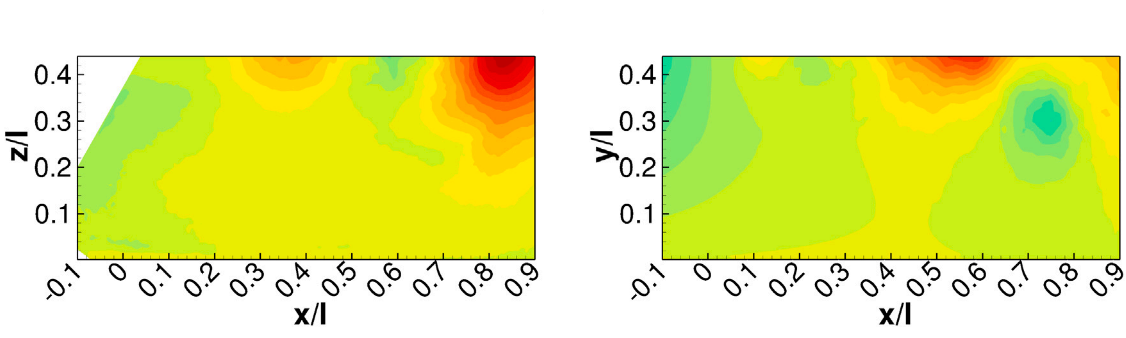
**Fig. D.24.** Comparison between measured and predicted data of the phase-averaged vertical velocity in the symmetry plane at different time instants of the gust generation. Experimental data taken from Wood and Breuer [13].



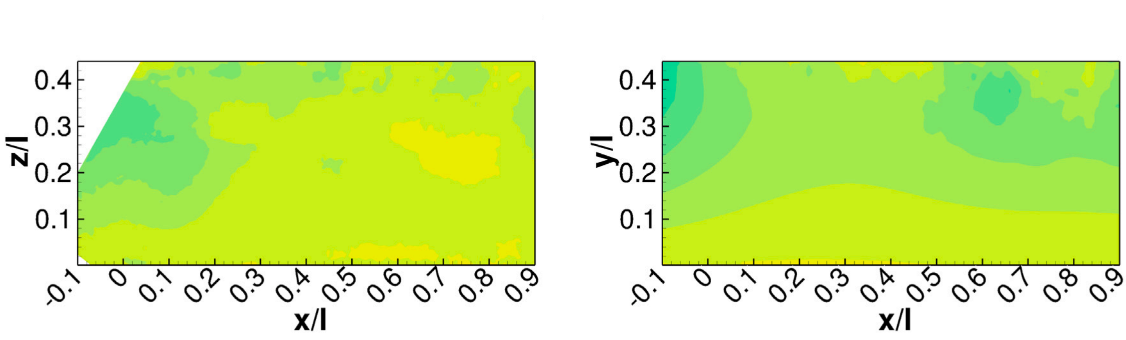

(e) $t/\Delta_{\text{gust}} = 0.7$: Experiment (left) vs. Simulation (right).




(f) $t/\Delta_{\text{gust}} = 0.8$: Experiment (left) vs. Simulation (right).




(g) $t/\Delta_{\text{gust}} = 1.0$: Experiment (left) vs. Simulation (right).




(h) $t/\Delta_{\text{gust}} = 1.6$: Experiment (left) vs. Simulation (right).

**Fig. D.24.** (*continued*).

## Appendix C. Transforming the saddle's translational motion to the nozzle's rotational motion

In this section, a detailed description on the transformation of the vertical motion of the WGG's saddle to the corresponding rotational motion of the nozzle is presented. The following two geometric conditions hold throughout the nozzle's motion (see Fig. 11):

$$b^2 = (x_s - x_n)^2 + (y_s - y_n)^2, \tag{C.1}$$

$$r^2 = x_n^2 + y_n^2, \quad \text{where } r = 0.336 \text{ m}. \tag{C.2}$$

By substituting $x_s = a = 0.295$ m, $x_n = r\cos\theta$ and $y_n = r\sin\theta$ in Eq. (C.1), one can find the relation between the coordinate $y_s$ and the angle of rotation $\theta$:

$$b^2 = (a - r\cos\theta)^2 + (y_s - r\sin\theta)^2,$$
$$(y_s - r\sin\theta)^2 = b^2 - (a - r\cos\theta)^2,$$
$$(y_s - r\sin\theta) = \sqrt{b^2 - (a - r\cos\theta)^2},$$
$$y_s = r\sin\theta + \sqrt{b^2 - (a - r\cos\theta)^2} \tag{C.3}$$

Differentiating Eq. (C.3) with respect to time and solving for $\omega_z$ yields:

$$v_s = \omega_z\, r\cos\theta - \frac{(a - r\cos\theta)\,\omega_z\, r\sin\theta}{\sqrt{b^2 - (a - r\cos\theta)^2}},$$
$$\omega_z = \frac{v_s}{x_n - \frac{(a-x_n)\,y_n}{\sqrt{b^2-(a-x_n)^2}}}. \tag{C.4}$$

The unknown coordinates of the point on the nozzle (i.e., $x_n(t)$ and $y_n(t)$) are determined by expanding Eq. (C.1) as:

$$(a - x_n)^2 + (y_s - y_n)^2 = b^2,$$
$$a^2 - 2x_n a + x_n^2 + y_s^2 - 2y_n y_s + y_n^2 = b^2. \tag{C.5}$$

Eq. (C.5) can be simplified replacing $x_n^2 + y_n^2$ by $r^2$ according to Eq. (38) leading to:

$$x_n a + y_n y_s = \underbrace{\frac{r^2 + a^2 + y_s^2 - b^2}{2}}_{k}.$$

This yields the following system of equations to be solved:

$$x_n a + y_n y_s = k, \qquad x_n^2 + y_n^2 = r^2,$$

whose solutions are:

$$y_n = \frac{-B - \sqrt{B^2 - 4AC}}{2A}, \qquad x_n = \frac{k - y_n y_s}{a},$$

where:

$$A = y_s^2 + a^2, \qquad B = -2\,k\,y_s, \qquad C = k^2 - r^2 a^2.$$

Note that in the solution for $y_n(t)$, the negative root is chosen to ensure $y_n(t) < y_s(t)$ and to maintain the consistency with the coordinate system defined in Fig. 11.

## Appendix D. Vertical velocity field

In Section 5.2.1 a comparison between the measured and the predicted phase-averaged streamwise velocity contours in the symmetry plane was depicted in Fig. 17. Fig. D.24 complements this comparison by providing the vertical velocity component of the phase-averaged velocity fields at different instants of the gust generation process.

## Data availability

Data will be made available on request.

## References

[1] Michalski A, Gawenat B, Gelenne P, Haug E. Computational wind engineering of large umbrella structures. J Wind Eng Ind Aerodyn 2015;144:96–107. http://dx.doi.org/10.1016/j.jweia.2015.05.002.

[2] Li S, Catarelli RA, Phillips BM, Bridge JA, Gurley KR. Physical simulation of downburst winds for civil structures: A review. J Wind Eng Ind Aerodyn 2024;254:105900. http://dx.doi.org/10.1016/j.jweia.2024.105900.

[3] Babaei R, Graat K, Chan C, Savory E. Experimental simulation of stationary and traveling density-driven thunderstorm downbursts using the two-fluid model. J Wind Eng Ind Aerodyn 2021;211:104553. http://dx.doi.org/10.1016/j.jweia.2021.104553.

[4] Junayed C, Jubayer C, Parvu D, Romanic D, Hangan H. Flow field dynamics of large-scale experimentally produced downburst flows. J Wind Eng Ind Aerodyn 2019;188:61–79. http://dx.doi.org/10.1016/j.jweia.2019.02.008.

[5] Aboutabikh M, Ghazal T, Chen J, Elgamal S, Aboshosha H. Designing a blade-system to generate downburst outflows at boundary layer wind tunnel. J Wind Eng Ind Aerodyn 2019;186:169–91. http://dx.doi.org/10.1016/j.jweia.2019.01.005.

[6] Alawode KJ, Azzi Z, Elawady A, Chowdhury AG. Dynamic properties of an aeroelastic transmission tower subjected to synoptic and downburst-like outflows. J Wind Eng Ind Aerodyn 2023;242:105557. http://dx.doi.org/10.1016/j.jweia.2023.105557.

[7] Butler K, Cao S, Kareem A, Tamura Y, Ozono S. Surface pressure and wind load characteristics on prisms immersed in a simulated transient gust front flow field. J Wind Eng Ind Aerodyn 2010;98(6):299–316. http://dx.doi.org/10.1016/j.jweia.2009.11.003.

[8] Catarelli RA, Fernández-Cabán PL, Phillips BM, Bridge JA, Masters FJ, Gurley KR, et al. Automation and new capabilities in the university of florida NHERI boundary layer wind tunnel. Front Built Environ 2020;6:558151. http://dx.doi.org/10.3389/fbuil.2020.558151.

[9] Wood JN, Breuer M, Neumann T. A novel approach for artificially generating horizontal wind gusts based on a movable plate: The paddle. J Wind Eng Ind Aerodyn 2022;230:105170. http://dx.doi.org/10.1016/j.jweia.2022.105170.

[10] Breuer M, Neumann T. Synchronous high-speed measurements of a flexible structure under wind gust load. Phys Fluids 2024;36(7):075147. http://dx.doi.org/10.1063/5.0215724.

[11] Boulbrachene K, Breuer M. A numerical method to mimic an experimental wind gust generator: The immersed boundary gust generator. Phys Fluids 2024;36(1):015146. http://dx.doi.org/10.1063/5.0187787.

[12] Breuer M, Boulbrachene K, De Nayer G. Combination of two FSI methods and their validation based on artificial wind gusts impacting a flexible T-structure. Comput & Fluids 2025;294:106621. http://dx.doi.org/10.1016/j.compfluid.2025.106621.

[13] Wood JN, Breuer M. Experimental study on a novel wind gust generator based on an adaptive nozzle design. J Wind Eng Ind Aerodyn 2025;261:106080. http://dx.doi.org/10.1016/j.jweia.2025.106080.

[14] Sen S, De Nayer G, Breuer M. A fast and robust hybrid method for block–structured mesh deformation with emphasis on FSI–LES applications. Int J Numer Meth. Engng 2017;111(3):273–300. http://dx.doi.org/10.1002/nme.5465.

[15] Peskin CS. The immersed boundary method. Acta Numer 2002;11:479–517. http://dx.doi.org/10.1017/S0962492902000077.

[16] Mittal R, Iaccarino G. Immersed boundary methods. Annu Rev Fluid Mech 2005;37(1):239–61. http://dx.doi.org/10.1146/annurev.fluid.37.061903.175743.

[17] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. J Comput Phys 2005;209(2):448–76. http://dx.doi.org/10.1016/j.jcp.2005.03.017.

[18] Roma AM, Peskin CS, Berger MJ. An adaptive version of the immersed boundary method. J Comput Phys 1999;153(2):509–34. http://dx.doi.org/10.1006/jcph.1999.6293.

[19] Stockie JM. Analysis and computation of immersed boundaries, with application to pulp fibres (Ph.D. thesis), Retrospective theses and dissertations, 1919-2007, University of British Columbia, Canada; 1997, http://dx.doi.org/10.14288/1.0080530.

[20] Bao Y, Kaye J, Peskin CS. A Gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance. J Comput Phys 2016;316:139–44. http://dx.doi.org/10.1016/j.jcp.2016.04.024.

[21] Wendland H. Local polynomial reproduction and moving least squares approximation. IMA J Numer Anal 2001;21(1):285–300. http://dx.doi.org/10.1093/imanum/21.1.285.

[22] Vanella M, Balaras E. A moving-least-squares reconstruction for embedded-boundary formulations. J Comput Phys 2009;228(18):6617–28. http://dx.doi.org/10.1016/j.jcp.2009.06.003.

[23] Krishnan S, Shaqfeh ES, Iaccarino G. Fully resolved viscoelastic particulate simulations using unstructured grids. J Comput Phys 2017;338:313–38. http://dx.doi.org/10.1016/j.jcp.2017.02.068.

[24] Saadat A, Guido CJ, Iaccarino G, Shaqfeh ESG. Immersed-finite-element method for deformable particle suspensions in viscous and viscoelastic media. Phys Rev E 2018;98:063316. http://dx.doi.org/10.1103/PhysRevE.98.063316.

[25] Li D, Wei A, Luo K, Fan J. An improved moving-least-squares reconstruction for immersed boundary method. Int J Numer Methods Eng 2015;104(8):789–804. http://dx.doi.org/10.1002/nme.4949.

[26] Bhalla APS. On immersed boundary kernel functions: A constrained quadratic minimization perspective. 2021, arXiv preprint arXiv:2111.11025.

[27] Bale R, Bhalla APS, Griffith BE, Tsubokura M. A one-sided direct forcing immersed boundary method using moving least squares. J Comput Phys 2021;440:110359. http://dx.doi.org/10.1016/j.jcp.2021.110359.

[28] Wood JN, Breuer M. Artificial wind gust generation based on an adaptive nozzle design. In: Rauh C, Ruck B, Leder A, editors. Proc. der 31. Gala–Fachtagung 'Experimentelle Strömungsmechanik', Sept. 3–5, 2024. Berlin; 2024, 9–1–9–8.

[29] Durst F, Schäfer M. A parallel block-structured multigrid method for the prediction of incompressible flows. Int J Numer Methods Fluids 1996;22(6):549–65. http://dx.doi.org/10.1002/(SICI)1097-0363(19960330)22:6<549::AID-FLD366>3.0.CO;2-7.

[30] Khosla PK, Rubin SG. A diagonally dominant second-order accurate implicit scheme. Comput & Fluids 1974;2(2):207–9. http://dx.doi.org/10.1016/0045-7930(74)90014-0.

[31] Ferziger JH, Perić M. Computational methods for fluid dynamics. 3rd ed.. Springer Berlin; 2002, http://dx.doi.org/10.1007/978-3-319-99693-6.

[32] Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing-edge separation. AIAA J 1983;21(11):1525–32. http://dx.doi.org/10.2514/3.8284.

[33] Breuer M, De Nayer G, Münsch M, Gallinger T, Wüchner R. Fluid–structure interaction using a partitioned semi–implicit predictor-corrector coupling scheme for the application of large–eddy simulation. J Fluids Struct 2012;29:107–30. http://dx.doi.org/10.1016/j.jfluidstructs.2011.09.003.

[34] Stone HL. Iterative solution of implicit approximations of multidimensional partial differential equations. SIAM J Numer Anal 1968;5(3):530–58. http://dx.doi.org/10.1137/0705044.

[35] Smagorinsky J. General circulation experiments with the primitive equations I: The basic experiment. Mon Weather Rev 1963;91(3):99–165. http://dx.doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

[36] De Nayer G, Kalmbach A, Breuer M, Sicklinger S, Wüchner R. Flow past a cylinder with a flexible splitter plate: A complementary experimental–numerical investigation and a new FSI test case (FSI-PfS-1a). Comput & Fluids 2014;99:18–43. http://dx.doi.org/10.1016/j.compfluid.2014.04.020.

[37] De Nayer G, Breuer M. Numerical FSI investigation based on LES: Flow past a cylinder with a flexible splitter plate involving large deformations (FSI-PfS-2a). Int J Heat Fluid Flow 2014;50:300–15. http://dx.doi.org/10.1016/j.ijheatfluidflow.2014.08.013.

[38] De Nayer G, Apostolatos A, Wood JN, Bletzinger KU, Wüchner R, Breuer M. Numerical studies on the instantaneous fluid–structure interaction of an air–Inflated flexible membrane in turbulent flow. J Fluids Struct 2018;82:577–609. http://dx.doi.org/10.1016/j.jfluidstructs.2018.08.005.

[39] De Nayer G, Breuer M, Wood JN. Numerical investigations on the dynamic behavior of a 2-DOF airfoil in the transitional Re number regime based on fully coupled simulations relying on an eddy-resolving technique. Int J Heat Fluid Flow 2020;85:108631. http://dx.doi.org/10.1016/j.ijheatfluidflow.2020.108631.

[40] De Nayer G, Breuer M, Boulbrachene K. FSI simulations of wind gusts impacting an air-inflated flexible membrane at Re=100,000. J Fluids Struct 2022;109:103462. http://dx.doi.org/10.1016/j.jfluidstructs.2021.103462.

[41] De Nayer G, Breuer M, Goldbach A-K. Fluid-structure interaction simulations of wind gusts impacting a hyperbolic paraboloid tensile structure. Phys Fluids 2024;36(10):105152. http://dx.doi.org/10.1063/5.0232369.

[42] Wood JN, De Nayer G, Schmidt S, Breuer M. Experimental investigation and large–eddy simulation of the turbulent flow past a smooth and rigid hemisphere. J Flow, Turbul Combust 2016;97(1):79–119. http://dx.doi.org/10.1007/s10494-015-9690-5.

[43] Schäfer F, Breuer M. Comparison of c–space and p–space particle tracing schemes on high–performance computers: Accuracy and performance. Int J Numer Methods Fluids 2002;39(4):277–99. http://dx.doi.org/10.1002/fld.324.

[44] Pinelli A, Naqavi I, Piomelli U, Favier J. Immersed-boundary methods for general finite-difference and finite-volume Navier-Stokes solvers. J Comput Phys 2010;229(24):9073–91. http://dx.doi.org/10.1016/j.jcp.2010.08.021.

[45] Zhou K, Balachandar S. An analysis of the spatio-temporal resolution of the immersed boundary method with direct forcing. J Comput Phys 2021;424:109862. http://dx.doi.org/10.1016/j.jcp.2020.109862.

[46] Zhou K, Ding Z, Sun K. Is Lagrangian weight crucial in the direct forcing immersed boundary method? J Phys: Conf Ser 2019;1324:012081. http://dx.doi.org/10.1088/1742-6596/1324/1/012081.

[47] Abdol Azis MH, Evrard F, van Wachem B. An immersed boundary method for incompressible flows in complex domains. J Comput Phys 2019;378:770–95. http://dx.doi.org/10.1016/j.jcp.2018.10.048.

[48] Moody LF. An approximate formula for pipe friction factors. Trans asME 1947;69(12):1005–11.

[49] Ducros F, Nicoud F, Poinsot T. Wall-adapting local eddy-viscosity models for simulations in complex geometries. In: Baines M, editor. Numerical methods for fluid dynamics VI. 1998, p. 293–9.

[50] Germano M, Piomelli U, Moin P, Cabot WH. A dynamic subgrid-scale eddy viscosity model. Phys Fluids A 1991;3:1760–5. http://dx.doi.org/10.1063/1.857955.

[51] IEC-Standard. 61400-21. Tech. Rep., measurement and assessment of power quality of grid connected wind turbines. 2002.

[52] Tritton DJ. Experiments on the flow past a circular cylinder at low Reynolds numbers. J Fluid Mech 1959;6(4):547–67. http://dx.doi.org/10.1017/S0022112059000829.

[53] Lu X-Y, Dalton C. Calculation of the timing of vortex formation from an oscillating cylinder. J Fluids Struct 1996;10(5):527–41. http://dx.doi.org/10.1006/jfls.1996.0035.

[54] Guilmineau E, Queutey P. A numerical simulation of vortex shedding from an oscillating circular cylinder. J Fluids Struct 2002;16(6):773–94. http://dx.doi.org/10.1006/jfls.2002.0449.

[55] Taira K, Colonius T. The immersed boundary method: A projection approach. J Comput Phys 2007;225(2):2118–37. http://dx.doi.org/10.1016/j.jcp.2007.03.005.

[56] Yi W, Corbett D, Yuan X-F. An improved Rhie–Chow interpolation scheme for the smoothed-interface immersed boundary method. Internat J Numer Methods Fluids 2016;82(11):770–95. http://dx.doi.org/10.1002/fld.4240.

[57] Liu C, Zheng X, Sung C. Preconditioned multigrid methods for unsteady incompressible flows. J Comput Phys 1998;139(1):35–57. http://dx.doi.org/10.1006/jcph.1997.5859.

[58] Mittal R, Dong H, Bozkurttas M, Najjar F, Vargas A, von Loebbecke A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. J Comput Phys 2008;227(10):4825–52. http://dx.doi.org/10.1016/j.jcp.2008.01.028.

[59] Yang J, Balaras E. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. J Comput Phys 2006;215(1):12–40. http://dx.doi.org/10.1016/j.jcp.2005.10.035.

[60] Schneiders L, Hartmann D, Meinke M, Schröder W. An accurate moving boundary formulation in cut-cell methods. J Comput Phys 2013;235(C):786–809. http://dx.doi.org/10.1016/j.jcp.2012.09.038.