Measurement Science
and Technology



**PAPER** • **OPEN ACCESS**

# Iterative domain partitioning in pressure from PIV

To cite this article: Alex M Donu *et al* 2026 *Meas. Sci. Technol.* **37** 245204

Measurement Science and Technology



PAPER

# Iterative domain partitioning in pressure from PIV

Alex M Donţu*, Abbas Daliri and Fulvio Scarano

Aerospace Engineering Department, Delft University of Technology, Delft, The Netherlands

* Author to whom any correspondence should be addressed.

E-mail: a.dontu@student.tudelft.nl

OPEN ACCESS

RECEIVED
26 January 2026

REVISED
30 April 2026

ACCEPTED FOR PUBLICATION
20 May 2026

PUBLISHED
15 June 2026

Keywords: pressure from PIV, particle image velocimetry, particle tracking velocimetry, domain partitioning, iterative domain partitioning



## Abstract

A technique that generalizes pressure reconstruction from planar and volumetric particle image velocimetry (PIV) velocity data is proposed to eliminate the need for a user-specified choice of boundary conditions prior to pressure gradient integration. The method extends the dual model concept proposed by Jux *et al* (2020 *Meas. Sci. Technol.* **31** 104001), which partitions the measurement domain into irrotational (*IRR*) and rotational (*ROT*) regions using the total pressure coefficient as a discriminant. The latter approach requires an educated choice of the *IRR* boundary condition location as well as a threshold (minimum) value set for the total pressure coefficient in the *IRR* region. Iterative domain partitioning (*IDP*) approaches the problem by evaluating *IRR* and *ROT* iteratively and thereby significantly reducing the result sensitivity to the user's initial choice of *IRR* boundary and threshold value. Furthermore, a near-wall adaptation of the integration, based on iso-potential lines, is introduced to avoid excessive error propagation when integrating PIV data along the object boundaries. The method is demonstrated to reliably converge towards a unique *IRR-ROT* boundary and minimize error propagation. The verification is carried out by examining three-dimensional (3D) particle tracking velocimetry data, reduced to planar, from the steady flow over a NACA0015 airfoil at $\alpha = [0°, 10°, 20°]$. Monte Carlo simulation with randomly varying boundary conditions yields the lowest dispersion of pressure distribution, as a metric to represent the method's accuracy. The approach is generalized to data from 3D PIV around a wall-mounted cube where the surface pressure reconstruction reveals a distribution consistent with the flow field topology. Also, in 3D, variation of the boundary conditions yields minimal fluctuations of the reconstructed pressure field when using *IDP*.

## 1. Introduction

Pressure reconstruction from particle image velocimetry (PIV) has, over the past few decades, emerged as a viable, non-intrusive alternative to conventional pressure measurements (van Oudheusden 2013) such as pressure orifices connected to transducers or the more advanced pressure-sensitive paint (Klein *et al* 2005). Advances in PIV techniques, including the introduction of high-speed hardware and three-dimensional (3D) systems (Elsinga *et al* 2006, Schanz *et al* 2016) have expanded measurement capabilities to 3D, temporally resolved velocity measurements (Scarano 2012), greatly enhancing the accuracy and applicability of PIV-based pressure reconstruction in complex aerodynamic flows.

The concept has been pioneered in the seminal studies by Gurka *et al* (1999) and Baur (1999), outlining two fundamental approaches to the numerical treatment of the problem: *direct integration* (*DI*) of the pressure gradient and solution of the Poisson pressure equation (*PPE*). The pressure gradient spatial distribution can be readily retrieved from the velocity data invoking the Navier–Stokes equations. For time-averaged turbulent flows, the following equation, obtained from the Reynolds averaging of the momentum equation, needs to be numerically evaluated:

$$\nabla \bar{p} = -\rho\ (\bar{\boldsymbol{u}} \cdot \nabla)\, \bar{\boldsymbol{u}} - \ \rho \nabla \cdot \left(\overline{\boldsymbol{u}'\boldsymbol{u}'}\right) + \ \mu\, \nabla^2 \bar{\boldsymbol{u}} \tag{1}$$

where $\overline{p}$ is the pressure field, $\rho$ and $\mu$ are the (known) fluid density and viscosity, $\overline{\boldsymbol{u}}$ is the time-averaged velocity field and $\left(\overline{\boldsymbol{u}'\boldsymbol{u}'}\right)$ is the Reynolds shear stress term. Dirichlet boundary conditions (*DBC*) are typically specified at a point upstream of the object of interest, where irrotationality can be safely assumed.

The pressure Poisson equation is obtained by applying the divergence operator to the above, yielding

$$\nabla^2\overline{p} = -\rho\nabla\cdot\left(\overline{\boldsymbol{u}}\cdot\nabla\right)\overline{\boldsymbol{u}} - \rho\nabla\cdot\nabla\cdot\left(\overline{\boldsymbol{u}'\boldsymbol{u}'}\right). \tag{2}$$

Treatment of the latter requires mixed boundary conditions. This formulation returns a global solution at a relatively low computational effort, with the useful property of providing a unique solution that minimizes jitters due to measurement noise (Li *et al* 2025).

Specifying boundary conditions along a complex curved geometry is non-trivial and the method's global nature has been shown to suffer from corruption of the entire pressure field reconstruction due to inaccurate values assigned at the object boundary. Rigorous error estimation studies by Charonko *et al* (2010) and Nie *et al* (2022) have assessed the correlation between velocity measurement error and the reconstruction error of *PPE*.

Conversely, *DI* techniques are less prone to the propagation of errors originating from noisy data at fluid-solid interfaces. Nevertheless, the *DI* approach remains sensitive to spatial error propagation, associated with the pressure gradient integration algorithm. The numerical problem is intrinsically ill-posed, due to the necessity of the pressure field to satisfy path independence of the integrated pressure.

Measurement noise or errors in the velocity measurements can lead to a considerably different pressure reconstruction when following different integration paths. The latter are determined by the choice of the boundary conditions, typically specified by the user.

Baur (1999) sought to mitigate these effects by locally enforcing consistency between integration paths within a space-marching framework, achieved by averaging pressure values integrated from neighboring nodes within a $3 \times 3$ kernel. However, the method remained inherently directional, promoting the accumulation and propagation of local measurement errors along extended integration paths and producing results that depend strongly on the choice of starting nodes.

An important advancement was introduced by Liu and Katz (2006), who developed an iterative omnidirectional (direct) integration (*ODI*) method, whereby the integration paths originate from a virtual, circular boundary placed at an arbitrary distance outside the measurement domain. By varying the starting points along this boundary and performing a weighted average of the resulting integrations, the path dependence is reduced under the assumption of zero-mean random errors. Liu *et al* (2016) later extended this concept by effectively placing the virtual boundary at infinity, leading to the concept of the rotating parallel rays omnidirectional integration (*RPR-ODI*) algorithm. The parallel integration paths represent an elegant solution to the problem of variable weights used in the averaging process previously dependent on the shape of the virtual boundary. The latter approach, however, requires large computational resources compared to *PPE* and scales unfavorably with the domain size. Attempts to apply the approach to 3D data are not reported in the literature. A recent advancement is the development of matrix-based omnidirectional solvers (Zigunov and Charonko 2024) that reformulate the *ODI* as a single matrix inversion problem, allowing the efficient, single-step solution of pressure on large 2D or 3D domains. A later investigation by Pryce *et al* (2024) shows that the solution to the *RPR-ODI* is numerically equivalent to a least-squares solution to *PPE* with Neumann boundary conditions.

Error quantification and uncertainty analysis are imperative for achieving reliable pressure reconstructions from PIV. In this regard, Charonko *et al* (2010) explicitly quantified the impact of velocity measurement errors on reconstructed pressure accuracy for both *DI* and *PPE* by introducing controlled levels of synthetic noise into benchmark datasets. Their results demonstrated that even a few percent of random velocity error can render instantaneous pressure estimates unreliable, unless tailored filtering or denoising is applied. Their study also concluded that the errors appear in the pressure field irrespective of the chosen reconstruction algorithm. More recently, Li *et al* (2025) derived theoretical upper bounds for pressure errors and proposed a Helmholtz–Hodge decomposition framework to impose curl-free and divergence-free corrections on velocity fields and pressure gradients prior to reconstruction. Similarly, Nie *et al* (2022) developed analytical expressions linking pressure errors to spatial resolution, velocity uncertainty, and characteristic flow scales, thereby establishing fundamental criteria for minimizing reconstruction errors.

Despite these advances, there remains no unified consensus on which methodology yields the most reliable results and assessments based on experimental conditions are more the exception than the rule. Factors such as the geometrical configuration and flow topology, measurement resolution, noise and the choice of boundary conditions strongly affect the performance of each approach, suggesting that

the optimal reconstruction strategy must be carefully tailored to the characteristics of the specific PIV dataset and flow regime under investigation.

Given that error propagation only occurs when integrating the pressure gradient, minimizing the extent of the domain of integration is generally accepted as a procedure to reduce errors. To this end, domain partitioning (*DP*) has been proposed as an approach to mitigate the sensitivity of the pressure reconstruction to user-specified boundary conditions. The core idea of the dual model proposed by Jux *et al* (2019) is to perform a *DP* which separates the measurement domain into two regions, based on the governing flow physics: (*IRR*) the irrotational flow domain (such as free-stream flow and external regions of potential flow), where an algebraic pressure-velocity relation (e.g. Bernoulli equation) can be hypothesized; (*ROT*) the rotational subdomain (viscous, often turbulent regions and separated flow) where pressure gradient integration is required. This approach largely simplifies the pressure computation in each partition. Such a technique relies on estimating the *IRR-ROT* boundary. Jux *et al* (2020) identified the total pressure coefficient $C_{\mathrm{P0}}$ as a discriminant, defined as

$$C_{\mathrm{P0}} = \frac{\bar{p}_0 - p_\infty}{q_\infty} = \frac{(\bar{p} - p_\infty) + \frac{1}{2}\rho|\bar{\boldsymbol{u}}|^2}{\frac{1}{2}\rho {u_\infty}^2}, \tag{3}$$

where $\bar{p}_0$ is the local value of the total pressure, $p_\infty$ is the free-stream static pressure, $q_\infty$ is the free-stream dynamic pressure, $u_\infty$ is the free-stream velocity and $\overline{\boldsymbol{u}}$ is the local, time-averaged velocity. In the *IRR* region, it is assumed that Bernoulli's principle holds true. Thus, pressure is obtained algebraically from the PIV-measured velocity via the steady Bernoulli's equation for time-averaged flows:

$$\bar{p} = \overline{p_0} - \frac{1}{2}\rho|\overline{u}|^2. \tag{4}$$

The inner edge of this partition provides the required *IRR-ROT* boundary, where *DBC* can be imposed for the subsequent gradient integration across the *ROT* region.

Therefore, the criterion to identify the irrotational flow region for *DP* reads as

$$C_{\mathrm{P0}} \geqslant 1 - \varepsilon, \tag{5}$$

where $\varepsilon$ is a positive, small ($O\ (10^{-2})$) quantity that accounts for uncertainties in determining the value of $C_{\mathrm{P0}}$ due to small errors in the velocity measurement.

Jux *et al* (2020) showed that by partitioning the domain, integration paths can be made significantly shorter, in turn mitigating error accumulation. Moreover, the integration from the inner edge of the *IRR-ROT* boundary towards the object surface reduces outward propagation of larger measurement errors more frequently incurred at the fluid-solid interface. The above-described method is referred to as single-step *DP* (*SSDP*) for simplicity. In this form, the method still relies upon performing direct pressure gradient integration throughout the whole measurement domain to estimate the total pressure coefficient, rendering the *DP* process susceptible (arguably to a lesser extent compared to *DI*) to the same error propagation it aims to mitigate. The method also introduces a second tuning parameter $\varepsilon$, pertaining to the total pressure coefficient threshold. This parameter needs to be chosen based upon an analysis of experimental uncertainties in the determination of $C_{\mathrm{P0}}$. In the *SSDP* method, such a threshold directly affects the estimation of the *IRR-ROT* boundary and, in turn, the accuracy of the pressure reconstruction.

The present work aims to generalize the *DP* technique by proposing a method that eliminates the dependence upon the choice of the initial boundary condition for the *IRR-ROT* boundary. The technique performs iterative *DP* (*IDP*) and addresses the above limitations by iteratively refining the estimate of the *IRR-ROT* boundary. Furthermore, *IDP* is conceived in a way that significantly reduces the effect of the threshold parameter $\varepsilon$ making use of a boundary regularization technique that permits retroactive error correction in the classification of nodes as either part of the *IRR* or the *ROT* domains.

The article first introduces the foundational aspects of the method (section 2), followed by a description of the experimental setups and procedures (section 3) and by *IDP*'s application to particle tracking velocimetry (*PTV*) measurements of a quasi-2D flow field around an airfoil to illustrate the principle and to compare the results with the state-of-the-art *SSDP* technique (section 4). The uncertainty of the proposed method is then assessed *a posteriori* using Monte Carlo analysis (section 5). Finally, the approach is generalized to complex 3D flows (section 6), where the surface pressure distribution of a turbulent flow around a wall-mounted cube is reconstructed.

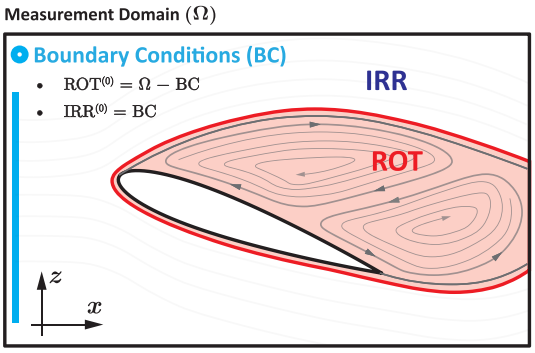


**Figure 1.** Initialization of the IDP algorithm. Regions labeled as IRR and ROT represent the partitions of irrotational and rotational flow respectively, that are to be estimated.

## 2. Working principle of IDP

The *IDP* method operates by repeatedly alternating between two steps: pressure reconstruction and flow-region classification. The initial iteration of *IDP* is equivalent to the *SSDP* algorithm. A preliminary estimate of $C_{P0}$ is obtained through pressure-gradient integration across the entire measurement domain, starting from user-defined boundary conditions. Flow regions in which $C_{P0}$ exceeds the prescribed threshold in equation (5) are identified as ***IRR*** and excluded from the integration in the following iterations. The inner edge of the *IRR-ROT* partitioning obtained during iteration $n$ is subsequently used as *DBC* for iteration $n+1$. As a result, the pressure integration that follows is restricted to a progressively smaller ***ROT*** region, which shortens the integration paths, potentially leading to a more accurate $C_{P0}$ estimate for further reclassification. This alternating sequence of integration and reclassification continues until the relative change in the reconstructed pressure field between successive iterations falls below a prescribed convergence threshold, indicating that both the flow-region partition and the corresponding pressure field have stabilized.

### 2.1. Zeroth-order classification

The algorithm is initialized by assuming no prior knowledge of the flow topology within the measurement domain $\Omega$. All interior nodes are therefore classified as *ROT*, while only the set of user-defined boundary condition nodes *BC* (blue segment or point in figure 1), typically selected in the outer edge, upstream region of $\Omega$, is considered *IRR* (figure 1):

$$ROT^{(0)} = \Omega - BC, IRR^{(0)} = BC, \tag{6}$$

### 2.2. Spatial integration

The pressure distribution is initially computed using a spatial propagation algorithm, as described by van Oudheusden (2008). For any grid node $\boldsymbol{x} = (x, y, z)$, let $N(\boldsymbol{x})$ denote its stencil (typically a $3 \times 3 \times 3$ neighborhood) and define the gradient integration kernel as

$$K^{(n)}(\boldsymbol{x}) = \{\boldsymbol{y} \in N(\boldsymbol{x}) | p^{(n)}(\boldsymbol{y}) \text{ is known}\} \tag{7}$$

Figure 2 shows a schematic illustration of neighboring nodes and gradient integration kernel. At each integration step, all the points $\boldsymbol{x} \in ROT^{(n)}$ with the largest number of known neighbors are selected, i.e.

$$\left|K^{(n)}(\boldsymbol{x})\right| = \max_{q \in ROT^{(n)}} \left|K^{(n)}(\boldsymbol{q})\right|, \tag{8}$$

where, $q$ is any point inside the $ROT^{(n)}$ region. The pressure value is updated as

$$p^{(n)}(\boldsymbol{x}) = \frac{1}{\left|K^{(n)}(\boldsymbol{x})\right|} \sum_{y \in K^{(n)}(\boldsymbol{x})} \left( p^{(n)}(\boldsymbol{y}) + \int_{y}^{x} \nabla \bar{p}^{(n)} \cdot d\boldsymbol{s} \right), \tag{9}$$

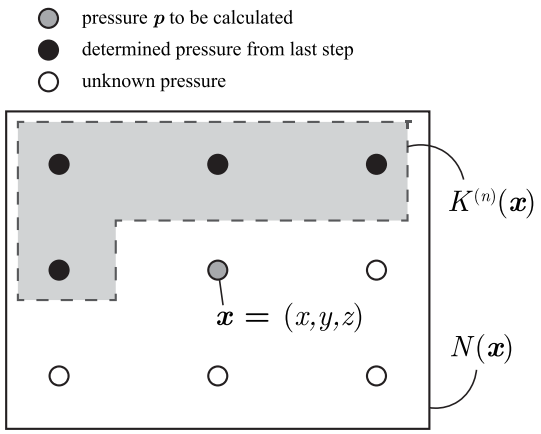


**Figure 2.** Schematic illustration of nodes used for spatial integration as described through equations (7)–(9).

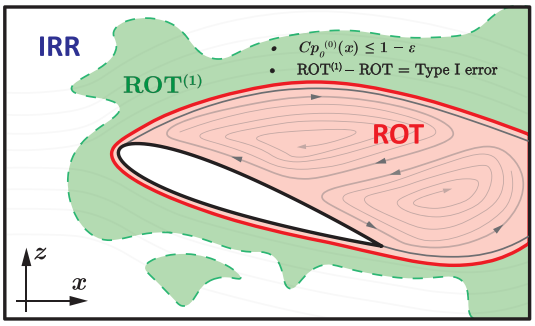


**Figure 3.** The first estimation of domain partitioning starting from DBC specified upstream of the body of interest typically yields a conservative estimate of the rotational region.

where $\nabla\bar{p}^{(n)}$ is calculated using equation (1). This approach enables the propagation of a connected line in 2D or surface in 3D. The process is analogous to that of a *solidification front*, as opposed to *percolation* through a medium, as is the case for methods such as the *ODI* where communication between integration paths is absent.

### 2.3. Domain partitioning

At the end of each spatial integration, nodes are classified based on the value of $C_{P0}^{(n)}$. Thus, the domain is split into regions $IRR^{(\mathrm{n}+1)}$ and $ROT^{(\mathrm{n}+1)}$ according to

$$\begin{cases} IRR^{(\mathrm{n}+1)} = \{\boldsymbol{x} \in \Omega | {C_{P0}}^{(n)}(\boldsymbol{x}) \geqslant 1 - \varepsilon\} \\ ROT^{(\mathrm{n}+1)} = \Omega - IRR^{(\mathrm{n}+1)} \end{cases}. \tag{10}$$

Within this framework, *SSDP* corresponds to the first iteration of *IDP*, providing the first estimation of the *domain partitioning*. The pressure value at nodes that lie inside the updated $IRR^{(\mathrm{n}+1)}$ region is overwritten using Bernoulli's equation, whereas pressure values for nodes inside $ROT^{(n+1)}$ remain to be updated in the next spatial integration starting from the newly updated *DBC* at the *IRR-ROT* interface. Figure 3 is a pictorial representation of the result from the first pass estimation of *DP*. Although the *ROT* region is roughly captured in the method, a first estimation is affected by initial large errors committed in $C_{\mathrm{p0}}$ estimation because of the pressure integration across the whole measurement domain. As a result, the conservative approach initially overestimates the extent of *ROT* to be eroded later with subsequent iterations.

### 2.4. Iterative IRR-ROT estimation

Redefining the boundary iteratively offers the advantage of minimizing the impact of local velocity measurement errors on the final global pressure field. At each iteration, the classification of points near the current $IRR - ROT^{(n)}$ boundary based on the $C_{\mathrm{P0}}$ criterion is strictly dependent on its immediate neighbors, such that the $IRR^{(0)}$ region is dilated from any initial *DBC* in a controlled manner.

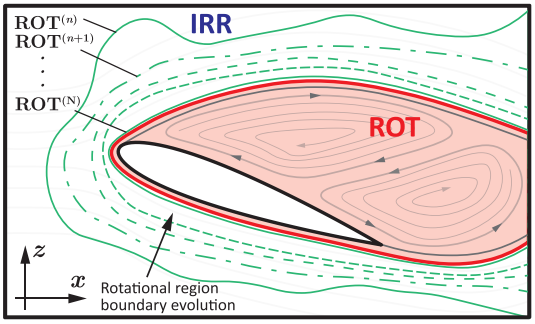


**Figure 4.** Illustration of IRR-ROT boundary evolution across iterations, progressively eroding the estimated ROT region and correcting type-I errors.

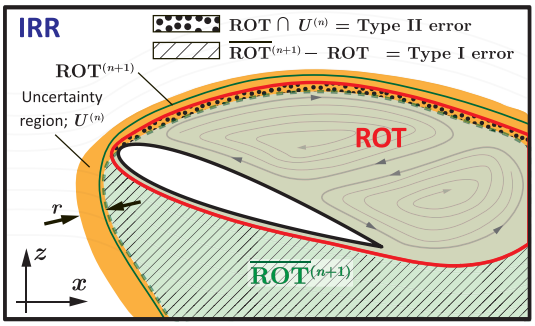


**Figure 5.** Uncertainty region (orange) built at the end of iteration n by dilation of ROT boundary. Re-evaluation of pressure values in $\boldsymbol{U}^{(n)}$ prevents propagation of type-II errors.

In the context of *IDP*, two types of classification errors may be committed for a given node $\boldsymbol{x}$ during iteration $n$. Type-I errors consist of incorrectly ascribing node $x$ to $ROT^{(n+1)}$, even though it belongs to *IRR* (false $\overline{ROT}$ positive). In the 0th order classification, all nodes (except those pertaining to the boundary conditions) are initially ascribed to $ROT^{(0)}$. This effectively initializes the algorithm with a set of nodes that constitute type-I errors. The latter are consistently corrected within the iterative procedure that progressively dilates *IRR* towards the actual (physical) boundary of *ROT*. Attention is given to the nodes placed at the *IRR-ROT* interface. These tend to be located closer to the updated boundary in each subsequent integration run, as illustrated in figure 4. Such nodes are increasingly more likely to be correctly reclassified into *IRR* in subsequent steps. This progressive correction mechanism mitigates the risk of error accumulation. Once the partition converges and the boundary stops evolving, there are no more detectable type-I errors.

A type-II error occurs when node $\boldsymbol{x}$ is incorrectly classified as belonging to the $IRR^{(n+1)}$ region during iteration $n$, even though it should be assigned to *ROT* (false *IRR* positive). Once a node is updated according to Bernoulli's equation, its value remains unchanged during later iterations if no additional correction mechanism is adopted. If such a misclassified node is not corrected promptly, the erroneous Bernoulli value may later act as a boundary condition and transmit errors into the surrounding *ROT* region. These errors tend to occur at the *IRR-ROT* boundary, and the risk is mitigated by introducing an uncertainty (buffer) region (due to the higher uncertainty of $C_{p0}$) at such a boundary. After each spatial integration step, the newly estimated region $ROT^{(n+1)}$ is subject to dilation with radius $r$ (in grid units):

$$U^{(n+1)} = \text{dilate}\left(\overline{ROT}^{(n+1)}, r\right) - \overline{ROT}^{(n+1)}. \tag{11}$$

A value of $r = 1$–2 grid units was found sufficient to balance error correction and numerical stability. Nodes within $U^{(n+1)}$ are temporarily reclassified as *ROT* and subsequently re-evaluated using updated neighbor information as shown in figure 5. This additional verification step increases the likelihood of correcting misclassified nodes, thereby reducing the persistence of type-II errors.

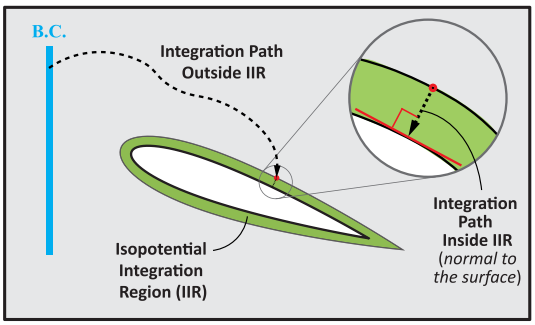


**Figure 6.** Schematic of isopotential integration in the vicinity of the airfoil surface.

### 2.5. Isopotential integration

Prioritizing candidates with the maximum number of available neighbors for spatial pressure-gradient integration (equation (8)) promotes local consistency between adjacent integration paths. Here the assumption is made that velocity measurement errors are uniformly and randomly distributed in space. This assumption breaks down in the vicinity of solid boundaries, where lack of spatial resolution, optical reflections, occlusions and reduced signal from particle tracers often corrupt the measurement, introducing larger velocity measurement errors. If not properly addressed, such localized errors can propagate through the integration process affecting the reconstructed pressure field to a large extent. An approach is devised here that aims to minimize the length of the integration path along the direction tangent to the object.

A dilated contour of the solid object bounds a region masked (i.e. excluded) by the iterative integration process as illustrated in figure 6. When aiming at pressure reconstruction down to the object surface, the pressure gradient integration is performed by enforcing pressure updates preferentially along local isopotential directions in near-wall regions. The approach exploits the physical observation that, in the immediate vicinity of an airfoil surface, streamlines tend to align with the surface geometry, such that isopotential lines naturally terminate normal to the surface. The contribution of each node $\boldsymbol{y}$ to the integration at node $x$ is weighted according to

$$w(\boldsymbol{x},\boldsymbol{y}) = \left| \frac{\vec{v}(\boldsymbol{x})}{\vec{v}(\boldsymbol{x})} \cdot \frac{\vec{r}_{\boldsymbol{xy}}}{\vec{r}_{\boldsymbol{xy}}} \right| \tag{12}$$

where, $w^{(n)}(\boldsymbol{x},\boldsymbol{y})$ is the assigned weight, $\vec{v}(\boldsymbol{x})$ is the velocity vector at the current node and $\vec{r}_{\boldsymbol{xy}}$ is the vector pointing from node $\boldsymbol{y}$ in the stencil to node $\boldsymbol{x}$. Thus, the expression for the pressure at node $\boldsymbol{x}$ with integration kernel $K(\boldsymbol{x})$ is

$$p^{(n)}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{y}\in K(\boldsymbol{x})} w(\boldsymbol{x},\boldsymbol{y}) \left[ p(\boldsymbol{y}) + \int_y^x \nabla \bar{p} \cdot d\boldsymbol{s} \right]}{\sum_{\boldsymbol{y}\in K(\boldsymbol{x})} w(\boldsymbol{x},\boldsymbol{y})}. \tag{13}$$

This weighting strategy suppresses error accumulation associated with unreliable near-wall velocity data, thereby limiting integration drift in both attached and separated flow regimes. Evidence of such errors is given in section 5.

### 2.6. Surface pressure reconstruction

In practice, the velocity field is rarely available up to the exact fluid–solid interface. This is mainly due to the absence of tracer particles in the immediate vicinity of the surface, optical reflections, masking requirements, limited spatial resolution, and possible occlusions. Consequently, the pressure reconstructed within the fluid domain must be extrapolated towards the solid surface to obtain surface pressure distributions and aerodynamic loads. This extrapolation is particularly sensitive to near-wall velocity errors and to the distance between the closest valid PIV vectors and the surface. Accurate reconstruction of surface pressure from PIV data is a long-standing challenge in experimental fluid mechanics (Ragni *et al* 2009, Tagliabue *et al* 2017, Jux *et al* 2020, Cakir *et al* 2024) particularly in regions with strong pressure gradients, such as near stagnation points, suction peaks, and in the presence of flow separation or unsteady effects (Ragni *et al* 2009, Tagliabue *et al* 2017).

A further challenge arises from the mismatch between the fluid-domain Cartesian mesh and the surface mesh at the fluid–solid interface. The problem has been addressed in 3D experiments by Jux *et al*

(2020), who proposed pressure-gradient integration from a 3 × 3 × 3 stencil centered on the nearest Cartesian grid point to each surface node.

Ragni *et al* (2009) treated 2D data locally linearly extrapolating the reconstructed pressure field along lines normal to the surface of the body. The latter approach is adopted here because of its simplicity and robustness. Although this extrapolation strategy may be less effective in regions with sharp pressure gradients, such as the suction peak, it provides a consistent and efficient procedure for comparing different pressure-estimation methods under the same surface-pressure reconstruction framework.

### 2.7. Process convergence

As will be shown in section 4.1, the *IDP* procedure converges once successive iterations no longer produce a displacement of the *IRR-ROT* boundary. In practice, the *ROT* region rapidly shrinks during the first few iterations, and only a small fraction of nodes near the evolving interface continue to switch classification. Beyond this point, the Dirichlet boundary for the integration step remains virtually unchanged, and the pressure reconstruction may be considered converged.

This condition is assessed by two complementary criteria. Pressure-field convergence is monitored through a relative $L^2$—norm:

$$\frac{\|C_{\mathrm{p}}^{(n+1)} - C_{\mathrm{p}}^{(n)}\|_2}{\|C_{\mathrm{p}}^{(n)}\|_2} < \delta_p, \tag{14}$$

where $\delta$ is a user-prescribed tolerance set to $10^{-3}$. Classification convergence is checked using the fraction of nodes that change state between iterations:

$$\frac{\left|IRR^{(n+1)} \Delta IRR^{(n)}\right|}{|\Omega|} < \delta_c, \tag{15}$$

where $\Delta$ denotes the symmetric difference between sets (i.e. all nodes whose classification flips). This criterion ensures that the *IRR-ROT* partition has stabilized and prevents termination in cases where the pressure update alone is small, but the interface continues its evolution. However, the classification convergence criterion alone is insufficient to assess convergence in cases where pressure fluctuations are strongly influenced by type-II errors. An example of such a situation is discussed in section 4.1. The algorithm terminates when both conditions are satisfied (logical AND condition). A flow-chart summarizes the *IDP* process in figure 7.

To facilitate reproducibility, a minimal MATLAB code of the IDP method is included in appendix B. This sample code together with a sample dataset is provided via GitHub repository being accessible using this link; https://github.com/adontu29/iterative-domain-partitioning-pressure-piv. The code annotations further indicate the correspondence between specific code sections and blocks 1–6 of flowchart of figure 7.

## 3. Description of experiments

Datasets from two experiments are employed in this study, obtained from time-averaged 3D-PTV measurements conducted on airfoil and canonical bluff-body configurations. These datasets provide well-characterized benchmark flows with varying degrees of complexity, enabling a systematic evaluation of the proposed *IDP* method.

### 3.1. Flow around NACA0015 airfoil (2D)

The first dataset relates to experiments performed in the open-jet facility of TU Delft Aerodynamics Laboratories. The wind tunnel features an exit cross section of 2.85 × 2.85 m$^2$ and a NACA0015 airfoil of chord length $c = 20$ cm and 1 m span is immersed in a stream at 15 m s$^{-1}$. The corresponding Reynolds number based on the airfoil chord is $Re_c = 2 \times 10^5$. The incidence angle is varied with a digitally controlled rotary stage. The cases with $\alpha = \{0°, 10°, 20°\}$ are selected for the present study. Figure 8(a) illustrates the experimental setup. The NACA0015 airfoil is installed vertically with endplates.

Helium filled soap bubbles (HFSB) of 0.4 mm mean diameter were used as tracer particles (Bosbach *et al* 2009). Scarano *et al* (2015) investigated and verified the fidelity of sub-millimeter HFSB tracers for quantitative velocimetry in wind tunnel flows. The seeding generator is a (2 × 1 m$^2$) rake comprising 400 generators delivering a seeded stream tube of 1.5 × 0.7 m$^2$ in the test section. Multiple LED

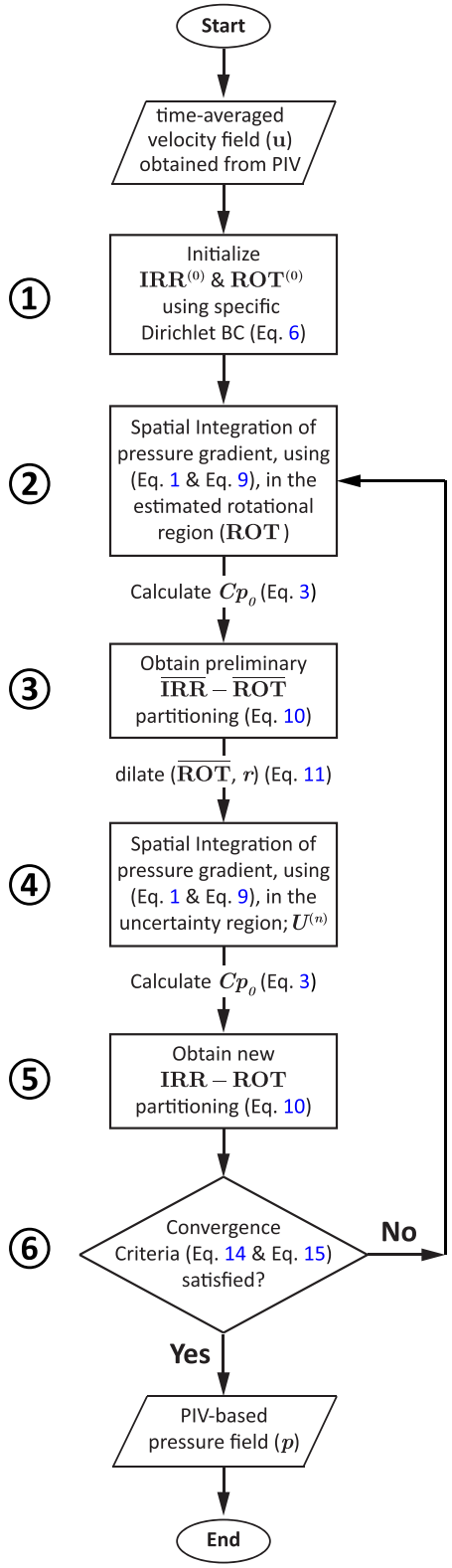


**Figure 7.** Flow chart illustrating the working principle of IDP.

arrays (LaVision *flashlight-300*) provide illumination around the airfoil. 3D imaging is obtained with an Omnidirectional PIV system (Hendriksen *et al* 2024) consisting of 10 CCD cameras assembled onto 2 antennas with 5 cameras, each observing the airfoil from its opposite sides. The CCD cameras had a sensor resolution of 2 megapixels and were operated in double-frame mode at an acquisition rate of 8 Hz.

The resulting measurement domain is $50 \times 25 \times 70$ cm$^3$. Sets of 1000 recordings are accumulated to ensemble average the velocity field from sparse data onto a Cartesian grid with cubic bins of side length $l = 10.8$ mm ($l/c = 0.0540$). Spatial overlap between neighboring bins of 75% yields a grid of velocity

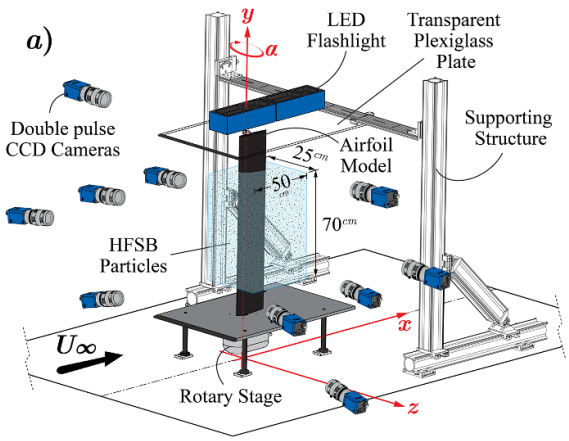


**Figure 8.** Experimental setup for (a) NACA0015 airfoil and (b) wall-mounted cube.

vectors spaced by $h = 2.7$ mm ($h/c = 0.0135$). Finally, a slice of the 3D data with 10 data points along the span is averaged, yielding the 2D distribution of time-averaged velocity field and its fluctuations for evaluation in equations (1) and (2).

PIV measurements are complemented with surface pressure measurements providing an experimental ground truth for the assessment of pressure-from-PIV measurement accuracy. A set of 64 pressure taps is placed along the mid-section of the airfoil. The pressure from each tap is read with a differential pressure transducer with maximum error of 0.1% of the full range.

### 3.2. Cube on a flat plate (3D)

The second dataset relates to the flow around a cube on a flat plate at zero incidence. Details of the latter have been described in the work by Hendriksen *et al* (2024). A schematic description of the experiment is given in figure 8(b). The cube of height $H = 12$ cm is installed on a flat plate and immersed in a flow at 10 m s$^{-1}$. The corresponding Reynolds number based on cube height is $Re_H = 0.8 \times 10^5$. Experiments are performed in the W-tunnel ($0.6 \times 0.6$ m$^2$ cross section, open jet) at the TU Delft Aerodynamics Laboratories. A seeding rake of 200 generators delivers HFSB of 0.4 mm diameter. Two LED arrays provide illumination around the object and high-speed imaging is provided by a surrounding set of 7 CMOS cameras (1 megapixel at 3000 fps). The system covers a domain of approximately $40 \times 40 \times 30$ cm$^3$. Data binning yields velocity vectors with mesh spacing of 3 mm. The purpose of this dataset is to demonstrate the generalization of the proposed method to geometrically nontrivial 3D problems.

## 4. Experimental verification—2D airfoil

The cross-sectional velocity distribution is illustrated in figure 9 at three selected values of the incidence angle. The flow is largely irrotational in the symmetrical condition ($\alpha = 0°$), with only a small velocity deficit in the thin wake (figure 9(a)). At $\alpha = 10°$, the flow pattern exhibits a minor separation as evident from a slight wake offset towards the suction side (figure 9(b)). Leading-edge separation characterizes the stalled condition at $\alpha = 20°$ (figure 9(c)).

Pressure reconstruction is compared using three approaches, *DI*, *SSDP* and *IDP*, with the results shown in figure 10. The pressure distribution based on Bernoulli's equation ($\Omega = IRR$) is highlighted with dashed contours. Furthermore, three different choices for initial *DBC* are considered: a vertical segment upstream of the airfoil (left), a horizontal segment above the airfoil (middle) and a single point in the top-left corner (right). The latter is the most conservative, yet commonly adopted choice.

The pressure field reconstruction based on *DI* (top row) returns the pronounced suction ($C_p \sim -2$) region above the airfoil consistently. Instead, the stagnation blob is poorly captured when the integration path progresses along the leading edge (figure 10(b)). Finally, the pressure distribution at the pressure side is largely distorted when integrating across the wake. In this case, placing the BC on a vertical line

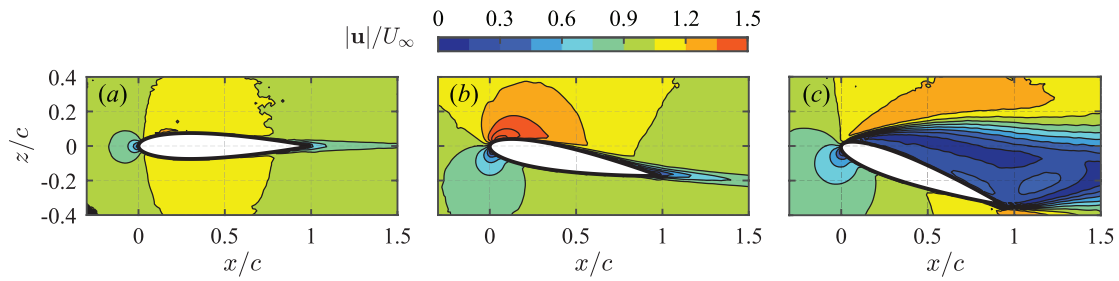


**Figure 9.** Time average velocity field for NACA 0015 at (a) $\alpha = 0°$, (b) $\alpha = 10°$, and (c) $\alpha = 20°$.

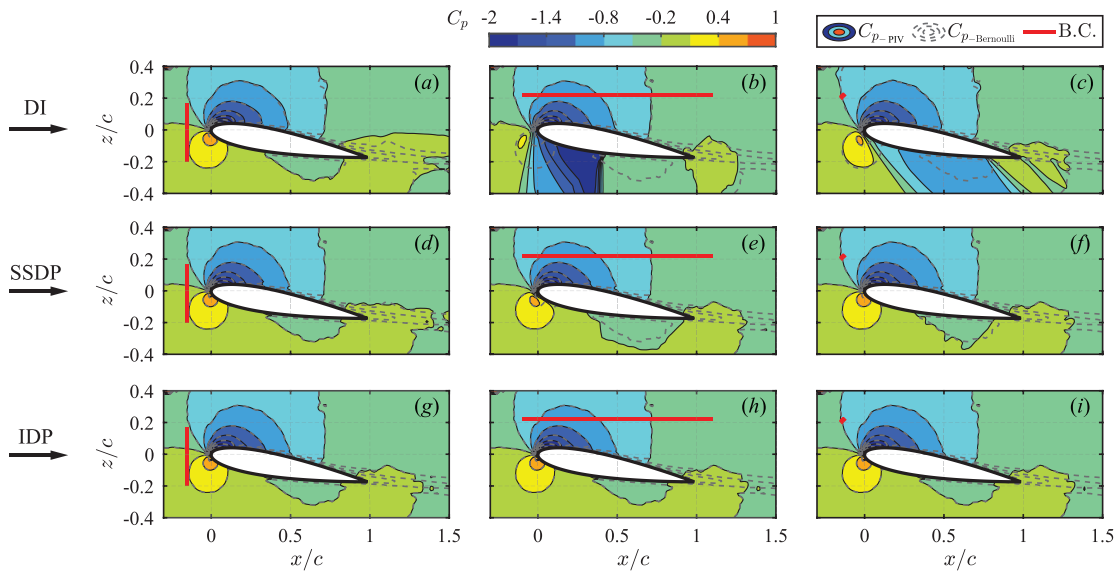


**Figure 10.** Pressure coefficient distribution as reconstructed $C_p$ with DI (top row), SSDP (middle row) and IDP (bottom row), for different DBCs (vertical line, horizontal line, and a single point). Values obtained with Bernoulli equations are given as dashed lines.

at the inlet yields the most consistent result. In contrast, the horizontal line above the airfoil or the single point yield large deviations at the pressure side in the *IRR* region. This is ascribed to the error accumulation when the integration front propagates near the surface of the airfoil, where the velocity measurement is most uncertain. These observations suggest that integration paths terminating at the surface of the body of interest minimize error propagation.

The results from the *SSDP* technique (figure 10, middle row) confirm *domain partitioning* as a suitable strategy for this type of flow. SSDP yields only minor differences in the stagnation region and at the pressure side of the airfoil (figure 10(e)) by the different choice of boundary-condition placement. The partitioned domain corresponding to the result shown in the second row of figure 10 is illustrated in figure A1 of appendix A. Finally, the bottom row illustrates the result obtained with *IDP*. In this case, the *IRR-ROT* boundary is estimated iteratively, and the resulting pressure coefficient distribution appears to be practically unaltered by varying the input *BC* region. The evolution of this boundary is detailed in figure A1 of appendix A, which shows that after only two iterations the *IRR-ROT* boundary becomes independent of the initial choice of *BC*.

### 4.1. Convergence

Across the varying flow patterns and choices of boundary conditions, *IDP* appears to consistently converge to the same final *IRR–ROT* partition within 15 iterations. The slowest convergence is observed for the case where $\alpha = 20°$, which is expected due to the larger *ROT* region. Figure 11 shows the main convergence indicators at $\alpha = 20°$. Figure 11(a) illustrates the evolution of the relative $L^2$ norm of the static pressure coefficient field between successive iterations. Figure 11(b) reports the percentage of grid nodes that are reclassified. Finally, figure 11(c) reports the percentage of nodes falling within the uncertainty band that change classification (type-II error correction). The complementary information provided by the first two metrics highlights why they both must satisfy their respective user-defined tolerances for

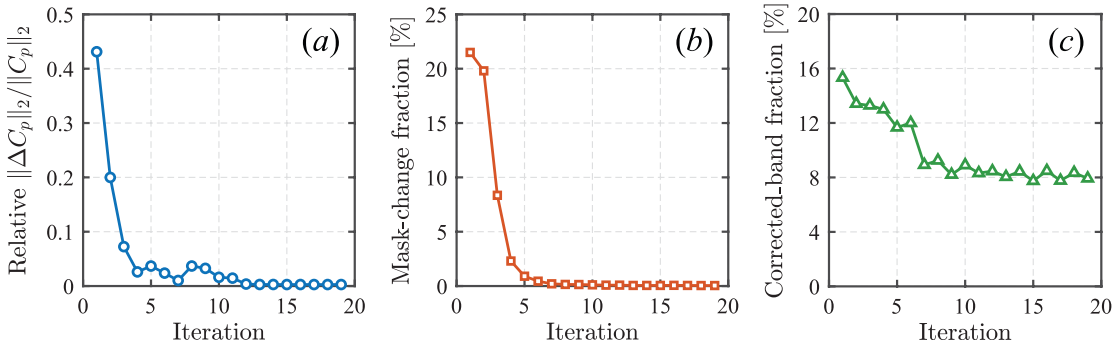


**Figure 11.** Convergence plots by (a) $L^2$ norm of $C_p$ updates between successive iterations, (b) percentage of grid nodes whose IRR-ROT classification changes between iterations, and (c) percentage of nodes inside the uncertainty region that are reclassified at each iteration.

the partitioning to be deemed converged. Five iterations deliver the largest decrease in the $L^2$ norm. This corresponds to the correction of a large amount of type-I errors naturally present during the initialization step when the entire domain, except for the *BC* region, is tentatively classified as *ROT*. It is important to note that approximately the same percentage of nodes are reclassified in the first two iterations, the first one corresponding to the *SSDP* estimation. Beyond iteration five, the magnitude of the $L^2$ norm exhibits a bounded oscillatory behavior, remaining within approximately 0–0.07 $C_p$ before settling to a final value of zero. Figure 11(c) reveals that in this phase, type-I and type-II errors become equally important to address, as the percentage of corrections in the uncertainty band stays relatively constant while the overall per-iteration reclassification displays an asymptotically monotonic decay throughout the entire iteration sequence, as the estimated *IRR-ROT* interface approaches the true physical boundary. In this regime, the algorithm alternates between refinement of the classification and re-integration within the uncertainty region, causing the boundary to oscillate locally around its final position until no further systematic corrections can be identified. Convergence is thus declared only when the relative $L^2$ norm between successive pressure fields and the percentage of flipped nodes both fall below their prescribed thresholds, ensuring the partitioning has stopped evolving meaningfully. Although the iterative method involves additional computations, the increase remains modest. The most computationally intensive part is the gradient integration, here performed exclusively across the *ROT* subdomain which contracts significantly within the first few iterations.

## 5. Errors and uncertainty

The measurement error is first evaluated by comparing the pressure reconstruction with the direct measurements from the pressure taps. It should be kept in mind, however, that pressure estimation at the solid surface is also affected by extrapolation errors as discussed in Ragni *et al* (2009). The discrepancies should therefore be considered an upper bound for the measurement errors.

The uncertainty of the pressure coefficient estimation is studied by Monte Carlo simulation, whereby the position of a single point *DBC* is varied randomly in the *IRR* domain and the corresponding dispersion of the pressure distribution is examined. Figure 12 illustrates the distribution of *DBCs* relative to the velocity field for $\alpha = 10°$.

### 5.1. Comparison to pressure taps

The surface pressure coefficient distributions obtained from the *SSDP* and *IDP* reconstructions are compared with pressure-tap measurements at incidence angles $\alpha = \{0°, 10°, 20°\}$ in figure 13. Surface pressure coefficients are extracted from the reconstructed pressure fields using the method of Ragni *et al* (2009), as described in section 2.6, and are directly compared with the corresponding pressure-tap data. For the *IDP* method, very good agreement with pressure-tap measurements is observed for all incidence angles, except for the suction-side peak at $\alpha = 10°$. In this region, the sharp velocity gradients near the leading edge cannot be fully resolved by the velocimetry data, leading to a slight underprediction of the suction peak. Aside from this discrepancy, the reconstructed pressure distributions closely follow the experimental measurements on both the pressure and suction sides of the airfoil. At $\alpha = 20°$, a sharp gradient in the pressure coefficient appears around $x/c \approx 0.8$ on the suction side. This feature originates

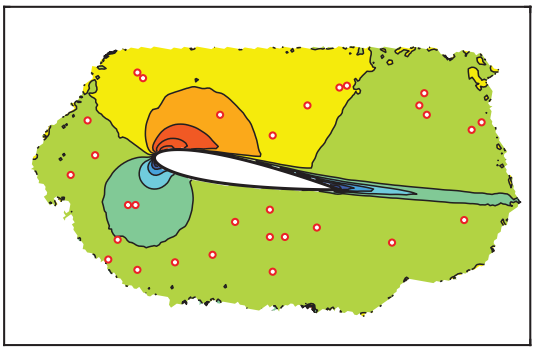

**Figure 12.** Dirichlet boundary condition locations (red circles) for the Monte Carlo simulation.

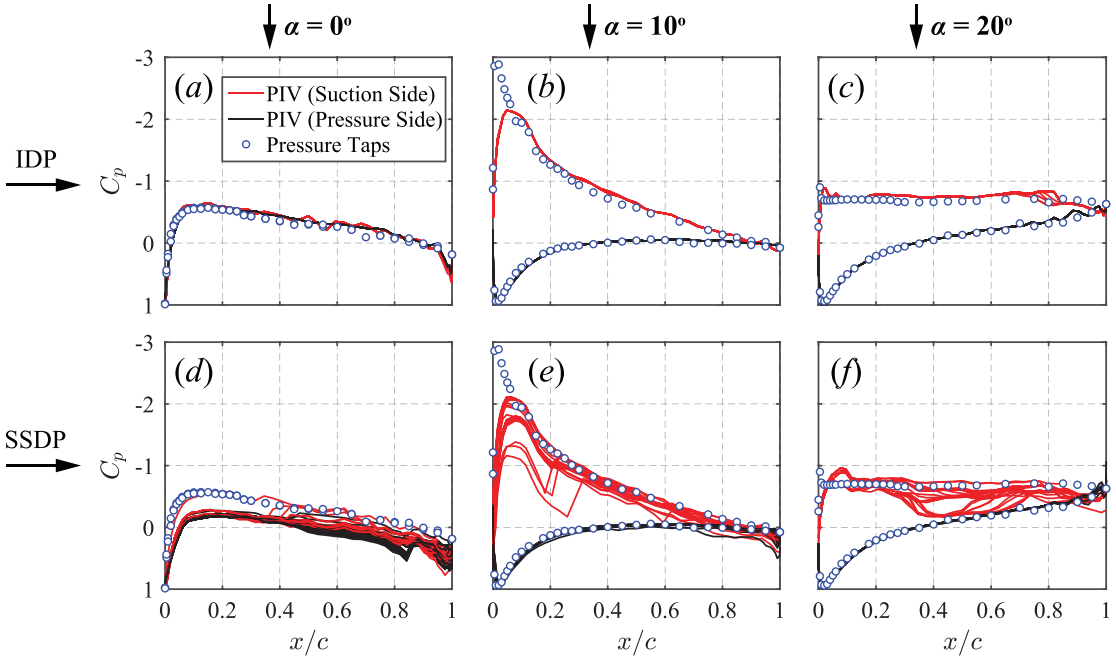


**Figure 13.** Surface pressure coefficient estimation from IDP (top row) and SSDP (bottom row) at $\alpha = \{0°, 10°, 20°\}$. Multiple lines are obtained following the Monte Carlo simulation. Red lines on the suction side and black lines on the pressure side respectively. Pressure taps data represented by hollow-blue circles.

from the intersection of two integration fronts advancing along the airfoil surface and reconciling conflicting pressure estimates at that chordwise location. Despite this localized effect, the overall agreement with pressure-tap measurements remains satisfactory.

### 5.2. Monte Carlo analysis

The position of DBC is chosen randomly at a point in the *IRR* domain. The process is repeated 30 times, as illustrated in figure 12 for the case of $\alpha = 10°$. The results in terms of surface pressure coefficient distribution are shown in figure 13. For the *IDP* method (figure 13 top row), the pressure distribution (red and black lines for the suction and pressure side respectively) overlaps across all 30 reconstructions with a small exception at $\alpha = 20°$ towards the trailing edge. This robustness is primarily attributed to the location and geometry of the *IRR-ROT* interface at $\alpha = \{0°, 10°\}$. When the partitioning closely conforms to the airfoil surface, the integration paths used for surface pressure recovery remain short and approximately normal to the surface, effectively suppressing error propagation during the final integration stage. At $\alpha = 20°$ instead, a wide band of rotational flow is present in the middle of the measurement domain, leading to longer final integration paths and a reduced effectiveness of the *domain partitioning*. While the pressure-side results remain independent of the *DBC* choice, the suction side exhibits increased sensitivity causing the pressure curves to split around $x/c \approx 0.8$ as discussed previously.

In contrast, the *SSDP* results display grouped variations in the pressure coefficient distributions that correlate with the relative placement of the imposed boundary conditions (figure 13 bottom row).

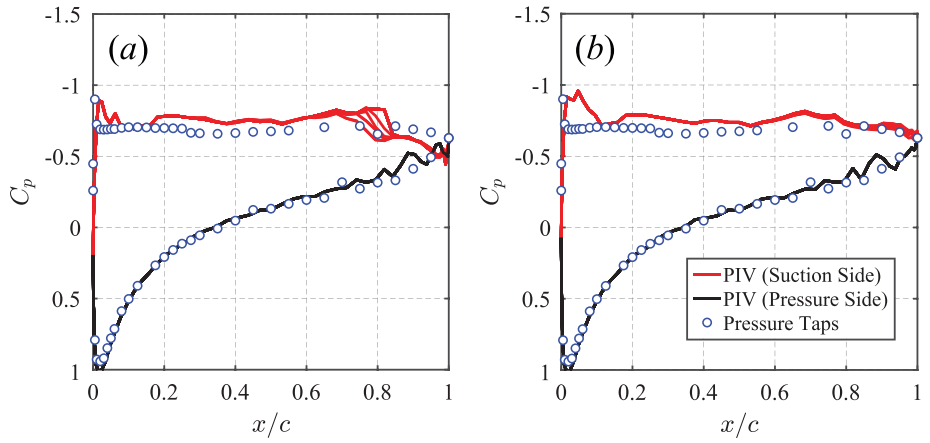


**Figure 14.** Surface pressure coefficient reconstruction at $\alpha = 20°$ with IDP (a) without and (b) with isopotential direction integration.

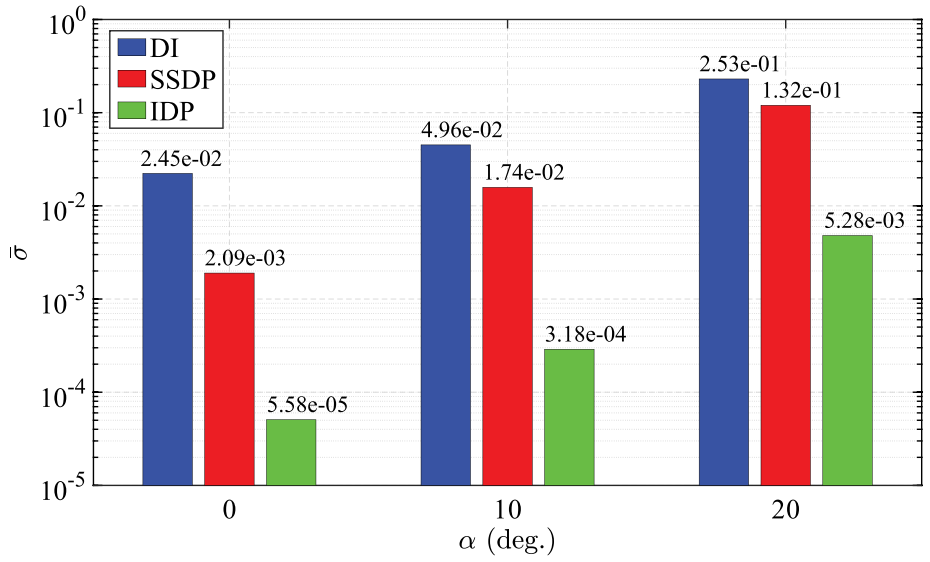


**Figure 15.** Sensitivity of mean standard deviation to changes in initial BC for baseline DI, SSDP and IDP algorithms.

Inaccurate *domain partitioning* leads to asymmetries even under attached flow conditions. Misalignment of the *IRR-ROT* boundary causes one side of the airfoil to be reached significantly earlier by the final marching front, promoting the propagation of errors originating from near-wall regions where velocity measurements are most uncertain.

Figure 14 illustrates the effect of enforcing isopotential integration near the airfoil surface for the *IDP* method at $\alpha = 20°$, which suppresses the primary mechanism of near-wall error propagation by wall parallel integration. The unphysical sharp gradients in the suction-side pressure-coefficient distribution are eliminated, resulting in a pressure distribution independent of the *DBC* position. The reconstructed pressure exhibits a favorable comparison with the pressure-tap measurements and is particularly improved near the trailing edge with the adoption of isopotential direction integration.

### 5.3. Mean standard deviation metric

The sensitivity of the three pressure reconstruction algorithms to user-specified boundary conditions is assessed through a domain-wide quantitative metric. To this end, a pointwise standard deviation field is defined as:

$$\sigma(x,y) = \sqrt{\frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left(Cp_i(x,y) - \overline{Cp}(x,y)\right)^2}. \tag{16}$$

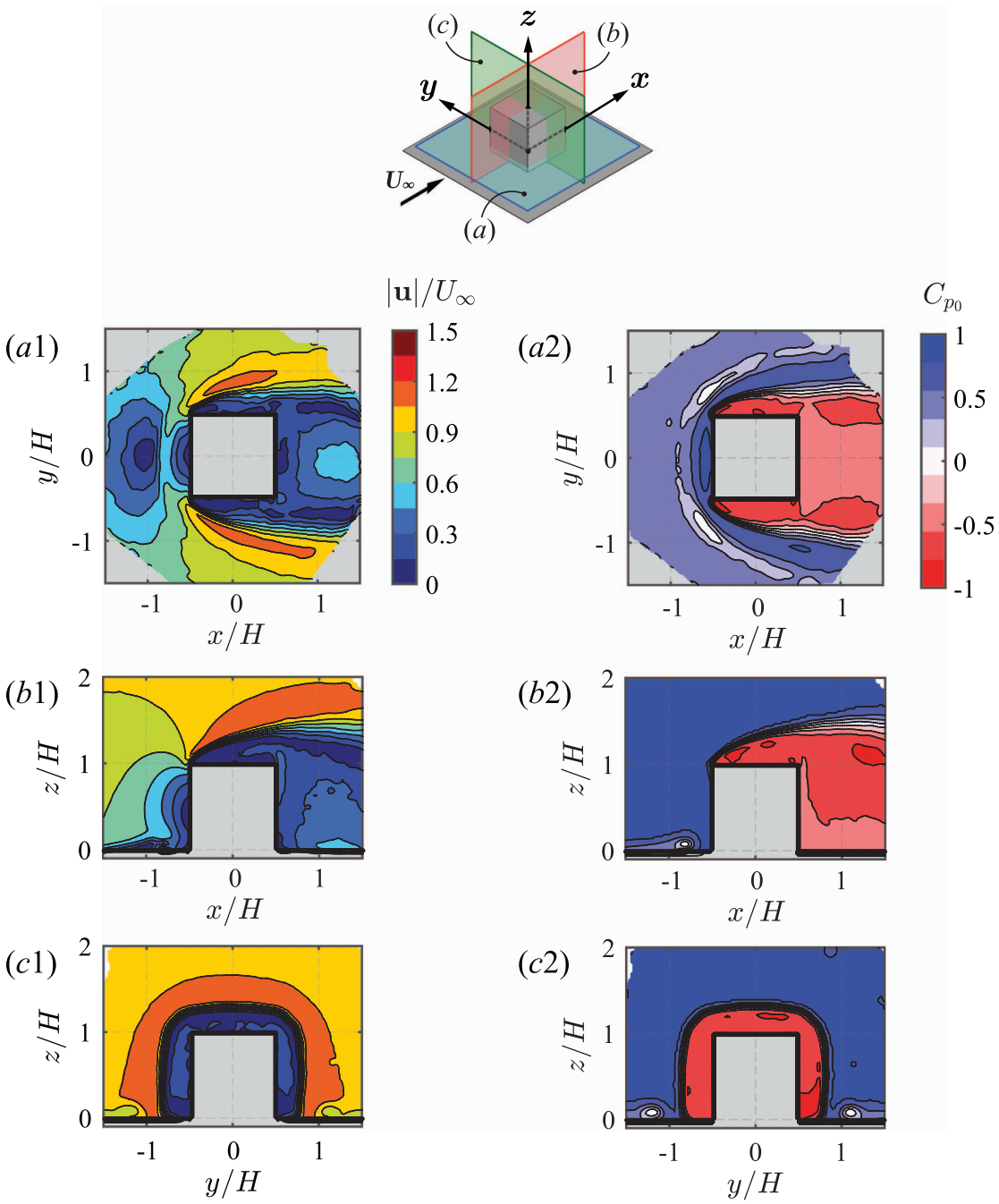


**Figure 16.** Velocity contours intersecting at the cube's midpoint (left column). Corresponding total pressure coefficient contours (right column).

Here $Cp_i$ represents the pressure coefficient spatial distribution resulting from the $i$th choice of boundary condition, whereas $\overline{Cp}$ is the averaged field over all reconstructions $N_{\mathrm{BC}}$. A global scalar indicator $\bar{\sigma}$, defined as the mean standard deviation, is then computed by spatially averaging $\sigma$ over the entire measurement domain. This quantity serves as a comprehensive figure of merit that reflects the sensitivity of each algorithm to boundary condition specification.

Figure 15 reports $\bar{\sigma}$ on a logarithmic scale for each angle of attack, comparing the baseline implementations of the *DI*, *SSDP* and *IDP* methods respectively. For low angles of attack, where the flow remains largely attached, *SSDP* achieves a significant reduction in reconstruction variability relative to *DI*. In this regime, the *IRR* region occupies most of the domain, and a single partitioning step is generally efficient in limiting error accumulation.

As the angle of attack increases and flow separation becomes more pronounced, the performance of *SSDP* degrades. At $\alpha = 20°$ where the *ROT* region makes up approximately half of the available measurement domain, the sensitivity of the pressure reconstruction to inaccuracies in the estimated *IRR-ROT*

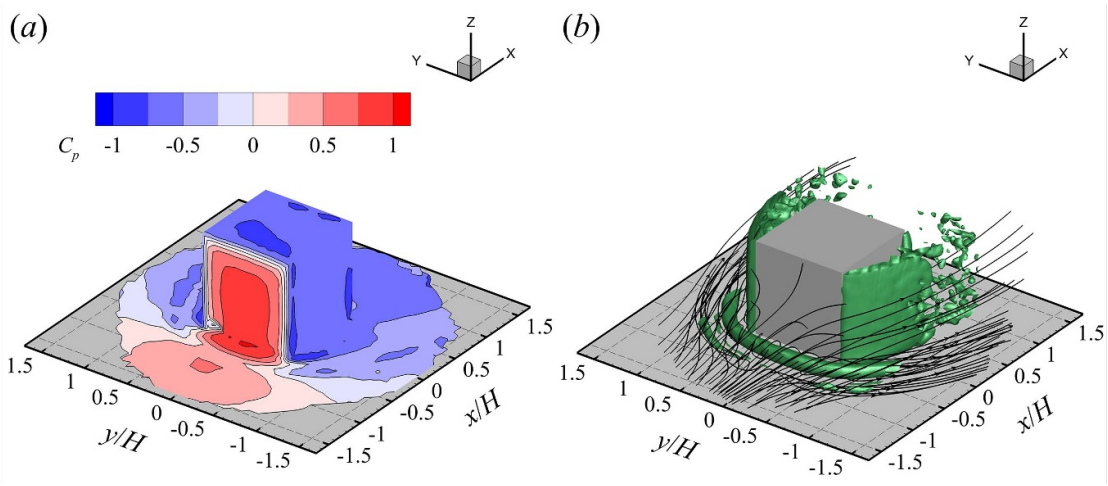


**Figure 17.** (a) Pressure coefficient distribution over the surface of the cube, (b) iso-surfaces of the Q-criterion colored by magnitude, overlaid with time-averaged streamlines, illustrating the dominant vortical structures.

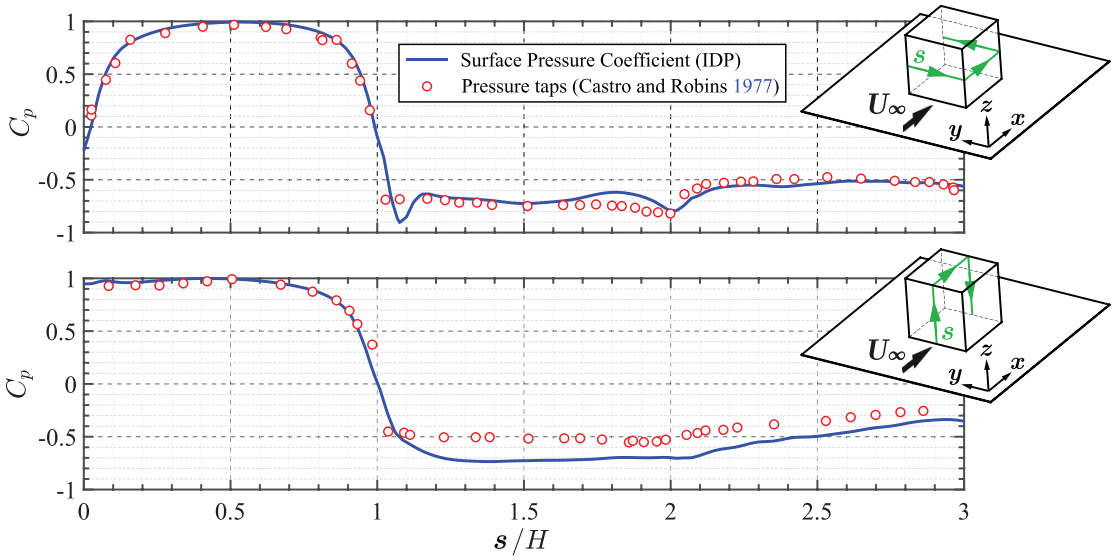


**Figure 18.** Comparison between surface pressure coefficient values obtained with IDP and pressure tap measurements according to (Castro and Robins 1977).

interface is significantly increased. Misclassification during the single partitioning step leads to longer integration paths originating near the airfoil surface, where velocity measurement uncertainty is highest. As a result, *SSDP* exhibits sensitivity levels comparable to those of *DI*, indicating substantial propagation of near-wall integration errors into the outer flow.

In contrast, *IDP* has a substantially lower variability, approximately two orders of magnitude, indicating very low sensitivity to boundary conditions in all flow regimes. At an angle of attack of 20°, where the flow exhibits extensive separation and accurate partitioning becomes critical, *SSDP* shows an order of magnitude increase in sensitivity to boundary conditions compared to the case of $\alpha = 10°$, while the $\bar{\sigma}$ associated with *IDP* remains below 1%.

## 6. Wall-mounted cube measurements (3D)

The rapid convergence of the *IDP* algorithm facilitates its extension to data of 3D flows, where the effective integration domain contracts significantly after the first evaluation of the *IRR-ROT* boundary. The present database is obtained from 3D large-scale, time-resolved Lagrangian particle tracking around a

wall-mounted cube introduced in section 3.2. Figure 16 (left column) presents the time-averaged contours of the streamwise velocity across three planes intersecting at the cube's mid-height.

The velocity distribution reproduces the main topology and findings of experimental studies in the literature at comparable Reynolds numbers (Castro and Robins 1977, Martinuzzi and Tropea 1993). The IDP method is applied to this dataset, and the reconstructed spatial distribution of the total pressure coefficient is shown in figure 16 (right column). The first iteration is initialized with a single point upstream of the cube at the mid-plane (plane (*b*) shown in the schematic of figure 16) of the measurement domain $(x/H, y/H, z/H) = (-1.5, 0, 1.5)$. The incoming boundary layer approaching the front face of the cube is subject to an adverse pressure gradient and separates forming the horseshoe vortex that propagates along the sides of the cube.

Figure 17(a) provides a 3D overview of pressure coefficient reconstruction along the surface of the cube. The development of the vortical structures is presented in figure 17(b) and confirmed by the drop in total pressure coefficient as given by *IDP* (see figure 16 right column). A frontal stagnation point is the source of a quasi-radial boundary layer that develops towards the top and side walls. The flow separates sharply at the leading edges, forming three free shear layers that encapsulate reverse and turbulent flow. Downstream of the cube, the recirculating fluid develops immediately behind the leeward face featuring an arch-like structure, followed by the reattachment line. Figure 18 displays the corresponding mid-plane pressure coefficient profiles extracted along the front, top, and rear faces. A pronounced positive pressure peak is observed at the center of the windward face, indicating flow stagnation. In contrast, the top and lateral faces exhibit negative values of the pressure coefficient. The reconstructed distributions are compared against pressure tap measurements from the experimental campaign of Castro and Robins (1977) at similar $Re_H$. The pressure distributions reconstructed with *IDP* show excellent agreement with the pressure tap measurements along the side and front faces. Top face reconstruction slightly overpredicts the $C_P$ drop, likely due to enhanced separation and uncertainty in experimental estimations of the Reynolds stresses. Despite this deviation, the reconstructed pressure captures the evolution along the separated flow region. Negative pressure coefficients of larger magnitude on the roof and rear of the cube than those found by Castro and Robins (1977) have also been observed by Lim *et al* (2009) and Gough *et al* (2019) who ascribed the differences to variations in upstream boundary layer characteristics and surface roughness, which alter the local wind shear and turbulence intensity at roof height. Overall, the robustness of the *IDP* approach is also demonstrated for complex 3D flows.

## 7. Conclusions and recommendations

This work introduced *IDP* as a generalized, robust framework for pressure reconstruction from PIV and PTV measurements using the gradient integration technique. The work addresses a long-standing limitation of pressure-from-PIV associated with boundary-condition sensitivity and error propagation. By iteratively refining the interface between rotational and irrotational flow regions and incorporating self-correcting mechanisms, *IDP* converges toward a physically consistent partitioning of the flow field while progressively shortening integration paths. As a result, the method strongly mitigates the influence of local measurement noise and eliminates the dependence on user-specified boundary conditions.

Validation on airfoil data shows that *IDP* converges within a few iterations and returns a pressure field largely insensitive to the initial placement of boundary conditions. The Monte Carlo analysis indicates that *IDP* reduces boundary conditions sensitivity by approximately two orders of magnitude compared to the predecessor *DP* method, *SSDP*. The extension to 3D data successfully reconstructs the surface pressure distributions around a wall-mounted cube at $Re = 80\,000$ and the comparisons with pressure-tap measurements show a good agreement over the front and side faces, while capturing the key trends in the separated flow region.

Overall, the results demonstrate that *IDP* is suited for a broad range of applications in aerodynamics for pressure reconstruction of both 2D and 3D time-averaged flows. The general formulation, rapid convergence, and low sensitivity to boundary-condition specification make it particularly suitable for experimental configurations with complex geometries, strong separation, or limited access to reliable boundary data.

The present results also open some directions for future developments to unsteady flows, dominated by vortex shedding, transient separation, or fluid-structure interaction. The algorithm is suited for parallelization, since integration and classification operations occur locally within the evolving *ROT* region. GPU or multi-core implementations could therefore reduce computational time significantly, enabling routine application of *IDP* to large-scale 3D PIV datasets.

## Acknowledgment

This project has received funding from the European Union's Marie Sklodowska-Curie Actions Postdoctoral Fellowship under Grant Agreement No. 101149435, project DIAMONDS.

Luuk Hendriksen is kindly acknowledged for making available the 3D database of the flow around the wall mounted cube.

## Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

## Author contributions

Alex M Donțu 0009-0007-7299-1877
Conceptualization (equal), Data curation (equal), Formal analysis (equal), Investigation (equal), Methodology (equal), Software (lead), Visualization (equal), Writing – original draft (lead), Writing – review & editing (equal)

Abbas Daliri 0000-0001-9399-4216
Conceptualization (equal), Data curation (equal), Funding acquisition (lead), Investigation (equal), Methodology (equal), Supervision (equal), Validation (equal), Visualization (equal), Writing – review & editing (equal)

Fulvio Scarano 0000-0003-2755-6669
Conceptualization (equal), Data curation (equal), Formal analysis (equal), Funding acquisition (equal), Methodology (equal), Project administration (lead), Resources (equal), Supervision (equal), Validation (equal), Writing – review & editing (lead)

## Appendix A. *IDP* boundary evolution

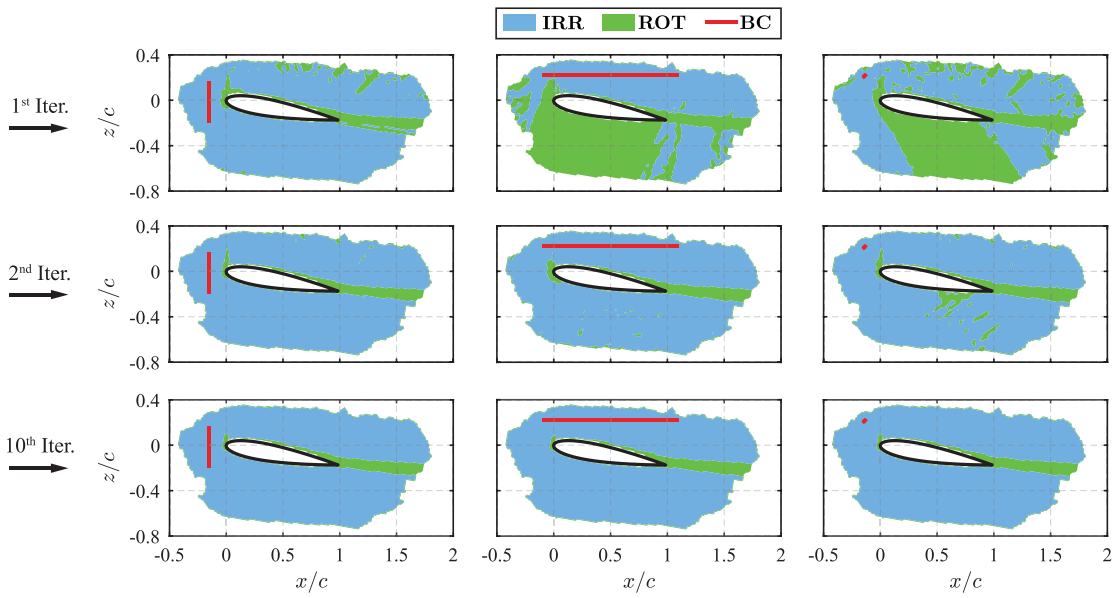


**Figure A1.** Evolution of IRR-ROT domain boundary for initial vertical DBC (left column), horizontal DBC (middle column), and point DBC (right column).

## Appendix B. Example *IDP* code

```
%INPUTS: read PIV field + define free-stream/reference pressures
dataTable = readtable(`csv_out/PIV2D_alpha10.csv'); nx = max(dataTable.i);
nz = max(dataTable.k);
Vinf = 14.5; P0 = 102 335; rho = 1.206; q_inf = 0.5 * rho * Vinf^2; P_inf = P0 -
q_inf;
X = reshape(dataTable.X, nx, nz); Z = reshape(dataTable.Z, nx, nz);
U = reshape(dataTable.U, nx, nz); W = reshape(dataTable.W, nx, nz);
dx = mean(diff(X(:,1)))/1000; dz = mean(diff(Z(1,:)))/1000;

%PREPROCESS: validity mask + Bernoulli pressure for IRR initialization
isValid = ~(U == 0 | W == 0); U(~isValid) = NaN; W(~isValid) = NaN;
speedSquared = U.^2 + W.^2; BernPressure = P0 - 0.5 * rho * speedSquared;

% BUILD ∇p: compute velocity gradients and form dp/dx, dp/dz
[Uz, Ux] = gradient(U, dz, dx); [Wz, Wx] = gradient(W, dz, dx);
Ux(isnan(Ux)) = 0; Uz(isnan(Uz)) = 0; Wx(isnan(Wx)) = 0; Wz(isnan(Wz)) = 0;
Px = -rho * (Ux .* U + Uz .* W); Pz = -rho * (Wz .* W + Wx .* U); % ∂p/∂x and
∂p/∂z

%% [Flowchart: Box 1] Initialize IRR^(0) and ROT^(0) by specifying DBC
xBC = − 100; zBC = 50; dist2 = (X---xBC).^2 + (Z---zBC).^2; dist2(isnan(U)) = inf;
[~, linearIdxBC] = min(dist2(:)); [iBC, kBC] = ind2sub([nx nz], linearIdxBC);

% [Flowchart: Box 1] Initialize pressure field + partition (IRR = 1, ROT = 0,
invalid=−1)
Pressure = zeros(nx, nz); Pressure(isnan(U)) = P0; Pressure(iBC,
kBC) = BernPressure(iBC, kBC);
PartitionMask = zeros(nx, nz); PartitionMask(isnan(U)) = − 1; PartitionMask(iBC,
kBC) = 1;

%Neighbour stencil + stopping tolerances (δ_p, δ_c)
neighborDi = [−1 1 0 0 −1 −1 1 1]; neighborDk = [0 0 1 −1 −1 1 −1 1];
maxOuterIters = 40; cpTotLower = 0.985; cpTotUpper = 1.05;
deltaP = 1e-2; deltaC = 1e-2; cpPrev =[]; maskPrev =[]; %tolerances

for iter = 1:maxOuterIters

    %% [Flowchart: Box 2] Perform spatial integration of ∇p in the
    estimated region
    %(Implementation: grow region by integrating from existing IRR neighbours)
    while any(PartitionMask(:) == 0)
        neighbors = conv2(double(PartitionMask == 1),ones(3),'same')-double
        (PartitionMask==1);
        neighbors(PartitionMask ~= 0) = -inf;
        bestNeighborCount = max(neighbors(:));
        [CandidateI, CandidateK] = find(neighbors == bestNeighborCount);
        for c = 1:numel(CandidateI)
            x0 = CandidateI(c); z0 = CandidateK(c);
            if PartitionMask(x0, z0) ~= 0, continue; end
            PressureSum = 0; numContrib = 0;

            % [Flowchart: Box 2] Local integration update from IRR neighbours
            using ∇p
            for n = 1:8
                x = x0 + neighborDi(n); z = z0 + neighborDk(n);
                if x < 1 || x > nx || z < 1 || z > nz || PartitionMask(x, z) ~= 1,
                continue;
```

```
                end
                dX = (X(x, z)---X(x0, z0))/1000; dZ = (Z(x, z)---Z(x0, z0))/1000;
                ds = hypot(dX, dZ);
                if ds == 0, continue; end
                unitDirX = dX/ds; unitDirZ = dZ/ds;
                dp_ds = 0.5*((Px(x0, z0)+ Px(x, z)) *unitDirX+(Pz(x0, z0)+
                Pz(x, z))*unitDirZ);
                PressureSum = PressureSum + (Pressure(x, z)---dp_ds * ds);
                numContrib = numContrib + 1;
            end

            Pressure(x0, z0) = (numContrib > 0) * (PressureSum/numContrib) + ...
            (numContrib == 0) * BernPressure(x0, z0);
            PartitionMask(x0, z0) = 2;
        end
        PartitionMask(PartitionMask == 2) = 1;
    end

    %Compute δ_p metric from successive Cp fields (equation (14)
    cpNow = (Pressure---P_inf)/q_inf; validCp = ~ isnan(cpNow); relL2 = inf;
    if ~isempty(cpPrev)
        relL2 = norm(cpNow(validCp)-cpPrev(validCp),2)/max(norm(cpPrev(validCp),
        2), eps);
    end

    cpPrev = cpNow;

    if iter ~= maxOuterIters
        Pressure(PartitionMask == -1) = P0;

        % Calculate Cp0 from current reconstructed pressure field (equation (3)
        cpTot = (Pressure---P_inf + 0.5 * rho * speedSquared)/q_inf;

        %% [Flowchart: Box 3] Obtain new preliminary IRR - ROT partitioning from
        Cp0 thresholds
        PartitionMask = double(cpTot > cpTotLower & cpTot < cpTotUpper);
        PartitionMask(isnan(U)) = - 1;

        %Dilate/regularize interface (remove IRR cells touching ROT/invalid)
        touchesRotOrInvalid = conv2(double(PartitionMask ⩽ 0),ones(3),'same') > 0;
        PartitionMask(PartitionMask > 0 & touchesRotOrInvalid) = 0;
        %% [Flowchart: Box 4 & 5] Conservative assumption (for simplicity):
        % all nodes in the uncertainty region are assigned to ROT. No integration
        is performed here.

        %Compute δ_c metric = fraction of nodes that flip class (equation (15)
        flipFrac = inf;
        if ~isempty(maskPrev)
            active = (PartitionMask~=-1);
            flipFrac = nnz(xor(PartitionMask==1, maskPrev==1) &
            active)/nnz(active);
        end
        maskPrev = PartitionMask;

        %% [Flowchart: Box 6] Terminate when both tolerances are satisfied
        if (relL2 < deltaP) && (flipFrac < deltaC)
            fprintf(`Converged at iter %d: relL2 = %.3 g, flipFrac = %.3 g\n',
            iter,relL2,flipFrac);
            break
```

```
        end

        % PIV-based pressure field
        Pressure = zeros(nx, nz); Pressure(PartitionMask == −1) = P0;
        Pressure(PartitionMask == 1) = BernPressure(PartitionMask == 1);
    end
end

%% [Flowchart: END] Final Cp from converged reconstructed pressure field
cp = (Pressure---P_inf)/q_inf; cp(PartitionMask == −1) = NaN;
figure; contourf(X, Z, cp, −2:0.25:1); clim([−2 1]); title(`C_p',
`FontSize', 32);
colormap(parula(12)); axis equal tight
```

## References


Baur T 1999 PIV with high temporal resolution for the determination of local pressure reductions from coherent turbulent phenomena *3rd Int. Workshop on Particle Image Velocimetry* (*Santa Barbara*)

Bosbach J, Kühn M and Wagner C 2009 Large scale particle image velocimetry with helium filled soap bubbles *Exp. Fluids* **46** 539–47

Cakir B O, Gonzalez Saiz G and Sciacchitano A 2024 Surface pressure reconstruction from LPT data with boundary conforming grids *Meas. Sci. Technol.* **35** 125303

Castro I P and Robins A G 1977 The flow around a surface-mounted cube in uniform and turbulent streams *J. Fluid Mech.* **79** 307–35

Charonko J J, King C V, Smith B L and Vlachos P P 2010 Assessment of pressure field calculations from particle image velocimetry measurements *Meas. Sci. Technol.* **21** 105401

Elsinga G E, Scarano F, Wieneke B and van Oudheusden B W 2006 Tomographic particle image velocimetry *Exp. Fluids* **41** 933–47

Gough H, King M-F, Nathan P, Grimmond C S B, Robins A, Noakes C J, Luo Z and Barlow J F 2019 Influence of neighbouring structures on building façade pressures: comparison between full-scale, wind-tunnel, CFD and practitioner guidelines *J. Wind Eng. Ind. Aerodyn.* **189** 22–33

Gurka R *et al* 1999 Computation of pressure distribution using PIV velocity data *Workshop on particle image velocimetry* vol 2 pp 1–6

Hendriksen L A, Sciacchitano A and Scarano F 2024 Object registration techniques for 3D particle tracking *Meas. Sci. Technol.* **35** 125202

Jux C, Scarano F and Sciacchitano A, 2019 Aerodynamic pressure reconstruction on generic surfaces from robotic PIV measurements *Proc. 13th Int. Symp. on Particle Image Velocimetry* (*Munich*)

Jux C, Sciacchitano A and Scarano F 2020 Flow pressure evaluation on generic surfaces by robotic volumetric PTV *Meas. Sci. Technol.* **31** 104001

Klein C, Engler R H, Henne U and Sachs W E 2005 Application of pressure-sensitive paint for determination of the pressure field and calculation of the forces and moments of models in a wind tunnel *Exp. Fluids* **39** 475–83

Li L, McClure J, Wright G B, Whitehead J P, Wang J and Pan Z 2025 Error propagation of direct pressure gradient integration and a Helmholtz–Hodge decomposition-based pressure field reconstruction method for image velocimetry *Exp. Fluids* **66** 92

Lim H C, Thomas T G and Castro I P 2009 Flow around a cube in a turbulent boundary layer: LES and experiment *J. Wind Eng. Ind. Aerodyn.* **97** 96–109

Liu X and Katz J 2006 Instantaneous pressure and material acceleration measurements using a four-exposure PIV system *Exp. Fluids* **41** 227–40

Liu X, Moreto J R and Siddle-Mitchell S, 2016. Instantaneous pressure reconstruction from measured pressure gradient using rotating parallel ray method *54th AIAA Aerospace Sciences Meeting*

Martinuzzi R and Tropea C 1993 The flow around surface-mounted, prismatic obstacles placed in a fully developed channel flow *J. Fluids Eng.* **115** 85–92

Nie M, Whitehead J P, Richards G, Smith B L and Pan Z 2022 Error propagation dynamics of PIV-based pressure field calculation (3): what is the minimum resolvable pressure in a reconstructed field? *Exp. Fluids* **63** 168

Pryce C, Li L and Pan Z, 2024. Revisiting Liu and Katz (2006) and Zigunov and Charonko (2024b): on the equivalence of the omnidirectional integration and the pressure Poisson equation *21st Int. Symp. on Applications of Laser and Imaging Techniques to Fluid Mechanics* (*Lisbon*)

Ragni D, Ashok A, Van Oudheusden B W and Scarano F 2009 Surface pressure and aerodynamic loads determination of a transonic airfoil based on particle image velocimetry *Meas. Sci. Technol.* **20** 074005

Scarano F 2012 Tomographic PIV: principles and practice *Meas. Sci. Technol.* **24** 012001

Scarano F, Ghaemi S, Caridi G C A, Bosbach J, Dierksheide U and Sciacchitano A 2015 On the use of helium-filled soap bubbles for large-scale tomographic PIV in wind tunnel experiments *Exp. Fluids* **56** 42

Schanz D, Gesemann S and Schröder A 2016 Shake-the-box: lagrangian particle tracking at high particle image densities *Exp. Fluids* **57** 70

Tagliabue A, Scharnowski S and Kähler C J 2017 Surface pressure determination: a comparison between PIV-based methods and PSP measurements *J. Vis.* **20** 581–90

van Oudheusden B W 2008 Principles and application of velocimetry-based planar pressure imaging in compressible flows with shocks *Exp. Fluids* **45** 657–74

van Oudheusden B W 2013 PIV-based pressure measurement *Meas. Sci. Technol.* **24** 032001

Zigunov F and Charonko J J 2024 A fast, matrix-based method to perform omnidirectional pressure integration *Meas. Sci. Technol.* **35** 065302