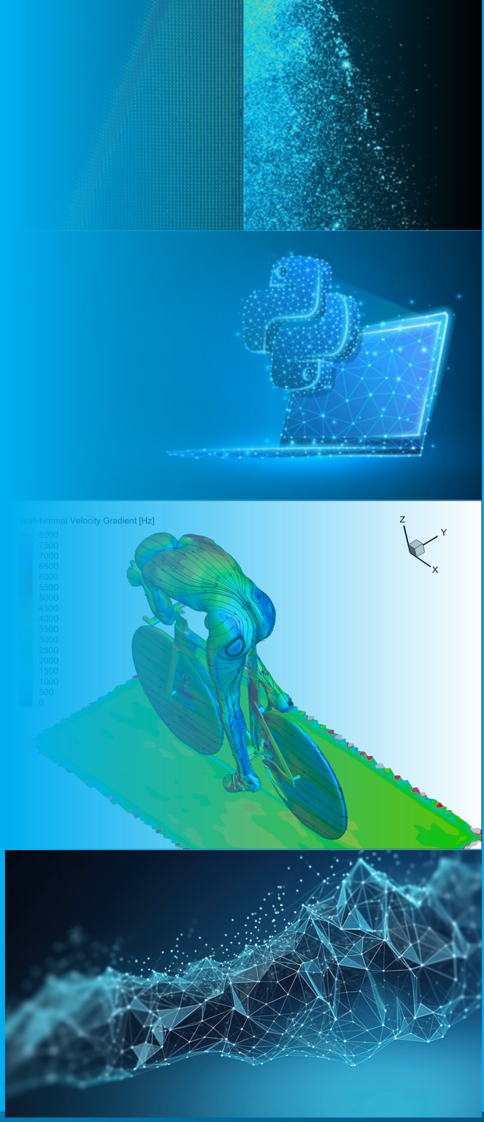
Technische Universiteit Delft

Near-surface fluid dynamics with large-scale Lagrangian particle tracking

Master of Science Thesis Report

E.P. Dedding



TU Delft

Near-surface fluid dynamics with large-scale Lagrangian particle tracking

Master of Science Thesis Report

by

E.P. Dedding

Front page images (top to bottom):
Planar PIV applied to a delta wing
Python emblem popping out of laptop
Wall-shear stress distribution around a cyclist
Low-poly mesh rendering of a flowing wave
VonKármán Institute for Fluid Dynamics (G. Paolillo)

Sources:
`https://www.spray.com/`
`https://pikbest.com/`
This report
`https://www.eetasia.com/`
`https://ls.vki.ac.be/media/product/images/2021_LS_PIV.jpg`

Student number:	4485092	
Committee members:	Dr. ir. B.W. van Oudheusden,	TU Delft, chairman
	Dr. ir. A.H. van Zuijlen,	TU Delft, external examiner
	Dr. A. Sciacchitano,	TU Delft, supervisor
	Prof. dr. F. Scarano,	TU Delft, supervisor
	L. Hendriksen,	TU Delft, advisor

TU Delft

Preface

E.P. Dedding
Delft, August 2024

Well, well, it has taken some eventful years to reach this point in my life. But hopefully, I will soon be able to embark on my next journey with a diploma under my belt. Experiences such as the student team Silverwing where we went to the U.S. for a drone flight competition or my internship at Alpine F1 team in Enstone, have really made me thankful of meeting new, bright people and having educational conversations. It has also taught me what I like most and what I like not so much. I have for example found a lot of joy in coding and creating software. It was one of the reasons I started a part-time job in IT. I have made some games on the side and have found new love and interest in combining it with my aerodynamics knowledge.

This thesis has given me the opportunity to experience academic research like I have never been able to before. As much as I love to just nitpick away at details and can lose myself for hours doing so, I have to be honest to myself and think that it has not always been so easy. Always keeping an eye on the bigger picture and truly understanding the core takeaways from research articles has not always been my forte. On the brightside, this experience has showed me what skills I should work on first and foremost.

In that aspect, I would like to express my gratitude to my thesis supervisors, Andrea and Fulvio, for supporting me throughout the project and preventing me from occasionally embarking on unnecessary (though very interesting) tangents. As I could sometimes feel lost in the world of articles and research, it was always refreshing to hear your thoughts on what is relevant pertaining to the bigger picture. With that, I want to really thank Luuk as well for not only sharing his experimental data and letting me partake in it, but for actively thinking along with me and proposing fresh insights when I sometimes have trouble seeing them myself.

Lastly, as you do, I want to thank all family and friends around me. My parents because they always have had faith that I will get there. My sibling for, well, being my siblings. Regardless of the moments, it is always comforting to think that I could rely on them if I'd have to. And lastly, thanks to my girlfriend Zahra who has not always experienced the best sides of me throughout last year. Late nights working on my thesis or staying away for long periods of time, because I just could not stop.

I am happy to have reached this point in my life and I am honestly buzzing with excitement to take it up a notch after graduating. I will be continuing my career at the Dutch meteorological institute KNMI, working on implementing a Numerical Weather Prediction model which makes use of data from hyperspectral IR satellites and has been developed at the English weather institute in Exeter. I really can't wait to start that - and actually visit England again, since that is part of the job :). Finally, to anyone reading this and thinking, hmm, I'd might want to get in touch with this person (either because you want to ask me something or maybe it is in regards to this thesis), please do not hesitate to reach out to me via my LinkedIn: `https://www.linkedin.com/in/erik-dedding`

Contents

Preface ii

1 Introduction 1

2 Background information on particle-based volumetric velocimetry techniques and near-wall flow measurements 5

2.1 Particle Image Velocimetry and Lagrangian Particle Tracking . . . 5
2.1.1 Short summary on the evolution of PIV and LPT . . . 6
2.1.2 Experimental data acquisition . . . 6
2.1.3 Data processing . . . 12
2.1.4 Post-processing volumetric velocimetry results . . . 17
2.1.5 Data assimilation techniques . . . 20
2.2 The boundary layer . . . 21
2.2.1 Boundary layer description . . . 21
2.2.2 Boundary layer parameters . . . 22
2.3 Object registration . . . 23
2.3.1 Experimental process . . . 23
2.3.2 Point-to-surface based object registration . . . 23
2.3.3 Uncertainty of the registration . . . 24

3 Literature survey on near-wall flows of generic 3D objects 27

3.1 Direct measurements of near-wall flows . . . 27
3.1.1 Oil Film interferometry . . . 28
3.1.2 Floating elements . . . 29
3.2 Near-wall flows with particle image velocimetry and Lagrangian particle tracking . . . 30
3.2.1 Flat plate flows . . . 30
3.2.2 Near-wall flows around generic objects . . . 32
3.3 Applying data assimilation to near-wall PIV and LPT . . . 35
3.3.1 Data assimilation approaches . . . 35
3.3.2 Implementing boundary conditions . . . 37
3.4 Research goal statement . . . 39
3.4.1 Research gap identification . . . 39
3.4.2 Research objective and questions . . . 39

4 Experimental dataset 41

4.1 Test facility: W-tunnel . . . 41
4.2 Surface-mounted test objects . . . 41
4.3 Experimental setup and procedures . . . 42
4.3.1 Imaging setup . . . 43
4.3.2 Tracer particles and flow seeding . . . 43
4.3.3 Data acquisition . . . 44
4.4 Data processing . . . 44

5 Local near-surface flow reconstruction 47

5.1 The mesh discrepancy . . . 47
5.1.1 The gap between the model and fluid domains . . . 48
5.1.2 Jux' dilated surface approach . . . 49
5.1.3 Linear interpolation onto a dilated surface . . . 49

5.2 The computational output domain . . . 52
5.2.1 Generating the output-mesh . . . 53
5.2.2 Local interrogation volume approach . . . 54
5.2.3 Merging the fluid domain and model domain - the complete picture . . . 56
5.3 Fluid reconstruction with LIVA interpolation schemes . . . 57
5.3.1 Trilinear fit with Delaunay triangulation including wall discretisation . . . 57
5.3.2 Quadratic fit of binned data with wall ghost points . . . 58
5.3.3 Quadratic fit of tracer data with wall ghost points . . . 61
5.3.4 Quadratic fit of tracer data with single wall constraint. . . . 62
5.3.5 Representing wall-shear stress distributions on 3D surfaces . . . 62
5.4 Object registration correction for ground plane . . . 63

6 Assessment of velocity reconstruction techniques **67**
6.1 Formulating a benchmark solution. . . . 67
6.2 Processing results with LIVA-console . . . 70
6.2.1 General outline of the console . . . 70
6.2.2 Configurable settings. . . . 73
6.2.3 Built-in viewers . . . 76
6.3 Comparison of LIVA-reconstruction schemes . . . 77
6.3.1 Velocity profiles . . . 78
6.3.2 Wall-shear stress distributions . . . 82

7 Skin friction topology around generic objects **85**
7.1 The surface-mounted cube case . . . 86
7.2 Processing the wing-body junction case . . . 87
7.3 The cyclist case. . . . 89

8 Conclusions and Recommendations **91**

Appendices **95**

A Fitting boundary layer profiles with wall functions **97**
A.1 Fitting with the Clauser chart and Coles’ method . . . 97
A.2 Results of wall function fitting . . . 98

B Tutorial on LIVA-console to process LPT results **101**
B.1 Generating the output-mesh . . . 101
B.2 Processing the surface-mounted cube with the LIVA-console . . . 102

C Data representation of object-mesh, fluid-mesh and output-mesh **107**
C.1 The object-mesh file format - .STL file. . . . 107
C.2 The fluid-mesh representation and file format - .DAT file . . . 108
C.2.1 Tracer-based representation . . . 108
C.2.2 Bin-based representation . . . 110
C.3 The output-mesh file format - VTK file format . . . 114

Bibliography **115**

List of Figures

2.1 Schematic of planar PIV. ([Raffel et al., 2018]) . . . 7
2.2 Schematic of an object in the object plane being projected onto the image plane, i.e. the camera. . . . 9
2.3 Depth of focus for an acceptable diameter of the geometric image sketch for both extreme positions of out-of-focus imaging. ([Raffel et al., 2018]) . . . 10
2.4 Cross-like and linear imaging configurations of tomographic PIV systems based on four cameras. ([Scarano, 2012]). . . . 11
2.5 Sketch of the optical setup used by Schröder et al. [2008]] which enabled the optimal usage of the laser pulse energy of only $21\,\mathrm{mJ}$ per pulse for illumination of the PIV measurement volume at 4 and $5\,\mathrm{kHz}$. . . . 12
2.6 Summations of disparity maps over 1–16 recordings. ([Wieneke, 2008]) . . . 13
2.7 Scheme of the OTF calibration. Two cameras are used to image the interrogation volume, divided into 8 subvolumes. The following steps can be identiftied. (1) The particles are sorted into their subvolumes. (2) The shape parameters of all detected particles are determined by fitting Gaussian distributions. (3) The shape parameters are subequently averaged over each subvolume. (4) The particles are corrected to fit a desired diameter, typically 2-3 pixels. (5) The results of all subvolumes are combined to create the complete OTF-maps for all cameras. ([Schanz et al., 2012]) . . . 14
2.8 Schematic showing the different steps in IPR. (1) Peaks are detected on the 2D images, which are used to (2) triangulate the 3D positions of the particles in space. (3) These are then optimised by slightly moving the particle around in space and (4) particles with an intensity which falls below the prescribed threshold are removed. (5) The particles and their detected intensity are removed from the residual images. These are then iteratively processed until no more new particles are detected. ([Schröder and Schanz, 2023]) . . 15
2.9 Schematic showcasing how adding the particle image shape is to the residual affects the fitting of the particle position ([Wieneke, 2012]) . . . 16
2.10 Schematic explaining the Shake-The-Box approach. First, particles are detected using IPR for a number of N_{init} time steps. When tracked consecutively, they are used to form particle tracks. These tracks are subsequently used as a prior for the particle's position in the next time step. This estimate is corrected using a ”shaking” approach similar to IPR. The detected particles are subtracted from the images. New particle tracks are formed and old tracks of which the particles have left the measurement domain are deleted. This is iteratively repeted until no more particles are detected. ([Schröder and Schanz, 2023]) 16
2.11 Schematic showing how particle tracks can pass through multiple bins at different times. 18
2.12 The source of residual error due to top-hat filter, Gaussian filter and a polynomial fit. ([Agüera et al., 2016]) . . . 18
2.13 The approach to functional binning of particle tracks. In the first filter, tracks with a midpoint very far away are removed. In the second filter, the closest distance is computed and only the tracks which pass through the bin are chosen. ([Godbersen and Schröder, 2020]) . . . 19
2.14 Results of vorticity reconstruction using (*left*) linear interpolation, (*middle*) VIC without $\mathrm{D}u/\mathrm{D}t$ term and (*right*) VIC+. ([Schneiders and Scarano, 2016]) . . . 21
2.15 Schematic description of the velocity profile near the wall in a turbulent boundary layer, showing the various layers which exist at different scales.(recreated from [Ting, 2016]) . 21

2.16 Reconstructed surface markers from a. integrated reflective markers, b. single laser pointer and c. a laser pointer with optical pattern module. The CAD model is shown in d. [Reprint from Hendriksen, 2024] . . . 24
2.17 Example of object registration which corrects an initial position of the cube to fitting it to the red surface markers. . . . 25

3.1 Example of how an oil drop develops to for a film, which indicates the interferometric pattern that can be used to determine the thickness. The images progress in time from left to right and the flow comes in from the bottom. [Örlü et al., 2010] . . . 28
3.2 Two OFI measurement setups varying from very simple (a) using only a light box to very complex (b) where the wind tunnel walls have been painted white to increase light reflection [Driver, 2003]. . . . 28
3.3 Sketch of a floating element balance Vinuesa and Örlü, 2017 . . . 29
3.4 Results showing (A) mean velocity profile and (b) Reynolds stress profile inside a flat plate boundary layer obtained via planar PIV and tomographic PIV [Kempaiah et al., 2020]. . . . 30
3.5 Plots of the instantaneous streamwise velocity with iso-surfaces of ejection events (parameterised as $u'v' < 0$) for (left) the stationary plate and (right) the oscillating plate, as reported by Scarano et al. [2022a]. . . . 30
3.6 Exemplary snapshot of STB evaluation in the turbulent boundary layer of an adverse pressure gradient flow at free-stream velocity of $14\,\mathrm{m\,s^{-1}}$. (Reproduced from [Schanz et al., 2019] . . . 31
3.7 Mean velocity profiles and Reynolds stress profiles in a turbulent boundary layer after binning multi-pulse STB results, compared against results from planar PIV, at edge velocity of $U_e = 35\,\mathrm{m\,s^{-1}}$. [Schröder et al., 2018] . . . 31
3.8 Results of the near-wall PIV measurement by Depardon et al. [2005] with velocities obtained at $0.5\,\mathrm{mm}$ (0.8% of the cube height) from the walls. . . . 33
3.9 Skin friction topology inferred from oil flow visualisation and near-wall PIV at high Reynolds numbers. Critical points are indicated, with green indicating saddle points, purples indicating nodes and red a focus. ([Depardon et al., 2005]) . . . 33
3.10 Example results of the three-dimensional flow around a surface-mounted cube. (see [Schröder and Schanz, 2023], reformatted from [Schröder et al., 2020]. . . . 34
3.11 Near-surface velocity streamlines plotted on a dilated surface of the cyclist, along with contours showing the velocity magnitude by Jux et al. [2018]. . . . 34
3.12 Near-surface velocity contour overlayed with skin friction topology at the lower back of a cyclist. [Schneiders et al., 2018]. . . . 35
3.13 Example results of the state-observer implemented into a 3D RANS model for the flow around a simplified car mirror by Saredi et al. [2021]. The left column shows the contours of the proportional feedback, whereas the right column shows the results of the proportional-integral feedback term. All plots are overlayed with the reference data. . . . 36
3.14 Results of the variational assimilation process applied to an idealised airfoil at high Reynolds number by Symon et al. [2017]. The left column shows the results from 2D PIV and the right column shows the assimilated results. The top plots show contours of the mean velocity and the bottom displays contours of the vorticity. . . . 37
3.15 Results of the ImVIC+ and ALE-VIC+ approaches for a period hill compared against the traditional VIC+. Moving from left to right the density of particles is increased. ([Cakir et al., 2022]) . . . 39

4.1 Overview of the W-Tunnel at the High Speed Laboratories of TU Delft. (Taken from [Delft University of Technology, 2023]) . . . 42
4.2 Image of the three models that were examined during the experimental campaign. Reflecting surface markers were integrated into the models on the left. The models on the right were marked using laser pointers. (Reprint from [Hendriksen, 2024]) . . . 42
4.3 Render of the wind tunnel test setup where the measurement domain is visualised using a glass box. Seven cameras are placed in a square configuration around the object and two LED-lights were used for illumination. . . . 43

4.4 Raw digital images of the (a) surface-mounted cube, (b) WBJ and (c) cyclist. 44
4.5 Settings used for the image preprocessing. 45
4.6 Settings used for the STB algorithm. 45

5.1 Schematic description of Cartesian-gridded bins lying (partially) inside a non-planar object. 47
5.2 Aliasing effect on binned results and the near-surface gap due to the presence of the curved model. The bin is shown in the top right. The black line indicates the aliased contour and represents the boundary of the interpolation domain, Ω_{int}. 48
5.3 Schematic of how the problem of aliasing and the near-surface gap would be resolved to increase Ω_{int} when the position of the model surface is included into the known data points. 49
5.4 Exampular triangulations which (a) do not meet Delaunay's criterion and (b) which does. 50
5.5 Barycentric coordinates $\lambda_{A,B,C}$ of a triangle simplex can be used to linearly interpolate inside the simplex. 50
5.6 The process of triangulating a set of two-dimensional points and consecutively computing the barycentric coordinates which is equivalent to the fitting of a plane inside each triangle. 51
5.7 Conventions of the object-mesh, fluid-mesh and output-mesh. 52
5.8 Example of (a) a volumetric output-mesh, generated around a cube and (b) a close-up of the output-mesh. 52
5.9 Example of (a) a surface output-mesh, generated to align with the cube and (b) a close-up of the output-mesh. 53
5.10 The two approaches to meshing a domain Ω by (a) first discretising the boundary and subsequently meshing the space that is bounded or (b) considering a global approach where the boundary and interior are meshed simultaneously. 54
5.11 A visual example indicating the LIVA-approach. A spherical volume is generated around an interrogation point in (a). Then both the fluid-mesh and the object-mesh are sliced with this sphere, yielding a local volume which contains part fluid-mesh data and part object-mesh data. 55
5.12 Example of the three-dimensional rotation using Rodrigues' transformation for (a) the general case where a vector $\mathbf{v}$ is rotated by an angle φ around the rotation axis $\mathbf{k}$ to obtain the new vector $\mathbf{w}$. The case of the global to local transformation is shown in (b), where the global up vector $\mathbf{v}$ must be rotated to align with the local normal $\mathbf{b}$, which requires the computation of the axis of rotation $\mathbf{k}$ and the angle ϑ. 56
5.13 Flowchart indicating the different components that are merged with the LIVA approach, after which a velocity reconstruction scheme can be applied to determine near-wall flow properties. 57
5.14 Visualisation of the three-dimensional triangulation for (a) a point on the top surface with (b) the set of triangulated tetrahedrons in a local coordinate system. 58
5.15 Results of Monte Carlo simulation to determine probability distribution of minimum radius for a randomly placed point to contain ten fluid-mesh nodes. The vertical red line represents the largest minimum radius $R_{min}/h = \sqrt{11/4}$, which corresponds to the interrogation point aligning with a fluid-mesh node. 59
5.16 Example of sampling strategies for (a) a cyclist CAD model, using (b) random sampling and (c) Poisson disk sampling. 59
5.17 Effect of double bin filtering applied to a signal s with Gaussian noise. The signal is made up of four wave numbers and a mean. The signal is subsequently binned with local polynomial fits, which crops the frequency spectrum. Applying a second binning on the signal s_b further distorts the frequency spectrum. 61
5.18 Plots of the misalignment of the tracers and the experimental models. Contours of the model are overlayed with tracer positions below $z = 10\,\mathrm{mm}$. 63
5.19 Schematic of the inner circle in red on which four outer circles of the same radius are drawn. In each outer circle the lowest tracer particle is sought. 63
5.20 Results of the proposed object registration correction for the ground plane, where (a) shows the local correction based on the search for lowest particles with a single inner circle and four outer circles. The results after the plane fitting are shown in (b). 65

6.1 Schematic of a single stack of coin-shaped bins. 68
6.2 Velocity profile of a binning with coin-shaped volumes stacked in the wall-normal direction. The velocity profile at (a) $(x, y) = (-180, 0)$ [mm] is shown next to the profile at (b) $(x, y) = (-120, 0)$ [mm]. The blue dot markers indicate the instantaneous velocity of each binned particle placed at the centre of the respective bin. The red cross markers depict are the result of the *Coin-Based-Tracer-LinReg* method. 68
6.3 Evolution of boundary layer velocity profile obtained from the coin-stack binning, ranging from $x = -180$ to $-72\,\mathrm{mm}$. 69
6.4 Evolution of boundary layer velocity profile obtained from the coin-stack binning, ranging from $x = -60$ to $60\,\mathrm{mm}$. 69
6.5 Functional flowchart of the LIVA-console. 70
6.6 Screenshot of the LIVA-console where the Geometry objects can be defined. 71
6.7 Screenshot of the LIVA-console where the Fluid objects can be defined. 72
6.8 Example of spherical interrogation volume cutting through the model such that points on the other side are also binned. 73
6.9 Screenshot of the LIVA-console where the Run settings can be defined. 74
6.10 Example of a semi-sphere where the sliced surface (a) is planar and (b) has a radius of curvature. 74
6.11 User-configurable settings in the LIVA-console for when the algorithm *Coin-Based-Tracer-LinReg* is chosen. 75
6.12 View of the 3D model viewer, when the object-mesh, fluid-mesh and output-mesh has been loaded. The object-mesh is shown in blue, the fluid-mesh in purple and the output-mesh in orange. 76
6.13 Example of the plot viewer for inspecting the results of the LIVA reconstruction. 76
6.14 Location of velocity profiles at plane $y = 0$ overlaid with the velocity magnitude contour and streamlines in the same plane. The cube has sides of length $L = 120\,\mathrm{mm}$. Note that the position of the cube in this figure is for illustration purposes and not exactly correct 77
6.15 Velocity profile (u-component) at location I. In (a) the mean velocity $\overline{u}$ is shown along with each respective reconstruction method ($\tilde{u}$), whereas in (b) the difference Δu is plotted defined as $\overline{u} - u_{GT}$. The red area indicates the uncertainty of the benchmark solution, whereas the orange dashed line indicates the data point closest to the surface, attainable with Jux' method. 78
6.16 Velocity profile (u-component) at location II. In (a) the mean velocity $\overline{u}$ is shown along with each respective reconstruction method ($\tilde{u}$), whereas in (b) the difference Δu is plotted defined as $\overline{u} - u_{GT}$. The red area indicates the uncertainty of the benchmark solution, whereas the orange dashed line indicates the data point closest to the surface, attainable with Jux' method. 79
6.17 Velocity profile (u-component) at location III. In (a) the mean velocity $\overline{u}$ is shown along with each respective reconstruction method ($\tilde{u}$), whereas in (b) the difference Δu is plotted defined as $\overline{u} - u_{GT}$. The red area indicates the uncertainty of the benchmark solution, whereas the orange dashed line indicates the data point closest to the surface, attainable with Jux' method. 79
6.18 Velocity profile (u-component) at location IV. In (a) the mean velocity $\overline{u}$ is shown along with each respective reconstruction method ($\tilde{u}$), whereas in (b) the difference Δu is plotted defined as $\overline{u} - u_{GT}$. The red area indicates the uncertainty of the benchmark solution, whereas the orange dashed line indicates the data point closest to the surface, attainable with Jux' method. 80
6.19 Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for ***Jux' method***. 81
6.20 Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for the ***Bin-Con-LinInterp*** method. 82
6.21 Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for the ***Bin-Nudge-QuadFit*** method. 83
6.22 Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for the ***Tracer-Nudge-QuadFit*** method. 84

6.23 Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for the ***Tracer-Con-QuadFit*** method. 84

7.1 isometric view of the skin friction magnitude around the surface-mounted cube overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 85
7.2 Views on the (a) front face and (b) rear face of the skin friction magnitude around the surface-mounted cube overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 86
7.3 Views on the (a) top face and (b) side face ($y = -60$) of the skin friction magnitude around the surface-mounted cube overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 86
7.4 Top view of the skin friction magnitude around the WBJ overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 87
7.5 isometric view from a forward position of the skin friction magnitude around the WBJ overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 88
7.6 isometric view from a rearward position of the skin friction magnitude around the WBJ overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 88
7.7 isometric view from a forward position of the skin friction magnitude around the cyclist overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 89
7.8 isometric view from a rearward position of the skin friction magnitude around the cyclist overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 90
7.9 Two views of (a) the front side and (b) the back side of the skin friction magnitude around the cyclist overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach. 90

A.1 Results of four wall function fitting strategies in table A.1 to the tracer particles at (a) the position $(x, y) = (-180, 0)\mathrm{mm}$ and (b) at $(x, y) = (-120, 0)\mathrm{mm}$. 99
A.2 Results of four wall function fitting strategies in table A.1 to the tracer particles at (a) the position $(x, y) = (0, 0)\mathrm{mm}$ and (b) at $(x, y) = (120, 0)\mathrm{mm}$. 99

B.1 View of the LIVA-console with surface-mounted cube object-mesh and output-mesh loaded. 102
B.2 View of the LIVA-console with surface-mounted cube fluid-mesh and tracer data loaded. 103
B.3 View of the side of the LIVA-console with only the tracer data points. This shows the error in the ground plane estimate. 104
B.4 View of the side of the LIVA-console after the ground-plane correction is computed. The corrected plane can be seen to fit the underside of the tracer data. 104
B.5 View of the LIVA-console after a single cell has been processed. The interrogation volume is displayed in yellow. 105
B.6 View of the LIVA-console with the 'Plot viewer' after executing the LIVA-algorithm for a cell 1765. 105

C.1 Digitisation of a cyclist model with (a) showing a rendered version and (b) a meshed version consisting of triangle surface elements. 107
C.2 The STL format representing a tessellated circular surface with eight triangle facets with a normal pointing out of the paper. 108
C.3 Example of four particles which are tracked over eight consecutive time steps. 109
C.4 The movie can be controlled with the buttons below it.
NOTE: Open the document in a PDF-viewer such as Adobe Acrobat Reader DC to play the movie. 110
C.5 Schematic description of ensemble binning. 110
C.6 Local coordinate system inside an interrogation bin. 111
C.7 Example of binning applied to a 1D sinusoidal with (a) a bin size or 2 and (b) a bij size of 1. 111
C.8 Mesh created from eight bins where velocity is sampled at the bin center. 112
C.9 Example of the three-dimensional flow field around a surface-moutned cube that is obtained from bin-based data. Three contours are plotted, one in each axis, displaying the contours of the velocity magnitude. 113
C.10 Overview of available linear cell types [Schroeder et al., 2006]. 115

List of Tables

4.1 Camera focal lengths and $f_{\#}$. 43
4.2 Summary of the model and experiment parameters 44

6.1 Table of shear stress τ_w magnitude $[\mathrm{mPa}]$ at different positions around the wall-mounted cube from various methods. 81

A.1 Summary of variables for the four fitting strategies with the Clauser chart method and Coles’ method and the role these play in the fitting. 98
A.2 Results of the nonlinear least squares fitting with uncertainty to velocity profile I for all four fitting approaches. 100

Listings

C.1 *.DAT* file example for tracer-based fluid data in ASCII format 109
C.2 *.DAT* file example for bin-based fluid data in ASCII format 113
C.3 *.VTK* file layout for unstructured grid, see also `https://docs.vtk.org/en/latest/design_documents/VTKFileFormats.html` 114

1

Introduction

In recent years, the volumetric flow measurement techniques Particle Image Velocimetry (PIV) and Particle Tracking Velocimetry (PTV) have provided valuable insight into turbulent fluid phenomena, volumetric pressure distributions and vortex topologies. In a world where the search for sustainability is reaching boiling points, the accurate determination of aerodynamic loads is of significant importance. To reach the goals of improving the fuel efficiency of aircraft by 2% per year up until 2050, the aircraft industry will have to rely on technological advancements such as blended wing-body airframes, hybrid propulsion and open rotors ([Bergero et al., 2023]). Assessing their aerodynamic performance will require appropriate flow measurement tools, capable of measuring the distribution of the three-dimensional loads.

As stated by Anderson [2017], the three-dimensional flow field over for example aircraft, automobiles or the blades of a wind turbine involves a complex interaction of the body geometry, air flow and ground interference. However, the aerodynamic loads that play a part in these cases can be reduced to only two sources, namely pressure distributions and shear stress distributions over the body surface. The above-mentioned appropriate fluid measurement techniques must allow an accurate measurement of these pressure and skin friction distributions, a requirement which could potentially be fulfilled using PIV or PTV. Note that for the remainder of this thesis report, the term particle tracking velocimetry is swapped in favour of Lagrangian particle tracking (LPT) when referring to the measurement of individual tracer trajectories as initially done by Ouellette et al. [2006]. This also emphasises the Lagrangian nature of the measurement technique, as it yields fluid velocities and accelerations in a Lagrangian reference frame.

Flows can be investigated via either numerical or experimental simulation. Numerical simulations of fluid are referred to as Computational Fluid Dynamics (CFD). The field of CFD has greatly benefited from the exponential increase in computational power, leading to [recent good example of “best” DNS (or other CFD) simulation]. However, numerical simulations are at the current stage still greatly limited by low Reynolds numbers (O(10^{3-4})) for three-dimensional flows. Experimental simulations greatly surpass that limit and flow measurement techniques employed therein remain a valuable tool for assessing fluid dynamics at higher Reynolds numbers (O(10^{5-6})).

Particle Image Velocimetry (PIV) and Lagrangian particle tracking (LPT), are common flow measurement techniques which can capture the complete three-dimensional flow. PIV and LPT rely on seeding of a fluid with tracer particles/bubbles to make the movement of the fluid observable. We can intuitively understand this concept from how leaves on a forest path may sometimes swirl indicating the presence of a small whirlwind. Using cameras to capture the bubble movement allows to quantify the displacement and determine a velocity field. PIV and LPT are used in many applications, such as for capturing vortex shedding behind cylinders [Scarano, 2010], investigating the efficiency of propulsors [Sciacchitano et al., 2018], capturing the flow around a cyclist [Jux et al., 2018] or visualising the turbulence-laden fluid around a surface-mounted cube [Schröder et al., 2020].

These are, however, all examples of bluff-body flow topology measurements and can be used for determining pressure distributions [van Oudheusden, 2013]. To also determine the wall-shear stress distribution, the near-surface fluid dynamics must be measured. An adequate technique applicable to generic objects is a complex task which to this day still has no straightforward solution due to the broad range of spatial and temporal scales in the boundary layer ([Örlü and Vinuesa, 2020]). Direct surface flow measurements are often done with oil flows or floating elements. Oil film interferometry (OFI) is regarded as one of the prime measurement techniques for quantitative time-averaged skin friction measurements ([Imai et al., 2023]). Boundary layer parameters such as the shape factor can not be determined. Additionally, oil films inevitably have to face the reality that is gravity, which poses difficulties for application to non-horizontal surfaces such as the sides of a cube. That is to say, applying oil to inverted surfaces will not work as the fluid will simply drip off the surface.

Indirect techniques using velocity measurements may provide a resolution for generic models when data close enough to the wall can be obtained ([Titchener et al., 2015]). Both boundary layer parameters and time-averaged wall-shear stress distributions can be determined via the fitting of velocity profiles ([Örlü and Vinuesa, 2020]), provided an accurate surface position can be determined. So far near-wall PIV has been applied and reported on in literature, but this requires the object to have only planar faces [Depardon et al., 2005] or taking very small measurement volumes in flat plate boundary layers (Kempaiah et al., 2020). Schröder et al. [2020] got large-scale measurements for a boundary layer, but did not report on skin friction. One research by Jux et al. [2018] did in fact report near-wall velocity measurements for a cyclist. Typically, this is not possible without accurately knowing the position of the model, but they were able to estimate within reasonable due to the large amount of fluid measurements taken.

Very recently, Hendriksen [2024] demonstrated an approach to accurately register a model using the original camera setup and an iterative closest point algorithm. This new development resolves one of the limitations for tomo-PIV and 3D LPT near surfaces, namely the inaccurate estimate of the model position. Coupled with the desire to push the operating window of PIV and PTV to measure near-surface fluid dynamical properties, this has motivated the presented MSc. thesis with the following objective:

The research objective is to reconstruct near-surface fluid dynamical properties in large-scale time-averaged PIV/LPT measurements by combining verified body position information with experimental results in the outer-field.

This thesis report proposes a method to combine the model position estimate and the fluid measurement techniques using state-of-the-art volumetric velocimetry results. The method employs interrogation volumes and a three-dimensional velocity functional to locally match the fluid measurement and surface geometry. A practical framework is described to merge information represented in the fluid domain $\Omega_{\mathcal{F}}$ and the model domain $\Omega_{\mathcal{M}}$ on a newly introduced output-mesh. The method is directly applied to real-world three-dimensional examples consisting of a surface-mounted cube and wing-body junction.

Firstly, an overview of the core concepts discussed and used throughout this report is given in chapter 2. Both the techniques PIV and LPT are discussed, along with boundary layers and the object registration approach by Hendriksen [2024]. A recap of the antecedent literature review is given in chapter 3. Recent examples of PIV and LPT are critically assessed along with the shortcomings which have not been appropriately addressed. Data assimilation techniques are included to shine light on the merging of measurement domains and computational domains. This is supplemented by an evaluation of near-surface fluid dynamical measurement techniques. The review paves the way to a gap in the current research which is addressed in this chapter, along with the research objectives and research questions which aim to achieve the stated objective.

This thesis report builds upon the experimental campaign of Hendriksen [2024]. The campaign measured the flow in large-scale LPT over a surface-mounted cube, wing-body junction and scaled cyclist. The experimental dataset and campaign is detailed in chapter 4. Next, the geometrical framework to bridge the mismatch between the fluid domain $\Omega_{\mathcal{F}}$ and the model domain $\Omega_{\mathcal{M}}$ is outlined in chapter 5. This chapter first describes the mismatch and gap between the two domains. A separate computational output-mesh is proposed onto which surface data can be interpolated, which was employed in a similar fashion by Jux et al. [2018] using a dilated model surface. Their approach is the most recent example in literature which provides (approximated) skin friction lines on a generic surface. It is considered the state-of-the-art for skin friction reconstruction. This approach is therefore also presented in more detail. Then, the family of velocity reconstruction methods which aim to merge the object and fluid domain are introduced. Several isotropic methods –hypothesised to improve the state-of-the-art– are proposed to reconstruct the skin friction topology.

To validate and quantitatively compare the proposed family of methods against a benchmark, a ”ground truth” is introduced in chapter 6, which exploits the anisotropic nature of boundary layers for planar geometries using flat cylindrical interrogation volumes. The outcomes are then assessed using a self-built LIVA-console which allows both processing and inspecting the velocity reconstruction. A tutorial for this console is included in appendix B. Finally, the methods are compared against each other based on velocity profiles and skin friction distribution plots with the surface-mounted cube case.

These methods are then assessed on the surface-mounted cube in a qualitative and quantitative manner. The experimental results of Depardon et al. [2005] and the computational results of Nigro et al. [2005] are used as reference data. The most suitable methods are then applied to the generic case of the wing-body junction and a complex scaled cyclist to assess the both the method’s flexibility and the resulting surface flow. The results of this assessment against literature references and the validation approach are presented and discussed in relation to the research questions in chapter 7. Lastly, the thesis is concluded in chapter 8. The report is summarised and recommendations are given for future work.

2

Background information on particle-based volumetric velocimetry techniques and near-wall flow measurements

PIV and LPT are techniques used (in controlled environments) to quantify and visualise fluid dynamics in three dimensions. The development of these techniques have advanced their use in both statistical and instantaneous investigations. They are similar in principle and experimental setup as they rely on the tracking of seeder particles in the fluid. These tracers are designed to follow the fluid movements with negligible lag, thereby allowing the measurement of the velocity field in three spatial dimensions and time. The main difference lies in the tracking objective as with PIV bins of tracer particles are cross-correlated through time to obtain an average displacement. On the other hand, LPT relies on the tracking of individual tracer particles through space and time and reconstructing their trajectories.

These techniques form the basis of this thesis report. Their procedures and working principles are therefore summarised in this chapter, providing background information to the reader. This chapter further includes the necessary information to understand the newly introduced methods and their evaluation, such as boundary layer flows and object registration. Firstly, the techniques Particle imaging velocimetry (PIV) and Lagrangian particle tracking (LPT) are detailed in section 2.1.The concepts for acquiring the experimental data and processing the digital images are treated and attention is given to post-processing techniques for Cartesian grid reduction (CGR) and data assimilation. Information on the representation of fluid measurements from LPT is considered as additional and included in appendix C. Next, an overview of the boundary layer is included in section 2.2. The various layers inside this viscous region are discussed and the classical boundary layer parameters are explained. Lastly, the newly proposed object registration method by Hendriksen [2024] is explained in section 2.3. The experimental process is covered together with the registration procedure. The uncertainty of the registration method is also discussed.

2.1. Particle Image Velocimetry and Lagrangian Particle Tracking

PIV and LPT are techniques used (in controlled environments) to quantify and visualise fluid dynamics in three-dimensions. The development of these techniques have pushed their use to both statistical and instantaneous investigations. They are similar in principle and experimental setup as they rely on tracking tracer particles. These are designed to follow the fluid movements, thereby allowing the measurement of the velocity field in three spatial dimensions and time. The main difference lies in the tracking objective as, with PIV, voxels (or bins) of tracer particles are cross-correlated through time to obtain an average displacement. On the other hand, LPT relies on the tracking of individual tracer particles through space and time.

As such, both techniques have developed into their own path of advantages, obstacles and possibilities with several similarities. To provide the reader with the necessary background on these particle tracer techniques, this section summarises their development and working principles. Firstly, a full overview of the method's history and evolution is provided for the interested reader in subsection 2.1.1. Then, the methods are split into an experimental acquisition part and a data processing part, respectively discussed in subsections 2.1.2 & 2.1.3, which together provide to the reader a complete summary of the working principles of PIV and LPT. Common post-processing techniques, such as CGR and pressure distribution determination, are denoted in subsection 2.1.4. Lastly, section 2.1.5 includes common data assimilation approaches which are considered to be part of the state-of-the-art.

2.1.1. Short summary on the evolution of PIV and LPT

Since its introduction in the 1980s, PIV has grown rapidly partly due to the arrival of electronic cameras and computers ([Raffel et al., 2018]) which allowed for more data to be gathered in a shorter time span. Better understanding of the physics coupled with technological advancements provided the scientific community with a suitable environment to expand on the technique. Initially with the introduction of planar PIV two velocity components were measured along two spatial directions ([Adrian, 1984] and [Pickering and Halliwell, 1984]). This was extended to three velocity components in two spatial directions with the introduction of stereo-PIV by Arroyo and Greated [1991]. Further development gave rise to techniques such as Scanning Light Sheet (SLS), Holographic PIV (holoPIV) and three-dimensional particle tracking velocimetry (3D PTV) which measure three velocity components (3D) along all three spatial components (3C) ([Scarano, 2012]). The benefit of such 3D3C-measurement techniques is the full reconstruction of the velocity gradient tensor. Although useful at the time, these techniques as described by Scarano [2012] had flaws which limited their development. Elsinga et al. [2006] brought in the novel tomographic PIV (tomoPIV) method that also enabled the full construction of the velocity gradient tensor. This technique sparked a range of developments offering the scientific community a non-intrusive measurement technique for the investigation of three-dimensional turbulence which quickly became an academical and industry standard.

Concurrently with the introduction of PIV, three-dimensional particle tracking was used in a similar manner to obtain 3D3C-velocity measurements. Initial work on this method delivered tracks for a single particle passing through a turbulent open channel water tunnel ([Jacobi, 1980]) or even 72 tracks using a stereo lens and almost a full day of manual post-processing ([Sheu et al., 1982]). It wasn't until a research group at the University of Tokyo devised an automated image processing technique for 3D PTV ([Nishino et al., 1989]) that it became a possible alternative method without significant manual labour. Further work focused on the feasibility and usability of the particle tracking ([Maas et al., 1993] and [Virant and Dracos, 1997]) which was also referred to as Lagrangian particle tracking (LPT) ([Ouellette et al., 2006]) due to the particle tracks which yield a Lagrangian representation of the fluid.

In the last decade much attention has gathered for the use of so-called data assimilation tools in post-processing experimental results. Instantaneous PIV and LPT data can still yield sparse results.

2.1.2. Experimental data acquisition

The experimental data acquisitions and working principles of 3D LPT and tomo-PIV are explained in detail in this section. Both methods depend on the flow being seeded with specialised tracer particles which satisfy specific criteria. The objective is to register these particles and track their movement in space. This is achieved by illuminating a desired volume of interest which is then captured using multiple cameras at high acquisition frequencies. The laser light that has been refracted by the bubbles is captured by the cameras as high intensity peaks. Each peak should coincide with a single particle. The velocity of individual particles or groups/bins of particles are calculated using their respective displacement and the acquisition frequency.

The general setup for (2D) PIV is drawn out in figure 2.1. Methods such as stereo- and tomo-PIV are increasingly more complex, yet the working principles are best demonstrated by considering the setup of planar PIV. The core components of a PIV experiment consist of the following:

- The flow is seeded with tracer/seeding particles. As the velocity field will be indirectly inferred from velocity of the tracer particles, their fluid mechanical properties are of high importance. Ad-

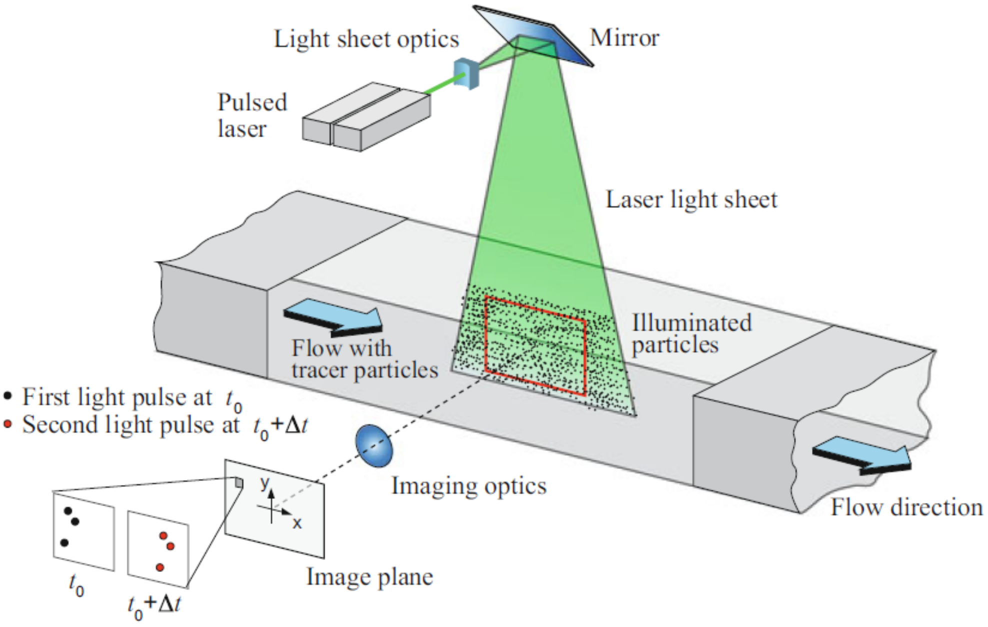


Figure 2.1: Schematic of planar PIV. ([Raffel et al., 2018])

ditionally, these particles must reflect sufficient light to be identified as clear peaks on the images. The amount of light reflected depends both on the light source as the choice of seeding particles. Optimal flow seeding conditions is often recognized as one of the most difficult aspects of a PIV experiment ([Scarano, 2010]), hence it is worth to review this aspect.

- The particles in the volume of interest are illuminated by a light source, optionally through a series of light optics. This volume can range from a thin light sheet to a rectangle or for example, as in the case of ([Schneiders et al., 2018]), a cone. Due to the intensity requirements, a high-powered laser is often chosen as the light source and optics are utilized to shape the laser light into the desired form.

- One or more cameras are focused on the illuminated region, optionally through a series of imaging optics. The camera(s) and light source(s) are generally controlled in parallel through a timing unit. A pair of snapshots separated by a short time span Δt are taken, which must be synchronized with the firing of the light sources.

- After the images are taken, they are processed to determine the velocity field. This is generally done through a statistical evaluation of image pairs. After a pre-processing operation, the recordings are divided into smaller interrogation windows (or volumes in 3D. Through a process of cross-correlation the average in-plane displacement of the interrogation windows is obtained.

The principal experimental elements of LPT are very similar to those of PIV. The primary differences are the sets of images that are taken during experimental campaigns and the post-processing of the image data. Whereas PIV relied on pairs of image to cross-correlate interrogation windows, particle tracking is usually done through a series of snapshots, called multi-frame or multi-pulse particle tracking ([Cierpka et al., 2013]). In the post-processing of the image data, the intensity peaks corresponding to individual peaks are iteratively determined through a process called iterative particle reconstruction (IPR) ([Wieneke, 2012]). Employing the information of individual particles a track can be fitted by connecting the correct particle through the time frames. When this is done dynamically, this particle track can be used as a predictor in the next time frame, which is the fundamental concept of Shake-The-Box ([Schanz et al., 2016]). These advancements have enabled the use of higher seeding densities while still reliably reproducing individual particle tracks.

Seeding particles

To visualise the movement of a fluid, it is seeded with a large number of tracer particles. For planar and stereo-PIV these particles are in the micrometer range and their seeding concentration typically lies between 10^9 and $10^{12}\,\mathrm{particles/m^3}$ ([Scarano, 2010]). These particles must adequately follow the flow and simultaneously scatter sufficient light to be distinguishable on the images. The manner in which particles follow a fluid is determined by the velocity lag or slip velocity U_s, which is the difference between the particle velocity U_p and the local fluid velocity U_f. A relation for the slip velocity U_s is obtained through Stokes' law:

$$\vec{U_s} = \vec{U_p} - \vec{U_f} = d_p^2 \frac{\rho_p - \rho_f}{18\mu_f} \frac{\mathrm{d}\vec{U_p}}{\mathrm{d}t}, \tag{2.1}$$

where d_p is the particle diameter, ρ_p and ρ_f are the particle and fluid densities, respectively, μ_f is the dynamic viscosity of air. It is apparent from equation (2.1) that the particle size must be as small as possible to reduce the velocity lag to an acceptable value. The dependency on the acceleration $\mathrm{d}\vec{U_f}/\mathrm{d}t$ additionally indicates that the slip velocity increases in turbulent flows, where fluctuations in time are observed. When the particle density and fluid density are approximately equal ($\rho_p \approx \rho_f$), the particles are referred to as neutrally buoyant. This is an optimal condition for minimization of the slip velocity, but is not easily attainable in gas flows. Therefore, tracer particles in gas flows are often several factors smaller than tracer particles in liquids.

The light scattering properties of particles are a fundamental property to consider for a successful PIV experiment, since it is directly responsible for the contrast of the recorded images. A convenient measure for the light scattering capabilities, according to Melling [1997], is the scattering cross section, C_s:

$$C_s = \frac{P_s}{I_0}, \tag{2.2}$$

where P_s is the total scattered power and I_0 the light source intensity incident on the tracer particle. Typical wavelengths for imaging lights are $527\,\mathrm{nm}$, whereas typical particle diameters for aerodynamics are about $1\,\mathrm{\mu m}$ [Raffel et al., 2018]. Thus, the particle size to wavelength fraction (d_p/λ) is generally larger than one, which corresponds to the Mie-scattering regime ([Mie, 1908]). In this regime, the side scatter efficiency is significantly less than the forward and backward scattering efficiency, which is the case with planar PIV. For stereo-PIV, tomo-PIV and 3D LPT, the cameras can be arranged in a forward configuration to maximise the signal intensity of the scattered light.

Production of the particles is important to achieve optimal seeding conditions. The generation of particles depends on the fluid medium and the test facility used. Seeding generators produce mixtures of air and seeding particles. Examples of seeding generators which generate liquid droplets are atomizers or hot plate vaporizers/condensers, while solid particles can be produced by entraining particles with a cyclone ([Scarano, 2010]). For the seeding of wind tunnel flows, the procedure varies with the facility. In a closed-loop (Göttingen Type) wind tunnel the flow is commonly seeded downstream of the test section, at the position of the diffuser. This minimises the interference of the seeding generator with the air in the test section. When the desired seeding level is attained, the production can be decreased to only account for the disappearing/evaporating bubbles. In the case of blow-down wind tunnels, the seeding generator is located upstream of the test section, which typically leads to sub-optimal, heterogeneous seeding conditions ([Raffel et al., 2018]).

The seeding density affects the resolution of the measured velocity field and the success of the cross-correlation or particle tracking algorithm. Typically higher seeding densities offer a higher spatial resolution, but negatively impact the chances of a successful cross-correlation or particle tracking. The most advance tomo-PIV and 3D LPT algorithms can handle a seeding density up to 0.15 - 0.2 ppp (particles per pixel) ([Schröder and Schanz, 2023)]. This value depends on the number of cameras N_C, the pixel-normalised particle image diameter d_τ^* and the particle image peak intensity ([Scarano, 2012]). Another parameter which relates to the seeding density is the source density N_S, which is defined by the fraction of the images which are occupied by particles.

The total number of particles N_P inside a measurement domain of volume V_M is proportional to the seeding concentration C as

$$N_P = V_M \cdot C \tag{2.3}$$

The resolution of the digital imaging D_R is a factor of the optical magnification M (defined by equation (2.9)) and the pixel pitch Δ_{px}. It indicates the number of pixels inside the digital image per millimeter or in other words the minimally identifiable length scale in the digital image.

$$D_R = \frac{M}{\Delta_{px}} \tag{2.4}$$

In tomo-PIV, the object space is discretized into voxels. Assuming that the voxel length is equal to the length of the pixel projection into the object space, the digital resolution also applies to $\mathrm{voxels\ mm^{-1}}$ in 3D ([Scarano, 2012]). The particle concentration inside a single voxel is then given as

$$PPV = \frac{C}{D_R^3}. \tag{2.5}$$

The source density N_S or particle seeding density in ppp can then be obtained as

$$N_S = \text{ppp} \cdot \frac{\pi}{4} \left(d_\tau^*\right)^2 = PPV \cdot W \cdot D_R \cdot \frac{\pi}{4} \left(d_\tau^*\right)^2 , \tag{2.6}$$

where $d_\tau^* = d_\tau / \Delta_{px}$ and W is the depth of the measurement domain (typically $L \times H \times W$). Lastly, the particle concentration C is related to the source density N_S as

$$C = \frac{N_S \cdot D_R^2}{W} \cdot \frac{4}{\pi \left(d_\tau^*\right)^2} \tag{2.7}$$

Helium-Filled Soap Bubbles

As was shown through equation (2.1), neutrally buoyant particles relax the stringent requirements on particle size. Larger particles have a significantly higher scattering efficiency, which had been one of the main limitations of upscaling PIV ([Scarano et al., 2015]). Neutrally buoyant particles are easier to realise in water flows, but for air flows helium-filled soap bubbles (HFSB) are a possiblity. Such particles were first used for qualitative visualisations ([Pounder, 1956], [Hale et al., 1971] and [Ferrel et al., 1985]). It was not until Kerho and Bragg [1994] investigated the use of HFSB as a quantitative measurement tool. The authors concluded that the bubbles faithfully followed the flow field, though it is important to ensure the particles are neutrally buoyant if quantitative measures are to be performed.

Most experiments hereinafter used HFSB as a qualitative technique and never inside wind tunnels due to the low production rates of commonly available generators. It was not until Scarano et al. [2015] demonstrated the quantitative feasibility of HFSB for flows up to $30\,\mathrm{m\,s^{-1}}$ with instantaneous tomo-PIV. Their result highlights both the significant increase in domain volume ($4.8\,\mathrm{L}$) and the high spatial resolution for an instantaneous investigation feasible with HFSB. Sub-millimetre HFSB were used, which remain the standard till this day as they input less momentum into the flow, have a higher generation efficiency, have higher sensitivity to small scale structures and consume less bubble fluid solution which reduces pollution ([Bosbach et al., 2009]).

Imaging

The imaging is a central part in both tomographic PIV and 3D LPT. To determine the position of particles in three-dimensional space, a minimum of three cameras are required. An accurate reconstruction

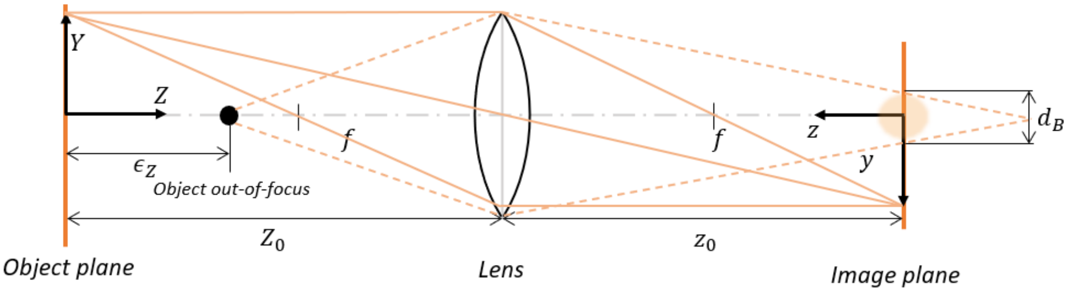


Figure 2.2: Schematic of an object in the object plane being projected onto the image plane, i.e. the camera.

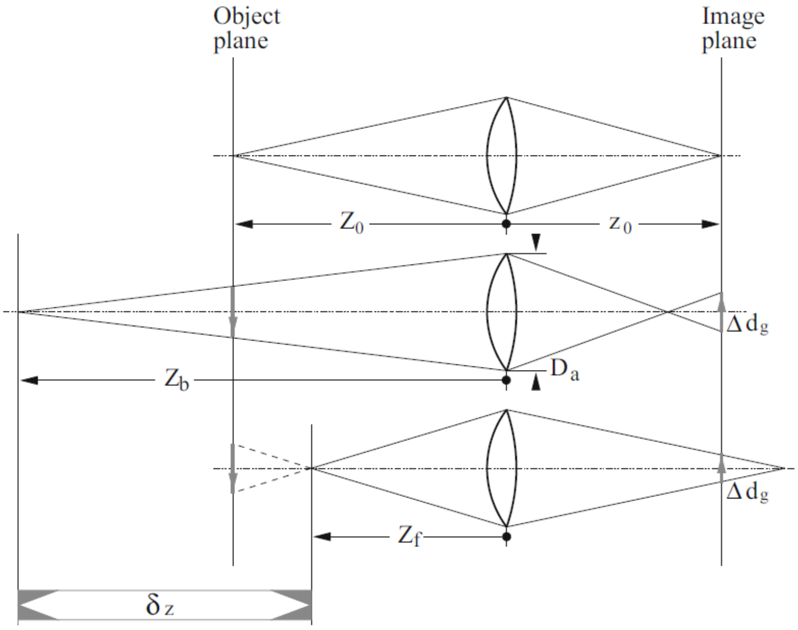


Figure 2.3: Depth of focus for an acceptable diameter of the geometric image sketch for both extreme positions of out-of-focus imaging. ([Raffel et al., 2018])

requires that all particles inside the measurement volume are in focus. When the light waves of any object passes through a circular aperture, a diffraction pattern is created behind the opening. The minimum diameter of this image diffraction spot d_{diff} is given by ([Goodman, 1996, p. 157])

$$d_{\text{diff}} = 2.44 \cdot \lambda \cdot f_{\#}\,(M+1)\,, \tag{2.8}$$

where λ is the wavelength of the light, $f_{\#}$ is the camera ratio between the focal length f and the aperture diameter D_a and M is the magnification factor. The magnification factor is defined as

$$M = \frac{z_0}{Z_0}, \tag{2.9}$$

where z_0 is the distance between the lens and image plane and Z_0 is the distance between the lens and object plane as defined in figure 2.2. The

The depth of focus δ_Z of the camera is also dependent on the aperture size $f_{\#}$. The depth of focus is illustrated in figure 2.3 and can be approximated, using the minimal image diffraction diameter as an acceptable diameter of the geometric image, as ([Raffel et al., 2018])

$$\delta_Z = 2 \cdot f_{\#} \cdot d_{\text{diff}} \frac{M+1}{M^2} = 4.88 \cdot \lambda \cdot f_{\#}^2 \left(1 + \frac{1}{M}\right)^2 \tag{2.10}$$

Equation (2.10) shows that for a large depth of view, one wants to maximise the f-stop $f_{\#}$, i.e. decrease the aperture diameter of the camera. On the other hand, increasing the aperture size, equation (2.8), yields *sharper* particle images as d_{diff} decreases. Increasing the aperture is also needed to ensure sufficient light is captured from each individual particle. The trade-off between these factors is important to consider for PIV and 3D LPT experiments ([Raffel et al., 2018]).

For particles which fall outside the region of focus, a blur is seen on the image plane. According to Scarano [2012] the diameter of this blur spot d_B can be approximated as

$$d_B \approx \frac{\epsilon_Z \cdot M^2}{f_{\#}} \tag{2.11}$$

where ϵ_Z is the distance between the object and the focal plane, shown in figure 2.2. This blur spot diameter can further be use in a simplified expression for the normalized particle image diameter as

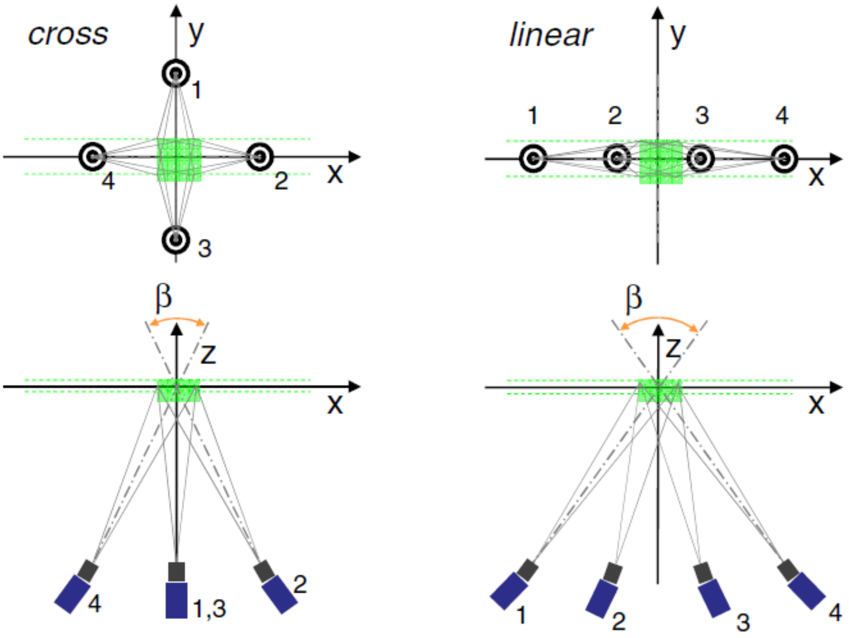


Figure 2.4: Cross-like and linear imaging configurations of tomographic PIV systems based on four cameras. ([Scarano, 2012]).

([Scarano, 2012])

$$d_\tau^* = \max\left\{1, \frac{\sqrt{d_{\text{geom}}^2 + d_{\text{diff}^2} + d_B^2}}{\Delta_{px}}\right\}, \tag{2.12}$$

where d_{geom} is the geometrical diameter of the particle. Scarano [2012] further gives an expression to update equation (2.10) as the f-stop $f_\#$ may be decreased to collect more light at the cost of a slight acceptable blur:

$$f_\# = \min\left\{\frac{M}{1+M}\frac{\sqrt{\delta_Z}}{4.88\lambda}, \frac{\delta_Z \cdot M^2}{4d_B}\right\} \tag{2.13}$$

According to Schanz et al. [2012] the blur diameter d_B may be increased and accounted for by using a variable optical transfer function which reduces these optical distortions.

Typical camera configurations for both 3D PIV and LPT are shown in figure 2.4. The cameras are positioned perpendicular to the illumination direction. Such configurations require a minimal depth of focus and offer flexibility in the choice of aperture angle β, which is maximised between $40\,\text{deg}$ and $80\,\text{deg}$ ([Elsinga et al. [2006]]). As reported by Scarano [2012], the one-dimensional linear configuration is 5% to 10% less accurate than the two-dimensional cross configuration. Such setups require optical access for both the cameras and the light source, which often limits the application of PIV and LPT. Schneiders et al. [2018] introduced a coaxial measurement configuration for use in volumetric velocimetry. The compact setup reduces the optical access requirements to a single direction of sight, at the cost of accuracy loss in the depth direction. This setup was then used to measure the near-wall velocity over a cyclist ([Jux et al., 2018]).

Illumination

PIV experiments are frequently done with lasers[1] as a light source. Lasers provide a bundle of monochromatic light with high energy density, which can be shaped into a thin or thick light sheet through an arrangement of optical lenses, such as a cylindrical diverging lens and a beam expander. The light intensity in the volume is inversely proportional to the light sheet thickness, which is a major limitation to upscale PIV measurements to larger volumes ([Scarano et al., 2015]).

---

[1] Lasers is an acronym for "Light Amplification by Stimulated Emission of Radiation"

Such limitations can be circumvented in various ways, such as special optical arrangement. Double-pass and multi-pass systems, shown in figure 2.5, can be employed to increase the gain factor as shown by e.g. [Schröder et al., 2008], [Scarano and Poelma, 2009] and [Schröder et al., 2011]. Ghaemi and Scarano [2010] showed that multi-pass systems can yield a gain factor 7 and 5 times larger compared to, respectively, single- and double-pass systems.

Recently, the advancements of Light Emitting Diodes (LEDs) have made it a viable option for PIV and LPT. One of the first demonstrations of LEDs used in volumetric velocimetry measurements is by Kühn et al. [2011]. The non-coherent actually yields very favorable imaging properties for use in particle tracking methods. Schanz et al. [2019] also used LEDs to measure a turbulent boundary inside a volume extending nearly $3\,\mathrm{m}$ – the total volume size was $450\,\mathrm{L}$.

2.1.3. Data processing

Once the experiment has been set up correctly, an adequate seeding density is achieved and the digital images have been acquired, the results will need to be post-processed to obtain the velocity field. The images are often processed in various stages, of which the most common ones are discussed in this subsection.

Volume self-calibration and optical transfer function

To improve the accuracy and even achieve sub-pixel accuracy for three-dimensional velocimetry, Wieneke [2008] introduced the volume self-calibration (VSC) method which was later improved ([Wieneke, 2018]). VSC was created from the insight that tomo-PIV required calibrations that were sub-pixel-accurate ([Schröder and Schanz, 2023]). The self-calibration is accomplished using the particle images at a lower seeding density without the model. The individual 2D particle peaks are determined in each camera and the corresponding 3D position are found by matching and triangulation. The particle positions (x_i,y_i) on the plane of each camera i are determined through an optimisation to obtain a best-fit for the real location (X,Y,Z), where minimisation of the sum of distances $|(x_i,y_i)-(x_i',y_i')|$ is frequently used. This distance is also termed the disparity d_i

$$d_i = (d_{i,x}, d_{i,y}) = (x_i, y_i) - (x_i', y_i'). \tag{2.14}$$

The disparity is then used to correct the mapping function $M_i(X,Y,Z)$ obtained from the initial calibration procedure as

$$M_i'(X,Y,Z) = M_i(X,Y,Z) - d_i(X,Y,Z). \tag{2.15}$$

These disparity vectors are usually obtained by first subdividing the complete measurement volume into $n_x \cdot n_y \cdot n_z$ sub-volumes. The disparities of all particles lying inside a single sub-volume are plotted and

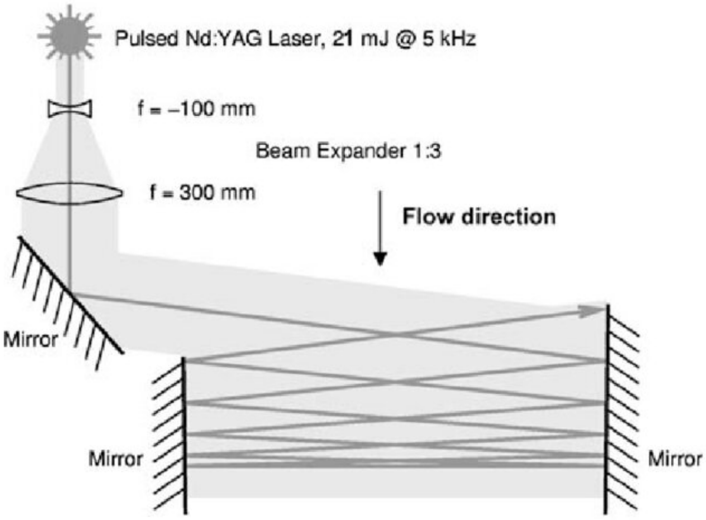


Figure 2.5: Sketch of the optical setup used by Schröder et al. [2008]] which enabled the optimal usage of the laser pulse energy of only $21\,\mathrm{mJ}$ per pulse for illumination of the PIV measurement volume at 4 and $5\,\mathrm{kHz}$.

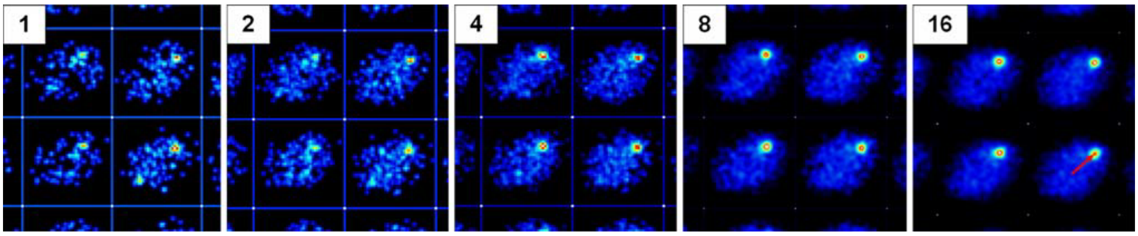


Figure 2.6: Summations of disparity maps over 1–16 recordings. ([Wieneke, 2008])

summed over all cameras over several images to improve the statistics. An example of the summed disparity maps is showcased in figure 2.6. With the addition of the disparities of more triangulated particles inside the sub-volume, an intensity peak becomes more pronounced, which corresponds to the most probable disparity vector. The final results can optionally be smoothed and even be repeated in an iterative manner. By fitting a Gaussian curve to the peak location, it is possible to achieve a sub-pixel self-calibration.

This approach was later improved to suppress ghost particles and enable detection of larger disparities ([Wieneke, 2018]). This enables the VSC method to be used without significantly decreasing the seeding density beforehand. Michaelis et al. [2021] demonstrated a calibration correction using non-parametric, 3D disparity fields. These were subsequently used to create correction fields for the pinhole model and mapping methods employing polynomials.

Building upon the VSC method, Schanz et al. [2012] introduced non-uniform optical transfer functions (OTFs). OTFs are used to map the 3D voxels to the corresponding camera pixels. These mappings had been homogeneous and uniform for the entire measurement volume and all cameras. This is often an incorrect assumption due to astigmatic distortions. Non-uniform OTFs are utilized to account for distortions per sub-volume and improve the voxel-pixel mapping accuracy.

The OTF routine is added to the VSC method as an additional processing step. When the individual particles are identified, their shape is identified using a two-dimensional Gaussian peak-fitter to find the parameters a, b, c and p of the weighting function

$$W(x,y) = p \cdot \exp\left(-\frac{1}{2}\begin{pmatrix}x\\y\end{pmatrix}^T \begin{pmatrix}a & b\\ b & c\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}\right), \tag{2.16}$$

where x and y are the pixel coordinates, a, b and c relate to the shape of the image and p is the peak height. This weighting is subsequently averaged over all particles inside a single sub-volume and accounted for by the desired particle diameter. The result is a non-uniform OTF-map for all cameras and all sub-volumes. This process is schematically shown for 2 cameras in figure 2.7.

Iterative particle reconstruction

In particle tracking velocimetry, the reconstructed particle positions are determined and tracked through several image frames. Through tomographic reconstruction techniques particle densities could be increased, but the accuracy was limited by the detection of 3D peaks and the high computational demands of tomo-PIV – in the order of 10 min/snapshot when using 1 Mpx images ([Scarano et al., 2021]). The introduction of the iterative particle reconstruction (IPR) method by Wieneke [2012] allowed a return to direct particle tracking at comparable seeding densities. The full IPR approach is shown schematically in figure 2.8. The approach starts off by finding particles in the 2D-images and performing 3D-triangulation to determine the position in physical space. This triangulation often involves errors of approximately $1\,\mathrm{px}$. These errors are reduced by optimising the 3D position and intensity of the triangulated particles, shown in step 3. First, the projected intensity of all triangulated particles is subtracted from the original image yielding a residual image for each camera i using

$$I^i_{\text{res}}(x_i, y_i) = I^i_{\text{orig}}(x_i, y_i) - I^i_{\text{proj}} x_i, y_i), \tag{2.17}$$

where the reprojected images of the particles are computed using a sum over all particles p as

$$I^i_{\text{proj}}(x_i, y_i) = \sum_p I^i_{\text{part}}(x_i, y_i, p). \tag{2.18}$$

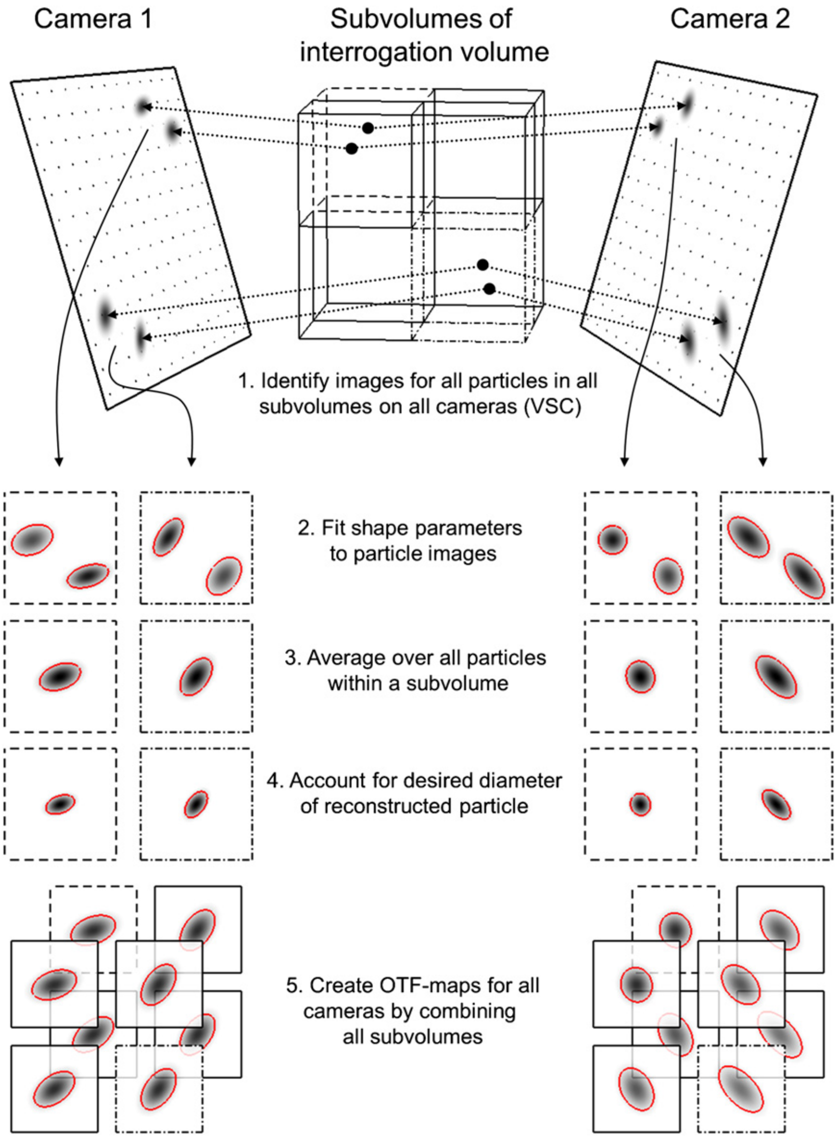


Figure 2.7: Scheme of the OTF calibration. Two cameras are used to image the interrogation volume, divided into 8 subvolumes. The following steps can be identiftied. (1) The particles are sorted into their subvolumes. (2) The shape parameters of all detected particles are determined by fitting Gaussian distributions. (3) The shape parameters are subequently averaged over each subvolume. (4) The particles are corrected to fit a desired diameter, typically 2-3 pixels. (5) The results of all subvolumes are combined to create the complete OTF-maps for all cameras. ([Schanz et al., 2012])

A particle-augmented residual image is then created by superimposing the residual image with the projection of the intensity of a single particle, visualised in figure 2.9, given by

$$I^i_{\text{res}+p}(x_i, y_i) = I^i_{\text{res}}(x_i, y_i) + I^i_{\text{part}}(x_i, y_i, p) \tag{2.19}$$

An optimisation approach is then employed to minimise the residual between the particle-augmented residual image and the projection of a single particle's intensity. The scheme optimises the coordinates (X'_p, Y'_p, Z'_p) of the particle along with its intensity I'_p through the following residual:

$$R = \sum_{i,x_i,y_i} \left(I^i_{\text{res}+p}(x_i, y_i) - I^i_{\text{part}}(x_i, y_i, X'_p, Y'_p, Z'_p, I'_p) \right)^2 \tag{2.20}$$

Remember that the physical particle position (X_i, Y_i, Z_i) is mapped to the camera plane using the mapping function of equation (2.15). For a more detailed description on how the particle position and intensity can be updated, the reader is referred to ([Wieneke, 2012]).

When the triangulation has been optimised and weak particles are filtered out (step 4 in figure 2.8), the residual images (following equation (2.17) are iteratively processed again. This process can be repeated until no more new particles are detected. The IPR approach achieved more accurately reconstructed particle positions up to source densities of $0.05\,\mathrm{ppp}$. Above this density, the fraction of wrongly detected particles increased significantly ([Wieneke, 2012]). Later, Jahn et al. [2021] demonstrated an improvement to the IPR method. They accomplished this by substituting the optimisation approach to minimise the residual defined in equation (2.20). Instead they used derivatives of the cost function, which allowed implementation of several different optimisation procedures such as the steepest-descent method.

Shake-the-box

The STB algorithm is displayed schematically in figure 2.10. The temporal information is seized by reversing the order of the reconstruction process. Instead of reconstructing the physical particle positions and, subsequently, matching them with the position in previous frames, the already reconstructed particle trajectories are used to initialise an estimate for the particle's position in the next time step. The actual position of the particles is then determined from the initialisation by a process later called "Shaking-the-particles" which was initially introduced by Wieneke [2012] and refers to the moving of the particle position to optimise a residual, such as the one in equation (2.20). The complete approach is outlined in this subsection ([Schröder and Schanz, 2023]), following figure 2.10.

1. *(Pre-)Initialistation phase:* Trajectories need to be initialised before they can be used in subsequent predictions. This is done using IPR ([Wieneke, 2012] or [Jahn et al., 2021]) over the first

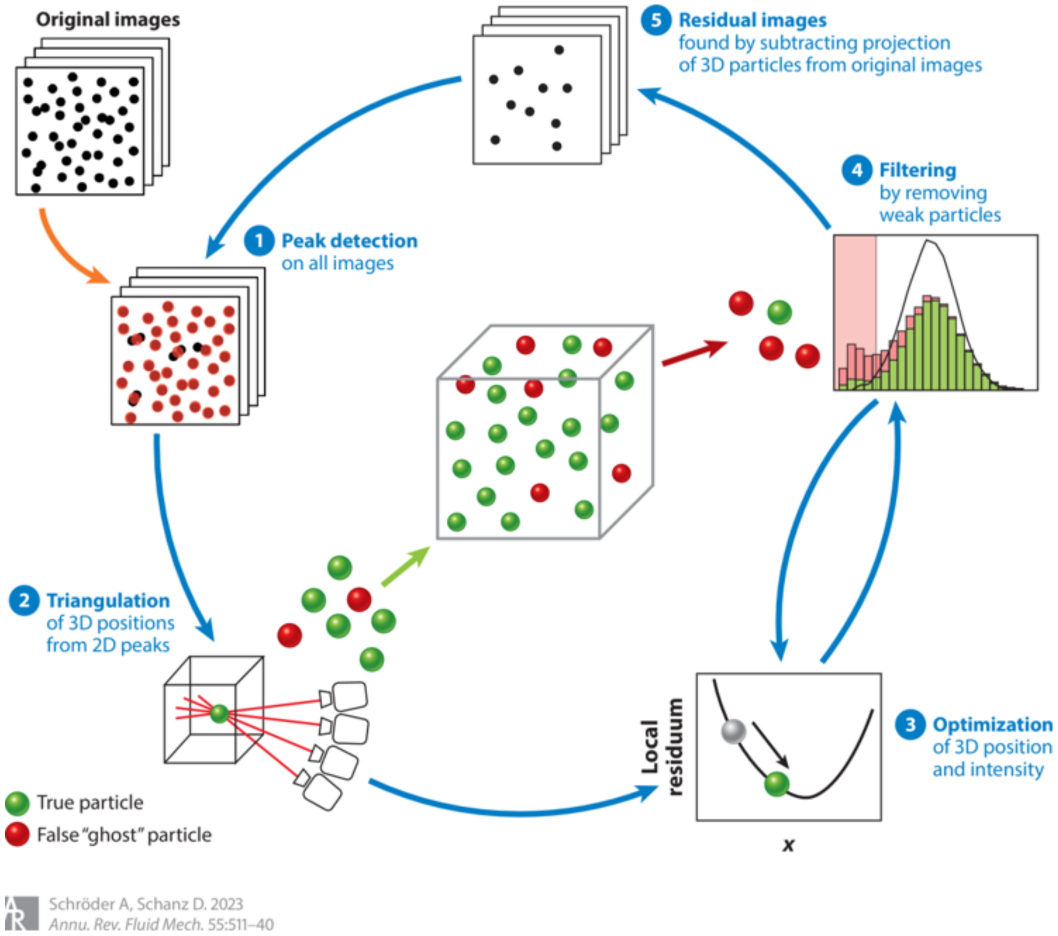


Figure 2.8: Schematic showing the different steps in IPR. (1) Peaks are detected on the 2D images, which are used to (2) triangulate the 3D positions of the particles in space. (3) These are then optimised by slightly moving the particle around in space and (4) particles with an intensity which falls below the prescribed threshold are removed. (5) The particles and their detected intensity are removed from the residual images. These are then iteratively processed until no more new particles are detected. ([Schröder and Schanz, 2023])

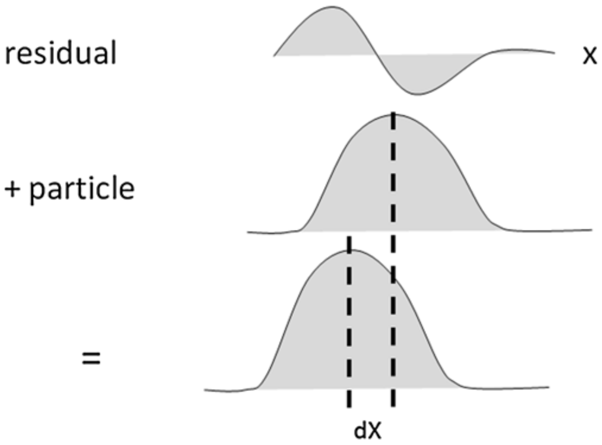
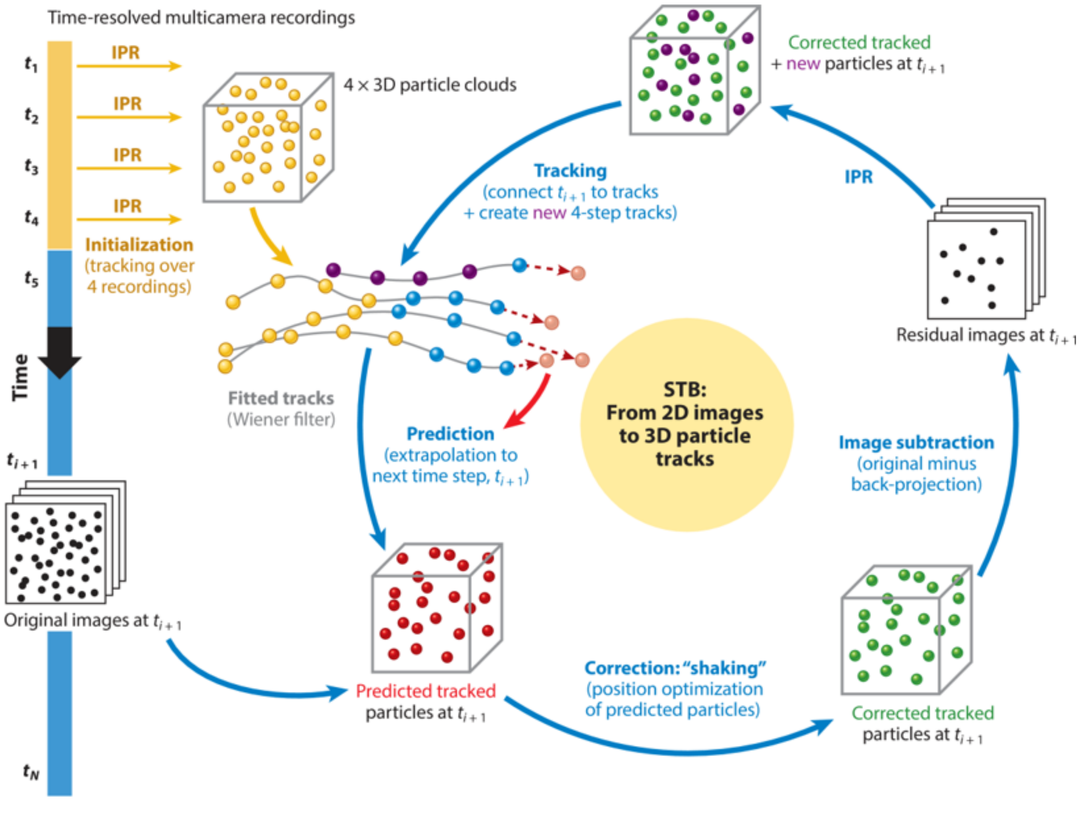


Figure 2.9: Schematic showcasing how adding the particle image shape is to the residual affects the fitting of the particle position ([Wieneke, 2012])

N_{init} time steps, typically taken as $N_{\text{init}} = 4$. Track candidates are identified in this initialisation phase. Matches amongst the N_{init} time steps can be determined in various manners such as applying a search radius around the particle position or using a predictor (e.g. mean velocity field measurements from other tools). The track candidates are checked for by applying velocity and



Schröder A, Schanz D. 2023
Annu. Rev. Fluid Mech. 55:511–40

Figure 2.10: Schematic explaining the Shake-The-Box approach. First, particles are detected using IPR for a number of N_{init} time steps. When tracked consecutively, they are used to form particle tracks. These tracks are subsequently used as a prior for the particle's position in the next time step. This estimate is corrected using a "shaking" approach similar to IPR. The detected particles are subtracted from the images. New particle tracks are formed and old tracks of which the particles have left the measurement domain are deleted. This is iteratively repeted until no more particles are detected. ([Schröder and Schanz, 2023])

acceleration thresholds and are passed to the box of tracked particles,

2. *Prediction of next particle position:* The track candidates can be used as predictors for the next time step. The position at the time $t_{N_{\text{init}}+1}$ can be extrapolated in different manners such as a predictive Wiener filter ([Wiener, 1949]) or a polynomial fit, e.g. a Savitzky-Golay filter ([Savitzky and Golay, 1964]). Schanz et al. [2016] reported that the Wiener filter provided more flexibility, easier application and a better reliability of the particle position prediction.
3. *Shaking the particles:* The initial particle positions are updated in a similar approach as outlined by Wieneke [2012] for IPR. The intensity of individual particles is re-projected onto residual images for each camera. Subsequently, the residual of equation (2.20) is minimised to obtain the new particle positions. The prediction of the particle position using trajectories are usually worse compared to predictions obtained through direct triangulation as done in IPR ([Schröder and Schanz, 2023]), however, most particles can still be correctly shifted to the correct position.
4. *Removing particles:* The particles which have already been tracked are subtracted from the images. The particles whose intensities fall below a certain threshold are deleted and the trajectories are stopped and removed from the system as to not be used in the next time step.
5. *Identifying new particles:* The remaining residual images contain particle candidates which can be triangulated using (advanced) IPR. Since most particles have been algorithmically removed from the measurement volume, the source density of the residual images is much lower benefitting the IPR. Once particle candidates are tracked for a number of time steps (usually equal to N_{init}), the trajectory is added to the system and used as a predictor. Tracking new particles can be done using a productor constructed from surrounding particle trajectories, by applying a Gaussian weighted average of the velocities ([Schanz et al., 2016]).
6. Convergent phase: After a number of time steps the system is considered converged once the number of tracked particles no longer changes significantly, and only particles which newly enter the measurement volume need to be triangulated. Once this state is reached, almost all of the other particles are accurately and correctly identified using the trajectories as predictors.

2.1.4. Post-processing volumetric velocimetry results

The volumetric velocimetry data can be used to detect Eulerian and, in the case of LPT, Lagrangian flow structures for flow analyses. Vorticity detection parameters such as the Q-criterion and lambda2-criterion can also be employed since the full three-dimensional velocity field is available. However, the applications of the complete flow field are not limited to determination of flow structures. LPT methods both provide a multitude of Lagrangian trajectories and a discretised velocity and acceleration field for each time step – and even between them through continuous filtering ([Schröder and Schanz, 2023]). Such data richness enables determination of two-point statistics up to the accuracy of the position determination ([Godbersen and Schröder, 2020]), which was previously limited by the imaging diameter of the particle ([C. Kähler et al., 2012]).

Bin-averaging of Lagrangian Particle Tracking results

Ensemble averaging is commonly used to convert Lagrangian velocimetry data to an Eulerian reference frame. An Eulerian reference frames permit solutions to be plotted from the perspective of the external spectator, which is how data is fluid dynamics is typically visualised. General quantities of interest are, amongst others, mean velocity (gradients), Reynolds stresses/fluctuations, strain rates and two-point correlations. The classic Reynolds decomposition of the instantaneous velocity $\vec{u}$ is given by

$$\vec{u}(x,y,z,t) = \overline{\vec{u}}(x,y,z) + \vec{u}'(x,y,z,t), \tag{2.21}$$

where $\overline{\vec{u}}$ is the mean velocity field and $\vec{u'}$ represents the fluctuations. Note how the mean is independent of time t. The classical ensemble averaging approach defines bins in which all fitted particles are averaged, allowing the recovery of turbulence statistics ([Kasagi and Nishino, 1991]). This is shown schematically in figure 2.11 for rectangular bins, which displays three particle tracks (*i.*, *ii.* and *iii.*) passing through several bins at different time steps (t_1 - t_4). All particles that fall inside a single bin are collected and averaged to obtain $\overline{\vec{u}}$. The top centre bin would for example use three particles, two from track *i.* at times t_2 and t_3 and one from track *ii.* at time t_1. The fluctuations $\vec{u'}$ and turbulence statistics

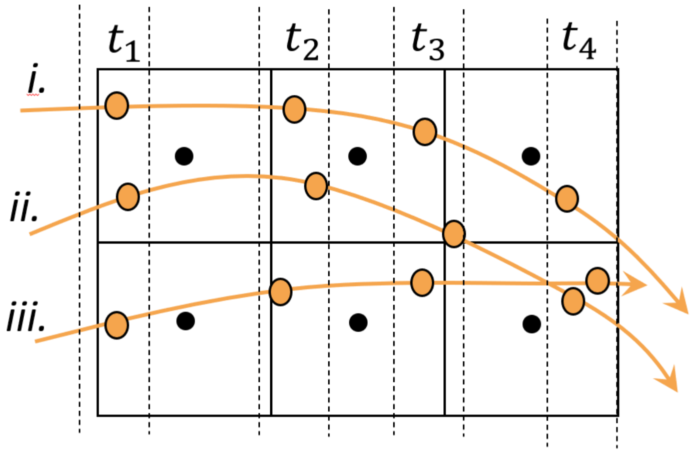


Figure 2.11: Schematic showing how particle tracks can pass through multiple bins at different times.

$\overline{u_i'u_j'}$ can be readily determined.

It is also possible to apply weighting to the particles based on their position. For example Agüí and Jiménez [1987] applied convolution with adaptive Gaussian windows to interpolate the tracers onto a uniform Eulerian grid. A different approach was taken by Agüera et al. [2016] who utilised polynomial fits inside bins based on the particle's position. This relied on fitting a quadratic velocity function of the form

$$u_n = a_0 + a_1\Delta x_n + a_2\Delta y_n + a_3\Delta z_n + a_4\Delta x_n^2 + a_5\Delta x_n\Delta y_n + a_6\Delta y_n^2 + a_7\Delta x_n\Delta z_n + a_8\Delta z_n^2 + a_9\Delta y_n\Delta z_n, \quad (2.22)$$

for each bin n, where $\Delta x_n, \Delta y_n$ and Δz_n are the distances to the grid point. This model fit is done for each of the three velocity components u, v and w. The reason to apply a polynomial fit over a top-hat filter (simple mean) or a Gaussian filter was illustrated through the sources of error due to residual velocity gradients, visually shown in figure 2.12. All three "filters" are applied in a one-dimensional problem. Given the right underlying functional, the polynomial fit is expected to yield the lowest residual fit. This argument can be bolstered by Taylor's theorem for approximation that any functional can be approximated in the vicinity by a polynomial fit using its derivatives ([Taylor, 1715]).

Recently, functional binning approaches have been proposed to use the full information of particle tracks, in the form of a parametric function over time, instead of the point-wise particle data ([Godbersen and Schröder, 2020]). In this functional approach the bin is represented by the spatial weighting function, part of which is dependent on the bin-dependent weighting

$$w_{\text{bin}}(\vec{x}) = \frac{1}{(2\pi)^{k/2}\sqrt{\det(\Sigma)}} \cdot \exp\left(-\frac{1}{2}\left(\vec{x} - \vec{\mu}^T\right)\Sigma^{-1}\left(\vec{x} - \vec{\mu}\right)\right), \quad (2.23)$$

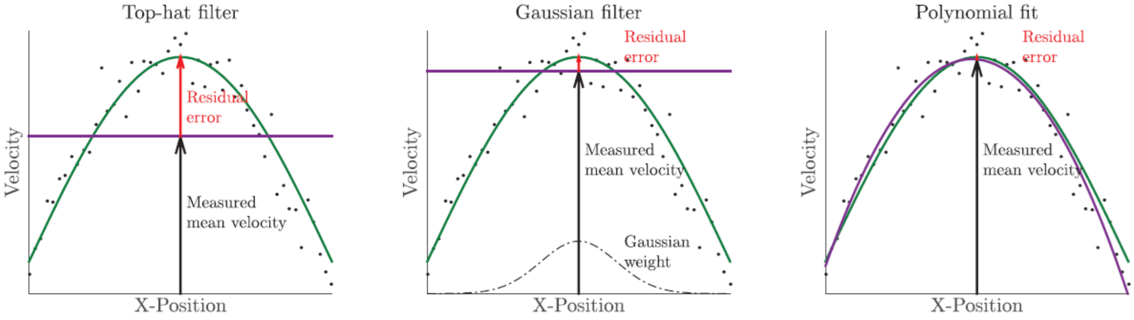


Figure 2.12: The source of residual error due to top-hat filter, Gaussian filter and a polynomial fit. ([Agüera et al., 2016])

a probability density function of a multivariate Gaussian distribution with dimension k, centered at the bin midpoint μ and the shape given by the co-variance matrix Σ. The choice of the co-variance matrix can be varied to adaptively shape bins and the bin centers μ need not be placed in a regular grid ([Godbersen and Schröder, 2020]). An additional temporal track-dependent weighting function $w_{\text{track}}(t)$ is multiplicatively added to the total weight function. This allows to introduce information on the quality of the data along the track. The complete weighting function is given by

$$w(t) = w_{\text{bin}}(\vec{p}(t)) \cdot w_{\text{track}}(t), \tag{2.24}$$

where the track position is parameterised, $\vec{x}(t) = \vec{p}(t) = (p_x(t), p_y(t), p_z(t))$. Finally, the mean velocity $\overrightarrow{u_{\text{bin}}}$ inside a bin is weighted using all N tracks following

$$\overrightarrow{u_{\text{bin}}} = \frac{\sum_{n=1}^{N} \overrightarrow{\overline{U_n}} \overline{W_n}}{\sum_{n=1}^{N} \overline{W_n}} = \frac{\sum_{n=1}^{N} \int_{t_0}^{t_1} \vec{u}_n(t) w_n(t) \mathrm{dt}}{\sum_{n=1}^{N} \int_{t_0}^{t_1} w_n(t) \mathrm{dt}}, \tag{2.25}$$

where the velocity along a given track is given by $\vec{u}_n(t)$.

To reduce the computational effort, a multi-stage filter process can be used to decrease the number of tracks that need to be integrated. The example by Godbersen and Schröder [2020] is shown in figure 2.13. In the first step, the tracks with midpoints inside a vicinity are chosen. This neighbouring distance is chosen generously as identification based on the midpoint will be an inaccurate approximation of the track's influence. In the second step, the minimum distance of the remaining tracks to the bin is then found by analytically solving a cubic equation. When the track falls inside a threshold proportional to the standard deviation, using the Gaussian bin weight function w_{bin} of equation (2.23), the track contributes to mean of the respective bin.

Pressure reconstruction from volumetric velocimetry data

Solving the (incompressible) momentum equation in classical fluid dynamics is generally complex (if not impossible) as it includes four unknowns namely three velocity components u, v and w and the pressure field p, whilst being non-linear in the velocity components. However, the availability of velocity field information has enlightened researchers to utilise the incompressible momentum equation, inputting experimental results for the complex, non-linear terms and leaving only the gradient of the pressure as an unknown:

$$\nabla p = -\rho\left(\frac{\partial \vec{u}}{\partial t} + [\vec{u} \cdot \nabla]\, \vec{u} - \nu \nabla^2 \vec{u}\right) \quad \text{or} \quad \nabla p = -\rho\left(\frac{\mathrm{D}\vec{u}}{\mathrm{D}t} - \nu \nabla^2 \vec{u}\right), \tag{2.26}$$

where the left equation is the Eulerian form and the right equation the Lagrangian. This can also be converted to a single elliptic partial differential equation for the pressure field p, known as the Poisson

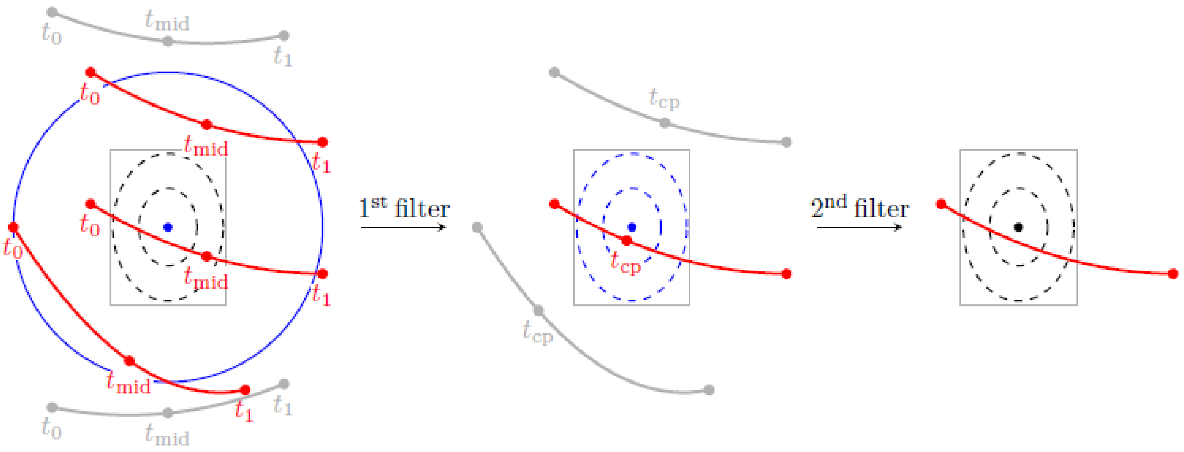


Figure 2.13: The approach to functional binning of particle tracks. In the first filter, tracks with a midpoint very far away are removed. In the second filter, the closest distance is computed and only the tracks which pass through the bin are chosen. ([Godbersen and Schröder, 2020])

pressure equation by taking the divergence of equation (2.26) ([van Oudheusden, 2013]). This equation is written in Eulerian form as:

$$\nabla^2 p = -\rho \nabla \cdot (\vec{u} \cdot \nabla)\,\vec{u} \tag{2.27}$$

Note how all of the right-hand side terms in equations (2.26) & (2.27) are measured in tomo-PIV and 3D LPT. The Lagrangian form of the pressure gradient could even be utilized along particle tracks with 3D LPT. These equations can be used in both unsteady approaches and time-averaged when a Reynolds decomposition is introduced with subsequent averaging. Early attempts to determine pressure from

PIV measurements focused on the Poisson equation of (2.27) in 2D. Gurka et al. [1999] applied a 2D Reynolds averaged version of equation (2.27) on PIV measurements of a pipe flow with water and a turbulent impinging jet flow. Instantaneous pressure computations of water waves from PIV were performed by Jakobsen et al. [1997] and Jensen et al. [2001]. Pressure inferral from PIV was extended by van Oudheusden [2008] to compressible flows which was validated with supersonic experimental data.

A range of suitable approaches to determine the pressure field from velocimetry data has been demonstrated ([van Gent et al., 2017]) to be suitable. The examples mentioned above solve the pressure Poisson equation (2.27). Another class of methods use the pressure gradient of equation (2.26) and integrate this along paths in the volume ([Liu and Katz, 2006]). The advantage of such an approach is the use of Lagrangian flow information from STB data, which is typically of higher spatial resolution ([van Gent et al., 2017]).

The main obstacle in the path-integral approach is to minimise the accumulation of errors along the path originating from spurious or under-resolved velocimetry data. An optimal path-choosing strategy defines the paths such that the region of large error is reached at the end. An example of such regions is near solid boundaries. This was implemented by Jux et al. [2020], who combined the approach with LPT measurements of a full-scale cyclist to determine the pressure coefficient at the model's surface. They also simplified the integration approach by employing Bernoulli's equation in regions of irrotational flow.

2.1.5. Data assimilation techniques

Data assimilation (DA) describes the mathematical discipline of combining theoretical models with experimental observations. Originating from the field of Numerical Weather Prediction (NWP) measurements of weather stations were combined with mathematical models to improve the reliability of weather prediction tools ([Navon, 2009]). The nonlinearity of the initial-value problem required an accurate determination of the initial conditions which explicitly take the dynamical evolution of the flow into account. This was accomplished by assimilating real-world measurements in time into a the numerical model ([Talagrand, 1997]).

In regards to instantaneous velocimetry measurements. (VIC)+ ([Leroux et al., 2014] and [Schneiders and Scarano, 2016]) is a well-established approach to improve both temporal and spatial resolution. VIC(+) is a variational data assimilation technique, which use an adjoint method to update a model with measurement values. VIC+ uses the vortex-in-cell description ([Christiansen, 1973]) as the model function, which is based on the inviscid, incompressible Navier-Stokes equations in terms of the vorticity and using the material derivative of the velocity. This works well in conjunction with the Lagrangian velocimetry technique, which directly determines this material derivative. The approach can improve the spatial resolution by a factor 4 ([Schneiders and Scarano, 2016]) and the time super-sampling approach has shown to be able to resolve a jet case sampled below the Nyquist frequency ([Schneiders et al., 2014]). Results of the VIC+ from Schneiders and Scarano are depicted here in figure 2.14. The Q-contours of a jet flow in water are reconstructed using linear interpolation, VIC (cost function without the term minimising $\mathrm{D}u/\mathrm{D}t$) and the full VIC+ method. The results clearly show how well VIC+ works.

Gesemann et al. [2016] proposed another optimisation method for time-resolved track-based results, i.e. from LPT. This method, termed *FlowFit*, use 3D B-splines to fit particle tracks by additionally penalising high frequency components and nonzero divergences. These methods were later extended to VIC# ([Jeon et al., 2018]) and VIC-TSA ([Scarano et al., 2022b]). The VIC# method implements three additional sub-cost functions based on the continuity and incompressible Navier-Stokes (NS) equations. Further, a multigrid method is proposed to accelerate the convergence and iteratively refine the grid size. VIC-TSA aimed at including more time-dependent particle track information from other time steps for the reconstruction at a single time step. This was done by performing the VIC+ approach

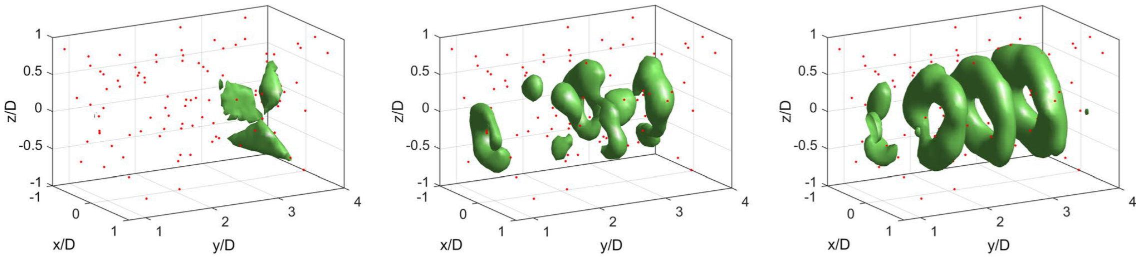


Figure 2.14: Results of vorticity reconstruction using (*left*) linear interpolation, (*middle*) VIC without $\mathrm{D}u/\mathrm{D}t$ term and (*right*) VIC+. ([Schneiders and Scarano, 2016])

for a set time interval. For a time t_0 a number of sampled positions $N_T + 1$ around t_0. For the solution at t_0 the respective cost function was then minimised over all time steps inside the sample. A similar, but simpler and computationally cheaper approach was taken by Jeon et al. [2019] and relied on the Taylor hypothesis. Particle information from neighbouring time steps (e.g. t_{k-1} and t_{k+1}) are projected onto the current time step t_k to increase the amount of available data. All methods have been show to handle Reynolds numbers of order $\mathcal{O}(10^5)$ and can provide slight improvements at the cost of a higher computational cost, higher complexity or more assumptions.

2.2. The boundary layer

This section outlines boundary layers in more depth and shortly explains common measurement techniques for near-wall flow properties, such as wall-shear stress, boundary layer thickness, moment thickness and shape factor. Section 2.2.1 reviews the description the boundary layer. The classical boundary layer parameters are outlined in section 2.2.2.

2.2.1. Boundary layer description

The presence of a solid object imposing zero-velocity induces a boundary layer over the surface where viscous effects dominate. For the incompressible flows discussed in this report, the characteristics of the boundary layer depend largely on the Reynolds number, the type of flow, namely laminar, turbulent or transitional. The Reynolds number defines the ratio of inertial forces to viscous forces and is given

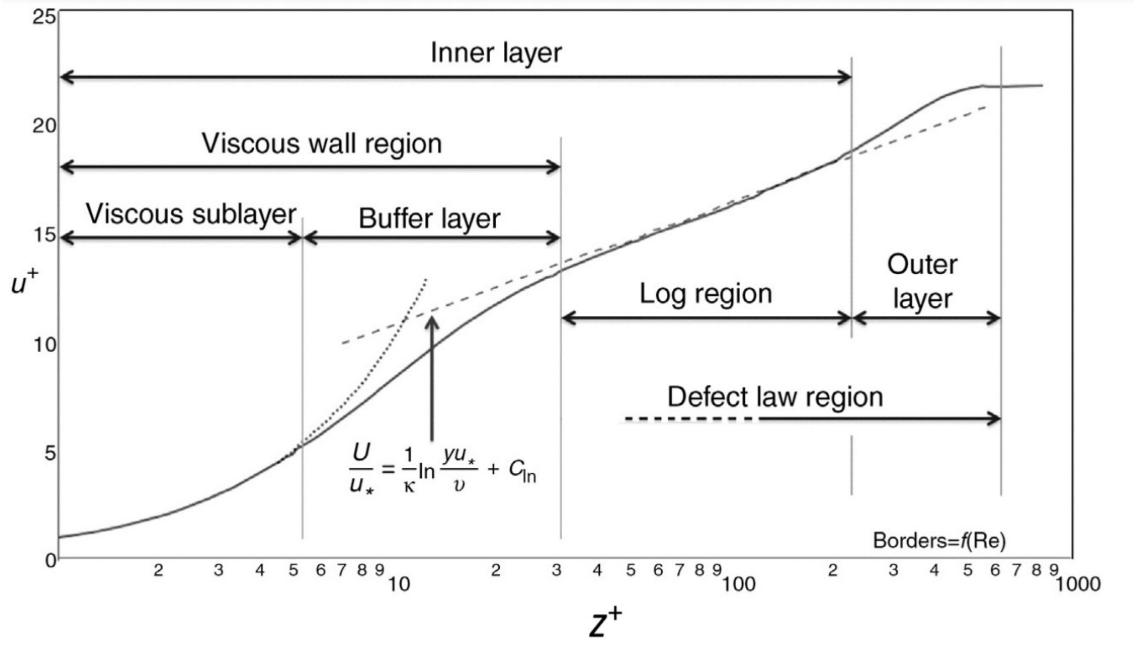


Figure 2.15: Schematic description of the velocity profile near the wall in a turbulent boundary layer, showing the various layers which exist at different scales.(recreated from [Ting, 2016])

by

$$Re = \frac{\rho \cdot U \cdot L}{\mu}, \tag{2.28}$$

where ρ is the density, U is the characteristic velocity, L is the characteristic length scale and μ is the dynamic viscosity.

The classic velocity profile in a **turbulent** boundary layer is depicted in figure 2.15. The velocity follows different ”laws” at different scales from the wall. Largely, the profile is divided into an inner layer and an outer layer. Note that both the velocity and the wall-normal distance are non-dimensionalised as

$$z^+ = \frac{z \cdot u_\tau}{\nu} \quad \text{and} \quad u^+ = \frac{U}{u_\tau}, \tag{2.29}$$

where z is the wall-normal distance, u_τ is the friction velocity (or shear velocity), ν is the kinematic viscosity and U is the velocity inside the boundary layer. In the following, z will always be used to indicate the wall-normal direction. Inside the inner layer, in the log region, the profile follows a logarithmic law, described as the law of the wall, here called the log-law ([Örlü and Vinuesa, 2020]). Very close to the wall in the viscous sublayer, the non-dimensionalised velocity and wall-normal distance exhibit a linear relation up to approximately $z^+ = 5$. In between these layers, the two profiles overlap which is termed the buffer layer.

2.2.2. Boundary layer parameters

Just as the velocity $\vec{u}$ can be Reynolds decomposed, the wall-shear-stress τ can also be decomposed into a mean component $\overline{\tau}$ and a fluctuating component τ'

$$\tau = \overline{\tau} + \tau'. \tag{2.30}$$

The wall-shear-stress components are defined using the Reynolds decomposed velocity as

$$\overline{\tau} = \mu \left. \frac{\partial \overline{\mathbf{u}}}{\partial \mathbf{n}} \right|_w \quad \text{and} \quad \tau' = \mu \left. \frac{\partial \mathbf{u}'}{\partial \mathbf{n}} \right|_w, \tag{2.31}$$

where the subscript w denotes that the the wall-normal gradient is evaluated at the wall. The skin friction can also be inferred from the velocity profile instead of the velocity gradient. If measurements are available inside the viscous sublayer, it is possible to directly determine the wall-shear stress due to the linearity between y^+ and u^+ as given by equation (2.29). For a simple $1\,\mathrm{m}$ flat plate flow at $Re = 10,000$, this would already yield a viscous sublayer size of $1\,\mathrm{mm}$ which is typically a limitation for most experimental techniques. A common other approach is the assumption of wall models, such as the Clauser chart/plot method ([Clauser, 1954]). This approach relies on fitting the logarithmic law of equation (2.32) to the measured velocity in the buffer layer:

$$u^+ = \frac{1}{\kappa} \ln\left(y^+\right) + B, \tag{2.32}$$

where κ is the von Kármán coefficient and B is the log-law intercept. Typically, constant values of the von Kármán coefficient and log-law intercept are assumed. That approach is often contested due to the variation of the constants in literature and the subjectivity in defining the limits of the log region ([Tavoularis, 2005]).

The boundary layer is traditionally characterised by the boundary layer thickness δ^*, the momentum thickness θ and the shape factor H, defined as

$$\delta^* = \int_0^\infty \left(1 - \frac{\rho u}{\rho_\infty u_\infty}\right) \mathrm{d}y, \tag{2.33}$$

$$\theta = \int_0^\infty \frac{\rho u}{\rho_\infty u_\infty} \left(1 - \frac{u}{u_\infty}\right) \mathrm{d}y, \quad \text{and}$$

$$H = \frac{\delta^*}{\theta}, \tag{2.34}$$

where the subscript ∞ denotes a parameter in the free-stream. These are important parameters when quantitatively comparing different boundary layer flows or investigating the dynamical state of the flow, i.e. laminar, transitional or turbulent. For incompressible flows (ρ is constant), the density drops out of the equations and only information on the velocity profile is required. Unlike wall-shear-stress measurements, however, determining the boundary layer parameters requires complete information on the velocity variation throughout the boundary layer.

2.3. Object registration

In a recent publication, Hendriksen [2024] demonstrated an approach to register objects in (large-scale) 3D-PIV experiments, where the velocity field and object position are obtained using the same set of cameras. The objective was to enhance the Shake-The-Box algorithm by including the distribution of the camera coverage rank R_C which requires information on the model position. R_C is defined as the number of cameras having optical access to a specific region of the measurement volume, which can be obtained through a ray-casting approach. These R_C maps provide information on which camera can physically detect a particle in which region.

An additional benefit of this approach is accurate information on the object position. Since the wall-position is of significant importance for wall-shear-stress estimates ([Titchener et al., 2015]), this development opens the door towards near-wall surface fluid dynamic evaluations with large-scale PIV. The representation space of the model is referred to as the model domain $\Omega_{\mathcal{M}}$. The model geometry is provided through a digital computer-aided design (CAD) model. The goal of the object registration is to seek a transformation matrix ${}_{\mathcal{M}}\mathbf{T}_{\mathcal{F}}$ that aligns the coordinate system of $\Omega_{\mathcal{M}}$ with the coordinate system of the fluid domain $\Omega_{\mathcal{F}}$, ${}_{\mathcal{M}}\mathbf{T}_{\mathcal{F}} : \Omega_{\mathcal{M}} \rightarrow \Omega_{\mathcal{F}}$. In other words, to place the CAD model in the correct position with respect to the fluid measurements. More detailed information on the data representation of the CAD model is available in appendix C.

The experimental process is denoted in subsection 2.3.1. Different approaches can be used to mark the surface, but the point-to-surface based registration is included here. This approach does not require adapting the physical model to include surface markers at predetermined positions. The procedure is explained in subsection 2.3.2. Lastly, the uncertainty of the object registration and its relation to uncertainty on boundary layer parameters is included in subsection 2.3.3.

2.3.1. Experimental process

The object registration relies on fitting a set of physical measurement points which represent the outer surface of the model to the CAD model. The process of obtaining the physical measurements points is referred to as *surface marking*. Several approaches to surface marking are compared by Hendriksen [2024]:

1. Incorporated markers; a set of markers are integrated into the model, such that these are individually distinguishable.
2. Single laser-beam pointer; a low-power laser pointer is employed to optically mark a single point on the model.
3. Multi-point laser beam pointer; a low-power laser pointer is equipped with an optical pattern module to split the laser beam into multiple points marking the model.

Figure 2.16 shows the point cloud results of each of the three surface marking approaches applied to a scaled cyclist model.

The point reflections are triangulated using the IPR method by Wieneke [2012]. To overcome the issue that not all cameras have optical access to the same set of markers, markers are registered with partitioned groups of the cameras ([Hysa et al., 2023]). Groups of three cameras can be used at low densities of the marker points, such as with approach **1.** and **3.**, whereas partitions of two cameras can be used for single point reconstructions.

2.3.2. Point-to-surface based object registration

The surface marking approach returns a point cloud which represents a point-wise discretisation of the physical model's surface. The registration algorithm must ensure a robust an accurate fitting of the digital CAD model to this point cloud, shown for a cube in figure 2.17. Hendriksen [2024] shortly describes and implements point-to-point registration and point-to-plane registration, which are variations of the iterative closest point (ICP) algorithm ([Rusinkiewicz and Levoy, 2001]). A point-to-plane registration is most applicable in the general case with static, non-moving models as it does not require the integration of surface markers into both the digital CAD model and corresponding physical model. Therefore, this method of registration is shortly outlined here and further discussed in section 5.4.

Given that the model geometry is known a priori, the output is a transformation matrix (rotation and translation), which moves the CAD closest to all points. A registration is typically started with an initial guess to bring the source close enough to the target. Then, the registration process commences which consists of six stages:

1. *Selection*, where the sets of points to be matched are defined.
2. *Matching*, where the points in the source set are linked to the points in the target set.
3. *Weighting*, where each point-pair is given a corresponding weight.
4. *Rejecting*, where certain point-pairs are not considered to partake in the matching.
5. *Objective function computing*, which quantifies the mismatch between the source and the target.
6. *Minimising*, where a transformation matrix $\mathbf{T}$ is sought such that the objective function is minimised.

The ICP methods considered here is the point-to-plane registration. In both cases the source points are the triangulated measurement points, whereas the target is the CAD-model. Note that once a transformation from the source to the target is found, this can be inverted to obtain the reverse transformation. The *Selection* step determines which source and target points are used. The target point set is selected by projecting the source point onto the CAD model. The selection of the source points will include all of the available measurement points, with large outliers removed.

Matching the points in both source and target is not a needed step for the point-to-plane approach. Similarly, no additional weighting is given to certain points in the source mesh. Points in the source mesh are rejected based on a threshold search radius. The objective function is defined as follows:

$$\epsilon(\mathbf{T}) = \sum_{\mathbf{s},\mathbf{t}\in\mathcal{K}} \left([\mathbf{Ts} - \mathbf{t}] \cdot \mathbf{d_n}\right)^2 \tag{2.35}$$

The error in equation (2.35) is minimised iteratively, where at step k a set of corresponding points is generated using the transformation matrix $\mathbf{T}_{k-1}$ and a new transformation $\mathbf{T}_k$ is found which minimises $\epsilon(\mathcal{T}_k)$. This process is repeated until the difference in the error metric between $k-1$ and k falls below a certain threshold. The resulting transformation matrix includes both a translation and rotation of the source points to match the target points.

2.3.3. Uncertainty of the registration

The uncertainty in registration is a significant aspect to consider when calculating the wall-shear-stresses and boundary layer parameters owing to the propagation of errors. As stated by Titchener et al. [2015] the errors ϵ on the displacement thickness δ^*, momentum thickness θ and shape factor H

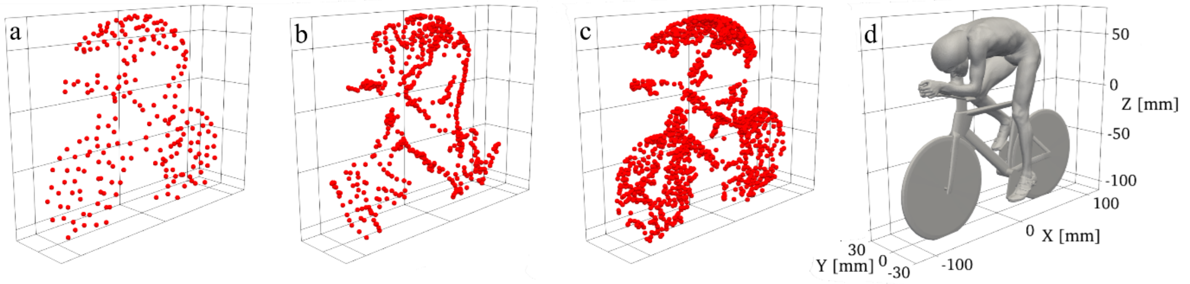


Figure 2.16: Reconstructed surface markers from a. integrated reflective markers, b. single laser pointer and c. a laser pointer with optical pattern module. The CAD model is shown in d. [Reprint from Hendriksen, 2024]

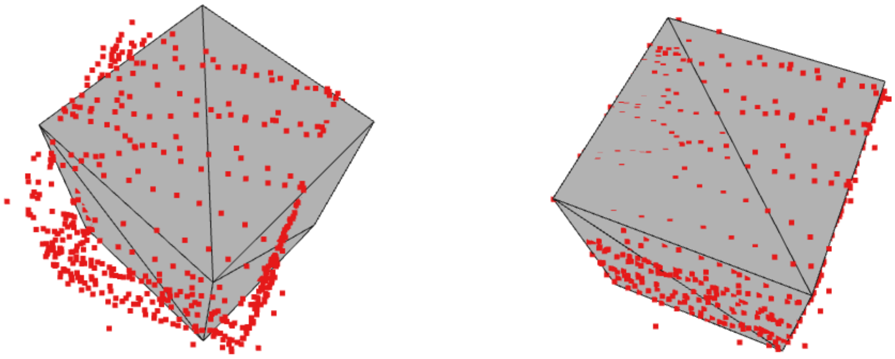


Figure 2.17: Example of object registration which corrects an initial position of the cube to fitting it to the red surface markers.

for attached turbulent boundary layers approximately follow:

$$\epsilon_{\delta^*} = 5.9 \cdot \frac{\Delta y_0}{\delta} \tag{2.36}$$

$$\epsilon_{\theta} = 1.3 \cdot \frac{\Delta y_0}{\delta} \tag{2.37}$$

$$\epsilon_{H} = 4.5 \cdot \frac{\Delta y_0}{\delta} \tag{2.38}$$

Placing that into perspective, to achieve an error of less than 5% on calculating the displacement thickness for a boundary layer that is $20\,\mathrm{mm}$ thick, the uncertainty of the wall position must be within $0.17\,\mathrm{mm}$. For the momentum thickness this must be within $0.77\,\mathrm{mm}$ and for the shape factor $0.22\,\mathrm{mm}$. Naturally, this scales with the size of the boundary layer and these estimates are valid for attached turbulent boundary layers, which exhibit a steep gradient at the wall.

The uncertainty of the object registration is given by the mean absolute distance, $\delta_{\mathbf{s},\mathbf{t}}$, defined as the mean distance between the triangulated source points $\mathbf{s}$ and their projection on the target CAD model $\mathbf{t}$. The results of the experiment exhibit an uncertainty in the order of $\mathcal{O}(10^{-1}\,\mathrm{mm})$ for models of $\mathcal{O}(10^{2}\,\mathrm{mm})$, amounting sub-1% accuracy on the position. The true effect of wall position offset on the object registration will vary over the model due to the dependence on not only the boundary layer thickness, but also the flow regime. For example, the flow around the cube consists of several separation and reattachment points, where new boundary layers are formed locally.

3

Literature survey on near-wall flows of generic 3D objects

Particle image velocimetry (PIV) and Lagrangian particle tracking (LPT) have matured into the preferred method for turbulent volumetric measurements ([Raffel et al., 2018]). Both techniques have demonstrated the capability to survey the three-dimensional characteristics of turbulent flows. Recent advancements have proposed the use of these volumetric velocimetry techniques to infer the surface pressure distribution on generic objects. Coupled with the rich availability of volumetric data ([Schröder and Schanz, 2023]) and developments to more accurately determine model positions relative to the vector positions of the fluid velocity field ([Hendriksen, 2024]), this motivates a study on inferring near-surface fluid dynamics using PIV and LPT.

To define the gap in research, recent literature is reviewed in this chapter, focused on near-wall flows around 3D objects. The goal is to formulate a research objective and divide it into research questions, which are tackled in the remainder of this thesis report. As a starting point, endeavours using direct methods for surface flows are reviewed in section 3.1. Firstly, common approaches to near-wall flow measurements are reviewed in section 3.1. Next, the volumetric velocimetry techniques, tomographic PIV (tomo-PIV) and three-dimensional LPT (3D LPT), and their applications to near-surface flows are reviewed in section 3.2. Then, to understand how fluid measurements and physical boundaries are merged in numerical schemes and the endeavours with data assimilation, section 3.3 evaluates the field of data assimilation with a focus on the implementation of boundary conditions. Lastly, the research objective is stated in section 3.4. This section also contains the research questions which are derived from the objective.

3.1. Direct measurements of near-wall flows

The wall-tangential skin friction and the wall-normal pressure are the only two sources that nature can use to communicate aerodynamic forces and moments ([Anderson, 2017]). Measuring the distribution of both quantities over objects is vital in most (industrial) applications. Subsection 2.1.4 showed that the wall-pressure could be reconstructed using PIV or LPT. Now, what remains is how the skin friction can be determined.

This section reviews direct measurement techniques for near-wall flows with a focus on wall-shear stress. Direct techniques are often favoured over indirect techniques as they require less assumptions on the wall-shear stress relations [Örlü and Vinuesa, 2020]. The two most common approaches are oil flow interferometry and floating elements [Vinuesa and Örlü, 2017]. Firstly, the oil film approach is discussed in subsection 3.1.1. Subsection 3.1.2 then reviews literature which employs floating elements.

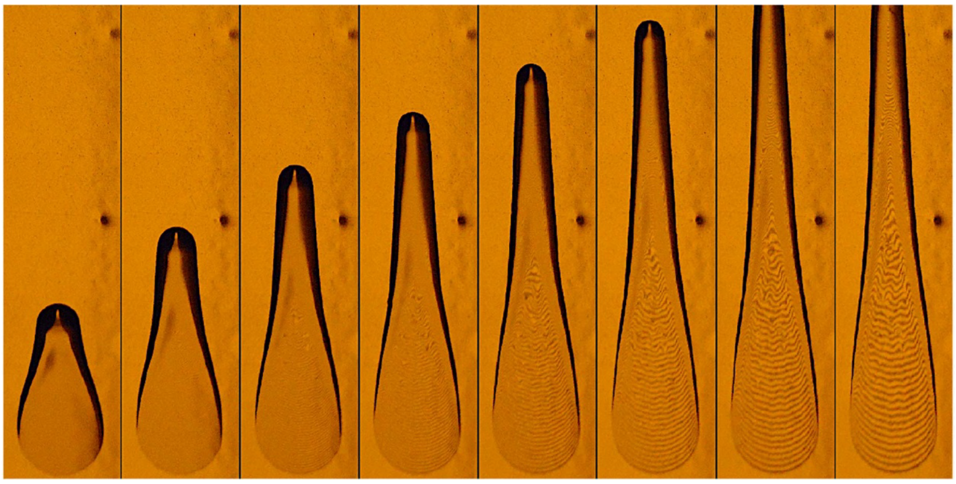
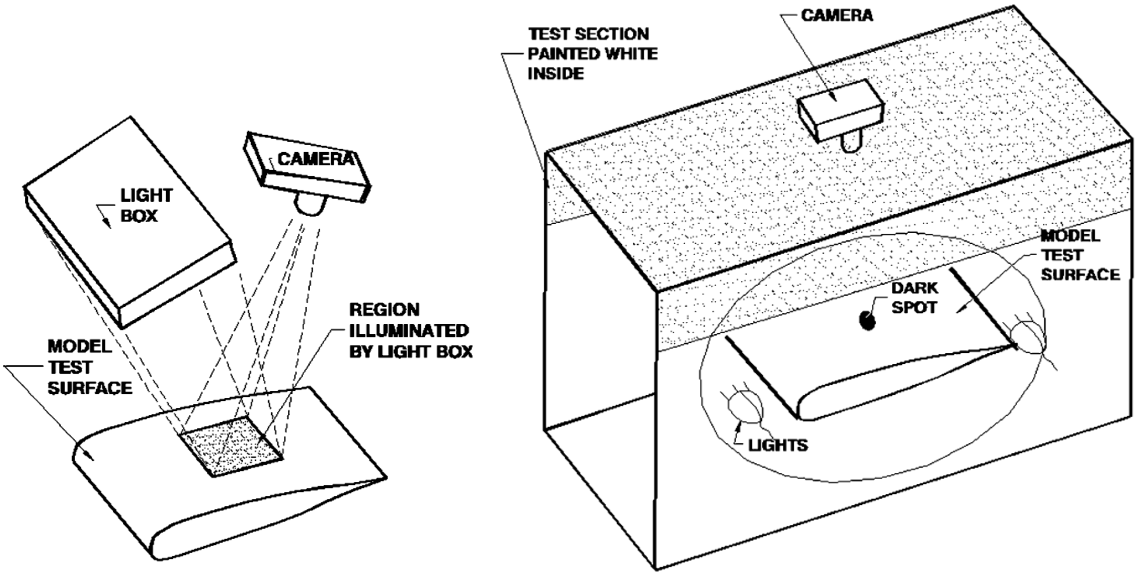


Figure 3.1: Example of how an oil drop develops to for a film, which indicates the interferometric pattern that can be used to determine the thickness. The images progress in time from left to right and the flow comes in from the bottom. [Örlü et al., 2010]

3.1.1. Oil Film interferometry

Oil flow interferometry (OFI) is one of the few techniques which directly measure time-averaged skin friction. It was originally proposed by Tanner and Blows [1976] and was later adopted for wall-bounded turbulence research [Monson et al., 1993]. One or more drops of oil are placed on a surface, which gets displaced by the incoming flow. The oil drop is stretched and forms a thin film which is typically several wavelengths of light thick. The thickness of the oil film and the wall-shear-stress can be correlated using an interferometric pattern, for example shown in figure 3.1. Monochromatic light is shone on the film which gets reflected by the surface of the oil and by the bottom wall. The height of the film at the kth fringe is given by [Fernholz et al., 1996]

$$h_k = h_0 + k\Delta h, \text{ with } \Delta h = \frac{\lambda}{2\sqrt{[n^2 - \sin^2 \alpha]}}, \tag{3.1}$$



(a) (b)

Figure 3.2: Two OFI measurement setups varying from very simple (a) using only a light box to very complex (b) where the wind tunnel walls have been painted white to increase light reflection [Driver, 2003].

where h_0 is the height of the first fringe at the edge of the film, where the drops starts, Δh is the difference in height between consecutive fringes, λ is the wavelength of the monochromatic light, n denotes the refractive index and α is the viewing angle of the observer. The slope of the oil film can be used to determine the wall-shear stress from the development of the film height over time, while also accounting for effects such as pressure gradients, surface tension and gravity [[Fernholz et al., 1996]]. Driver [2003] reviewed several approaches that make use of OFI and applied these to models in large wind tunnels from NASA. Some of these setups are shown in figure 3.2, ranging from a compact setup using only a light box and a small illuminated region, subfigure 3.2a, to a large setup where the walls of the wind tunnel test section were painted white, subfigure 3.2b. The technique required minimal setup experience and costs and attained accurate measurements with errors less than 5%, though one of the major drawbacks was the productivity of the techniques.

In recent times, the accuracy of OFI has been improved. Vinuesa et al. [2014] applied OFI to measure skin frictions of a turbulent channel flow at a friction Reynolds number up to $\mathrm{Re}_\tau = 900$ to assess the developing flow. They estimated the uncertainties of the friction velocity and wall shear stress to be respectively 0.58% and 0.85%. Further understanding of the technique was given by Segalini et al. [2015]. They provide a detailed analysis of perturbation effects on the different boundary layer regions caused by an oil film to establish corrections to the classical OFI interferometry method. They conclude that the accuracy of OFI with their proposed corrections for determining the friction velocity is similarly around 1%.

OFI is the method of choice for measuring mean wall-shear-stress, but it can not be used to determine the instantaneous wall-shear stress. And even though corrections have been proposed, the nature of the technique means that OFI is still affected by curvature of the surface, surface tension of the oil, the pressure gradient and gravity [Squire, 1961]. It would therefore be infeasible to apply this technique to highly curved surfaces or even inverted surfaces. And since the film must be captured by a camera, the method still requires optical access.

3.1.2. Floating elements

In experimental aerodynamics, floating elements (FE) have become a vital tool for measuring wall-shear stresses and skin friction. These devices consist of small, isolated sections of a surface that are allowed to ”float” or move in response to shear forces exerted by the fluid flow. By measuring the displacement or force on these elements, one can directly quantify the wall-shear stress [Vinuesa and Örlü, 2017]. A skectch of a floating element is depicted in figure 3.3. Floating elements have been used since the early 1900s [Kempf, 1929] and are still used today.

Recent advancements have significantly improved the sensitivity and accuracy of floating elements. The design by Ferreira et al. [2018] for example is capable of accurately measuring rough-wall boundary layer flows. They demonstrated an accuracy to within 2% of comparable hot-wire anemometry measurements. This was later extended to improve the accuracy of FE in low speed boundary layer flows using an analytical model [Aguiar Ferreira et al., 2024].

Typically, larger FE are desired as these yield a higher signal-to-noise ratio [Örlü and Vinuesa, 2020] and are therefore more suited to measurements of the atmospheric boundary layer. A recent work by Baars et al. [2016] assessed the accuracy of modern FE devices and found good agreement with theory for the case of a 3x$1\,\mathrm{m}$ device. Nonetheless, recent works have attempted to decrease the size of FE devices ([Naughton and Sheplak, 2002]), classified as microelectromechanical systems (MEMS). If such a device would ever reach the accuracy of OFI, it could become extremely versatile as it can be essentially incorporated in any surface without any need of for example optical access. However, unless millions of miniature MEMS were to be used Floating elements could also not ever reach a point where they could measure the complete skin friction distribution around a generic object such as a

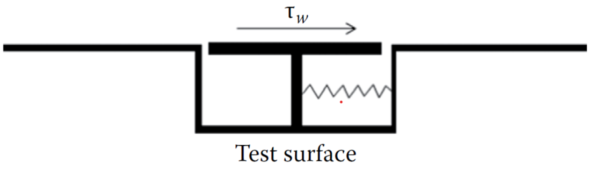


Figure 3.3: Sketch of a floating element balance Vinuesa and Örlü, 2017

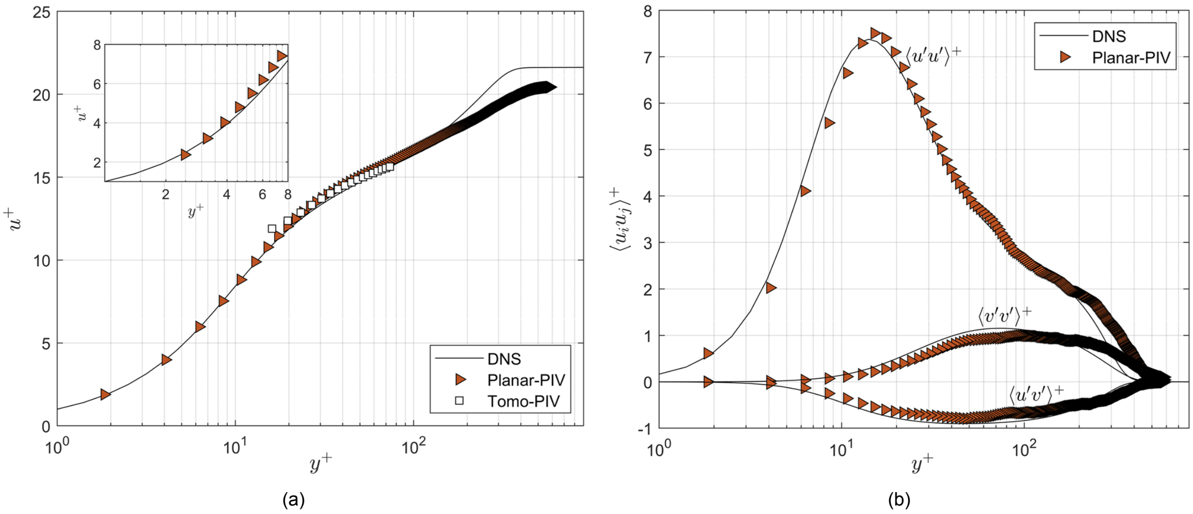


Figure 3.4: Results showing (A) mean velocity profile and (b) Reynolds stress profile inside a flat plate boundary layer obtained via planar PIV and tomographic PIV [Kempaiah et al., 2020].

cyclist.

3.2. Near-wall flows with particle image velocimetry and Lagrangian particle tracking

Section 3.1 reviewed the techniques for wall-shear stress measurements in the form of OFI and floating elements. Another category of measurement techniques for wall-shear stresses are the indirect methods. These rely on the relation between velocity and wall-shear stress, given by equation (2.31). PIV and LPT are herein considered as several works have already demonstrated the use of volumetric measurement techniques to near-surface flows.

This section reviews the works that employ tomo-PIV and 3D LPT for near-surface flows to determine the state-of-the-art. Most work has focused on (semi-)flat plate boundary layers and these are discussed in subsection 3.2.1. Few works have been presented which attempt to broaden the perspective to complete objects and these are reviewed in subsection 3.2.2.

3.2.1. Flat plate flows

Particle image and tracking velocimetry can yield high resolution velocity measurements, from which the wall-shear stress can be directly determined. As explained in section 2.2, this requires that the

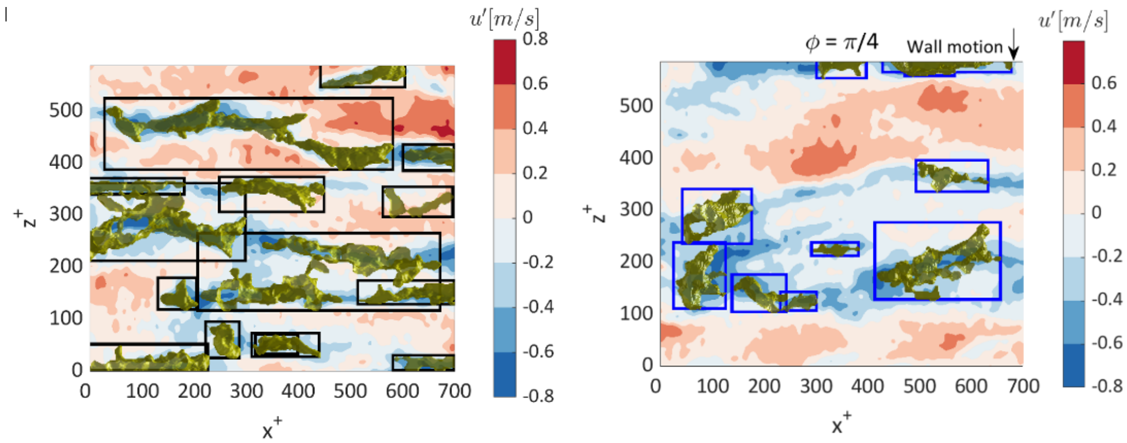


Figure 3.5: Plots of the instantaneous streamwise velocity with iso-surfaces of ejection events (parameterised as $u'v' < 0$) for (left) the stationary plate and (right) the oscillating plate, as reported by Scarano et al. [2022a].

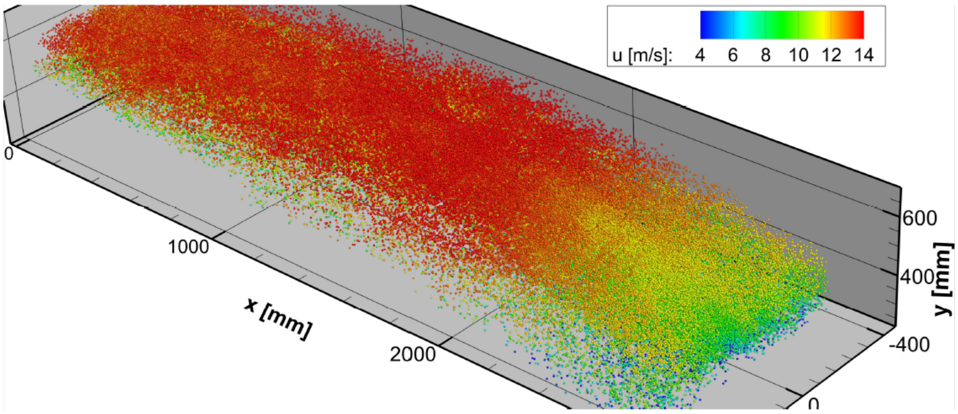


Figure 3.6: Exemplary snapshot of STB evaluation in the turbulent boundary layer of an adverse pressure gradient flow at free-stream velocity of $14\,\mathrm{m\,s^{-1}}$. (Reproduced from [Schanz et al., 2019]

scale of the PIV-experiment is compatible with the scale of the viscous sublayer such that velocity can be measured up to the viscous sublayer [C. J. Kähler, 2004]. For example, the results in figure 3.4 are excellent illustrations by Kempaiah et al. [2020] (see also [Scarano et al., 2022a]) of how PIV measurements can be used to directly obtain the wall shear stress. A light sheet was placed perpendicular to the wall, such that the velocity along the wall-normal coordinate could be measured in a single plane/. The non-dimensionalised velocity U^+ is then plotted against the non-dimensionalised wall-normal coordinate y^+. Note how in the current thesis report, the $z-$coordinate is used to denote the wall-normal direction. Planar PIV measurements are available up to $y^+ = 1$, which is well inside the viscous sublayer (i.e. $y^+ < 5$). A linear relation exists between u^+ and y^+ using equation (2.29) as

$$u^+ = y^+ \text{ or } \frac{U}{u_\tau} = \frac{z \cdot u_\tau}{\nu}. \tag{3.2}$$

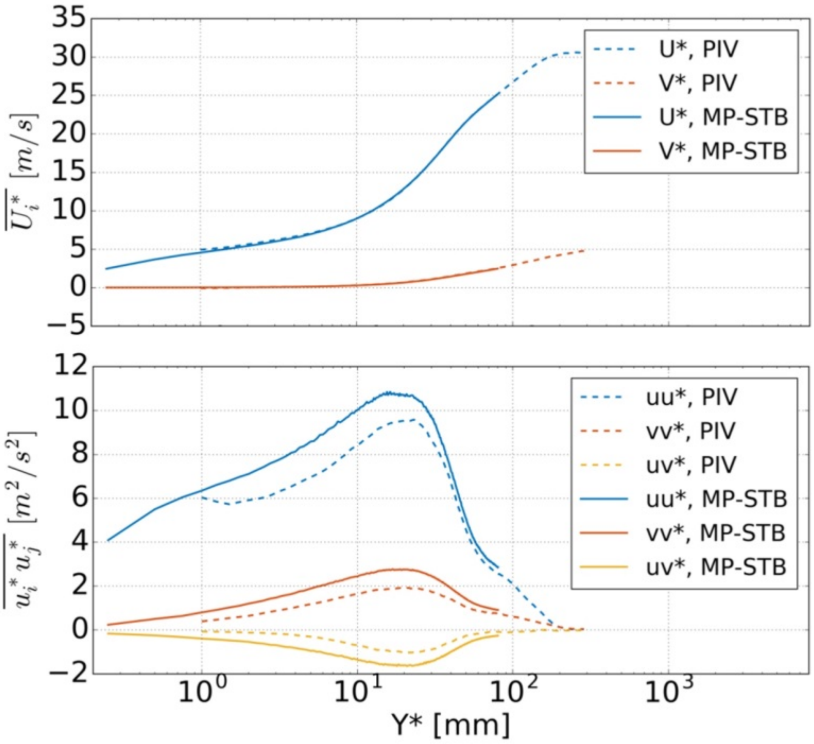


Figure 3.7: Mean velocity profiles and Reynolds stress profiles in a turbulent boundary layer after binning multi-pulse STB results, compared against results from planar PIV, at edge velocity of $U_e = 35\,\mathrm{m\,s^{-1}}$. [Schröder et al., 2018]

The measurements can be used directly to determine the friction velocity u_τ. u_τ can in turn be used to determine the wall-shear stress via [White, 2006]

$$\tau_w = u_\tau^2 \cdot \rho. \tag{3.3}$$

The results of Kempaiah et al. [2020] also highlight how tomo-PIV suffers more from light reflections and reduction of light intensity than two-component planar PIV. Results were obtained within the log-layer region of the flow, which still agrees well with the results from planar PIV. The measurement volume was $7{\times}6{\times}0.6\,\mathrm{cm}^3$ and a vector spacing of $0.25\,\mathrm{mm}$ was achieved using an overlap of 75%. This yielded velocity measurements in the buffer layer ($y^+ = 20 - 70$) of the boundary layer. Due to the low resolution the authors resorted to planar PIV to determine the skin friction and used the tomo-PIV results for vorticity analysis. As displayed in figure 3.5, the tomo-PIV investigation did visualise the instantaneous streamwise velocity and iso-surface of ejection events for both a stationary and oscillating plate surface, providing valuable insight into the effect that oscillating walls have on the near-surface fluid structures. When it comes to near-wall resulotion, the dynamic spatial range (DSR) may actually be improved by particle tracking techniques compared to correlation-based techniques since a large number of particles need to be captured at a single time for a reliable correlation [C. J. Kähler et al., 2012].

The advent of 3D LPT has lead Schröder et al. [2018] to investigate the turbulent boundary layer using regular and multi-pulse STB. The authors chose to utilise DHES particles with a mean diameter of $1\,\mu\mathrm{m}$. They implemented specialised wall-seeding devices to introduce significantly more particles near the walls and achieved high resolution velocity and turbulence statistics data up to the wall in several unconnnected regions. The particles were tracked in small volumes which mainly extended along the wall-normal direction. They adapted the multi-pulse STB approach [Novara et al., 2019], which allowed creating bins with very high wall-normal resolution, $0.07\,\mathrm{mm}$ or $2\,\mathrm{px}$ or about 5 viscous units. Results of their experiment are included here in figure 3.7. Velocity is obtained. This result may have been expanded to determine skin friction distributions, for example by fitting wall models [Rodriguez et al., 2015]. However, only velocity profiles are reported.

In a similar manner Schanz et al. [2019] measured the turbulent boundary layer flow of a sequentially zero-pressure, favourable and adverse pressure gradient. They used HFSB with a mean diameter of $300\,\mu\mathrm{m}$ in a single measurement volume of $450\,\mathrm{L}$ spanning up to $3\,\mathrm{m}$. Schanz et al. tracked hundreds of thousands bubbles over the length of the measurement volume which are shown in figure 3.6. The results showed large-scale turbulent structures in the outer turbulent boundary layer region and even medium-scale structures were resolved. However, close to the wall the undersampling of flow gradients due to relatively large observation space and bubble diameters was apparent and the small scales could not be resolved.

3.2.2. Near-wall flows around generic objects

The examples outlined above demonstrate the applicability of PIV and in particular LPT to investigate near-wall solutions. The requirement of optical access and adequate lighting, however, often limit the application of near-wall PIV to flat plate boundary layers, with occasionally a minor adverse or favourable pressure gradient. However, several works have afforded the same problem of applying near-wall PIV to generic objects. One such example is by Drouin et al. [2002], who applied near-wall planar PIV to the rear of an Ahmed body. This is possible since an Ahmed body has planar surfaces at the rear and light sheets could be oriented tangentially. Similarly, Depardon et al. [2005] used near-wall PIV to determine the velocity distribution very close to the walls of a surface-mounted cube using parallel light sheets. Results of the velocity magnitude and skin friction lines (partly using oil flows) are shown in figures 3.8 and 3.9, respectively. The velocity was measured at a distance of $0.5\,\mathrm{mm}$ (0.8% of the cube height) from the walls. They demonstrated that these near-wall velocity streamlines can be used to infer the skin friction lines, though this approach was only used to infer skin friction topologies, not magnitude distributions.

A data-rich example showcasing the full extent of 3D LPT with STB and multi-pulse STB was orchestrated by Schröder et al. [2020]. They used seven high-speed cameras to capture the 3D velocity field around a cube of $1\,\mathrm{cm}^3$ mounted to a flat surface gathering up to 55,000 images. This experiment was performed inside a water tunnel at inflow velocities up to $0.8\,\mathrm{m/s}$ yielding a Reynolds number of approximately 8×10^3 based on the cube length scale. The striking experimental results are in part due to the high temporal and spatial resolution that was achieved, showcasing a state-of-the-art implementation of 3D LPT. Such an experiment is, however, extraordinary and, in most cases, simply unattainable. The regions occluded by the cube were marked by hand to define the particle search area for each camera, which is only possible due to the simple geometry. Schröder et al. do obtain velocity information well within the viscous sublayer, but they do not provide any indication on the wall-shear-stress and only comment ”*For this specific case [..] the displayed flow structure at the wall is a good and direct approximation of the mean skin friction velocity vector distribution. [..] Access to a direct estimation of the surface drag based on the flow field around an obstacle is considered a very interesting novelty for aerodynamic investigations.*” [Schröder et al., 2020].

Lastly, the results by Jux et al. [2020] are highlighted, using what has been termed ”robotic PIV”[1]. The flow around a full-scale cyclist was measured using the co-axial volumetric velocimetry configuration proposed by Schneiders et al. [2018] and a robotic arm. Jux et al. subsequently used the velocity measurements and the pressure gradient-path integral approach to determine time-averaged wall pressures on the surface of the cyclist. However, the results of interest here are from an earlier publication [Jux et al., 2018], where they used the same dataset determine the near-surface time-averaged velocity field around the cyclist, displayed in figure 3.11. The authors first created a dilated, offset mesh from the contour of the cyclist. Due to the high availability of data up to the model and relatively small bins, only a small dilation of $5\,\mathrm{mm}$ or 0.5% of the characteristic model length scale was needed. This brought the velocity contour inside the domain of binned data points and it could be interpolated onto the dilated mesh. The authors were thereby able to display near-surface velocity streamlines around a generic cyclist body. This approach is thus seen as the state-of-the-art and will be the starting point of comparison for the proposed family of methods. Therefore, more attention is given to this approach in section 5.1. The near-surface velocity streamlines provide an indication of the skin friction lines, akin to the results of Drouin et al. [2002] and Depardon et al. [2005]. Schneiders et al. [2018] went one step further with these results and displayed the skin friction topology around different parts of the cyclist. For example, the topology at the lower back of the cyclist is included in figure 3.12. They identified critical points, such as nodes, spiral and saddle points. However, as was the case with the results by Depardon et al. [2005], the step to the wall-shear stress had not been taken. This may have been due

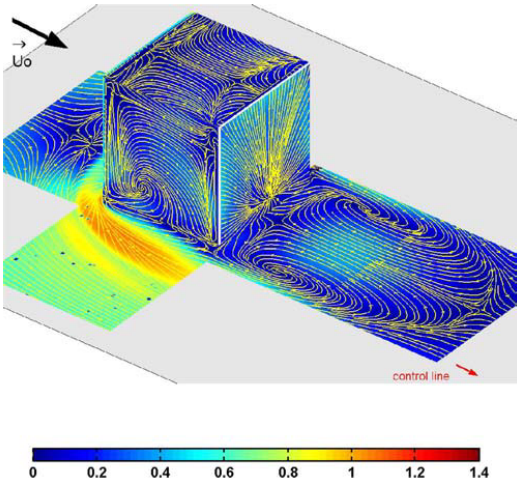
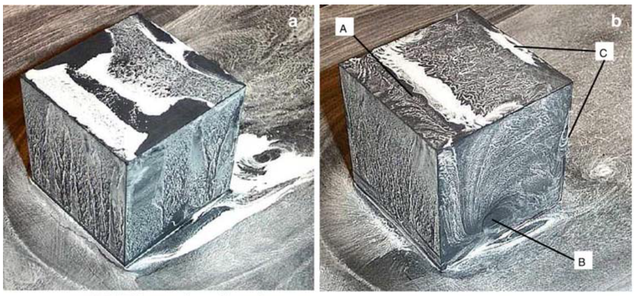
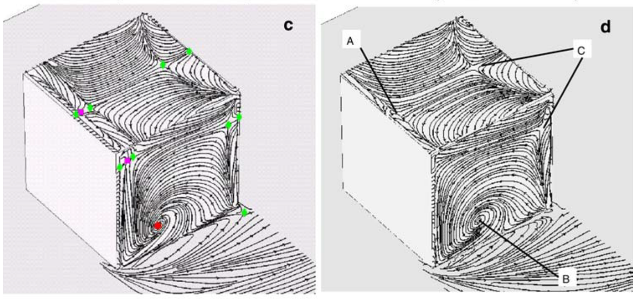


Figure 3.8: Results of the near-wall PIV measurement by Depardon et al. [2005] with velocities obtained at $0.5\,\mathrm{mm}$ (0.8% of the cube height) from the walls.



Oil Flow visualization ($Re_H = 40000$)

Oil Flow visualization ($Re_H = 160000$)



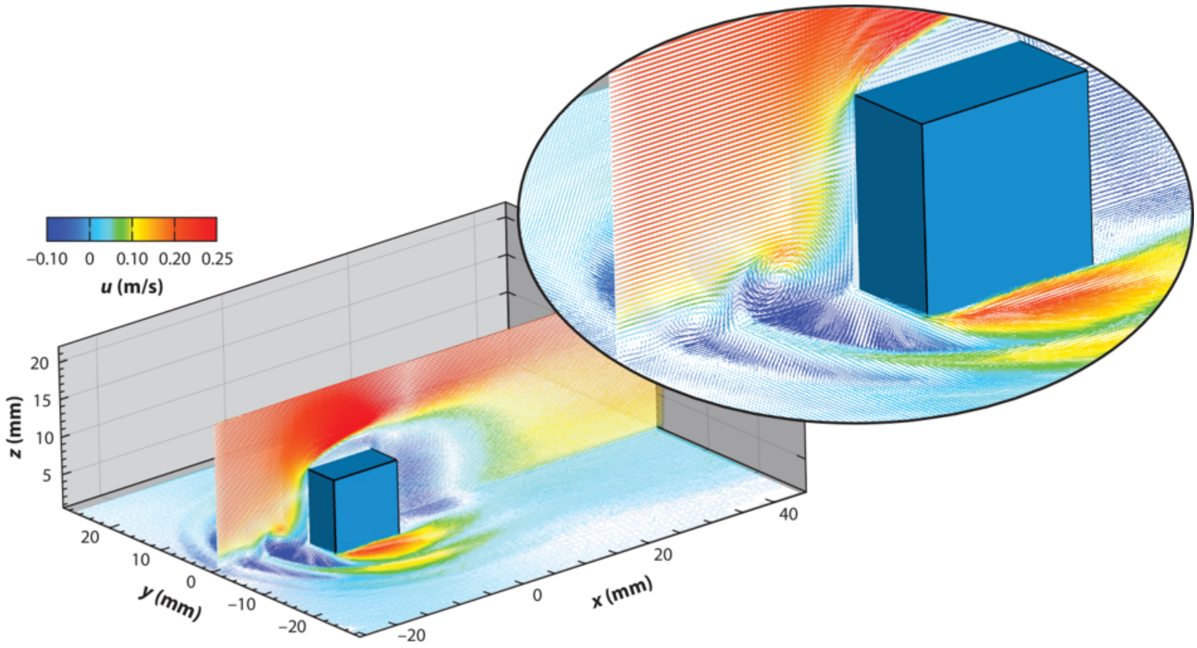
PIV measurements ($Re_H = 40000$)

PIV measurements ($Re_H = 160000$)

Figure 3.9: Skin friction topology inferred from oil flow visualisation and near-wall PIV at high Reynolds numbers. Critical points are indicated, with green indicating saddle points, purples indicating nodes and red a focus. ([Depardon et al., 2005])

---

[1] Note that in reality this is a particle tracking approach, so robotic PTV or LPT would have been more applicable.



Schröder A, Schanz D. 2023
Annu. Rev. Fluid Mech. 55:511–40

Figure 3.10: Example results of the three-dimensional flow around a surface-mounted cube. (see [Schröder and Schanz, 2023], reformatted from [Schröder et al., 2020].

to the lack of an accurate model position estimate. A problem that can now be resolved with the object registration method of Hendriksen [2024].

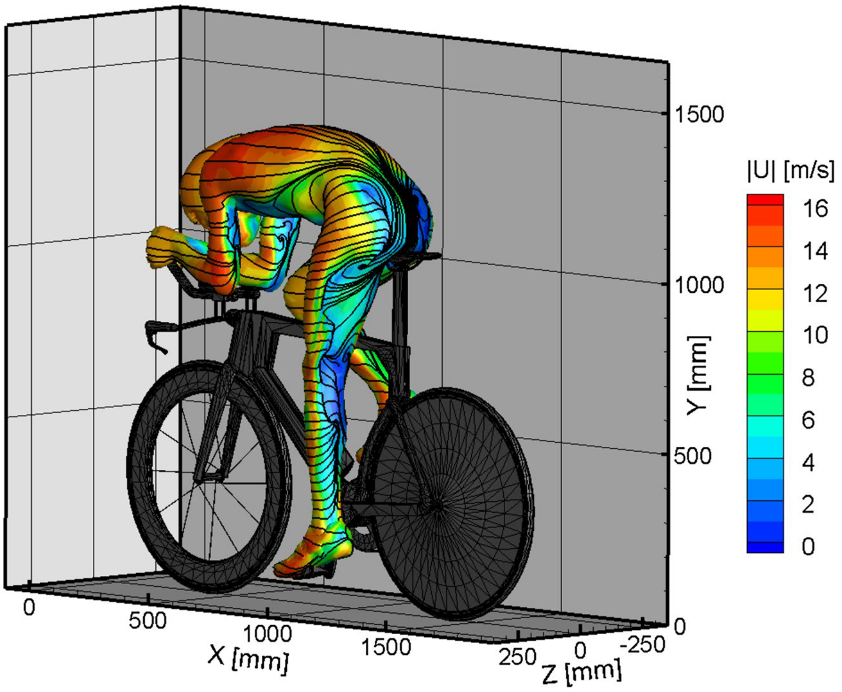


Figure 3.11: Near-surface velocity streamlines plotted on a dilated surface of the cyclist, along with contours showing the velocity magnitude by Jux et al. [2018].

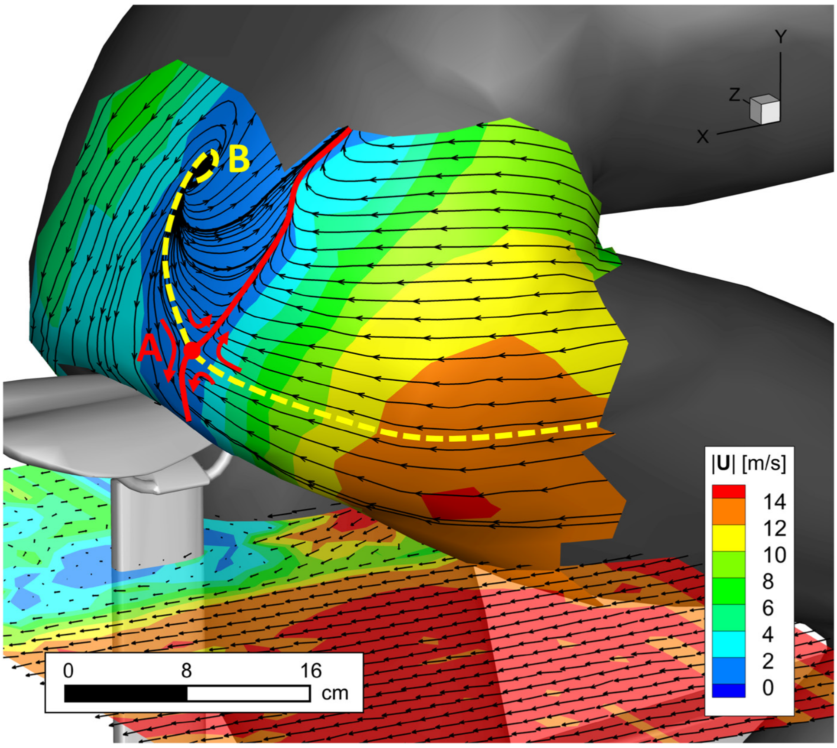


Figure 3.12: Near-surface velocity contour overlayed with skin friction topology at the lower back of a cyclist. [Schneiders et al., 2018].

3.3. Applying data assimilation to near-wall PIV and LPT

Reviewing current implementations of DA approaches can provide insight into how experimental data is generally combined with numerical models. DA is applied in either one of the following two similar goals; (1) improving numerical simulations by introducing real-world measurements or (2) improving measurement accuracy and resolution by introducing numerical models. The latter goal can be considered as a model-informed interpolation. With regards to 3D LPT, DA has been successfully utilised to improve the temporal and spatial resolution of instantaneous measurements ([Schneiders and Scarano, 2016] and [Gesemann et al., 2016]).

There exist two main types of DA methods mentioned throughout literature ([Talagrand, 1997] and [Hayase, 2015b]), sequential DA methods and variational DA methods. Though sequential DA methods have seen much use in other fields, when it comes to PIV and LPT, most approaches have employed variational methods. Therefore, these are primarily covered in subsection 3.3.1. The incorporation of boundary conditions is reviewed in subsection 3.3.2 which is essential in the context of near-surface fluid dynamics.

3.3.1. Data assimilation approaches

There are two main types of DA methods mentioned throughout literature:

- *Sequential methods*, divided into statistical and state observer methods. Examples of statistical methods are Optimal Interpolation (OI), Kalman filters (KF), ensmble Kalman filters (enKF) and particle filters (PF).
- *Variational (non-sequential) methods*, particularly three-dimensional (3D-VAR) and four-dimensional (4DVar) variational methods.

The advent of data assimilation techniques with PIV and LPT has risen since the last decade. As of now, most works seem to favour the variational approach over the sequential. Sequential DA methods perform assimilation in a serial manner, iteratively stepping the solution to the next value. The core assumption is that the state of a system $\psi(\vec{x}_k)$ at time t_k only depends on the previous state of the system $\psi(\vec{x}_{k-1})$ and the observations $\vec{y}_k$ only depend on the state $\psi(\vec{x}_k)$ ([Bertino et al., 2003]). Sequential DA methods are further divided into statistical methods and state observer methods. Deng

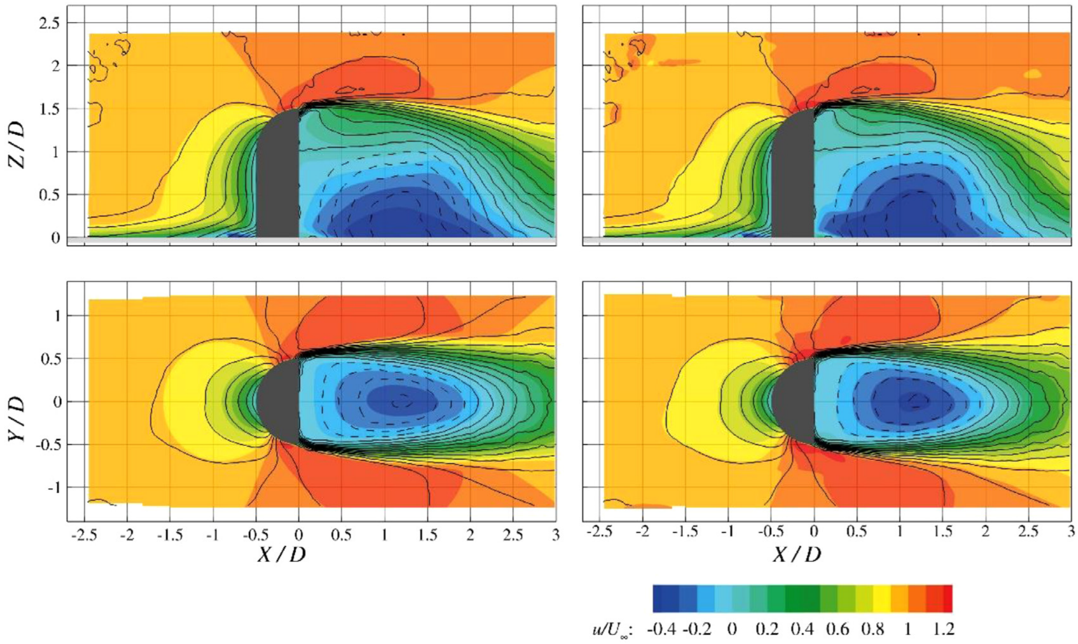


Figure 3.13: Example results of the state-observer implemented into a 3D RANS model for the flow around a simplified car mirror by Saredi et al. [2021]. The left column shows the contours of the proportional feedback, whereas the right column shows the results of the proportional-integral feedback term. All plots are overlayed with the reference data.

et al. [2018] made use of the ensemble Kalman filter ([Evensen, 1994]) to assimilate local measurement data into Reynolds-averaged Navier-Stokes (RANS) models. They compared the EnKF with four different turbulence models, the Spalart and Allmaras (SA, [Spalart and Allmaras, 1994]), $\kappa - \epsilon$ ([Launder and Spalding, 1974]), $\kappa - \omega$ ([Wilcox, 1988]) and the shear stress transport (SST, [Menter, 1994]) models. The measurement data was obtained from planar PIV at $Re = 6000$ for a jet flow and different subsamples of the measurements were assimilated and compared against each other.

Early examples in literature on state-observers can be found from [Hayase, 2015a], though few are validated/applied to PIV measurements. An example using PIV is given by Yamagata et al. [2008] at a Reynolds number of 1200, using a body force term in the 2D RANS equations. They investigated the effect of the feedback data rate by changing the feedback frequency and the feedback area (the percentage of measured data which is assimilated). Saredi et al. [2021] introduced a state-observer into the 3D-RANS equations. The goal was to improve the accuracy of RANS-based CFD simulations for unsteady flows at high Reynolds numbers. Two feedback control laws were compared using the flow around a simplified car mirror, one with a proportional feedback and one with a proportional-integrated feedback. Some results are shown in figure 3.13. The result showed that the assimilated measurement results are largely advected by the flow and a high density of measurement points is required to significantly affect to solution in regions where advection dominates.

Variational DA methods have been extensively used with PIV and LPT. Most well-known examples are VIC and its variations which have been introduced in subsection 2.1.5. These do not follow a sequential approach, but are known as an adjoint method ([Talagrand and Courtier, 1987]). The principal is to adjust one global model solution towards the available observations, using both past and future information. A scalar, the cost function, is defined which defines the distance between the solution and the observations. The goal is to find the solution which minimises this scalar, i.e. the distance. This is known as a 'constrained variational' problem, where the unknowns (e.g. velocity or pressure) should minimise the scalar, while simultaneously adhering to a given set of constraints (e.g. the incompressible Navier-Stokes equations) ([Talagrand and Courtier, 1987]). The constraints can be introduced into the optimisation through Lagrangian multipliers ([Bertsekas, 1996]). The gradient of this Lagrangian can be used to iteratively minimise the cost function. This gradient is known as the adjoint equation. Typically conjugate-gradient or quasi-Newton methods are used for this iterative procedure ([Evensen et al., 2022]). Though effective, one of the main obstacles of variational methods is the need for the tangent linear and adjoint model operators, which can be a complicated task to obtain. Additionally, for very nonlinear problems, the gradient-based 4DVar may get trapped in a local minimum.

The model and measurement operators in the equation for the cost function are generally nonlinear. This yields a non-quadratic cost function $\mathcal{J}$, which is often difficult and computationally expensive to minimise ([Evensen et al., 2022]). An incremental approach which makes use of the incremental Gauss-Newton formulation can offer a computationally cheaper and more efficient solution. In this approach, the minimisation of the cost function is replaced with a loop to minimise an inner incremental (linearised) cost function which is quadratic. The inner iterations can be solved with various minimisation methods, such as the conjugate gradient method or the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.

Mons et al. [2016] compared the ensemble Kalman smoother (EnKS, an extension to the EnKF) against 4DVar methods. The Benefit of the EnKS was the availability of posterior statistics next to an assimilated result, however, the 4DVar yielded the largest error reduction given the same computational cost and fixed observation and prior statistics. An implementation of 4DVar using planar PIV is given by Foures et al. [2014]. They give a complete description on how the weak-constraint variant can be implemented into the RANS equations. The divergence of the Reynolds stress tensor was included as one of the design variables, contrary to using a turbulence model. Using this approach they reconstructed mean velocities, pressures and, of course, Reynolds stress tensors at a very low Reynolds number of 150. This method was later extended to higher Reynolds numbers by Symon et al. [2017] and applied to an idealised 2D airfoil at $Re = 13,500$. Their results are displayed in figure 3.14 showing the PIV-measurement results and the assimilated results.

3.3.2. Implementing boundary conditions

At its core, data assimilation relies on solving a dynamical model defined by (partial) differential equations. DA applications to fluid mechanics are no exception to that and this essentially results in a merger between the field of computational fluid dynamics (CFD) and experimental fluid dynamics (EFD). The merger comes with several complications as data in the experimental (or measurement/observation) space needs to be mapped to the computational (or model) space. Information in the experimental space lies on an experimental mesh which must be connected to the computational mesh which stores the information in the computational space. Such a mapping can be done using linear projections or interpolations, but other nonlinear approaches (e.g. convolutions) [Evensen et al., 2022] can also be utilised. When assimilating data from an experimental mesh to a computational mesh, a correct implementation of boundary conditions becomes crucial since slight mismatches between the physical boundaries and the computational boundaries can yield significant errors. These position errors can be included into the data assimilation framework, either as an alignment error or using boundary conditions as design variables [Nehrkorn et al., 2015].

The issue of measurement position and boundaries does raise the question on how boundary conditions are generally implemented in data assimilation frameworks for fluid mechanics. Perhaps the most

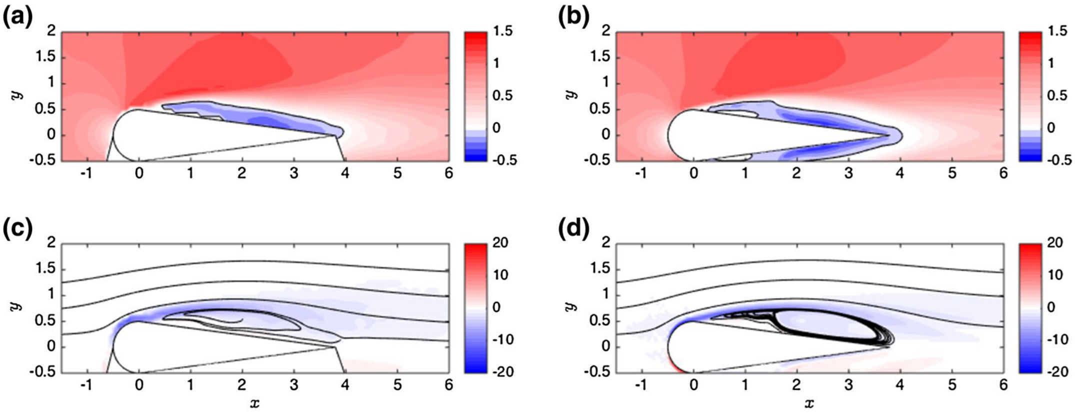


Figure 3.14: Results of the variational assimilation process applied to an idealised airfoil at high Reynolds number by Symon et al. [2017]. The left column shows the results from 2D PIV and the right column shows the assimilated results. The top plots show contours of the mean velocity and the bottom displays contours of the vorticity.

straightforward method to handle a boundary is through the use of a conformal mesh. In this manner the mesh is generated such that it aligns with the boundary. Conformal meshes can be easily created for rectangular and spherical models and domains. Conformal meshes for more complex geometries such as hills or airfoils can be handled through mapping techniques ([Baker, 2005]). The coordinate system (x, y) would be mapped to the space (η, ξ) which has a simple geometry such as a circle or rectangle. The model can be evaluated with the alternative coordinates and similarly, the boundary conditions can be imposed there. The advantage is an analytical representation of the domain and a simplification of the numerical derivatives. In reality, however, very few complex geometries can be mapped to a simpler domain ([Baker, 2005]).

With the advent of numerical mesh generation, more complex configurations could be conformally meshed using multi-block approaches ([Baker, 2005]). This involves a hybrid meshing strategy using multiple mesh types such as C-, O- and H-meshes. This approach could handle more model geometries, though to this day it still requires significant human error to set up such a multi-block mesh. This need for human interaction was resolved with the development of automatic unstructured meshing techniques ([Owen, 2000]). Several techniques exist for three-dimensional unstructured meshing, mostly based on tri/tetrahedral (triangulation) meshing or quad/hexahedral meshing. Typical triangulation methods are the Delaunay triangulation, advancing front, and octree decomposition. The Delaunay triangulation make use of the Delaunay criterion ([Delaunay, 1934]) which states that no node in the mesh must be within the circumsphere of any triangle/tetrahedron. A triangulation is sought that matches this criterion. The advancing front approach progressively meshes a domain starting at a boundary. New nodes are inserted based on certain criteria, which may also be the Delaunay criterion. A special form of the advancing front method is the advancing layers ([Owen, 2000]), which is used to create mesh elements along the boundary which are stretched in the fluid direction. An octree decomposition is obtained by recursively subdividing a domain into smaller elements until the desired resolution. In its simplest form this can be done by splitting domains into cube-shapes, which can be split in half to form the tris/tetrahedra.

After a conformal mesh is created, boundary conditions must be imposed on it. Generally speaking two boundary types are recognised for fluid mechanical applications, namely Dirichlet and Neumann boundary conditions ([Tu et al., 2018]). The Dirichlet boundary condition specifies the value that the solution needs to take along the boundary and the Neumann boundary condition specifies the value that a derivative of the solution must have.

Another approach to handle boundaries is through the use of the immersed boundary (IB) method, which is often used in problems of fluid-structure interactions. In this method a force distribution over all grid points replaces the solid boundaries. The method was first proposed by Peskin [1972] as a method to investigate the fluid flow around heart valves. Since its introduction, the method has found its place in various applications due the benefit of avoiding complicated mesh generation, particularly in the situation of moving and deforming objects ([Huang and Tian, 2019]).

The technique was originally designed to solve problems with IBs which are both moved by the fluid and exert a force on the fluid. It was adapted for rigid bodies using the penalty IB method ([Goldstein et al., 1993]). In this approach the Lagrangian forcing is modelled as a feedback term based on the prescribed velocity at the boundary and the velocity of the solution ([Verzicco, 2023]). A special case of the penalty IB method is the projection IB method. In this approach, which treats the force as a Lagrangian multiplier for imposing the no-slip condition ([Huang and Tian, 2019]).

A main problem of the IB method for fluid dynamics is the treatment of near-wall regions, particularly in the case of high-Reynolds number flows ([Verzicco, 2023]). The IB method must cope with thin boundary layers to yield a sufficient resolution, yet does not benefit from the anisotropic element dimensions. Body-fitted meshes can typically be resolved with either wall-resolved or wall-modelled approaches, but for IB a wall model must be implemented ([Cai et al., 2021]).

A comparison of immersed boundaries and conformal boundaries with the data assimilation approach has been given by Cakir et al. [2022]. They modified the VIC+ method with an immersed boundary (ImVIC+) and with a conformal mesh using the Arbitrary Lagrangian-Eulerian approach (ALE-VIC+) to specify appropriate boundary conditions near surfaces. Results of both approaches and of the VIC+ approach applied to a periodic hill are reproduced from ([Cakir et al., 2022]) and shown in figure 3.15. Cakir et al. also used the updated approaches to estimate wall-pressures. They concluded that the ImVIC+ implementation slightly outperformed the ALE-VIC+ approach. Proper inclusion of the boundary conditions has had an expected impact on the assimilated near-wall velocity, however,

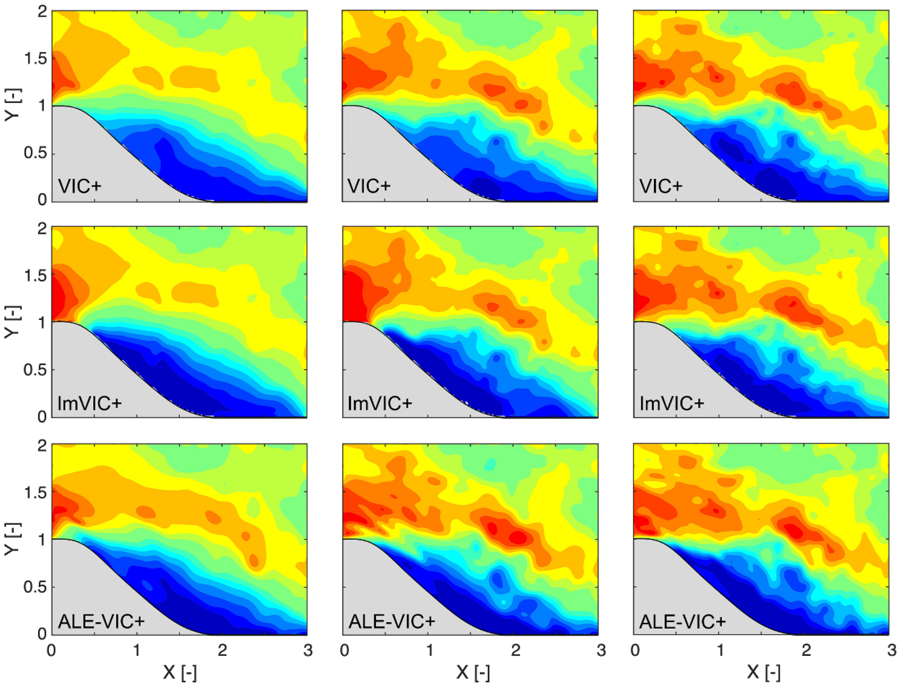


Figure 3.15: Results of the ImVIC+ and ALE-VIC+ approaches for a period hill compared against the traditional VIC+. Moving from left to right the density of particles is increased. ([Cakir et al., 2022])

neither approaches have incorporated the viscous terms which dominate near the walls. A proper comparison of the ImVIC+ and ALE-VIC+ approaches to implement the no-slip boundary conditions need realistically be updated with an inclusion of viscous terms, which might require the inclusion of turbulence models. Additionally, the results are currently limited to 2D and simple geometries.

3.4. Research goal statement

Chapter 2 and the previous sections have provided an overview of the current state-of-the-art of PIV and LPT and near-wall flow measurements. This section focuses on translating that review into a specific gap in literature, which is done in subsection 3.4.1. Then, the research objective and questions meant to fill this gap are formulated in subsection 3.4.2.

3.4.1. Research gap identification

Skin friction measurements are often limited to simple geometries and few examples exist which yield a full skin friction distribution on any geometry. Near-wall velocity data can be resolved using PIV and LPT, up to a small distance from the model surfaces. Current data assimilation approaches utilising large-scale PIV and LPT results have mainly focused on far-body flows. Near-surface methods have been proposed and experimented with, but this has been limited to the two-dimensional case and viscous terms have been neglected. This leaves a research gap to utilise large-scale PIV/LPT to reconstruct near-surface fluid dynamics, with a specific focus on skin friction distribution, for generic, complex bodies. It is assumed that the location of the physical walls with respect to the measurement data can be accurately determined, which was demonstrated by Hendriksen [2024].

3.4.2. Research objective and questions

The following main objective of the research is proposed:

The research objective is to reconstruct near-surface fluid dynamical properties in large-scale time-averaged PIV/LPT measurements by combining verified body position information with experimental results in the outer-field.

The main objective is divided into three research questions, which deal with I) reconstruction of the near-surface velocity, II) computation of near-surface properties such as skin friction and boundary layer parameters and III) implementation requirements of the method into the current state-of-the-art.

- **I** How can the information of the object position be utilised to reconstruct the near-surface velocity gaps in measurements from the PIV/LPT?
  - **Ia** How can the model domain (verified body position) be merged with the fluid domain (experimental results)?
    *Topic of **Domain discretisation (DD)**.*
  - **Ib** What method can be used to reconstruct the near-surface velocity field from the merger between object and fluid domain while implementing (basic) flow physics?
    *Topic of **Computational (fluid) reconstruction models (CfRM)**.*
  - **Ic** How are the boundary conditions implemented (verified body position) implemented?
    *Topic of **Boundary condition implementation (BCI)***
  - **Id** How does the proposed method compare to what is currently available with the state-of-the-art in terms of the near-surface velocity, skin friction topology (critical point positions) and skin friction magnitude?
    *Topic of **Method verification & validation (MVV)***
- **II** How are the fluid dynamical (near-)surface properties computed from the reconstructed velocity field?
  - **IIa** What method can be used to compute the wall shear stress distribution from the reconstructed velocity field?
    *Topic of **Wall shear stress computation (WSSC)**.*
  - **IIb** How accurate is this method in reconstructing the topology, when comparing the position of critical points to literature
    *Topic of **Skin friction topology (SFT)**.*
  - **IIc** How accurate is this method in reconstructing the skin friction magnitude, compared to literature and benchmark solutions.
    *Topic of **skin friction magnitude (SFM)**.*
- **III** How is the method implemented into the current state-of-the-art?
  - **IIIa** What are the requirements on the PIV/LPT data for use of this method in terms of particle density and tracer data availability near the wall?
    *Topic of **Measurement data requirements (MDR)**.*
  - **IIIb** How can the method be generally implemented into PIV/LPT data assimilation given any verified object mesh (such as a .STL) and flow field data points?
    *Topic of **General method implementation and use (GMIU)**.*

This report aims to answer the above-mentioned questions. A method has been devised to reconstruct the near-surface velocity gaps. The skin friction distribution has been determined from the results of this method and a graphical user interface has been designed to allow the post-processing of volumetric velocimetry data. Note that the method is verified and validated using 3D LPT techniques. The state-of-the-art for near-surface velocity description by Jux et al. [2018] is used as a starting point, who incorporated 3D LPT data which had undergone CGR.

4

Experimental dataset

The experimental data used through this thesis report is obtained during the test campaign in [Hendriksen, 2024]. The goal of this experiment was to validate various object registration approaches of three wall-mounted models with a LPT-camera setup. This chapter details the experimental datasets and processing steps first before the proposed method is outlined in the following chapters, since the data is occasionally utilised to highlight different stages of the method. Firstly, section 4.2 introduces the wall-mounted models which are investigated. Then, the test facility is shortly described in section 4.1. Section 4.3 describes the experimental procedures and acquisition approach for each of the three models. Lastly, the post-processing of the digital images to obtain the Lagrangian particle trajectories is described in section 4.4

4.1. Test facility: W-tunnel

The experiments are conducted in the W-tunnel at the High Speed Laboratories of TU Delft, shown in figure 4.1. This wind tunnel is a low-speed, open jet facility with an exit of $400\,\mathrm{mm}\times 400\,\mathrm{mm}$. It is typically used for smaller experiments and allows testing in wind speeds up to $35\,\mathrm{m/s}$. The level of turbulence can be reduced to about 0.5% (Delft University of Technology, 2023).

The free-stream velocity during the experiments is $10\,\mathrm{m\,s^{-1}}$ for the cube and WBJ models and $8\,\mathrm{m\,s^{-1}}$ for the scaled cyclist, which yields Reynolds numbers of 80,000 for the cube, 120,000 for the WBJ (based on the height) and 50,000 for the cyclist (based on the torso size of $90\,\mathrm{mm}$). The w-tunnel offers a convenient way to place the seeding generator in the settling chamber before the contraction section, which minimises the increase in turbulence intensity ([Giaquinta, 2018]). An aluminium frame was constructed to suspend the plywood board with the surface mounted models upside down, such that the cameras and light source can be placed near the ground and be pointed upward. This made the cameras more accessible to tune the focal lengths.

4.2. Surface-mounted test objects

Three models were investigated during the experimental campaign with increasing levels of geometrical complexity:

- A cube with sides of $120\,\mathrm{mm}$ in length.
- A wing-body junction (WBJ) with a chord length of $180\,\mathrm{mm}$, a span of $108\,\mathrm{mm}$ and maximum thickness of approximately $36\,\mathrm{mm}$.
- A cyclist with 1:8 scaling, a wheel-to-wheel distance of $220\,\mathrm{mm}$, a height of $180\,\mathrm{mm}$ and a width of approximately $50\,\mathrm{mm}$.

These models are shown in figure 4.2, displaying both the models with and without reflective markers. These markers were integrated into the model surface for the object registration. The cube and WBJ

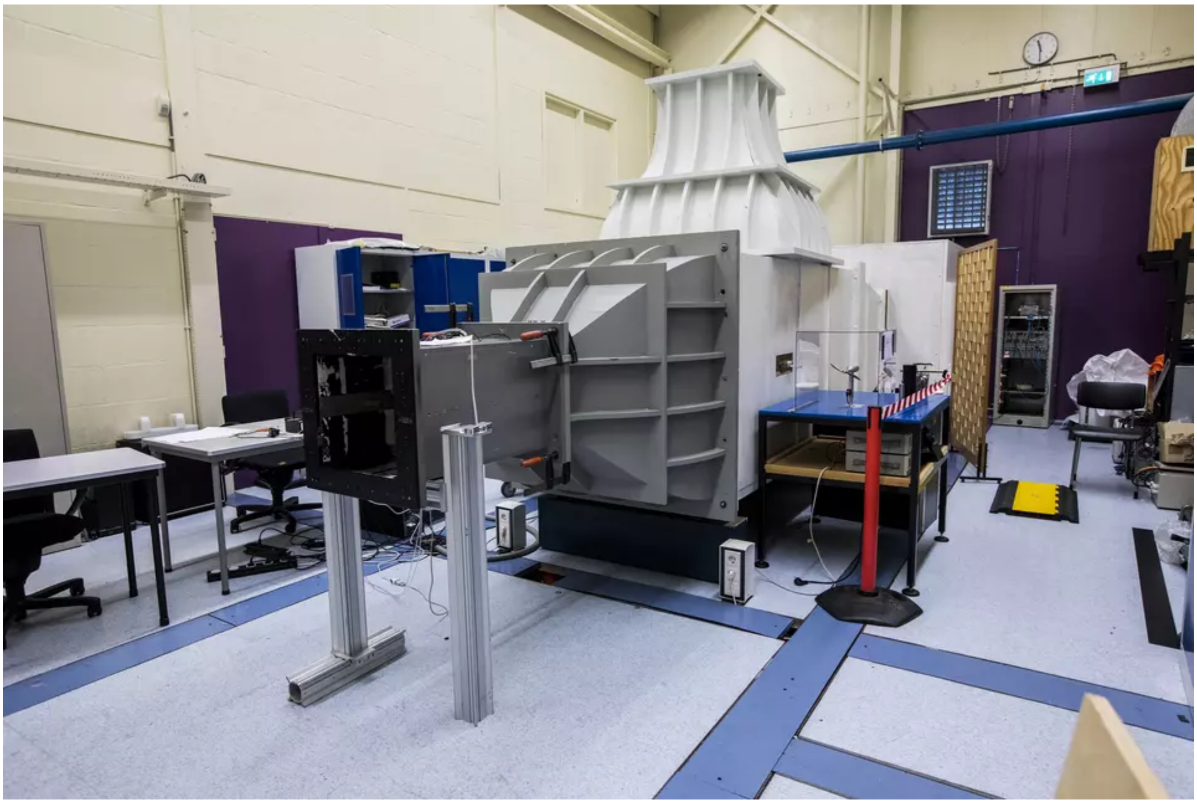


Figure 4.1: Overview of the W-Tunnel at the High Speed Laboratories of TU Delft. (Taken from [Delft University of Technology, 2023])

models were produced with a Fused Deposition Modeling printer and the nominal manufacturing accuracy is respectively $0.3\,\mathrm{mm}$ and $0.1\,\mathrm{mm}$. The cyclist model was manufactured with a stereolithography printer and its nominal manufacturing accuracy is $0.1\,\mathrm{mm}$. All models were mounted to a plywood board with an elliptical leading edge using an adapter plate at a position $800\,\mathrm{mm}$ from the leading edge of the board. A cut-out was manufactured in the board where the adapter plate was placed. A trip strip was placed on the plywood board approximately $500\,\mathrm{mm}$ forward of the models.

4.3. Experimental setup and procedures

Next, the experimental setup and procedures are shortly discussed. An overview of the camera and light source setup is provided in subsection 4.3.1. The helium-filled soap bubbles and the seeding

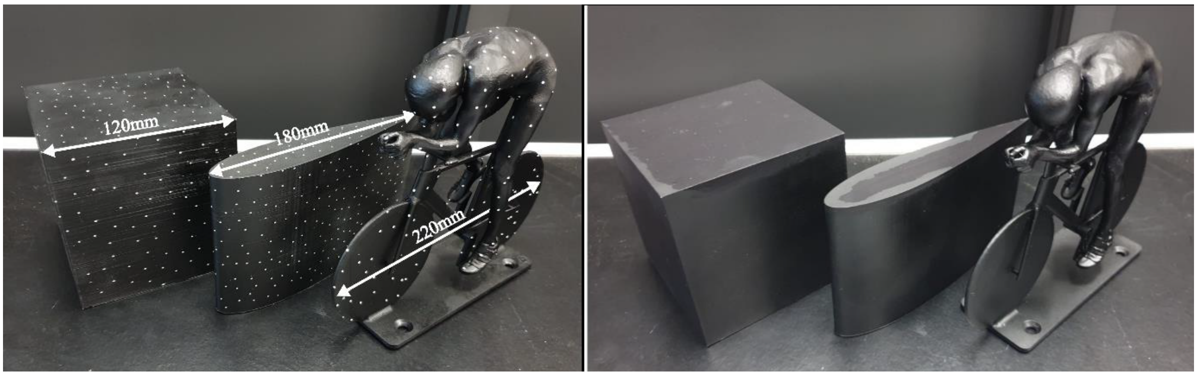


Figure 4.2: Image of the three models that were examined during the experimental campaign. Reflecting surface markers were integrated into the models on the left. The models on the right were marked using laser pointers. (Reprint from [Hendriksen, 2024])

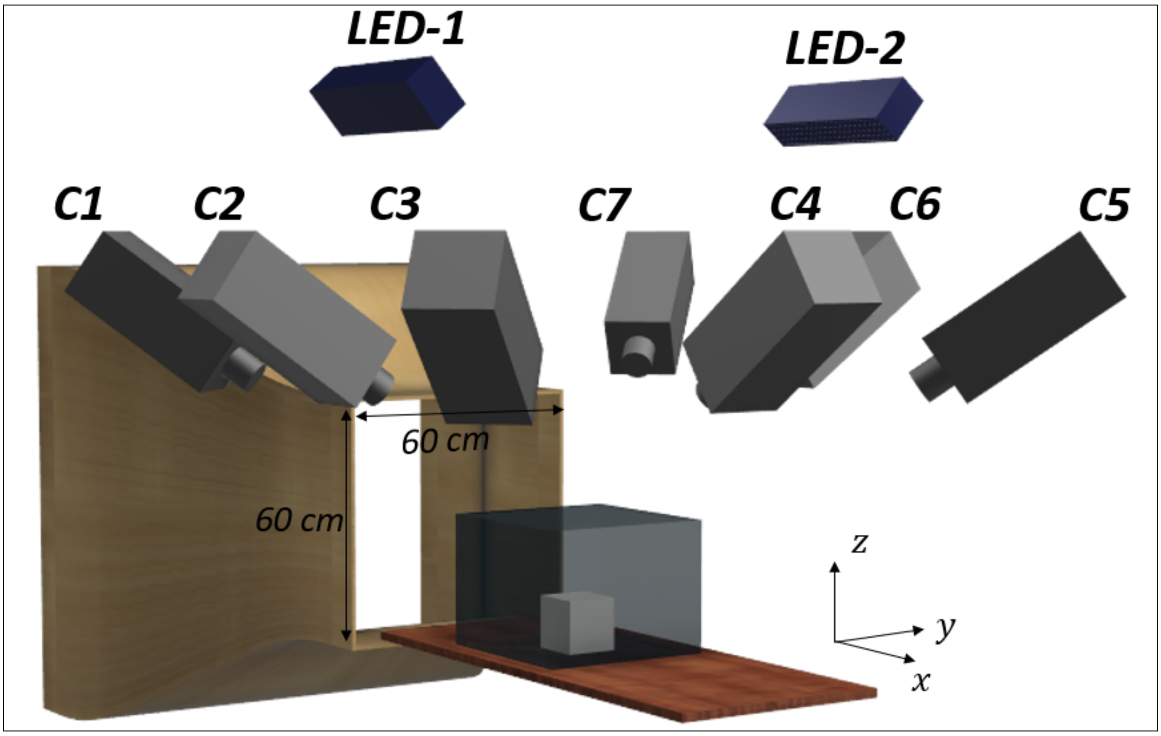


Figure 4.3: Render of the wind tunnel test setup where the measurement domain is visualised using a glass box. Seven cameras are placed in a square configuration around the object and two LED-lights were used for illumination.

generator are introduced in subsection 4.3.2. Subsection 4.3.3 outlines the data acquisition procedure from calibration to imaging.

4.3.1. Imaging setup

The imaging setup consists of seven high-speed complementary metal-oxide semiconductor (CMOS) cameras yielding a measurement volume of $400 \times 400 \times 300$ mm ($L \times W \times H$) around the model. This also includes a small segment under the ground plane. The cameras are placed in a square configuration around the model, as shown in figure 4.3. All cameras have a resolution of 1024×1024 px (1 Mpx), a pixel pitch of 20 µm, a magnification M of 0.05, an angular camera offset of 45° and can be operated up to 5400 fps. The focal length and $f_\#$ of the cameras varied depending on their position and is summarised in table 4.1. Two large *flashlight-300* LED lights from LaVision are employed to illuminate the measurement volume.

Table 4.1: Camera focal lengths and $f_\#$.

	C1	C2	C3	C4	C5	C6	C7
Focal length [mm]	60	50	60	50	60	50	60
$f_\#$ [-]	32	22	22	16	22	22	32

4.3.2. Tracer particles and flow seeding

Neutrally buoyant HFSB with a mean diameter of 0.35 mm are used to seed the flow. The bubbles are generated with a seeding rake of 0.5×1 m. The rate of production is 6×10^6 particles per second ([González Saiz et al., 2022]). As mentioned above, the seeding rake is placed in the settling chamber of the w-tunnel. With the free-stream velocity of 10 m s^{-1} and 8 m s^{-1}, the seeding density is respectively around 1.2 and 1.5 particles per cubic centimeter (p cm^{-3}) and the source density is 0.02 and 0.025 particles per pixel (ppp).

Table 4.2: Summary of the model and experiment parameters

	Cube	**WBJ**	**Cyclist**
Model size ($L \times W \times H$) $[\mathrm{mm}^3]$	$120 \times 120 \times 120$		$220 \times 50 \times 180$
Tracer particles [-]	Helium-filled soap bubbles		
Free-stream velocity $[\mathrm{m\ s}^{-1}]$	10	10	8
Reynolds number [-]	80,000	120,000	48,500
Measurement volume ($L \times W \times H$) $[\mathrm{mm}^3]$	$400 \times 400 \times 300$		
Seeding density $[\mathrm{p\ cm}^{-3}]$	1.2		1.5
Source density $[\mathrm{ppp}]$	0.02		0.025
Acquisition frequency $[\mathrm{Hz}]$	3,000		
Number of images acquired [-]	5,000		

4.3.3. Data acquisition

Next the data acquisition procedure is discussed. The model and experiment paremeters are summarised in table 4.2. First, the optical system was calibrated using a two-dimensional calibration target which consists of an equally spaced array of dots. The plate was positioned three different positions and orientations. A calibration function based on the pinhole model is set up for each z-position (considering midplane at $z = 0$) ([Soloff et al., 1997] and [Willert, 1997]). These calibration functions are combined in a single mapping function $M_i(X, Y, Z)$ for each camera i, which relate the physical 3D location of the particle to its 2D position in the camera image.

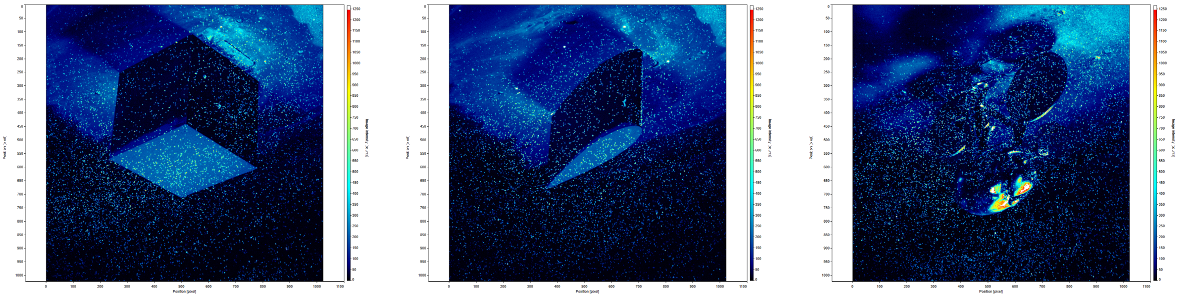
After the physical calibration the wind tunnel was turned on together with the seeding rake at a lower production rate to obtain particle images for a VSC. Then, one of the models was carefully installed and the seeder production rate was reset to the previously mentioned production rate. With each run, 5,000 images are obtained at an acquistion frequency of $3.000\,\mathrm{Hz}$. For purposes of consistency, the calibration and VSC setup was repeated when switching to the other models. As will be discussed in the next section, the digital images were post-processed.

4.4. Data processing

This section discusses the processing of the digital images to obtain the Lagrangian particle trajectories. Examples of the raw digital images taken with Camera C7 for each test object is shown in figure 4.4. First, a minimum filter was applied to the raw images. This filter is computed over a width of $N_{\text{filt}} = 11$ images using a symmetrical kernel. The minimum filter subtracts the background noise from the raw images and increase to signal-to-noise ratio (SNR).

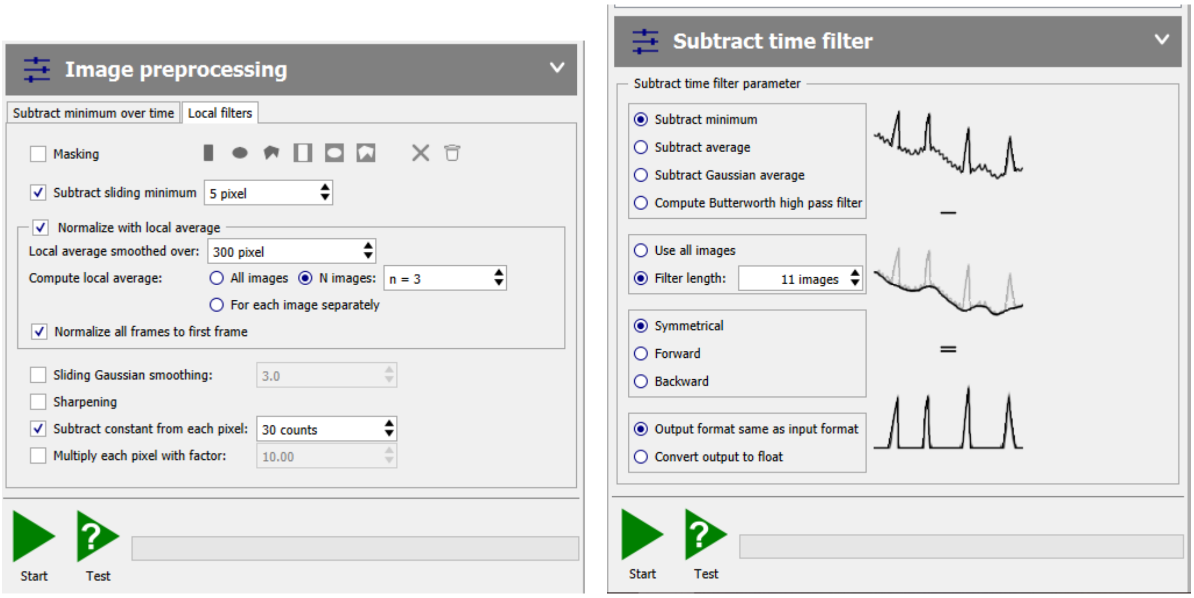
Next, the images are preprocessed with local filters. A sliding minimum is subtracted using a kernel size of $5\,\mathrm{px}$. The pixel intensities are normalised with a local average. The local average is smoothed over $300\,\mathrm{px}$ and computed using three images. All frames are then normalises with respect to the first frame. Additionally, a constant pixel count of 30 is subtracted from each pixel.

STB is performed inside a rectangular measurement volume of approximately (LxWxH) $400\mathrm{x}400\mathrm{x}$ $320\,\mathrm{mm}$ with a single pass. A threshold of 10 counts is used for detecting particles, allowing for a triangulation error of 1.5 voxels. For the tracking of the bubbles, velocity ranges of $-10 < u < 20$



(a) (b) (c)

Figure 4.4: Raw digital images of the (a) surface-mounted cube, (b) WBJ and (c) cyclist.



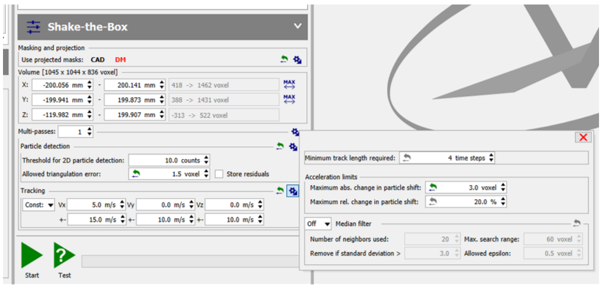
☐ Masking
☑ Subtract sliding minimum 5 pixel
☑ Normalize with local average
Local average smoothed over: 300 pixel
Compute local average: ○ All images ● N images: n = 3
○ For each image separately
☑ Normalize all frames to first frame
☐ Sliding Gaussian smoothing: 3.0
☐ Sharpening
☑ Subtract constant from each pixel: 30 counts
☐ Multiply each pixel with factor: 10.00

Subtract time filter parameter
● Subtract minimum
○ Subtract average
○ Subtract Gaussian average
○ Compute Butterworth high pass filter
○ Use all images
● Filter length: 11 images
● Symmetrical
○ Forward
○ Backward
● Output format same as input format
○ Convert output to float

(a) (b)

Figure 4.5: Settings used for the image preprocessing.

ms^{-1}, $-10 < v < 10$ ms^{-1} and $-10 < w < +10$ ms^{-1} are used. A minimum track length of 4 time steps is imposed before a bubble trajectory is used for the shaking process (i.e. $N_{\text{init}} = 4$). Lastly, two physical limits are imposed on the trajectory matching. The maximum absolute change in particle shift is set 3 voxels and the maximum relative change in particle shift is set to 20%. The settings for the image preprocessing and the STB algorithm are included in figures 4.5 & 4.6, respectively.



Masking and projection
Use projected masks: CAD DM
Volume [1045 x 1044 x 836 voxel]
X: -200.056 mm - 200.141 mm 418 -> 1462 voxel
Y: -199.941 mm - 199.873 mm 388 -> 1431 voxel
Z: -119.982 mm - 199.907 mm -313 -> 522 voxel
Multi-passes: 1
Particle detection
Threshold for 2D particle detection: 10.0 counts
Allowed triangulation error: 1.5 voxel ☐ Store residuals
Tracking
Const: Vx 5.0 m/s Vy 0.0 m/s Vz 0.0 m/s
+- 15.0 m/s +- 10.0 m/s +- 10.0 m/s

Minimum track length required: 4 time steps
Acceleration limits
Maximum abs. change in particle shift: 3.0 voxel
Maximum rel. change in particle shift: 20.0 %
Off Median filter
Number of neighbors used: 20 Max. search range: 60 voxel
Remove if standard deviation > 3.0 Allowed epsilon: 0.5 voxel

Figure 4.6: Settings used for the STB algorithm.

5

Local near-surface flow reconstruction

A literature review which addresses the gap in near-surface fluid analysis with PIV and LPT has been reported in chapter 3. A research objective has been proposed to employ flow physics and a verified body position to enable the reconstruction of skin friction topology. This problem is split into two parts, a geometrical and a physical. Firstly, the mismatch between the fluid domain $\Omega_{\mathcal{F}}$ and the model domain $\Omega_{\mathcal{M}}$ is highlighted in section 5.1 and resolved by introducing a third domain, the computational domain $\Omega_{\mathcal{C}}$. Then, section 5.2 introduces the binning approach using spherical interrogation volumes that is employed everywhere inside $\Omega_{\mathcal{C}}$, where the fluid-mesh and object-mesh are combined. This approach is referred to as the Local Interrogation Volume Approach (LIVA). Five methods of increasing complexity are presented in section 5.3 which impose the no-slip boundary condition on the velocity measurements inside the spherical interrogation volumes. Lastly, the object registration experiment discussed in chapter 4 shows considerable errors in the approximation of the ground plane position. This problem is addressed in section 5.4. A comment is given in appendix A on fitting the fluid-mesh and object-mesh with wall functions rather than with the proposed (low-order) polynomial functionals.

5.1. The mesh discrepancy

Results of particle tracking and PIV are generally transformed to Cartesian grids, either via CGR or via Cartesian interrogation volumes used with PIV. The binning causes aliasing issues near curved surfaces as shown in figure 5.1. The issue of non-body-conformal interrogation bins can be partially addressed with the functional fit, though the presence of the wall introduces a boundary layer with a high wall-normal velocity gradient, which is not accurately reproduced by only fitting the tracers' velocity. The region near the wall typically sees far fewer bubbles due to a lower entrainment into the boundary layer and the bubble size prevents measurements of the fluid velocity very near to the surface. Therefore, introducing information on the position of the wall and thereby imposing a no-slip

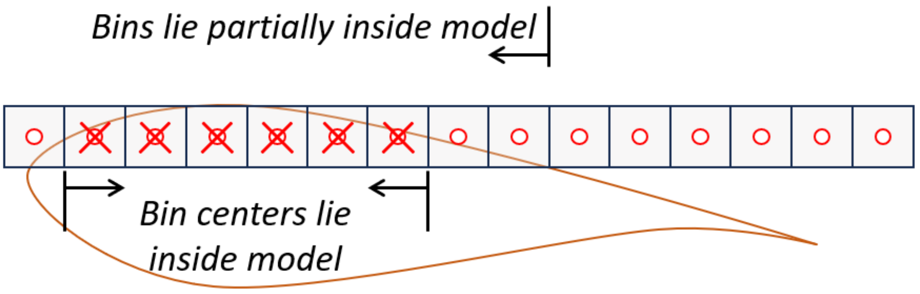


Figure 5.1: Schematic description of Cartesian-gridded bins lying (partially) inside a non-planar object.

condition is hypothesised to improve the near-wall velocity reconstruction.

First, subsection 5.1.1 outlines the gap that exists due to the discrepancy between the fluid and object domain. The problem was similarly addressed by Jux et al. [2018]. Their approach and solution is reviewed in subsection 5.1.2. This also serves as the baseline of comparison for the proposed velocity reconstruction techniques. Therefore, the linear interpolation employed by Jux et al. [2018] is detailed and explained in subsection 5.1.3.

5.1.1. The gap between the model and fluid domains

Consider the two-dimensional situation sketched in figure 5.2, which shows a binned fluid-mesh around a curved model. Only the bins are considered which centre does not fall inside the model. The nearest fluid-mesh nodes are connected to indicate the aliased boundary. There is no data available between the model and aliased boundary and therefore, it is not possible to interpolate the results within this region. The region bounded by the fluid-mesh boundaries and the aliased boundary is referred to as the interpolation domain Ω_{int}. It is apparent that fluid measurement results can not be interpolated outside of Ω_{int} and a gap exists between the $\Omega_{\mathcal{F}}$ and $\Omega_{\mathcal{M}}$.

To enable interpolation onto any point, the point must lie in the domain of the data that is interpolated. Points lying outside the domain can be estimated through extrapolation which rely on the curvature of the extrapolated dataset ([Atkinson, 1989]). The binned LPT results lie on a Cartesian grid which introduces aliasing near curved objects as bins in the same row, column and tube[1] can (partially) lie inside the model.

Ω_{int} can be enlarged up to the object's surface, by including the information that is given by the model surface, the no-slip condition. Such a situation is sketched in figure 5.3, where the model surface is discretised and forms the boundary of Ω_{int}. The gap that was present earlier in figure 5.2 is removed and the velocity can be interpolated in this region. The choice of interpolator is critical as it must be able to reconstruct the non-linear boundary layer profile. Several approaches are proposed in subsection 5.3 and assessed in the next chapter, varying from linear interpolations to fits of quadratic functionals.

It is imperative to understand the role of the object registration demonstrated by Hendriksen [2024]. Including the no-slip condition as a zero velocity boundary condition requires a very accurate determination of the position of the wall [Örlü et al., 2010; Titchener et al., 2015], as was discussed in subsection 2.3.3. The object registration provides the transformation matrix ${}_{\mathcal{M}}\mathcal{T}_{\mathcal{F}}$, aligning the coordinate systems of $\Omega_{\mathcal{M}}$ and $\Omega_{\mathcal{F}}$ within an error of less than 1%, which has opened the door to the approach outlined in this report.

It is therein worth to also investigate the solution by Jux et al. [2018]. They did not have an accurate estimate of the wall position, yet did demonstrate the near-surface velocity field around a cyclist. The remainder of this section focuses on their solution and thereby explains what is considered the state-of-the-art against which the results from the herein proposed methods will be compared.

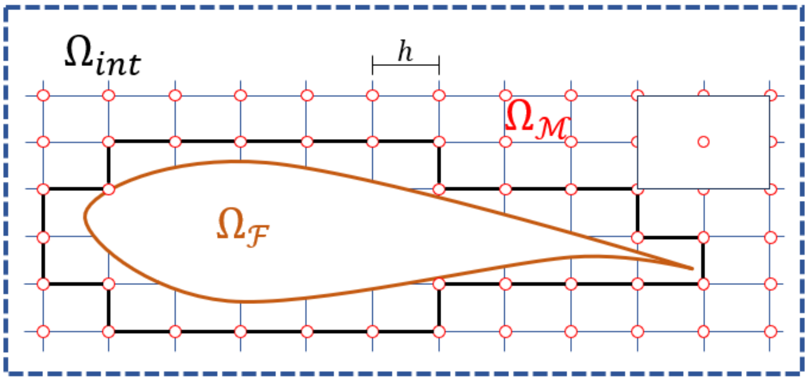


Figure 5.2: Aliasing effect on binned results and the near-surface gap due to the presence of the curved model. The bin is shown in the top right. The black line indicates the aliased contour and represents the boundary of the interpolation domain, Ω_{int}.

---

[1] Equivalent to row and column in the third direction ([Kolda and Bader, 2009])

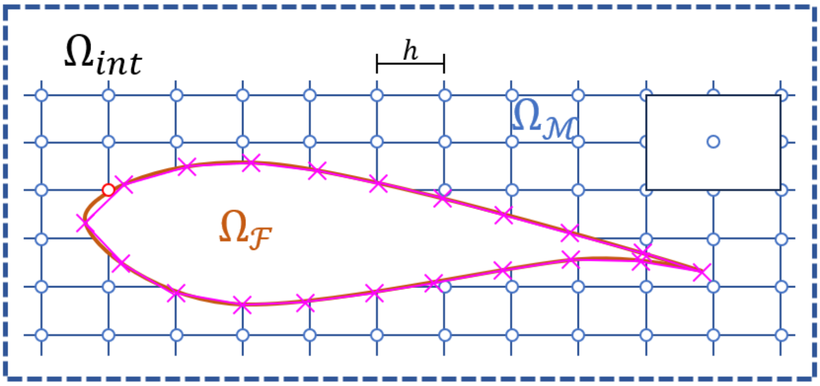


Figure 5.3: Schematic of how the problem of aliasing and the near-surface gap would be resolved to increase Ω_{int} when the position of the model surface is included into the known data points.

5.1.2. Jux' dilated surface approach

Jux et al. [2018] had to address the same data gap problem when determining the near-surface velocity over a cyclist. They were necessitated by the availability of fluid data near the surface after binning. Additionally, the position of the cyclist was not known accurately. This was firstly estimated from the missing presence of tracer particles, similar to [Jux et al., 2021]. To overcome the gap, they generated a dilated mesh from the surface of the cyclist, as if blowing up the model slightly, to introduce an equivalent, enlarged contour offset from the estimated model's walls. This brings the computational mesh inside the (local) convex hull of binned fluid measurement data and the velocity can be linearly interpolated onto this second mesh. The minimum offset distance is determined by the vector grid spacing h and can via simple geometry be determined to be $\sqrt{n} \cdot h$ for the two-dimensional $n = 2$ and three-dimensional $n = 3$ cases.

As explained in chapter 3, the results obtained by Jux et al. [2018] are considered the state-of-the-art and will be the point of departure for this report. In that work, multiple particle tracks were measured around a cyclist using robotic PIV, capturing the flow dynamics in the entirety of the measurement domain $\Omega_{\mathcal{F}}$. CGR was performed using binning to yield an Eulerian perspective and time-average the dataset. Due to the high density of particle tracks compared to the model size, it was possible to visualize velocity contours at $5\,\text{mm}$ distance (the size of a single vector spacing) from the model, which has dimensions of approximately (LxWxH) $1.6\text{x}0.4\text{x}1.3\,\text{m}$[2]. Taken the characteristic length scale to be $L = 0.9\,\text{m}$ (=$\sqrt[3]{\text{Volume}}$), the velocity contours were placed at about 0.5% of the characteristic length of the model. As explained above the authors generated a dilated surface while estimating the position of the cyclist based on tracer data availability. This approach will hereafter be referred to as *Jux' method*. The next subsection reviews and outlines an implementation of Jux' method. More detail is given to this implementation as it firstly provides a primer for the proposed family of interpolation methods in the remainder of the chapter and secondly, acts as a benchmark against which the other interrogation schemes will be compared.

5.1.3. Linear interpolation onto a dilated surface

After dilating the surface of the cyclist and placing this mesh inside Ω_{int}, the binned data is interpolated. It is possible to differentiate between interpolation from structured and unstructured grids. Structured grids enable fast interpolation for example with trilinear interpolation though higher order polynomial fits can be utilised. Meshfree interpolators such as radial basis functions (RBFs) or Kriging ([Fasshauer, 2007]) can be employed for unstructured grids. The aliasing effect yields an unstructured grid with structured components. Therefore, interpolation schemes that rely on structured grids are unavailable. The meshfree interpolators are suitable for interpolation inside Ω_{int}, though Kriging is a non-linear interpolator and RBF approximations require various considerations such as the placement of RBF-centres and the domain of the local RBFs. Therefore, a linear interpolation based on Delaunay triangulation, shorthanded with LIDT, is utilised which can handle unstructured grids.

Given an array of unstructured points, LIDT first triangulates the points based on the Delaunay cri-

---

[2] The reader is directed to [Jux et al., 2018] for the specifics on the model dimensions

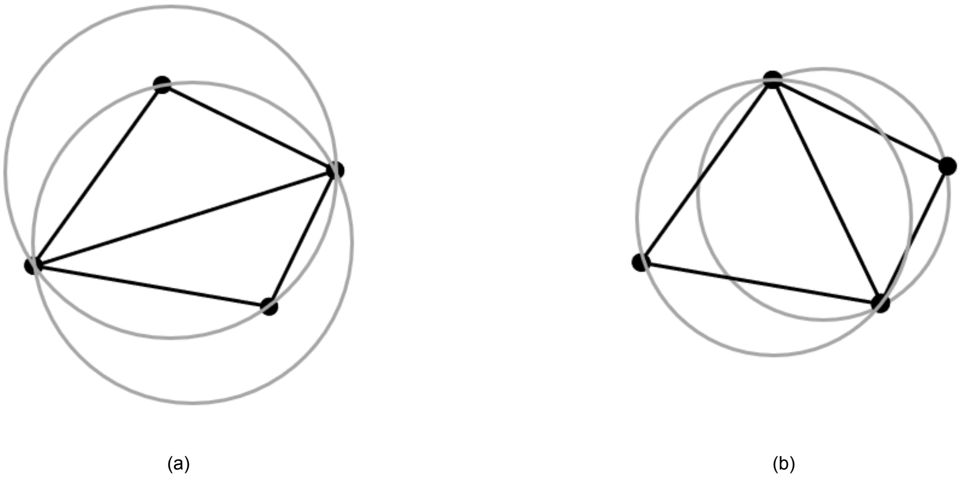


Figure 5.4: Exampular triangulations which (a) do not meet Delaunay's criterion and (b) which does.

terion ([Delaunay, 1934]) which states that no node in the mesh must be within the circumsphere of any triangle/tetrahedron (generally called a simplex). This is schematically drawn for the two-dimensional case in figure 5.4, where figure 5.4a presents a triangulation which does not fit the Delaunay criterion while the triangulation of figure 5.4b does. In three dimensions the triangle simplex is replaced with a tetrahedron.

Next, the barycentric coordinates are computed for each point inside a simplex. Barycentric coordinates λ can be defined for any points inside the simplex and interpreted as masses placed at the vertices of the simplex which determine a unique mass centroid P ([Coxeter, 1969]). This is displayed schematically for the triangle simplex in figure 5.5. λ are coordinates in the barycentric coordinate system of the simplex, where the coordinates of any point P are $(\lambda_A, \lambda_B, \lambda_C)$ for triangle simplices or $(\lambda_A, \lambda_B, \lambda_C, \lambda_D)$ for the tetrahedron simplex. The discussion next will focus on LIDT in three-dimensions. The coordinates $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ and $(0,0,0,1)$ correspond to the vertices A, B, C and D of the tetrahedron. After determining that a point P indeed lies inside the simplex, the goal is to find the barycentric coordinates given the Cartesian coordinates (x, y, z) of P ([Coxeter, 1969]). Firstly they satisfy

$$\lambda_A + \lambda_B + \lambda_C + \lambda_D = 1. \tag{5.1}$$

Next, the coordinates (x, y, z) of P are written as a linear combination of the barycentric coordinates and the vertex coordinates:

$$\begin{aligned} x &= \lambda_A x_A + \lambda_B x_B + \lambda_C x_C + \lambda_D x_D \\ y &= \lambda_A y_A + \lambda_B y_B + \lambda_C y_C + \lambda_D y_D \\ z &= \lambda_A z_A + \lambda_B z_B + \lambda_C z_C + \lambda_D z_D \end{aligned} \tag{5.2}$$

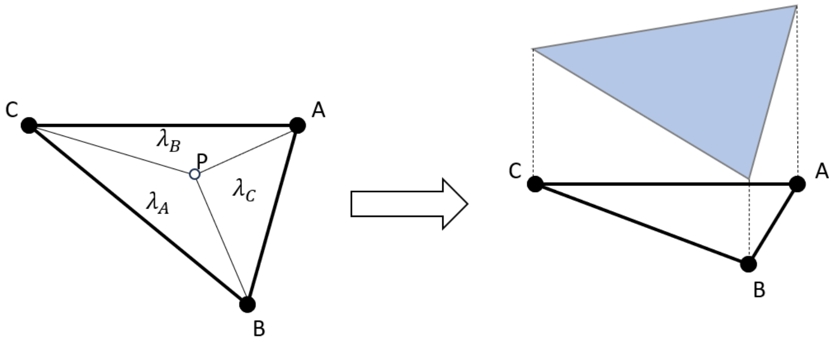
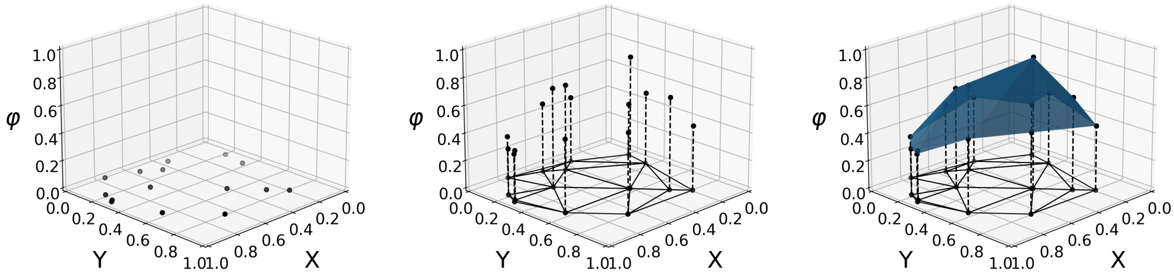


Figure 5.5: Barycentric coordinates $\lambda_{A,B,C}$ of a triangle simplex can be used to linearly interpolate inside the simplex.



(a) (b) (c)

Figure 5.6: The process of triangulating a set of two-dimensional points and consecutively computing the barycentric coordinates which is equivalent to the fitting of a plane inside each triangle.

Combined with equation (5.1), this can be written into the matrix form

$$\mathbf{A}\vec{\lambda} = \mathbf{r} - \mathbf{r_D},$$
$$\text{with } \mathbf{A} = \begin{pmatrix} x_A - x_D & x_B - x_D & x_C - x_D \\ y_A - y_D & y_B - y_D & y_C - y_D \\ z_A - z_D & z_B - z_D & z_C - z_D \end{pmatrix}, \; \vec{\lambda} = \begin{pmatrix} \lambda_A \\ \lambda_B \\ \lambda_C \end{pmatrix} \text{ and } \mathbf{r} - \mathbf{r_D} = \begin{pmatrix} x - x_D \\ y - y_D \\ z - z_D \end{pmatrix}. \tag{5.3}$$

The matrix $\mathbf{A}$ is invertible when none of the vertices of the triangle are colinear, which is true after a Delaunay triangulation. Equation (5.3) can be solved for the vector of λ values and the last barycentric coordinate is then given by equation (5.1). This equation can also be added directly to equation (5.3) which yields the single linear system of equations

$$\begin{pmatrix} x_A - x_D & x_B - x_D & x_C - x_D & 0 \\ y_A - y_D & y_B - y_D & y_C - y_D & 0 \\ z_A - z_D & z_B - z_D & z_C - z_D & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_A \\ \lambda_B \\ \lambda_C \\ \lambda_D \end{pmatrix} = \begin{pmatrix} x - x_D \\ y - y_D \\ z - z_D \\ 1 \end{pmatrix}. \tag{5.4}$$

After the barycentric coordinates are computed based on the coordinates of P, these can be used as weights to linearly interpolate any quantity φ based on the value of φ at the vertices. This is done so with the following relation

$$\varphi_P = \lambda_A\varphi_A + \lambda_B\varphi_B + \lambda_C\varphi_C + \lambda_D\varphi_D \tag{5.5}$$

This process is graphically depicted in figure 5.6 for the two-dimensional case. First the collection of points are triangulated based on the Delaunay criterion. The LIDT implementation in this report utilises the `scipy.spatial.Delaunay()` function of the openly available Scipy package ([Virtanen et al., 2020]) to perform the triangulation. Then, the trilinear interpolation is performed using the barycentric coordinates from equation (5.4) and the weighting of equation (5.5). As shown in subfigure 5.6c, this is equivalent to fitting a plane to the coordinates (x, y, φ) at the vertices for triangle simplices, where x and y are the Cartesian coordinates and φ is the quantity value.

Moving ahead in this effort, Jux' technique will be compared to the against the other methods at the model surface. In that case Jux' method needs to be evaluated outside Ω_{int} and thus the interpolation method outlined above is no longer valid. In that case, Jux' method will be linearly extrapolated. This is done as follows. First, the vector normal $\mathbf{n}$ to the geometry is computed as it defines the direction along which the solution curve will be extrapolated. The point of evaluation P iteratively steps along $\mathbf{n}$ towards Ω_{int} until it falls inside the interpolation domain. Then, the LIDT approach is used to determine both the quantity value φ and the (linear) gradient of the quantity along the wall-normal line $\partial\varphi/\partial n$. The quantity value is then linearly extrapolated along the wall-normal direction to the model wall with the wall-normal gradient as

$$\varphi_M = \varphi_P - \frac{\partial\varphi}{\partial n} n_P, \tag{5.6}$$

where the subscript M denotes the model surface, subscript P refers to the point of evaluation which lies just inside Ω_{int} and n_P is the wall-normal distance travelled to move from M to P. Trivially, the wall-normal gradient of the velocity $\partial\mathbf{u}/\partial n$ remains constant for a linear extrapolation.

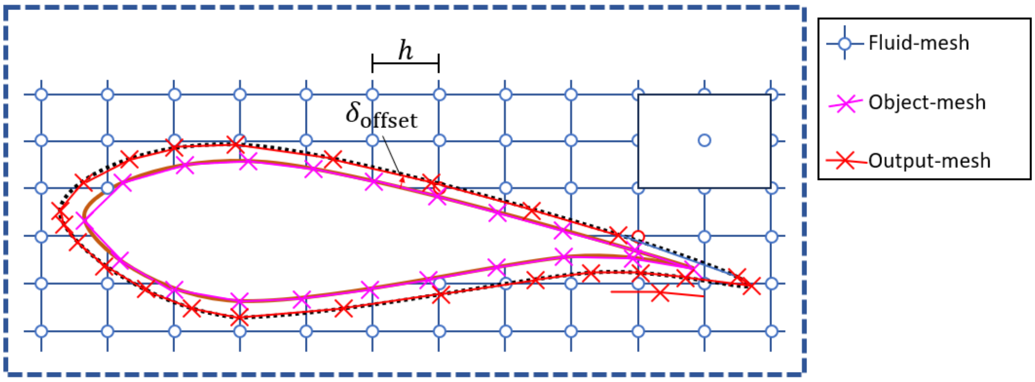
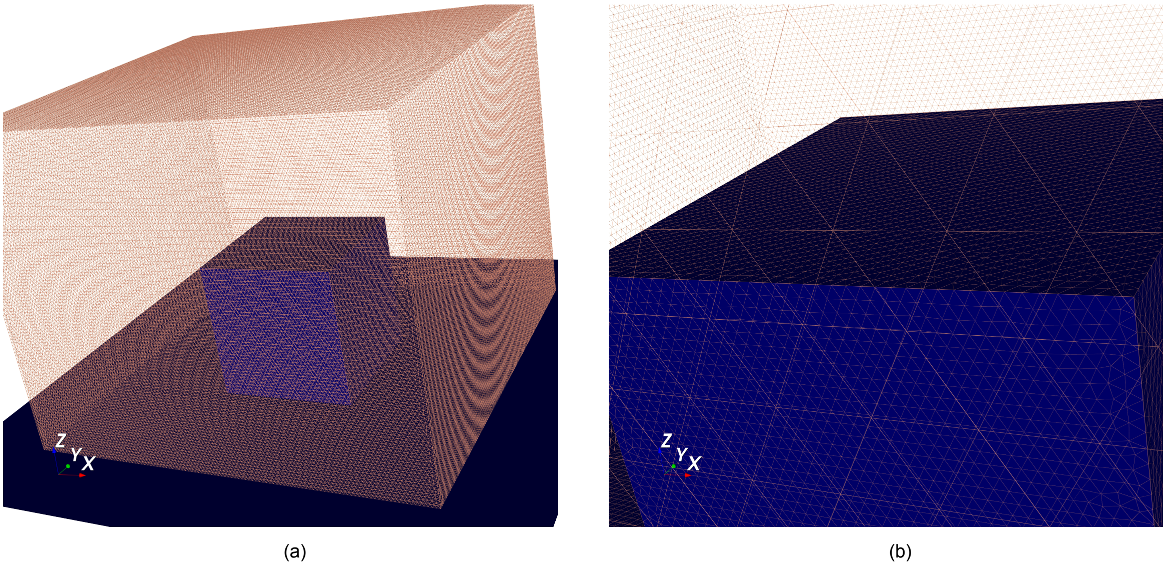


Figure 5.7: Conventions of the object-mesh, fluid-mesh and output-mesh.

5.2. The computational output domain

The model domain $\Omega_{\mathcal{M}}$ and the fluid domain $\Omega_{\mathcal{F}}$ are disconnected and incompatible in their current implementation. The transformation matrix ${}_{\mathcal{M}}\mathcal{T}_{\mathcal{F}}$, obtained from the object registration, aligns the coordinate systems of $\Omega_{\mathcal{M}}$ and $\Omega_{\mathcal{F}}$ such that these can be merged together. The classic approach to CGR, subdividing the entire domain $\Omega_{\mathcal{F}}$ with bins, slices through the model's surface and does not account for its presence. To incorporate information from on the one hand the fluid measurements and on the other hand the model geometry, a new mesh, similar to the approach by Jux et al. [2018], is introduced onto which the information is interpolated. This mesh is referred to as the output-mesh. For convenience, the mesh which holds the model geometry information is shorthandedly referred to as the object-mesh and the mesh which holds the fluid information is termed the fluid-mesh. This convention is shown graphically in figure 5.7. This figure depicts an output-mesh which is offset (dilated) from the model surface. Note that the term fluid-mesh does not specify whether the data is represented as tracer-based or bin-based.

Although depicted in 2D as a surface mesh, the output-mesh can generally be any type of mesh, e.g. triangular or quadrilateral for two-dimensional surface meshes and tetrahedral or hexahedral for three-dimensional volume meshes, or any combination. In this thesis report, all meshes will be constructed with triangular and tetrahedral elements. Contrary to the data stored for geometrical information, the output-mesh holds both geometrical information and information on variables which lie inside this mesh. Such information can lie at vertices, along edges, on faces or inside cells. This can be viewed as a



(a) (b)

Figure 5.8: Example of (a) a volumetric output-mesh, generated around a cube and (b) a close-up of the output-mesh.

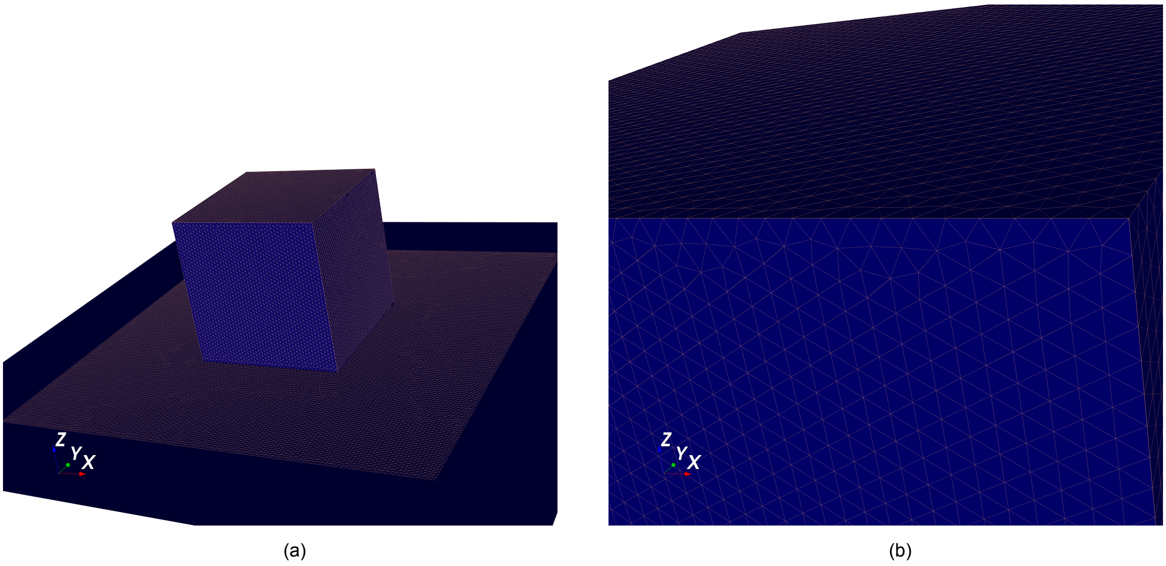


Figure 5.9: Example of (a) a surface output-mesh, generated to align with the cube and (b) a close-up of the output-mesh.

superposition of the data format for the LPT fluid measurements and the model geometry. Figures 5.8 & 5.9 respectively show a volumetric and surface mesh generated around a surface-mounted cube. Note that all images of the object-mesh, fluid-mesh and output-mesh (such as figure 5.11) are generated using the LIVA-console, explained in section 6.2. Concerning the topic of near-surface fluid reconstruction the region of interest is evidently near the model, though the object-informed interpolation presented in this report can be employed everywhere in Ω_{int}.

This section outlines the generation of the output-mesh in subsection 5.2.1. The interpolation onto the output-mesh is localised by using spherical interrogation volumes around the point of interest. This procedure is introduced and discussed in subsection 5.2.2. Lastly, subsection 5.2.3 gives the complete picture how fluid and object measurement are integrated onto an output-mesh.

5.2.1. Generating the output-mesh

The output-mesh occupies the Ω_{int} domain and is bounded by the model geometry. To generate such a mesh an unstructered mesh generation algorithm is employed as reviewed in subsection 3.3.2. These have the advantage that they can handle arbitrarily shaped domains and are thereby more versatile. The construction of a mesh boils down to (1) creating vertices/nodes and (2) connecting these vertices to build the elements of the mesh ([George et al., 2017]). The approach to this unstructured mesh generation can be divided into two branches, shown in figure 5.10. Either the procedure is handled in twofold where first a boundary mesh is created by subdividing the boundary and subsequently a spatial mesh is created from the boundary mesh. On the other hand both steps can be considered simultaneously in a global approach.

The purpose of the mesh is to infer connectivity between quantities in space. When the fluid quantities are only assigned to vertices, it is also possible to define an array of points which occupies the given Ω_{int} or any other domain such as the surface of the model. However, in such a case the points are unconnected which significantly complicates plotting of contours and streamlines in three dimensions. For more information on mesh geometries and generation, the reader is referred to [de Berg et al., 2008] and [Tu et al., 2018].

In this report the unstructured meshes are generated using the open-source mesh generator *Gmsh* [Geuzaine and Remacle, 2009], unless mentioned otherwise. Gmsh offers a generic and simple interface to define mesh boundaries, load model geometries and generate unstructured surface or volume meshes. The meshes shown in figure 5.8 & 5.9 have both been created with Gmsh. Refining and maximising the output-mesh quality is not considered in this report, partly due to the reconstruction approach which is dissimilar from CFD numerical solutions, which requires refinements of the mesh to accomodate for the scales in the flow, such as finite-element methods. The only requirement imposed on the mesh is to yield a homogeneous distribution of surface elements with similar aspect ratios to

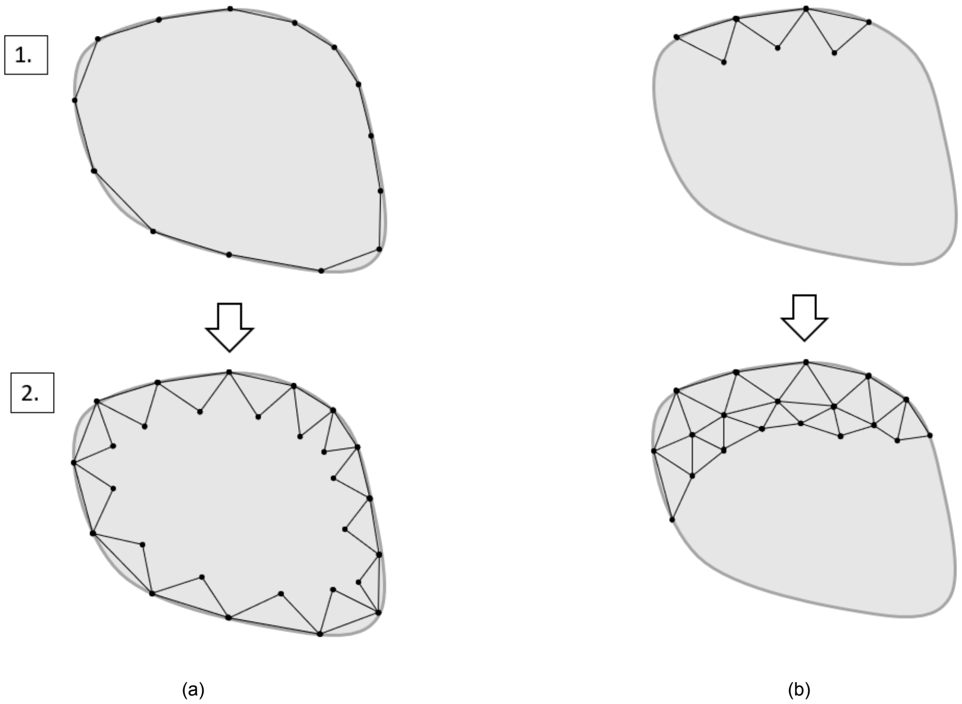


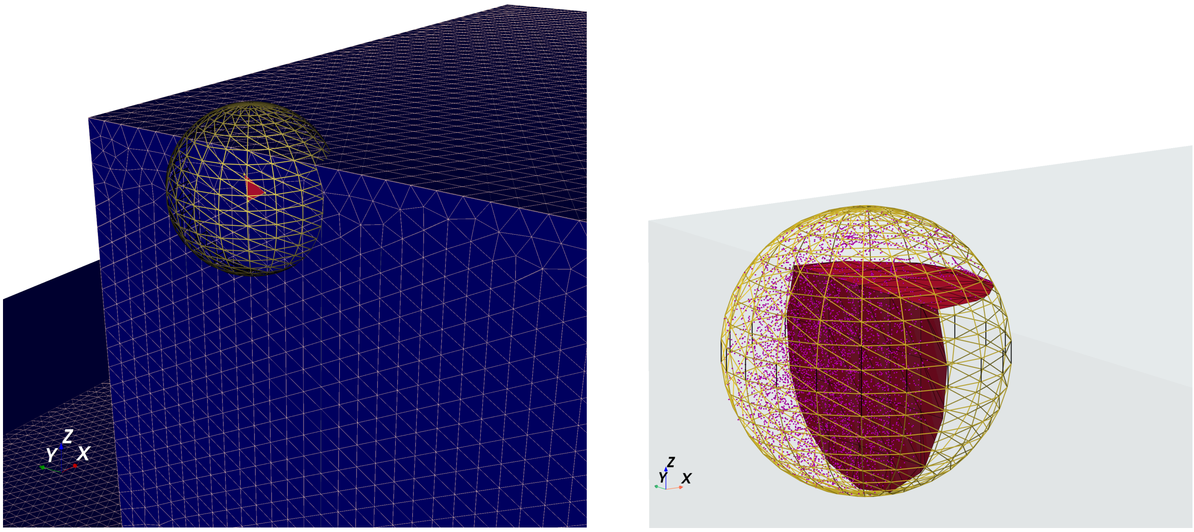
Figure 5.10: The two approaches to meshing a domain Ω by (a) first discretising the boundary and subsequently meshing the space that is bounded or (b) considering a global approach where the boundary and interior are meshed simultaneously.

avoid sparsely populated areas. For this reason, simply subdividing a tessellated STL-mesh is also not recommend. An STL file may contain high aspect ratio triangle facets. Subdividing such facets by splitting into two does not adequately resolve the high aspect ratio and leads to an inhomogeneous cell distribution similar to a pie that is subdivided into high-aspect ratio slices.

5.2.2. Local interrogation volume approach

The information from the fluid measurements and the model geometry (i.e. no-slip) are combined in an interpolation scheme which approximates a local part of the fluid volume around the point of interest. This localisation is performed with spherical interrogation volumes that capture the section of the fluid measurements and model geometry that lie inside it. Spheres are initially used as these isotropic geometries do not introduce a bias in the interpolation scheme and are expected to behave similarly for planar and curved surfaces. This approach is comparable to binning, discussed in subsection C.2.2, yet the interrogation volume also considers and accounts for the presence of the model's surface. Using this local interrogation volume approach (LIVA) has the advantage that the global problem is split into several smaller local problems. Each local problem is solved to return the respective local interpolation function.

Two geometrical operations are required with LIVA, namely the registration of fluid-mesh nodes and object-mesh cells. The registration of fluid-mesh nodes in the neighbourhood of a point can be solved using spatial locators, such as octrees and kd-trees. In this report, the problem is solved using a standard VTK-library tool for locating points ([Schroeder et al., 2006]). This function utilises a spatial search object which divides the specified domain into a regular array of buckets (again comparable to bins) and keep a list of points which fall inside each bucket. The registration of the model geometry inside the sphere is handled by means of a Boolean mesh operations. A discretised sphere is generated centred at the point of interest. Then the boolean intersection is computed using a standard VTK-library tool ([Schroeder et al., 2006]). The registration of the fluid-mesh and object-mesh in the LIVA is depicted for a point near the upper edge of a cube in figure 5.11. The fluid-mesh is represented with tracers, though the same operation is done for a fluid-mesh from binned LPT results. The most important characteristic of the interrogation volume is the sphere radius, which is dependent on the interpolation scheme, the concentration of fluid-mesh nodes and, in the case of a tracer-based representation, the



(a) (b)

Figure 5.11: A visual example indicating the LIVA-approach. A spherical volume is generated around an interrogation point in (a). Then both the fluid-mesh and the object-mesh are sliced with this sphere, yielding a local volume which contains part fluid-mesh data and part object-mesh data.

criterium for statistical convergence. These aspects are discussed with the interpolation scheme in chapter 5. Inside the interrogation volume, a local Cartesian coordinate system (x^*, y^*, z^*) is defined.

The origin of the local coordinate system is placed at the interrogation point, i.e. the centre of the interrogation volume. When handling a volumetric output-mesh, the local coordinate system need not be rotated and aligns with the global Cartesian coordinate system. However, in the case of a surface output-mesh, the local coordinate system is rotated, such that the local z^*-coordinate aligns with the normal direction $\mathbf{n}$ of the cell. This rotation simplifies operations such as introducing a semi-sphere –only considering $z^* > 0$– or determining the wall-normal velocity gradient. It should be noted that the direction of the local x^*- and y^*-coordinates need not be known explicitly and may remain arbitrary.

The three-dimensional translation from the global coordinate system to the local coordinate system centred at the point $P(x_P, y_P, z_P)$ consists of a translation first and, when handling a surface output-mesh, a rotation based on the Rodrigues' rotation ([Friedberg, 2022]). The translation is simply given by the four-dimensional rotation matrix

$$\mathbf{T}_{\text{tl}} = \begin{pmatrix} 1 & 0 & 0 & -x_p \\ 0 & 1 & 0 & -y_p \\ 0 & 0 & 1 & -z_p \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.7}$$

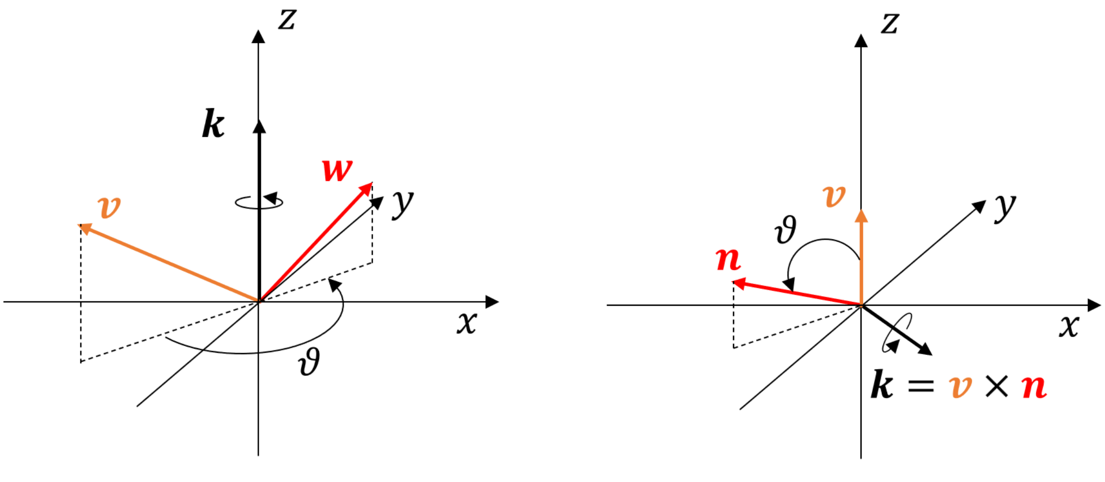
The 4×4 transformation matrix is applied to a slightly modified point vector $P = [x_p, y_p, z_p, 1]^T$. If the local coordinate system need also be rotated, the Rodrigues' rotation matrix is constructed to align the local z^*-coordinate with the normal $\mathbf{n}$. The Rodrigues' rotation swivels a vector $\mathbf{v}$ around a rotation axis $\mathbf{k}$ by the angle ϑ, using the right-hand system to define the positive angle direction.

This process is shown schematically in figure 5.12. First, the traditional Rodrigues' transformation is defined in figure 5.12a, where the source vector $\mathbf{v}$ is rotated around the rotation axis $\mathbf{k}$ by an angle ϑ to obtain the destination vector $\mathbf{w}$. To align the local z^*-coordinate with the normal $\mathbf{n}$, it is necessary to determine both the rotation axis $\mathbf{k}$ and the angle ϑ. This is shown in figure 5.12b, where the source vector $\mathbf{v}$ is the unit vector $(0, 0, 1)^T$. The destination vector is $\mathbf{n}$. The rotation vector $\mathbf{k}$ is then the cross product of the source vector and destination vector:

$$\mathbf{k} = \hat{\mathbf{v}} \times \hat{\mathbf{n}}, \tag{5.8}$$

where the hat-operator $(\hat{\phi})$ denotes a normalised vector. The angle of rotation ϑ is determined using the scalar product relation via

$$\vartheta = \arccos\left(\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}\right). \tag{5.9}$$



(a) (b)

Figure 5.12: Example of the three-dimensional rotation using Rodrigues' transformation for (a) the general case where a vector $\mathbf{v}$ is rotated by an angle φ around the rotation axis $\mathbf{k}$ to obtain the new vector $\mathbf{w}$. The case of the global to local transformation is shown in (b), where the global up vector $\mathbf{v}$ must be rotated to align with the local normal $\mathbf{b}$, which requires the computation of the axis of rotation $\mathbf{k}$ and the angle ϑ.

Then, the Rodrigues' rotation matrix can be composed. This 4×4 transformation matrix is defined as ([Friedberg, 2022]):

$$\mathbf{T}_{\text{rot}} = \mathbf{I} + (\sin\vartheta)\mathbf{K} + (1 - \cos\vartheta)\mathbf{K}^2, \tag{5.10}$$

$$\text{where } \mathbf{I} = \text{eye}(4), \text{ and } \mathbf{K} = \begin{pmatrix} 0 & -\hat{k}_z & \hat{k}_y & 0 \\ \hat{k}_z & 0 & -\hat{k}_x & 0 \\ -\hat{k}_y & \hat{k}_x & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.11}$$

Lastly, the translation and rotation can be combined in a single global-to-local transformation matrix ${}_G\mathbf{T}_L$ as

$${}_G\mathbf{T}_L = \mathbf{T}_{\text{rot}}\mathbf{T}_{\text{tl}}. \tag{5.12}$$

5.2.3. Merging the fluid domain and model domain - the complete picture

The output-mesh proposed in this thesis report serves a different purpose, but is nonetheless similar to the one employed by Jux et al. [2018]. In this situation the output-mesh offers a platform to combine both LPT fluid measurements and the results from the object registration through a locally defined interpolation function. The proposed approach aims to answer research question **Ia**, regarding the ***Domain discretisation***. Figure 5.13 depicts the flow of information to derive the output-mesh, introduce isotropic interrogation volumes to define local interrogation functions and estimate near-surface fluid dynamical quantities such as wall-shear stresses. After performing the LPT experiment, the images are processed and particle trajectories are reconstructed using IPR and STB as per the state-of-the-art methods. The results can be be returned either in the tracer-based representation or the bin-based representation, referred to as the fluid-mesh. Simultaneously surface markers are registered on the object with the same LPT camera setup. This allows an object registration as discussed in section 2.3, yielding the object-mesh. The domain Ω_{int} of the fluid-mesh and the bounding object-mesh are used to generate an output-mesh. This can be a volumetric or surface mesh when connectivity between cells and vertices is desired, or simply an array of data points when this is not required. Spherical interrogation volumes are introduced to turn the global problem into multiple local problems, which consider and account for the presence of the model's surface. Inside these spherical volumes, a velocity reconstruction method is utilised which combines the fluid data and model geometry and allows an estimation of near-surface fluid dynamics. The velocity reconstruction method is the topic of the next chapter.

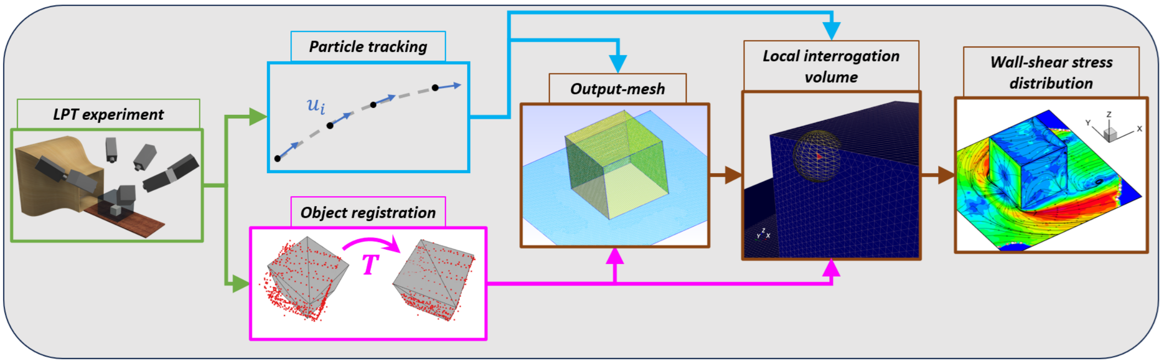


Figure 5.13: Flowchart indicating the different components that are merged with the LIVA approach, after which a velocity reconstruction scheme can be applied to determine near-wall flow properties.

5.3. Fluid reconstruction with LIVA interpolation schemes

The lack of information between the domain Ω_{int}, created from a binned fluid-mesh, and the model surface imposed the need for dilating the outer surface. As explained in section 5.1, this lack of data is alleviated by the introduction of the model geometry. The model surface adds important information on the velocity through the no-slip condition ($\mathbf{u}|_w = \mathbf{0}$). The LIVA establishes smaller fluid volumes that contain fluid-mesh nodes and object-mesh cells. The fluid-mesh carries with it the fluid motion, while the object-mesh cells impose zero velocity. The information from both domains is combined through an interpolation scheme to reconstruct the gap between Ω_{int} and the model surface.

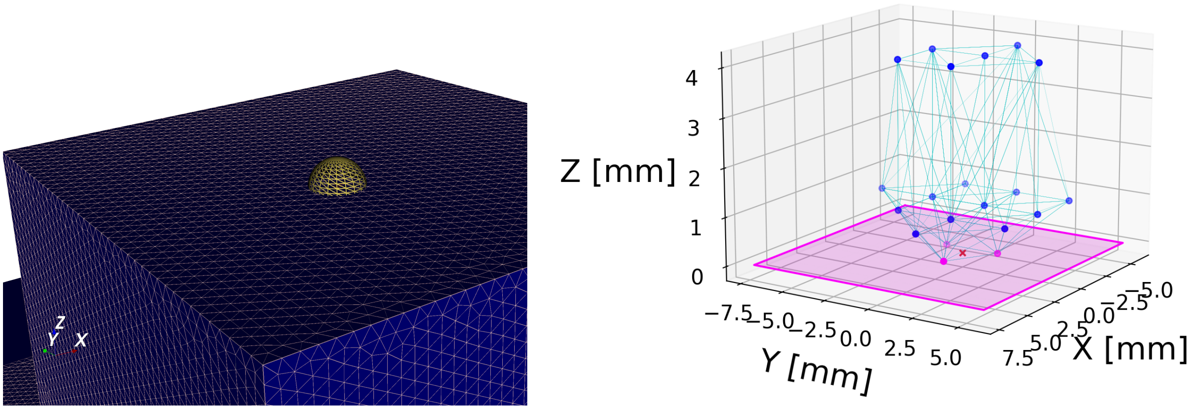
This section introduces a family of interpolation schemes which built upon the method of Jux et al. using LIVA. Therefore, a binned-representation of the fluid-mesh using polynomial fits is used first. Subsection 5.3.1 a scheme similar to Jux' method, but including a vertex-discretised representation of the object-mesh referred to as *wall ghost particles*. Next, the order of the interpolation scheme is increased and a quadratic functional is fitted inside the LIVA volume with a bin-based fluid-mesh, outlined in subsection 5.3.2. Then, subsection 5.3.3 introduces the use of a tracer-based fluid-mesh over a bin-based fluid-mesh. Lastly, the wall ghost particles are replaced for a constraint fit of the quadratic functional in subsection 5.3.4. Lastly, The representation of wall-shear stresses on a surface output-mesh is explained in subsection 5.3.5.

5.3.1. Trilinear fit with Delaunay triangulation including wall discretisation

Continuing from the discussion of Jux' method, the first in the family of proposed methods is a linear reconstruction approach which incorporates a discretised representation of the model surface. This method is termed *Bin-Con-Lin* (Bin-Constrained-Linear). The object-mesh is discretised with a similar spacing as the fluid-mesh, h. These vertices are then included into the LIDT-approach which was outlined in subsection 5.1.3. The discrete vertex-representation of the object-mesh is displayed in figure 5.3. By including wall points with a known zero velocity, the interpolation domain is increased up to the model surface. A spherical interrogation volume of radius $R_{\text{LIVA-BCL}}$ is created around the interrogation point. The fluid-mesh and object-mesh nodes which fall inside the volume are registered and a three-dimensional Delaunay triangulation is generated. The object-mesh nodes are determined by projecting the interrogation point onto the object geometry and extracting the object-mesh cell that contains the projected point.

This process is shown for an interrogation point on top of a cube in figure 5.14, where the interrogation sphere is shown in subfigure 5.14a and the Delaunay triangulation with a single cell from the object-mesh in subfigure 5.14b. The interrogation point lies on the surface geometry in the example and is denoted by the red cross. Note that the plot is given in terms of the local coordinate system (x^*, y^*, z^*) which is centred at the interrogation point. The corresponding cell of the object-mesh is represented by the triangle with three fuchsia vertices. The seventeen fluid-mesh points that fall inside the sphere are denoted by the blue nodes. The triangulation with three-dimensional simplices is depicted by the cyan edges.

The radius of the sphere $R_{\text{LIVA-BCL}}$ is chosen such that the interrogation point can be guaranteed to be in the convex hull ([de Berg et al., 2008]) of the union of fluid-mesh and object-mesh nodes, cap-



(a) (b)

Figure 5.14: Visualisation of the three-dimensional triangulation for (a) a point on the top surface with (b) the set of triangulated tetrahedrons in a local coordinate system.

tured by the local interrogation volume. This condition naturally forms a minimal radius requirement. The maximum radius is limited by computational cost as the triangulation becomes more expensive with more nodes. The bins form a Cartesian grid with a spacing given by h. Therefore, the radius is defined such that the interrogation sphere always includes the eight nodes of a single ”grid bin” (with edges of size h) in which it lies. The edge case is where the interrogation point lies near one of the nodes and the furthest grid bin node is the opposite node. From this it can be determined that $R_{\text{LIVA-BCL}} > \sqrt{3}h$.

5.3.2. Quadratic fit of binned data with wall ghost points

To alleviate the expected shortcomings of the trilinear fit from subsection 5.3.1, the next method proposes an increased order and range of the interpolation scheme. This is achieved by replacing the trilinear interpolation with the fit of a quadratic polynomial over the interrogation volume, which is called method *Bin-Nudge-QuadFit* (Bin-Nudged-Quadratic). Fitting of higher-order velocity functions is desired to better approximate the boundary layer velocity profiles which is non-linear along the normal distance due to the boundary layer presence. This approach additionally extends the effect of the no-slip condition to points which lie further inside the fluid-mesh and away from the object-mesh.

The radius of the interrogation sphere $R_{\text{LIVA-BNQ}}$ determines the domain in which a quadratic velocity functional is locally fitted, as was similarly done by Agüera et al. [2016]. The second-order, three-dimensional polynomial is defined in local coordinates as

$$\begin{aligned} u(x^*,y^*,z^*) = a_0{+}a_1 \cdot x^* + a_2 \cdot y^* + a_3 \cdot z^* + a_4 \cdot \left(x^*\right)^2 + a_5 \cdot \left(y^*\right)^2 + \dots \\ a_6 \cdot \left(z^*\right)^2 + a_7 \cdot x^* y^* + a_8 \cdot x^* z^* + a_9 \cdot y^* z^* \end{aligned} \tag{5.13}$$

The velocity functional is then fitted to measurement points which come from the fluid-mesh and measurements belonging to the object-mesh. Wall-ghost (WG) points are introduced which aim to enforce the no-slip condition at the wall. WG points are generated on the object-mesh cells and introduce the no-slip condition. Two criteria are important in for this approach, namely the radius of the sphere, $R_{\text{LIVA-BNQ}}$, and the number of WG points that are introduced, $N_{p,WG}$.

Firstly, the requirements on the sphere radius $R_{\text{LIVA-BNQ}}$ are discussed. An initial lower bound on the minimum radius can be determined from the conditions for a least-squares fit. The velocity functional of equation (5.13) contains ten parameters m, which requires at least ten measurement points to solve the least-squares fit. Using a Monte Carlo simulation it is then possible to determine the minimum radius, R_{min}, in a three-dimensional Cartesian grid with grid vector spacing h to ensure to encapsulate ten fluid-mesh nodes. This is done by distributing points inside a cube with edges of size h with a uniform random distribution. This cube is placed in a Cartesian grid of sample fluid-mesh nodes with the same grid spacing, such that the cube vertices coincide with a fluid-mesh node, Then, for each point the radius at which ten fluid-mesh nodes are encapsulated is determined. The results of this Monte Carlo simulation with $N = 10^6$ are shown in figure 5.15. The approximated probability density function

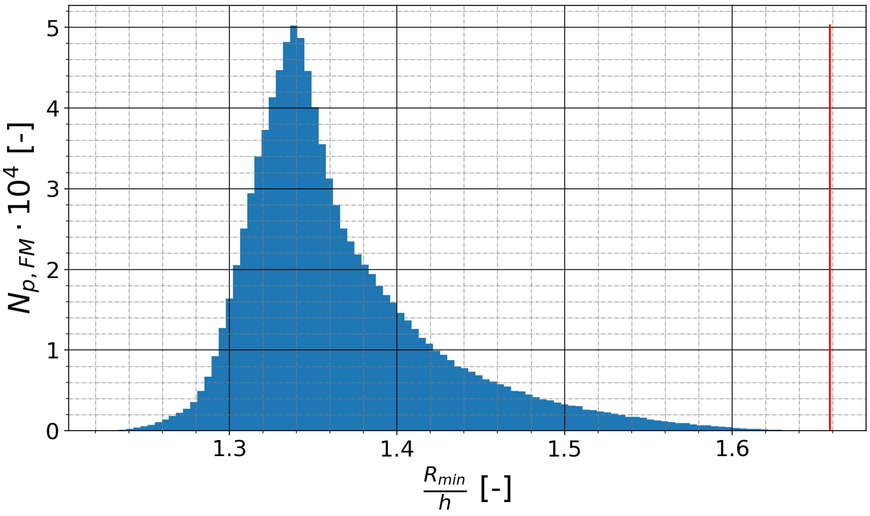


Figure 5.15: Results of Monte Carlo simulation to determine probability distribution of minimum radius for a randomly placed point to contain ten fluid-mesh nodes. The vertical red line represents the largest minimum radius $R_{min}/h = \sqrt{11/4}$, which corresponds to the interrogation point aligning with a fluid-mesh node.

indicates that most point locations require a minimum radius of $R_{\text{min}}/h = 1.34$ and that ten fluid-mesh nodes can be guaranteed for a minimum radius of $R_{\text{min}}/h \approx 1.66$. It can be verified that this location coincides with the point that lies equidistant from the surrounding eight fluid-mesh nodes, i.e. it lies in the centre of the Cartesian grid cube. This allows for an exact expression of the minimum radius as $R_{\text{min}}/h > \sqrt{11/4} \approx 1.6583$. Thus to guarantee a least-squares velocity fit of equation (5.13), the lower bound of the radius $R_{\text{LIVA-BNQ}}$ is $\sqrt{11/4}\ h$ to capture enough fluid-mesh nodes.

Next, WG points are generated on the intersection of the interrogation volume and the model surface. These points will be included in the least-squares velocity fit and can be interpreted as weights which pull (or nudge) the velocity to zero at the model surface. When not explicitly adding weights to these points in the least-squares fit, the number of generated WG points constitute a weight in and of itself. Generating few WG points shall very weakly nudge the velocity towards no-slip at the boundary, whereas generating many WG points may adversely affect the velocity away from the wall. To cut through this dilemma, the following is proposed. The number of generated WG points are chosen such that the inter-point distance is similar to the inter-nodal distant of the fluid-mesh. For the case of the binned data, the inter-nodal distance is simply the grid vector spacing h. The WG points will thus be generated with a similar inter-point distance.

The WG point distribution is obtained via a Poisson disk sampling scheme. The samples in a Poisson disk distributions are at least a distance r apart from each other, yielding a homogeneous representation of a surface. Consider for example the sampling of cyclist model in figure 5.16. First, the cyclist surface mesh (represented by a .STL file) is shown in figure 5.16a. Next, points are uniformly sampled on the mesh which yields the sample shown in figure 5.16b, with the colour varying from blue to red

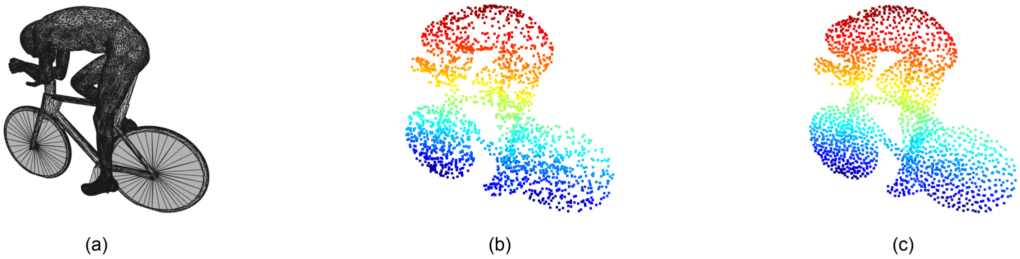

(a) (b) (c)

Figure 5.16: Example of sampling strategies for (a) a cyclist CAD model, using (b) random sampling and (c) Poisson disk sampling.

based on z-position. The uniformly sampled points clearly do not yield a homogeneous distribution. A Poisson disk sampling of the same cyclist mesh is shown in figure 5.16c, where the colour varies based on the z-coordinate. The Poisson disk sampling evidently provides a more homogeneously distributed point sample. The point samples in figure 5.16b and 5.16c are generated with 1000 points.

Typically, Poisson disk sampling is initialised by a user-supplied radial distance r, which defines the Poisson disk radius ([Bridson, 2007]). The Open3D ([Zhou et al., 2018]) library provides a library with convenient tools for point sampling. Their implementation allows to specify a number of points for sampling, i.e. a number of WG points, $N_{p,WG}$. The implemented Poisson disk sampling algorithm is based on [Yuksel, 2015]. This requires the determination of the number of WG points. The number of points per area C_A $[\mathrm{p\ m^{-2}}]$ can simply be obtained from the grid vector spacing as

$$C_A = \frac{1}{h^2} \tag{5.14}$$

and the number of sampled WG points is then obtained with the area of the intersected wall geometry, $A_{\text{LIVA-geom}}$, via

$$N_{p,WG} = C_A \cdot A_{\text{LIVA-geom}}. \tag{5.15}$$

The sampled WG points and fluid-mesh nodes are combined in a single array. The point coordinates (x^*, y^*, z^*) are used to set up the coordinate matrix $\mathbf{X}$ as

$$\mathbf{X}_{ij} = \phi_j(\mathbf{x}_i^*), \text{ with } u^* = \sum_{j=1}^{m} a_j \phi_j(\mathbf{x}_i^*), \text{ following equation (5.13).} \tag{5.16}$$

The velocity values u, v and w are fitted separately. The quantities respectively form the matrices $\mathbf{Y}^{(u)}$, $\mathbf{Y}^{(v)}$ and $\mathbf{Y}^{(w)}$ as

$$\mathbf{Y}_j^{(\varphi)} = \varphi(\mathbf{x}_\mathbf{j}^*), \tag{5.17}$$

where φ represents a velocity component. Then, the vector of coefficients $\mathbf{a}$ is obtained per velocity component via

$$\mathbf{a}^{\varphi} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}^{\varphi}. \tag{5.18}$$

The fitting of a polynomial velocity functional in the local coordinate system provides a convenient way to determine the velocity and velocity gradient in the normal direction –when using a surface output-mesh and rotating the local coordinate system as explained in subsection 5.2.2. The velocity at the interrogation point is simply the coefficient a_0. The velocity gradients at the interrogation point in the x^*-, y^*- and z^*-direction are given by the coefficients a_1, a_2 and a_3, respectively. Now, remembering that the z^*-coordinate is aligned with the outward normal, the gradient $\partial u/\partial z^*$ is equivalent to $\partial u/\partial n$. These quantities are directly used for representing wall-shear stress distributions on three-dimensional surfaces, which is detailed in subsection 5.3.5.

The application of a LIVA interrogation volume with a local polynomial fit can be interpreted as a spatial averaging filter, similar to the Savitsky-Golay filter. The reconstruction scheme outlined above and termed method *Bin-Nudge-QuadFit* applies local interpolation on a binned fluid-mesh and object-mesh represented by WG points, as a continuation from the state-of-the-art. This double filtering can, however, introduce some unwanted effects. An example a double binning with local polynomials is shown for a one-dimensional signal, s in figure 5.17. The signal is plotted in the top row consisting of four wave numbers and a nonzero mean. The wave numbers are clearly identifiable in the frequency spectrum. Next, s is binned with quadratic polynomials as per the method of Agüera et al. [2016] to yield s_b, which is shown in the middle row. The size of the bin is shown in the lower left. The combination of the bin size and grid vector spacing –and to some extent the order of the polynomial– allows for reconstructing s up to a cut-off frequency. The mean and first and second wave numbers are resolved nearly exactly and so is most of the signal energy (area under the frequency curve). However, the third and fourth wave numbers are lost and an additional frequency is incorrectly introduced. Then, a second binning is done on the already binned signal, yielding s_{b-b}. At best, this second binning will return exactly s_b and not the true signal s. Additionally, the frequency spectrum of s_{b-b} can be seen to be further distorted due to the second binning. The largest wave numbers are still present and identifiable, yet the peaks decrease and energy diffuses to other wave numbers. Therefore, a LIVA interpolation scheme is likely better applied to the original tracer-based representation of the fluid-mesh. This forms the basis of the next proposed method in subsection 5.3.3.

5.3.3. Quadratic fit of tracer data with wall ghost points

Subsection 5.3.2 proposed the fitting of a quadratic velocity functional over a trilinear fit based on a Delaunay triangulation. As explained at the end, the double binning can likely be improved by directly using the tracer-based representation of the fluid-mesh. This subsection details such an approach, which is termed *Tracer-Nudge-QuadFit* (Tracer-Nudged-QuadraticFit). Note that this method is nearly identical to the method *Bin-Nudge-QuadFit*, baring two differences.

Firstly, the tracer-based fluid-mesh representation is used, which was outlined in subsection C.2.1. The tracers are measured along trajectories over multiple time steps. Thus for each time step k, flow information is available on a set of tracers. These are combined into a single large ensemble to which the LIVA interrogation volumes are applied. The intersection of the wall model is determined and WG points are generated on this surface via a Poisson disk sampling.

The second difference is the determination of the concentration for this approach. Using a bin-based representation, the particle concentration could be simply determined from h via equation (5.14). For the tracer-based representation the particle concentration varies throughout the domain, as the bubbles are not easily advected through shear layers and boundary layers. However, to simplify and reduce computational costs, a single concentration, C, is used, dependent on the mean inter-particle distance λ_p. This is determined from the global particle to volume ratio $N_p/\mathcal{V}$ as

$$\lambda_p = \sqrt[3]{\frac{3\cdot\mathcal{V}}{4\cdot N_p}}. \tag{5.19}$$

The factor $N_p/\mathcal{V}$ is determined by taking a large bounding box around the model. The number of particles inside this bounding box defines N_p, whereas $\mathcal{V}$ is given by the bounding box volume minus the volume of the model. The areal concentration is then determined with equation (5.14), using λ_p instead of h.

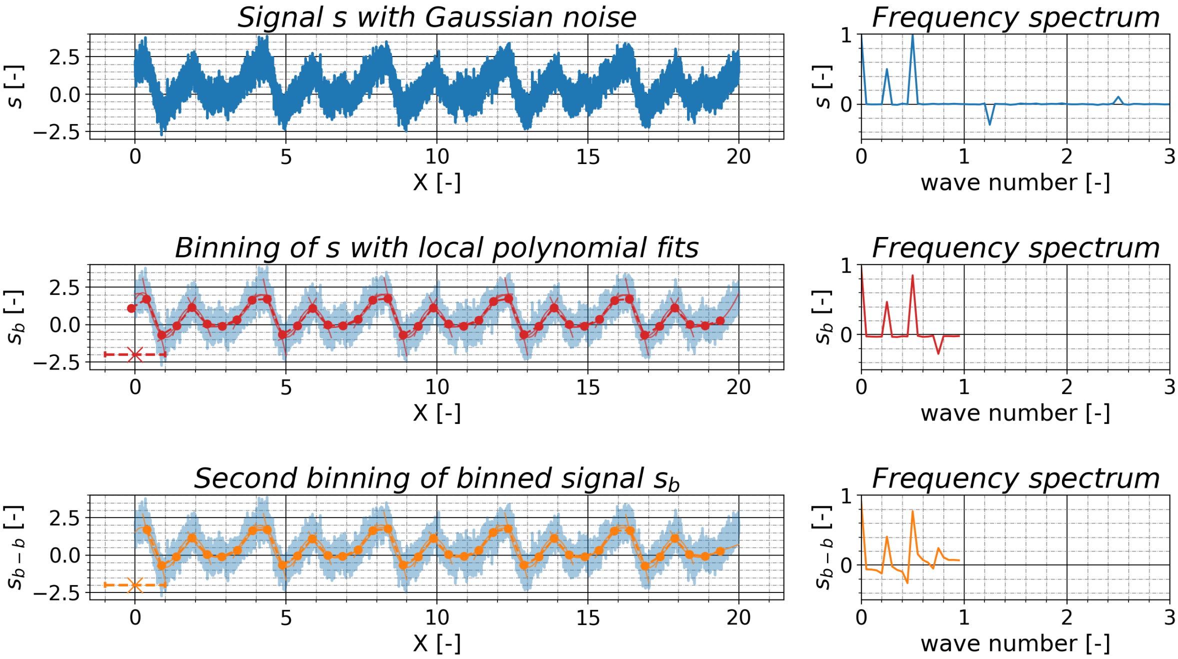


Figure 5.17: Effect of double bin filtering applied to a signal s with Gaussian noise. The signal is made up of four wave numbers and a mean. The signal is subsequently binned with local polynomial fits, which crops the frequency spectrum. Applying a second binning on the signal s_b further distorts the frequency spectrum.

5.3.4. Quadratic fit of tracer data with single wall constraint

The nudging of the three-dimensional quadratic velocity functional towards zero at the wall using WG points does not pose a hard constraint on the solution. Realistically a nonzero velocity will be returned when the solution is evaluated at the wall with method *Tracer-Nudge-QuadFit*. To circumvent this issue, the linear least squares fit is replaced by a *constrained least squares fit*. This approach is termed *Tracer-Con-QuadFit* (Tracer-Constrained-QuadraticFit).

The WG points are substituted for a single constraining point at the wall. This wall-constrained (WC) point P_{WC} is obtained by projecting the interrogation point onto the model geometry. The projection is equivalent to determining the nearest point on the model surface. The constrained least squares is then set up as follows

$$\begin{aligned} \text{minimise} \quad & ||\mathbf{Xa} - \mathbf{Y}^{(\varphi)}||^2 \\ \text{subject to} \quad & \mathbf{Ca} = \mathbf{0}, \end{aligned} \tag{5.20}$$

where the solution is sought which minimises the squared distance between the measurements $\mathbf{Y}^{(\varphi)}$ and the quadratic velocity fit, while the solution is subjected to the no-slip condition at the point P_{WC} expressed via the linear constraint

$$\mathbf{Ca}, \text{ with } C_i = \phi_i(P_{WC}). \tag{5.21}$$

The constrained least squares problem can be solved by the introduction of Lagrange multipliers which have been previously reviewed in subsection 3.3.1 when discussing data assimilation. This problem can be solved using the *KKT equations* ([Boyd and Vandenberghe, 2018]). These equations combine the the minimisation problem and constraint into the single matrix equation

$$\begin{pmatrix} 2X^T X & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \lambda \end{pmatrix} = \begin{pmatrix} 2X^T \mathbf{Y}^{(\varphi)} \\ \mathbf{0} \end{pmatrix}, \tag{5.22}$$

where λ is the Lagrange multiplier introduced to satisfy the constraint. Equation (5.22) can then be solved for the vector containing $\mathbf{a}$ and $\vec{\lambda}$.

5.3.5. Representing wall-shear stress distributions on 3D surfaces

After the velocity is reconstructed using one of the approaches outlined above, the wall shear stress can be determined. As was shortly discussed in subsection 5.3.2, the rotation of the local coordinate system such that the z^* aligns with the normal $\mathbf{n}$ offers a convenient way to extract the velocity gradient along the normal. The wall shear stress is given by

$$\tau_w = \mu \left. \frac{\partial \mathbf{u}_{//}}{\partial n} \right|_{\text{wall}}, \tag{5.23}$$

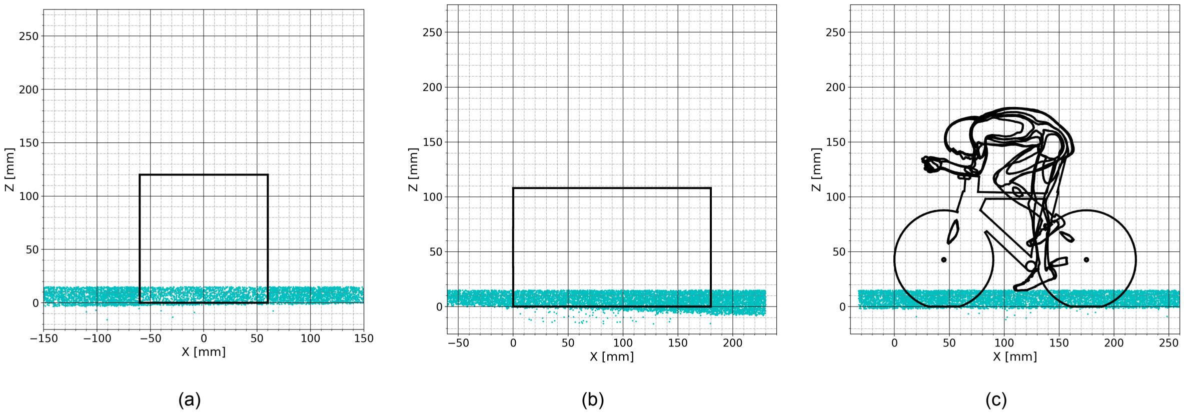
where μ is the dynamic viscosity and $\mathbf{u}_{//}$ is the tangential velocity vector. Although the position of the velocity vectors are transformed with the matrix ${}_G\mathcal{T}_L$, the direction of the velocity vectors is expressed in the global coordinate system with u, v and w which lie along the x-, y- and z-coordinates, respectively. Using the reconstruction schemes outlined in section 5.3, the total velocity gradient along the normal $\partial \mathbf{u}/\partial n$ can be computed, either using a forward difference for the trilinear fit or simply with the coefficient a_3 for the approaches utilising the quadratic fit of equation (5.13).

Next, the tangential velocity gradient along the normal direction can be obtained via the decomposition of the gradient vector $\partial \mathbf{u}/\partial n$ into a normal and tangential component

$$\frac{\partial \mathbf{u}}{\partial n} = \left(\frac{\partial \mathbf{u}}{\partial n}\right)_{//} + \left(\frac{\partial \mathbf{u}}{\partial n}\right)_{\perp} = \frac{\partial \mathbf{u}_{//}}{\partial n} + \frac{\partial \mathbf{u}_{\perp}}{\partial n}. \tag{5.24}$$

The tangential direction was not explicitly determined from the Rodrigues' rotation as discussed in subsection 5.2.2 and this is not needed either. The normal vector is known explicitly and can be used to determine the wall-normal component of equation (5.24) via

$$\frac{\partial \mathbf{u}_{\perp}}{\partial n} = \left(\frac{\partial \mathbf{u}}{\partial n} \cdot \hat{\mathbf{n}}\right) \hat{\mathbf{n}}. \tag{5.25}$$



(a) (b) (c)

Figure 5.18: Plots of the misalignment of the tracers and the experimental models. Contours of the model are overlayed with tracer positions below $z = 10\,\mathrm{mm}$.

Then equation (5.25) is substituted into equation (5.24) to yield an equation for the tangential component of the velocity gradient along the normal direction:

$$\frac{\partial \mathbf{u}_{//}}{\partial n} = \frac{\partial \mathbf{u}}{\partial n} - \frac{\partial \mathbf{u}_{\perp}}{\partial n} = \frac{\partial \mathbf{u}}{\partial n} - \left(\frac{\partial \mathbf{u}}{\partial n} \cdot \hat{\mathbf{n}} \right) \hat{\mathbf{n}} \tag{5.26}$$

5.4. Object registration correction for ground plane

The experimental datasets outlined in chapter 4 are the results of particle tracking the three-dimensional flow around three wall-mounted models. An object registration was performed for each case to seek the transformation matrix ${}_{\mathcal{M}}\mathcal{T}_{\mathcal{F}}$, which aligns the model domain and fluid domain with an accuracy in the order of $10^{-1}\,\mathrm{mm}$. The same can not be said for the ground plane which position could not be accurately determined with an object registration due to various reasons.

Firstly, few surface markers registered the ground plane as the subject of the experimental campaign

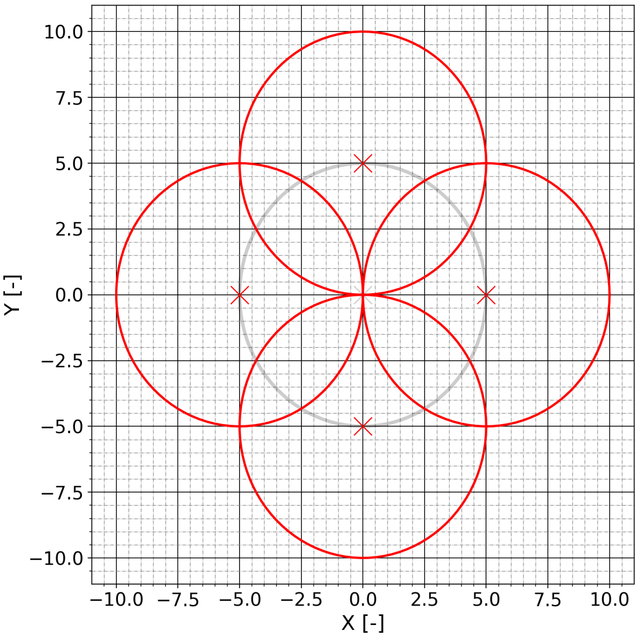


Figure 5.19: Schematic of the inner circle in red on which four outer circles of the same radius are drawn. In each outer circle the lowest tracer particle is sought.

was the object registration of the models, i.e the cube, WBJ and the cyclist. Secondly, the surface-mounted models were mounted to a board through an intermediary adapter plate that was 3D-printed. A cut-out was created in the board, but nonetheless the adapter plate was not perfectly flush with the board. Lastly, the ground plane board itself was made of plywood. In the experimental setup this board was rigidly clamped which slightly deformed it. The results of all experimental cases indicate that the measured ground plane is not aligned with the bottom of the models. This can be seen in figure 5.18, which shows a sample of the particle tracers for $z < 10$ projected onto the $x - z$ plane along with the model contours. The WBJ shows the largest misalignment, which was due to its mounting. To allow a large angle of attack, the WBJ was fixed with a single bolt near the leading edge around which it could be pitched. This slightly tilted the rear up and away from the ground plane board, which is clearly visible in figure 5.18b.

As discussed in subsection 2.3.3, minor misalignments in the wall position can yield significant errors in the near-surface fluid dynamical quantities. To apply the LIVA to both the model and the ground plane, a registration correction for the ground plane is introduced. This correction relies on the tracer data information and attempts to fit a ground plane to the ”underside” of the tracers.

A sample of points is evenly distributed on an initial estimate of the ground plane. In this case the initial estimate is the plane $z = 0$ and points are sampled with a similar inter-nodal spacing as the output-mesh. For each point $P_{GP,i}$ on the estimated ground plane, a search radius r_{search} is defined, which firstly defines an inner circle. On this inner circle $N = 4$ evenly distributed points are sampled around which a search volume is defined using r_{search}. This is shown schematically in figure 5.19. The inner circle is shown in black, whereas the N search circles are displayed in red. The search circles correspond to a cylindrical interrogation volume, a search cylinder. This cylinder extends a set distance above and below the estimated ground plane and registers particles of the complete tracer ensemble which fall inside it. After ordering the tracers inside the volume based on the z-coordinate, the lowest placed tracer is obtained for each search cylinder. The N lowest tracers are averaged to yield the mean lowest tracer in the vicinity of $P_{GP,i}$ and, based on the assumption that the mean particle size r_{HFSB} is $0.35\,\text{mm}$, the local ground position is estimated to then be half of the bubble size lower. This process is repeated for all $P_{GP,i}$ to yield a distribution of local ground plane estimates.

Next, the local ground plane estimates are combined and fitted to a plane, which represents the corrected ground plane. A plane can be fitted using a singular value decomposition (SVD) ([Brown, 1976]), which is one of the standard tools in the scikit-spatial library[3]. First the geometric centroid of the point set is determined by averaging along each spatial dimension. Then, the point set is centred around the geometric centroid. The normal to the plane is obtained by solving a least squares problem using singular value decomposition applied to the centred point set cast into an $N_{GP} \times 3$ matrix **S** such that

$$\mathbf{S} = \mathbf{U}\Sigma\mathbf{V} \tag{5.27}$$

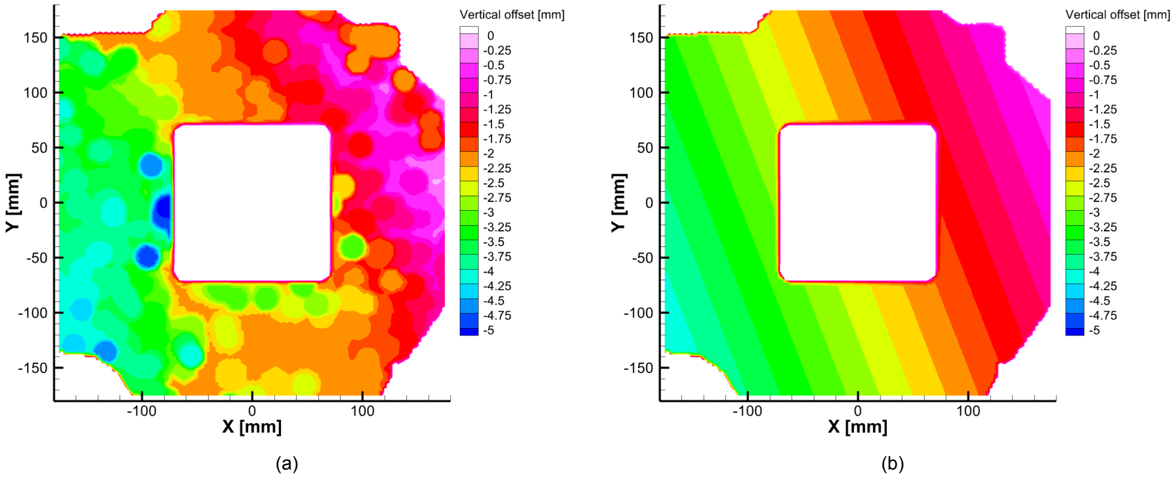
and the last column of the matrix $\mathbf{V}$ then yields the vector normal to the fitted plane.

Results of this approach are shown for the cube case in figure 5.20. Note that the correction is only computed away from the model for points that lie a distance $2r_{\text{search}}$ away. The local offsets are shown in subfigure 5.20a and indicate a trend of increasing offset towards negative x and negative y. The ground plane fit shown in figure 5.20b shows a similar trend as expected.

It should be noted that the plywood board is assumed to be planar in this correction, whereas in reality the bending of the material may yield non-planar geometries. Additionally, there is no extra ghost particle filter applied outside the one explained in chapter 4. The pre-weighting approach aims to filter out sparsely distributed ghost particles by weighting these with real physical particles. The ground plane correction is only applied to the surface output-mesh nodes which coincide with the ground plane and do not lie in the vicinity of the model. For convenience, all results are ultimately stored and displayed on the original surface output-mesh.

---

[3] URL: `https://scikit-spatial.readthedocs.io/en/stable/`



(a) (b)

Figure 5.20: Results of the proposed object registration correction for the ground plane, where (a) shows the local correction based on the search for lowest particles with a single inner circle and four outer circles. The results after the plane fitting are shown in (b).

6

Assessment of velocity reconstruction techniques

A family of five reconstruction methods has been proposed in chapter 5, along with the method by Jux et al. [2018] which was extended to extrapolate the shear stress results to the object's surface. Next, the methods are assessed and the most suitable method is eventually picked to determine the skin friction topology on all experimental models of chapter 4. Jux' method is quickly shown to be infeasible for reconstructing the skin friction, both for topology and magnitude distribution. Therefore, a benchmark solution is introduced first in section 6.1. The benchmark solution employs anisotropic, model-conformal bins which are stretched in the boundary layer. The tangential velocity is determined from a linearised mean Which depends only on the wall-normal coordinate. Then, the LIVA-console is shortly introduced in section 6.2, which addresses research question **IIIb** focused on the implementation of the proposed method into the current state-of-the-art. The console is similar to software like DaVis and is used to process, visualise and check the LIVA-reconstruction techniques, along with Jux' method and the anisotropic benchmark solution. The techniques are finally assessed in section 6.3 using the surface-mounted cube case. Firstly, several velocity profiles are shown, which demonstrate the application of the LIVA-methods for reconstructing boundary layer velocity profiles. Lastly, the skin friction distribution, topology and magnitude, is determined using each method and discussed. The best method is picked and used in chapter 7 to evaluate the skin friction topology over all experimental models, matching them with the off-the-surface flow field and results in literature.

6.1. Formulating a benchmark solution

To quantitatively evaluate the performance of each method, they would need to be compared against a benchmark which acts as a ground truth. This benchmark needs to be a more accurate representation of the flow. There are various ways to achieve a benchmark result against which a methodology's performance can be quantified. For example in fluid simulations, these benchmarks could be achieved by increasing computational resolution, employing industry-proven methods or turning to Direct Numerical Simulations. For experimental simulations, verification can similarly be done by utilising other fluid measurement techniques, such as SOFV or near-wall 2D PIV for the cases here. The specific choice is dependent on the use case and available resources.

The current state-of-the-art available in literature on reconstructing the near-surface velocity on generic objects from particle tracking results is Jux' method. As explained in section 5.1, Jux' method is on average valid up to half of the grid size (depending on the size of Ω_{LPT}), leaving a slight gap to the wall. This state-of-the-art solution can thus not be considered a more accurate representation for evaluation the LIVA reconstruction models.

Therefore, a different approach is taken to obtain a ground truth for benchmarking the family of

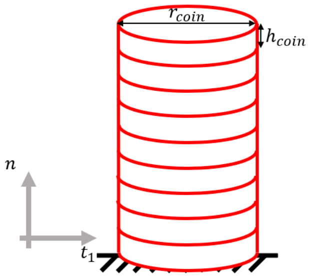
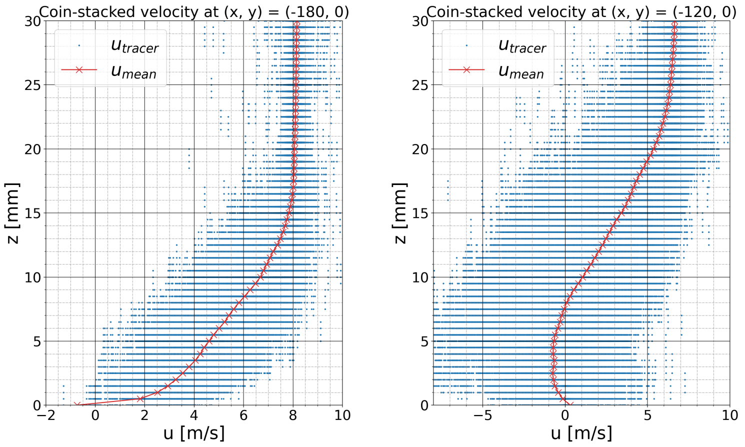


Figure 6.1: Schematic of a single stack of coin-shaped bins.



(a) (b)

Figure 6.2: Velocity profile of a binning with coin-shaped volumes stacked in the wall-normal direction. The velocity profile at (a) $(x, y) = (-180, 0)$ [mm] is shown next to the profile at (b) $(x, y) = (-120, 0)$ [mm]. The blue dot markers indicate the instantaneous velocity of each binned particle placed at the centre of the respective bin. The red cross markers depict are the result of the *Coin-Based-Tracer-LinReg* method.

methods in this thesis report. This approach assumes that the velocity variation close to a wall is dominated by the wall-normal coordinate and varies little in wall-tangential directions, which is a common assumption for the creation of finite element cells in boundary layers ([Tu et al., 2018]). Measured tracers are collected in flat, coin-like bins such that the particles with highly correlated velocities are grouped to determine the mean velocity inside a single coin. Moreover, by increasing the radius and decreasing the height, a relatively high wall-normal resolution can be achieved. A sketch of such a coin stack is shown in figure 6.1. This approach is referred to as the *Coin-Based-Tracer-LinReg* method.

Two velocity profiles with such coin stacks are plotted in figure 6.2 at location I and II. In these results, the coin stacks had a radius of $r_{\text{coin}} = 12\,\mathrm{mm}$. a height $h_{\text{coin}} = 2\,\mathrm{mm}$ and an overlap of 75%. The first coin was centered at the ground plane after the object registration correction of section 5.4 was applied. The mean velocity inside each bin is determined using a linearised mean. The mean velocity is indicated in the plots by the red cross markers and the velocities of the binned tracers are indicated by the blue dot markers. Note that the particles which lie in the same bin are plotted at the $z-$centre of the corresponding bin, thereby visually showing the fluctuations of velocities inside a single bin. The profile in figure 6.1a is taken before the point of separation at $(x, y) = (-180, 0)$ [mm]. The result is typical for a developed boundary layer and it is clear from the presence of the fluctuations that the flow is turbulent. Figure 6.1b depicts the velocity profile after the point of separation at $(x, y) = (-120, 0)$ [mm] with reverse flow and large fluctuations in the separated region.

The coin-stacked velocity plots further highlight the effectiveness of the object registration correction. Note that at the velocity profiles I and II, the ground plane is corrected by respectively -3.85 and $-3.35\,\mathrm{mm}$. The use of a linearised mean with a slightly thicker coin –a height of $2\,\mathrm{mm}$ entails approximately 33 viscous counts ($y^+ = 1 \sim y = 0.06\,\mathrm{mm}$), going up to the logarithmic region– offers a slight flexibility in the wall position estimate. A linearised mean automatically accounts for a linear variation in velocity when it only registers particles in the upper half of the coin, unlike a regular mean. A case could be made that a next iteration of an object registration correction may be performed by finding the $z-$coordinate at which the coin-stacked velocity profile goes through $u = 0\,\mathrm{m\,s^{-1}}$. This is, however, considered outside the scope of this thesis.

Next, the limits to the validity of the coin-stack binning approach are examined. A range of velocity profiles is plotted along the $y = 0$-plane. The boundary layer evolution on the ground plane in front of the cube from $x- = -180\,\mathrm{mm}$ to $x = -72\,\mathrm{mm}$ is displayed in figure 6.3. The x-location of each profile is denoted by the black, dotted line and the velocity is plotted with respect to the corresponding profile line. The evolution shows how the adverse pressure gradient, inflects the velocity profile, until the point of separation which occurs at approximately $x = -140\,\mathrm{mm}$. The velocity profile at $x = -100\,\mathrm{mm}$ has

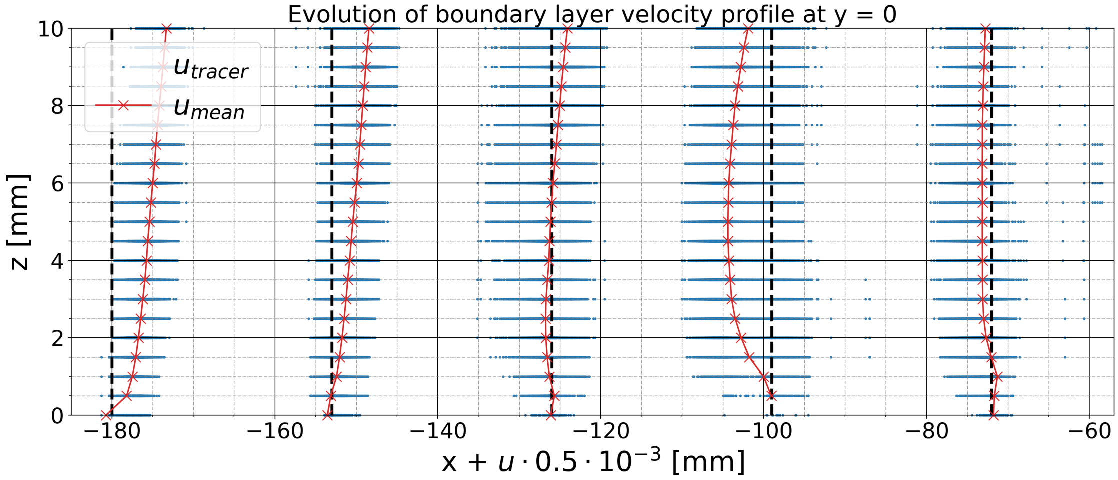


Figure 6.3: Evolution of boundary layer velocity profile obtained from the coin-stack binning, ranging from $x = -180$ to $-72\,\mathrm{mm}$.

an insufficient number of particles. The standard error to the mean velocity σ_U can be determined via ([Agüera et al., 2016])

$$\sigma_U = \frac{\sqrt{u' + \sigma}}{\sqrt{N_p}}, \tag{6.1}$$

where u' is the standard deviation of the fluctuations and σ is the random error due to erroneous particle positioning. This relation is dominated by the standard error of the fluctuations. Hence, assuming a turbulence intensity of 10%, which is typical for boundary layers ([White, 2006]), at least $N_p = 100$ tracers are needed to statistically reduce the error on the mean to <1%. Hence, bins with less than 100 particles are rejected. This condition typically occurs after separated flow, creating strong shear layers which are infrequently penetrated by the tracers. After the flow has separated due to the forward horse shoe vortex it reattaches behind at approximately $x = -80\,\mathrm{mm}$ (see figure 6.14), where more tracers are found again.

Next, consider the boundary layer evolution profile at the top surface of the cube along the plane $y = 0$, displayed in figure 6.4. The front and back face of the cube are located at $x = -60\,\mathrm{mm}$ and $x = 60\,\mathrm{mm}$. The first boundary layer profile at $x = -60\,\mathrm{mm}$ exhibits an unexpected profile. The sharp edge of the cube will induce leading edge separation and thus the flow is expected to move in $+x$ direction directly above this edge. This is confirmed by the streamline-contour plot of figure 6.14. The

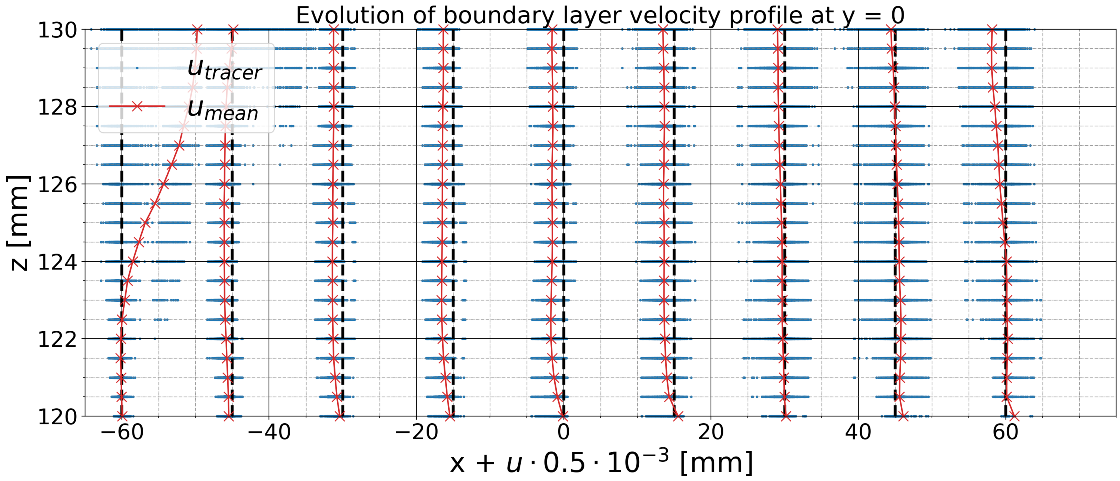


Figure 6.4: Evolution of boundary layer velocity profile obtained from the coin-stack binning, ranging from $x = -60$ to $60\,\mathrm{mm}$.

boundary layer profile at $x = -60\,\mathrm{mm}$ does not match this behaviour. This is due to the relatively large radius of the coins. s. The core assumption of the anisotropically shaped coins is that velocities along the radial direction of the coin are highly correlated and variations can be neglected with respect to the variation along the normal disk axis. In other words, all velocities can be reduced to a single dependency, namely the wall-normal direction. This assumption holds up well for attached turbulent boundary layers, however, in the vicinity of any shear layer this assumption is no longer valid.

6.2. Processing results with LIVA-console

Part of the research goal presented in section 3.4 focuses on the implementation and use of the reconstruction method, namely question **IIIb**. To this end, a console was developed and coded in Python which allows a user to perform three tasks:

- Easily define the inputs and settings for the data processing, such as the fluid measurements, CAD-file with the object information, output-mesh file and sphere radius.
- Check the results for a single cell in the output-mesh, which can be selected by the user. The results are displayed in a built-in viewer.
- To process the results for the whole output-mesh after having selected the desired settings and input files.

This section shortly discusses the main aspects of the console, whereas the reader is referred to appendix B for a link to the Github page which contains the source code and a tutorial which explains how to use the console to process the case of the surface-mounted cube.

An overview of the console is given in subsection 6.2.1. The main functionalities are explained and put in relation to the flowchart of figure 5.13. The user-configurable settings are explained in subsection 6.2.2 and, lastly, the viewers, which are present within the console and used for inspecting the LIVA-method, are clarified and demonstrated in subsection 6.2.3. A big aspect of the LIVA-console is working with the various data files and file formats. For more information on this topic, the reader is directed to appendix C.

6.2.1. General outline of the console

The LIVA-console can generally be divided into three functionalities, which are visualised in figure 6.5 and referred to as the 'Setup', 'Processing' and 'Inspection'. However, before initialising the console, several preparations need to be completed first. These are indicated by the grey box and refer to additional tasks which need to be done after the experiment as outlined in chapter 4. In this subsection, the steps of the flowchart are expanded upon.

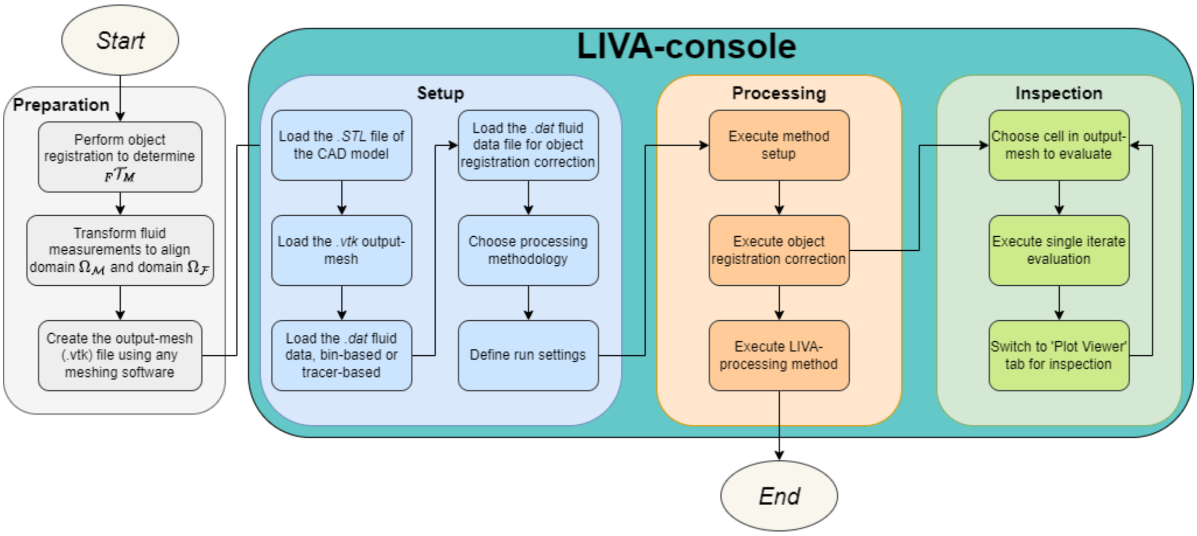


Figure 6.5: Functional flowchart of the LIVA-console.

Preparation

Firstly, an object registration is to be performed to obtain the transformation matrix ${}_{\mathcal{F}}\mathcal{T}_{\mathcal{M}}$. In this report, the method and results by Hendriksen [2024] are used directly without modification. The transformation matrix is subsequently used to transform the fluid measurements in DaVis. This aligns the two meshes and enables the use of the LIVA-methods. The fluid measurements can be outputted in a *.DAT* file as bin-based data or as tracer-based, depending on the method to be performed. A *.STL* file which contains the object-mesh should also be available at this point. This *.STL* file could then be used to generate an output-mesh, for example using the Gmsh software [Geuzaine and Remacle, 2009]. The output-mesh must be then be written to a *.VTK* file to be able to load it into the LIVA-console.

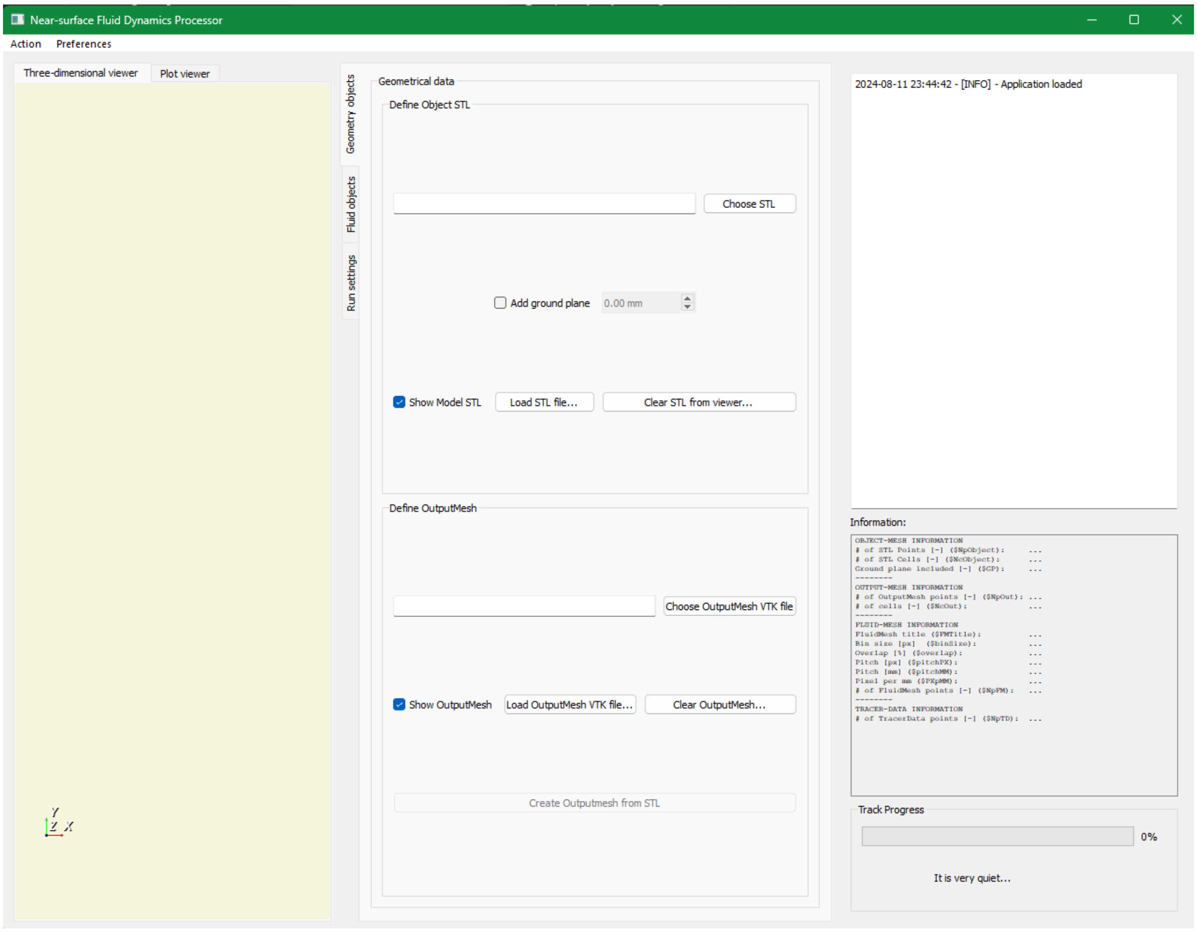
Setup

After the preparation is completed, the LIVA-console can be opened. The first steps are to load in the three data files generated during the preparation phase:

- *.STL* file, which contains the object-mesh information.
- *.VTK* file, which contains the output-mesh information.
- *.DAT* file, which contains the fluid-mesh information.
- (Optionally) *.DAT* file, which contains the tracer positions for the object registration correction as explained in section 5.4.

The tab in the console where the geometrical objects are selected and loaded is shown in figure 6.6. The *.STL* file can be loaded with an initial ground plane, which is always placed orthogonal to the z−axis. The other functionalities are explained in appendix B.

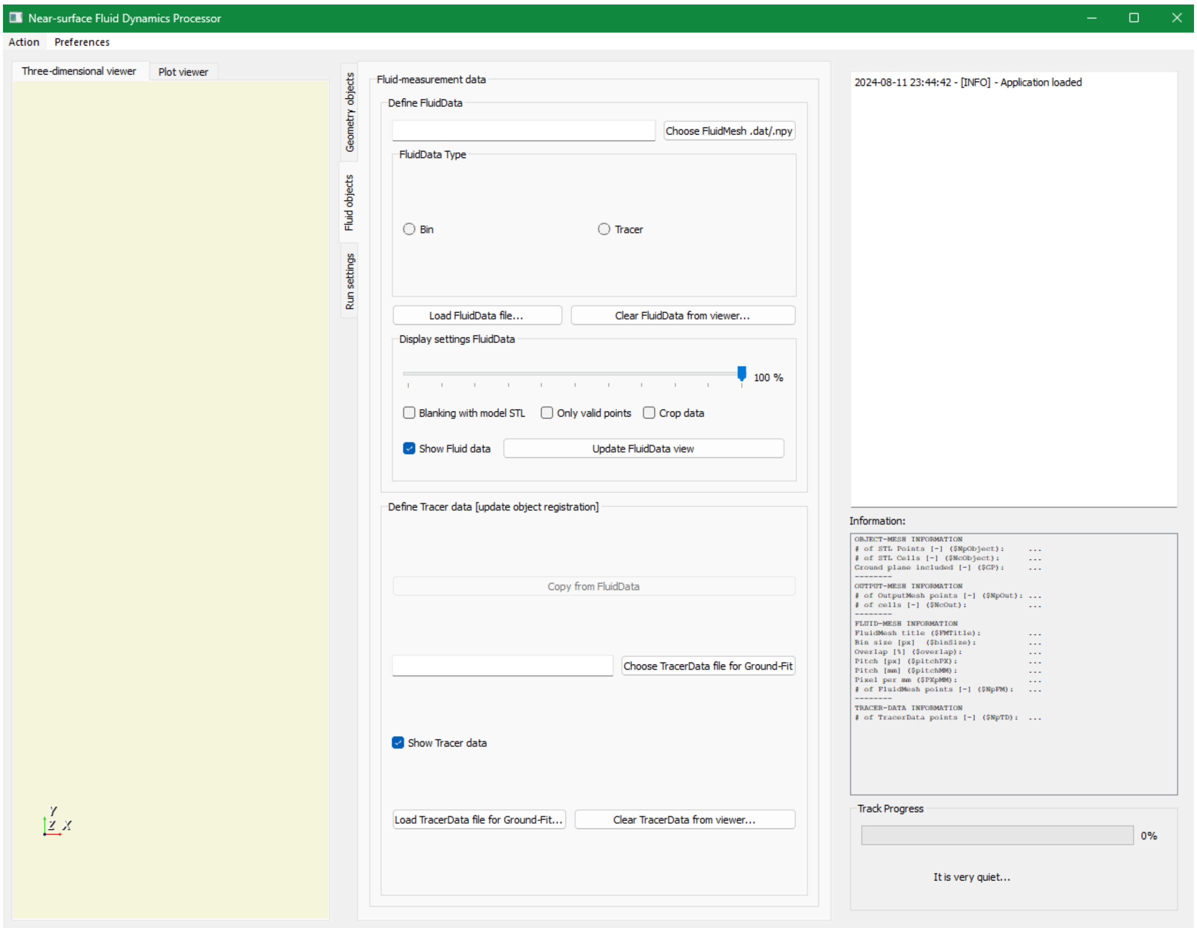
The tab where the fluid objects are selected and loaded is shown in the screenshot of figure 6.7. Note that the user must select if the fluid-mesh represents bin-based or tracer-based data, though the



```
2024-08-11 23:44:42 - [INFO] - Application loaded
```

```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):     ...
# of STL Cells [-] ($NcObject):      ...
Ground plane included [-] ($GP):     ...
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): ...
# of cells [-] ($NcOut):             ...
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):          ...
Bin size [px]  ($binSize):           ...
Overlap [%] ($overlap):              ...
Pitch [px] ($pitchPX):               ...
Pitch [mm] ($pitchMM):               ...
Pixel per mm ($PXpMM):               ...
# of FluidMesh points [-] ($NpFM):   ...
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  ...
```

Figure 6.6: Screenshot of the LIVA-console where the Geometry objects can be defined.



```
2024-08-11 23:44:42 - [INFO] - Application loaded
```

```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):     ...
# of STL Cells [-] ($NcObject):      ...
Ground plane included [-] ($GP):     ...
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): ...
# of cells [-] ($NcOut):             ...
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):          ...
Bin size [px]  ($binSize):           ...
Overlap [%] ($overlap):              ...
Pitch [px] ($pitchPX):               ...
Pitch [mm] ($pitchMM):               ...
Pixel per mm ($PXpMM):               ...
# of FluidMesh points [-] ($NpFM):   ...
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  ...
```

Figure 6.7: Screenshot of the LIVA-console where the Fluid objects can be defined.

software guesses this when selecting the file. Additionally, the fluid-mesh typically contains several million points. These visuals can be filtered with the tools available in the section 'Display settings FluidData'. The file which contains the tracer positions for the object registration correction is also loaded via this view. If the user is working with a tracer-based fluid mesh, this information can be copied from the fluid-mesh. However, it is also possible to select a different data file.

Next, the user chooses one of the available reconstruction approaches, which depends on the available data. The console-view where this is selected, is displayed in figure 6.9. The three algorithms in the left column require a bin-based fluid-mesh, while the algorithms in the right column require a tracer-based fluid-mesh. The user further selects the interrogation sphere radius and chooses cropping dimensions for both the 'Ground Estimate', for the object registration correction, and the 'Fluid Mesh', for processing the results onto the output-mesh. Note that this can reduce the computational cost, though care must be taken that the actual ground plane and output-mesh lies inside the respective bounds. Lastly, the user may also use a semi-sphere or a semi-sphere where the mid-plane has prescribed a radius of curvature. This is useful when binning data near thin objects, such as the cyclist's wheels. In such cases, a complete sphere would also bin fluid-mesh and object-mesh points which lie on the other side of model, as for example shown in figure 6.8. After the settings have been initialised, the setup can be started. After the setup is completed, the user can continue to the next step.

Processing

The next step is the processing of the results, during which the user will typically look at individual results, before commencing the complete run. The first step in the processing is to optionally execute an object registration correction. Such a correction is done for all the cases described in this report, but may not be necessary, either when the accuracy ground plane position estimate is of sufficient accuracy or when there is no ground plane to be considered. After the ground plane correction, the user may

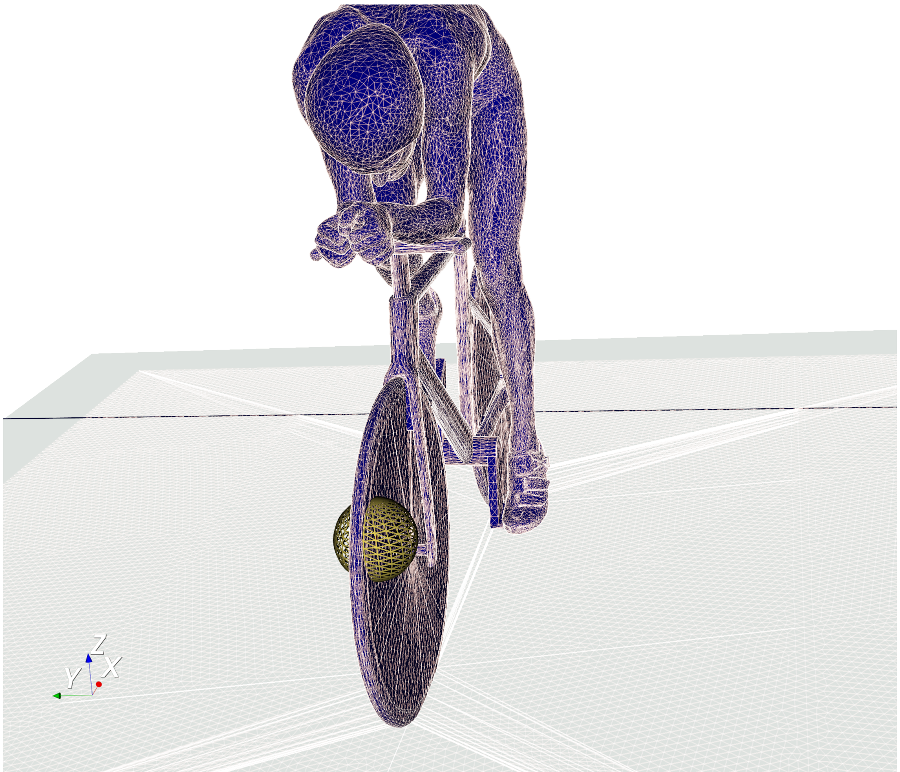


Figure 6.8: Example of spherical interrogation volume cutting through the model such that points on the other side are also binned.

either process individual results by first choosing a cell of the output-mesh or they may execute the process for the entire output-mesh. In the latter case, the results are written to the selected directory as a *.VTU* file. This file can then be loaded into for example TecPlot 360 [TECPLOT] or ParaView [Ahrens et al., 2005] to visualise the results. At that point, the user no longer requires the LIVA-console.

Inspection
When the user chooses to process a single cell, the results of the velocity reconstruction are displayed in the console itself for inspection. Firstly, a cell of the output-mesh is chosen, either selecting the cell number or by selecting one via the built-in viewer. The exact definition of the results in the built-in viewer are discussed in subsection 6.2.3, but these are split into two:

- Results of the binning with (semi-)spherical interrogration volumes around the object.
- Results of the velocity reconstruction via either linear interpolatior or fitting quadratic functionals.

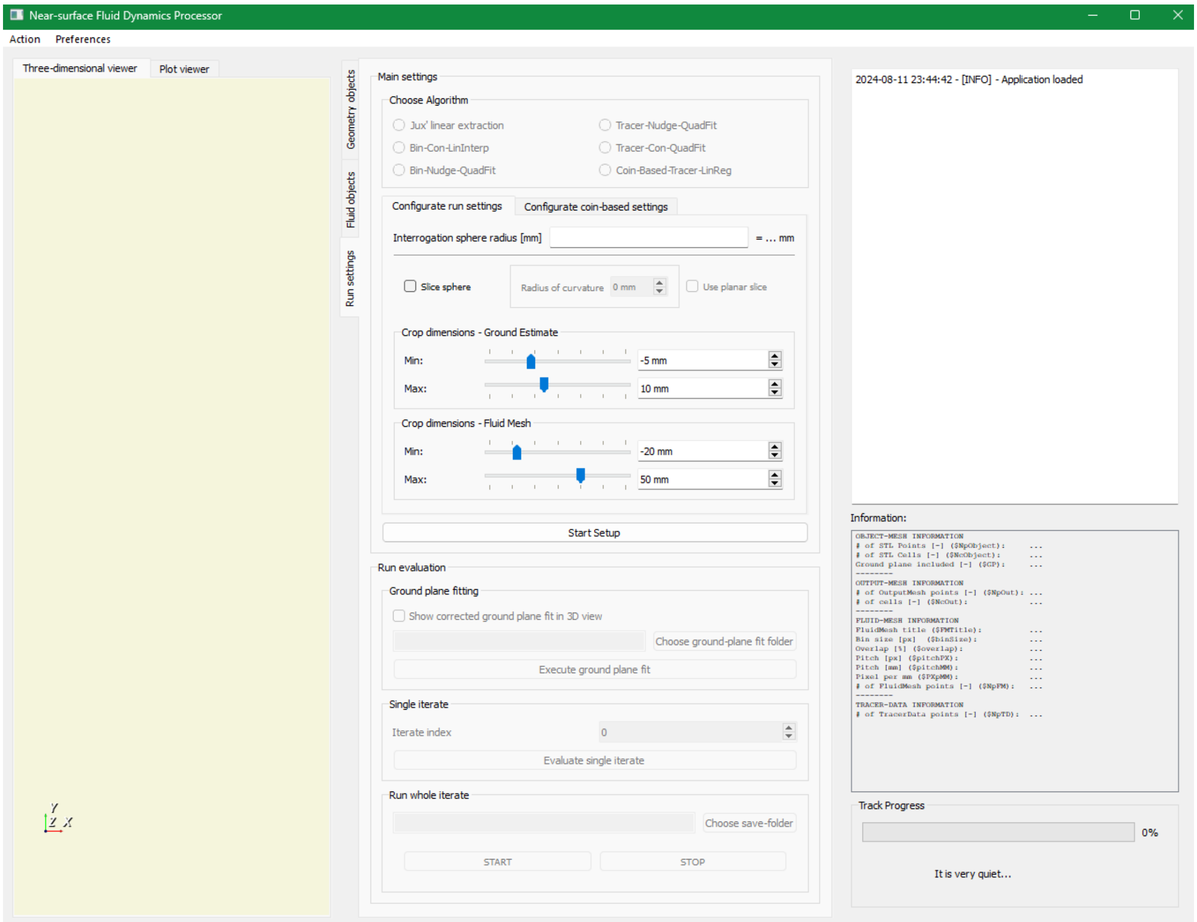
New points can be selected and processed serially. Once the user is finished with the inspection, they can execute the process for the entire output-mesh.

6.2.2. Configurable settings

A concise overview of all user-configurable settings is given in this subsection. These are divided into three categories, namely *Geometrical data settings*, *Fluid data settings* and *Algorithm settings*:

Geometrical data settings

- File with object-mesh information; a *.STL* file which contains the CAD model of the object can be selected.
- File with output-mesh information; a *.VTK* file which contains the output-mesh geometry can be selected.
- Add ground plane; a checkbox in the console allows the user to automatically add a ground plane upon loading the *.STL* file. When done so, the ground plane will be included when executing one of the (LIVA-)methods.



Radius of curvature: 0 mm

Crop dimensions - Ground Estimate
Min: -5 mm
Max: 10 mm

Crop dimensions - Fluid Mesh
Min: -20 mm
Max: 50 mm

Iterate index: 0

```
2024-08-11 23:44:42 - [INFO] - Application loaded
```

```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):      ...
# of STL Cells [-] ($NcObject):       ...
Ground plane included [-] ($GP):      ...
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): ...
# of cells [-] ($NcOut):              ...
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):           ...
Bin size [px]  ($binSize):            ...
Overlap [%] ($overlap):               ...
Pitch [px] ($pitchPX):                ...
Pitch [mm] ($pitchMM):                ...
Pixel per mm ($PXpMM):                ...
# of FluidMesh points [-] ($NpFM):    ...
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  ...
```

Figure 6.9: Screenshot of the LIVA-console where the Run settings can be defined.

Fluid data settings

- File with fluid-mesh information; a *.DAT* file which contains either bin-based or tracer-based fluid velocity data can be selected.
- File with tracer position information; a *.DAT* file which contains tracer positions can be selected. This must be the same format as the tracer-based fluid velocity data and, therefore, the same file

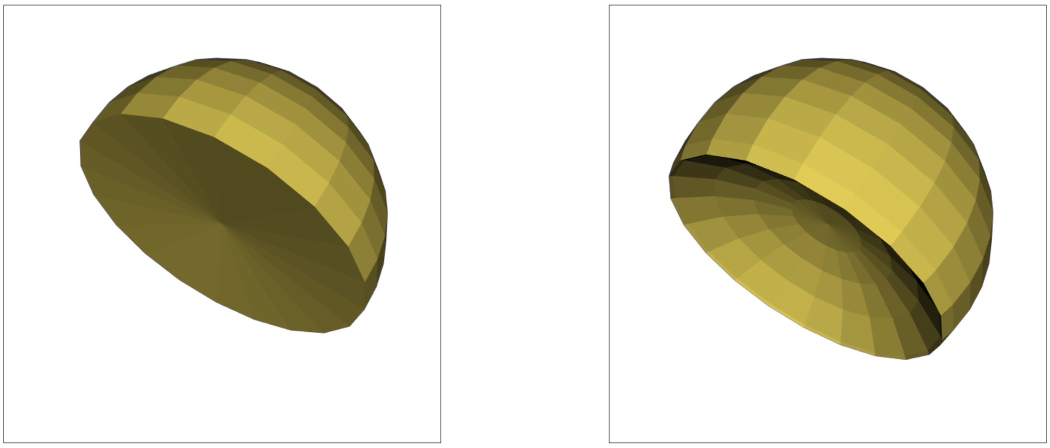
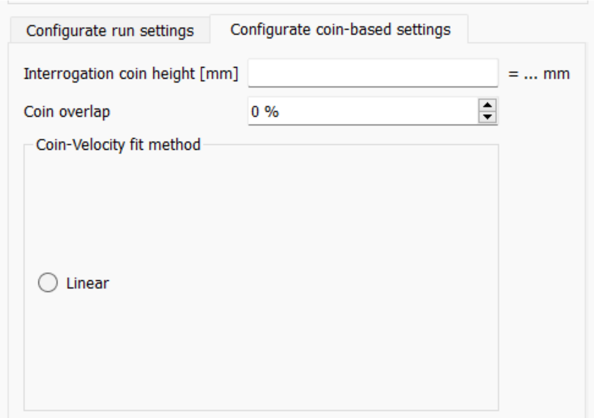

(a) (b)

Figure 6.10: Example of a semi-sphere where the sliced surface (a) is planar and (b) has a radius of curvature.



Interrogation coin height [mm] = ... mm

Coin overlap 0 %

Coin-Velocity fit method

○ Linear

Figure 6.11: User-configurable settings in the LIVA-console for when the algorithm *Coin-Based-Tracer-LinReg* is chosen.

as for the fluid-mesh can be reused.

- Display settings for fluid data; For displaying the fluid-mesh and tracer positions in the three-dimensional model viewer, several settings can be used to improve the display. Firstly, the percentage of points that are shown to the user can be decreased. The points can be masked using the model STL, which does not show points which fall inside the model or under the ground. If an IsValid flag is included with the *.DAT* file, this attribute can be used to only display valid points. If present, this flag is always used in the processing. Then, the data can be cropped using the 'Crop Dimensions' which are part of the *Algorithm settings*.

Algorithm settings

- Processing algorithm; One of the six processing algorithms can be chosen, either one which is one of the four proposed LIVA-methods (*Bin-Con-LinInterp*, *Bin-Nudge-QuadFit*, *Tracer-Nudge-QuadFit*, *Tracer-Con-QuadFit*, section 5.3), Jux' method using additional linear extrapolation (*Jux' linear extraction* subsection 5.1.2 & 5.1.3) or the anisotropic coin stacking approach (*Coin-Based-Tracer-LinReg*, section 6.1).
- Sphere radius; The user defines the radius of the spherical interrogation volume here. This may also be an expression using one of the variables displayed in the information table below the logging widget, e.g. "*3 * $pitchPX*" defines a radius that is thrice the pitch (in mm).
- Sphere splitting; The interrogation volume can also be split to form a semi-sphere. The user has two options, either to use a planar slice (true semi-sphere) or semi-sphere where the slice surface has a radius of curvature. These semi-spheres are visualised in figure 6.10.
- Coin-based settings; When the method *Coin-Based-Tracer-LinReg* is chosen, the user must additionally define settings for the coin-stack. These are shown separately in figure 6.11. The height of the coin-stack is defined, along with the overlap of the coins and the fitting algorithm. In this report, only the linearised mean is used, hence this is the only available option. The radius of the coin is taken from the *Sphere radius*-setting.
- Crop dimensions; The fluid-mesh and tracer position data can be cropped to improve the computation cost. A lower and upper bound can be selected, though care must be taken to ensure that the cropping does not negatively affect the ground plane fit or main processing algorithm. The actual ground plane must lie inside the 'Ground Estimate' cropping dimensions and the output-mesh must lie within the 'Fluid Mesh' cropping dimensions.
- Show corrected ground plane fit; When executing the ground plane fit, it is possible to show the corrected plane in the three-dimensional viewer.
- Iterate index; Before processing a single iterate, the user can select an output-mesh cell of choice.
- Save folder; After executing the algorithm for the whole output-mesh the results are saved to a *.VTU* file in the selected directory.

Figure 6.12: View of the 3D model viewer, when the object-mesh, fluid-mesh and output-mesh has been loaded. The object-mesh is shown in blue, the fluid-mesh in purple and the output-mesh in orange.

6.2.3. Built-in viewers

Lastly, the console includes two viewers and a logging widget, one of the viewers displays the geometrical model, the other the binning and velocity reconstruction results for a single interrogation volume and the logging widget communicates log information to the user. Below the logging widget is a small table which shows information about the loaded data. The viewers are explained in more detail in this section.

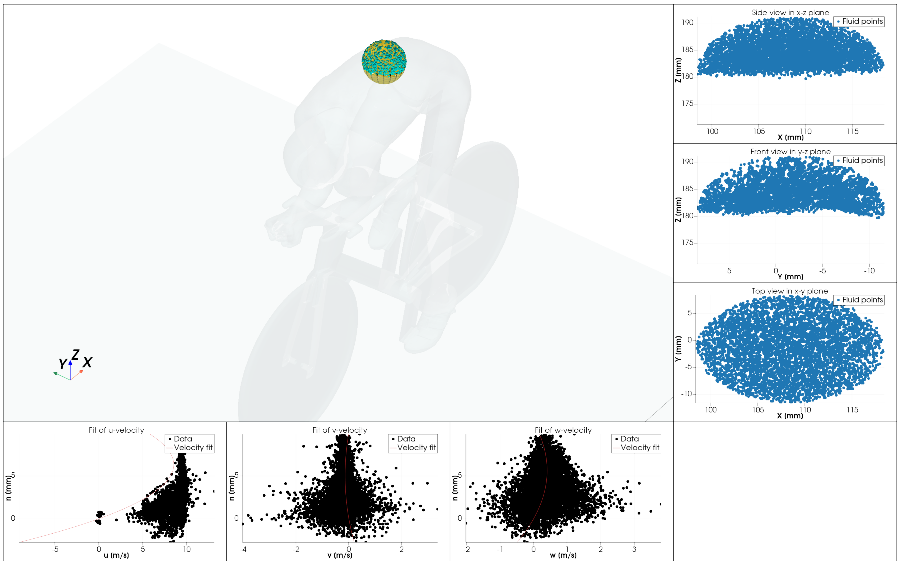


Figure 6.13: Example of the plot viewer for inspecting the results of the LIVA reconstruction.

Three-dimensional model viewer

The *3D model viewer* shows the meshes upon loading the files. The user can move the objects around in space and verify that the correct meshes have been chosen. When the setup is completed and the user process a single cell of the output-mesh, the chosen interrogation volume is also displayed in the 3D model viewer. This provides the user with a better feel of for example the size of the interrogation volume relative to the object. An example of the viewer, with all data loaded is shown in figure 6.12, whereas a view with a single interrogation volume is depicted in subfigure 5.11a.

Plot viewer

The *Plot viewer* shows the results of reconstructing a single interrogation volume. Examples of this viewer were shown above in chapter 5. Two views are available to the reader. First, a view similar to the 3D model viewer discussed above is available, which shows the results of the binning. An example is shown in subfigure 6.13. The fluid-mesh points which fall inside the interrogation volume are displayed in blue, with the object-mesh in red. A discretised version of the spherical interrogation volume is shown in yellow. The model and ground plane are still included in the view, though greyed out. Static side views of the binning are also available.

Additionally, the results of the velocity reconstruction are included. For each velocity-component, u, v and w, the fit (linear or quadratic) is plotted against the wall-normal coordinate n. The data points are also included which allows the user to verify the fit. Note that all fluid-mesh points are collapsed to only the wall-normal coordinate. An example of such a fit is shown in figure 6.13.

6.3. Comparison of LIVA-reconstruction schemes

Comparison and verification of the methods is done both qualitatively and quantitatively, through the reconstruction of velocity profiles and by examining the wall-shear stress distribution. For the quantitative evaluation a ”ground truth” solution is introduced, which exploits the anisotropic nature of the boundary layer to increase the wall resolution. This ground truth is introduced and explained in subsection 6.1. This subsection also discusses the validity of this benchmark solution. Then, the proposed family of four methods, Jux’ state-of-the-art method and the benchmark solution are applied to four ve-

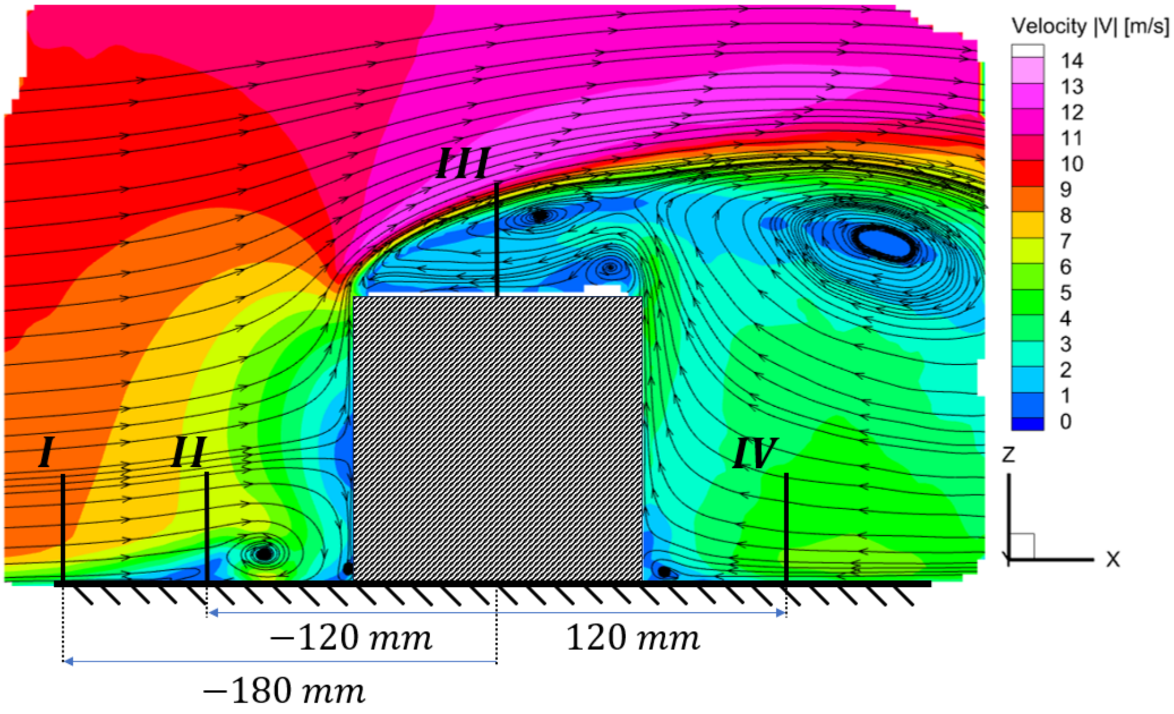


Figure 6.14: Location of velocity profiles at plane $y = 0$ overlaid with the velocity magnitude contour and streamlines in the same plane. The cube has sides of length $L = 120\,\mathrm{mm}$. Note that the position of the cube in this figure is for illustration purposes and not exactly correct

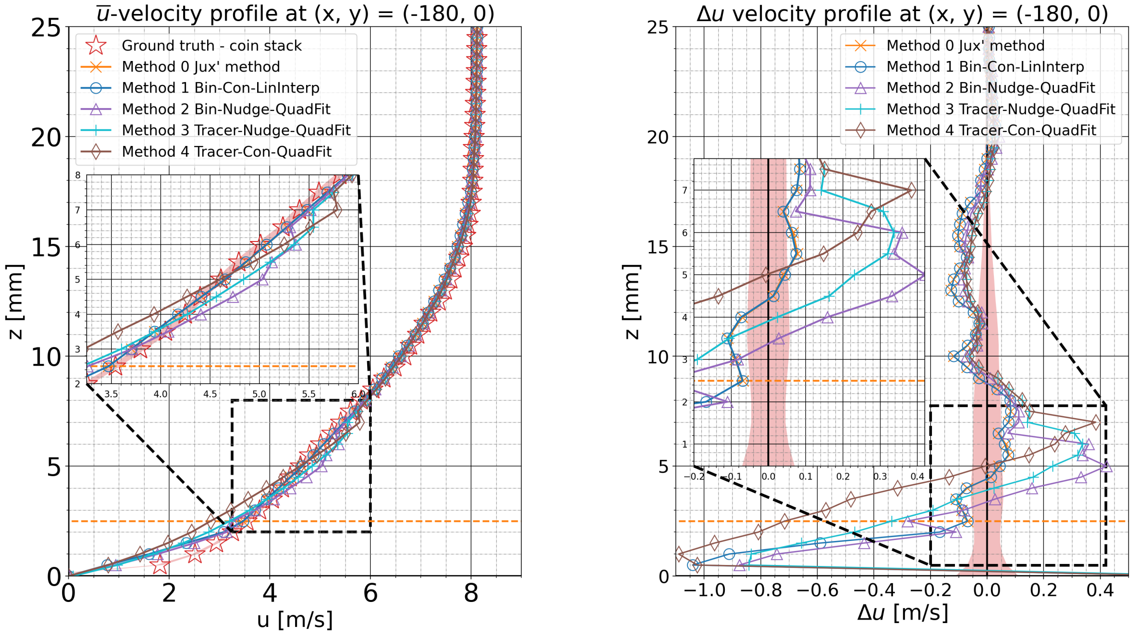
locity profiles around the cube –when valid– and presented in subsection 6.3.1. Lastly, the wall-shear stress distributions are introduced and discussed to both verify the reconstruction methods and further compare them against each other.

6.3.1. Velocity profiles

The methods are first applied to reconstruct a boundary layer profile. Four different positions are chosen on the ground plane and object itself. These positions are shown in figure 6.14. First, velocity profile I is placed far ahead of the cube, such that it falls inside the attached turbulent boundary layer. Profile II is positioned directly after the point of separation and reverse flow is expected to occur near the ground. The upper portion of the boundary layer experiences forward flow. The next velocity profile III is located on top of the cube in the wake of the leading edge separation which mostly contains reversed flow. Lastly, velocity profile IV is chosen which lies in the wake region of the cube. This profile only experiences reverse flow.

The mean velocity profile is shown for location I in figure 6.15a. All methods, except for Jux' method, go to zero near the wall due to the introduction of the no-slip boundary condition in the reconstruction. As Jux' method interpolates the fluid state between binned velocities without imposing boundary conditions, there are no more data points available beyond $z < 2.5\,\mathrm{mm}$. The *Bin-Con-LinInterp* method follows the same approach, but adds linear interpolation between the fluid bins and the wall which bridges the gap. The family of methods all show a smooth velocity profile, apart from the *Bin-Nudge-QuadFit* method. This first overshoots the ground truth and then jaggedly reduces to the ground truth profile as can be seen in the zoomed-in portion. The plot of the differences in subfigure 6.15b reveals maximum differences occur with respect to the benchmark solution of around $1\,\mathrm{m\,s^{-1}}$. The methods with quadratic velocity fit, behave in a similar way. Close to the wall, these under predict the velocity, then slightly after these over predict the velocity. This is typical behaviour when fitting a low-order polynomial that is unable to reach the steep gradients that are expected in boundary layers.

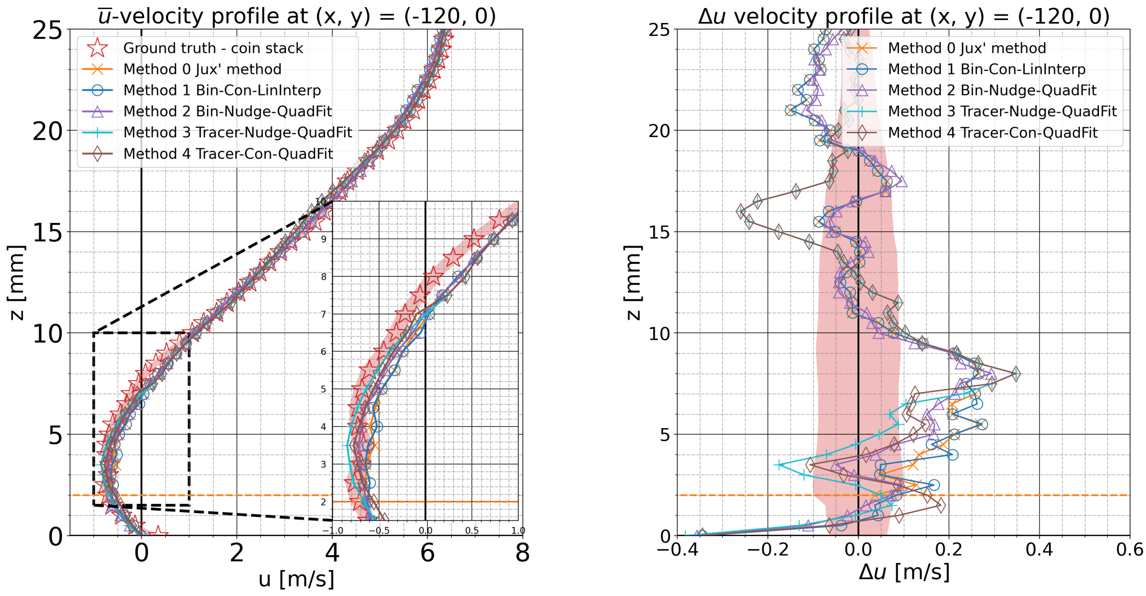
Next, the velocity profile at location II is displayed in figure 6.16. The separated boundary layer flow



(a)

(b)

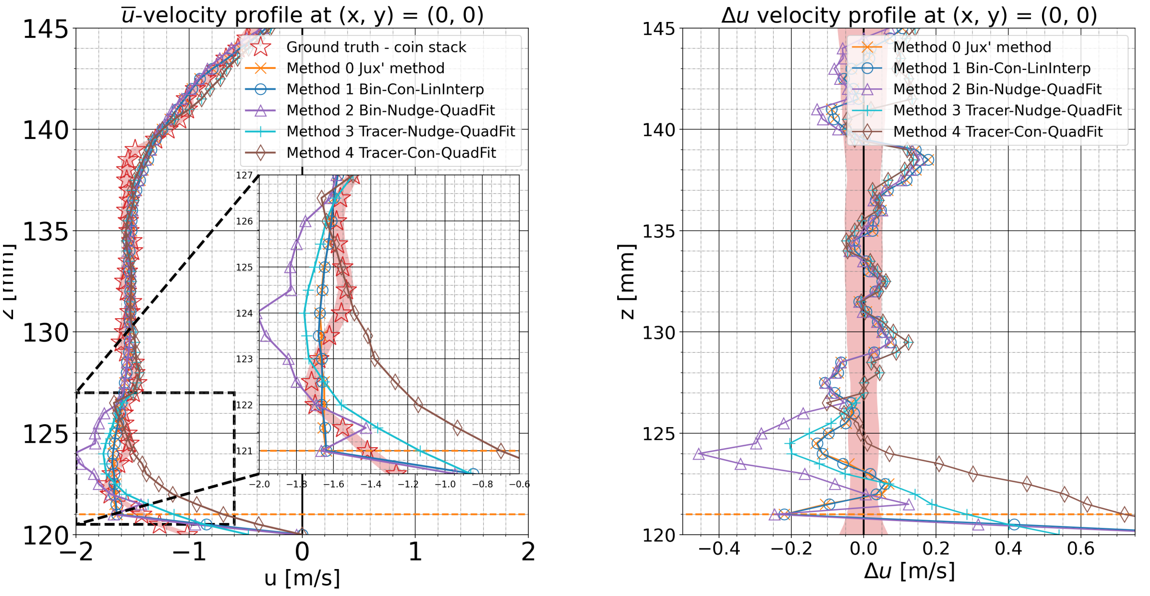
Figure 6.15: Velocity profile (u-component) at location I. In (a) the mean velocity $\overline{u}$ is shown along with each respective reconstruction method ($\tilde{u}$), whereas in (b) the difference Δu is plotted defined as $\overline{u} - u_{GT}$. The red area indicates the uncertainty of the benchmark solution, whereas the orange dashed line indicates the data point closest to the surface, attainable with Jux' method.



(a)

(b)

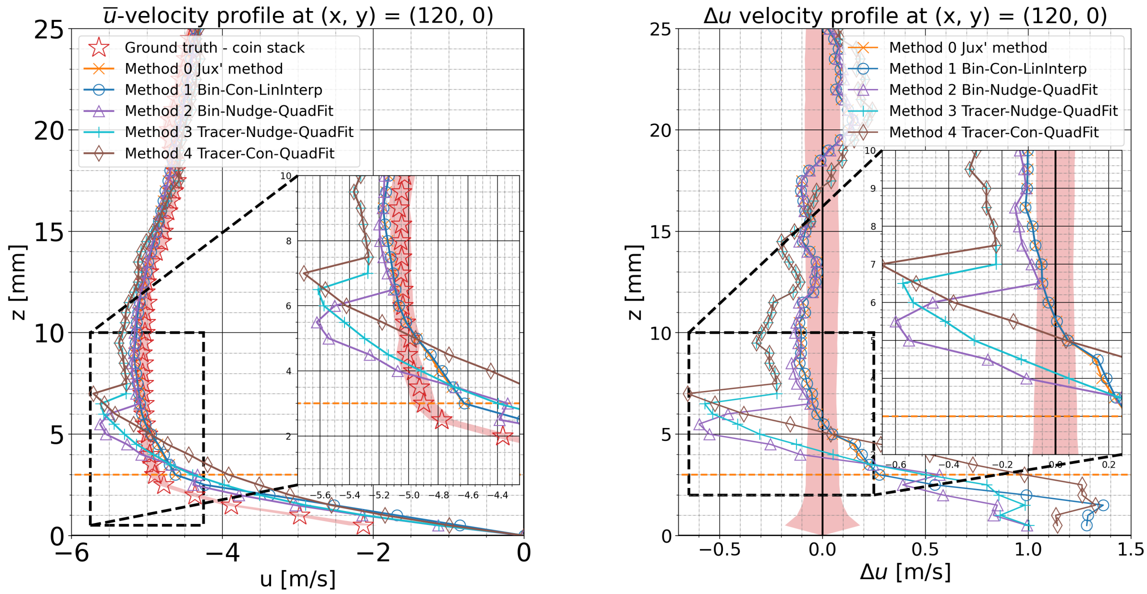
Figure 6.16: Velocity profile (u-component) at location II. In (a) the mean velocity $\overline{u}$ is shown along with each respective reconstruction method ($\tilde{u}$), whereas in (b) the difference Δu is plotted defined as $\overline{u} - u_{GT}$. The red area indicates the uncertainty of the benchmark solution, whereas the orange dashed line indicates the data point closest to the surface, attainable with Jux' method.



(a)

(b)

Figure 6.17: Velocity profile (u-component) at location III. In (a) the mean velocity $\overline{u}$ is shown along with each respective reconstruction method ($\tilde{u}$), whereas in (b) the difference Δu is plotted defined as $\overline{u} - u_{GT}$. The red area indicates the uncertainty of the benchmark solution, whereas the orange dashed line indicates the data point closest to the surface, attainable with Jux' method.



(a)

(b)

Figure 6.18: Velocity profile (u-component) at location IV. In (a) the mean velocity $\overline{u}$ is shown along with each respective reconstruction method ($\tilde{u}$), whereas in (b) the difference Δu is plotted defined as $\overline{u} - u_{GT}$. The red area indicates the uncertainty of the benchmark solution, whereas the orange dashed line indicates the data point closest to the surface, attainable with Jux' method.

shows significantly lower gradients, which highly benefits all methods. Again, all proposed methods go approximately to zero due to the enforcing of the boundary condition at the wall. Around the infliction region, the methods over predict the velocity compared to the ground truth. This may more likely be a fallacy of the benchmark solution, which has limited validity in regions of (high) shear, as explained in section 6.1. The difference, both amongst the methods themselves and compared to the benchmark solution is at most $0.5\,\mathrm{m\ s^{-1}}$.

Velocity profile III on the upper surface of the cube is depicted in figure 6.17. This also presents the first example for which the ”nudged” *Tracer-Nudge-QuadFit* method does not (approximately) go to zero at the surface. The distribution of WG points is not strong enough to pull the solution to zero, likely due to the very thin local boundary layer. This boundary layer is identifiable by the relatively large velocity at only $1\,\mathrm{mm}$ above the surface. The effect of the constrained *Tracer-Con-QuadFit* becomes very clear.

This method forces the solution through zero, however, the introduction of the single constraint reduces the degree of freedom of the quadratic fit. This can similarly be compared to lowering the order of the polynomial. Thus lower gradients are expected. The differences for velocity profile III in figure 6.17b indicate that the reconstruction solutions are within $0.5\,\mathrm{m\ s^{-1}}$ of the benchmark solution, with the *Tracer-Nudge-QuadFit* having the lowest offset.

Lastly, figure 6.18 presents the results for velocity profile IV. This profile is placed in the wake of the cube and experiences purely reversed flow. Similar to the reversed flow on the top of the cube, the local, secondary boundary layer is in the order of a few millimeters. The quadratic fits are not able to fit such a steep gradient and both under and over predict the velocity. This velocity profile shows an artefact of the boundary condition implementation, which was faintly visible in figure 6.15, particularly in the plot of Δu. The overshoot of the solution is caused by the fitting of a quadratic polynomial, which is (weakly) constrained to also pass through the zero velocity at the wall inside the local interrogation volume. The boundary condition is included as long as the wall geometry is intersected by this volume. When the volume no longer intersects the model geometry, no boundary conditions are included and the only information on the presence of the model is indirectly supplied by the fluid measurements. This causes a kink, which only increases with thinner boundary layers. The kink may be overcome by shrinking the size of the interrogation volume, though this complicates the requirement for sufficient particles. However, when the interrogation volume is several orders smaller than the boundary layer

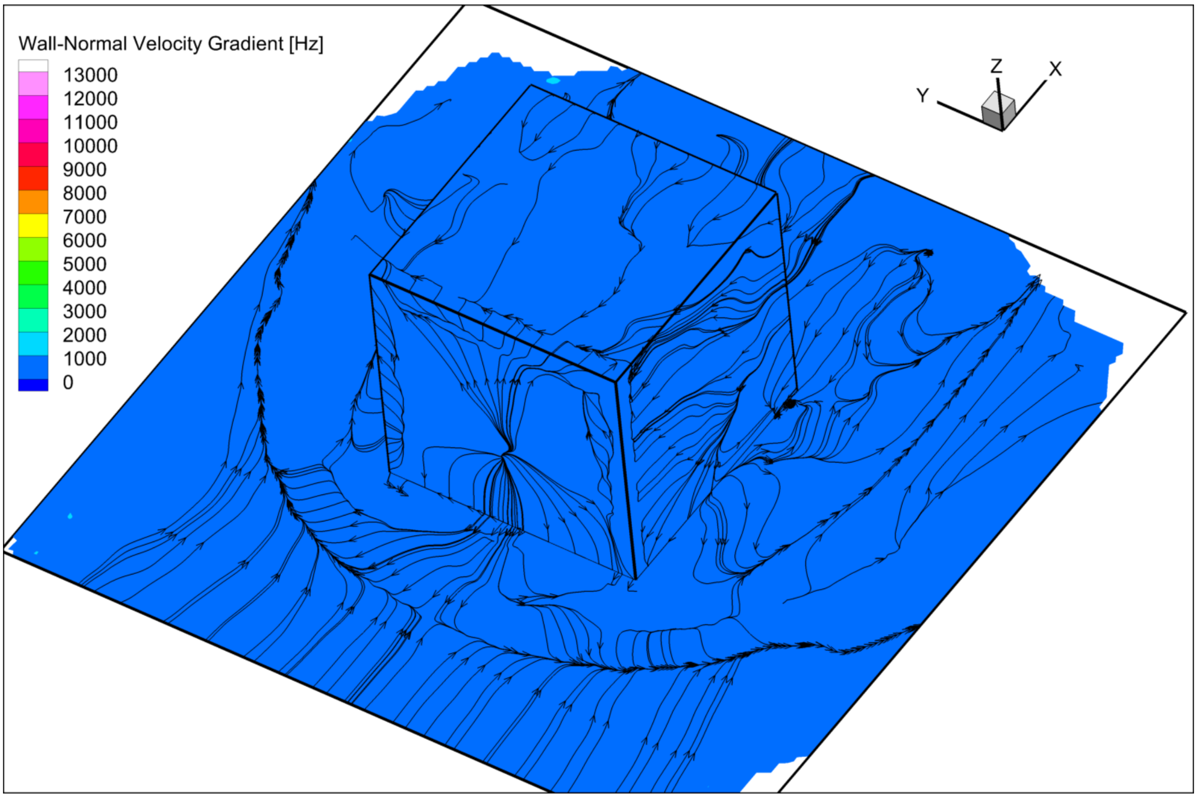


Figure 6.19: Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for ***Jux' method***.

height and improvement to the velocity profile reconstruction is.
Taking the boundary layer of velocity profile I, the interrogation volume radius would need to be at least half the size of the developed boundary layer for a reconstruction without over- and underfitting of the data. Given the characteristic heights of the boundary layer in the experiment, one could determine the needed interrogation radius and subsequently the required ensemble particle density can be estimated. For example, in a flow where the smallest characteristic boundary layer is $10\,\mathrm{mm}$, the interrogation volume must be at most $5\,\mathrm{mm}$ in radius. To then obtain at least 100 particles in the semi-sphere, a ensemble particle concentration of approximately 200 particles per cubic centimeter is needed. Ensuring homogeneous seeding, if the particle density in the shear layer during the experiment would be 0.1 particle per cubic centimeter, at least 2,000 images would be needed, assuming that most particles are reconstructed using STB.

Table 6.1: Table of shear stress τ_w magnitude $[\mathrm{mPa}]$ at different positions around the wall-mounted cube from various methods.

Ground position (x, y, z) *[mm]*	Flow type	Benchmark	Coles' method ((A.5))	Jux' method	*Tracer-Nudge-QuadFit*	*Tracer-Con-QuadFit*
(-170, 0, 0)	Turbulent BL	52.0	73.1	8.3	24.1	24.5
(-150, 0, 0)	Horseshoe vortex	25.4	15.78	7.9	7.4	8.5
(90, -110, 0)	Vortex streak	79.2	N/A	8.6	76.5	72.9
(-60, 0, 90)	Laminar BL	21.1	16.2	8.2	22.3	22.7
(120, 0, 0)	Separation	62.5	N/A	6.5	33.6	34.1

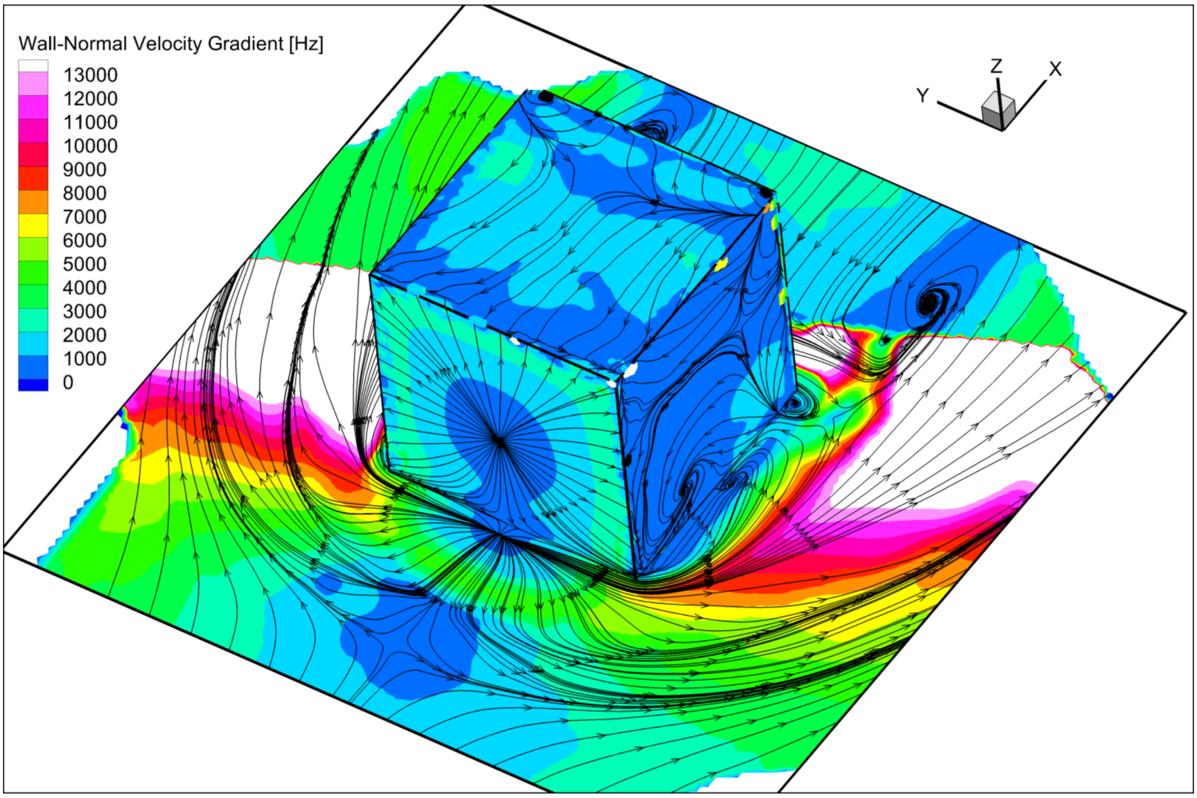


Figure 6.20: Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for the ***Bin-Con-LinInterp*** method.

6.3.2. Wall-shear stress distributions

Lastly, the methods are applied to a surface output-mesh of the surface-mounted cube case. The results are displayed and discussed in this subsection. The wall-shear-stresses are determined via the process of subsection 5.3.5. Note that for consistency, all skin friction magnitudes are displayed with the same colour map range, which cause oversaturation with some plots and undersaturation with others. For the tailored results which focus on the aerodynamic evaluation of the surface-mounted cube case, the reader is referred to chapter 7.

Firstly, Jux' method is displayed in figure 6.19. The skin friction lines are plotted on top of the wall-normal velocity gradient contour. The skin friction lines far forward of the cube are qualitatively okay, however, it is apparent that without including the wall surface no-slip condition both the skin friction topology and magnitude greatly differ from reality. The first proposed method *Bin-Con-LinInterp* is presented in figure 6.20. The small adaptation from Jux' method can be seen to significantly improve both the topology and relative magnitude contours. The forward separation point on the ground plane, stagnation point on the forward face and the streaklines of the horseshoe vortex which curve around the cube are all visible. Further, the footprint of the arch vortex behind the cube is visible along with the corner vortex on the side and the reattachment saddle point on the top surface. From a qualitative perspective, these agree well with results in literature ([Depardon et al., 2005] and [da Silva et al., 2021]). Contrary, the magnitude contour in figure 6.20 exhibits unexpected behaviour. The skin friction significantly increases after the line of separation. Typical skin friction profiles show a decrease in the skin friction magnitude aft of the separation line. However, in this case the effect is artificially induced by the non-conformal fluid-meshbin.

After the object registration correction of section 5.4, the ground plane of the wall-mounted cube is placed at a slight angle with respect to the cube's faces. This aligns perfectly with the diagonal line of maximum wall-shear stress visible in the contour plot. The trilinear approach of method *Bin-Con-LinInterp* directly scales the wall-normal velocity gradient with the distance between a fluid-mesh node and the surface-mesh. When the surface-mesh is not conformal with the fluid-mesh, such an alias-effect occurs. The same alias-effect occurs for the case of method *Bin-Nudge-QuadInterp*. The

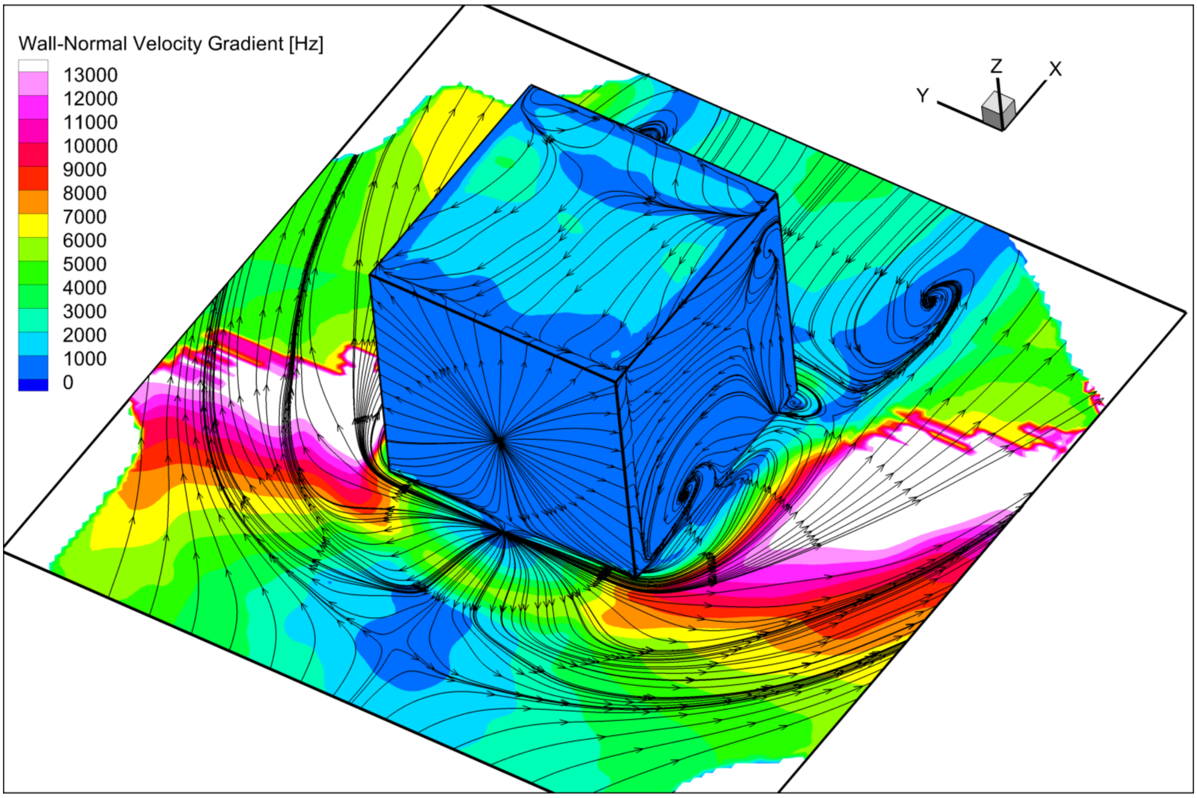


Figure 6.21: Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for the ***Bin-Nudge-QuadFit*** method.

distribution of the bin-based fluid-mesh nodes is very sparse, to the extent that the addition or subtraction of a single additional fluid-mesh node can yield jagged results.

The first tracer-based method *Tracer-Nudge-QuadFit* is displayed in figure 6.22. As expected, the homogeneous ensemble of tracer particles does not experiencing the aliasing problem contrary to the Cartesian bin-based representation. The magnitude of the wall-normal gradient reaches up to $7.000\,\mathrm{Hz}$ in the streaklines of the horseshoe vortex. The results for the method *Tracer-Con-QuadFit* are displayed in figure 6.23. These are nearly indistinguishable from the results of the nudged method. It should be noted, however, that the results for the *Tracer-Nudge-QuadFit* method are highly dependent on the number of WG points and thereby the concentration fixed before the processing. This approach is also computationally more expensive due to the Poisson sampling step on an intersection of the geometry. The *Tracer-Nudge-QuadFit* method only requires a single projection onto the surface-mesh to include the boundary constraint.

The results are sampled at different locations around the cube to indicate the wall-normal shear magnitude for all methods. This is displayed in table 6.1. Three locations are sampled inside the attached turbulent boundary layer, close to the point of separation and in the wake of the horseshoe vortex. In all cases, Jux' method shows little variation between the locations. The LIVA-methods proposed in this report, consistently underpredict τ_w in attached turbulent boundary layer. However, outside the classic boundary layer profile, the LIVA methods are more applicable and perform relatively well. There is also little difference between the *Tracer-Nudge-QuadFit* and the *Tracer-Con-QuadFit* method. Since the latter is less computationally expensive, it is considered more favourable. Hence, this algorithm is also selected to process the skin friction topology on the remaining objects, presented in chapter 7.

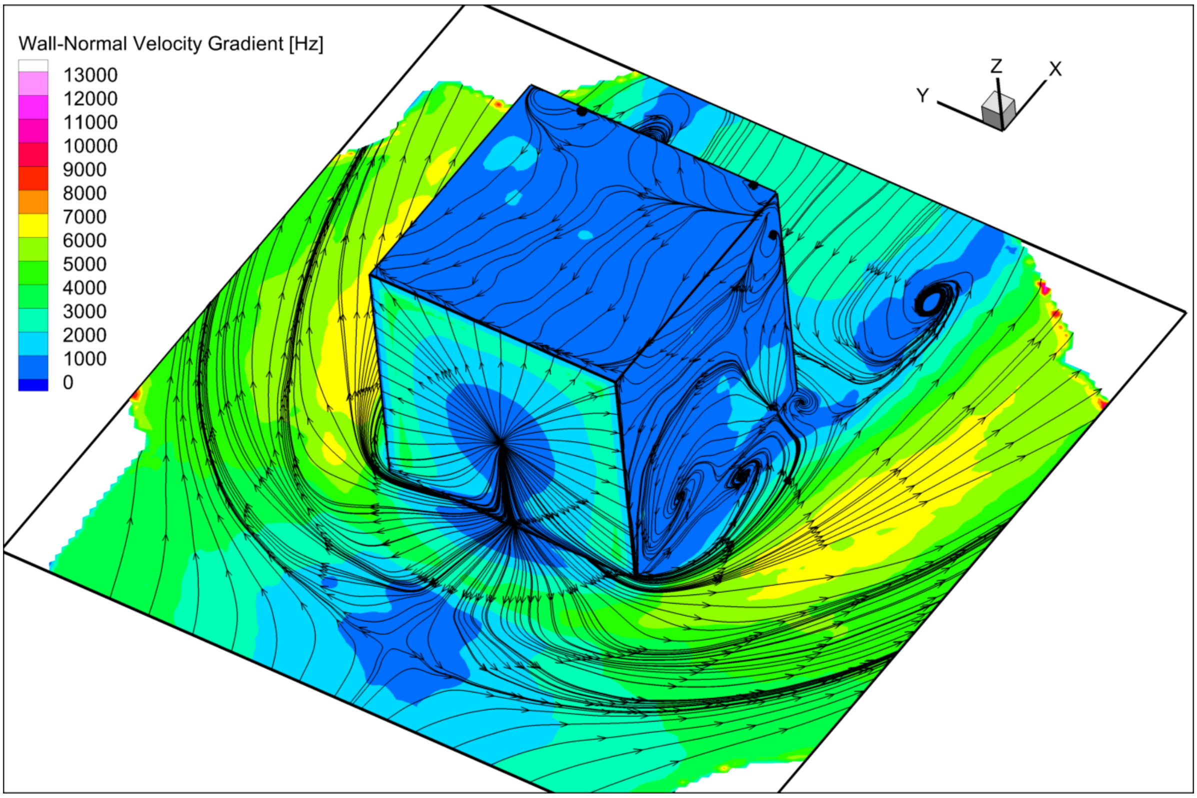


Figure 6.22: Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for the ***Tracer-Nudge-QuadFit*** method.

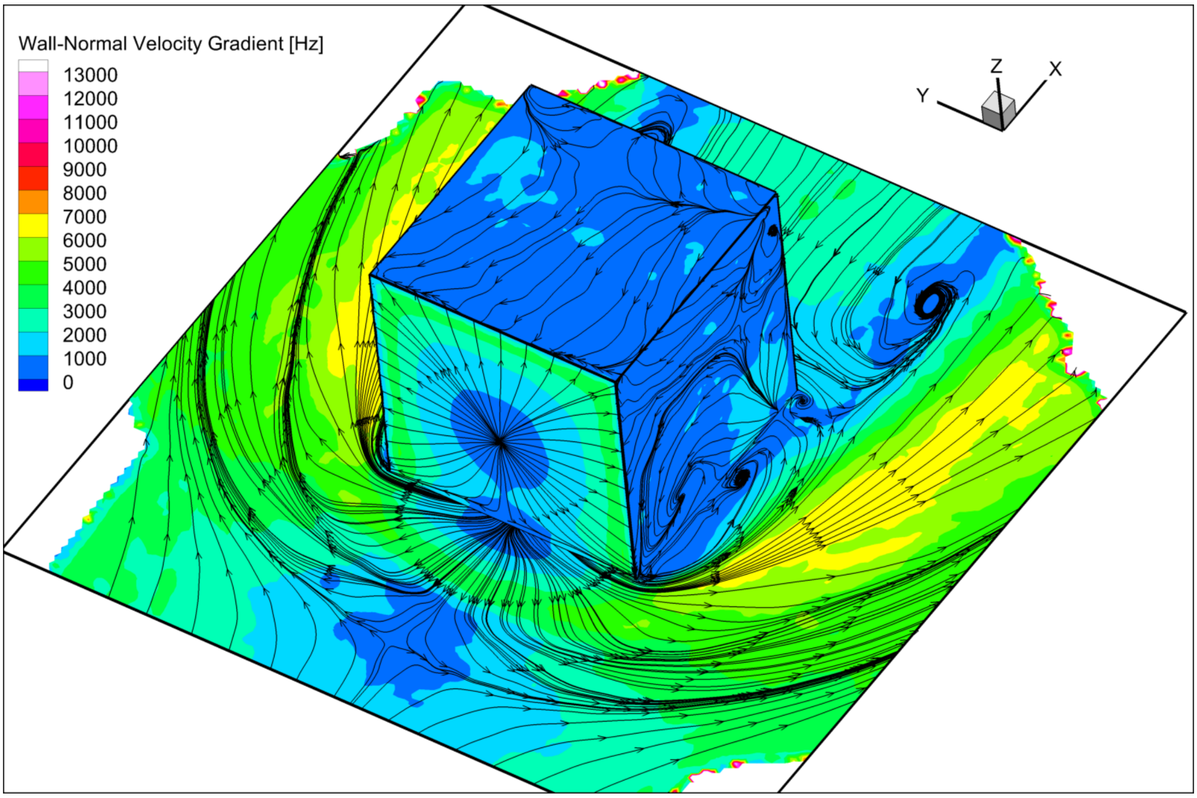


Figure 6.23: Contour plot with magnitude of wall-normal velocity gradient overlaid with skin friction lines for the ***Tracer-Con-QuadFit*** method.

7

Skin friction topology around generic objects

The three cases of the cube, wing-body junction and the cyclist, presented in chapter 4 are processed with the *Tracer-Con-QuadFit* method. This chapter presents the topology of the skin friction for each case, along with the magnitude contours. The purpose of this chapter is to highlight the flexibility of the LIVA approach, proposed in this thesis report.

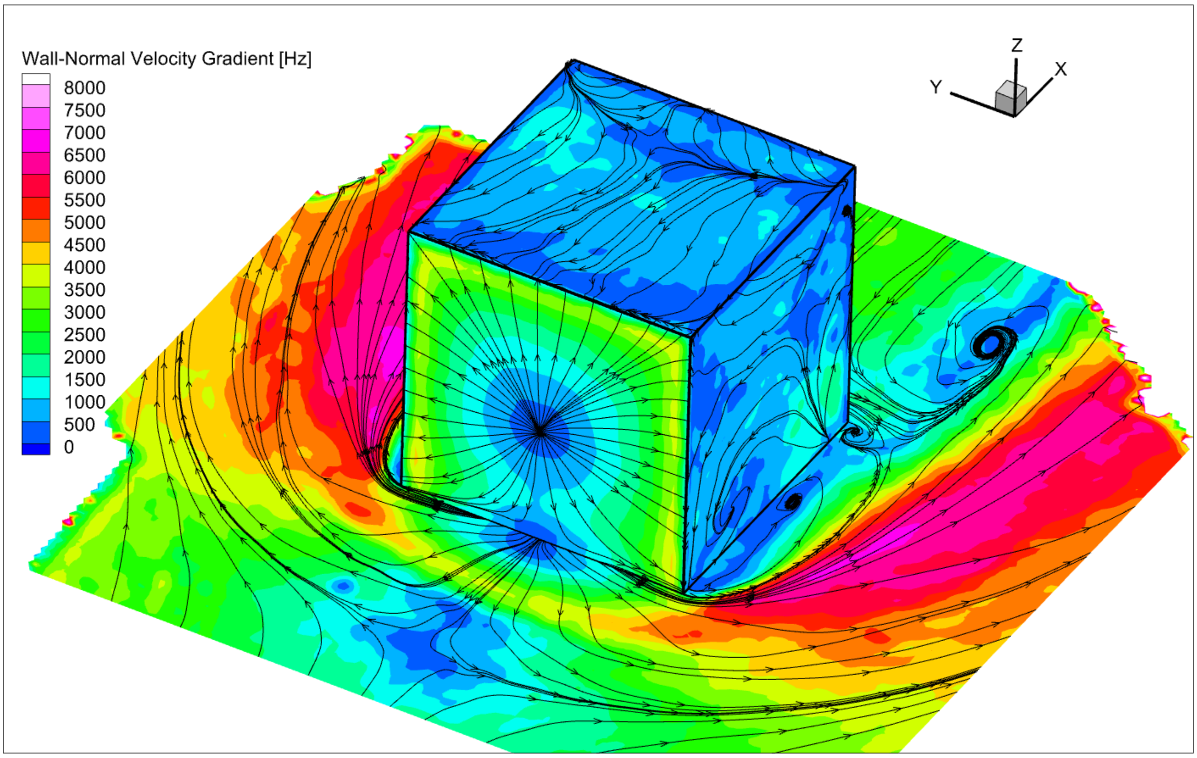
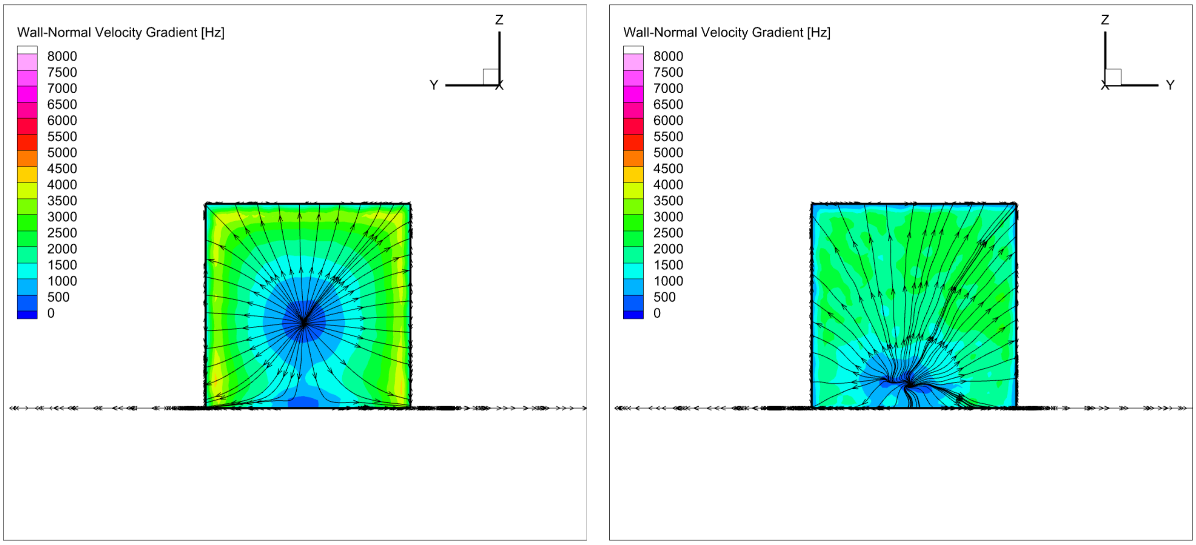


Figure 7.1: isometric view of the skin friction magnitude around the surface-mounted cube overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

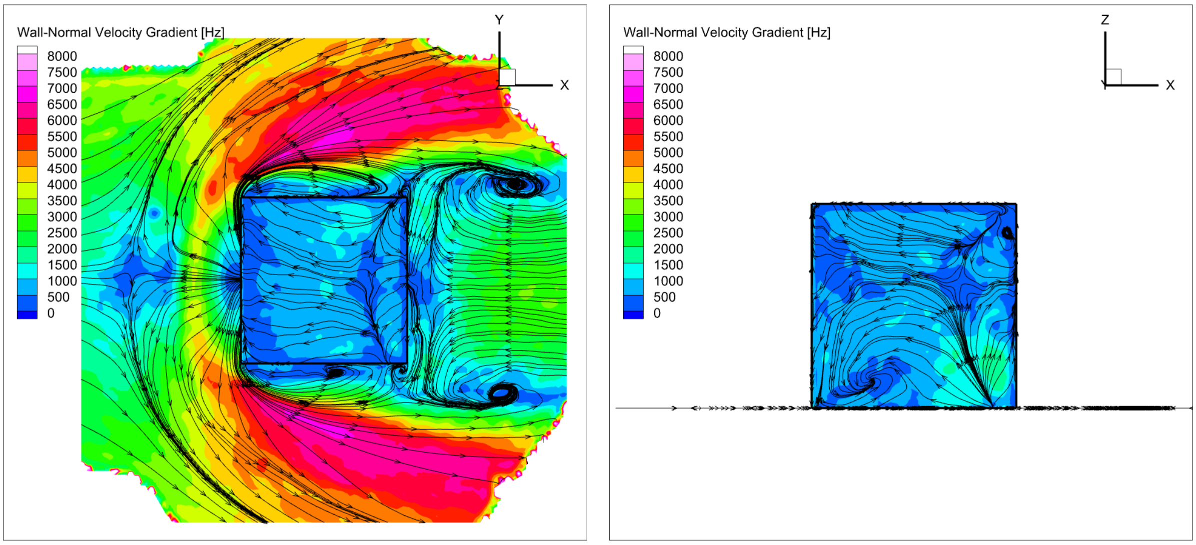


(a) (b)

Figure 7.2: Views on the (a) front face and (b) rear face of the skin friction magnitude around the surface-mounted cube overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

7.1. The surface-mounted cube case

Firstly, the results are presented for the surface-mounted cube. This dataset has been extensively covered in chapter 5 and the results will be summarised in this section. The cube is considered the geometrically simplest case, but the bluff body and sharp edges induces a flow that may informally be referred to as ”riddled with shedding vortices”. An isometric view of the cube is depicted in figure 7.1. The incoming flow separates forward of the cube and the skin friction lines converge from this point on to single lines which arch around the object. This is indicative of the typical horseshoe vortex that exists in front of the cube. An attachment point is present on the front face of the cube, from which the attached flow spreads evenly to all edges, representative of an impinging jet flow (see also subfigure 7.2a). The sharp edges on the cube induce separation, as a real flow is unable to sustain an infinite acceleration to turn a $90°$ corner.



(a) (b)

Figure 7.3: Views on the (a) top face and (b) side face ($y = -60$) of the skin friction magnitude around the surface-mounted cube overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

The wake induced by the leading edge of the cube extends to far aft of the cube (see also figure 6.14). Part of the separated flow reattaches on the upper face and hence a large portion of the upper face experiences negative friction in the negative x-direction. A saddle point is present at the point where the flow reattaches to the upper surface. Part of the fluid attaching at this saddle point flows towards the rear of the cube, which is another separation for the flow impinging the cube from behind. The attachment of the flow on the rear can be seen in subfigure 7.2b.

Turning the attention to the top view of the cube in subfigure 7.3a, two large vortices can be seen located behind the cube and positioned approximately one cube length apart. These are the footprints of the arch vortices which are induced by flow separation from the front face of the cube. The reattachment point of the large leading edge separated wake is not captured inside the measurement domain and lies further aft of the cube. Smaller vortices can be detected on the ground plane near the sides of the cube and near the vertical edges of the rear face, which are a result of separation of forward and backward flow, respectively, from the side edges. Lastly, two saddle points can be viewed between the horseshoe vortex and the rear face of the cube.

A side view of the cube is presented in subfigure 7.3b. The skin friction topology on this side includes two vortices, one lower near the front face and one higher near the rear face. A saddle point is present slightly aft of the centre, and two more attachments points appear to be located at the face edges above and below this saddle point.

7.2. Processing the wing-body junction case

The next case is the wing-body junction, which is a typical flow case for aircraft. A top view of the WBJ is displayed in figure 7.4. The angle of attack can be estimated to be around $6°$ from this view and the suction side is located towards $-y$. Contrary to literature, the skin friction topology does not show the presence of the characteristic horseshoe vortex on the ground forward of the leading edge.

An isometric view of the skin friction topology and distribution is given in figure 7.5. The flow attaches evenly on the leading edge of the airfoil. The suction side can be clearly recognised by the region of high wall-shear, which reaches a maximum approximately where the thickness is largest. The flow over the middle of the airfoil is attached up to the trailing edge. The flow near the top of the wing shows

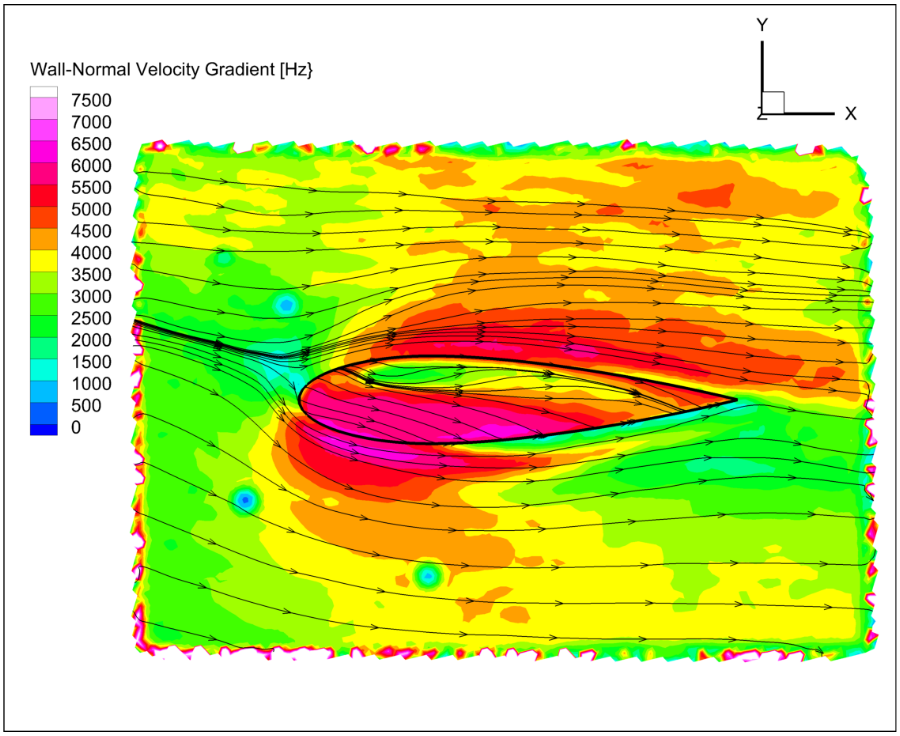


Figure 7.4: Top view of the skin friction magnitude around the WBJ overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

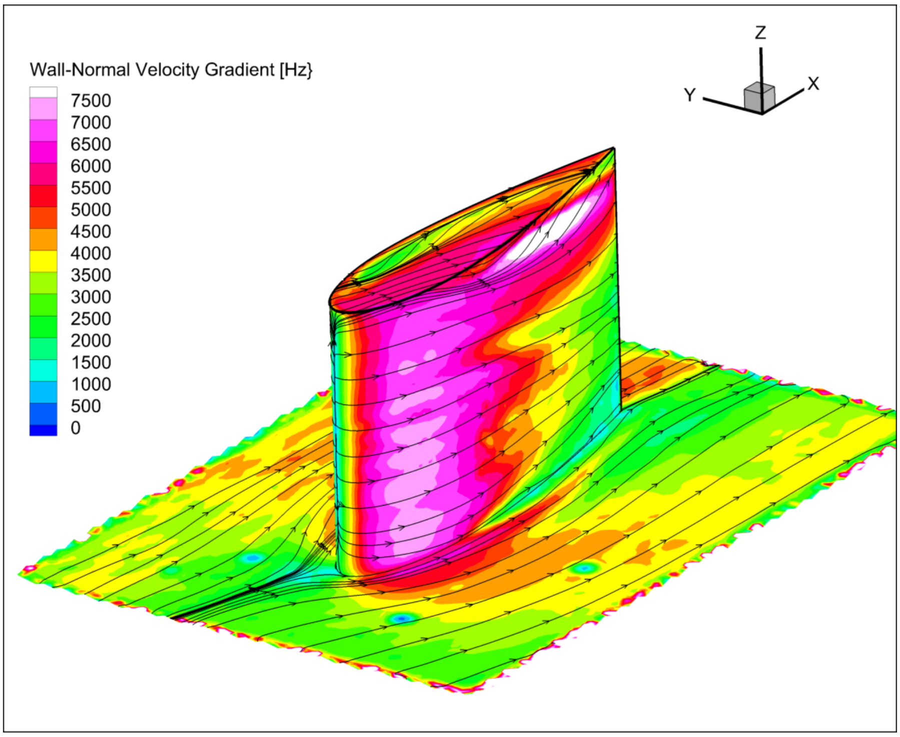


Figure 7.5: isometric view from a forward position of the skin friction magnitude around the WBJ overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

a different behaviour due to the three-dimensionality of the flow. At the top of the leading edge, a tip vortex is created which moves up from the pressure side and rolls over the edge on the pressure side,

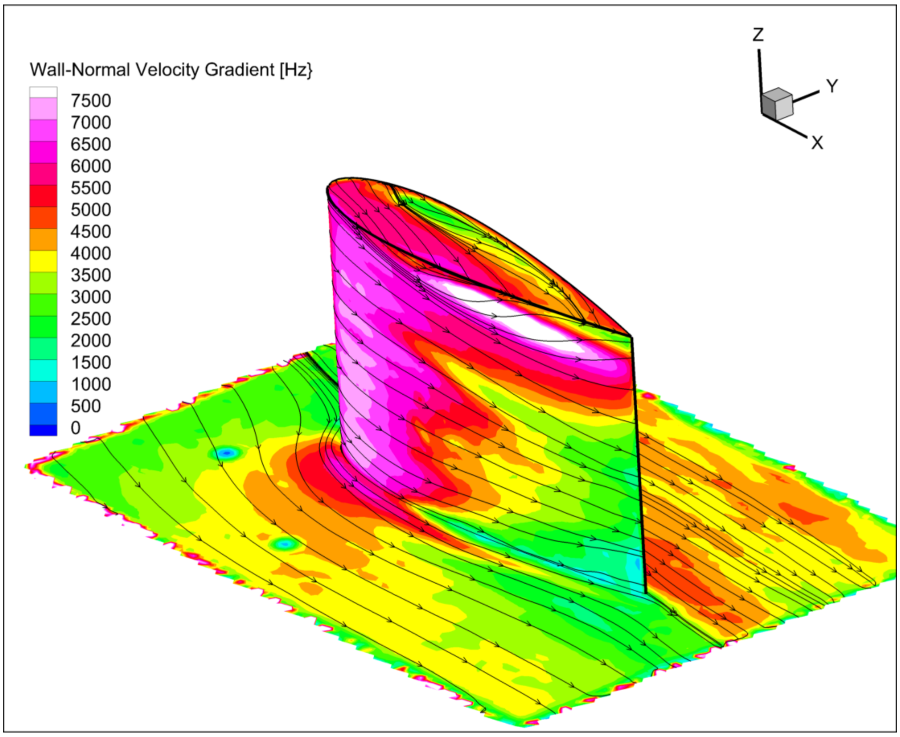


Figure 7.6: isometric view from a rearward position of the skin friction magnitude around the WBJ overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

rotating in clockwise direction along the x-axis. The tip vortex flow reattaches again on the side of the airfoil and rolls up on near the edge of the pressure side. This causes the *S*-shaped friction lines moving towards the pressure side and is generally paired by a secondary separation underneath the tip vortex.

A second tip vortex occurs due to flow on the WBJ's side face rolling up over the edge on the suction side. A streak of increased wall-shear stress is visible on the suction side towards the tip region. This may be indicative of a pressure-driven vortex that accelerates the flow near the top.

7.3. The cyclist case

The skin friction distribution around the scaled cyclist is discussed last. This is topologically the most complicated model. An isometric view of the skin friction lines and wall-normal gradient is presented in figure 7.7. The flow over the cyclist is largely attached and only separates via the two vortices on the rear of the cyclist and a separation line on the back of the upper arms and legs, which matches the results by Jux et al. [2018]. A stagnation point exists at the tip of the rider's head where the flow attaches to the cyclist. A separation region can be recognised in the neck from the decrease in skin friction magnitude and the converging skin friction lines. The flow also attaches on the biceps of the cyclist after which it separates again on from the triceps.

Turning to the isometric view of the cyclist's back in figure 7.8, the separation lines and vortices are clearly visible. Long separation lines are present on the back of the arms and legs. A small vortex can also be identified near the left elbow of the cyclist. Compared to the findings of Jux et al., the two vortices on the lower rear of the cyclist are positioned slightly higher up the cyclist's body in the present case. This is likely a consequence of the lower Reynolds number, which induces earlier separation. The wall-normal velocity gradient varies from 0 to approximately $5.000\,\mathrm{Hz}$ with the regions of highest shear found on the cyclist's back and the sides of the legs where the attached flow reaches a local maximum.

Next, the front and rear view of the cyclist are displayed in figure 7.9. The front view shows the

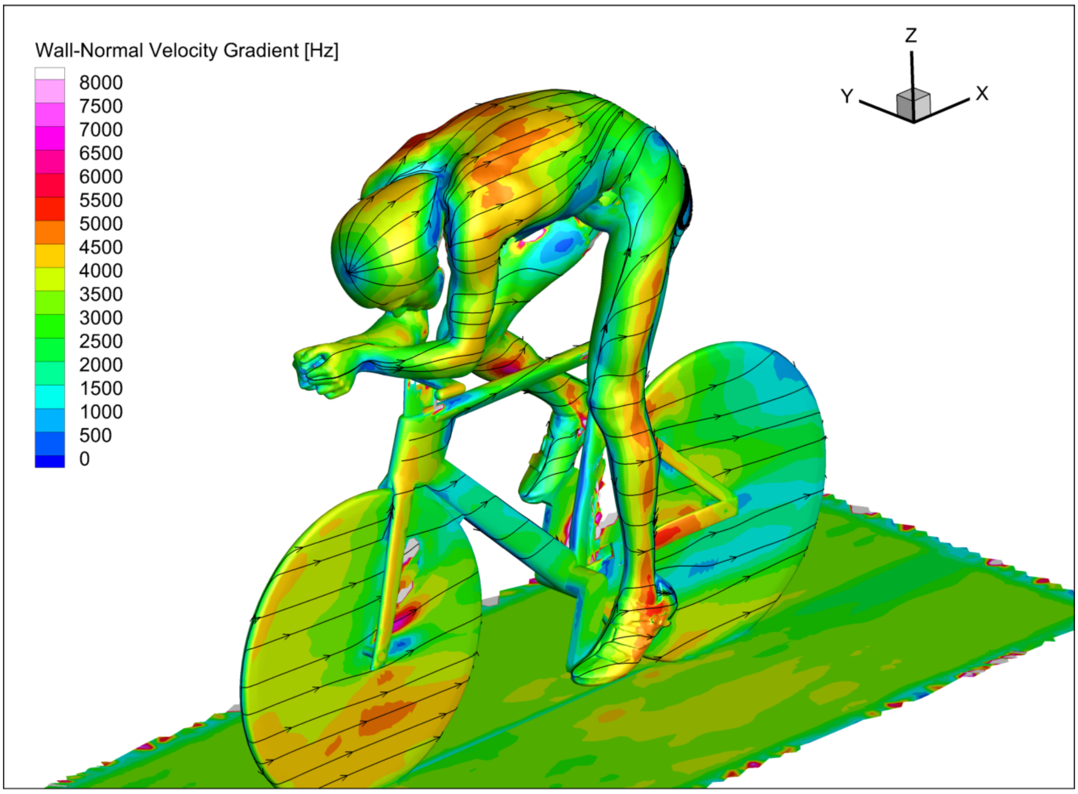


Figure 7.7: isometric view from a forward position of the skin friction magnitude around the cyclist overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

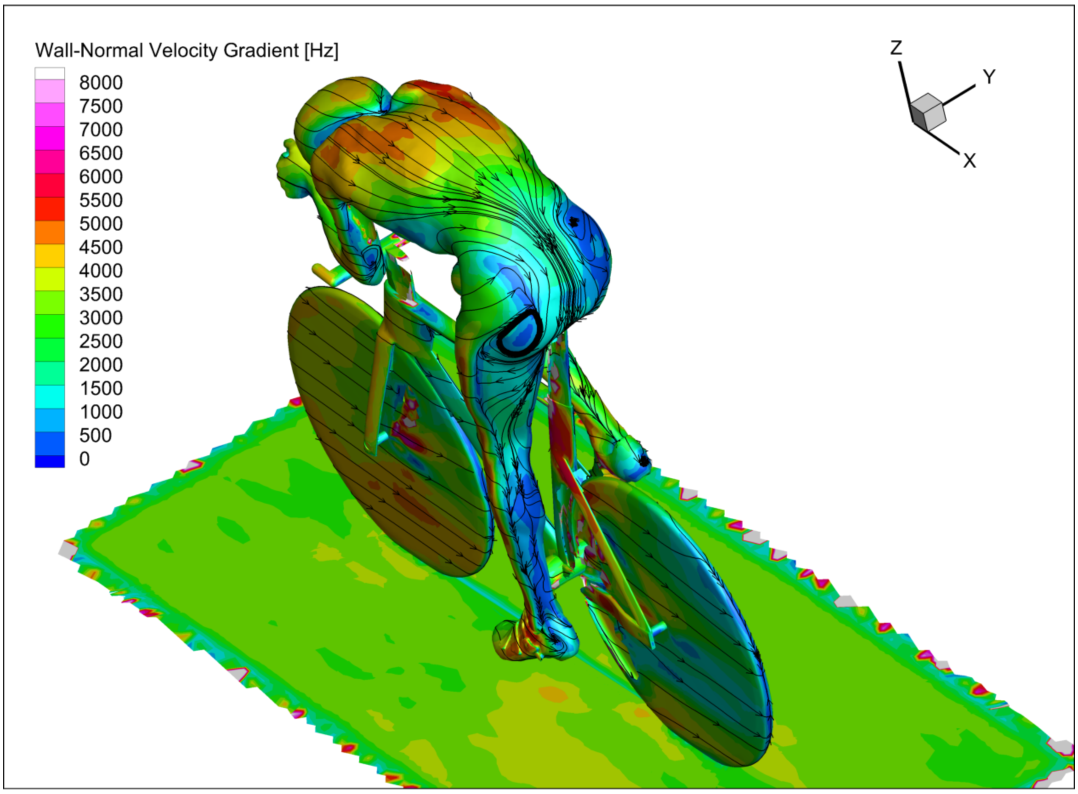
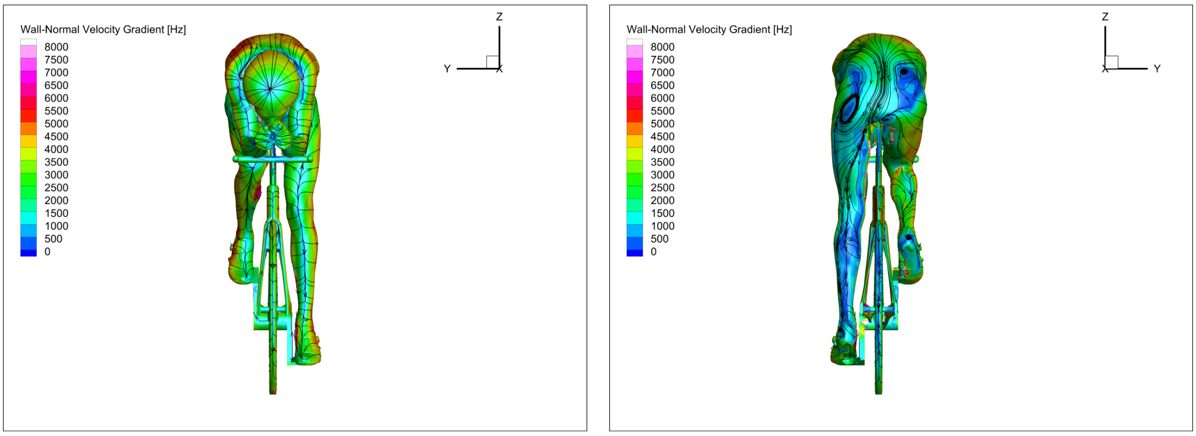


Figure 7.8: isometric view from a rearward position of the skin friction magnitude around the cyclist overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

extend of the attachment lines running from the hip of the cyclist down the the toes. A saddle point can be identified on both knees and a saddle point is present between the right shoe and shin of the cyclist. The rear view shows long separation lines which run from the gluteus of the cyclist down to the ankles. The lower left vortex is about twice the size of the upper left vortex. Down from the upper right vortex, another separation line can be seen which moves towards the inner thighs, on the back of the right upper leg, the friction lines appear to move upward towards the gluteus.



(a) (b)

Figure 7.9: Two views of (a) the front side and (b) the back side of the skin friction magnitude around the cyclist overlaid with skin friction lines, obtained with the *Tracer-Con-QuadFit* approach.

8

Conclusions and Recommendations

The recent advancements in the three-dimensional flow measurement techniques, particle image velocimetry (PIV) and Lagrangian particle tracking (LPT), towards large-scale experiments, coupled with the introduction of the object registration approach by Hendriksen [2024] motivated this thesis to investigate the inference of near-wall surface fluid dynamics from particle tracking results. Sustainability plays a crucial role in the design and analysis of new aircraft, vehicles, wind turbines and more. Both PIV and LPT are currently widely applied in both state-of-the-art industrial and academic pursuits due to their rich data acquisition. Various examples have been presented demonstrating its capabilities such as the measurement around a full-scale cyclist by Schneiders et al. [2018] and Jux et al. [2018] or the complete characterisation of the flow around a surface-mounted cube by Schröder et al. [2020].

It was previously demonstrated that pressure distributions can also be inferred from the three-dimensional flow field measurements with both PIV ([de Kat and van Oudheusden, 2012]) and LPT ([Jux et al., 2020]) up to model surfaces. To obtain a complete picture of the aerodynamic forces, only the skin friction need still be determined ([Anderson, 2017]). This is typically done through oil film interferometry ([Örlü and Vinuesa, 2020]), however, these oils may not always be applicable when gravity effects becomes significant. The ability to also infer skin friction distributions from volumetric measurements would complete the story and add much value to the already packed toolbox that are PIV and LPT.

Current data assimilation approaches utilising large-scale PIV and LPT results have mainly focused on far-body flows. Near-surface methods have been proposed and experimented with, but this has been limited to the two-dimensional case and viscous terms have often been neglected. This leaves a research gap to utilise large-scale PIV/LPT to reconstruct near-surface fluid dynamics, with a specific focus on skin friction distribution, for generic, complex bodies, summarised by the following research objective:

The research objective is to reconstruct near-surface fluid dynamical properties in large-scale time-averaged PIV/LPT measurements by combining verified body position information with experimental results in the outer-field.

The research objective was divided into three main topics, which dealt with I) reconstruction of the near-surface velocity, II) computation of the skin friction and III) implementation of the method into the current state-of-the-art. To tackle the first question of how the model domain and fluid domain can be merged together, the local interrogation volume approach (LIVA) was introduced along with an output-mesh onto which the merger can be interpolated. An object registration of the model using the same optical setup yields a transformation matrix. This transforms the coordinate system of a digital computer-aided design (CAD) model (typically provided in *.STL* format) to the fluid domain. Local spherical interrogation volumes were used to bin both fluid measurements, which are represented by a fluid-mesh, and portions of the model geometry, represented by an surface-mesh.

When dealing with LPT results, the fluid-mesh may be represented by the a single ensemble of all individual tracers or reduced to a Cartesian grid using bins. Next, the question arose as to how the near-surface velocity can be reconstructed from the merger in the LIVA. A family of four different methods was introduced which aim to interpolate the local velocity inside the interrogation volume by accounting for the model geometry. These are summarised as follows:

- *Bin-Con-LinInterp* - A three-dimensional Delaunay triangulation is generated using an ensemble of bin-based fluid-mesh points and a point-discretised surface-mesh. A trilinear interpolation is subsequently employed using barycentric coordinates of the three-dimensional simplices to interpolate the velocity onto an interrogation point inside the volume.
- *Bin-Nudge-QuadFit* - Wall ghost (WG) points are introduced and distributed over the intersection of the surface-mesh which lies inside the interrogation volume. The WG points are sampled with the same inter-nodal distance as the bin-based fluid-mesh nodes, which is the grid vector spacing h. The WG points with zero velocity and fluid-mesh nodes are concatenated into a single array of data points. A three-dimensional quadratic velocity functional is then fitted to these points using a least-squares approach.
- *Tracer-Nudge-QuadFit* - Similar to *Bin-Nudge-QuadFit* WG points are sampled over the intersection of the surface-mesh and the interrogation volume. The tracer-based representation of the fluid-mesh is used and a single, fixed inter-nodal distance is determined from the global particle density. All data points are concatenated into a single array and the least-squares method is used to fit a quadratic polynomial to the data points.
- *Tracer-Con-QuadFit* - The WG points are swapped for a single constraint at the projection of the interrogation point onto the wall. A constrained least squares problem is set up to solve the minimisation problem subject to the single boundary condition. The same quadratic velocity polynomial is used for this approach.

The implementation of the boundary conditions is included in the descriptions above. To lastly answer how the family of methods compare against the state-of-the-art, the boundary layer velocity profiles are computed, along with the skin friction distribution. In regards to inferring the skin friction distribution from particle tracking results, Jux' mehthod ([Jux et al., 2018]) is considered the state-of-the-art. Jux' method employs a dilated output-mesh which was then extrapolated to the model surface to acts as the state-of-the-art solution in this report. These methods were further compared against a benchmark solution, which employed the anisotropic nature of boundary layers. Flat, coin-shaped bins were used as interrogation volumes and a linearised mean was computed inside each disk.

The methods were applied to the experimental dataset from a large-scale LPT experiment for three models, a surface-mounted cube, wing-body junction and scaled cyclist. First the velocity profiles were constructed. This revealed the under- and over-prediction that occurs with quadratic fits, when the interrogation volume is several orders higher than the boundary layer. This places a requirement on the experimental data and the LIVA approach. TO reconstruct near-wall flows without over-and under-predicting, the maximum radius of the interrogation volume must be approximately half the size of the characteristic boundary layer in the flow. Based on this requirement, the minimum number of images could be determined.

Further, the skin friction topology was accurately reconstructed by all methods from a qualitative perspective. The state-of-the-art method does not allow for determining the topology via the wall-normal gradient and the near-surface streamlines would need to be employed. The effect of the non-conformal bin-based fluid-mesh with respect to the model geometry was clear in the case of the surface-mounted cube. Due to the object registration correction the ground plane was no longer aligned with the bin-based fluid-mesh. As the linear method directly scales the wall-normal gradient with the distance to the wall, an artificial aliasing effect was noticeable distorting the skin friction magnitude. The same effect was apparent for the *Bin-Nudge-QuadFit* method. The difference between the methods *Tracer-Nudge-QuadFit* and *Tracer-Con-QuadFit* was minimal with respect to the skin friction distribution. The method which employs WG points yielded smoother results, whereas the single wall constraint approach is computationally cheaper.

Lastly, the research question on how this method could be integrated into the state-of-the-art was addressed. A metric on the experimental parameters was mentioned above and what remains is the practical implementation. Typically, digital particle tracking images are processed using specialist software such as DaVis ([LaVision, 2023]). To that end, a processing console was created and presented which can be used to set up the processing, get visual insight into the processing with local interrogation volumes and finally simplify the processing of a particle tracking experiment for the user. A tutorial on this software is provided in the appendix.

Various scientific domains were touched upon during this thesis project, from computational geometry and numerical analysis to experimental measurements and even software design. The outcomes of this thesis can be further sustained by suggesting improvements to the different facets of the project. Firstly, the interrogation volume was chosen to be sphere as an initial suggestion for an isotropic approach. For the same reasoning as the coin-shaped bins, the sphere may be swapped by an anisotropically shaped element, which may align with the direction of largest correlations. Inspiration may be drawn from the adaptive interrogation volumes proposed in [Novara et al., 2012], which used *cigar*-like and *disk*-like volumes for binning to improve resolution in the shear layer. The need for the object registration correction on the ground plane was demonstrated as this offset reached up to $5\,\mathrm{mm}$ for the case of the wall-mounted cube and even higher for the wing-body junction. The cause of this misalignment was largely due to the wooden material used which easily bent and the fact that the experimental campaign was focused on a correct registration of the models and little attention was given to the ground plane. For further experiments which will partake in a similar endeavour, it is recommended to source rigid materials, choose the manufacturing technique based on the allowable model position uncertainty and, lastly, include the ground plane (and potentially other objects) in the object registration process.

Next, it was clear that tracer particles provided the best fluid-mesh representation. This was largely due to the double binning effect which would occur otherwise. Additionally, the large binning may already wash out the required spatial resolution for an accurate near-surface velocity reconstruction. Further improvements could firstly be made to the the velocity functional. As per [Agüera et al., 2016], a quadratic polynomial was used, but this is likely a sub-optimal functional for the near-wall velocity. One option is the use of different interpolation schemes, for example a mimetic interpolation ([Pletzer and Hayek, 2019]) which satisfies vector calculus identities, such as divergence and Stokes' theorem. A comment was also given on the use of wall functions for the near-surface velocity fitting, which worked well for attached turbulent boundary layer. A recommendation would be to not only adapt the quadratic functional, but further to use a hybrid approach. Coles' method (Coles, 1968) could be used for attached flows and a switch could be made to a higher-order polynomial when deviation from the canonical profile is detected, for example due to separation.

Lastly, the LIVA console to process particle tracking results could be further developed, but a more powerful direction of development might be the integration of the console into other software such as DaVis. This way the necessary processes could be launched directly from the DaVis software. Currently, the console outputs a single file in *.vtu* format which could be loaded by any fluid data viewer, such as TECPLOT$^{\mathrm{TM}}$. To make the data generation more compact and move toward a computational fluid dynamical treatment of this data, e.g. with finite element meshes, it is recommended to investigate how the volumetric and surface outputs can be combined in a single data file.

Appendices

A

Fitting boundary layer profiles with wall functions

As discussed in subsection 2.2.2, a common approach to determine wall-shear stresses is by fitting wall functions, such as the Clauser chart method. Fitting a wall function is generally applied to different regimes of the boundary layer, which may be the entire region or a subset such as only the inner layer, depending on data availability. The classic velocity profile in a turbulent boundary layer has been discussed in subsection 2.2.1. The profile is divided into the inner and outer layer. Inside the inner layer, the profile follows a logarithmic law and close to the wall in the viscous sublayer, the non-dimensionalised velocity and wall-normal distance exhibit a linear relation up to approximately $z^+ = 5$. In between these layers, in the buffer layer, these profiles overlap.

The mean radius of the HFSB is approximately $0.35\,\mathrm{mm}$. Typical near-wall turbulence measurements are executed with tracer particles which sizes are in the order of $1\,\mu\mathrm{m}$ ([Schröder et al., 2015] and [Kempaiah et al., 2020]). In the turbulent boundary layer on the ground forward of cube, the z-value corresponding to $z^+ = 1$ (or $z = \nu/u_\tau$) would be $\mathcal{O}(10^{-2}\,\mathrm{mm})$ ([da Silva et al., 2021]). On the faces of the cube, this value is about 25% smaller. Hence, in this case one could only expect to find tracked particles in the boundary layer region from $z^+ > 35 - 50$, which corresponds to the log region and outer layer. Introducing a single wall point to specify the zero-velocity constraint may improve the wall function modelling, however, as was shown by Örlü et al. [2010] inaccuracies in the wall position mostly affect the reconstructed velocity profile in the viscous wall region and not so much in the log region and outer layer, see also subsection 2.3.3.

This appendix demonstrates four fitting strategies using the Clauser chart method and Coles' method. The strategies are first explained in subsection A.1 and applied to four velocity profiles. The results are obtained using a nonlinear least squares fit, which are presented and discussed in subsection A.2

A.1. Fitting with the Clauser chart and Coles' method

As outlined by Rodriguez et al. [2015], the condition of tracked particles from $z^+ > 35 - 50$ can be fit using the Clauser chart method, when only the log region is used, or Coles' method when both log and wake region (= outer layer) are utilised. For the purpose of this demonstration, both methods are applied to the wall-mounted cube case at different positions on the ground plane and model. These positions are the same as in subsection 6.3.1 and displayed in figure 6.14.

The Clauser chart method ([Clauser, 1954]) utilises the LogLaw equation (2.32), which is repeated here for convenience, using z as the wall-normal direction:

$$u^+ = \frac{1}{\kappa} \ln\left(z^+\right) + B \tag{A.1}$$

Table A.1: Summary of variables for the four fitting strategies with the Clauser chart method and Coles' method and the role these play in the fitting.

	Dependent variable	*Measured variable*	*Fixed variables*	*Design variables*
Clauser chart method I - equation (A.1)	z	U	κ, B, ν	u_τ
Clauser chart method II - equation (A.1)	z	U	ν	u_τ, κ, B
Coles' method I - equation (A.5)	z	U	κ, B, ν	u_τ, Π, δ
Coles' method II - equation (A.5)	z	U	ν	u_τ, Π, δ, κ, B

The introduction of the non-dimensional parameters z^+ and u^+ is expected to collapse all boundary layer velocity profiles onto a single profile, dependent on the friction velocity u_τ. Hence, to explicitly show this dependency, it may be convenient to rewrite above equations in the dimensional parameters as

$$U = \left[\frac{1}{\kappa}\ln\left(\frac{z \cdot u_\tau}{\nu}\right) + B\right] \cdot u_\tau, \tag{A.2}$$

which shows that with a fixed von Kármán constant κ and loglaw intersect B, the parameter u_τ is the only variable (since the kinematic viscosity is determined by the ambient conditions). However, as has been discussed in subsection 2.2 and various other literature ([Örlü and Vinuesa, 2020]), the values of κ and B are contested ([Wei et al., 2005]) and even their universality for various flow types ([Nagib and K. Chauhan, 2008]). Therefore, given sufficient flow field data, it may be possible to include κ and B in the set of design variables. By fitting the data to the logarithmic region of the boundary layer $50 < z^+ < 200$, the friction velocity can thus be determined.

To include data of the outer layer region into the wall function, Coles' method introduces a wake term ([Coles, 1956]) meant to correct the defect from the log law. This correction is parallelly named the 'Law of the Wake' and is given in non-dimensional parameters by

$$u^+ = f\left(z^+\right) + \frac{\Pi}{\kappa} w\left(\eta\right). \tag{A.3}$$

The attentive reader notices the introduction of two new variables, the wake component Π and the non-dimensional wall-normal variable $\eta = z/\delta$, where δ is the boundary layer height. The functional f is simply the law of the wall, however, the functional w represents the law of the wake. For the purposes of this demonstration, the analytical expression for $w(\eta)$ by Coles [1968] is used here and presented in equation (A.4), though more recent expressions are available ([K. A. Chauhan et al., 2009]).

$$w\left(\frac{z}{\delta}\right) = 2 \cdot \sin^2\left(\frac{\Pi z}{2\delta}\right) \tag{A.4}$$

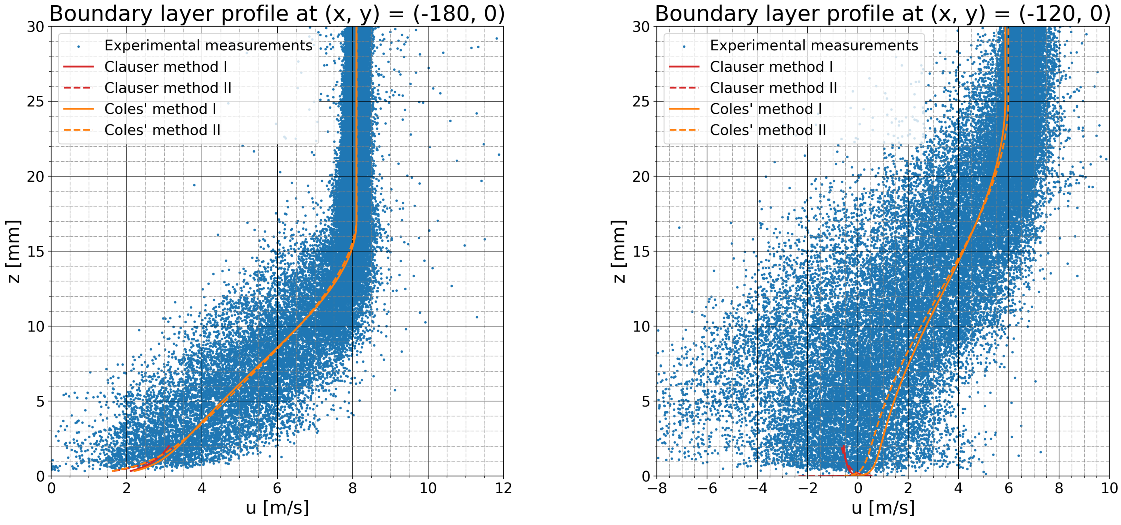
Introducing the law of the wall may extent the region of applicability to the boundary layer edge, δ. By substituting equation (A.4) into equation (A.3) and introducing the dimensional parameters, the dependencies on all variables is clearly indicated:

$$U = \left[\frac{1}{\kappa}\ln\left(\frac{z \cdot u_\tau}{\nu}\right) + B + \frac{2\Pi}{\kappa} \cdot \sin^2\left(\frac{\pi z}{2\delta}\right)\right] \cdot u_\tau \tag{A.5}$$

Both methods are applied to fitting the four boundary layer profiles shown in figure 6.14 using the tracer-based fluid-mesh, assuming a constant $\kappa = 0.41$ and $B = 5.0$ and optimising for these constants. This yields in total four solutions, which are summarised in table A.1. To gather a sufficient number of particles for the fitting, a cylindrical interrogation volume with radius of $12\,\mathrm{mm}$ is used which axis aligns with the velocity profile position. Note that only the u-component of the velocity is used for the boundary layer fitting. The problem is solved using a nonlinear least squares fit.

A.2. Results of wall function fitting

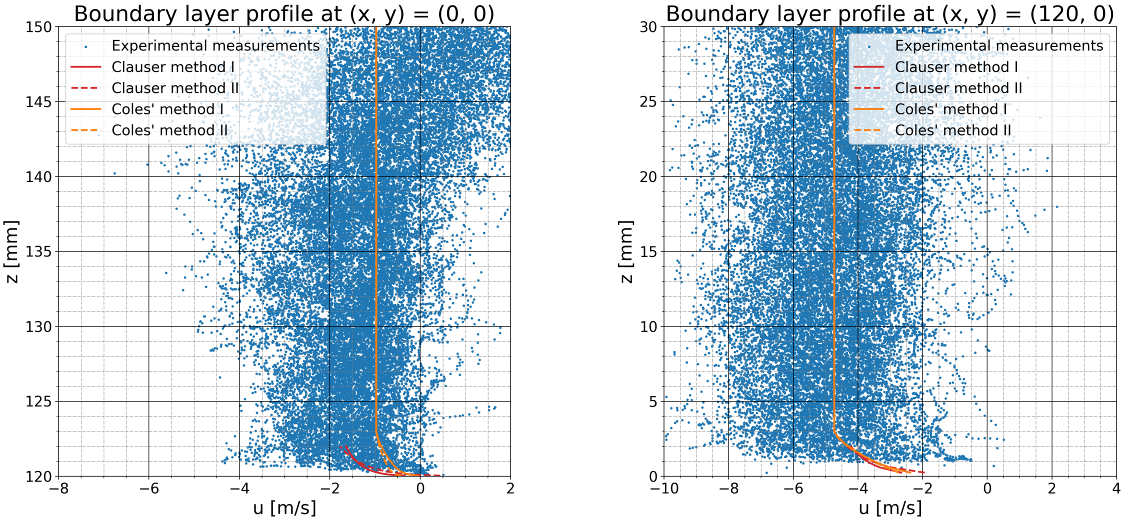
The results are shown in figures A.1 & A.2. Firstly, velocity profile I is shown in subfigure A.1a and can be recognised as a classic attached turbulent boundary layer. Large fluctuations occur in the



(a) (b)

Figure A.1: Results of four wall function fitting strategies in table A.1 to the tracer particles at (a) the position (x, y) = $(-180, 0)\mathrm{mm}$ and (b) at (x, y) = $(-120, 0)\mathrm{mm}$.

boundary layer, which decrease near the edge. For all profiles, the Clauser chart method is applied up to $z = 2$, which correspond to $z^+ \approx 2\,\mathrm{mm}$. Both the Clauser chart method and Coles' method can accurately fit the turbulent velocity profile. Clauser method I yields a friction velocity of $0.233\,\mathrm{m\,s^{-1}}$ and Coles' method I yields a friction velocity of $0.241\,\mathrm{m\,s^{-1}}$ which are similar outcomes. Given the friction velocity the viscous count ν/u_τ can be updated and found to be approximately $0.061\,\mathrm{mm}$. The advantage of applying Coles' method is the additional information on the wake component $\Pi = 3.06$ and the boundary layer height δ $=16.6\,\mathrm{mm}$. The additional information from Coles' method could be used to directly compute the skin friction coefficient C_f, the displacement thickness δ^*, the momentum



(a) (b)

Figure A.2: Results of four wall function fitting strategies in table A.1 to the tracer particles at (a) the position (x, y) = $(0, 0)\mathrm{mm}$ and (b) at (x, y) = $(120, 0)\mathrm{mm}$.

Table A.2: Results of the nonlinear least squares fitting with uncertainty to velocity profile I for all four fitting approaches.

	u_τ [m s^{-1}]	Π [-]	δ [mm]	κ [-]	B [-]
Clauser chart method I - equation (A.1)	$0.233 \pm 1.95 \times 10^{-3}$	-	-	-	-
Clauser chart method II - equation (A.1)	6.39×10^{5}	-	-	7.04×10^{5}	2.46×10^{7}
Coles' method I - equation (A.5)	$0.241 \pm 6.17 \times 10^{-4}$	1.95×10^{-2}	4.03×10^{-5}	-	-
Coles' method II - equation (A.5)	9.91×10^{3}	5.59×10^{-2}	6.72×10^{-5}	1.11×10^{4}	3.88×10^{5}

thickness θ and lastly, the shape factor H via the relations ([White, 2006])

$$\tau_w = u_\tau^2 \cdot \rho \tag{A.6}$$

$$C_f = \frac{\tau}{\frac{1}{2}\rho U_e}, \tag{A.7}$$

$$\lambda = \sqrt{\frac{2}{C_f}}, \tag{A.8}$$

$$\frac{\delta^*}{\delta} \approx \frac{1+\Pi}{\kappa\lambda}, \tag{A.9}$$

$$\frac{\theta}{\delta} \approx \frac{\delta^*}{\delta} - \frac{2 + 3.2\Pi + 1.5\Pi^2}{\kappa^2\lambda^2}. \tag{A.10}$$

For the boundary layer velocity profile at location I, this yields $\tau = 0.071\,\mathrm{Pa}$, $C_f = 0.0018$, $\lambda = 33.6$, $\delta^* = 4.89 \times 10^{-3}$, $\theta = 2.6 \times 10^{-3}$ and $H = 1.86$.

The results of the fitting approaches Clauser method II and Coles' method II are plotted in the same figures. The results seem to fit the experimental measurements well, however, these do not tell the full story. Upon performing the nonlinear least-squares, the uncertainty in the estimate of each design variable can be obtained. In both cases, adding κ and B to the set of design variables negatively impacts the accuracy of fitting the friction velocity, as well as κ and B. This is visible from the values in table A.2 in which the uncertainties for all approaches are summarised.

The other velocity profiles all include a region of separation. Profile II has a small region of adverse flow, but this can not be accurately fitted by equations (A.1) or (A.5) which are specially derived for turbulent attached boundary layers. A small "trick" was applied to fit the velocity profiles III and IV, namely that the sign of the fit was flipped for adverse flow. The highly separated regions at III and IV contain a secondary attached boundary layer, which moves in the direction of negative x. At first sight, the Clauser method still holds up in this separated region, to fit the secondary boundary layer. Upon closer inspection, however, the experimental measurement data shows a lower wall-normal velocity gradient than the Clauser fit for velocity profile III and a higher wall-normal gradient for the velocity profile IV. Similarly, Coles' method fails as the loglaw defect can not be corrected by the introduced law of the wake.

This appendix aimed to highlight the potential advantages and apparent disadvantages of employing analytical wall functions for fitting. The wall-mounted cube provides a case with largely separated flows, which is incompatible with the Clauser chart method and Coles' method that rely on attached turbulent boundary layers (in fact, there is no reliable, analytical equation for separated flows). An approach can be envisioned, however, in which part of the solution which encompasses attached turbulent flow can be fitted with an appropriate wall function, whereas the other part containing separated flows may be fitted with a generic functional such as a polynomial.

B

Tutorial on LIVA-console to process LPT results

Research question III focused on the implementation of the proposed method into the current state-of-the-art. The digital images gathered during an LPT experiment are typically processed by specialised software such as DaVis ([LaVision, 2023]). As mentioned in section 6.2 a console has been developed with tools to inspect the data, define the algorithm settings and launch the processing. The user can inspect the geometry through a three-dimensional viewer. The LIVA process settings can be defined after which the user can process the data step-by-step. This appendix contains a tutorial that explains how the case of the surface-mounted cube was processed from start to finish. A Github project has been created which contains the necessary Python source files, test data files, tools and environment information to follow along with the tutorial presented here. This project can be accessed via the following URL:

`https://github.com/Erikd1997/LIVA_processor.git`

There are two datasets available in this Github project in the *data* folder. First is the surface-mounted cube case with bin-based data. Second is the WBJ case with tracer-based data. The tutorial in this appendix will walk through the surface-mounted cube case, but the same steps are applied to the case of the WBJ. To access the *.DAT* files which hold the fluid-mesh information, a Google Drive folder is available. The Github projects do no allow for (multiple) large files to be shared on their platform. A few files are available, but not all. The Google Drive folder does contain all required data files[1] The folder can be accessed via the following URL:

`https://drive.google.com/drive/folders/1Db2HCmJ40mIjJbutxNf6YhfdDiN044Po?usp=drive_link`

The processing can be divided into three steps, where first all necessary data files are obtained such as the object STL, the fluid measurement data and the surface output-mesh. This is detailed for generating the output-mesh in section B.1. After, one can start the console to load in all data and process the results to determine the skin friction distribution, which is described in section B.2. Once the algorithm has completed, the results can be opened in a data visualisationtool such as Tecplot 360 or ParaView.

B.1. Generating the output-mesh

Next to the fluid-mesh, object-mesh and the transformation matrix ${}_{\mathcal{M}}\mathcal{T}_{\mathcal{F}}$, an output-mesh must be generated. In this tutorial, the mesh is generated using *Gmsh*, however, any other software can be

---

[1] Note that these will likely not be available for multiple years as I personally pay for data storage and will not keep these files forever.

used which can produce a *.VTK* file. The Gmsh software is available in the Github project or can be downloaded from the website, `https://gmsh.info/`. It is also possible to skip this step and use the output-mesh that is included in the Github project at *./data/cube/geometry/cube-outputmesh-7mm_spacing.vtk*. A second, more refined outputmesh is also available.

In this case, we generate an output-mesh that is aligned with the surface of the cube. Firstly, however, we convert the STL file to a different file format which is better supported by Gmsh. Otherwise, we Gmsh would need to parametrise the *.STL* data. Open the .STL file into a CAD-modelling software, such as Fusion 360. If the output-mesh must include a ground plane, this should be drafted and combined with the outer surface of the cube. At the end, the outer surface of the cube must remain and be exported as a *.STEP* file.

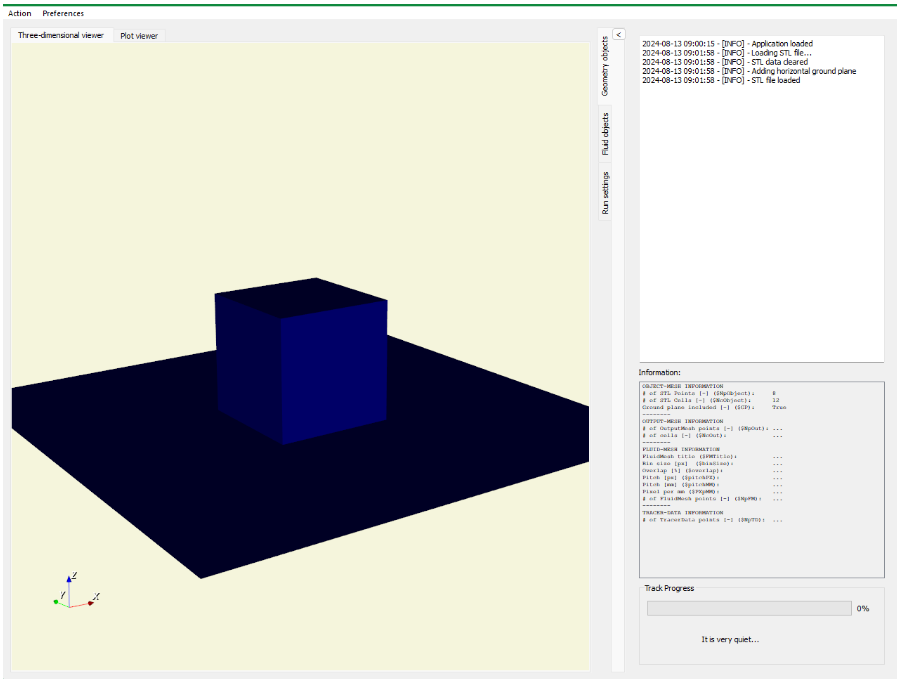
The *.STEP* file is then loaded into the Gmsh software using **File** > **Merge**. Next, we open the 'Options' menu to adjust the maximum cell size. Navigate to **Tools** > **Options** and in the new window that opens, navigate to **Mesh**. Under the tab **General**, the minimum and maximum element sizes can be selected. In this case, we set the maximum to 7 and leave the minimum at 1.

The 'Options' window can be closed again and it is time to generate the mesh. Unfold 'Menu' in the tree on the left and press '2D'. This automatically generates a surface-mesh using the prescribed algorithm (default is Frontal-Delaunay). Now, the mesh should be exported so navigate to **File** > **Export** which prompts a file dialog where the name of the meshfile can be selected. Ensure that the extension *.VTK* is added to the filename, such that Gmsh knows that the file must be written in the VTK format. Once the mesh is exported (eitherin ASCII or BINARY), it is time to proceed to the LIVA-console.

B.2. Processing the surface-mounted cube with the LIVA-console

After cloning the source code, one can run the main Python script *applicationMain.py*. This launches the LIVA-console as depicted in figure 6.6. Choose the *.STL* file *./data/cube/geometry/cube_12cm.stl*. Add a ground plane at $z = 0\,\mathrm{mm}$ to that and click 'Load STL file...'. Next, choose the *.VTK* file that stores the output-mesh, discussed in section B.1. Load this mesh by clicking on 'Load OutputMesh VTK file. At this point the cube is shown in the 3D model viewer with the output-mesh, as seen in figure B.1.

Tip: Click on the active tab (it should be 'Geometry objects' at this point) to minimise this pane and enlarge the 3D model viewer. Click on the '<' symbol to restore the view again.



```
2024-08-13 09:00:15 - [INFO] - Application loaded
2024-08-13 09:01:58 - [INFO] - Loading STL file...
2024-08-13 09:01:58 - [INFO] - STL data cleared
2024-08-13 09:01:58 - [INFO] - Adding horizontal ground plane
2024-08-13 09:01:58 - [INFO] - STL file loaded
```

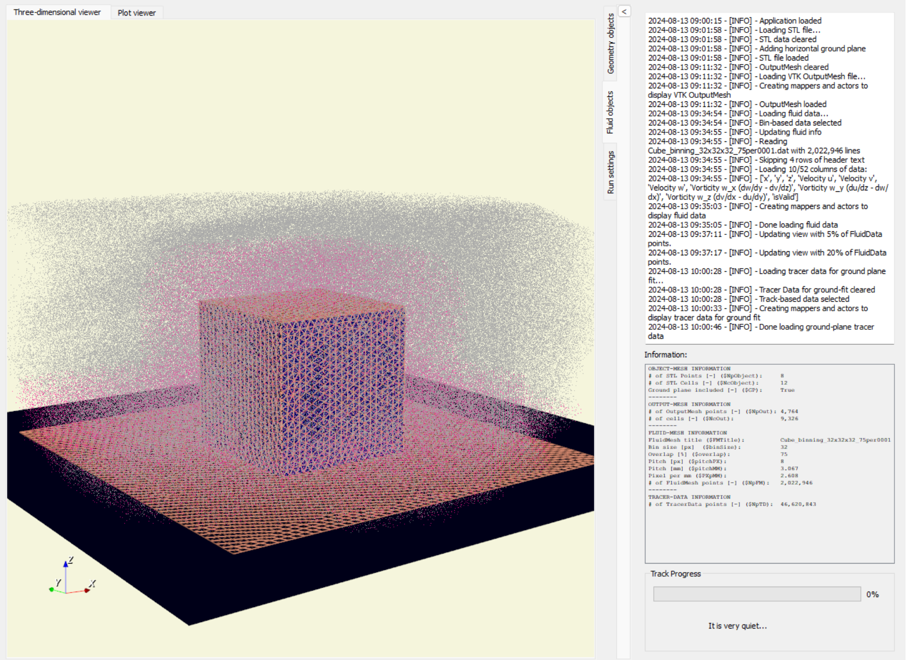
Information:

```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):     8
# of STL Cells [-] ($NcObject):      12
Ground plane included [-] ($GP):     True
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): ...
# of cells [-] ($NcOut):             ...
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):          ...
Bin size [px]  ($binSize):           ...
Overlap [%] ($overlap):              ...
Pitch [px] ($pitchPX):               ...
Pitch [mm] ($pitchMM):               ...
Pixel per mm ($PXpMM):               ...
# of FluidMesh points [-] ($NpFM):   ...
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  ...
```

Figure B.1: View of the LIVA-console with surface-mounted cube object-mesh and output-mesh loaded.

Enlarge the settings pane again and navigate to the next tab 'Fluid objects'. In this section, we will load the fluid-mesh information and tracer positions. It is possible to either load the bin-based data or the tracer-based data. For the surface-mounted cube tutorial here, we will work with the available bin-based data. Choose the data file *./data/cube/fluid data/Cube_binning_32x32x32_75per0001.dat*. Ensure that the ”Bin” type of data is selected and click ’Load FluidData file...’. After loading the data, the points are displayed in the viewer. Less points can be modified by dragging the slider bar to the left, blanking points which fall inside the model or ground plane, including only valid points and cropping data. Update the view by clicking ’Update FluidData view’. Here, we will set the slider bar to 20%, and check all the boxes below it. You should see that the cube is now visible again behind the fluid-mesh. The cropping dimensions are taken from the tab ’Run settings’.

Next, we will add the fluid tracer data which is available through the Google Drive folder link shared above. Open the data file *./data/cube/fluid data/Cube_tracer_fluid_data.npy* and load the data by clicking ’Load TracerData file for Ground-Fit...’. After the data is loaded, the view should look as shown in figure B.2. Note how the metadata has also been included in the ’Information’ table displayed below the logging widget.



```
2024-08-13 09:00:15 - [INFO] - Application loaded
2024-08-13 09:01:58 - [INFO] - Loading STL file...
2024-08-13 09:01:58 - [INFO] - STL data cleared
2024-08-13 09:01:58 - [INFO] - Adding horizontal ground plane
2024-08-13 09:01:58 - [INFO] - STL file loaded
2024-08-13 09:11:32 - [INFO] - OutputMesh cleared
2024-08-13 09:11:32 - [INFO] - Loading VTK OutputMesh file...
2024-08-13 09:11:32 - [INFO] - Creating mappers and actors to
display VTK OutputMesh
2024-08-13 09:11:32 - [INFO] - OutputMesh loaded
2024-08-13 09:34:54 - [INFO] - Loading fluid data...
2024-08-13 09:34:54 - [INFO] - Bin-based data selected
2024-08-13 09:34:55 - [INFO] - Updating fluid info
2024-08-13 09:34:55 - [INFO] - Reading
Cube_binning_32x32x32_75per0001.dat with 2,022,946 lines
2024-08-13 09:34:55 - [INFO] - Skipping 4 rows of header text
2024-08-13 09:34:55 - [INFO] - Loading 10/52 columns of data:
2024-08-13 09:34:55 - [INFO] - ['x', 'y', 'z', 'Velocity u', 'Velocity v',
'Velocity w', 'Vorticity w_x (dw/dy - dv/dz)', 'Vorticity w_y (du/dz - dw/
dx)', 'Vorticity w_z (dv/dx - du/dy)', 'isValid']
2024-08-13 09:35:03 - [INFO] - Creating mappers and actors to
display fluid data
2024-08-13 09:35:05 - [INFO] - Done loading fluid data
2024-08-13 09:37:11 - [INFO] - Updating view with 5% of FluidData
points.
2024-08-13 09:37:17 - [INFO] - Updating view with 20% of FluidData
points.
2024-08-13 10:00:28 - [INFO] - Loading tracer data for ground plane
fit...
2024-08-13 10:00:28 - [INFO] - Tracer Data for ground-fit cleared
2024-08-13 10:00:28 - [INFO] - Track-based data selected
2024-08-13 10:00:33 - [INFO] - Creating mappers and actors to
display tracer data for ground fit
2024-08-13 10:00:46 - [INFO] - Done loading ground-plane tracer
data
```

```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):     8
# of STL Cells [-] ($NcObject):      12
Ground plane included [-] ($GP):     True
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): 4,764
# of cells [-] ($NcOut):             9,326
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):          Cube_binning_32x32x32_75per0001
Bin size [px]  ($binSize):           32
Overlap [%] ($overlap):              75
Pitch [px] ($pitchPX):               8
Pitch [mm] ($pitchMM):               3.067
Pixel per mm ($PXpMM):               2.608
# of FluidMesh points [-] ($NpFM):   2,022,946
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  46,620,843
```

Figure B.2: View of the LIVA-console with surface-mounted cube fluid-mesh and tracer data loaded.

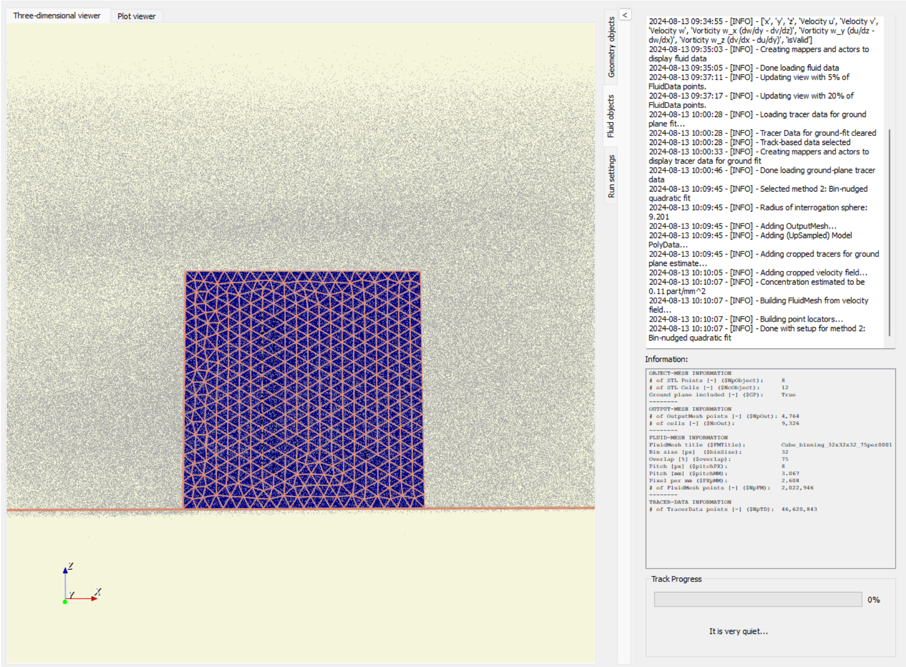
Next, we will define the settings for the interrogation volume and run the setup. Navigate to the ’Run settings’ tab. Choose the *Bin-Nudge-QuadFit* algorithm. Then, set as interrogation sphere radius *3 * $pitchMM*, this should amount to $9.201\ \mathrm{mm}$. Any of the other loaded variables shown in the ’Information’ pane may also be used. Their variable names are preposed by the $-sign. We will not work with semi-spheres here and we leave the cropping dimensions the same as the default. Next click the button ’Start Setup’. This triggers the console to initialise the data and compute all necessary information to iterate over each cell. Note that this setup may take about 20 seconds.

After the setup is completed, you should notice that the widgets below have been enabled. Next, we will execute the object registration correction for the ground plane. First, we will show the discrepancy by examining the tracer positions. Navigate to the ’Fluid objects’ tab and unselect the ’Show Fluid data’ checkbox. Now, only the grey tracer data should be visible. Align the camera such that you are looking at the side of the cube facing in positive $y-$direction.

Tip: you can hold the CTRL button to rotate the view along the axis perpendicular to the screen.

The view should look similar to figure B.3. Note, how grey tracers appear below the ground plane in front of the cube (towards the left, in negative $x-$direction). We will be resolving this issue by computing a new ground plane estimate.

Enlarge the settings pane and navigate back to the ’Run settings’ tab. Select the ’Show corrected ground plane fit in 3D view’ checkbox and choose a ground-plane fit folder. This folder will store the



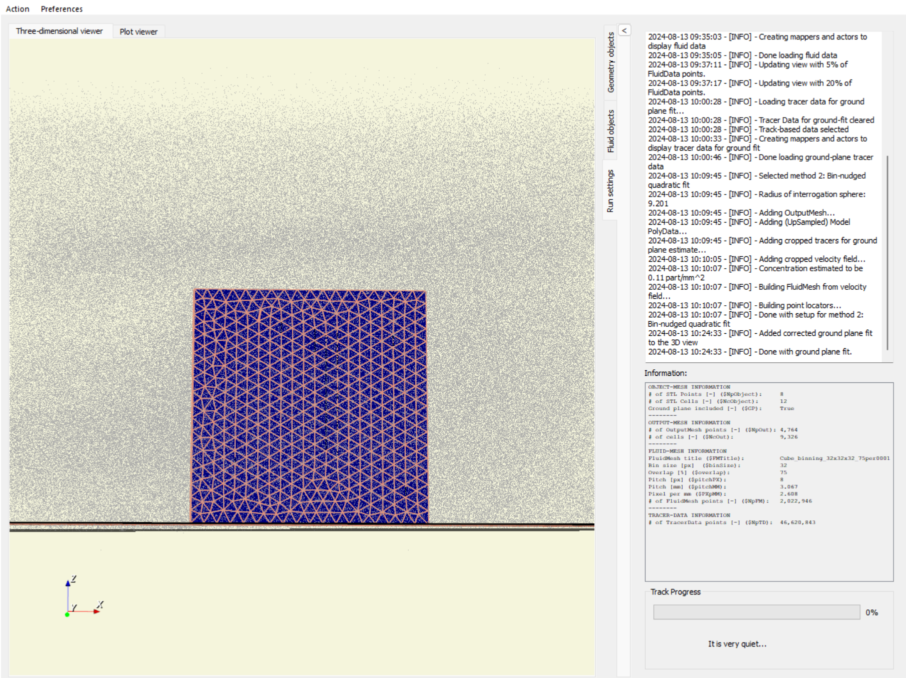
```
2024-08-13 09:34:55 - [INFO] - ['x', 'y', 'z', 'Velocity u', 'Velocity v',
'Velocity w', 'Vorticity w_x (dw/dy - dv/dz)', 'Vorticity w_y (du/dz -
dw/dx)', 'Vorticity w_z (dv/dx - du/dy)', 'isValid']
2024-08-13 09:35:03 - [INFO] - Creating mappers and actors to
display fluid data
2024-08-13 09:35:05 - [INFO] - Done loading fluid data
2024-08-13 09:37:11 - [INFO] - Updating view with 5% of
FluidData points.
2024-08-13 09:37:17 - [INFO] - Updating view with 20% of
FluidData points.
2024-08-13 10:00:28 - [INFO] - Loading tracer data for ground
plane fit...
2024-08-13 10:00:28 - [INFO] - Tracer Data for ground-fit cleared
2024-08-13 10:00:28 - [INFO] - Track-based data selected
2024-08-13 10:00:33 - [INFO] - Creating mappers and actors to
display tracer data for ground fit
2024-08-13 10:00:46 - [INFO] - Done loading ground-plane tracer
data
2024-08-13 10:09:45 - [INFO] - Selected method 2: Bin-nudged
quadratic fit
2024-08-13 10:09:45 - [INFO] - Radius of interrogation sphere:
9.201
2024-08-13 10:09:45 - [INFO] - Adding OutputMesh...
2024-08-13 10:09:45 - [INFO] - Adding (UpSampled) Model
PolyData...
2024-08-13 10:09:45 - [INFO] - Adding cropped tracers for ground
plane estimate...
2024-08-13 10:10:05 - [INFO] - Adding cropped velocity field...
2024-08-13 10:10:07 - [INFO] - Concentration estimated to be
0.11 part/mm^2
2024-08-13 10:10:07 - [INFO] - Building FluidMesh from velocity
field...
2024-08-13 10:10:07 - [INFO] - Building point locators...
2024-08-13 10:10:07 - [INFO] - Done with setup for method 2:
Bin-nudged quadratic fit
```

Information:

```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):     8
# of STL Cells [-] ($NcObject):      12
Ground plane included [-] ($GP):     True
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): 4,764
# of cells [-] ($NcOut):             9,326
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):          Cube_binning_32x32x32_75per0001
Bin size [px]  ($binSize):           32
Overlap [%] ($overlap):              75
Pitch [px] ($pitchPX):               8
Pitch [mm] ($pitchMM):               3.067
Pixel per mm ($PXpMM):               2.608
# of FluidMesh points [-] ($NpFM):   2,022,946
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  46,620,843
```

Figure B.3: View of the side of the LIVA-console with only the tracer data points. This shows the error in the ground plane estimate.

results of the ground-plane estimate for this specific tracer dataset and output-mesh and can be loaded in the next time. Then, click the button 'Execute ground plane fit'. Once the correction is computed, the view automatically updates and show the new ground plane. The contour of the previous estimate is still included to see the difference. If we position the camera in the same way as before, such as in figure B.4, we can see that the ground plane is now fit to the underside of the tracer data.



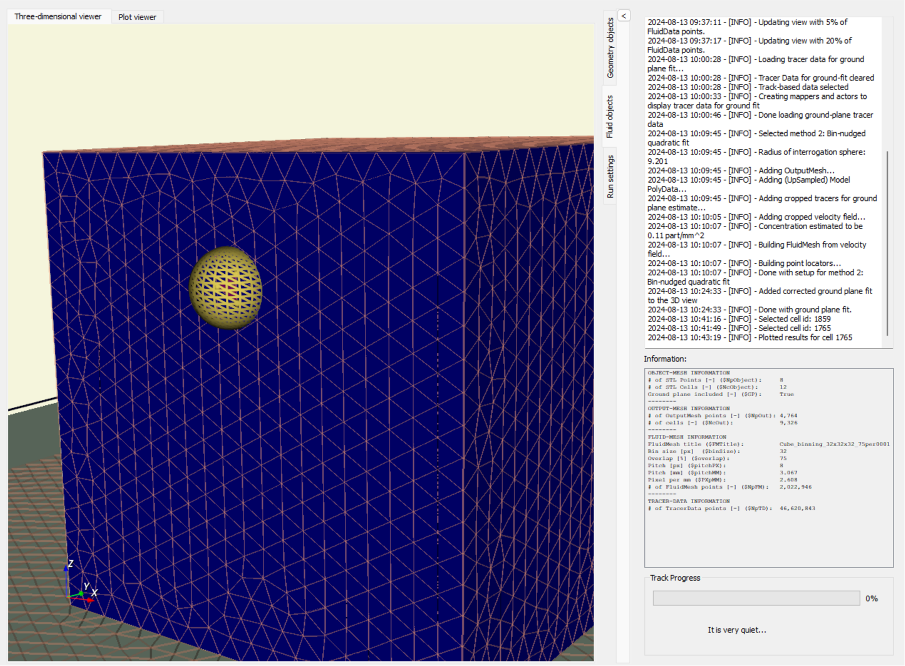
```
2024-08-13 09:35:03 - [INFO] - Creating mappers and actors to
display fluid data
2024-08-13 09:35:05 - [INFO] - Done loading fluid data
2024-08-13 09:37:11 - [INFO] - Updating view with 5% of
FluidData points.
2024-08-13 09:37:17 - [INFO] - Updating view with 20% of
FluidData points.
2024-08-13 10:00:28 - [INFO] - Loading tracer data for ground
plane fit...
2024-08-13 10:00:28 - [INFO] - Tracer Data for ground-fit cleared
2024-08-13 10:00:28 - [INFO] - Track-based data selected
2024-08-13 10:00:33 - [INFO] - Creating mappers and actors to
display tracer data for ground fit
2024-08-13 10:00:46 - [INFO] - Done loading ground-plane tracer
data
2024-08-13 10:09:45 - [INFO] - Selected method 2: Bin-nudged
quadratic fit
2024-08-13 10:09:45 - [INFO] - Radius of interrogation sphere:
9.201
2024-08-13 10:09:45 - [INFO] - Adding OutputMesh...
2024-08-13 10:09:45 - [INFO] - Adding (UpSampled) Model
PolyData...
2024-08-13 10:09:45 - [INFO] - Adding cropped tracers for ground
plane estimate...
2024-08-13 10:10:05 - [INFO] - Adding cropped velocity field...
2024-08-13 10:10:07 - [INFO] - Concentration estimated to be
0.11 part/mm^2
2024-08-13 10:10:07 - [INFO] - Building FluidMesh from velocity
field...
2024-08-13 10:10:07 - [INFO] - Building point locators...
2024-08-13 10:10:07 - [INFO] - Done with setup for method 2:
Bin-nudged quadratic fit
2024-08-13 10:24:33 - [INFO] - Added corrected ground plane fit
to the 3D view
2024-08-13 10:24:33 - [INFO] - Done with ground plane fit.
```

Information:

```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):     8
# of STL Cells [-] ($NcObject):      12
Ground plane included [-] ($GP):     True
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): 4,764
# of cells [-] ($NcOut):             9,326
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):          Cube_binning_32x32x32_75per0001
Bin size [px]  ($binSize):           32
Overlap [%] ($overlap):              75
Pitch [px] ($pitchPX):               8
Pitch [mm] ($pitchMM):               3.067
Pixel per mm ($PXpMM):               2.608
# of FluidMesh points [-] ($NpFM):   2,022,946
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  46,620,843
```

Figure B.4: View of the side of the LIVA-console after the ground-plane correction is computed. The corrected plane can be seen to fit the underside of the tracer data.

Next, we will execute the LIVA-algorithm for a single cell and inspect the results. First, we will select a cell of the output-mesh, for example on the front face of the cube. Hold CTRL+SHIFT before clicking in the 3D model viewer to select a cell. To follow along with this tutorial select the cell with ID 1765. It is also possible to choose a different cell in the output-mesh. Once selected, the cell is highlighted in the viewer. Navigate to the 'Run settings' tab once again. Press the button 'Evaluate single iterate'. The interrogation volume is now shown in the 3D model viewer, as can be seen in figure B.5. Select the 'Plot viewer' tab in the top left, and a new view appears. Here, the results of the LIVA-binning is shown in the main plot. On the right, the LIVA-binning is shown in three side views. The results of the velocity reconstruction are shown in the bottom row, plotted against the wall-normal coordinate



```
2024-08-13 09:37:11 - [INFO] - Updating view with 5% of
FluidData points.
2024-08-13 09:37:17 - [INFO] - Updating view with 20% of
FluidData points.
2024-08-13 10:00:28 - [INFO] - Loading tracer data for ground
plane fit...
2024-08-13 10:00:28 - [INFO] - Tracer Data for ground-fit cleared
2024-08-13 10:00:28 - [INFO] - Track-based data selected
2024-08-13 10:00:33 - [INFO] - Creating mappers and actors to
display tracer data for ground fit
2024-08-13 10:00:46 - [INFO] - Done loading ground-plane tracer
data
2024-08-13 10:09:45 - [INFO] - Selected method 2: Bin-nudged
quadratic fit
2024-08-13 10:09:45 - [INFO] - Radius of interrogation sphere:
9.201
2024-08-13 10:09:45 - [INFO] - Adding OutputMesh...
2024-08-13 10:09:45 - [INFO] - Adding (UpSampled) Model
PolyData...
2024-08-13 10:09:45 - [INFO] - Adding cropped tracers for ground
plane estimate...
2024-08-13 10:10:05 - [INFO] - Adding cropped velocity field...
2024-08-13 10:10:07 - [INFO] - Concentration estimated to be
0.11 part/mm^2
2024-08-13 10:10:07 - [INFO] - Building FluidMesh from velocity
field...
2024-08-13 10:10:07 - [INFO] - Building point locators...
2024-08-13 10:10:07 - [INFO] - Done with setup for method 2:
Bin-nudged quadratic fit
2024-08-13 10:24:33 - [INFO] - Added corrected ground plane fit
to the 3D view
2024-08-13 10:24:33 - [INFO] - Done with ground plane fit.
2024-08-13 10:41:16 - [INFO] - Selected cell id: 1859
2024-08-13 10:41:49 - [INFO] - Selected cell id: 1765
2024-08-13 10:43:19 - [INFO] - Plotted results for cell 1765
```

Information:

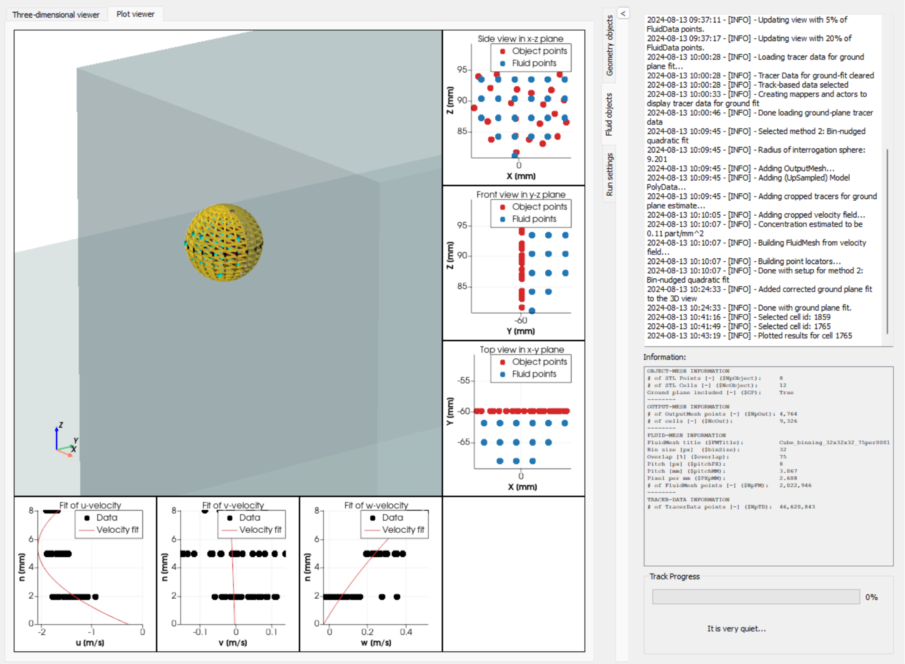
```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):     8
# of STL Cells [-] ($NcObject):      12
Ground plane included [-] ($GP):     True
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): 4,764
# of cells [-] ($NcOut):             9,326
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):          Cube_binning_32x32x32_75per0001
Bin size [px]  ($binSize):           32
Overlap [%] ($overlap):              75
Pitch [px] ($pitchPX):               8
Pitch [mm] ($pitchMM):               3.067
Pixel per mm ($PXpMM):               2.608
# of FluidMesh points [-] ($NpFM):   2,022,946
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  46,620,843
```

Track Progress

0%

It is very quiet...

Figure B.5: View of the LIVA-console after a single cell has been processed. The interrogation volume is displayed in yellow.



```
2024-08-13 09:37:11 - [INFO] - Updating view with 5% of
FluidData points.
2024-08-13 09:37:17 - [INFO] - Updating view with 20% of
FluidData points.
2024-08-13 10:00:28 - [INFO] - Loading tracer data for ground
plane fit...
2024-08-13 10:00:28 - [INFO] - Tracer Data for ground-fit cleared
2024-08-13 10:00:28 - [INFO] - Track-based data selected
2024-08-13 10:00:33 - [INFO] - Creating mappers and actors to
display tracer data for ground fit
2024-08-13 10:00:46 - [INFO] - Done loading ground-plane tracer
data
2024-08-13 10:09:45 - [INFO] - Selected method 2: Bin-nudged
quadratic fit
2024-08-13 10:09:45 - [INFO] - Radius of interrogation sphere:
9.201
2024-08-13 10:09:45 - [INFO] - Adding OutputMesh...
2024-08-13 10:09:45 - [INFO] - Adding (UpSampled) Model
PolyData...
2024-08-13 10:09:45 - [INFO] - Adding cropped tracers for ground
plane estimate...
2024-08-13 10:10:05 - [INFO] - Adding cropped velocity field...
2024-08-13 10:10:07 - [INFO] - Concentration estimated to be
0.11 part/mm^2
2024-08-13 10:10:07 - [INFO] - Building FluidMesh from velocity
field...
2024-08-13 10:10:07 - [INFO] - Building point locators...
2024-08-13 10:10:07 - [INFO] - Done with setup for method 2:
Bin-nudged quadratic fit
2024-08-13 10:24:33 - [INFO] - Added corrected ground plane fit
to the 3D view
2024-08-13 10:24:33 - [INFO] - Done with ground plane fit.
2024-08-13 10:41:16 - [INFO] - Selected cell id: 1859
2024-08-13 10:41:49 - [INFO] - Selected cell id: 1765
2024-08-13 10:43:19 - [INFO] - Plotted results for cell 1765
```

Information:

```
OBJECT-MESH INFORMATION
# of STL Points [-] ($NpObject):     8
# of STL Cells [-] ($NcObject):      12
Ground plane included [-] ($GP):     True
--------
OUTPUT-MESH INFORMATION
# of OutputMesh points [-] ($NpOut): 4,764
# of cells [-] ($NcOut):             9,326
--------
FLUID-MESH INFORMATION
FluidMesh title ($FMTitle):          Cube_binning_32x32x32_75per0001
Bin size [px]  ($binSize):           32
Overlap [%] ($overlap):              75
Pitch [px] ($pitchPX):               8
Pitch [mm] ($pitchMM):               3.067
Pixel per mm ($PXpMM):               2.608
# of FluidMesh points [-] ($NpFM):   2,022,946
--------
TRACER-DATA INFORMATION
# of TracerData points [-] ($NpTD):  46,620,843
```

Track Progress

0%

It is very quiet...

Figure B.6: View of the LIVA-console with the 'Plot viewer' after executing the LIVA-algorithm for a cell 1765.

n. Note that the velocities u, v and w correspond to the global coordinates however. The results for cell ID 1765 are displayed in figure B.6. In the case shown here, there is a shear towards the global $w-$direction, which translates to flow moving upwards from the center of the cube's front face. Note that that $u-$component is perpendicular to the cell surface and, as explained in subsection 5.3.5, this is not included in the (sall-tangential) shear stress.

That leaves one last action, namely to execute the LIVA-algorithm for the entirety of the output-mesh. Navigate once again to the 'Run settings' tab and select a folder where the results are to be saved. Once chosen, press the 'START' button to execute the algorithm. Once the loading bar has turned to 100% the results are saved to a *.VTU* file. This file can be imported into a fluid visualisation software such as Tecplot or ParaView. That concludes the tutorial on the LIVA-console.

C

Data representation of object-mesh, fluid-mesh and output-mesh

This appendix contains additional information on the representation formats of the various meshes and mesh-files used throughout this report. It may assist any reader who will work with the LIVA-console and thereby reading/writing such files.

C.1. The object-mesh file format - .STL file

The representation of an object's surface in CFD is often presented via a CAD model, so much so that common CFD codes either offer an interface for creating CAD models and/or facilitate the import of such models ([Tu et al., 2018]). The CAD model aims to geometrically represent a digitised version of a physical object via a collection of geometrical elements meshed together and is typically used in designing products. For geometrical purposes, an object need only be defined by its bounding surface, defined as the surface domain Γ_{CAD}. The domain Γ_{CAD} is made up of several smaller surface elements which are connected along their edges and at their vertices. The digitisation of a cyclist is displayed in figure C.1, showing the render of a cyclist CAD-model in figure C.1a and the CAD model in C.1b made up of triangle surface elements. The triangle surface is obtained via a process called tessellation and is a common approach to store CAD model information, for example via the

(a)

(b)

Figure C.1: Digitisation of a cyclist model with (a) showing a rendered version and (b) a meshed version consisting of triangle surface elements.

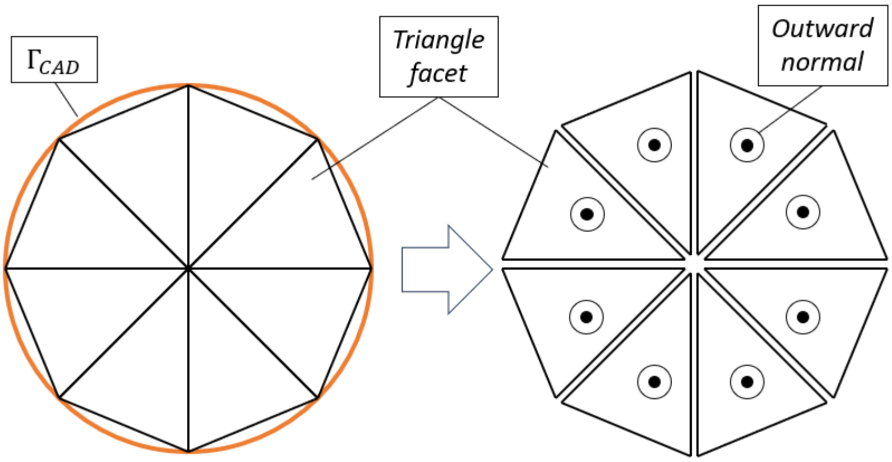


Figure C.2: The STL format representing a tessellated circular surface with eight triangle facets with a normal pointing out of the paper.

STereoLitographic (STL) format ([Kai et al., 1997]). The STL format represents triangular facets with a set of x-, y- and z-coordinates for the three vertices and a normal vector indicating which side faces outward, schematically depicted in figure C.2. Note how each facet has its own three vertices leading to redundancy and complicating the processing of STL formats to reconstruct coinciding points and edges ([Kai et al., 1997]). CAD model information in this report will be communicated via the STL format, though other data formats such as IGES, OBJ and STEP are possible ([McHenry and Bajcsy, 2008]).

C.2. The fluid-mesh representation and file format - .DAT file

The fluid data, consisting of coordinates, velocities and possibly many more quantities such as acceleration, vorticity, et cetera, has its own data structure for representing the fluid state in the domain $\Omega_{\mathcal{F}}$. Results from Lagrangian particle tracking experiment are used here as an example, though this can be transformed into a representation akin to particle image velocimetry via ensemble binning. The LPT characterisation is discussed in subsection C.2.1 and is referred to as tracer-based data. Subsection C.2.2 outlines the structure of ensemble binned tracer data, referred to as bin-based data.

C.2.1. Tracer-based representation

The tracer-based representation of LPT data consists of a collection of tracers at a single time instance k with at minimum a position and velocity. Four reconstructed particle trajectories are plotted in figure C.3 per example. Each trajectory is given a track ID through which the sampled tracers along a path are linked. This form is not directly useful to experimentalists or analysts which typically visualise and analyse fluids from an Eulerian perspective. In an Eulerian perspective the fluid is visualised by the observer as it flows through a fixed region in space. On the other hand, in a Lagrangian reference frame, a small fluid element is visualised by an observer which moves with the element, thereby tracing out a path in space.

The tracer-based format in which LPT results are presented can be considered as ”meshless”, yet the position is given in any coordinate system, such as a Cartesian (figure C.3), spherical and cylindrical. This is slightly different from the ”true” Lagrangian perspective where the frame of reference follows the object (like a passenger on a boat floating along the river). The change of any quantity ϕ along a single trajectory is given by the material derivative (or total derivative) ([Batchelor, 2000]):

$$\frac{\mathrm{D}\phi}{\mathrm{D}t} = \frac{\partial \phi}{\partial t} + (\nabla \cdot \mathbf{u})\phi, \tag{C.1}$$

where the left-hand side denotes the material derivative of ϕ and the right-hand side the local temporal and local convective derivatives of the quantity ϕ. The local temporal and convective derivatives are

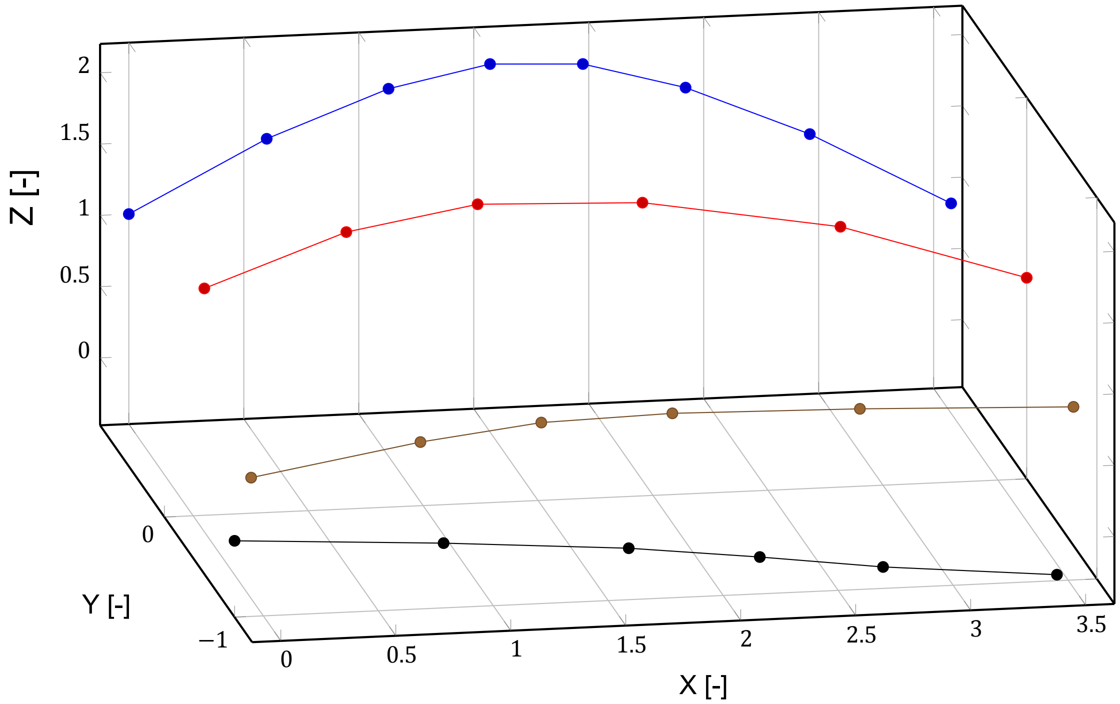


Figure C.3: Example of four particles which are tracked over eight consecutive time steps.

obtained from an Eulerian perspective, whereas the material derivative is obtained from the Lagrangian perspective.

To write and read tracer-based fluid data, the information can be stored to *.DAT* files in ASCII or binary format. Such files consist of header data and segments or snapshots with the fluid information representing the time frames. An example of such a file is given in Listing C.1. Each zone is given a time instance as `Snapshot XXXX`. The collection of datasets (or set of zones) that represents the same region of the solution over time is referred to as a strand and these are marked with an integer-valued `STRANDID`. The `SOLUTIONTIME` marks a specific time in the strand. Next, the particles tracked in each snapshot are `I`-ordered, hence `J=1` and `K=1`. The total number of tracers tracked at the given `SOLUTIONTIME` is equal to `I`. Tracer-based data files are always denoted by `ZONETYPE=Ordered`, since the information is ordered. Other zone types are reserved for finite-element meshes. The infor-

Listing C.1: *.DAT* file example for tracer-based fluid data in ASCII format

```
TITLE = ”Cube_cropped_tracks(z100-200) Polynomial order: 2 Length: 5”
VARIABLES = ”x [mm]” ”y [mm]” ”z [mm]” ”u [m/s]” ”v [m/s]” ”w [m/s]” ”trackID” ...
ZONE T=”Snapshot 0000”
STRANDID=1, SOLUTIONTIME=2.059831199
I=901, J=1, K=1, ZONETYPE = Ordered
DATAPACKING = POINT
-81.4127 -38.8966 102.193 4.32118 -2.60797 1.7744 0
4.83548 3.16288 158.609 6.8023 0.20414 3.48972 1
.
ZONE T=”Snapshot 0001”
STRANDID=1, SOLUTIONTIME=2.060164538
I=952, J=1, K=1, ZONETYPE = Ordered
DATAPACKING = POINT
-79.8447 -39.8276 103.01 5.0869 -2.97778 3.12764 0
6.76605 3.10175 159.441 4.78093 -0.570945 1.50237 1
.
.
```

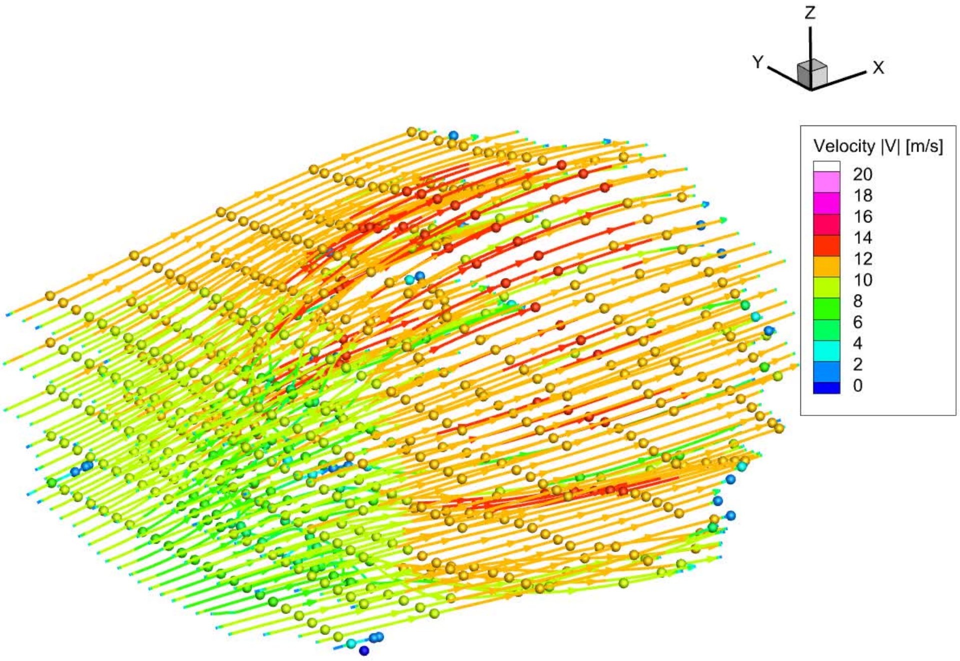


Figure C.4: The movie can be controlled with the buttons below it.
NOTE: Open the document in a PDF-viewer such as Adobe Acrobat Reader DC to play the movie.

mation may be written in two orders, determined by `DATAPACKING`. First is the `POINT` format, where the values of all variables are written for the first point, then the second point and so forth. The other option is the `BLOCK` format, where the values of all points are written for the first variable in a block, then for the second variable, then for the third and so forth. (Tecplot Inc., 2006])

The combination of all particle trajectories could be used to visualise a flow field. As an example, figure C.4, created with TECPLOT 360^{TM}, portrays a fluid field around a surface-mounted cube visualised by particles moving along streamlines. The stream-tracers are not constrained to a mesh, so the trajectories curve and bend as they are convected with the fluid. The Lagrangian perspective with information along trajectories was employed to compute two-point statistics at close distances, beyond the typical limitation of particle imaging ([Godbersen and Schröder, 2020]). Lagrangian descriptions have also been used to extract vortical structures, even with highly sparse data ([Martins et al., 2021]). To obtain one-, two- or multipoint statistics and convert the flow field to an Eulerian description, the tracers can be binned in small interrogation volumes.

C.2.2. Bin-based representation

The output of an LPT experiment provides a set of particle tracers with at minimum a position, velocity and time (or frame number). To collect all tracers in a time-averaged approach and visualise the results

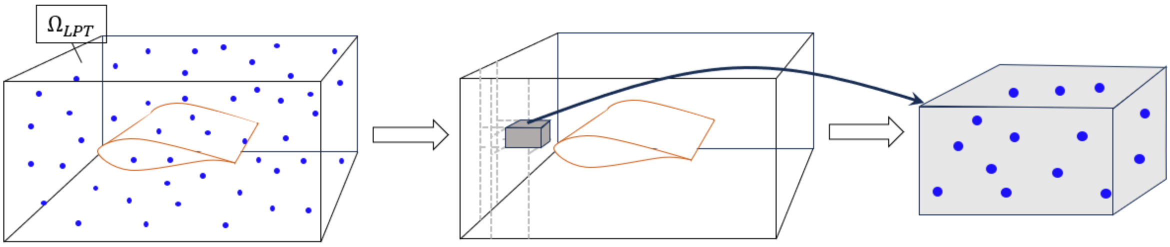


Figure C.5: Schematic description of ensemble binning.

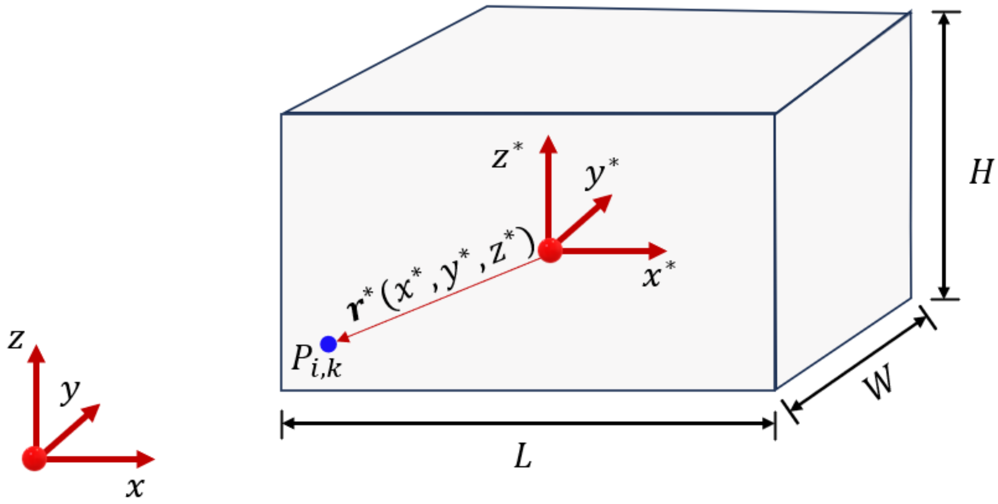


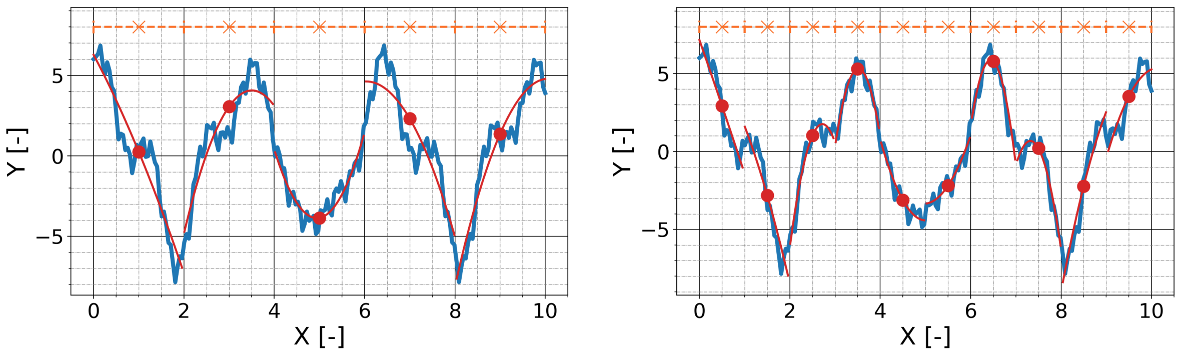
Figure C.6: Local coordinate system inside an interrogation bin.

in an Eulerian reference frame –referred to as Cartesian Grid Reduction (CGR, [Scarano et al., 2022b])–, ensembles are collected in bins and averaged, hence the term ensemble binning. Consider the LPT measurement volume Ω_{LPT} sketched on the left in figure C.5. A wing is placed inside the volume which contains a set of tracer particles N_k at a time k. Ω_{LPT} is subdivided into J smaller bins B, which fill the complete volume. A tracer $N_{i,k}$ at a time k is assigned to a bin B_j via the integer truncated division ([Godbersen and Schröder, 2020])

$$\text{bin}_{\text{index}} = \left\lfloor \frac{x_{N_{i,k}} - x_0}{\Delta x} \right\rfloor, \tag{C.2}$$

where $x_{N_{i,k}}$ denotes the particle position, x_0 the corner of the volume Ω_{LPT} and Δx is the bin size. This process is repeated for all tracer particles N_p over all time steps k, such that an ensemble of tracers is obtained as shown on the right in figure C.5. The ensemble can be used to estimate the mean velocity or obtain bin statistics, such as velocity fluctuations and Reynolds stresses. The tracers can be combined via a Gaussian weighting approach ([Agüí and Jiménez, 1987]) or using a polynomial fit over the volume ([Agüera et al., 2016]). A recent publication also proposed to combine the full particle track information for the velocity and bin statistics ([Godbersen and Schröder, 2020]). In this report, the ensemble is combined by using a polynomial fit of the velocity based on the tracer's positions, which is implemented into the LPT post-processing software DaVis ([LaVision, 2023]).

The bin has a length L, width W and height H. A local coordinate system is defined with its origin at the centre of the bin and orientation aligned with the global Cartesian coordinate system (x, y, z). The position in this coordinate system is defined by the local coordinates (x^*, y^*, z^*). This is shown



(a)

(b)

Figure C.7: Example of binning applied to a 1D sinusoidal with (a) a bin size or 2 and (b) a bij size of 1.

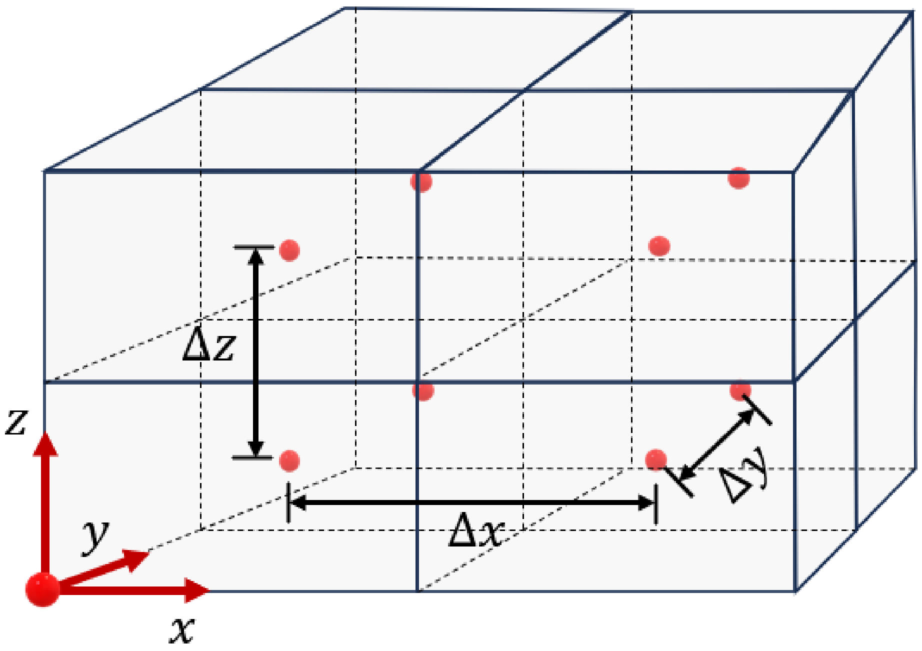


Figure C.8: Mesh created from eight bins where velocity is sampled at the bin center.

schematically in figure C.6, where the position of tracer $P_{i,k}$ is given by the position vector $\mathbf{r}^*(x^*, y^*, z^*)$. The polynomial fit of the velocity in a bin B_j is of second order and contains cross-terms:

$$u_j \approx a_0 + a_1 \cdot x^* + a_2 \cdot y^* + a_3 \cdot z^* + a_4 \cdot (x^*)^2 + a_5 \cdot (y^*)^2 + a_6\,(z^*)^2 + a_7 \cdot x^* y^* + a_8 \cdot y^* z^* + a_9 \cdot x^* z^*, \quad \text{(C.3)}$$

where the coefficients $a_0, \ldots, a_9$ are referred to as design variables. The same velocity functional is applied to the other two velocity components v and w.

The polynomial fit is a powerful approach and can theoretically fit every sufficiently smooth real valued functional when properly employed, as per Taylor's theorem ([Taylor, 1715]). The quadratic fit of equation (C.3) with ten terms approximates the velocity with an error of order $\mathcal{O}\left(||\mathbf{r}^*||^3\right)$. Take for example a one-dimensional functional approximated by piecewise unconnected quadratic splines, displayed in figure C.7. The domain (X) is split into smaller bins, indicated by the orange lines. A second order polynomial is fitted to the blue functional inside each bin indicated by the red piecewise function. It is apparent that the local fits improve when the bin size decreases. Additionally, increasing the order of the fitting function, i.e. including third-order terms and higher, decreases the error of the Taylor's series approximation. Contrarily, this is at odds with the requirements of curve fitting, e.g. with a least-squares fit, to solve equation (C.3) for the design variables, namely a sufficient number of sample points. Increasing the order of the velocity functional introduces more design variables $a_{10}, a_{11}, \ldots$ which requires more sample points and shrinking the bin size decreases the number of tracers registered per bin.

After applying the polynomial fit, the mean velocity is determined by evaluating the functional at the bin centre. The fitted functional represents the time-averaged mean inside the interrogation volume and bin statistics can be obtained with respect to a locally adapted mean. This maps the volume onto a Cartesian grid which is drawn for eight bins in figure C.8 with spacing between the bin centres as respectively Δx, Δy and Δz. The interrogation bins can overlap to increase the grid resolution, while ensuring a sufficient number of tracers are registered inside the bins. The cubic interrogation volumes are created by sampling the measurement volume Ω_{LPT} and do not directly take into account the presence of the model. A situation as sketched in figure 5.1 is, therefore, a common occurrence for curved, non-rectangular objects[1]. The issue of non-conformal interrogation bins is partially addressed

---

[1] Note that this can also occur for rectangular objects when the bins do not align with the object's planar faces

Listing C.2: *.DAT* file example for bin-based fluid data in ASCII format

```
TITLE = ”Cube_binning_32x32x32_75per0001”
VARIABLES = ”x [mm]”, ”y [mm]”, ”z [mm]”, ”U [m/s]”, ”V [m/s]”, ”W [m/s]”, ...
ZONE T=”Frame 0”, I=137, J=138, K=107, F=POINT
STRANDID=1, SOLUTIONTIME=2.893014529
-209.480 195.692 -17.0443 8.68060 -0.118692 0.542636 ...
-206.413 195.692 -17.0443 8.67508 -0.134776 0.583171 ...
.
209.480 195.692 -17.0443 8.66576 -0.160836 0.589231 ...
-209.480 192.625 -17.0443 8.64864 -0.189547 0.584634 ...
-206.413 192.625 -17.0443 8.61377 -0.216365 0.580128 ...
.
.
209.480 -224.438 -17.0443 8.55799 -0.235607 0.582623 ...
-209.480 195.692 -13.9776 8.48781 -0.244959 0.596807 ...
.
.
.
```

with the functional fit, though the presence of the wall introduces a boundary layer with a high wall-normal velocity gradient, which is not accurately reproduced by simply fitting the tracers' velocity. The region near the wall typically sees far fewer bubbles due to a lower entrainment into the boundary layer and the bubble size prevents measurements of the fluid velocity very near to the surface. Introducing information on the position of the wall and thereby imposing a no-slip condition is hypothesised to improve the near-wall velocity reconstruction.

The Cartesian grid of binned velocity can be written to a *.DAT* file, as shown in listing C.2. Similar to the *.DAT* file for tracer-based data, this file consists of header data and snapshots with the fluid information representing time frames. The example of listing C.2 contains a single snapshot with a single `SOLUTIONTIME`, as the CGR of the tracer-based data also averages the results in time. The

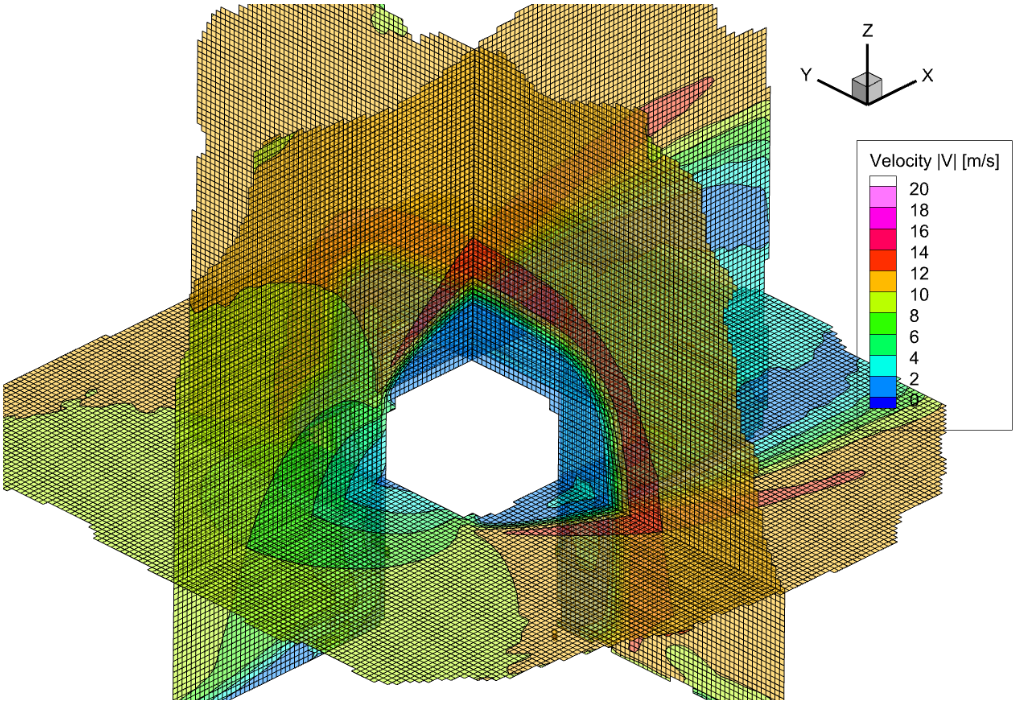


Figure C.9: Example of the three-dimensional flow field around a surface-moutned cube that is obtained from bin-based data. Three contours are plotted, one in each axis, displaying the contours of the velocity magnitude.

file contains a `TITLE` and defines the `VARIABLES`. The mesh that is created from the binning is a IJK-ordered and the indices vary from 1 to *Imax*, *JMax* and *KMax*, indicated by `I`, `J` and `K`, respectively. The variables are stored in the nodes (denoted by the data packing type `POINTS`) and the total number of nodes is given by *IMax* × *JMax* × *KMax*. Lastly, a `STRANDID` is associated to the zone information. Each line holds the mean velocity and statistics information for a single bin along with the position. The mesh connection information is inferred from the structured grid format of the file. The `I`-index varies the fastest, then the `J`-index and lastly the `K` index varies the slowest. These *.DAT* files are written in ASCII or binary format. The final binned, Cartesian-gridded fluid domain is plotted with TECPLOT 360$^{\text{TM}}$ in figure C.9 for the same flow case as figure C.4, showing contours of the velocity with the mesh in all three directions. Apart from the bin statistics, the Eulerian framework is typically used for detecting vortical structures using classic methods such as the Q-, λ_2- and ω- criterions ([Zhan et al., 2019]).

C.3. The output-mesh file format - VTK file format

The output-mesh data is represented by the Visualisation ToolKit (VTK) file format [Schroeder et al., 2006]. The VTK software provides readily available tools for reading and writing mesh information in this format in several computer languages, amongst which is Python. The *.VTK* extension refers to a legacy file format, which is not commonly supported by other software. More recent adaptions employ an XML-based VTK format, where the file extension denotes the data type and file type. For example, the output of the LIVA-console is a *.VTU* file which refers to an unstructured grid.

An example of the contents of a *.VTK* file is shown in listing C.3. The first line denotes the version number and the second line provides a message header. Then, the file format is defined, being either ASCII or BINARY. The dataset information in the VTK file format for unstructured grids pertains to three parts, the coordinates of the points, the connectivity of points to form a cell of any type and the cell type. First the x, y and z-coordinate of all points in the mesh are written in a section POINTS. The order is important as each point is implicitly assigned an index depending on their row. The next section CELLS defines the connectivity of each cell. This depends on their type, as vertices have one point, lines have two points, triangles have three points and so forth. For each row, the first integer defines the number of points that define the cell and the next $\mathbf{numPoints_i}$ indices relate to the point $\mathbf{p_i}$ in the section above.

Listing C.3: *.VTK* file layout for unstructured grid,
see also https://docs.vtk.org/en/latest/design_documents/VTKFileFormats.html

```
# vtk DataFile Version x.x
Data header with maximum 256 characters
ASCII
DATASET UNSTRUCTURED_GRID
POINTS n dataType
p_0x p_0y p_0z
p_1x p_1y p_1z
...
p_n-1x p_n-1y p_n-1z

CELLS n size
numPoints_0, i_0, j_0, k_0, ...
numPoints_1, i_1, j_1, k_1, ...
numPoints_2, i_2, j_2, k_2, ...
...
numPoints_n-1, i_n-1, j_n-1, k_n-1, ...

CELL_TYPES n
type_0
type_1
type_2
...
type_n-1
```

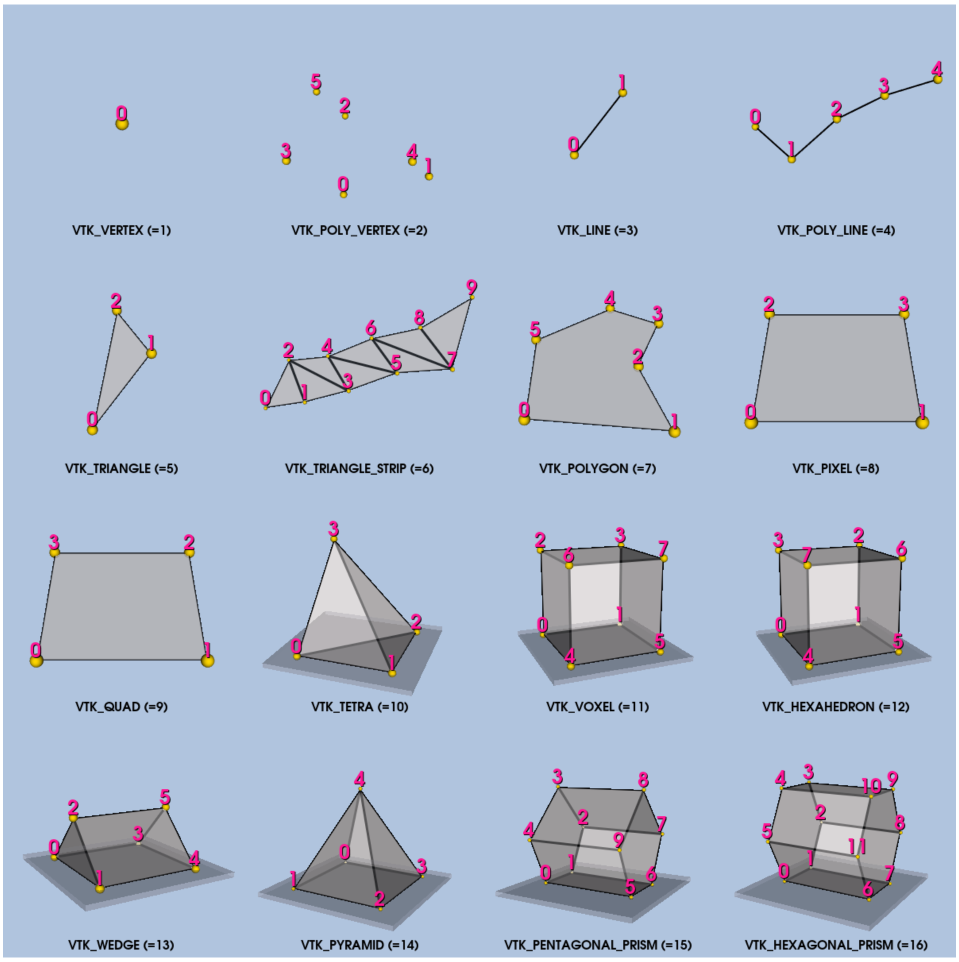


Figure C.10: Overview of available linear cell types [Schroeder et al., 2006].

Lastly, a section CELL_TYPES is inserted which specifies the VTK cell type. All available linear cell types are shown in figure C.10. For the cases used in this report, only cells of type 5 were used, namely triangular cells.

When additional variables are associated to a mesh, they are added to a new section, similar to the *.DAT* file in section C.2. Data can either be any of the cells in the mesh. This means that variables can not be stored in vertices, lines, triangle simplices and so forth. Throughout this report, results have always been written to triangle simplices.

Bibliography

Adrian, R. (1984). Scattering particle characteristics and their effect on pulsed laser measurements of fluid flow: Speckle velocimetry vs. particle image velocimetry. In: *Applied optics* 23.11, pp. 1690–1691. DOI: `10.1364/AO.23.001690`.

Agüera, N., Cafiero, G., Astarita, T., and Discetti, S. (Oct. 2016). Ensemble 3D PTV for high resolution turbulent statistics. In: 27.12, p. 124011. DOI: `10.1088/0957-0233/27/12/124011`.

Agüí, J. C. and Jiménez, J. (1987). On the performance of particle tracking. In: *Journal of Fluid Mechanics* 185, pp. 447–468. DOI: `10.1017/S0022112087003252`.

Aguiar Ferreira, M., Costa, P., and Ganapathisubramani, B. (Mar. 2024). Wall shear stress measurement using a zero-displacement floating-element balance. In: *Experiments in Fluids* 65.4. ISSN: 1432-1114. DOI: `10.1007/s00348-024-03785-1`.

Ahrens, J., Geveci, B., and Law, C. (2005). "ParaView: An End-User Tool for Large Data Visualization". In: *Visualization Handbook*. ISBN 978-0123875822. Elesvier.

Anderson, J. D. (2017). *Fundamentals of aerodynamics*. 6th. McGraw-Hill. ISBN: 978-1-259-25134-4.

Arroyo, M. P. and Greated, C. A. (1991). Stereoscopic particle image velocimetry. In: *Measurement Science and Technology* 2.12, p. 1181. DOI: `10.1088/0957-0233/2/12/012`.

Atkinson, K. E. (1989). *An Introduction to Numerical Analysis*. 2nd. John Wiley & Sons, Ltd. ISBN: 0-471-62489-6.

Baars, W. J., Squire, D. T., Talluru, K. M., Abbassi, M. R., Hutchins, N., and Marusic, I. (May 2016). Wall-drag measurements of smooth- and rough-wall turbulent boundary layers using a floating element. In: *Experiments in Fluids* 57.5. ISSN: 1432-1114. DOI: `10.1007/s00348-016-2168-y`.

Baker, T. J. (2005). Mesh generation: Art or science? In: *Progress in Aerospace Sciences* 41.1, pp. 29–63. ISSN: 0376-0421. DOI: `10.1016/j.paerosci.2005.02.002`.

Batchelor, G. K. (2000). *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library. Cambridge University Press.

Bergero, C., Gosnell, G., Gielen, D., Kang, S., Bazilian, M., and Davis, S. J. (2023). Pathways to net-zero emissions from aviation. In: *Nature Sustainability* 6, pp. 404–414. DOI: `10.1038/s41893-022-01046-9`.

Bertino, L., Evensen, G., and Wackernagel, H. (2003). Sequential Data Assimilation Techniques in Oceanography. In: *International Statistical Review / Revue Internationale de Statistique* 71.2. ISSN: 03067734, 17515823.

Bertsekas, D. P. (Jan. 1996). *Constrained Optimization and Lagrange Multiplier Methods*. 1st ed. Athena Scientific. ISBN: 1-886529-04-3.

Bosbach, J., Kühn, M., and Wagner, C. (Mar. 2009). Large scale particle image velocimetry with helium filled soap bubbles. In: *Experiments in Fluids* 46, pp. 539–547. DOI: `10.1007/s00348-008-0579-0`.

Boyd, S. and Vandenberghe, L. (2018). *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. Cambridge: Cambridge University Press.

Bridson, R. (2007). “Fast Poisson disk sampling in arbitrary dimensions”. In: *ACM SIGGRAPH 2007 Sketches*. SIGGRAPH ’07. San Diego, California: Association for Computing Machinery, 22–es. ISBN: 9781450347266. DOI: `10.1145/1278780.1278807`.

Brown, C. M. (1976). *Principal Axes and Best-Fit Planes, with Applications*. Tech. rep. Rochester, New York: University of Rochester.

Cai, S.-G., Degrigny, J., Boussuge, J.-F., and Sagaut, P. (2021). Coupling of turbulence wall models and immersed boundaries on Cartesian grids. In: *Journal of Computational Physics* 429, p. 109995. ISSN: 0021-9991. DOI: `10.1016/j.jcp.2020.109995`.

Cakir, B. O., Saiz, G. G., Sciacchitano, A., and van Oudheusden, B. W. (2022). Dense interpolations of LPT data in the presence of generic solid objects. In: *Measurement Science and Technology* 33 (12), p. 124009. DOI: `10.1088/1361-6501/ac8ec7`.

Chauhan, K. A., Monkewitz, P. A., and Nagib, H. M. (Mar. 2009). Criteria for assessing experiments in zero pressure gradient boundary layers. In: *Fluid Dynamics Research* 41.2, p. 021404. DOI: `10.1088/0169-5983/41/2/021404`.

Christiansen, I. (1973). Numerical simulation of hydrodynamics by the method of point vortices. In: *Journal of Computational Physics* 13.3, pp. 363–379. ISSN: 0021-9991. DOI: `10.1016/0021-9991(73)90042-9`.

Cierpka, C., Lütke, B., and Kähler, C. (May 2013). Higher order multi-frame particle tracking velocimetry. In: *Experiments in Fluids* 54. DOI: `10.1007/s00348-013-1533-3`.

Clauser, F. H. (1954). Turbulent Boundary Layers in Adverse Pressure Gradients. In: *Journal of the Aeronautical Sciences* 21.2, pp. 91–108. DOI: `10.2514/8.2938`.

Coles, D. (1956). The law of the wake in turbulent boundary layer. In: *Journal of Fluid mechanics* 2, pp. 191–226.

Coles, D. (1968). “The young person’s guide to the data. Compuation of turbulent boundary layers”. In: *AFOSR-IFP Stanford Conference*. Vol. 2.

Coxeter, H. S. M. (1969). “Affine geometry”. In: *Introduction to Geometry*. New York: Wiley. Chap. 13.

de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O. (2008). *Computational Geometry: Algorithms and Applications*. English. 4th ed. Germany: Springer Berlin. ISBN: 978-3-540-77974-2. DOI: `10.1007/978-3-540-77974-2`.

Delaunay, B. N. (1934). Sur la Sphere Vide. In: *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk* 7, pp. 793–800.

Delft University of Technology (2023). *W-Tunnel*. Accessed: 27-08-2023. URL: `https://www.tudelft.nl/lr/organisatie/afdelingen/flow-physics-and-technology/facilities/low-speed-wind-tunnels/w-tunnel`.

Deng, Z., He, C., Wen, X., and Liu, Y. (Aug. 2018). Recovering turbulent flow field from local quantity measurement: turbulence modeling using ensemble-Kalman-filter-based data assimilation. In: *Journal of Visualization* 21. DOI: `10.1007/s12650-018-0508-0`.

Depardon, S., Lasserre, J., Boueilh, J., Brizzi, L., and Jacques, B. (Nov. 2005). Skin friction pattern analysis using near-wall PIV. In: *Experiments in Fluids* 39, pp. 805–818. DOI: `10.1007/s00348-005-0014-8`.

Driver, D. M. (June 2003). Application of oil-film interferometry skin-friction measurement to large wind tunnels. In: *Experiments in Fluids* 34.6, pp. 717–725. ISSN: 1432-1114. DOI: `10.1007/s00348-003-0613-1`. URL: `http://dx.doi.org/10.1007/s00348-003-0613-1`.

Drouin, V., Giovannini, A., and Gillieron, P. (2002). “Topology characterization of the vortical near-wake flow over a simplified car model”. In: *Conference on bluff body wakes and vortex-induced vibrations*. Port Douglas, Australia.

Elsinga, G., Scarano, F., Wieneke, B., and van Oudheusden, B. W. (2006). Tomographic Particle Image Velocimetry. In: *Experiments in Fluids* 41, pp. 933–947. DOI: 10.1007/s00348-006-0212-z.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. In: *Journal of Geophysical Research: Oceans* 99.C5, pp. 10143–10162. DOI: 10.1029/94JC00572.

Evensen, G., Vossepoel, F., and Van Leeuwen, P. J. (Jan. 2022). *Data Assimilation Fundamentals: A Unified Formulation of the State and Parameter Estimation Problem*. ISBN: 978-3-030-96708-6. DOI: 10.1007/978-3-030-96709-3.

Fasshauer, G. E. (2007). *Meshfree Approximation Methods with Matlab*. Vol. 6. WORLD SCIENTIFIC. DOI: 10.1142/6437.

Fernholz, H. H., Janke, G., Schober, M., Wagner, P. M., and Warnack, D. (Oct. 1996). New developments and applications of skin-friction measuring techniques. In: *Measurement Science and Technology* 7.10, p. 1396. DOI: 10.1088/0957-0233/7/10/010.

Ferreira, M. A., Rodriguez-Lopez, E., and Ganapathisubramani, B. (Sept. 2018). An alternative floating element design for skin-friction measurement of turbulent wall flows. In: *Experiments in Fluids* 59.10. ISSN: 1432-1114. DOI: 10.1007/s00348-018-2612-2.

Ferrel, G., Aoki, K., and Lilley, D. (1985). “Flow visualization of lateral jet injection into swirling crossflow”. In: *23rd Aerospace Sciences Meeting*. Chap. AIAA Paper 85(0059), pp. 14–17. DOI: 10.2514/6.1985-59.

Foures, D. P. G., Dovetta, N., Sipp, D., and Schmid, P. J. (2014). A data-assimilation method for Reynolds-averaged Navier–Stokes-driven mean flow reconstruction. In: *Journal of Fluid Mechanics* 759, pp. 404–431. DOI: 10.1017/jfm.2014.566.

Friedberg, R. (2022). *Rodrigues, Olinde: ”Des lois géométriques qui régissent les déplacements d’un système solide...”, translation and commentary*. arXiv: 2211.07787 [math.HO].

van Gent, P., Michaelis, D., van Oudheusden, B., Weiss, P.-É., Kat, R., Laskari, A., Jeon, Y., David, L., Schanz, D., Huhn, F., Gesemann, S., Novara, M., McPhaden, C., Neeteson, N., Rival, D., Schneiders, J., and Schrijer, F. (Mar. 2017). Comparative assessment of pressure field reconstructions from particle image velocimetry measurements and Lagrangian particle tracking. In: *Experiments in fluids* 58:33. DOI: 10.1007/s00348-017-2324-z.

George, P. L., Borouchaki, H., Alauzet, F., Laug, P., Loseille, A., Marcum, D., and Maréchal, L. (2017). “Mesh Generation and Mesh Adaptivity: Theory and Techniques”. In: *Encyclopedia of Computational Mechanics Second Edition*. John Wiley & Sons, Ltd, pp. 1–51. ISBN: 9781119176817. DOI: 10.1002/9781119176817.ecm2012.

Gesemann, S., Huhn, F., Schanz, D., and Schröder, A. (2016). “From Noisy Particle Tracks to Velocity, Acceleration and Pressure Fields using B-splines and Penalties”. In: *8th International Symposium on Applications of Laser Techniques to Fluid Mechanics*. DOI: 10.48550/arXiv.1510.09034.

Geuzaine, C. and Remacle, J.-F. (Sept. 2009). Gmsh: A 3-D Finite Element Mesh Generator with built-in Pre- and Post-Processing Facilities. In: *International Journal for Numerical Methods in Engineering* 79.11, pp. 1309–1331. ISSN: 00295981. DOI: 10.1002/nme.2579. URL: http://dx.doi.org/10.1002/nme.2579.

Ghaemi, S. and Scarano, F. (Nov. 2010). Multi-pass light amplification for tomographic particle image velocimetry applications. In: *Measurement Science and Technology* 21.12, p. 127002. DOI: 10.1088/0957-0233/21/12/127002.

Giaquinta, D. (2018). “The Flow Topology of the Ahmed Body in Cross-Wind”. PhD thesis. Delft University of Technology.

Godbersen, P. and Schröder, A. (June 2020). Functional binning: improving convergence of Eulerian statistics from Lagrangian particle tracking. In: *Measurement Science and Technology* 31.9, p. 095304. DOI: 10.1088/1361-6501/ab8b84.

Goldstein, D., Handler, R., and Sirovich, L. (1993). Modeling a No-Slip Flow Boundary with an External Force Field. In: *Journal of Computational Physics* 105.2, pp. 354–366. ISSN: 0021-9991. DOI: `10.1006/jcph.1993.1081`.

González Saiz, G., Sciacchitano, A., and F., S. (Aug. 2022). On the closure of Collar's triangle by optical diagnostics. In: *Experiments in Fluids* 63. DOI: `10.1007/s00348-022-03468-9`.

Goodman, J. (1996). *Introduction to Fourier Optics*. 2nd ed. Electrical Engineering Series. McGraw-Hill. ISBN: 9780070242548.

Gurka, R., Liberzon, A., D, H., and Rubinstein, D. (Jan. 1999). "Computation of Pressure Distribution Using PIV Velocity Data". In: *3rd International workshop on particle image velocimetry*.

Hale, R., Tan, P., and Ordway, D. (1971). Experimental Investigation of Several Neutrally-Buoyant Bubble Generators for Aerodynamic Flow Visualization. In: *Naval Research Reviews* 24, pp. 19–24.

Hayase, T. (Jan. 2015a). A Review of Measurement-Integrated Simulation of Complex Real Flows. In: *Journal of Flow Control, Measurement & Visualization* 03, pp. 51–66. DOI: `10.4236/jfcmv.2015.32006`.

Hayase, T. (Sept. 2015b). Numerical simulation of real-world flows. In: *Fluid Dynamics Research* 47.5, p. 051201. DOI: `10.1088/0169-5983/47/5/051201`.

Hendriksen, L. (2024). Object registration. In: *Measurement Science and Technology*, to appear.

Huang, W.-X. and Tian, F.-B. (2019). Recent trends and progress in the immersed boundary method. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 233.23-24, pp. 7617–7636. DOI: `10.1177/0954406219842606`.

Hysa, I., Tuinstra, M., Sciacchitano, A., and Scarano, F. (2023). "A multi-directional redundant 3D-PIV system for ship deck wind interactions". In: *15th International Symposium on Particle Image Velocimetry*. San Diego, CA.

Imai, T., Kondo, K., Suzuki, Y., and Miki, Y. (Apr. 2023). Measurement of wall shear stress on an airfoil surface by using the oil film interferometry with PIV analysis applied to Fizeau fringes. In: *Journal of Fluid Science and Technology* 18. DOI: `10.1299/jfst.2023jfst0022`.

Jacobi, O. (1980). "PHOTOGRAMMETRIC TRACKING OF A MOVING PARTICLE IN RUNNING WATER". In: *14th Congress of the International Society of Photogrammetry*. Vol. 23. International Society for Photogrammetry and Remote Sensing.

Jahn, T., Schanz, D., and Schröder, A. (Aug. 2021). Advanced iterative particle reconstruction for Lagrangian particle tracking. In: *Experiments in Fluids* 62. DOI: `10.1007/s00348-021-03276-7`.

Jakobsen, M. L., Dewhirst, T. P., and Greated, C. A. (Dec. 1997). Particle image velocimetry for predictions of acceleration fields and force within fluid flows. In: *Measurement Science and Technology* 8.12, p. 1502. DOI: `10.1088/0957-0233/8/12/013`.

Jensen, A., Sveen, J. K., Grue, J., Richon, J., and Gray, C. A. (2001). Accelerations in water waves by extended particle image velocimetry. In: *Experiments in Fluids* 30 (5), pp. 500–510. DOI: `10.1007/s003480000229`.

Jeon, Y. J., Müller, M., Michaelis, D., and Wieneke, B. (2019). "Data assimilation-based flow field reconstruction from particle tracks over multiple time steps". In: *13th International Symposium on Particle Image Velocimetry*. Munich, Germany.

Jeon, Y. J., Schneiders, J., Müller, M., Michaelis, D., and Bernhard, W. (2018). "4D flow field reconstruction from particle tracks by VIC+ with additional constraints and multigrid approximation". In: *18th International Symposium Flow Visualization*. Zürich, Switzerland.

Jux, C., Sciacchitano, A., Schneiders, J., and Scarano, F. (2018). Robotic volumetric PIV of a full-scale cyclist. In: *Experiments in Fluids* 59 (4). DOI: `10.1007/s00348-018-2524-1`.

Jux, C., Sciacchitano, A., and Scarano, F. (2020). Flow pressure evaluation on generic surfaces by robotic volumetric PTV. In: *Measurement Science and Technology* 31. DOI: 10.1088/1361-6501/ab8f46.

Jux, C., Sciacchitano, A., and Scarano, F. (Feb. 2021). Object surface reconstruction from flow tracers. In: *Experiments in Fluids* 62. DOI: 10.1007/s00348-021-03139-1.

Kähler, C., Cierpka, C., and Scharnowski, S. (Feb. 2012). On the resolution limit of digital particle image velocimetry. In: *Experiments in Fluids* 52, pp. 1629–1639. DOI: 10.1007/s00348-012-1280-x.

Kähler, C. J., Scharnowski, S., and Cierpka, C. (May 2012). On the uncertainty of digital PIV and PTV near walls. In: *Experiments in Fluids* 52.6, pp. 1641–1656. ISSN: 1432-1114. DOI: 10.1007/s00348-012-1307-3. URL: http://dx.doi.org/10.1007/s00348-012-1307-3.

Kähler, C. J. (2004). "The significance of coherent flow structures for the turbulent mixing in wall-bounded flows". PhD thesis. University Goettingen Repository. DOI: 10.53846/goediss-2691. URL: http://dx.doi.org/10.53846/goediss-2691.

Kai, C. C., Jacob, G. G., and Mei, T. (Aug. 1997). Interface between CAD and rapid prototyping systems. Part 1: A study of existing interfaces. In: *The International Journal of Advanced Manufacturing Technology* 13.8, pp. 566–570. DOI: 10.1007/bf01176300.

Kasagi, N. and Nishino, K. (1991). Probing turbulence with three-dimensional particle-tracking velocimetry. In: *Experimental Thermal and Fluid Science* 4.5, pp. 601–612. ISSN: 0894-1777. DOI: 10.1016/0894-1777(91)90039-T.

de Kat, R. and van Oudheusden, B. W. (2012). Instantaneous planar pressure determination from PIV in turbulent flow. In: *Experiments In Fluids* 52, pp. 1089–1106.

Kempaiah, K. U., Scarano, F., Elsinga, G. E., van Oudheusden, B. W., and Bermel, L. (2020). 3-dimensional particle image velocimetry based evaluation of turbulent skin-friction reduction by spanwise wall oscillation. In: *Physics of Fluids* 32 (8). DOI: 10.1063/5.0015359.

Kempf, G. (1929). Neue Ergebnisse der Widerstandsforschung. In: *Werft, Reederei, Hafen*, pp. 234–239.

Kerho, M. and Bragg, M. (1994). Neutrally Buoyant Bubbles Used as Flow Tracers in Air. In: *Experiments in Fluids* 16, pp. 393–400. DOI: 10.1007/BF00202064.

Kolda, T. G. and Bader, B. W. (2009). Tensor Decompositions and Applications. In: *SIAM Review* 51.3, pp. 455–500. DOI: 10.1137/07070111X.

Kühn, M., Ehrenfried, K., Bosbach, J., and Wagner, C. (Apr. 2011). Large-scale tomographic particle image velocimetry using helium-filled soap bubbles. In: *Experiments in Fluids* Vol. 50, pp. 929–948. DOI: 10.1007/s00348-010-0947-4.

Launder, B. and Spalding, D. (1974). The numerical computation of turbulent flows. In: *Computer Methods in Applied Mechanics and Engineering* 3.2, pp. 269–289. ISSN: 0045-7825. DOI: 10.1016/0045-7825(74)90029-2.

LaVision (Aug. 27, 2023). *DaVis*. Version 11. URL: https://www.lavision.de/en/products/davis-software/.

Leroux, R., Chatellier, L., and David, L. (Apr. 2014). Bayesian inference applied to spatio-temporal reconstruction of flows around a NACA0012 airfoil. In: *Experiments in Fluids* 55. DOI: 10.1007/s00348-014-1699-3.

Liu, X. and Katz, J. (Aug. 2006). Instantaneous pressure and material acceleration measurements using a four-exposure PIV system. In: *Experiments in Fluids* 41, pp. 227–240. DOI: 10.1007/s00348-006-0152-7.

Maas, H.-G., Gruen, A., and Papantoniou, D. (1993). Particle tracking velocimetry in three-dimensional flows Part 1. Photogrammetric determination of particle coordinates. In: *Experiments in Fluids* 15, pp. 133–146. DOI: 10.1007/BF00190953.

Martins, F., Sciacchitano, A., and Rival, D. (2021). Detection of vortical structures in sparse Lagrangian data using coherent-structure colouring. English. In: *Experiments in Fluids* 62.4. ISSN: 0723-4864. DOI: 10.1007/s00348-021-03135-5.

McHenry, K. and Bajcsy, P. (Dec. 2008). *An overview of 3D data content, file formats and viewers*. Tech. rep. Urbana, Illinois: Image Spatial Data Analysis Group, National Center for Supercomputing Applications.

Melling, A. (Dec. 1997). Tracer particles and seeding for particle image velocimetry. In: *Measurement Science and Technology* 8.12, p. 1406. DOI: 10.1088/0957-0233/8/12/005.

Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. In: *AIAA Journal* 32.8, pp. 1598–1605. DOI: 10.2514/3.12149.

Michaelis, D., Mychkovsky, A. G., Wiswall, J. T., Prevost, R., Neal, D., and Wieneke, B. (2021). Calibration correction of arbitrary optical distortions by non-parametric 3d disparity field for planar and volumetric piv/lpt. In: *14th International Symposium on Particle Image Velocimetry* 1 (1). DOI: 10.18409/ispiv.v1i1.116.

Mie, G. (1908). Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen. In: *Annalen der Physik* 330.3, pp. 377–445. DOI: 10.1002/andp.19083300302.

Mons, V., Chassaing, J.-C., Gomez, T., and Sagaut, P. (2016). Reconstruction of unsteady viscous flows using data assimilation schemes. In: *Journal of Computational Physics* 316, pp. 255–280. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.04.022.

Monson, D. J., Mateer, G. G., and Menter, F. R. (Sept. 1993). “Boundary-Layer Transition and Global Skin Friction Measurement with an Oil-Fringe Imaging Technique”. In: *SAE Technical Paper Series*. AEROTECH. SAE International. DOI: 10.4271/932550. URL: http://dx.doi.org/10.4271/932550.

Nagib, H. and Chauhan, K. (Oct. 2008). Variations of von Kármán coefficient in canonical flows. In: *Physics of Fluids - PHYS FLUIDS* 20. DOI: 10.1063/1.3006423.

Naughton, J. W. and Sheplak, M. (Aug. 2002). Modern developments in shear-stress measurement. In: *Progress in Aerospace Sciences* 38.6–7, pp. 515–570. ISSN: 0376-0421. DOI: 10.1016/s0376-0421(02)00031-3.

Navon, I. M. (2009). Data assimilation for numerical weather prediction: a review. In: *Data assimilation for atmospheric, oceanic and hydrologic applications*, pp. 21–65. DOI: 10.1007/978-3-540-71056-1_2.

Nehrkorn, T., Woods, B. K., Hoffman, R. N., and Aulігné, T. (2015). Correcting for Position Errors in Variational Data Assimilation. In: *Monthly Weather Review* 143.4, pp. 1368–1381. DOI: 10.1175/MWR-D-14-00127.1.

Nigro, N., Filippini, G., Franck, G., Storti, M., and D’Elía, J. (2005). “Flow Around a Sharp-Edged Surface-Mounted Cube by Large Eddy Simulation”. In: vol. 24.

Nishino, K., Kasagi, N., and Hirata, M. (1989). Three-Dimensional Particle Tracking Velocimetry Based on Automated Digital Image Processing. In: *Journal of Fluids Engineering* 111.4, pp. 384–391. ISSN: 0098-2202. DOI: 10.1115/1.3243657.

Novara, M., Ianiro, A., and Scarano, F. (Dec. 2012). Adaptive interrogation for 3D-PIV. In: *Measurement Science and Technology* 24.2, p. 024012. DOI: 10.1088/0957-0233/24/2/024012. URL: https://dx.doi.org/10.1088/0957-0233/24/2/024012.

Novara, M., Schanz, D., Geisler, R., Gesemann, S., Voss, C., and Schröder, A. (Feb. 2019). Multi-Exposed recordings for 3D Lagrangian Particle Tracking with Multi-Pulse Shake-The-Box. In: *Experiments in Fluids* 60. DOI: 10.1007/s00348-019-2692-7.

Örlü, R. and Vinuesa, R. (2020). Instantaneous wall-shear-stress measurements: advances and application to near-wall extreme events. In: *Measurement Science and Technology* 31 (11), p. 112001. DOI: 10.1088/1361-6501/aba06f.

Örlü, R., Fransson, J. H., and Alfredsson, P. H. (2010). On near wall measurements of wall bounded flows-The necessity of an accurate determination of the wall position. In: *Progress in Aerospace Sciences* 46 (8), pp. 353–387. ISSN: 03760421. DOI: `10.1016/j.paerosci.2010.04.002`.

van Oudheusden, B. W. (2008). Principles and application of velocimetry-based planar pressure imaging in compressible flows with shocks. In: *Experiments in Fluids: experimental methods and their applications to fluid flow* 45.4, pp. 657–674. ISSN: 0723-4864. DOI: `10.1007/s00348-008-0546-9`.

van Oudheusden, B. W. (Jan. 2013). PIV-based pressure measurement. In: *Measurement Science and Technology* 24.3, p. 032001. DOI: `10.1088/0957-0233/24/3/032001`.

Ouellette, N., Xu, H., and Bodenschatz, E. (Feb. 2006). A quantitaive study of three-dimensional Lagrangian particle tracking algorithms. In: *Experiments in Fluids* 40, pp. 301–313. DOI: `10.1007/s00348-005-0068-7`.

Owen, S. (May 2000). A Survey of Unstructured Mesh Generation Technology. In: *7th International Meshing Roundtable* 3.

Peskin, C. S. (1972). Flow patterns around heart valves: A numerical method. In: *Journal of Computational Physics* 10.2, pp. 252–271. ISSN: 0021-9991. DOI: `10.1016/0021-9991(72)90065-4`.

Pickering, C. J. D. and Halliwell, N. A. (1984). Laser speckle photography and particle image velocimetry: photographic film noise. In: *Appl. Opt.* 23.17, pp. 2961–2969. DOI: `10.1364/AO.23.002961`.

Pletzer, A. and Hayek, W. (2019). Mimetic Interpolation of Vector Fields on Arakawa C/D Grids. In: *Monthly Weather Review* 147.1, pp. 3–16. DOI: `10.1175/MWR-D-18-0146.1`.

Pounder, E. (1956). *Parachute Inflation Process Wind-Tunnel Study*. Tech. rep. WADC Technical Report 56-391. Ohio, USA: Equipment Laboratory, Wright-Patterson Air Force Base, pp. 17–18.

Raffel, M., Kähler, C., Willert, C., Wereley, S., Scarano, F., and Kompenhans, J. (2018). *Particle Image Velocimetry: A Practical Guide*. English. 3rd. Springer. ISBN: 978-3-319-68851-0. DOI: `10.1007/978-3-319-68852-7`.

Rodriguez, E., Bruce, P., and Buxton, O. (Apr. 2015). A robust post-processing method to determine skin friction in turbulent boundary layers from the velocity profile. In: *Experiments in Fluids* 56. DOI: `10.1007/s00348-015-1935-5`.

Rusinkiewicz, S. and Levoy, M. (2001). “Efficient variants of the ICP algorithm”. In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152. DOI: `10.1109/IM.2001.924423`.

Saredi, E., Tumuluru Ramesh, N., Sciacchitano, A., and Scarano, F. (2021). State observer data assimilation for RANS with time-averaged 3D-PIV data. In: *Computers & Fluids* 218, p. 104827. ISSN: 0045-7930. DOI: `10.1016/j.compfluid.2020.104827`.

Savitzky, A. and Golay, M. (1964). Smoothing and differentiation of data by simplified least squares procedures. In: *Analytical Chemistry* 36 (8), pp. 1627–1639. DOI: `10.1021/ac60214a047`.

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. In: *Experiments in Fluids* 56 (2). DOI: `10.1007/s00348-015-1909-7`.

Scarano, F., Jux, C., and Sciacchitano, A. (2021). Recent advancements towards large-scale flow diagnostics by robotic piv. In: *Fluid Dynamics Research* 53 (1), p. 011401. DOI: `10.1088/1873-7005/abe034`.

Scarano, F., Ujjaini Kempaiah, K., and Kotsonis, M. (2022a). “PIV analysis of skin friction and coherent structures in turbulent drag reduction regimes”. English. In: *20th International Symposium on the Application of Laser and Imaging Techniques to Fluid Mechanics*.

Scarano, F. (Dec. 2010). Particle Image Velocimetry. In: *Encyclopedia of Aerospace Engineering*. DOI: `10.1002/9780470686652.eae074`.

Scarano, F. (2012). Tomographic PIV: Principles and practice. In: *Measurement Science and Technology* 24, p. 012001. DOI: 10.1088/0957-0233/24/1/012001.

Scarano, F. and Poelma, C. (July 2009). Three-Dimensional Vorticity Patterns of Cylinder Wakes. In: *Experiments in Fluids* 47, pp. 69–83. DOI: 10.1007/s00348-009-0629-2.

Scarano, F., Schneiders, J., Saiz, G., and Sciacchitano, A. (2022b). Dense velocity reconstruction with VIC-based time-segment assimilation. In: *Experiments in Fluids: experimental methods and their applications to fluid flow* 63.6. ISSN: 0723-4864. DOI: 10.1007/s00348-022-03437-2.

Schanz, D., Gesemann, S., and Schröder, A. (2016). Shake-The-Box: Lagrangian particle tracking at high particle image densities. In: *Experiments in Fluids* 57. DOI: 10.1007/s00348-016-2157-1.

Schanz, D., Gesemann, S., Schröder, A., Wieneke, B., and Novara, M. (Dec. 2012). Non-uniform optical transfer functions in particle imaging: Calibration and application to Tomographic reconstruction. In: *Measurement Science and Technology* 24.2, p. 024009. DOI: 10.1088/0957-0233/24/2/024009.

Schanz, D., Schröder, A., Novara, M., Geisler, R., Agocs, J., et al. (July 2019). “Large-Scale Volumetric Characterization of a Turbulent Boundary Layer Flow”. In: *13th International Symposium on Particle Image Velocimetry*. Munich, Germany.

Schneiders, J., Dwight, R. P., and Scarano, F. (2014). Time-supersampling of 3d-piv measurements with vortex-in-cell simulation. In: *Experiments in Fluids* 55 (3). DOI: 10.1007/s00348-014-1692-x.

Schneiders, J. and Scarano, F. (2016). Dense velocity reconstruction from tomographic ptv with material derivatives. In: *Experiments in Fluids* 57 (9). DOI: 10.1007/s00348-016-2225-6.

Schneiders, J., Scarano, F., Jux, C., and Sciacchitano, A. (2018). Coaxial volumetric velocimetry. In: *Measurement Science and Technology* 29 (6), p. 065201. DOI: 10.1088/1361-6501/aab07d.

Schröder, A., Willert, C., Schanz, D., Geisler, R., Jahn, T., Gallas, Q., and Leclaire, B. (2020). The flow around a surface mounted cube: a characterization by time-resolved piv, 3d shake-the-box and lbm simulation. In: *Experiments in Fluids* 61 (9). DOI: 10.1007/s00348-020-03014-5.

Schröder, A., Geisler, R., Elsinga, G., Scarano, F., and Dierksheide, U. (Feb. 2008). Investigation of a turbulent spot and a tripped turbulent boundary layer flow using time-resolved tomographic PIV. In: *Experiments in Fluids* Vol.: 44, pp. 305–316. DOI: 10.1007/s00348-007-0403-2.

Schröder, A. and Schanz, D. (2023). 3D Lagrangian Particle Tracking in Fluid Mechanics. In: *Annual Review of Fluid Mechanics* 55.1, pp. 511–540. DOI: 10.1146/annurev-fluid-031822-041721.

Schröder, A., Schanz, D., Geisler, R., Novara, M., and Willert, C. (2015). “Near-wall turbulence characterization using 4D-PTV “Shake-The-Box””. In: *11th International Symposium on Particle Image Velocimetry - PIV2015*. Conference Proceedings on USB-stick 238, pp. 1–14. URL: https://elib.dlr.de/95223/.

Schröder, A., Schanz, D., Heine, B., and Dierksheide, U. (2011). “Transitional Flow Structures Downstream of a Backward-Facing-Step—An Investigation Using High Resolution Tomo PIV”. In: *9th Internation Symposium PIV*. Kobe, Japan.

Schröder, A., Schanz, D., Novara, M., Philipp, F., Geisler, R., Knopp, T., Schroll, M., and Willert, C. (July 2018). “Investigation of a high Reynolds number turbulent boundary layer flow with adverse pressure gradients using PIV and 2D- and 3D- Shake-The-Box”. In: *19th International Symposium on the Application of Laser and Imaging Techniques to Fluid Mechanics*. Conference Proceedings online 257. 2.2 Turbulent Boundary Layers I.

Schroeder, W., Martin, K., and Lorensen, B. (2006). *The Visualization Toolkit (4th ed.)* Kitware. ISBN: 978-1-930934-19-1.

Sciacchitano, A., Giaquinta, D., Schneiders, J. F. G., Scarano, F., van Rooijen, B., and Funes, D. E. (2018). “Quantitative visualization of the flow around a turboprop aircraft model by robotic volumetric velocimetry”. In: *ISFV 18—Int. Symp. Flow Visualization*. Zurich, Switzerland. DOI: 10.3929/ethz-b-000279206.

Segalini, A., Rüedi, J.-D., and Monkewitz, P. A. (May 2015). Systematic errors of skin-friction measurements by oil-film interferometry. In: *Journal of Fluid Mechanics* 773, pp. 298–326. ISSN: 1469-7645. DOI: 10.1017/jfm.2015.237. URL: http://dx.doi.org/10.1017/jfm.2015.237.

Sheu, Y.-H., CHANG, T., TATTERSON, G., and DICKEY, D. (1982). A THREE-DIMENSIONAL MEASUREMENT TECHNIQUE FOR TURBULENT FLOWS. In: *Chemical Engineering Communications* 17.1-6, pp. 67–83. DOI: 10.1080/00986448208911615.

da Silva, B. L., Sumner, D., and Bergstrom, D. J. (July 2021). Mean and Dynamic Aspects of the Wakes of a Surface-Mounted Cube and Block. In: *Journal of Fluids Engineering* 144.1, p. 011302. ISSN: 0098-2202. DOI: 10.1115/1.4051428.

Soloff, S. M., Adrian, R. J., and Liu, Z.-C. (Dec. 1997). Distortion compensation for generalized stereoscopic particle image velocimetry. In: *Measurement Science and Technology* 8.12, p. 1441. DOI: 10.1088/0957-0233/8/12/008.

Spalart, P. and Allmaras, S. (Jan. 1994). A One-Equation Turbulence Model for Aerodynamic Flows. In: *AIAA* 439. DOI: 10.2514/6.1992-439.

Squire, L. C. (Jan. 1961). The motion of a thin oil sheet under the steady boundary layer on a body. In: *Journal of Fluid Mechanics* 11, pp. 161–179. DOI: 10.1017/S0022112061000445.

Symon, S., Dovetta, N., McKeon, B., Sipp, D., and Schmid, P. (Apr. 2017). Data assimilation of mean velocity from 2D PIV measurements of flow over an idealized airfoil. In: *Experiments in Fluids* 58. DOI: 10.1007/s00348-017-2336-8.

Talagrand, O. (Jan. 1997). Assimilation of Observations, an Introduction. In: *Journal of the Meteorological Society of Japan* 75, pp. 191–209. DOI: 10.2151/jmsj1965.75.1B_191.

Talagrand, O. and Courtier, P. (1987). Variational Assimilation of Meteorological Observations With the Adjoint Vorticity Equation. I: Theory. In: *Quarterly Journal of the Royal Meteorological Society* 113.478, pp. 1311–1328. DOI: 10.1002/qj.49711347812.

Tanner, L. H. and Blows, L. G. (Mar. 1976). A study of the motion of oil films on surfaces in air flow, with application to the measurement of skin friction. In: *Journal of Physics E: Scientific Instruments* 9.3, p. 194. DOI: 10.1088/0022-3735/9/3/015.

Tavoularis, S. (2005). *Measurement in Fluid Mechanics*. Cambridge, UK: Cambridge University Press.

Taylor, B. (1715). *Methodus Incrementorum Directa & Inversa*. Typis Pearsonianis.

TECPLOT (n.d.). *TECPLOT 360*.

Tecplot Inc. (2006). *Tecplot 360 Data Format Guide*.

Ting, D. S.-K. (2016). *Basics of Engineering Turbulence*. Ed. by D. S.-K. Ting. Academic Press, pp. 119–138. ISBN: 978-0-12-803970-0. DOI: 10.1016/B978-0-12-803970-0.00006-4.

Titchener, N., Colliss, S., and Babinsky, H. (Aug. 2015). On the calculation of boundary-layer parameters from discrete data. In: *Experiments in Fluids* 56. DOI: 10.1007/s00348-015-2024-5.

Tu, J., Yeoh, G. H., and Liu, C. (2018). *Computational Fluid Dynamics: A Practical Approach*. 2nd ed. USA: Butterworth-Heinemann. ISBN: 978-0-08-101127-0.

Verzicco, R. (2023). Immersed Boundary Methods: Historical Perspective and Future Outlook. In: *Annual Review of Fluid Mechanics* 55.1, pp. 129–155. DOI: 10.1146/annurev-fluid-120720-022129.

Vinuesa, R., Bartrons, E., Chiu, D., Dressler, K., Rüedi, J.-D., Suzuki, Y., and Nagib, H. (June 2014). New insight on flow development and two-dimensionality of turbulent channel flows. In: *Experiments in Fluids* 55. DOI: 10.1007/s00348-014-1759-8.

Vinuesa, R. and Örlü, R. (2017). “Measurement of wall-shear stress”. In: *Experimental Aerodynamics*. CRC Press Taylor & Francis Group. DOI: 10.1201/9781315371733-12.

Virant, M. and Dracos, T. (1997). 3D PTV and its application on Lagrangian motion. In: *Measurement Science and Technology* 8.12, p. 1539. DOI: `10.1088/0957-0233/8/12/017`.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. In: *Nature Methods* 17, pp. 261–272. DOI: `10.1038/s41592-019-0686-2`.

Wei, T., Schmidt, R., and Mcmurtry, P. (May 2005). Comment on the Clauser chart method for determining the friction velocity. In: *Experiments in Fluids* 38, pp. 695–699. DOI: `10.1007/s00348-005-0934-3`.

White, F. (2006). *Viscous Fluid Flow*. McGraw-Hill international edition. McGraw-Hill. ISBN: 9780071244930.

Wieneke, B. (Sept. 2008). Volume self-calibration for 3D particle image velocimetry. In: *Experiments in Fluids* 45, pp. 549–556. DOI: `10.1007/s00348-008-0521-5`.

Wieneke, B. (Dec. 2012). Iterative reconstruction of volumetric particle distribution. In: *Measurement Science and Technology* 24.2, p. 024008. DOI: `10.1088/0957-0233/24/2/024008`.

Wieneke, B. (Dec. 2018). Improvements for volume self-calibration. In: *Measurement Science and Technology* 29.8, p. 084002. DOI: `10.1088/1361-6501/aacd45`.

Wiener, N. (Aug. 1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. The MIT Press. ISBN: 9780262257190. DOI: `10.7551/mitpress/2946.001.0001`.

Wilcox, D. C. (1988). Multiscale model for turbulent flows. In: *AIAA Journal* 26.11, pp. 1311–1320. DOI: `10.2514/3.10042`.

Willert, C. (Dec. 1997). Stereoscopic digital particle image velocimetry for application in wind tunnel flows. In: *Measurement Science and Technology* 8.12, p. 1465. DOI: `10.1088/0957-0233/8/12/010`.

Yamagata, T., Hayase, T., and Higuchi, H. (June 2008). Effect of Feedback Data Rate in PIV Measurement-Integrated Simulation. In: *Journal of Fluid Science and Technology* 3. DOI: `10.1299/jfst.3.477`.

Yuksel, C. (2015). Sample Elimination for Generating Poisson Disk Sample Sets. In: *Computer Graphics Forum*. DOI: `10.1111/cgf.12538`.

Zhan, J.-m., Li, Y.-t., Wai, W.-h. O., and Hu, W.-q. (Dec. 2019). Comparison between the Q criterion and Rortex in the application of an in-stream structure. In: *Physics of Fluids* 31.12, p. 121701. ISSN: 1070-6631. DOI: `10.1063/1.5124245`.

Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3D: A Modern Library for 3D Data Processing. In: *arXiv:1801.09847*.