Lagrangian Flow Field Reconstruction Based on Constrained Stable Radial Basis Function

by

Lanyu Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Recent advances in three-dimensional (3D) high seeding density time-resolved Lagrangian particle tracking (LPT) techniques have made diagnosing fluid flows at high resolution in space and time under a Lagrangian framework feasible and practical. But challenges persist in developing LPT data processing methods. A promising processing method should accurately and robustly reconstruct, for example, particle trajectories, velocities, and differential quantities from noisy experimental data. Despite numerous algorithms available in the LPT community, they may suffer from some issues, such as unfavorable accuracy and robustness, lack of physical constraints, and unnecessary projection from Lagrangian data onto Eulerian meshes. These challenges may limit the application of the 3D high seeding density time-resolved LPT techniques.

In this thesis, a novel 3D Lagrangian flow field reconstruction method is proposed to address these challenges. The proposed method is based on a stable radial basis function (RBF) and constrained least squares (CLS). The stable RBF serves as a model function to approximate trajectories and velocity fields. The CLS provides a framework to facilitate regression and enforce physical constraints, further enhancing the reconstruction performance. The stable RBF and CLS work together to reconstruct 3D Lagrangian flow fields with high accuracy and robustness.

The proposed method offers several advantages over the algorithms currently used in the LPT community. First, it accurately reconstructs particle trajectories, velocities, and differential quantities in 3D, even from noisy experimental data, while satisfying physical constraints such as divergence-free for incompressible flows. Second, it does not require projecting Lagrangian data onto Eulerian meshes, allowing for direct flow field reconstruction at scattered data locations. Third, it effectively mitigates experimental noise in particle locations. Last, the proposed method enables smooth spatial and temporal super-resolution with ease. These advantages exhibit that the proposed method is promising for LPT data processing and further applications in data assimilation and machine learning.

Systematic tests were conducted to validate and verify the proposed method. Two-dimensional and 3D validations were performed using synthetic data based on exact solutions of the Taylor-Green vortex and Arnold-Beltrami-Childress flow with added artificial noise. The validations show that the proposed method outperforms baseline algorithms (e.g., finite difference methods and polynomial fittings) under different flow conditions. The method was then verified using experimental data from a 3D low-speed pulsing jet, showing its reliable performance. Based on these validations and verification, it is demonstrated that the proposed method can process experimental LPT data and reconstruct Lagrangian flow fields with accuracy and robustness.

Acknowledgements

I would like to express my appreciation to all those who made this thesis possible. First and foremost, I am grateful to my advisor, Dr. Zhao Pan, for his meticulous instructions and insightful discussions. I also extend my thanks to Dr. Elisabeth Larsson for her inspiring discussions and for sharing source codes of 3D RBF-QR, which is the core algorithm used in this thesis. Additionally, I am thankful to Dr. Md Nazmus Sakib for conducting experiments and providing data with timely support. Lastly, I would like to acknowledge Compute Canada for high-performance computational resources.

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

1D	One-Dimension or One-Dimensional
2D	Two-Dimensions or Two-Dimensional
3D	Three-Dimensions or Three-Dimensional
ABC	Arnold-Beltrami-Childress
BC	Boundary Condition
CLS	Constrained Least Squares
FDM	Finite Difference Method
LPT	Lagrangian Particle Tracking
LS	Least Squares
LS-POLY	Least Squares Polynomials
PIV	Particle Image Velocimetry
POLY	Polynomials
PTV	Particle Tracking Velocimetry
PUM	Partition-of-Unity Method
RBF	Radial Basis Function
RBF-CP	Radial Basis Function Contour-Padé

RBF-FD	Radial Basis Function-generated Finite Difference
RBF-RA	Radial Basis Function Rational Approximation
STB	Shake-The-Box
TGV	Taylor-Green Vortex
TR-LPT	Time-Resolved Lagrangian Particle Tracking
WCLS	Weighted Constrained Least Squares

Accents, Subscripts, and Superscripts

$\dot{a}$	first-order temporal derivative of a
$\hat{a}$	observation or measurement
$\langle a \rangle$	arithmetic average of a
$\tilde{a}$	reconstruction or approximation
a^{div}	evaluation of a with divergence-free constraints
a^{up}	updated evaluation of a in *Step 3*
a_0	initial value of a
$\bar{\mathbf{A}}$	3D extension of matrix $\mathbf{A}$
$\mathbf{A}^{+}$	generalized inverse of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of matrix $\mathbf{A}$
$\mathbf{A}^{\mathrm{T}}$	transpose of matrix $\mathbf{A}$
$\mathbf{A}_t$	first-order temporal differentiation of matrix $\mathbf{A}$
$\mathbf{A}_{tt}$	second-order temporal differentiation of matrix $\mathbf{A}$

Roman Symbols

$\mathcal{D}$	generalized linear differentiation operator
$\mathcal{E}$	relative error of reconstruction

$\mathcal{L}$	Lagrangian objective function
$\mathcal{R}$	residual between measurement and model function
$\max(a)$	maximum of a
d_x, d_y, d_z	longitudinal, lateral, and normal spatial displacements (phase) of exact solution in Cartesian coordinate system
M_{trj}	number of temporal reference points along pathline
N_{sns}	number of snapshots
N_{trj}	number of temporal centers along pathline or number of frames
p	polynomial coefficient
P_{trj}	number of temporal evaluation points along pathline
r	Euclidean norm (distance) between two points
t	time
u, v, w	longitudinal, lateral, and normal velocity components in Cartesian coordinate system
x, y, z	longitudinal, lateral, and normal spatial coordinates in Cartesian coordinate system
$\mathbf{B}$	RBF-QR system matrix
$\mathbf{C}$	constraint matrix
$\mathbf{C}_{\mathcal{D}}$	differentiation constraint matrix
$\mathbf{C}_{\mathcal{O}}$	constraint matrix associated with fixed function values
$\mathbf{D}$	RBF-Direct system matrix
$\mathbf{E}$	RBF-QR evaluation matrix
$\mathbf{E}_{\mathcal{D}}$	RBF-QR differentiation matrix
$\mathbf{F}$	integrated RBF-QR value matrix

$\mathbf{G}$	integrated RBF-QR system matrix
$\mathbf{H}$	integrated trajectory and velocity matrix
$\mathbf{K}$	integrated trajectory and velocity evaluation matrix
$\mathbf{P}$	polynomial basis matrix
$\mathbf{Q}$	orthogonal matrix from QR-factorization
$\mathbf{R}^*$	upper triangular matrix from QR-factorization
$\mathbf{U}$	velocity vector
$\mathbf{V}$	intermediate divergence-free velocity matrix
$\mathbf{X}$	modified trajectory matrix
$\boldsymbol{d}$	vector of constraint values
$\boldsymbol{p}$	vector of polynomial coefficients
$\boldsymbol{R}$	rotation-rate tensor
$\boldsymbol{S}$	strain-rate tensor
$\boldsymbol{t}^{\mathrm{ref}}$	vector of temporal reference points
$\boldsymbol{t}^{\mathrm{eva}}$	vector of temporal evaluation points
$\boldsymbol{x}$	vector of spatial coordinates
$\boldsymbol{x}^{\mathrm{cst}}$	vector of spatial coordinates of constraint points
$\boldsymbol{x}^{\mathrm{eva}}$	vector of spatial coordinates of evaluation points
$\boldsymbol{x}^{\mathrm{ref}}$	vector of spatial coordinates of reference points
$\hat{\boldsymbol{t}}^{\mathrm{c}}$	vector of time sequence of temporal centers from measurement
$\hat{\boldsymbol{x}}_i^{\mathrm{c}}$	vector of spatial coordinates of center i from measurement
$\hat{f}_i^{\mathrm{c}}$	scalar data at center i from measurement
$\mathbb{R}$	Set of real numbers in one-dimension

$\mathbb{R}^d$	Set of real numbers in d-dimensions
$\tilde{s}_{\mathcal{D}}$	differentiation approximation function
$\tilde{s}$	approximation function
d	dimension
J	number of constraint points
L	spatial span of domain in one direction
M	number of reference points
N	number of centers
P	number of evaluation points
Q	iso-value of Q-criterion

Greek Symbols

$\mathbf{\Phi}$	RBF-Direct basis matrix
$\mathbf{\Phi}_{\mathcal{D}}$	RBF-Direct differentiation basis matrix
$\mathbf{\Psi}$	RBF-QR basis matrix
$\mathbf{\Psi}_{\mathcal{D}}$	RBF-QR differentiation basis matrix
η_0	temporal oversampling ratio
η_1	spatial oversampling ratio in *Step 2*
η_2	spatial oversampling ratio in *Step 4*
κ	index of frame
α	amplitude of exact solution
$\mathbf{\Lambda}$	update expansion coefficient matrix
$\boldsymbol{\lambda}$	vector of RBF expansion coefficients
$\boldsymbol{\lambda}^*$	vector of Lagrangian multipliers

$\boldsymbol{\lambda}^{\mathrm{frm}}$	vector of RBF expansion coefficients in frame
$\boldsymbol{\lambda}^{\mathrm{trj}}$	vector of RBF expansion coefficients along pathline
λ	RBF expansion coefficient
λ^*	Lagrangian multiplier
Ω	fluid flow domain
ω	spatial frequency of exact solution
ϕ	RBF-Direct kernel
$\phi_{\mathcal{D}}$	RBF-Direct differentiation kernel
ψ	RBF-QR kernel
$\psi_{\mathcal{D}}$	RBF-QR differentiation kernel
$\sigma(a)$	standard deviation of a
ε	shape factor
ξ	noise level
Δt	time interval between two consecutive frames
δt	time interval between two consecutive snapshots
Δx	particle displacement between two consecutive frames

Chapter 1

Introduction

In fluid mechanics, there are two fundamental perspectives to describe a flow field, i.e., the Eulerian and Lagrangian perspectives [7, 32, 95]. In the Eulerian perspective, fluid flows are observed within a fixed control volume, and individual fluid parcels are not identified. Functions of independent field variables describe the fluid flow. For example, a velocity component u can be expressed as $u = u(x, y, z, t)$, where x, y, z, and t are independent field variables that represent the spatial coordinates in a 3D domain and the time, respectively. On the other hand, the Lagrangian perspective focuses on individual fluid parcels and tracks their motions independently. The flow variables are associated with a specific fluid parcel and depend on each other. For example, if a fluid parcel is tracked, its velocity component u is given by $u = u(x_0, y_0, z_0, t)$, where x_0, y_0, z_0 are the initial spatial coordinates of the fluid parcel. In other words, the particle velocity is uniquely determined by the identity of a fluid parcel and time. The Lagrangian and Eulerian perspectives lay the foundations for fluid mechanics.

Despite classic textbooks [7, 32, 95] giving equal attention to the Lagrangian and Eulerian descriptions when first introducing them in the first chapters, in later chapters and real-world practices, the Lagrangian descriptions are overshadowed by their Eulerian counterparts. This is true for the computational and experimental fluid mechanics where most of the numerical [14, 44, 97, 171] and experimental [133, 164, 166] techniques are built upon the Eulerian description. In experiments, one possible reason is the lack of feasible and practical Lagrangian experimental techniques that can track particle motions with high spatial and temporal resolutions in 3D [3, 33]. In turn, the Eulerian description has been dominating fluid experiments.

However, with the recent development of 3D high-seeding density time-resolved Lagrangian particle tracking (3D TR-LPT or four-dimensional LPT) [141, 142, 176], the Lagrangian description is back to focus and demonstrates some superiority in experimental fluid mechanics. The 3D TR-LPT and other Lagrangian experimental techniques have been intensively researched and increasingly used in recent years [20, 85, 121, 141, 146, 147, 149, 151, 153, 167]. The superiority of the 3D TR-LPT stems from the high-seeding density and time-resolved properties in addition to its Lagrangian nature. The high-seeding density can achieve a high spatial resolution and the time-resolved property enables an LPT[1] system to identify transient and time-correlated flow structures [11, 126]. LPT techniques keep developing today, driven by the unique advantages of the Lagrangian description.

The 3D TR-LPT, including some other Lagrangian experimental techniques, has three major advantages over the Eulerian ones. First, the Lagrangian description is perhaps more intuitive than the Eulerian one since fluid flows are inherently Lagrangian. A volume of fluid consists of an 'infinite' number of fluid parcels that densely fill the flow domain. Each fluid parcel follows its unique pathline, and the collective motion of fluid parcels reflects the fluid flow. For example, the secondary flow phenomenon can be observed by stirring a cup of tea and tracking the movements of tea leaves at the bottom of the cup [17]. Following the movements of tea leaves is essentially diagnosing the flow from a Lagrangian perspective. This property facilitates straightforward analysis and a heuristic understanding of fluid flows.

Second, the LPT systems can potentially achieve higher spatial resolution than that of typical Eulerian experimental techniques [2, 33, 73, 155], such as particle image velocimetry (PIV). The PIV techniques rely on tracking the patterns of a few particles within finite-size interrogation windows. The interrogation windows are often chosen based on a rule of thumb [131]. Reconstruction based on the interrogation windows can lead to a spatially averaged (smoothing or smearing) velocity field and inherently inaccuracy in regions with high velocity gradients [9, 29, 65, 66, 78, 81, 136, 175]. In contrast to PIV, the LPT techniques do not require interrogation windows, which facilitates the measurement of fine-scale flow structures and accurate fluid flow diagnosis.

Third, the LPT techniques enable trajectory-based measurement. By tracking the trajectories of individual particles, the history of particle movements can be recovered. This allows the detection of where a particle originated from and where it is moving towards (e.g., identifying the sources and sinks of fluid flows). Examples of LPT experiments leveraging the trajectory-based measurement include indoor airflow studies [12, 51], Lagrangian coherent structure identification [55, 135] and fluid mixing research [6, 88, 134].

---

[1] Although the terms LPT (Lagrangian particle tracking) and PTV (particle tracking velocimetry) are often used interchangeably, this thesis exclusively uses the term LPT, as recommended by [149].

This advantage is particularly attractive as it expands the application of fluid experiments to under-explored areas. The demand for LPT experiments, Lagrangian data, and the corresponding data processing techniques is increasing. Therefore, this thesis emphasizes the processing of experimental data obtained through the LPT techniques.

Developing an accurate and robust algorithm for LPT data processing is not trivial. It challenges the community for two major reasons: i) the raw LPT data, which are directly output from LPT experiments, technically offer nothing but locations of particles as time series. Important flow information, such as velocities and velocity gradients, cannot be directly accessed without proper flow reconstruction. Therefore, effective LPT data processing methods are needed to reconstruct trajectories, velocities, and differential quantities based on the raw LPT data. ii) The raw LPT data are inaccurate due to various sources of noise and bias, including those from measurement and processing. Noise and bias can significantly degrade reconstruction quality depending on the choice of the reconstruction method. Hence, suppressing noise is crucial to have reliable reconstruction results. Many LPT processing algorithms have been proposed in the community. They are summarized in the next chapter.

The thesis is organized as follows: Chapter 2 provides a survey on the flow field reconstruction methods. In Chapter 3, the two core algorithms (i.e., constrained least squares and a stable radial basis function) of the proposed method are summarized. Chapter 4 elaborates on the two core algorithms combined as a novel method for flow reconstruction. In Chapter 5, validations and verification using the proposed method are discussed. Last, Chapter 6 concludes the thesis and provides future outlooks.

Chapter 2

Literature Review

This chapter reviews the methods used, or with the potential to be used, to reconstruct particle trajectories, velocities, and differential quantities for the LPT data. Section 2.1 summarizes the trajectory and velocity reconstruction methods. Section 2.2 reviews the differential quantity reconstruction methods. Section 2.3 focuses on methods for noise reduction and constraint enforcement. Section 2.4 highlights two particular flow field reconstruction methods, which are more systematic than the rest and may be considered state-of-the-art. Section 2.5 points out some limitations of current methods in the LPT community. This literature review provides the motivation for the proposed method.

2.1 Trajectory and Velocity Reconstruction

The primary objective of LPT experiments is to recover particle trajectories and measure velocities based on the raw LPT data. The raw LPT data are converted from a set of LPT images of the particles. The LPT image conversion usually involves three steps: particle image recognition, particle coordinate determination, and particle tracking. The particle image recognition (also known as particle detection [74, 108, 111] or identification [126]) identifies tracer particles from the LPT images based on the brightness and size of pixel clusters. It also determines the planar spatial coordinates of the identified tracer particles in the images. The particle coordinate determination (also called particle correspondence [108, 111]) converts particle planar spatial coordinates in images to the volumetric spatial coordinates in a global reference frame in the physical space [126]. In the particle tracking step [126] (also referred to as particle pairing [34], linking [108] or matching [74]), the same particle appears in different frames is recognized and labeled as one particle. After these three steps, the LPT images are converted to the raw LPT data that can be used for trajectory and velocity reconstruction.

Perhaps the most straightforward way to recover trajectories and measure velocities is by directly invoking particle tracking results and using the first-order finite difference method, respectively. For example, particle trajectories can be represented by line segments that connect the same particles identified by particle tracking algorithms in successive frames [119] or just a series of dots without explicit connections [9, 60, 61, 79, 80, 84, 112, 130, 158, 173]. This method was commonly used in the early stage of LPT application [1, 60, 61, 80, 84, 119, 130]. The first-order finite difference method calculates a particle velocity $\mathbf{U}$ by dividing the particle displacement $\Delta \boldsymbol{x}$ between two consecutive frames by the corresponding time interval Δt: $\mathbf{U} \approx \Delta \boldsymbol{x} / \Delta t$ [9, 60, 61, 79, 80, 84, 94, 111, 119, 158, 173]. The velocity between two consecutive frames can be interpolated by piece-wise linear functions [119] if a temporal super-resolution is needed. These trajectory and velocity reconstruction methods are widely used in the LPT community.

Two alternative algorithms that are similar to the first-order finite difference method can be found in some literature. One algorithm measures particle velocities based on long-exposure images. In a long-exposure image, a particle leaves a pathline. The particle pathline is approximated by straight line segments, assuming that the pathline has low curvature. The velocity is then calculated by dividing the length of the line segment by the corresponding exposure time. This approach allows velocity measurement in a single frame and has been used in several studies [1, 83, 84]. The other algorithm involves more consecutive frames to construct higher-order stencils. This can increase velocity evaluation accuracy and is usually used for multi-frame LPT data. For example, a central difference scheme has a three-point stencil along a trajectory with second-order accuracy [10, 111, 159] and a central difference scheme has a five-point stencil with fourth-order accuracy [111]. These two algorithms can either extend the applicability of the finite difference methods or increase the accuracy of velocity measurement.

However, using the finite difference method can potentially lead to inaccurate velocity reconstruction due to several reasons. First, the finite difference method is based on a weak assumption that a particle travels a short distance and its pathline is nearly straight [29]. If the pathline is curved, this assumption may not hold and lead to inaccurate reconstruction. Second, the finite difference method can amplify the noise present in the raw LPT data during velocity computation. Presumably, the derivative of noisy data is often even noisier, and some filtering or smoothing techniques are usually required. However, the direct use of the finite difference method does not provide any filtering and results in highly noisy velocity calculations. These two reasons restrain the application of finite difference methods for accurate and robust reconstruction.

An alternative approach to recover trajectories and measure velocities is using continuous functions, and a polynomial is one of them. The trajectories of particles can be approximated by polynomials with different orders. The order of polynomials depends on the desired accuracy and the number of available frames. The velocity is the first-order temporal differentiation of the trajectory functions. This method can be considered coupled reconstruction, as the velocity is derived from the same polynomial function as the trajectory. Common choices for the order of polynomials are the first-order [34, 83, 84], second-order [29, 83, 84, 130, 177], third-order [29, 63, 90, 91, 108, 109, 141], and fourth-order [43]. Polynomial functions are mostly applied with a least squares algorithm. The least squares algorithm can suppress the noise and improve the robustness of the reconstruction. This polynomial-based method outperforms finite difference methods [29, 108] (see Section 5.3 for some tests) and is also widely used in the LPT data processing.

However, polynomial-based reconstruction methods have two major drawbacks: (i) a polynomial function cannot be infinitely smooth, which is not an ideal choice for trajectory functions. The infinite smoothness of trajectory functions derives infinitely smooth acceleration functions, which is important for downstream applications such as data assimilation and force reconstruction. Although the smoothness of polynomial functions can be increased by using high-order polynomials, constructing a high-degree smoothness polynomial is difficult when the number of frames is limited. (ii) Polynomial-based reconstruction often faces a dilemma in choosing between low- or high-order polynomials. The low-order polynomials may not accurately approximate trajectories with high curvature, while the high-order polynomials may lead to numerical oscillations [137]. This problem is particularly pronounced when approximating long particle trajectories, where both high and low curvature exists simultaneously on one pathline. These drawbacks may hinder the application of polynomials in reconstruction for realistic experiments.

The use of splines is an alternative way to reconstruct particle trajectories [15, 53, 94, 141], sharing many similarities with polynomials but being superior to typical polynomials. The basis spline (B-spline) [93, 148] is one of them, being the cornerstone in computer graphics [30, 127, 178, 180] and computer-aided design [24, 107, 143]. A B-spline is essentially a piece-wise polynomial. When a particle trajectory is approximated by a B-spline, its velocity is calculated by the first-order temporal differentiation of the trajectory function. The B-splines are flexible because of their piece-wise and compact support nature, which avoids the aforementioned dilemma of polynomials and is apt to reconstruct long particle trajectories. They have been increasingly used for reconstruction in LPT experiments in recent years [15, 53]. However, they cannot have high-degree smoothness either, especially when the number of frames is limited. Moreover, the choice of knots and types of B-splines, such as uniform or non-uniform B-splines, must be carefully considered and is usually flow-dependent. These shortcomings may limit the application of splines in flow field reconstruction.

2.2 Differential Quantity Reconstruction

In addition to the trajectory and velocity, differential quantities, such as velocity spatial derivatives including velocity gradients [23, 108, 113, 159], vorticity [5, 31, 72, 75, 146, 161], strain- and rotation-rate tensors [92, 103, 104], coherent structures [23, 103, 104, 135], and even pressure and pressure gradients [15, 53, 157], are also of interest. The differential quantities can reveal valuable flow characteristics and aid in-depth analysis of complex flows. For example, strain- and rotation-rate tensors are essential for flow kinematics studies in turbulence. Among the above differential quantities, the velocity gradient is perhaps the most fundamental one, and other differential quantities can be derived directly or indirectly based on the velocity gradient. Differential quantity reconstruction is one step forward beyond the trajectory and velocity reconstruction and a promising LPT data processing method is expected to provide reliable differential quantity reconstruction. This section will review the reconstruction methods for differential quantities.

The methods for reconstructing differential quantities can be categorized as either mesh-based or meshless. The primary difference between these two families of methods is the need for data conversion to map Lagrangian data onto a Eulerian mesh. Mesh-based methods require this conversion, while meshless methods directly compute differential quantities at scattered data points.

The mesh-based methods perform data conversions before differential quantity computation. In a data conversion, continuous functions are used to approximate velocities at scattered points first. Then the velocities at the nodal points on the structured mesh are interpolated by the continuous functions. Several data conversion functions are widely used, including polynomial-based functions such as polynomials [31, 72, 73, 159], B-splines [53], and Taylor expansions of multivariate polynomials [113]; and radial basis functions (RBFs) such as Gaussian RBFs [5, 23, 75, 76, 145, 146, 159], multi-quadratic RBFs [60, 61]. After data conversions, differential quantities are computed on Eulerian meshes.

The finite difference methods (FDM) are perhaps the most convenient way to compute velocity gradients on a structured mesh. They have been widely used in experiments [5, 31, 75, 76, 146, 159]. The accuracy of the FDM-based differential quantity reconstruction is sensitive to the size of the stencil and grid spacing. To achieve high accuracy, finite difference methods require fine meshes and large stencils, which can be challenging due to limited computational resources. Furthermore, the fine mesh applied to noisy experimental data may lead to high errors in the computed gradients.

Alternatively, direct differentiation of data conversion functions can also be used to compute differential quantities. Since the data conversion functions are continuous, direct spatial differentiation is possible at nodal points on the Eulerian meshes. For polynomial-based conversion functions [31, 53, 72, 73, 113, 159], because the smoothness of reconstructed differential quantity fields is limited by the low order of their underlying polynomials, the reconstruction accuracy might be undermined. Particularly, in the Taylor expansion [113], the accuracy also depends on the value of the residual term, which can be negligible only when the grid spacing of the Eulerian mesh is small and the spatial resolution of the Lagrangian measurement is high. If these conditions are not met, reconstruction accuracy may be compromised.

For the RBF data conversion functions, although they can possess infinite smoothness, the choice of flatness of the RBF kernel is often empirical and not trivial. The flatness of the RBF kernel is controlled by a parameter called the shape factor. In the methods that adopt Gaussian RBF [5, 23, 159], the shape factor is set to be equal to a constant times the mean distance between particles. This is an attempt at an adaptive shape factor. However, the 'optimal' value of the shape factor involves many other considerations beyond the mean distance between particles: flow profiles (e.g., velocity gradients) and the pattern of the particle distribution (e.g, random or quasi-uniform distribution) may also play a role [103]. An inappropriate shape factor may lead to numerical instability and inaccurate approximation for flows with different length scales and data points with various densities. More information on this topic is available in Section 3.1.2 and literature in the numerical analysis community [45, 48, 98]. Most importantly, data conversions inevitably introduce additional errors when mapping Lagrangian data onto Eulerian meshes.

For the meshless methods, differential quantity reconstruction is directly performed at Lagrangian data without mapping them onto a Eulerian mesh. A velocity field is described by continuous functions, and the velocity derivatives are calculated by the spatial differentiation of the continuous functions. Popular choices of the continuous functions are polynomials [108, 161], splines [15, 158], and RBFs [103, 104, 157]. Although polynomials, splines, and RBFs can be used in both mesh-based and meshless methods, their missions are different. In a mesh-based method, these functions are used as interpolant or regression functions to map scattered data onto a structured mesh. In a meshless method, these functions are used to approximate the velocity field based on the scatter data, and projecting the data onto a structured mesh is avoided. The continuous function in a meshless method is usually constructed by the least squares algorithm to provide smooth reconstruction [15, 103, 108, 157, 161]. The least squares fitting carries a low-pass filtering feature (see Section 2.3) and has advantages over interpolation (see Section 3.2). The meshless methods avoid additional errors introduced by converting Lagrangian data to Eulerian meshes, and they are promising for high-accuracy differential quantity reconstruction. In the thesis, a meshless method has been chosen to reconstruct differential quantities.

2.3 Noise Reduction and Constraint Enforcement

Noise and errors are inevitable in experiments including the LPT. Suppressing noise is necessary and can be achieved by different methods such as moving average filter methods [5, 159] and least squares fitting [15, 31, 53, 103, 108, 113, 157, 158, 161]. The moving average filter methods replace the data at one point with a weighted average of its neighboring data [5, 159]. The moving filter methods assume that the data of a particle can not be drastically different from its nearby particles. However, in some complex flows with high gradients, moving average filter methods may smear fine flow structures, which undermines reconstruction accuracy. In addition, the moving average filter can be viewed as a low-pass filter and its cut-off frequency is determined by the range of the neighboring data. Determining the range of the nearby points is flow-dependent and empirical. Instead of directly applying a low-pass filter on the LPT data, the least squares fitting can also achieve denoising [53, 108]. The least squares fitting minimizes the total residual between the reconstructed functions and the experimental measurements. The resulting reconstructed function inherently smooths out the high-frequency noise and highlights the low-frequency flow structures. In this thesis, the least squares fitting is adopted to suppress noise and improve the accuracy and robustness of the proposed method.

In addition to noise reduction, imposing physical constraints could improve the quality of reconstruction. These constraints can arise from any knowledge about the fluid flow such as the flow properties, boundary conditions, and even governing equations. Examples include the divergence-free property for an incompressible flow [23, 75, 76, 157, 159], Dirichlet and/or Neumann conditions for a pressure field [23, 75, 76, 146, 157], and Navier-Stokes equations for a fluid flow [53]. Imposing a physical constraint means enforcing the experimental data to respect the governing equation to some extent. This is the fundamental philosophy behind data assimilation [96, 100] and many physics-informed machine learning techniques [21, 114, 132]. In this thesis, divergence-free constraints are imposed for incompressible flows as a first attempt.

2.4 Comprehensive Reconstruction

Comprehensive reconstruction methods can evaluate trajectory, velocity, and differential quantity altogether incorporating noise reduction and constraint enforcement. These methods first initialize trajectories and velocities based on the raw LPT data. Then they calculate differential quantities using either mesh-based or meshless algorithms based on initial trajectories and velocities. Additional data processing algorithms, such as noise filter, and outlier removal, can be also implemented as comprehensive reconstruction. In this section, two comprehensive reconstruction methods are highlighted.

Gesemann et al. [53] proposed a mesh-based reconstruction method. This method consists of two algorithms, i.e., Trackfit and Flowfit, and they work together as one complete LPT data processing strategy. The Trackfit initializes particle trajectories, velocities, and accelerations based on raw LPT data. The trajectory of a particle is approximated by a 1D cubic B-spline. The velocity and acceleration functions are calculated by the first- and second-order temporal differentiation of the trajectory functions, respectively. An over-determined system in the least squares sense is established when reconstructing particle trajectories. The over-determined system is based on a cost function consisting of two parts: i) the total residuals between trajectory measurements and their B-spline approximation, and ii) a weighted acceleration variation.[1] The B-spline coefficients can be retrieved by minimizing the cost function and solving the over-determined system. By substituting the coefficients back to the B-splines, the particle trajectory, velocity, and acceleration functions are recovered.

The Flowfit completes the remaining flow field reconstruction. It utilizes the particle velocities and accelerations initialized by the Trackfit as input. In the first step of the Flowfit, the Lagrangian data (i.e., the particle velocities and accelerations initialized by the Trackfit) are converted onto a Eulerian mesh using 3D weighted cubic B-splines. Another cost function that consists of residuals between measurements and the 3D B-spline approximation with weights and penalties is established. Weights and penalties are determined by

---

[1] The LPT systems typically use two criteria for particle tracking: minimum acceleration [123, 173] or minimum change in acceleration [112, 173]. The minimum acceleration criterion assumes that a particle has a low rate of change of velocity between two successive frames. Therefore, the particle tracking algorithms should choose the candidate particle that has the minimum acceleration as the same particle along the pathline. The minimum change in acceleration criterion assumes that a particle has a low averaged jerk between two successive frames. Hence, the particle tracking algorithms should choose the candidate particle that has the minimum change in acceleration as the same particle along the pathline. The minimum change in acceleration criterion is adopted implicitly in [53] by minimizing a cost function that comprises acceleration variation.

measurement error and/or flow properties, such as the velocity measurement error and velocity divergence. The coefficients of the 3D B-splines are then retrieved by minimizing the cost function using the least squares algorithm. The cost function regarding the coefficients of the B-splines may be linear or nonlinear, depending on the types of constraints. The linear and nonlinear cost function is optimized by a conjugate gradient method for least squares [122] and limited-memory Broyden-Fletcher-Goldfarb-Shanno [120] algorithm, respectively. After retrieving the expansion coefficients, velocity fields are reconstructed, and differential quantities can be calculated by the spatial differentiation of the velocity field functions.

The Trackfit plus Flowfit strategy has been applied in many works. For example, it has been used for studies in turbulent Rayleigh-Bénard convection [16], turbulent bubbly jet [86], large-scale turbulent boundary layer [140], impinging jet flow [68, 69, 87], pure thermal plume [67], and flow around a surface mounted cube [150, 154]. However, in the Trackfit, the trajectory cost function that adopts the minimum change in acceleration criterion, enforcing minimum jerk on the trajectory of a particle, may not be always valid. This criterion implies that the time interval between two adjacent frames is small, which may not be suitable for all applications. In addition, the Flowfit still needs data conversions, which may introduce additional errors due to mapping Lagrangian data onto Eulerian meshes. Besides, the velocity fields reconstructed by the Flowfit are not fully explored. For example, these divergence-free velocities could be used to update the initial trajectories since the velocity fields reconstructed by Flowfit respect many physical constraints while the trajectories from Trackfit are not. Such an update may further improve the accuracy and robustness of trajectory reconstruction.

The other comprehensive reconstruction method was proposed by Lüthi [108]. In this method, a localized least squares cubic polynomial function, named “moving cubic spline” by the author, is used to reconstruct particle trajectories. The trajectory function is essentially approximated by third-order polynomials. The trajectory reconstruction possesses a noise reduction feature because of the use of the least squares algorithm. The velocity and acceleration functions are calculated by the first- and second-order temporal differentiation of the trajectory functions, respectively. Next, linear piece-wise polynomial functions are employed to approximate the velocity field in each frame. An over-determined system about the coefficients of the linear piece-wise polynomial functions is constructed to map measured data onto evaluation points. By solving the over-determined system, the coefficients of polynomials are retrieved. The polynomial coefficients are then substituted into linear piece-wise polynomial functions to obtain the velocity field function in each frame. The velocity gradient field is calculated by the spatial differentiation of the corresponding velocity field function. In addition, optional weights on measured data can be applied to the least square algorithm in trajectory and acceleration reconstruction. The weights are based on neighboring particle distance and/or relative velocity divergence. For the particles that are far from an evaluation point, their data are assigned low weights, which implies that their influence on reconstruction is low. The velocity divergence is evaluated by the reconstructed velocity gradient fields. Particles with low relative velocity divergence are assigned high weights for incompressible flows, while particles with high relative velocity divergence are assigned low weights. The weights make reconstruction more closely match the data with low relative divergence than those with high. The weighted least squares algorithm improves reconstruction quality to some extent.

Lüthi's method has been applied in many LPT experiments, such as vorticity dynamics in turbulence [109], 3D scanning PTV [63], small-scale and large-scale PTV studies [91], shear-free turbulence with confinements and rotations [90], two-scale turbulence measurement in a rotational system [89], turbulent and non-turbulent interface study [62], and others [177]. This method attempted to 'softly' incorporate divergence-free property for incompressible flows to improve the reconstruction by leveraging a nonzero divergence as a normalized weight, however, the divergence-free is not strictly enforced. Additionally, the linear piece-wise polynomial functions only provide piece-wise linear smoothness throughout the domain. The limited smoothness is inadequate for accurately representing complex velocity gradient fields. In summary, following Lüthi's approach, a meshless method with constrained least squares is adopted in this thesis to provide an infinitely smooth reconstruction with enforced physical constraints.

2.5 Summary

Despite many methods for processing LPT data being available (see Table 2.1 for an overview of these methods), several challenges still need to be addressed. These include: (i) the use of mesh-based methods that require data conversions, which may introduce unnecessary numerical errors and divert from the original purpose of LPT experiments; (ii) the lack of enforcement of physical constraints, which can be crucial for realistic and reliable flow reconstruction; (iii) the significant degradation of reconstruction quality due to the noise in particle spatial coordinates from measurement; and (iv) the fixed-order polynomials (including some splines) for trajectory reconstruction cannot universally capture trajectories with varying curvatures. Therefore, the constrained least squares (CLS) radial basis function (RBF) Lagrangian reconstruction method (referred to as the CLS-RBF method) is proposed in this thesis to address these challenges and reconstruct Lagrangian flow fields.

Table 2.1: An overview of flow field reconstruction methods.[1]

Author	Data Conv.	Traj.[2]	Velocity Field[3]	Diff. Quantity	Noise Supp.	Physical Const.	Traj. Update[4]
Agüí & Jimenez [5]	✓	—	Gaussian RBF	FDM	✓	×	×
Bobro et al. [15]	×	B-splines	Taylor Exp.	Taylor Exp.	✓	✓	×
Casa & Krueger [23]	✓	—	Gaussian RBF	Spatial Diff	✓	✓	×
Gesemann et al. [53]	✓	B-splines	B-splines	Spatial Diff.	✓	✓	×
Hassan et al. [60, 61]	✓	—	MQ-RBF	Spatial Diff.	✓	×	×
Jeon et al. [75]	✓	—	RBF	VIC#	✓	✓	×
Lüthi [108]	×	cubic Poly.	linear Poly.	Spatial Diff.	✓	×	×
Malik et al. [113]	✓	—	Taylor Exp.	Taylor Exp.	✓	×	×
Schneiders et al. [146]	✓	—	RBF	VIC+	✓	✓	×
Sperotto et al. [157]	×	—	Gaussian RBF+Poly.	Spatial Diff.	✓	✓	×
Stegeman [158]	×	—	splines	Spatial Diff.	✓	×	×
Takehara & Etoh [161]	×	—	cubic Poly.	Spatial Diff.	✓	×	×
CLS-RBF method	×	RBF-QR	RBF-QR	Spatial Diff.	✓	✓	✓

[1] The abbreviations in this table: Data Conv.: data conversion; Traj.: trajectory; Diff.: differential or differentiation; Noise Supp.: noise suppression; Physical Const.: physical constraints; Poly.: polynomials; Taylor Exp.: Taylor expansion of multivariate polynomials; MQ-RBF: multiquadric RBF.

[2] The function used to approximate a particle trajectory along a pathline.

[3] The function used to approximate a velocity field in a frame.

[4] Trajectory update based on the reconstructed velocity fields.

Chapter 3

Radial Basis Function and Constrained Least Squares

In this chapter, two pillars of the CLS-RBF Lagrangian reconstruction method will be elaborated. In Section 3.1, the classic radial basis function (RBF) interpolation and differentiation will be summarized first, and then stable RBF algorithms proposed in the mathematics community are reviewed. One of the stable RBF algorithms, i.e., RBF-QR, is focused on. Section 3.2 presents the RBF-QR embedded into the constrained least squares (CLS) framework to achieve divergence-free reconstruction in 3D.

3.1 Overview of RBF

3.1.1 RBF-Direct

The classic RBF, also called RBF-Direct in some literature [45, 46, 47, 98, 179], is a kernel-based meshless algorithm for approximation. It uses the Euclidean norm (distance) between fixed points as independent variables and can fit or interpolate scattered data. Depending on the choice of a specific kernel (also known as a basis), the RBF approximation can be infinitely differentiable and easily extended to high dimensions. The RBF methods have wide applications in data regression, computer graphics [22, 110], machine learning [25, 64, 82, 105], etc. It is also suitable for flow field reconstruction [103, 104, 157].

The RBF-Direct can be used for fundamental interpolation and differentiation. Given the scalar data $\hat{f}_i^{\mathrm{c}} \in \mathbb{R}$ at a scattered center (also called a nodal point) $\hat{\boldsymbol{x}}_i^{\mathrm{c}} \in \mathbb{R}^d$ in flow domain Ω, i.e., $(\hat{\boldsymbol{x}}_i^{\mathrm{c}}, \hat{f}_i^{\mathrm{c}})$, where $i \in \{1, 2, ...N\}$, N is the number of centers and d is dimensions, the RBF-Direct interpolation function $\tilde{s}(\boldsymbol{x})$ is a linear combination of weighted kernels [42, 45]:

$$\tilde{s}(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda_i \phi(\varepsilon, \boldsymbol{r}_i) = \sum_{i=1}^{N} \lambda_i \phi(\varepsilon, \|\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\mathrm{c}}\|), \tag{3.1}$$

where $\phi(\varepsilon, \boldsymbol{r}_i)$ is an RBF-Direct kernel (e.g., a Gaussian kernel $\phi = e^{-\varepsilon^2 \boldsymbol{r}_i^2}$), $\boldsymbol{r}_i = \|\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\mathrm{c}}\|$ is the vector of Euclidean norm between a location $\boldsymbol{x}$ in the domain Ω and the center $\hat{\boldsymbol{x}}_i^{\mathrm{c}}$, λ_i is the expansion coefficient that controls the weight of a radial basis, and ε is the shape factor that controls the profile of the kernel. Eqn. (3.1) can be written in a matrix form:

$$\tilde{s}(\boldsymbol{x}) = \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{\lambda}, \tag{3.2}$$

where $\boldsymbol{\lambda}$ is the vector of expansion coefficients with entries $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_N)^{\mathrm{T}}$, and the RBF-Direct basis matrix $\boldsymbol{\Phi}(\boldsymbol{x})$ is based on $\phi(\varepsilon, \boldsymbol{r}_i)$. The differentiation of Eqn. (3.2) can be computed with the same $\boldsymbol{\lambda}$ in Eqn. (3.2):

$$\tilde{s}_{\mathcal{D}}(\boldsymbol{x}) = \boldsymbol{\Phi}_{\mathcal{D}}(\boldsymbol{x})\boldsymbol{\lambda}, \tag{3.3}$$

where $\mathcal{D}$ is a generalized linear differential operator, $\boldsymbol{\Phi}_{\mathcal{D}}(\boldsymbol{x})$ is an RBF-Direct differentiation basis matrix with entries $\phi_{\mathcal{D}}(\varepsilon, \boldsymbol{r}_i) = \phi_{\mathcal{D}}(\varepsilon, \|\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\mathrm{c}}\|)$. $\boldsymbol{\Phi}_{\mathcal{D}}$ and $\phi_{\mathcal{D}}$ can be calculated by the chain rule. For example, in the x direction, the spatial differentiation basis matrix $\boldsymbol{\Phi}_x$ has the entries based on ϕ_x, and ϕ_x is calculated by

$$\phi_x = \frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial r}\frac{\partial r}{\partial x}. \tag{3.4}$$

In general, the Eqns. (3.2) and (3.3) are used for fundamental RBF-Direct interpolation and differentiation, respectively.

However, when using Eqn. (3.2) or (3.3), some details must be considered. These details include the choice of kernels, the calculation of the vector of expansion coefficients $\boldsymbol{\lambda}$, and the value of the shape factor. The RBF-Direct kernel can take different forms. For example, popular choices are a Gaussian kernel $\phi = e^{-\varepsilon^2 r^2}$, a multiquadric kernel $(\phi = \sqrt{1 + (\varepsilon r)^2})$ [58], and an inverse multiquadric kernel $(\phi = 1/\sqrt{1 + (\varepsilon r)^2})$ [50]. In this thesis, the Gaussian kernel has been chosen to accommodate a stable RBF algorithm [45]. The vector of expansion coefficients $\boldsymbol{\lambda}$ is the key to the interpolation and differentiation. Once the vector of expansion coefficients $\boldsymbol{\lambda}$ is calculated, the interpolation and differentiation are almost completed. To calculate the vector of expansion coefficients

$\boldsymbol{\lambda}$, the evaluation on interpolant $\tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}})$ is forced to be equal to the given data $\hat{\boldsymbol{f}}^{\mathrm{c}}$ based on Eqn. (3.1):

$$\begin{aligned} \tilde{s}(\boldsymbol{x})\big|_{\hat{\boldsymbol{x}}_i^{\mathrm{c}}} = \tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) &= \hat{\boldsymbol{f}}^{\mathrm{c}} \\ \hat{\boldsymbol{f}}^{\mathrm{c}} &= \mathbf{D}\boldsymbol{\lambda} \qquad\qquad , \\ \boldsymbol{\lambda} &= \mathbf{D}^{-1}\hat{\boldsymbol{f}}^{\mathrm{c}} \end{aligned} \tag{3.5}$$

where $\hat{\boldsymbol{f}}^{\mathrm{c}}$ the vector of given scalar data with entries $\hat{\boldsymbol{f}}^{\mathrm{c}} = (\hat{f}_1^{\mathrm{c}}, \hat{f}_2^{\mathrm{c}}, \ldots, \hat{f}_N^{\mathrm{c}})^{\mathrm{T}}$ and $\mathbf{D}$ is an RBF-Direct system matrix with entries $D_{ij} = \phi\big(\varepsilon, \big\|\hat{\boldsymbol{x}}_i^{\mathrm{c}} - \hat{\boldsymbol{x}}_j^{\mathrm{c}}\big\|\big)$, $i, j = 1, 2, \ldots, N$. The shape factor ε controls the flatness of the radial basis. A small shape factor leads to a near-flat kernel, while a large shape factor is associated with a spiky one (see Figure 3.1 for illustration). The choice of shape factors can significantly influence the approximation quality. This topic will be discussed in the next section. By choosing a proper kernel function, calculating the vector of expansion coefficients $\boldsymbol{\lambda}$ correctly, and setting a suitable value of the shape factor, the RBF-Direct can be applied for reliable interpolation and differentiation.

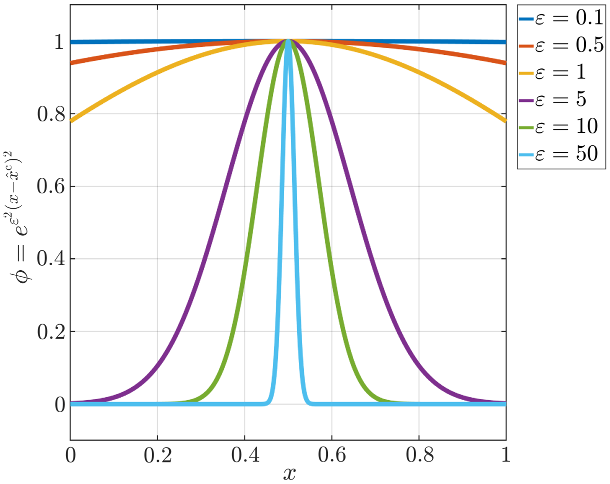


Figure 3.1: RBF-Direct Gaussian kernel with different shape factors.

3.1.2 Stable RBF algorithms

Although RBF-Direct can achieve fundamental interpolation and differentiation and has been widely used in various communities [22, 25, 64, 82, 105, 110], a critical dilemma regarding the choice of the shape factor exists in the RBF-Direct application. On the one hand, a small shape factor corresponding to the near-flat basis can achieve an accurate approximation but ill-conditioning may occur leading to unstable computation. This phenomenon may be exacerbated if a large number of centers are involved [45, 48, 98]. On the other hand, a large shape factor results in well-conditioned computation, but it causes inaccurate approximation at the same time [45, 48, 98]. It was conventionally believed that one had to carefully tune the shape factor to strike a balance between accuracy and stability until some stable RBF algorithms were invented [46, 48].

The stable RBF algorithms can achieve numerical stability without compromising accuracy. They overcome the ill-conditioning problem caused by a small shape factor. To the best of the author's knowledge, four stable RBF algorithms have been proposed in the numerical analysis community up to now, i.e., RBF Contour-Padé (RBF-CP) [48], RBF Rational Approximation (RBF-RA) [179], RBF-GA [46], and RBF-QR [45]. The RBF-CP algorithm resolves the ill-conditioning arising from a small shape factor by removing the restriction that the shape factor has to be a real number. The RBF-CP approximation function consists of a rational function in ε and power series in ε to achieve stable computation with a near-flat basis. It uses fast Fourier transfer, Laurent expansion, and Padé rational form [48] to derive the RBF-CP. However, this algorithm is computationally expensive and can only approximate a limited number of centers (e.g., less than a hundred in 2D and a few hundred in 3D) [46, 48, 179]. The RBF-RA algorithm is based on vector-value rational approximation, which is similar to the RBF-CP. But the RBF-RA computes the rational approximation by constructing a coupled over-determined linear system that contains the shape factor [179]. Although the RBF-RA is more accurate, robust, and computationally efficient than the RBF-CP, it can only interpolate up to approximately a hundred points in 2D and a few hundred in 3D [46, 179]. The RBF-GA algorithm is specifically based on the Gaussian kernel and aims to improve the radial basis function-generated finite difference (RBF-FD) method. It transforms an ill-conditioned RBF-Direct basis to a well-conditioned RBF-GA basis using the incomplete gamma function without truncating an infinite expansion. However, the RBF-GA can only interpolate a few hundred nodal points in 2D and about five hundred in 3D. Furthermore, its differentiation is complex because it is based on a modified Gaussian basis and the differentiation matrix is not straightforward [46]. Although these three algorithms are applicable in approximation to achieve stable computation, they suffer from various issues and are not the optimal choices for flow field reconstruction.

In this thesis, the RBF-QR has been chosen as the stable RBF algorithm in the CLS-RBF method. The RBF-QR is proposed by Fornberg et al. [45]. Unlike the RBF-CP and RBF-RA, which use vector-valued rational approximation, the RBF-QR is based on the idea that the ill-conditioned basis can be spun to a well-conditioned one while maintaining the same approximation space. In the RBF-QR, the RBF-Direct basis matrix $\boldsymbol{\Phi}(\varepsilon, \boldsymbol{r})$ is converted to a more stable basis matrix $\boldsymbol{\Psi}(\varepsilon, \boldsymbol{r})$ by a process of factorization, Taylor expansion, coordinate conversion, and Chebyshev polynomial substitution. The new basis matrix $\boldsymbol{\Psi}(\varepsilon, \boldsymbol{r})$ enjoys well-conditioning and stability for $\varepsilon \to 0$. Unlike the RBF-Direct, the RBF-QR is capable of approximating straight lines when using proper expansion coefficients, while the RBF-Direct cannot achieve this approximation because of the limitation in kernel profiles (see Figure 3.1 and Figure 3.2 for comparison). The RBF-QR can interpolate more points than the RBF-CP, RBF-GA, and RBF-RA, up to thousands of points in 2D and 3D. Therefore, the RBF-QR stands out among stable RBFs and becomes the core algorithm in the CLS-RBF method. More details about the RBF-QR can be found in [45, 98, 99].

Applying the RBF-QR in interpolation and differentiation is straightforward and similar to that of the RBF-Direct. In practice, an RBF-Direct basis matrix $\boldsymbol{\Phi}$ can be directly replaced by a corresponding RBF-QR basis matrix $\boldsymbol{\Psi}$. Similar to Eqns. (3.2) and (3.3), the RBF-QR interpolation function $\tilde{s}(\boldsymbol{x})$ and its differentiation function $\tilde{s}_{\mathcal{D}}(\boldsymbol{x})$ are

$$\begin{aligned} \tilde{s}(\boldsymbol{x}) &= \boldsymbol{\Psi}(\boldsymbol{x})\boldsymbol{\lambda} \\ \tilde{s}_{\mathcal{D}}(\boldsymbol{x}) &= \boldsymbol{\Psi}_{\mathcal{D}}(\boldsymbol{x})\boldsymbol{\lambda}, \end{aligned} \tag{3.6}$$

respectively, where $\boldsymbol{\Psi}(\boldsymbol{x})$ is an RBF-QR basis matrix and $\boldsymbol{\Psi}_{\mathcal{D}}(\boldsymbol{x})$ is an RBF-QR differentiation basis matrix. The same as the RBF-Direct, the expansion coefficient of the RBF-QR is the key to interpolation and differentiation. The vector of expansion coefficients $\boldsymbol{\lambda}$ is calculated by forcing $\tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) = \hat{\boldsymbol{f}}^{\mathrm{c}}$:

$$\begin{aligned} \tilde{s}(\boldsymbol{x})|_{\hat{\boldsymbol{x}}_i^{\mathrm{c}}} &= \tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) = \hat{\boldsymbol{f}}^{\mathrm{c}} \\ \hat{\boldsymbol{f}}^{\mathrm{c}} &= \mathbf{B}\boldsymbol{\lambda} \\ \boldsymbol{\lambda} &= \mathbf{B}^{-1}\hat{\boldsymbol{f}}^{\mathrm{c}} \end{aligned} \quad , \tag{3.7}$$

where the RBF-QR system matrix $\mathbf{B}$ has entries: $B_{ij} = \psi\big(\varepsilon, \big\|\hat{\boldsymbol{x}}_i^{\mathrm{c}} - \hat{\boldsymbol{x}}_j^{\mathrm{c}}\big\|\big)$, $i, j = 1, 2, \ldots, N$. Then the vector of expansion coefficients $\boldsymbol{\lambda}$ is substituted to Eqn. (3.6). Up to now, the interpolation and differentiation based on the RBF-QR is completed.

3.2 Constrained Least Squares RBF-QR

Interpolation is not ideal for Lagrangian flow field reconstruction. Interpolation requires the reconstructed functions to pass through every single data point, implying that the data are assumed to be fully trustworthy. However, the LPT data are contaminated in experiments, and forcing the interpolant to precisely pass through these inaccurate data is unnecessary and, perhaps, inappropriate. Besides, interpolation may suffer from unstable computation, especially in high-gradient regions. Numerical oscillation may occur in these areas and lead to divergence in computation and/or inaccuracy in results. Furthermore, it is almost impossible for interpolation to impose physical constraints as the interpolant is solely determined by the given data, and forcing any constraints may lead to an ill-posed problem. On the contrary, the CLS does not require the reconstructed functions to pass through every given data point. This relaxation provides two key benefits: i) smooth and stable reconstruction even in the presence of contaminated data; ii) realistic reconstruction by imposing physical constraints. Therefore, in this thesis, the CLS algorithm has been chosen, instead of interpolation.

In the CLS-RBF method, the CLS and RBF-QR are two foundational building blocks serving different roles. The RBF-QR is a model function for Lagrangian flow field reconstruction. The model function is essentially a trial function that approximates the 'true' trajectory, velocity, and velocity gradient functions in flow fields. The CLS is a framework for approximation. It is compatible with various model functions, such as polynomials, splines, and RBFs. Under the CLS framework, the model functions fit given data, instead of interpolating them. The model functions can also be enforced by constraints. In summary, the CLS provides a platform for the RBF-QR, in which the RBF-QR is used as model functions to approximate 'true' Lagrangian flow fields, with smooth fitting and physical constraints enforcement.[1]

3.2.1 Constrained least squares RBF-QR

In this section, the least squares algorithm without constraints is discussed first, and then constraints are imposed. The least squares algorithm minimizes the total residuals between model functions and measurements, and it is widely used for regression [36, 39, 49], system identification [8, 26, 170], and machine learning [18, 19, 168], etc. In the context of the least squares (LS) RBF-QR, the model function is specified as the RBF-QR. The same as RBF-Direct and RBF-QR interpolation, the key to RBF-QR fitting is finding the corresponding expansion coefficients. To calculate the vector of expansion coefficients $\boldsymbol{\lambda}$, a residual $\mathcal{R}$

---

[1] The basic assumption of using RBF-QR is that the flow fields can be described by a continuous function. However, in the case of high-speed and compressible fluid flows, discontinuities such as shock waves can occur, rendering this assumption invalid.

between measurements and the model functions is minimized:

$$\min \ \mathcal{R} = \sum_{i}^{N} \left\| \tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) - \hat{f}_i^{\mathrm{c}} \right\|^2 , \tag{3.8}$$

where $\hat{f}_i^{\mathrm{c}}$ is the given scalar data at center i from measurement and $\tilde{s}(\hat{\boldsymbol{x}}_{i=1}^{\mathrm{c}}) = \mathbf{B}\boldsymbol{\lambda}|_{\hat{\boldsymbol{x}}_i^{\mathrm{c}}}$ is the reconstructed value of the RBF-QR model function at the same location as the center i, N is the number of the centers. Eqn. (3.8) is an objective function in the context of optimization. Note that $\tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) = \mathbf{B}\boldsymbol{\lambda}|_{\hat{\boldsymbol{x}}_i^{\mathrm{c}}}$ is different from $\tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) = \mathbf{B}(\hat{\boldsymbol{x}}_i^{\mathrm{c}})\boldsymbol{\lambda}$ because $\tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) = \mathbf{B}\boldsymbol{\lambda}|_{\hat{\boldsymbol{x}}_i^{\mathrm{c}}}$ is a generalized RBF-QR approximation function at a center $\hat{\boldsymbol{x}}_i^{\mathrm{c}}$, regardless of interpolation or fitting, while $\tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) = \mathbf{B}(\hat{\boldsymbol{x}}_i^{\mathrm{c}})\boldsymbol{\lambda}$ is specified as the RBF-QR interpolation at $\hat{\boldsymbol{x}}_i^{\mathrm{c}}$ based on Eqn. (3.6). To calculate $\boldsymbol{\lambda}$, the extrema of $\mathcal{R}$ should be found by setting the gradient of $\mathcal{R}$ with respect to $\boldsymbol{\lambda}$ to zero (i.e., $\partial\mathcal{R}/\partial\boldsymbol{\lambda} = 0$, [42, 52, 99]). Consequently, the vector of expansion coefficients is:

$$\boldsymbol{\lambda} = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\hat{\boldsymbol{f}}^{\mathrm{c}}. \tag{3.9}$$

Like the RBF interpolation, once the vector of expansion coefficients $\boldsymbol{\lambda}$ is computed, the RBF-QR least squares fitting is almost completed.

However, different from the RBF interpolation, a ‘trick’ needs to be applied here to achieve a proper fitting. As shown in Eqn. (3.9), the RBF-QR system matrix $\mathbf{B}$ must be non-squared, otherwise $(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}$ reduces to $\mathbf{B}^{-1}$, in which least squares fitting turns to interpolation. To implement least squares fittings, reference points are introduced in computation [42, 98, 99]. M reference points $\boldsymbol{x}_j^{\mathrm{ref}} \in \mathbb{R}^d$, $j = 1, 2, \ldots, M$ are created in the domain Ω. Some details about the reference points and other points will be discussed shortly. The RBF-QR system matrix $\mathbf{B}$ is constructed using M reference points $\boldsymbol{x}_j^{\mathrm{ref}}$ and N centers $\hat{\boldsymbol{x}}_i^{\mathrm{c}}$, with entries:

$$B_{ij} = \psi\left(\varepsilon, \left\|\hat{\boldsymbol{x}}_i^{\mathrm{c}} - \boldsymbol{x}_j^{\mathrm{ref}}\right\|\right), \tag{3.10}$$

where ψ is an RBF-QR kernel, $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$. The ratio of the number of centers N and reference points M is the over-sampling ratio $\eta = N/M$. An oversampling ratio that is larger than unity (i.e., $\eta > 1$) is required to formulate an over-determined mapping for regression [98, 99, 103]. After constructing the non-squared RBF-QR system matrix $\mathbf{B}$, the vector of expansion coefficients $\boldsymbol{\lambda}$ is calculated by Eqn. (3.9).

With the known expansion coefficient $\boldsymbol{\lambda}$, the RBF-QR fitting and differential approximation functions are calculated using corresponding matrices. The RBF-QR evaluation matrix $\mathbf{E}$ and its differentiation matrix $\mathbf{E}_{\mathcal{D}}$ are mapped between P evaluation points

$\boldsymbol{x}_k^{\text{eva}} \in \mathbb{R}^d$ and M reference points $\boldsymbol{x}_j^{\text{ref}}$ with entries:

$$\begin{aligned}
\mathbf{E} &= E_{kj} = \psi\left(\varepsilon, \left\|\boldsymbol{x}_k^{\text{eva}} - \boldsymbol{x}_j^{\text{ref}}\right\|\right), \\
\mathbf{E}_{\mathcal{D}} &= E_{\mathcal{D},kj} = \psi_{\mathcal{D}}\left(\varepsilon, \left\|\boldsymbol{x}_k^{\text{eva}} - \boldsymbol{x}_j^{\text{ref}}\right\|\right),
\end{aligned} \tag{3.11}$$

where $j = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, P$. The least squares RBF-QR approximation function is given by:

$$\tilde{s}(\boldsymbol{x}) = \mathbf{E}\boldsymbol{\lambda} = \mathbf{E}\mathbf{B}^{+}\hat{\boldsymbol{f}}^{\text{c}}, \tag{3.12}$$

where $\mathbf{B}^{+} = (\mathbf{B}^{\text{T}}\mathbf{B})^{-1}\mathbf{B}^{\text{T}}$ is generalized inverse of $\mathbf{B}$ constructed in Eqn. (3.10). For stable differentiation computation, QR-factorization of $\mathbf{B}^{+} = \mathbf{Q}\mathbf{R}^{*}$ is recommended and least squares RBF-QR differentiation function accordingly is [98, 99]:

$$\tilde{s}_{\mathcal{D}}(\boldsymbol{x}) = \mathbf{E}_{\mathcal{D}}\boldsymbol{\lambda} = \mathbf{E}_{\mathcal{D}}\mathbf{R}^{*-1}\mathbf{Q}^{\text{T}}\hat{\boldsymbol{f}}^{\text{c}}. \tag{3.13}$$

After constructing evaluation matrices and calculating the least squares RBF-QR approximation functions, flow fields can be approximated by RBF-QR model functions.

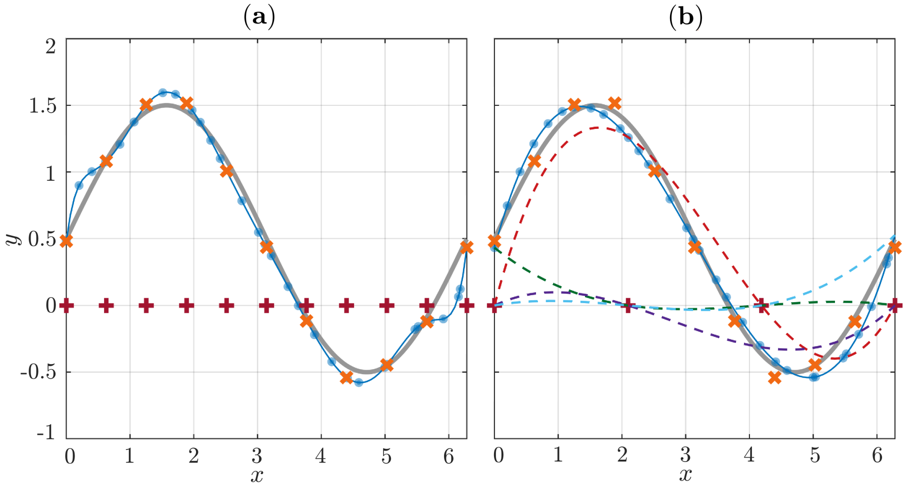


Figure 3.2: A demonstration of least squares RBF-QR fitting using centers and reference points in one dimension. Gray solid curves: the ground truth from exact solutions; blue solid curves: RBF-QR reconstruction results; dashed curves: least squares RBF-QR bases; orange crosses: given data that are sampled from the ground truth with random perturbation; scarlet crosses: the x coordinates of the reference points; blue dots: reconstruction results at the evaluation points. (a): 1D RBF-QR interpolation (a special case of least squares fitting when reference points and centers share the same x coordinates); (b): 1D least squares RBF-QR fitting. Note that only the x coordinates of the reference points are marked for visualization. In the least squares fitting, only the coordinates of the reference points are required (e.g., $x = \pi$ in this case). However, both coordinates and values are used for the centers and evaluation points (e.g., $(x, y) = (\pi, 0.5)$ in this case).

Up to now, three types of points are introduced, i.e., centers, evaluation points, and reference points. These three types of points play different roles in the computation. Data sets in one dimension (1D) are used to demonstrate the concepts of these points (see Figure 3.2 for illustration). Centers $\hat{\boldsymbol{x}}^{\mathrm{c}}$ (orange crosses) are the location where the data are measured. The locations of centers are determined by experimental measurement and are usually 'randomly' distributed in the domain in the case of an LPT experiment. $\hat{\boldsymbol{f}}^{\mathrm{c}}$ are the data measured at centers $\hat{\boldsymbol{x}}^{\mathrm{c}}$. Reference points $\boldsymbol{x}^{\mathrm{ref}}$ (scarlet crosses) are imaginary points that are intentionally added to the domain. A linear combination of basis functions centered at the reference points (see dashed curves in Figure 3.2(b)) is expected to approximate the data located at centers. The number of reference points should be smaller than that of the centers, corresponding to an over-sampling ratio larger than unity, to implement a regression. This linear combination of these basis functions results in a continuous function (i.e., the approximation function $\tilde{s}(\boldsymbol{x})$, see the blue solid curves) that allows evaluating at an arbitrary location in the domain. The location where $\tilde{s}(\boldsymbol{x})$ is evaluated is the location of evaluation points $\boldsymbol{x}^{\mathrm{eva}}$. There is no limitation to the number or the location of the evaluation points. If the evaluation points are densely placed in the domain, super-resolution can be achieved (see blue dots). Similarly, down-sampling can be executed on demand. If the evaluation points are placed at the centers, direct evaluation at the Lagrangian data locations is achieved. All these three types of points work together to reconstruct flow fields in the CLS-RBF method. These three types of points are also used in the CLS framework.

Next, the CLS RBF-QR algorithm is constructed based on the least squares RBF-QR with additional constraints. To derive the CLS RBF-QR algorithm, an equality constraint $\mathbf{C}\boldsymbol{\lambda} = \boldsymbol{d}$ is added to Eqn. (3.8):

$$\begin{aligned} \min\ \mathcal{R} &= \sum_{i}^{N} \left\| \tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) - \hat{f}_i^{\mathrm{c}} \right\|^2, \\ \text{subject to } & \mathbf{C}\boldsymbol{\lambda} = \boldsymbol{d}, \end{aligned} \tag{3.14}$$

where $\mathbf{C}$ is a generalized RBF-QR constraint matrix, $\boldsymbol{d}$ is a column vector of corresponding constraint values. The RBF-QR constraint matrix $\mathbf{C}$ is constructed using M reference points $\boldsymbol{x}^{\mathrm{ref}}$ and J constraint points $\boldsymbol{x}^{\mathrm{cst}}$. The constraint points can be arbitrarily placed in the domain to enforce constraints at any location. In this thesis, constraints are enforced at the same locations as the centers, i.e., $\boldsymbol{x}_l^{\mathrm{cst}} = \hat{\boldsymbol{x}}_j^{\mathrm{c}}$, $l = j = 1, 2, \ldots, N$, and $N = J$. The matrix $\mathbf{C}$ can be a constraint matrix $\mathbf{C}_{\mathcal{O}}$ associated with fixed function values and/or a differentiation constraint matrix $\mathbf{C}_{\mathcal{D}}$ for function derivatives[2]:

$$\begin{aligned} \mathbf{C}_{\mathcal{O}} &= C_{\mathcal{O},lj} = \psi\left(\varepsilon, \left\|\boldsymbol{x}_l^{\mathrm{cst}} - \boldsymbol{x}_j^{\mathrm{ref}}\right\|\right) \\ \mathbf{C}_{\mathcal{D}} &= C_{\mathcal{D},lj} = \psi_{\mathcal{D}}\left(\varepsilon, \left\|\boldsymbol{x}_l^{\mathrm{cst}} - \boldsymbol{x}_j^{\mathrm{ref}}\right\|\right), \end{aligned} \tag{3.15}$$

where $j = 1, 2, \ldots, M$, $l = 1, 2, \ldots, J$. To calculate the vector of expansion coefficients $\boldsymbol{\lambda}$ in Eqn. (3.14), a new Lagrangian objective function is formulated by:

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^*) = \mathcal{R} + \boldsymbol{\lambda}^*(\mathbf{C}\boldsymbol{\lambda} - \boldsymbol{d}), \tag{3.16}$$

where $\boldsymbol{\lambda}^* = (\lambda_1^*, \lambda_2^*, \ldots, \lambda_J^*)^{\mathrm{T}}$ is the vector of Lagrangian multipliers. By setting the gradi-

---

[2] If the constraints are applied on the boundary, the $\mathbf{C}_{\mathcal{O}}$ constraint can be thought of as an analogy to Dirichlet boundary conditions (BCs), whereas the $\mathbf{C}_{\mathcal{D}}$ constraint is analogous to Neumann BCs.

ent of $\mathcal{L}$ with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^*$ to zero, respectively, a linear system is obtained:

$$\begin{aligned} \frac{\partial \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^*)}{\partial \boldsymbol{\lambda}} &= \mathbf{0} = \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{B}^{\mathrm{T}}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) \mathbf{B}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) - \hat{\boldsymbol{f}}^{\mathrm{cT}} \mathbf{B}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) + \boldsymbol{\lambda}^* \mathbf{C} \\ \frac{\partial \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^*)}{\partial \boldsymbol{\lambda}^*} &= \mathbf{0} = \mathbf{C}\boldsymbol{\lambda} - \boldsymbol{d} \end{aligned} . \tag{3.17}$$

Note that, here, $\tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) = \mathbf{B}\boldsymbol{\lambda}|_{\hat{\boldsymbol{x}}_i^{\mathrm{c}}}$ is equivalent to $\tilde{s}(\hat{\boldsymbol{x}}_i^{\mathrm{c}}) = \mathbf{B}(\hat{\boldsymbol{x}}_i^{\mathrm{c}})\boldsymbol{\lambda}$ because $\mathbf{B}$ is formulated between centers and reference points, instead of centers and centers. Eqn. (3.17) can be written in a matrix form:

$$\begin{bmatrix} \mathbf{G} & \mathbf{C}^{\mathrm{T}} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\lambda}^* \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \boldsymbol{d} \end{bmatrix}, \tag{3.18}$$

where $\mathbf{G} = \mathbf{B}^{\mathrm{T}}(\hat{\boldsymbol{x}}_i^{\mathrm{c}})\mathbf{B}(\hat{\boldsymbol{x}}_i^{\mathrm{c}})$ is an integrated RBF-QR system matrix and $\mathbf{F} = \mathbf{B}^{\mathrm{T}}(\hat{\boldsymbol{x}}_i^{\mathrm{c}})\hat{\boldsymbol{f}}^{\mathrm{c}}$ is an integrated RBF-QR value matrix. Now, the vector of expansion coefficients $\boldsymbol{\lambda}$ can be extracted from the vector $[\boldsymbol{\lambda}, \boldsymbol{\lambda}^*]^{\mathrm{T}}$ that is calculated in Eqn. (3.18). With the known vector of expansion coefficients $\boldsymbol{\lambda}$, a generalized CLS RBF-QR fitting and differential approximation is completed using Eqns. (3.12) and (3.13), respectively. With the above preparation, a generalized framework based on the CLS and RBF-QR is ready to be extended and applied in a 3D vector field.

3.2.2 3D extension with divergence-free constraints

To apply the generalized CLS RBF-QR framework in 3D flow field reconstruction, an extension of Eqn. (3.18) is necessary. This is because (i) the given values at centers in 3D flow fields are velocity vectors $\mathbf{U} = (u, v, w)^{\mathrm{T}}$ instead of scalars; (ii) the velocity divergence-free constraint in 3D consists of spatial differentiation in all three directions instead of one.

The extension of Eqn. (3.18) is straightforward: each element in the block matrix of Eqn. (3.18) is extended to a new block matrix that contains similar elements for the three directions in a 3D space. On the left-hand side of Eqn. (3.18), a new constraint block matrix $\bar{\mathbf{C}}$ is given by:

$$\bar{\mathbf{C}} = \begin{bmatrix} \mathbf{C}_x & \mathbf{C}_y & \mathbf{C}_z \end{bmatrix}, \tag{3.19}$$

where $\mathbf{C}_x$, $\mathbf{C}_y$, and $\mathbf{C}_z$ are first-order spatial differential constraint matrices in the x, y, and z directions, respectively. The constraint matrices $\mathbf{C}_x$, $\mathbf{C}_y$, and $\mathbf{C}_z$ are calculated by Eqn. (3.15). The matrix $\mathbf{G}$ is extended to a diagonal block matrix $\bar{\mathbf{G}}$:

$$\bar{\mathbf{G}} = \begin{bmatrix} \mathbf{B}^{\mathrm{T}}\mathbf{B} & & \\ & \mathbf{B}^{\mathrm{T}}\mathbf{B} & \\ & & \mathbf{B}^{\mathrm{T}}\mathbf{B} \end{bmatrix}. \tag{3.20}$$

Similarly, on the right-hand side of Eqn. (3.18), the vector of given values at the centers becomes:

$$\hat{\boldsymbol{f}}^{\mathrm{c}} = \left(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}\right)^{\mathrm{T}}, \tag{3.21}$$

where $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ are the column vectors consisting of velocity components at centers in the x, y, and z directions, respectively. For example, in the x direction,

$$\boldsymbol{u} = \left(u_1, u_2, ...u_N\right)^{\mathrm{T}}, \tag{3.22}$$

where N is the number of centers. The matrix $\mathbf{F}$ is extended to

$$\bar{\mathbf{F}} = \begin{bmatrix} \mathbf{B}^{\mathrm{T}} & & \\ & \mathbf{B}^{\mathrm{T}} & \\ & & \mathbf{B}^{\mathrm{T}} \end{bmatrix} \hat{\boldsymbol{f}}^{\mathrm{c}}, \tag{3.23}$$

where $\hat{\boldsymbol{f}}^{\mathrm{c}}$ is from Eqn. (3.21) and $\bar{\mathbf{F}}$ has the dimensions of $3M \times 1$. $\boldsymbol{d}$ is a null column vector with dimensions of $J \times 1$ so that the divergence is zero at the constraint points. These extensions guarantee that the extended Eqn. (3.18) is compatible with a 3D flow field and divergence-free constraints. Other constraints and flow reconstruction tasks in 2D can be implemented in a similar way.

The same as the interpolation and fitting approximation discussed earlier, the vector of expansion coefficients $\boldsymbol{\lambda}$ needs to be calculated and used for the final evaluation. After extending each block in Eqn. (3.18), the vector $\left[\bar{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^*\right]^{\mathrm{T}}$ is calculated by:

$$\begin{bmatrix} \bar{\boldsymbol{\lambda}} \\ \boldsymbol{\lambda}^* \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{G}} & \bar{\mathbf{C}}^{\mathrm{T}} \\ \bar{\mathbf{C}} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \bar{\mathbf{F}} \\ \boldsymbol{d} \end{bmatrix}, \tag{3.24}$$

where $\bar{\boldsymbol{\lambda}}$ is an extended expansion coefficient vector with entries $\bar{\boldsymbol{\lambda}} = (\boldsymbol{\lambda}_x, \boldsymbol{\lambda}_y, \boldsymbol{\lambda}_z)^{\mathrm{T}}$ and the dimensions of $3M \times 1$. $\boldsymbol{\lambda}_x$, $\boldsymbol{\lambda}_y$, and $\boldsymbol{\lambda}_z$ are expansion coefficient vectors in the x, y, and z directions, respectively. Extract $\bar{\boldsymbol{\lambda}}$ from $[\bar{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^*]^{\mathrm{T}}$ and the CLS RBF-QR approximation function $\tilde{s}(\boldsymbol{x})$ and differentiation functions $\tilde{s}_{\mathcal{D}}(\boldsymbol{x})$ are

$$\begin{aligned} \tilde{s}(\boldsymbol{x}) &= \bar{\mathbf{E}}\bar{\boldsymbol{\lambda}} \\ \tilde{s}_{\mathcal{D}}(\boldsymbol{x}) &= \bar{\mathbf{E}}_{\mathcal{D}}\bar{\boldsymbol{\lambda}}' \end{aligned} \tag{3.25}$$

respectively, where $\bar{\mathbf{E}}$ and $\bar{\mathbf{E}}_{\mathcal{D}}$ are extended diagonal block matrices based on $\mathbf{E}$ and $\mathbf{E}_{\mathcal{D}}$ in

Eqn. (3.11):

$$
\begin{aligned}
\bar{\mathbf{E}} &= \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E} & \\ & & \mathbf{E} \end{bmatrix} \\
\bar{\mathbf{E}}_{\mathcal{D}} &= \begin{bmatrix} \mathbf{E}_{\mathcal{D}} & & \\ & \mathbf{E}_{\mathcal{D}} & \\ & & \mathbf{E}_{\mathcal{D}} \end{bmatrix}
\end{aligned}, \tag{3.26}
$$

respectively. Up to this point, a framework based on the CLS and RBF-QR is established for 3D Lagrangian flow fields with divergence-free constraints.

Chapter 4

CLS-RBF Lagrangian Reconstruction

In this chapter, the CLS-RBF Lagrangian reconstruction method will be elaborated, focusing on computing particle trajectories, velocities, and differential quantities based on the CLS and RBF-QR framework introduced in Chapter 3. In Section 4.1, four steps of the CLS-RBF method are introduced. Super-resolution, as a byproduct, and the choice of optimal parameters will also be discussed in Section 4.1. Section 4.2 will focus on computing differential quantities such as strain- and rotation-rate tensors.

4.1 CLS-RBF Lagrangian Reconstruction

In this section, the CLS-RBF Lagrangian reconstruction method and its four steps are presented. Before delving into the details of the CLS-RBF method, the structure of the raw LPT data must be discussed first in the context of this method. The CLS-RBF method concerns using the 'raw' data from LPT systems, in which only the spatial coordinates of each particle in each frame are recorded. In the space dimension, in each frame, a particle location in the flow domain Ω is described as scattered coordinates in space, which is called a spatial center hereafter: $\hat{\boldsymbol{x}}_i^{\mathrm{c}} = (\hat{x}_i^{\mathrm{c}}, \hat{y}_i^{\mathrm{c}}, \hat{z}_i^{\mathrm{c}})$, $i \in \{1, 2, \ldots, N\}$, where N is the number of spatial centers in the domain. The number of spatial centers is equal to the number of particles measured by LPT experiments in the field of view unless down-sampled. In the time dimension, for each particle, a time instant $\hat{t}_\kappa^{\mathrm{c}}$, $\kappa \in \{1, 2, \ldots, N_{\mathrm{trj}}\}$ in a time sequence $\hat{\boldsymbol{t}}^{\mathrm{c}} = (\hat{t}_1^{\mathrm{c}}, \hat{t}_2^{\mathrm{c}}, \ldots, \hat{t}_{N_{\mathrm{trj}}}^{\mathrm{c}})^{\mathrm{T}}$ is called a temporal center, where N_{trj} is the number of frames. The time sequence $\hat{\boldsymbol{t}}^{\mathrm{c}}$ is presumed to have equal interval Δt, i.e., $\Delta t = \hat{t}_{\kappa+1}^{\mathrm{c}} - \hat{t}_\kappa^{\mathrm{c}} = \mathit{const}$, in this thesis for convenience. Hereafter, $\boldsymbol{x}$ is reserved for particle spatial coordinates (e.g.,

$\boldsymbol{x} = (x, y, z)$), and t is reserved for time. In summary, the CLS-RBF method uses raw LPT data that only contain spatial coordinate data $\boldsymbol{x}$ and the corresponding time t as the input for the reconstruction.

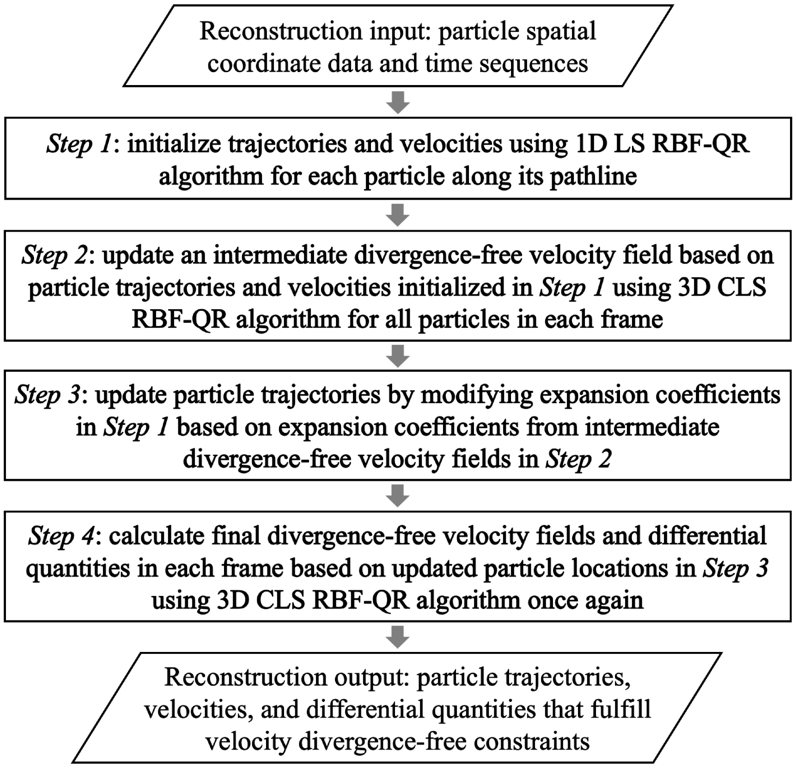


Figure 4.1: A flowchart illustrating the four steps of the CLS-RBF Lagrangian reconstruction method.

Starting with these raw LPT data, the CLS-RBF method reconstructs a 3D flow field via four steps, which are outlined briefly in a flowchart (see Figure 4.1). A diagram[1] is used to illustrate the input and output of each step (see Figure 4.2). As Figure 4.2 shows, the gray dots represent the raw LPT data as the input of the CLS-RBF method. In *Step 1*, for each particle along its pathline, the trajectories (green dashed curves) and velocities (green dashed arrows) are calculated based on the input LPT data (gray dots). This reconstruction directly uses the 1D least squares (LS) RBF-QR without any constraints. The velocities are obtained by taking the first-order temporal differentiation of the trajectories. The intersections of the trajectories and frames are the modified particle locations (green

---

[1] Only four frames in a 2D domain are drawn in this example for demonstration purposes.

crosses). In *Step 2*, for all particles in each frame, based on the modified particle locations (green crosses) and initialized velocities (green dashed arrows), intermediate divergence-free velocity fields (blue solid arrows) are calculated using the 3D CLS RBF-QR algorithm. In *Step 3*, the locations of each particle are updated (indicated by the red crosses), by replacing the expansion coefficients of initialized trajectory functions in *Step 1* with the corresponding expansion coefficients that are based on the intermediate divergence-free velocity fields. In *Step 4*, for all particles in each frame, the final divergence-free velocity fields (red solid arrows) and trajectories (red solid curves) are reconstructed based on the updated particle locations (red crosses) and the intermediate divergence-free velocity fields (blue solid arrows) using the 3D CLS RBF-QR algorithm again. The differential quantities are also output in *Step 4* but they are not shown in Figure 4.2. After these four steps, the CLS-RBF Lagrangian reconstruction is completed. A detailed mathematical description of each step is provided in Sections 4.1.1 – 4.1.4.

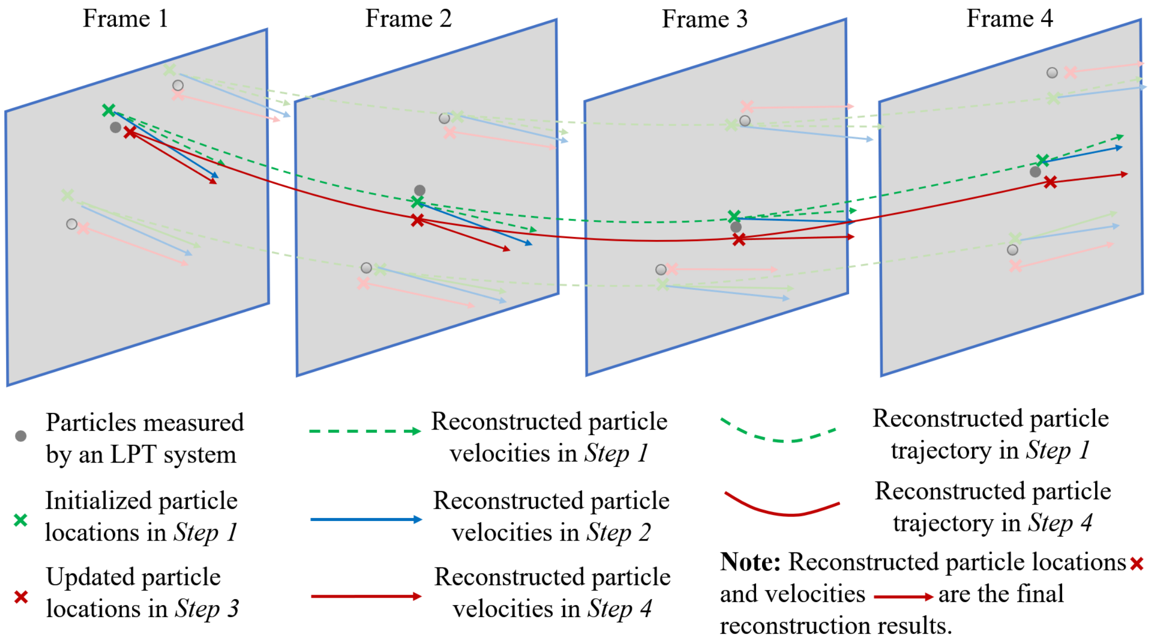


Figure 4.2: A 2D demonstration of the CLS-RBF Lagrangian reconstruction method. The input and output data in each step are represented by symbols (for particle locations), arrows (for velocity vectors), and curves (for particle trajectories).

4.1.1 Step 1: initialize particle trajectory and velocity

The trajectories and velocities are initialized using the one-dimensional (1D) least squares (LS) RBF-QR algorithm for each particle along its pathline. For example, for a specific particle in the x direction, its spatial coordinates[2] $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{N_{\text{trj}}})^{\mathrm{T}}$ are given at N_{trj} temporal centers (frames) $\hat{\boldsymbol{t}}^{\text{c}} = (\hat{t}_1^{\text{c}}, \hat{t}_2^{\text{c}}, \ldots, \hat{t}_{N_{\text{trj}}}^{\text{c}})^{\mathrm{T}}$. Based on Eqn. (3.12), the initialized trajectory function $\tilde{\boldsymbol{x}}(t)$ is calculated by the 1D LS RBF-QR:

$$\tilde{\boldsymbol{x}}(\boldsymbol{t}^{\text{eva}}) = \mathbf{E}(\boldsymbol{t}^{\text{eva}})\boldsymbol{\lambda}^{\text{trj}} = \mathbf{E}(\boldsymbol{t}^{\text{eva}})\mathbf{B}^{+}(\hat{\boldsymbol{t}}^{\text{c}})\hat{\boldsymbol{x}}, \tag{4.1}$$

where $\mathbf{B}^{+}(\hat{\boldsymbol{t}}^{\text{c}}) = (\mathbf{B}^{\mathrm{T}}(\hat{\boldsymbol{t}}^{\text{c}})\mathbf{B}(\hat{\boldsymbol{t}}^{\text{c}}))^{-1}\mathbf{B}^{\mathrm{T}}(\hat{\boldsymbol{t}}^{\text{c}})$ is a generalized inverse of the RBF-QR system matrix $\mathbf{B}(\hat{\boldsymbol{t}}^{\text{c}})$ and $\boldsymbol{t}^{\text{eva}} = (t_1^{\text{eva}}, t_2^{\text{eva}}, \ldots, t_{P_{\text{trj}}}^{\text{eva}})^{\mathrm{T}}$ is the column vector of the temporal evaluation points.[3] The matrix $\mathbf{B}(\hat{\boldsymbol{t}}^{\text{c}})$ is calculated by Eqn. (3.10) and is formulated between M_{trj} temporal reference points $\boldsymbol{t}^{\text{ref}} = (t_1^{\text{ref}}, t_2^{\text{ref}}, \ldots, t_{M_{\text{trj}}}^{\text{ref}})^{\mathrm{T}}$ and N_{trj} temporal centers $\hat{\boldsymbol{t}}^{\text{c}}$ with entries: $B_{ij} = \psi\left(\varepsilon, \left\|\hat{t}_i^{\text{c}} - t_j^{\text{ref}}\right\|\right)$, where $i = 1, 2, \ldots, N_{\text{trj}}$, $j = 1, 2, \ldots, M_{\text{trj}}$. The RBF-QR evaluation matrix $\mathbf{E}(\boldsymbol{t}^{\text{eva}})$ is calculated by Eqn. (3.11) and is mapped between P_{trj} temporal evaluation points $\boldsymbol{t}^{\text{eva}}$ and M_{trj} temporal reference points $\boldsymbol{t}^{\text{ref}}$ with entries: $E_{kj} = \psi\left(\varepsilon, \left\|t_k^{\text{eva}} - t_j^{\text{ref}}\right\|\right)$, where $j = 1, 2, \ldots, M_{\text{trj}}$, $k = 1, 2, \ldots, P_{\text{trj}}$. A temporal oversampling ratio is defined as $\eta_0 = N_{\text{trj}}/M_{\text{trj}}$. $\eta_0 > 1$ is a condition to construct an over-determined system and be solved by the least squares algorithm. The choice of η_0 will be discussed in Section 4.1.6. Now, Eqn. (4.1) can be used to initialize the velocity and acceleration of the same particle in the x direction along the same pathline.

By definition, the velocity and acceleration functions are the first-order and second-order temporal differentiation of a trajectory function, respectively. To initialize them, only the RBF-QR evaluation matrix $\mathbf{E}(\boldsymbol{t}^{\text{eva}})$ needs to be changed to $\mathbf{E}_t(\boldsymbol{t}^{\text{eva}}) = d\mathbf{E}(\boldsymbol{t}^{\text{eva}})/dt$ and $\mathbf{E}_{tt}(\boldsymbol{t}^{\text{eva}}) = d^2\mathbf{E}(\boldsymbol{t}^{\text{eva}})/dt^2$ for the velocity and acceleration evaluation, respectively. The evaluation matrices $\mathbf{E}_t(\boldsymbol{t}^{\text{eva}})$ and $\mathbf{E}_{tt}(\boldsymbol{t}^{\text{eva}})$ are constructed using Eqn. (3.11). Based on Eqn. (4.1), the LS RBF-QR velocity and acceleration functions are given by:

$$\begin{aligned} \tilde{\boldsymbol{u}}(\boldsymbol{t}^{\text{eva}}) &= \mathbf{E}_t(\boldsymbol{t}^{\text{eva}})\boldsymbol{\lambda}^{\text{trj}} = \mathbf{E}_t(\boldsymbol{t}^{\text{eva}})\mathbf{B}^{+}(\hat{\boldsymbol{t}}^{\text{c}})\hat{\boldsymbol{x}} \\ \tilde{\boldsymbol{a}}(\boldsymbol{t}^{\text{eva}}) &= \mathbf{E}_{tt}(\boldsymbol{t}^{\text{eva}})\boldsymbol{\lambda}^{\text{trj}} = \mathbf{E}_{tt}(\boldsymbol{t}^{\text{eva}})\mathbf{B}^{+}(\hat{\boldsymbol{t}}^{\text{c}})\hat{\boldsymbol{x}}, \end{aligned} \tag{4.2}$$

respectively. Initializing trajectories, velocities, and accelerations in the y and z directions follow a similar procedure as that in the x direction. In this thesis, the acceleration of

---

[2] Note that $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{N_{\text{trj}}})^{\mathrm{T}}$ is different from $\hat{\boldsymbol{x}}_i^{\text{c}}$: the vector $\hat{\boldsymbol{x}}$ denotes one particle's spatial coordinates in the x direction along its pathline while $\hat{\boldsymbol{x}}_i^{\text{c}} = (\hat{x}_i^{\text{c}}, \hat{y}_i^{\text{c}}, \hat{z}_i^{\text{c}})$ represents one particle's 3D spatial coordinates in one frame.

[3] Note that, the vector $\boldsymbol{t}^{\text{eva}}$ should at least contain the temporal evaluation points that coincide with all temporal centers; thus, the particle locations in the measured frames are re-calculated as well.

particles will be left for future work. Once the velocities along each pathline are initialized for all particles, the velocity field in each frame is also known in turn. Up to now, initializing particle trajectories and velocities is completed.

During the initialization, spatial centers have been modified. The spatial centers $\hat{\boldsymbol{x}}_i^{\mathrm{c}} = (\hat{x}_i^{\mathrm{c}}, \hat{y}_i^{\mathrm{c}}, \hat{z}_i^{\mathrm{c}})$ in the raw LPT data (gray dots in Figure 4.2) have been adjusted to $\tilde{\boldsymbol{x}}_i^{\mathrm{c}} = (\tilde{x}_i^{\mathrm{c}}, \tilde{y}_i^{\mathrm{c}}, \tilde{z}_i^{\mathrm{c}})$, $i \in \{1, 2, \ldots, N\}$ in *Step 1* (green crosses in Figure 4.2), because of the use of the least squares fitting in Eqn. (4.1). Hereafter, $\tilde{\boldsymbol{x}}_i^{\mathrm{c}}$ is called the modified spatial centers. Accordingly, the output velocity vector (green dashed arrows in Figure 4.2) at a modified spatial center $\tilde{\boldsymbol{x}}_i^{\mathrm{c}}$ in the κ-th frame is written as $\tilde{\mathbf{U}}(t_\kappa)|_{\tilde{\boldsymbol{x}}_i^{\mathrm{c}}} = (\tilde{u}(t_\kappa), \tilde{v}(t_\kappa), \tilde{w}(t_\kappa))^{\mathrm{T}}|_{\tilde{\boldsymbol{x}}_i^{\mathrm{c}}}$, where the velocity components are calculated based on Eqn. (4.2). The modified spatial center $\tilde{\boldsymbol{x}}_i^{\mathrm{c}}$ can be viewed as the intersection of the initialized trajectory (green dashed curves in Figure 4.2) and the κ-th frame. The initialized trajectories passing through the modified spatial centers $\tilde{\boldsymbol{x}}_i^{\mathrm{c}}$ are expected to be smooth and close to the true particle pathlines. From now on, the spatial centers from the raw LPT data are not used anymore and they are replaced by the modified spatial centers.

4.1.2 Step 2: calculate intermediate velocity field

Step 2 computes an intermediate divergence-free velocity field in each frame using the 3D CLS RBF-QR algorithm. The input data of *Step 2* are the modified spatial centers and initialized velocity fields, which both are the output from *Step 1*. To reconstruct the intermediate divergence-free velocity field, all relevant matrices are constructed in each frame using the modified spatial centers. Based on Eqn. (3.10) and in the κ-th frame, an RBF-QR spatial system matrix $\mathbf{B}(\tilde{\boldsymbol{x}}^{\mathrm{c}})$ is formulated based on N modified spatial centers $\tilde{\boldsymbol{x}}_i^{\mathrm{c}}$ and M spatial reference points $\boldsymbol{x}_j^{\mathrm{ref}}$:

$$\begin{aligned} \mathbf{B}(\tilde{\boldsymbol{x}}^{\mathrm{c}}) = B_{ij} &= \psi\left(\varepsilon, \left\|\tilde{\boldsymbol{x}}_i^{\mathrm{c}} - \boldsymbol{x}_j^{\mathrm{ref}}\right\|\right) \\ &= \psi\left(\varepsilon, \left\|(\tilde{x}, \tilde{y}, \tilde{z})_i^{\mathrm{c}} - (x, y, z)_j^{\mathrm{ref}}\right\|\right), \end{aligned} \tag{4.3}$$

where $\tilde{\boldsymbol{x}}^{\mathrm{c}} = [\tilde{\boldsymbol{x}}_1^{\mathrm{c}}, \tilde{\boldsymbol{x}}_2^{\mathrm{c}}, \ldots, \tilde{\boldsymbol{x}}_i^{\mathrm{c}}, \ldots, \tilde{\boldsymbol{x}}_N^{\mathrm{c}}]^{\mathrm{T}}$, $\tilde{\boldsymbol{x}}_i^{\mathrm{c}} = (\tilde{x}, \tilde{y}, \tilde{z})_i^{\mathrm{c}}$ is calculated by Eqn. (4.1) in the κ-th frame; $\boldsymbol{x}_j^{\mathrm{ref}} = (x, y, z)_j^{\mathrm{ref}}$, $i = 1, 2, \ldots, N$, and $j = 1, 2, \ldots, M$. Based on Eqn. (3.15), an RBF-QR spatial differential constraint matrix $\mathbf{C}_{\mathcal{D}}$ is established between J constraint points $\boldsymbol{x}_l^{\mathrm{cst}}$ and M spatial reference points $\boldsymbol{x}_j^{\mathrm{ref}}$:

$$\begin{aligned} \mathbf{C}_{\mathcal{D}} = C_{\mathcal{D},lj} &= \psi_{\mathcal{D}}\left(\varepsilon, \left\|\boldsymbol{x}_l^{\mathrm{cst}} - \boldsymbol{x}_l^{\mathrm{ref}}\right\|\right) \\ &= \psi_{\mathcal{D}}\left(\varepsilon, \left\|(x, y, z)_l^{\mathrm{cst}} - (x, y, z)_j^{\mathrm{ref}}\right\|\right), \end{aligned} \tag{4.4}$$

where $\boldsymbol{x}_l^{\text{cst}} = (x, y, z)_l^{\text{cst}}$ and $l = 1, 2, \ldots, J$. In this step, the constraint points are placed at the same locations as the modified spatial centers. This implies that divergence-free constraints are only enforced at the particle locations measured by LPT experiments. The RBF-QR spatial evaluation matrix $\mathbf{E}(\boldsymbol{x}^{\text{eva}})$ and its differentiation matrix $\mathbf{E}_{\mathcal{D}}(\boldsymbol{x}^{\text{eva}})$ are constructed between P spatial evaluation points $\boldsymbol{x}_k^{\text{eva}}$ and M spatial reference points $\boldsymbol{x}_j^{\text{ref}}$:

$$\begin{aligned} \mathbf{E}(\boldsymbol{x}^{\text{eva}}) = E_{kj} &= \psi\left(\varepsilon, \left\|\boldsymbol{x}_k^{\text{eva}} - \boldsymbol{x}_j^{\text{ref}}\right\|\right) \\ &= \psi\left(\varepsilon, \left\|(x, y, z)_k^{\text{eva}} - (x, y, z)_j^{\text{ref}}\right\|\right), \\ \mathbf{E}_{\mathcal{D}}(\boldsymbol{x}^{\text{eva}}) = E_{\mathcal{D},kj} &= \psi_{\mathcal{D}}\left(\varepsilon, \left\|\boldsymbol{x}_k^{\text{eva}} - \boldsymbol{x}_j^{\text{ref}}\right\|\right) \\ &= \psi_{\mathcal{D}}\left(\varepsilon, \left\|(x, y, z)_k^{\text{eva}} - (x, y, z)_j^{\text{ref}}\right\|\right), \end{aligned} \tag{4.5}$$

where $\boldsymbol{x}^{\text{eva}} = [\boldsymbol{x}_1^{\text{eva}}, \boldsymbol{x}_2^{\text{eva}}, \ldots, \boldsymbol{x}_k^{\text{eva}}, \ldots, \boldsymbol{x}_P^{\text{eva}}]^{\text{T}}$, $\boldsymbol{x}_k^{\text{eva}} = (x, y, z)_k^{\text{eva}}$, and $k = 1, 2, \ldots, P$. In this step, the evaluation points are also placed at the same locations as that of the modified spatial centers, since in *Step 3*, only the modified spatial centers need to be updated. Once the above matrices are constructed, they are used to calculate the expansion coefficients.

To calculate the expansion coefficients, the same matrix extensions outlined in Section 3.2.2 are performed. The extended matrices $\bar{\mathbf{G}}$, $\bar{\mathbf{C}}$, and $\bar{\mathbf{F}}$ in Eqn. (3.24) are constructed based on matrices in Eqns. (4.3) and (4.4). The expansion coefficient $\bar{\boldsymbol{\lambda}}$ is then calculated using Eqn. (3.24). Once $\bar{\boldsymbol{\lambda}}$ is solved, the intermediate divergence-free velocity vector field $\tilde{\mathbf{U}}_\kappa^{\text{div}} = (\tilde{\boldsymbol{u}}_\kappa^{\text{div}}, \tilde{\boldsymbol{v}}_\kappa^{\text{div}}, \tilde{\boldsymbol{w}}_\kappa^{\text{div}})^{\text{T}}$ (blue solid arrows in Figure 4.2) in the κ-th frame is calculated based on Eqn. (3.25):

$$\tilde{\mathbf{U}}_\kappa^{\text{div}}(\boldsymbol{x}^{\text{eva}}) = \bar{\mathbf{E}}\bar{\boldsymbol{\lambda}}, \tag{4.6}$$

where $\bar{\mathbf{E}}$ is constructed based on the matrices in Eqn. (4.5). The column vectors of velocity components $\tilde{\boldsymbol{u}}_\kappa^{\text{div}}$ (i.e., $\tilde{\boldsymbol{u}}_\kappa^{\text{div}} = (\tilde{u}_1^{\text{div}}, \tilde{u}_2^{\text{div}}, \ldots, \tilde{u}_N^{\text{div}})^{\text{T}}|_\kappa$), $\tilde{\boldsymbol{v}}_\kappa^{\text{div}}$, and $\tilde{\boldsymbol{w}}_\kappa^{\text{div}}$ can be extracted from $\tilde{\mathbf{U}}_\kappa^{\text{div}}(\boldsymbol{x}^{\text{eva}})$. A spatial oversampling ratio in *Step 2* is defined as $\eta_1 = N/M$. η_1 should be larger than one to formulate a regression problem to be solved in the least squares sense. The choice of η_1 will be discussed in Section 4.1.6. So far, the intermediate divergence-free velocity field has been computed in one frame. The intermediate velocity fields in the other frames can be calculated in the same way. These intermediate velocity fields will be used to update particle trajectories in *Step 3*.

4.1.3 Step 3: update particle location

In *Step 3*, the particle locations obtained in *Step 1* are updated based on the intermediate divergence-free velocity fields in *Step 2*. The *Step 3* is motivated by the intuition that the velocity initialized in *Step 1* using temporal derivatives along a pathline in a frame should be equal to the intermediate divergence-free velocity reconstructed by *Step 2* in the same frame for the same particle if the flow is incompressible, i.e.,

$$\tilde{\mathbf{U}}(t_\kappa)|_{\hat{\boldsymbol{x}}_i^{\mathrm{c}}} = \tilde{\mathbf{U}}_\kappa^{\mathrm{div}}(\tilde{\boldsymbol{x}}_i^{\mathrm{c}}), \tag{4.7}$$

where $\tilde{\mathbf{U}}(t_\kappa)|_{\hat{\boldsymbol{x}}_i^{\mathrm{c}}}$ is the velocity initialized in *Step 1* and $\tilde{\mathbf{U}}_\kappa^{\mathrm{div}}(\tilde{\boldsymbol{x}}_i^{\mathrm{c}})$ is the velocity calculated in *Step 2*, for the particle i in the κ-th frame. Despite that the initialized velocity is computed by definition in *Step 1*, it is not exactly divergence-free. This is because the initialized trajectories are not necessarily accurate due to the processing and/or the noise in experiments. One idea to improve the initialized trajectories is to introduce some constraints based on physical knowledge. For example, leveraging the divergence-free property of a velocity field, one can update particle trajectories. Therefore, the updated trajectory[4] is expected more accurate than the initial one.

To update trajectories, only the RBF-QR expansion coefficients of trajectories in *Step 1* need to be recalculated. The update in the x direction is used as an example here. First, a linear system is established using a modified trajectory matrix $\tilde{\mathbf{X}}(\hat{\boldsymbol{t}}^{\mathrm{c}})$ and a divergence-free velocity matrix $\tilde{\mathbf{V}}(\tilde{\boldsymbol{x}}^{\mathrm{c}})$:

$$\tilde{\mathbf{X}}(\hat{\boldsymbol{t}}^{\mathrm{c}}) = \boldsymbol{\Lambda}\mathbf{E}(\hat{\boldsymbol{t}}^{\mathrm{c}}), \tag{4.8a}$$

$$\tilde{\mathbf{V}}(\tilde{\boldsymbol{x}}^{\mathrm{c}}) = \boldsymbol{\Lambda}\mathbf{E}_t(\hat{\boldsymbol{t}}^{\mathrm{c}}), \tag{4.8b}$$

where $\boldsymbol{\Lambda}$ is an expansion coefficient matrix to be determined. The matrix $\tilde{\mathbf{X}}(\hat{\boldsymbol{t}}^{\mathrm{c}})$ is calculated by Eqn. (4.1) in *Step 1*. It is based on the modified particle locations and has entries:

$$\tilde{\mathbf{X}}(\hat{\boldsymbol{t}}^{\mathrm{c}}) = \begin{bmatrix} \tilde{x}_1(\hat{t}_1^{\mathrm{c}}) & \tilde{x}_1(\hat{t}_2^{\mathrm{c}}) & \dots & \tilde{x}_1(\hat{t}_{N_{\mathrm{trj}}}^{\mathrm{c}}) \\ \tilde{x}_2(\hat{t}_1^{\mathrm{c}}) & \tilde{x}_2(\hat{t}_2^{\mathrm{c}}) & \dots & \tilde{x}_2(\hat{t}_{N_{\mathrm{trj}}}^{\mathrm{c}}) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}_N(\hat{t}_1^{\mathrm{c}}) & \tilde{x}_N(\hat{t}_2^{\mathrm{c}}) & \dots & \tilde{x}_N(\hat{t}_{N_{\mathrm{trj}}}^{\mathrm{c}}) \end{bmatrix}, \tag{4.9}$$

and dimensions of $N \times N_{\mathrm{trj}}$. The matrix $\tilde{\mathbf{V}}(\tilde{\boldsymbol{x}}^{\mathrm{c}})$ is calculated by Eqn. (4.6) in *Step 2*. It is

---

[4] To distinguish between the particle trajectories obtained in different steps, the trajectories output from *Step 3* are referred to as 'updated' particle trajectory, while the trajectories output from *Step 1* are referred to as 'initialized' particle trajectories. This distinction applies to other particle-related quantities as well, such as 'updated' particle locations (red crosses in Figure 4.2).

based on the corresponding velocity fields and has the entries:

$$\tilde{\mathbf{V}}(\tilde{\boldsymbol{x}}^{\mathrm{c}}) = \begin{bmatrix} \tilde{u}_1^{\mathrm{div}}(\tilde{\boldsymbol{x}}_1^{\mathrm{c}}) & \tilde{u}_2^{\mathrm{div}}(\tilde{\boldsymbol{x}}_1^{\mathrm{c}}) & \dots & \tilde{u}_{N_{\mathrm{trj}}}^{\mathrm{div}}(\tilde{\boldsymbol{x}}_1^{\mathrm{c}}) \\ \tilde{u}_1^{\mathrm{div}}(\tilde{\boldsymbol{x}}_2^{\mathrm{c}}) & \tilde{u}_2^{\mathrm{div}}(\tilde{\boldsymbol{x}}_2^{\mathrm{c}}) & \dots & \tilde{u}_{N_{\mathrm{trj}}}^{\mathrm{div}}(\tilde{\boldsymbol{x}}_2^{\mathrm{c}}) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{u}_1^{\mathrm{div}}(\tilde{\boldsymbol{x}}_N^{\mathrm{c}}) & \tilde{u}_2^{\mathrm{div}}(\tilde{\boldsymbol{x}}_N^{\mathrm{c}}) & \dots & \tilde{u}_{N_{\mathrm{trj}}}^{\mathrm{div}}(\tilde{\boldsymbol{x}}_N^{\mathrm{c}}) \end{bmatrix}, \tag{4.10}$$

and dimensions of $N \times N_{\mathrm{trj}}$. The matrices $\mathbf{E}(\hat{\boldsymbol{t}}^{\mathrm{c}})$ and $\mathbf{E}_t(\hat{\boldsymbol{t}}^{\mathrm{c}})$ are the same as those in Eqns. (4.1) and (4.2).

Second, the linear system in Eqns. (4.8a) and (4.8b) can be integrated into an overdetermined system, using one update expansion coefficient matrix $\boldsymbol{\Lambda}$:

$$\mathbf{H} = \boldsymbol{\Lambda}\mathbf{K}, \tag{4.11}$$

where $\mathbf{K}$ is an integrated trajectory and velocity evaluation matrix

$$\mathbf{K} = \left[\mathbf{E}(\hat{\boldsymbol{t}}^{\mathrm{c}}) \quad \mathbf{E}_t(\hat{\boldsymbol{t}}^{\mathrm{c}})\right], \tag{4.12}$$

with the dimensions of $M_{\mathrm{trj}} \times 2N_{\mathrm{trj}}$, and $\mathbf{H}$ is an integrated trajectory and velocity matrix

$$\mathbf{H} = \left[\tilde{\mathbf{X}}(\hat{\boldsymbol{t}}^{\mathrm{c}}) \quad \tilde{\mathbf{V}}(\tilde{\boldsymbol{x}}^{\mathrm{c}})\right], \tag{4.13}$$

with the dimensions of $N \times 2N_{\mathrm{trj}}$.

Third, the update expansion coefficient matrix $\boldsymbol{\Lambda}$ is solved by:

$$\boldsymbol{\Lambda} = \mathbf{H}\mathbf{K}^{+}, \tag{4.14}$$

where $\mathbf{K}^{+} = (\mathbf{K}^{\mathrm{T}}\mathbf{K})^{-1}\mathbf{K}^{\mathrm{T}}$ is a generalized inverse of $\mathbf{K}$. The update expansion coefficient matrix $\boldsymbol{\Lambda}$ in Eqn. (4.14) has dimensions of $N \times M_{\mathrm{trj}}$ with entries:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_{1,1} & \lambda_{1,2} & \dots & \lambda_{1,M_{\mathrm{trj}}} \\ \lambda_{2,1} & \lambda_{2,2} & \dots & \lambda_{2,M_{\mathrm{trj}}} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{N,1} & \lambda_{N,2} & \dots & \lambda_{N,M_{\mathrm{trj}}} \end{bmatrix}. \tag{4.15}$$

In each row of $\boldsymbol{\Lambda}$, i.e., $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_1^{\mathrm{trj}}, \boldsymbol{\lambda}_2^{\mathrm{trj}}, \dots, \boldsymbol{\lambda}_N^{\mathrm{trj}}]^{\mathrm{T}}$, the expansion coefficients are used to approximate a trajectory for a certain particle, while in each column of $\boldsymbol{\Lambda}$, i.e., $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_1^{\mathrm{frm}}, \boldsymbol{\lambda}_2^{\mathrm{frm}}, \dots, \boldsymbol{\lambda}_{M_{\mathrm{trj}}}^{\mathrm{frm}}]$, the expansion coefficients are used to approximate a velocity field for all particles in a certain frame.

Last, each row of the expansion coefficient matrix $\mathbf{\Lambda}$ is used to update trajectories. For example, for a particle i and in the x direction, an updated trajectory $\tilde{\boldsymbol{x}}_i^{\mathrm{up}}(\boldsymbol{t})$ is:

$$\tilde{\boldsymbol{x}}_i^{\mathrm{up}}(\boldsymbol{t}) = \mathbf{E}(\boldsymbol{t})\boldsymbol{\lambda}_i^{\mathrm{trj}}, \tag{4.16}$$

where $\mathbf{E}(\boldsymbol{t})$ comes from *Step 1*. The updated particle trajectory in the y and z directions follow the same principles as that in the x direction. So far, the particle trajectories are updated in all three directions in *Step 3* and they are expected to be more accurate than those in *Step 1*.

In addition to the computational procedure of *Step 3*, the physical interpretation of Eqns. (4.8a) and (4.8b) are emphasized here. From the temporal perspective, the Lagrangian trajectory of each particle reconstructed by the 1D LS RBF-QR algorithm (right-hand side of Eqn. (4.8a)) should follow the modified particle locations (left-hand side of Eqn. (4.8a), which are the current best estimation based on LPT measurement) as closely as possible. From the spatial perspective, the flow field at each time instant should respect physical constraints such as divergence-free. Therefore, the particle velocities initialized by definition along pathlines (i.e., right-hand side of Eqn. (4.8b), in which velocities are computed by the temporal differentiation of trajectories) should be equal to the divergence-free velocity field reconstructed by the 3D CLS RBF-QR algorithm in each frame (left-hand side of Eqn. (4.8b)). By incorporating both velocities along pathlines and divergence-free velocity fields in frames into one linear system (see Eqn. (4.11)), no 'discrimination' is projected to time and space. The solution of Eqn. (4.11) results in an expansion coefficient matrix $\mathbf{\Lambda}$ that simultaneously respects the temporal and spatial information (experimental measurement and physical constraints) of a flow field. This practice estimates trajectories more accurately than it would be without divergence-free constraints. In summary, *Step 3* is the key step in the CLS-RBF method, which intrinsically connects spatial and temporal dimensions.

4.1.4 Step 4: update final divergence-free velocity field

Step 4 calculates the final divergence-free velocity and differential quantity fields. As the particle locations are updated in *Step 3*, the intermediate velocity fields computed in *Step 2* are no longer divergence-free at the updated particle locations. To resolve this conflict, the 3D CLS RBF-QR algorithm is applied again to compute a final divergence-free velocity field at the updated particle locations. *Step 4* finalizes the calculation of velocities and differential quantities that are both used as the output of the CLS-RBF method.

The overall computation in *Step 4* is similar to that in *Step 2*. *Step 4* employs the updated particle trajectories from *Step 3* and the intermediate divergence-free velocity fields from *Step 2* as input. The velocity gradients are calculated based on Eqn. (3.25). For example, in the x direction, the velocity gradient tensor component $\partial\tilde{u}/\partial x$ in the κ-th frame is given by:

$$\left.\frac{\partial\tilde{u}(\boldsymbol{x})}{\partial x}\right|_{\kappa} = \mathbf{E}_x(\boldsymbol{x})\boldsymbol{\lambda}_{\kappa}, \tag{4.17}$$

where $\boldsymbol{\lambda}_{\kappa}$ is the vector of the expansion coefficient in the x direction and the κ-th frame. The vector $\boldsymbol{\lambda}_{\kappa}$ is calculated in the same way as that in Eqn. (4.6) and is extracted from $\bar{\boldsymbol{\lambda}}$. $\mathbf{E}_x(\boldsymbol{x})$ is the RBF-QR differentiation matrix in the x direction, which is based on Eqn.(4.5). Other velocity gradient tensor components can be calculated by equations similar to Eqn. (4.17), with the corresponding differentiation matrices and expansion coefficients. Calculating other differential quantities (see Section 4.2) is directly based on the velocity gradients. A spatial oversampling ratio in *Step 4* is defined as $\eta_2 = N/M$, where M is the number of spatial reference points in *Step 4* and N is the number of spatial centers. The optimal choice of η_2 will be discussed shortly. After computing the final velocity and differential quantity fields, the Lagrangian flow field reconstruction is completed.

4.1.5 Spatial and temporal super-resolution

The CLS-RBF Lagrangian reconstruction method allows for easy spatial and temporal super-resolution. This means that a divergence-free velocity field is accessible at any time instant at on-demand spacial resolution without any restraints on the location and ‘seeding density’ of pseudo-particles. This method can be used as a reliable tool for LPT data conversion.

To demonstrate the implementation of super-resolution in the CLS-RBF method, a pseudo-particle is placed at an arbitrary location in the domain (say $(x_\mathrm{s}, y_\mathrm{s}, z_\mathrm{s}) \in \Omega$) at any time instant t between the first and last frames measured by LPT experiments. To recover its velocity and differential quantity, *Step 1* – *Step 4* are performed. In *Step 1*, locations and velocities of all measured particles are calculated at time t using Eqns. (4.1) and (4.2), respectively. Next, based on the particle locations and velocities calculated in *Step 1*, an intermediate divergence-free velocity field at time t is reconstructed in *Step 2*. Then, the particle locations at time t are updated in *Step 3* based on the intermediate velocity fields from *Step 2*. Last, the final velocities and differential quantities at time t are calculated based on the updated particle locations from *Step 3* and the intermediate velocity fields from *Step 2*. Because the final velocity and differential quantity field in *Step 4* are described by continuous functions, the velocity and velocity gradient at the location of the pseudo-particle $(x_\mathrm{s}, y_\mathrm{s}, z_\mathrm{s})$ can be calculated. The same procedure can be applied to a pseudo-particle at any other location and time instant. Since the pseudo-particle location and time instant are both arbitrary, any number of pseudo-particles can be placed densely in both space and time, which results in spatiotemporal super-resolution.

4.1.6 Optimal parameter choice

The CLS-RBF method has four free parameters to adjust the performance of reconstruction. These four free parameters are i) the shape factor ε, ii) the temporal oversampling ratio η_0 in *Step 1*, iii) the spatial oversampling ratio η_1 in *Step 2*, and iv) the spatial oversampling ratio η_2 in *Step 4*. Selecting the optimal parameters requires a rough estimation of the trajectory curvature and noise in particle spatial coordinates. Based on the tests, the below suggestions have been made: (a) the shape factor ε can be any small value, such as $0 < \varepsilon \leq 1 \times 10^{-4}$ to ensure high accuracy by virtue of the RBF-QR algorithm. (b) When particle trajectories have high curvatures, a small η_0, such as $\eta_0 \in [1.5, 5]$, can resolve the curved trajectories. (c) When the noise in the particle location is high, large η_0, η_1, and η_2 are preferred for smooth reconstruction and noise suppression. The spatial oversampling ratios η_1 and η_2 can be selected between 2.0 and 10.0. (d) In practice, if smooth velocity and differential quantities are desired, η_2 can be slightly larger than η_1. In summary, the optimal parameter choice is empirical and case-dependent for now. Strategies for improving optimal parameter selection will be discussed in future work (see Chapter 6).

4.2 Differential Quantity Computation

In this section, two types of differential quantities, i.e., strain- and rotation-rate tensors and coherent structure, are discussed. Section 4.2.1 elaborates on strain- and rotation-rate tensor calculation. Section 4.2.2 focuses on identifying coherent structures based on the Q-criterion. Computing these differential quantities is part of *Step 4*, which follows the evaluation of velocity gradients. The CLS-RBF method can also be applied to other quantities based on velocity gradients, such as pressure gradients.

4.2.1 Strain-rate and rotation-rate tensor

Strain-rate and rotation-rate tensors are fluid mechanics quantities that describe the rate of change of a fluid parcel regarding deformation and rotation, respectively. The strain-rate tensor and rotation-rate tensor are the symmetric and anti-symmetric parts of a velocity gradient, respectively. The velocity gradient in 3D is defined as

$$\nabla \mathbf{U} = U_{i,j} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix}, \tag{4.18}$$

where $\mathbf{U} = (u, v, w)^{\mathrm{T}}$ is a velocity vector, $i, j = x, y, z$ by index notation. The velocity gradient can be decomposed by $\nabla \mathbf{U} = \frac{1}{2}(U_{i,j} + U_{j,i}) + \frac{1}{2}(U_{i,j} - U_{j,i})$. Based on the above definitions, the strain-rate tensor $\boldsymbol{S}$ and rotation-rate tensor $\boldsymbol{R}$ are:

$$\begin{aligned} \boldsymbol{S} = S_{ij} = \frac{1}{2}(U_{i,j} + U_{j,i}) \\ \boldsymbol{R} = R_{ij} = \frac{1}{2}(U_{i,j} - U_{j,i}) \end{aligned}. \tag{4.19}$$

The off-diagonal elements of the strain-rate tensor (i.e., S_{ij} $(i \neq j)$) describe the shear deformation of a fluid parcel and the diagonal elements of the strain-rate tensor (i.e., S_{ii}) represent the compression and expansion. The rotation-rate tensor describes the rate of rotation along axes. After all components of the velocity gradient tensor $U_{i,j}$ are calculated in *Step 4*, $\boldsymbol{S}$ and $\boldsymbol{R}$ can be evaluated by Eqn. (4.19).

4.2.2 Coherent structure and Q-criterion

The coherent structure is an orderly flow structure that shows spatiotemporal coherence and is consistent over a sufficiently long time [71, 128, 144]. Vortical structures are perhaps the most important coherent structures, and they can be recognized by vortex identification methods. The Q-criterion [70] is one of the vortex identification methods [106, 182]. The Q-criterion uses the second invariant of the characteristic equation based on eigenvalues of a velocity gradient tensor to identify vortices. The definition of the Q-criterion is [56, 70]:

$$Q = \frac{1}{2}(\|\boldsymbol{R}^2\| - \|\boldsymbol{S}^2\|) > 0, \tag{4.20}$$

where $\boldsymbol{R}$ is the rotation-rate tensor, and $\boldsymbol{S}$ is the strain-rate tensor. $Q > 0$ indicates that the rotation is stronger than non-rotational motion [56, 70, 77, 181]. After $\boldsymbol{S}$ and $\boldsymbol{R}$ are calculated by Eqn. (4.19), vortices can be found using Eqn. (4.20).

Chapter 5

Results and Discussions

In this chapter, validations and verification using the CLS-RBF Lagrangian reconstruction method will be presented. In Section 5.1, synthetic data generation and reconstruction error evaluation are discussed. Section 5.2 elaborates on several baseline algorithms used for benchmarking. Sections 5.3 and 5.4 provide validations and verification of the CLS-RBF method, respectively.

5.1 Synthetic Data and Error Evaluation

5.1.1 Synthetic data generation

To simulate LPT data measured by experiments, synthetic data are generated by adding artificial noise to the ground truth based on exact solutions. The Taylor-Green vortex (TGV) and Arnold-Beltrami-Childress (ABC) flow are adopted as exact solutions for flow fields. Both the TGV and ABC flows are governed by the incompressible Navier-Stokes equations, which will be detailed in each validation case. The ground truth is essentially particle trajectory data that consist of particle spatial coordinates and their corresponding time information. To generate such trajectories, the velocity of a particle is integrated using a small time increment along its pathline starting from an initial location. A certain number of particles are randomly placed in the flow domain Ω in the first snapshot. The particle locations in the $(\kappa+1)$-th snapshot are calculated by the forward Euler method with small time step δt:

$$\boldsymbol{x}_{\kappa+1} = \boldsymbol{x}_{\kappa} + \delta t \cdot \mathbf{U}_{\kappa}, \tag{5.1}$$

where $\boldsymbol{x}_{\kappa+1} = (x_{\kappa+1}, y_{\kappa+1}, z_{\kappa+1})$ is the particle location in the $(\kappa+1)$-th snapshot, $\boldsymbol{x}_\kappa = (x_\kappa, y_\kappa, z_\kappa)$ is the particle location in the κ-th snapshot, $\mathbf{U}_\kappa = (u_\kappa, v_\kappa, w_\kappa)$ is the velocity evaluated by the exact solution in the κ-th snapshot. After the integration, particle trajectories based on small time intervals between snapshots (usually the number of snapshots is more than ten thousand) are obtained with high spatial and temporal accuracy. The above trajectory is down-sampled to generate the ground truth for the synthetic LPT data. A certain number of frames with an equal time interval Δt is chosen from the snapshots. In this thesis, the number of frames is chosen to be 11. The frame interval is much larger than that in the snapshots (i.e., $\Delta t \gg \delta t$), and this practice simulates the real LPT experiments with finite resolution in time. So far, the down-sampled ground truth is obtained and they can be imposed with artificial noise.

The artificial noise is added to the particle spatial coordinates of the ground truth. The artificial noise is zero-mean Gaussian noise whose standard deviation σ is proportional to the spatial span of the domain in one direction. For example, in the x direction, the standard deviation σ of the noise is $\sigma = \xi L$, where ξ is the noise level, L is the spatial span of the domain in the x direction. The standard deviation of artificial noise in the y and z directions is assumed the same as that in the x direction for convenience in the scope of this thesis. $\xi = 0.1\%$ or $\xi = 1\%$ are chosen to represent typical low and high noise levels in an LPT experiment. Hereafter, the two noise levels are called 0.1% and 1% noise, respectively. A flowchart of the synthetic data generation is presented in Figure 5.1. Up to now, the synthetic data can be used to validate the CLS-RBF method.

5.1.2 Error evaluation

The relative errors in particle location, velocity, and velocity gradient reconstruction are quantified to assess reconstruction quality. The relative error $\mathcal{E}$ is defined as:

$$\mathcal{E} = \frac{\left|\tilde{f} - f_0\right|}{\|f_0\|_{L^\infty(\Omega)}} \times 100\% \tag{5.2}$$

where f_0 is the ground truth based on the exact solution and $\tilde{f}$ is the reconstruction result, Ω is the computation domain. The statistics (i.e., mean, standard derivation, and maximum) of the logarithm of the relative error ($\log_{10} \mathcal{E}$) are also evaluated for errors with large variation.

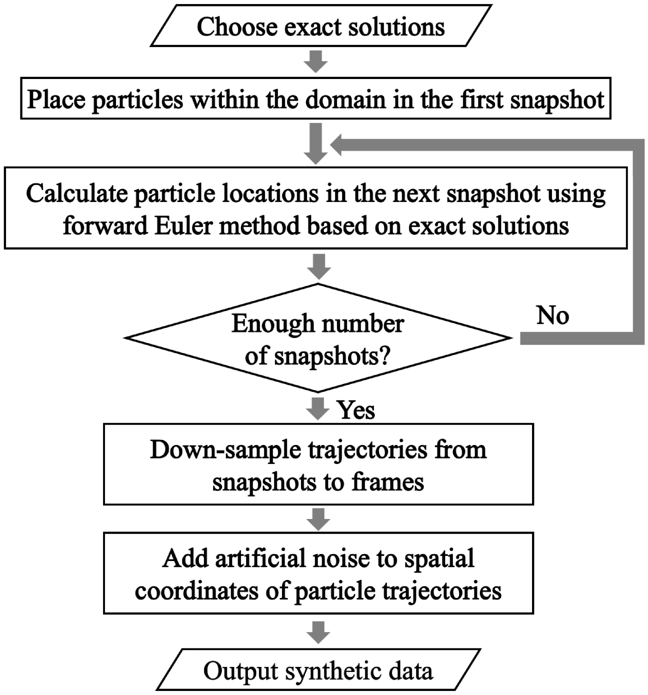


Figure 5.1: A flowchart of synthetic data generation.

5.2 Baseline Algorithms

Particle trajectories and velocities reconstructed by six 'baseline' algorithms are used for benchmarking. Most of the baseline algorithms have been widely used in the LPT community for trajectory and velocity reconstruction. The six baseline algorithms are: (i) first-order finite difference method (1st FDM) [79, 80], (ii) second-order finite difference method (2nd FDM) [10, 111], (iii) second-order least squares polynomial fitting (2nd LS-POLY) [29, 130], (iv) third-order least squares polynomial fitting (3rd LS-POLY) [29, 63], (v) fourth-order least square polynomials fitting (4th LS-POLY) [43], and (vi) ninth-order least squares polynomial fitting (9th LS-POLY). Note that, the 9th LS-POLY method has not yet been used to reconstruct particle trajectories and velocities in the LPT community. In this thesis, it has been chosen to compare reconstruction results with the CLS-RBF reconstruction especially when the trajectory has high curvature. The detailed implementation of these baseline algorithms can be found in Appendix A.

5.3 Validation

Section 5.3.1 discusses the reconstruction parameters that were used for synthetic data generation and Lagrangian flow field reconstruction. Section 5.3.2 analyzes trajectory and velocity reconstruction results. Section 5.3.3 discusses differential quantity reconstruction results.

5.3.1 Synthetic flows and parameters

For the synthetic data generation, a specific exact solution of a velocity field is determined. In the 2D validation, the Taylor-Green vortex (TGV) [165] is adopted. The velocity components of the 2D TGV are given by [165]:

$$\begin{aligned} u &= \alpha_1 \sin(\omega x)\cos(\omega y) \\ v &= \alpha_2 \cos(\omega x)\sin(\omega y) \end{aligned}, \tag{5.3}$$

where $\alpha_1 = 1$ and $\alpha_2 = -1$ are the amplitudes and $\omega = 2\pi$ is the spatial frequency of the 2D TGV. In the 3D validation, the TGV and ABC flow fields are used. The velocity components of the 3D TGV flow field are given by:

$$\begin{aligned} u =& \alpha_1 \cos(\omega(x-d_x))\sin(\omega(y-d_y))\sin(\omega(z-d_z)) \\ v =& \alpha_2 \sin(\omega(x-d_x))\cos(\omega(y-d_y))\sin(\omega(z-d_z)), \\ w =& \alpha_3 \sin(\omega(x-d_x))\sin(\omega(y-d_y))\cos(\omega(z-d_z)) \end{aligned} \tag{5.4}$$

where $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, $\alpha_3 = -1$, and $\omega = 2\pi$; $d_x = 0.25$, $d_y = 0.25$ and $d_z = 0.25$ are displacements of the TGV flow structure in the x, y and z directions, respectively. The velocity components of the 3D ABC flow are given by [37, 183]:

$$\begin{aligned} u &= \alpha[\sin(\omega z) + \cos(\omega y)] \\ v &= \alpha[\sin(\omega x) + \cos(\omega z)], \\ w &= \alpha[\sin(\omega y) + \cos(\omega x)] \end{aligned} \tag{5.5}$$

where $\alpha = 0.1$ and $\omega = 8\pi$. Following the steps described in Section 5.1.1, synthetic data are generated. The corresponding velocity functions (e.g., Eqns. (5.3), (5.4), and (5.5)) are used to calculate velocity fields in Eqn. (5.1). The parameters for synthetic data generation of each validation can be found in Table 5.1. The parameters used for CLS-RBF Lagrangian reconstruction are listed in Table 5.2. With these parameters, the CLS-RBF Lagrangian reconstruction can be conducted in 2D and 3D validations.

Table 5.1: Parameters for synthetic data generation.

Case	N	$x \times y \times z$	Δt	δt	N_{sns}
2D TGV	500	$[0, 1]^2$	0.1	5×10^{-7}	2×10^6
3D TGV	700	$[0, 0.5]^3$	0.1	5×10^{-6}	2×10^5
3D ABC	800	$[0, 0.5]^3$	0.05	5×10^{-6}	1×10^5

N: number of particles in each frame
$x \times y \times z$: domain dimensions, e.g., $x \times y \times z \in [0, 1]^3 = [0, 1] \times [0, 1] \times [0, 1]$
Δt: time interval between two consecutive frames
δt: time interval between two consecutive snapshots
N_{sns}: number of snapshots

Table 5.2: Parameters for CLS-RBF Lagrangian reconstruction.

Case	ξ	ε	η_0	η_1	η_2
2D TGV	0.1%	1×10^{-4}	11/9	6	6
	1.0%	1×10^{-4}	11/6	6	8
3D TGV	0.1%	1×10^{-4}	11/8	5	9
	1.0%	1×10^{-4}	11/6	8	8
3D ABC	0.1%	1×10^{-4}	11/5	2.5	2.5
	1.0%	1×10^{-4}	11/4	3	3

ξ: noise level
ε: shape factor
η_0: temporal oversampling ratio
η_1: spatial oversampling ratio in *Step 2*
η_2: spatial oversampling ratio in *Step 4*

5.3.2 Trajectory and velocity reconstruction

The trajectory reconstruction results using the CLS-RBF method and baseline algorithms are analyzed qualitatively in Figures 5.2 – 5.6. Figure 5.2 illustrates the trajectory reconstruction with two noise levels in the 2D TGV flow field. As shown in this figure, the CLS-RBF method can reconstruct trajectories with no significant deviation from the ground truth regardless of the noise levels. However, high-order polynomial fitting (e.g., 9th LS-POLY) showed numerical oscillation, especially at two ends of a trajectory when the noise level was high. Reconstructed trajectories based on low-order polynomial fitting (e.g., 2nd LS-POLY and 3rd LS-POLY) had significant deviations from the ground truth, particularly for trajectories with high curvature variation.

Figures 5.3 and 5.4 represent the reconstruction results in the 3D TGV flow field. As shown in these two figures, the trajectories reconstructed by the CLS-RBF method closely matched the ground truth. In contrast, the baseline algorithms exhibited significant deviations for low-order polynomial fitting and numerical oscillation for high-order polynomial fitting.

Figures 5.5 and 5.6 present the reconstruction results in the 3D ABC flow field. It is noteworthy that the trajectories of particles in the ABC flows differ from those in the TGV flow fields. The former exhibit low curvature and approximate straight lines, while the latter are characterized by high curvature variations.[1] As shown in these two figures, the CLS-RBF method and low-order polynomial fitting can reconstruct trajectories closely matching the ground truth. On the contrary, the high-order polynomial fitting exhibited numerical oscillations along trajectories, which was further exacerbated by the high noise level. In summary, the CLS-RBF method outperforms the baseline algorithms that suffer from various issues in trajectory reconstruction. The CLS-RBF method can reconstruct trajectories close to the ground truth, regardless of the noise levels and trajectory curvature in both 2D and 3D flow fields.

---

[1] Another feature of the ABC flows is that: particles in the ABC flows tend to move out of the domain, in contrast to those in the TGV fields that are restrained to the flow domain. Although particles outside the domain move into the domain and replenish the empty space, they will not be shown because they are absent in the first snapshot. This feature explains why the central area in the ABC flows appears hollowed out but the flows are still considered incompressible.

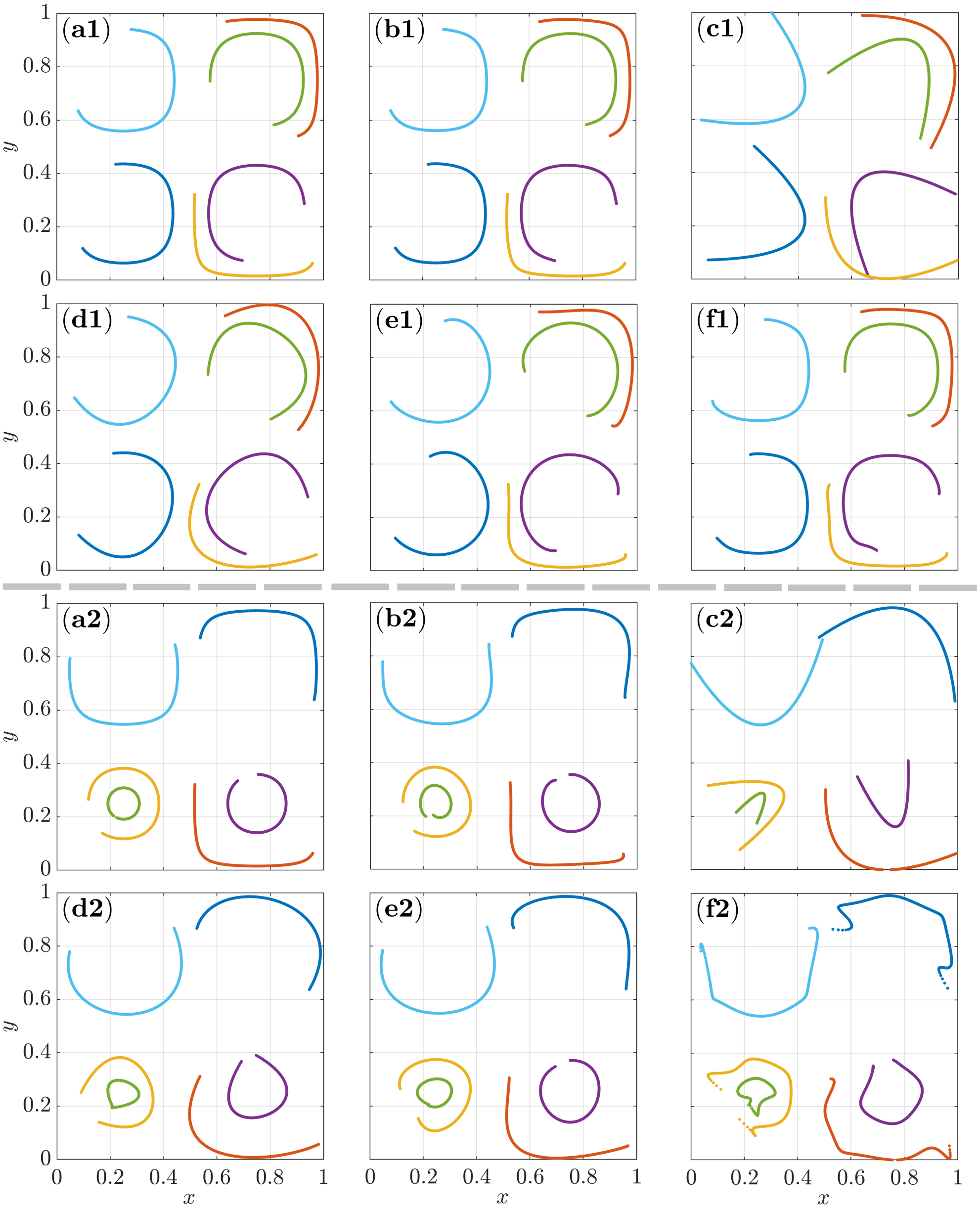


Figure 5.2: Particle trajectory super-resolution reconstruction in the 2D validation using different methods based on the synthetic data with two noise levels. Only six trajectories out of 500 are shown in sub-figures. (a1) & (a2): the ground truth; (b1) & (b2): CLS-RBF method; (c1) & (c2): 2nd LS-POLY; (d1) & (d2): 3rd LS-POLY; (e1) & (e2): 4th LS-POLY; (f1) & (f2): 9th LS-POLY. (a1) – (f1): reconstruction based on the synthetic data with 0.1% noise. (a2) – (f2): reconstruction based on the synthetic data with 1% noise.

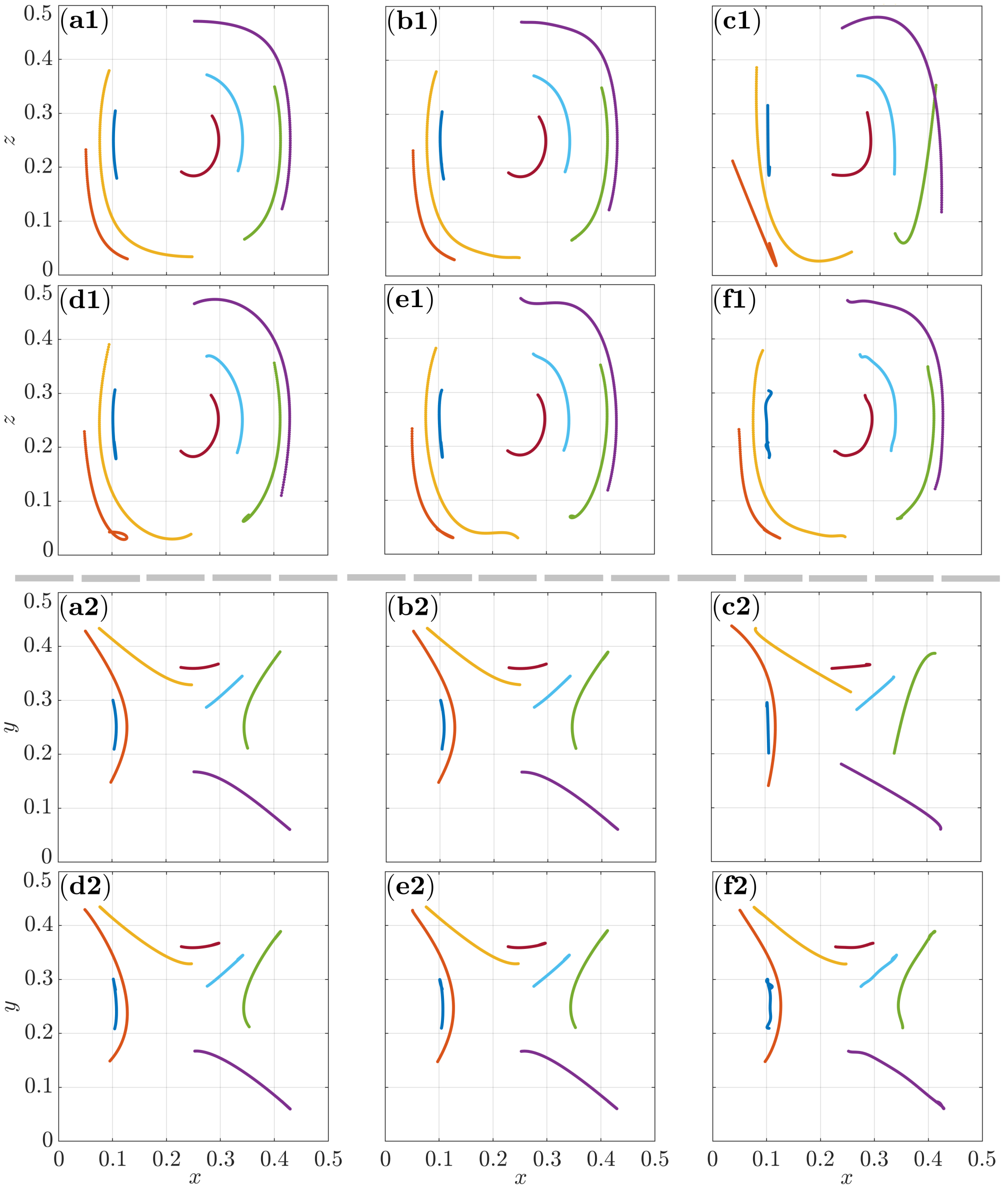


Figure 5.3: Particle trajectory super-resolution reconstruction in the 3D TGV validation using different methods based on the synthetic data with 0.1% noise. Only seven trajectories out of 700 are shown in each sub-figure. (a1) & (a2): the ground truth; (b1) & (b2): CLS-RBF method; (c1) & (c2): 2nd LS-POLY; (d1) & (d2): 3rd LS-POLY; (e1) & (e2): 4th LS-POLY; (f1) & (f2): 9th LS-POLY. (a1) – (f1): views from the $-y$ axis; (a2) – (f2): views from the $+z$ axis.

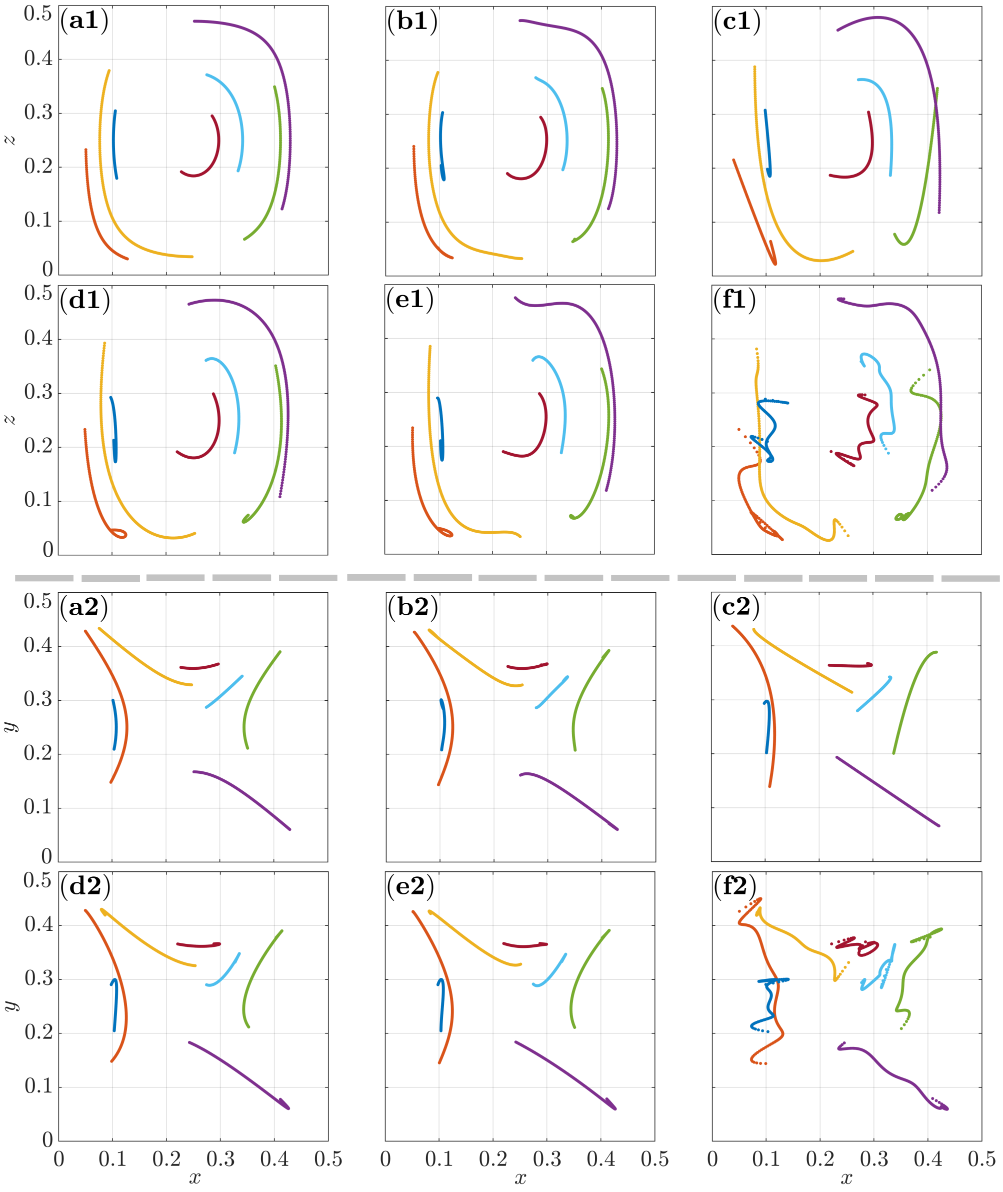


Figure 5.4: Particle trajectory super-resolution reconstruction in the 3D TGV validation using different methods based on the synthetic data with 1% noise. Only seven trajectories out of 700 are shown in each sub-figure. (a1) & (a2): the ground truth; (b1) & (b2): CLS-RBF method; (c1) & (c2): 2nd LS-POLY; (d1) & (d2): 3rd LS-POLY; (e1) & (e2): 4th LS-POLY; (f1) & (f2): 9th LS-POLY. (a1) – (f1): views from the $-y$ axis; (a2) – (f2): views from the $+z$ axis.

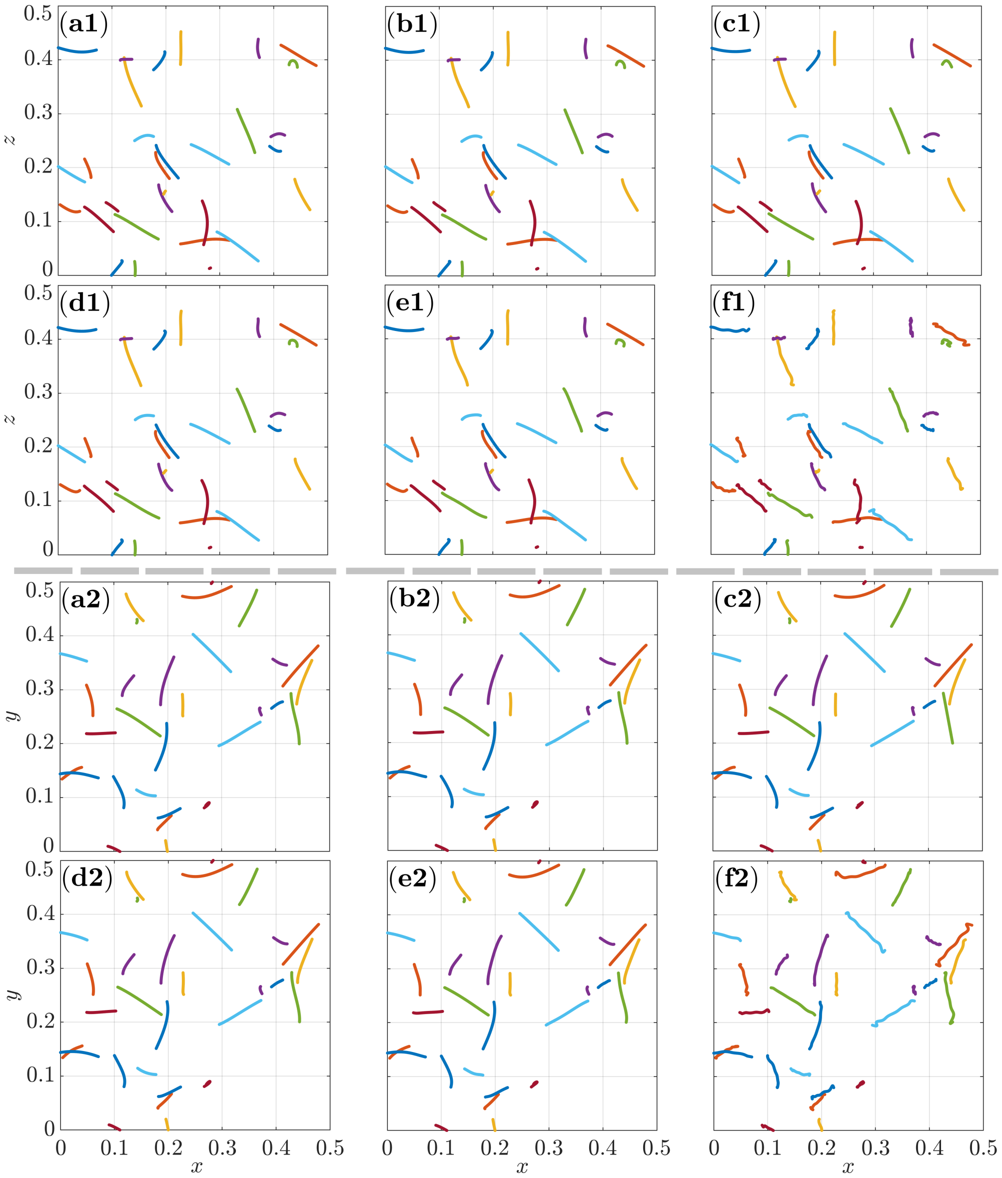


Figure 5.5: Particle trajectory super-resolution reconstruction in the 3D ABC validation using different methods based on the synthetic data with 0.1% noise. About 30 trajectories out of 800 are shown in each sub-figure. (a1) & (a2): the ground truth; (b1) & (b2): CLS-RBF method; (c1) & (c2): 2nd LS-POLY; (d1) & (d2): 3rd LS-POLY; (e1) & (e2): 4th LS-POLY; (f1) & (f2): 9th LS-POLY. (a1) – (f1): views from the $-y$ axis; (a2) – (f2): views from the $+z$ axis.

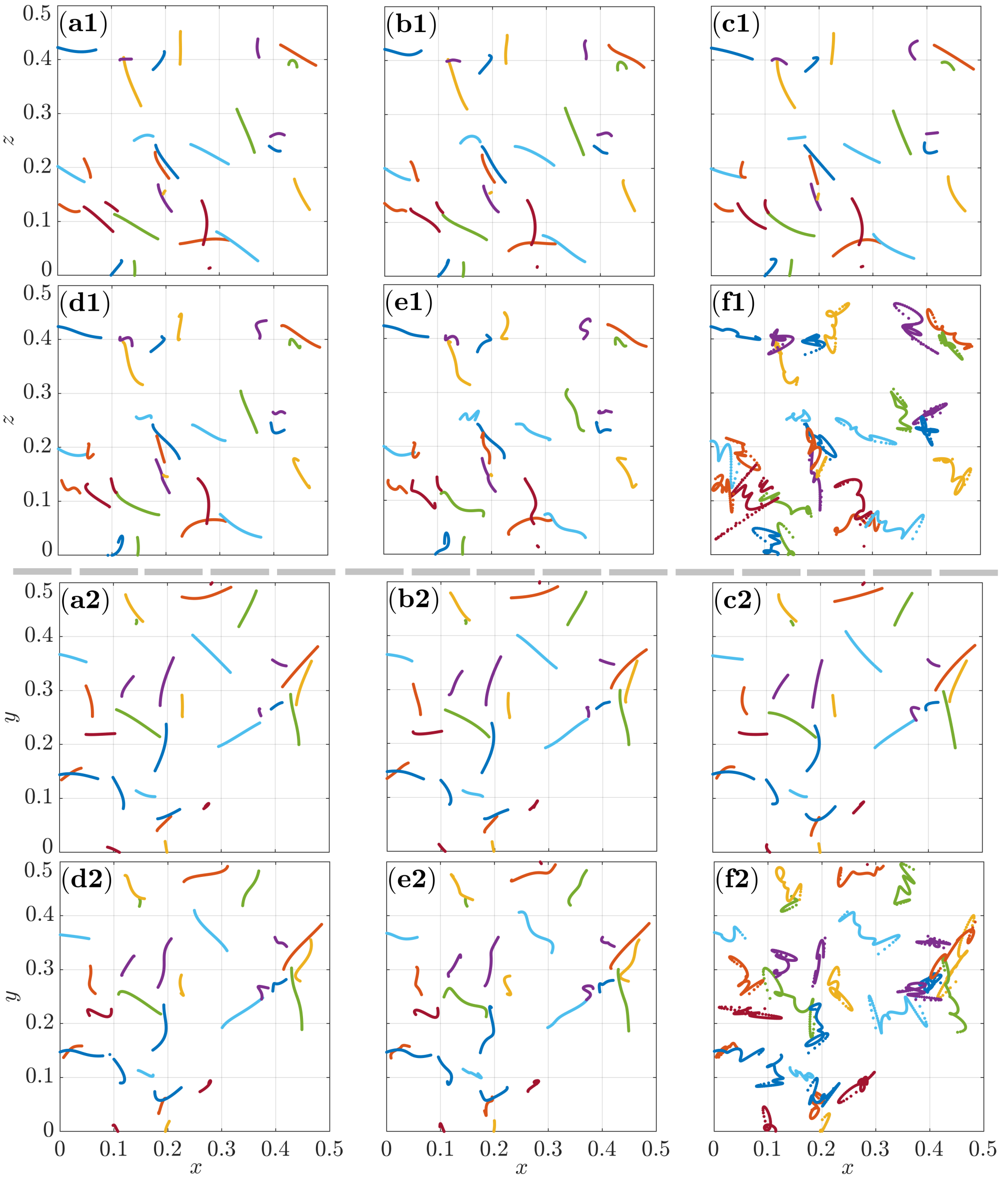


Figure 5.6: Particle trajectory super-resolution reconstruction in the 3D ABC validation using different methods based on the synthetic data with 1% noise. About 30 trajectories out of 800 are shown in each sub-figure. (a1) & (a2): the ground truth; (b1) & (b2): CLS-RBF method; (c1) & (c2): 2nd LS-POLY; (d1) & (d2): 3rd LS-POLY; (e1) & (e2): 4th LS-POLY; (f1) & (f2): 9th LS-POLY. (a1) – (f1): views from the $-y$ axis; (a2) – (f2): views from the $+z$ axis.

Trajectory and velocity reconstruction results are analyzed quantitatively in Figures 5.7 – 5.12. Red solid lines in these figures represent the results reconstructed by the CLS-RBF method; green dashed and solid lines indicate results of the 1st FDM and 2nd FDM, respectively;[2] orange dashed and solid lines represent the 2nd LS-PLOY and 3rd LS-POLY, respectively; blue dashed and solid lines are plotted according to the 4th LS-POLY and 9th LS-POLY, respectively. From Figures 5.7 – 5.12, it is observed that the CLS-RBF method can almost always reconstruct trajectories and velocities with lower error (regarding the mean, standard deviation, and maximum relative errors) than those based on the baseline algorithms, regardless of the noise levels. Although in the first and last frames, i.e., at the two ends of trajectories, the relative errors increased, they were still lower than those of the baseline algorithms. Furthermore, the CLS-RBF method effectively mitigated noise in the spatial coordinates of the particles obtained from the synthetic data. This was evidenced by that the red lines, which represent the CLS-RBF reconstruction results, lie below the green lines representing the original synthetic data.

---

[2] Note that the green dashed and green solid lines are overlapped in Figure 5.7 and 5.12(a1) – (a3) and (b1) – (b3) because both of the finite difference methods only evaluate the velocity and do not correct particle trajectories. Therefore, the particle locations from the synthetic data are directly used as the trajectory output of the finite difference methods.

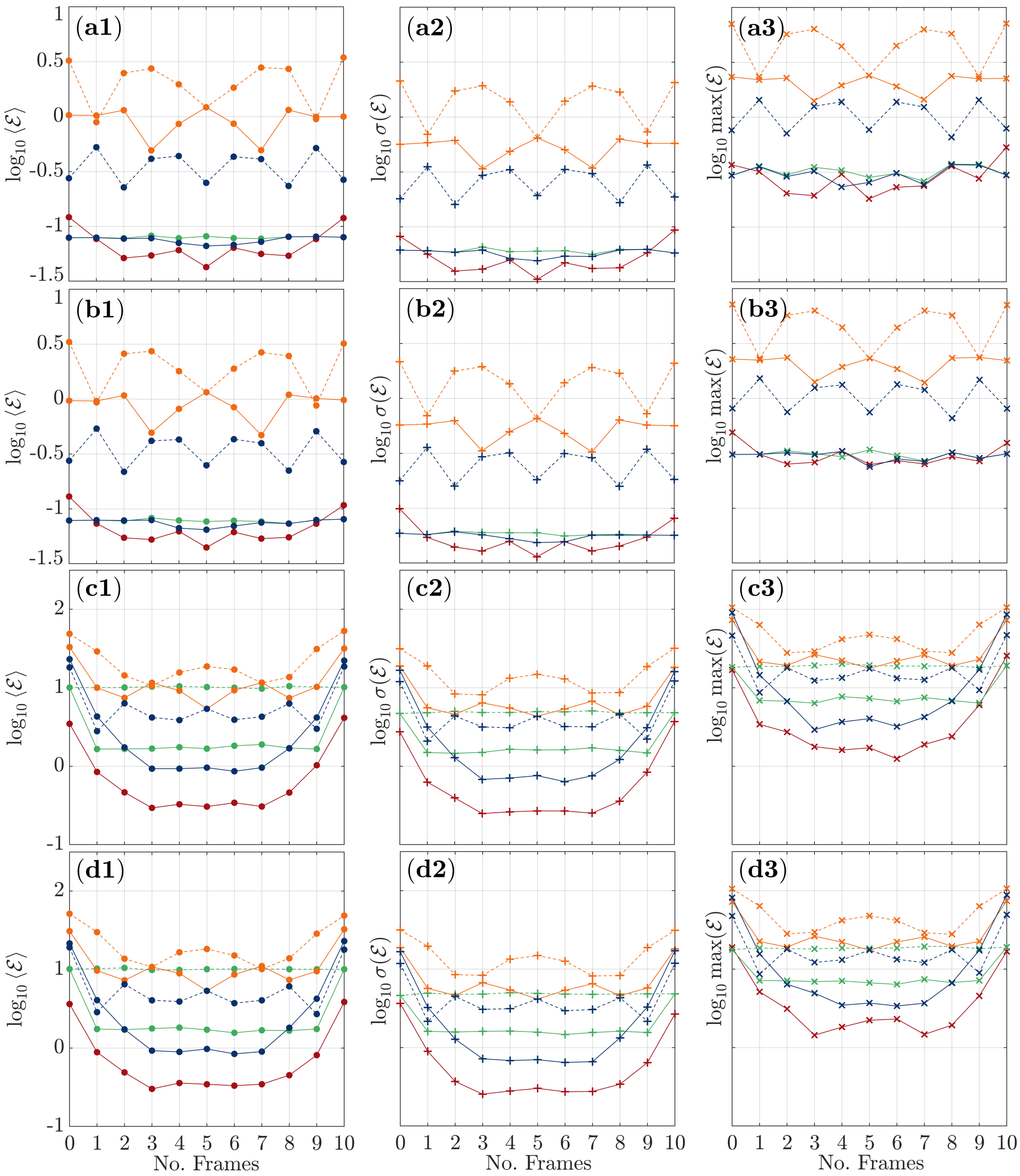


Figure 5.7: Reconstruction error comparison using different methods in the 2D validation based on the synthetic data with 0.1% noise. Left column to right: the mean, standard deviation, and maximum of the relative error, respectively. Top row to bottom: reconstruction error of particle locations in the x and y coordinates, and the velocity components u and v, respectively.

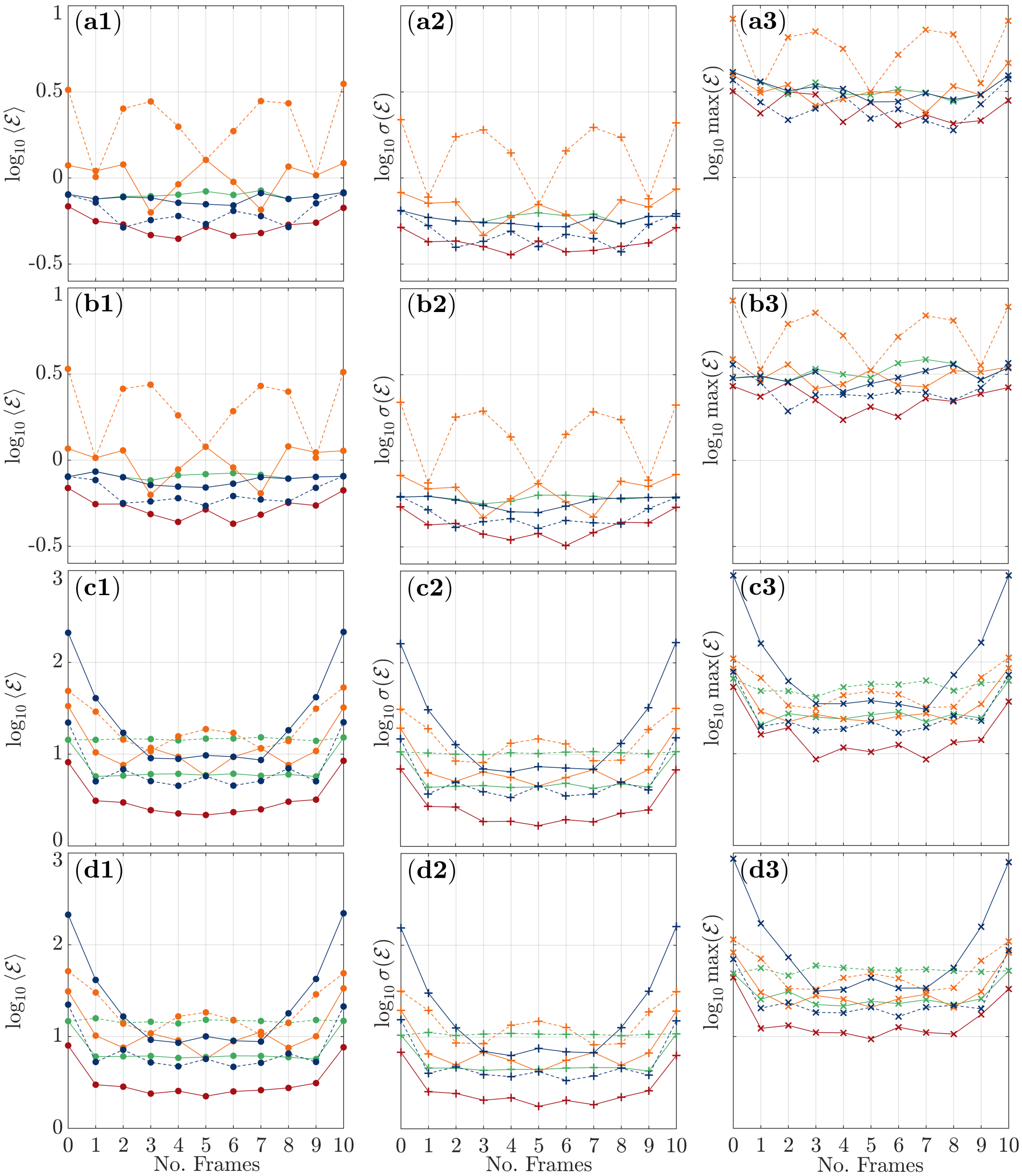


Figure 5.8: Reconstruction error comparison using different methods in the 2D validation based on the synthetic data with 1% noise. Left column to right: the mean, standard deviation, and maximum of the relative error, respectively. Top row to bottom: reconstruction error of particle locations in the x and y coordinates, and the velocity components u and v, respectively.

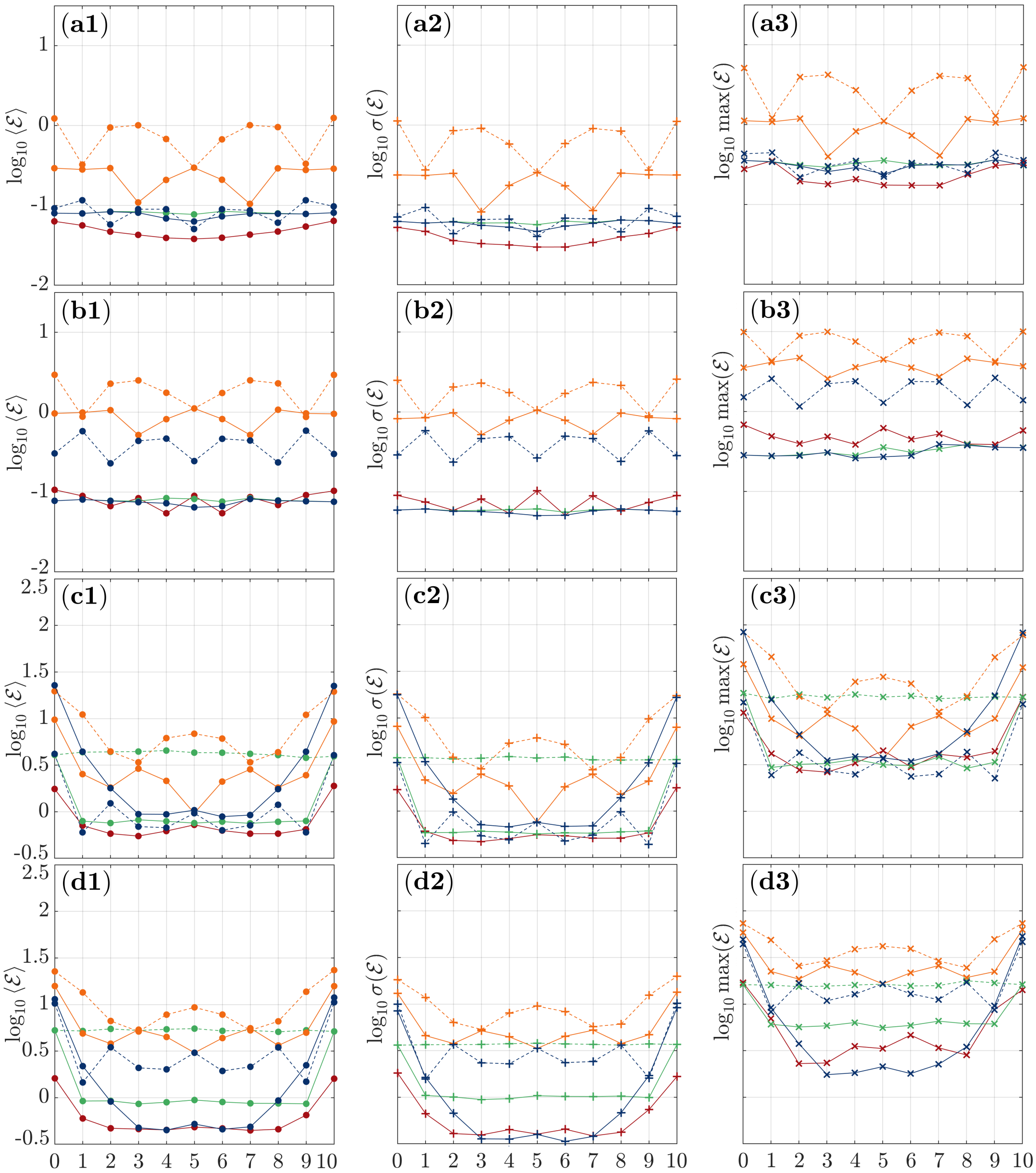


Figure 5.9: Reconstruction error comparison using different methods in the 3D TGV validation based on the synthetic data with 0.1% noise. Left column to right: the mean, standard deviation, and maximum of the relative error, respectively. Top row to bottom: reconstruction error of particle locations in the x and z coordinates, and the velocity components u and w, respectively.

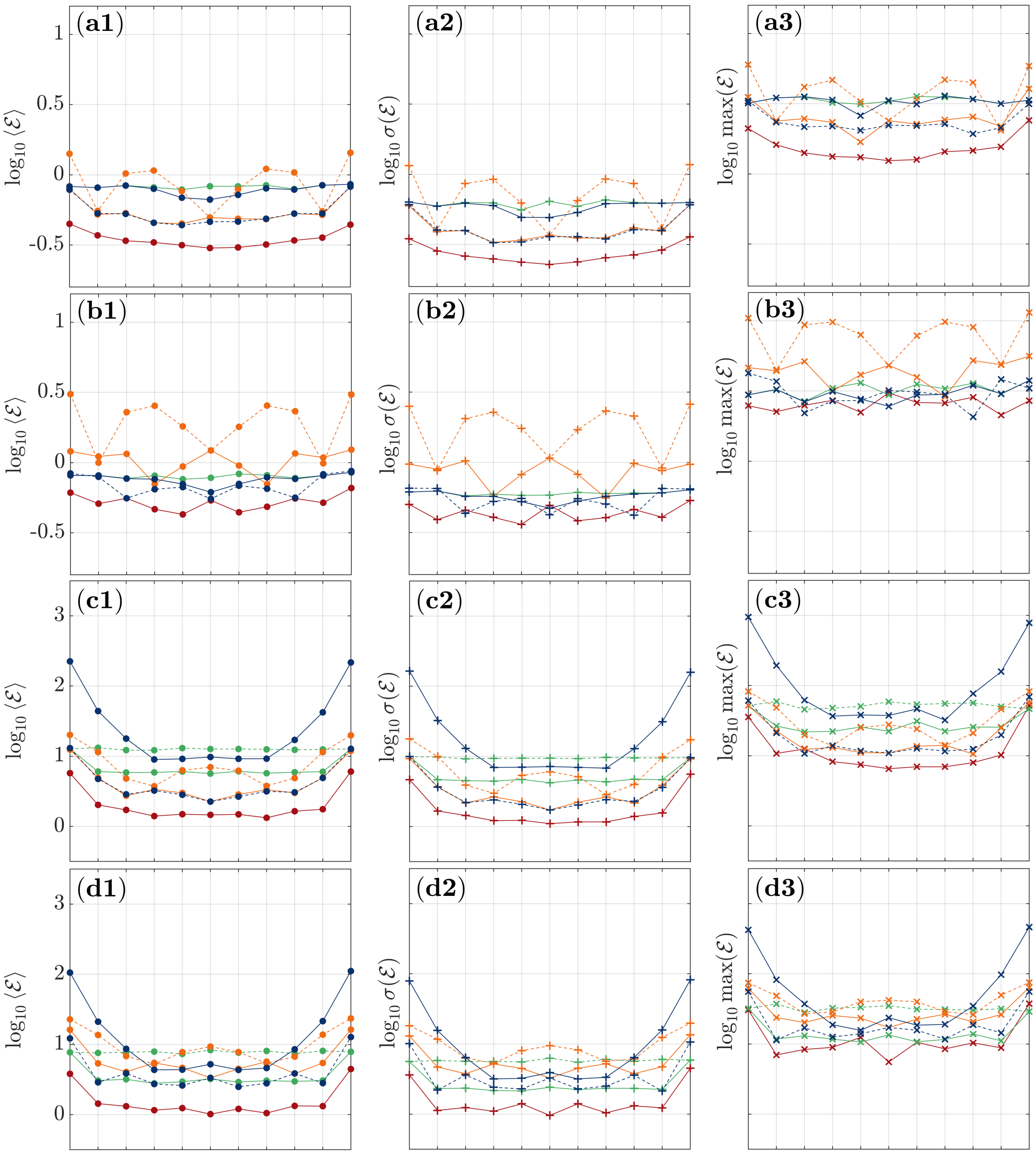


Figure 5.10: Reconstruction error comparison using different methods in the 3D TGV validation based on the synthetic data with 1% noise. Left column to right: the mean, standard deviation, and maximum of the relative error, respectively. Top row to bottom: reconstruction error of particle locations in the x and z coordinates, and the velocity components u and w, respectively.

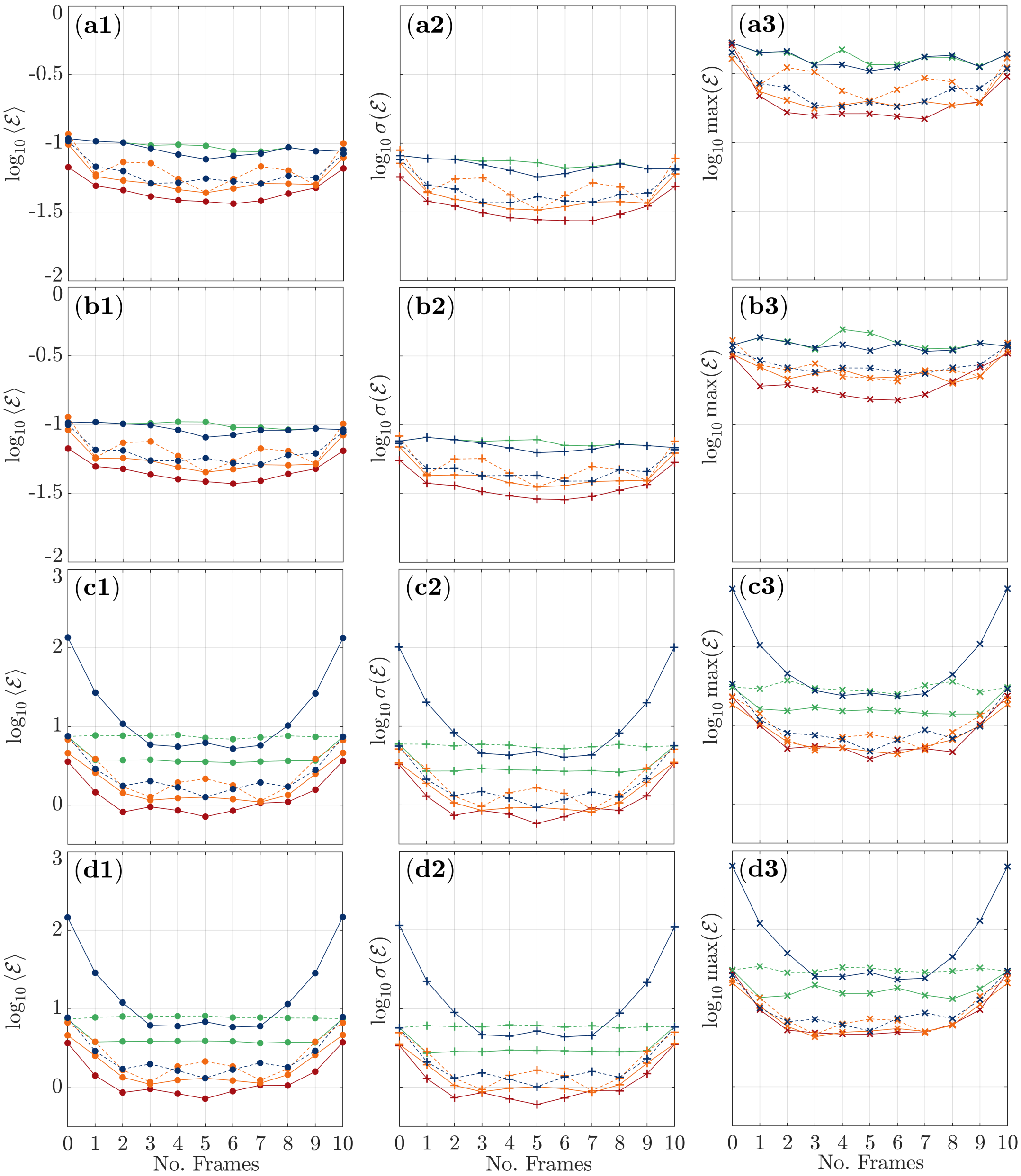


Figure 5.11: Reconstruction error comparison using different methods in the 3D ABC validation based on the synthetic data with 0.1% noise. Left column to right: the mean, standard deviation, and maximum of the relative error, respectively. Top row to bottom: reconstruction error of particle locations in the x and z coordinates, and the velocity components u and w, respectively.

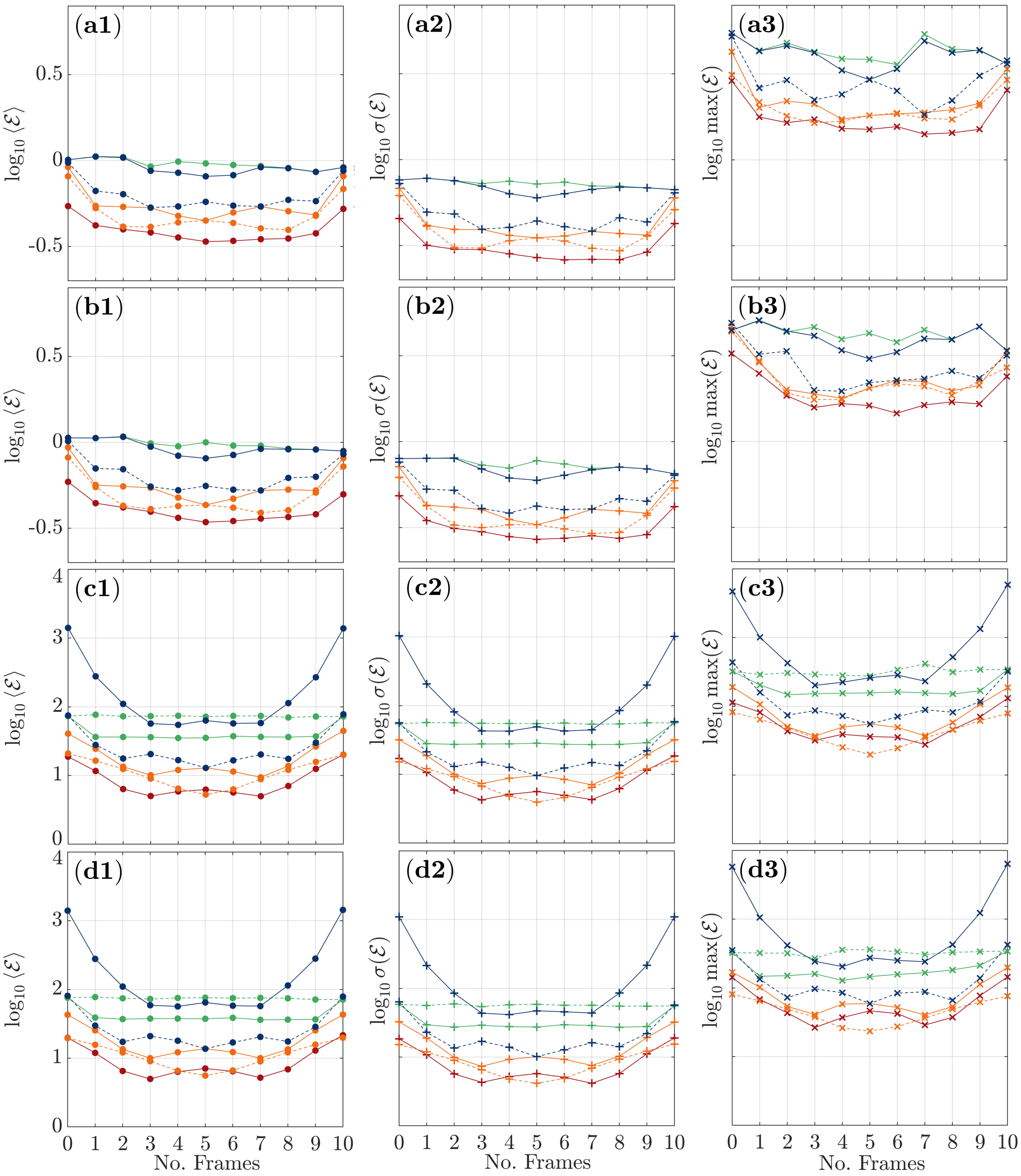


Figure 5.12: Reconstruction error comparison using different methods in the 3D ABC validation based on the synthetic data with 1% noise. Left column to right: the mean, standard deviation, and maximum of the relative error, respectively. Top row to bottom: reconstruction error of particle locations in the x and z coordinates, and the velocity components u and w, respectively.

In summary, the qualitative and quantitative assessments collectively suggest that the CLS-RBF method is an accurate trajectory and velocity reconstruction method. It outperforms the baseline algorithms and also effectively suppresses noise in synthetic data, enjoying smooth reconstruction.

5.3.3 Differential quantity reconstruction

The differential quantity reconstruction of the 2D TGV validation is presented in Figures 5.13 and 5.14, as well in Tables. 5.3 and 5.4. Figures 5.13 and 5.14 illustrate reconstruction results—based on the synthetic data with the noise levels of 0.1% and 1%, respectively—for a velocity gradient component $\partial\tilde{u}/\partial x$, velocity divergence $\nabla \cdot \tilde{\mathbf{U}}$, and the relative error of the reconstructed velocity gradient component $\partial\tilde{u}/\partial x$. As shown in these two figures, the reconstructed $\partial\tilde{u}/\partial x$ was almost identical to that of the ground truth in all three frames. The divergence $\nabla \cdot \tilde{\mathbf{U}}$ was mostly within a range of $\pm 1 \times 10^{-8}$, despite it was calculated based on noisy data, with only a few particles falling outside this range near the boundaries. For reconstruction based on the noise level $\xi = 1\%$, the velocity divergence further deviated from zero in general, but at most particles, it was still within the range of $\pm 1 \times 10^{-8}$. Considering the relatively high noise in the synthetic data, the near-zero velocity divergence suggests that the reconstructed velocity fields can be considered divergence-free. The reconstruction relative error of $\partial\tilde{u}/\partial x$ was below 7% at most particles, with the high relative error appearing mainly near the boundaries.

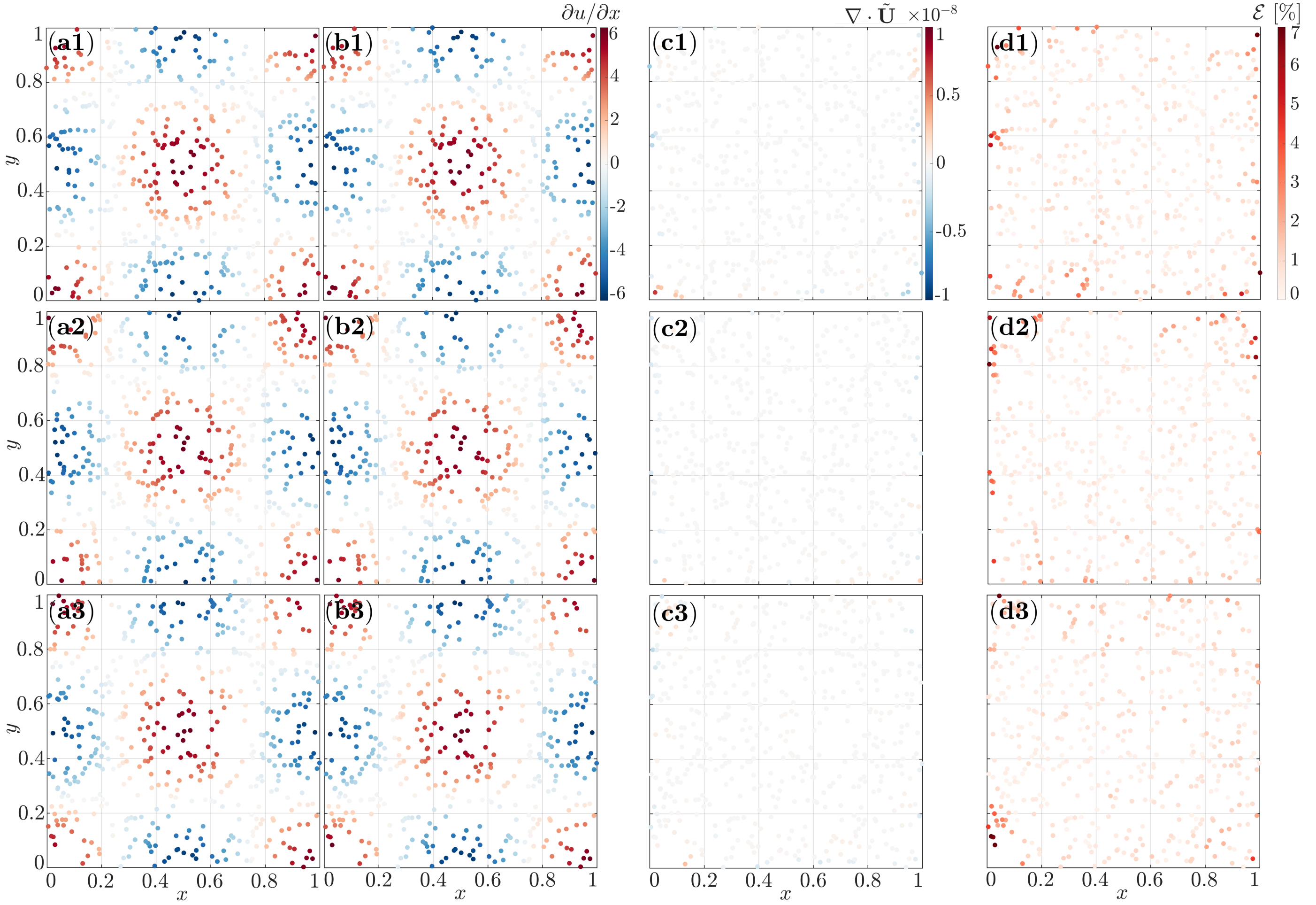


Figure 5.13: Reconstructed velocity gradient component $\partial\tilde{u}/\partial x$, velocity divergence $\nabla \cdot \tilde{\mathbf{U}}$, and the relative error of $\partial\tilde{u}/\partial x$ in the 2D validation using the CLS-RBF method based on the synthetic data with 0.1% noise. (a1) – (a3): the ground truth of $\partial u/\partial x$. (b1) – (b3): reconstructed $\partial\tilde{u}/\partial x$. (c1) – (c3): reconstructed $\nabla \cdot \tilde{\mathbf{U}}$. (d1) – (d3): the relative error of the reconstructed $\partial\tilde{u}/\partial x$. Top row to bottom: the reconstruction results of the third, sixth, and ninth frames, respectively.

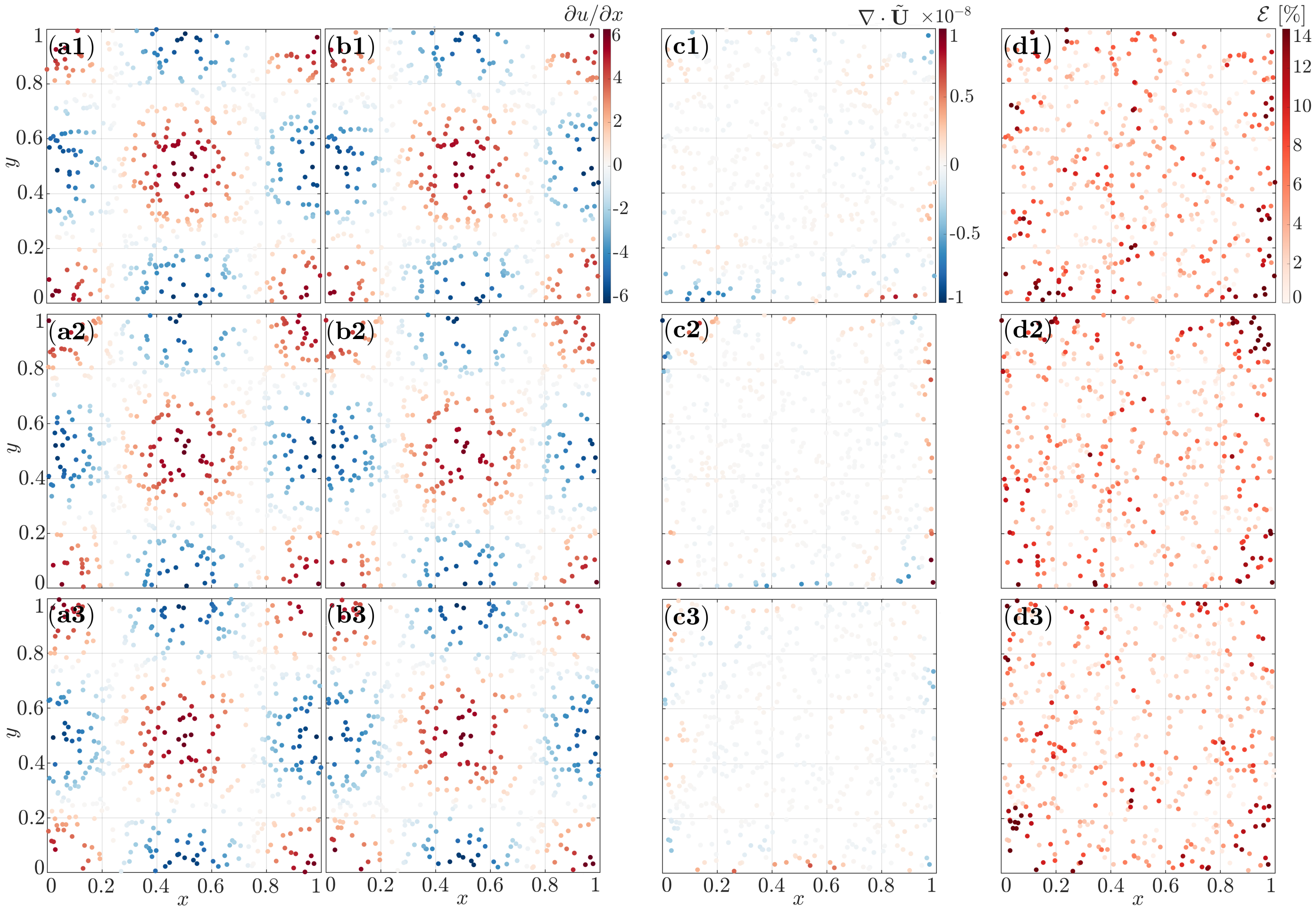


Figure 5.14: Reconstructed velocity gradient component $\partial \tilde{u}/\partial x$, velocity divergence $\nabla \cdot \tilde{\mathbf{U}}$, and the relative error of $\partial \tilde{u}/\partial x$ in the 2D validation using the CLS-RBF method based on the synthetic data with 1% noise. (a1) – (a3): the ground truth of $\partial u/\partial x$. (b1) – (b3): reconstructed $\partial \tilde{u}/\partial x$. (c1) – (c3): reconstructed $\nabla \cdot \tilde{\mathbf{U}}$. (d1) – (d3): the relative error of the reconstructed $\partial \tilde{u}/\partial x$. Top row to bottom: the reconstruction results of the third, sixth, and ninth frames, respectively.

A detailed comparison of reconstruction relative errors is in Tables 5.3 and 5.4. In these two tables, the statistics of relative error of the reconstructed velocity gradient tensor $\nabla\tilde{\mathbf{U}}$ for the 11 frames are listed. It can be observed that the diagonal elements of the velocity gradient tensors had lower relative errors than those of the off-diagonal elements. This is probably because the diagonal elements of the velocity gradient tensors participated in the constrained reconstruction. Since the reconstruction quality significantly degraded at the two ends of trajectories,[3] the relative error of velocity gradient components nearly halved after excluding the reconstructed results in the first and last frames. In summary, the CLS-RBF method can reconstruct differential quantities with accuracy and robustness, fulfilling divergence-free constraints in 2D TGV flows.

Table 5.3: Relative error of velocity gradient reconstruction based on the synthetic data with 0.1% and 1% noise in the 2D validation.

Quantity	$\xi = 0.1\%$			$\xi = 1\%$		
	$\langle\mathcal{E}\rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$	$\langle\mathcal{E}\rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$
$\partial\tilde{u}/\partial x$	1.87%	4.54%	94.11%	5.73%	8.34%	312.41%
$\partial\tilde{u}/\partial y$	3.67%	9.64%	282.28%	9.80%	23.73%	885.09%
$\partial\tilde{v}/\partial x$	3.87%	10.62%	270.11%	9.11%	14.62%	220.75%
$\partial\tilde{v}/\partial y$	1.88%	4.54%	94.11%	5.73%	8.38%	312.41%

Table 5.4: Relative error of velocity gradient reconstruction based on the synthetic data with 0.1% and 1% noise in the 2D validation (excluding the first and last frames).

Quantity	$\xi = 0.1\%$			$\xi = 1\%$		
	$\langle\mathcal{E}\rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$	$\langle\mathcal{E}\rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$
$\partial\tilde{u}/\partial x$	0.83%	1.26%	31.38%	4.62%	4.47%	67.46%
$\partial\tilde{u}/\partial y$	1.62%	3.74%	82.98%	6.20%	8.87%	135.98%
$\partial\tilde{v}/\partial x$	1.73%	3.60%	70.41%	6.31%	8.87%	122.77%
$\partial\tilde{v}/\partial y$	0.83%	1.26%	31.38%	4.62%	4.47%	67.46%

The 3D TGV differential quantity reconstruction is shown in Figure 5.15 – 5.18 and Tables 5.5 and 5.6. Figure 5.15 presents iso-surfaces of coherent structures based on

---

[3] This degradation in quality exists for all baseline methods, not only for the CLS-RBF method.

the Q-criterion. The coherent structures were visualized at synthetic particles in Figure 5.15(a1) – (b1), while in (a2) – (b2), they are reconstructed and visualized with spatial super-resolution.[4] As illustrated in Figure 5.15, with or without spatial super-resolution, the iso-surfaces of coherent structures based on the Q-criterion were almost identical to those in the ground truth, with only minor distortions near the boundaries. Figures 5.16 – 5.18 show iso-surfaces of strain-rate and rotation-rate tensors in the sixth frame with two different noise levels. As shown in these three figures, the iso-surfaces of reconstructed diagonal elements in the strain-rate tensor were almost identical to those in the ground truth. However, the iso-surfaces of off-diagonal elements in the strain-rate tensor (i.e., $\tilde{S}_{21}$, $\tilde{S}_{31}$, and $\tilde{S}_{32}$) and non-zero elements in the rotation-rate tensor (i.e., $\tilde{R}_{13}$ and $\tilde{R}_{23}$) showed discernible distortions near the boundaries. It may be due to the involvement of these diagonal elements in the constrained reconstruction. Although the distortion on iso-surfaces was pronounced when the noise level increased to 1%, the profiles of the iso-surface remained smooth and the flow features maintained recognizable.

---

[4] Hereafter, by default, the iso-surfaces visualized in figures are based on spatial super-resolution reconstruction.

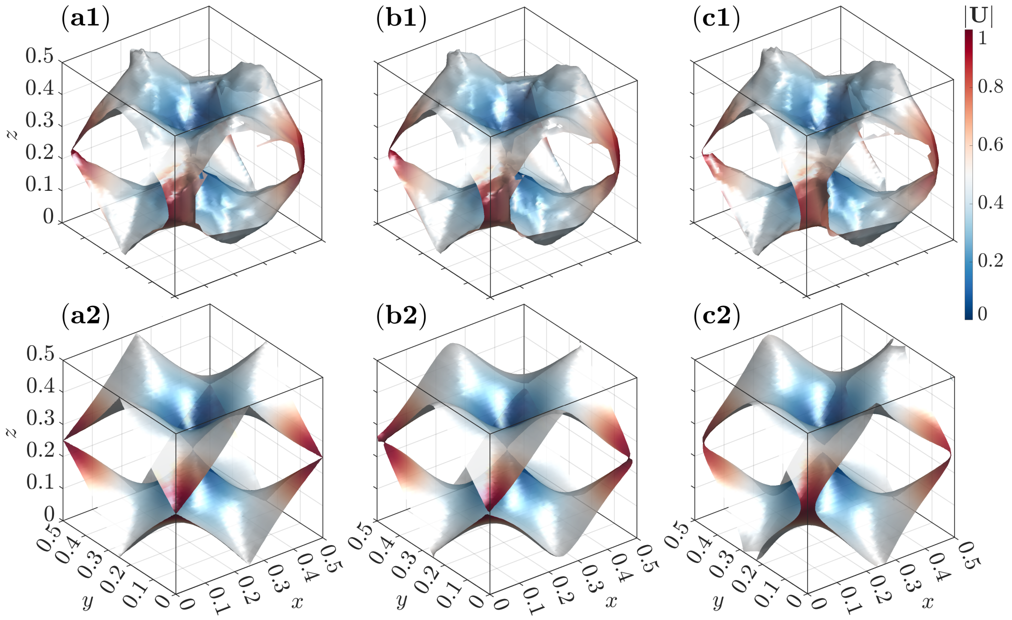


Figure 5.15: Iso-surfaces of coherent structures based on the Q-criterion (iso-value = ±0.001) in the sixth frame of the 3D TGV validation. The iso-surfaces are colored by particle velocities. Left column to right: the ground truth, reconstruction results using CLS-RBF method based on the synthetic data with 0.1% and 1% noise, respectively. Upper row: reconstruction at particle locations of the synthetic data; lower row: spatial super-resolution reconstruction.

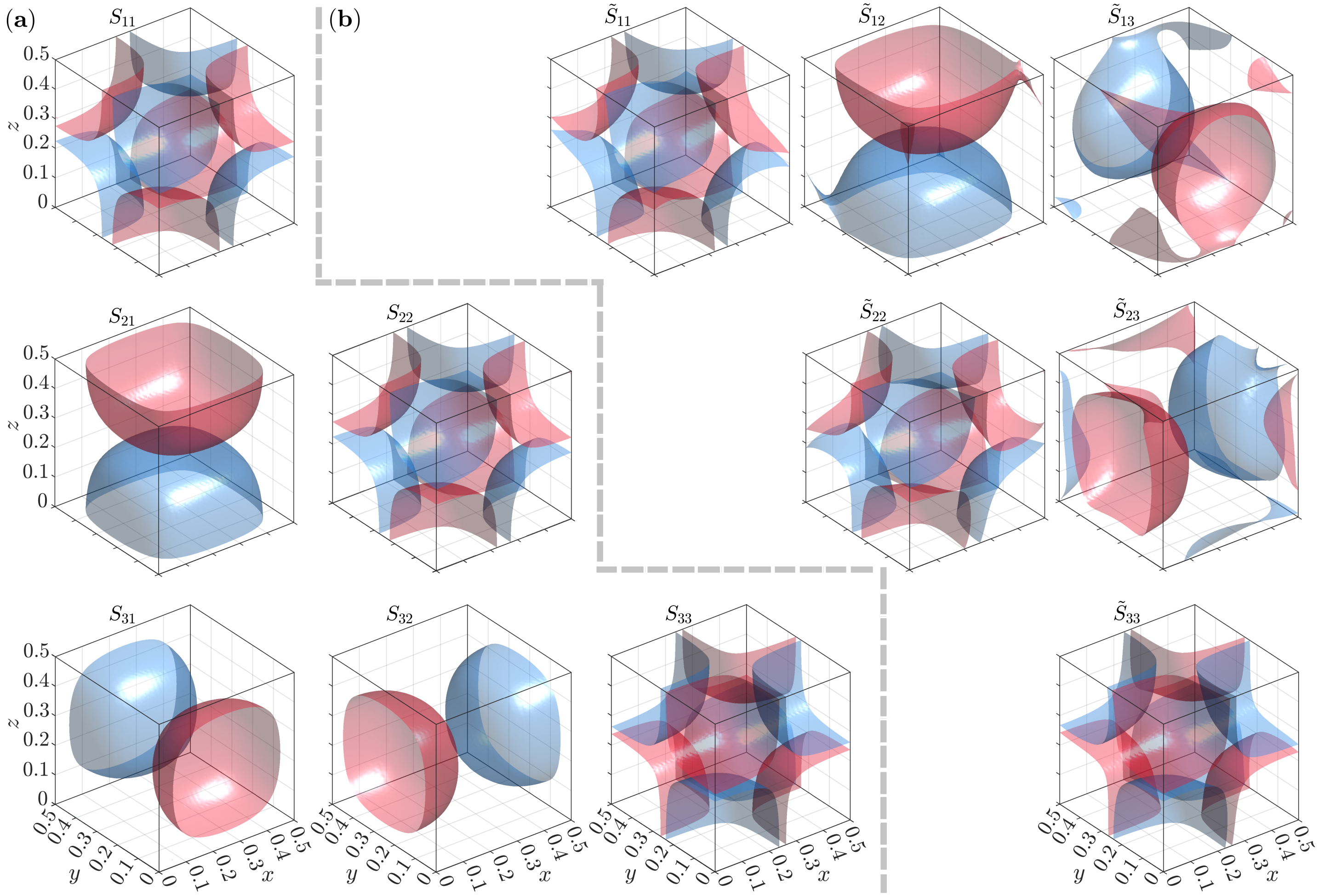


Figure 5.16: Iso-surfaces of strain-rate tensors (iso-value = ± 0.5) in the sixth frame based on the synthetic data with 0.1% noise in the 3D TGV validation. Group (a) the ground truth; group (b) reconstruction using CLS-RBF method. The colors red and blue correspond to positive and negative iso-values, respectively.

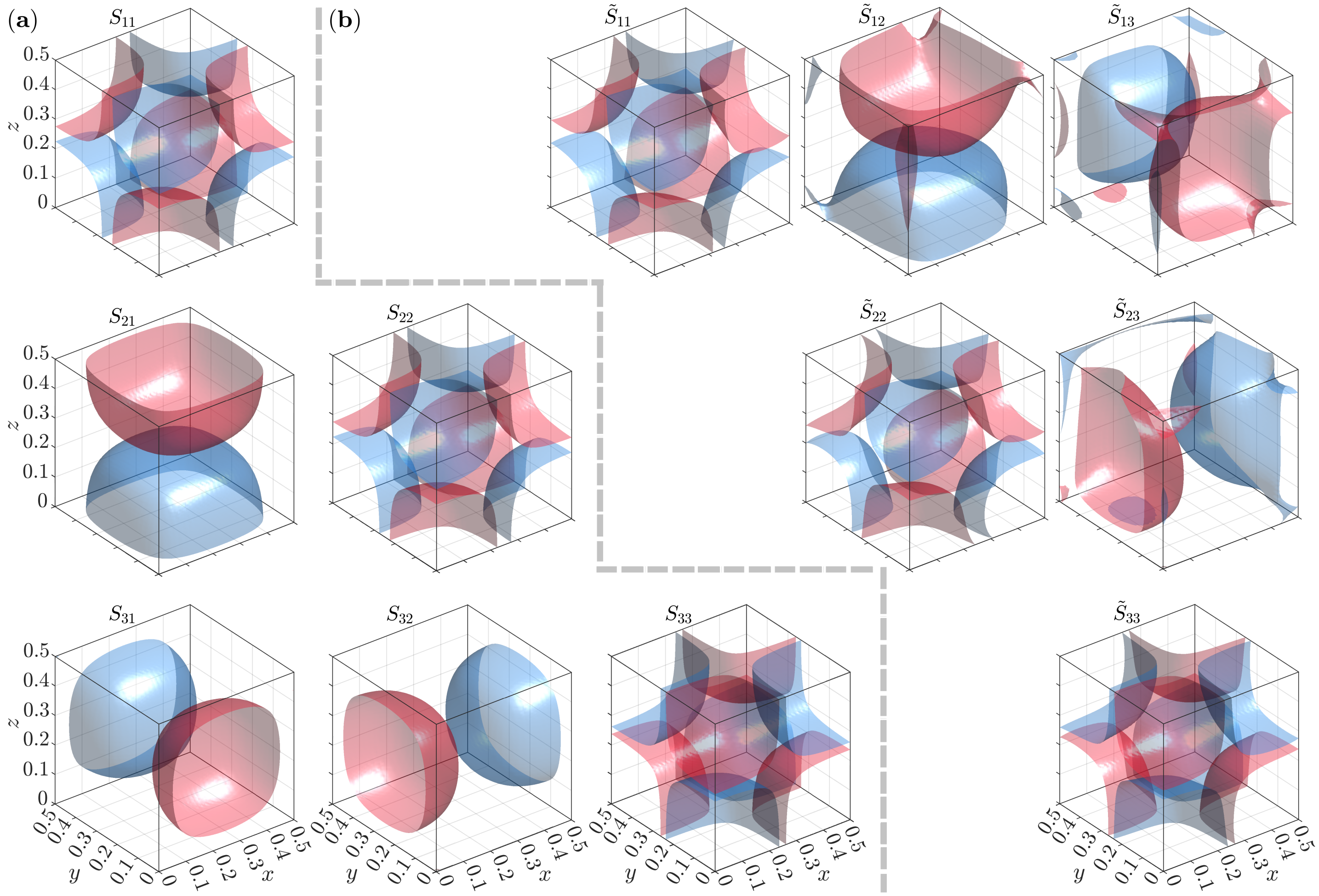


Figure 5.17: Iso-surfaces of strain-rate tensors (iso-value = ±0.5) in the sixth frame based on the synthetic data with 1% noise in the 3D TGV validation. Group (a) the ground truth; group (b) reconstruction using CLS-RBF method. The colors red and blue correspond to positive and negative iso-values, respectively.

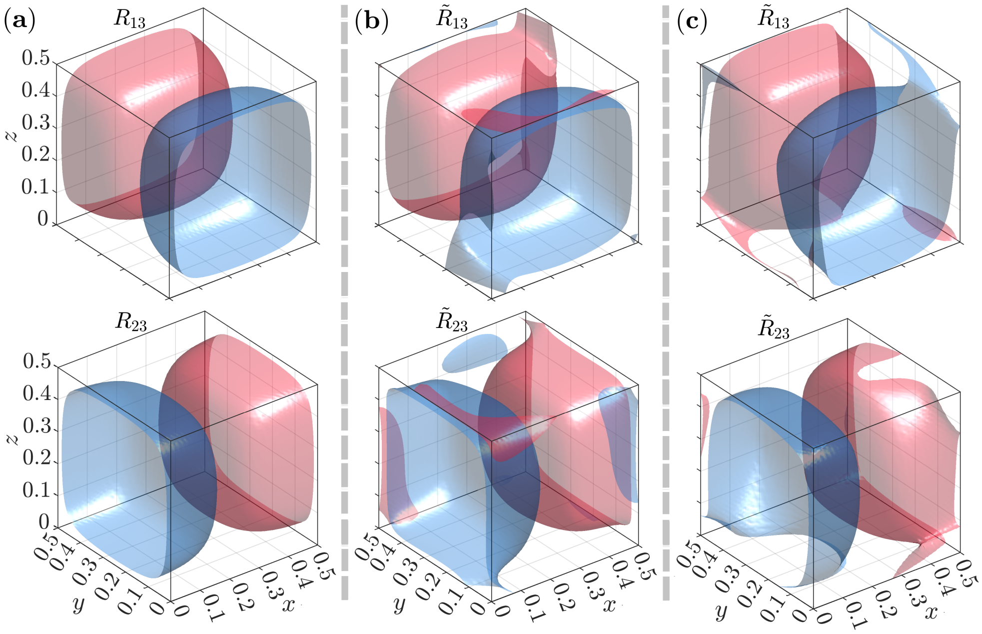


Figure 5.18: Iso-surfaces of rotation-rate tensors (iso-value = ±0.5) in the sixth frame based on the synthetic data with 0.1% & 1% noise in the 3D TGV validation. (a) the ground truth; reconstruction using CLS-RBF method based on synthetic data with 0.1% (b) and 1% (c) noise levels. The colors red and blue correspond to positive and negative iso-values, respectively.

The statistics of reconstruction relative error of velocity gradients are listed in Tables 5.5 and 5.6. The statistics of the 3D TGV validation exhibited similar characteristics to those in the 2D TGV validation. Considering the noise imposed in the synthetic data and extra errors introduced by differentiation, the error in the reconstructed velocity gradients was sufficiently low. In addition, the velocity divergence in all frames (not shown in tables here) fell into ranges of $[-5 \times 10^{-7}, 5 \times 10^{-7}]$ and $[-2 \times 10^{-6}, 2 \times 10^{-6}]$ for the synthetic data with 0.1% and 1% noise levels, respectively. The near-zero divergence implies that the reconstructed velocity field was solenoidal. In summary, the CLS-RBF method can reconstruct differential quantities with accuracy and robustness, reveal important flow structures, and fulfill divergence-free constraints in 3D TGV flows.

Table 5.5: Relative error of velocity gradient reconstruction based on the synthetic data with 0.1% and 1% noise in the 3D TGV validation.

Quantity	$\xi = 0.1\%$			$\xi = 1\%$		
	$\langle \mathcal{E} \rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$	$\langle \mathcal{E} \rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$
$\partial \tilde{u}/\partial x$	2.16%	3.23%	62.99%	5.17%	8.74%	162.72%
$\partial \tilde{u}/\partial y$	2.21%	3.00%	64.55%	6.54%	11.11%	203.33%
$\partial \tilde{u}/\partial z$	3.22%	4.58%	82.74%	6.87%	11.45%	179.92%
$\partial \tilde{v}/\partial x$	2.65%	4.07%	72.38%	7.18%	13.42%	181.37%
$\partial \tilde{v}/\partial y$	2.21%	3.06%	50.67%	5.45%	10.90%	276.72%
$\partial \tilde{v}/\partial z$	3.27%	4.77%	75.60%	7.91%	15.09%	234.21%
$\partial \tilde{w}/\partial x$	2.44%	3.99%	60.14%	4.75%	9.27%	187.65%
$\partial \tilde{w}/\partial y$	2.58%	4.22%	80.62%	4.56%	7.66%	126.05%
$\partial \tilde{w}/\partial z$	1.49%	2.04%	42.40%	3.59%	5.79%	127.11%

Table 5.6: Relative error of velocity gradient reconstruction based on the synthetic data with 0.1% and 1% noise in the 3D TGV validation (excluding the first and last frames).

Quantity	$\xi = 0.1\%$			$\xi = 1\%$		
	$\langle \mathcal{E} \rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$	$\langle \mathcal{E} \rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$
$\partial \tilde{u}/\partial x$	1.59%	1.80%	23.33%	3.41%	4.00%	42.23%
$\partial \tilde{u}/\partial y$	1.78%	2.01%	21.07%	4.49%	6.21%	108.49%
$\partial \tilde{u}/\partial z$	2.63%	3.39%	34.23%	4.39%	4.99%	60.78%
$\partial \tilde{v}/\partial x$	2.09%	2.70%	38.47%	4.51%	6.51%	83.23%
$\partial \tilde{v}/\partial y$	1.82%	2.22%	30.30%	3.40%	4.04%	43.88%
$\partial \tilde{v}/\partial z$	2.75%	3.61%	32.14%	4.59%	5.96%	74.31%
$\partial \tilde{w}/\partial x$	1.76%	2.55%	43.18%	3.15%	4.37%	49.91%
$\partial \tilde{w}/\partial y$	2.11%	3.01%	49.00%	2.91%	3.50%	44.86%
$\partial \tilde{w}/\partial z$	1.18%	1.37%	26.70%	2.58%	3.00%	28.57%

The 3D ABC differential quantity reconstruction are shown in Figures 5.19 – 5.22 and Tables 5.7 and 5.8. Figures 5.19 – 5.22 represent the iso-surfaces of reconstructed strain-rate and rotation-rate tensors. In these four figures, smooth tubular structures can be observed. These structures closely matched the ground truth when the noise level was low. However, the high noise level in the synthetic data may lead to distorted tubular structures. Nevertheless, the major structures of the strain-rate and rotation-rate tensors remained recognizable and smooth.

In Tables 5.7 and 5.8, the statistics of relative error of reconstructed velocity gradients are listed.[5] Table 5.7 and 5.8 show that the relative error of the reconstructed velocity gradients was low when the noise level was 0.1%. Although the relative error increased significantly when the noise level was 1%, it was still acceptable, considering that the synthetic data were severely contaminated.[6] Furthermore, the velocity divergence for all particles in every frame (not shown in tables here) fell into the ranges of $[-2\times10^{-7}, 2\times10^{-7}]$ and $[-5\times10^{-6}, 5\times10^{-6}]$ for the synthetic data with 0.1% and 1% noise levels, respectively. These results imply that the reconstructed velocity gradient was accurate and the velocity fields could be considered divergence-free. In summary, the CLS-RBF method exhibits comparable performance in the differential quantity reconstruction in the ABC validation as it does in the 3D TGV validation. The CLS-RBF method is capable to reconstruct differential quantities with accuracy and robustness for different flows.

The above 2D and 3D validations show that the CLS-RBF method can reconstruct particle trajectories, velocities, and differential quantities with accuracy and robustness, satisfying the divergence-free constraints for different flow fields. The CLS-RBF method outperforms the baseline algorithms in trajectory and velocity reconstruction and has low reconstruction errors. It can calculate differential quantities with low errors. The resolved flow structures are almost identical to the ground truth despite the presence of noise in the synthetic data. The performance of the CLS-RBF method in the Lagrangian flow field reconstruction is consistent and stable under different flow conditions with different noise levels. In summary, the CLS-RBF method is promising for LPT data processing.

---

[5] Since the diagonal elements of velocity gradients in the ABC flows are always zero, the relative errors of these components are absent in the tables.

[6] The trajectories of the synthetic data with 1% noise level showed significant zigzagging features and trajectories were nearly unrecognizable.

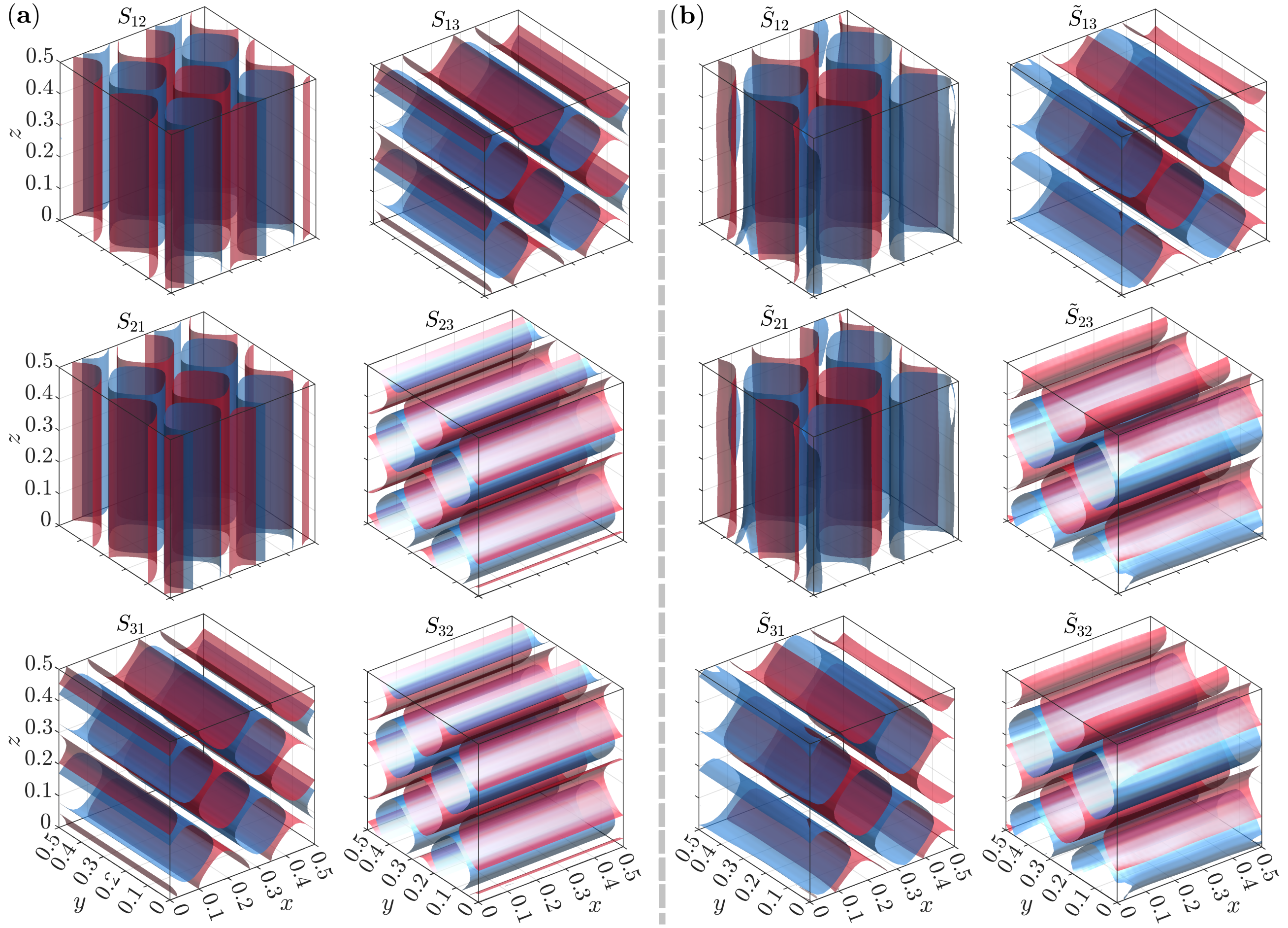


Figure 5.19: Iso-surfaces of strain-rate tensors (iso-value = ±0.5) based on the synthetic data with 0.1% noise in the 3D ABC validation. Group (a): the ground truth; group (b): reconstruction using the CLS-RBF method. The colors red and blue correspond to positive and negative iso-values, respectively.

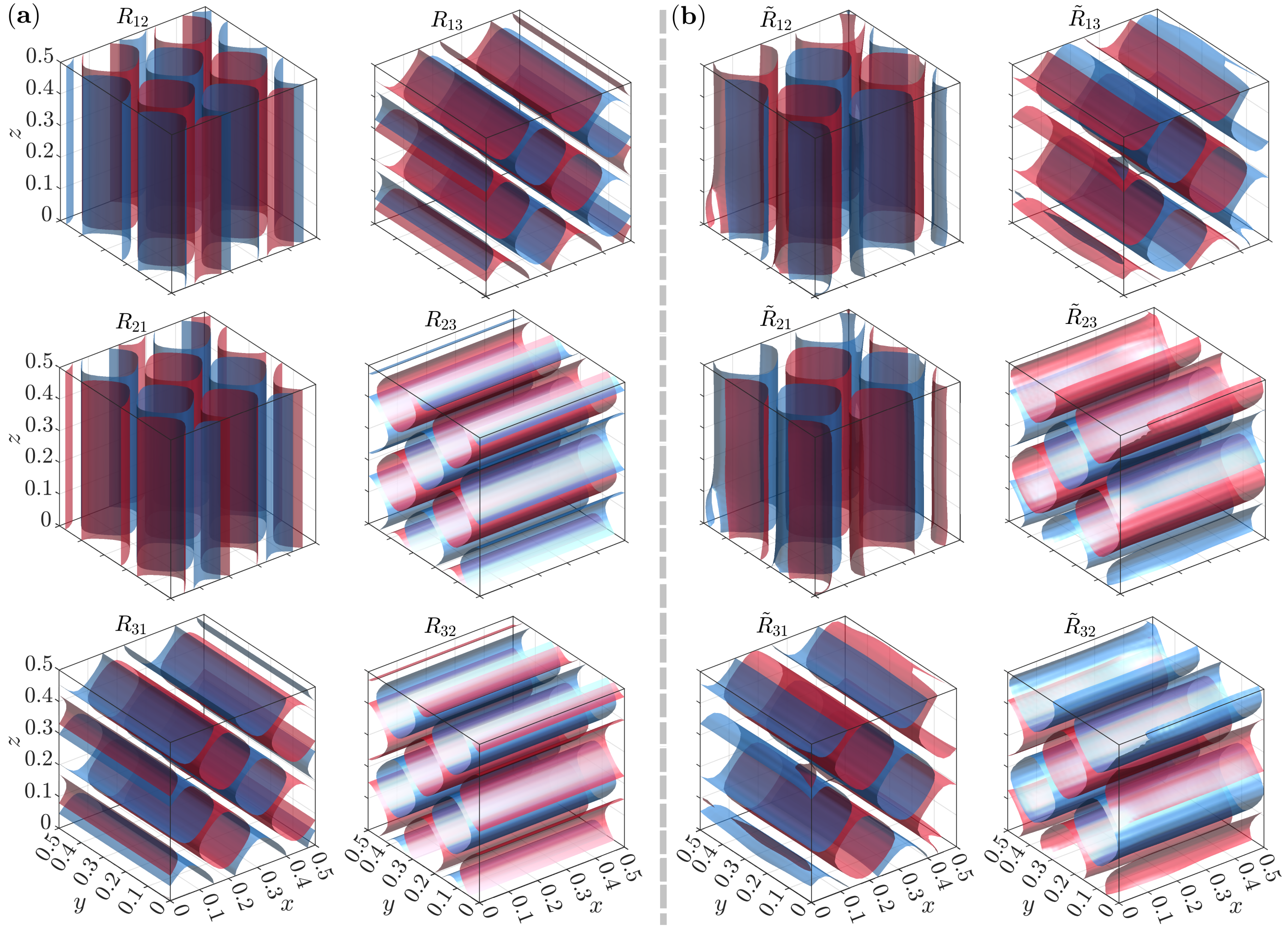


Figure 5.20: Iso-surfaces of rotation-rate tensors (iso-value = ±0.5) based on the synthetic data with 0.1% noise in the 3D ABC validation. Group (a): the ground truth; group (b): reconstruction using the CLS-RBF method. The colors red and blue correspond to positive and negative iso-values, respectively.

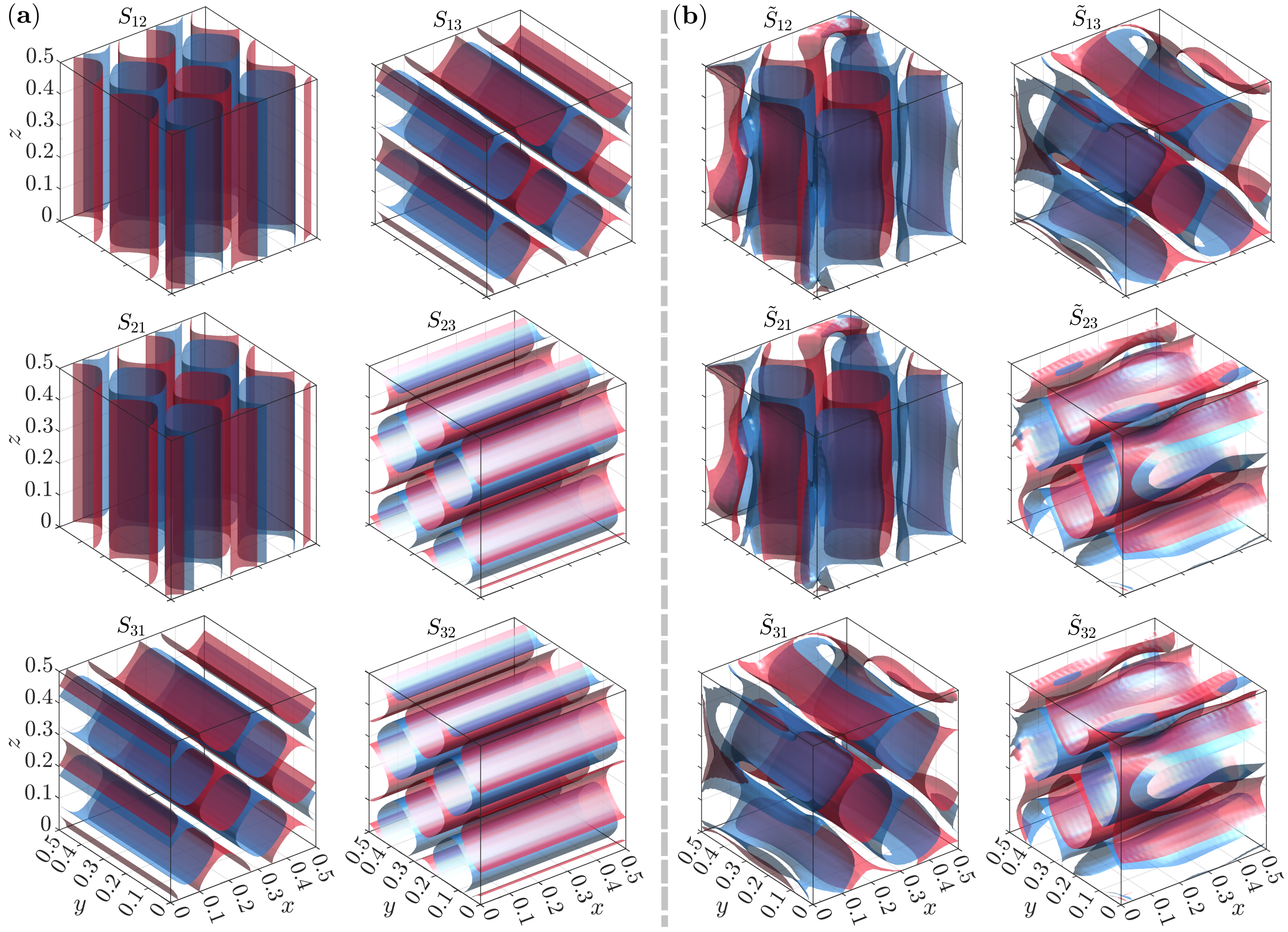


Figure 5.21: Iso-surfaces of strain-rate tensors (iso-value = ±0.5) based on the synthetic data with 1% noise in the 3D ABC validation. Group (a): the ground truth; group (b): reconstruction using the CLS-RBF method. The colors red and blue correspond to positive and negative iso-values, respectively.

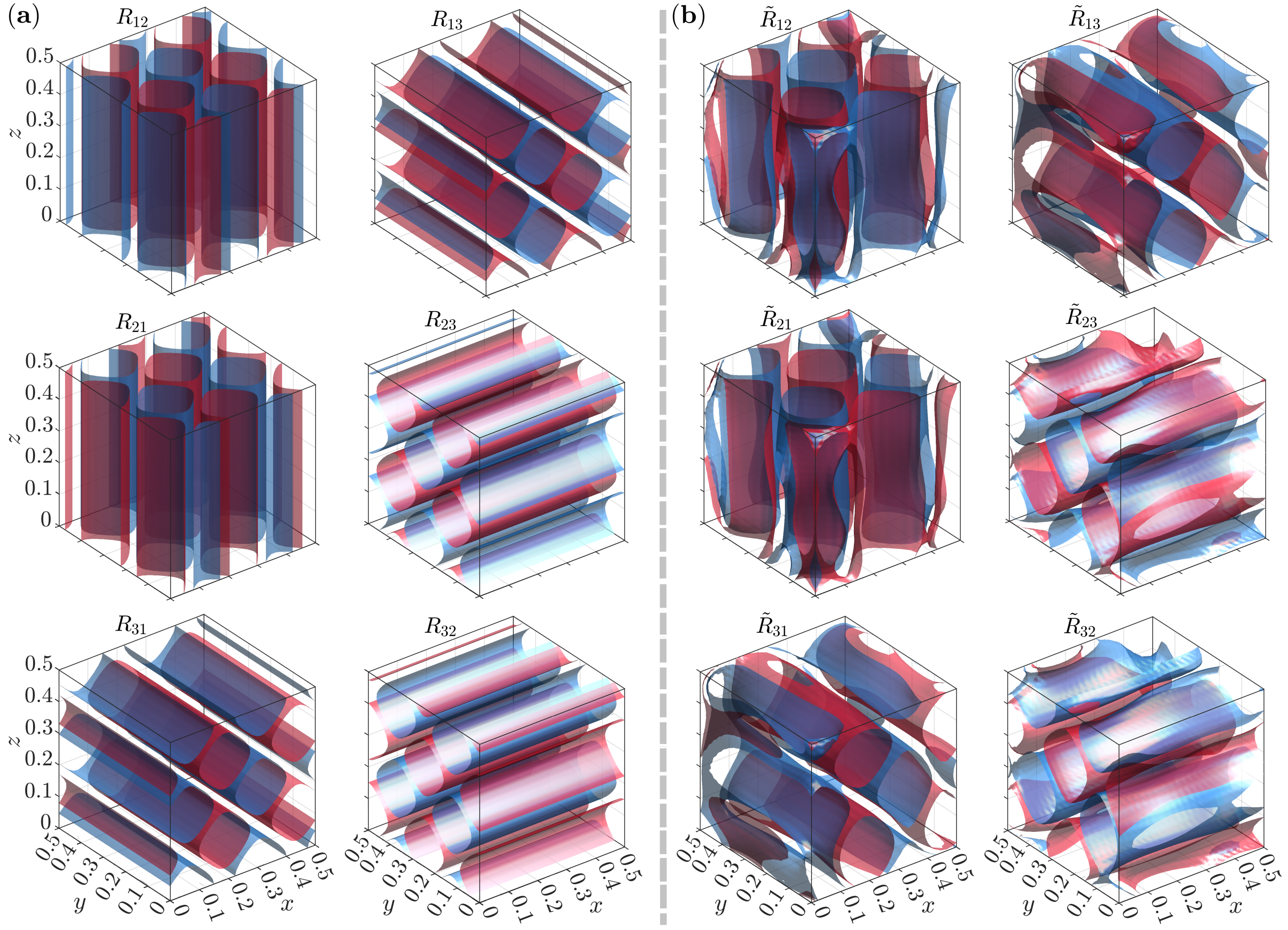


Figure 5.22: Iso-surfaces of rotation-rate tensors (iso-value = ± 0.5) based on the synthetic data with 1% noise in the 3D ABC validation. Group (a): the ground truth; group (b): reconstruction using the CLS-RBF method. The colors red and blue correspond to positive and negative iso-values, respectively.

Table 5.7: Relative error of velocity gradient reconstruction based on the synthetic data with 0.1% and 1% noise in the 3D ABC validation.

Quantity	$\xi = 0.1\%$			$\xi = 1\%$		
	$\langle \mathcal{E} \rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$	$\langle \mathcal{E} \rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$
$\partial \tilde{u}/\partial x$	—	—	—	—	—	—
$\partial \tilde{u}/\partial y$	6.27%	21.43%	1152.75%	30.70%	77.31%	2463.09%
$\partial \tilde{u}/\partial z$	7.16%	25.86%	1570.60%	30.24%	71.01%	2084.87%
$\partial \tilde{v}/\partial x$	6.79%	32.96%	1846.56%	29.29%	86.00%	3700.34%
$\partial \tilde{v}/\partial y$	—	—	—	—	—	—
$\partial \tilde{v}/\partial z$	6.09%	18.25%	790.34%	32.55%	98.07%	4297.90%
$\partial \tilde{w}/\partial x$	5.53%	14.12%	304.77%	31.20%	71.91%	1843.19%
$\partial \tilde{w}/\partial y$	6.89%	18.82%	656.89%	27.61%	71.86%	3235.89%
$\partial \tilde{w}/\partial z$	—	—	—	—	—	—

Table 5.8: Relative error of velocity gradient reconstruction based on the synthetic data with 0.1% and 1% noise in the 3D ABC validation (excluding the first and last frames).

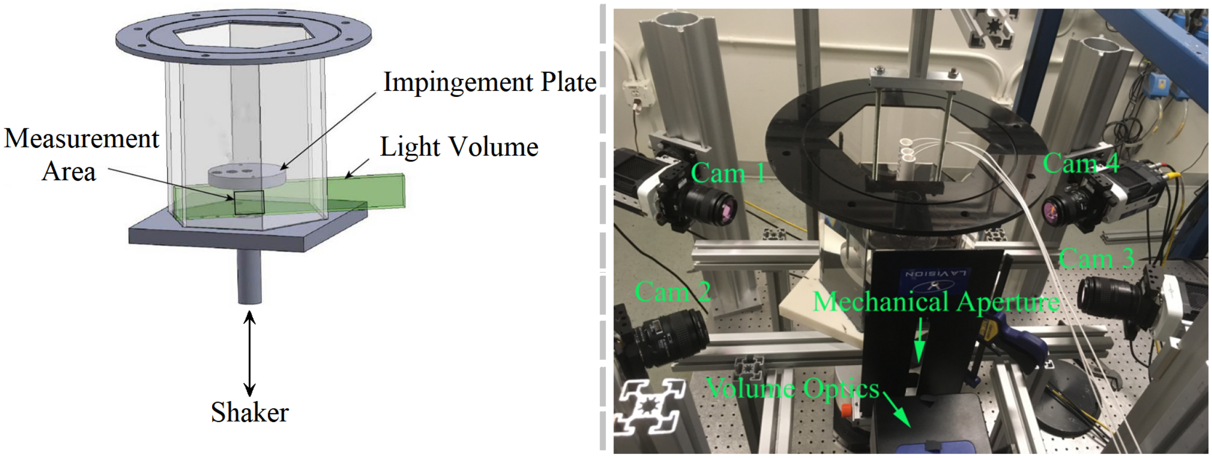
Quantity	$\xi = 0.1\%$			$\xi = 1\%$		
	$\langle \mathcal{E} \rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$	$\langle \mathcal{E} \rangle$	$\sigma(\mathcal{E})$	$\max(\mathcal{E})$
$\partial \tilde{u}/\partial x$	—	—	—	—	—	—
$\partial \tilde{u}/\partial y$	4.23%	11.30%	319.15%	22.41%	44.30%	940.18%
$\partial \tilde{u}/\partial z$	5.26%	13.07%	335.47%	23.47%	50.80%	1027.49%
$\partial \tilde{v}/\partial x$	4.65%	14.20%	571.40%	22.34%	52.99%	2013.46%
$\partial \tilde{v}/\partial y$	—	—	—	—	—	—
$\partial \tilde{v}/\partial z$	4.27%	11.18%	227.48%	24.95%	64.12%	2641.30%
$\partial \tilde{w}/\partial x$	3.86%	9.97%	289.61%	23.73%	46.03%	1061.64%
$\partial \tilde{w}/\partial y$	5.02%	12.45%	317.65%	20.80%	43.41%	1245.04%
$\partial \tilde{w}/\partial z$	—	—	—	—	—	—

5.4 Experimental Verification

The CLS-RBF Lagrangian reconstruction method was verified by experimental data collected from a low-speed pulsing jet flow measured by a volumetric LPT system. This experiment was conducted by Dr. Md Nazmus Sakib et al. at Utah State University, US [138, 139]. Section 5.4.1 briefs the experimental setups. In Section 5.4.2, the experimental data are pre-processed and the parameters for the CLS-RBF Lagrangian reconstruction are listed. Section 5.4.3 discusses the 3D experimental reconstruction results.

5.4.1 Experiment setup

The experimental facility comprised a transparent hexagonal water tank, a cylindrical piston equipped with an electromagnetic shaker, an impingement plate, and optical equipment (see Figure 5.23 for illustration). The piston was located at the bottom of the water tank and the shaker oscillated the piston to generate a simulated pulsing jet. The water was then pushed through a circular orifice, which impinged on a circular plate at the top of the measurement area. The flow domain was seeded with neutrally buoyant phosphorescent micro-plastic particles with a diameter of 50 μm. A dual cavity high-speed laser illuminated the measurement area whose dimensions were 60 mm × 57 mm × 20 mm. Four high-speed cameras captured the jet flow and provided time-resolved LPT images at two thousand frames per second. More details about the LPT experimental setups can be found in [138, 139]. The built-in software for LPT data acquisition and processing was DaVis 10 developed by LaVision. The DaVis 10 processed LPT raw images and reconstructed particle trajectories using the Shake-The-Box (STB) technique [141]. It can output particle locations, velocity, and acceleration in each frame. After obtaining the raw LPT data, data pre-processing and Lagrangian flow field reconstruction were next performed.



(a) (b)

Figure 5.23: The LPT experimental facility. A schematic diagram (a) and an image (b) of the experimental facility. Both (a) and (b) were adopted with permission from Dr. Md Nazmus Sakib at Utah State University [138, 139].

5.4.2 Pre-processing and reconstruction

Due to limitations of the current version of the CLS-RBF Lagrangian reconstruction method,[7] pre-processing the experimental LPT data needs to be performed via two steps before reconstruction. First, the measurement area was cropped to a computational domain that had dimensions of $x \times y \times z \in [-5, 15] \times [0, 20] \times [-10, 10]$ mm^3. This step reduced the scale of LPT data and the number of particles. Second, the raw LPT data were down-sampled to ensure that trajectories were long enough. This practice provides two features to challenge the CLS-RBF method: i) long trajectories with varying curvature on the pathlines and ii) low resolution in time. After these two steps, the pre-processed LPT data were compatible with the current version of the method and ready for reconstruction.

To challenge the robustness of the CLS-RBF Lagrangian reconstruction method, highly

---

[7] The limitations of the current version of the CLS-RBF Lagrangian reconstruction method will be summarized in Chapter 6. One of the limitations is the inability to process large-scale LPT data with long trajectories (e.g., ten thousand particles with trajectories containing tens of frames). However, this limitation can be overcome by using the partition-of-unity method (PUM) in space and time. The PUM can partition the domain into small subdomains with few particles in space and time. In each subdomain, computation is independent, but the results in the entire domain are unity. The CLS-RBF Lagrangian reconstruction method can work with these few particles (usually several hundred) and the computational results in the entire domain can be reconstructed by the PUM.

contaminated LPT data were intentionally introduced. These contaminated LPT data were generated by adding extra artificial noise to the particle spatial coordinates in the pre-processed LPT data. The STB system reported an average nominal uncertainty of 0.0267% for the spatial coordinates of the particles. The extra artificial noise level was 0.5%, which was one order of magnitude higher than that of the raw data, representing a typical low-quality data set. The generation of extra artificial noise was based on the same principle as that of the synthetic data used for the 3D validation (see Section 5.1.1). In this sense, it can be considered that the pre-processed LPT data were the 'ground truth', and the contaminated LPT data were synthetic data.

Next, the CLS-RBF method was applied to reconstruct the flow field based on the pre-processed and contaminated LPT data sets. The parameters for reconstruction are listed in Table 5.9. The reconstruction was performed on a workstation equipped with AMD Ryzen 9 3900X CPU and 128 GB of RAM, running the operating system Windows 11. The CPU had 12 cores with a clock speed of 3.79 GHz.[8] The RAM was composed of four channels with a frequency of 2,666 MHz. The CLS-RBF Lagrangian reconstruction method was developed on MATLAB R2022b. The average reconstruction wall-clock time was 19 s, 55 s, and 84 s for direct reconstruction at measured particles, spatial super-resolution, and temporal super-resolution, respectively. In the spatial super-resolution reconstruction, $31 \times 31 \times 31$ evaluation points were uniformly distributed in the domain with 11 frames; while in the temporal super-resolution reconstruction, 51 frames were reconstructed and the evaluation points coincided with measured particles. No significant fluctuations in wall-clock time were observed when different nominal uncertainty data were input. The current reconstruction codes were efficient, even with super-resolution in space and time. The codes have not been optimized yet, but future optimization, such as implementing parallel computing, could potentially increase the efficiency of the reconstruction. The results, including particle trajectories, velocities, and differential quantities, are visualized and analyzed in the next section.

Table 5.9: Parameters for CLS-RBF Lagrangian reconstruction in the 3D experimental verification.

Nominal uncertainty	ε	η_0	η_1	η_2
0.0267%	1×10^{-10}	11/10	4	5
0.5%	1×10^{-10}	11/7	4.5	4.5

---

[8] Parallel computing was not activated in the current reconstruction, but it can be applied in theory.

5.4.3 Results and discussions

In this section, the reconstruction results using the CLS-RBF method in the 3D experimental verification are discussed. The particle trajectory reconstruction based on the pre-processed LPT data is shown in Figure 5.24. Figure 5.24(a2) – (c2) are the zoomed-in views of (a1) – (c1). These sub-figures focus on the region of $x \times y \times z \in [0, 10] \times [5, 15] \times [-5, 5]$ mm^3, which were mainly occupied by a jet core. In Figure 5.24, the trajectories reconstructed by the CLS-RBF method (see Sub-figures (b1) and (b2)) were in good agreement with those reconstructed by the DaVis 10 (see Sub-figures (a1) and (a2)). The temporal super-resolution reconstruction (see Sub-figures (c1) and (c2)) exhibited the same trajectory profiles as those directly reconstructed at the measured particles (see Sub-figures (b1) and (b2)). Furthermore, the temporal super-resolution reconstruction offered a more detailed visualization of the pathlines. These tests indicate that the CLS-RBF method can reconstruct trajectories from experimental data. Temporal super-resolution is helpful, especially for experimental data with low resolution in time.

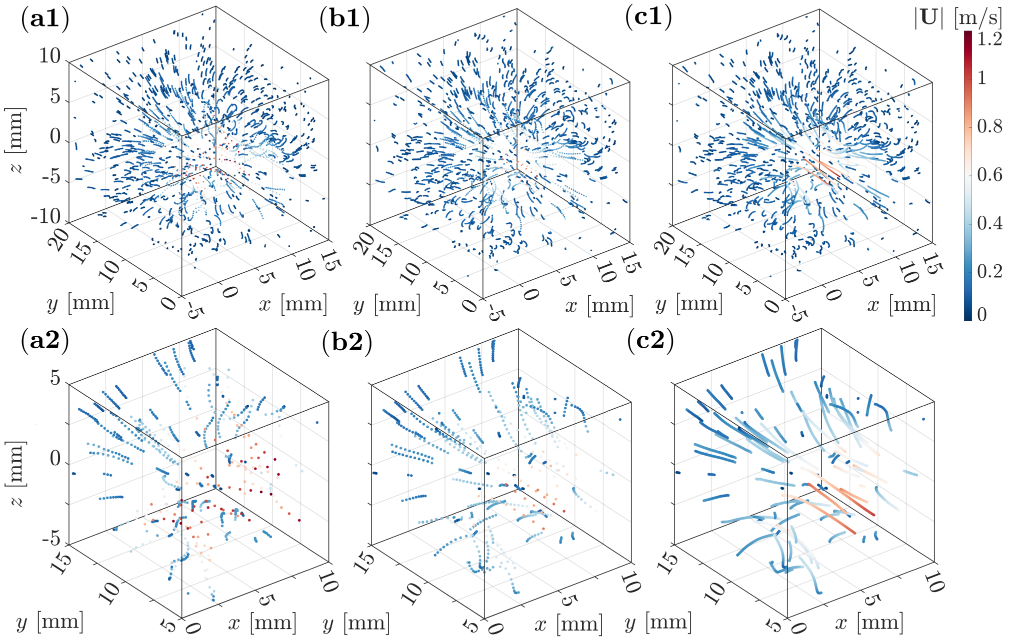


Figure 5.24: Reconstructed particle trajectories in the 3D experimental verification. Left column to right: reconstructed trajectories using the DaVis 10, raw data with and without temporal super-resolution based on the CLS-RBF Lagrangian reconstruction, respectively. Sub-figures (a2) – (c2) are the zoomed-in views of (a1) – (c1) near the jet core.

The velocity and differential quantity reconstruction are shown in Figures 5.25 – 5.29. Figure 5.25 represents reconstruction results based on the pre-processed LPT data. As shown in Sub-figures (b1) – (b3), a smooth toroidal structure represented a vortex moving from about $y = 5$ mm to $y = 15$ mm, which were in good agreement with the velocity vector fields in Sub-figures (a1) – (a3). The toroidal vortical structure is a typical structure in a low-speed pulsing jet flow and it was successfully detected by the CLS-RBF method. In Sub-figures (c1) – (c3), the reconstructed velocity divergence was within $\pm 1 \times 10^{-8}$ s^{-1} for most particles, indicating that the reconstructed velocity fields can be considered solenoidal in each frame.

Figure 5.26 shows the reconstruction results based on the contaminated LPT data. A major flow structure (i.e., toroidal vortical structures) was captured but with more distortions than the structures depicted in Figure 5.25. Some trivial structures emerged near the domain boundaries due to the added noise. Nevertheless, the main flow features were still recognizable (see Figure 5.26(b1) – (b3)). Besides, the velocity divergence was mostly within $\pm 1 \times 10^{-7}$ s^{-1} (see Figure 5.26(c1) – (c3)), suggesting that the reconstructed velocity fields can be considered divergence-free, despite the highly contaminated data were used. In summary, the CLS-RBF method is capable to reconstruct particle trajectories, velocities, and coherent structures satisfying divergence-free constraints with temporal and spatial super-resolution, even if experimental data are severally contaminated.

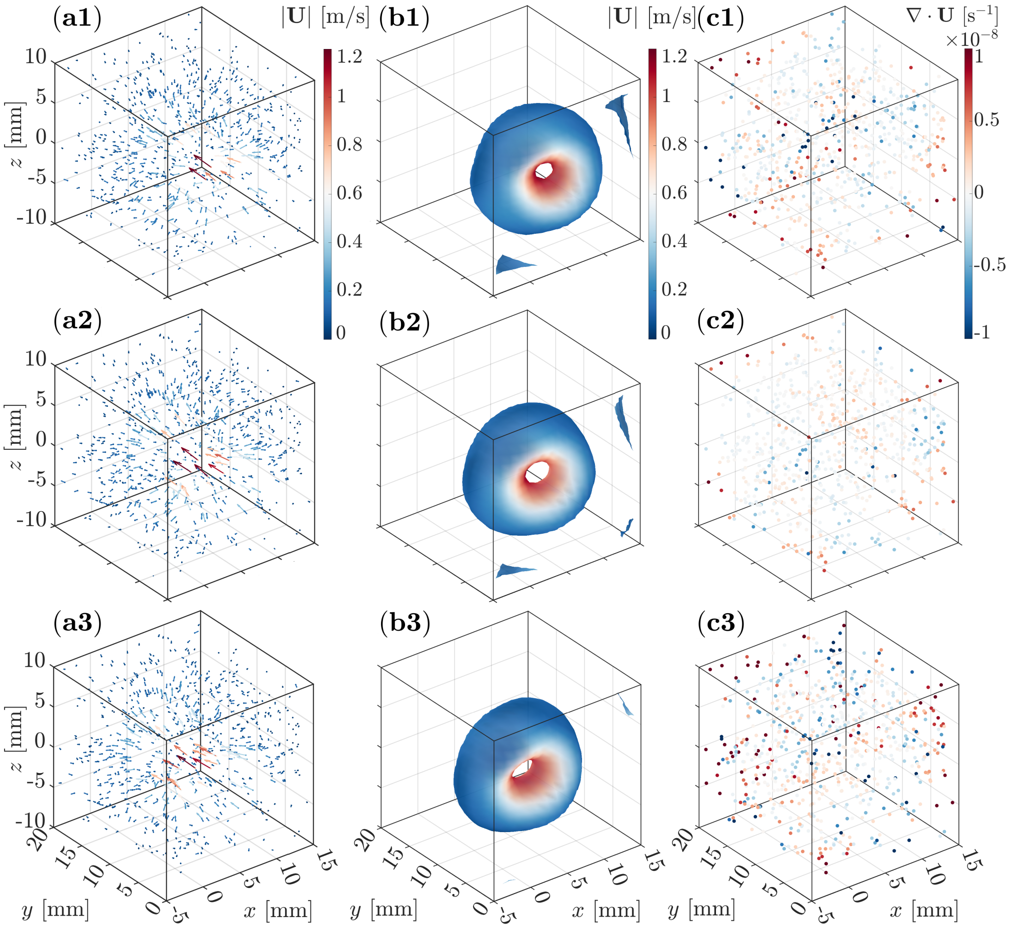


Figure 5.25: Reconstruction of velocity vector fields, iso-surfaces of coherent structures based on the Q-criterion, and scatter-plots of velocity divergence based on the pre-processed LPT data in the 3D experimental verification. The iso-surfaces are colored by particle velocities. Left column: reconstructed velocity vector fields. Central column: iso-surfaces of reconstructed coherent structure based on the Q-criterion (iso-value = ±0.0025). Right column: scatter-plots of the reconstructed velocity divergence at each measured particle. Top row to bottom: reconstruction from the third, sixth, and ninth frames.

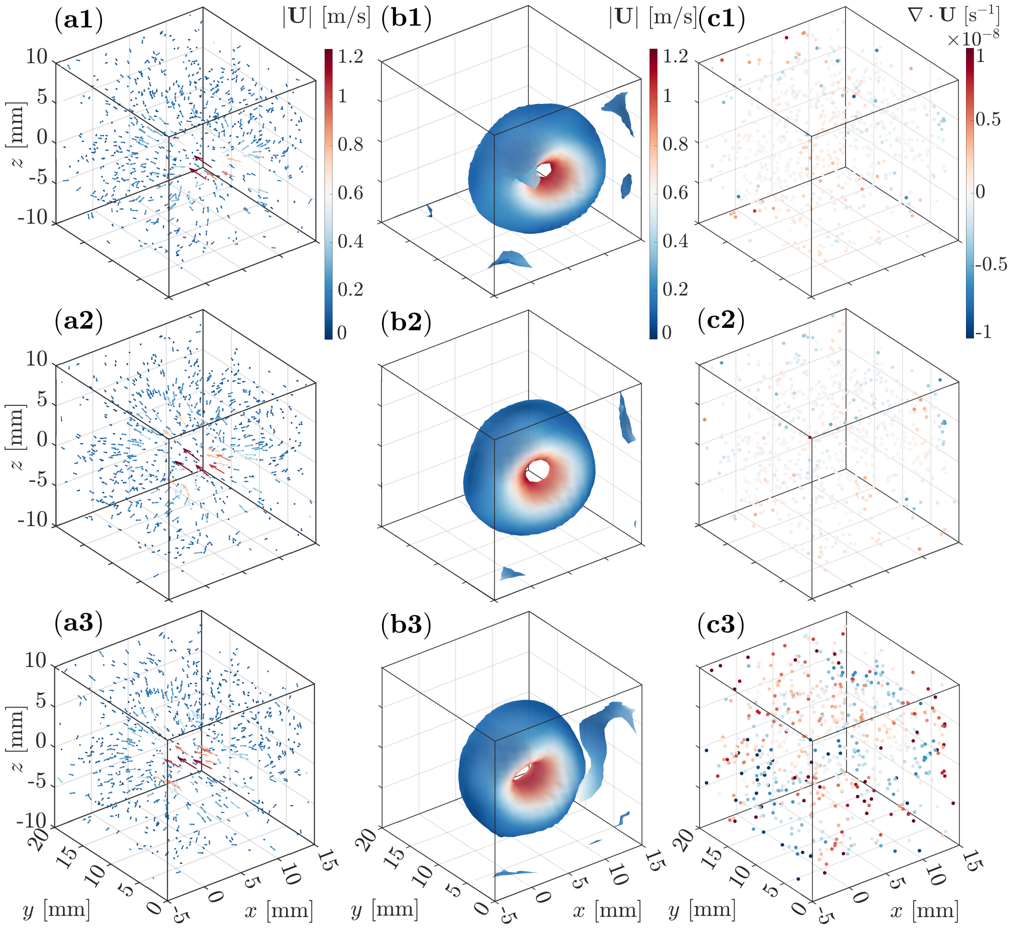


Figure 5.26: Reconstruction of velocity vector fields, iso-surfaces of coherent structures based on the Q-criterion, and scatter-plots of velocity divergence based on the contaminated LPT data in the 3D experimental verification. The iso-surfaces are colored by particle velocities. Left column: reconstructed velocity vector fields. Central column: iso-surfaces of reconstructed coherent structure based on the Q-criterion (iso-value = ±0.0025). Right column: scatter-plots of the reconstructed velocity divergence at each measured particle. Top row to bottom: reconstruction from the third, sixth, and ninth frames.

In addition to the coherent structures, strain- and rotation-rate tensors are also presented for an intuitive reconstruction assessment. Figure 5.27 and 5.28 show iso-surfaces of reconstructed strain-rate and rotation-rate tensors based on the pre-processed LPT data, respectively. The structures of the strain-rate tensor in Figure 5.27 can reveal key features of shear deformation, expansion, and compression of the jet. For example, as depicted in Figure 5.27, the two major tubular structures of $\tilde{S}_{12}$ were observed in the domain. These major structures developed along the y axis with two minor vortical structures with reversed colors warping them. As shown in Figure 5.29(a), viewing from the $+z$ axis, the two major structures with reversed colors indicated that the fluid parcels located at two sides of the center-line near the jet core underwent shear deformation in opposite directions. These deformations were caused by the nonuniform velocity distribution of the jet along the x axis. These shear deformations can be visualized as the faces of a fluid parcel close to the jet core (e.g., faces a and c) moving faster than those far from the jet core (e.g., faces b and d). In contrast, the two minor structures implied that the fluid parcels near the vortices experienced shear deformation in opposite directions to its closest major structure.

Similarly, the structures of the rotation-rate tensor in Figure 5.28 can unveil some key features of the flow rotation. For example, as shown in Figure 5.28, two large hemispherical structures with reversed colors, adherent to each other, appeared in the middle of the iso-surfaces of $\tilde{R}_{12}$. This indicates that the fluid parcels on the red iso-surfaces rotated along the $-z$ axis, while the fluid parcels on the blue iso-surfaces rotated along the $+z$ axis (see Figure 5.29(b) for illustration). The large empty areas in $\tilde{R}_{13}$ and $\tilde{R}_{31}$ implied that there was negligible rotation along the y axis.

The interpretation of the strain-rate and rotation-rate tensor was consistent with the behavior of the pulsing jet flow. In this jet flow, the jet core had higher speeds than the periphery jet areas, forming a velocity gradient along the x axis. The reversed flows created vortices that brought fluid parcels back. No significant rotation was generated along the y axis. The reconstructed strain- and rotation-rate results aligned well with the experimental observations, indicating that the CLS-RBF method is capable of reconstructing strain- and rotation-rate tensors.

In summary, the CLS-RBF Lagrangian reconstruction method is able to process experimental LPT data, i.e., reconstruct trajectories, velocities, and differential quantities based on experimental Lagrangian data. This reconstruction enjoys robustness and satisfies divergence-free constraints for incompressible flows. The CLS-RBF method can process low-quality LPT data, in which the data are highly contaminated and/or spatiotemporal resolution is low.

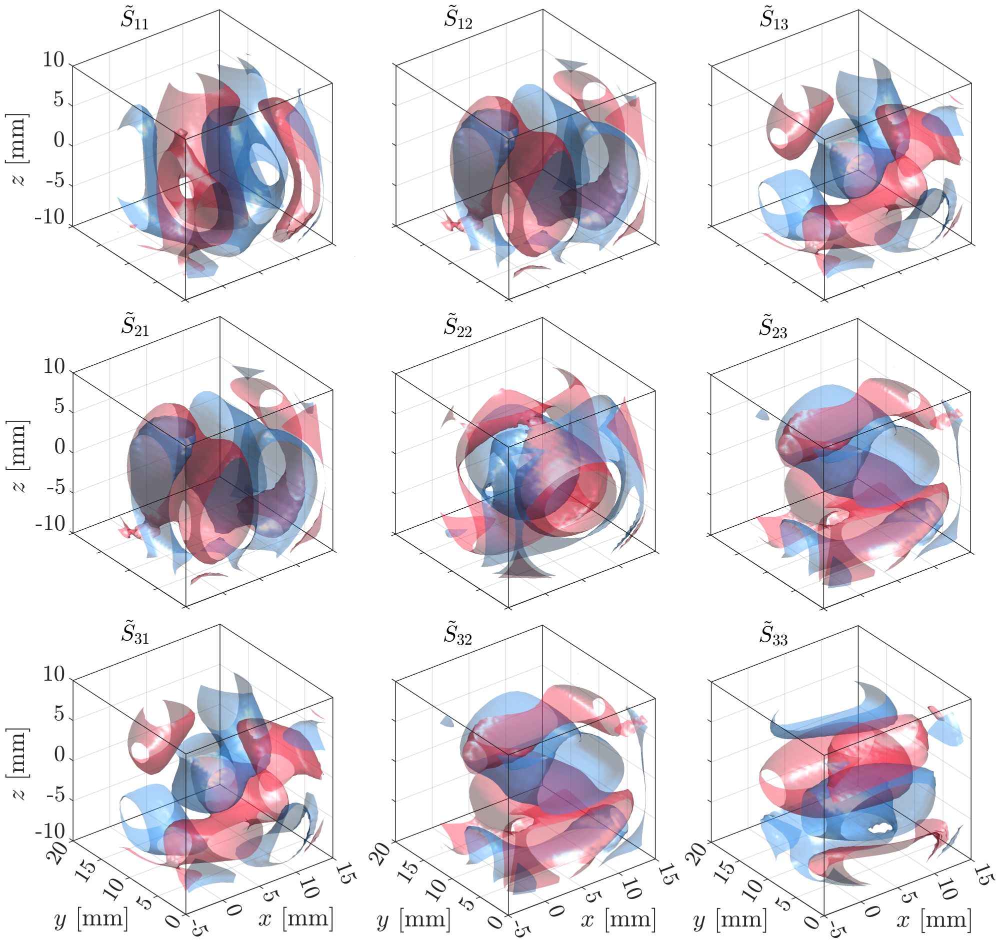


Figure 5.27: Iso-surfaces of reconstructed strain-rate tensors (iso-value = ±10) in the sixth frame of the 3D experimental verification. The colors red and blue correspond to positive and negative iso-values, respectively.

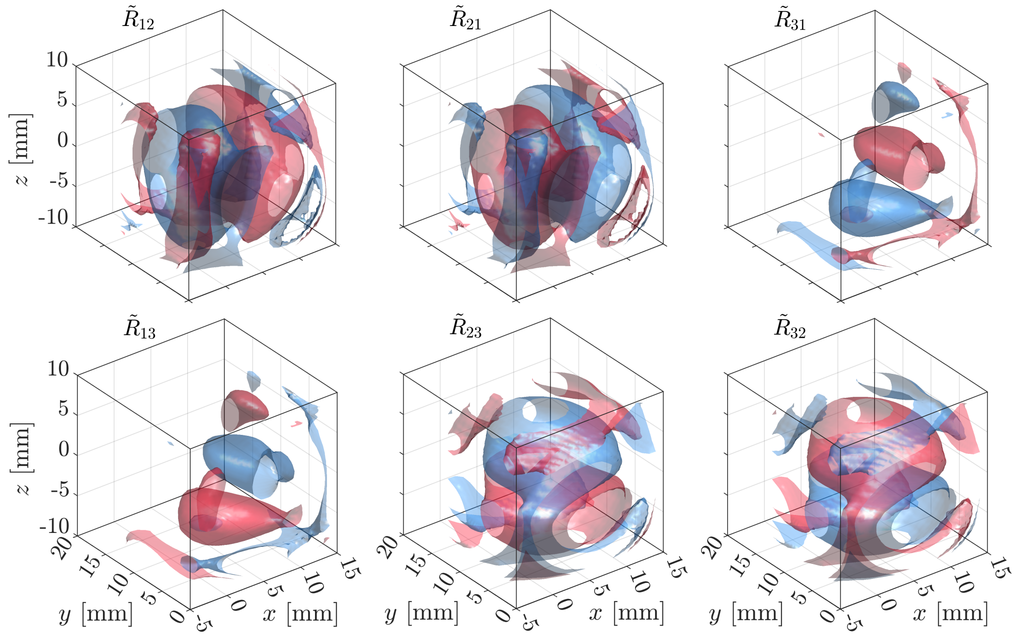


Figure 5.28: Iso-surfaces of reconstructed rotation-rate tensors (iso-value = ±10) in the sixth frame of the 3D experimental verification. The colors red and blue correspond to positive and negative iso-values, respectively.

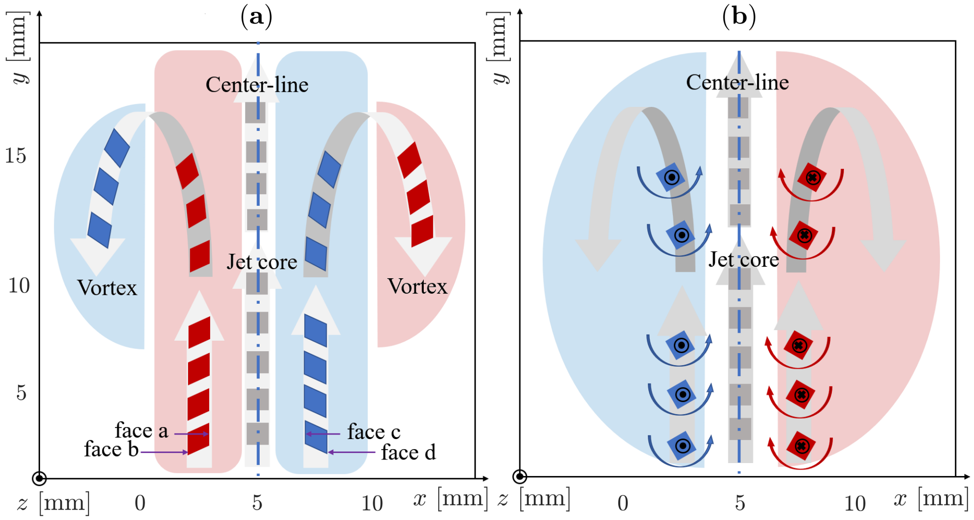


Figure 5.29: Schematic diagrams of shear deformation and rotation of fluid parcels based on $\tilde{S}_{12}$ and $\tilde{R}_{12}$ reconstruction. (a) Shear deformation of fluid parcels based on $\tilde{S}_{12}$ reconstruction in Figure 5.27; (b) rotation of fluid parcels based on $\tilde{R}_{12}$ reconstruction in Figure 5.28. Red, gray, and blue quadrilaterals: fluid parcels with positive, zero, and negative shear deformation or rotation, respectively; gray arrows: pathways of the jet flow; dash-dotted line: the center-line of the jet core. The diagram presents a slice of the domain at $z = 0$ mm.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, a novel 3D divergence-free Lagrangian flow field reconstruction method based on the constrained least squares (CLS) and stable radial basis function (RBF) algorithm is developed. The stable RBF, i.e., RBF-QR, serves as robust model functions of trajectories along each pathline (1D functions of the time) and of velocity fields in each frame (3D functions of the spatial coordinates). The CLS algorithm fits scattered data based on RBF-QR model functions and enforces divergence-free constraints for incompressible flows, ensuring that reconstruction is consistent with physical constraints. The proposed method consists of four steps (see Chapter 4). These four steps comprehensively reconstruct a Lagrangian flow field by integrating the CLS and RBF-QR algorithms. Besides, the temporal and spatial perspectives of flow fields are intrinsically connected and equally respected by the reconstruction.

Combining the CLS and RBF-QR, the proposed Lagrangian flow reconstruction method overcomes the limitations in the previous methods (see Chapter 2 and Table 2.1) and shows favorable features. The CLS-RBF Lagrangian reconstruction method can avoid data conversions from Lagrangian data onto Eulerian meshes, allowing flow field variables to be directly computed under the Lagrangian framework. The method also enforces divergence-free constraints for incompressible flows and can be extended to other physical constraints. Furthermore, it can accurately capture both high and low curvature trajectories with proper parameter settings. An immediate 'by-product' of the CLS-RBF method is smooth super-resolution in space and time, guaranteeing physical constraints at arbitrary time and location. These properties indicate that the CLS-RBF method could outperform previous methods and is promising for LPT data processing.

The validation and verification studies in this thesis demonstrate the competence of the CLS-RBF method. Validations based on the synthetic data show that the CLS-RBF method outperforms the baseline algorithms (i.e., polynomials and finite difference methods) widely used in the LPT community regarding trajectory and velocity reconstruction. Besides, the reconstructed particle trajectories enjoy lower errors than those in the synthetic data while preserving divergence-free constraints. The differential quantity reconstruction is also accurate and robust, which enables resolving flow structures based on noisy and low-resolution data. The verification base on a low-speed pulsing jet from experiments confirms the performance of the CLS-RBF method. In summary, the CLS-RBF Lagrangian reconstruction method can process LPT data even if the data are (highly) contaminated and have low spatiotemporal resolution, providing an accurate and robust reconstruction of particle trajectories, velocities, and differential quantities with divergence-free constraints for 3D incompressible flows.

6.2 Future Work

Here, some improvements of the current CLS-RBF Lagrangian reconstruction method are pointed out:

1. Incorporate the partition-of-unity method (PUM) to process large-scale LPT data with long trajectories. Large-scale LPT data with long trajectories (e.g., tens of thousands of particles in the domain recorded on tens frames or even more) can potentially lead to numerical instability and computational inefficiency if the current method is directly applied. The PUM can partition the flow domain and trajectories into spatial and temporal subdomains, allowing for localized flow field reconstruction and preserving global smoothness. Therefore, the CLS-RBF method can theoretically process arbitrarily large-scale LPT data with numerical stability and relatively low computational cost.

2. Upgrade the CLS to a weighted constrained least squares (WCLS) algorithm to enhance reconstruction accuracy and robustness. In experiments, the uncertainty of the particle location may vary over time and space. Particles with lower measurement uncertainty are considered more accurate and trustworthy than those with higher measurement uncertainty. Assigning higher weights to particles with low uncertainties under a WCLS framework increases their contribution to the computation. Thus,

the reconstructed functions tend to be close to the accurate and trustworthy data. This practice is expected to improve the quality of reconstruction.

3. Develop an adaptive approach that automatically chooses optimal parameters to fine-tune reconstruction for largely varying curvature trajectories. Both high and low curvature trajectories could appear simultaneously, especially for a complex flow in a large domain and particles are recorded on long trajectories. The current tests show that a high curvature trajectory needs a smaller temporal over-sampling ratio than that of a low curvature trajectory. However, the current version of the CLS-RBF method adopts fixed reconstruction parameters in the entire flow field. An improved trajectory reconstruction should assign independent parameters to different trajectories and frames.

4. Develop data assimilation or interpretable machine learning framework based on the CLS-RBF method for LPT data. The method proposed in this thesis can be considered a one-layer neural network, which is 'transparent' and interpretable to the users, and physical constraints can be explicitly imposed. These features are essential to data assimilation or interpretable machine learning tasks where constraints arise from physics (e.g., divergence-free in the present thesis). Other constraints, such as boundary conditions and governing equations, can also be enforced.

References

[1] AA Adamczyk and L Rimai. 2-Dimensional particle tracking velocimetry (PTV): technique and image processing algorithms. *Experiments in fluids*, 6(6):373–380, 1988.

[2] RJ Adrian. Dynamic ranges of velocity and spatial resolution of particle image velocimetry. *Measurement Science and Technology*, 8(12):1393, 1997.

[3] Ronald J Adrian. Particle-imaging techniques for experimental fluid mechanics. *Annual review of fluid mechanics*, 23(1):261–304, 1991.

[4] Ronald J Adrian and Jerry Westerweel. *Particle image velocimetry*. Number 30. Cambridge university press, 2011.

[5] Juan C Agüí and J Jimenez. On the performance of particle tracking. *Journal of fluid mechanics*, 185:447–468, 1987.

[6] Federico Alberini, Li Liu, EH Stitt, and MJH Simmons. Comparison between 3-D-PTV and 2-D-PIV for determination of hydrodynamics of complex fluids in a stirred vessel. *Chemical Engineering Science*, 171:189–203, 2017.

[7] John Anderson. *EBOOK: Fundamentals of Aerodynamics (SI units)*. McGraw Hill, 2011.

[8] Karl Johan Åström and Peter Eykhoff. System identificationâa survey. *Automatica*, 7(2):123–162, 1971.

[9] SJ Baek and SJ Lee. A new two-frame particle tracking algorithm using match probability. *Experiments in Fluids*, 22(1):23–32, 1996.

[10] RJM Bastiaans, GAJ Van der Plas, and RN Kieft. The performance of a new PTV algorithm applied in super-resolution PIV. *Experiments in Fluids*, 32(3):346–356, 2002.

[11] Steven J Beresh. Time-resolved particle image velocimetry. *Measurement Science and Technology*, 32(10):102003, 2021.

[12] Pascal Henry Biwole, Wei Yan, Yanhui Zhang, and Jean-Jacques Roux. A complete 3D particle tracking algorithm and its applications to the indoor airflow study. *Measurement Science and Technology*, 20(11):115403, 2009.

[13] Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[14] Jiri Blazek. *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann, 2015.

[15] Maxim Bobrov, Mikhail Hrebtov, Vladislav Ivashchenko, Rustam Mullyadzhanov, Alexander Seredkin, Mikhail Tokarev, Dinar Zaripov, Vladimir Dulin, and Dmitriy Markovich. Pressure evaluation from Lagrangian particle tracking data using a grid-free least-squares method. *Measurement Science and Technology*, 32(8):084014, 2021.

[16] Johannes Bosbach, Daniel Schanz, Philipp Godbersen, and Andreas Schröder. Spatially and temporally resolved measurements of turbulent Rayleigh-Bénard convection by Lagrangian particle tracking of long-lived helium-filled soap bubbles. In *Proceedings of 14th International Symposium on Particle Image Velocimetry*, volume 1, pages 1–12. ILLINOIS Tech/Paul V. Galvin Library, 2021.

[17] Kent A Bowker. Albert Einstein and meandering rivers. *Earth Sciences History*, 7(1):45, 1988.

[18] Justin A Boyan. Technical update: Least-squares temporal difference learning. *Machine learning*, 49(2):233–246, 2002.

[19] Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

[20] Matthew Bross, Thomas Fuchs, and Christian J Kähler. Analysis of an adverse pressure gradient turbulent boundary layer with 2d-and 3d-time-resolved particle tracking. In *5th International Conference on Experimental Fluid Mechanics ICEFM*, 2018.

[21] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021.

[22] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, 2001.

[23] LDC Casa and PS Krueger. Radial basis function interpolation of unstructured, three-dimensional, volumetric particle tracking velocimetry data. *Measurement Science and Technology*, 24(6):065304, 2013.

[24] Julie S Chalfant and Takashi Maekawa. Design for manufacturing using B-spline developable surfaces. *Journal of Ship Research*, 42(03):207–215, 1998.

[25] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks*, 10(5):1055–1064, 1999.

[26] Sheng Chen, Stephen A Billings, and Wan Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50(5):1873–1896, 1989.

[27] Liang Cheng and Steven Armfield. A simplified marker and cell method for unsteady flows on non-staggered grids. *International journal for numerical methods in fluids*, 21(1):15–34, 1995.

[28] C Cierpka, M Rossi, R Segura, F Mastrangelo, and CJ Kähler. A comparative analysis of the uncertainty of astigmatism-μPTV, stereo-μPIV, and μPIV. *Experiments in fluids*, 52:605–615, 2012.

[29] Christian Cierpka, Benjamin Lütke, and Christian J Kähler. Higher order multi-frame particle tracking velocimetry. *Experiments in Fluids*, 54(5):1–12, 2013.

[30] Elaine Cohen, Tom Lyche, and Richard Riesenfeld. Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer graphics and image processing*, 14(2):87–111, 1980.

[31] Richard K Cohn and Manoochehr M Koochesfahani. The accuracy of remapping irregularly spaced velocity data onto a regular grid and the computation of vorticity. *Experiments in fluids*, 29(Suppl 1):S061–S069, 2000.

[32] Iain G Currie. *Fundamental mechanics of fluids*. CRC press, 2016.

[33] Dana Dabiri and Charles Pecora. *Particle tracking velocimetry.* IOP Publishing Bristol, 2020.

[34] Stuart B Dalziel. Decay of rotating turbulence: some particle tracking experiments. *Applied scientific research*, 49(3):217–244, 1992.

[35] Carl De Boor. On calculating with B-splines. *Journal of Approximation theory*, 6(1):50–62, 1972.

[36] Annette J Dobson and Adrian G Barnett. *An introduction to generalized linear models.* Chapman and Hall/CRC, 2018.

[37] Thierry Dombre, Uriel Frisch, John M Greene, Michel Hénon, A Mehr, and Andrew M Soward. Chaotic streamlines in the ABC flows. *Journal of Fluid Mechanics*, 167:353–391, 1986.

[38] Th Dracos. *Three-Dimensional Velocity and Vorticity Measuring and Image Analysis Techniques: Lecture Notes from the Short Course held in Zürich, Switzerland, 3–6 September 1996*, volume 4. Springer Science & Business Media, 1996.

[39] Norman R Draper and Harry Smith. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998.

[40] Sergey V Ershkov. About existence of stationary points for the Arnold–Beltrami–Childress (ABC) flow. *Applied Mathematics and Computation*, 276:379–383, 2016.

[41] Francesco Fambri and Michael Dumbser. Spectral semi-implicit and space-time discontinuous Galerkin methods for the incompressible Navier–Stokes equations on staggered Cartesian grids. *Applied Numerical Mathematics*, 110:41–74, 2016.

[42] Gregory E Fasshauer. *Meshfree approximation methods with MATLAB*, volume 6. World Scientific, 2007.

[43] Simone Ferrari and Lionel Rossi. Particle tracking velocimetry and accelerometry (PTVA) measurements applied to quasi-two-dimensional multi-scale flows. *Experiments in fluids*, 44(6):873–886, 2008.

[44] Joel H Ferziger, Milovan Perić, and Robert L Street. *Computational methods for fluid dynamics*, volume 3. Springer, 2002.

[45] Bengt Fornberg, Elisabeth Larsson, and Natasha Flyer. Stable computations with Gaussian radial basis functions. *SIAM Journal on Scientific Computing*, 33(2):869–892, 2011.

[46] Bengt Fornberg, Erik Lehto, and Collin Powell. Stable calculation of Gaussian-based RBF-FD stencils. *Computers & Mathematics with Applications*, 65(4):627–637, 2013.

[47] Bengt Fornberg and Cécile Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM Journal on Scientific Computing*, 30(1):60–80, 2008.

[48] Bengt Fornberg and Grady Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers & Mathematics with Applications*, 48(5-6):853–867, 2004.

[49] John Fox. *Applied regression analysis and generalized linear models*. Sage Publications, 2015.

[50] Richard Franke. Scattered data interpolation: tests of some methods. *Mathematics of computation*, 38(157):181–200, 1982.

[51] Sijie Fu, Pascal Henry Biwole, and Christian Mathis. Particle tracking velocimetry for indoor airflow field: A review. *Building and Environment*, 87:34–44, 2015.

[52] Rao Veerabhadra Garimella. A simple introduction to moving least squares and local regression estimation. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2017.

[53] Sebastian Gesemann, Florian Huhn, Daniel Schanz, and Andreas Schröder. From noisy particle tracks to velocity, acceleration and pressure fields using B-splines and penalties. In *18th international symposium on applications of laser and imaging techniques to fluid mechanics, Lisbon, Portugal*, volume 4, 2016.

[54] Faranak Gholampour, Esmail Hesameddini, and Ameneh Taleei. A global RBF-QR collocation technique for solving two-dimensional elliptic problems involving arbitrary interface. *Engineering with Computers*, 37(4):3793–3811, 2021.

[55] Melissa A Green, Clarence W Rowley, and George Haller. Detection of Lagrangian coherent structures in three-dimensional turbulence. *Journal of Fluid Mechanics*, 572:111–120, 2007.

[56] George Haller. An objective definition of a vortex. *Journal of fluid mechanics*, 525:1–26, 2005.

[57] George Haller. Lagrangian coherent structures. *Annual review of fluid mechanics*, 47:137–162, 2015.

[58] Rolland L Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, 76(8):1905–1915, 1971.

[59] Francis H Harlow and J Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, 8(12):2182–2189, 1965.

[60] YA Hassan and RE Canaan. Full-field bubbly flow velocity measurements using a multiframe particle tracking technique. *Experiments in fluids*, 12(1):49–60, 1991.

[61] Yassin A Hassan, Thomas K Blanchat, and CH Seeley Jr. PIV flow visualisation using particle tracking techniques. *Measurement Science and Technology*, 3(7):633, 1992.

[62] M Holzner, A Liberzon, N Nikitin, W Kinzelbach, and A Tsinober. Small-scale aspects of flows in proximity of the turbulent/nonturbulent interface. *Physics of fluids*, 19(7):071702, 2007.

[63] Klaus Hoyer, Markus Holzner, Beat Lüthi, Michele Guala, Alexander Liberzon, and Wolfgang Kinzelbach. 3D scanning particle tracking velocimetry. *Experiments in Fluids*, 39(5):923–934, 2005.

[64] Guang-Bin Huang, Lei Chen, Chee Kheong Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006.

[65] HT Huang, HE Fiedler, and JJ Wang. Limitation and improvement of PIV: Part I: limitation of conventional techniques due to deformation of particle image patterns. *Experiments in fluids*, 15(3):168–174, 1993.

[66] HT Huang, HE Fiedler, and JJ Wang. Limitation and improvement of PIV: part II: particle image distortion, a novel technique. *Experiments in fluids*, 15(4-5):263–273, 1993.

[67] Florian Huhn, Daniel Schanz, Sebastian Gesemann, Uwe Dierksheide, Remco van de Meerendonk, and Andreas Schröder. Large-scale volumetric flow measurement in a pure thermal plume by dense tracking of helium-filled soap bubbles. *Experiments in Fluids*, 58(9):1–19, 2017.

[68] Florian Huhn, Daniel Schanz, Peter Manovski, Sebastian Gesemann, and Andreas Schröder. Time-resolved large-scale volumetric pressure fields of an impinging jet from dense Lagrangian particle tracking. *Experiments in Fluids*, 59(5):1–16, 2018.

[69] Florian Huhn, Andreas Schröder, Daniel Schanz, Sebastian Gesemann, and Peter Manovski. Estimation of time-resolved 3D pressure fields in an impinging jet flow from dense Lagrangian particle tracking. In *Proceedings 18th International Symposium on Flow Visualization*. ETH Zurich, 2018.

[70] Julian CR Hunt, Alan A Wray, and Parviz Moin. Eddies, streams, and convergence zones in turbulent flows. *Studying turbulence using numerical simulation databases, 2. Proceedings of the 1988 summer program*, 1988.

[71] AKM Fazle Hussain. Coherent structures and turbulence. *Journal of Fluid Mechanics*, 173:303–356, 1986.

[72] T Ido, Y Murai, and F Yamamoto. Postprocessing algorithm for particle-tracking velocimetry based on ellipsoidal equations. *Experiments in Fluids*, 32(3):326–336, 2002.

[73] Takehiro Ido and Yuichi Murai. A recursive interpolation algorithm for particle tracking velocimetry. *Flow Measurement and Instrumentation*, 17(5):267–275, 2006.

[74] Thomas Janke, R Schwarze, and K Bauer. Part2track: a MATLAB package for double frame and time resolved particle tracking velocimetry. *SoftwareX*, 11:100413, 2020.

[75] Young Jin Jeon, Markus Müller, and Dirk Michaelis. Fine scale reconstruction (VIC#) by implementing additional constraints and coarse-grid approximation into VIC+. *Experiments in Fluids*, 63(4):70, 2022.

[76] Young Jin Jeon, Jan FG Schneiders, Markus Müller, Dirk Michaelis, and Bernhard Wieneke. 4D flow field reconstruction from particle tracks by VIC+ with additional constraints and multigrid approximation. In *Proceedings 18th International Symposium on Flow Visualization*. ETH Zurich, 2018.

[77] Jinhee Jeong and Fazle Hussain. On the identification of a vortex. *Journal of fluid mechanics*, 285:69–94, 1995.

[78] Christian J Kähler, Sven Scharnowski, and Christian Cierpka. On the resolution limit of digital particle image velocimetry. *Experiments in fluids*, 52:1629–1639, 2012.

[79] Nobuhide Kasagi and Akio Matsunaga. Three-dimensional particle-tracking velocimetry measurement of turbulence statistics and energy budget in a backward-facing step flow. *International journal of heat and fluid flow*, 16(6):477–485, 1995.

[80] Nobuhide Kasagi and Koichi Nishino. Probing turbulence with three-dimensional particle-tracking velocimetry. *Experimental thermal and fluid science*, 4(5):601–612, 1991.

[81] Richard D Keane and Ronald J Adrian. Theory of cross-correlation analysis of PIV images. *Applied scientific research*, 49:191–215, 1992.

[82] S Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

[83] B Khalighi. Study of the intake tumble motion by flow visualization and particle tracking velocimetry. *Experiments in Fluids*, 10(4):230–236, 1991.

[84] Bahram Khalighi and Yong H Lee. Particle tracking velocimetry: an automatic image processing algorithm. *Applied Optics*, 28(20):4328–4332, 1989.

[85] Ali Rahimi Khojasteh, Yin Yang, Dominique Heitz, and Sylvain Laizet. Lagrangian coherent track initialization. *Physics of Fluids*, 33(9):095113, 2021.

[86] Dong Kim, Daniel Schanz, Matteo Novara, Hyunduk Seo, Youngwoo Kim, Andreas Schröder, and Kyung Chun Kim. Experimental study of turbulent bubbly jet. Part 1. Simultaneous measurement of three-dimensional velocity fields of bubbles and water. *Journal of Fluid Mechanics*, 941, 2022.

[87] Mirae Kim, Daniel Schanz, Matteo Novara, Andreas Schröder, and Kyung Chun Kim. Volumetric Lagrangian particle tracking measurements of jet impingement on convex cylinder. In *14th International Symposium on Particle Image Velocimetry*, volume 1, 2021.

[88] Todd A Kingston, Taylor A Geick, Teshia R Robinson, and Theodore J Heindel. Characterizing 3D granular flow structures in a double screw mixer using X-ray particle tracking velocimetry. *Powder technology*, 278:211–222, 2015.

[89] M Kinzel, M Wolf, M Holzner, B Lüthi, C Tropea, and W Kinzelbach. Simultaneous two-scale 3D-PTV measurements in turbulence under the influence of system rotation. *Experiments in fluids*, 51:75–82, 2011.

[90] Matthias Kinzel. *Experimental investigation of turbulence under the influence of confinement and rotation*. PhD thesis, Darmstadt, Techn. Univ., Diss., 2010, 2010.

[91] Matthias Kinzel, Markus Holzner, Beat Lüthi, Alexander Liberzon, Cameron Tropea, Wolfgang Kinzelbach, and M Oberlack. Two-scale experiments via Particle Tracking Velocimetry: a feasibility study. *Imaging Measurement Methods, Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, 106:103–111, 2009.

[92] J Kitzhofer, P Westfeld, O Pust, T Nonn, HG Maas, and C Brücker. Estimation of 3D deformation and rotation rate tensor from volumetric particle data via 3D least squares matching. In *Proc. 15th Int. Symp. on Applications of Laser Techniques to Fluid Mechanics*, pages 3–1, 2010.

[93] Gary D Knott. *Interpolating cubic splines*, volume 18. Springer Science & Business Media, 1999.

[94] Aloke Kumar, Christian Cierpka, Stuart J Williams, Christian J Kähler, and Steven T Wereley. 3D3C velocimetry measurements of an electrothermal microvortex using wavefront deformation PTV and a single camera. *Microfluidics and nanofluidics*, 10:355–365, 2011.

[95] Pijush K Kundu, Ira M Cohen, and David R Dowling. *Fluid mechanics*. Academic press, 2015.

[96] Boris Khattatov William Lahoz and Richard Menard. *Data assimilation*. Springer, 2010.

[97] Culbert B Laney. *Computational gasdynamics*. Cambridge university press, 1998.

[98] Elisabeth Larsson, Erik Lehto, Alfa Heryudono, and Bengt Fornberg. Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions. *SIAM Journal on Scientific Computing*, 35(4):A2096–A2119, 2013.

[99] Elisabeth Larsson, Victor Shcherbakov, and Alfa Heryudono. A least squares radial basis function partition of unity method for solving PDEs. *SIAM Journal on Scientific Computing*, 39(6):A2538–A2563, 2017.

[100] Kody Law, Andrew Stuart, and Kostas Zygalakis. Data assimilation. *Cham, Switzerland: Springer*, 214:52, 2015.

[101] Charles L Lawson and Richard J Hanson. *Solving least squares problems.* SIAM, 1995.

[102] Julie Lelouvetel, Masashi Nakagawa, Yohei Sato, and Koichi Hishida. Effect of bubbles on turbulent kinetic energy transport in downward flow measured by time-resolved PTV. *Experiments in fluids*, 50:813–823, 2011.

[103] Lanyu Li, Nazmus Sakib, and Zhao Pan. Robust Strain/Rotation-Rate Tensor Reconstruction Based on Least Squares RBF-QR for 3D Lagrangian Particle Tracking. In *Fluids Engineering Division Summer Meeting*, volume 85833, page V001T02A009. American Society of Mechanical Engineers, 2022.

[104] Lanyu Li, Prabu Sellappan, Peter Schmid, Jean-Pierre Hickey, Louis Cattafesta, and Zhao Pan. Lagrangian strain-and rotation-rate tensor evaluation based on multi-pulse particle tracking velocimetry (MPTV) and radial basis functions (RBFs). In *14th International Symposium on Particle Image Velocimetry*, volume 1, 2021.

[105] Nan-Ying Liang, Guang-Bin Huang, Paramasivan Saratchandran, and Narasimhan Sundararajan. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on neural networks*, 17(6):1411–1423, 2006.

[106] Chaoqun Liu, Yi-sheng Gao, Xiang-rui Dong, Yi-qian Wang, Jian-ming Liu, Yuning Zhang, Xiao-shu Cai, and Nan Gui. Third generation of vortex identification methods: Omega and Liutex/Rortex based systems. *Journal of Hydrodynamics*, 31(2):205–223, 2019.

[107] Shutian Liu, Quhao Li, Junhuan Liu, Wenjiong Chen, and Yongcun Zhang. A realization method for transforming a topology optimization design into additive manufacturing structures. *Engineering*, 4(2):277–285, 2018.

[108] Beat Lüthi. *Some aspects of strain, vorticity and material element dynamics as measured with 3D particle tracking velocimetry in a turbulent flow.* PhD thesis, ETH Zurich, 2002.

[109] Beat Lüthi, Arkady Tsinober, and Wolfgang Kinzelbach. Lagrangian measurement of vorticity dynamics in turbulent flow. *Journal of Fluid mechanics*, 528:87–118, 2005.

[110] Ives Macêdo, Joao Paulo Gois, and Luiz Velho. Hermite radial basis functions implicits. In *Computer Graphics Forum*, volume 30, pages 27–42. Wiley Online Library, 2011.

[111] NA Malik, Th Dracos, and DA Papantoniou. Particle tracking velocimetry in three-dimensional flows: Part II: Particle tracking. *Experiments in fluids*, 15:279–294, 1993.

[112] Nadeem A Malik and Th Dracos. Lagrangian PTV in 3D flows. *Applied Scientific Research*, 51:161–166, 1993.

[113] Nadeem A Malik and Th Dracos. Interpolation schemes for three-dimensional velocity fields from scattered data using Taylor expansions. *Journal of Computational Physics*, 119(2):231–243, 1995.

[114] Zhiping Mao, Ameya D Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.

[115] Sean McKee, Murilo F Tomé, Valdemir G Ferreira, José A Cuminato, Antonio Castelo, FS Sousa, and Norberto Mangiavacchi. The MAC method. *Computers & Fluids*, 37(8):907–930, 2008.

[116] Fadi Mishriky and Paul Walsh. Towards understanding the influence of gradient reconstruction methods on unstructured flowsimulations. *Transactions of the Canadian society for mechanical engineering*, 41(2):169–179, 2017.

[117] Nathan J Neeteson, Samik Bhattacharya, David E Rival, Dirk Michaelis, Daniel Schanz, and Andreas Schröder. Pressure-field extraction from Lagrangian flow measurements: first experiences with 4D-PTV data. *Experiments in Fluids*, 57:1–18, 2016.

[118] Nathan J Neeteson and David E Rival. Pressure-field extraction on unstructured flow data using a Voronoi tessellation-based networking algorithm: a proof-of-principle study. *Experiments in Fluids*, 56:1–13, 2015.

[119] Koichi Nishino, Nobuhide Kasagi, and Masaru Hirata. Three-dimensional particle tracking velocimetry based on automated digital image processing. 1989.

[120] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[121] Matteo Novara, Daniel Schanz, Nico Reuther, Christian J Kähler, and Andreas Schröder. Lagrangian 3D particle tracking in high-speed flows: Shake-The-Box for multi-pulse systems. *Experiments in fluids*, 57:1–20, 2016.

[122] Christopher C Paige and Michael A Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):43–71, 1982.

[123] D Papantoniou and Th Dracos. Analyzing 3-D turbulent motions in open channel flow by use of stereoscopy and particle tracking. In *Advances in Turbulence 2: Proceedings of the Second European Turbulence Conference, Berlin, August 30–September 2, 1988*, pages 278–285. Springer, 1989.

[124] D Papantoniou and Th Dracos. Lagrangian statistics in open channel flow by 3-D particle tracking velocimetry. *Eng. Turb. Model. Expt. Elsevier*, 1990.

[125] JS Park and KD Kihm. Three-dimensional micro-PTV using deconvolution microscopy. *Experiments in Fluids*, 40:491–499, 2006.

[126] Charles Pecora. Particle Tracking Velocimetry: A Review. 2018.

[127] Les Piegl. On NURBS: a survey. *IEEE Computer Graphics and Applications*, 11(01):55–71, 1991.

[128] Stephen B Pope and Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000.

[129] Shijie Qin and Shijun Liao. Ultra-chaos in the ABC flow and its relationships to turbulence. *arXiv preprint arXiv:2204.14056*, 2022.

[130] RG Racca and JM Dewey. A method for automatic particle tracking in a three-dimensional flow field. *Experiments in Fluids*, 6(1):25–32, 1988.

[131] Markus Raffel, Christian E Willert, Jürgen Kompenhans, et al. *Particle image velocimetry: a practical guide*, volume 2. Springer, 1998.

[132] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[133] Ethirajan Rathakrishnan. *Instrumentation, measurements, and experiments in fluids*. CRC press, 2007.

[134] Manuele Romano, Federico Alberini, L Liu, Mark Simmons, and E Stitt. Lagrangian investigations of a stirred tank fluid flow using 3d-PTV. *Chemical Engineering Research and Design*, 172:71–83, 2021.

[135] Giuseppe A Rosi, Andrew M Walker, and David E Rival. Lagrangian coherent structure identification using a Voronoi tessellation-based networking algorithm. *Experiments in Fluids*, 56:1–14, 2015.

[136] GI Roth and J Katz. Five techniques for increasing the speed and accuracy of PIV interrogation. *Measurement Science and Technology*, 12(3):238, 2001.

[137] Carl Runge. Über empirische funktionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik*, 46(224-243):20, 1901.

[138] Md Nazmus Sakib. *Pressure from PIV for an Oscillating Internal Flow*. PhD thesis, Utah State University, 2022.

[139] Nazmus Sakib, Alexander Mychkovsky, James Wiswall, Randy Samaroo, and Barton Smith. Effect of the boundary conditions, temporal, and spatial resolution on the pressure from PIV for an oscillating flow. In *14th International Symposium on Particle Image Velocimetry*, volume 1, 2021.

[140] Daniel Schanz, Reinhard Geisler, Christina Voss, Matteo Novara, Tobias Jahn, Uwe Dierksheide, and Andreas Schröder. Large scale volumetric flow measurements in a turbulent boundary layer. In *Proceedings 18th international symposium on flow visualization*. ETH Zurich, 2018.

[141] Daniel Schanz, Sebastian Gesemann, and Andreas Schröder. Shake-The-Box: Lagrangian particle tracking at high particle image densities. *Experiments in fluids*, 57(5):1–27, 2016.

[142] Daniel Schanz, Andreas Schröder, Sebastian Gesemann, Dirk Michaelis, and Bernhard Wieneke. Shake the box: a highly efficient and accurate tomographic particle tracking velocimetry (TOMO-PTV) method using prediction of particle positions. 2013.

[143] Dominik Schillinger, Luca Dede, Michael A Scott, John A Evans, Michael J Borden, Ernst Rank, and Thomas JR Hughes. An isogeometric design-through-analysis

methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249:116–150, 2012.

[144] Peter J Schmid. Data-driven and operator-based tools for the analysis of turbulent flows. In *Advanced Approaches in Turbulence*, pages 243–305. Elsevier, 2021.

[145] Jan FG Schneiders, Richard P Dwight, and Fulvio Scarano. Time-supersampling of 3D-PIV measurements with vortex-in-cell simulation. *Experiments in Fluids*, 55:1–15, 2014.

[146] Jan FG Schneiders and Fulvio Scarano. Dense velocity reconstruction from tomographic PTV with material derivatives. *Experiments in fluids*, 57:1–22, 2016.

[147] Johannes Schobesberger, Petr Lichtneger, Christoph Hauer, Helmut Habersack, and Christine Sindelar. Three-dimensional coherent flow structures during incipient particle motion. *Journal of Hydraulic Engineering*, 146(5):04020027, 2020.

[148] Isaac Jacob Schoenberg. Spline functions and the problem of graduation. *Proceedings of the National Academy of Sciences*, 52(4):947–950, 1964.

[149] Andreas Schröder and Daniel Schanz. 3D Lagrangian Particle Tracking in Fluid Mechanics. *Annual Review of Fluid Mechanics*, 55, 2023.

[150] Andreas Schröder, Daniel Schanz, Reinhard Geisler, Tobias Jahn, Christian E Willert, Benjamin Leclaire, and Quentin Gallas. A characterization of the flow around a surface mounted cube by TR-PIV and 3D Shake-The-Box. In *13th International Symposium on Particle Image Velocimetry*, number 209, pages 285–300. Universität der Bundeswehr München: AtheneForschung, 2019.

[151] Andreas Schröder, Daniel Schanz, Reinhard Geisler, Matteo Novara, and Christian Willert. Near-wall turbulence characterization using 4D-PTV Shake-The-Box. 2015.

[152] Andreas Schröder, Daniel Schanz, Dirk Michaelis, Christian Cierpka, Sven Scharnowski, and Christian J Kähler. Advances of PIV and 4D-PTV Shake-The-Box for turbulent flow analysis–the flow over periodic hills. *Flow, Turbulence and Combustion*, 95:193–209, 2015.

[153] Andreas Schröder, Daniel Schanz, Christoph Roloff, and Dirk Michaelis. Lagrangian and Eulerian dynamics of coherent structures in turbulent flow over periodic hills using time-resolved tomo PIV and 4D-PTV Shake-the-box 17th Int. In *Symp. on*

Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal, pages 07–10, 2014.

[154] Andreas Schröder, Chris Willert, Daniel Schanz, Reinhard Geisler, Tobias Jahn, Quentin Gallas, and Benjamin Leclaire. The flow around a surface mounted cube: a characterization by time-resolved PIV, 3D Shake-The-Box and LBM simulation. *Experiments in Fluids*, 61(9):1–22, 2020.

[155] Prabu Sellappan, Farrukh S Alvi, and Louis N Cattafesta. Lagrangian and Eulerian measurements in high-speed jets using Multi-Pulse Shake-The-Box and fine scale reconstruction (VIC#). *Experiments in Fluids*, 61:1–17, 2020.

[156] Shawn C Shadden. Lagrangian coherent structures. *Transport and mixing in laminar flows: from microfluidics to oceanic currents*, pages 59–89, 2011.

[157] Pietro Sperotto, Sandra Pieraccini, and Miguel A Mendez. A meshless method to compute pressure fields from image velocimetry. *Measurement Science and Technology*, 33(9):094005, 2022.

[158] Yw Yvonne Stegeman. Particle tracking velocimetry. 1995.

[159] Heinrich Stüer and Stefan Blaser. Interpolation of scattered 3D PTV data to a regular grid. *Flow, turbulence and combustion*, 64:215–232, 2000.

[160] Yasuhide Takano, Goji T Etoh, and Gozo Tsujimoto. Measurements of vortex generation and motion at wind wave surfaces. *Coastal Engineering Proceedings*, (33):28–28, 2012.

[161] K Takehara and TG Etoh. Accurate estimation of vorticity from PTV velocity distribution. In *Proc. of 8th International Symposium on Particle Image Velocimetry-PIV09*, volume 185, 2009.

[162] K Takehara and TG Etoh. Direct evaluation method of vorticity combined with PTV. In *Selected Papers from the 31st International Congress on High-Speed Imaging and Photonics*, volume 10328. SPIE, 2017.

[163] Maurizio Tavelli and Michael Dumbser. A staggered space-time discontinuous Galerkin method for the three-dimensional incompressible Navier–Stokes equations on unstructured tetrahedral meshes. *Journal of Computational Physics*, 319:294–323, 2016.

[164] Stavros Tavoularis. *Measurement in fluid mechanics.* Cambridge University Press, 2005.

[165] Geoffrey Ingram Taylor and Albert Edward Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 158(895):499–521, 1937.

[166] Cameron Tropea, Alexander L Yarin, John F Foss, et al. *Springer handbook of experimental fluid mechanics*, volume 1. Springer, 2007.

[167] PL Van Gent, Dirk Michaelis, BW Van Oudheusden, P É Weiss, Roeland de Kat, Angeliki Laskari, Young Jin Jeon, Laurent David, Daniel Schanz, Florian Huhn, et al. Comparative assessment of pressure field reconstructions from particle image velocimetry measurements and Lagrangian particle tracking. *Experiments in Fluids*, 58:1–23, 2017.

[168] Tony Van Gestel, Johan AK Suykens, Bart Baesens, Stijn Viaene, Jan Vanthienen, Guido Dedene, Bart De Moor, and Joos Vandewalle. Benchmarking least squares support vector machine classifiers. *Machine learning*, 54(1):5–32, 2004.

[169] Prakash Vedula and RJ Adrian. Optimal solenoidal interpolation of turbulent vector fields: application to PTV and super-resolution PIV. *Experiments in fluids*, 39:213–221, 2005.

[170] Michel Verhaegen and Vincent Verdult. *Filtering and system identification: a least squares approach.* Cambridge university press, 2007.

[171] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method.* Pearson education, 2007.

[172] Malene H Vested, F Gökhan Ergin, Stefan Carstensen, and Erik D Christensen. High-resolution planar two-component PTV measurements in a breaking wave. In *19th International Symposium on the Application of Laser and Imaging Techniques to Fluid Mechanics, Springer-Verlag*, 2018.

[173] Marko Virant and Themistocles Dracos. 3D PTV and its application on Lagrangian motion. *Measurement science and technology*, 8(12):1539, 1997.

[174] Longwei Wang, Chong Pan, Jianhua Liu, and Chujiang Cai. Ratio-cut background removal method and its application in near-wall PTV measurement of a turbulent boundary layer. *Measurement Science and Technology*, 32(2):025302, 2020.

[175] Jerry Westerweel. On velocity gradients in PIV interrogation. *Experiments in Fluids*, 44:831–842, 2008.

[176] Bernhard Wieneke. Iterative reconstruction of volumetric particle distribution. *Measurement Science and Technology*, 24(2):024008, 2012.

[177] Jochen Willneff. *A spatio-temporal matching algorithm for 3D particle tracking velocimetry.* PhD thesis, ETH Zurich, 2003.

[178] Charles D Woodward. Cross-sectional design of B-spline surfaces. *Computers & graphics*, 11(2):193–201, 1987.

[179] Grady B Wright and Bengt Fornberg. Stable computations with flat radial basis functions using vector-valued rational approximations. *Journal of Computational Physics*, 331:137–156, 2017.

[180] Zhouwang Yang, Jiansong Deng, and Falai Chen. Fitting unorganized point clouds with active implicit B-spline curves. *The Visual Computer*, 21:831–839, 2005.

[181] Jie-min Zhan, Yu-tian Li, Wing-hong Onyx Wai, and Wen-qing Hu. Comparison between the Q criterion and Rortex in the application of an in-stream structure. *Physics of Fluids*, 31(12):121701, 2019.

[182] Yuning Zhang, Kaihua Liu, Haizhen Xian, and Xiaoze Du. A review of methods for vortex identification in hydroturbines. *Renewable and Sustainable Energy Reviews*, 81:1269–1285, 2018.

[183] Xiao-Hua Zhao, Keng-Huat Kwek, JI-Bin Li, and Ke-Lei Huang. Chaotic and resonant streamlines in the ABC flow. *SIAM Journal on Applied Mathematics*, 53(1):71–77, 1993.

APPENDICES

Appendix A

Baseline Algorithms

A.1 Finite difference method

Since the finite difference method (FDM) is only used to evaluate velocities based on the coordinates of particles, the particle locations that are output from the FDM are assumed to be the same as those in the synthetic data. For example, in the x direction, $\tilde{x}(\hat{t}_\kappa) = \hat{x}(\hat{t}_\kappa)$ in the κ-th frame.

For velocity reconstruction, for example in the x direction, the 1st FDM calculates velocity component u in a given frame by the forward Euler method:

$$\tilde{u}(\hat{t}_\kappa) = \frac{\hat{x}_{\kappa+1} - \hat{x}_\kappa}{\Delta t} \tag{A.1}$$

where $\hat{x}_\kappa$ is the particle coordinate along the x direction in the κ-th frame measured by LPT experiments, Δt is the interval between two consecutive frames, $\hat{t}_\kappa$ is the time instant in the κ-th frame. The velocity of the last frame is assumed to be equal to that in the second last frame. The velocity calculated by Eqn. (A.1) has first-order accuracy. The 2nd FDM calculates velocities in a given frame by the central difference method:

$$\tilde{u}(\hat{t}_\kappa) = \frac{\hat{x}_{\kappa+1} - \hat{x}_{\kappa-1}}{2\Delta t}. \tag{A.2}$$

The velocities in the first and last frames are calculated by Eqn. (A.1). This central difference method has second-order accuracy in velocity evaluation (except for the first and last frames). Velocities in the y and z directions are the same as that in the x direction. Now the particle trajectories and velocities calculated by finite difference methods can be used for benchmarking.

A.2 Least squares polynomial fitting

In the least squares polynomial fittings, polynomials are used as model functions to approximate particle trajectories and velocities. For example, in the x direction, the trajectory polynomial model function $\tilde{x}(t)$ is given by:

$$\tilde{x}(t) = \sum_{j=0}^{m} p_{m,j} t^j \tag{A.3}$$

where $p_{m,j}$ is the polynomial coefficient, m is the order of polynomials, and t is the time. The trajectory and velocity model functions of the 2nd LS-POLY, for instance, are

$$\begin{aligned} \tilde{x}(t) &= p_{2,2}t^2 + p_{2,1}t + p_{2,0} \\ \tilde{u}(t) &= \dot{\tilde{x}}(t) = 2p_{2,2}t + p_{2,1} \end{aligned}, \tag{A.4}$$

respectively. The trajectory and velocity approximation functions of the other least squares polynomial fitting are similar to those in the x direction of the 2nd LS-POLY.

The key to least squares polynomial fitting is to find the polynomial coefficient $p_{m,j}$. To calculate the polynomial coefficient $p_{m,j}$, for example in the x direction, the residual $\mathcal{R}$ between particle location measurements $\hat{x}_\kappa$ and the polynomial model function $\tilde{x}(\hat{t}_\kappa)$ is minimized:

$$\min\ \mathcal{R} = \sum_{\kappa=1}^{N_{\mathrm{trj}}} \left\| \tilde{x}(\hat{t}_\kappa) - \hat{x}_\kappa \right\|^2, \tag{A.5}$$

where $\hat{t}_\kappa$ is the time instant in the κ-th frame. To find the extrema of $\mathcal{R}$, the gradient of the residual $\mathcal{R}$ regarding to $p_{m,j}$ is set to zero (i.e., $\partial\mathcal{R}/\partial p_{m,j} = 0$) [42, 52, 99]. The polynomial coefficient $p_{m,j}$ can be organized as a vector $\boldsymbol{p} = (p_{m,0}, p_{m,1}, \ldots, p_{m,m})^{\mathrm{T}}$ with dimensions of $(m+1) \times 1$ and the base t^j is arranged as a polynomial basis matrix

$$\mathbf{P} = \begin{bmatrix} 1 & t_1 & t_1^2 & \ldots & t_1^m \\ 1 & t_2 & t_2^2 & \ldots & t_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_{\mathrm{trj}} & t_{\mathrm{trj}}^2 & \ldots & t_{\mathrm{trj}}^m \end{bmatrix}, \tag{A.6}$$

with dimensions of $N_{\mathrm{trj}} \times (m+1)$. The vector of polynomial coefficient $\boldsymbol{p}$ is calculated by:

$$\boldsymbol{p} = \mathbf{P}^+ \hat{\boldsymbol{x}}, \tag{A.7}$$

where $\mathbf{P}^+ = (\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}$ is a generalized inverse of $\mathbf{P}$, and $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_{N_{\mathrm{trj}}})^{\mathrm{T}}$. The polynomial coefficients in the y and z directions are the same as that in the x direction. Once all polynomial coefficients are retrieved, the polynomial model functions are constructed and can be used to approximate particle trajectories and velocities.