

# Lagrangian Strain- and Rotation-Rate Tensor Evaluation Based on Multi-pulse Particle Tracking Velocimetry (MPTV) and Radial Basis Functions (RBFs)

Lanyu Li<sup>1</sup>, Prabu Sellappan<sup>2</sup>, Peter Schmid<sup>3</sup>, Louis N. Cattafesta<sup>2\*</sup>,  
Jean-Pierre Hickey<sup>1</sup> and Zhao Pan<sup>1†</sup>

<sup>1</sup> University of Waterloo, Department of Mechanical and Mechatronics Engineering, ON N2L 3G1, Canada

<sup>2</sup> Florida Center for Advanced Aero-Propulsion (FCAAP), FAMU-FSU College of Engineering, Tallahassee, Florida, 32310, US

<sup>3</sup> Department of Mathematics, Imperial College London, London, UK

\* lcattafesta@fsu.edu † zhao.pan@uwaterloo.ca

## Abstract

Physical conservation laws are inherently Lagrangian. However, analyses in fluid mechanics using the Lagrangian framework are often forgone in favor of those using the Eulerian framework. This is perhaps due to a lack of experimental techniques with high temporal and spatial resolution that track the movement of fluid tracers in a flow domain. The development of time-resolved Particle Tracking Velocimetry/Accelerometry (TR-PTV/A) that measures flows with high seeding density has made the use of the Lagrangian framework more accessible. A challenge facing PTV/A is the need for robust mesh-free numerical schemes that handle random particle locations. Such a scheme can be created with high-order accuracy using Radial Basis Functions (RBFs). RBFs allow direct evaluation of derivatives of vector and scalar fields at random locations with infinite-order smoothness. The current work uses RBF-based differential schemes to develop a post-processing tool for PTV/A data, which can accurately evaluate spatial derivatives directly from Lagrangian particle tracks. This RBF-based strain/rotation-rate tensor evaluation tool is validated with two and three-dimensional flows from analytical solutions and is then tested with experimental data measured by a multi-pulse PTV/A system.

## 1 Introduction

Classic textbooks in fluid mechanics often start with the concept of a fluid parcel and Eulerian versus Lagrangian descriptions of fluid motion (Panton, 2006). In addition to the inherent conservation laws developed in a Lagrangian framework, particle tracking is perhaps the most intuitive means to probe a fluid flow. A common example is examining the motion of tea leaves in a cup. However, examples of Lagrangian methods in the experimental fluid dynamics literature pale in comparison to Eulerian approaches. This is perhaps due to a lack of experimental techniques with sufficient temporal and spatial resolution that can track the three-dimensional motion of many fluid parcels in a flow domain. This deficiency has persisted until the recent development of time-resolved Particle Tracking Velocimetry/Accelerometry (TR-PTV/A) that can measure Lagrangian tracer particle tracks with high seeding density (Schanz et al., 2016).

In addition to the above experimental challenge, robust mesh-free numerical schemes that handle randomly scattered vector field are lacking in the experimental fluid mechanics community. One of the basic tasks for data analysis is evaluating the three-dimensional (3D) velocity gradient tensors, namely the strain-rate and rotation-rate tensors. Strain/rotation-rate tensors are essential to understand the kinematics of complex flows (e.g., coherent structures and invariants). For example, Kitzhofer et al. (2010) developed an algorithm based on least squares matching to estimate the strain/rotation-rate tensor using Tomo-PIV. Although this work successfully extended kinematic analysis from two dimensions (2D) to 3D, it used an Eulerian framework because the velocity data on a structured mesh are from PIV. Methods such as VIC+

can handle scattered velocity data on ‘unstructured mesh’ from PTV (Jeon et al., 2018). However, interpolating scattered data to a structured mesh before any other analysis is a common practice. This step can introduce unnecessary numerical error; more importantly, such a procedure may ‘degrade’ the often sparse but accurate PTV-based Lagrangian data to PIV-like Eulerian data on a regular mesh. Direct analysis of velocity fields on scattered data relies on a mesh-free numerical scheme. Recent progress by Rosi et al. (2015) used Voronoi tessellation to evaluate spatial gradients for Lagrangian coherent structure identification. This work introduced a direct analysis tool for Lagrangian data. But, it was limited to 2D, and extending to 3D would be much more difficult (in terms of implementation and computation) due to the nature of the Voronoi diagram. As another example, Takehara and Etoh (2009) reconstructed the 2D vorticity field directly from PTV data using a Moving Least Square (MLS) method. The authors applied this algorithm to visualize fluid motion and vortices in wind-driven waves (Takano et al., 2012). However, the algorithm was not extended to 3D. Consequently, some parameters in the algorithm were limited to 2D. For example, particle distribution was quantified by particle per pixel (PPP). This is a standard measure of image resolution in planar PIV or PTV, and PPP is also a global indicator of average particle seeding density in 2D physical space with proper calibration. However, the particle data layout is important for a mesh-free method, and the local spatial particle density in 3D must be considered when applying such methods. As such, a robust, accurate, and convenient mesh-free method is much needed to directly compute spatial velocity gradients in 3D.

In the current work, we establish a three-step data analysis method, which can accurately evaluate spatial derivatives (and then strain/rotation-rate tensors) directly from Lagrangian velocimetry data. The three-step method consists of i) a *data processing* step that calculates spatial derivatives using a stable mesh-free method based on Radial Basis Functions (RBFs), ii) a *pre-processing* step that improves the data quality by leveraging the data layout and smoothing the potential noise from the experiments, and iii) a *post-processing* that removes possible outliers. This data analysis method is verified using simulated PTV data in 2D and 3D and then tested on experimental data from an MPTV system in 3D.

## 2 Data processing: RBF-QR and PUM for stable gradient evaluation

### 2.1 RBF-QR as stable mesh-free method

Radial Basis Functions (RBFs) are a class of mesh-free methods that are increasingly applied in many fields, such as computational physics and machine learning. RBFs are often used as an interpolation and approximation tool that can handle high-dimensional scattered data. A typical RBF-based interpolation problem can be stated as follows. Given the known scalar data  $f_i^c \in \mathbb{R}$  at  $N$  scattered node centers  $\mathbf{x}_i^c \in \mathbb{R}^d$ ,  $i = 1, 2, 3, \dots, N$ , ( $d$  is the number of dimension of  $\mathbf{x}_i^c$ ), we look for a reconstructing function  $s(\mathbf{x})$  that interpolates these known data (also called centers). This function is a linear combination of  $N$  RBFs in the form of

$$s(\mathbf{x}) = \sum_{i=1}^N \lambda_i \phi(\varepsilon, r_i) = \sum_{i=1}^N \lambda_i \phi(\varepsilon, \|\mathbf{x} - \mathbf{x}_i^c\|), \quad (1)$$

where  $\phi(\varepsilon, r_i)$  is a RBF, and  $\varepsilon$  is a parameter called the ‘shape factor’ that controls the shape (e.g., spiky or flat) of the RBF, and  $r_i = \|\mathbf{x} - \mathbf{x}_i^c\|$  is the Euclidean distance from a location in the domain to a center. One of the common choices for  $\phi(\varepsilon, r)$  is the Gaussian RBF (i.e.,  $\phi(r) = \exp(-\varepsilon^2 r^2)$ ). The coefficient  $\lambda_i$  can be determined by enforcing  $s(\mathbf{x}_i^c) = f_i^c$ . The corresponding linear system is  $\mathbf{f} = \mathbf{H}\boldsymbol{\lambda}$ , where  $\mathbf{H}$  has entries  $H_{ij} = \phi(\varepsilon, \|\mathbf{x}_i^c - \mathbf{x}_j^c\|)$ ,  $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_N)^T$ , and  $\mathbf{f} = (f_1^c, f_2^c, \dots, f_N^c)^T$ . By inverting  $\mathbf{H}$ ,  $\boldsymbol{\lambda} = \mathbf{H}^{-1}\mathbf{f}$  can be solved for eq. (1). Derivatives of  $s(\mathbf{x})$  can be easily calculated by applying a corresponding differential operator to  $\phi(\varepsilon, \|\mathbf{x} - \mathbf{x}_i^c\|)$ . For example,  $s_y(\mathbf{x}) = \sum_{i=1}^N \lambda_i \phi_y(\varepsilon, r_i)$  is the spatial derivative of  $s(\mathbf{x})$  with respect to coordinate  $y$ . This straightforward approach is commonly called ‘RBF-Direct’.

For data interpolation using a smooth function, small  $\varepsilon$  corresponding to flat RBFs leads to accurate interpolation. However, if the RBF-Direct approach is used, a flat RBF also results in an ill-conditioned  $\mathbf{H}$  and, thus, numerical instability. To overcome this problem, there are currently three primary versions of stable RBF algorithms (Wright and Fornberg, 2016): the Contour-Padé (RBF-CP), RBF-GA, and RBF-QR methods. RBF-CP was the first algorithm that removed the restriction that the shape factor must be a real number (Fornberg and Wright, 2004). In RBF-CP, a basis function incorporates a rational function of  $\varepsilon$ , and stable computing can be achieved with flat RBFs. A recent extension to RBF-CP is the RBF-RA (Wright and Fornberg, 2016), which is more accurate and efficient compared to RBF-CP. RBF-GA is specifically based on the Gaussian kernel (Fornberg et al., 2013). It transforms an ill-conditioned basis

to well-conditioned ones which include the incomplete gamma function instead of a truncated numerical approximation. RBF-QR was introduced by [Fornberg et al. \(2011\)](#) and can handle a relatively large number of points than RBF-GA and RBF-CP/RA. These methods are all stable at flat limit of RBFs (e.g.,  $\epsilon \rightarrow 0$  for Gaussian RBFs) and can calculate derivatives on scattered data ([Larsson et al., 2013](#)) in high dimensions with a finite number of data points.

RBF-CP and RBF-RA are limited to about a hundred and a few hundred points in 2D and 3D, respectively ([Fornberg and Wright, 2004](#); [Fornberg et al., 2013](#); [Wright and Fornberg, 2016](#)). RBF-GA can increase these to a few hundred in 2D and greater than 500 in 3D ([Fornberg et al., 2013](#)), while RBF-QR can use hundreds of points in 2D and thousands of points in 3D with double precision ([Fornberg et al., 2011](#)). In the current work, we use RBF-QR as the core numerical method to compute gradients on scattered data. Implementation details of RBF-QR in 2D and 3D can be found in [Fornberg et al. \(2011\)](#); [Larsson et al. \(2013\)](#).

## 2.2 Localization by Partition of Unity Method (PUM)

As mentioned above, flat RBFs are often used to achieve high accuracy reconstruction at the expense of ill-conditioning. However, this is not only caused by using flat RBFs (or the choice of value of shape factor). For a fixed value of shape factor, a large number of data points can also lead to numerical instability. The same issue persists for stable RBF-based methods. In addition, computing a large number of data points in a domain at once can be prohibitively expensive. There are two popular strategies to overcome the challenge associated with large data: i) RBF-generated finite differences ([Fornberg and Wright, 2004](#); [Wright and Fornberg, 2016](#)), and ii) the Partition of Unity Method (PUM) ([Larsson et al., 2013](#)) as a localization approach.

In the current work, we use PUM to localize the RBF-QR to handle large data. First, a set of overlapped sub-domains ( $\Omega_j, j = 1, \dots, P$ , called PUM patches) are constructed to cover the whole domain  $\Omega$ . [Figure 1](#) shows examples where 9 circular and 8 spherical PUM patches cover the domain<sup>1</sup> in 2D and 3D, respectively. Each patch is then assigned a weight function  $W_j(\mathbf{x})$  so that i) for a certain patch, the weight function outside that patch is zero (i.e.,  $W_j(\mathbf{x}) = 0$ , for  $\mathbf{x} \notin \Omega_j$ ), and ii) the summation of all weight functions from all the patches is unity at any location in the domain (i.e.,  $\sum_{j=1}^P W_j(\mathbf{x}) = 1$ ). The weights can be constructed by applying Shepard’s method ([Shepard, 1968](#)):  $W_j(\mathbf{x}) = \psi_j(\mathbf{x}) / \sum_{k=1}^P \psi_k(\mathbf{x}), j = 1, 2, 3 \dots P$ , where  $\psi(\mathbf{x})$  is a compactly supported generating function such as the Wendland  $C^2$  function ([Wendland, 1995](#)). A global function  $s(\mathbf{x})$  on the domain  $\Omega$  can then be assembled by a weighted summation of local functions on  $\Omega_j$ :  $s(\mathbf{x}) = \sum_{j=1}^P W_j(\mathbf{x}) s_j(\mathbf{x})$ . For example, if  $s_j(\mathbf{x})$  is a spatial gradient of a velocity component of a flow, when  $s_j(\mathbf{x})$  is calculated using RBF-QR in every patch, then a localized stable RBF that can handle a large number of data points is achieved. In addition, combining PUM and RBF-QR also reduces computational cost and memory requirements. Example implementations of RBF and PUM can be found in, for example, [Larsson et al. \(2017\)](#).

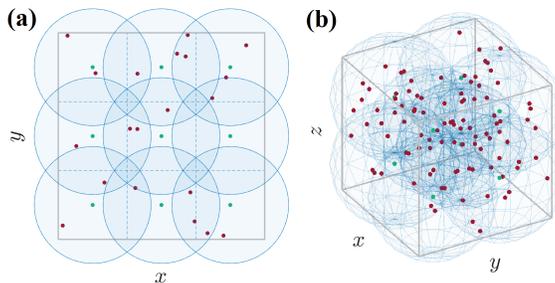


Figure 1: Example PUM patch layout in 2D, (a) and 3D, (b). The grey lines indicate the edges of domain  $\Omega$  (square for 2D and cube for 3D). Circles and spheres in transparent light blue are PUM patches for 2D and 3D, respectively, whose centers are marked by green dots. Red dots indicate the locations of scattered data points in the domain.

<sup>1</sup>The domain  $\Omega$  does not have to be regular-shaped boxes, and the PUM patches do not have to be regular-shaped circles or spheres.

### 3 Pre-Processing: data layout enhancing and smoothing

Velocity from a real experiment is imperfect. In addition to contamination due to measurement noise, PTV data may not be well distributed. The location of single tracer particle cannot be controlled and local seeding density may be too high or too low, which may lend difficulties to RBF-based gradient computation. Thus, before data processing, pre-processing may be necessary to improve the data quality. In the current work, we suggest a two-stage pre-processing including i) data layout enhancement and ii) smoothing before processing experimental data.

A mesh-free method (including RBFs) in general performs better on quasi-uniformly distributed data (Fasshauer, 2007). Although RBFs can in principle evaluate gradients on arbitrarily scattered data, randomly distributed data can lead to results with low accuracy. This challenge arises from the ‘conflicting’ nature of PTV experiments and mesh-free numerical methods. We here use a simple example to demonstrate this particular issue. We use RBF-QR with PUM to directly calculate velocity gradient ( $\partial u/\partial x$ ) in a 2D domain based on simulated PTV data (see §5 and Fig. 3 for more details about the simulated data and the flow field). Here,  $1 \times 10^4$  virtual tracing particles are injected in the domain so that they are either randomly distributed (coordinates of particle locations are generated using random numbers with uniform distribution) or have a much more uniformly distributed particle layout (particle locations are generated using a Halton sequence). We want to emphasize that the particle layout generated from random numbers with uniform distribution is not quasi-uniform. Uniform distribution generated random particles, similar to the particle distribution in a well-seeded PTV experiment, have high randomness in particle location leading to excessive local sparse or dense particle layout. The relative errors of the gradient calculation on the velocity data contaminated by small noise (zero mean Gaussian noise with a standard deviation of 1% of maximum corresponding velocity component) on each data layout are examined. Figure 2 shows the relative error ( $\mathcal{E}$ , defined by Eq. (2)) in the calculated velocity gradient against the distance of every two neighboring particles ( $d_n$ ).

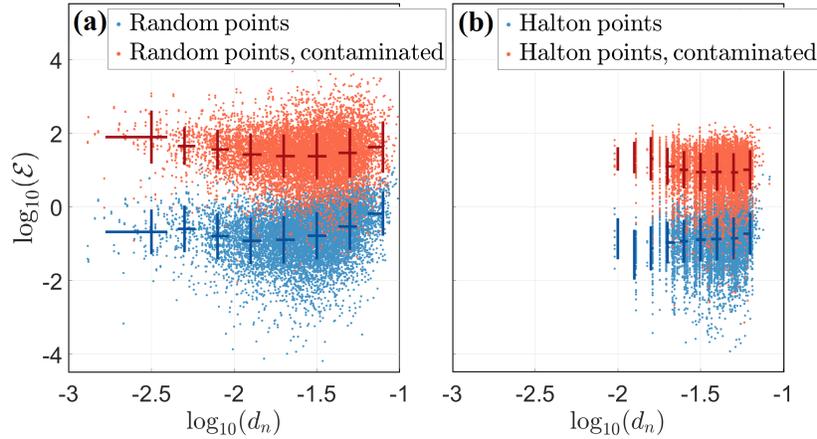


Figure 2: Relative error ( $\mathcal{E}$ ) of the computed gradient ( $\partial u/\partial x$ ) on each data point vs. distance of two neighboring data points ( $d_n$ ), on random (a) and Halton points (b). The light red and blue point clouds indicate the relative error in the computed gradient at each simulated PTV particles with and without artificial noise, respectively. The overlaid crosses with darker colors summarize the trend of point clouds in groups. The left and right end of a horizontal bar represents  $\pm 1$  standard deviation band of  $\log_{10}(d_n)$  for a group of data, and the vertical bar represents the  $\pm 1$  standard deviation band of  $\log_{10}(\mathcal{E})$  for the same group of data. The intersection of vertical and horizontal bars indicates the mean of the error for the group of the data.

The results show that the gradient calculation based on randomly distributed particles (Figure 2(a)), which is close to the particle distribution of real PTV data, in general has higher error than that based on a Halton particle distribution (Figure 2(b)). As shown in Fig. 2(a), when  $d_n$  is too large, too sparse data cannot resolve the fine flow structures and lead to high error, which is intuitive. When  $d_n$  is too small (implying that high local particle density), elevated error also results. This is due to low uniformity of the particle layout, which is unfriendly to the RBFs. This observation may not be surprising either. It resembles a low quality mesh, in the context of finite element analysis, caused by neighboring nodal points that are too close and lead to high-aspect ratio cells. To enhance the data layout, we interrogate neighboring particle pairs in each PUM patch, and iteratively mask out one of the two neighboring points if the distance between them is much closer than the rest of neighbors. In our tests, a few iterations are sufficient to significantly improve

the performance of RBF-based calculation.

It is worth noting that when the velocity data is contaminated by noise, the observation about the data layout is even more obvious (see the red data points and error bars in Figure 2): random noise on close data points further exacerbates the effect of non-uniformity on numerical differentiation. However, being noisy is unfortunately the case of real experimental data. In addition, numerical differentiation of noisy data may significantly amplify the noise especially when the ‘grid spacing’ is small and/or the order of the differentiation scheme is high. For example, Fig. 2 shows that small artificial noise in the velocity field can significantly corrupt the gradient calculation. Thus, we propose applying a smoother to the velocity data in each PUM patch after data layout enhancement. In the current work, a two layer neural network (‘NEWRBE’, a MATLAB built-in function) is used to smooth out potential noise. An adaptive spread parameter ( $S_p$ ) is used in the ‘NEWRBE’ function is used:  $S_p = Kd_n^{min} / \sqrt[d]{N}$ , where  $d_n^{min}$  is the closest distance between two neighboring points within a PUM patch after data layout enhancing,  $N$  is the number of points in that patch,  $d$  is dimension of domain. We tested on isotropic turbulent flows (see §5) and found that  $K = 1000$  and  $250$  are robust choices for 2D and 3D computation, respectively.

## 4 Post-Processing: outlier detection and removal

It is possible to have few extreme values (likely suffering large error) in the calculated gradient fields, even if pre-processing is applied and the overall accuracy of the calculation is significantly improved. Thus, an outlier detector and remover may be necessary for post-processing. In the current work, we adopt an adaptive outlier detector developed by Cheng and Wang (2019), which labels an extreme value in a neighborhood as an outlier and removes it for the final result.

## 5 Validation on Simulated PTV data

We next validate the performance of our data analysis method. A direct numerical simulation (DNS) of forced isotropic turbulence is used as ground truth to validate the algorithm. The simulations were conducted using the Pencil code (Brandenburg et al., 2021). The solver uses a high-order, explicit finite-difference scheme combined with a hyperviscosity filter for stability. A solenoidal forcing is applied to randomly selected low-wave numbers in the velocity spectrum to sustain the turbulence at a Reynolds number (based on the Taylor length scale) of  $Re_\lambda \approx 80$ . An overly conservative grid resolution of  $512^3$  was selected to adequately resolve all relevant dynamic scales in the flow. Pseudo-particles are injected in the domain at random locations. The velocity and velocity gradients at the particle locations are obtained using linear interpolation from the DNS data. Small artificial noise is introduced to each velocity component at a particle to simulate the noise in experiments. The noise is zero mean Gaussian noise with a standard deviation of 1% of the maximum corresponding velocity component, and we use this noise throughout the current paper. We use this contaminated synthetic PTV data as the input to test the performance of the data analysis tool.

A baseline algorithm based on RBF-Direct with adaptive shape factor from literature Jeon et al. (2018) is also used for compression (i.e.,  $\phi(r) = \exp(-\frac{r^2}{2.4h^2})$ , where we define  $h$  as the average distance between the particles). The relative error in the quantity of interest is used to assess the accuracy of the computation. For example, when calculating the vorticity of a 2D flow field, the relative error at each particle is defined as

$$\mathcal{E} = \frac{|\omega - \tilde{\omega}|}{\|\omega\|_{L^\infty(\Omega)}} \times 100\%, \quad (2)$$

where  $\omega$  is the true value of the vorticity, and  $\tilde{\omega}$  is the computed value that may have error. Note, we found that the distribution of error is close to logarithm-normal.

Figure 3 shows a 2D validation calculating vorticity from simulated 2D PTV data, where  $\mathbf{u} = (u, v)$  is the velocity field measured at  $1 \times 10^4$  randomly distributed particles on a  $(x, y) \in [-\pi, \pi] \times [-\pi, \pi]$  domain. The velocity (Fig. 3(a)) and vorticity (Fig. 3(b)) fields are based on a slice of the aforementioned 3D DNS data. Calculated vorticity field based on simulated PTV data with (Fig. 3(c3 and c4)) and without (Fig. 3(c1 and c2)) noise are shown. The corresponding relative error at each particle is presented in Fig. 3(d1 – d4) accordingly, and summarized as histograms in Fig. 3(e1 – e4). When no artificial noise is introduced to the velocity field, both the baseline algorithm and our method can evaluate the vorticity accurately. The median

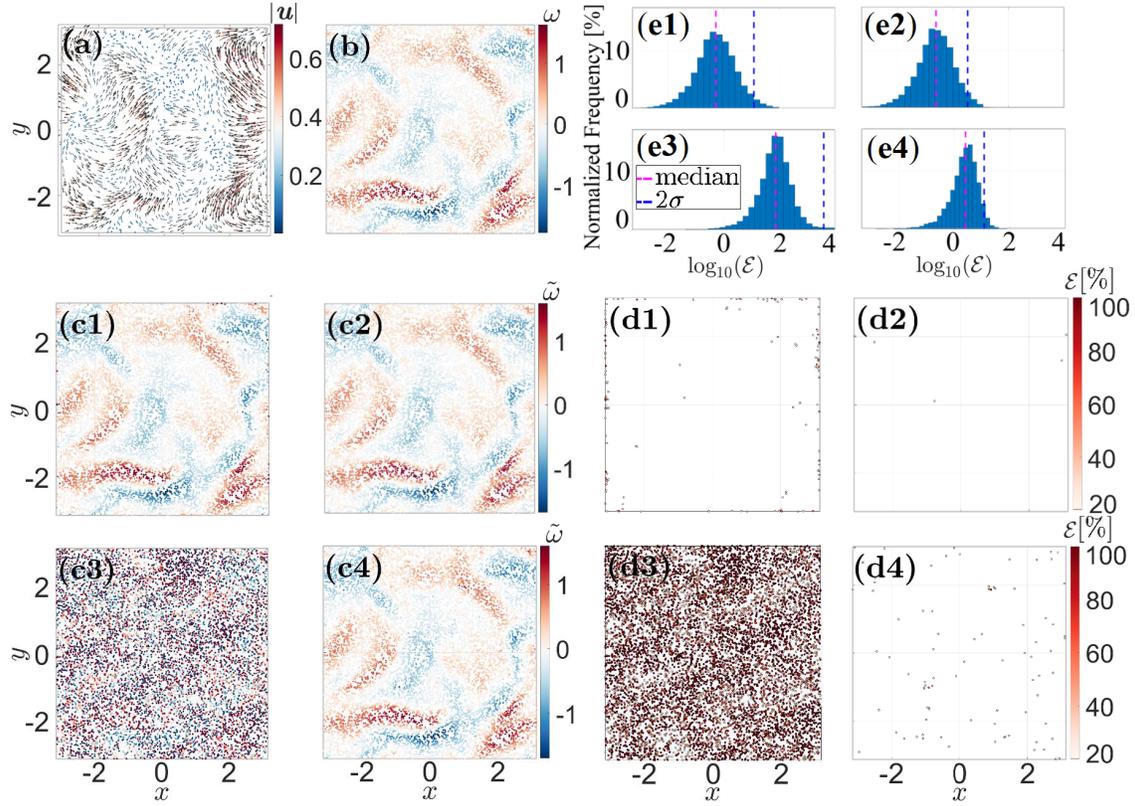


Figure 3: 2D validation on simulated PTV data. (a) Velocity field of the simulated PTV data. Quivers indicate the direction of the flow field and particles are colored by the amplitude of the velocity. Particles are down-sampled to 25% of the actual seeding density for clear illustration. (b) Ground truth of vorticity, colored at each particle. Computed vorticity based on the simulated PTV data without noise contamination using the baseline algorithm (c1) and our three-step method (c2). Computed vorticity based on the simulated PTV data with artificial noise introduced, using the baseline algorithm (c3) and our three-step method (c4). (d1–d4) Relative error in the computed vorticity, corresponding to c1–c4, respectively. Only particles with error in vorticity higher than 20% are shown. (e1–e4) Histogram of relative error in the computed vorticity, corresponding to d1–d4. The vertical magenta and blue dashed lines indicate the median and +2 standard deviation above the mean, respectively.

values of the relative error are 0.52% and 0.24%, as shown in Fig. 3(e1 and e2), respectively. However, when the velocity is slightly contaminated by random noise, our method is significantly more robust.

We next validate the algorithm in 3D with the simulated data on a cropped domain,  $x \times y \times z \in [-\pi, 0] \times [-\pi, 0] \times [-\pi, -0.75\pi]$ , of the DNS result. Noise contaminated velocity field at  $2 \times 10^4$  randomly distributed particles in the domain is used to reconstruct strain-rate tensor ( $\tilde{S}_{ij}$ , Fig. 4(b)) and rotation-rate tensor ( $\tilde{R}_{ij}$ , Fig. 5(b)). The velocity gradient tensor in 3D is  $u_{i,j} = \partial u_i / \partial x_j$ , where  $u_i$ ,  $i = 1, 2, 3$  are the velocity components, and  $x_j$ ,  $j = 1, 2, 3$  are the Cartesian coordinates, which can be directly evaluated at each particles using our method.  $u_{i,j}$  can be decomposed into symmetric and anti-symmetric parts (Blazek, 2015) as follows:  $u_{i,j} = \frac{1}{2}(u_{i,j} + u_{j,i}) + \frac{1}{2}(u_{i,j} - u_{j,i})$ . The first part is strain-rate tensor  $S_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$  and the second part is rotation-rate tensor  $R_{ij} = \frac{1}{2}(u_{i,j} - u_{j,i})$ .

Compared to the ground truth (Fig. 4(a) for  $S_{ij}$ ) against the computed values from noisy PTV data (Fig. 4(b) for  $\tilde{S}_{ij}$ ). The median values of relative error in  $\tilde{S}_{ij}$  are less than 2% and two standard deviations above average of are less than about 20%. Therefore, our method can compute the gradients and the tensor with robustness and sufficient accuracy. The flow structures are well visualized with the iso-surfaces. Figure 5 shows the ground truth ( $R_{ij}$ , Fig. 5(a)) of the rotation-rate tensor and the computed results ( $\tilde{R}_{ij}$ , Fig. 5(b)). Noting that  $R_{ij} = -R_{ji}$ , the iso-surfaces in Fig. 5(a) and Fig. 5(b) have the almost the same shape

for the corresponding components, but with reversed color. The relative error in  $\tilde{R}_{ij}$ , as examples, are summarized in Fig. 5(c1–c3). The median of the relative error is around 2% and two standard deviation above average of the error is less than 15%.

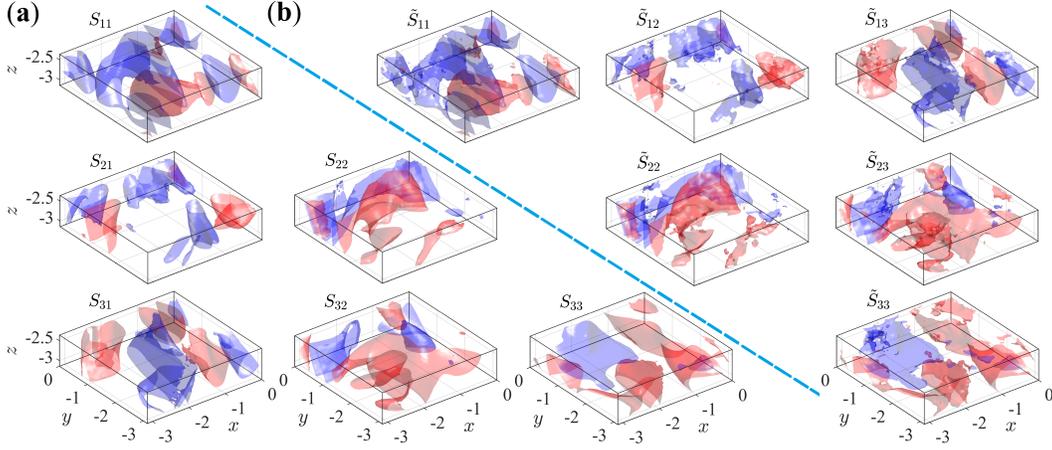


Figure 4: Visualization of the strain-rate tensor ( $S_{ij}$ ) of a 3D turbulent flow. (a) Lower six components of the strain-rate tensor ( $S_{ij}$ ,  $i \geq j$ ), ground truth from DNS. (b) Upper six components of strain-rate tensor,  $\tilde{S}_{ij}$ ,  $i \leq j$ , computed from simulated PTV data with artificial noise introduced in the velocity field. (a) and (b) are separated by the blue dashed line. The surface iso-values are  $+0.3$  for red and  $-0.3$  for blue.

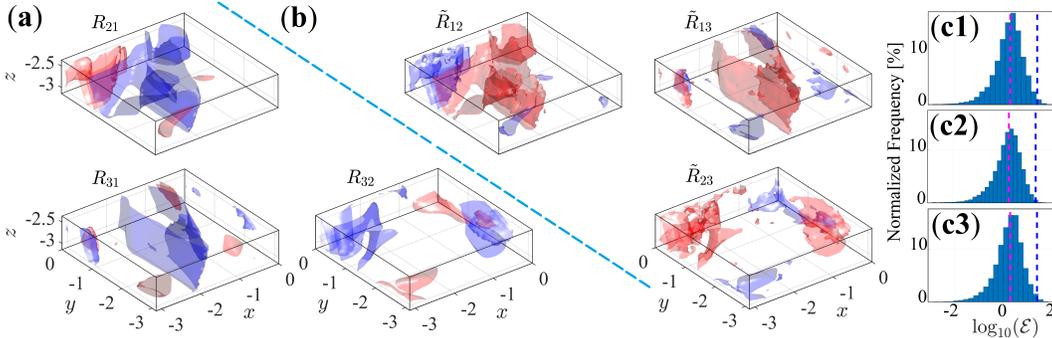


Figure 5: Visualization of the rotation-rate tensor ( $R_{ij}$ ) of a 3D turbulent flow. (a) Lower three components of rotation-rate tensor ( $R_{ij}$ ,  $i > j$ ), ground truth from DNS. (b) Upper three components of rotation-rate tensor,  $\tilde{R}_{ij}$ ,  $i < j$ , computed from simulated PTV data with artificial noise introduced in the velocity field. The surface iso-values are  $+0.3$  for red and  $-0.3$  for blue. (c1–c3) Histogram of the relative error in the computed  $\tilde{R}_{ij}$  in (b),  $\tilde{R}_{23}$ ,  $\tilde{R}_{13}$ ,  $\tilde{R}_{12}$ , respectively.

## 6 Application to experimental data

Finally, we apply our method to real experimental data. Velocity field of a free jet issuing from a circular nozzle (diameter  $D = 2.54$  cm) at Mach 0.3 is obtained using Multi-Pulse Shake-The-Box (Fig. 6). Details of the setup and data can be found in Sellappan et al. (2020). Our three-step method is used to process  $\sim 2.6 \times 10^4$  particles in a volumetric domain (average seeding density is 0.25 particles/mm<sup>3</sup>) and evaluate strain/rotation-rate tensors (Fig. 7 and Fig. 8, respectively). In  $S_{12}$ ,  $S_{21}$  and  $R_{12}$ ,  $R_{21}$ , two stripes with different colors (opposite signs of iso-values for iso-surfaces) are shown as they are dominated by  $\partial u/\partial y$ . The similar large scale strip are also shown in  $S_{13}$ ,  $S_{31}$  and  $R_{13}$ ,  $R_{31}$  for the dominant  $\partial u/\partial z$  contribution. In  $S_{11} = \partial u/\partial x$ , we observe blue and red bands appearing alternatively in the stream-wise direction, which is from roll-up

due to strong shear and can be better visualized by a vortex identification method. Figure 9 shows vortical structures of the jet flow identified by Q-criterion, where  $Q = \frac{1}{2} (|R^2| - |S^2|) > 0$ . Toroidal vortex structures of the high speed jet present near the nozzle due to Kelvin-Helmholtz instabilities, and break up downstream.

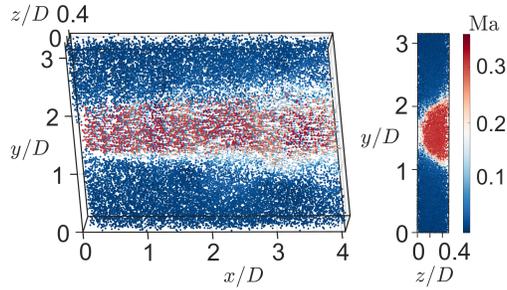


Figure 6: Nondimensionalized velocity field of a Ma 0.3 free jet shooting from left to right. The side view of the velocity field is projected to the left.

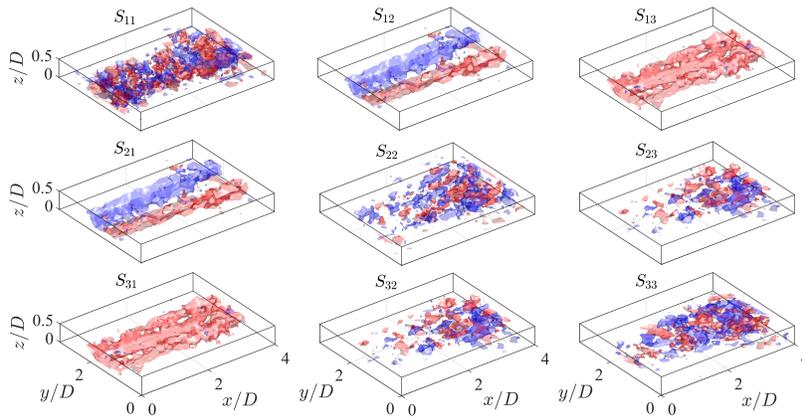


Figure 7: Nondimensionalized strain-rate tensor of the Ma 0.3 free jet. Surface iso-values for  $S_{12}$ ,  $S_{21}$ ,  $S_{13}$ ,  $S_{31}$  are  $\pm 0.2$ , and other tensor components adopt iso-values of  $\pm 0.1$ . Red surfaces for positive iso-values and blue for negative.

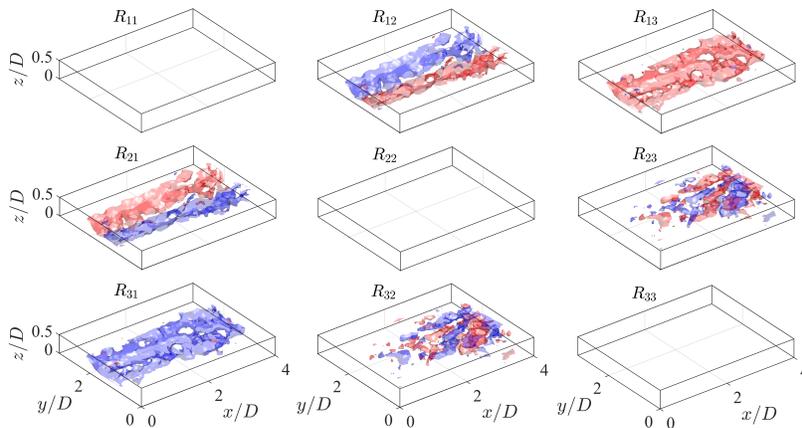


Figure 8: Nondimensionalized rotation-rate tensor of the Ma 0.3 free jet. Surface iso-values for  $R_{23}$ ,  $R_{32}$  are  $\pm 0.1$ , and other tensor components adopt iso-values of  $\pm 0.2$ . Red surfaces for positive iso-values and blue for negative.

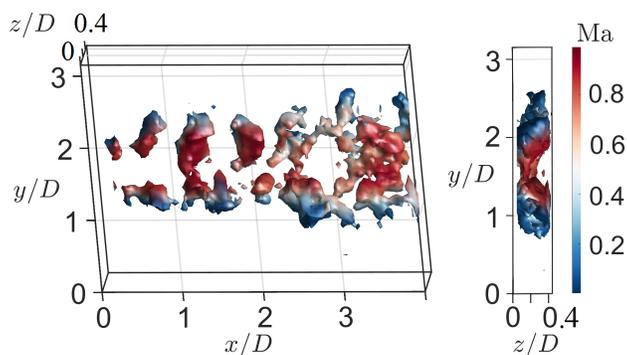


Figure 9: Iso-surface of  $Q$ -criterion ( $Q = 0.025$ ), colored by  $Ma$ , of the  $Ma$  0.3 jet flow.

## 7 Conclusion and discussions

We propose a three-step method to calculate spatial velocity gradients and strain/rotation-rate tensors for volumetric PTV/A data. RBF-QR with PUM, serving as the core of the processing step, ensures that the method can handle large scattered data in 3D with numerical stability. Pre-processing improves data quality by enhancing data layout and reducing potential noise from experimental data. The post-processing detects and removes potential outliers for more robust final results. Validation on simulated data in 2D and 3D show that this method is robust and accurate enough to be applied to real volumetric PTV/A data, where particle layout may not be ideal and the velocity is uncertain. Our method directly reads, operates, and stores Lagrangian data for each particle, and only some visualization may require interpolation on Eulerian mesh.

Lastly, we want to point out a special issue that the data layout (local particle distribution) can significantly affect a gradient calculation, especially when noise is present in the PTV data. To be more specific, tracer particles with quasi-uniform layout is preferred by our method, and many other mesh-free schemes. However, a quasi-uniform distribution of particles may never be achieved in a real PTV measurement for many reasons. Examples include agglomeration of tracer particles in high shear flows, low seeding densities near the wall, and non-uniform illumination and scattering, etc. Due to the non-uniformity of the particle layout, raw PTV data has low efficiency in terms of resolving flow structures. Local high-density seeding may appear at low gradient regions, where high spatial resolution sampling is not needed. Moreover, excessively high-density seeding in a local region may corrupt a mesh-free numerical scheme, despite the fact that global high seeding density is much preferred to resolve small scale structures in the flow. Even for a well-seeded flow, randomly distributed particles do not yield the most friendly data layout for a mesh-free scheme and data layout enhancement is perhaps needed.

## Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the U.S. Office of Naval Research Grant N00014-21-1-2454 (monitored by Dr. David Gonzalez) and insightful discussions with Prof. Larsson at Uppsala University and Prof. Wright from Boise State University.

## References

- Blazek J (2015) *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann
- Brandenburg A, Johansen A, Bourdin PA, Dobler W, Lyra W, Rheinhardt M, Bingert S, Haugen NEL, Mee A, Gent F, Babkowskaia N, Yang CC, Heinemann T, Dintrans B, Mitra D, Candelaresi S, Warnecke J, Käpylä PJ, Schreiber A, Chatterjee P, Käpylä MJ, Li XY, Krüger J, Aarnes JR, Sarson GR, Oishi JS, Schober J, Plasson R, Sandin C, Karchniwy E, Rodrigues LFS, Hubbard A, Guerrero G, Snodin A, Losada IR, Pekkilä J, and Qian C (2021) The pencil code, a modular mpi code for partial differential equations and particles: multipurpose and multiuser-maintained. *Journal of Open Source Software* 6:2807

- Cheng Z and Wang J (2019) Quantification of the strain field of sands based on x-ray micro-tomography: A comparison between a grid-based method and a mesh-based method. *Powder Technology* 344:314–334
- Fasshauer GE (2007) *Meshfree approximation methods with MATLAB*. volume 6. World Scientific
- Fornberg B, Larsson E, and Flyer N (2011) Stable computations with gaussian radial basis functions. *SIAM Journal on Scientific Computing* 33:869–892
- Fornberg B, Lehto E, and Powell C (2013) Stable calculation of gaussian-based rbf-fd stencils. *Computers & Mathematics with Applications* 65:627–637
- Fornberg B and Wright G (2004) Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers & Mathematics with Applications* 48:853–867
- Jeon Y, Schneiders J, Müller M, Michaelis D, and Wieneke B (2018) 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
- Kitzhofer J, Westfeld P, Pust O, Nonn T, Maas H, and Brücker C (2010) 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*
- Larsson E, Lehto E, Heryudono A, and Fornberg B (2013) Stable computation of differentiation matrices and scattered node stencils based on gaussian radial basis functions. *SIAM Journal on Scientific Computing* 35:A2096–A2119
- Larsson E, Shcherbakov V, and Heryudono A (2017) A least squares radial basis function partition of unity method for solving pdes. *SIAM Journal on Scientific Computing* 39:A2538–A2563
- Panton RL (2006) *Incompressible flow*. John Wiley & Sons
- Rosi GA, Walker AM, and Rival DE (2015) Lagrangian coherent structure identification using a voronoi tessellation-based networking algorithm. *Experiments in Fluids* 56:1–14
- Schanz D, Gesemann S, and Schröder A (2016) Shake-the-box: Lagrangian particle tracking at high particle image densities. *Experiments in Fluids* 57:70
- Sellappan P, Alvi FS, and Cattafesta LN (2020) 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
- Shepard D (1968) A two-dimensional interpolation function for irregularly-spaced data. in *Proceedings of the 1968 23rd ACM national conference*. pages 517–524
- Takano Y, Etoh GT, and Tsujimoto G (2012) Measurements of vortex generation and motion at wind wave surfaces. *Coastal Engineering Proceedings* pages 28–28
- Takehara K and Etoh T (2009) Accurate estimation of vorticity from ptv velocity distribution. in *Proc. of 8th International Symposium on Particle Image Velocimetry-PIV09*. volume 185
- Wendland H (1995) Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics* 4:389–396
- Wright GB and Fornberg B (2016) Stable computations with flat radial basis functions using vector-valued rational approximations. *Journal of Computational Physics* 331