SoftwareX 12 (2020) 100604

Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

VortexFitting: A post-processing fluid mechanics tool for vortex identification

Guilherme Lindner [a,*], Yann Devaux [b], Sanja Miskovic [a]

[a] Norman B. Keevil Institute of Mining Engineering, University of British Columbia, Vancouver, Canada
[b] Institut P', CNRS - Université de Poitiers - ISAE-ENSMA - UPR 3346, SP2MI - Téléport 2, 11 Bvd Marie & Pierre Curie, F-86962 Futuroscope Chasseneuil, France

ARTICLE INFO

Article history:
Received 29 July 2020
Received in revised form 28 September 2020
Accepted 2 October 2020

Keywords:
Vortex
Detection methods
Fluid dynamics
Python

ABSTRACT

VortexFitting is a fluid mechanics post-processing tool developed in Python. It aims to detect the presence of vortices in a flow and evaluate their properties. Data obtained from both numerical simulations and experimental flow imaging techniques can be used as inputs. The software supports a number of input file formats such as NetCDF, HD5, TecPlot, and raw text files. The first stage of the vortex search procedure, which is identification of vortex candidates, is accomplished using a set of detection methods: swirling strength, Q criterion, and Δ criterion. The candidate vortices are then fitted to a Lamb–Oseen vortex model using a non-linear least-squares method, and the correlation between the model and the original velocity field is evaluated. If the correlation is deemed high enough, based on a user defined threshold, the vortex is accepted, and properties such as vortex radius and circulation, and vortex center are obtained. Each vortex can be tracked in a transient flow, and its trajectory is reconstructed with its decay characteristics. Two applications are presented in this paper: (i) an experimental columnar vortex moving through a free-surface water channel, and (ii) a numerical simulation of a bubbling fluidized bed. We demonstrate that VortexFitting can successfully identify the presence of vortices and characterize their features in both applications.

Code metadata

Current code version	1.0.1
Permanent link to repository used for this code version:	https://github.com/ElsevierSoftwareX/SOFTX-D-20-00015
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	Python 3.6+
Compilation requirements & dependencies	Linux/Mac/Windows scipy, numpy, netCDF4, matplotlib
Link to developer documentation	https://guilindner.github.io/VortexFitting/
Support email for questions	lindner.guilherme@alumni.ubc.ca

1. Motivation and significance

Turbulent flows are consisted of a great number of vortices, which are coherent in both time and space. While these coherent structures are the main components of unsteady flows, and significant research has been focused on their generation, evolution, and interaction with the mean flow, they are still not fully understood. Also, the terms coherent structure and vortices are often used interchangeably, and one may find it difficult to give a clear definition. In general, any form of pattern arising in a flow that has an effect on the flow itself is considered a coherent structure. On the other hand, the vortices are commonly associated with the existence of some rotational/swirling pattern in the flow. According to Shawn et al. (2005) [1], a definition of coherent structures is given as:

* Corresponding author.
E-mail address: lindner.guilherme@alumni.ubc.ca (G. Lindner).

https://doi.org/10.1016/j.softx.2020.100604

"It has long been recognized that flows with general time dependence admit emergent patterns which influence the transport of tracers, these structures are often generically referred to as Coherent Structures"

Hence, we can consider vortices to be specific types of coherent structures. The detection, tracking, visualization, and analysis of coherent structures is critical to explain the phenomena of turbulent motions, and this knowledge can be used to improve turbulent flow modeling and predictions. However, identification of vortices is difficult, which stems from the fact that it is not always clear how to measure the extension of the vortex from its center of rotation. The issue becomes more pronounced when multiple coherent structures interact.

The main goal of the developed software is to identify the vortices inside different velocity fields, which are typical for fluid mechanics applications. The emergence of coherent structures, which could be characterized by their scales, strength, direction, among other features, contains information about the type of the flow regime. However, a robust and reliable method for the extraction of these features is not fully established and readily available. To obtain experimentally the velocity fields, imaging methods such as Particle Image Velocimetry (PIV), Particle Tracking Velocimetry (PTV), and Laser Doppler Velocimetry (LDV) are available. They could also be obtained employing numerical methods used to simulate complex flows in high resolution such as Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES), which are capable of capturing the details of flow fields containing sharp gradients and small disturbances.

The VortexFitting algorithm is designed to minimize the need for user interaction and to automate the vortex detection process, followed by the theoretical vortex model fit and extraction of valuable flow characteristics. This platform can assist with detection, characterization, and tracking of vortices in a wide range of flow fields. Some of the potential application examples are: fluidized beds, vortex emission behind an obstacle in a flow, turbulence of a boundary-layer, ship wakes, wingtip vortices, and complex oceanic and meteorological flow structures, just to name a few. Another interesting use case is for the design optimization of a Vortex Tube. By outputting numerical results of the stagnation point (vortex center) and other characteristics, a tube geometry could be optimized with an automated routine. Raffie & Sadeghiazad [2,3] investigated the stagnation point using visual analysis, which could be quantified by using VortexFitting. The software has also been successfully used to characterize the propagation of a columnar vortex in a water channel and to study its decay with time [4].

2. Software description

The software detects vortices in a fluid flow, before extracting their relevant characteristics (scale, strength, location). The input data file should contain, at least, a two dimensional velocity field, namely $(u; v)$, with a uniform spatial mesh $(x; y)$ in a Cartesian referential. The third component, z — spatial and w — velocity, is optional. Once the velocity field is read, the VortexFitting program performs the three main operations: (i) the calculation of a local function to detect the vortex; (ii) the localization of the maxima of this function; and (iii) the adjustment (fitting) of each local maximum to a theoretical model.

2.1. Detection methods

In this section, the detection methods implemented in the code for vortex identification are presented. These methods are based on the velocity gradient tensor, $\bar{D}$, that can be written as Eq. (1), where, for $i = 1, 2, 3$, x_i and u_i correspond to $x; y; z$ and $u; v; w$, respectively.

$$D_{ij} = \frac{\partial u_i}{\partial x_j} \tag{1}$$

As this is a second order tensor, it can be decomposed into a symmetric, S_{ij}, and anti-symmetric, Ω_{ij}, components, where $D_{ij} = S_{ij} + \Omega_{ij}$. S_{ij} (Eq. (2)) is known as the rate-of-strain tensor and Ω_{ij} (Eq. (3)) is the vorticity tensor.

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2}$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{3}$$

The characteristic equation for the velocity gradient tensor is given by Eq. (4), where P, Q, and R are the first, second, and third invariants of $\bar{D}$, respectively.

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0 \tag{4}$$

Using the decomposition into symmetric and anti-symmetric parts, these invariants can be expressed as (Eqs. (5)–(7)):

$$P = -\operatorname{tr}(\bar{D}) \tag{5}$$

$$Q = \frac{1}{2}\left(\operatorname{tr}(\bar{D})^2 - \operatorname{tr}\left(\bar{D}^2\right)\right) = \frac{1}{2}\left(\|\Omega\|^2 - \|S\|^2\right) \tag{6}$$

$$R = -\det(\bar{D}) \tag{7}$$

Due to VortexFitting modularity, other mathematical definitions for a vortex, such as the Rortex [5], can be implemented as a detection method.

2.1.1. Q criterion

The Q criterion proposed by Hunt et al. (1988) [6] identifies the vortices as flow regions with positive second invariant of $\bar{D}$. An additional condition is that the pressure in the eddy region should be lower than the ambient pressure. Chakraborty et al. (2005) [7] explained that *"in an incompressible flow, Q is a local measure of the excess rotation rate relative to the strain rate"*. In practical terms, a vortex is detected in the case of $Q > 0$.

2.1.2. Δ Criterion

Chong et al. (1990) [8] defined a vortex core to be the region where $\bar{D}$ has complex eigenvalues. In order to determine if the eigenvalues are complex, the discriminant Δ of the characteristic equation is examined by Eq. (8) considering the flow is incompressible (i.e. $P = 0$).

$$\Delta = \left(\frac{Q}{3}\right)^3 + \left(\frac{R}{2}\right)^2 > 0 \tag{8}$$

2.1.3. Swirling strength criterion

The swirling strength criterion, λ_{ci}, was developed by Zhou et al. (1999) [9]. It defines a vortex core to be the region where $\bar{D}$ has complex eigenvalues. It is based on the idea that the velocity gradient tensor in Cartesian coordinates can be decomposed as follows (Eq. (9)):

$$\bar{D} = [\bar{v_r}\,\bar{v_{cr}}\,\bar{v_{ci}}]^T \begin{bmatrix} \lambda_r & 0 & 0 \\ 0 & \lambda_{cr} & \lambda_{ci} \\ 0 & -\lambda_{ci} & \lambda_{cr} \end{bmatrix} [\bar{v_r}\,\bar{v_{cr}}\,\bar{v_{ci}}]^T \tag{9}$$

where, λ_r is the real eigenvalue related to the eigenvector $\bar{v_r}$, and $\lambda_{cr} \pm i\lambda_{ci}$ is the complex conjugate pair of complex eigenvalues related to the eigenvectors $\bar{v_r} \pm i\bar{v_{ci}}$. The strength of this swirling motion can be quantified by λ_{ci}, which is called the local swirling

strength of the vortex. The threshold for λ_{ci} is not well-defined, but, theoretically, any value greater than zero should be considered as a vortex. Experimental results [9] show that values of $\lambda_{ci} \geq 1.5$ give smoother results.

In the case of a non-homogeneous direction, the normalization is applied to the swirling strength field, which produces smoother results. From here, the local maxima of the detected vortices can then be identified. The swirling strength is divided by the wall-normal profile of its RMS value given by Eq. (10).

$$\bar{\lambda}_{ci}(x_{1/3}, x_2) = \frac{\lambda_{ci}(x_{1/3}, x_2)}{\lambda_{ci,RMS}(x_2)} \tag{10}$$

Fig. 1 shows the swirling strength derived from a near-wall velocity field, demonstrating the effect of normalization data treatment on vortex detection. Data are obtained by a numerical simulation of a PIV case. The original swirling strength field is given in Fig. 1a, where a total of 104 vortices were found, mostly near the wall where the boundary layer plays an important role in increasing the swirling strength. Fig. 1b presents the swirling strength fluctuations by applying the Reynolds decomposition approach. Here, 202 vortices are found and the wall influence on the vortex detection is reduced. The results after the normalization step are given in Fig. 1c. Note that the distance between two detected vortices can be modified by increasing the box size at the detected peaks. In Fig. 1c, the box is set to 12 (in mesh-units) instead of box size of 6 used in Figs. 1a and 1b. The tool draws a square box with the desired size around each peak location, accepting only the peaks with the larger size (based on the built-in `scipy` function `maximum_filter`). This particular setting removes the overlapping vortices, reducing their number to 154 for Fig. 1b.

2.2. Fitting of coherent structures

Using the peak of maximum swirling strength or by identifying the locations where the Q or Δ criteria are higher than 0 provides a rough estimation of a possible vortex and position of its center. However, even when a threshold is applied with these methods, the detection of a real vortex is not always guaranteed. To improve the detection accuracy, a Lamb–Oseen vortex model is fitted on top of the primary detected peaks to provide an additional level of validation that the detected feature is indeed a vortex.

The correlation coefficient between the fitted model and the velocity field is calculated; if the correlation is higher than the minimum specified by the user, the vortex is accepted. The correlation threshold can be lowered (e.g. for a vortex sheared and deformed in a mean flow) by the users in VortexFitting based on their preference.

2.2.1. Lamb-Oseen vortex

The Lamb–Oseen vortex is a mathematical model for the flow velocity in the circumferential direction, $\vec{e}_\theta$:

$$\vec{u}(r,t) = \vec{u}_a + \frac{\Gamma}{2\pi r}\left[1 - \exp\left(-\left(\frac{r}{r_0(t)}\right)^2\right)\right]\vec{e}_\theta \tag{11}$$

where, r is the radius, $r_0 = \sqrt{4\nu t}$ is the vortex core radius, ν is the viscosity, and Γ is the circulation contained in the vortex. It models a line vortex that decays due to viscosity. Since the coherent structures are in movement, an advective velocity term, $\vec{u}_a$, is added to the Lamb–Oseen vortex model.

Eq. (11) is solved by using a non-linear least squares method to get different vortex parameters. Therefore, for each detected vortex, the software returns its location (x and y coordinates of the center), the circulation Γ, the core radius r_0, and the advective velocity components (u_a; v_a). The fitting is managed by the `scipy` package (function `least_squares`), which supports three algorithms: Levenberg–Marquardt, Powell's dogleg, or Trust Region Reflective.

2.3. Software architecture

The software is constituted of the following main blocks allowing for an easy way to add new methods, models, tools, or schemes:

- `_main_.py`: main program, which contains a parser to specify arguments and options. The algorithm is detailed in Fig. 2.
- `classes.py`: definition of the input data type, including examples using netCDF, HD5, Tecplot, and OpenFOAM. The user can adapt this file to accept any custom input file.
- `detection.py`: implementations of vortex detection methods such as Q criterion, Δ criterion, and swirling strength.
- `fitting.py`: algorithms involving the core functionalities of the VortexFitting. Functions such as normalization of the swirling strength field, location of the maxima, fitting to Lamb–Oseen vortex model, and plotting are included here.
- `schemes.py`: implements difference schemes such as 2nd, 4th order, and least squares filter [10].
- `output.py`: writes the output of fitted vortices characteristics.

The routine of the main program is described in Fig. 2, which is also detailing the fitting routine. The iteration loop allows the fitting window to be adjusted according to the core radius of the vortex and simultaneously change the center position of the vortex if needed. The software leverages the vectorized operations from `numpy` to achieve good calculation performance.

2.4. Software functionalities

The software offers several options to refine and automatize the detection. If a mean flow field is available, it can be specified by an optional argument and subtracted from the current velocity field. In the case of a sequence of images, the inline arguments (`first`, `last`, `step`) allow batch processing; the same detection features are applied to the complete image sequence and saved in a single directory. Two threshold parameters can be specified by the user: (i) a correlation threshold for the vortex fitting, and (ii) a swirling strength threshold for the vortex detection.

The software can handle all kind of spatial scales, without limitation for their number or size, but the user needs to provide the initial core radius magnitude. This is done using the `rmax` parameter; its value can be entered by the user or left at the mesh scale.

The `fitting.py` module contains some functions to help users investigate their data. User can plot a specific region from the velocity field, locate manually the potential vortices, and plot the vorticity field.

The software saves a plot (PDF or png format) of the spatial domain with the locations of the detected vortices (e.g. Figs. 1 and 4). The comparison between each vortex field and its Lamb–Oseen model is also saved in a separate file (e.g. Fig. 5). A text file is generated with all the relevant values recorded, including time step, core radius, circulation, correlation coefficient, and center location. Auxiliary data such as the detection method, the difference scheme used, and the mean field are stored to permit the reproducibility of the results.

3. Illustrative examples

3.1. Experimental vortex generation

VortexFitting has been used by Devaux et al. [4] to track vortices in an experimental system. Experiments took place in a free surface water channel, where a vertical thin plate is pitched

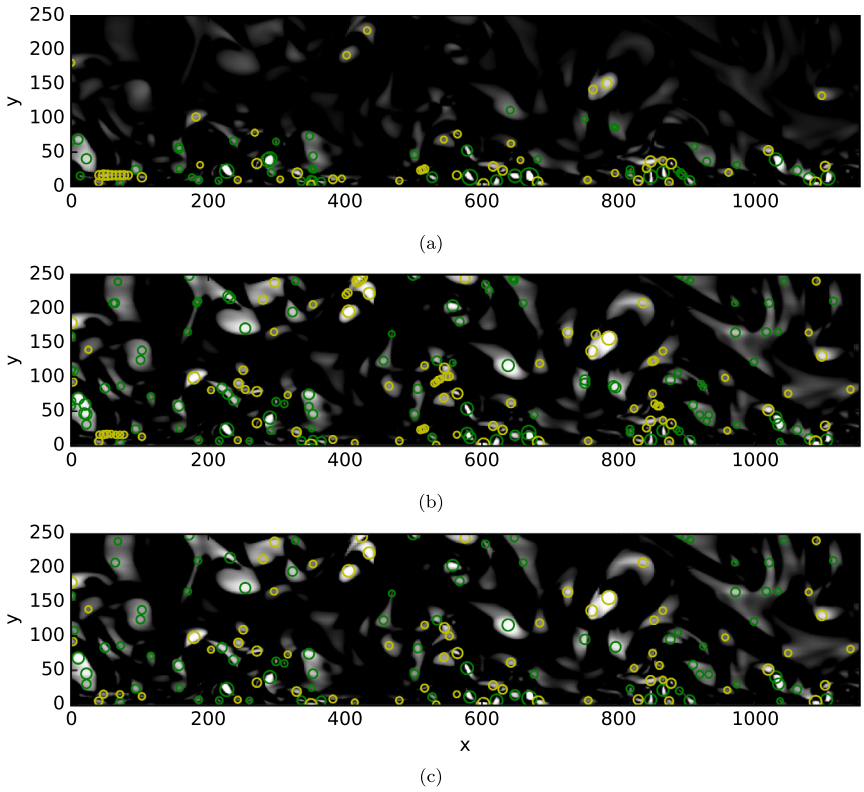


Fig. 1. Swirling strength field, (a) original, (b) fluctuation field, (c) normalized field with `boxsize=12` from a PIV velocity field. The yellow circles correspond to the vortices rotating in the clockwise and the green circles in the counter-clockwise direction. Spatial variables are measured in mesh units. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

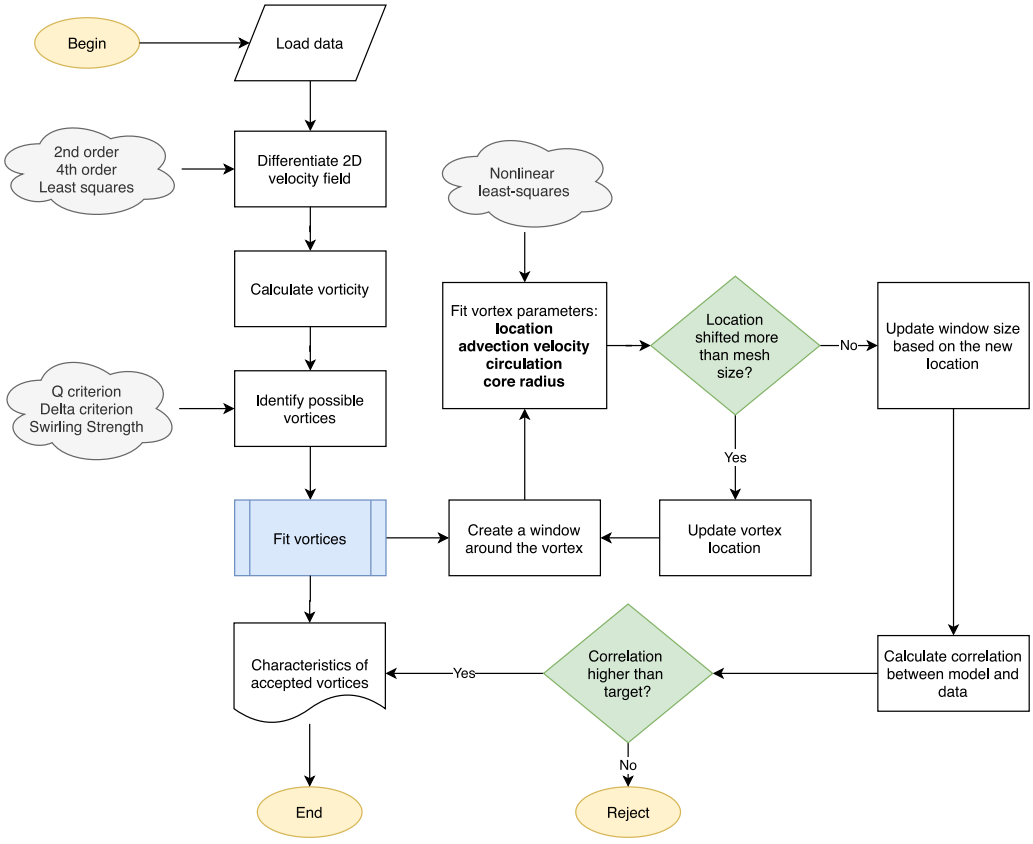
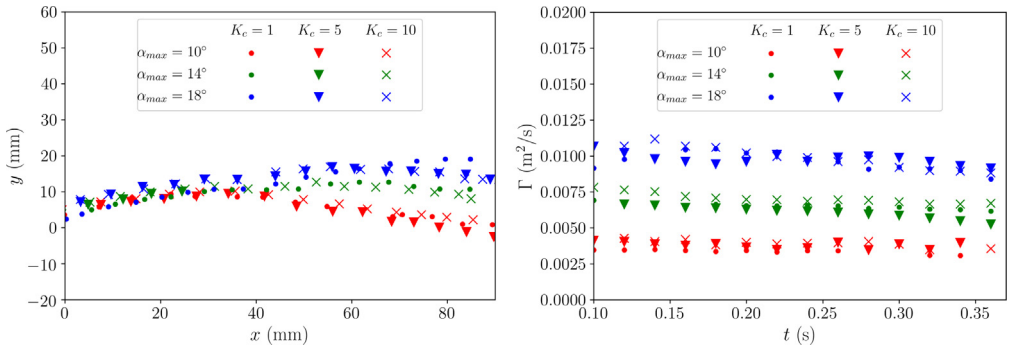


Fig. 2. Flowchart of the main program.



(a) Trajectories at $z = 50$ mm from the bottom (b) Circulation Γ at $z = 5$ mm from the bottom

```
TITLE="5runs_avg_mean_subtracted"
Variables="X","Y","Z","VX","VY","VZ"
ZONE T="10", I=289, J=214, SOLUTIONTIME=10
 0.0      −74.6006  5.0  28.97  112.33   59.95
 0.3283   −74.6006  5.0  12.90   −7.16  −12.91
 0.6566   −74.6006  5.0  12.45  −18.34  −16.08
 0.9849   −74.6006  5.0  −7.38   −7.81    3.58
 ...
```

(c) Example of used raw data (Tecplot format). X, Y, Z are in mm, VX, VY, VZ are in mm/s.

Fig. 3. Example of used raw data (Tecplot format). X, Y, Z are in mm, VX, VY, VZ are in mm/s.

and generates a reproducible vortex advected by the mean flow in the channel. Two parameters were investigated, namely the maximum angle $\alpha_{\max}$ of the pitching plate and its velocity K_c (non-dimensional parameter: ratio of the tip velocity and the flow velocity). Horizontal stereoscopic PIV measurements provided two-dimensional fields $(x; y)$ with three velocity components $(u; v; w)$ using 50 Hz acquisition frequency. Data are stored in a `.dat` file, with a Tecplot format (e.g. Fig. 3c).

For this experiment, VortexFitting was used to track the temporal dynamics of the vortex. An average velocity field is computed and subtracted with the software, and the core radius remained at a unique value ($r_0 \approx 5$ mm) for all cases. The $\alpha_{\max}$ and K_c parameters have been shown to have a direct influence on the vortex trajectory (Fig. 3a). Fig. 3b illustrates how the mechanical parameters $\alpha_{\max}$ and K_c influence the vortex strength obtained via the circulation Γ. The circulation was shown to decrease with time and the vortex eventually fades out.

3.2. Numerical fluidized bed

A two phase fluidized bed reactor, originally designed by the National Energy Technology Laboratory (NETL) [11], was simulated to test VortexFitting capabilities on highly turbulent particle laden flows. The geometry of the reactor is 0.23×0.076×1.2 m (W×D×H) and the inlet air flow rate was fixed at 2.19 m/s. The solid phase in the reactor consists of 92,947 glass beads with 3.25 mm in diameter with a total weight of 1.9 kg.

In this specific case, since the pattern of the flow is rapidly changing, both temporally and spatially, we first calculated the average air velocity over time. The average velocity served as an input to VortexFitting, which is then subtracted from the flow field at a desired time frame.

Fig. 4a shows the output from VortexFitting, where 70 locations are marked as possible vortices. Green dots correspond to counter-clockwise rotating vortices, while the yellow dots represent vortices with the clockwise rotation. Fig. 4b shows the accepted vortices, or those that achieved a correlation with the Lamb–Oseen vortex model criteria higher than 0.6. In Fig. 5 three accepted vortices are represented and compared with the correlated model.

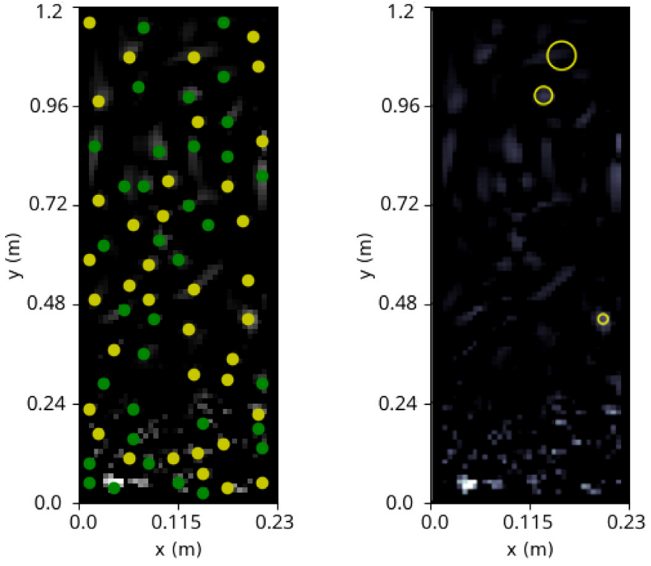


Fig. 4. Results of VortexFitting of fluidized bed simulation data. Possible vortices (green and yellow full dots) are shown on the left, and the validated vortices are given on the right (yellow circles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Impact and conclusions

VortexFitting provides a tool to identify and track vortices from the experimental or numerical velocity fields. In turbulent flow problems — from the studies of wall turbulence, flows downstream a foil or a turbine, to the ship wakes, coherent structures are present in various shapes and scales. The automation of the vortex detection process is of great interest to the research community, due to the immense quantity of the flow

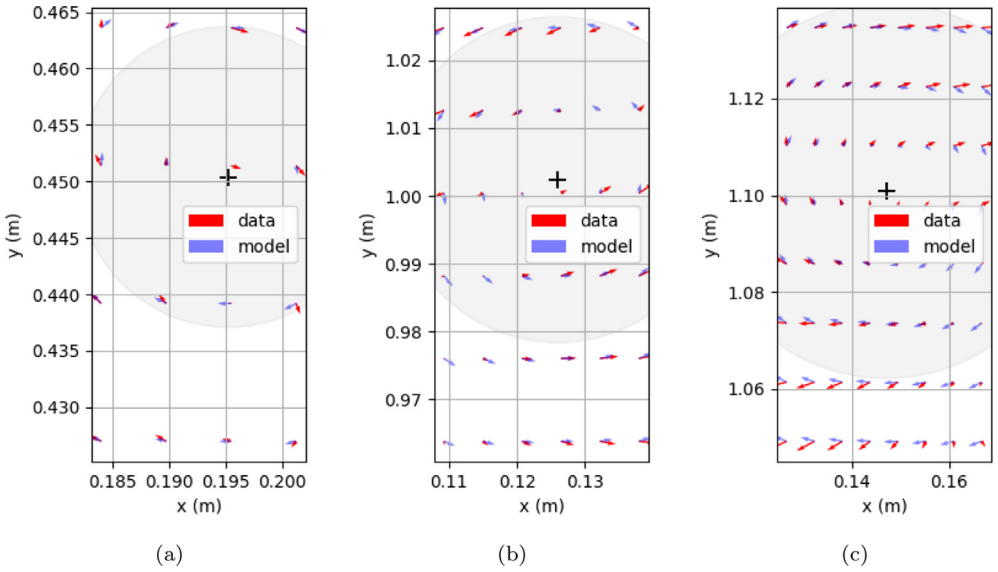


Fig. 5. Fitted vortices on the fluidized bed at$t = 14$ s.

data being generated. Considering experimental studies, spatial and time resolution of collected results using state-of-the art methods, such as PIV, are becoming more refined. The same trends could be observed with the turbulent flow simulations employing advanced numerical methods.

The implementation of the vortex detection and fitting algorithms in VortexFitting is divided into three main steps: (i) detection of possible vortices, (ii) fitting with the Lamb–Oseen model, and (iii) tracking of the vortex (for transient cases). From the several proposed vortex detection methods found in literature, three have been implemented in VortexFitting: Q criterion [6], Δ criterion [8], and swirling strength criterion, λ_{ci} [9]. The approximate location and quantity of vortices present in the flow are obtained in the first step. Once the vortices are located, they are fitted to a Lamb–Oseen model, where a correlation threshold ensures retention of only the real vortices. Critical vortex parameters, including circulation Γ, core radius r_0, tangential velocity at the core radius u_θ, center coordinates, among others, are extracted. The last step involves spatial and temporal tracking of the vortices, in addition to the comparison of the fitting parameters allowing investigation of the vortices scales and their dissipation over time.

The software has been successfully used for the tracking of a columnar vortex in a free-surface water channel, and for the structure identification in a highly turbulent bubbling fluidized bed. The platform can also be employed in many other cases related to fluid dynamics, as long as the velocity fields of the flow are available.

Currently, vortices are fitted to the Lamb–Oseen model with a 2D velocity field, but it can be extended to the third velocity component. For example, the w component can be predicted by a Turner equation [12] for near-wall flows. The implementation of this model would enable extraction of more features from the velocity fields ultimately providing a better description of the flow.

In some flows, vortices may be stretched in a particular direction. To improve the software further, the possibility of ellipsoidal fitting can be considered. Analysis of the stretching and the orientation with respect to the main flow would provide additional information about the underlying flow.

Declaration of competing interest

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

Acknowledgments

We acknowledge UBC and Doctoral Recruitment Fellowship as the primary source of support. The authors would like to thank the faculty members from Université de Lille, Jean-Philippe Laval and Jean-Marc Foucaut for the supervision during the first prototype of VortexFitting and Ilkay Solak for improving the code efficiency during early stages of its development. The authors also wish to thank Johan Fourdrinoy from Institut P′ for his constructive feedbacks on the software.

References

[1] Shadden Shawn C, Lekien Francois, Marsden Jerrold E. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. Phys D 2005;212:271–304. http://dx.doi.org/10.1016/j.physd.2005.10.007.

[2] Rafiee SE, Sadeghiazad MM. Experimental study and 3D CFD analysis on the optimization of throttle angle for a convergent vortex tube. J Mar Sci Appl 2016;15:388–404.

[3] Rafiee SE, Sadeghiazad MM. Experimental and 3D CFD analysis on optimization of geometrical parameters of parallel vortex tube cyclone separator. Aerosp Sci Technol 2017;63:110–22.

[4] Y. Devaux, L. Thomas, D. Calluaud, G. Pineau. Isolated columnar vortex generation: influence of momentum impulsion characteristics and wall roughness. Fluid Dyn Res 2020;52(2):025511. http://dx.doi.org/10.1088/1873-7005/ab7ebf.

[5] Liu Chaoqun, Gao Yisheng, Tian Shuling, Dong Xiangrui. Rortex—A new vortex vector definition and vorticity tensor and vector decompositions. Phys Fluids 2018;30(3):035103.

[6] J. Hunt, A. Wray, P. Moin. Eddies, streams, and convergence zones in turbulent flows, centre for turbulence research rep. CTR-S88, 1988.

[7] Chakraborty P, Balachandar S, Adrian RJ. On the relationships between local vortex identification schemes. J Fluid Mech 2005;535:189–214. http://dx.doi.org/10.1017/s0022112005004726.

[8] Chong MS, Perry AE, Cantwell BJ. A general classification of three-dimensional flow fields.. Phys Fluids 1990;2:765–77. http://dx.doi.org/10.1063/1.857730.

[9] Zhou J, Adrian RJ, Balachandar S, Kendall TM. Mechanisms for generating coherent packets of hairpin vortices in channel flow. J Fluid Mech 1999;387:353–96. http://dx.doi.org/10.1017/s002211209900467x.

[10] M. Raffel, C. Willert, F. Scarano, C. Kähler, S. Wereley, J. Kompenhans. Particle image velocimetry: A practical guide. Springer; 2018, http://dx.doi.org/10.1007/978-3-540-72308-0.

[11] B. Gopalan, F. Shaffer. Higher order statistical analysis of eulerian particle velocity data in CFB risers as measured with high speed particle imaging. Powder Technol 2013;242:13–26. http://dx.doi.org/10.1016/j.powtec.2013.01.046.

[12] J. Turner. The constraints imposed on tornado-like vortices by the top and bottom boundary conditions. J Fluid Mech 1966;25(2):377–400. http://dx.doi.org/10.1017/S002211206600171X.