Unsupervised extraction of vortices objectively defined from vorticity

1 Introduction

This MATLAB code allows for the extraction of objectively defined vortical structures from three dimensional turbulent flow fields. The code is based on the so-called Lagrangian Averaged Vorticity Deviation (LAVD) method by [1] and it constitutes a development of the one used by [2].

In this new version, several new features for the Vortex center line identification were added and control parameters with a clear range of variability were set-up for a user-independent employment. Together with the code, a test case of homogeneous and isotropic turbulence (HIT) LAVD field is provided. The HIT-LAVD field was downloaded from the John Hopkins data set.

2 Main Functions of the Code

To run the first two main functions of the code, it is necessary to download ”turbmat-master” folder from ”https://github.com/idies/turbmat” and add the path of John Hopkins database to the matlab functions.

2.1 Download of HIT Eulerian data sets

main00_JHVelExtractor.m provides the user with the possibility to download HIT flow velocity fields from the John Hopkins repository.

User-control parameters:

- *dt* time resolution
- *dd* space resolution
- *npoints_max* number of points

2.2 Computation of LAVD field from Eulerian data sets

main01_EulerianLAVDCalculator.m computes the LAVD field from flow velocity fields.

User-control parameters:

- $t0$ initial time for LAVD computation
- T time span for LAVD computation
- *dt_extr* time increment for extraction

2.3 Direct computation of LAVD field from JH data sets

Instead of the two-step procedure described above, **main01_JHLAVDExtractor.m** can be used to compute the LAVD field using HIT data set from JH repository.

2.4 Unsupervised ridge point location determination through gradient climbing

main02_RidgesDetectionGradientClimbing.m identifies the center of the structures (ridges of the LAVD field) through a gradient climbing algorithm. The ridge detection is composed of two main iterations: firstly, the ridges are identified by climbing the gradient of the LAVD field; secondly, to distinguish single ridges, an unsupervised clustering algorithm is employed, capable of classifying different ridges. Finally, a smoothing algorithm is applied to the different ridges.

To initialize the ridge climbing, a set of parameters are to defined. These parameters help for faster convergence of the algorithm, however their employment is not necessary. The following user-control parameters shall be defined:

- $thr0EigPrc$ threshold on the larger eigenvalue of the Hessian of the LAVD field. This value is expressed in percentage and shall be close to 0 (a ridge is a saddle with one null and two negative eigenvalues).
- $thrdVmPrc$ threshold on the intensity of the gradient of the LAVD field, expressed as the percentile of the grad(LAVD).
- $thrVmPrc$ threshold on the intensity of the LAVD field, expressed as the percentile of LAVD.

To stop the gradient climbing, an algorithm based on the comparison between subsequent binary images is applied. To this end, the 3D flow field is binarized at each time step based on the presence of trajectories of the grad(LAVD) field. When the relative difference between two subsequent binary images is below a threshold, the gradient climbing is stopped.

- $relDiff$ relative difference between two subsequent binary images expressed as a percentage of the initial difference.
- $stepNum$ maximum number of iterations of the gradient climbing.

For the final step, namely clustering and smoothing of the ridges, the following two parameters are to be set:

- $distmin$ minimum distance between points to belong to the same cluster, selected as fraction of the flow field resolution.
- $minlr$ number of left-right points to be discarded for the smoothing spline
- ws maximum window size for the smoothing spline

2.5 Unsupervised ridge point location determination through streamlines

main02_RidgesDetectionStreamLines.m is an alternative to the gradient climbing algorithm for the ridge identification. This module is based on MATLAB in built function ”stream3”. After computing the streamlines of the gradient of the LAVD field, the 3D space is subdivided into voxels and a bincounting algorithm counts the number of streamlines per voxel. The ridges position is thus selected as the the voxel position with the highest number of streamlines.

The user-control parameter of the ”Stream-Lines” method is the following:

- $Nprc$ threshold expressed in percentile of the number of streamline points in a bin.

For ridge clustering and smoothing, refer to the above section.

2.6 Vortex boundary identification

The boundaries of the vortices are computed as in [2], as the outermost almost convex contours of the LAVD filed in planes locally perpendicular to the vortex center line (**main03_OLCSeduction.m**).

The user-control parameters of this step are the following:

- $MinLength$ minimum length of the contours selected as a multiple of the flow filed resolution.
- $DeficiencyThresh$ a threshold that defines the convexity of the contour, defined as the difference in percentage between the area of inside to contour and the area of a contour convex hull.
- $Nval$ number of contours to be extracted.

References

[1] George Haller, Alireza Hadjighasem, Mohammad Farazmand, and Florian Huhn. Defining coherent vortices objectively from the vorticity. *Journal of Fluid Mechanics*, 795:136–173, 2016.

[2] Marius M Neamtu-Halic, Dominik Krug, George Haller, and Markus Holzner. Lagrangian coherent structures and entrainment near the turbulent/non-turbulent interface of a gravity current. *Journal of fluid mechanics*, 877:824–843, 2019.