Clustering-based Threshold Estimation for Vortex Extraction and Visualization

Kavya Padmesh1, Simon Ferrari1, Yaoping Hu1 and Robert J. Martinuzzi2

1 Dept. of Electrical and Computer Engineering
2 Dept. of Mechanical and Manufacturing Engineering
University of Calgary
Calgary, AB, CANADA
{kavya.padmesh, swferrar, huy, rmartinu}@ucalgary.ca

Abstract— Research efforts have been devoted to extraction and visualization of vortices in an unsteady (turbulent) flow. Characterizing the behaviors of the flow, vortices are identifiable as regions using a vortex detector known as the lambda2-criterion. Isosurface visualization renders vortex regions based on a chosen isovalue. However, it is highly challenging to choose one isovalue suitable for visualizing vortex regions of the entire flow field. A solution is the approach of maxima score that localizes vortex regions identified by the lambda2-criterion based on similarity scores relative to local extrema. The approach is however sensitive to noise or floating-point errors in the flow, leading to clutter in vortex visualization. As a feasibility study, this paper presents a threshold estimation to overcome this sensitivity. The estimation involves clustering on local minimum differences in lambda2 scalar values derived from the gradient tensor of the velocity field, and yields multiple values of the threshold without user intervention. Tested on several flows in various size and Reynolds number, the results of the threshold estimation confirmed overcoming the sensitivity of the maxima score approach. This indicates a potential of the threshold estimation to improve the robustness of the approach for vortex extraction and visualization.

Keywords— Information visualization, cluster analysis, threshold estimation, clutter reduction, vortices, unsteady flows

I. INTRODUCTION

The effectiveness of scientific visualization is largely dependent on accuracy and ease of recognizing features in complex data. In fluid dynamics, features like vortices characterize the behaviors of such complex data—an unsteady (turbulent) flow. A significant amount of research has been devoted to the extraction and visualization of vortices in the flow [1]. A region around high swirling motion of the flow is commonly considered as the location of a vortex. The high resolution of advanced empirical measurements and computational fluid dynamics (CFD) often yields a raw velocity field of hundreds or thousands of timeframes, resulting in terabytes of flow data. Traditional methods of vortex visualization, like streamlines and stream surface, suffer from clutter (i.e., congestion of graphical entities) due to high vortex densities [2, 3]. Such clutter obscures vortex features and reduces information available to the user. Hence, a robust technique of vortex extraction and visualization is imperatively essential to the concise presentation of vortex features for understanding flow behaviors.

Vortex extraction of an unsteady flow has a great importance in many industrial applications such as drag reduction of aerodynamic bodies [4], robust design of combustion engines [5] and efficiency of wind turbines [6]. Several vortex detectors—like \lambda_2-criterion [7], Q-criterion [8], vorticity [9] and helicity [10]—identify vortex regions by examining local neighborhoods of a velocity field [11]. The identification extracts vortex regions in 3D space at each timeframe. The extracted vortex regions are commonly rendered using the technique of isosurface visualization according to user-defined isovales (i.e., constants). Among the vortex detectors aforementioned, the \lambda_2-criterion is the one widely accepted. The criterion involves computing eigenvalues of the strain and spin matrix, \( S^2 + \Omega^2 \), that is derived from the gradient tensor of a velocity field [7]. The \lambda_2-criterion identifies vortices at regions, where the scalar values of \lambda_2 are negative. Isosurfaces of these scalar values render the vortices by setting suitable isovales below zero.

One shortcoming of these vortex detectors is their inability to identify all vortex regions [12]. Moreover, the selection of an appropriate isovale for differentiating vortex regions requires domain knowledge of fluid dynamics and can rather be subjective [11]. It is highly challenging to choose one isovale suitable for the entire field of an unsteady flow. A lower isovale might lead to lumping individual high-intensity vortices together as a single feature, resulting poor differentiation among the vortices. In contrast, a higher isovale might reject low-intensity vortices as features, causing omission of the vortices. Thus, there is a need for a solution to normalize isovales according to the variation of relative intensities in vortex regions.

A solution is the approach of maxima score [13]. The approach localizes vortex regions identified by the \lambda_2-criterion based on similarity in vortex intensities. However, the outcome of the approach is sensitive to a chosen value of the threshold, T, which accommodates for inherent noise or floating-point errors in the raw velocity field. As a feasibility study, this work aims to overcome the sensitivity of the maxima score approach with the following requirements:

Research supported by the Natural Sciences and Engineering Research Council of Canada (Grants: RGPIN-2015-06601 to Y. Hu and STPGP 478870 to Y. Hu and R. Martinuzzi).
• Threshold estimation for the maxima score approach of vortex extraction and visualization;
• Robustness improvement of the approach for various flows with different size and Reynolds number.

II. RELATED WORK

Vortex detectors analyze the gradient tensor of a velocity field. Acquired by empirical measurements or by discrete CFD computations, the velocity field often contains noise and floating-point errors. Based on this field, vortex identification uses detectors such as \( \lambda_2 \)-criterion [7], Q-criterion [8], vorticity [9] and helicity [10]. Each detector performs differently with flow datasets of various size and Reynolds number; and can lead to false positives in vortex extraction [14].

Solutions of reducing false positives involves multiple vortex detectors. For example, Biswas et al. [15] developed a fuzzy based algorithm to combine uncertainty in the outputs of several vortex detectors. A fuzzy index relating to the outputs of each vortex detector differentiates vortex and non-vortex regions. Bremer et al. [16] suggested a relevance metric to overcome ambiguity in vortex extraction. They used topological encoding of multiple vortex detectors to identify vortex regions. Zhang et al. [17] incorporated an algorithm (AdaBoost) of machine learning with an expert-in-loop approach to improve the outputs of their vortex detectors. These solutions enhance the accuracy of vortex extraction, but increase computational complexity because of using multiple vortex detectors. Furthermore, vortex extraction requires domain knowledge of fluid dynamics to determine user-defined criteria such as shape or spatial extent of vortices [11]. Manual labeling vortex regions by domain experts (i.e., users) is a way of incorporating domain knowledge into the detectors.

Due to inabilities of the aforementioned detectors to identify vortex regions without user intervention, an alternative approach is to extract and visualize vortices based on “significance”. This leads to our previous work of developing a normalized significance metric – maxima score [13]. The approach of maxima score identifies “significance” of regions in a flow field by comparing them to their neighborhoods. The maxima score approach localizes vortices as most “significant” regions based on the \( \lambda_2 \) scalar field, because the \( \lambda_2 \)-criterion is most reliable for vortex identification among existing vortex detectors [18]. However, the maxima score approach is sensitive to noise and floating-point errors, causing clutter in vortex visualization.

In brief, the maxima score approach scores each grid point on a given scalar field based on its similarity to local extrema (i.e., local minima/maxima) within a size-fixed neighborhood [13]. The center of the neighborhood is on each grid point and moves over the entire field. To overcome the misidentification of “significant” regions, a difference within the neighborhood is comparable to a reference threshold, \( T \). The comparison is to reduce the effect of noise or floating-point errors on the vortex identification. Neighboring points with a difference above the threshold contribute towards a maxima score. However, the estimation of an appropriate \( T \) value is a challenging task for complex flows, as undertaken in our previous work [13].

To overcome the sensitivity of the maxima score approach, there is a need of multiple thresholds estimated from an input scalar field. The estimation shall ideally be without user intervention. A suitable candidate for the estimation is cluster analysis – a technique of unsupervised machine learning. The suitability arises from its application to vortex analyses [19, 20, 21]. For replacing \( T \), we herein undertook a feasibility study on using the clustering analysis to estimate data-dependent thresholds, \( T_{clust} \). The thresholds obtained for each grid point of a flow field are then applicable to the maxima score approach for vortex extraction and visualization.

III. METHOD

Figure 1 depicts the process of vortex extraction and visualization for a given unsteady flow, i.e., a velocity field on a 3D rectilinear grid with many timeframes. For each timeframe, the \( \lambda_2 \)-criterion is a vortex detector to convert the velocity field into its corresponding \( \lambda_2 \) scalar field on the same grid. The dash-lined box in Fig. 1 contains steps for extracting vortices based on the \( \lambda_2 \) field before their visualization. The gray shaded box within the dash-lined box highlights the focus of this feasibility study to estimate thresholds, \( T_{clust} \), using cluster analysis. Based on local minimum differences among all grid points, the thresholds scale to variations in local intensities of the \( \lambda_2 \) field. For vortex extraction, the maxima score approach uses these thresholds to assign scores to each grid point based on its similarity to local extrema.

Estimation of the thresholds, \( T_{clust} \), has two sub-steps: local difference computation and K-means cluster analysis.

A. Local difference computation

This sub-step computes local differences between each grid point and its neighbors, and stores the minimum of the differences in a 1D array, \( X \). Each minimum difference is within a neighborhood of the fixed size, \( 5 \times 5 \times 5 \). This size balances computational details and time, as described in our earlier work [13]. The center of the neighborhood is on a grid point of the
field. Let the center of the neighborhood be \( p_0 \), the absolute differences between \( p_0 \) and all its neighbors in the neighborhood are sortable in an ascending order. The first value of this order is thus the local minimum difference of the neighborhood for storing into the array, \( X \). The computation and storing are repetitive through moving the center, \( p_0 \), of the neighborhood one point on the grid until covering the entire field. Being univariate, the 1D array, \( X \), corresponds to the local minimum differences of the field and functions as an input to K-means cluster analysis in the next sub-step.

### B. K-means cluster analysis

K-means cluster analysis is widely applicable [22], due to its simplicity and computational efficiency. In this feasibility study, the cluster analysis takes the array, \( X \), as an input and groups the local minimum differences of the \( \lambda_2 \) field into several clusters. Thereby, the centroid, \( \mu \), of a cluster is the average of all array elements in the cluster and corresponds to one value of the thresholds, \( T_{clust} \).

The outcome of the K-means analysis is greatly dependent on the initialization of centroids, \( \mu_{\text{init}} \). Based on a probability of each array element, \( x_i \), of \( X \), the initialization takes place using a k-means++ algorithm [23]. The probability is proportional to the squared Euclidean distance, \( D(x_i) \), of the element from its nearest centroid. The initialization with k-means++ promotes fast convergence for clustering [23].

Algorithm 1 describes our K-means algorithm with the k-means++ initialization. Each iteration of the K-means algorithm saves a heterogeneity value, \( H \), for the chosen number of clusters \( K \). Ideal clustering should result in minimum heterogeneity, where the similar array elements of \( X \) are in the same cluster and separate clearly from the dissimilar elements of \( X \) in other clusters. A minimum heterogeneity determines the final centroids and their associated elements. For a maximum number of iterations, \( \text{maxiter} \), specified empirically [22], the K-means algorithm terminates the iterations until reaching this number. Each iteration yields a cluster assignment, \( Z \), to produce centroids, \( \mu \), and their associated heterogeneity, \( H \). The centroids corresponding to the minimum value of the heterogeneity is then the estimation of the thresholds, \( T_{clust} \).

The K-means algorithm requires specifying the number of clusters, \( K \), for clustering. Since an optimal number of clusters is unknown in most flow data, certain internal validation measures like silhouette coefficient [24] enable to determine the cluster number in the data. Silhouette coefficient serves to evaluate the quality of clusters. The higher the average silhouette coefficient is, the better the clustering accuracy will be. The silhouette coefficient, \( S(i) \), of each cluster element, \( x_i \), ranges from -1.0 to 1.0 for a given input array, \( X \). The value of -1.0 indicates a poor cluster assignment of \( x_i \), and the value of 1.0 corresponds to a proper cluster assignment of \( x_i \). The maximum average silhouette coefficient among all possible \( K \) clusters gives the optimal number of the clusters. For the input array, \( X \), with \( K \) clusters, the silhouette coefficient of each array element, \( x_i \), is thus obtainable by

\[
S(i) = \frac{b(i) - a(i)}{\text{max}(a(i), b(i))},
\]

where \( a(i) \) is the average Euclidean distance between \( x_i \) and all other elements of the cluster it belongs to, and \( b(i) \) is the minimum Euclidean distance between \( x_i \) and all elements within other clusters.

The final cluster centroids, \( \mu_{\text{final}} \), are computable based on the chosen number of clusters, \( K \), for the given input array, \( X \). Each of those centroids is the average of the array elements belonging to a cluster in the final cluster assignment, \( Z \). Since \( X \) contains the minimum differences of \( \lambda_2 \) values for all grid points, the cluster centroids produce thresholds, \( T_{clust} \), which are applicable for computing a maxima score of corresponding grid point.

For the flow with multiple timeframes, the clustering-based threshold estimation computes the thresholds, \( T_{clust} \), on a single timeframe to reduce computation time. A short computation time offers an incentive to investigate the feasibility of applying the estimated thresholds in vortex extraction and visualization of other timeframes. Thus, the maxima score approach uses the thresholds to extract vortices, and isosurface visualization renders the outcomes of the extraction for analysis. We implemented the K-means algorithm and isosurface rendering using C++ and the visualization toolkit (VTK).

### IV. RESULTS AND DISCUSSION

Using three flow datasets, we evaluated the feasibility of clustering-based thresholds, \( T_{clust} \), for vortex extraction and visualization based on their \( \lambda_2 \) scalar fields. We rendered the outcomes of the extraction using isosurface visualization, with
an isovalue of 0.0 for the $\lambda_2$ scalar field of each dataset and an isovalue of -0.6 for the maxima scores of the dataset [13].

Table I describes these three datasets with various levels of the Reynolds number ($Re$), different grid dimensions and dissimilar numbers of timeframes. Each dataset was a 3D velocity field on a 3D rectilinear grid with multiple timeframes. The first two datasets, Pyramid high cycle (HC) and Pyramid low cycle (LC), were the outcomes of empirical measurements in the wake of a pyramid-shaped object (45 x 45 x 38 mm). The grid of the dataset was uniform along the x-, y- and z-axis. With the same Reynolds number ($Re = 28000$), both datasets had their flow direction mainly along the x-axis. This $Re$ is very high, indicating strong turbulence in the flows. Both HC and LC corresponded to two different intensities of vortex shedding in the pyramid wake. The third dataset, Plate, was the result of a CFD simulation in the wake of a plate-shaped object (147 x 87 x 32 mm). The grid of the dataset was non-uniform; and the flow direction of the dataset was perpendicular to the plate-shaped object along the x-axis. With a moderate $Re (= 1200)$, the dataset had repetitive turbulence along the y-axis and complex shedding in the x-axis of the wake.

Table II gives the processing time of the K-means algorithm (Algorithm 1) for different numbers of clusters on all three datasets. The processing time was obtainable on a graphic workstation with an Intel Xeon E5540 CPU at 2.5GHz. As shown in Table II, both Pyramid HC and Pyramid LC have the same processing time for each cluster number, because of their same grids. For the same cluster number, the time needed for processing the Plate dataset is longer due to its much larger grid size than that of both Pyramid datasets. The increase in the cluster number leads to raising the processing time for all three datasets. Figure 2 depicts the relationships among clustering-based thresholds, silhouette coefficients and the number of clusters for all three datasets. The relationship between clustering-based thresholds, $T_{clus}$, and the number of clusters, $K$, is representative in Fig. 2(a) with $K$ varied up to 7 for each dataset. The thresholds increase as the number of clusters raises. As shown in Fig. 2(b), the average silhouette coefficient of each dataset is inversely related the number of clusters. The maximum average silhouette coefficient is at $K = 2$ for all datasets. To minimizing computation time and maximizing the average silhouette coefficient, the number of clusters is thus set as $K = 2$ for further analyses.

To evaluate the effect of $T_{clus}$ on the maxima score approach, we computed three $T_{clus}$ values on timeframe 1, timeframe 25 and timeframe 50, respectively, among all available 50 timeframes of the Pyramid HC dataset and then applied each $T_{clus}$ value to the same other timeframes of 5, 31 and 46 for comparison. The use of these timeframes were random among all timeframes of the dataset. The comparison used a manual threshold, $T = 0.000001$ [13], as reference on the same other timeframes. Figure 3 presents comparatively isosurface visualizations of vortices extracted by using the maxima score approach with a manual threshold $T$ and clustering-based thresholds $T_{clus}$. For the case in Fig. 3(a), the manual threshold was applicable to other timeframes of 5, 31 and 46 to serve as illustrative references. The three cases in Figs. 3(b), 3(c) and 3(d) gave vortex visualizations of the same timeframes by using the $T_{clus}$ value computed on timeframe 1, timeframe 25 and timeframe 50, respectively. As shown in Fig. 3, the visualizations of all cases are similar for each timeframe in comparison. This similarity indicates that the thresholds, $T_{clus}$, computed on anyone timeframe might be

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Reynolds number ($Re$)</th>
<th>Grid dimensions (X x Y x Z)</th>
<th>Number of timeframes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pyramid HC</td>
<td>28000</td>
<td>30 x 31 x 18</td>
<td>50</td>
</tr>
<tr>
<td>Pyramid LC</td>
<td>28000</td>
<td>30 x 31 x 18</td>
<td>100</td>
</tr>
<tr>
<td>Plate</td>
<td>1200</td>
<td>256 x 204 x 64</td>
<td>20</td>
</tr>
</tbody>
</table>

**TABLE II. PROCESSING TIME FOR CLUSTERING FLOW DATASETS**

<table>
<thead>
<tr>
<th>Number of clusters $K$</th>
<th>Pyramid HC, Pyramid LC (s)</th>
<th>Plate (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.40</td>
<td>125.25</td>
</tr>
<tr>
<td>3</td>
<td>0.55</td>
<td>167.70</td>
</tr>
<tr>
<td>4</td>
<td>0.70</td>
<td>222.30</td>
</tr>
<tr>
<td>5</td>
<td>0.90</td>
<td>281.90</td>
</tr>
<tr>
<td>6</td>
<td>1.12</td>
<td>335.13</td>
</tr>
<tr>
<td>7</td>
<td>1.13</td>
<td>401.44</td>
</tr>
</tbody>
</table>

Fig. 2. Relationships among clustering-based thresholds, silhouette coefficients and the number of clusters for all three datasets: (a) clustering-based thresholds, $T_{clus}$, vs. the number of clusters; and (b) the average silhouette coefficient vs. the number of clusters.
suitable for multiple timeframes of the dataset. A similar observation was true for the Pyramid LC dataset. Hence, the use of clustering-based threshold \( T_{\text{clus}} \) to replace the manual threshold \( T \) could be feasible for the maxima score approach, as verified in the Plate dataset.

Using the Plate dataset, Figure 4 exemplifies the application of \( T_{\text{clus}} \) to deal with the sensitivity issue of the maxima score approach. Both Figs. 4(a) and 4(b) give isosurface visualizations of vortices in this dataset at timeframe 1 with \( T = 0.000001 \) and \( T = 0.000001 \), respectively. As the vortices circled by ellipses indicate, the smaller \( T \) produced over-extraction of vortices to cause clutter in the visualization, whereas the larger \( T \) failed to capture vortices downstream. The incorporation of \( T_{\text{clus}} \), computed at the same timeframe, yielded a vortex visualization with reduced clutter while retaining vortices downstream, as depicted in Fig. 4(c). Thus, the use of \( T_{\text{clus}} \) overcomes the sensitivity of the maxima score approach.

In order to quantify the effect of \( T_{\text{clus}} \) on vortex extraction using the maxima score approach, a statistical analysis was conducted to measure the accuracy of vortex extraction. The analysis was based on a F1 score [25] - a conventional way of quantifying data classification, because vortex extraction identifying vortices versus non-vortices in a flow is similar to data classification. The outcomes of vortex extraction were either as positive for vortices or negative for non-vortices. In the absence of a truth ground, we considered vortices identified by the \( \lambda^2 \)-criterion (where \( \lambda^2 < 0 \)) as a true reference in the analysis. Regions (i.e., grid points) in a flow with a normalized score ranging from -0.9 to -0.5 were part of vortices, according to the same range in our previous work [13]. Thus, a grid point met both the \( \lambda^2 \)-criterion and this score range belonged to a vortex as a true positive (TP). A grid point failed the criterion but fulfilled the range indicated a false positive (FP), whereas a grid point satisfied the criterion but failed the range revealed a false negative (FN). A F1 score is thus a weighted average of precision and recall based on TP, FP, and FN [25].

Figure 5(a) shows the relationship between the F1 score and different \( T \) values in vortex extraction by using the maxima score approach. The F1 score of the Plate dataset decreases drastically in the range of \( T \geq 1.00 \times 10^{-5} \), whereas the F1 score of the Pyramid HC dataset changes for \( T \geq 1.00 \times 10^{-2} \). The F1 score of the Pyramid LC dataset remains constant over all \( T \) values in consideration. A stable F1 score signifies the suitability of a single value of \( T \leq 1.00 \times 10^{-7} \) for the entire flow of both Pyramid HC and Pyramid LC datasets. However, a large variation in the F1 score of the Plate dataset constrains the value of \( T \leq 1.00 \times 10^{-5} \) for vortex extraction. In comparison, the F1 score of each dataset is relatively stable over the number of clusters, \( K \), when vortex extraction uses \( T_{\text{clus}} \) values as depicted in Fig. 5(b). This observation is especially significant for the Plate dataset due to its large variation of vortex intensities in the flow. The stability of the F1 scores over different \( K \) indicates not only the suitability of using \( K = 2 \) to obtain \( T_{\text{clus}} \) for saving computation time, but also the reduced sensitivity of the maxima score approach for all three datasets. This reduction indicates a feasibility of the clustering-based threshold estimation for improving vortex extraction and visualization. However, more datasets with high variations of vortex intensities would warrant a better determination of the \( K \) value for obtaining \( T_{\text{clus}} \) for the extraction and visualization.

Two observations are derivable from the above results: the K-means clustering analysis is feasible to estimate thresholds
**Figs. 5.** F1 score of vortex extraction by the maxima score approach over: (a) a range of $T$ values and (b) $T_{c1ust}$ values corresponding to varying $K$.

$T_{c1ust}$ for vortex extraction and visualization; and the thresholds $T_{c1ust}$ improve vortex extraction by overcoming the sensitivity of the maxima score approach. For the given datasets, $T_{c1ust}$ computed at any single timeframe is suitable for vortex extraction over remaining timeframes by using the maxima score approach. Nevertheless, the applicability of $T_{c1ust}$ obtained at one timeframe to vortex extraction over numerous timeframes would require further evaluation with very large-scale datasets containing thousands of timeframes.

**V. Conclusion**

We developed clustering-based threshold estimation for vortex extraction and visualization. Tested on several datasets of varied size and Reynolds number, the results of the test indicate the fulfillment of two requirements defined for this work. Thus, the clustering-based threshold estimation is a feasible solution to reduce the sensitivity of the maxima score approach. Future work is to compare the performance of multiple clustering algorithms in the threshold estimation and evaluate its effectiveness on very large-scale datasets with high variations of vortex intensities.

**ACKNOWLEDGMENT**

The authors would like to thank our collaborators for offering their flow datasets leading to feasibility study on threshold estimation.

**REFERENCES**


