Towards In-Situ Vortex Identification for Peta-Scale CFD Using Contour Trees

Marius K. Koch*
Dept. of Aeronautics
Imperial College London

Paul H. J. Kelly†
Dept. of Computing
Imperial College London

Peter E. Vincent‡
Dept. of Aeronautics
Imperial College London

Abstract

Turbulent flows exist in many fields of science and occur in a wide range of engineering applications. While in the past broad knowledge has been established regarding the statistical properties of turbulence at a range of Reynolds numbers, there is a lack of understanding of the detailed structure of these flows. Since the physical processes involve a vast number of structures, extremely large data sets are required to fully resolve a flow field in both space and time. To make the analysis of such data sets possible, we propose a framework that uses state-of-the-art contour tree construction algorithms to identify, classify and track vortices in turbulent flow fields produced by large-scale high-fidelity massively-parallel computational fluid dynamics solvers such as PyFR. Since disk capacity and I/O have become a bottleneck for such large-scale simulations, the proposed framework will be applied *in-situ*, while relevant data is still in device memory.

Index Terms: Applied Computing—Physical Sciences and Engineering —Physics; Applied Computing—Physical Sciences and Engineering —Aerospace;

1 Introduction

The emergence of unsteady peta-scale turbulent flow simulations is leading to generation of a significant amount of high-fidelity data. Insight from this data could help address a range of open problems in the field of turbulent flow physics. There are lots of unanswered questions regarding the dynamics of turbulent flow features, e.g. whether hairpin vortices persist in a fully turbulent regime.

Fig. 1 shows iso-surfaces of Q-criteria, which define vortical structures, in the wake of a single T106D low pressure turbine blade [7] obtained using the PyFR [9] computational fluid dynamics (CFD) solver. The flow is highly complex and includes a vast number of turbulent structures across a wide range of scales in both space and time, resulting in an extremely large data set.

Here we propose a framework to identify, track and classify all vortical structures in such large time-dependent data sets using contour trees (CT) of the Q-criterion field, isosurfaces of which define vortical structures. A CT is a representation of a scalar field that records when contours join, split, appear and disappear, when the isovalue is varied. Hence the CT provides nesting relations of vortices. Moreover, each branch of the CT maps to the volume within a vortex in physical space, allowing one to gain further information about each vortex, e.g. its internal pressure distribution, or its shape. Using these results algorithms based on e.g. machine learning approaches can be trained to classify vortices. Finally, once identified and characterised, CTs can also be used to help track vortices in time, providing information on their evolution and interaction. In this way we can obtain statistics on the abundance of different types of vortices, as well as their characteristic behaviour.

*e-mail: marius.koch16@imperial.ac.uk
†e-mail: p.kelly@imperial.ac.uk
‡e-mail: p.vincent@imperial.ac.uk

Figure 1: Snapshot of iso-surfaces of Q-criteria coloured by velocity magnitude in the wake of a single T106D low pressure turbine blade [7].

Currently, disk capacity and I/O have become a bottleneck for large-scale turbulent flow simulations, prohibiting the classical approach of writing data to disk and analysing it in a post-processing step. This proposed framework will address this challenge by applying the outlined analysis technologies *in-situ*, so the analysis happens concurrently with the simulation, extracting insight on the fly while the data is still in system memory. Specifically, in a first instance the proposed framework will be integrated with the PyFR CFD solver. PyFR is well suited to producing the flow field, since its highly parallel nature allows simulations of large problems and its high-order capability enables efficient resolution of the unsteady turbulent flow physics. The proposed framework could furthermore be applied to other scalar fields, isosurfaces of which are also considered to identify vortices, e.g. λ_2. This can pave the way to a comparison of their quality.

Thus far algorithmic choices for the proposed framework have been made (Sect. 2). Furthermore a new method to identify and classify saddle points on grids of linear hexahedral elements has been developed and implemented in the current version of the framework (Sect. 3). To date, the framework is able to read a snapshot of a scalar field from disk onto a GPU, augment it with saddles, and apply the CT construction algorithm. Correctness of the implementation thus far has been demonstrated using a verification test case (Sect. 4). Finally, Sect. 5 provides an overview of next steps that need to be undertaken in order to complete the proposed framework, along with discussion around two turbulent flow physics problems to which the framework, once complete, could be applied.

2 Contour Tree Algorithm Choices

The proposed framework is designed from the ground up to exploit massively parallel multi-GPU systems. Specifically, the current version of the framework employs the data-parallel CT algorithm by Carr et al. [1] to construct the local merge trees (MT) on each GPU. From the local MTs one can compute the distributed MT introduced by Morozov and Weber [4], a local global representation of the MT. The combination of all distributed MTs implicitly stores the information of the global MT without the need of constructing it on a single GPU. By storing the minimal and maximal root of every vertex the information of distributed CTs is provided [5].

In order to track the vortices over multiple time steps, the algorithm of Saikia and Weinkauf [6] will be implemented. The proposed framework will in this way be enabled to analyse the evolution of a statistically significant amount of vortices within a turbulent flow field.

3 Saddle Point Identification on Meshes of Linear Hexahedra

In order to construct the MTs with Carr's algorithm from scalar data on the computational mesh, it must include all saddles points of the data as well as all splits and merges. These requirements are inherently satisfied by linear simplex meshes. However, PyFR is a mixed element solver, and solutions can also be provided on meshes of linear hexahedra, for which these requirements are not inherently satisfied.

The current version of the framework incorporates new algorithms that we have developed to address this issue by identifying saddles on the faces or in the volume of hexahedra. This is a critical prerequisite for constructing MTs. Furthermore, these new algorithms provide information on whether a split or a merge appears at a given saddle. This information can be used to increase the performance of MT construction. After augmenting the mesh with the saddles and connecting them to the mesh through appropriate edges, Carr's algorithm can be applied.

4 Verification Test Case

To test the correctness of the current version of the framework, it has been applied to a simple verification test case involving a mesh of eight linear hexahedra. All hexahedra include at least one face saddle and two of them include volume saddles. The resulting CT can be seen in Fig. 2.

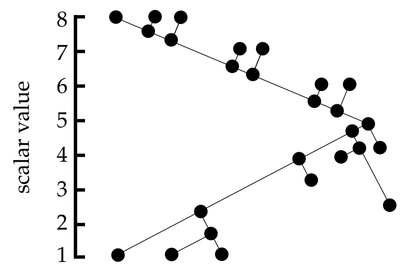

Figure 2: CT obtained from the simple verification test case; black dots indicate minima, maxima and saddle points.

5 What is left to do?

Various next steps need to be undertaken in order to complete the proposed framework. Firstly implementation of the distributed CT algorithm must be completed to enable analysis of data spread over multiple GPUs. Secondly, pruning algorithms need to be implemented to removed leaves of the CT that are caused by noise. Thirdly, once all structures can be robustly identified, algorithms to extract information about each vortex, e.g. its internal pressure distribution, or its shape need to be developed and implemented. Using this information, vortices can be classified into types, e.g. hairpin vortices or coherent and incoherent structures. Fourthly, the tracking algorithm of Saikia and Weinkauf [6] must be implemented. Finally, the data structure of the proposed framework needs to be adapted to align with that employed by PyFR, and both frameworks need to be merged to enable highly-efficient *in-situ* analysis of the CFD simulations.

Two candidate test cases for the proposed framework have been identified. This first involves analysing hairpin vortices in a flat plate turbulent boundary layer (TBL), and the second involves distinguishing between rod shaped vortices and disc shaped vortices in homogeneous isotropic turbulence (HIT). Applied to a TBL the framework can provide direct evidence for hairpin vortices in turbulent boundary layers [2]. In HIT we are interested in the abundance and production of rod and disc shaped vortices. Also direct evidence of the tendency for rod-like structures to change to disc-like structures [3] could be provided by tracking and classifying a statistically significant amount of vortices.

This knowledge could be used to obtain statistics on different types of vortices, leading to a deeper understanding of turbulent flow physics. More specifically, this knowledge may lead to a better understanding of mixing processes, transport of passive scalars and flow separation, and provide the foundation for new Reynolds-Averaged Navier Stokes and Large-Eddy Simulation turbulence models [8].

6 Conclusion

We have proposed a framework that uses state-of-the-art CT construction algorithms to identify, classify and track vortices in turbulent flow fields produced by large-scale high-fidelity massively-parallel computational fluid dynamics solvers such as PyFR. Since disk capacity and I/O have become a bottleneck for such large-scale simulations, the proposed framework will be applied *in-situ*, while relevant data is still in device memory. Thus far algorithmic choices for the proposed framework have been made. Furthermore a new method to identify and classify saddle points on grids of linear hexahedral elements has been developed and implemented in the current version of the framework. To date, the framework is able to read a snapshot of a scalar field from disk onto a GPU, augment it with saddles, and apply the CT construction algorithm. Correctness of the implementation thus far has been demonstrated using a verification test case. Once complete, the proposed framework will be able to obtain statistics on different types of vortices, leading to a deeper understanding of turbulent flow physics.

Acknowledgments

Marius Koch and Peter Vincent would like to thank the Leverhulme Trust for their support via a Philip Leverhulme Prize. Peter Vincent would also like to thank the EPSRC for their support via an Early Career Fellowship (EP/K027379/1).

References

[1] H. A. Carr, G. H. Weber, C. M. Sewell, and J. P. Ahrens. Parallel peak pruning for scalable SMP contour tree computation. In *6th Symposium on LDAV*. IEEE, 2016.

[2] G. Eitel-Amor, R. Örlü, P. Schlatter, and O. Flores. Hairpin vortices in turbulent boundary layers. *Physics of Fluids*, 27(2), 2015.

[3] C. J. Keylock. The Schur decomposition of the velocity gradient tensor for turbulent flows. *Journal of Fluid Mechanics*, 848, 2018.

[4] D. Morozov and G. Weber. Distributed merge trees. In *SIGPLAN Notices*, vol. 48. ACM, 2013.

[5] D. Morozov and G. H. Weber. Distributed contour trees. In *Topological Methods in Data Analysis and Visualization III*. Springer, 2014.

[6] H. Saikia and T. Weinkauf. Global feature tracking and similarity estimation in time-dependent scalar fields. In *Computer Graphics Forum*, vol. 36. Wiley Online Library, 2017.

[7] P. Vincent, F. Witherden, B. Vermeire, J. S. Park, and A. Iyer. Towards green aviation with Python at petascale. In *International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2016.

[8] J. Weatheritt and R. Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *Journal of Computational Physics*, 325, 2016.

[9] F. D. Witherden, A. M. Farrington, and P. E. Vincent. PyFR: An open source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach. *Computer Physics Communications*, 185(11), 2014.