Work in Progress: Vortex Detection and Visualization for Design of Micro Air Vehicles and Turbomachinery

Rhonda J. Vickery, High Performance Technologies, Inc., rvickery@hpti.com
Thomas Wischgoll, Wright State University, thomas.wischgoll@wright.edu
Chris Koehler, Wright State University, koehler.11@wright.edu
Matthew Pickett, Wright State University, acrolith_@hotmail.com
Neal Eikenberry, Wright State University, neal62689@gmail.com
Lance Harris, Central State University, l.harris313@gmail.com
Richard Snyder, AFRL/RBSD, Richard.Snyder@wpafb.af.mil
Darius Sanders, AFRL/RZTF, Darius.Sanders@wpafb.af.mil
Haibo Dong, Wright State University, haibo.dong@wright.edu
Randall Hand, Lockheed Martin IS&GS, Hand, randall.e.hand@usace.army.mil
Hugh Thornburg, High Performance Technologies, Inc., hthornburg@hpti.com

Abstract

Vortex detection and visualization is an important technique for CFD modelers and analysts. Since vortices are often not just local phenomena, algorithms for detecting the vortex core can be expanded by the use of streamline placement and termination methodologies to appropriately visualize the vortex. We are enhancing an existing VCDetect software tool for vortex detection to include new algorithms applicable to small scale micro air vehicles (MAVs), improve the interface, and integrate it with DoD Visualization Toolkit (VTK) based production tools. The code is being updated to include the latest parallelization features for efficient usage. The current VCDetect VTK-based code was developed with an XML file-based interface and was partially parallelized with multi-threading. The code was tested with a few common examples and some turbomachinery data test cases. In this work we are integrating newly developed visualization and vortex detection algorithms. An improved interface is added to allow the tool to be used both interactively and to easily set up multiple batch runs. Since this interface is expected to run on the user's local desktop, we are securing the communications using a previously developed direct through ssh methodology. We are also further parallelizing the code using the Compute Unified Device Architecture (CUDA), and we are applying it to a new problem domain in the area of MAV design. Finally this code will be made available to more DoD users by coordinating with the High Performance Computing Modernization Program (HPCMP) Data Analysis and Assessment Centers (DAACs) to include it in the Computational Science Environment (CSE) suite of production tools and libraries. We describe the extent of our work to date, and provide information on our path forward to completion of this project by 31 August 2011.

Introduction

This work enhances an existing VCDetect software tool for vortex detection to include new algorithms applicable to small scale micro air vehicles (MAVs), improve the interface, and integrate it with DoD VTK-based production tools. The code is being updated to include optimizations for execution on heterogeneous CPU/GPU architectures. We are applying it to a new problem domain in the area of MAV design, and are integrating newly developed visualization and vortex detection algorithms as described herein. The code will be made available to DoD users by coordinating with the High Performance Computing Modernization Program (HPCMP) Data Analysis and Assessment Centers (DAACs) to include it in the Computational Science Environment (CSE) production tool suite. The next section describes our methodology, including our motivation for the problem domain and application design. We also include some preliminary results in our analysis of dragonfly data described in (Koehler et al., 2011). Finally we close with some conclusions and ideas for future work.

Background

In many CFD applications, we are interested in studying the vortices within the simulated flow to gain insight into the data at hand. This is also the case when investigating the wings of insects to derive better wing designs for MAVs that are more optimal with respect to flight characteristics as well as energy efficiency. In this particular case, the most important features of the flow field are the leading edge vortices (LEVs) on each insect wing. The wide variation of experimental results surrounding research into how insects use the LEV to generate lift suggests that there is a lack of understanding about how it forms and behaves as the insects wings flap. With the most recent deformable wing simulations it is also important to study the relationship between how the wings deform and how the LEV is produced. The trailing edge vortex (TEV) is also potentially utilized by insects for certain maneuvers, such as rotation, takeoff and landing. Hence, flow visualizations that highlight the relationship between deformable wing kinematics and vortex (LEV and TEV) production are required.

When visualizing the flow to study the deformable wing of an insect, the focus should be on the LEV and TEV. Hence, the visualization algorithm has to extract these features from the flow data and then visualize the flow properties in their vicinity. Typically, stream lines or stream surfaces are used to depict such flow properties. Due to the high complexity of the flow in the vicinity of the vortices as a result of the highly turbulent flow characteristics in those areas, stream lines are often chosen over stream surfaces since the latter tend to overlap on themselves significantly in such complex flows. Even with stream lines, special care has to be taken in seeding those stream lines and terminating them in such a way that ensures a visualization that is easy to comprehend by the user, yet emphasizes all necessary features of the flow. This is one of the major advantages of the proposed approach over existing techniques. Automatic seeding strategies are utilized based on the flow features, i.e. the leading edge vortex and the trailing edge vortex, and stream line termination keeps overlapping within the visualization at a minimum, while properly visualizing the important characteristics of the flow.

Methodology

In the initial stages of this project, we planned to integrate recent work in the area of vortex detection and visualization into a tool that could be transitioned into the DoD HPCMP production environment. This involved enhancing a Visualization Toolkit (VTK) based tool developed with an XML file-based interface that was partially parallelized with multi-threading (van der Zwaag et al., 2009). The code was tested with a few common examples and some turbomachinery data test cases. This interface is useful for running multiple test cases in the batch environment, but becomes tedious to use when setting up a large number of test cases covering many parameter options. Although VCDetect had two vortex detection algorithms: an Eigenvector method developed by Sujidi and Haimes and the Parallel Vectors method developed by Roth and Peikert, often the vortices of interest are difficult to discern, and other vortex methods were needed (Jiang et al., 2004). The Lambda2 method developed by Jeong and Hussain had been implemented and integrated with a visualization technique known as “vortex tubes” and the code was obtained to be implemented with VCDetect (Hand et al., 2006). Additionally, a new streamline placement visualization technique has been recently developed in conjunction with recent research into the deformable dragonfly wing kinematics which provided the data for this work, including photogrammetry setup and parametric surfaces used to digitally reconstruct a useful dragonfly model (Koehler et al., 2011). Integrating these new features into an easy to use tool for vortex detection is the thrust of this work.

A high-fidelity in-house direct numerical simulation tool was used to simulate the flow induced by the flapping dragonfly wings. It is based on an existing Navier-Stokes immersed-boundary solver (Mittal et al., 2008), which is capable of simulating flows with complex moving boundaries including solid and membranous objects on stationary Cartesian grids (Dong et al., 2006). The solver employs a non-dissipative, 2nd-order, central-difference scheme, which is critical for accurately predicting vortices.

Using global metrics such as vorticity or helicity for streamline placement can be computationally intense when dealing with large vector fields and many time steps. Koehler et al. developed an interactive streamline seeding algorithm which alleviates this problem by placing seed points based on the evolution of objects immersed in the flow field rather than metrics taken from the flow field itself. An example using this method is shown in Figure 1. Streamlines are particularly effective at visualizing vortices in insect flight and micro air vehicle simulation data due to the easily identifiable helix shape. However, streamline seed placement is challenging in highly unsteady 3D flapping flight data. Uniform coverage of the entire flow domain produces very busy results.

Experiments have been executed with both seed generation and rendering techniques to more accurately display the key vortices without having them be occluded by less important features of the flow (Koehler et al., 2011). This is particularly important in order to study formation and movement of the LEV as well as the shedding of the wing tip vortices and TEVs. Initial seed placement experiments were done based on the vorticity magnitude in the area surrounding the wing leading edges. Targeting specific vorticity ranges allows the visual capturing of different portions of the vortices that form on the leading edge of each wing during the upstroke and downstroke. Experiments were also done with semi transparent streamlines. Making streamlines whose seed points are above a certain vorticity magnitude more transparent allows users to visualize vortices with less self occlusion of the streamlines. Figure 2 shows the result of one of the prototypes where this criteria was applied.

Figure 1. Streamlines colored by vorticity magnitude were generated using the dragonfly wings leading edge as seed points.

Figure 2. Streamlines distributed based on vorticity criteria.

We recently started transitioning the VCDetect tool into a ParaView plugin that can be deployed as part of the CSE (Squillacote, 2008, Renteria and Mark, 2008). One advantage of using this approach is the ability to implement a standard interface with only a few lines of XML. ParaView also offers a large number of readers, including Tecplot and CGNS readers needed for this project.

We have applied the VCDetect tool to automatically find corelines, and core surfaces that show the vortex locations and structure for the underside of a dragonfly wing (see Figure 3a and b). We have also used ParaView to create streamlines using the corelines as a starting point to highlight vortices which could be overlooked during initial analysis (Figure 4). These results may help provide additional insight into the flow characteristics, and therefore MAV design that mimics this type of flight.

Figure 3. Automatically generated vortex corelines (a), surfaces (b).

Figure 4. ParaView generated streamlines using corelines as a starting point colored by velocity magnitude.

Summary and Future Work

Currently we are in the process of integrating several features into a VCDetect plugin for ParaView to be included in the CSE. We have also applied this technology to turbomachinery data and expect that it will be applicable to other CFD datasets.

In recent years the architectural design of software has changed to include the proliferation of GPU's for computation. This provides exciting opportunities to gain significant speedup in applications with data parallelism. The previously developed VCDetect code, though multi-threaded, did not take advantage of this new approach. As shown in Figure 3, VCDetect locates vortex cores and generates surfaces defining the vortices of interest. Currently this takes 10 minutes or more per test case on a high end workstation with 8 CPUs, and so becomes very time consuming when multiple parameters and vortex algorithms need to be applied. In this project we have several workstations with NVIDIA Tesla cards for development, as well as access to a new GPU based state-of-the-art mini-supercomputer. The availability of such hardware gives us the development platform needed to further parallelize the code for heterogeneous CPU/GPU architectures. We plan to compare the results of our optimized code with our previous implementation. Although using CUDA or OpenCL are logical choices for this, there are other parallel streaming technologies that we are also considering (Vo et al., 2010).

We take advantage of the expertise of a multi-disciplinary team to provide value to all phases of the project. We are working jointly with our DoD partners to prioritize the desired features through a living requirements document in order to maximize our time and resources. We are using the Agile software development approach to iteratively deliver and refine the software to keep on track with evolving customer requirements while receiving valuable feedback for improvement. This approach maximizes team productivity and allows for modification during the project lifetime. The only real technical risk is inherent in the field of vortex detection; a chosen algorithm may not find the vortices of interest in the particular data set. Hence we mitigate that risk by including multiple algorithms in an infrastructure that makes it easy to add others. Since our code is VTK based, adding it to other DoD production visualization suites is straightforward.

Acknowledgments

This work was funded by the HPCMP under the User Productivity, Enhancement, Technology Transfer and Training (PETTT) preplanned effort PP-ACE-KY02-013-P3, GSA contract GS04T09DBC0017.

References

Dong, H., R. Mittal and F.M. Najjar, "Wake topology and hydrodynamic performance of low aspect-ratio flapping foils," Journal of Fluid Mechanics, vol. 566, pp. 309-343, November, 2006.

Dong, H., C. Koehler, Z. Liang, H. Wan, and Z. Gaston, “An Integrated Analysis of a Dragonfly in Free Flight,” AIAA 2010.-4390, 28 June – 1 July 2010, Chicago, IL.

Hand, R., and P. Adams, “Visualization of Time Dependant, Quasi Streamwise Vortex Tubes in a Bubble-Laden Turbulent Boundary Layer over a Flat Plate,” 2006 DoD High Performance Computing Modernization Program Users Group Conference.

Jiang, M., R. Machiraju, and D. Thompson, “Detection and Visualization of Vortices,” in C. Johnson and C. Hansen, editors, *Visualization Handbook,* Academic Press, 2004.

Koehler, C., T. Wischgoll, H. Dong, and Z. Gaston, "Vortex Visualization in Ultra Low Reynolds Number Insect Flight," Under Review by IEEE Transactions on Visualization and Computer Graphics.

Mittal, R., H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas and A. von Loebbecke, "A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries," Journal of Computational Physics, vol. 227, pp. 4825-4852, May, 2008.

Renteria, J.C, and E.R. Mark, “The Computational Science Environment (CSE),” 2008 DoD High Performance Computing Modernization Program Users Group Conference, 2008, pp. 382-386.

Roth, M., and R. Peikert, “A Higher-Order Method for Finding Vortex Core Lines,” IEEE Visualization '98, pp. 143-150, October 1998.

Schroeder, W., K. Martin, and B. Lorensen, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, *4th Edition*, Kitware, Inc., 2006.

Squillacote, A.H., *The ParaView Guide*, *Version 3*, Kitware, Inc., 2008.

Sujudi, R., and R. Haimes. “Identification of Swirling Flow in 3D Vector Fields,” AIAA 12th Computational Fluid Dynamics Conference, pp. 95-1715, June 1995.

Van der Zwaag, J., R.J. Vickery, and R. Moorhead, "Vortex Detection Through the Visualization Toolkit," 2009 DoD High Performance Computing Modernization Program Users Group Conference, 2009, pp. 395-397.

Vo, H., D. Osmari, B. Summa, J. Comba, V. Pascucci and C. Silva, “Streaming-Enabled Parallel Dataflow Architecture for Multicore Systems,” to appear, IEEE/VGTC EuroVis 2010.