Open Problems in Computational Steering of Massive Parallel Unstructured Grid Based CFD Simulations

Christian Wagner[1]

1 German Aerospace Center
Brunswig, Germany
christian.wagner@dlr.de
and
University of Kaiserslautern
Kaiserslautern, Germany
wagner@cs.uni-kl.de

Abstract

Traditionally, analysis of flow fields resulting from computational fluid dynamics (CFD) calculations is a sequential process. The flow area defined by surrounding geometry is tessellated, a mesh is generated and divided into subregions, transferred to a cluster or supercomputer and the result is transferred back. Then, a variety of post-processing tasks should give insights to the physical problem. At that point, parameters chosen wrong can be identified and the simulation has to be done again with tweaked parameters. This is an iterative process that can be time consuming, especially if one iteration lasts more than a few days. In general, aiming at reducing the simulation times by shortening the time used to identify wrong parameters results in high productivity enhancements.

In this paper, the need for on-line monitoring and computational steering approaches for massive parallel unstructured flow simulators are presented with aircraft design as one of many possible application domains. This involves software integration aspects, data streaming and explorative visualization. Many challenges still have to be solved and this paper summarizes most important ones.

Keywords and phrases Computational Steering, CFD simulation, Interactive Visualization, Explorative Visualization, Virtual Reality

Digital Object Identifier 10.4230/OASIcs.VLUDS.2010.82

1 Introduction

Aircraft design has many challenges in the near future forcing the design to heavily rely on numerical simulations. The number of expected passengers in 2020 is as triple as high as nowadays number and the price per passenger has to be reduced by half. Additionally, aviation has to deal with governmental restrictions like massive reductions in emissions and noise levels.

These requests need technological leaps and cannot be achieved through continuous improvements of traditional techniques like the continuous improvements of wing shapes. New aircraft configurations have to be discovered and it becomes necessary to account for the whole flow field around an aircraft. The aim in numerical CFD research is to enable aircraft simulations over its entire flight.

To support engineers in the development of advanced designs, CFD as the main future design tool also needs to provide high usability. The nowadays very sequential CFD work-flow

Visualization of Large and Unstructured Data Sets– IRTG Workshop, 2010.
Editors: Ariane Middel, Inga Scheler, Hans Hagen; pp. 82–89
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is therefore the topic of this paper. Traditionally, CAD design, mesh generation, simulation and post-processing are separate tasks, most of them have effort of more than a few days. This sequential pattern has the effect that scientific results as well as errors are recognizable at very late stages and decreases the efficiency of the whole work-flow.

The following paper is structured as follows. In the next section we will discuss how CFD work-flows can be enhanced with computational steering methods. Then, we will cite work related to the stated CFD work-flow. After that, we will discuss the open problems of software integration, data streaming and explorative visualization needed to be solved followed by a conclusion section.

2 CFD Optimizations

Computational Fluid Dynamics became the most important tool to give researchers insight into complex flow structures. Different strategies can be used to support results more quickly. On the one hand, the CFD simulation itself can be improved by using faster algorithms and hardware acceleration. On the other hand, the overall work-flow can be restructured to enable higher responsiveness for simulation systems. For the latter approach, this chapter will discuss the benefits of computational steering capabilities.

2.1 Computational Steering to optimize CFD work-flows

The traditional work-flow used in computational fluid dynamics is very sequential. The tasks of setting up the flow conditions, the simulation and the analysis of the results are strongly separated. If some parameters were set inappropriately they have to be tuned and the work-flow starts again from its beginning. Since the simulation itself can last for a few days or longer, the process is full of very long waiting periods.

Especially in production processes long waiting periods have to be avoided. Different approaches to overcome these problems are still in development. In-situ visualization approaches try to move most of the analytic calculations into the simulation that have to be done in the post-processing otherwise. Besides the challenge to bring visualization algorithms to the same scaling as the simulation code, sophisticated knowledge about the expected results are needed.

Another approach is the usage of computational steering techniques. Computational Steering normally comes in combination with on-line monitoring and visualization of the ongoing simulation, giving the ability to evaluate the actual solver runs. Interactive steering then allows to tweak simulation parameters and guide the running simulation. The resulting fast feedback gives a lot of potential to the researchers. Having the possibility to change simulation parameters on the fly shortcuts the traditional sequential work-flow, because inappropriate parameters can be identified much earlier before the post-processing task and can be tuned immediately instead of starting at the setup task again. Also, being capable of fixing issues in the underlying simulation mesh like multi-block decompositions or mesh refinement levels can guide the simulation to a faster convergence. Last but not least, through getting an immediate visual feedback the researcher gains additional insights in the simulated effects.

Even in the setting of attaching a computational steering system to an ongoing simulation, a fast and effective visualization is needed for the analysis of not well-known physical phenomena. Explorative visualization has proven to support researchers many times in the past. But the high amount of data in physically correct computational fluid dynamics still challenges data management as well as visualization algorithms.

Some of the open problems needed to solve to make such a steerable system available are presented in section 4.

3 Related work

To support the computational steering approach described in the last section we are introducing a very general solution that future research will aim on, as described in section 4.1. This section gives an overview of work with close relation to at least one aspect to our whole system view.

The idea to use visualization guided computational steering systems is not new. [5] describes a framework to develop steered algorithms, but focuses more on the combination of algorithms. To allow for a better visual responsiveness [6] introduces a parallel pipeline-driven front-end. A more CFD-specific system can be found in [8]. An overview of the different approach of in-situ visualization can be found in [12].

The benefits of virtual environments for the analysis of flow phenomena are discussed in [11]. [18] introduces a parallel back-end to support post-processing in virtual environments. These two articles also discuss the advantages of interactive and explorative post-processing.

In the field of progressive streaming a lot of work has already been done. [17] shows how to use space-filling curves to reorganize data on regular grids and give them a multi-resolution meaning that can easily be streamed. For unstructured meshes the additional problem with indexing schemes from cells to nodes arises. [7] therefore introduces an interleaved streaming file format that is used in [13] to enable for streaming iso-surfaces. These streaming formats are used for online monitoring in [16] for a simulation on regular grids and in [3] for progressive volume rendering of unstructured grids.

The addressed problems of computational fluid dynamics systems in this paper are guided by the TAU flow solver used and developed by the German Aerospace Center [20]. [19] shows how this system was wrapped into a python-scripting interface.

4 Open Problems

As described in section 2.1, design processes relying on CFD simulations can benefit a lot from computational steering capabilities. In this section, we will summarize the work-flow with computational steering capabilities as an aimed solution in section 4.1. After that, we will discuss arising problems. We will describe software architecture problems in section 4.2, communication problems in section 4.3, and finally visualization issues in section 4.4.

4.1 Aimed solution

The aimed solution proposed for CFD work-flows can be found in figure 1. Instead of iterating through the time-consuming traditional work-flow again and again until an appropriate solution is found setup and output should be necessary only once.

After the computation of the simulation is started it can be observed via online monitoring. For this purpose, data from the simulation is transferred to the visualization system and the running simulation can be evaluated immediately. To help the viewer in unknown situations, explorative visualization techniques will be at hand to find the interesting features in the simulation. In the field of computational fluid dynamics this can be various flow specific visualization techniques like the interactive seeding of stream lines and stream surfaces as well as interactive iso-value selection techniques for the scalar fields in the simulation. If inappropriate parameters or meshing is found the researcher can change them on the fly

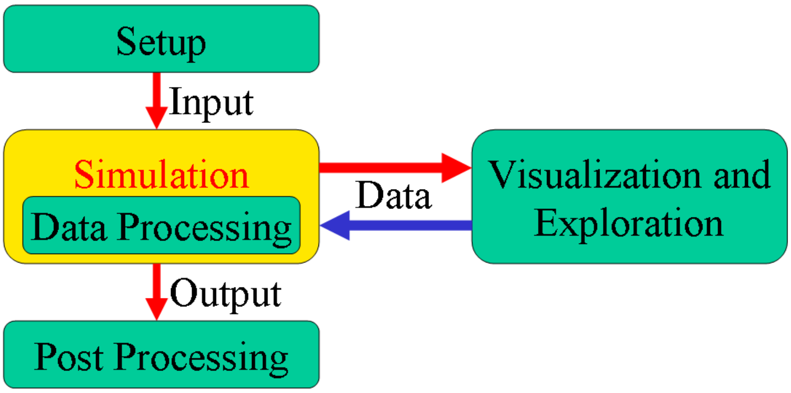

Figure 1 Aimed solution. A running simulation can be observed by an On-line Monitoring visualization. Explorative visualization guides the viewer through unknown behavior. For the determination of visualization features causing high CPU load a Data Processing module assists the visualization module. Since the simulation can be guided to appropriate solutions, setup and has to be done only once. Finally, the one-time written output has to go through additional post processing steps only once.

and the simulation will continue with the new values. Since the analysis in the traditional post-processing task involves normally also higher order visualization techniques, e.g. vortex region and vortex core line extraction, these should also be possible. For that reason, a module supporting these tasks should be combined in the system.

For the realization of such a system a long way is still to go. As shown in the last section, a lot of the problems are already solved or at least got research attention. But there are still many open questions. Some of them are described in the following sections.

4.2 Software Architecture

To enable CFD simulations for computational steering a system architecture with supporting functionalities has to be developed.

Traditional simulation systems were built as a collection of executables concerning different steps of the simulation. Therefore, the simulation was done by calling these programs in a way suitable for the simulation purpose, each of them reading input data from the file system and writing output data for the following one. This results in a huge file I/O overhead, that is nowadays canceled out by wrapping the functionalities in a scripting wrapper, the Python language proved to be effective, and keeping the data in memory. As an additional advantage for the researcher, the growing amount of functionalities can be combined in a wide and fine-granular variety enabling new and optimized simulation settings.

A computational steering system has to take account of the manifold functionalities of these simulation scripts while not constraining the researcher in the freedom of writing simulation scripts. Therefore, well-designed and flexible interfaces are required. The question how to build these interfaces between simulation, computational steering system and visualization is still unsolved. Some interfaces to bring data out of the simulation are used in existing on-line monitoring systems. Mainly copying raw geometry and field data, they are not very flexible and the simulation has to deal with the on-line monitoring data format. The way back into the simulation is even harder and a computational steering system has to know all the parameters that can be changed.

It is not very likely that a common interface standard can be established, however, application domain-specific computational steering standards could eventually evolve like the CGNS file format did for saving fluid dynamic data.

In the existing computational steering approaches concurrency between the simulation and the visualization system was addressed seldom. Normally, the data is copied between every iteration step to the visualization as well as changed parameters.

Especially, when the visualization becomes more distant to the simulation system, as described in section 4.3, this becomes an unwanted scenario. If the time to transmit data becomes much longer than the time to calculate one iteration step concurrent transmission and visualization of the data is very reasonable for not slowing down the simulation while monitoring a running simulation. However, from computational steering point of view the changes into the simulation have to be serialized or the simulation might have to get back to the time step the viewer is actually seeing.

4.3 Communication

The system architecture needed for large CFD simulations is running distributed on large supercomputers or cluster systems and communication is therefore inherent in computational steering. For on-line monitoring data has to be transferred to the visualization host. In the first computational steering systems the visualization host was located nearby the simulation host and the bandwidth compared to the amount of data was high.

This changes in the usual setting of computational fluid dynamic simulations. In figure 2 the computational environment often found in research institutes is depicted. On the one hand, you can find a simulation cluster or supercomputer with a very high bandwidth and low latency intercommunication network. On the other hand, for visualization purposes a different system is used, sometimes as an additional small visualization cluster system with a virtual reality system. Normally, these two systems are connected by only a local area network with a rather low bandwidth and high latency. Additionally, the computation systems mostly have only one or a few login nodes with high firewall protections that can be used to transfer data. The separation becomes even much stronger if distant visualization is used in a collaborative system supporting a few visualizations at very distant locations.

The usage of such a system causes auxiliary trouble for computational steering systems. The concurrency issues where already discussed in section 4.2. Through domain decomposition techniques additional blocking of the data is introduced that should be handled without slowing down the simulation by using a traditional transfer through the simulations master node.

To deal with low bandwidth, progressive streaming approaches were already introduced years ago. This streaming technique is to reorganizing data to show results from the very beginning of data transmissions. First, a lower quality preview is transmitted providing quick overviews. Further data is then gradually increasing details. For data on structured grids this is a solved problem for most applications and for data on unstructured grids some cases out of the variety are already addressed.

For the usual kind of computational fluid dynamics systems shown in section 4.3 new challenges come into play. View-dependent progressive streaming algorithms have to take into account the additional domain decomposition that should not be handled by loading the simulations master node.

In order to additionally minimize the necessary data to transfer, domain specific algorithms should be introduced handling the conditions of computational fluid dynamics situations. For unsteady simulations grid adaptations are often required to support complex flow features

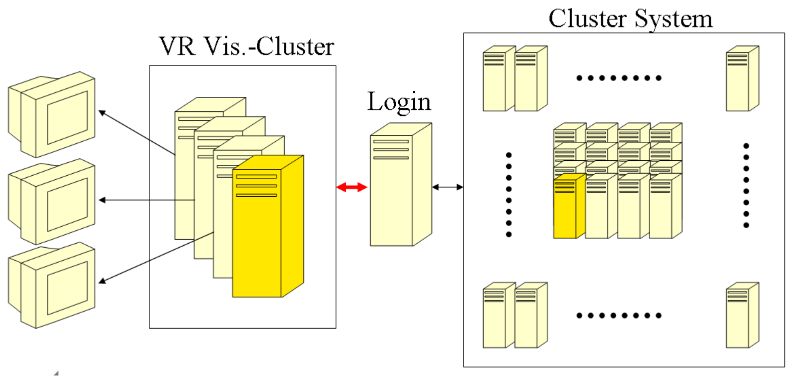

Figure 2 Usual computational system in CFD environments. Because of their scaling performance, a cluster system is often used on the computational side. The visualization is located at a distant place, here a virtual reality system with its own visualization cluster is depicted. These two systems are mostly connected through a local area network with a low bandwidth resulting in an additional bottleneck.

changing only portions of the underlying simulation grid. Transferring incrementally only the updated portions results in less data to handle in a progressive streaming. The same is true for resulting data on top of the grid that has to be updated after each simulation iteration step.

4.4 Visualization

Finally, a complete computational steering environment needs to support adequate visualization capabilities. For the analysis of unknown flow structures explorative visualization techniques have shown to support the researcher in a very effective way. The existence of real time feature extraction algorithms is required. For flow fields arising from CFD solutions stream lines are suitable to guide the researcher through the flow field and algorithms to extract them in real time are already at hand. Especially in complex flow situations stream surfaces provide better visual clues, but porting to graphics processing units to provide interactive extraction techniques has just started.

A system to determine complex features causing high CPU load should be introduced to assist the explorative visualization on-demand. But the best system architecture is unclear. Two solutions are possible. On the one hand, an additional visualization cluster can be attached to a running simulation. Data then has to be copied from the simulation to the visualization, resulting visualization features are copied to the visualization front-end. On the other hand, feature extraction can be done on the simulation nodes, this approach is called in-situ visualization. This approach promises more accurate results, since the full amount of data can be accessed. But the extraction algorithms fight for computation time with the running simulation and have to scale as well as the simulation does.

Another unsolved problem is how to provide the graphics processing units with all the data needed for the computation. Since the resulting flow fields are way to large for the GPUs memory, adapted data has to be streamed over the connecting bus system.

5 Conclusion

In this paper we had a look at the traditional work-flow for computational fluid dynamics systems. Even if the evolution of CFD system by adding capabilities of scripting and coupling with many other solvers transforms them to a more general toolbox, it is hard to use their full potential. This is caused by long waiting periods and manually iterated tasks as a result of lacking compatible steering frameworks.

Therefore we stated an idealized work-flow consisting of a one-time data load at startup that has the potential to also write the results only once. This can be achieved by tuning all parameters and guiding the simulation to an appropriate result without the bottlenecks of the traditional CFD work-flow.

To realize the idealized work-flow we determined three main research fields. First, flexible software architectures need to be developed that are capable of handling the variety of application scenarios and with interfaces flexible enough to support future applications and hardware architectures. Second, communication between subsystems will get more important in the future. With increasing supercomputer sizes their availability becomes more centralized and therefore distant analyzes and visualization becomes more and more common. Progressive streaming techniques are promising to handle bandwidth bottlenecks and provide better responsiveness. And thirdly, enhanced visualization techniques are required to support engineers with analyzes techniques suitable to their application domain. For efficient feature extraction, these algorithms need to scale as well as the hardware they are running on.

Nevertheless, although research is going on now for over forty years, we have to keep in mind that also computational fluid dynamic is far away from reaching its end. If the performance of nowadays simulation systems can be increased by a factor of at least 1000, scientists might start to think of modeling and simulating most aircraft design situations.

That is why computational steering systems still need to be flexible and application dependent in order to support efficient work-flows.

References

1 Allen Bierbaum, Christopher Just, Patrick Hartling, Kevin Meinert, Albert Baker, and Carolina Cruz-Neira. Vr juggler: A virtual platform for virtual reality application development. In *Virtual Reality Conference, IEEE*, 1:89–96, Los Alamitos, CA, USA, 2001. IEEE Computer Society.

2 Steve Bryson. Virtual reality in scientific visualization. *Commun. ACM*, 39(5):62–71, 1996.

3 Steven P. Callahan, Louis Bavoil, Valerio Pascucci, and Claudio T. Silva. Progressive volume rendering of large unstructured grids. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1307–1314, 2006.

4 Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, and Rosemary M. Simpson. Immersive vr for scientific visualization: A progress report. *IEEE Comput. Graph. Appl.*, 20(6):26–52, 2000.

5 J. Davison de St. Germain, John McCorquodale, Steven G. Parker, and Christopher R. Johnson. Uintah: A massively parallel problem solving environment. *High-Performance Distributed Computing, International Symposium on*, 0:33, 2000.

6 Aurelien Esnard, Nicolas Richart, and Olivier Coulaud. A steering environment for online parallel visualization of legacy parallel simulations. In *DS-RT '06: Proceedings of the 10th IEEE international symposium on Distributed Simulation and Real-Time Applications*, pages 7–14, Washington, DC, USA, 2006. IEEE Computer Society.

7 Martin Isenburg, and Peter Lindstrom. Streaming meshes. In *Proceedings of Visualization'05*, pages 231–238, October 2005.

8 O. Kreylos, A. M. Tesdall, B. Hamann, J. K. Hunter, and K. I. Joy. Interactive visualization and steering of cfd simulations. In S. Müller and W. Stürzlinger, editors, *VISSYM '02: Proceedings of the symposium on Data Visualisation 2002*, pages 25–34, Aire-la-Ville, Switzerland, 2002. Eurographics Association.

9 Oliver Kreylos, Tony Bernardin, Magali I. Billen, Eric S. Cowgill, Ryan D. Gold, Bernd Hamann, Margarete Jadamec, Louise Kellogg, Oliver G. Staadt, and Dawn Y. Sumner. Enabling scientic worksflows in virtual reality. In *Proceedings of the ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications VRCIA 2006*, pages 155 – 162. ACM SIGGRAPH, ACM SIGGRAPH, 2006.

10 Oliver Kreylos, E. Wes Bethel, Terry J. Ligocki, and Bernd Hamann. Virtual-reality based interactive exploration of multiresolution data. *Hierarchical and Geometrical Methods in Scientific Visualization*, 2003.

11 T. Kuhlen, A. Gerndt, I. Assenmacher, B. Hentschel, M. Schirski, M. Wolter, and C. Bischof. Analysis of flow phenomena in virtual environments - benefits, challenges, and solutions. In *Proceedings of the 11th International Conference on Human Computer Interaction*, 2005.

12 Kwan-Liu Ma. In situ visualization at extreme scale: Challenges and opportunities. *IEEE Computer Graphics and Applications*, 29:14–19, 2009.

13 Ajith Mascarenhas, Martin Isenburg, Valerio Pascucci, and Jack Snoeyink. Encoding volumetric grids for streaming isosurface extraction. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 665–672, Washington, DC, USA, 2004. IEEE Computer Society.

14 Tom Meyer and Al Globus. Direct manipulation of isosurfaces and cutting planes in virtual environments. Technical report, NAS Systems Division, NASA Ames Research, 1993.

15 Jurriaan D. Mulder, Jarke van Wijk, and Robert Van Liere. A survey of computational steering environments. *Future Generation Computer Systems*, 13, 1998.

16 V. Pascucci, L. Linsen, B. Hamann, and F. Gygi. Real-time monitoring of large scientific simulations. In *In ACM Symposium on Applied Computing'03, ACM*, pages 194–198. Press, 2003.

17 Valerio Pascucci and Randall J. Frank. Global static indexing for real-time exploration of very large regular grids. In *Supercomputing '01: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)*, pages 2–2, New York, NY, USA, 2001. ACM.

18 M. Schirski, A. Gerndt, T. van Reimersdahl, T. Kuhlen, P. Adomeit, O. Lang, S. Pischinger, and C. Bischof. Vista flowlib - framework for interactive visualization and exploration of unsteady flows in virtual environments. In *EGVE '03: Proceedings of the workshop on Virtual environments 2003*, pages 77–85, New York, NY, USA, 2003. ACM.

19 Andreas Schütte, Gunnar Einarsson, Axel Raichle, Britta Schöning, Matthias Orlt, Jens Neumann, Jürgen Arnold, Wulf Mönnich, and Thomas Forkert. Numerical simulation of maneuvering aircraft by aerodynamic, flight mechanics and structural mechanics coupling. In *AIAA-Journal of Aircraft*, 2007.

20 Dieter Schwamborn, Thomas Gerhold, and Ralf Heinrich. The dlr tau-code: Recent applications in research and industry. In *ECCOMAS CFD 2006*, The Netherlands, 2006. TU Delft.