8th. World Congress on Computational Mechanics (WCCM8)
5th. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 – July 5, 2008
Venice, Italy

Large and heterogeneous data handling for helicopter CFD

Marc Poinot[1], Michel Costes[2]

[1] ONERA DSNA
BP 72, 92322 Châtillon

[2] ONERA DAAP
BP 72, 92322 Meudon

{poinot,costes}@onera.fr

Key Words: *CFD, Numerical simulation, CGNS.*

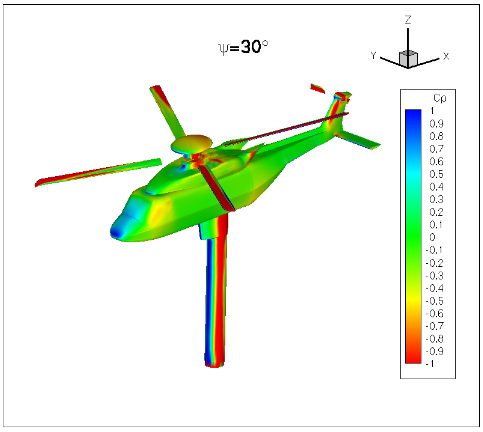
ABSTRACT

Context

The CFD projects at ONERA are involving several codes on super-computers, they use and produce very large amount of data. In the workflow of these CFD simulations, the data are heterogeneous, time-dependent and most of the time generated during a parallel computation. The usual file based storage can be achieved by means of a local and precise procedure giving rules for directories, file names, contents of files... but the features provided by CGNS [1] offer a more formal, documented, interoperable and actually operational way of handling these data. We illustrate such a use through the EU GOAHEAD project in which the applied aerodynamics department of ONERA had to compute a complete helicopter configuration with a set of chimera grids provided by an external partner using a different CFD solver [2].

Standardized complex data

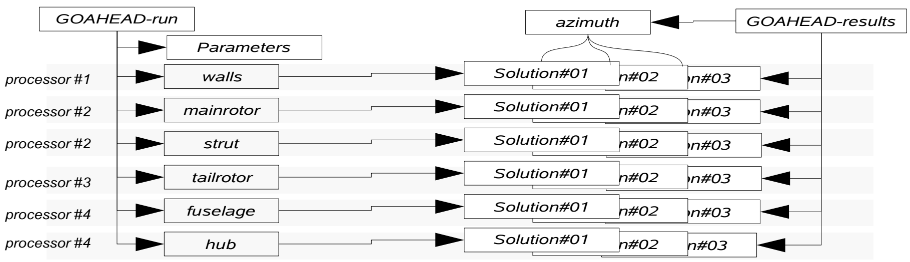
The GOAHEAD application includes a high speed CFD simulation of a complete NH90-like helicopter model in the DNW-LLF wind tunnel, including main and tail rotors, strut, fuselage, hub and wind tunnel walls. The computational tools have to be interoperable and eventually re-used for other cases. Thus, the open standard for data management is a strong requirement. Also the very large amount of unsteady data could not be handled without the help of a data model: we have to handle traceability between grids, computation parameters, restart data, results... The grids were defined and provided by DLR and were not to be modified by the application. Some data changes had nevertheless to be introduced in these grids. For example, the



different grids have to be moved in their expected location for each computation run, the assembly of files in a global CGNS tree led to name collision. All modifications are performed as decorations, this means the actual data is unchanged. The decoration pre-processing needs not to be aware of the configuration, number of zones, zone names, boundary conditions and all other information. Thanks to the public standard it discovers the required information during the files parse. The chimera technique being used to compute the motion of the different blades with respect to the fuselage and the wind-tunnel, the masking, blanking, neighbourhood search is also performed by pre-processing tools using *pyCGNS* [3]. The tools are looking for zones, their neighbourhood and sets the correct parameters and boundary conditions for the elsA fluid solver [4]. Therefore, a new release of DLR grids is taken into account by parsing again all the information of the CGNS files, no other external data is needed.

Computation facts

The computation is unsteady and runs on a linux cluster after a distribution of the 145 structured zones using a compliant balancing tool. The CGNS data model is perfectly suited for a parallel computation. When a zone (i.e. block, domain) is separated from the global CGNS tree and sent to a specific processor, it comes with all the solver required data, including connectivities, BCs, solutions... Each run is producing about 1,5Gb of data per degree of rotor azimuth. The computation is restarted every 30 degrees, and thus a post processing has to be performed to build a new restart tree using the solution fields from the previous run: we have to make a loop over solutions becoming the initialization of the next step. We use the CGNS link feature to create an empty skeleton. This skeleton represents the results we would have at the end of one step. When the fluid solver saves the data on disk, one file is saved per processor and the structure of this file is the one expected by the skeleton. Thus, without any other post processing, the user can run visualization tools on the skeleton file now filled. Moreover, the actual solution data is shared between tools through these links, no file duplication is done even if the different pre/post processing tools need different views of the same data.
The data, their relationship and their dependency against the time are shown in the following schema:



Conclusion

We succeeded in managing large and heterogeneous data using a CGNS data model, including when complex CFD techniques such as chimera are used. This standard data allows interoperability by means of a parsing and understanding of a self-descriptive set of files. An actual GOAHEAD computation has been performed and lessons have been drawn for more interoperability in forthcoming international projects.

REFERENCES

[1] "The CGNS System," D. Poirier, S. R. Allmaras, D. R. McCarthy, M. F. Smith, F. Y. Enomoto (AIAA Paper 98-3007).

[2] “The blind-test activity of the GOAHEAD project”, O. Boelens et al, 33rd European Rotorcraft Forum, Kazan, September 2007.

[3] `http://elsa.onera.fr/pyCGNS`

[4] “The status of elsA CFD software for flow simulation and multi-disciplinary applications.” L.Cambier, J.-P. Veuillot (AIAA-2008-0664).