11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20 - 25, 2014, Barcelona, Spain

FLUID STRUCTURE INTERACTION WITH INCONSISTENT SOFTWARE PLATFORMS

David Grasselt*, Prof. Dr.-Ing. Klaus Höschler and Aris Konstantinidis

Institute of Traffic Engineering, Brandenburg University of Technology Cottbus-Senftenberg (BTU),
Siemens-Halske-Ring 14, 03046 Cottbus, Germany,
David.Grasselt@tu-cottbus.de, http://www.tu-cottbus.de/flugtriebwerksdesign

KEY WORDS: *Fluid Structure Interaction (FSI), Multiphysics Problems, Applications, Computing Methods, inconsistent Software Platforms*

INTRODUCTION

The FSI problem class describes the mutual dependence of the multiphysical interaction between fluid dynamic forces and structural mechanic deformation. The FSI effects become more significant and may even influence partially the safety analysis when the dependence between the influence and response becomes stronger, e.g. the pumping of blood by the ventricles of the human heart or the fluttering of aero-engine blades. [1]

For mechanical engineers the design of a product is influenced by the fluid flow related forces that act onto the structural components. For a fluid dynamics engineer the design is mostly influenced by structural components controlling or transiently changing the flow field. When the solutions of both disciplines mutually depend on each other, a FSI simulation process chain must be considered. A robust simulation process chain is the basis for profound investigations. A kind of investigation could be the aerodynamic design of a race car or the structural mechanic construction of an umbrella. Both interact with structural and fluid-dynamical values but both require a different approach. Applications where bidirectional FSI – coupling of fluid flow analysis to structural analysis and vice versa – has to be applied could for example be a self-regulating valve or nozzle.

MOTIVATION

The modelling of fluid dynamic problems requires an approach different from those, which is relevant for structural mechanic issues. The coupling of modern numerical methods and tools enables the analysis of mutual dependencies. Although software companies develop more and more integrated solutions, inconsistent application of software solutions (SW) play an important role for several reasons like:

a) end-user availability,
b) business case and
c) certification reasons.

In some cases it is not possible for users who deal with FSI problems to develop on integrated solutions, especially open-source-solver users have to take an inconsistent approach to reach a fully coupled FSI solution. Another reason could be that the computational engineer has no access to integrated tools because investment in new software solutions which support such strategies is not supported. Finally, some kind of constructions need an approved engineering process and approved software solution approaches to obtain a legal certificate. To hold a specific certified combination of approved tools it can be advantageous to combine inconsistent tools to enable automated analysis approaches. The advantages can e.g. be:

a) the reduction in time effort for such routine tasks,
b) the multiple execution of similar jobs for an optimisation and
c) the reduction of errors due to human interaction.

Nowadays, some tools seem to be better suited for a specific application, so that the user has a preference on a specific tool for mechanical issues and on another for fluid dynamic issues. Additionally, the accuracy of several computational fluid dynamic (CFD) as well as structure mechanical finite element method (FEM) tools are not benchmarked against each other, like it is typical for computer hardware components. It is getting even harder to compare the quality of integrated software solutions. All this reveals that a free combination of software is desirable. This can only be achieved when Fluid-Structure Interaction is realised with inconsistent software platforms.

The paper describes a coupling approach, in the case of obligatory use of inconsistent SW for the separate, mutual depending challenges of a FSI problem description, especially for large displacement applications.

METHODOLOGY

General

The most popular FSI approaches base on the slight coordinate variation of the spatial discretisation of a geometrical model, a typical approach for vibration investigations.
For flow pattern influencing changes driven by a varying geometric boundary condition, the displacements, which have to be gained from the structural analysis, have to be much larger than typically known in above meant approaches.
This is an important fact which emphasises the need for an approach which aims at the reconstruction of a CAD model after having obtained the finite element analysis results.
An example of very small displacements could be the behaviour of a windsock, where the displacements of the solid body are negligible small and do not influence the much larger scaled flow pattern. A counterexample where the displacements do influence the flow field very strongly could be a tilt window in a storm which closes and reopens alternating.

Figure 1 shows the applicable approach presented in this paper, where the problem definition with its boundary conditions and constraints is solved by creating a FE and a CFD CAD sub-model from a parental model. The structural analysis receives information from the problem definition itself, which defines the geometric constraints and boundary conditions, as well as temperature or pressure loads from the flow analysis.

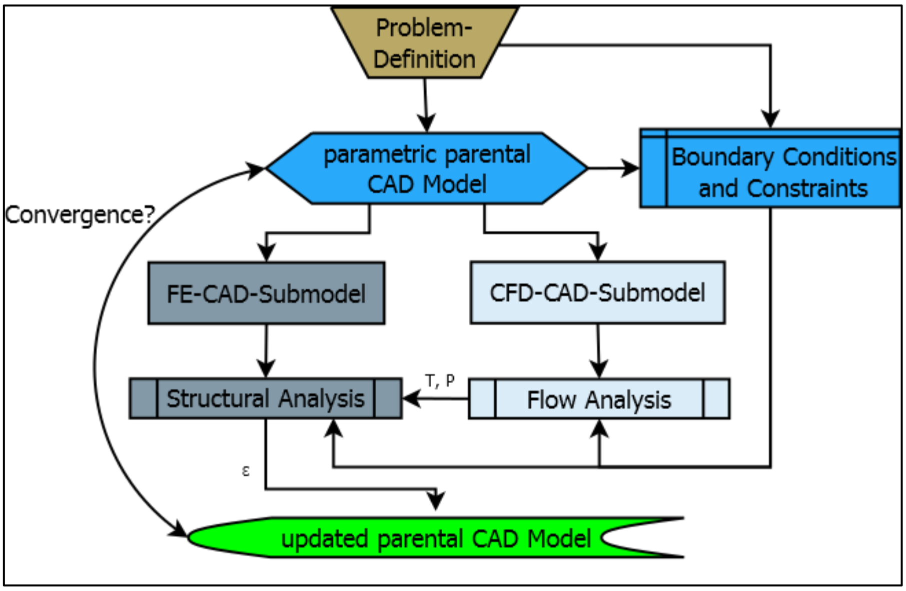

Figure 1 General FSI Proces Structure

The stress results from the structural analysis can be used to evaluate the construction itself, the displacements will be utilised to update the parental model. When the changes of displacements are below a certain limit, the FSI process chain has converged.

Unfortunately, direct interfaces between the different tools are typically not supported. Only a few combinations can be found, for example between Siemens NX and Ansys Fluent. The most widespread approach is to use universal formats as data exchange platforms between different tools. But these formats, e.g. STEP, STL or Parasolid, sometimes do not support the transfer of all required information. This makes a well-considered choice even more important when dealing with such interfaces.

CAD & CFD

The first step of the developed FSI process chain is the solution of a fluid dynamic problem, which is then followed by a coupled FE Analysis. The fluid dynamic problem has to be subdivided into the derivation of an aero-mechanical CAD model and the CFD problem setup, solution and evaluation. The aero-mechanical CAD model is derived from a main model which is the basis for both, CFD and FE investigations. The presented approach uses different software platforms for

a) CAD setup and manipulation,
b) CFD grid generation and
c) Boundary Condition (BC) application, solver setup, solution and result extraction.

The option of an automated CAD manipulation opens the opportunity for preparation of multiple similar cases and is based on a parameterised CAD description. By (de-) activating the relevant features, it is possible to choose between aero- and structure-mechanical items. The export to a compatible format (IGES: international graphics exchange standard) finally allows the transfer of, e.g., surface labels.

These Labels have to be available to allow the detection during the grid generation sub-step. The kind of discretisation scheme (tri, quad, tets, hexa, pyramids …) on surfaces or in volumes near a surface has to be set depending on surface type (inlets/outlets, walls …). Depending on the kind of investigation and the number of models to evaluate, the discretisation strategy has to be robust, uniform and quick. Depending on the time-effort which is planned to or can be spend for the solution of a, sometimes very complex, problem description, it is important to focus the objective of such an analysis: detecting tendencies or absolute magnitudes. This should have an influence on the grid modelling depth. Additionally, specific geometrical features must be detected to guarantee a relevant resolution. This can also be managed by coupling the flow solver with the grid generator, which is supported for only few inconsistent combinations, e.g. Ansys Fluent and Centaur.
The interface between grid generator and CFD solver can again be a data file. Typically the file-format of the solver or a grid file can be set by the grid generator as export format. The approach to create similar, comparable grids can be, to transfer all the settings of grid A to grid B or just to pass through the grid generation process again identically as done before. The modelling of BC's, the solver setup and the result extraction is case dependent. Some platforms support the scripting of this sub-step, which enables comparable cases again. During the setup of a case the identification and separate treatment of specific faces is important and can be supported by surface labelling again. Finally, the FEA relevant outputs and the magnitudes which describe the aerodynamic quality have to be gathered. While single magnitudes can be extracted relatively easy, the extraction of FEA BC from a CFD Analysis is more difficult due to the fact that the target-format has to be determined by a solution is an intersection of both, the CFD and the FE processor: the exported pressure or temperature distribution on a surface has to be in an applicable format for the FEA software platform.

Transition from CFD to FEA

Along with the regular case dependent FEA setup via scripting, there are FSI relevant procedures which have to be introduced to get a coupled solution. The CFD supported FEA BCs are typically given in context of the fluid domain surfaces' grid point distribution, which makes it more complex to transfer to the FEA BC information. Usually CFD and FEA Grid points are not coincident, even the numbers of grid points for the FE analysis and for CFD analyses are not the same. This leads to the situation that some grid points may be coincident and others are not, see the principle sketch of layered Grids in Figure 2.

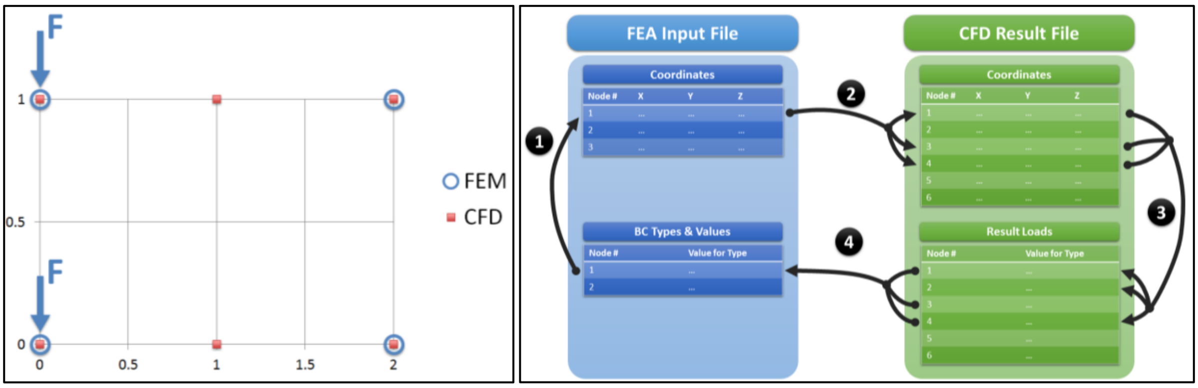

Figure 2 Layered Grids for CFD and FEA (left) and Triangulation (right)

Triangulation of CFD grid points in the near of a FEA grid point can help to increase the accuracy of the BCs' distribution. After analysing the FEA input file for grid points with a predefined uniform dummy load and corresponding node number in step (1) the second step is to compare the FEA grid coordinates determined in the first step against the coordinates of the CFD result file to find three of the nearest CFD grid points. Step (3) looks for the related CFD result values of the three coordinates evaluated in step (2). In step (4) the arithmetic average of these three result values will be written to the FEA input file related entry of step (1). This will be done for each of the entries existing in the input file.

The start of the FE solution process has to be divided into three steps, to transfer the results from the CFD to the FE solver BC input before the solver starts:

a) the pre-processing and generation of a finite-element-model input file,
b) the manipulation of the input file with an external code, and the required corresponding time and data management, as illustrated in Figure 2 and
c) the execution of the FEM solver with the manipulated input file.

Typical finite-element-codes generate an input file and process this file with a dedicated interface. This interface can be addressed directly by the application using a graphical user interface (GUI) button. Alternatively, the solver is not executed immediately, but a file is created which can be manipulated by hand. Subsequently, the manipulated input file can be processed by the solver immediately. Apart from that, the manipulated input file can also be processed using an interface which the GUI normally uses but is not seen by the user. Most software platforms utilise a modularised, object-oriented approach which leads to the point that an interface is supported in the installation directories of that software. Another option is that the software allows executing older or hand-edited input files through internal routines.
A self-coded Java SW can use one of these interfaces to manipulate the structural solvers' input file, Figure 3.

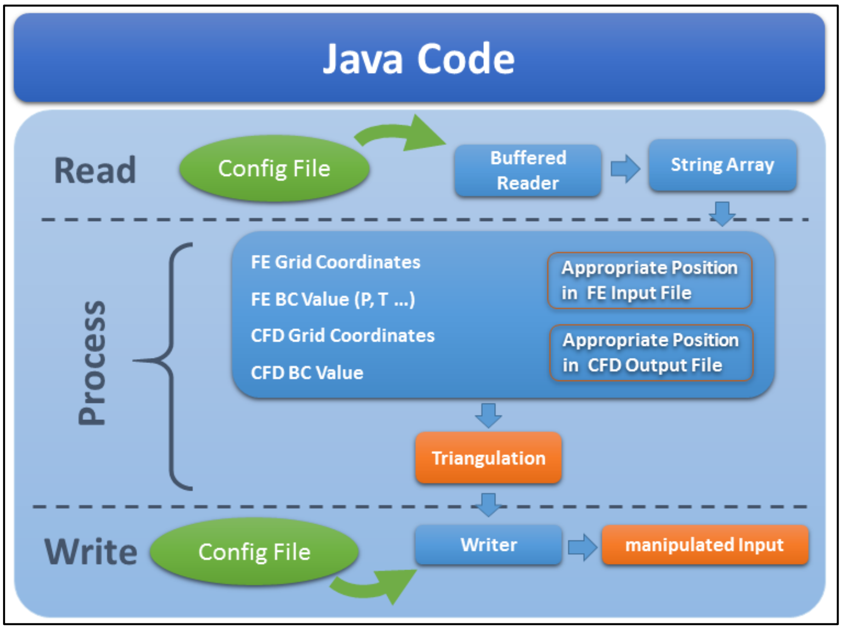

Figure 3 Coupling of CFD Results to FEA Boundary Condition

This additionally requires access to CFD solution data, which can be obtained from a CFD post-processor in form of text files or likewise as described above. Furthermore the access to the Java Runtime Environment and a compiler enables the transmission of high-resolution load distributions. The final submission of the manipulated input file to the FE solver can be realised with direct interfaces (installation directory links) or in-solution coding, but this might require access to the FE Code itself and a special license type for interaction with the source code of the FE solution.

Results of a FEA

The outcome of the FE solution can be divided into two types of information. The most obvious is numerical output in form of stress or displacement results which e.g. can be compared against material properties. This output can be used to evaluate whether the component is still functional or structural unimpaired. The information can be obtained by SW tools given or by self-coded functions. These may be capable to present exactly the information in a format which is required.

Additional FSI process relevant information is required to allow the coupling of FEA to the CFD analysis: boundary conditions, especially the surface pressure provoked deformation of the component of interest. To compute a deformed model with inconsistent platforms it is necessary to access simulation data (nodal displacements) on the one side and the grid topology (nodal coordinates) on the other side. These information have to be utilized to transform the base CAD model by a technique which creates one or multiple new surfaces on the basis of existing loaded surfaces. The generated CAD surface has to be processed thereafter to create a new valid parental or at least CFD CAD model. Here, an approach of importing a deformed surface tessellation language (STL) described surface which then can be used to match the former model. A key method is the description of such surfaces using non-uniform rational B-splines [2].

The coupling of the 2nd CFD-FEA iteration becomes more complex, due to the fact that the load information for the non-deformed FE-model has to be recovered from the deformed CFD-CAD-model of the 1st iterations' FEA deflection, see Figure 4.

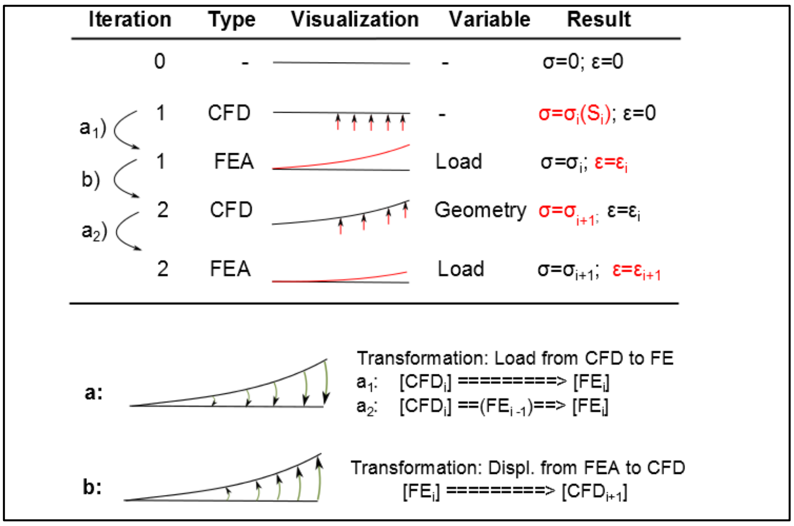

Figure 4 FSI Parameter-Exchange

The load extraction of the first CFD result file can be used directly for the following structural analysis due to the fact that this CFD model is based on the same CAD model which is used for the FE-analysis. The second and every following iteration of the CFD CAD model is influenced by the displacements of the previously executed structural analysis' displacement results, while the FEA CAD model has to be the original. This means that the matching of CFD and FEA grid points has to recognise the additional displacements as illustrated.

Automation

The automation of interface processing, FEA and CFD solution leads to significant exploitation potential for example for design-optimisation applications and design evaluation with target solver-, Monte Carlo-, six sigma-, Taguchi, stochastic design improvement methods or is capable to simply accelerate the design process.

For automated process chains the relevance of approaches which work non-interactive is high, when the process chain is going to be applied on a server. Another reason to work non-interactive is that the hardware system can be unloaded by processes which are not reasonable necessary. Automated process chains should therefore be mainly driven and controlled by script and not by human interaction. To create an approach which can be used for other FSI applications, again it is required to strictly separate method code and application dependent information. The discussed FSI method uses scripting exclusively, most often it is a kind of in-solution coding which also opens wide interaction capabilities and extended handling opportunities, when compared against the methods available for GUI users.

RESULTS

Effort in automation of all subtasks, implemented between and into inconsistent software platforms, including the manipulation of CAD-models, CFD discretization & solution through macros, journaling or other tool depending utilities, lead to strategies for individual SW combinations. This opens a wide advantageous access to several, often required or preferred independent software platforms and the benefits of rapid analysing of thousands of designs by high performance computing (HPC) exploitation. The presented methods claim a non-neglectable one-off expense in method modelling that can be worth when the constituted assumptions of the motivation chapter get significant.

REFERENCES

[1] H.-J. Bungartz, A.H. van Zuijlen and H.Bijl; Multi-Level Accelerated Sub-Iterations for Fluid-Structure Interaction, *Fluid Structure Interaction II*; p. 2, Springer 2007.
[2] Y. Bazilevs, K. Takizawa and T.E. Tezduyar; *Wiley Series in Computational Mechanics*; John Wiley & Sons Ltd 2013.
[3] J.H. Ferziger and M.Peric; *Computational Methods for Fluid Dynamics*; 3rd rev. ed. Springer 2002
[4] O.C. Zienkiewicz and R.C. Taylor, *The finite element method*, 6th Edition, Elsevier, 2005.