Optimized working process in a practice-oriented CFD project

Richard Markl-Gološ

CAE Simulation & Solutions GmbH, Vienna, Austria

Summary:

The present lecture shows an optimized and practicable working process in common CFD projects. With the used commercial and non-commercial software tools, especially for the pre-processing, the effective processing time for modelling could be significantly reduced based on applying a direct modeller, automated own-written scripts and a script-based meshing software. Afterwards with the obtained mesh the numerical simulation and the post-processing can be executed in a common used way. Experience has shown that this working process leads to lower working times and a lower error rate.

As a consequence, this lecture should give an understanding how to effectively simplify the commonly occurring huge CAD data from customers or other construction departments and finally end up in CFD results.

Keywords:

Optimized working process, CFD simulation, cfMesh, direct-modelling, ANSYS Spaceclaim, ANSYS Fluent

1 Introduction

Nowadays one of the biggest challenges of a computational engineer is on the one hand to derive from complex, extensive and detailed CAD data (on the part of the customer/construction) a clean and geometry which is appropriate for the simulation goals and on the other hand to create quickly and effectively a computational grid (numerical mesh). Additionally, there should be the possibility of transferring geometry changes that occur in the course of the project to the mesh and to integrate them into the existing working process easily. In practice, an efficient interaction between the customer requirement and the finally used mesh for the solver plays an important role.

The present lecture investigates the execution of an optimized and proven process in a CFD project. This process starts with the receiving of the geometry data, includes all relevant intermediate steps (geometry preparation, meshing, numerical simulation) and ends with the evaluation of the results or its documentation. Here, a combination of commercial and non-commercial (open-source, for free) products has proven particularly useful.

In summary this whole working procedure leads to a work time reduction, especially for the normally huge time-consuming pre-processing (deriving the relevant geometry and obtaining a numerical mesh) as well as license cost reduction.

2 Working Process

On the basis of a practice-oriented project it will be shown how the direct-modeller *ANSYS Spaceclaim* can be used to quickly simplify/derive (extensive) geometries and to prepare for the meshing software *cfMesh* with specific own-written scripts. The non-commercial (for free) meshing software *cfMesh* is then able to generate a computational grid automatically with scripts. This mesh is initially available in an *OpenFOAM* format. With the utilization of *OpenFOAM*-specific conversion commands the computational grid is translated into a mesh, which is readable for the numerical simulation software *ANSYS Fluent*. After obtaining results from *ANSYS Fluent* the postprocessing software *ANSYS CFD-Post* (or other Python written post-processing tools) can be used to visualize these results. All these steps are summarized in Fig. 1.

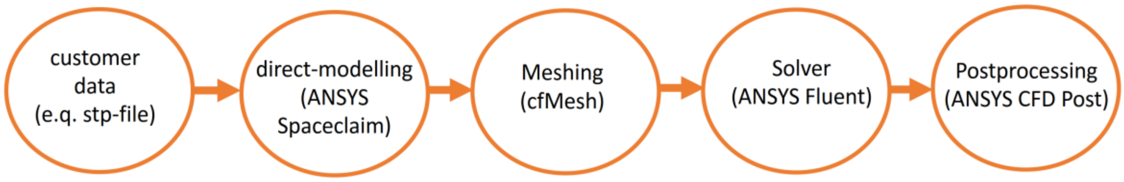

Fig. 1, working process

2.1 Geometry handling

Largely the geometry provided from the customer or from the construction department (internal or external) is very complex and overloaded with construction details which are not relevant for the flow simulation. So, there are two options:

- draw/construct a new geometry with a CAD-software or
- directly use some relevant parts of the existing construction to derive a practical geometry for the meshing software.

In the sum of the projects realized in our company we noticed that the second option (using the direct-modeller *ANSYS Spaceclaim* Ref. [3]) leads to quicker and more accurate CFD models with a lower error rate.

Moreover, it may occur during the project that some changes according to the customer's request should be included in the geometry. With this direct modeller is easy to add or to remove some parts quickly without destroying the dependencies in the geometry file in contract to parameterized CAD software. Nevertheless, *ANSYS Spaceclaim* contains the possibility to organize the geometry in an independent structure tree with respect to defining the boundary conditions.

ANSYS Spaceclaim is able to export triangulated detailed stl-files which is one of the options to transfer the geometry to the meshing software *cfMesh*.

Fig. 2 shows on the one hand the complex and overloaded geometry (provided raw data from the customer) and on the other hand the clean and for the CFD simulation relevant geometry.

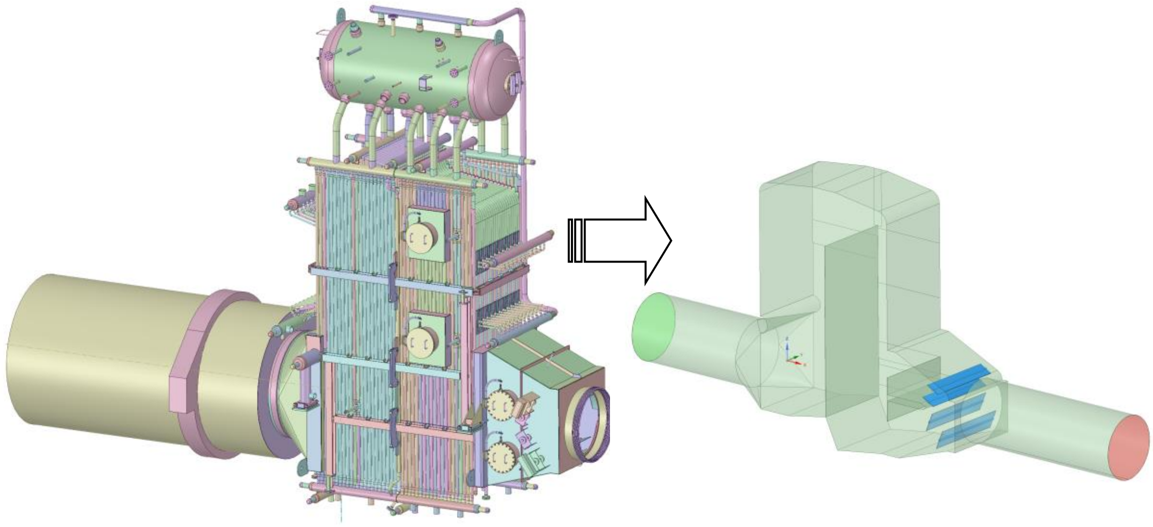

Fig. 2, model simplification by using the direct-modeller *ANSYS Spaceclaim* (lhs: original CAD Data, rhs: CAD data used for the mesh)

2.2 Generate a numerical mesh

The non-commercial meshing software *cfMesh* (Ref. [1]) is used to create a numerical mesh. The benefits of using this unstructured hex-dominant mesh generator are as follows:

- generates the mesh automated
- runs robust
- simple to use und to learn
- captures complex geometries
- tolerates small gaps or cracks
- runs on parallel processors
- no license cost

Before using this script-based software the geometry which comes from *ANSYS Spaceclaim* should be convert into a format readable for *cfMesh*. Basically, *ANSYS Spaceclaim* exports for each boundary condition (inlet, outlet, wall etc.) a single stl-file. Now these stl-files are manipulated and merged by own-written scripts und finally converted to a ftr-file, where all the geometrical informations for the boundary conditions are implemented. *cfMesh* has the ability to generate a numerical mesh script-based with different kinds of object (box, cone or individual geometry defined by a separate stl-file etc.) and local (e.q. surfaces or lines) refinements, and/or of course boundary layers. Furthermore, in some cases optionally anisotropic meshes are also useful and with *cfMesh* feasible.

Similar to *snappyHexMesh* (the standard meshing tool for *OpenFOAM*) *cfMesh* controls all the individual mesh requirements by using a file called *meshDict* (see Fig. 4). Basically, in this control file these following important parameters can be defined:

- geometry input file
- basic cell size
- definition of the boundary layers
- local and object refinements
- boundary renaming
- definition of an anisotropic mesh

ParaView (Ref. [4] and [5]) is a commonly used and non-commercial software to visualize and to check the created mesh. Moreover, the *OpenFOAM* specific command *checkMesh* is another powerful and efficient opportunity/tool to check the validity of the mesh.

Fig. 3 shows exemplary a mesh with a detailed view near the guide plates, where the refinements and the boundary layers are shown. *cfMesh* generates automatically an unstructured numerical mesh with a high quality boundary layer. To convey a sense of the time management of the meshing software *cfMesh* the shown mesh (about 2 700 00 cells) in Fig. 3 is generated in 80 seconds on a 8 core Linux machine (32 GB RAM).

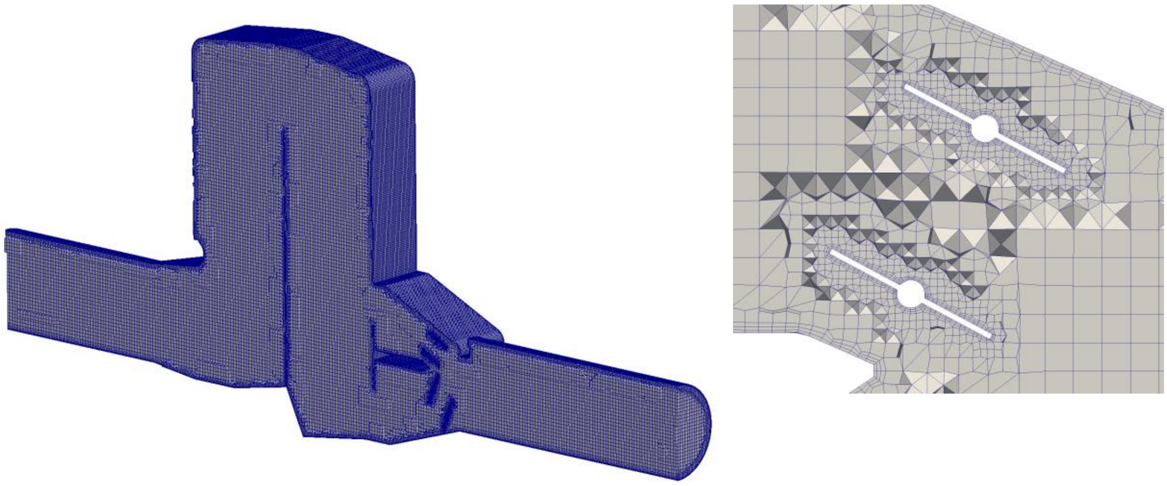

Fig. 3, Mesh generated with *cfMesh* (lhs: total model, rhs: detailed view)

Finally, *cfMesh* produces a mesh in an *OpenFOAM* format, which can be converted by using some *OpenFoam* specific conversation commands to an *ANSYS Fluent* readable mesh.

Eventually this script-based mesh generation with *cfMesh* is predestined for an automatic workflow, where changes in the geometry can be easily implemented.

As an example only a few parameters are required to get a mesh as depicted in Fig. 3:

```
surfaceFile "all.ftr"; // --> geometry file
maxCellSize 50e-3; // --> basic cell size
boundaryLayers
{
    nLayers 0;
    patchBoundaryLayers
    {
        "wall.*|leit.*"
        {
             nLayers    3;
             thicknessRatio 1.2;  // --> boundary layer definition
        }
    }
}
localRefinement
{
    "wall.*"
    {
        additionalRefinementLevels 1; // --> local refinement step based on the
        maxCellSize (e.q. 1 means maxCellSize/2)
    }
    "leit.*"
    {
        additionalRefinementLevels 2;
    }
}
```

Fig. 4, structure of the main control file *meshDict* (file extract, explanation of the parameters in green)

2.3 Using *ANSYS Fluent* (as a solver as well)

The numerical solver *ANSYS Fluent* (Ref. [2]) is able to import the *cfMesh* generated numerical mesh to do the numerical simulation and uses all the previously in the *meshdict* defined boundary condition names.

A huge benefit is that with even rough geometry data and the corresponding mesh it is possible to set up the case. In the progressing project status details in the geometry or other changes can be easily implemented by replacing a new mesh without further changes to the *ANSYS Fluent* case necessary.

It should be noticed if more than one zone is necessary for numerical simulation (e.g. multi-reference-frame for a rotor) *ANSYS fluent* is able to couple (append) different zones from *cfMesh* via interface (lhs in Fig. 5). Otherwise simple geometries (e.q. a box for a heat exchanger) can be picked and separated directly in *ANSYS fluent* (rhs in Fig. 5).

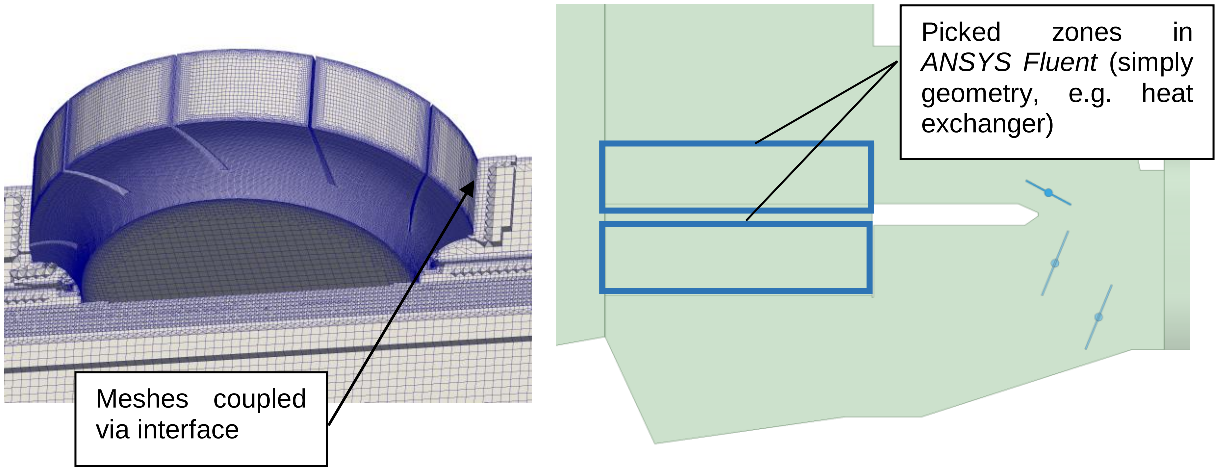

Fig. 5, lhs: 2 zones connected with an interface, rhs: picked Zones in *ANSYS Fluent*

2.4 Post-processing

Mainly the post-processing software *ANSYS CFD-Post* is applied to visualize the results obtained from *ANSYS Fluent*. This software is powerful to illustrate the flow by using streamlines, vectors, contour-plots or animations. In special cases own-written python scripts are used to document results as well.

3 Conclusion

Through this optimization of the working process the total cycle time of a CFD project compared to conventional procedures (e.q. ANSYS workbench) could be significantly reduced. This working process includes the following essential steps:

- Simplification of the geometry by using the software *ANSYS Spaceclaim*
- Script-based automated mesh generation by basically using the software *cfMesh* and own-written scripts
- Mesh manipulation with *ANSYS Fluent*

So the main difference to the commonly used methods so far is the geometry preparation in combination with an automated and robust meshing software.

Another major and decisive advantage is that the solver settings can be done parallel in time with the refinement or detail improvements of the numerical mesh. In addition, as part of the project progress, changes to the geometry or optimization measures are easy to implement.

With these optimizations it is possible to invest more time in thinking about a physical optimization or to put the focus on the postprocessing or the documentation.

4 References

[1] Juretić, F. "cfMesh User Guide v. 1.1.", Creative Fields, Ltd, 2015

[2] “ANSYS Fluent User Guide”, Ansys Inc., 2011

[3] “ANSYS Discovery SpaceClaim“ , http://help.spaceclaim.com/v19.0/en/ [Online, Accessed 1st October 2019], 2019

[4] Ahrens, J., Geveci, B., Law, C., “ParaView: An End-User Tool for Large Data Visualization”, Visualization Handbook, Elsevier, 2005

[5] Ayachit, U., “The ParaView Guide: A Parallel Visualization Application”, Kitware, 2015