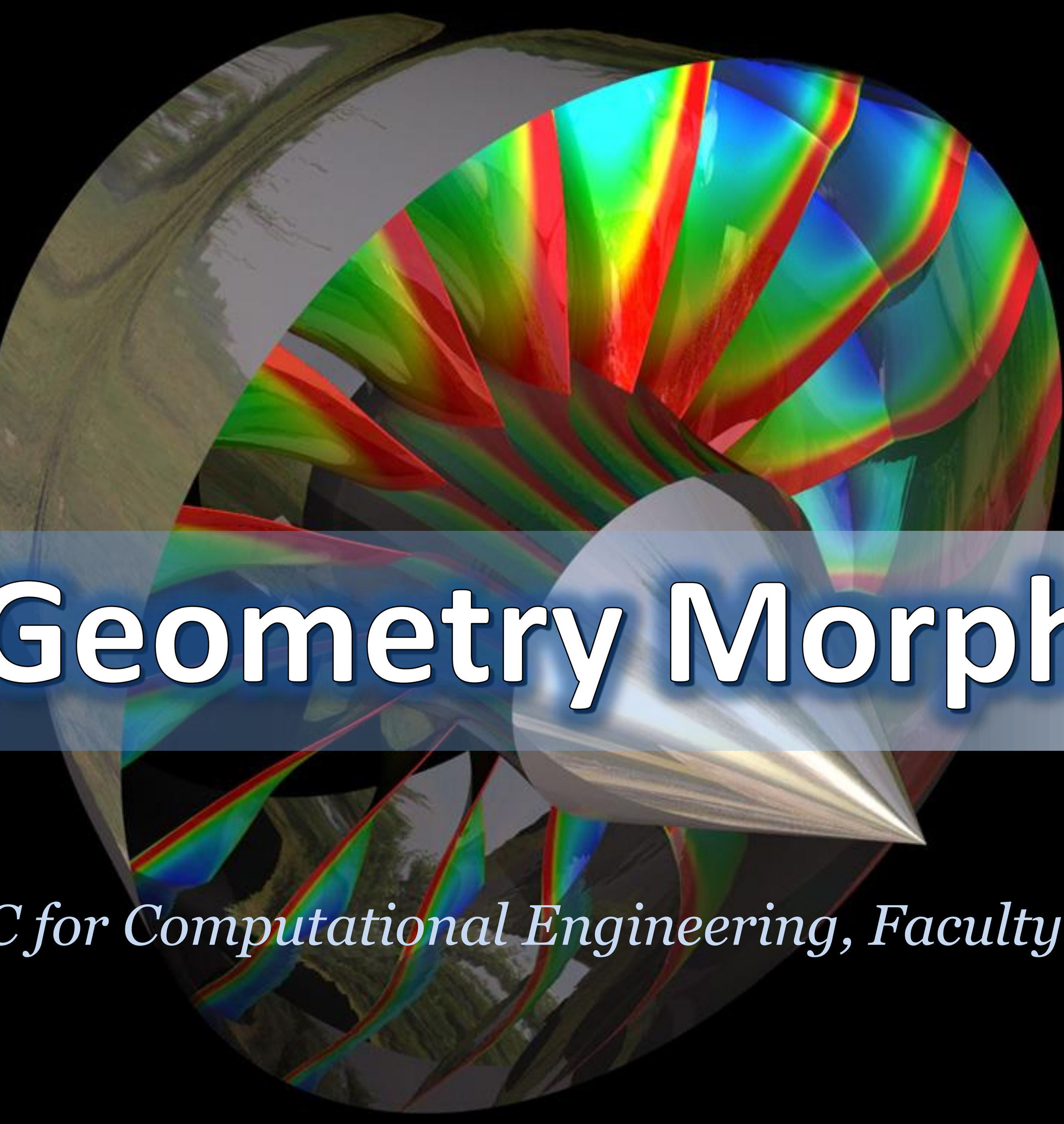

UNIVERSITY OF Southampton

Engineering and the Environment

Aeronautics, Astronautics and Computational Engineering

Geometry Morphing for Combustor Design

Xu Zhang, David Toal, Andy Keane

UTC for Computational Engineering, Faculty of Engineering and Physical Sciences, University of Southampton

Mike Caruso, Marco Zedda, Fred Witham

Rolls-Royce plc.

Project Overview

The aim of this project is to morph a Computational Fluid Dynamics (CFD) mesh by employing displacements calculated as a result of a Finite Element Analysis (FEA). As shown in Fig. 1, the workflow commences with a Computer Aided Design (CAD) model. From this, an FEA model is generated which includes a low fidelity representation of the CFD fluid volume with an extremely low elastic modulus. After applying all boundary conditions the FEA model is solved producing a set of the nodal displacements. The nodal displacements of the low fidelity CFD volume representation are then used to morph the much finer CFD fluid volume constructed by, for example, the Prometheus system.

Morphing Mesh *vs*. Morphing CAD?

There are a couple of reasons to use this morphing approach rather than manipulating the CAD model. As long as the coarse FEA and finer CFD meshes are generated in the first place, there is no need to revisit the CAD model - it could be skipped completely by applying this approach. Although this project is aimed at hot-to-cold or cold-to-hot combustor morphing, it can be applied to many other engine components like turbine or compressor blades. In the design processes of these components, the FEA models have to be solved anyway - why not use the FEA results directly?

Overview of the Tetrahedron Interpolation

Fig. 2 illustrates the overview of CFD node interpolation process. Firstly, both CFD and FEA meshes are parsed to memory (the FEA mesh considered in this project is a three-dimensional tetrahedron volume mesh). Then, the mesh mapping process creates a relationship map between CFD nodes and FEA tetrahedrons. The Barycentric coordinate of one CFD node will be calculated as well if it is inside an FEA tetrahedron. The next step morphs the CFD mesh by interpolating all CFD nodes using the Barycentric coordinate information. Finally, the new morphed CFD mesh will be written out.

Future Work

Potential applications of this approach include:

- Cold-to-hot/hot-to-cold geometry morphing of combustor system
- Rapid assessment of combustor concessions
- Geometry morphing based on measurement scans
- Robustness assessment of engine components and sub-systems

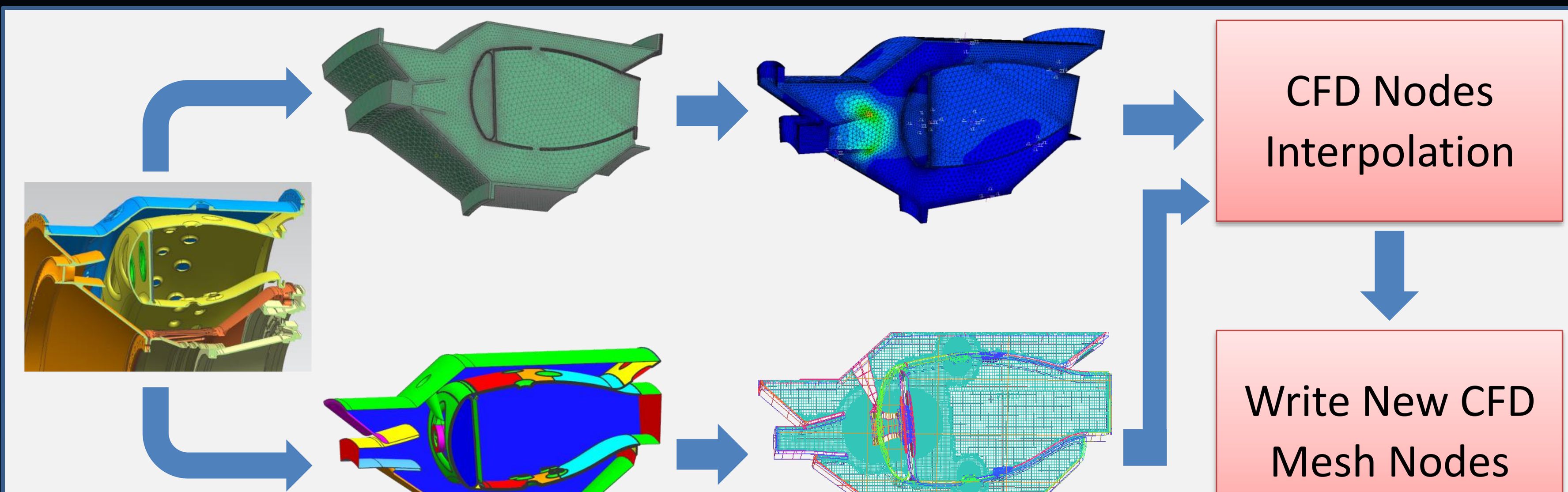

Fig. 1

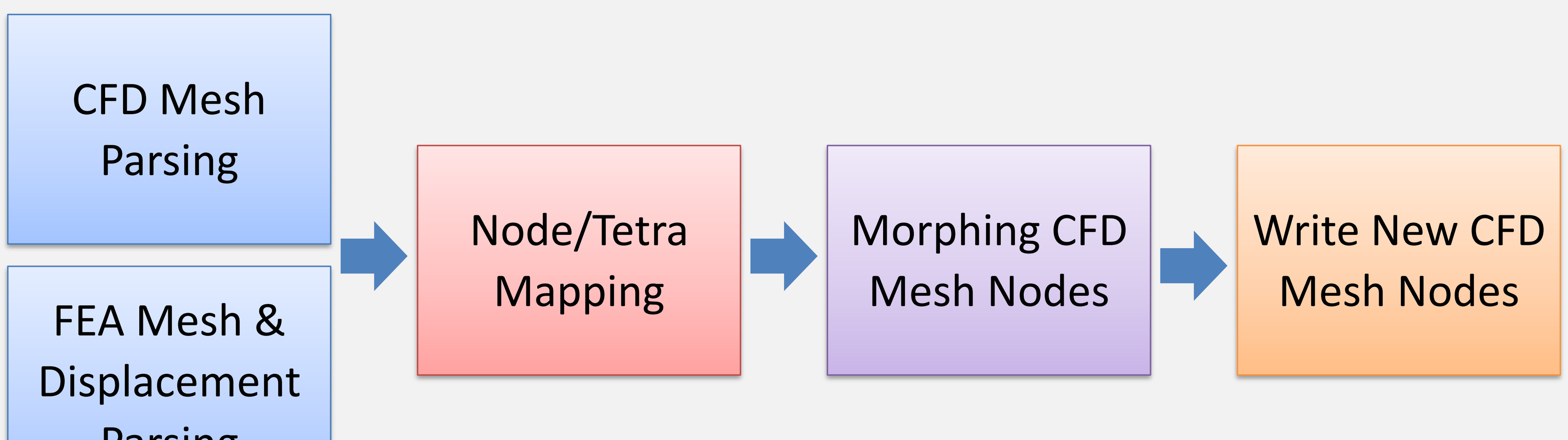

Fig. 2

Implementation and Performance

A prototype application has been implemented in Matlab. The performance has been tested against different sizes of the dataset generated using a generic combustor CAD model. An equivalent version has been implemented in C++ to the Input/Output (I/O) and CFD/FEA mesh mapping process by applying memory mapping and k-dimensional (k-d) tree approach respectively. As k-d tree archives the complexity of $\boldsymbol{O}(\log \boldsymbol{n})$ on average, the complexity of the morphing process has been reduced from $O(n^2)$ to $O(n \cdot \log n)$, which results in significant improvements on large CFD meshes. The comparison result is shown in Table 1. Including fluid volume in the FEA mesh certainly increases the number of elements. Depending on the complexity of the simulation, the solving time varies. However, it is not affected noticeably. Take the test case with over 18-million CFD nodes and more than 300-thousand FEA tetrahedrons shown in Fig. 1, SC03 simulations always finished within 5 minutes if the fluid domain is included or not. Compared to generating a new CFD mesh using ICEM, which can take a number of hours, the overall performance of this morphing approach speeds up the design process significantly.

CFD Nodes	SC03 Elements	Matlab Approach	C++ k-d Tree Approach
18,405,139	332,081	21,177.65 s ≈ 5.88 h	56.322 s

Table 2

This work is funded by GEMinIDS project

http://www.soton.ac.uk/engineering/research/groups/CED/posters.page | email: xu.zhang@soton.ac.uk

Computational Engineering & Design Group, University of Southampton, SO16 7QF, U.K.