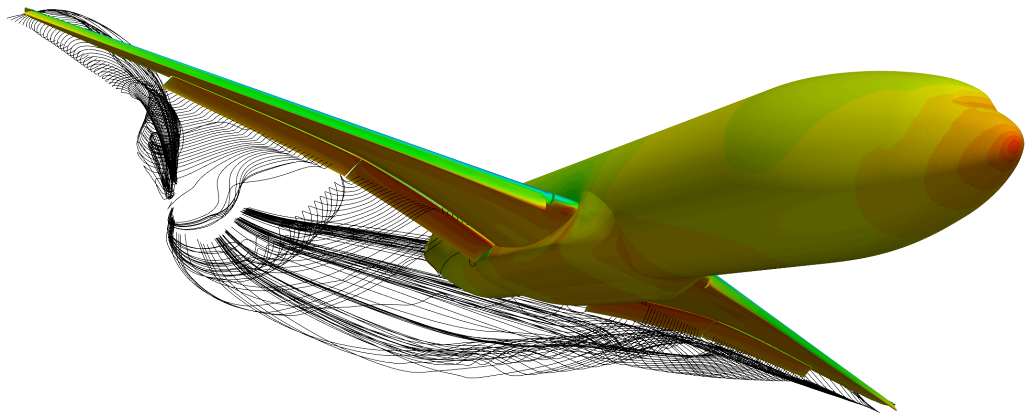


Google Summer of Code

Differentiated Meshing for CFD-based Design

Project Size: Medium (175 hours)

Contributor: Amartya K Prusty
Mentor: Dr. Nitish Anand

Stichting SU2
Faculty of Aerospace Engineering
Kluyverweg 1
Delft 2926HS
The Netherlands

1. About the Contributor

I'm a Masters student in Engineering Mechanics at KTH Royal Institute of Technology in Stockholm. I specialise in the fluid mechanics track in my studies, where I have done projects and courses on CFD theory and simulations. A lot of focus, in theory, was on meshing patterns having a significant impact on the consistency and convergence of the methods used. Thus, I understand the impact the topic chosen here can have in facilitating open-source software users and researchers. Some of the significant aspects in my portfolio include :

1. Two years of professional experience at Goldman Sachs developing ML, Statistical algorithms and SDLC of products (Python, Git, etc.)
2. Extensive competitive programming experience using C++ (CodeChef highest - 4 stars)
3. Theoretical and project experience in CFD using OpenFOAM/ANSYS and development proficiency in SU2

Here's my LinkedIn profile for a better overview!

One of my long-term goals is to resolve current intricacies in mechanical engineering problems using computational methodologies. Thus, this opportunity with SU2 and GSoC would be a perfect stepping stone for my career interests!

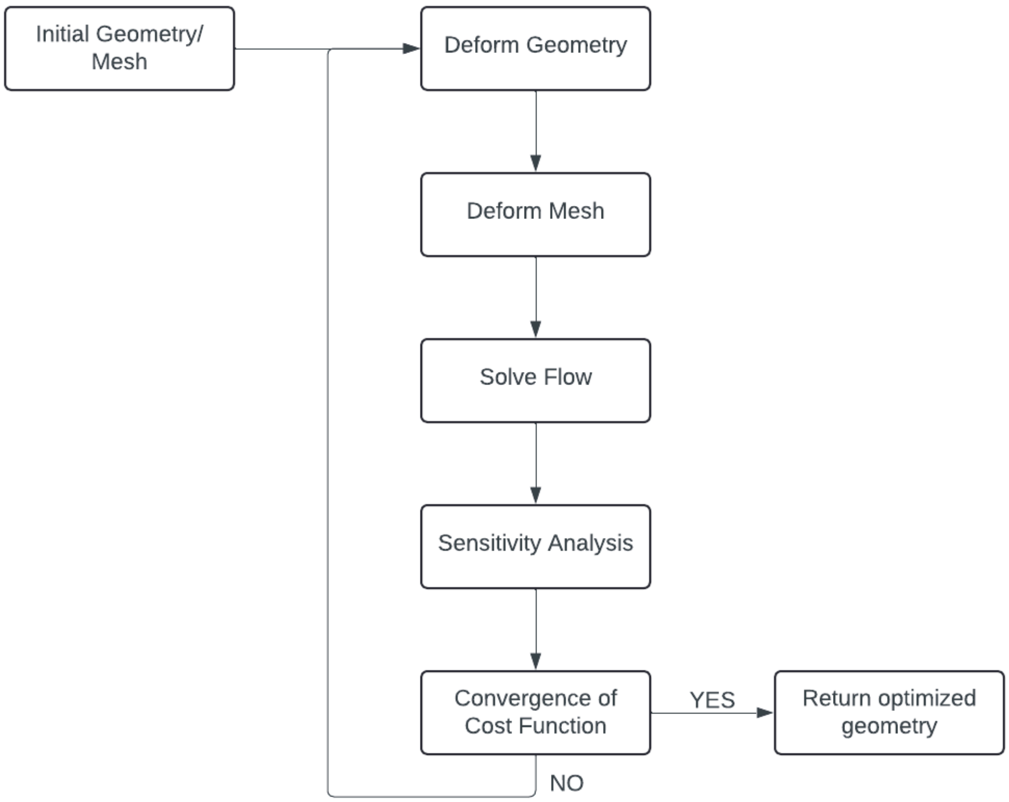
2. Project Introduction

Conventional Shape Optimization in simulations requires the minimisation of a cost function based on grid parameters. This involves updating the mesh structure at each iteration to accommodate the changing geometry. The key metrics in the analysis of mesh quality are orthogonality, skewness, smoothness, and aspect ratio. Typical implementations in SU2 for mesh deformation are based on spring systems and elasticity models. These implementations impact the mesh domain's orthogonality and skewness, adding numerical diffusion and wiggles to the solution via the advective and diffusive terms in Navier Stokes. Control of the meshing is challenging near boundaries as well because elements may cross each other, giving rise to invalid elements (negative volume), which is a common issue during deformation.

To avoid this, we propose implementing a re-meshing algorithm to provide volume-positive elements and accurate node sensitivities to replace the current deformation-based functionality in SU2.

The steps to be worked on are as follows :

1. Implementing a constrained orthogonal-structured grid generation algorithm [1]
2. Computation of node gradients of loss function w.r.t. surface parameters for the new mesh
3. Integration with SU2 for post-processing and calculations of the next step



Current Pipeline for Shape Optimisation

We would implement an orthogonal-structured grid generation algorithm to replace the mesh deformation module shown in the diagram above. The algorithm would be augmented with constraints of gradient smoothening and element conservation to ensure continuity between initial and updated geometry. This can be achieved by modelling it as an optimisation problem, as shown by Parthasarathy et al. [2]
For the second step, an automatic differentiation library must be used to compute the gradients of the loss function at the nodes w.r.t to the surface parameters. I propose using CoDiPack, an open-source C++ library, to

achieve this task, which should be straightforward for structured mesh equations.

The pipeline created would be integrated with the CDeformationDriver class to update mesh and gradient information in a manner similar to the architecture for CFreeFormDeformationBox.

3. Milestones

The key milestones in the project would be as follows below :

1. Implementation of structured-orthogonal meshing of a domain

2. Implementation of continuity constraints for the meshing algorithm

3. Wrapping Python-based meshing implementation in C++ for low-latency processing downstream

4. Addition of CoDiPack to the meshing algorithm for automatic differentiation of the loss function at new mesh nodes

5. Mid-term evaluation - calculation of the difference in mesh quality metrics compared to the FFD-Elasticity method in SU2

6. Complete integration of pipeline with the existing SU2 architecture

7. Benchmarking of the implementation using documented shape optimisation test cases

8. End-term evaluation - comparison of numerical solution accuracy and other improvements to mesh deformation methods

4. Timeline

As the total coding timeline is from May 27 to August 26, I divide the tasks in my proposal according to the weeks, each consisting of 5 working days, as follows :

Weeks	**Dates**	**Task Outline**
1-2	27th May-7th June	1. Implement orthogonal meshing algorithm for 2D case 2. Implement orthogonal meshing for a surface 3. Calculate node sensitivities
3-4	10th June -21st June	1. Implement optimisation problem for adding constraints to the algorithm 2. Add constraint for the number of elements 3. Add constraint for gradient smoothening
5-6	24th June - 5th July	1. Add CoDiPack for automatic differentiation w.r.t surface parameters 2. Wrap the developed Python modules in C++
7	8th July - 12th July	1. Midterm evaluation submission - Generate orthogonal mesh and gradients given an initial mesh/domain with boundaries, measure improvement of quality metrics 2. Start replacement of Mesh Deform module
8	15th July - 19th July	A break for some rest and travel!

9	22nd July - 26th July	1. Continue integration of pipeline with SU2, replacing Mesh Deform module 2. Run shape-optimization pipeline for a test case
10	29th July - 2nd Aug	Buffer week to accommodate for unanticipated delays and bugs
11	5th Aug - 9th Aug -	Benchmarking of implementation against shape-optimisation test cases
12-13	12th Aug - 23rd Aug	1. Raising a pull request and performing code reviews 2. Documentation of work 3. Final evaluation submission - code performance comparison in meshing quality and numerical accuracy

Each working day, I would like to work around 3 - 3.5 hours a day to distribute my workload in a uniform way. This amounts to approximately 175 working hours in the total duration.

5. Specific Deliverables

The specific deliverables of the project according to the milestones are as follows :

1. A document defining the theory for orthogonal mesh generation
2. A module which returns a new mesh with node gradients given an input mesh
3. A document which shows the improvement in mesh quality for a single iteration through the module
4. A pull request for the integrated pipeline in SU2

5. Documentation of the entire implementation and modules implemented
6. A document comparing the results of the new pipeline to the FFD/Elasticity-based optimisation results

6. References

1. Thompson, J.F., Soni, B.K., & Weatherill, N.P. (Eds.). (1998). Handbook of Grid Generation (1st ed.). CRC Press. https://doi.org/10.1201/9781420050349
2. R. Durand, B.G. Pantoja-Rosero, V. Oliveira, A general mesh smoothing method for finite elements, Finite Elements in Analysis and Design, Volume 158, 2019, Pages 17-30, ISSN 0168-874X, https://doi.org/10.1016/j.finel.2019.01.010