Robust Mesh Quality Improvement Feature for Structural Shape Optimization

Software Lab Project 2017 – Chair of Structural Analysis

Students: Bodhinanda Chandra, Yuheng Zhou, Zheng Wang

Supervisors: Aditya Ghantasala, Armin Geiser

Abstract

In a node-based structural shape optimization, usually, only the surface mesh is considered during the optimization process. This means that only the surface nodal coordinates are updated at each optimization iteration. However, generally, the internal nodes are not moving accordingly, and thus, the quality of the internal mesh may become problematic as the surface deformation increases. While some of the traditional optimizations perform geometry updates and remeshing at every iteration, a more efficient approach using mesh motion algorithm can save a lot of computational time. A mesh motion algorithm by Radial Basis Functions (RBFs) is one of the methods which can interpolate the nodal displacements and fit them according to the surface movement so that the distortion in the internal mesh can be reduced and the quality of the mesh can be preserved at an acceptable level. Furthermore, the method is also chosen since the problem definition does not depend on the type of mesh used at all. This means that the RBFs mesh motion only needs point cloud information and can be used in every kind of element types. In this project, RBF interpolation algorithm is implemented within the structural optimization framework developed by the Chair of Structural Analysis, where several types of basis function are tested for different kind of problem setups.

Radial Basis Functions

The radial basis function is a well-established numerical tool to interpolate scattered data, fitting a solution for the mesh morphing from a list of source points and their displacements. The RBF problem definition does not depend on the mesh. It can be used in different kinds of mesh element type, in both 2D and 3D.

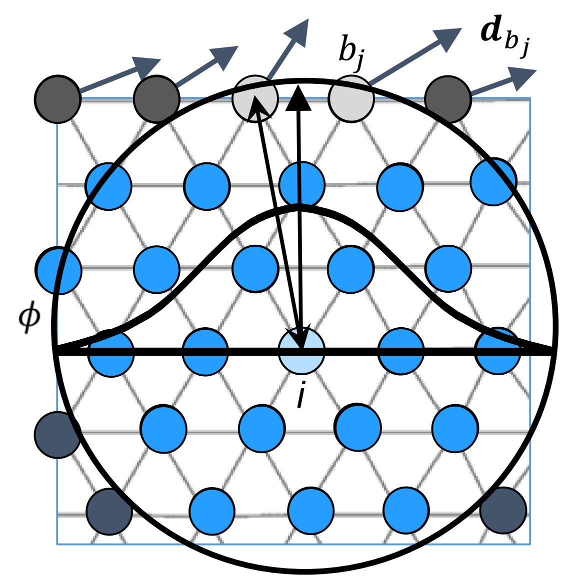
Interpolation Function

Polynomial $p(\mathbf{x}_i) = \beta_0 + \beta_1 x_i + \beta_2 y_i + \beta_3 z_i$

$$s(\mathbf{x}_i) = \sum_{j=1}^{n_b} \alpha_j \underbrace{\phi\left(\overbrace{||\mathbf{x}_i - \mathbf{x}_{b_j}||}^{\text{Point Distance}}\right)}_{\text{Basis Function}} + p(\mathbf{x}_i)$$

Interpolation Condition

1. $s\left(\mathbf{x}_{b_j}\right) = \mathbf{d}_{b_j}$
2. $\sum_{j=1}^{n_b} \alpha_j q(\mathbf{x}_{b_j}) = 0$, where $\deg(q) \leq \deg(p)$



Linear System of Equation

$$\begin{bmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}$$

- $\mathbf{M}_{n_b \times n_b} = M_{b_i b_j} = \phi\left(||\mathbf{x}_{b_i} - \mathbf{x}_{b_j}||\right)$
- $\mathbf{P}_{n_b \times 4} = \begin{bmatrix} 1 & x_{b_1} & y_{b_1} & z_{b_1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{b_{n_b}} & y_{b_{n_b}} & z_{b_{n_b}} \end{bmatrix}$

Solving the Equation

- $\boldsymbol{\alpha}_j = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_{nb}]^T$
- $\boldsymbol{\beta}_j = [\beta_0 \quad \beta_1 \quad \beta_2 \quad \beta_3]^T$

De Boer, A., van der Schoot, M. S., Bijl, H., *Mesh deformation based on radial basis function interpolation*, Computers and Structures 85 (2007) 784–795

Software Implementation

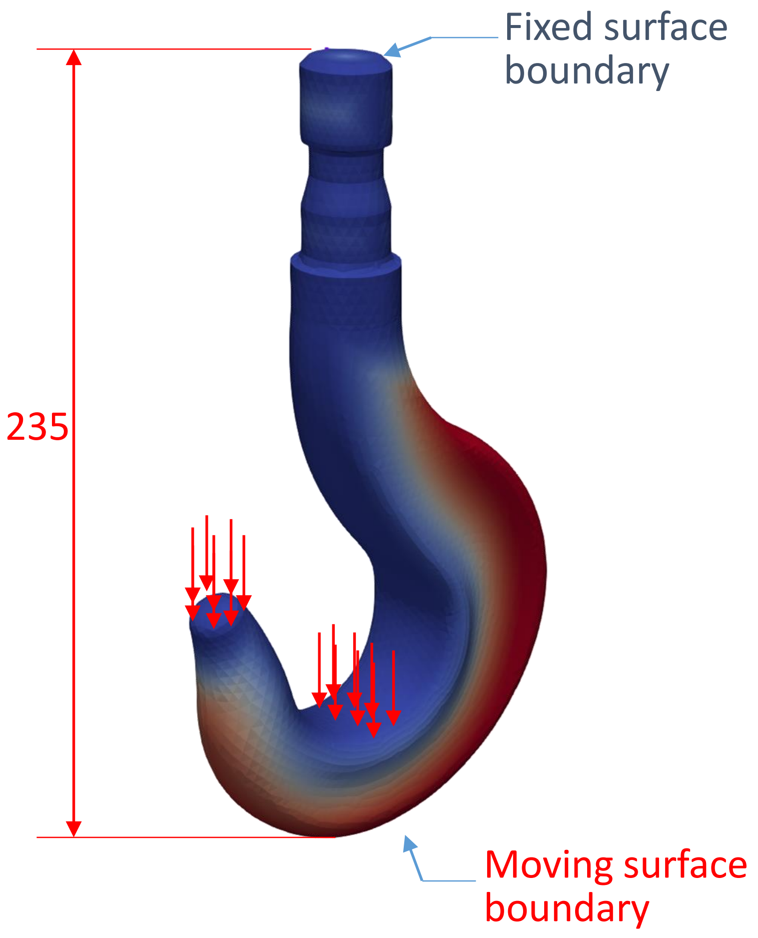
The mesh motion RBF algorithms are implemented as a C++ class which then will be wrapped and accessed as a python class together with the other essential structural optimization functionalities, e.g. Finite Element Method software, optimization algorithms, etc.

Class in C++: MeshmotionRBF.h, MeshmotionRBF.cpp → **Application Programming Interface (API):** ShapeModule.cpp → **Wrapper (using Ctypes):** MeshmotionWrapper.py, MeshmotionWrapper.pyc → **Main program:** Runtest.py

Input parameters: Parameter.json, Nodeset.dat, Mesh.dat → **Main program:** Runtest.py

Other functionalities: Structural analysis using FEM, optimization algorithms, etc. → **Main program:** Runtest.py

Numerical Results: Minimum Strain Energy Shape Optimization of a Hook Structure



1. Finite Element Method structural analysis using tetrahedral mesh in Carat

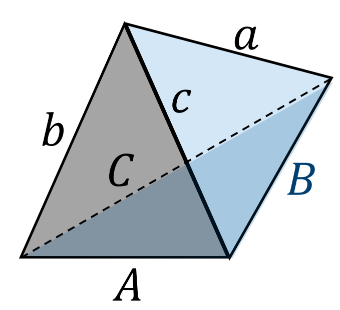
2. Structural Optimization using steepest descent gradient-based algorithm
Objective function: minimum strain energy
Design variable: the surface nodal position of the moving boundary

3. Mesh Motion using RBF to morph the internal volume mesh
Radius: 25.0
Type of Basis Function: CTPSC1

Mesh Quality Metrics for 3D Tetrahedral

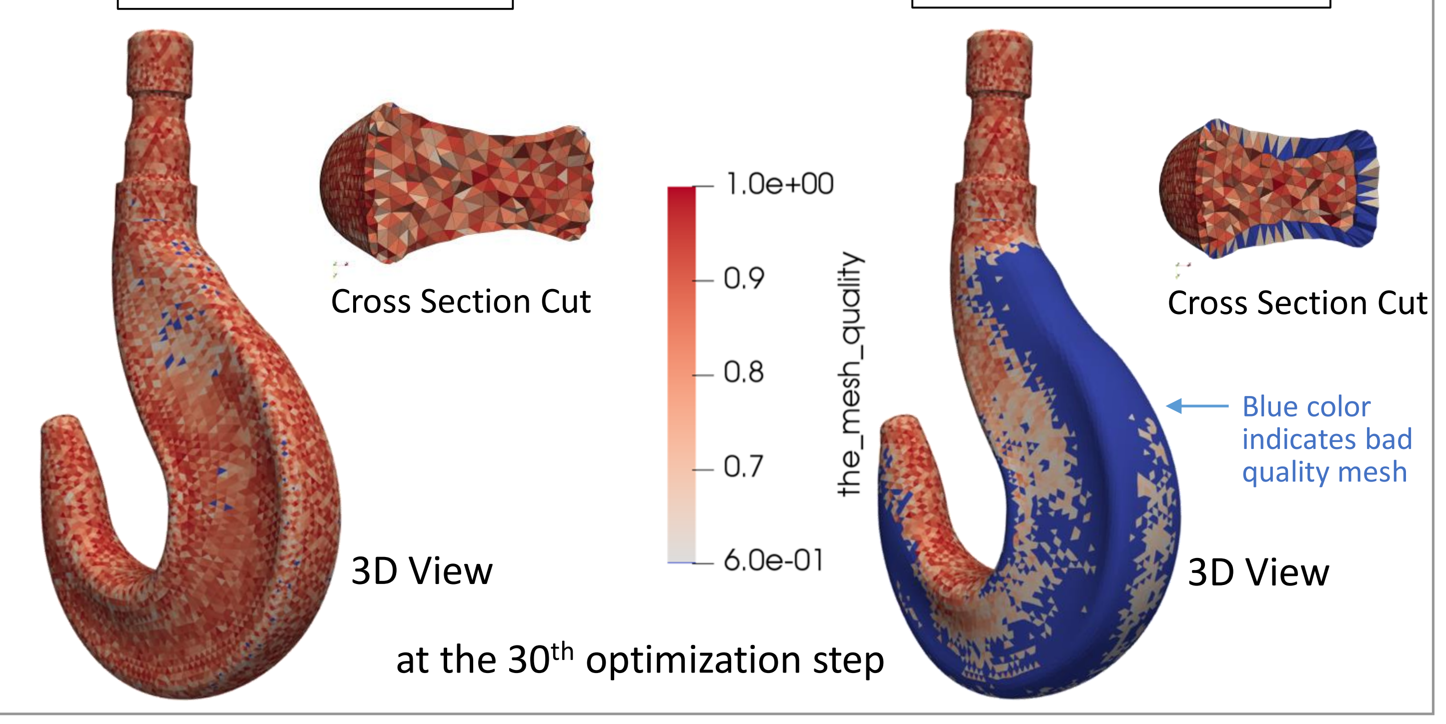
$$q = \frac{12(3V)^{\frac{2}{3}}}{(A^2 + B^2 + C^2 + a^2 + b^2 + c^2)}$$

- A, B, C, a, b, c are the sides of the tetrahedral
- V is the volume of the tetrahedral



If $q > 0.6$ the mesh is of acceptable quality

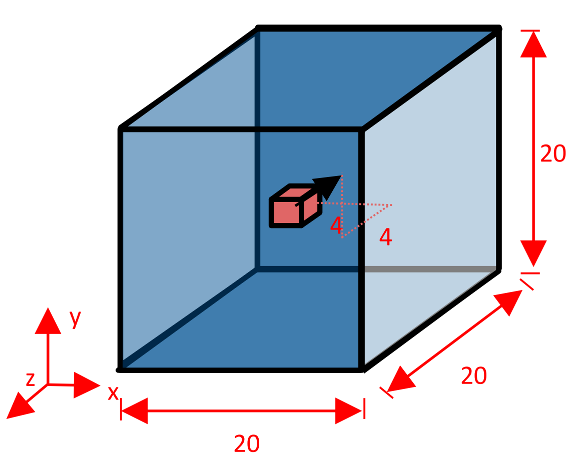
Bank, Randolph E., *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, User's Guide 6.0*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.



Efficiency Test

Computer Environment:
- Processor: intel i7-4770 CPU 3.4GHz
- Memory: 16GB
- OS: Ubuntu 14.04 LTS (64 bits)

Linear Solver: Eigen-SparseQR
Node Search Algorithm: FLANN



NO.	Name	$f(\xi)$
1	CP C^2	$(1-\xi)^4(4\xi+1)$
2	CTPS C^1	$1+\frac{80}{3}\xi^2 - 40\xi^3 + 15\xi^4 - \frac{8}{3}\xi^5 + 20\xi^2\log(\xi)$
3	CTPS C_a^2	$1 - 30\xi^2 - 10\xi^3 + 45\xi^4 - 6\xi^5 - 60\xi^3\log(\xi)$
4	CTPS C_b^2	$1 - 20\xi^2 + 80\xi^3 - 45\xi^4 - 16\xi^5 + 60\xi^4\log(\xi)$