Rigid Motion Mesh Morpher: a robust morphing tool for adjoint-based shape optimization

George Eleftheriou and Guillaume Pierrot

ESI Group, Rungis, 94513 CEDEX, France

A complete discrete adjoint-based shape optimization process necessitates the use of a reliable and robust mesh deformation tool. The Rigid Motion Mesh Morpher is a mesh deformation tool based on the principle of gracefully deforming the mesh in a way that keeps the motion of its parts (groups of nodes) as-rigid-as-possible. This is achieved by minimizing a distortion metric, hence solving the problem of minimization of a mesh deformation energy functional.

I. Introduction

In adjoint based shape optimization problems, there are two ways of dealing with the required changes to the mesh. The first one is regenerating a mesh (re-meshing) based on the new shape, while the second one is adapting the existing mesh (by moving the nodes) to fit the new shape (morphing or mesh deformation). Re-meshing may be time consuming, as it is introduced as a separate step inside the optimization loop, and tedious as it may require user-intervention. It also introduces inconsistencies in the process as the sensitivities are normally supposed to be computed at isoconnectivity. Morphing, on the other hand, is also challenged, mainly by the need to maintain the mesh quality (avoiding distorted and negative cells) while deforming it. Within this context, various mesh morphing techniques have been developed. The Spring Analogy[1] is simple but may suffer robustness issues. Laplacian smoothing[6] is suitable for translation but does not account for rotation. The Linear Elasticity approach[3] does not account for mesh anisotropy and is difficult to implement for general meshes because finite elements are used to solve the equations. Finally, the Radial Basis Functions[5] are promising but computationally heavy as the matrices involved are full, restricting the mesh size and complicating the implementation.

II. Rigid Motion Mesh Morpher

The Rigid Motion Mesh Morpher approach, as is shown in the present study, overcomes the above-mentioned limitations, being more flexible and essentially mesh-less, since it does not require any inertial quantities or cell connectivities related to the mesh. Firstly, the set of surface nodes (or boundary nodes, the ones defining the shape) of the mesh is identified. The prescribed motion of these nodes (their velocities) is known. Then, all nodes are grouped into “stencils” and all the stencils are required to deform in an as-rigid-as-possible way. Thus, for every stencil we define a rotation velocity and a translation velocity. Those, along with the velocities of all volume nodes (or internal nodes), form the set of unknowns. The as-close-to-rigid-as-possible motion is achieved by attempting to minimize a metric representing the difference of the actual deformation from a perfectly rigid motion (simply a translation plus a rotation). The resulting system of equations is solved, using the prescribed motion of the boundary nodes as input to form the boundary conditions.

The aforementioned qualities hold true for as long as the prescribed displacements of the boundary nodes are tiny enough, in order to remain within the linear range of the problem. For larger displacements, subcycling must be used in order to subdivide the prescribed displacements into smaller ones. Hence, a “rigid-motion” history of the stencils is kept at all times during the morphing cycles to compute the extent to which some stencils need to be rigidified.

An improvement upon this concept, is the development of a non-linear variant, namely, the Finite Transformation Rigid Motion Mesh Morpher, that eliminates the need for subcycling and keeping track of the “rigid-motion” history of the stencils for all subcycles. It employs the Polar Decomposition[4] to compute a rotation matrix and bears some similarity to.[2] Finally, there is a significant gain in terms of morphing efficiency and the quality of the resulting mesh.

III. Results

The tool has been tested and benchmarked both as a standalone tool (no optimization case involved) and as a link of an adjoint-based shape optimization toolchain. Mid-size industrial test cases have been run and the results will be presented.

References

[1] Carlo L. Botasso, David Detomi, and Roberto Serra. The Ball-Vertex Method: a New Simple Spring Analogy Method for Unstructured Dynamic Meshes. *Computer Methods in Applied Mechanics and Engineering*, 194:4244–4264, 2005.

[2] Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. A simple geometric model for elastic deformations. In *ACM Transactions on Graphics (TOG)*, volume 29, page 38. ACM, 2010.

[3] Richard P. Dwight. Robust Mesh Deformation using the Linear Elasticity Equations. In *Computational Fluid Dynamics 2006*, pages 401–406. Springer, 2009.

[4] Nicholas J Higham. Computing the polar decomposition-with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986.

[5] Stefan Jakobsson and Olivier Amoignon. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers & Fluids*, 36(6):1119–1136, 2007.

[6] Z. Su, S. Wang, C. Yu, F. Liu, and X. Shi. A Novel Laplacian Based Method for Mesh Deformation. *Journal of Information and Computational Science*, 7(4):877–883, 2010.