21st International Conference **ENGINEERING MECHANICS 2015**
Svratka, Czech Republic, May 11 –14, 2015
Paper **#123, pp. 268–269**

Parallel Data Mapping between Simplex Finite Element Meshes using Mesh Topology

Daniel Rypl[a], Jan Vesecký[b]

Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague,
Th´akurova 7; 166 29, Prague; Czech Republic

[a]daniel.rypl@fsv.cvut.cz, [b]jan.vesecky@fsv.cvut.cz

Keywords: finite element data transfer, finite element mesh mapping, walking algorithm, mesh traversal, parallelization, threads

Abstract: Presented paper focuses on the transfer of finite element data between simplex finite element meshes representing different discretizations of the same geometrical model. The construction of the localization map, defining for each node of the target mesh, onto which the data are to be transferred, the closest element of the original source mesh, is based on the walking algorithm, performing a traversal between neighbouring elements of the source mesh from an initial element towards the processed node. To make the algorithm efficient and reliable, individual nodes of the target mesh are processed in special order given by the nodal connectivity of the target mesh. The implementation also relies on the classification of the processed meshes to the underlying geometrical model. Once the localization map is available, data from the source mesh are mapped to nodes of the target mesh in parallel (using threads) by the help of shape functions of the elements of the source mesh.

Introduction

A need for transfer of finite element data between finite elements meshes appears commonly in many finite element method based applications. There are several approaches for mapping the field variables. The most common one is based on shape functions of element comprising given point [1, 2], the others use typically the nearest point or set of points [3]. The common denominator of these techniques is the need to localize the node or element of the original mesh nearest to a particular node (or point) of the mesh on which the field variables are to be recovered. Since the recovery is likely to be performed many times during the analysis, in the past the attention was primarily focused on the computational efficiency of the localization process. In this context, the spatial index based approaches using various tree or dynamic cell data structures proved to be especially powerful. However, these approaches use typically only geometrical location/distance to determine the index (position of the node/point within the data structure), which may lead to incorrect results in special cases when dealing, for example, with geometry with curved discrete cracks. In these cases the topological distance should be also considered.

Walking algorithm

In the present approach, the localization process does not rely on any spatial index and the search is performed by a walking algorithm [4, 5] based on the topological information of the mesh and the knowledge from previous searches. The basic idea comes from the fact that elements of the original mesh comprising adjacent nodes of the new mesh are likely to be relatively close to each other. Thus the candidates for localization are always not yet processed neighbours of already localized nodes which are inside or on the boundary of the processed region. The order in which the nodes are processed is controlled by a queue, which must be initialized by (at least) one node, the source element of which is known. This should be either a vertex node, in which case the target element can be quickly identified as one of elements on the original mesh incident to the same (classified to the same vertex or having

identical coordinates) vertex node, or interior node (or a virtual point inside the region (as a last resort)), in which case target element comprising the node (or point) is found by inspecting successively individual elements (classified to the processed region) of the original mesh until the one comprising the node is localized. Once the queue is initialized, it is processed on the first-in first-out basis until it becomes empty. If it is empty but there are still some unprocessed nodes in the target mesh, the queue must be newly initialized by a not yet processed target node(s) and the processing of the queue is invoked again. Once localization of a particular node in the queue is completed, its shape functions within the localized source element are computed and stored for later use during the mapping of the field variables. If the node is outside of its source element then it is marked as provisionally localized. If the node is inside of the source element and it is either interior or boundary classified to an entity of the next lower dimension (with respect to the processed region) then all its not yet processed or provisionally localized neighbour nodes are added to the queue using its source element localized by the walking algorithm as the initial element for their own localization. The advantage of the proposed approach is that it needs no additional searching data structure and relies only on the mesh topology and optionally also on the classification of the mesh with respect to the underlying geometrical model. The disadvantage consists in the fact that all the nodes, not only the selected ones, of the new mesh must be processed to ensure convergence and efficiency of the algorithm. Also the localization of the first node, when performed by intensive search, can be considered a drawback (despite the fact that this search is likely to be computationally less demanding than building the complete searching data structure), especially if performed several times when processing disconnected meshes, for example.

Summary

In this paper, a topology based methodology for the transfer of data between simplex finite element meshes has been introduced. While the standard approaches are based on various spatial decomposition techniques using typically Euclidean distance as the measure of proximity of the objects, the proposed method relies on topological features of the processed meshes and their classification to the underlying geometrical model. The approach proved to be not only reliable but also efficient. Its efficiency is further increased by running the actual mapping of data in several concurrent threads.

Acknowledgments: This work was supported by the Technology Agency of the Czech Republic under project No. TA02011196. Its financial assistance is gratefully acknowledged.

References

[1] S.M. Afazov, A.A. Becker, T.H. Hyde, Development of a Finite Element Data Exchange System for Chain Simulation of Manufacturing Processes, Advances in Engineering Software, 47 (2012) 104–113.

[2] D. Scrimieri, S.M. Afazov, A.A. Becker, S.M. Ratchev, Fast Mapping of Finite Element Field Variables Between Meshes with Different Densities and Element Types, Advances in Engineering Software, 67 (2014) 90–98.

[3] Y. Luo, A Nearest-Nodes Finite Element Method with Local Multivariate Lagrange Interpolation, Finite Elements in Analysis and Design, 44 (2008) 797–803.

[4] O. Devillers, S. Pion, M. Teillaud, Walking in a Triangulation, International Journal of Foundations of Computer Science, 13 (2002) 181–199.

[5] R. Soukal, V. Purchart, I. Kolingerová, Surface Point Location by Walking Algorithm for Haptic Visualization of Triangulated 3D Models, Advances in Engineering Software, 75 (2014) 58–67.