Available online at www.sciencedirect.com
ScienceDirect
Computers and Structures 85 (2007) 944–954
Computers & Structures
www.elsevier.com/locate/compstruc

A unified approach to the deformation of simplicial and non-simplicial meshes in two and three dimensions with guaranteed validity ☆

Nazmiye Acikgoz [a], Carlo L. Bottasso [b,*]

[a] Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, 270 Ferst Dr., Atlanta, GA 30332-0150, USA
[b] Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, Via La Masa 34, Milano 20156, Italy

Received 9 June 2006; accepted 21 November 2006
Available online 12 January 2007

Abstract

In this paper we present a unified formulation that embraces the problem of deformation of simplicial and non-simplicial, structured and unstructured, two and three dimensional meshes. The method is formulated so as to avoid, by construction, the generation of invalid elements. At first, we show that in all cases above, invalid elements are generated by the same collapse mechanism. Specifically, an invalid element is formed when a mesh vertex leaves its ball, i.e. the polyhedral cavity formed by all triangular faces (in three dimensions) or edges (in two dimensions) that are edge-connected to the vertex. Next, we show that, in all cases above, collapse of elements is avoided by connecting with a spring each vertex in the grid with its normal projection on the ball boundaries.

The proposed method is demonstrated on a number of difficult example problems denoted by severe grid deformations, and with the help of a variety of grid types.

Keywords: Mesh deformation; Dynamic meshes; Fluid–structure interaction; Coupled problems; Aeroelasticity; Ball-vertex method

1. Introduction

Many challenging multi-physics and multi-field problems are unsteady in nature and are characterized by moving boundaries and/or interfaces. In all these cases, the motion of a portion of the domain boundary is known and one wants to deform the rest of the mesh in order to accommodate these imposed displacements. For this purpose a vertex repositioning problem must be solved in a robust and efficient way, and such that invalid elements are avoided even for large amplitude motions. The basic idea behind all methods for this class of problems is to define a suitable fictitious elasticity problem over the domain.

The fictitious problem can either be continuous [7] or discrete. For the former approach, partial differential equations are discretized in space, for example using the finite element method. Alternatively, a lumped-parameter discrete structural model can be used. To our knowledge, a thorough comparison of these two classes of methodologies is still lacking, so the question of which of the two is the best remains open. In this study we consider only the discrete case, where the fictitious problem defines a suitable network of springs associated with the mesh. The problem then becomes how to construct the best possible network of springs that: (a) is inexpensive to compute; (b) does not contain collapse mechanisms; and (c) leads to graded and well shaped deformed grids, even for large imposed displacements.

☆ Submitted to the Fourth MIT Conference on Computational Fluid and Solid Mechanics, June 13–15, 2007, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

* Corresponding author. Tel.: +39 02 2399 8315; fax: +39 02 2399 8334.
E-mail address: carlo.bottasso@polimi.it (C.L. Bottasso).

0045-7949/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2006.11.009

The most widely used and the simplest mesh deformation technique is the spring analogy method [2], where each edge is replaced by a spring, whose stiffness is inversely proportional to the edge length. While this classical method performs reasonably well in a number of cases, it does indeed fail as soon as the local grid motion is not small compared to the local mesh size. Unfortunately, in many practical cases the necessary grid displacements are not small.

Indeed, there is a need for methods that can specifically deal with large deformation problems. To address this issue, torsional springs were added to the linear edge springs by Farhat et al. [6,5]. The torsional springs are designed so as to ensure that mesh entanglement will be avoided. The method works well in practice, but it becomes quite cumbersome in three dimensions. Furthermore, its use for non-simplicial meshes would first require to split quads into triangles and hexas into tetrahedra.

A new simple method of controlling collapse mechanisms for simplicial meshes was presented in Ref. [3]. In this work we develop a unified formulation that covers the simplicial and the non-simplicial structured and unstructured cases. Furthermore, the method is applicable to hybrid grids composed of a mix of simplicial and non-simplicial elements. The method is based on a network of linear face-vertex springs in three dimensions or linear edge-vertex springs in two dimensions. These springs effectively constrain each vertex within the polyhedral ball that encloses it, contrasting the possible collapse mechanisms of the grid elements. In fact, the condition for a valid grid to remain valid even after deformation is that each vertex must remain confined to its ball. This condition holds both for simplicial and non-simplicial meshes in two and three dimensions. Therefore, this method avoids mesh entanglement during deformation by explicitly enforcing the vertex containment condition.

This work is organized as follows. The problem of mesh motion is first formulated in Section 2. Next, in Section 3 we study the collapse mechanisms of both two and three dimensional simplicial and non-simplicial meshes, and we show that the condition for a valid mesh to remain valid after deformation can be expressed in terms of the containment condition of each vertex within its ball. In Section 4 we formulate the ball-vertex method, by inserting springs which resist the motion of each vertex towards each boundary entity of its ball. The method is applied to two and three dimensional problems using simplicial and non-simplicial elements in Section 5. The results are quantitatively assessed using objective mesh quality measures, rather than using numerical coupled simulations which would raise questions about the solver type, the coupling, the details of the time marching procedures, etc. Finally, conclusions are briefly discussed in Section 6.

2. Problem formulation

We consider a deforming mesh problem, i.e. a problem where we are interested in deforming the domain Ω, and hence the grid $\mathscr{T}_h$ associated with it, in order to accommodate some given displacement on a portion of its boundary. The basic idea behind all methods for this class of problems is to define suitable fictitious structural properties for the domain. The deformed configuration of the elastic domain can then be computed, under the action of the driving displacements.

In general, the domain boundary Γ can be partitioned according to the following criterion:

$$\Gamma = \Gamma_{\mathrm{m}} + \Gamma_0 + \Gamma_{\mathrm{s}}, \tag{1}$$

as depicted in Fig. 1. The moving boundary of the domain Ω is noted Γ_{m}. On the moving boundary, displacements are known and drive the mesh deformation process. On Γ_0 the grid displacements are null, while on the sliding boundary Γ_{s} displacements are constrained to be tangential to the boundary.

In the simulation of a transient process, the driving boundary conditions for the mesh deformation problem are to be regarded as functions of time. Consequently, a mesh deformation problem is solved at each time step during the transient simulation. After imposing the boundary conditions on Γ_0 and Γ_{s}, one obtains the following fictitious elasticity problem:

$$\boldsymbol{M}\ddot{\boldsymbol{u}}_M + \boldsymbol{C}\dot{\boldsymbol{u}}_M + \boldsymbol{K}\boldsymbol{u}_M = \boldsymbol{B}\boldsymbol{g}, \tag{2}$$

where $\boldsymbol{u}_M$ is the vector of displacements of the moving vertices of the grid, $\boldsymbol{g}$ is the vector of imposed displacements on Γ_{m}, $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ are the inertia, damping and stiffness matrices, respectively, and $\boldsymbol{B}$ is found by imposing the boundary conditions on the various portions of the domain boundary. More often than not, and in the present work, only the static version of the problem is used, i.e. one sets the time derivatives of $\boldsymbol{u}_M$ to zero and solves

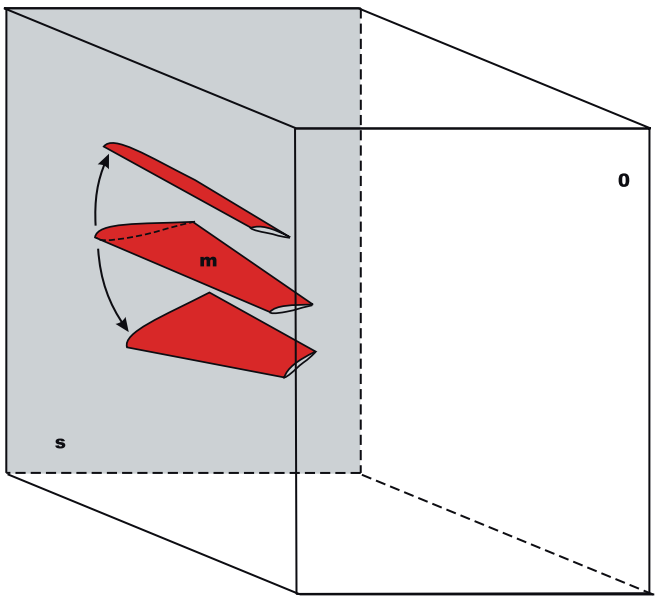

Fig. 1. Partitioning of domain boundary. Moving boundary: Γ_{m}; null displacement boundary: Γ_0; sliding boundary: Γ_{s}.