Computers and Structures 193 (2017) 87–109

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Robust *h*-adaptive meshing strategy considering exact arbitrary CAD geometries in a Cartesian grid framework

Onofre Marco *, Juan José Ródenas, José Manuel Navarro-Jiménez, Manuel Tur

Centro de Investigación en Ingeniería Mecánica (CIIM), Universitat Politècnica de València, Valencia 46022, Spain

ARTICLE INFO

Article history:
Received 24 February 2017
Accepted 4 August 2017
Available online 17 August 2017

Keywords:
Cartesian grids
h-refinement
NURBS
NEFEM

ABSTRACT

Geometry plays a key role in contact and shape optimization problems in which the accurate representation of the exact geometry and the use of adaptive analysis techniques are crucial to obtaining accurate computationally-efficient Finite Element (FE) simulations. We propose a novel algorithm to generate 3D *h*-adaptive meshes for an Immersed Boundary Method (IBM) based on Cartesian grids and the so-called NEFEM (NURBS-Enhanced FE Method) integration techniques. To increase the accuracy of the results at the minimum computational cost we seek to keep the efficient Cartesian structure of the mesh during the whole analysis process while considering the exact boundary representation of domains given by NURBS or T-Splines.

Within the framework of Cartesian grids, the two significant contributions of this paper are: (a) the methodology used for the mesh-geometry intersection, which represents a considerable challenge due to their independence; and (b) the robust procedure used to generate the integration subdomains that exactly represent the CAD model. The numerical examples given show the proper convergence of the method, its capacity to mesh complex 3D geometries and that Cartesian grid-based IBM can be considered a robust and reliable tool in terms of accuracy and computational cost.

1. Introduction

It has recently become clear that a major drawback to the rapid structural analysis of geometrically elaborated 3D domains using Finite Element Analysis (FEA) is the time allotted to creating an appropriate finite element mesh. Even after the development of sophisticated mesh generators, a significant amount of skilled human resources is required to create good quality finite element meshes for the geometrically complex models required for the solution of common industrial problems.

The aim of adaptive mesh generation and automatic error control in FEA is to eliminate the need for manual re-meshing and re-running design simulations to check the numerical accuracy. In ideal circumstances, the user should only input the component model and a coarse finite element mesh. The software should then autonomously and adaptively reduce the element size where required, reducing the error in the solution fields to a predetermined value.

Adaptive methods of finite element simulations were first proposed in the late 70's [1,2]. The most common criterion in general engineering use is that of prescribing a limit for a global magnitude, such as the error computed in the energy norm, though it is possible to define magnitudes of interest to evaluate the goodness of the evaluated meshes [3–5].

The procedures for the refinement of finite element meshes fall mostly into two categories:

1. The *h*-refinement, in which the element type is maintained but the elements are changed in size. In some locations of the mesh the element sizes are made smaller, or larger (not very common), where needed to provide maximum computational economy in reaching the desired solution.
2. The *p*-refinement, in which the element size is kept constant and the order of the polynomial, used in its definition is increased where necessary, generally by using hierarchical shape functions [6,7].

There exists a third category, the *hp*-refinement [8,9], which consists of simultaneously adapting the size of the elements and their approximation degree.

In this contribution we will present an *h*-adaptive refinement strategy based only on the size of the element keeping the polynomial order of the interpolation constant.

* Corresponding author.

E-mail addresses: onmaral@upvnet.upv.es (O. Marco), jjrodena@mcm.upv.es (J.J. Ródenas), jonaji@upvnet.upv.es (J.M. Navarro-Jiménez), manuel.tur@mcm.upv.es (M. Tur).

http://dx.doi.org/10.1016/j.compstruc.2017.08.004
0045-7949/

Decades after the development of functional meshing techniques [10–12], mesh generation still has to evolve in order to minimize the design cycle time because real industrial applications are, in general, geometrically complex and traditionally require a skilled workforce to generate an analysis-suitable finite element mesh.

One alternative to reduce the meshing burden, related to the proposals in this paper, is to use mesh generators based on simple discretizations such as *octrees* [13–15]. In octree-based mesh generators [12,16,17] an embedding cube-shaped domain is created and meshed following a Cartesian hierarchy through the mesh generation process for efficiency. After adapting the octree mesh to the geometry and splitting the cut cells into tetrahedrons to capture the boundary of the model, the octree is broken up into a valid body-fitted mesh and then smoothing techniques are used achieve good quality finite elements.

It is apparent that mesh generation could be greatly simplified by using implicit meshing approaches in which (as in octree techniques) the geometrically complex domain is embedded into a geometrically simpler domain whose meshing is simple if not trivial. As opposed to octree techniques, in the approach described here the non-conforming FE mesh is not modified to fit the boundary. Instead, the matching between geometry and mesh is done during the evaluation of element integrals, which are defined only by the part of the elements cut by the boundary that lies within the domain. Many methods are described in the literature in which the geometrically complex domain is embedded into a geometrically simpler domain. Among many other names used to describe these FE techniques in which the mesh does not match the domain's geometry, there is the Immersed Boundary Method (IBM) [18], the Immersed Finite Element Method (IFEM) [19] or the Finite Cell Method (FCM) [20–22]. Immersed boundary methods, often referred to as embedded methods, have been studied by a number of authors for very different problems such as, for example, shape optimization [23,24] or bio-mechanics [25–27]. Most of these techniques rely on an integration submesh in the elements cut by the boundary to perform the body-fitted numerical integration appearing in the weak formulation.

Implementing the Finite Element Method in combination with the embedded-domain concept offers a powerful alternative due to the potential benefits: virtual automatic domain discretization, suitable for creating hierarchical data structures for simple data transfer and re-use of calculations, ability to easily create adapted domain discretizations, a natural platform for efficient structural shape optimization processes, multigrid and multiscale analyses, etc. However, there are also tradeoffs with this approach related to the fact that the boundary of the domain does not necessarily coincide with the element faces. For example, there are difficulties in accurately integrating the weak form of the governing equations over the elements intersected by the boundary. There are also difficulties in imposing essential boundary conditions as the nodes do not necessarily lie on the Dirichlet boundary and the direct enforcement of the essential boundary conditions is in general not possible.

As an efficient solution for these drawbacks, we used a methodology based on the use of Cartesian grids independent of the geometry. This methodology, known as cgFEM [28,29], was implemented in a computer code for the structural analysis of 3D components considering uniform meshes. The first 3D version of this methodology, known as FEAVox [30] was described in a previous paper. The aspect that distinguishes FEAVox from other immersed boundary approaches is that it is able to consider the exact CAD representation of the boundary of the domain, given by NURBS [31,32] or T-Splines [33], in the evaluation of volume integrals. To perform the numerical integration, instead of simplifying the embedded geometry, for instance using triangular facets for its definition, FEAVox includes novel techniques to perform the exact integration (up to the accuracy of the quadrature rule) over the true computational domain. In particular, these integration techniques are the techniques considered by the NURBS-Enhanced Finite Element Method (NEFEM) [34,35].

The accurate evaluation of integrals in elements cut by the boundary, it is necessary to maintain the optimal convergence rate of the error of the FE solution. This is, therefore, an active area of research. In fact, several methodologies have recently emerged to perform high-order integration in embedded methods, such as the so-called 'smart octrees' tailored to Finite Cell approaches [36] or techniques in which the geometry is defined implicitly by level sets [37]. We used the NURBS-Enhanced integration techniques because their consistency considering the exact geometric description [38] is of major importance when dealing with CAD models in applications such as shape optimization or contact between bodies.

This contribution will show how the capabilities of the cgFEM methodology have been improved by developing *h*-adaptive analysis techniques. These techniques have been successfully implemented in FEAVox in order to handle complicated CAD models without renouncing the traditional properties of embedded methods as well as developing a robust enhanced procedure for geometry-mesh intersection.

The paper is organized as follows: a brief review of the basic features of the cgFEM methodology is given in Section 2. Section 3 explains how the mesh-geometry intersection problem is solved. Section 4 describes an extended scheme to efficiently integrate elements intersected by the boundary. Section 5 gives details of the refinement strategies. Section 6 contains numerical results showing the behavior of the proposed technique. Finally, the conclusions are given in Section 7. The derivation of the *h*-adaptive refinement criterion for 3D meshes is given in Appendix A.

2. Cartesian grids with exact representation of the geometry: FEAVox

The present work is the logical continuation of [30], which introduced the new cgFEM methodology implemented in an FE code, called FEAVox, for the analysis of structural 3D components. Its main novelty was its ability to perform accurate numerical integration in non-conforming meshes independent of the geometry. A brief review of cgFEM and its features is given here as a background to the present paper.

The foundations of mesh generation in cgFEM consists of defining an embedding domain Ω such that a bounded domain $\Omega_{\mathtt{Phys}}$ fulfills $\Omega_{\mathtt{Phys}} \subset \Omega$. Let us assume that the embedding domain is a cube, although rectangular cuboids could also be considered. This means Ω is much easier to mesh than the domain of interest $\Omega_{\mathtt{Phys}}$. Fig. 1 gives an example of the different domains defined. Fig. 1b only gives the elements of the embedding domain interacting with $\Omega_{\mathtt{Phys}}$ denoted by $\Omega_{\mathtt{Approx}}$ and Fig. 1c shows a representation of the submesh used only for integration purposes.

The original version of FEAVox considered a sequence of uniformly refined Cartesian meshes to mesh the Ω, where the different levels of the Cartesian meshes were connected by predefined hierarchical relationships. The term Cartesian grid set, denoted by $\{\mathcal{Q}_h^i\}_{i=1,\ldots,m}$, is used to define the sequence of m meshes utilized to discretize the embedding 3D domain Ω. For each level i of refinement, the embedding domain Ω is partitioned into $\mathtt{n}_{\mathtt{el}}^i$ disjoint cubes of uniform size, where $\mathtt{n}_{\mathtt{el}}^{i+1} = 8\mathtt{n}_{\mathtt{el}}^i$. While in a uniform refinement process this operation was carried out for every element in the mesh, in our *h*-adaptive approach, the subdivision step will be guided either by local geometrical parameters or a