ARTICLE IN PRESS

Computer-Aided Design ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer-Aided Design

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

Array-based, parallel hierarchical mesh refinement algorithms for unstructured meshes☆

Navamita Ray[a,*], Iulian Grindeanu[a], Xinglin Zhao[b], Vijay Mahadevan[a], Xiangmin Jiao[b]

[a] Mathematics and Computer Science, Argonne National Laboratory, Argonne, IL 60439, USA

[b] Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794, USA

ARTICLE INFO

Keywords:
Uniform mesh refinement
Hierarchical meshes
High-order surface reconstruction
Parallel computation
Half-facet

ABSTRACT

In this paper, we describe an array-based hierarchical mesh refinement capability through uniform refinement of unstructured meshes for efficient solution of PDE's using finite element methods and multigrid solvers. A multi-degree, multi-dimensional and multi-level framework is designed to generate the nested hierarchies from an initial coarse mesh that can be used for a variety of purposes such as in multigrid solvers/preconditioners, to do solution convergence and verification studies and to improve overall parallel efficiency by decreasing I/O bandwidth requirements (by loading smaller meshes and in-memory refinement). We also describe a high-order boundary reconstruction capability that can be used to project the new points after refinement using high-order approximations instead of linear projection in order to minimize and provide more control on geometrical errors introduced by curved boundaries.

The capability is developed under the parallel unstructured mesh framework "Mesh Oriented dAtaBase" (MOAB Tautges et al. (2004)). We describe the underlying data structures and algorithms to generate such hierarchies in parallel and present numerical results for computational efficiency and effect on mesh quality. We also present results to demonstrate the applicability of the developed capability to study convergence properties of different point projection schemes for various mesh hierarchies and to a multigrid finite-element solver for elliptic problems.

Published by Elsevier Ltd.

1. Introduction

In the numerical solution of complex partial differential equations (PDE's) using finite element methods for unstructured meshes, the two most computationally intensive steps are mesh generation and linear solvers. An initial coarse mesh representing the computational domain might not be of sufficient resolution to get meaningful results out of the discretizations for physical scales that might be present. As a result, the capability to refine a mesh is an essential part of any simulation process. Additionally, it is well known that multi-level methods such as geometric multigrid methods (GMG) can theoretically deliver optimal time complexity for solving sparse linear systems from PDE discretizations. Thus, it would be advantageous to use nested multi-level (i.e., hierarchical) meshes to achieve high-degree of verifiable accuracy and computational efficiency, especially in the context of large-scale parallel computing, as both the number of processors and the mesh resolution increase. Uniform mesh refinement (UMR) provides a simple and efficient way to generate such hierarchies via successive refinement of the mesh at a previous level. It also provides a natural hierarchy via parent and child type of relationship between entities of meshes at different levels that enable queries that support computation of projection operators between levels. While UMR is a relatively simple process, it is by no means trivial especially in a parallel setting. Notable challenges include maintenance of mesh quality, multi-level and multi-degree refinement, and data structure and software design.

In this paper, we develop parallel uniform refinement-based algorithms to generate multi-degree, multi-dimensional and multi-level meshes from coarse unstructured meshes. The generated mesh hierarchies can be used for a variety of purposes such as convergence studies, multilevel methods, generating large meshes in parallel to overcome IO bottlenecks, etc. While the multi-degree refinement allows achieving uniformly greater resolution faster, the multi-dimensional refinement preserves the hierarchy over explicit lower dimensional entities such as curves in surfaces, or surfaces embedded in volumes.

☆ This paper has been recommended for acceptance by Franck Ledoux and Katherine.

* Corresponding author.
E-mail address: nray@mcs.anl.gov (N. Ray).

http://dx.doi.org/10.1016/j.cad.2016.07.011
0010-4485/Published by Elsevier Ltd.

The key contributions of the paper include: (1) developing a template-based refinement strategy for subdividing each entity into smaller entities to support multi-degree refinement patterns, (2) extending the array-based half-facet (AHF) data structure [1] to support hierarchy generation and efficient mesh traversals, (3) developing efficient parallel communication strategies to resolve shared entities along processor boundaries after refinement, and (4) link with various high-order point projection strategies based high-order boundary reconstruction. We develop the capability under the parallel array-based unstructured mesh framework "Mesh Oriented dAtaBase" a.k.a MOAB [2]. The developed mesh hierarchy generation supports 1D (edges), 2D (triangles, quadrilaterals), and 3D (tetrahedral, hexahedral) meshes and mixed-dimensional meshes. We present results to demonstrate the computational efficiency, memory requirements and effect on mesh quality due to template-based UMR. We also demonstrate the effectiveness of the hierarchical meshes through a multigrid application.

A key aspect of the refinement algorithm is the positioning of the new vertices as entities are refined and we currently use a linear point projection scheme. However, using linear point projection scheme for the new vertices compromises the accuracy of the geometry and in turn that of the finite element solver. To address this issue, we make use of a recently added discrete geometry module in MOAB that provides high-order point projection schemes. The discrete geometry module provides high-order boundary reconstruction strategies that are based on weighted averaging of local polynomial fittings as described in [3,4]. We present convergence studies of the geometrical errors using different point projection schemes for various mesh hierarchies.

This work is an extension of the conference paper [5] and improves the original paper on two aspects. First, we introduce an alternative approach based on a combinatorial matching algorithm to resolve shared interface entities. Secondly, we include a high-order boundary reconstruction capability to project the new points after refinement using high-order approximations instead of linear projection so that geometrical errors introduced by curved boundaries can be minimized. In [6], a parallel hierarchical tetrahedral–octahedral subdivision refinement scheme is described. The proposed scheme was specific to tetrahedral meshes whereas our framework is applicable to a larger set of entity types. Also, the refinement template in [6] leads to a significant jump in the number of entities comparing to using lower-degrees of refinements where the mesh size increase is more graded. Finally, the parallel model follows a master–slave communication model whereas we follow a purely distributed model with mesh partitioning along with various asynchronous communication strategies to overlap communication with computation for latency hiding. Our template-based approach is further extended in [7] to support both non-conformal and conformal adaptive refinement while using an array-based tree-like structure for storing the hierarchies. In [8–10], various strategies for adaptive mesh refinement for both unstructured and structured meshes are provided. Our approach, though still in preliminary stages, differs significantly. It first of all allows storing the hierarchy information and subsequently support various inter and intra level mesh traversal queries are required by multi-level methods. Secondly, unstructured meshes and as a result complex geometries can be directly supported without posing any constraint on the geometry. We refer to [11] for an overview of the existing adaptive strategies as detailed comparison is beyond the scope of this work.

The remainder of the paper is organized as follows. Section 2 reviews some background knowledge and related mesh data structures. Section 3 describes the refinement templates, underlying mesh hierarchy storage and extended half-facet data structures. Section 4 describes the algorithms for the parallel communication and mesh hierarchy generation. Section 5 presents numerical results for computational efficiency, memory requirements, effect on mesh quality, convergence properties of different point projection schemes, and an example of multigrid solver. Section 6 concludes the paper with a discussion.

2. Related work and background

2.1. Background

Mesh data structures are fundamental to meshing algorithms and mesh-based numerical methods. The underlying data structure strongly influences the overall performance of the algorithms or simulations, since it is used to perform all the mesh-based combinatorial operations and as a result has been investigated since the inception of mesh generation and computational geometry. We review some terminology before describing our data structures and mesh frameworks. We say a mesh is a *manifold* or *non-manifold* if its geometric realization is a manifold or non-manifold, respectively. In a d-dimensional mesh, we refer to the d-dimensional entities as *elements*, and refer to the $(d-1)$-dimensional sub-entities as its *facets*. Each facet in a 2-D or 3-D mesh has an orientation with respect to the containing element. For example, each edge of a triangle has a direction, and all the edges form an oriented loop. Thus, it makes sense to call the facets *half-facets*. Each facet may have multiple incident elements, especially for non-manifold entities. We refer to all such half-facets as *sibling half-facets*. A mesh is said to be *conformal* if the pairwise intersection of any two entities is either another entity (lower-dimensional) or is empty. In this paper, we consider only conformal meshes, which may be manifold or non-manifold. In some engineering applications, especially in coupled or multi-component systems, the domain of interest may be composed of a union of topologically 1-D, 2-D, and 3-D objects, such as a mixture of cables, thin-shells, and solids. We refer to such a domain and its mesh as *mixed-dimensional*. We refer to a subset of the mesh corresponding to a 1-D, 2-D, and 3-D object in the domain as a *sub-mesh*.

2.1.1. Array-based half-facet (AHF) data structure

There are a number of mesh data structures such as entity-based, boundary representations, corner table, radial-edge, winged, half-edge/face, incidence graphs, etc., that are used for mesh representation and queries. The two data structures that are relevant in our context are the half-edge and half-face data structures. The half-edge data structure is for 2D and surface meshes. It uses edge as the core object where the edge within each face is called a directed or half-edge. Typical implementations (e.g., CGAL [12,13], OpenMesh [14] and Surface_Mesh [15]) store mappings from each half-edge to its opposite half-edge, its previous and next half-edge within its face, its vertices, its incident face, as well as the mapping from each vertex and each face to an incident half-edge. More compact representations, such as [16], can be obtained by storing only the mapping between opposite half-edge, optionally the mapping from each vertex to an incident half-edge, along with the element connectivity. The half-edge concept was generalized to half-faces (e.g., [16,17]) for volume meshes where half-faces refer to the oriented faces within a cell. These basic half-edge and half-face data structures are simple and are restricted to oriented, manifold meshes (with or without boundary) in 2-D and 3-D, respectively.

In [1], an efficient, compact and general array-based half-facet (AHF) mesh data structure with support for mixed-dimensional meshes, which may be non-manifold and/or non-oriented was proposed. The core object of AHF is *half-facet* as defined previously and is represented as an *implicit entity*. The concept of *sibling half-facets* unifies the half-vertex, half-edge, and half-face data

Please cite this article in press as: Ray N, et al. Array-based, parallel hierarchical mesh refinement algorithms for unstructured meshes. Computer-Aided Design (2016), http://dx.doi.org/10.1016/j.cad.2016.07.011