CONF-9509183--2
SAND95-1755C

The Design of a Parallel Adaptive Paving All-Quadrilateral Meshing Algorithm*

Timothy J. Tautges
Randy R. Lober
Courtenay Vaughan
Joseph Jung

RECEIVED
AUG 17 1995
OSTI

Sandia National Laboratories
PO Box 5800, Albuquerque, NM, 87112

ABSTRACT

Adaptive finite element analysis demands a great deal of computational resources, and as such is most appropriately solved in a massively parallel computer environment. This analysis will require other parallel algorithms before it can fully utilize MP computers, one of which is parallel adaptive meshing. A version of the paving algorithm is being designed which operates in parallel but which also retains the robustness and other desirable features present in the serial algorithm. Adaptive paving in a production mode is demonstrated using a Babuska-Rheinboldt error estimator on a classic linearly elastic plate problem. The design of the parallel paving algorithm is described, and is based on the decomposition of a surface into "virtual" surfaces. The topology of the virtual surface boundaries is defined using mesh entities (mesh nodes and edges) so as to allow movement of these boundaries with smoothing and other operations. This arrangement allows the use of the standard paving algorithm on subdomain interiors, after the negotiation of the boundary mesh.

INTRODUCTION

With the arrival of faster computers, more sophisticated finite element analysis numerical methods, and many other factors, adaptive finite element analysis has become possible. Because the computational resources demanded by adaptive analyses are so great, much research has been done to move the analyses to massively parallel (MP) computers [1][2]. There are many tools required by an adaptive analysis loop, for example error estimators and adaptive mesh generation. In order to take full advantage of the MP environment, these other algorithms as well as the adaptive FEA code should operate locally. The topic of this paper is the design of a massively parallel adaptive surface meshing algorithm based on the paving algorithm.

The paving algorithm computes an all-quadrilateral mesh for 2D or 3D surfaces [3]. This algorithm was modified to perform adaptive meshing as part of an earlier effort [4], but this effort was only a proof of concept for adaptive meshing. The purposes of this work are two-fold: First, to develop a production-quality adaptive paving capability that can be used in an iterative adaptive analysis loop, and second, to develop a parallel adaptive paving

*This work was supported by the U.S. Department of Energy under contract DE-AC04-76DO00789.

DISCLAIMER

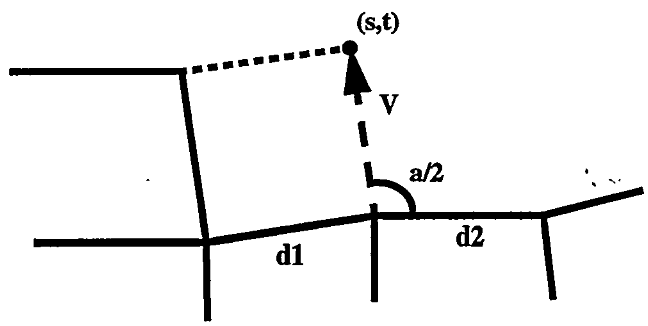
capability which retains the "good" characteristics of paving but which can operate in an adaptive analysis loop on a MP computer.

This paper is organized as follows. Section 2 describes the development of production-quality adaptive paving within the CUBIT mesh generation toolkit [5] and its operation with a parallel adaptive analysis loop. Section 3 describes the parallel paving algorithm. Section 4 describes briefly the parallel adaptive analysis loop we envision, and finally Section 5 gives conclusions and plans for future work.

ADAPTIVE PAVING

Previous adaptive meshing algorithms typically subdivide or combine elements locally to get elements with the desired size [6]. Adaptive paving seeks to remesh the surface based on a field function containing target edge lengths. This method eliminates the need for modeling constrained nodes in the analysis codes, and also increases the number of degrees of freedom added with each added element. These characteristics come at the cost of increased adaptive meshing time.

Paving operates by projecting quad elements along an advancing front (see Figure 1). In



Normal Paving:

$$|V| = \frac{0.5\,(d1 + d2)}{\sin\,(a/2)}$$

Adaptive Paving:

$$|V| = F\,(s, t)$$

Figure 1. Projecting a new element in paving, and projection distances used in normal and adaptive paving.

normal paving, the projection distance is determined by the lengths and relative angles of the surrounding mesh edges. In adaptive paving, this projection length is simply adjusted to be the value of the target size in the neighborhood of the mesh node being created to form the next quad element. In addition, the smoothing algorithm used during the paving process weights edge lengths toward the target edge length, rather than using a more typical length-weighted laplacian smooth.

As an example of an adaptive analysis for a solid mechanics problem, consider the classic problem of a hole in a linearly elastic plate subjected to uniaxial far-field tension. This problem was performed using the Babuska-Rheinboldt error estimator [6]. Based on this error estimate and the theoretical rate of convergence of the energy norm for the bi-linear elements used, the required element sizes were calculated and passed to the paving algorithm. The sequence of meshes obtained is shown in Figure 1. The maximum error in mesh 1 is 0.30%, which is the error in the strain energy as a percentage of the average strain energy per unit volume in the problem. This starting mesh has 145 nodes and 122

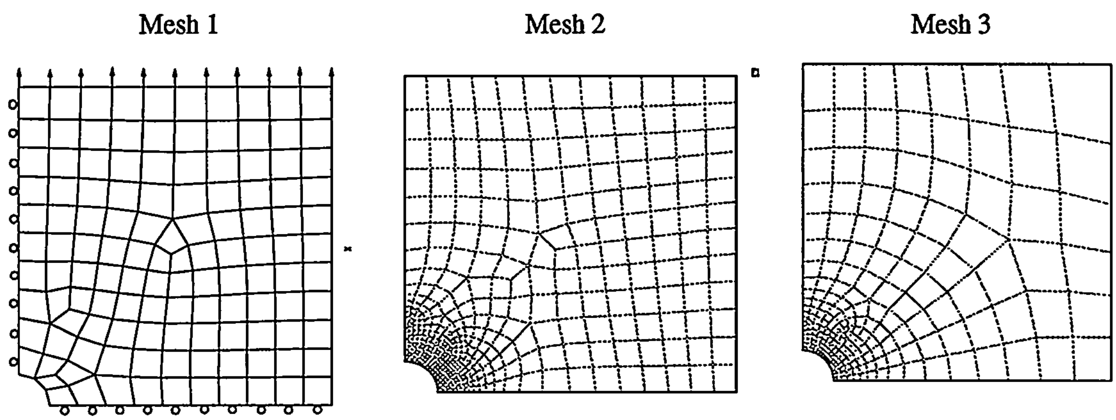


Figure 2. Three iterations of an adaptive analysis and remeshing loop.

elements. Mesh 2 was created with the objective of obtaining a error no greater than a target of 0.05%. In fact, the maximum error for that mesh is 0.036%, which was obtained with 634 nodes and 441 elements. An additional analysis was performed to determine if the adaptive procedure could obtain a more optimized mesh. The resulting mesh 3 has a maximum error of 0.045% with 739 nodes and 216 elements.

PARALLEL ADAPTIVE PAVING

Paving is advancing front method which is boundary sensitive (i.e. it places poorly-connected elements away from the boundary) and orientation insensitive (i.e. rotating the part and remeshing produces a similar quality mesh). These two characteristics make paving an inherently serial algorithm. The goal of this research is to design a parallel paving-type algorithm which retains these beneficial characteristics.

The parallel adaptive paving algorithm will be described in two parts. First, a non-adaptive parallel algorithm will be described. Then, the data communication necessary to allow an adaptive algorithm will be discussed.

The parallel paving algorithm is based on the decomposition of the surface to be paved into subdomains. Ideally, these subdomains are the same ones used in the parallel finite element analysis. Subdomains and their boundaries are represented using "virtual" geometry entities (vertices, curves, surfaces); the spatial position of these entities is evaluated using underlying normal geometry, while their topology is defined using mesh entities. For example, in Figure 3) a surface S1 is bounded by curves C1 and C2 and vertices V1 and V2. This surface is decomposed into two subdomains, s1 and s2, which can each be bounded by a combination of "normal" and "virtual" geometry. In Figure 3, s2 is bounded by normal geometry entities C2, V1 and V2, and virtual entities c2-c4 and v1-v2. The spatial position of each virtual entity is constrained to an underlying normal geometry entity. So, for example, v1 is constrained to lie on C1, which means it can move only along C1. The underlying geometry for a virtual entity can be either a higher order entity, as C1 is for v1, or can be the same order, as C1 is for c2. The topology of a virtual entity is defined entirely using mesh entities; that is, virtual vertices are defined using a node, curves are

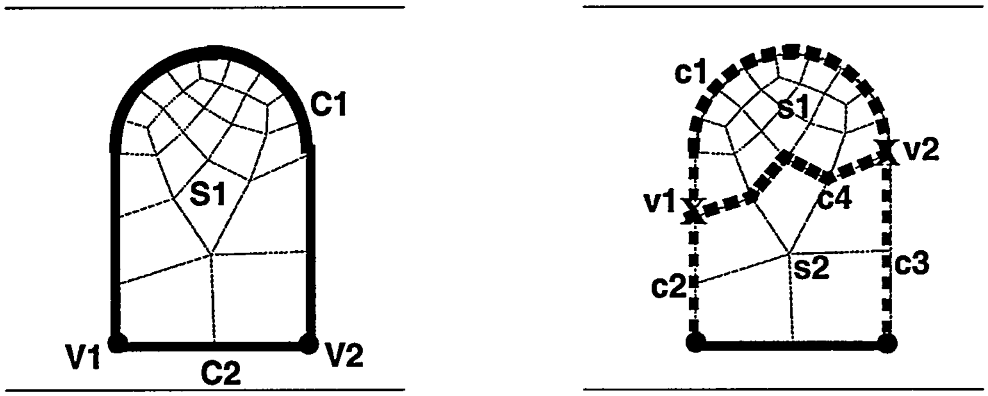


Figure 3. Comparison of geometry entities defining a "normal" surface (left), and "virtual" geometry entities defining the subdivided surface (right). Normal geometry entities are denoted with S (surface), C (curve) and V (vertex), while virtual geometry entities use s (surface), c (curve) and v (vertex).

defined as a series of mesh edges, and surface are defined using a group of mesh faces. It is assumed here that these faces are all contiguous in space.

Defining virtual geometry in this manner allows us to use almost the identical meshing algorithm for subdomains as is used for normal geometries. It also allows the virtual subdomain boundaries to move as a result of smoothing or other operations, which removes any constraint that would be an artifact of the parallel solution procedure. Finally, defining virtual geometry topology using mesh entities simplifies the movement of elements between subdomains for load balancing.

Using the above definition for geometry, the parallel meshing algorithm consists of first cooperatively meshing the subdomain boundaries, then independently meshing subdomain interiors. The boundary meshing step will involve communication between neighboring processors, which should be minimized for an efficient algorithm. Two methods for meshing these boundaries are being considered, one where a single processor "owns" each virtual curve and is responsible for its meshing, and another where all sharing processors perform their own meshing of that boundary using a deterministic algorithm.

PARALLEL ADAPTIVE ANALYSIS LOOP

As was mentioned in the Introduction, parallel adaptive meshing is only one part of an overall adaptive loop. A parallel dynamic adaptive loop would consist of the following steps:

- Start-up (problem decomposition, processor initialization, data loading)
- Adaptive remeshing
- Remapping of state variables from old mesh to new mesh
- Load balancing
- Finite element analysis

where all but the last step would be repeated several times in the problem. As was stated earlier, the parallel adaptive remeshing algorithm is only one of several algorithms which must be in place before true dynamic adaptive finite element analyses can be done effectively on MP computers.

CONCLUSIONS

One key step in dynamic adaptive finite element analyses on MP computers is the remeshing step. This step should be executed locally on the MP machine, to avoid the cost and complexity of generating the meshes remotely.

The paving algorithm has proven to be very capable of doing adaptive mesh generation. This capability has been demonstrated in a production mode by using the Babuska-Rheinboldt error estimator in combination with adaptive meshing to converge on the energy norm for a classic linear elastic plate problem. The error for this problem was reduced from 0.30%to 0.045%, while arriving at an optimum mesh size of 739 nodes and 216 elements.

The design of the parallel adaptive paving algorithm has as a goal to retain the "good" characteristics of paving, namely boundary sensitivity and orientation insensitivity, while executing in an MP environment. This design is focused on the use of "virtual" geometry, which fits in rather well with the overall parallel adaptive analysis process. The surface to be meshed is partitioned into subdomains which correspond to a set of contiguous mesh faces (ideally this partition is identical to that used in the parallel finite element analysis). After cooperatively meshing the subdomain boundaries, each processor is able to mesh its subdomain interior independently.

The design of this algorithm has been completed, and is now in the implementation phase. Datastructures for the virtual geometry, decomposition algorithms, and subdomain meshing have all been implemented in a serial environment. Our plans are to proceed with the implementation of subdomain boundary meshing in a parallel environment, after which meshing of the subdomain interiors should be trivial. Parallel mesh smoothing algorithms will then be explored.

REFERENCES

1. Karen D. Devine, Joseph E. Flaherty, "A Parallel Adaptive hp-Refinement Finite Element Method with Dynamic Load Balancing for the Solution of Hyperbolic Conservation Laws", submitted to *Appl. Numer. Maths.*, May, 1995.

2. C.L. Bottasso, H.L. deCougny, M. dindar, J.E. Flaherty, C. Ozturan, Z. Rusak, M.S. Shephard, "Compressible Aerodynamics Using a Parallel Adaptive Time-Discontinuous Galerkin Least-Squares Finite Element Method." *AIAA* Paper 94-1888, *12th AIAA Appl. Aerodyn. Conf.*, June 20-22, 1994, Colorado Springs, CO.

3. Ted D. Blacker, Michael B. Stephenson, "Paving: A New Approach To Automated Quadrilateral Mesh Generation", *Int. J. Num. Meth. Engr.*, 32, 811-847 (1991).

4. Ted D. Blacker, Joseph Jung, Walter R. Witkowski, "An Adaptive Finite Element Technique Using Element Equilibrium and Paving", ASME 90-WA/CIE-2 (1990).

5. Ted D. Blacker et al., "CUBIT Mesh Generation Environment, Volume 1: User's Manual", SAND94-1100, Sandia National Laboratories, Albuquerque, New Mexico, May 1994.

6. K. Clark, J.E. Flaherty, M.S. Shephard, Eds., *Appl. Numer. Maths.*, 14 (1994), special issue on Adaptive Methods for Partial Differential Equations.

7. I. Babuska and W. C. Rheinboldt, 'A-posteriori error estimates for the finite element method', *Int. J. Num. Meth. Eng.*, 11, 1597-615, 1978.