Chapter 1

COMPONENT-BASED ADAPTIVE MESH CONTROL PROCEDURES

Mark S. Shephard
Scientific Computation Research Center, Rensselaer Polytechnic Institute
shephard@scorec.rpi.edu

Eunyoung Seol
Scientific Computation Research Center, Rensselaer Polytechnic Institute
seole@scorec.rpi.edu

Jie Wan
Scientific Computation Research Center, Rensselaer Polytechnic Institute
jwan@scorec.rpi.edu

Andrew C. Bauer
Scientific Computation Research Center, Rensselaer Polytechnic Institute
acbauer@scorec.rpi.edu

Abstract In this paper we demonstrate the use of loosely-coupled adaptive loops created from already available software components. Using interoperable functions, we demonstrate the effectiveness of the adaptive loops through two examples, an electromagnetics simulation and a metal forming simulation. Both are completely automated and give accurate results.

Keywords: adaptive loop, mesh, model, field, mesh adaptation, error estimation

1. Introduction

A large number of methods and associated analysis codes for the numerical solution of partial differential equations are in common use today. Although these analysis codes are capable of providing results to the required levels of accuracy for many classes of problems, the ability to provide these predictions is not automatically controlled by the analysis code, but instead is a strong function of the input information provided. This deficiency is further complicated by the fact that most of the effort required to execute an analysis is associated with the generation of the input, and that substantial expertise and training is required to successfully define this input to provide reliable results. Efforts to address these problems are focused on automatically constructing analysis code input (e.g., [6, 18]) and employing the development of adaptive analysis procedures [4, 21]. The ultimate results of these efforts will be the production of automated adaptive analysis software.

Commercial software is available that can accept a general problem definition and automatically generate the analysis code input. Such tools are dramatically reducing the time and effort required to perform analyses. However, the ability to automatically generate input for and execute an analysis does not ensure that results to the level of accuracy needed are obtained. For the current discussion it is assumed that there is sufficient *a-priori* information to know that the results of interest can be obtained by solving sets of partial differential equations. In this case, the errors that must be controlled are the discretization errors associated with using a finite-dimensional basis over a mesh. Although adaptive methods to control discretization errors are well known, their application is limited to some research and specific special purpose codes.

Consideration of the steps of an automated adaptive analysis process and the interactions of the components used in those steps helps to explain why adaptive analysis methods are not more common. The steps of a finite element based automated adaptive analysis system are:

1 **Create a general geometry-based problem definition** In terms of a non-manifold boundary representation solid models as supported by commercial CAD systems are well suited to provide the geometric domain definition. When supplemented with an attributing capability, general problem definitions can be completed [14].

2 **Create an initial mesh** Using an automatic mesh generator, meshes are generated directly from CAD representation [6], including procedures that provide flexible spatially-based control of the mesh [17].

3 **Perform analysis** Using a finite element procedure which constructs the element level contributions, assembles them into a global system and solves the global system. A wide variety of codes have been developed to support this process. In most cases these codes operate on a mesh-based problem specification and employ data structures for a fixed mesh.

4 **Postprocess analysis results** Using error estimation and correction indication procedures. These procedures are responsible for determining useful estimates of the discretization error on the current mesh and where and how to modify the mesh to most effectively reduce the errors to an acceptable level [2, 3, 21].

5 **Improve mesh** By altering it to obtain the mesh sizes indicated by the correction indication procedures. These mesh improvements can be carried out by regenerating a new mesh, or by the modification of the existing mesh. The procedures used in the current paper are based on generalized mesh modification procedures [12, 13, 20].

6 **Repeat the steps 3-5** until the desired level of accuracy is obtained.

Steps 1 and 2 interact with the geometry-based problem definition and generate the mesh-based problem definition operated on by the finite element analysis procedures. Because the only interaction between these components and the analysis code is the output from the two components, the introduction of the automatic mesh generation is straightforward. In the common case of performing the analysis on a single mesh, the analysis procedures (performed in step 3) operate on the mesh-based problem definition only. The proper execution of steps 4 and 5 require interactions with both the mesh-based and geometry-based problem definitions (e.g., improving the geometric approximation as the mesh is refined and associating the appropriate traction values to newly defined boundary nodes). In addition, as a result of executing step 5 the mesh is modified which must be reflected in the mesh-based problem description used in the next analysis. The complexity of dealing with these interactions has dramatically slowed the introduction of adaptive analysis methods into practice.

One approach to address the mismatch between the needs of fixed mesh and adaptive mesh analysis procedures is to alter the analysis code to directly interact with the adaptive analysis processes. The advantage of this approach is that the resulting code can minimize the total computation and data manipulation time required (which appears to be important for transient adaptive analyses using explicit time stepping [15]). Although it is possible to construct such adaptive codes that interact directly with the geometry-based problem specification [5], the modification of an existing fixed mesh code requires the introduction of entirely new data structures thus forcing an extensive rewrite of the code. The expense and time required to do this is large and in most cases considered prohibitive, particularly for well established codes.

The alternative approach for use with existing fixed mesh analysis codes, which is the focus of this paper, is to leave the analysis code unaltered and to use a set of interoperable information communication tools [23] to control the flow of information between the set of components used for creating the problem definition (step 1), mesh generation (step 2), error estimation and correction indication (step 4), and mesh improvement procedure (step 5). Section 2 overviews the components and information flow between components required in an automated adaptive process. Section 3 will demonstrate the effectiveness of this approach for the construction of automated adaptive loops using existing analysis codes, one for electromagnetic field simulations and one for large deformation forming problems. In both these examples the finite element systems are solved using implicit methods and it is found that the added information transfer cost associated with constructing the adaptive loop with a set of components that are external to the analysis code is small.

2. Integration of Interoperable Components

The steps of the adaptive loop need to be integrated such that software components can properly share information to be able to perform the steps. The components necessary for performing the adaptive loop which interact with more than one adaptive loop steps are:

- **Geometry interface** is a high level topological model of the domain which supports the integration to multiple CAD systems. The interoperable API of the modeler enables interactions with mesh generation, mesh modification and analysis code input construction to obtain all domain geometry information needed [5, 19].

- **Mesh interface** provides the services for storing and modifying mesh data [5, 7, 19] during the adaptive process. The current procedures used the Algorithm-Oriented Mesh Database (AOMD) [16] to support these processes.

- **Field interface** provides complex functions to obtain the solution information needed for error estimation and to support the transfer of solution fields as the mesh is adapted [5, 19].

The modules used are the CAD modeler, analysis engine, mesh generator and mesh modification procedures. The geometry interface is used in all steps except step 3. The mesh interface interacts' with all steps except for the first one. The field interfaces interacts with steps 3 through 5. The analysis engine is only used during step 3, the mesh generator is only used during step 2 and the mesh modification procedures are only used during step 5. The analysis engine, mesh generator and mesh modification procedures do not interact among each other and thus are easily replaceable using the geometry, mesh, and/or field interfaces. With a well-defined interoperable interface to the geometry, mesh and field modules, various adaptive loops are easily built through vertical integration of the other components. The interoperable components used here are being designed to be compliant with the TSTT interface [23] as they evolve so as to provide *interchangeability* that allows horizontal integration across a number of different tools that provide similar functionalities.

3. Examples

Given a flexible set of adaptive error control components, adaptive loops have been built around two fixed mesh finite element codes. The first is a frequency domain electromagnetics simulation code Omega3P [11] developed at the Stanford Linear Accelerator Center (SLAC). The second is a commercial metal forming simulation code, DEFORM-3D$^{\mathrm{TM}}$ [8] where the adaptive loop tracks the evolving geometry.

3.1 Adaptive Loop for Accelerator Design

SLAC's eigenmode solver Omega3P, which is used in the design of next generation linear accelerators, has been integrated with adaptive mesh control [12, 13] to improve the accuracy and convergence of wall loss (or quality factor) calculations in accelarating cavities. The simulation procedure consists of interfacing Omega3P to solid models, automatic mesh generator, general mesh modification, and error estimator components to form an adaptive loop as depicted in Figure 1.1.

The accelerator geometries are defined as ACIS$^{\mathrm{TM}}$ solid models [1] and physical parameters required by the simulation are associated with geometric model entities. Using functional interfaces between geometric model and meshing techniques [6], the automatic mesh generation tools of Simmetrix$^{\mathrm{TM}}$ [20] generates an initial mesh. After Omega3P

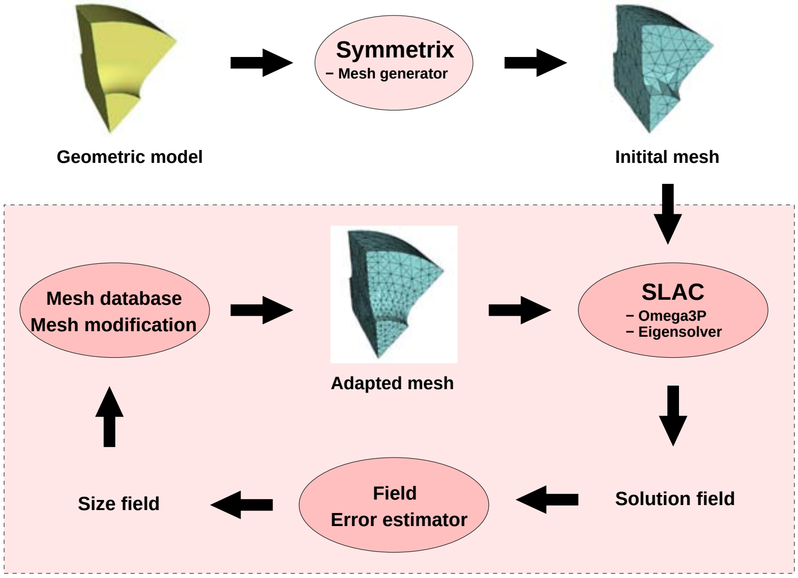


Figure 1.1. Framework of adaptive loop for accelerator design

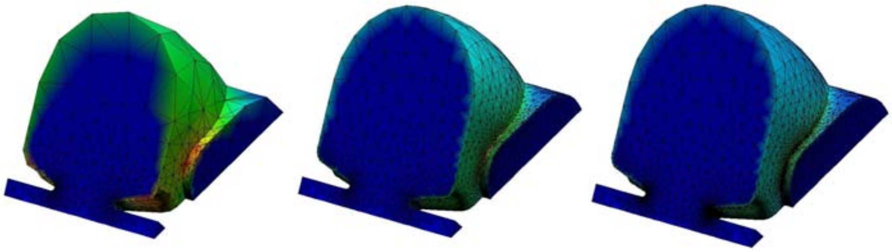


Figure 1.2. Mesh and wall-loss distribution for 3 adaptive steps

calculates the solution fields, the error indication procedure determines a new mesh size field and the mesh modification procedures modify the mesh to generate a new mesh for the next execution of Omega3P. This iterative procedure repeats until the desired accuracy is reached.

The adaptive procedure has been applied to a Trispal 4-petal accelerating cavity. Figure 1.2 shows the mesh and wall loss distribution on the cavity surface for three adaptive steps with an increasingly denser mesh in the area of highfield concentration (from left to right). The procedure has been shown to reliably produce results of the desired accuracy for approximately one-third the number of unknowns the previous user controlled procedure produced [9].

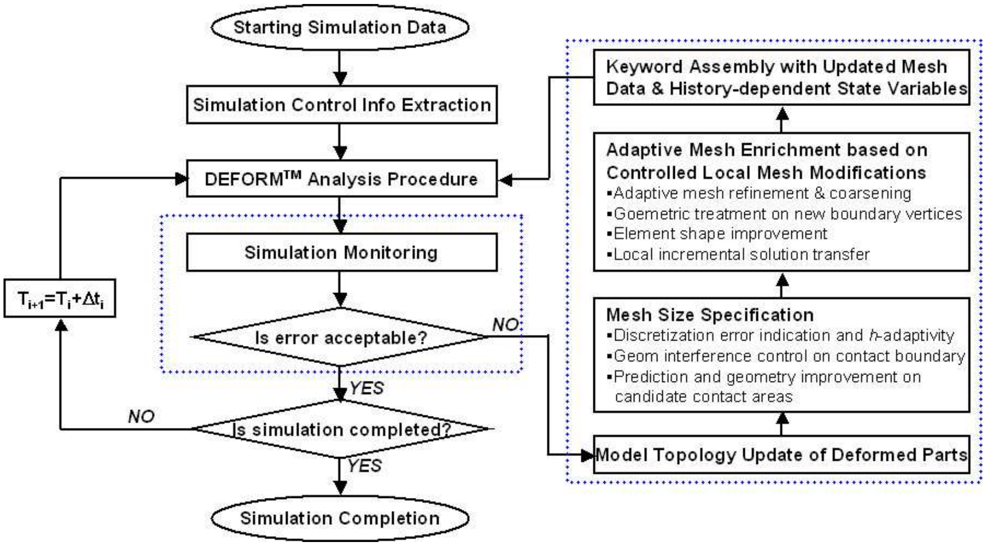


Figure 1.3. Automated adaptive forming simulation process.

3.2 Metal Forming Simulation

In 3D metal forming simulations the deformable parts undergo large plastic deformations that result in major changes in the analysis domain geometry. The meshes of the deforming parts typically need to be frequently modified to continue the analysis due to large element distortions, mesh discretization errors and/or geometric approximation errors. In these cases, it is necessary to replace the deformed mesh with an improved mesh that is consistent with the current configuration [8, 10, 24]. History dependent field variables also need to be accurately transferred from the old mesh to the new mesh [8, 10, 24]. Remeshing generates an entirely new mesh even though there might be only a limited number of elements that need to be modified. To more effectively address the needed mesh updates and field transfers and to provide higher solution accuracy, a component-based adaptive mesh control procedure as depicted in Figure 1.3 was developed. Detailed discussions on the involved components are presented in [24].

A steering link manufacturing problem as shown in Figure 1.6 is investigated to demonstrate the developed capabilities. A total stroke of 41.7 mm is simulated. The allowed maximum geometric interference is 0.60 mm. The initial workpiece mesh consists of 6,765 mesh vertices and 28,885 mesh regions (Figure 1.5a). The simulation is completed with 20 mesh modification steps performed by adopting the automated adap-

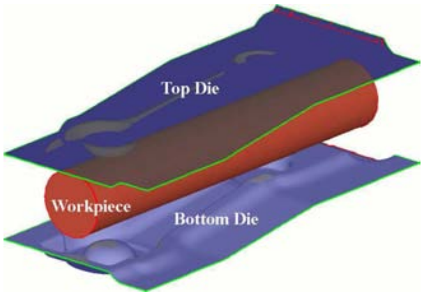
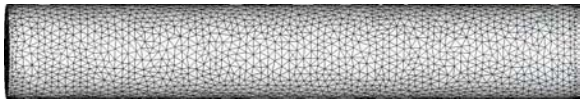
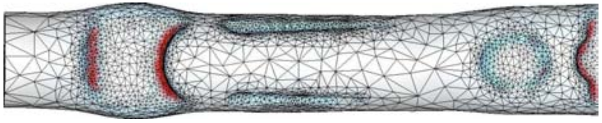


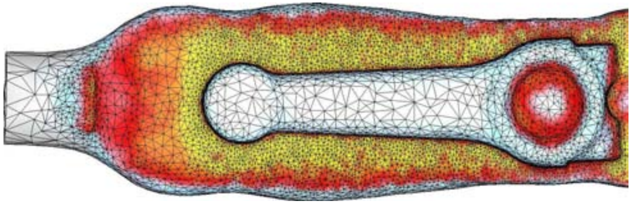
Figure 1.4. Setup of the steering link problem.



(a) Initial workpiece mesh (28,885 mesh regions)



(b) Stroke =27.6 mm (50,424 mesh regions)



(c) Final workpiece mesh (102,249 mesh regions)

Figure 1.5. Mesh adapted consistently with the effective strain profile.

tive mesh control procedure. The final mesh of the achieved workpiece consists of 23,525 mesh vertices and 102,249 mesh regions (Figure 1.5c). The workpiece mesh is adapted to control solution error and the geometric approximation. The effects of mesh adaptivity are demonstrated in Figure 1.5. The large elements as seen in the far left of these pictures are satisfactory because of the low strain gradient. In the regions of high strain gradient, smaller elements are needed to control the discretization error while fine mesh is needed near the contact boundaries to control the geometric approximation errors [8, 24].

The element quality of the workpiece mesh measured in terms of the maximum dihedral angles before and after the mesh modification steps

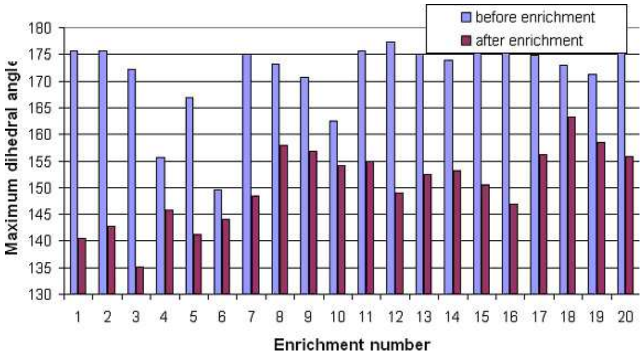


Figure 1.6. Element quality control through remeshings and mesh modifications.

is shown in Figure 1.6. It can be seen that throughout the mesh modification based simulation, the element quality of the workpiece mesh is effectively controlled through the element distortion monitoring and local mesh modification based shape improvement.

4. Closing Remarks

This paper has shown two automated adaptive loops which accurately simulate their corresponding physics. The adaptive loops are created using a set of interoperable components that link analysis codes with geometry-based problem definitions, automatic mesh generation, error estimation procedures and generalized mesh modification procedures. Other adaptive loops can be easily constructed using this approach.

References

[1] Spatial Inc. (2004) http://www.spatial.com/components/acis/.

[2] Ainsworth M, Oden JT (2000) *A Posteriori Error Estimation in Finite Element Analysis.* Wiley-Interscience, John Wiley & Sons.

[3] Babuška I, Strouboulis T (2001) *The Reliability of the FE Method.* Oxford Press.

[4] Bangerth W, Rannacher R (2003) *Adaptive Finite Element Methods for Differential Equations, Lectures in Mathematics* VIII Vol. 207. Birkhauser.

[5] Beall MW, Shephard MS (1999) An Object-Oriented Framework for Reliable Numerical Simulations. *Eng. w. Comp.* 15:61-72.

[6] Beall MW, Walsh J, Shephard MS (2004) A comparison of techniques for geometry access related to mesh generation. *Eng. w. Comp.* 20(3):210-221.

[7] Beall MW, Shephard MS (1997) A general topology-based mesh data structure. *Int. J. Numer. Meth. Eng.* 40:1573-1596.

[8] Fluhrer J (2004) DEFORM-3D$^{\mathrm{TM}}$ Versoin 5.0 User's Manual, Scientific Forming Technologies Corporation.

[9] Ge L, Lee LQ, Li Z, Ng C, Ko K, Luo Y, Shephard (2004) MS Adaptive Mesh Refinement for High Accuracy Wall Loss Determination in Accelerating Cavity Design. *Eleventh Biennial IEEE Conference on Electromagnetic Field Computation* Seoul, Korea, June 6-9.

[10] Kobayashi S, Oh S-I, Altan T (1989) *Metal Forming and the Finite Element Method*. Oxford University Press.

[11] Lee L-Q, et al. (2004) Solving large sparse linear systems in end-to-end accelerator structure simulations. SLAC-PUB-10320 January.

[12] Li X, Shephard MS, Beall MW (2002) Accounting for curved domains in mesh adaptation. *Int. J. Numer. Meth. Eng.* 58:247-276.

[13] Li X, Shephard MS, Beall MW (2003) 3-D Anisotropic Mesh Adaptation by Mesh Modifications. to appear in *Comp. Meth. App. Mech. and Eng.*.

[14] O'Bara RM, Beall MW, Shephard MS (2002) Attribute Management System for Engineering Analysis. *Eng. w. Comp.* 18:339-351.

[15] Remacle J-F, Flaherty JE, Shephard MS (2003) An adaptive discontinuous Galerkin technique with an orthogonal basis applied compressible flow problems. *SIAM Review* 45(1):53-72.

[16] Remacle J-F, Shephard MS (2003) An algorithm oriented mesh database. *Int. J. Numer. Meth. Eng.* 58:349-374

[17] Shephard MS (2000) Meshing environment for geometry-based analysis. *Int. J. Numer. Meth. Eng.* 47:169-190.

[18] Shephard MS, Beall MW, O'Bara RM, Webster BE (2004) Toward simulation-based design. *Fin. Elem. in An. and Des.* 40:1575-1598.

[19] Shephard MS, Fisher P, Chand KK, Flaherty JE (2003) Simulation Information Strucutures http://www.tstt-scidac.org/.

[20] Simmetrix Inc. (2004) Simulation Modeling Suite http://www.simmetrix.com/.

[21] Stein E, ed. (2002) *Error-Controlled Adaptive Finite Elements in Solid Mechanics*. J. Wiley & Sons.

[22] Thompson JF, Soni BK, Weatherill NP, eds. (1999) *CRC Handbook of Grid Generation*. CRC Press, Inc.

[23] http://www.tstt-scidac.org (2004).

[24] Wan J, Kocak S, Shephard MS (2005) Automated Adaptive 3-D Forming Simulation Processes. To appear in *Eng. w. Comp.*