UCRL-JC-126796
PREPRINT

3-D Finite Element Model Development for Biomechanics: A Software Demonstration

K. Hollerbach
A.M. Hollister
E. Ashby

This paper was prepared for submittal to the
Sixth International Symposium on Computer Simulation Biomechanics
Tokyo, Japan
August 21-23, 1997

March 1997

3-D FINITE ELEMENT MODEL DEVELOPMENT FOR BIOMECHANICS: A SOFTWARE DEMONSTRATION

K Hollerbach, Lawrence Livermore National Laboratory, Livermore, California,
holler@llnl.gov
AM Hollister , LSU Medical Center, Shreveport, Louisiana, *anne@www.ortho.lsumc.edu*
E Ashby, Lawrence Livermore National Laboratory, Livermore, California, *ashby3@llnl.gov*

INTRODUCTION

Finite element analysis is becoming an increasingly important part of biomechanics and orthopaedic research, as computational resources become more powerful, and data handling algorithms become more sophisticated. Until recently, tools with sufficient power did not exist or were not accessible to adequately model complicated, three-dimensional, nonlinear biomechanical systems. In the past, finite element analyses in biomechanics have often been limited to two-dimensional approaches, linear analyses, or simulations of single tissue types. Today, we have the resources to model fully three-dimensional, nonlinear, multi-tissue, and even multi-joint systems. The authors will present the process of developing these kinds of finite element models, using human hand and knee examples, and will demonstrate their software tools.

The process of developing finite element models (Fig. 1) begins with the acquisition of data that will be used to define the three-dimensional geometry of the joint tissues. These data can come from several imaging modalities, including CT and MRI. Three-dimensional data sets are acquired and segmented, i.e., each tissue type of interest to the modeler is labeled within the data set. From the segmented data, three-dimensional surfaces are calculated, and fully volumetric meshes (the geometric portion of the finite element model) are generated. In the finite element analysis, the tissue types described by the finite element models are assigned specific material characteristics, and the simulation is completed, with externally calculated boundary conditions defining the specific joint behavior (e.g., flexion of the joint due to flexor tendon action). Each of these steps is described in more detail below and will be demonstrated. All simulations are run on Silicon Graphics, Inc. workstations (Mountain View, California).

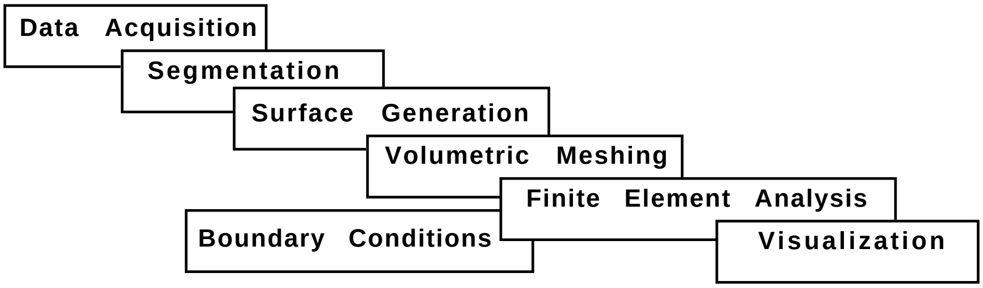

Figure 1: The process of three-dimensional finite element model development. The order of modeling steps is indicated by the overlapping boxes: Data are acquired and segmented; surfaces are generated from the segmentation results and subsequently meshed; the finite element analysis depends on the meshes, which represent the finite element model geometry, and on calculated boundary conditions; finally, finite element results are visualized.

DATA ACQUISITION

Accurate geometry is one key to successfully modeling joint behavior using finite element techniques. As a result, tissues, especially near the articular surfaces, must be defined with a high degree of spatial resolution. We use both CT and MRI data in our model development. Once the data are acquired, model development is independent of the imaging modality. As a result, only acquisition of bone surfaces from CT data will be discussed here.

Typically, scanners used in the medical field have a spatial resolution that is not acceptable for a precise definition of articular surfaces. Thus, an amputated hand was scanned (Fig. 2) with one of the industrial scanners that have been designed and constructed at the Lawrence Livermore National Laboratory (LLNL). The pixel size of the scanner is 150 microns, and is equal to the distance between CT slice planes. The scanner uses a high energy X-ray source and a scintillating screen lens coupled to a camera. The detector helps acquire 720 projections (2.5 Gigabytes) rapidly. The reconstruction step is computationally intensive and is performed using a parallelized Convolution Back-Projection algorithm.

SEGMENTATION

Segmentation of the imaged, three-dimensional data sets is the process of identifying tissues and their boundaries. When large data sets are used, this process is quite time consuming, and automating the procedure as much as possible becomes desirable. The authors have developed software that performs semi-automatic segmentation and allows as much user input as is required to refine the results.

Precisely identifying the boundaries of whole bones from very high resolution CT data is made difficult by the inhomogeneous trabecular structure of the bones. Despite the commonly accepted method of matching Hounsfield units to gray values in CT scans, simple thresholding of the data to accurately define bone surfaces is inadequate, as are edge detectors which produce a large number of spurious edges. Our approach relies on a simple model of the bone attenuation profiles, which is used to perform an initial, automated, coarse segmentation. Watershed lines, which are by construction located on gradient peaks and hence on sharp boundaries, improve the automatic segmentation results. This approach was satisfactory in most cases. However, human interaction will always be needed to refine the coarse segmentation. This kind of interaction requires computational tools, such as our Visu software (LLNL), that allow the user to visualize large data sets, manipulate the color map to display false colors, perform interactive thresholding, overlay the segmentation mask, and save the corrected results (Fig. 3).

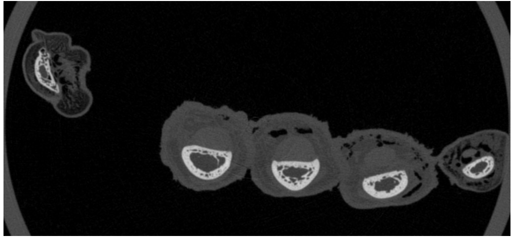

Figure 2: A sample of high resolution CT data, in a transverse cut through a human hand.

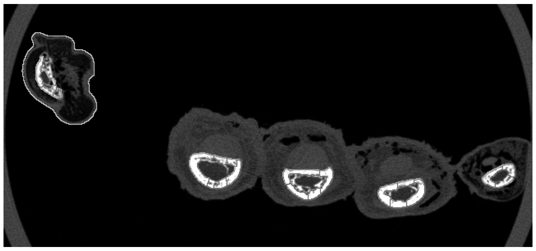

Figure 3a: Edges generated by automated segmentation.

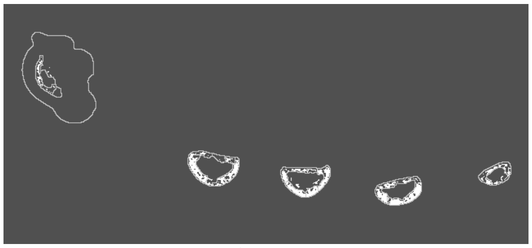

Figure 3b: Edges filled in by automated segmentation and partially manually corrected.

Figure 3c: Final mask resulting from semi-automated segmentation.

SURFACE GENERATION

Three-dimensional surfaces are generated directly from the masks that are the final product of the segmentation step. The authors have implemented the marching cubes algorithm (in standard use among computer graphics code developers) and use it as their primary surface generation tool. The algorithm generates a closed surface for each connected set of volumetric elements identified as a particular tissue. The surface is described by a set of triangles in three-dimensional space. One disadvantage of this method is the large number of triangles used to define the surface. When the number of triangles used exceeds that which is required to minimally but adequately define the surface structure, a decimation algorithm (General Electric Corp., Schenectady, New York) is applied to reduce the data (Fig. 4).

VOLUMETRIC MESHING

Most papers on mesh generation algorithms focus on volumetric tetrahedral meshes. The methods used generally rely on a subdivision algorithm of the volume, such as in the octree approach. A mesh is then built by triangulating each of the cells of the volume. Slightly changing the coordinates of the vertices helps smooth the mesh and improve its quality. However, tetrahedral meshes are not suited for the dynamic simulations required in our applications. Structural engineers prefer hexahedral meshes, which help speed up the convergence of the numerical algorithms. Automated hexahedral meshing is more challenging than tetrahedral meshing, since global topology must be taken into consideration from the start.

Our work relies on the TrueGrid (XYZ Scientific Applications, Inc., Livermore, California) meshing package. Hexahedral mesh generation was performed manually, to allow for fine control of mesh quality, especially near contacting surfaces. To form the volumetric mesh, the (originally block shaped) mesh is placed inside the bone, and its vertices and faces are projected in a multi-step process in such a way that each vertex lies at the intersection of the perpendicular cut plane and the outer edge of the closest radial surface, and each face approximates the original surface grid. With all vertices and faces in place, the internal nodes are then arranged to optimize the grid quality. The final result obtained is a high quality mesh that is suitable for finite element modeling. Diagnostic measures, such as orthogonality of the elements, may be applied to confirm the mesh quality.

BOUNDARY CONDITIONS

All simulations must be driven by boundary conditions of some kind. In finite element analysis, these boundary conditions typically consist of forces or displacements that are applied to the model. One example in biomechanics is a stress-strain model of a single tissue type, where a block of tissue may be stressed and the resulting strains calculated, based on some modeled set of material characteristics. An optimization algorithm may be applied to match the numerical parameters of the material model to experimental data. Alternatively, in whole joint modeling, assumptions may be made about the kinematic mechanism, and the analysis may be used to determine forces that drive the joints. The authors use separate experimental and modeling tools to determine net joint reaction and musculo-tendinous forces and use these calculations to drive the finite element analysis and to produce biomechanically relevant joint behavior. For modeling the hand, a biomechanics model (originally developed by Giurintano et al., 1995) is used, in which the user can assign any desired kinematic mechanism and can define the tendon paths that cross each joint. Tendon forces are calculated and used to drive the finite element joint simulation. In joints where tendons have not yet been modeled, external forces are applied.

FINITE ELEMENT ANALYSIS

The finite element simulations are performed using the NIKE3D code (LLNL). NIKE3D is a nonlinear, implicit, three-dimensional finite element code developed for the purposes of studying dynamic, finite deformations. Articular contact algorithms are based on a master-slave approach that provides general multi-body contact capability, including sliding with contact, gaps, and friction, as needed. Contact algorithms use the penalty method, where a penalty is generated when interpenetration between contacting surfaces is calculated.

In the hand model example, the joints are placed in a configuration appropriate to a particular grasp, and the corresponding tendon forces are applied. Since the hand is frequently used statically, as in grasp, these simulations can provide valuable information on joint loading at and near the articular surface during many common activities. In addition, we are able to simulate fully dynamic articulation of the joints in three dimensions. In these simulations, the motion is determined entirely by the tendon forces, the ligament forces constraining the joint, and the joint geometry. The kinematic mechanism is not arbitrarily determined by the analyst, but is determined by the three-dimensional geometry of the articular surfaces. Material properties for the joint tissues are taken from the literature. Results obtained from the simulations include the kinematics, which can be compared against experimental data, the articular surface stresses, and the ligament stresses.

VISUALIZATION

Joint kinematics and tissue stresses calculated by the finite element code are visualized on a workstation, using the Griz (LLNL) package (Fig. 5). Griz is an interactive program for visualizing finite element analysis results on three-dimensional, unstructured grids. Griz provides techniques such as isosurface display, cutting planes, and vector data display. It can also animate simulation results and store animation frames for video production. In the biomechanics simulations, Griz is used primarily to animate the joint kinematics. On Silicon Graphics, Inc. workstations, high quality graphics produce images, with color maps that easily distinguish regions of high stress in the soft tissues, thereby providing the user with an intuitive, yet quantitative presentation of analysis results.

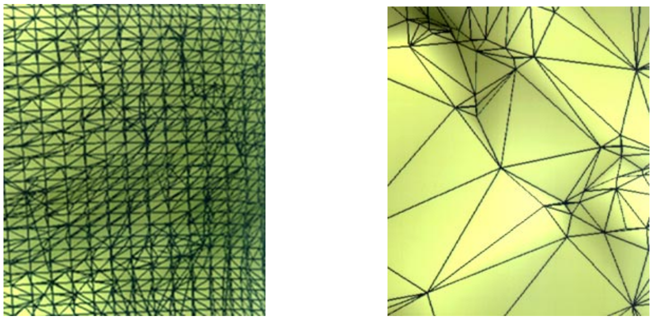

Figure 4 (a)

Figure 4 (b)

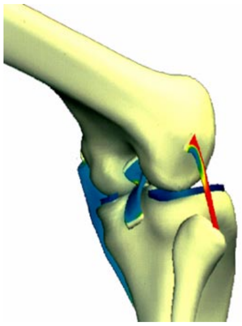

Figure 5

Figure 4: Close-up of bone surface generated from CT segmentation by marching cubes algorithm, before (a) and after (b) the decimation process.

Figure 5: Griz image showing knee ligament stresses during flexion.

DISCUSSION

The authors have developed and applied computational tools to human joint biomechanics and to prosthetic implant analysis. These tools can also be applied to other biomechanic systems, provided that imaged data, material characteristics, and boundary conditions are available. Since human anatomy and, therefore, human joint biomechanics are inherently three-dimensional, nonlinear processes, they cannot be adequately modeled in two-dimensional analyses. Modeling software developed and used by the authors can provide the tools that have the potential to produce more accurate results than analyses that have previously been attempted. Demonstrations of the Visu, TrueGrid, and NIKE3D/Griz codes will be presented.

REFERENCES

Hollerbach K, et al. (1996) "Modeling the biomechanics of human joints and prosthetic implants", UCRL - TB - 118601 - Rev. 2. Lawrence Livermore National Laboratory.

Giurintano DG, Hollister AM (1995) "Force Analysis of the Thumb for a Five-link System" *Med Eng Phys* **17**:297-303.

ACKNOWLEDGMENTS
This work was supported in part by the US Department of Energy under contract # W-7405-ENG-48 with the Lawrence Livermore National Laboratory.

Technical Information Department **• Lawrence Livermore National Laboratory**
University of California • Livermore, California 94551