ARL-MR-0890 • JUN 2015

US Army Research Laboratory

Progress in Finite Element Modeling of the Lower Extremities

by Adam Sokolow

NOTICES

Disclaimers

ARL-MR-0890 ● JUN 2015

US Army Research Laboratory

Progress in Finite Element Modeling of the Lower Extremities

by Adam Sokolow
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
June 2015

2. REPORT TYPE
Memorandum Report

3. DATES COVERED (From - To)
October 2014–February 2015

4. TITLE AND SUBTITLE
Progress in Finite Element Modeling of the Lower Extremities

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Adam Sokolow

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-WMP-B
Aberdeen Proving Ground, MD 21005

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-MR-0890

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Human body modeling efforts for the purpose of Soldier protection need to address the current threats as well as have a vision for the future. Modeling the human body is a challenging endeavor due to its geometric complexity, numerous interacting layers, rich anisotropy, and wide variability. Developing a model for predictive injury capability, therefore, needs to be versatile and flexible to address different levels of modeling complexity. The vision presented here surrounds a flexible mix-and-match assembly approach. This assembly process has the capability to take a collection of source body part meshes that may have different resolutions, deform the meshes based on the individual to be simulated, and posture them into different positions so that the end result can be exported into multiple finite element solvers. The primary focus of the present effort is the mounted Soldier's response to accelerative loading from underbody blast events. Many of the challenges in modeling the human body remain the same for applications such as the response of dismounted Soldiers. This report presents a progress report of our current efforts and documents some major improvements to the lower leg model with a vision of the future in mind. We also introduce significant details regarding an assembly architecture that is currently under development.

15. SUBJECT TERMS
lower extremities, FEM, accelerative loading, biological variability, solver independent

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
Unclassified	Unclassified	Unclassified	UU	22

19a. NAME OF RESPONSIBLE PERSON
Adam Sokolow

19b. TELEPHONE NUMBER (Include area code)
410-306-2985

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

Developing a finite element model of the human body that has predictive injury capability presents many challenges, the first of which is identifying the types of conditions where injury prediction is needed. Our focus is on lower-leg injuries due to an underbody blast event beneath a vehicle. During such an event, the explosive gases and soil impart momentum to the vehicle hull. Deformations of the hull affect the floor plate, which can be in direct contact with the vehicle occupants. The floor plate can also undergo rapid accelerations that result in occupant injury.

Injuries that result from these events are often dubbed accelerative injuries, although this nomenclature is misleading in that it combines 2 types of injuries under the same terminology. The 2 cases separate into what can be considered a shock perspective and a structural perspective. To model these events properly requires 2 models to be developed by 2 classes of injury mechanisms: one model that can adequately resolve wave propagation and a second that is suitable for accumulated strain. While both are necessary in the long run, the current emphasis is placed on a model that is adequate to predict injuries due to accumulated strain. This type of model can be generated for the entire body and has a more immediate impact on improvements in Soldier protection.

For the shock case, wave propagation within the tissues that make up the structure is incredibly important, as well as understanding the full constitutive response and the types of interfaces a propagating wave might encounter. This case would be most concerned with the initial impact of the floor plate with the occupant. After the initial strike, a shock wave, or a large and rapidly changing impulse, will propagate through the substructures and across the material interfaces. The complicated interplay that results from anisotropy, rate-dependence, fracture, and interface dynamics can all play a critical role in injury. Wave propagative models should be developed at the component level and take into consideration the numerous layers that are present. To account for the types of injuries seen in the calcaneus, such component models would likely require multiple elements through the flesh, fat, muscles, and tendons of the heel, and account for the grading of the cortical to trabecular bone of the calcaneus. While we currently do not have accurate geometries for all these layers, we may have them in the future. Thus, our modeling effort needs to be adaptable to allow for including refined meshes in future work.

For the structural perspective, the model is most concerned with accumulated deformation imposed on the load-bearing structures. This accumulated deformation might result from the somewhat simple case of compression along the loading path, or the more complicated case where inertial effects result in bending and subsequent injury, e.g., the distal tibia motion results in bending of the tibia rather than the tibia rotating about the knee joint. Although still an active research question, intuitively it seems that calcaneal fractures are more likely to result from wave propagative effects whereas tibial fractures result from accumulated strain. Namely, rapidly varying propagating waves may fracture the microstructure of the calcaneus, while the tibia fails as it bends like a beam or under compressive loads. Either or both of these cases might be necessary to model when developing a fully predictive model. However, the level of detail required and the refinement of the mesh is vastly different for the 2 cases.

The second challenge in modeling the human is accounting for variability when necessary. In the context of underbody blast events, there is a large amount of variability in the loading that reaches the occupant. These events occur in theater, and there is a wide range of postures that the occupants can be positioned in during an event. There are also large variations between Soldiers, some of which are readily visible like height and weight, while others are more subtle, e.g., femur length, bone structure and density, or completely unknown, e.g., prior injury. The materials themselves also have variability, e.g., cortical bone from the skull is structurally different from the femur and therefore exhibits different material characteristics. Thus, a true statistical modeling approach needs to incorporate posture and structural differences, as well as variations in the materials. Often these are not independent characteristics and thus a larger understanding of covariates of the system is necessary. However, the degree to which any type of variability matters in the context of predicting injury is an open research question.

These challenges and the unknowns introduce numerous practical concerns, including version control for the multitude of parts—whether it is material model, mesh resolution, contact, connectivity, or posturing. Practical issues exist like numbering parts uniquely, producing documented input files for the solvers, and facilitating presentation quality images. These challenges also introduce numerous research concerns such as parameterizing structural features like bone thicknesses or bone dimensions, or material features like the defined directionality of tissue anisotropy. In this report, both the practical and the research concerns are addressed by centralizing the efforts into a single assembler.

This report is organized as follows. First, a summary of the meshing efforts is presented, as it represents a substantial improvement from the previous version of the lower leg. The remaining sections outline the assembler program starting with an overview and then continuing into a description of its current capabilities that include model management, controlled generation of beam elements, optional subdivision of parts, and model morphing capabilities.

2. Hexahedron Meshes

A typical criticism of previous modeling efforts has been that the model is built on tetrahedron elements. These types of elements behave too stiffly at large strain, and do not allow for accurate shock wave propagation. In an effort to improve both accuracy and simulation run time, a major effort has been made to develop hexahedral meshes.

The source geometries that we are working with are from Zygote Media Group, Inc. in a sterolithographic (STL) format. This format is inherently different from the typical computer-aided design (CAD) file because it is a surface mesh and not B-spline based geometry. The full capability of a meshing software is rarely available, e.g., CUBIT (Sandia National Laboratory) treats geometries in a CAD engine as a separate engine from STLs and they cannot be mixed. The stability of CUBIT is also highly questionable when using STL files and repeated crashes during mesh development are common. Fortunately, these structures can be meshed using a blocking scheme in meshing software that has this capability. We use ICEM CFD (ANSYS) meshing software to produce highly controlled and parameterized meshes of the bones, muscles, and flesh.

Due to the complexity of the geometries, the general modeling approach is to treat the bones, muscles, and skin as separate components and mesh them individually. Meshes are then joined at a later stage through combinations of contact definitions and 1-dimensional (1-D) structural elements, and where appropriate, node merging. These choices are all flexible and can be changed in later versions.

Hexahedral meshes were developed for 23 bones in the right-lower extremity, including the pelvis, femur, patella, tibia, fibula, talus, calcaneus, cuboid, navicular, the cuneiforms, metatarsals, and phalanges. Figure 1 shows the lower leg below the knee to provide an idea of the mesh resolution of the coarse model. We note that each toe has been simplified so that the 2 or 3 small phalanges that form a single toe are merged into one. This results in a single phalange for each toe. The intent of the mesh resolution is to capture loading events that are on the order of milliseconds, i.e., accumulated strain of the bone structures and some kinematics.

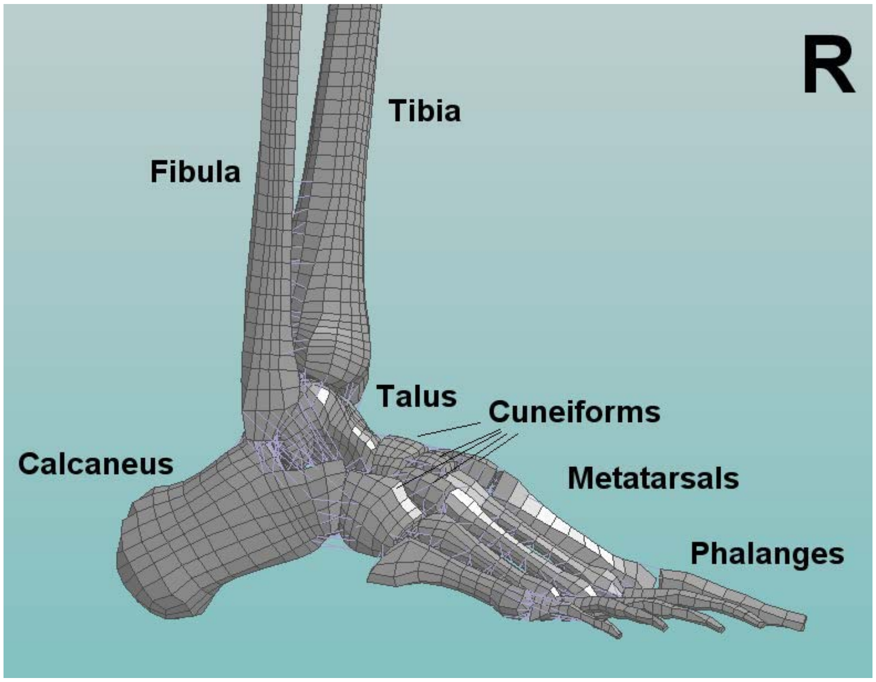
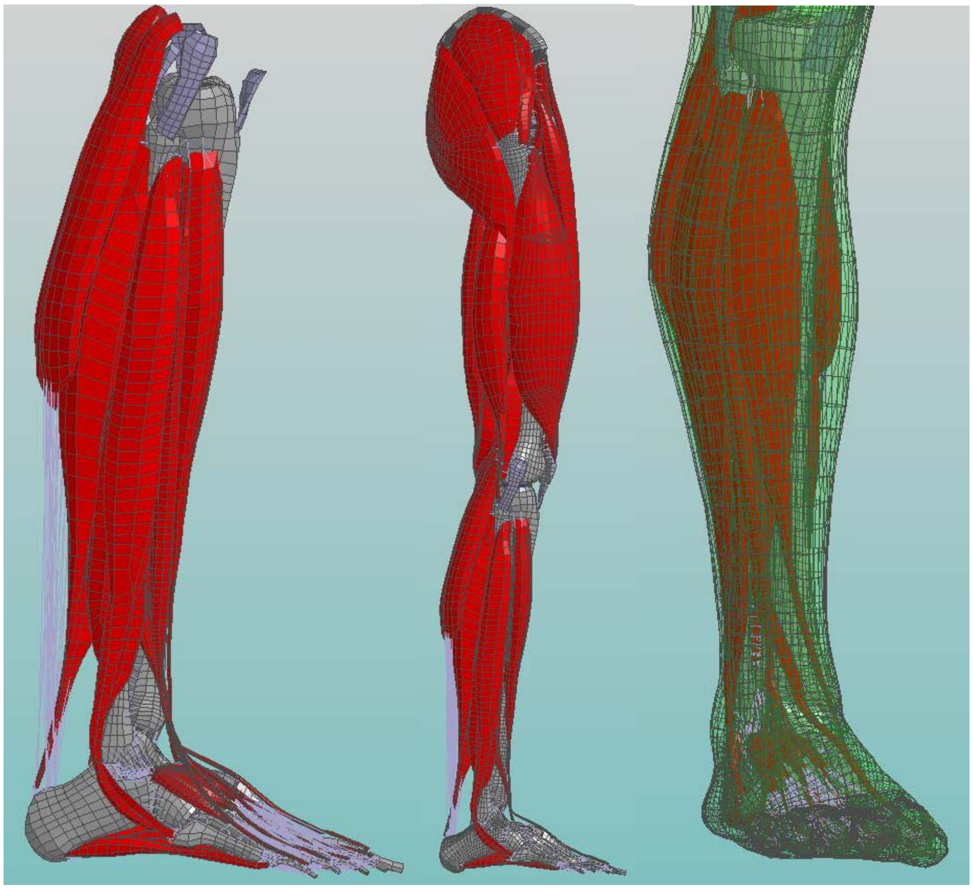


Fig. 1 **Coarse hexahedral meshes of the bones of the foot and lower leg**

In addition to the 23 bones, there are 19 muscles that are currently meshed below the knee, these are shown in the left panel of Fig. 2a (red elements). There are also 23 muscles between the pelvis and patella that have been meshed (Fig. 2b). Figure 2c also shows the current flesh layer (green elements) that envelops the muscles and bones. This flesh layer is a homogenization of the multiple layers of skin, fat, and other connective tissues that surround the musculoskeletal systems. Figure 2 also illustrates a number of 1-D structural elements (purple line segments) used to represent connective tissues such as tendons and ligaments. Where necessary, these 1-D structural elements will be replaced with solid elements in future work.



(a) (b) (c)

Fig. 2 Coarse hexahedral meshes of the muscles and flesh of the lower leg

Using LS-Dyna (Livermore Software Technology Corporation), simulations without the flesh layer (corresponding to roughly 30,000 elements) can reach 50-ms simulation time on 16 cores in under an hour. Incorporation of the flesh layer into a running simulation is underway. The difficulty lies, in part, in the choice to mesh parts separately, but more specifically challenges are associated with errors in the Zygote Media Group, Inc. source geometries. The source geometries, as mentioned before, are STL format and have numerous surfaces that interpenetrate one another. This becomes an issue when the solver attempts to resolve a defined contact between 2 interpenetrating surfaces. In these cases, the softer material is moved in such a way that it can create a small or negative volume. Element erosion occasionally can fix this, but it can lead to a zero time step. Parallel efforts are currently underway to convert the meshed structures into CAD formats where Boolean operations can be performed to remove the interpenetrating surfaces as well as identify critical interpenetrating surfaces and attempt to resolve them at the blocking stage. Enabling contact definitions also may reduce the time step to maintain stability, thereby causing a reduction in performance.

3. Assembler Overview

The challenges and questions raised in the introduction need to be answered for any model, and ideally, changing the specific configuration within a model should be fairly simple and require minimal work. The remainder of this report describes some of the features of an assembler program that is in development.

The assembler program is divided into 4 main components: a collection of finite-element data structures called Gilgamesh, a model parser and assembler called Gargamel, a model deformer Gumby, and a model translator Galvatron.

Gilgamesh handles the objects that are related to the model and the mesh, i.e., nodes, elements, parts, materials, boundary conditions, etc. Gargamel takes the input meshes and performs some practical operations including adding comments, material assignment, and colormap generation but also includes subdividing source meshes into multiple subparts, determining contact pairs from sets, and creating additional connective elements such as 1-D beams. In this way, Gargamel takes specified inputs and operators to meld together a model that is represented as a Gilgamesh object. Gumby operators apply deformations to the nodes stored in Gilgamesh using the information specified by Gargamel. These deformations (discussed in Section 6) are used to accommodate biological variability in the structures and will be used in the future to accommodate some posturing. Galvatron is the last stage of the assembler code. Galvatron translates the model contained in Gilgamesh to a format that can be parsed by an external finite element method (FEM) solver.

The program is written in C++ and heavily makes use of operator overloading, function pointers, and stated-structures. The end goal is to allow a highly specialized but simple interface for a human model developer. At this point, the interface is within the code itself and requires a recompile, but a parser could easily be added in a future revision.

Current off-the-shelf software largely addresses either the meshing step or the solver step. Both sides have features that are similar to what is being described here, and a brute force approach could be used to accomplish these tasks. It is common for a modeler to have dozens of disjoint scripts to accomplish some of the tasks that are discussed in Sections 4–6. For example, LS-Dyna has an accompanying pre-processing software that allows both scripted and graphical interface interaction to manipulate the mesh and materials. Similarly, TrueGrid (XYZ Scientific Applications) is capable of outputting Dyna3D (Lawrence Livermore National Laboratory) input decks. However, in both cases, swapping out a part that is multiply connected by 1-D structural elements and inserting a finer mesh would require manual manipulation and changes in multiple locations in the input deck. Similarly, deciding to subdivide a part into multiple parts according to a parameterized function requires external scripts or mesh manipulation and re-importing of the model. In the case of a functionally graded material, new parts to the mesh would have to be included, requiring new material definitions and new contacts to be defined manually.

The mesh is often considered a single entity at the input stage of FEM solver codes, making version control of the model combinatorial. In the ideal case, the model should be the mesh, material models, and boundary conditions, and it should be somewhat solver independent. This procedure would also largely depend on the target solver. As one can see, it can become quite cumbersome for a modeler to toggle back and forth between solvers. However, by centralizing some of these capabilities and adding in very specific features that would otherwise appear in a tangled mess of scripts, the assembler organizes research projects into one program so that they can be used later on in a practical way.

Another key strength of using a programming language like C++ is polymorphism and object-oriented programming. An object is said to be polymorphic if it behaves differently in different contexts. There are many different ways to accomplish the polymorphic design and our current approach is to utilize a state structure. For example, in the context of solver input decks, a material is output to a file in different formats depending on whether the assembler is in an LS-Dyna, Dyna3D, or SIERRA state (Sierra Solid Mechanics, Sandia National Laboratory). Thus, the modeler can work with the constitutive model and parameters instead of worrying about properly formatting the material for the code. This simple task can be prone to multiple errors. In the case of the lower leg, if each component has a unique set of material models and parameters, a few hundred material definitions would have to be translated from one solver to another. Since the material models and parameters are still being updated, this task would be very inefficient as it would have to be revisited numerous times. By utilizing polymorphic design, the model development is solver independent. In the current version of the code, the assembler outputs to LS-Dyna input decks, and is partially implemented to output Dyna3D input decks. Since this is done utilizing a single class structure, the Dyna3D portion can be omitted entirely and not affect the code. Similarly, future additions like a SIERRA output class could be added in the future, thus making it a flexible approach. What is important to note is that the polymorphic design lets the user interact with the assembler using the same commands, and it will produce different output depending on the solver format that is desired. This allows the user to view the model as mesh entities, materials, boundaries, and contacts, and the code properly converts it to the syntax needed for the FEM parser. These translator subroutines can be added as needed to accommodate the specific material models that are used, i.e., we are not attempting to recreate or accommodate for the entire material library.

4. Contact, Material, Part Management, and Structural Element Generation

Since we are using a higher-level assembler program that maintains an abstraction of the FEM model, the contacts defined, the materials, and the parts are all kept in data structures. The map data structure in C++ is utilized to associate common names of the part, e.g., “tibia” to class structures that have details about that part, e.g., the material of the tibia. The data structures allow for iteration and therefore simple renumbering when necessary (as is needed in Dyna3D). The part management also enables useful comments to be generated within the input decks that help isolate nodes, elements, parts to be associated with their “common name” as well as their internal solver identification numbers. These same keys are used to create a colormap customized to the model.

Contacts are defined pairwise and have a similar data structure for their management. Similarly for materials, sets of data structures help organize the model. These are managed in a larger structure that assigns numbers to them properly and interacts with the elements and part objects to do similar assignments. Contacts can also be defined as one-to-many using sets or all combinations. For example, the tibia can be set to have contact with all the muscles below the knee, or bones of the foot can be assigned to have contact with each other.

The mesh used for a part is included in its internal information, so making a substitution from a coarse mesh to a fine mesh is a simple file name substitution (see Fig. 3). The data structures that manage the parts also communicate with the contact and 1-D structural element generation. This allows the user to simply comment out a part on a single line and the contacts, 1-D structures, material files, elements, and nodes will all be renumbered and accounted for automatically. Both of these cases are particularly useful in Dyna3D since any change in element count directly affects the nodes, elements, parts, in header files and also impacts the number of “cards” the solver is expecting. Similar interdependencies exist in LS-Dyna as well, although they are not as pervasive.

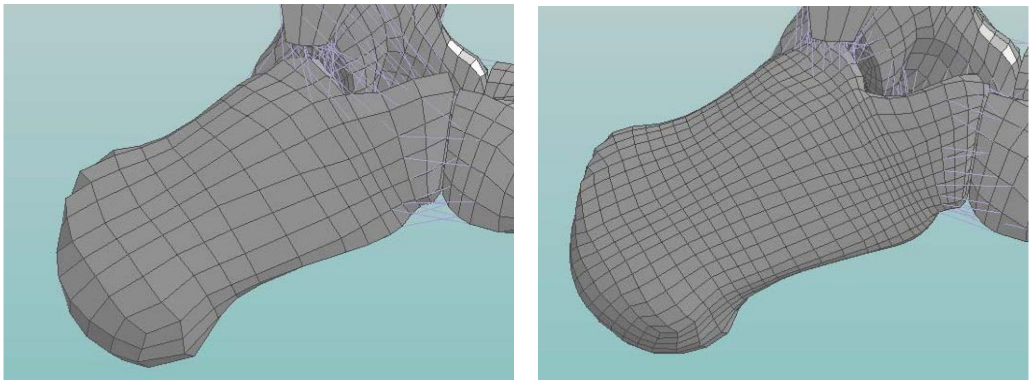


Fig. 3 A coarse mesh of the calcaneus (left) is replaced with a refined mesh (right) by changing a single line of code. The 1-D structural elements are automatically updated during the assembly process in the model management software.

A major drawback to the Zygote Media Group, Inc. geometries that we currently have is a lack of detail in the connective tissues. This results in gaps between bones, which is clearly problematic for simulating wave propagation. Figure 4 shows a close-up of the cuneiforms of the foot. In the body, these bones are tightly coupled to one another and have cartilage filling the gaps between bones. Due to the lack of detail in the source geometries, the choices on the modeling side are either to manually create cartilage structures or to fill these gaps with a simpler approximation (background grid approaches are also being explored). The auto 1-D–element calculators are the simplest approach to connect adjacent bones. In Gargamel, one can request a connection between 2 objects. This connection can be parameterized so that the connected nodes between 2 objects can be adjusted. Figure 4 shows an example where the number of connections and their desired minimum and maximum lengths are changed. One can also keep the same parameters, and make a mesh substitution and the beam connectors will be recalculated (Fig. 3).

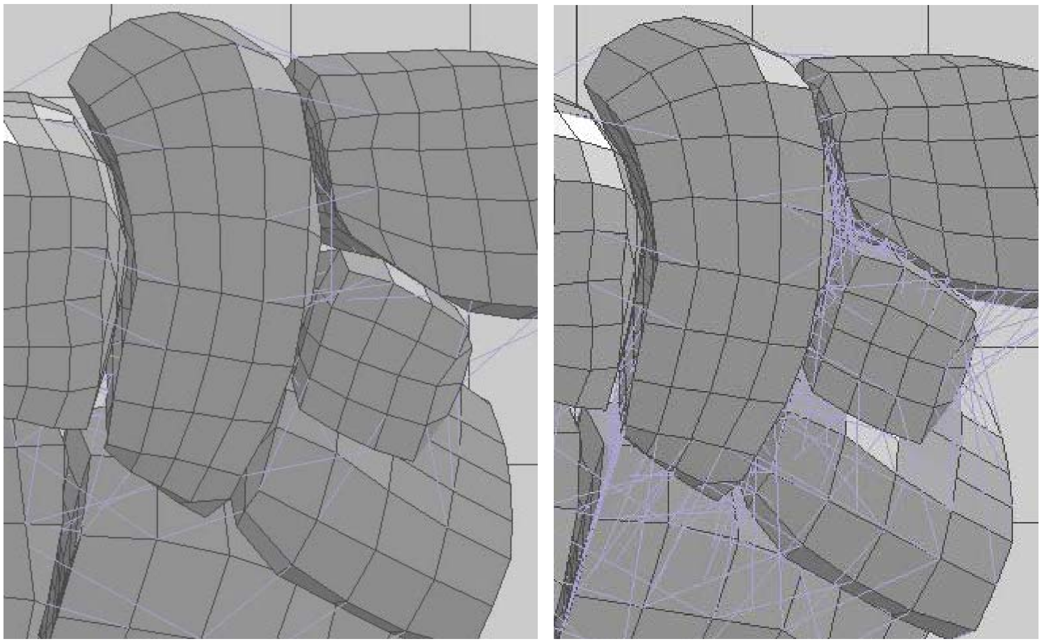


Fig. 4 Close-up of the cuneiforms and navicular. The left panel shows a set of 1-D structural elements (purple) automatically determined from the model management software. The right panel shows a second set of elements determined from a different set of parameters.

5. Subdividing and Grading Materials

There are numerous transitions of materials within the body. Some of these transitions can be considered as disjoint layers while others are more appropriately modeled as functionally graded materials, e.g., trabecular bone. However, this is not a simple material swap in the case of switching between a functionally graded material and 2 distinct layers.

The source geometries correspond to a major anatomical component. In the case of the bones, it is only the surface of a particular bone. Therefore, any detail within that volume needs to be generated by the modeler. Specifically, the thickness of the cortical shell is an issue. This outer region of the bone varies in thickness from one bone to the next and within a bone itself. Some researchers choose to represent this layer with shell elements, while others assume a uniform thickness of solid elements, while others take into account some of the varying thicknesses. A shell representation may allow for a considerably larger time step as well as fewer elements. The current version of the assembler program lets the user switch between representing the surface by shell elements or a uniform thickness by changing one line of code. This is done within Gargamel by taking the input mesh of the entire part and subdividing it into multiple parts according to a thickness parameter. Figure 5 shows an example result of this operation for the calcaneus and talus. This procedure is limited by the resolution of the mesh and therefore can produce a stair-stepped interior, as seen in Fig. 5. This procedure can be generalized in the future to account for graded materials or alternate methods of subdividing the part.

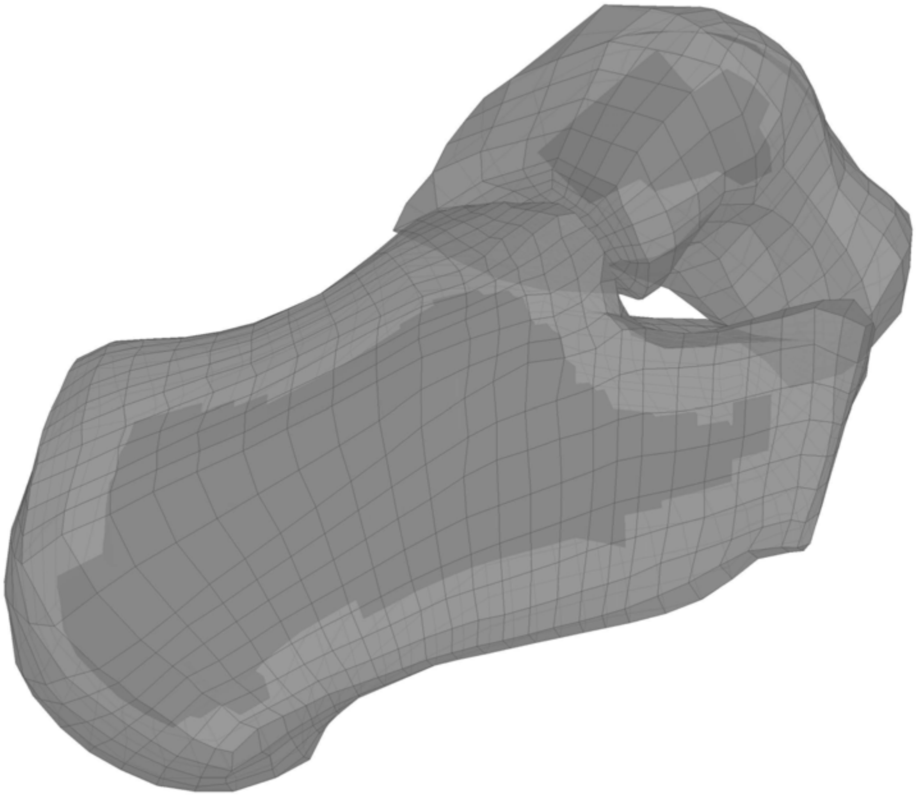


Fig. 5 The calcaneus and talus are subdivided to account for a solid cortical layer (light gray) and trabecular interior (darker gray). The stair-stepped trabecular interior is due to the coarse mesh resolution.

6. Accounting for Biological Variability

Biological variability is another issue to be accounted for when modeling the human. The goal of the assembler is to have a single set of source geometries and their associated meshes that can be distorted to represent other individuals. Often a simple uniform scaling can be used to match a single length, e.g., femur length. Unfortunately, femur length and other measurements do not correlate to all other possible anthropometric measurements. Here, we briefly describe a capability of the assembler that enables a single FEM model to undergo simple modifications in the code to output a different model for different Soldiers.

In Gumby, various functional nodal mappings can be applied to and taken relative to specific structures. Figure 6 shows a particularly extreme example where the relative length of the femur to the tibia is taken as an input parameter. In Gumby, a source object like the tibia or femur is used to define a coordinate system upon which a smoothly varying function defines the nodal mapping. In this case, the nodes of the leg are either compressed or stretched to translate the relative location of the knee up or down. Aside from shortening or lengthening the surrounding muscles, this mapping does not change the other dimensions of the leg. Although this method is still under development and can be improved to minimize knee distortion, it shows potential for accommodating multiple Soldiers from one FEM model. The 9 legs shown were generated in seconds by adjusting a single input parameter.

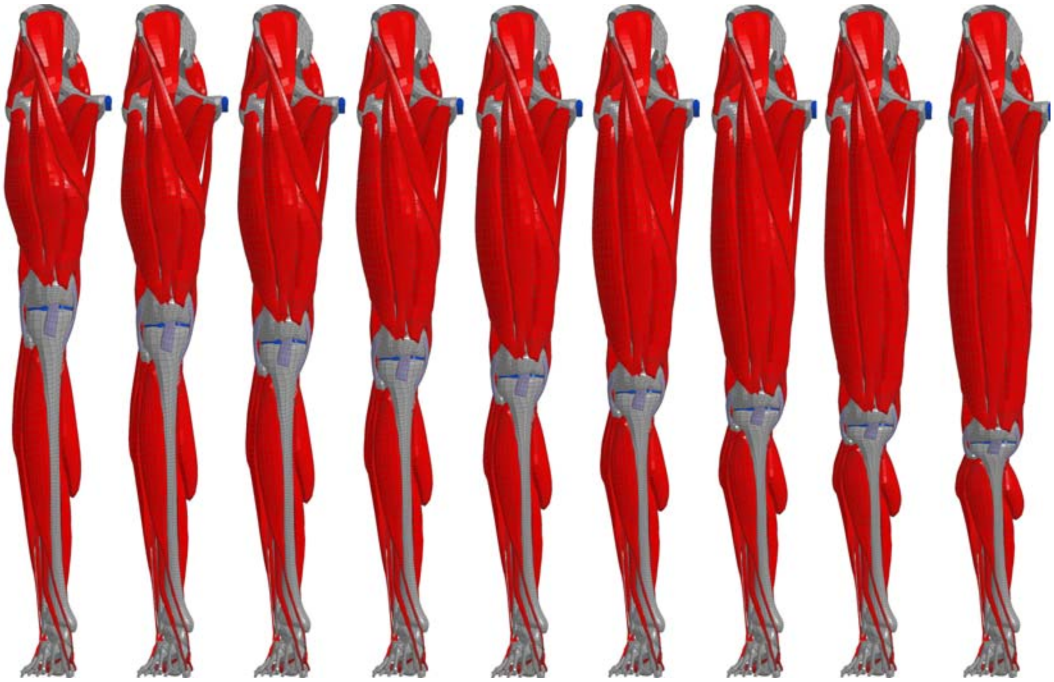


Fig. 6 The relative length of the femur to tibia is taken as an input parameter and scaled independently from the overall model dimensions

Another Gumby-related distortion is shown in Fig. 7. Here the relative thigh and calf thicknesses are adjusted by an input parameter. Using the surface location of the long bones, a functional mapping can be defined to only alter the nodal locations of the flesh. This type of morphing provides the capability of modeling individuals with a larger soft tissue mass relative to their bone structure.

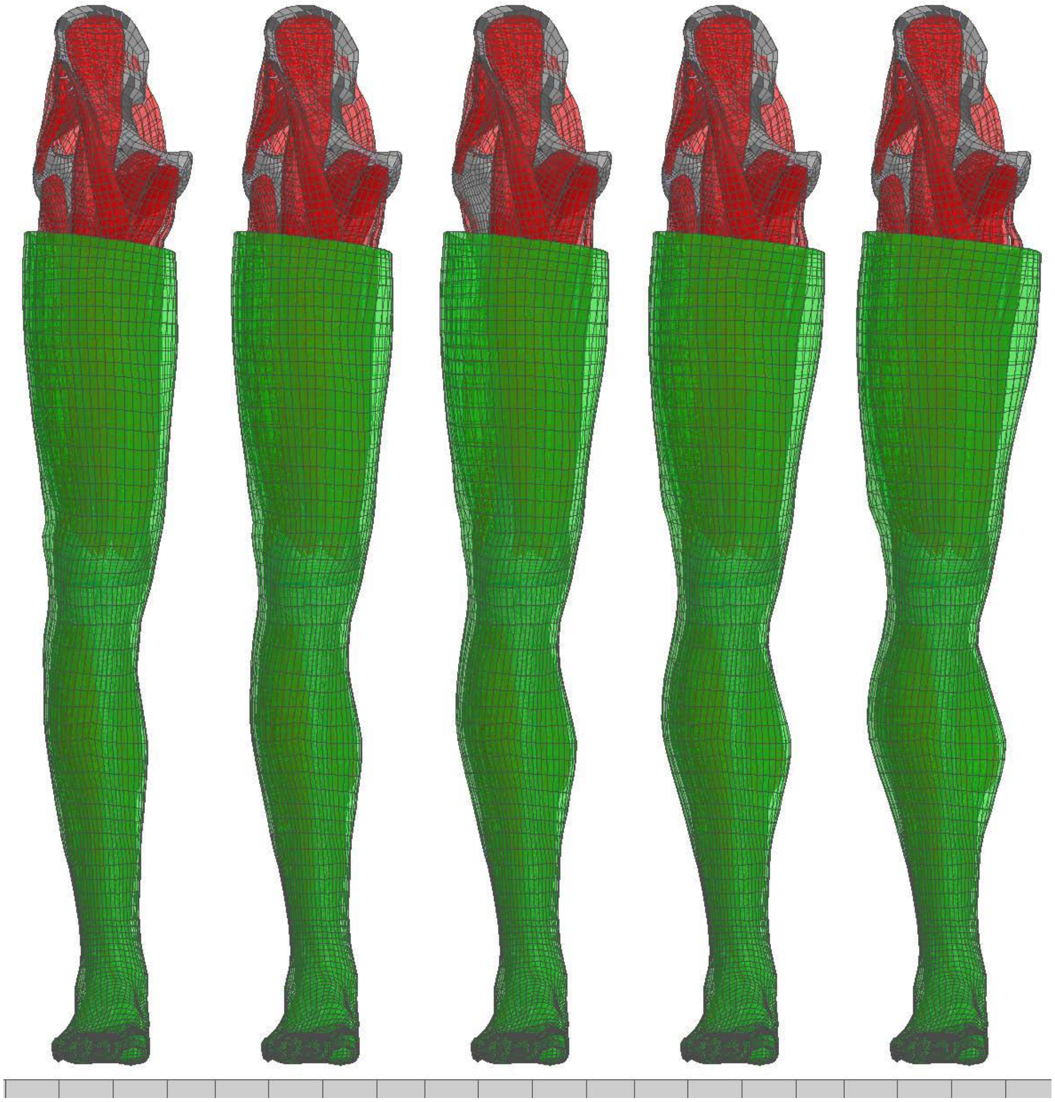


Fig. 7 **The relative size of the thigh and calf are taken as input scaled independently from the overall model dimensions, and do not affect the underlying bone structure**

Both of these approaches can be combined together with an overall scaling of the model. All of these features in combination would enable FEM simulations of individuals not represented by the original geometries. This procedure can easily be extended to the other regions of the body and enables the vision of a having an army of human body models to subject to the same threat.

7. Conclusions

We have briefly presented some of the progress on modeling the lower leg. A major effort has been put into mesh development. The coarser resolution and the update to hexahedral elements have tremendously reduced computational run time compared to the earlier version of the model. The model still requires a lot of work due to interpenetrating volumes that are present in the source geometries. A framework is being developed to manage and assemble the human body model to allow for rapid prototyping and incorporating enhancements that are specific to the human body. The goal of the framework is to be solver independent, to organize and streamline the human body modeling effort, and to mix-and-match component models from different resolutions and configurations to account for numerous Soldiers and postures.

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO LL
IMAL HRA MAIL & RECORDS
MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

43 DIR USARL
(PDF) RDRL WM
S KARNA
RDRL WML C
T PIEHLER
RDRL WML H
B SCHUSTER
RDRL WMM B
B LOVE
RDRL WMM G
L PIEHLER
N ZANDER
RDRL WMP
S SCHOENFELD
RDRL WMP B
A DAGRO
A DILEONARDI
W EVANS
C GUNNARSSON
C HOPPEL
Y HUANG
M KLEINBERGER
M LYNCH
J MCDONALD
P MCKEE
S SATAPATHY
A SOKOLOW
C WEAVER
T WEERASOORIYA
S WOZNIAK
T ZHANG
K ZIEGLER
RDRL WMP C
R BECKER
S BILYK
T BJERKE
D CASEM
J CLAYTON
D DANDEKAR
M GREENFIELD
B LEAVY
RDRL WMP D
R DONEY
J RUNYEON
RDRL WMP E
P SWOBODA
RDRL WMP F
D FOX
E FIORAVANTE
N GNIAZDOWSKI
R GUPTA
R KARGUS
RDRL WMP G
R BANTON
N ELDREDGE
S KUKUCK

INTENTIONALLY LEFT BLANK.