Published in final edited form as:
Comput Aided Des. 2010 December 1; 42(12): 1108–1116. doi:10.1016/j.cad.2010.07.005.

FEATURE-BASED MULTIBLOCK FINITE ELEMENT MESH GENERATION

Kiran H. Shivanna[1,2], **Srinivas C. Tadepalli**[1,2], and **Nicole M. Grosland**[1,2,3]

[1] Center for Computer Aided Design, The University of Iowa, Iowa City, IA 52246, USA

[2] Department of Biomedical Engineering, The University of Iowa, Iowa City, IA 52246, USA

[3] Department of Orthopaedics and Rehabilitation, The University of Iowa, Iowa City, IA 52246, USA

Abstract

Hexahedral finite element mesh development for anatomic structures and biomedical implants can be cumbersome. Moreover, using traditional meshing techniques, detailed features may be inadequately captured. In this paper, we describe methodologies to handle multi-feature datasets (i.e., feature edges and surfaces). Coupling multi-feature information with multiblock meshing techniques has enabled anatomic structures, as well as orthopaedic implants, to be readily meshed. Moreover, the projection process, node and element set creation are automated, thus reducing the user interaction during model development. To improve the mesh quality, Laplacian- and optimization-based mesh improvement algorithms have been adapted to the multi-feature datasets.

Keywords

Finite element mesh; Preprocessor; Multiblock; Feature; Orthopaedics; Mesh improvement

1. Introduction

Computational models of joint anatomy and function provide a means for biomechanists, physicians, and physical therapists to understand the effects of repetitive motion, acute injury, and degenerative diseases. Furthermore, such models may be used to improve the design of prosthetic implants. The ability to develop models on a patient/subject specific basis, in a rapid and reliable manner, can facilitate the understanding of biomechanics and can be used in a variety of applications in the medical and safety arenas. Amongst the existing numerical methods, Finite Element Analysis (FEA) is the most widely used due to its broad range of applicability and its ability to represent complex geometries such as anatomical structures [1–3]. Of the many applications, FEA is commonly used for orthopaedic implant design [4–6], implant structure interaction [7], bone remodeling [8] and joint contact mechanics [9]. FEA gives insight into the behavior of the structures being analyzed by augmenting physical experimentation. Oftentimes FEA is the only choice where physical experimentation is not feasible.

Mesh generation is a necessary step in any finite element analysis. Inevitably, it constitutes the bulk of the setup time for a problem. The time devoted to model development varies with the

user, the pre-processing software, and the structure under consideration. By and large, the time required to generate a mesh increases with increased structural complexity. This is especially true of anatomic modeling where the structures to be meshed are anything but regular. The increase in time is attributed in part to the fact that the quality of the computed solution and the solution time are highly dependent on the quality of the mesh and the chosen element type. While free-form meshing schemes using tetrahedral elements are widely employed, it is well known that hexahedral elements would, in many cases, be more effective for analysis. Tetrahedral meshes typically require 4–10 times more elements than a hexahedral mesh to achieve the same level of accuracy [10–12]. As a result, there is an increase in computational cost. Likewise, in response to high deformations and/or nearly compressible materials, low order tetrahedral elements tend to be overly stiff and lock. Unfortunately higher order tetrahedral elements are prone to inversion, or turning inside out. Mesh generation consisting entirely of hexahedral elements, however, can be a daunting task, which is amplified when irregularly shaped structures are considered.

Few meshing strategies, or more specifically pre-processing packages, have been designed with an emphasis on anatomic model development. Commercial meshing tools are typically geared toward meshing engineering components (in the form of geometric data) with regular geometry and dimensions. Hence, they are often not suitable for meshing the irregularities of anatomic structures (in the form of discrete data), especially when a mesh consisting of hexahedral elements is desired [13,14]. The amount of time required to generate a mesh of high quality with traditional commercial software can be prohibitively expensive, oftentimes precluding their use in analyzing a large number of data sets. Hexahedral meshing methods targeting anatomic structures have recently been proposed. For example, octree-based [15], octree coupled with dual-contouring [16], and marching cubes coupled with sheet insertion [17] are some of the proposed methods. Each of the aforementioned techniques start with a voxel mesh from which the final mesh is sculpted along with the use of buffer layer insertion and node movement based mesh quality optimization [18]. Although such techniques yield meshes of high quality, element size and orientation control remains a challenge. This challenge can be addressed with the multiblock meshing technique which affords greater control over element size and orientation [19] as compared to the previously mentioned techniques. This is important because properly chosen element size and orientation aids in efficient modeling of anatomic structures.

Toward addressing the challenges of anatomic mesh generation, we established an open source finite element pre-processing environment called IA-FEMesh (Iowa FE Mesh) to accelerate the development and improve the quality of anatomic finite element models (http://www.ccad.uiowa.edu/mimx/IA-FEMesh/) [19]. IA-FEMesh employs an interactive multiblock meshing technique. In brief, the models initiate with a triangulated isosurface (STL or VTK format) of the structure of interest. Anatomic models, for example, may be generated directly from a segmented image data set (i.e., CT or MR), while triangulated surfaces generated from geometric surfaces may be used to mesh an implant. Thereafter, an arrangement of building blocks is positioned about the surface to best replicate the basic geometry of the structure. Building block generation is an important and often challenging step in the meshing process. The meshing algorithms implemented in IA-FEMesh to date, rely on closest point projection to map the multiblock nodes/elements to the surface of interest. Interactive tools provide the user control over block vertex placement, and hence indirectly the control over the placement of the nodes assigned to the block edges and faces.

As the level of structural complexity increases, the current multiblock meshing technique may prove insufficient (e.g., the dataset is limited to a single closed surface). Such techniques alone may fail to place/maintain nodes at the desired locations, thereby resulting in a loss of mesh fidelity, distorted elements, and/or an overall poor mesh quality. This is most prevalent in

regions of high curvature or at abrupt changes in geometry. For example, consider a cylindrical surface (Fig. 1a). Unfortunately, the closest point projection technique does not guarantee that the nodes along the edges of the block project to, nor stay on, the periphery of the cylindrical end cap (Fig. 1b). As illustrated in Fig. 1b, geometric features such as ridges and corners frequently require special attention/treatments to achieve a high degree of geometric fidelity (i.e., the mesh illustrated in Fig. 1c). Moreover, mesh improvement strategies must strive to preserve such features while improving the mesh quality. Consequently, our goal is to use the features inherent to a structure as a meshing aid.

The objective of this study was to develop proof of concept meshing strategies for addressing the challenges of meshing anatomic structures and biomedical implants. The ultimate goal is to accommodate multi-feature datasets in a multiblock framework while reducing/easing the user interaction during mesh development. Towards addressing this issue, the following techniques have been devised and tested: (1) automated and user defined feature edge definitions; (2) automated entity projection calculations based on a novel 'number of neighbors' rule; (3) node and element set creation based on the projection entity calculation; and (4) Laplacian- and optimization based mesh improvement algorithms for multi-feature data sets. Orthopaedic related examples are highlighted to illustrate the capability of the proposed methods.

2. Methods

In brief, each mesh initiates with a triangulated surface, or surfaces, of the structure of interest. Thereafter, additional structural features may be generated either automatically through feature extraction or manually via interactive user tracings. A multiblock structure is sculpted about the surface(s) such that the block vertices capture the features. The computed projection entities (i.e., edge, face, and block) are used to generate the mesh and define associated node and element sets. If need be, the element quality of the mesh is improved. The meshing workflow is depicted by the flow chart in Figure 2.

2.1. Feature edge definition

Feature edges may be defined automatically by extracting the boundary edges and/or sharp surface edges from a polygonal dataset (i.e., triangulated surface) (Fig. 3a). A boundary edge is defined as an edge that is used by only one polygon, while a sharp surface edge is shared by two polygons whose dihedral angle is greater than a specified value (in Fig. 3b a value of 90^0 is specified). Situations arise when a structure would benefit from user defined feature edges. Consequently, we have enhanced our earlier tracing methods [20] to interactively draw a contour directly on the surface using an advanced interactive 3D widget (Fig. 3c). This widget allows for flexible assignment of control points on the surface of interest; the number and location of the points can be varied interactively. Once the control points have been defined, the shortest path between successive control points along the surface of interest [21] is used to define the feature edge (Fig. 3c inset).

2.2. Block structure entities

The basic entities of a block structure, in the order of dimension, include: vertices (0D), edges (1D), faces (2D), and blocks (3D). During the projection procedure, a series of projection entity lists were established/maintained. These lists catalog all building block vertices and edges to be projected onto a given feature edge, all building block faces to be projected onto a given surface, and all blocks associated with a given surface. The techniques used to generate and maintain the respective lists are detailed below for each block entity. Hereafter, for illustration purposes 'V', 'E', 'F' and 'B' are used to denote the aforementioned block entities, respectively.

The following conditions hold true during the projection process: (1) If a vertex of the building block structure is within a user-specified distance of a given feature edge, then the vertex is designated to be projected onto that feature edge as opposed to the closest point on the surface; (2) If the two vertices that define an edge of a block project onto a feature edge, then the edge of the block projects onto the feature edge; and (3) If all four vertices of a block face project onto a surface, then the block face is designated to project onto the surface.

The following assumptions were enforced to ensure satisfactory performance of the multi-feature projection techniques: (1) if a feature edge is an open contour, the open ends of the contour cannot be represented by a single building block edge (Fig. 4a). Hence, the edge should either be divided (Fig. 4. b) or the contour should be closed (Fig. 4c) so as to ensure satisfactory projection; (2) a feature edge must lie on the surface of interest; (3) Unintended boundary edges caused by missing surface polygons must be avoided; and (4) If a surface is an open volume with a topological hole, the hole in the surface cannot be represented by the building block faces that are designated to be projected onto the surface (Fig. 5a). Hence, the building block face should be subdivided such that the subdivided faces are no longer projecting onto the surface (Fig. 5b) or the topological hole on the surface should be patched (Fig. 5c) for satisfactory projection.

2.2.1. Vertex—In an effort to accommodate the multi-feature definitions and to ease the mesh generation process, visual feedback is provided to the user by assigning feature dependent colors. Accordingly, we have developed building block editing operations to accommodate the color-coded feature definitions (Fig. 3d). For example, as the user manipulates the vertices of a building block structure within a specified tolerance of a given feature, the sphere representing that vertex changes color to match that of the feature edge. This visual feedback helps to ensure that vertices, edges, and faces will be projected onto the desired feature of interest. This color coding scheme facilitates vertex visualization especially in cases of complicated building block structures. Based on the vertex projection information, the projection of the remaining building block entities can be established.

2.2.2. Edge—Once the vertices of a block are designated for projection onto a given feature edge, the associated edges, must too project accordingly. Care was taken to ensure that internal edges represented by vertices assigned for projection, must not be projected onto the feature. For example, consider two building blocks used to define a cylindrical surface (Fig. 6a). Circular contours defining the periphery of each end cap were automatically extracted as feature edges (Fig. 3b). With six vertices positioned within tolerance of each feature edge, ideally only the six external edges (E_1–E_6) bounding an individual contour would be projected onto this feature (i.e., contour). To avoid projecting the internal edge(s), both the edges (E_1–E_7) and faces (F_1 & F_2) defined by these vertices were identified (Fig. 6b). All edges common to adjacent faces were removed from the list designated for projection. This would correspond to eliminating edge E_7. The resulting entities to be projected are shown in Figure 6c. In some instances, an internal edge may not be common to two faces designated for projection (Fig. 6d). For example, using the aforementioned protocol, edges E_1–E_6 were retained and designated for projection onto the contour (Fig. 6e). Edge E_1, however, represents an internal edge and should be removed from the list of projected edges. Consequently, a ‘number of neighbors’ rule was implemented, such that a given edge is retained if, and only if, it has a total of two neighboring edges sharing a common vertex. For example, as illustrated in Figure 6e, edge E_1 has four neighbors, while edges E_2–E_6 each have three, or fewer, neighbors. Consequently, the edge with the most neighbors (E_1) is removed from the projection list (Fig. 6f). The number of neighbors associated with each edge remaining in the list is recalculated, and the process repeats itself until only those edges with two or less neighbors remain. As illustrated, after the removal of edge E_1, the number of neighbors for each of the edges E_2–E_6 is two.

2.2.3. Face—The aforementioned logic was extended to the block faces (Fig. 7). If, for example, two blocks (B_1 and B_2) share a common face, the common face (F_1) is removed from the list of entities to be projected. Moreover, a check of neighboring faces is performed. Neighboring faces share an edge. In order to be retained, the count of neighboring faces may not exceed four. As illustrated in Figure 7a, Face F_1, has eight neighboring faces. Consequently, had both blocks not been considered for projection, the face would be eliminated based on the number of neighbors rule.

2.2.4. Block—Types of feature surfaces used for mesh generation can broadly be classified as: 1) Open surface (e.g. articular cartilage surface), 2) Closed surface within a closed surface (e.g. brain containing a tumor), and 3) Surfaces abutting each other without overlap (e.g. labrum surface abutting the cartilage surface). The procedure used to generate a list of blocks associated with a given surface is illustrated with an example of a cylinder within a cylinder (Fig. 8a). If an exterior face of the block structure projects onto a surface, the block coupled to the face becomes associated with that surface. As a result, all of the outer blocks would be associated with the outer cylinder (Fig. 8b). For the remaining internal blocks, a block is associated with the smallest closed surface volume within which the centroid of the block lies. It should be noted that a block can only be associated with one surface. As illustrated in Fig. 8a, the centroid of the inner block lies within both the inner and the outer cylinder. The inner cylinder has the smaller volume thus requiring that the inner block be associated with the inner cylinder (Fig. 8c).

2.3. Mesh generation

Once the building block structure has been established, mesh seeds are assigned to the individual blocks. Mesh seeding specifies the subdivision of individual blocks, which in turn controls the number of elements in the final mesh. The steps for the creation of the finite element mesh are as follows,

1. A rectilinear grid structure is created from the initial mesh seeding.
2. The grids corresponding to the blocks are considered one by one.
3. The nodes making up the edges of the block structure are either projected (if the edge is a part of the edge projection list) onto the designated feature edge or recalculated using linear interpolation.
4. The faces are either projected (if the face is part of the face projection list) onto the designated surface or interpolated using elliptical interpolation [22].
5. The interior nodes of the grid are interpolated using elliptical interpolation.
6. Steps 3–6 are repeated for all blocks in the block structure to form the final projected mesh.

2.4. Node and element set creation

The aforementioned meshing strategy readily enables the nodes and elements to be grouped automatically for material and load/boundary assignments (Fig. 9). Consider the mesh created (Fig. 9a) from the features and block structure shown in Figure 8a. For example, nodes projecting onto a feature are grouped together to form a node set. Figure 9b illustrates the nodes corresponding to the feature edges and nodes corresponding to the outer cylinder surface (the nodes corresponding to the inner cylinder are not shown). Elements associated with the blocks that are in turn associated with feature surfaces are grouped together to form element sets. Figure 9c shows the element sets created for both the inner and outer cylinders. The elements rendered in wireframe constitute the element set representing the outer cylinder and the element set rendered in solid corresponds to the inner cylinder.

2.5. Mesh quality improvement

The closest point projection scheme used to place nodes on features can result in a node distribution that is less than optimal. The node distribution directly impacts the quality of the mesh which in turn impacts the accuracy of the finite element analysis. Hence, mesh improvement strategies involving Laplacian and optimization based methods have been implemented [23] in the meshing workflow. Here nodes are divided into (1) feature associated nodes (nodes projecting onto a feature edge or surface) and (2) independent nodes (all other nodes). Feature associated nodes are repositioned using Laplacian smoothing [24]; followed by the repositioning of independent nodes using optimization based methods [18,25]. Laplacian smoothing is carried out in two stages. In the first stage of the Laplacian smoothing, nodes associated with feature edges are smoothed. In the second stage of the Laplacian smoothing, nodes associated with surfaces are smoothed. The feature edge node sets are constrained in the second stage of the Laplacian smoothing. The nodes during Laplacian smoothing are always projected back onto the feature of interest during smoothing.

In the next step of mesh improvement, independent nodes are repositioned to improve the mesh quality by optimization based methods while feature associated nodes are constrained. In optimization based methods, mesh quality metrics are used as the objective functions for minimization. Of the mesh quality metrics, Jacobian based metrics are commonly used due to their strong mathematical background [26]. The optimization based mesh improvement method used here incorporates mesh untangling [18] followed by shape improvement [25]. In the mesh untangling phase, nodes in an element with non-positive Jacobian are repositioned so as to make the Jacobian positive. In the shape improvement phase, the inverse mean metric ratio function is used to make the element angles as orthogonal as possible thus improving the mesh quality.

2.6. Orthopaedic Applications

To test the applicability of the methods described, models representing a femoral component of a total knee replacement (TKR) (Fig. 10) and a distal femur with cartilage (Fig. 11) were created. Three feature edges were extracted from the triangulated surface of the TKR to aid the meshing processing (Fig. 10a). The distal femur model consisted of two triangulated surfaces, one characterizing the femoral surface and the other defining the articular cartilage surface. The feature edge indentifying the periphery of the articular cartilage was projected onto the surface of the femur as a second feature definition (Fig. 11a, inset). Building block structures (Fig. 10b and 11b) closely resembling the structures of interest were generated. The tolerance used for the placement of block vertices was 0.5mm. Mesh seeds were assigned to the ensuing block structure and the nodes projected according to the techniques described herein. The resulting mesh definitions are shown in Figure 10c and Figure 11c. Mesh improvement strategies described previously were applied to the mesh definitions. In each case, three iterations of Laplacian smoothing were applied to the feature associated nodes. For the independent nodes, a minimum scaled Jacobian of 0.05 was specified for all the meshes during the mesh untangling step. After the mesh untangling step, three shape improvement iterations were applied.

3. Results and Discussion

In the absence of additional feature information, the minimum scaled Jacobian of the resulting cylindrical mesh (Fig. 1b) was 0.22. This metric increased to 0.44 with the supplemental feature edge data (Fig. 1c). Moreover, the average scaled Jacobian for the respective models increased from 0.58 to 0.76, thus indicating an overall improvement in mesh quality. The cylindrical mesh exemplifies the ability to enhance model fidelity by capturing the features and improving the overall element quality.

Figure 10d and Figure 11d illustrate the resulting mesh definitions for the TKR and the distal femur models, respectively. The resulting meshes were generated in a matter of minutes. The elements of each mesh (Fig. 10d and Fig. 11d) exhibit a positive scaled Jacobian after mesh quality improvement operations (Table 1); thus, making the meshes suitable for analysis. The minimum and maximum values of the scaled Jacobian for both examples were 0.05 and 0.99, respectively. The average scaled Jacobian value for the TKR was 0.85, while that of the distal femur was 0.81.

Numerous finite element models of orthopaedic structures exist and based on the type of elements, the models can be classified as containing tetrahedral, hexahedral, and mixed (containing both tetrahedral and hexahedral elements) meshes. It is generally accepted that hexahedral elements perform better in non-linear analyses [10]. Hence, the majority of the models contain hexahedral meshes [27–29] as compared to tetrahedral [30] or mixed [31] meshes. Bendjaballah et al. [29] have used a method to generate the hexahedral mesh but do not document the procedure to vary the mesh size, which is important in mesh convergence studies. In our method, the mesh density can be changed by changing the mesh seeding, thus simplifying mesh convergence studies. The knee mesh used by Haut-Donahue et al. [28] in the tibio-femoral contact is generated from planar projection which works well if the surface being meshed is nearly planar. The planar projection method might not be applicable for the surfaces with complex topology. In our method, use of closest-point projection ensures satisfactory meshing of surfaces with complex topology. Mixed mesh has been used to study the joint contact in the hip by Anderson et al. [31] and in the knee by Baldwin et al. [32]. The cartilage (region of interest) is modeled using hexahedral elements and the bone with tetrahedral elements. The tetrahedral and hexahedral elements are tied at their interface. But, mixed meshes cannot be used where the studies in interfacial stress transfer is of importance. In our method, the elements at the interface (cartilage and bone in Fig. 11d) have node to node correspondence, thus making the method suitable for interfacial stress transfer studies.

Other multiblock methods are available to mesh the examples highlighted herein [13,14]. Rarely, however, do the existing methods consider discrete triangulated surfaces as input, but use geometric/parametric data such as spline curves and NURBS surfaces. If the existing methods were applied to discrete datasets their use would be cumbersome at best, as the methods would require additional tools to convert discrete data to geometric/parametric data. The data conversion is required to generate a four sided geometric surface patch for each external block structure face. The surface patches are used in deciding the location of external nodes of the ensuing mesh. The size and the location of these geometric patches should be decided by the user thus requiring inordinate amount of preplanning. In our method, the closest point projection employed automatically places the external nodes on the surface without any user intervention thus reducing the time required to generate a mesh. To quantify the time required, a comparative study has been carried out previously comparing our method to other meshing techniques [33] and the study found a marked decrease in the time required to generate meshes. Multiblock based preprocessors [13,14] require the user to specify whether a given block entity is to be projected, and if so, where it projects. The complexity of manual projection specification increases with block structure complexity. In our method, automated calculation of projection edge and face lists using the ‘numbers of neighbors’ rule addresses this difficulty. For the examples chosen herein, the models were generated in matter of minutes. Moreover, the accommodation of multiple surface designations enables the node and the element set definitions to be automated, thereby further reducing the time devoted to mesh development. This is particularly helpful because anatomic structures are oftentimes multi-material datasets.

Future endeavors include generalizing the multi-feature techniques, thereby enabling countless models to be developed with ease. Moreover, we are addressing mesh improvement techniques aimed at both the internal and external node definitions. For example, automating various

gridding patterns at the block level are being considered. The internal mesh quality may be improved if the blocks are arranged in a butterfly pattern [34,35], as opposed to the rectilinear pattern. Rather than rely on manually defining such patterns, our goal is to automate the definitions thereof. Laplacian smoothing is used to improve the 3D surface mesh quality. The Laplacian smoothing is a heuristic method which repositions the nodes based on the average distance to the connected neighbor. Deterministic methods such as optimization-based methods perform better as compared to heuristic methods. Hence, optimization6 based methods are being explored for the improvement of the 3D surface mesh.

4. Conclusions

Our new method of mesh generation involves multiple discrete features, which makes it ideal for generating meshes of anatomic structures as well as implants. Automated calculation of the projection entities along with automated node and element set creation is incorporated to reduce user interaction. Mesh improvement algorithms have been implemented to improve mesh quality. We have demonstrated the application of the method for orthopaedic structures. Our goal is to generalize the techniques such that they may be applied to wide range of problems.

Acknowledgments

The authors gratefully acknowledge financial support provided by an award R01EB005973 from the National Institute of Biomedical Imaging and Bioengineering, National Institutes of Health.

References

1. Huiskes R, Chao EY. A survey of finite element analysis in orthopedic biomechanics: the first decade. Journal of Biomechanics 1983;16(6):385–409. [PubMed: 6352706]
2. Huiskes R, Hollister SJ. From structure to process, from organ to cell: recent developments of FE-analysis in orthopaedic biomechanics. Journal of Biomechanical Engineering 1993;115(4B):520–527. [PubMed: 8302035]
3. Prendergrast PJ. Finite element models in tissue mechanics and orthopaedic implant design. Clinical Biomechanics 1997;12(6):343–366. [PubMed: 11415744]
4. Halloran JP, Petrella AJ, Rullkoetter PJ. Explicit finite element modeling of total knee replacement mechanics. Journal of Biomechanics 2005;38(2):323–331. [PubMed: 15598460]
5. Schmidt H, Midderhoff S, Adkins K, Wilke HJ. The effect of different design concepts in lumbar total disc arthroplasty on the range of motion, facet joint forces and instantaneous center of rotation of a L4-5 segment. European Spine Journal 2009;18:1695–1705. [PubMed: 19763638]
6. Tai CL, Shih CH, Chen WP, Lee SS, Liu YL, Hsieh PH, Chen WJ. Finite element analysis of the cervico-trochanteric stemless femoral prosthesis. Clinical Biomechanics 2003;18:S53–S58. [PubMed: 12828915]
7. Gonzalez CD, New AM, Browne M. Probabilistic finite element analysis of the uncemented hip replacement - effect of femur characteristics and implant design geometry. Journal of Biomechanics 2010;43:512–520. [PubMed: 19896129]
8. Fernandes PR, Folgado J, Jacobs C, Pellegrini V. A contact model with ingrowth control for bone remodelling around cementless stems. Journal of Biomechanics 2002;35(2):167–176. [PubMed: 11784535]
9. Dunbar WL, Un K, Donzelli PS, Spilker RL. An Evaluation of Three-Dimensional Diarthrodial Joint Contact Using Penetration Data and the Finite Element Method. Journal of Biomechanical Engineering 2001;123(4):333–340. [PubMed: 11563758]
10. Cifuentes AO, Kalbag A. A performance study of tetrahedral and hexahedral elements in 3D finite element analysis. Finite Elements in Analysis and Design 1992;12:313–318.
11. Ramos A, Simoes J. Tetrahedral versus hexahedral finite elements in numerical modelling of the proximal femur. Medical Engineering and Physics 2006;28(9):916–924. [PubMed: 16464628]
12. Weingarten VI. The controversy over hex or tet meshing. Machine Design 1994:74–78.

13. Kluess D, Souffrant R, Mittelmeier W, Wree Andreas, Schmitz K, Bader R. A convenient approach for finite-element-analyses of orthopaedic implants in bone contact: Modeling and experimental validation. Computer Methods and Programs in Biomedicine 2009;95:23–30. [PubMed: 19231021]
14. Schonning A, Oommen B, Ionescu I, Conway T. Hexahedral mesh development of free-formed geometry: The human femur exemplified. Computer-Aided Design 2009;41:566–572.
15. Ito Y, Shih AM, Soni BK. Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates. International Journal for Numerical Methods in Engineering 2009;77:1809–1833.
16. Zhang Y, Hughes TJR, Bajaj CL. An automatic 3D mesh generation method for domains with multiple materials. Computer Methods in Applied Mechanics and Engineering 2010;199:405–415. [PubMed: 20161555]
17. Shepherd JF, Johnson CR. Hexahedral mesh generation for biomedical models in SCIRun. Engineering with Computers 2009;25:97–114.
18. Knupp P. Hexahedral and tetrahedral mesh untangling. Engineering With Computers 2001;17(3): 261–268.
19. Grosland NM, Shivanna KH, Magnotta VA, Kallemeyn NA, DeVries NA, Tadepalli SC, Lisle C. IA-FEMesh: An open-source, interactive, multiblock approach to anatomic finite element model development. Computer Methods and Programs in Biomedicine 2009;94(1):96–107. [PubMed: 19157630]
20. Shivanna KH, Grosland NM, Russell ME, Pedersen DR. Dirathrodial joint contact models: Finite element model development of the human hip. Engineering With Computers 2008;24(2):155–163.
21. Surazhsky V, Surazhsky T, Kirsanov D, Gortler SJ, Hoppe H. Fast exact and approximate geodesics on meshes. ACM SIGGRAPH 2005. 2005
22. Spekreijse SP. Elliptic grid generation based on laplace equations and algebraic transformations. Journal of Computational Physics 1995;118:38–61.
23. Shivanna, KH.; Tadepalli, SC.; Magnotta, VA.; Grosland, NM. A framework for finite element mesh quality improvement and visualization in orthopaedic biomechanics. ASME Summer Bioengineering Conference; 2009; Lake Tahoe, CA.
24. Field D. Laplacian smoothing and delaunay triangulations. Communications in Applied Numerical Methods 1988;4:709–712.
25. Munson M. Mesh shape-quality optimization using inverse mean-ratio metric. Math Program Ser A 2007;110:561–590.
26. Knupp PM. Algebraic mesh quality metrics. SIAM Journal on Scientific Computing 2001;23(1):193–218.
27. Papaioannou G, Nianios G, Mitrogiannis C, Fyhrie D, Tashman S, Yang KH. Patient-specific knee joint finite element model validation with high-accuracy kinematics from biplane dynamic Roentgen stereogrammetric analysis. Journal of Biomechanics 2008;41:2633–2638. [PubMed: 18675422]
28. Haut-Donahue TL, Hull ML, Rashid MM, Jacobs CR. A finite element model of the human knee joint for the study of tibio-femoral contact. Journal of Biomechanical Engineering 2002;124(3):273–280. [PubMed: 12071261]
29. Bendjaballah MZ, Shirazi-Adl A, Zukor DJ. Finite element analysis of human knee joint in varus-valgus. Clinical Biomechanics 1997;12(3):139–148. [PubMed: 11415685]
30. Hao Z, Jin D, Zhang Y, Zhang J. A finite element 3D model of in vivo human knee joint based on MRI for the tibiofemoral joint contact analysis. Lecture Notes in Computer Science 2007;4561:616–622.
31. Anderson AE, Ellis BJ, Maas SA, Peters CL, Weiss JA. Validation of finite element predictions of cartilage contact pressure in the human hip joint. Journal of Biomechanical Engineering 2008;130:1–10.
32. Baldwin MA, Clary C, Maltesky LP, Rullkoetter PJ. Verification of predicted specimen-specific natural and implanted patellofemoral kinematics during simulated deep knee bend. Journal of Biomechanics 2009;42:2341–2348. [PubMed: 19720376]
33. DeVries NA, Shivanna KH, Tadepalli SC, Magnotta VA, Grosland NM. IA-FEMesh: Anatomic FE models - A check of mesh accuracy and validity. Iowa Orthopaedic Journal 2009;29:48–54. [PubMed: 19742085]

34. Thompson, JF.; Soni, BK.; Weatherill, NP. Handbook of grid generation. CRC Press; 1999.
35. Ertbruggen C, Hirsch C, Paiva M. Anatomically based three-dimensional model of airways to simulate flow and particle transport using computational fluid dynamics. Journal of Applied Physiology 2005;98:970–980. [PubMed: 15501925]

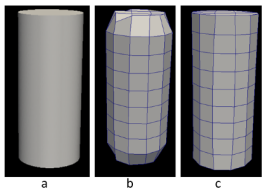

Fig. 1.

Cylinder with ends capped: (a) The triangulated surface, (b) the finite element mesh without capturing the cylindrical end caps, and (c) the finite element mesh generated by capturing the cylindrical end caps via feature edge definitions.

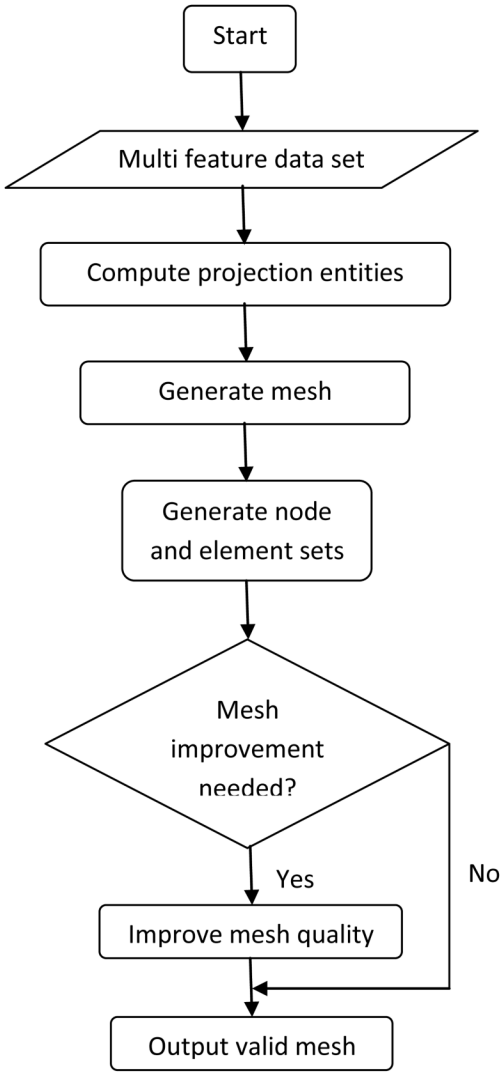

Fig. 2.
Flow chart of meshing workflow.

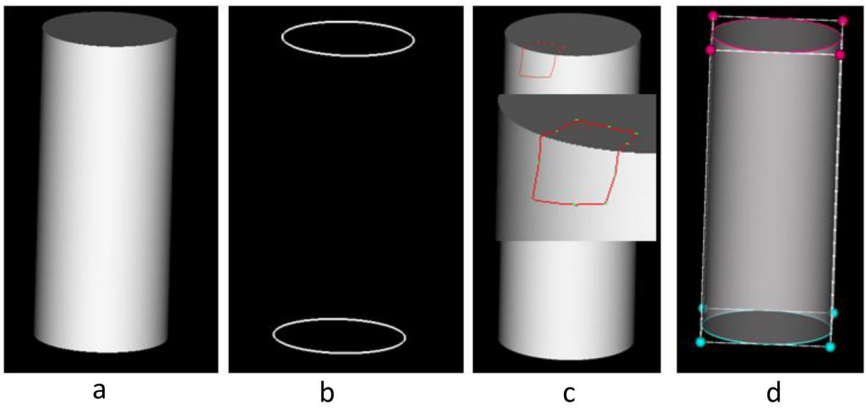

Fig. 3.

Feature edge definition: (a) The triangulated surface, (b) Automated feature edge extraction, (c) An example of a manually defined feature edge, (d) the proposed building block editing operation with color-coded feature dependent attributes.

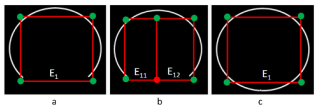

Fig. 4.

Building block edge definition for open contour, vertices colored green - designated to be projected: (a) Nodes on edge E_1 should not be projected because of open contour, (b) split the edge E_1 resulting in edges E_{11} and E_{12}, the resulting edges are not designated to be projected onto the feature edge, and (c) closed contour ensures satisfactory projection of edge E_1.

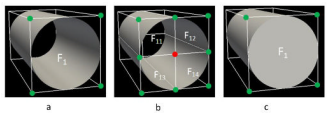

Fig. 5.

Building block face definition for open surface (cylinder without end caps), vertices colored green - designated to be projected: (a) Nodes on face F_1 should not be projected because of open surface, (b) split the face F_1 resulting in faces F_{11}, F_{12}, F_{13} and F_{14}, the resulting faces are not designated to be projected onto the surface as the center vertex is not designated to be projected, and (c) Patched surface ensures satisfactory projection of face F_1.

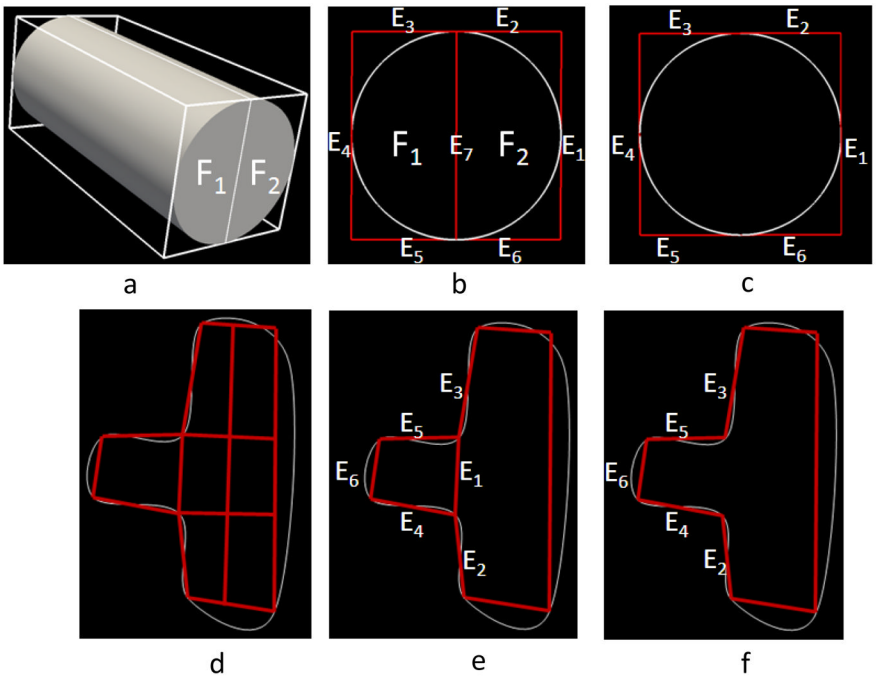

Fig. 6.

Computation of projection edges: (a) Block structure with two blocks with an edge common between faces $F_1$and F_2,(b) edges identified to be projected onto the feature edge, with interior edge E_7 falsely identified to be projected, (c) edge E_7 removed from the list, (d) interior edge not common to two faces, (e) list of projection edges with interior edge E_1 falsely identified to be projected, and (f) edge E_1 removed after the application of 'Number of neighbors rule'.

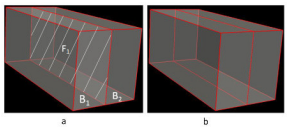

Fig. 7.
Computation of projection faces for block structure with two blocks: (a) Face F_1 falsely identified to be projected, and (b) updated projection face list after removal of face F_1.

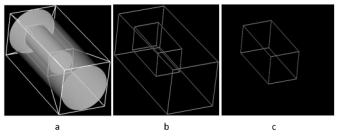

Fig. 8.
Computation of associated blocks: (a) Block structure for cylinder within a cylinder data set, (b) blocks associated with the outer cylinder, and (c) block associated with the inner cylinder.

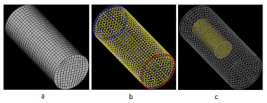

Fig. 9.
Automated creation of node sets: (a) Mesh corresponding to cylinder within a cylinder dataset, (b) nodes colored yellow associated with the outer surface(blue and red – associated with the feature edges), and (c) nodes associated with the inner surface.

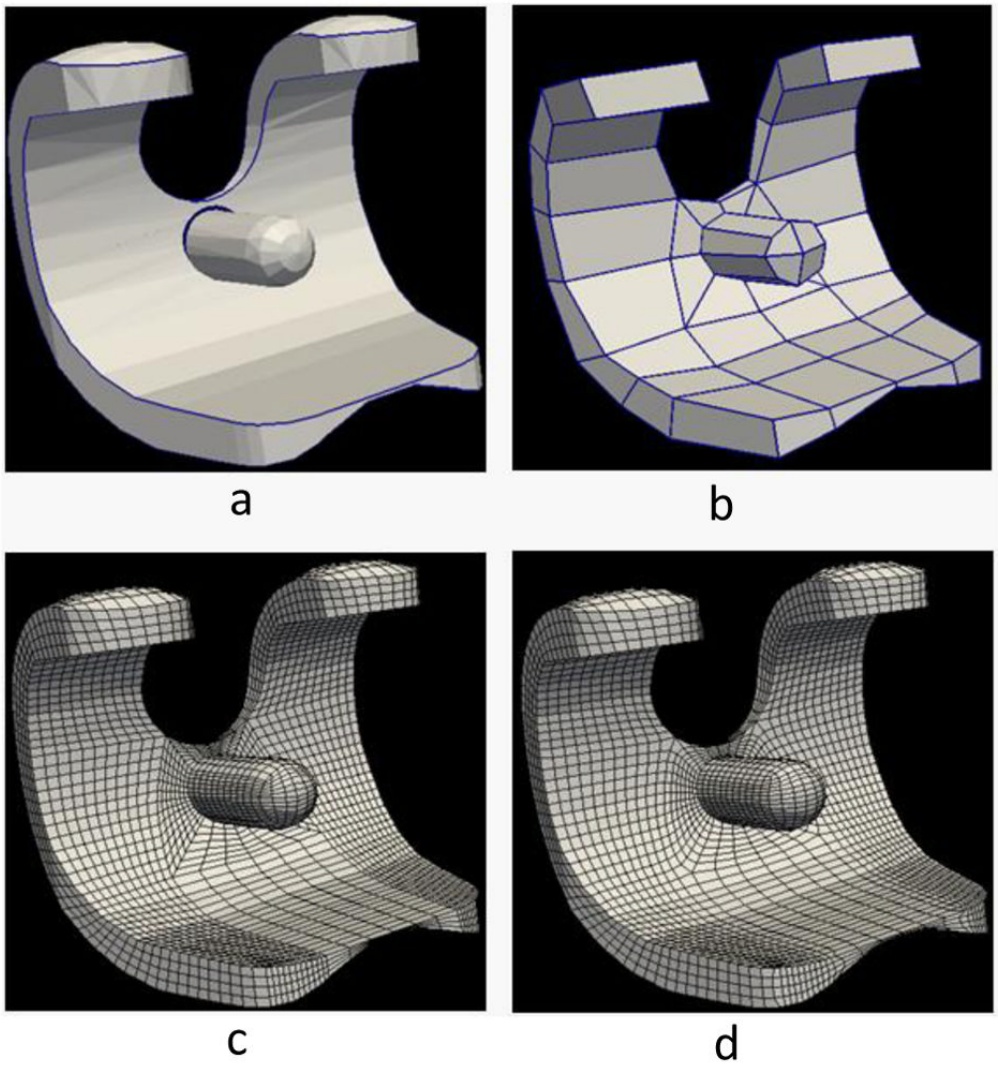

Fig. 10.
Femoral component of a total knee replacement device: (a) Triangulated surface definition (white) with feature edges (blue), (b) building block structure, (c) femoral component mesh before mesh improvement, and (d) after mesh improvement.

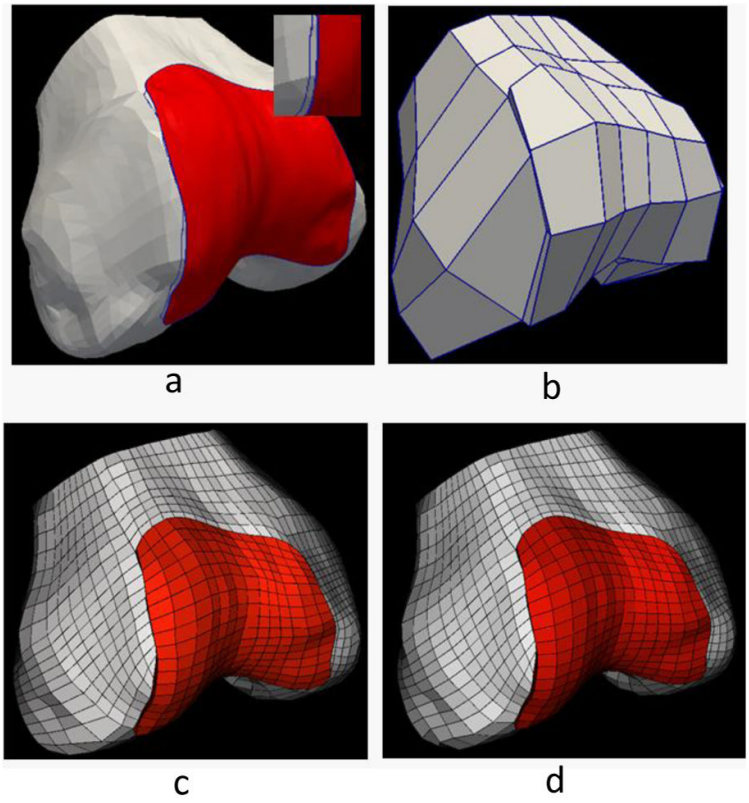

Fig. 11.

Distal femur with cartilage dataset: (a) Triangulated surface definition of femur (white), cartilage (red) and feature edges (blue), (b) building block structure, (c) corresponding mesh (femur(white) and cartilage(red)) before mesh improvement, and (d) after mesh improvement.

Table 1

Resulting mesh quality – Scaled Jacobian.

	# of Elements	Scaled Jacobian			
		Minimum	Maximum	Average	Std Dev
Cylinder	81	0.22	0.96	0.58	0.29
Cylinder (Multi-feature)	81	0.44	1.0	0.76	0.27
TKR	7682	0.05	0.99	0.85	0.13
Femur with Cartilage	4872	0.05	0.99	0.81	0.16