Development of the VHP-Female CAD model including Dynamic Breathing Sequence

by

Anh Le Tran

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Master of Science

in

Electrical and Computer Engineering

April 2017

APPROVED:

Dr. Sergey Makarov
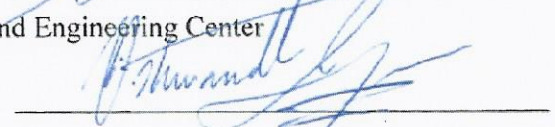Worcester Polytechnic Institute

Prof. Ara Nazarian
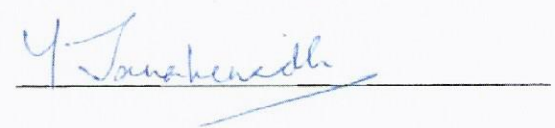Beth Israel Deaconess Medical Center, Harvard Medical School

Dr. Gregory Noetscher
US Army Natick Soldier Research, Development and Engineering Center

Dr. Vishwanath Iyer
The MathWorks, Inc.

Dr. Janakinadh Yanamadala
The MathWorks, Inc.

Development of the VHP-Female CAD model including Dynamic Breathing Sequence

by

Anh Le Tran

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Master of Science

in

Electrical and Computer Engineering

April 2017

APPROVED:

Dr. Sergey Makarov   _____
Worcester Polytechnic Institute

Prof. Ara Nazarian   _____
Beth Israel Deaconess Medical Center, Harvard Medical School

Dr. Gregory Noetscher   _____

US Army Natick Soldier Research, Development and Engineering Center

Dr. Vishwanath Iyer   _____

The MathWorks, Inc.

Dr. Janakinadh Yanamadala   _____

The MathWorks, Inc.

To my family and friends

Abstract

Mathematics, physics, biology, and computer science are combined to create computational modeling, which studies the behaviors and reactions of complex biomedical problems. Modern biomedical research relies significantly on realistic computational human models or "virtual humans". Relevant study areas utilizing computational human models include electromagnetics, solid mechanics, fluid dynamics, optics, ultrasound propagation, thermal propagation, and automotive safety research. These and other applications provide ample justification for the realization of the Visible Human Project® (VHP)-Female v. 4.0, a new platform-independent full body electromagnetic computational model.

Along with the VHP-Female v. 4.0, a realistic and anatomically justified Dynamic Breathing Sequence is developed. The creation of such model is essential to the development of biomedical devices and procedures that are affected by the dynamics of human breathing, such as Magnetic Resonance Imaging and the calculation of Specific Absorption Rate. The model can be used in numerous application, including Breath-Detection Radar for human search and rescue.

I defend:
1. Creation of the breathing sequence on the CAD model VHP-Female v3.1
2. Development of the full-body CAD model VHP-Female v4.0

List of Figures

List of Tables

List of Published and Accepted Papers

1. **A. L. Tran**, and S. N. Makarov, "Degree of RF MRI Coil Detuning for an Anatomically Realistic Respiratory Cycle Modelled with the Finite Element Method", *39th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (EMBC 2017), Jeju Island, Korea, Jul. 11-15 2017 (accepted).

2. **A. L. Tran**, and S. N. Makarov, "Degree of RF MRI Coil Detuning and SAR Variations over an Anatomically Realistic Respiratory Cycle Modelled with the Finite Element Method", *25th Annual Meeting if the International Society for Magnetic Resonance in Medicine* (ISMRM 2017), Honolulu, HI, Apr. 22-24 2017 (accepted).

3. **A. L. Tran**, G. Noetscher, S. Louie, A. Prokop, A. Nazarian, and S. N. Makarov, "FEM Human Body Model with Embedded Respiratory Cycles for Antenna and E&M Simulations," *2016 Antennas Appl. Sym.*, Monticello, IL, Sep. 20-22 2016.

4. **A. L. Tran**, R. Borwankar, H. Tankaria, G. N. K. Srichandhru, J. Yanamadala, S. Louie, and S.N. Makarov, "Construction of Approximate Breathing Sequence for a CAD Human Model via a Native Parametric Sweep in ANSYS," *38th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (EMBC 2016), Orlando, FL, Aug. 16-20 2016.

5. H. Tankaria, X. J. Jackson, R. Borwankar, G. N. K. Srichandhru, **A. L. Tran**, J. Yanamadala, G. M. Noetscher, A. Nazarian, S. Louie, and S.N. Makarov, "VHP-Female Full-Body Human CAD Model for Cross-Platform FEM Simulations – Recent Development and Validations," *38th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (EMBC 2016), Orlando, FL, Aug. 16-20 2016.

6. J. Yanamadala, R. Borwankar, G. N. K. Srichandhru, **A. L. Tran**, H. V. Tankaria, X.J. Jackson, S.N. Makarov, and A. Pascual-Leone, "Full-Body FEM Pregnant Woman Model and Applications," *38th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (EMBC 2016), Orlando, FL, Aug. 16-20 2016.

7. J. Yanamadala, V. K. Rathi, S. Maliye, H. A. Win, **A. L. Tran**, M. Zagalskaya, G. M. Noetscher, S. N. Makarov, "Full-Body Visible Human Project Female Computational Phantom and Its Applications for Biomedical Electromagnetic Modeling", *The IEEE Signal Processing in Medicine and Biology Symposium (SPMB14)*, Philadelphia, Pennsylvania, USA, 2014.

8. J. Yanamadala, V. K. Rathi, S. Maliye, H. A. Win, **A. L. Tran**, M. K. Kozlov, G. M. Noetscher, A. Nazarian, S.N. Makarov, "Segmentation of the Visible Human Project (VHP) Female Cryosection Images within MATLAB Environment", *23rd International Meshing Roundtable*, London, 2015.

Table of Contents

I. Chapter 1: Introduction and Review of Prior Work

1.1. Development of Human Model

Virtual humans are full-body human models which can be brought to "life" in the computer. A high-resolution model includes in excess of 200 individual tissue parts and all major body systems. There are about 40 detailed anatomical virtual humans in the world to date (see Table 1.1). These models are increasingly used in regulatory safety research and modern biomedical design since they respond *in a similar manner* to real humans.

Table 1. 1. Major Anatomical Full-Body Human Models for CEM and Radiological Simulations (After 2004) [1]

Entity/Country	Model Name	G/A/H/W	Da	TYPE	RES, mm²	FV	D
IT'IS Found. Switzerland	Glenn	m/84/173/61.1	N	V	0.5x0.5x1.0h 0.9x0.9x2b	N	Y
IT'IS Found. Switzerland	FATS	m/37/182/119	N	V	0.5x0.5x1.0h 0.9x0.9x2b	N	Y
IT'IS Found. Switzerland	DUKE	m/34/177/70.3	N	V/S	0.5x0.5x1.0h 0.9x0.9x2b	Y	Y
IT'IS Found. Switzerland	ELLA	f/26/163/57.3	N	V/S	0.5x0.5x1.0h 0.9x0.9x2b	Y	Y
IT'IS Found. Switzerland	LOUIS	m/14/168/49.7	N	V	0.5x0.5x1.0h 0.9x0.9x2b	N	Y
IT'IS Found. Switzerland	BILLIE	f/11/149/34.0	N	V/S	0.5x0.5x1.0h 0.9x0.9x2b	Y	Y
IT'IS Found. Switzerland	EARTHA	f/8/136/29.9	N	V	0.5x0.5x1.0h 0.9x0.9x2b	N	Y
IT'IS Found. Switzerland	DIZZY	m/8/137/25.4	N	V	0.5x0.5x1.0h 0.9x0.9x2b	N	Y
IT'IS Found. Switzerland	THELONIOUS	m/6/115/18.6	N	V/S	0.5x0.5x1.0h 0.9x0.9x2b	Y	Y
IT'IS Found. Switzerland	ROBERTA	f/5/109/17.8	N	V	0.5x0.5x1.0h 0.9x0.9x2b	N	Y
IT'IS Found. Switzerland	NINA	f/3/92/13.9	N	V	0.5x0.5x1.0h 0.9x0.9x2b	N	N
IT'IS Found. Switzerland	CHARLIE	f/8w/na/4.3	N	V	0.5x0.5x1.0h 0.9x0.9x2b	N	N
China Acad. of Tel. Res.	CHINESE MALE	m/35/172/64	Y	V/S/CAD	1x1x1	N	N
China Acad. of Tel. Res.	CHINESE FEMALE	f/22/162/54	Y	V/S/CAD	1x1x1	N	N

			Da	TYPE	RES	FV	D
Huazhong Univ, China	CDH M2	m/166	Y	V/S/CAD	0.1x0.1x0.2	N	N
Huazhong Univ, China	CHINESE REF. MAN	m/166	Y	S/CAD	2x2x2	N	Y
Natl. Inst. of Inform. and Comm. Technol., Japan	NAGAOKA MAN (TARO)	m/22/173/65	N	V	2x2x2	N	N
Natl. Inst. of Inform. and Comm. Technol., Japan	NAGAOKA WOMAN	f/22/160/53	N	V	2x2x2	N	N
ETRI, Korea	KOREAN MAN	m/21/176/67	N	V	1x1x1 (head) 3x3x3 (body)	N	N
ETRI, Korea	KOREAN CHILD	m/7/122.4/25.5	N	V	1x1x3	N	N
Nat. Radiological Protection Board, UK	NAOMI	f/23/163/60	N	V	2x2x2	N	N
REMCOM, PennState	Male/Female	m/f	Y	V	5x5x5 (both)	N	Y
CST AG, Germany	BABY	f/8w/57/4.2	N	V	0.85x0.85x4	N	N
CST AG, Germany	CHILD	f/7/115/21.7	N	V	1.54x1.54x8	N	N
CST AG, Germany	DONNA	f/40/176/79	N	V	1.88x1.88x10	N	N
CST AG, Germany	EMMA	f/26/170/81	N	V	0.98x0.98x10	N	N
CST AG, Germany	GUSTAV	m/38/176/69	N	V	2.1x2.1x8	N	N
CST AG, Germany	LAURA	f/43/163/51	N	V	1.88x1.88x5	N	N
CST AG, Germany	HUGO, posable	m/38/187/113	Y	V	1x1x1	N	Y
U Texas Austin, USA	AUSTIN MAN	m/38/187/113	Y	V	1x1x4	Y	Y
U Texas Austin, USA	AUSTIN WOMAN	f/60/162/88	Y	V	1x1x4	Y	Y
Duke University Medical Center, USA	XCAT FAMILY	Orig: f/60/162/88 m/38/187/113	Y	CAD, NURBS	variable	Y	Y
NEVA EM LLC, WPI, BIDMC	VHP-Female	f/60/162/88	Y	CAD	variable		

Abbreviations: G/A/H/W – Gender/Age/Height/Weight; Da - Original image dataset made available for independent evaluation (Y/N); TYPE (V – voxel; S – surface-based model, but without explicit FEM compatibility; CAD – surface-based FEM-compatible model); RES – Lowest image resolution (before or after postprocessing) of the model declared by the provider (h=head, b=body); FV - Free version for available (Y/N); D – Deformable/posable (Y/N);

However, the majority of these available virtual humans are *voxel models* made of millions of small equal cubes or bricks. Voxel models cannot be simulated with the Finite-Element Method (FEM). Voxel models are also "cast in stone"; they cannot be easily deformed. Therefore, there is a need for *Computer Aided Design* (CAD) human models made of surface-based objects – individual tissues and tissue parts – and compatible with the FEM. The CAD virtual humans can breathe like real humans, move like real humans, and can also suffer from injuries and have various implants.

In electromagnetics applications, performance safety and evaluations of medical devices are developed with computational human model [2]. These devices include electrophysiology monitoring devices such as Electroencephalogram (EEG), Electrocardiogram (ECG), Magnetoencephalography (MEG), and Magnetic Resonance Imaging (MRI) systems, MR conditional implanted devices, radiofrequency

ablation, optical coherence tomographic devices, fluorescence spectroscopy devices, laser surgery devices, and optical therapy devices. Moreover, computational human models are used in antenna research such as the safety and performance evaluations of cellphone radiation, implanted antennas, and microwave imaging systems.

One major application area is the impact of MRI, cellphones, and electromagnetic radiation on the human body. There are only a few anatomical *true-CAD* virtual humans in the world directly applicable to *electromagnetic safety and performance evaluations*. Those are the XCAT Family of Dr. W. P. Segars (Duke) and the Chinese Male/Female from China Academy of Tel. Research. The XCAT family based on B-spline modeling has limited tissue parts.

1.2. Use of the model

Regulatory organizations (e.g. FDA, FCC, or IEC) require the determination of the Specific Absorption Rate (SAR) for *safety assessment* of biomedical electromagnetic devices *prior to going to market*. Such devices include MRI coils, Electroencephalography (EEG), Electrocardiography (ECG), Magnetoencephalography (MEG), brain-computer interfaces, MR conditional passive or active implanted devices, neurostimulators and electrical stimulators, devices for radio frequency ablation, cellphone antennas, and on-body and implanted antennas.

However, SAR measurements are not easily performed for human subjects *in-vivo*. To put it simply, we cannot drill a hole in a body to do the necessary measurements. Therefore, SAR and device performance estimates are typically derived from Computational Electromagnetics (CEM) simulations performed with virtual humans. Other emerging applications include modern automotive safety studies especially related to *self-driving cars*. Although vehicular crash safety performances are evaluated with crash test dummies regulated by federal agencies (U.S. Dept. of Transportation Fed. Highway Administration), it is believed that the impact biomechanics with *CAD virtual humans* would provide significantly better understandings of injury mechanisms.

Many in the industry and academia today either lack CAD human models or use non-anatomical models, resulting in a need that is either met inadequately or not met at all. Accurate virtual humans are difficult and time consuming to create. Consequently, the models are expensive – for example, a license to use the modern virtual human family product from the IT'IS Foundation of Switzerland can easily run up to $60,000 annually.

The need for CAD virtual human body models in industry and academia is currently met on a very limited basis. Examples of prospective customers are: MathWorks, Inc [3]; COMSOL, Inc [4]; ANSYS, Inc [5]; CST AG [6]; IMST GmbH [7]; Remcom, Inc. [8]; Altair Engineering, Inc. [9]; Toyota Central R&D Labs, Inc. [10]

To demonstrate that the model is able to satisfy this need, Fig. 1.1 compares two recent virtual human models. On the left, a soldier CAD model (called *anatomy avatar*) from the U.S. Army Research Institute of Environmental Medicine (March 2016, [11]) is shown. Such models could make an impact on the design of military gear, protective equipment and vehicles, and other applications across military medicine [11]. On the right, the CAD VHP-Female v.3.1 model is shown (March 2016) with the resolution of 1-7 mm and 250 individual parts, including blood vessels and state-of-the art peripheral nerves. Although the USARIEM model is subject-specific, which makes its semi-automatic construction much more complicated, the difference in the anatomical quality and accuracy of both models is remarkable.
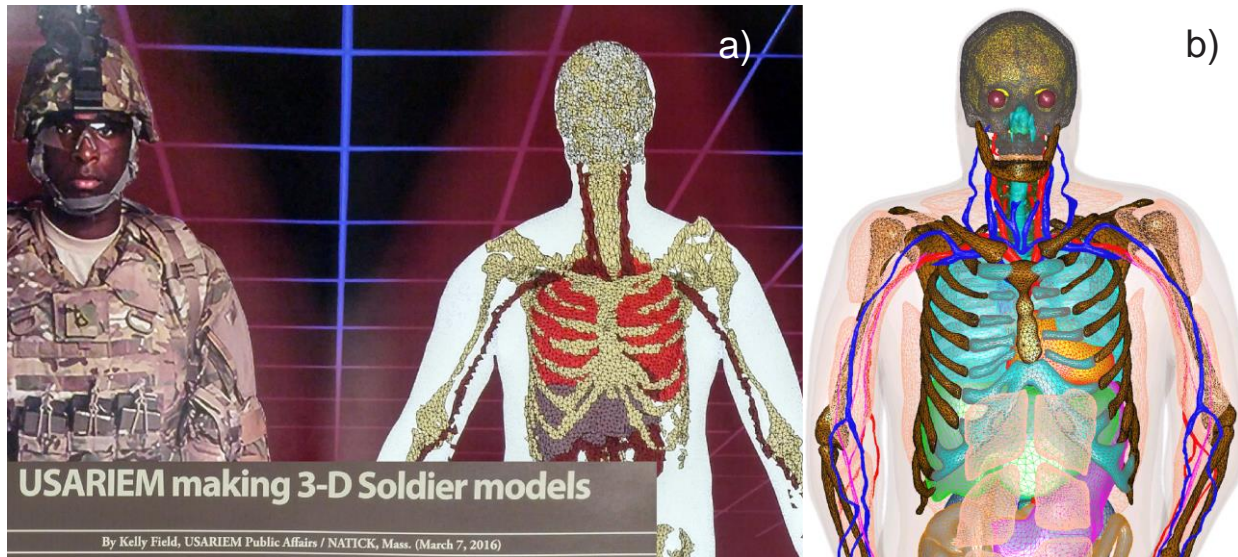
Figure 1. 1. A) Left – soldier CAD model from U.S. Army Research Institute of Environmental Medicine (March 2016); b) Right – VHP-Female v. 3.1 CAD human model with major blood vessels and peripheral nerves.

1.3. Advantages of VHP Female

A. Cross-platform compatibility of our CAD models

The majority of available domestic and international human models are *voxel models*. The voxel model is a very large set of very small cubes – unit cells – with different material properties characterizing different tissues. This means that every tissue is a volume inside a watertight curved shell consisting of many small triangles. The voxel human models can be simulated with the help of time-domain electromagnetic solvers but cannot be simulated via Finite-Element Modeling (FEM). The CAD models, on the other hand, are appropriate for both FEM and time-domain methods, and are thus *cross-platform* (ANSYS Electronic Desktop, MATLAB, CST, FEKO). In other words, CAD models are compatible with the simulation software of *all target customers*. The significance of this *critical* feature is validated by the ability to serve *all targeted customers simultaneously*.

B. Anatomical accuracy of our CAD models

The existing VHP-Female human model has 26 tissues and 250 tissue parts; the resolution in the cranium is 1-3 mm, and 1-7 mm in the main body.

C. True CAD of our models

Even though the IT'IS Foundation models look excellent at first sight [12], they are not true CAD models. The shells representing different tissues are not watertight and intersect. A joint check of the DUKE model with ANSYS, Inc. [12] generated approximately 100 intersection/non-manifold errors. Therefore, these models are *not FEM compatible* and their parts cannot be 3D printed. The significance of this feature is validated by the fact that existing educational and commercial models run flawlessly using FEM – the major simulation tool for biomechanics and other applications – and by the fact that any part of them could be 3D printed (have previously printed femur bones for experimental analysis).

D. Dynamic behavior of the CAD models

The real human *in-vivo* is a *dynamic model* subject to breathing, blood flow, and movement. The respiratory motion is one major cause of image blurriness in MRI systems, even with respiratory triggered sequences. The Swiss Virtual Family mentioned above does not have this difficult and complicated model extension.

E. Access to the original image dataset

All virtual humans are constructed by segmentation of a stack of images. For models, the original high-resolution image dataset is available to *any user* or to any interested person through the U.S. National Library of Medicine [13]. However, all full-body models distributed by IT'IS Foundation, Switzerland *do not make* the original datasets available to users. Therefore, we simply do not know their

anatomical accuracy. The significance of this feature might potentially result in making models to be a completed *standard* for radio-frequency safety estimates.

1.4. Human Model Construction

Computational phantoms or virtual humans are created via a set of 3D mathematical algorithms commonly called *image segmentation*. Segmentation is one of the most studied problems in the field of biomedical image analysis. Consider one body image (a slice in the xy-plane) in Fig. 1.2a which shows a cross-section of a human leg, including the patella [14]. The complete stack of images continues in the z-direction. A skilled operator traces the patella boundary with a set of discrete points in the xy-plane (a polygon) shown by crosses in Fig 1.2a. A z-coordinate corresponding to the global coordinate system is added. Then, another cross-sectional image is traced, and all 3D points are collected, image by image. The end result is a complete patella boundary in three dimensions given in the form of a point cloud shown in Fig. 1.2b. This process is known as *manual segmentation*, still the "gold standard" of image segmentation. Other tissues are segmented similarly. The inner volume of the point cloud is either empty or can be filled with a set of (uniformly distributed) inner nodes. In the latter case, we arrive at a volumetric *voxel model* of a tissue, which is a *typical final result* of image segmentation. A large number of such volumes may exist, either interconnected or separate.
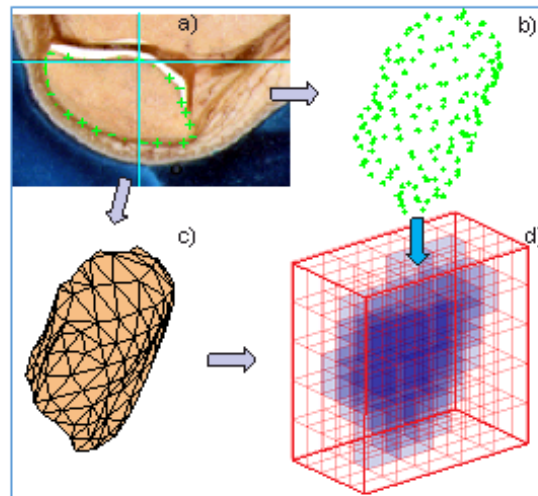
Figure 1. 2. A) – Image of a patella with a traced boundary; b) – resulting point cloud; c) – patella CAD model; d) – patella voxel model.

A true CAD model must meet the following conditions:

A. It must be watertight, i.e. it does not have surface holes
B. The surface of the mesh must be manifold. A mesh is 2-manifold if every node of the mesh has a disk-shaped neighborhood of triangle, which can be continuously deformed to an open disk. Every edge of a 2-manifold mesh is a manifold edge with only two attached triangles. All other meshes are non-manifold meshes and are not suitable for FEM analysis. Fig. 1.3 gives examples of a non-manifold mesh with a non-manifold edge and a non-manifold mesh with a non-manifold node. [15]
C. Different tissue meshes (in Fig. 1.4) must not intersect
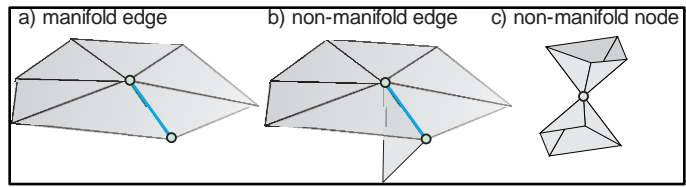
Figure 1. 3. A) – Examples of a manifold edge; b) – non-manifold edge, and c) – non-manifold node.
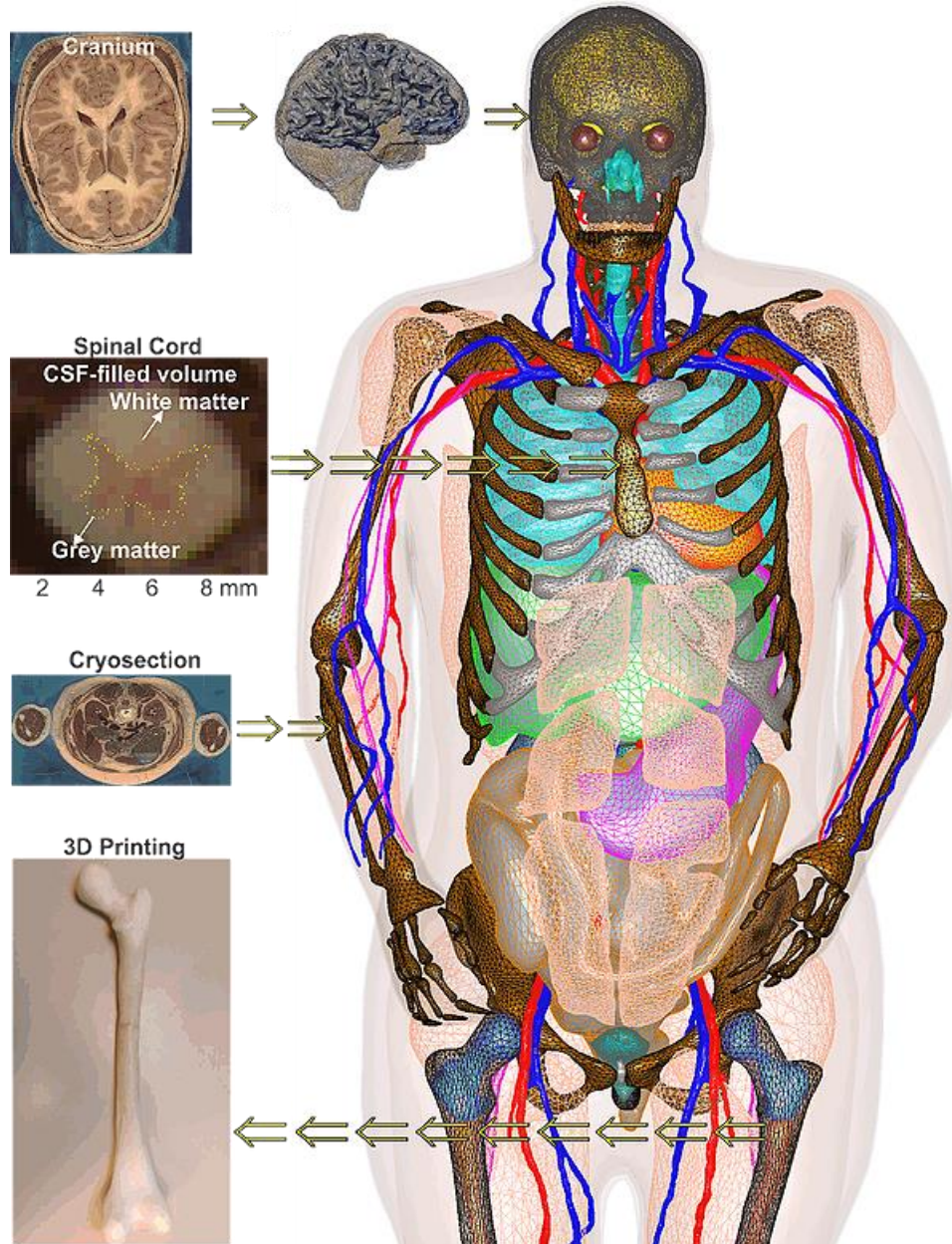
Figure 1. 4. VHP-Female CAD representation

References

[1]. A. Nazarian et al, "Application of Virtual Humans in Medial Research: VHP-Female CAD Model" *AP Magazine Article*

[2]. U.S. Food and Drug Administration, Center for Devices and Radiological Health "Reporting of Computational Modeling Studies in Medical Device Submissions: Draft Guidance for Industry and Food and Drug Administration Staff," Jan 17th 2014.

[3]. MathWorks, Inc. Available: http://www.mathworks.com/

[4]. COMSOL, Inc., Available: https://www.comsol.com/

[5]. ANSYS Website. Available: http://www.ansys.com/

[6]. Computer Simulation Technology (CST) Allgemeine Gesellschaft (AG) Website. Available: https://www.cst.com/

[7]. IMST GmbH Website. Available: https://www.imst.de/

[8]. Remcom, Inc. Available: http://www.remcom.com/

[9]. Altair Engineering, Inc. Available: http://www.altair.com/

[10]. Toyota Central R&D Labs., Inc. Available: http://www.tytlabs.com/

[11]. USARIEM Models. Available: https://www.rt.com/usa/338688-us-army-avatar-soldiers/ ; http://www.army-technology.com/news/newsus-army-to-develop-3-d-full-anatomy-avatar-of-soldiers-4859541

[12]. Available: http://www.itis.ethz.ch/virtual-population/virtual-population-cvip-vip/cvip3-and-vip1/duke/duke-cv3-1-1/

[13]. M. J. Ackerman, "The Visible Human Project," Proc. IEEE, vol. 86, no. 3, pp. 504-511, 1998. Available: https://www.nlm.nih.gov/research/visible/visible_human.html

[14]. V. Spitzer, M. J. Ackerman, A. L. Scherzinger, and D. W. Whitlock, "The Visible Human Male: A Technical Report," J. of the Amer. Medical Informatics Assoc., vol. 3, no. 2, pp.118-130, 1996.

[15]. S. N. Makarov et al, "Virtual Humans Models for Electromagnetic Studies and Their Applications,"

[16]. IT'IS Foundation. Available: http://www.itis.ethz.ch/virtual-population/

[17]. IT'IS Foundation: Services. Available http://www.itis.ethz.ch/services/

[18]. J. Yanamadala, V.K. Rathi, S. Maliye, H.A. Win, A.L. Tran, M. K. Kozlov, G.M. Noetscher, A. Nazarian, and S.N. Makarov, "Segmentation of the Visible Human Project® (VHP) Female Cryosection Images within MATLAB® Environment," *23rd International Meshing Roundtable (IMR23)*, London, England, Oct. 12-15, 2014. Available: http://www.imr.sandia.gov/papers/imr23.html

[19]. A. M. Helderman, N. S. Thang, T. Dolma, T. T. Trinh, M. W. Piazza, J. Zhang, S. Xie, J. Yanamadala, and S. N. Makarov, "Using MATLAB® Platform for Image Segmentation and FEM-Compatible Triangular Surface Mesh Generation –Vertebral Column of Visible Human Project – Female", *37th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (EMBC 2015), Milano, Italy, Aug. 25-29 2015, pp. 3237-3241.

[20]. J. Yanamadala, G. Noetscher, V. Rathi, S. Maliye, H.A. Win, A.Le Tran, X.J. Jackson, A.T. Htet, M. Kozlov, A. Nazarian, S. Louie, and S. Makarov, "New VHP-Female V. 2.0 Full-Body Computational Phantom and Its Performance Metrics Using FEM Simulator ANSYS HFSS," *37th Ann. Int. Conf. of the IEEE Eng. In Medicine and Biology Soc.* (EMBC 2015), Milano, Italy, Aug. 25-29 2015, pp. 3237-3241.

[21]. H. Tankaria, X.J. Jackson, R. Borwankar, G.N.K. Srichandhru, A.L. Tran, J. Yanamadala, G. M. Noetscher, A. Nazarian, S. Louie, and S.N. Makarov, "VHP-Female

Full-Body Human CAD Model for Cross-Platform FEM Simulations – Recent Development and Validations," *38ᵗʰ Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (EMBC 2016), Orlando, FL, Aug. 16-20 2016.

[22]. H. Tankaria, X.J. Jackson, R. Borwankar, G.N.K. Srichandhru, A.L. Tran, J. Yanamadala, S.N. Makarov, and A.-P. Leone, "Full-Body FEM Pregnant Woman Model and Applications," *38ᵗʰ Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (EMBC 2016), Orlando, FL, Aug. 16-20 2016.

II. Chapter 2: Mesh Processing Algorithms

Introduction

This Chapter reviews selected basic facts related to triangular surface mesh generation for computational electromagnetics problems. Understating mesh generation, mesh properties, possible mesh errors, and the process of adaptive mesh refinement is an important part of any intelligent electromagnetic computations. Most of the materials in this chapter are attributed to my Advisor Sergey Makarov's book:

S. N. Makarov, G. N. Noetscher, and A. Nazarian, "Low-Frequency Electromagnetic Modeling for Electrical and Biological Systems Using MATLAB", Wiley, New York, June 2015, 648 p., Chapter 2. Triangular surface mesh generation.

Section 2.1 defines a triangular mesh, discusses structured and unstructured meshes, introduces the meaning of a 2-manifold mesh, and gives the metrics for triangle quality, triangle size, and mesh uniformity.

Section 2.2 introduces the basic 2D Delaunay triangulation algorithm. It gives examples of unconstrained and constrained Delaunay triangulation. It also introduces three-dimensional tetrahedral mesh generation. Perhaps the most important part of this section is three-dimensional surface mesh generation applied in particular to biomedical imaging. Biomedical imaging methods and image processing represent a huge and rapidly developing area of research.

Section 2.3 begins with a discussion of basic mesh operations (translation, rotation, scaling). Laplacian smoothing is then studied, including several permutations of the algorithm. Section 2.3 introduces Boolean mesh operations. Section 2.3 finishes with the description of two "collision" algorithms: ray-triangle intersection and segment-triangle intersection. These algorithms have a broad application space. For example, they may be used to find surface normal vectors and evaluate intersections of two distinct triangular surface meshes, triangle by triangle. The latter method is the foundation for performing Boolean operations on realistic surface meshes of human tissues.

Section 2.4 introduces perhaps the most important computational aspect of mesh transformation: the process of adaptive mesh refinement. The physical motivation for adaptive mesh refinement is discussed. Laplacian smoothing may be employed to maintain mesh quality.

2.1. Triangular mesh and its quality
2.1.1. Arrays of vertices and faces. Structured meshes
Arrays of vertices and faces

A triangular surface mesh is the base of any surface representation including various numerical methods in electrical and biomedical engineering, computer graphics, etc. Consider a planar rectangle on the size $a{\times}b$ in the xy-plane shown in Fig. 2.2. The goal is to "cover" its surface with triangles – *simplexes* in 2D. Many ways of doing so exist. One such way is shown in Fig. 2.1.
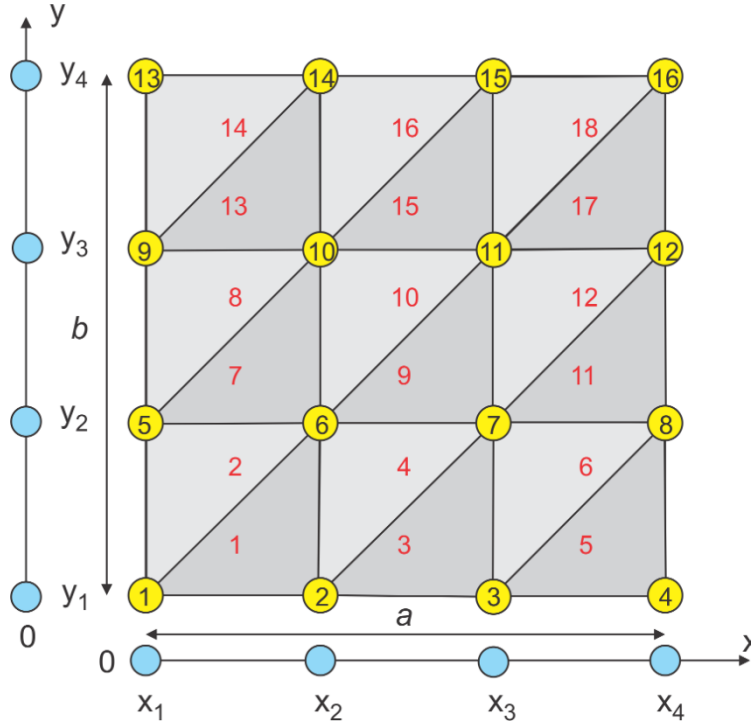
Figure 2. 1. Mesh generation for a planar rectangle.

We first define uniformly spaced *x*-nodes and uniformly spaced *y*-nodes. Assume that there are $N_x + 1$ nodal points along the *x*-axis and $N_y + 1$ nodal points along the *y*-axis. In Fig.2.1, $N_x = 3$, $N_y = 3$. In a general case, one has

$$x_m = \left(\frac{m-1}{N_x}\right)a, \quad m = 1,...,N_x + 1$$

$$y_n = \left(\frac{n-1}{N_y}\right)b, \quad n = 1,...,N_y + 1$$

(2.1)

There is a common way of describing triangular meshes, which originates from old NASTRAN programs written in the 1970s and 1980s. In order to define a triangular mesh, we need the *array of vertices* (or nodes), *P*. This array consists of rows; every row includes three Cartesian coordinates of the corresponding nodal point. The row number in array *P* is simply the vertex number. Further, we need an *array of faces* (or triangles), *t* also called *connectivity matrix*. This array also consists of rows; every row includes three integer numbers of triangle vertices; each such number is simultaneously the row number

of the array P. The row number in the array t is the number of the triangular face. For the mesh shown in Fig. 2.1, there are 16 nodes and 18 triangles, giving:

$$
P = \begin{bmatrix}
x_1 & y_1 & 0 \\
x_2 & y_1 & 0 \\
x_3 & y_1 & 0 \\
x_4 & y_1 & 0 \\
x_1 & y_2 & 0 \\
x_2 & y_2 & 0 \\
x_3 & y_2 & 0 \\
x_4 & y_2 & 0 \\
x_1 & y_3 & 0 \\
x_2 & y_3 & 0 \\
x_3 & y_3 & 0 \\
x_4 & y_3 & 0 \\
x_1 & y_4 & 0 \\
x_2 & y_4 & 0 \\
x_3 & y_4 & 0 \\
x_4 & y_4 & 0
\end{bmatrix}
\qquad
t = \begin{bmatrix}
1 & 2 & 6 \\
1 & 5 & 6 \\
2 & 3 & 7 \\
2 & 6 & 7 \\
3 & 4 & 8 \\
3 & 7 & 8 \\
... \\
11 & 15 & 16
\end{bmatrix}
\Rightarrow t = \begin{cases}
\text{odd row :} \\
\big[m,\, m+1,\, m+N_x+2\big] + (n-1)(N_x+1) \\[2mm]
\text{even row :} \\
\big[m,\, m+N_x+1,\, m+N_x+2\big] + (n-1)(N_x+1) \\
m = 1,...,N_x, \quad n = 1,...,N_y
\end{cases}
$$

$$(2.2)$$

Structured mesh

The triangular mesh shown in Fig. 2.1 is called a *structured surface mesh*. In a structured mesh, the indices of the neighbor vertices for any particular vertex could in principle be calculated using a simple addition rule – see Fig. 2.1 and Eq. (2.2). As long as this rule is known, the array of triangular patches, t, is not really necessary. However, in *unstructured meshes*, studied in the next section, we must use a list of each node's neighbors – the connectivity array, t.

Non-uniform mesh

Often, it is desired to increase the triangle density close to the certain areas, for example, close to rectangle edges. In this case, Eqs. (2.1), which are the generating equations for nodal points, may be modified accordingly. For example, the generator

$$
x_m = \left[1 - \cos\left(\frac{m-1}{N_x} \right) \right] \frac{a}{2}, \quad m = 1,...,N_x + 1
$$

$$
y_n = \left[1 - \cos\left(\frac{n-1}{N_y} \right) \right] \frac{b}{2}, \quad n = 1,...,N_y + 1
$$

$$(2.3)$$

will create the triangular mesh shown in Fig. 2.2. Note that the array, t, remains the same for meshes in both Fig. 2.1 and Fig. 2.2.
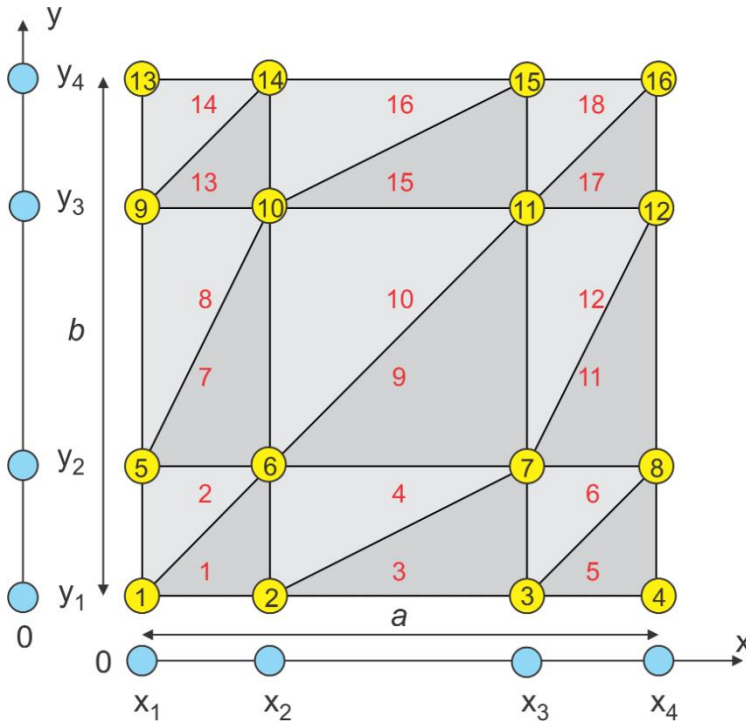
Figure 2. 2. Mesh generation for a planar rectangle with non-uniform nodes.

Mesh storage formats

Many modern mesh storage formats also include *edge information*. For example, the *winged edge format* used in computer graphics for each edge gives: two vertices of the edge, two faces attached to the edge, and the four edges attached to the edge of interest. Indeed, the edge information can be retrieved from the face information and vice versa. The above format is only valid for manifold meshes as explained in the following text.

2.1.2. A 3D triangular mesh. 2-Manifold meshes

When a third spatial coordinate z is added to the array of nodes in Eq. (2.2), a 3D triangular mesh is obtained. Figure 2.3 shows a triangular mesh for a sphere. This mesh has also been generated "by hand". The idea here is to use an initial octahedron mesh, then divide each edge of the mesh, and push all the new vertices in the direction of their outer normal so that they all belong to the sphere surface. This process is repeated as long as necessary. A problem with the triangular mesh shown in Fig. 2.3 is in slightly different triangle sizes on the sphere surface, which is a deficiency if a *uniform* mesh is needed with the approximately equal triangles.
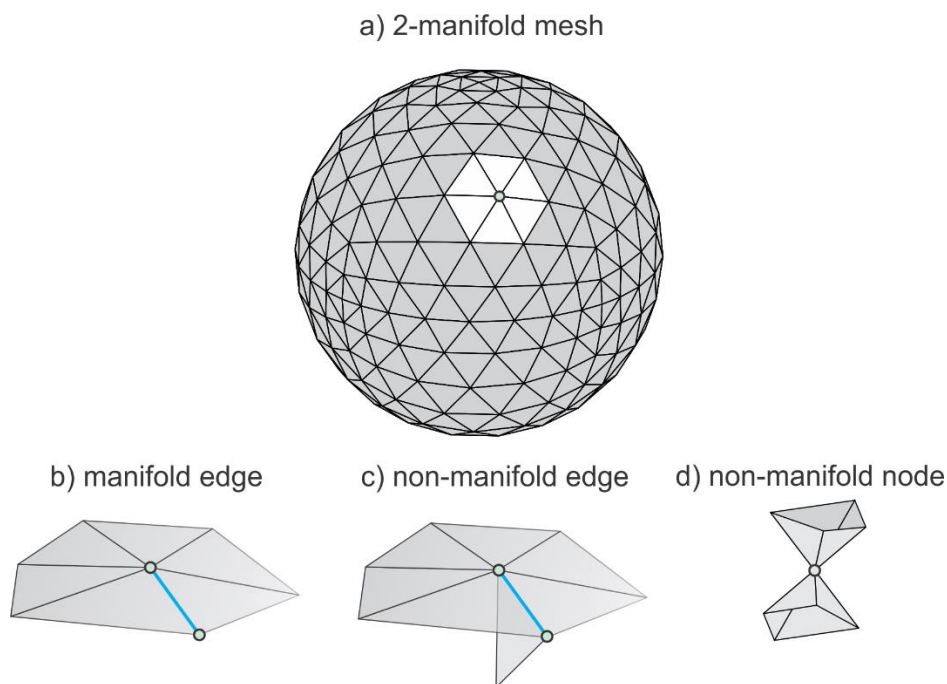
Figure 2. 3. A) – A 3D triangular mesh for a sphere with 512 triangles – a 2-manifold mesh; b) – manifold edge; c) – non-manifold edge; d) – non-manifold node.

The 3D triangular meshes are most important for the numerical analysis, computer graphics, and pattern recognition. The following two properties are of note:

1. A 3D triangular mesh usually represents a physical solid object. Therefore, it must be *watertight*, i.e., does not have missing triangles (surface holes).
2. The surface of a well-behaved triangular mesh in 3D must satisfy one critical condition, which is the so-called *manifold condition*. A mesh is *2-manifold* if every node of the mesh has a disk-shaped neighborhood of triangles – see Fig. 2.3a. This neighborhood can be continuously deformed to an open disk. Every edge of a 2-manifold mesh is a *manifold edge* – see Fig. 2.3b – with only two attached triangles.
3. All other meshes *are non-manifold meshes*. Fig. 2.3c gives an example of a non-manifold mesh with a *non-manifold edge*. A non-manifold mesh with a *non-manifold node* is shown in Fig. 2.3d.

The non-manifold meshes cannot be used for the numerical analysis, in most of the cases. Therefore, they must be *healed* prior to computations.

The definition of 2-manifold meshes is derived from the definition of 2-manifold surfaces. A surface is *2-manifold* if and only if for each point r on the surface there exists an open ball with center r and sufficiently small radius so that the intersection of this ball and the surface can be continuously deformed to an open disk [1].

2.1.3. Triangle quality and mesh quality

The triangular mesh should be used to obtain a solution to differential or integral equations. The long narrow triangles seen in particular in Fig. 2.2 are generally not desirable since electric charge and current distributions may significantly vary along their lengths. Such a feature is in contradiction with the general idea of surface discretization, where we typically assume that the electric (or other) parameters are approximately constant for every small triangle.

The best triangle is an equilateral triangle, with all three triangle angles equal to 60 degrees, i.e. $\alpha = \beta = \gamma = 60°$. In reality, the mesh cannot consist of only equilateral triangles. The *quality of the triangular element* is therefore introduced, which is essentially a measure of the deviation from an equilateral triangle. One common quality measure is the ratio between the radius of the inscribed circle (times two), r_{in}, and the radius of the circumscribed circle, r_{out}, – see Fig. 2.4.
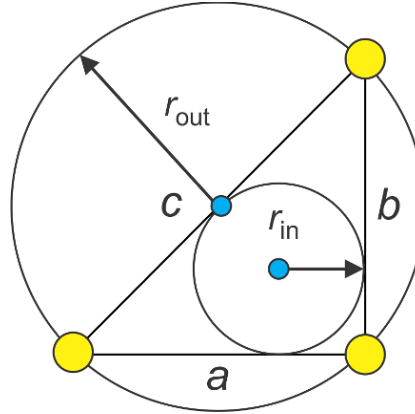
Figure 2. 4. Radii of the inscribed circle (largest circle contained in the triangle) and the circumscribed circle (smallest circle containing the triangle), respectively, for a right-angled isosceles triangle. r_{in} is called the inradius and r_{out} is the circumradius.

The quality factor q is expressed by [2]:

$$q \equiv \frac{2r_{in}}{r_{out}} = \frac{(b+c-a)(c+a-b)(a+b-c)}{abc} \tag{2.4}$$

where a, b, c are the triangle sides. Another useful expression is [2]:

$$q = \frac{2r_{in}}{r_{out}} = 2(\cos\alpha + \cos\beta + \cos\gamma - 1) \tag{2.5}$$

The overall quality factor of the entire triangular mesh may be the lowest q-factor of an individual triangle in the mesh. For example, the mesh shown in Fig. 2.3 has an overall mesh quality 0.83. Alternatively, a histogram may be used, which displays the number of triangles of a certain quality. The quality factor introduced above is not the only quality measure. Many other measures of element quality exist [3], [4], [5].

2.1.4. Triangle size and mesh uniformity

A natural measure of triangle size is the radius of the circumscribed circle, r_{out} in Fig. 2.4 [6]. Such a radius is half of the longest edge for degenerate triangles. For equilateral triangles with side a,

$$r_{out} = a / \sqrt{3} \tag{2.6}$$

2.2. Delaunay triangulation and three dimensional volume and surface meshes

2.2.1. Structured vs. unstructured meshes

Triangular meshes can be categorized as *structured* or *unstructured*. Figs. 2.1 and 2.2 of the previous section illustrate examples of structured meshes. According to Ref.[7], "structured meshes exhibit a uniform topological structure that unstructured meshes lack. A functional definition is that in a structured mesh, the indices of the neighbors of any node can be calculated using simple addition, whereas an unstructured mesh necessitates the storage of a list of each node's neighbors." An example of the structured mesh is given by Eqs. (2.2) of section 2.1. There, all rows of the triangle array t are computed analytically.

The structured surface meshes might be used for simple geometries (a rectangle or a brick in 3D). Inclusion of non-trivial (polygonal or curved) boundaries usually results in the unstructured meshes. Furthermore, the unstructured meshes can provide finer resolution in certain (sometimes not a priori known) domains of interest. In what follows, we will study the unstructured meshes.

2.2.2. Mesh generation and its properties

Surface mesh generation is a creation of an array of nodes, P, for a geometrical structure and subdivision of this structure into small planar triangles described by the array t. There are several desirable properties of mesh generation [7]:

1. The triangles should not intersect the boundaries (in other words, should "respect" the boundaries). Consecutive triangle edges should approximate actual curved boundaries by closest piecewise-linear boundaries.
2. The mesh generation should offer as much control as possible over the sizes of elements in the mesh. This control means the ability to grade from small to large triangles over a relatively short distance.
3. A third (most difficult) goal of mesh generation is that all the triangles should have a relatively high quality as described in section 2.1.

2.2.3. Delaunay triangulation in two dimensions

According to Ref.[8], "despite an abundance of recent work on procedures of generating good triangulation, none of the modern approaches compare in elegance and generality to a procedure developed over fifty years ago by the Russian mathematician Delaunay. Delaunay [9] derived a simple procedure for triangulating an arbitrary set of points on a plane in such a way that *the sum of minimum angle(s) in each triangle would be maximized*. Since finite-element solutions are most accurate with nearly equilateral triangle grids, and since the Delaunay triangulation procedure comes as close as possible to this, it is an excellent method to use with the finite-element method." More precisely, the Delaunay triangulation in two dimensions maximizes the minimum angle of all the angles of the triangles in the triangulation [7]. In other words, it maximizes minimum triangle quality of the mesh according to Eq. (2.5).

2.2.4. Algorithm

Assume that we have the array of nodes, P shown by small circles in Fig. 2.5. We need an array of nonintersecting triangles, t. The following theorem applies: "Three points p_1, p_2, p_3 are vertices of the same triangle of the Delaunay triangulation of P if and only if the circle through p_1, p_2, p_3 (the circumcircle of triangle p_1, p_2, p_3) contains no point, P, *in its interior*. Points on the circle boundary are permitted."

This theorem is illustrated in Fig. 2.5. Triangle p_1, p_2, p_3 in Fig. 2.5a is Delaunay since its circumcircle is empty (does not contain any other nodes of P). However, triangle p_1, p_2, p_3 in Fig. 2.5b is not Delaunay since its circumcircle contain another node.

One popular algorithm for computing Delaunay triangulation is the *edge flip* demonstrated in Fig. 2.5. It begins with an arbitrary triangulation. Then, we use the following theorem: "An edge of the triangulation is Delaunay if and only if there exists an empty circle that passes through its vertices."

Thus, we check every edge in an arbitrarily-created mesh. If this edge is not Delaunay, such as the edge p_1, p_3 in Fig. 2.5b, we simply flip it as shown in Fig. 2.5a. The flipped edge p_2, p_3 is Delaunay. The algorithm requires $O(n^2)$ edge flips for a set P of n points [7].
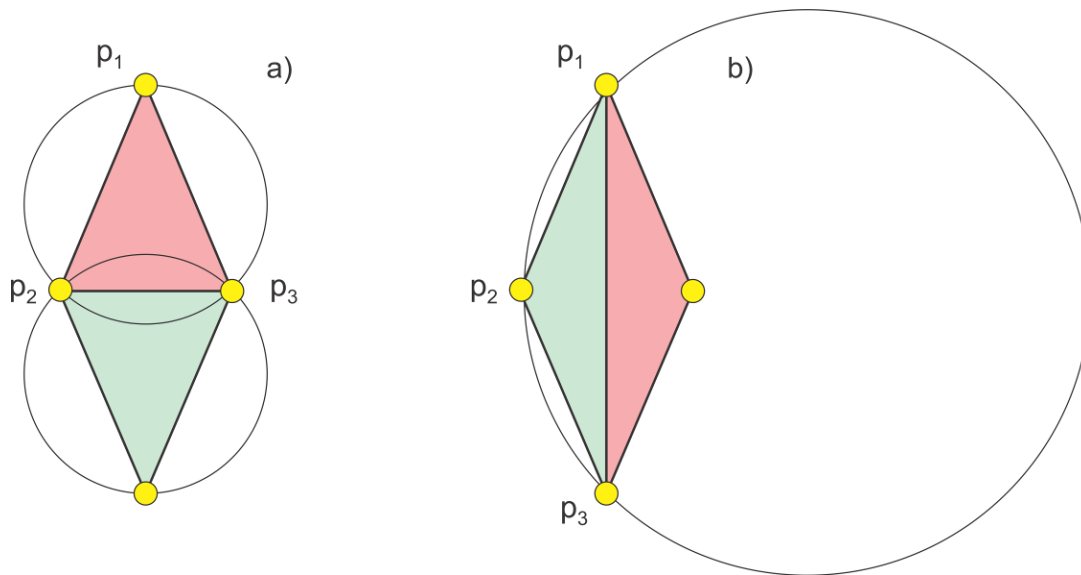
Figure 2. 5. A) – Delaunay triangulation of a set of four nodes; b) – non-Delaunay triangulation of the same node set.

The Delaunay triangulation of a given vertex set is *unique*. The triangle test or the edge test uniquely determines if the triangle or an edge is a part of the triangulation. The known exceptions are a line with more than two points on it, a circle with more than three points on it and no points inside, a sphere with more than four points on its surface and no points inside, and a few other similar cases [7].

Many algorithms compute the Delaunay triangulation by a fast check of whether there is a node inside a triangle in question or not. Today, the Delaunay triangulation of set P of n points in the plane can be computed in O (*nlog (n)*) expected time, using O (*n*) expected storage – see, for example Ref. [10]. The online documentation, algorithms, and examples are given in Ref.[11].

2.2.5. Constrained Delaunay triangulation explained through an example

The Delaunay triangulation should respect the boundary of the object. Therefore, it may start with the explicit definition of the boundary nodes and the boundary edges. The explicit inclusion of boundary edges is not necessary. However, for non-convex boundaries and multiple boundaries the explicit inclusion of the boundary edges is a must. The Delaunay triangulation constructed in this manner is the *constrained Delaunay triangulation*. Figure 2.6 shows an example [12]. In Fig. 2.6a, we have a polygon with eight boundary nodes and no interior nodes. An application of Delaunay triangulation gives the mesh

in Fig. 2.6b, which does not respect the boundaries. When, however, the boundary edges are forced to be a part of the Delaunay triangulation, the mesh of Fig. 2.6c is obtained. The boundary is respected, but there are extra triangles. They are excluded from the mesh by checking the in/out (Boolean) status with respect to a closed boundary of an object – see Fig. 2.6d. Hence, the mesh of a non-convex polygon has been created.
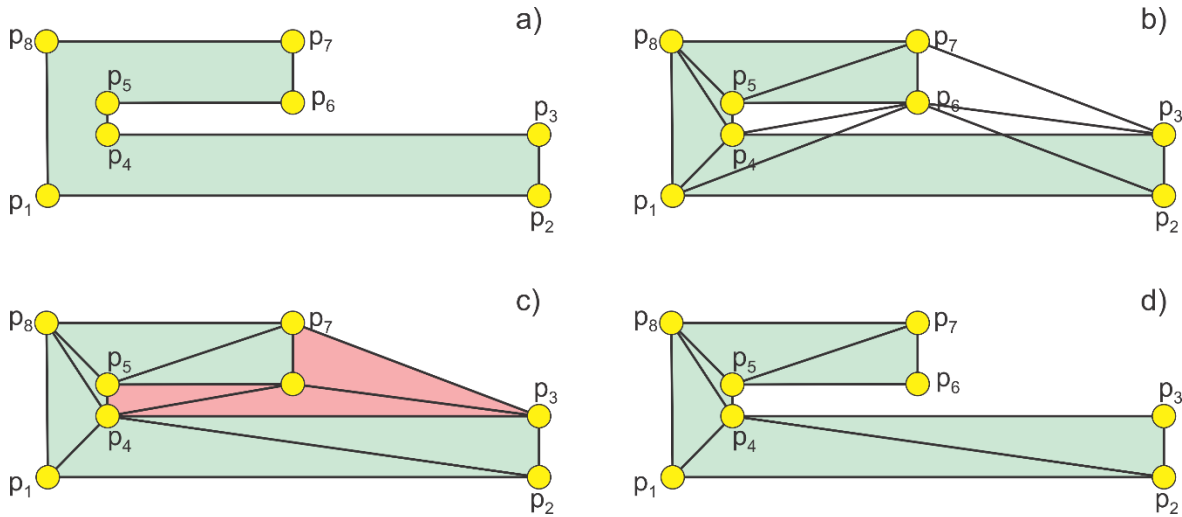
Figure 2. 6. Top – unconstrained Delaunay triangulation of a non-convex polygon; bottom – constrained Delaunay triangulation with boundary edges included into the mesh and removal of unnecessary triangles.

Note that reference [13] discusses edge-flip based algorithms for updating and constructing constrained Delaunay triangulations and constrained regular triangulations. A large collection of mesh generators can be found in online Ref. [14].

2.2.6. Delaunay triangulation in three dimensions (tessellation or tetrahedralization)

Given a set of points P in 3D one can build a tetrahedralization of the *convex hull* (or a convex envelope) that is, a partition of this convex volume into 17quine17dral, in such a way that the circumscribing sphere of each tetrahedron does not contain any other point of P than the vertices of this tetrahedron, Such a tetrahedralization is called a *3D Delaunay triangulation* or *tessellation* or *tetrahedralization* [17]-[20]. Under non degeneracy assumptions (no three points on a line, etc.) it is unique. Many different techniques have been proposed for the computation of Delaunay triangulation in 3D [15]-[17]. One flipping-based algorithm is as follows [17], [18]. At the beginning the triangulation is initialized as a single tetrahedron, with vertices "at infinity", that contains all points of P. At each step a new point from P is inserted and the corresponding tetrahedron, in which this point lies, is split. Then, the Delaunay property is re-established by "flipping" 17quine17dral. This method is thus similar to the 2D triangulation method described above in Fig. 2.5.

2.2.7. Three-dimensional surface mesh generation

This problem is perhaps one of the most complicated tasks of mesh generation. An example is biomedical imaging, which is a very large area of research. A workflow for computational biomedical phantoms is a set of 3D mathematical surface meshing algorithms for anatomical structures, which are extracted from medical imaging data. This data includes Computed Tomography (CT), Magnetic Resonance Imaging (MRI), etc. A typical sequence for three-dimensional triangular surface mesh generation in particular includes:

1. Algorithm(s) for image registration and segmentation – creating a dense point cloud in the form of a shell corresponding to the boundary of a 3D shape from a stack of images [19]-[22] – see Fig. 2.7.
2. Algorithm(s) for surface reconstruction/extraction – creating a triangular surface mesh corresponding to this point cloud [20], [23]-[27] – see Figure 2.7b.
3. Algorithms for healing the resulting mesh [20], [23]-[27], mesh smoothing, and coarsening [28], [35] – see Fig. 2.7b.

a)   b)
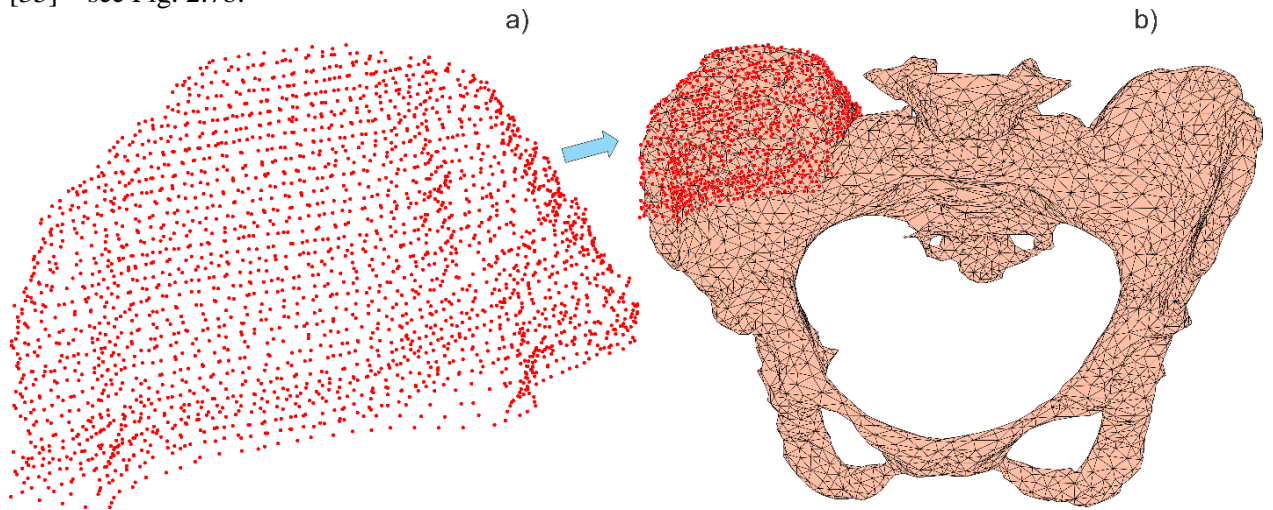
Figure 2. 7. Illustration of three-dimensional surface mesh generation for a pelvic bone from the stack of images from the Visible Human (VHP) Project using MATLAB tools. Only a part of the original point cloud, which is the starting point of mesh generation, is shown in Fig. 2.7a.

2.2.8. Algorithms for three-dimensional surface mesh generation

A naïve but straightforward and simple way is to apply a 3D Delaunay triangulation to a point cloud, create a tetrahedral mesh, and extract the surface (boundary) faces. Unfortunately, most of the tessellation methods create a final convex tetrahedral mesh, which will mask all non-convex details in Fig. 2.7b. Therefore, this method works only for convex objects without holes. For arbitrary bodies, the problem greatly complicates. A very important concern is noise present in real scanned 3D data. Given that the entire problem is very complex, only a brief outline of two methods are discussed: a sculpting-based volumetric method [17] and a region-growing surface method – *the ball-pivoting method* [25].

Volumetric method

In sculpting-based methods, a volume tetrahedralization is computed from the data points, typically the 3D Delaunay triangulation. This may be done by surrounding the original dataset by a shield of extra points – see Fig. 2.8a where a projection of a 3D problem is shown. Tetrahedra are then removed from the convex hull to extract the original shape. It is easy to remove the 18quine18dral, which contain boundary nodes. It is difficult to remove other 18quine18dral under question. The concept is shown in Fig. 2.8a. A certain distance function from the domain surfaces [17] should be available and applicable.

original shell

tetrahedra under question

tetrahedra with boundary nodes removed automatically
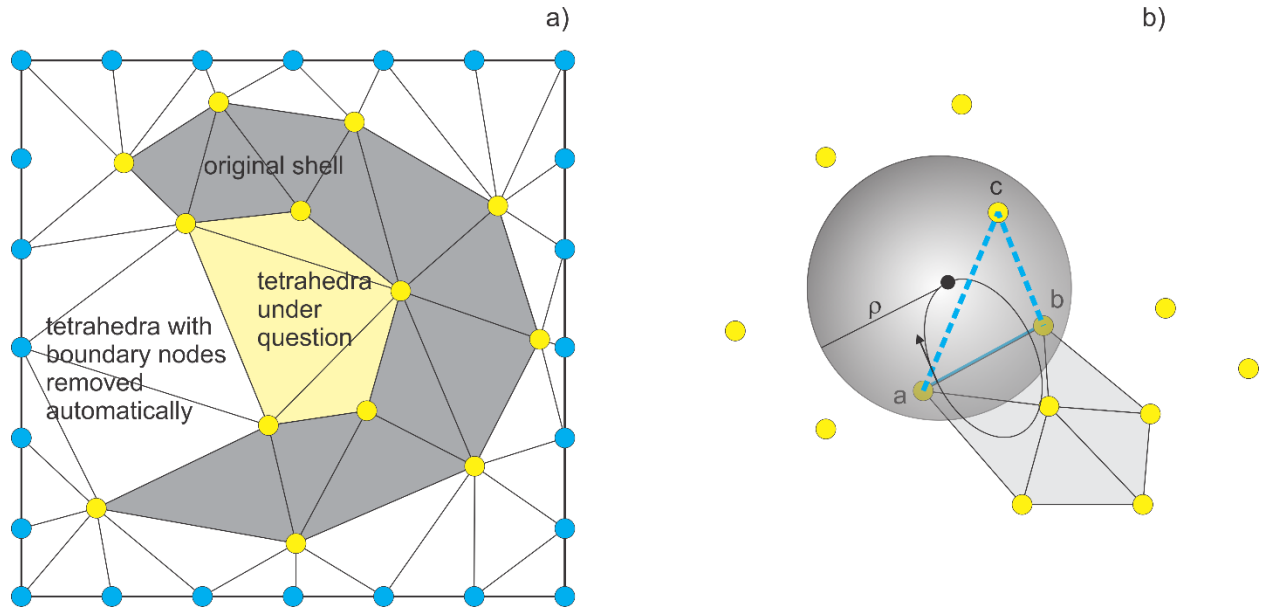
Figure 2. 8. Two methods of three-dimensional mesh generation. In Fig. 2.8a, only a projection of a tetrahedral mesh is shown.

Region-growing surface method (ball pivoting)

The principle of the Ball-Pivoting Algorithm (BPA) is shown in Fig. 2.8b, cited in Ref.[25]. Three points form a triangle if a ball of a user-specified radius ρ touches them without containing any other point. Starting with a seed triangle, the ball pivots around an edge ab (i.e. it revolves around the edge while keeping in contact with the edge's endpoints) until it touches another point c, forming another triangle. The rotation direction is shown in Fig. 2.8b. The process continues until all reachable edges have been tried, and then starts from another seed triangle, until all points have been considered. Parts of the surface mesh so created are then stitched together.

Oversampling

For both methods, oversampling (creation of a very large dense nodal set P) may be a big plus. However, the oversampling is limited by the image resolution and other factors.

2.3. Mesh operations and transformations

2.3.1. Topology-preserving mesh transformations

Once created, a triangular mesh (more specifically, the array of vertices P) as a whole could be subject to a shift, rotation, or other operations in 3D. A third coordinate may need to be added to a 2D surface mesh; the starting point is to use $P(:, 3) = 0$. Most common mesh operations include mesh translation, rotation, scaling, and deformation. Those operations correspond to the translation, rotation, or deformation of the original object. It is important to emphasize that the array of faces or triangles, t, *always* remains the same. *Mesh translation* (shift) is the movement as a whole, along some vector (x, y, z). It is given by:

$$P(:, 1) \rightarrow P(:, 1) + x, \quad P(:, 2) \rightarrow P(:, 2) + y, \quad P(:, 3) \rightarrow P(:, 3) + z \qquad (2.7)$$

Another simple operation is *mesh scaling*. Mesh scaling by the factor (S_x, S_y, S_z) in three dimensions is done according to:

$$P(:, 1) \rightarrow S_x P(:, 1), \quad P(:, 2) \rightarrow S_y P(:, 2), \quad P(:, 3) \rightarrow S_z P(:, 3) \tag{2.8}$$

Mesh rotation about the *x*-axis by angle θ following the right-hand rule is given by:

$$
\begin{aligned}
P(:, 1) &\rightarrow P(:, 1) \\
P(:, 2) &\rightarrow +\cos\theta\, P(:, 2) + \sin\theta\, P(:, 3) \\
P(:, 3) &\rightarrow -\sin\theta\, P(:, 2) + \cos\theta\, P(:, 3)
\end{aligned}
\tag{2.9}
$$

Mesh rotation about the *z*-axis by angle φ is given by a similar expression:

$$
\begin{aligned}
P(:, 3) &\rightarrow P(:, 3) \\
P(:, 1) &\rightarrow +\cos\theta\, P(:, 1) + \sin\theta\, P(:, 2) \\
P(:, 2) &\rightarrow -\sin\theta\, P(:, 1) + \cos\theta\, P(:, 2)
\end{aligned}
\tag{2.10}
$$

A very efficient method for rotating an entire mesh in space, given an arbitrary axis with the unit vector **k** and an angle of rotation θ is the *Rodrigues' rotation formula*. It has the form (noting that rotation about the axis follows the right-hand rule):

$$P_{\text{new}} = P\cos\theta + (K \times P)_{\text{rowwise}}\sin\theta + K \circ K'(1 - \cos\theta) \tag{2.11}$$

Here, $\times$ denotes the cross-product and array K has the same dimensions as P; it consists of the required number of copies of the vector **k**. Array K' also has the same dimensions as P; it contains three identical columns of dot products $(K \cdot P)_{\text{rowwise}}$. The symbol $\circ$ means *Hadamard product* (element by element multiplication).

2.3.2. Necessity of mesh smoothing

Often, the (constrained) 2D Delaunay triangulation algorithm *for planar meshes* is readily available, in particular using the basic MATLAB platform. Then, a mesh for a 2D convex or any non-convex *complicated shape* represented by multiple polygonal boundaries might be created as follows:

1. We select a *larger domain* of a simple shape that contains all boundaries (a rectangle, circle, etc.). We fill this domain with nodes distributed either randomly or, alternatively, as nodes of equilateral triangles of a certain size *excluding* the nodes outside the domain. The resulting node array is P.

2. We analytically specify the *required* boundary nodes, P_b, and the corresponding boundary edges, C. The boundary nodes are put *up front* in the resulting node array, $P \rightarrow [P_b; P]$, in order to keep the connectivity in C. This is a very critical step.

3. We apply the constrained Delaunay triangulation with constrains on C and select subdomains if necessary. Such subdomains may be, for example, electrodes attached to a conducting object.

2.3.3. Topology-preserving Laplacian smoothing

According to Ref.[32], "there are mainly three types of mesh improvement methods: (1) refinement or coarsening, (2) edge swapping, and (3) mesh smoothing. The refinement and the coarsening mainly try to optimize the mesh density, while edge swapping and mesh smoothing mainly aim to optimize the shape regularity. There are mainly two types of smoothing methods, namely Laplacian smoothing and optimization-based smoothing."

In its basic form, Laplacian smoothing implies moving each vertex to the arithmetic average of the neighboring vertices while keeping the boundary nodes and boundary connectivity (boundary edges) *unchanged* as shown in Fig. 2.9. In other words, a free vertex of the mesh is simply relocated to the centroid of the vertices connected to that vertex. In Fig. 2.9, the node p_5 is moved to the average of the neighboring points p_1, p_2, p_3, p_4. In this way, the local triangle quality greatly improves. Indeed, in order to perform the smoothing we should know the neighboring nodes (neighboring triangles) for every mesh node.
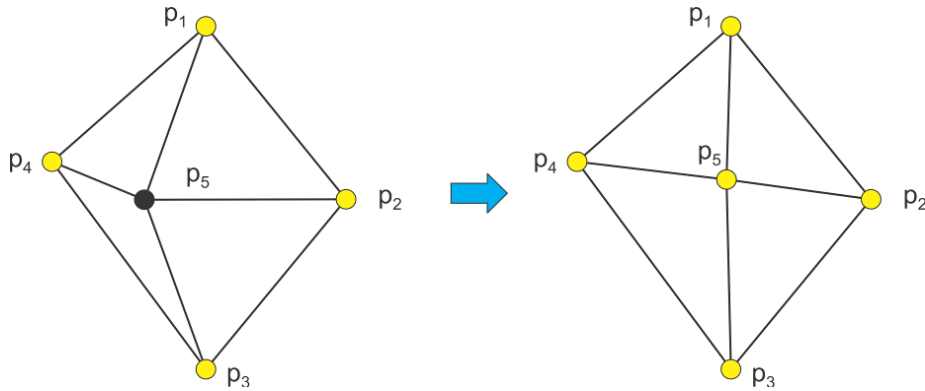
Figure 2. 9. Concept of Laplacian smoothing.

Reference [34] provides a comprehensive overview of different Laplacian smoothing methods. Some of them are briefly listed below.

Standard Laplacian smoothing shown in Fig. 2.9:

$$p^* = \frac{1}{k} \sum_{p_j \in \Omega_i, p_j \neq p_i} p_j \qquad (2.12)$$

where Ω_i is the "star" of the vertex p_i having k points and p^* is the new location of p_i.

Lumped Laplacian smoothing,

$$p^* = \frac{1}{3}p_i + \frac{2}{3}\frac{1}{k}\sum_{p_j \in \Omega_i, p_j \neq p_i} p_j \qquad\qquad (2.13)$$

Centroid Voronoi Tessellation (CVT) smoothing utilizing attached triangle centers t_j and areas A_j,

$$p^* = \sum t_j A_j / \sum A_j \qquad\qquad (2.14)$$

Weighted Centroid of Circumcenters (WCC) smoothing utilizing attached triangle circumcenters c_j and areas A_j,

$$p^* = \sum c_j A_j / \sum A_j \qquad\qquad (2.15)$$

Equally-weighted versions of Eqs (2.14), (2.15) may be used.

2.3.4. Laplacian smoothing with re-triangulation. Iterative algorithm

After the creation of the initial mesh, the Laplacian smoothing of any type is applied to all free nodes except the boundary nodes. After that, the constrained Delaunay triangulation with the constraints is defined. Then, the Laplacian smoothing is applied again and the process repeats itself iteratively. One measure of convergence is the control of the resulting mesh quality. The process may be stopped when the triangle quality ceases to increase or it oscillates about a certain value. Note that the Laplacian smoothing does not necessarily converge except for the algorithm ***Lumped Laplacian smoothing*** [30].

There are two techniques to calculate new positions, p^*. The first method is to modify *all* positions by one step. This method is called the *simultaneous version*. The second variant is to update the new positions of p^* immediately. This variant is called the *sequential version*. In this latter case, a position p^* may not solely depend on the "set" of old positions but can also depend on previously calculated new positions. The Laplacian algorithm ***Standard Laplacian smoothing*** has the property that the limits of the two techniques are the same if they exist [30].

2.3.5. Weaknesses of Laplacian smoothing

The Laplacian smoothing algorithm described in previous subsections has the following weaknesses:
1. It is a *local* algorithm with fixed boundary nodes; therefore it is not able to provide high-quality uniform meshes from an arbitrary set of given nodes. If the nodes were initially concentrated in one local area, they will stay there forever.
2. For the same reason, it is not well suited for creating high-quality, non-uniform meshes with different triangle sizes from a given set of data points;
3. The proper handling of boundary nodes is an important subject. The method of fixed nodes used in this subsection is definitely not the best; it only works when the length of a boundary edge is approximately the best anticipated edge length for a given mesh. Also, intersecting boundaries require special care.

Laplacian smoothing in three dimensions

An undesirable effect is also the obvious "shrinkage" of 3D triangular surface meshes after Laplacian smoothing; the entire 3D mesh actually becomes smaller that it is in reality after several iterations [30]. A simple algorithm to avoid the shrinkage has been developed [30]. The idea is to push the modified points toward the previous points and the original points of the mesh.

2.3.6. Boolean operations with meshes

Combining multiple meshes into one mesh

Consider two *non-intersecting* meshes P_1, t_1 and P_2, t_2. The meshes are combined into one mesh following the rule:

$$P = [P_1; P_2], \quad t = [t_1; t_2 + n_1] \tag{2.16}$$

This rule is easily extended to three and more meshes. Here, n_1 is the number of nodes in the first mesh.

Constructive solid geometry

Boolean operations with simple shapes in 2D and 3D are a part of *Constructive Solid Geometry* (CSG) or *Constructive Volume Geometry* (CVG) algorithms [37]-[39], also implemented in MATLAB [38] for simple shapes. This very interesting subject is beyond the scope of the present section.

Boolean operations with realistic 3D surface meshes

While the CSG works quite well for simple shapes, union, subtraction, and intersection of two realistic 3D surface meshes is a very difficult task [40]-[44]. All intersecting triangles have to be identified, properly split, and re-triangulated in some way or other.

Boolean operations with 2D surface meshes

The problem simplifies in two dimensions where we only need to split intersecting edges. If a function that checks the in/out status of a node in an arbitrary polygon is readily available, the Boolean operations may be implemented with the explicit boundary descriptions (through given boundary nodes and edges). This is the case with the basic MATLAB package where such a function is inOutStatus of the class DelaunayTri. An outer boundary (outer polygon enclosing all objects) must be given.

2.3.7 Collision algorithms for 3D surface meshes

Algorithms that find ray-triangle, segment-triangle, and triangle-triangle intersections [40]-[48] are in particular a basic component of all collision detection data in computer animation. As far computational purposes are concerned, these algorithms allow us to:
1. Find outer surface normal vectors for a 3D shell;
2. Perform the inside/outside check for an arbitrary node;
3. Perform Boolean operations (union, subtraction, intersection) on realistic 3D surface meshes [40]-[44] including meshes of various tissues obtained from the medical image data.

Ray-triangle intersection with Möller and Trumbore algorithm [47]

This algorithm is perhaps most common. We define a ray, $\mathbf{R}(t)$ as $\mathbf{R}(t) = \mathbf{O} + t \cdot \mathbf{d}$ where $\mathbf{O}$ is the origin of the ray and $\mathbf{d}$ is the normalized direction vector. We define a triangle by three vertices: $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. We define the point, $\mathbf{T}(u, \upsilon)$ on the triangle as:

$$\mathbf{T}(u, \upsilon) = (1 - u - \upsilon)\mathbf{p}_1 + u\mathbf{p}_2 + \upsilon\mathbf{p}_3 \tag{2.17}$$

where (u, υ) are *barycentric coordinates*, which, by definition, meet the following conditions:

$$u \geq 0, \ \upsilon \geq 0, \ u + \upsilon \leq 1. \tag{2.18}$$

To find the intersection point between the ray and the triangle, Eqs. (2.17), (2.18) are to be solved simultaneously, which yields

$$\mathbf{O} + t \cdot \mathbf{d} = (1 - u - \upsilon)\mathbf{p}_1 + u\mathbf{p}_2 + \upsilon\mathbf{p}_3 \tag{2.19}$$

Rearranging the terms leads to the matrix equation,

$$\begin{bmatrix} -\mathbf{d} & \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix} \cdot \begin{bmatrix} t \\ u \\ \upsilon \end{bmatrix} = \mathbf{O} - \mathbf{p}_2 \tag{2.20}$$

where $\mathbf{e}_1 = \mathbf{p}_2 - \mathbf{p}_1$ and $\mathbf{e}_2 = \mathbf{p}_3 - \mathbf{p}_1$. By solving Eq. (2.20), we can find the barycentric coordinates (u, υ) and the distance, t, from the ray origin to the intersection point. The solution to Eqs. (2.20) is obtained using the Cramer's rule:

$$\begin{bmatrix} t \\ u \\ \upsilon \end{bmatrix} = \frac{1}{(\mathbf{d} \times \mathbf{e}_2) \cdot \mathbf{e}_1} \begin{bmatrix} (\mathbf{tvec} \times \mathbf{e}_1) \cdot \mathbf{e}_2 \\ (\mathbf{d} \times \mathbf{e}_2) \cdot \mathbf{tvec} \\ (\mathbf{tvec} \times \mathbf{e}_2) \cdot \mathbf{d} \end{bmatrix} = \frac{1}{\det} \begin{bmatrix} \mathbf{qvec} \cdot \mathbf{e}_2 \\ \mathbf{pvec} \cdot \mathbf{tvec} \\ \mathbf{qvec} \cdot \mathbf{d} \end{bmatrix} \tag{2.21}$$

where $\mathbf{tvec} = \mathbf{O} - \mathbf{p}_1$, $\mathbf{pvec} = (\mathbf{d} \times \mathbf{e}_2)$, $\mathbf{qvec} = (\mathbf{tvec} \times \mathbf{e}_1)$ and $\det = \mathbf{pvec} \cdot \mathbf{e}_1$. Fig. 2.10a illustrates the implementation of this algorithm in MATLAB for a ray passing through a human eye surface mesh. The intersection points are marked.
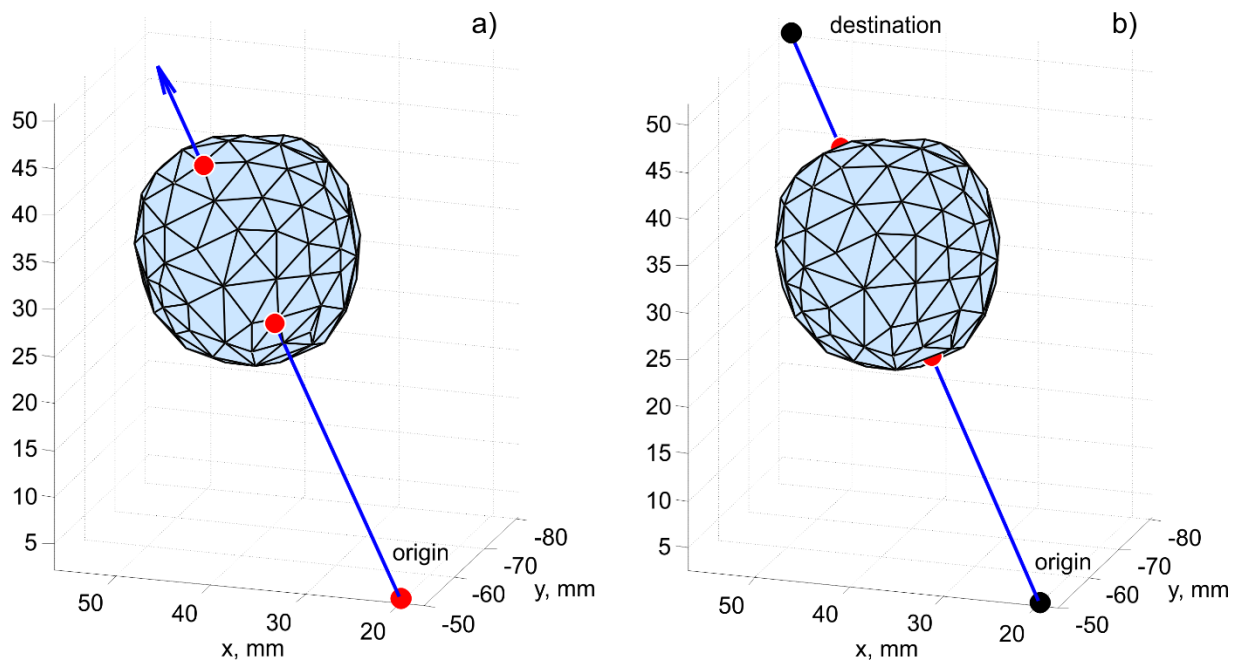
Figure 2. 10. A) – Ray-triangle intersection for a human eye; b) – segment-triangle intersection.

Segment –triangle intersection

The same algorithm applies (an alternative is the algorithm of Ref. [46], which projects the point and triangle onto a 2D coordinate plane where inclusion is tested.). Similar to the ray-triangle intersection, we find the point(s) of intersection but the distance of this point from the origin should be less than or equal to the length of the segment. Fig. 2.10b illustrates the implementation of this algorithm in MATLAB for a ray passing through a human eye surface mesh. The intersection points are marked.

Triangle – triangle intersection and mesh – mesh intersection

A triangle is a set of three segments. Therefore, a triangle-triangle intersection problem can be reduced to the segment-triangle intersection problem considered previously. However, separate fast algorithms may be developed too [44], [47]. Next, the mesh-mesh intersection problem can be reduced to the triangle-triangle intersection problem since every mesh is a combination of triangles. Consider a master mesh X and a slave mesh Y. Both meshes are 2-manifold. For any triangle from the master mesh there exist three different intersection cases as shown in Fig. 2.11. Cases #1 and #3 in Fig. 2.11 would become equivalent if we could treat the master and slave meshes as one set of triangles.
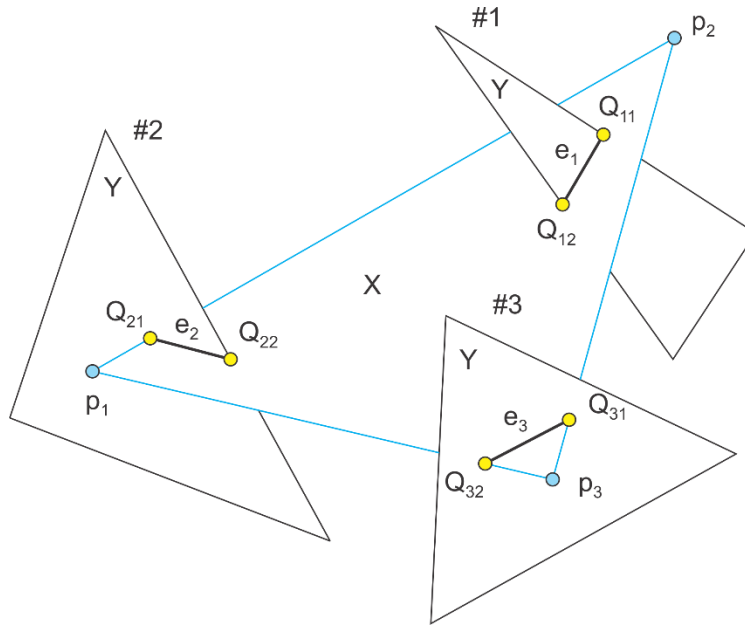
Figure 2. 11. Three types of intersection of a triangle from a master mesh X with various triangles of a slave mesh Y. Cases #1 and #3 are equivalent if we treat the master and slave meshes as one set of triangles.

With reference to Fig. 2.11 we could formulate one possible mesh-mesh intersection algorithm as follows:

1. For each triangle of the master mesh under question, we find intersecting edges $e_i, i = 1,2,3,...$ in Fig. 2.25.
2. Next, we apply a constrained 2D Delaunay triangulation to triangle's plane and subdivide the master triangle into sub-triangles, which respect intersections.
3. The same procedure is applied to each triangle under question of the slave mesh.
4. We construct refined master and slave meshes, which respect all intersections.
5. Boolean operations on meshes are performed by checking in/out status of separate triangles.

Apparently, the above algorithm is quite slow. However it is simple and makes use of the existing constrained 2D Delaunay triangulation in MATLAB.

Other algorithms implemented in MATAB

MATLAB central provides a number of related vectorized scripts – see [48], [49]– for ray-triangle and segment-triangle intersection.

2.3.8. Checking in/out status and finding outer normal vectors for 2-manifold 3D surface meshes

In/out status

Assume that an observation point in Fig. 2.12a lies outside a 2-manifold shell in three dimensions. A ray emanating from this point may or may not intersect this mesh. However, the number of intersections will always be even: 0, 2, 4, etc. Similarly, if the observation point lies inside the shell, the number of intersections will always be odd: 1, 2, 3, etc.
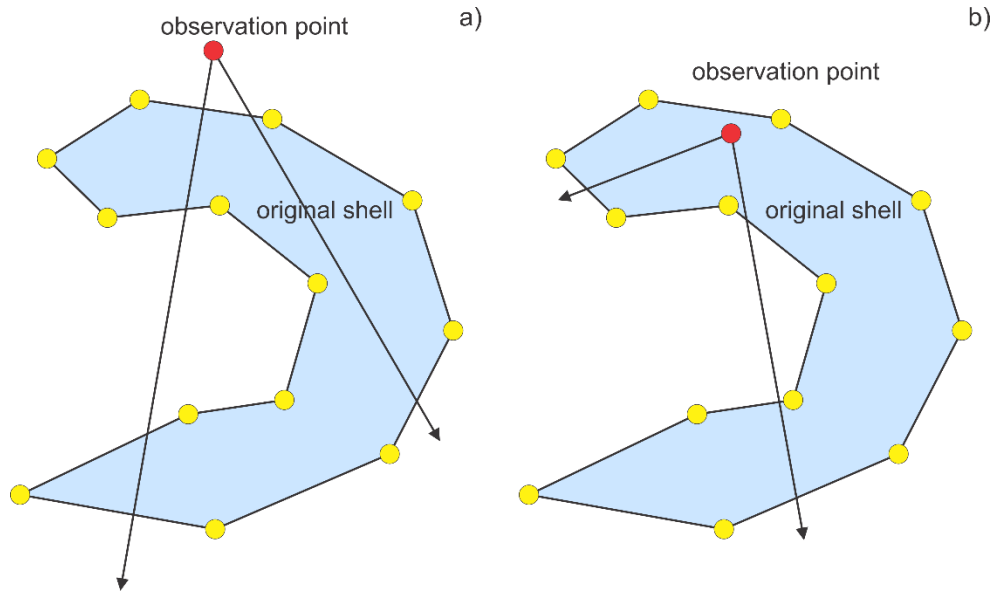
Figure 2. 12. Checking in/out status for a 2-manifold mesh. Only mesh cross-section is shown.

The critical point is that we do not have to check all rays; this can be done only once (excluding some degenerate cases). Therefore, a single ray can be constructed that points toward the center of a selected triangle in the mesh. Then, the ray-triangle intersection algorithm shown in Fig. 2.12 is applied and the number of intersections is counted.

Finding outer normal vectors

The normal vector is defined as the normalized cross-product of two triangle edges; its sign is important. To select only the outer normal vector, we may set an observation point slightly above each triangle center, in the direction of the normal vector. If this point is outside the mesh, the normal direction is correct. If not, the direction is reversed.

2.4. Adaptive mesh refinement and mesh decimation

Theory of adaptive mesh refinement in applications to the Boundary Element Method (BEM) or the Method of Moments (MoM) is developed in many references – see, for example, [50]-[58]. Its detailed analysis is not the subject of this section. This theory depends on the particular method used for computations. Here, we present one simple concept. This concept implies the MoM/BEM Galerkin method with the accurate calculation of potential integrals, makes it possible to evaluate the electric potential and the field at any point of the conducting (metal) boundary or dielectric/magnetic boundary. This also utilizes the simple mesh generation methods described in this Chapter.

2.4.1. Concept of adaptive mesh refinement

As an example, we consider a surface of a metal object in the electrostatic case. Such a surface always possesses a given potential/voltage value. However, the charge distribution is to be evaluated numerically, for every surface triangle. After the numerical solution for the charge distribution is complete, the electric potential may be computed back. Some possible steps for adaptive mesh refinement are then as follows:

(1a) ***Finding local solution error – collocation method***. First, an error of the numerical solution must be found for every triangle in the mesh. In the *collocation method* [50], [52], [53] the electric potential on the object surface is matched to a given potential value, e.g. 1 V only for the collocation nodes (positions) –

typically triangle centers or vertices. A relative potential (or charge/electric field) mismatch for *other positions* could be used as a local *error indicator* [50], [52], [53]. An excellent early paper on the local error behavior is Ref.[50].

(1b) ***Finding local solution error – Galerkin method***. In the Galerkin method, there are no collocation nodes. However, once the numerical solution is available, the potential and the field may still be recalculated at any point of the boundary. Consider the boundary of a metal object with the constant impressed potential of 1 V. Using the existing BEM solution, the electric potential values are recalculated at every triangle center separately. Those values are never 1 V but rather slightly vary about this value. The corresponding absolute difference is an error.

(2) ***Using tangential electric field***
A similar treatment applies to the tangential electric field, which must be zero at the metal surfaces (for example, at triangle centers), but is never really zero. The corresponding absolute deviation is again an error. For dielectric-dielectric interfaces, the error in the boundary condition for the normal *E*-field component may be computed; for magnetic boundaries, the error in the normal H-field is found to be more complicated and physically justified; this is the subject of a separate discussion.

(3) ***Triangle refinement***. Triangles with the largest error (say for example the worst 15% of the total number of triangles) are refined by introducing new mesh nodes exactly at the centers of their edges as shown in Fig. 2.13. Two scenarios are possible: a) three new nodes at three edges of any triangle in question and b) new nodes only on the boundary edges and on edges adjacent to the two triangles in question. All boundary nodes including the old and new ones may be put up front in the new node array *P*.

(4) ***Mesh refinement***. Constrained Laplacian smoothing described in the previous section may be applied to the new node array *P*. The Laplacian smoothing may involve Delaunay triangulation(s). The boundary nodes are not subject to movement. If an inner node crosses the boundary, it is deleted. In the present section, this method will be developed only for a circle and rectangle, but it is applicable to any polygonal domain.

(5) ***Iterative adaptive mesh refinement***. The numerical solution is calculated again and the process repeats itself until a stop criteria defined by the algorithm designer is met.
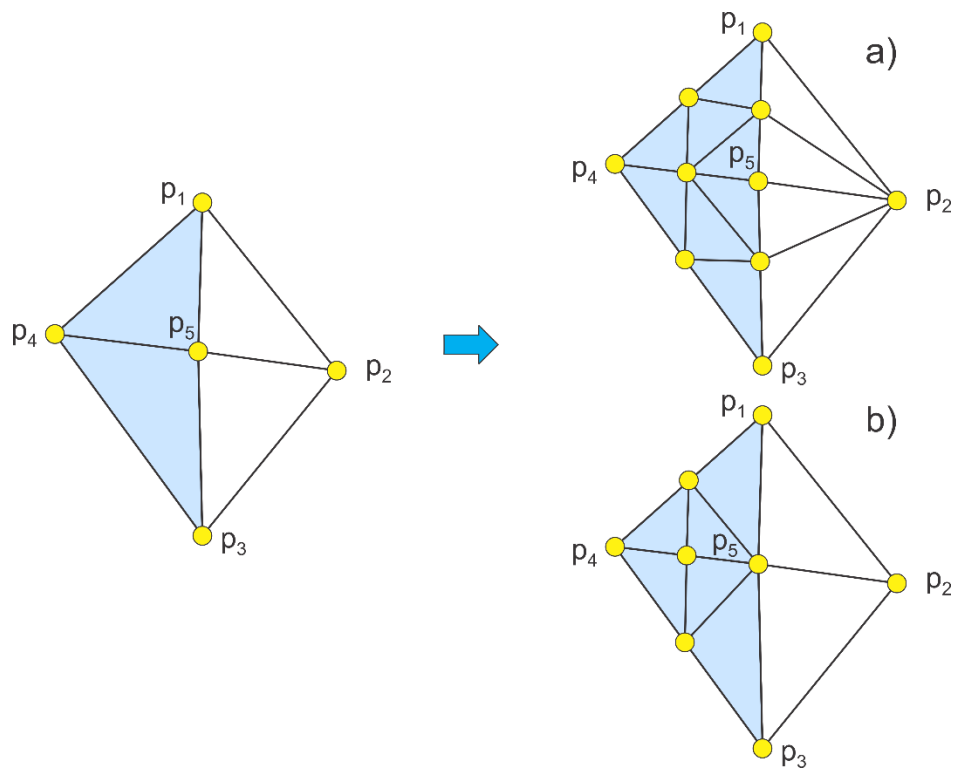
Figure 2. 13. Two scenarios of edge subdivisions for triangles with the largest error.

Note that the adaptive mesh refinement implies final meshes with very different triangle sizes; the max/min triangle size ratio is at least 10:1 and routinely reaches 100:2. Another feature of the iterative procedure is reusing the previous meshes. (see Fig. 2.14)

first iteration, min quality =0.37, 399 triangles

local error

initial mesh 234 triangles

second iteration, min quality =0.66, 687 triangles

sixth iteration, min quality =0.50, 5630 triangles
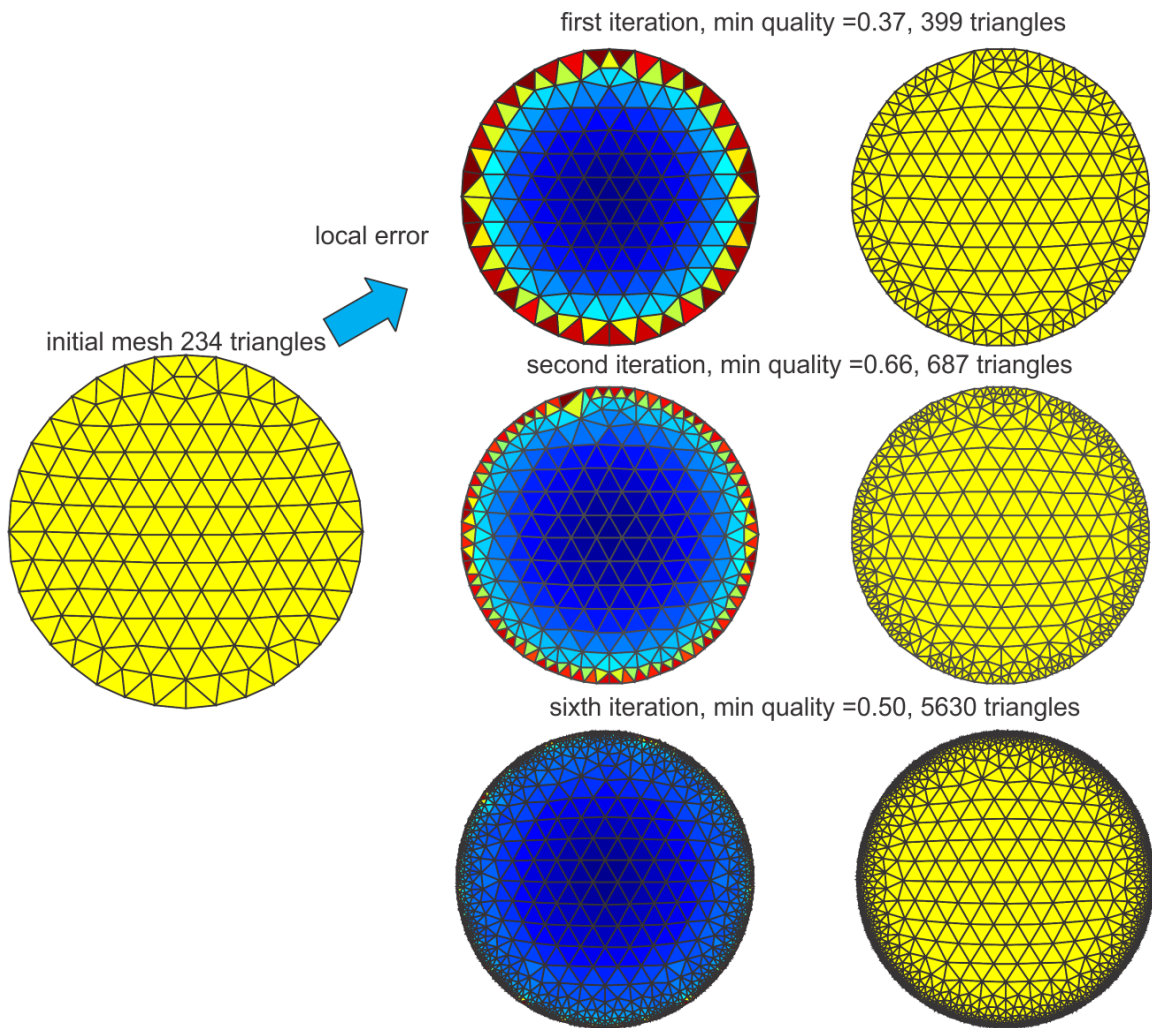
Figure 2. 14. Iterative meshes in the adaptive mesh refinement process. Left column shows the error plot at the previous iteration step; right column – the mesh refined according to this error.

2.4.2. Mesh decimation

Quite often, a dense mesh in certain areas is not required. Depending on factors including the problem geometry, field gradient, and value of the error function, large triangles may provide an adequate solution, enabling reduction of the problem size and an increase in the speed of the calculation. The process of reducing mesh density in certain areas is known as *mesh decimation*. One popular algorithm for mesh decimation is the (incremental) *edge collapse* schematically shown in Fig. 2.15a (see, for example, Ref. [7]). Two nodes p_1, p_2 of an edge (usually the shortest edge in the area of interest) are replaced by one node p_{12} at the edge center and the mesh topology is updated. Yet another algorithm is the *vertex removal* schematically shown in Fig. 2.15b. Node p_1 and all triangles attached to this node are removed and the resulting hole is re-triangulated.
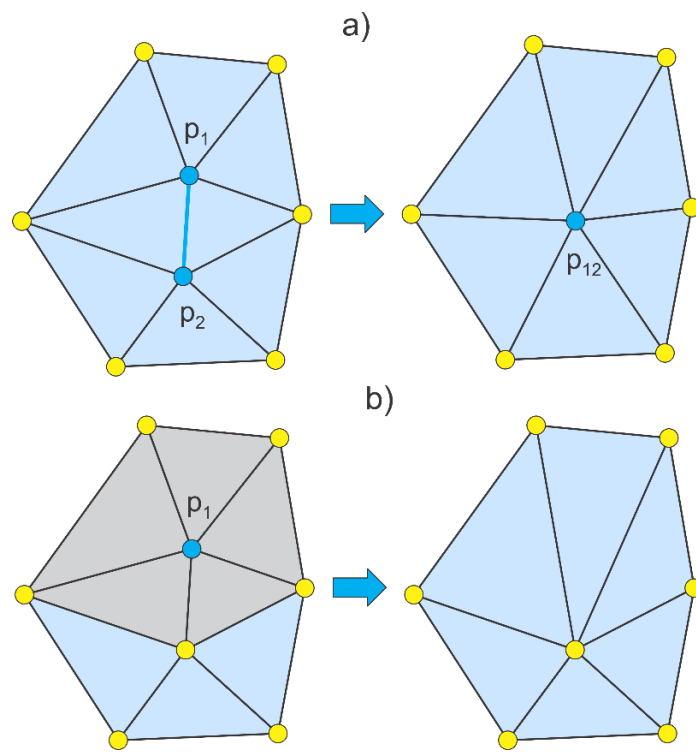
Figure 2. 15. A) – Edge collapse method; b) – vertex removal.

References

[1] Manifolds online: Dr. Ching-Kuang Shene, http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/model/manifold.html

[2] R. A. Johnson, *Modern Geometry: An Elementary Treatise on the Geometry of the Triangle and the Circle*, Boston, MA, Houghton Mifflin, 1929, p. 189-192. Online: http://mathworld.wolfram.com/Inradius.html

[3] D. A. Field, "Qualitative measures for initial meshes," *Intern. J. Numer. Methods Engineering*, vol. 47, pp. 887–906, 2000.

[4] P. M. Knupp, "Remarks on Mesh Quality," 45th AIAA Aerospace Sciences Meeting and Exhibit, 7-10 January, 2007, Reno, NV, 10 p.

[5] Online: http://cubit.sandia.gov/help-version8.1/Chapter_5/Mesh_Quality_Assessment/Metrics_for_Triangular_Elements.html

[6] P.-O. Persson, Mesh Generation for Implicit Geometries, PhD Thesis, MIT, Feb. 2005, 126 p. Online: http://persson.berkeley.edu/thesis/persson-thesis-color.pdf

[7] J. R. Shewchuk, *Delaunay Refinement Mesh Generation*, PhD thesis, Computer Science Department, Carnegie Mellon University, May 18, 1997, Pittsburgh, PA.

[8] Z. Cendes and D. Shenton, "Adaptive mesh refinement in the finite element computation of magnetic fields," *IEEE Trans. On Magnetics*, vol. 21, no. 5, Sep. 1985, pp. 1812.1816.

[9] B. Delaunay, "Sur La Sphe're Vide," *Bull. Acad. Science USSR VII: Serie Sci. Math. Nat.*, 1934, pp. 783-800.

[10] R. A. Dwyer, "A faster divide-and-conquer algorithm for constructing Delaunay triangulations," *Algorithmica*, vol. 2, no.2.4, 1987, pp. 137-152.

[11] The Geometry Center Home Page: http://www.qhull.org/

[12] MATLAB Online Help: *Computational geometry/Delaunay triangulation*.

[13] J. R. Shewchuk, "Updating and constructing constrained Delaunay and constrained regular triangulations by flips," *Proc. Of the 19th Annual Symposium on Computational Geometry (San Diego, California)*, pp. 182-190, Association for Computing Machinery, June 2003. Online: http://www.cs.cmu.edu/~jrs/jrspapers.html

[14] R. Schneiders: http://www-users.informatik.rwth-aachen.de/~roberts/software.html

[15] F. P. Preparata and M. I. Shamos, *Computational Geometry: an Introduction*, Springer-Verlag, New York, NY, 1985.

[16] H. Edelsbrunner, "*Algorithms in Combinatorial Geometry*," vol. 10: EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Heidelberg, Germany, 1987.

[17] C. Bajaj, F. Bernardini, and G. Xu, "Automatic reconstruction of surfaces and scalar fields from 3D scans," in *Computer Graphics Proceedings*, 1995, Annual Conference Series. Proceedings of SIGGRAPH 95, pp. 109–118.

[18] H. Edelsbrunner and N. R. Shah, Incremental topological flipping works for regular triangulations. In Proc. *8th Annual ACM Sympos. Comput. Geom.* (1992), pp. 43–52.

[19] P.A. Yushkevich, P. Joseph, C. Heather, et. Al., "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. 31(3), pp.1116-28, 2006.

[20] P.G. Young, T.B.H. Beresford-West, S.R.L. Coward, et. Al., "An efficient approach to converting 3D image data into highly accurate computational models," *Philos Trans A Math Phys Eng Sci.* A-13; vol. 366 (1878):pp. 3155-73, 2008.

[21] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging,* vol. **13**(1), pp. 46–165, Jan. 2004.

[22] Q. Fang and D. Boas, "Tetrahedral mesh generation from volumetric binary and gray-scale images," *IEEE International Symposium on Biomedical Imaging* 2009, pp. 1142-1145, http://iso2mesh.sourceforge.net/cgi-bin/index.cgi

[23] Lorensen, W.E. and Cline, H.E. "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics*, vol. 21(4): 163-169, 1987.

[24] Hilton, A., Stoddart, A.J., Illingworth, J., and Windeatt, T., "Marching Triangles: Range Image Fusion for Complex Object Modeling," *Image Processing*, vol. 1, pp. 381–384. Sep 1996.

[25] Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G., "The ball pivoting algorithm for surface reconstruction," *IEEE Trans. Visualization and Computer Graphics*, vol. **5**, no 4, Oct-Dec. 1999.

[26] Kazhdan, M., Bolitho, M., and Hoppe, H., "Poisson Surface Reconstruction," *Eurographics Symposium on Geometry Processing*, 2006.

[27] Wang, M. and Feng, J.Q., "2D-manifold boundary surfaces extraction from heterogeneous object on GPU," *J. of Computer Science and Technology*, vol. **27**(4): 862–871, 2012.

[28] Field, D., "Laplacian smoothing and Delaunay triangulations," *Communications in Applied Numerical Methods*, vol. 4, pp. 709–712, 1988.

[29] Freitag, L.A., "On Combining Laplacian And Optimization-Based Mesh Smoothing Techniques," In: *Trends in Unstructured Mesh Generation*, pp. 37-43, 1997.

[30] Vollmer, J., Mencl, R., and Müller, H., "Improved Laplacian smoothing of noisy surface meshes," *Computer Graphics Forum*, vol. 18, no. 3, pp. 131-138, 1999.

[31] Mukherje, N., "A Hybrid, variational 3D smoother for orphaned shell meshes," In *Proc. 11th International Meshing Roundtable*, Ithaca, New York, Sep.15-18 2002.

[32] Chen, L., "Mesh smoothing schemes based on optimal Delaunay triangulations," *In Proc. 13th International Meshing Roundtable*, Williamsburg, VA, Sandia National Laboratories, SAND #2004-3765C, pp.109-120, Sep. 19-22, 2004.

[33] Garimella, R.V., Shashkov, M.J., and Knupp, P.M., "Triangular and Quadrilateral Surface Mesh Quality Optimization Using Local Parametrization," *LA-UR-02-7190*, Nov. 2002.

[34] Erten, H., Ungor, A., and Zhao, C., "Mesh Smoothing Algorithms for Complex Geometries," in: B. W. Clark (ed.), *Proc. 18th Int. Meshing Roundtable*, Springer-Verlag, Berlin-Heidelberg, 2009, pp. 175-193.

[35] Garland, M. and Heckbert, P.S., "Surface simplification using quadric error metrics," SIGGRAPH '97 *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 209-216, NY, 1997.

[36] P.-O. Persson and G. Strang, "A Simple mesh generator in MATLAB," *SIAM Review*, vol. 46, no. 2, pp. 329-345, June 2004.

[37] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice in C*, Addison-Wesley Professional, 1996, 2nd ed.

[38] MATLAB codes by Mr. Bruce R. Land of Cornell University, Online: http://www.nbb.cornell.edu/neurobio/land/projects/hierarchy/

[39] M. Chen and J. V. Tucker, "Constructive volume geometry," *Computer Graphics Forum*, vol. 19, no. 4, pp. 282.293, 2000. Online: http://www-compsci.swan.ac.uk/~csjvt/JVTPublications/CVG-Forum(published).pdf

[40] S. H. Lo and W. X. Wang, "A fast robust algorithm for the intersection of triangulated surfaces," *Engineering with Computers*, vol. 20, pp. 11–21, 2004.

[41] A. H. Elsheikh and M. Elsheikh, "A reliable triangular mesh intersection algorithm and its application in geological modelling," *Engineering with Computers* vol. 30, pp. 143–157, 2014.

[42] L. C. Coelho, M. Gattass, and L. H. De Figueiredo, Intersecting and Trimming Parametric Meshes on Finite-Element Shells," *Int. J. for Numerical Methods in Engineering*, vol. 0(0), 1–100, 1999.

[43] W. M. Lira, L. C. G. Coelho, and L. F. Martha, "Multiple Intersections of Finite-Element Surface Meshes," *11th International Meshing Roundtable*, Ithaca, New York, USA, Sep. 15-18, 2002.

[44] O. Devillers and P. Guigue, "Faster Triangle-Triangle Intersection Tests," Institut National de Recherche en Informatique et en Automatique, Preprint # 4488, June 2002.

[45] T. Möller and B. Trumbore, "Fast, Minimum Storage Ray/Triangle Intersection", *Journal of Graphics Tools*, 2(1):21—28, 1997.

[46] D. Badouel, "An Efficient Ray-Polygon Intersection", *Graphics Gems*, pp. 390-393.

[47] T. Möller, "A Fast Triangle-Triangle Intersection Test," Online: http://web.stanford.edu/class/cs277/resources/papers/Moller1997b.pdf

[48] J. Tuszynski, "Triangle/Ray Intersection," Online: http://www.mathworks.com/matlabcentral/fileexchange/33073-triangle-ray-intersection

[49] J. P. Mena-Chalco, "Ray/Triangle Intersection," Online: http://www.mathworks.com/matlabcentral/fileexchange/25058-ray-triangle-intersection

[50] T. Watanabe, Y. Kuno, and Y. Uchikawa, "Robustness of the boundary integral equation method for potential problems," *IEEE Trans. Magnetics*, vol. 26, no. 2, pp. 610-613, March 1990.

[51] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary Element Techniques, Theory, and Applications in Engineering*, Springer-Verlag, Berlin-Heidelberg, 1994.

[52] M. Bachtold, M. Emmenegger, J. G. Korvink and H. Baltes, "An error indicator and automatic adaptive meshing for electrostatic boundary element simulations," *IEEE Tran. Computer-Aided Design of Integrated Circuits and Systems,* vol. 16, no. 12, pp. 1439-1446, Dec. 1997.

[53] D.-K. Park, C.-H. Ahn, S.-Y. Lee, and J.W. Ra, "Adaptive mesh refinement for boundary element method and its application to stripline analysis," *IEEE Trans. On Magnetics*, vol. 26, no. 2, pp. 783-786, March 1990.

[54] M. Liang, J. Chen, and S. Yang, "Error estimation for boundary element method, "*Engineering Analysis with Boundary Elements*, vol. 23, pp. 257-265, 1999.

[55] V. Jandhyala, S. Savage, and Z. Cendes, "Efficient capacitance computation for structures with non-uniform adaptive surface meshes," *Presented at 36th Annual ACM/IEEE Design Automation Conference*, 1999.

[56] N. Kamiya, Y. Aikawa, and K. Kawaguchi, "Adaptive boundary element for the problem with mixed boundary condition," In: C. A. Brebbia, S. Kim, T. A. Osswald, and H. Power, Eds., *Boundary Elements XVII*, WIT, 1995.

[57] M. Aurada, M. Ebner, S. Ferraz-Leite, M. Mayr, P. Goldenits, M. Karkulik, and D. Praetorius, "HILBERT — A Matlab implementation of adaptive BEM, ASC Report 44/2009, Wien, Austria. Online: http://www.asc.tuwien.ac.at/abem/hilbert/

[58] M. Aurada, S. Ferraz-Leite, P. Goldenits, M. Karkulik, M. Mayr, and D. Praetorius, "Convergence of adaptive BEM for some mixed boundary value problem," *ASC Report No. 12/2010*, Wien, Austria. Online: http://www.asc.tuwien.ac.at/preprint/2010/asc12x2010.pdf

[59] M. Bikson, A. Rahman, A. Datta, "Computational models of transcranial direct current stimulation", Clinical EEG and Neuroscience, vol. 43, pp. 176-183, 2012.

[60] P.C. Miranda, M. Lomarev, M. Hallett, "Modeling the current distribution during transcranial direct current stimulation", Clinical Neurophysiology, vol. 117, pp. 1623-1629, 2006.

[61] P.C. Miranda, A. Mekonnen, R. Salvador, G. Ruffini, "The electric field in the cortex during transcranial current stimulation", NeuroImage, vol. 70, pp. 48-58, 2013.

[62] M. Bikson, A. Rahman, A. Datta, F. Fregni, L. Merabet, "High-resolution modeling assisted design of customized and individualized transcranial direct current stimulation protocols", Neuromodulation: Technolgy at the Neural Interface, vol. 15, pp. 306-315, 2012.

[63] M. Bikson, A. Datta, "Guidelines for precise and accurate computational models of tDCS", Brain Stimulation, vol. 5, pp. 430-434, 2012.

[64] G.M. Noetscher, J. Yanamadala, S.N. Makarov, A. Pascual-Leone, "Comparison of cephalic and extracephalic montages for Transcranial Direct Current Stimulation – A numerical study," IEEE Trans. Biomedical Engineering, vol. 61, no. 9 (Sep. 2014), pp. 2488-2498.

[65] T. Möller and B. Trumbore, Fast, Minimum Storage Ray/Triangle Intersection, J. of Graphics Tools, vol. 2 (1997) 21–28.

[66] M. Garland and P.S Heckbert, "Surface simplification using quadric error metrics," SIGGRAPH '97 *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 209-216, NY, 1997.

[67] P.-A. Ubach, C. Estruch, and J. Garcia-Espinosa, On the interpolation of normal vectors for triangle meshes, Int. J. Numer. Meth. Eng. 2013; 96:247–268.

[68] G. Thürmer and C. Wüthrich, Computing vertex 35quine from polygonal facets. Journal of Graphics, GPU and Game Tools 1998; 3(1):43–46.

[69] H. Erten, A. Ungor, and C. Zhao, "*Mesh Smoothing Algorithms for Complex Geometries*," in: B. W. Clark (ed.), *Proc. 18th Int. Meshing Roundtable*, Springer-Verlag, Berlin-Heidelberg, 2009, pp. 175-193.

[70] J. Yanamadala, V.K. Rathi, S. Maliye, H.A. Win, A.L. Tran, M.K. Kozlov, G.M. Noetscher, A. Nazarian, and S.N. Makarov, "Segmentation of the Visible Human Project® (VHP) Female Cryosection Images within MATLAB® Environment," 23rd International Meshing Roundtable (IMR23), London, England, Oct. 12-15, 2014.

III. Chapter 3: Development of Breathing Sequence

An approximate method to model respiratory motion in a CAD human model subject to electromagnetic (or acoustic, thermal) finite-element analysis is suggested and described. Its concept implies using affine transformations, which are implemented in commercial FEM software packages, in the form of a parametric sweep. This method does not require multiple copies of the CAD model or multiple project files. It enables use of arbitrary sampling times and an automatic reposition of on-body and in-body devices. The method was applied to the platform-independent full-body electromagnetic computational model Visible Human Project® (VHP)-Female v. 3.1. Examples of scattering calculations and MRI Coil application will be given.

3.1. Background

Human respiration is the exchange of air between the lungs and the ambient atmosphere. The section below briefly summarizes some major facts pertinent to our study.

Mechanics. Respiratory mechanics represent a complex multi-object deformation process. It predominantly involves the non-rigid motion of the (i) diaphragm, (ii) thoracic cage including ribs, cartilage, and sternum, (iii) lungs, (iv) heart, (v) liver, (vi) kidneys, and (vii) intestine. For inhalation, the diaphragm contracts and pushes the contents of the abdomen in the inferior direction as shown in Fig. 3.1 [1] and Fig. 3.2 [2]. Simultaneously, the external intercostal muscles expand the rib cage and slightly raise it. For exhalation, the diaphragm and the external intercostal muscles relax.
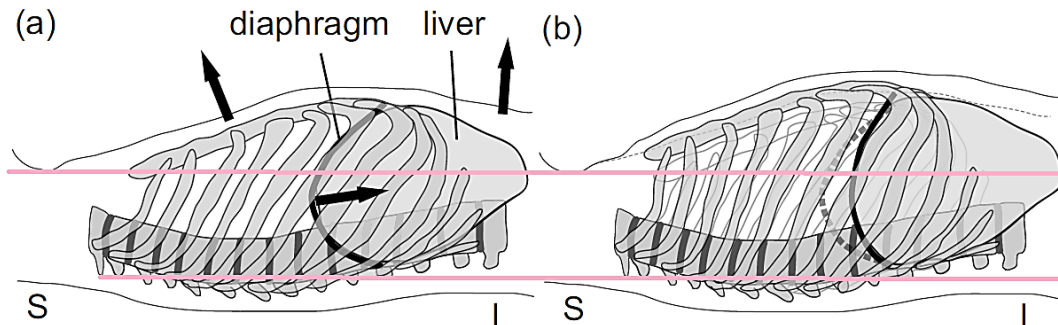
Figure 3. 1. (a) Maximum exhalation position; (b) Maximum inhalation position, after [1].

Diaphragm motion. Respiration is chiefly driven by the diaphragm with primary motion in the superior-inferior direction; total travel is estimated as 10-30 mm during quiet breathing [1]. Other studies report 20±7.0 mm average [2]. A simplified 1D diaphragm motion, $x(t)$, is non-harmonic, and the exhalation portion dominates the inhalation. Given the exhalation at origin of a **healthy** subject in normal condition, one has

$$x(t) = -A\cos^4 \omega t \qquad\qquad (3.1)$$

where A is the corresponding amplitude [3],[4]. Furthermore, the respiratory motion often exhibits hysteresis in space, with an amplitude on the order of 2-4 mm [1].

Adjacent tissues. Closely adjacent structures (i.e., liver, etc.) show comparable motion amplitudes. Furthermore, the following motion amplitudes have been observed for a healthy subject (cf. a review in Ref. [1]):
 - Motion with an average amplitude of 12mm in the lung for tissue parts not attached to rigid structures;
 - 1-25 mm superior-inferior motion of the kidneys, 13 mm superior-inferior motion of the spleen, 2-8 mm motion of the heart (the heart motion is mostly a simple rigid-body translation [5], [6]), and 1-7 mm motion of the trachea;

- 13 mm superior-inferior motion of the spleen.
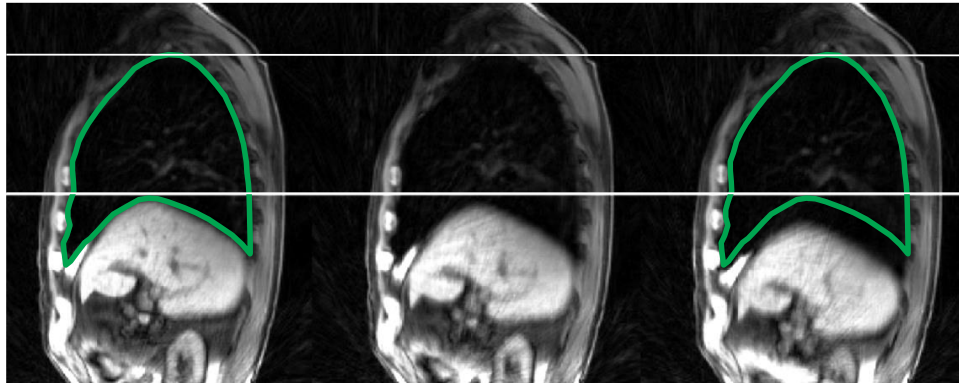
Figure 3. 2. Respiratory motion captured via MRI retrospective gating and averaging over multiple cycles, after [2]. The green contour indicates lung volume at maximum exhalation.

Thoracic cage kinematics. During respiration, the ribs rotate about an axis through their costal necks to affect the anteroposterior and transverse diameters of the thoracic cavity as shown in Fig. 3.3 [5],[7].
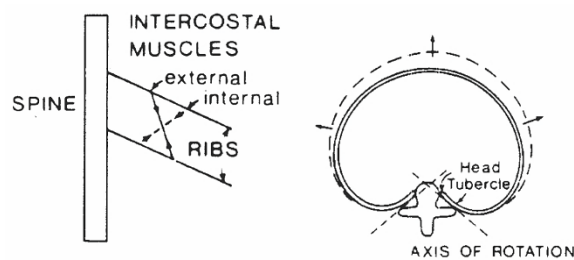
Figure 3. 3. Motion of the ribs during respiration, after [5],[7]. The ribs rotate about an axis through their costal neck.

CAD B-Spline modeling. Modeling of the breathing cycle to date has been mostly performed via deformable NURBS surfaces (B-splines) for the lungs and surrounding tissues. The changes the phantoms undergo are then typically splined over time to create time continuous 4D respiratory models [5],[8],[9], which indeed utilize free-form deformations.

Challenges of FEM CAD Modeling. Commercial FEM codes do not operate with B-spline surfaces but rather with triangulated surfaces and tetrahedral/hexahedral volumes. This is in particular valid for most frequency-domain electromagnetics solvers such as ANSYS EM Suite/Maxell 3D and CST Microwave Studio. Therefore, a free-form breathing sequence has to be ultimately converted to a (large) *discrete* series of separate (full-body) triangulated CAD models, even if the original data were in the form of parametric B-splines. Generally, this conversion requires ssignificant additional meshing times.

The size of one detailed FEM full-body model is quite large (about 200-1000 Mbytes in ANSYS) and a computation with 20-30 such models would be a significant challenge from several points of view. For example, a user will need to create, run, and then post-process a number of large distinct project files, each of which must replicate his own excitation setup (e.g. a coil, an antenna, or a radar) and employ a new human model.

3.2. Approach

Built-in affine transformations. A commercial CEM package typically includes a set of nine affine transformations:

3 translations (in the x, y, z directions);

3 rotations (about the x, y, z axes);

3 directional scaling transformations (along the x, y, z axes);

applicable to any object (including a triangular tissue mesh) or to a group of objects and in the form of a *parametric sweep*. These transformations can be performed in either global or local coordinate systems. The user can initialize a discrete generic global variable, x_n, $n = 0,..., N$, define object geometry parameters as certain unique functions of x_n , and then move/rotate/deform every object of a multi-object structure independently within the framework of the *same* project file.

*Approach. B*uilt-in parameterized affine transformations to construct breathing cycles (quiet, deep, shallow) were applied using only *one* base full-body human model and using only *one* project file. Along with the base static human CAD model, this project file includes a parametric sweep or sweeps modeling deformations of involved tissues. Such an approach is **_not exact_**, but it may have **_sufficient accuracy_** when the parametric sweep is carefully designed. It will allow us to employ any temporal resolution, which is impossible with discrete models. To construct an anatomically relevant breathing cycle, the project aim is to follow the anatomical data collected from Refs. [1]-[9] as close as possible.

Challenges. To design an FEM-compatible and anatomically justified multi-tissue affine parametric sweep, a very extensive preprocessing of the static human CAD model is necessary, which is a significant undertaking.

3.3. Selecting a sweeping variable

The natural sweeping variable x_n is proportional to *the diaphragm motion*. Since the breathing cycle is periodic, only one period T must be considered. According to Eq. (1), physical time, t, is expressed through a sweep variable x_n, $n = 0,..., N$ by

$$t = T \left\{ \frac{1}{2} - \frac{1}{\pi} \cos^{-1} \left(\sqrt[4]{\frac{x_n}{N}} \right) \right\} \qquad \text{when } 0 \le t \le T/2 \tag{3.2}$$

This result can be programmed in MATLAB as

```
E = 11; t_=0:E; T = 1; t = T*(pi/2-acos((t_/E).^0.25))/pi; plot(t_, t, '-*'); grid on;
```

Table 3.1 gives the corresponding numerical time values. Sweeping variable x_n runs from zero to N=11 in twelve uniform steps. Its zero value corresponds to maximum exhalation; its maximum value of 11 corresponds to maximum inhalation.

Table 3. 1. Time values in terms of period, T, corresponding to the sweeping variable x_n, $n = 0,..., N$ *for N=11.*

x_n	t/T
0	0.0000
1	0.1850
2	0.2265
3	0.2571
4	0.2830/

5	0.3066
6	0.3292
7	0.3515
8	0.3747
9	0.4000
10	0.4308
11	0.5000
10	0.5692
9	0.6000
8	0.6253
7	0.6485
6	0.6708
5	0.6934
4	0.7170
3	0.7429
2	0.7735
1	0.8150
0	1.0000

Higher *N* values can be considered for a better accuracy.

3.4.   Static CAD model

As a base human model at maximum exhalation, the VHP-Female v.3.1 CAD model, (http://www.nevaem.com/) shown in Fig. 3.4, was chosen. The model has 26 individual tissues, 270 individual tissue parts, major blood vessels and peripheral nerves, and a superior resolution in the spinal cord/cranium. The sweep for the respiratory motion will be implemented for the BASE model versus ACCURATE model (Section 4.14). Only the results for the BASE model will be reported here.

The subject is a ~60 year old white female with a height *h* of 162 cm measured from top of the scalp to the average center of both heels. The body mass *M*, computed using standard tissue densities [11] and assigning the average body shell, which includes internal tissues, the density of muscle, is ~88 kg. The computed BMI is ~33.5 (moderately obese). The subject has a heart pathology.
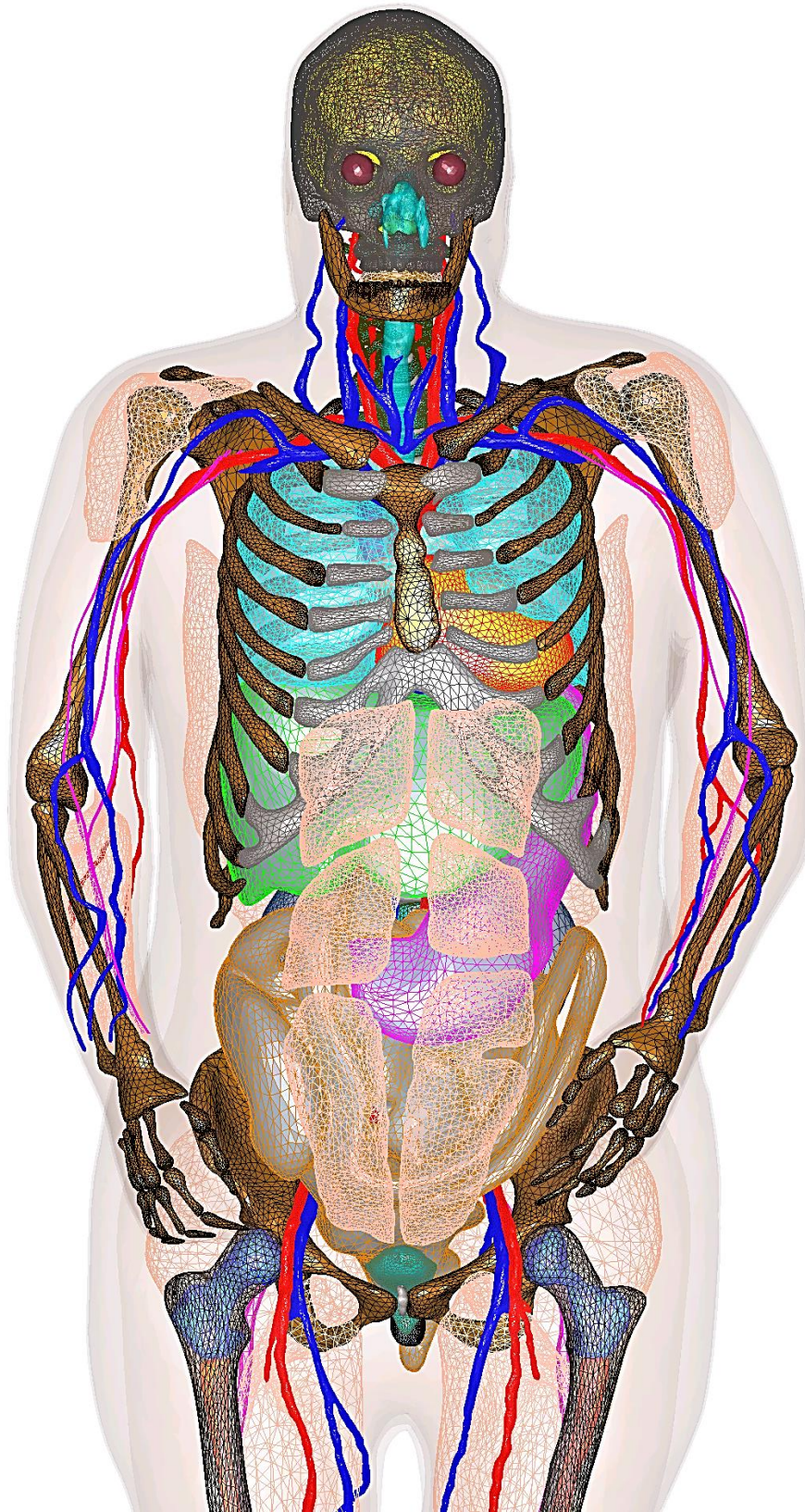
Figure 3. 4. Static VHP-Female v.3.1 CAD model at maximum exhalation (http://www.nevaem.com/).

3.5. Respiratory cycle and CAD tissues affected by respiratory motion

The overall change in lung volume is set at ***0.32 L***, which is close to a normal-to-shallow breathing sequence for this subject. Default temporal resolution includes ***12 discrete uniform steps*** from 0 to 11 in steps of 1 from maximum exhalation to maximum inhalation. The default full cycle includes ***23 discrete steps***. Breathing cycles with finer resolution may be trivially constructed.

The following major set of tissue parts (35 in total) are affected by the respiratory motion:

- Lungs;
- Ribcage with 24 ribs (every rib is moved ***independently***);
- Thoracic cage cartilage;
- Sternum;
- Pectoralis muscles (major/minor);
- Abdominal muscles;
- Erector spinae muscles;
- Heart (muscle);
- Liver;
- Stomach;
- Outer shell – average body;
- Outer shell – fat;
- Outer shell – skin.

These objects are transformed so that there are no intersections between any of them at any time moment, ***with the minimum deformation factors***. These transformations are to be performed in global or local coordinate systems.

3.6. Required accuracy – total body mass

Since the respiratory motion modeled with multiple deformed CAD objects is an approximation, a requirement should be made with regard to the total mass error. It is required that the maximum relative body mass variation shall not exceed 0.1% during the entire respiratory cycle.

3.7. Algorithm

Below, suggested kinematics and dynamics for the individual tissues are reviewed. All quantitative approximations and the final formulas are thoroughly described in Appendix A. Note that the sequence below is the development sequence. In actual implementation of the respiratory sequence in CEM software, all motions of different objects occur in **parallel**.

Lung dynamics. This is the first deformation step described in detail in Appendix A. In a local co-ordinate system associated with the top of the lung, the lung is deformed in all three directions and is moved in one direction in order to guarantee the expected diaphragm movement of 20 mm and simultaneously the volume change of 0.32 L, while maintaining anatomically sound overall deformations.

Thoracic cage kinematics. This is the second deformation step. Since the rotation axes are very loosely defined for the actual anatomical data, we have rotated each rib pair about a fixed axis passing through the heads of two ribs (the end parts closest to the spine). We have also rotated slightly the rib pairs about the vertical axis. Thus, every rib pair is subject to rotation about two axes. All permissible variations of rotation angles have been tested, for every rib pair, in order to satisfy two criteria: (i) avoid intersections with the lung; and (ii) stay as close to the lung as possible.

Sternum/cartilage dynamics. This is the next deformation step. The sternum is subject to a translation motion, without rotation. Fixed *control points* on its surface are introduced. Those control points, along with the rib tips, form lines, along which the corresponding cartilage parts will further be deformed (moved and expanded).

Muscles dynamics. In this case, we apply rotations, movements, and slight deformations. The goal is to minimize overall movement while avoiding intersections with the thoracic cage.

Heart kinematics. The heart is moved in two directions (along top to bottom axis and back to front axis) without rotations and deformations. The cardiac cycle is not considered.

Liver/stomach kinematics. Liver and stomach are moved in two respective directions and are slightly deformed; see Appendix A.

Outer full-body shells. This is the only case where we cannot apply affine transformations. However, we may apply Boolean operations with the tissue CAD objects. A number of deformed chests objects are created internally, and then they are united with the otherwise static full-body shells. This operation requires greater care since we have two very closely spaced (1 mm) body shells.

3.8. Polynomial interpolation with MATLAB Curve Fitting Tool

After a discrete set of affine transformations has been established, this set was converted to polynomials using the MATLAB Curve Fitting Tool applicable to any temporal resolution and reported in Appendix A. The polynomial approximations have been independently tested with a fine grid. As an example, Table 3.2 reports affine polynomial approximations for several muscles. Note that the dynamic variable t in Table 3.1 is not the actual time, but is proportional to the diaphragm motion $x(t)$ in Eq. (3.1). Affine transformations are performed with local coordinate system or global coordinate system. Local coordinate system of each object is its center (mean(x,y,z)).

Table 3. 2. Affine transformations of some muscles (inhalation only) of the VHP-Female model.

Muscles			Polynomials of deformation factors (angles are recorded in degrees)
Pectoralis minor (in local Coordinate System)	Left	Rot z	$-7.149e-5*t^6 + 0.00252*t^5 - 0.03393*t^4 + 0.2181*t^3 - 0.681*t^2 + 1.406*t + 0.005579$
		Move y	$0.0002042*t^6 - 0.007194*t^5 + 0.09695*t^4 - 0.6231*t^3 + 1.946*t^2 - 4.016*t - 0.01594$
		Scale y	$2.042e-7*t^6 - 7.194e-6*t^5 + 9.695e-5*t^4 - 0.0006*t^3 + 0.001946*t^2 - 0.004*t + 0.999984$
	Right	Rot z	$1.083e-05*t^6 + 8.348e-05*t^5 - 0.00957*t^4 + 0.1344*t^3 - 0.7021*t^2 + 1.774*t + 0.01398$
		Move y	$-3.095e-5*t^6 - 0.0002385*t^5 + 0.02734*t^4 - 0.3841*t^3 + 2.006*t^2 - 5.067*t - 0.03994$
		Scale y	$3.09e-8*t^6 - 2.385e-7*t^5 + 2.73e-5*t^4 - 0.00038*t^3 + 0.002*t^2 - 0.005067*t + 0.99996$
Pectoralis major	Left	Move z	$-0.00013*t^6 + 0.004709*t^5 - 0.0656*t^4 + 0.4366*t^3 - 1.388*t^2 + 2.397*t - 0.005142$

(in local Coordinate System)		Move y	$-0.05 + 0.0002525 * t^6 - 0.00914 * t^5 + 0.1272 * t^4 - 0.8445 * t^3 + 2.673 * t^2 - 4.721 * t + 0.01132$
		Scale y	$2.6e - 7 * t^6 - 9.418e - 6 * t^5 + 0.0001312 * t^4 - 0.00087 * t^3 + 0.0028 * t^2 - 0.0048 * t + 1.00001028$
	Right	Move z	$-4.642e - 05 * t^6 + 0.001553 * t^5 - 0.02174 * t^4 + 0.1655 * t^3 - 0.6858 * t^2 + 1.801 * t + 0.02082$
		Move y	$9.284e - 05 * t^6 - 0.003107 * t^5 + 0.04348 * t^4 - 0.331 * t^3 + 1.372 * t^2 - 3.602 * t - 0.04165$
		Scale y	$9.284e - 8 * t^6 - 3.107e - 6 * t^5 + 4.348e - 5 * t^4 - 0.00033 * t^3 + 0.00137 * t^2 - 0.0036 * t + 0.99995835$
Erector Spinae (in local CS)	*Left*	Scale y	$8.987e - 06 * t^6 - 0.0003339 * t^5 + 0.004839 * t^4 - 0.03445 * t^3 + 0.1242 * t^2 - 0.2474 * t + 0.998756$
		Scale x	$4.493e - 06 * t^6 - 0.0001669 * t^5 + 0.00242 * t^4 - 0.01723 * t^3 + 0.06208 * t^2 - 0.1237 * t + 0.9993778$
		Move y	$-1.123e - 05 * t^6 - 0.0004174 * t^5 + 0.006049 * t^4 - 0.04307 * t^3 + 0.1552 * t^2 - 0.3093 * t - 0.3093$
	Right	Scale y	$-0.02 * t + 1$
		Scale x	$-0.02 * t + 1$
		Move y	$-0.05 * t$
Abdominal (in global CS)	Move z		$0.09091 * t * 10^{-3}$
	Move y		$-0.35 * t * 10^{-3}$

3.9.   Results

The corresponding full-body VHP-Female model with the embedded respiratory motion in the form of a parametric sweep described in Appendix A has been independently realized in

- ANSYS Electronics Desktop software package
- CST Studio Suite software package
- MATLAB

The maximum body mass variation during the entire respiratory cycle is 80 g, ***which is less than 0.1% of the total body mass***. The parametric sweep may be adjusted/modified at any time in response to further anatomical evaluations and customer needs.

3.10.   RF test at 300 MHz

The problem geometry is shown in Fig. 3.5. An incident plane wave at 300 MHz has a horizontal polarization. The simulations have been performed in ANSYS HFSS with three adaptive mesh refinement passes and with the final meshes approaching 1M 44quine44dral.

Near field. Fig. 3.5 shows the near-field results at three observation points given a 1 V/m incident wave. The scattered field is plotted. In the illuminated zone, the co-pol near field data may vary by about 3% due to the respiratory motion. In the shadow zone, the corresponding variation is negligibly small. *Cross-polarization components may exhibit considerably larger relative near-field variations.*
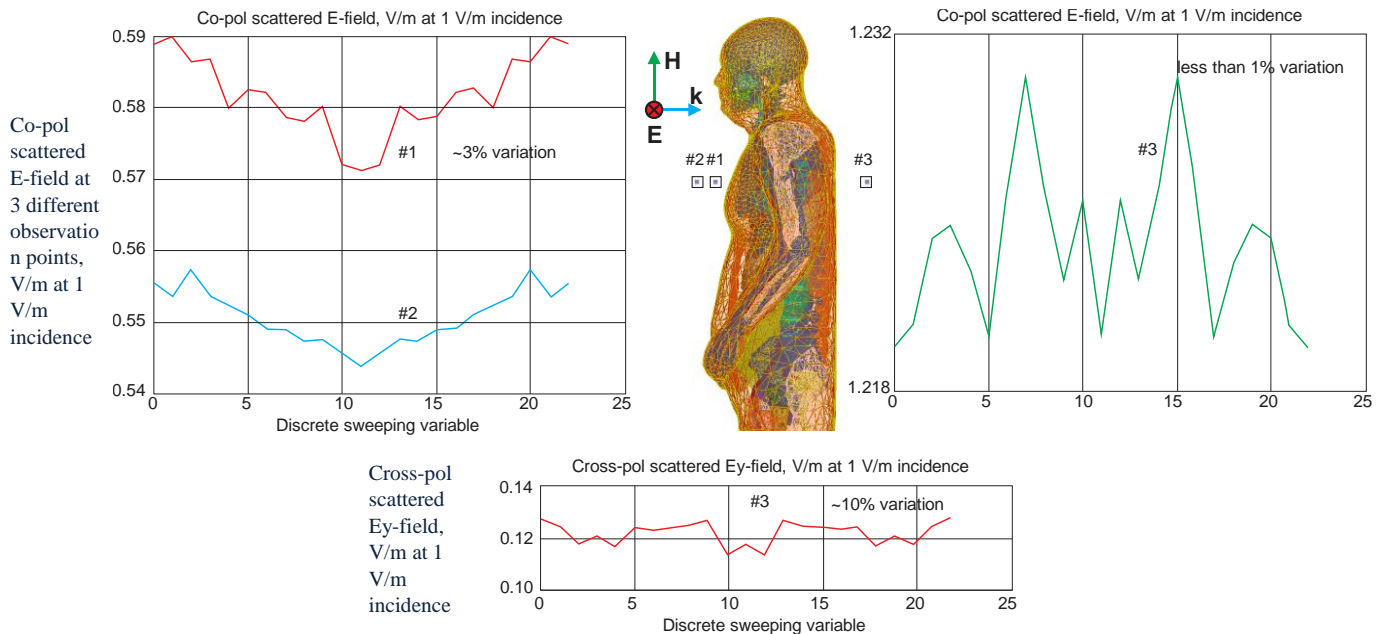
Figure 3. 5. Scattered field in the Fresnel region at 300 MHz.

RCS (Radar Cross Section). RCS is the measure of a target's ability to reflect radar signals in the direction of the radar receiverFig 3.6 shows the monostatic RCS of the heterogeneous breathing VHP-Female Model during the respiratory cycle. The RCS variations are about 1%.
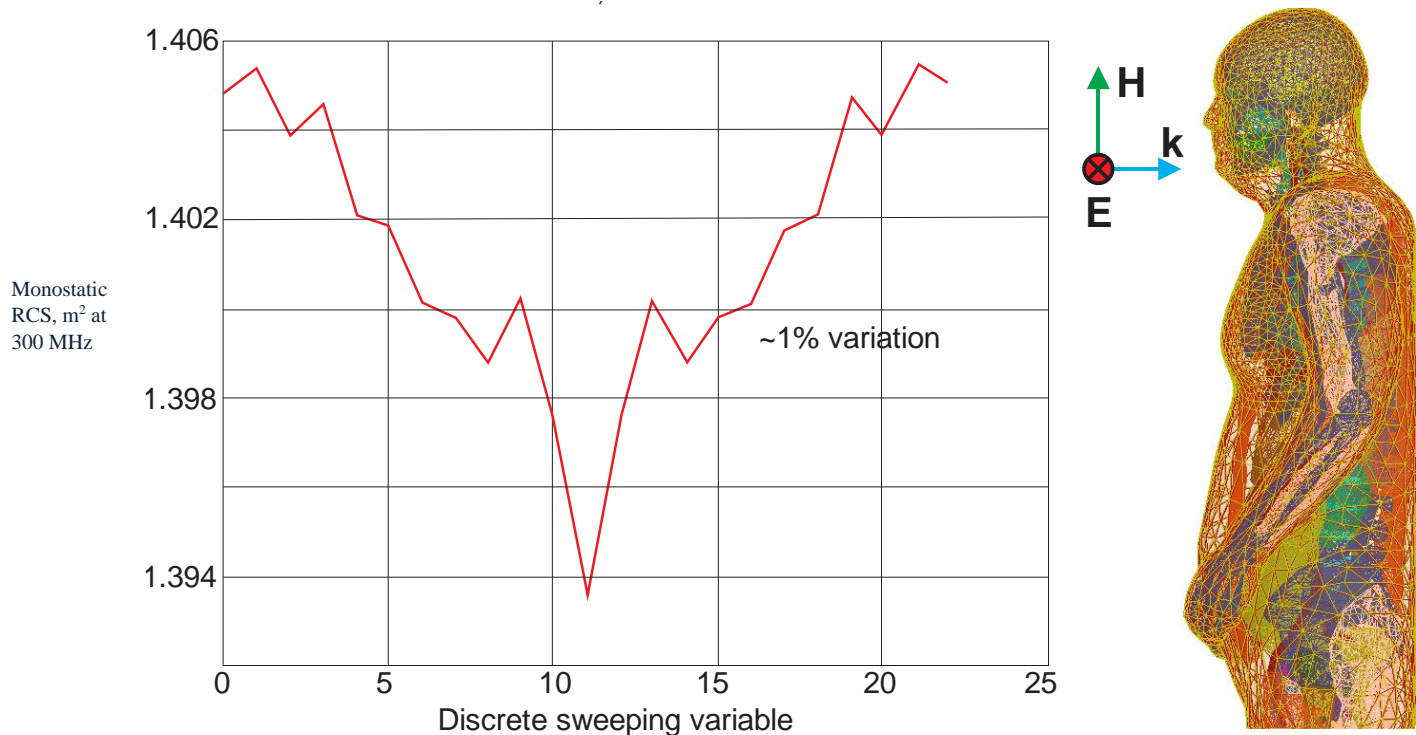
Figure 3. 6. Monostatic RCS during the respiratory cycle.

3.11. MRI Coil Application

Any RF MRI coil such as a high-pass birdcage coil shown in Fig. 3.7 is a familiar LC resonator [14]. The number of distinct coil resonances is equal to the number of the tuning capacitors ("springs") in Fig. 3.7. The resonances appear in pairs corresponding to the pairs of upper and lower capacitors in Fig. 3.7 Magnetic fields of each pair of resonant models are perpendicular to each other. When combined, these two perpendicular fields create a circular polarization [12].

In order to find all resonances of the coil and then select the main resonant mode pair, we will consider the coil in Fig. 3.7 as a generic N-port linear network. Every tuning capacitance is now becoming an independent port driven by an RF generator. The N-port network is routinely modeled in any commercial CEM software at the known resonant frequency. As a result, an impedance matrix $\hat{Z}$ is obtained, which relates port voltages $\vec{V}$ and currents $\vec{I}$. The N distinct coil resonances correspond to N eigenvalues and N eigenvectors of the impedance matrix $\hat{Z}$, very similar to the eigenvalues of the stiffness matrix in mechanical engineering of lumped resonant systems. We find those eigenvalues in MATLAB.

The pair of desired resonant modes must generate a magnetic field that is as homogeneous as possible within the coil. In this case, the current distribution $\vec{I}$ for every mode in the pair must be a sinusoidal function of the polar angle φ around the coil with exactly one period. The corresponding mode strength is computed via a Fourier coefficient. The analysis of mode strength will show two nearly equal clear peaks (10-50 times higher than the rest of the values) corresponding to the two desired modes.

In practice, only a limited number of ports are driven (typically two or four). In order to obtain an "ideal" solution exactly corresponding to the two resonant modes, we will drive *all* ports and assume ideal port matching to the characteristic impedance. Given the known impedance (and scattering) matrices, this condition yields all port sources necessary to excite the best possible B_1^+.
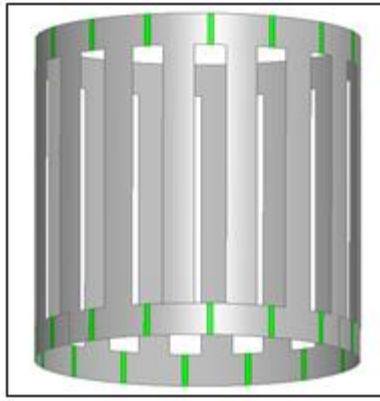
Figure 3. 7. Generic high-pass birdcage MRI RF coil. Tuning capacitors and/or driving ports are marked green.

We constructed and simulated models mimicking maximum exhalation ($t_n = 0$), zero pressure gradient ($t_n = 5$), and maximum inhalation ($t_n = 11$) using the perfectly matched RF coil model. This method uses a resonant-model of a coil optimally driven at all possible ports. The coil is loaded with a dynamic virtual human VHP-Female v. 4.0 (height 162cm/weight 88 kg). Each tissue has been augmented with accurate material properties.
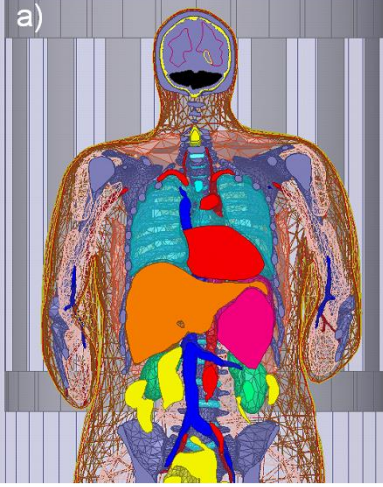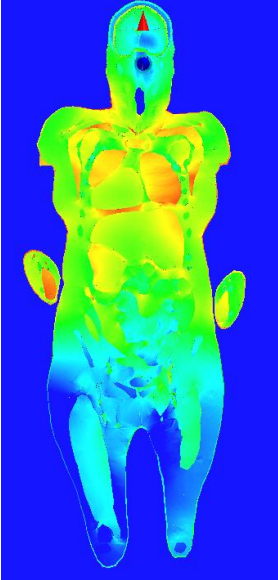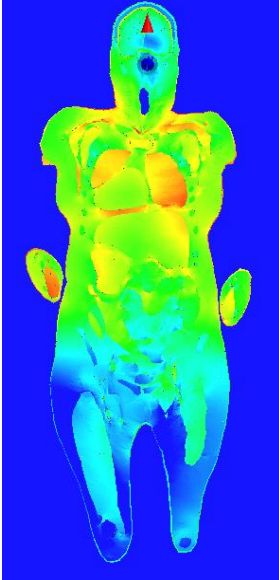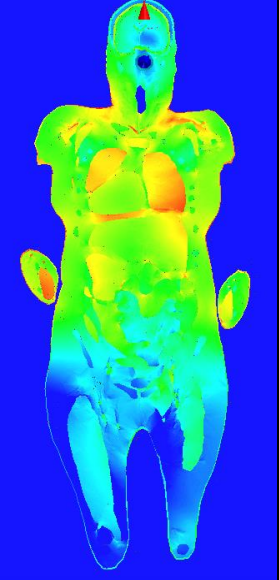
Table 3.3 shows averaged simulation results for the virtual human VHP-Female 4.0 in a high-pass full body birdcage coil shown in Fig. 3.7 at 1.5 T field (64 MHz) at the **shoulder** landmark. The coil has 16 rungs, 32 matching capacitors (each of which can be connected to a driving port), the diameter of 640 mm and the length of 689 mm, which is close to the dimensions given in Ref. [13]. Below, we present resonant capacitance values of the high-pass MRI coil in Fig. 3.7. Our numerical results from Table 3.3 obtained with FEM software ANSYS Electronics Desktop reveal that coil detuning due to respiration is generally *very small*. The maximum deviation occurs somewhere in between maximum exhalation and minimum inhalation.

Table 3. 3. Computed resonant capacitance values for a perfectly-tuned loaded high-pass full-body birdcage coil during the respiratory cycle.

	Max exhalation ($t_n = 0$)	Intermediate time ($t_n = 5$)	Max inhalation ($t_n = 11$)
Resonant Capacitance Mode 1 (pF)	112.0477	112.0863	112.0467
Resonant Capacitance Mode 2 (pF)	112.2804	112.3204	112.2538

A series of figures illustrating Specific Absorption Rate (SAR) variations for various breathing times is given in Table 3.4.

Table 3. 4. SAR 10g plot

Model	SAR: Max exhalation ($t_n = 0$)	SAR: Zero pressure gradient ($t_n = 5$)	SAR: Max inhalation ($t_n = 11$)

Finite-element simulations performed for the anatomically realistic breathing cycle provided quantitative estimates of MRI RF coil detuning. Results demonstrate that detuning is quite small. Such a result is qualitatively expected since the respiratory motion causes negligibly small changes in the body mass.

References

[1] M. V. Siebenthal, "Analysis and Modelling of Respiratory Liver Motion using 4DMRI", Ph.D. dissertation, Elect. Eng. And Inform. Technology Dept., ETH Zurich, Switzerland, 2008.

[2] R. Grimm *et.al*, "Self-gated MRI motion modeling for respiratory motion compensation in integrated PET/MRI," *Medical Image Analysis J.*, vol.19, pp., 110-120, 2015.

[3] A. E. Lujan *et.al*, "A method for incorporating organ motion due to breathing into 3D dose calculations," *Medical Physics*, vol. 26, no. 5, pp. 715-720, 1999.

[4] A. E. Lujan, J. M. Balter, and R. K. Ten Haken, "A method for incorporating organ motion due to breathing into 3D dose calculations in the liver: Sensitivity to variations in motion," *Medical Physics*, vol.30, no.10, pp. 2643-2649, 2003.

[5] W. P. Segars, D. S. Lalush, and B. M. W. Tsui, "Modelling respiration mechanics in the MCAT and spline-based MCAT phantom," *Nuclear Science Symp.*, Seattle, WA, 1999, vol. 2, pp. 985-989.

[6] Y. Wang, S. Riederer, and R. Ehman, "Respiratory motion of the heart: Kinematics and the implications for the spatial resolution in coronary imaging," *Magnetic Resonance in Medicine*, vol. 33, no. 5, pp.713-719, 1995.

[7] J. West, *Respiratory Physiology*, 5th ed. Baltimore, Williams and Wilkins, 1995.

[8] R. Zeng, "Estimating Respiratory Motion from CT Images via Deformable Models and Priors," Ph.D. dissertation, Elect. Eng. Dept., University of Michigan, Ann Arbor, Michigan, 2007.

[9]   J. Eom *et.al*, "Predictive modeling of lung motion over the entire respiratory cycle using measured pressure-volume data, 4DCT images, and finite element analysis," *Medical Physics*, vol. 37, no.8, pp. 4389 – 4400, 2010.

[10] J.Yanamadala, G. M. Noetscher, S. Louie, A. Prokop, M. Kozlov, A. Nazarian, S. N. Makarov, "Multi-Purpose VHP-Female Version 3.0 Cross-Platform Computational Human Model,"*10th European Conference on Antennas and Propagation 2016 (EuCAP16)*, Davos, Switzerland, April 2016.

[11] P.A. Hasgall, F. Di Gennaro, C. Baumgartner, *et al.*, "IT'IS Database for thermal and electromagnetic parameters of biological tissues," Version 2.6, Jan. 13th, 2015.

[12] Lemdiasov, R, Obi A, Ludwig R. A Numerical Postprocessing Procedure for Analyzing Radio Frequency MRI Coils. Concepts in Magnetic Resonance Part A. 2011; 38A(4): 133–147.

[13] Cabot E, Christ A, Kuster N. Whole Body and Local SAR in Anatomical Phantoms Exposed to RF Fields from Birdcage Coils. Proceedings of the 29th General Assembly of the International Union of Radio Science. 2008 August 7-16. Chicago, USA.

[14] Tankaria, "Validation of CAD Phantoms", Master Thesis. 2017 April 24th. Worcester, MA, USA

IV. Chapter 4: Development of VHP Female 4.0 and Review of Prior Work

4.1. Review of Prior Work

4.1.1. Source

The VHP-Female CAD model is a full-body computational human model based on the open-source Visible Human Project®-Female cryosection dataset of the U.S. National Library of Medicine (NLM) [1]. Fig. 4.1 shows an example of the color cryosection image.

Figure 4. 1. Abdominal image of the cryosection dataset.

4.1.2. Construction

The model includes more than 250 parts that were created with manual and semi-automatic image segmentation, in particular via ITK-SNAP [2], [3] and MATLAB. Mesh processing algorithms [4]-[6] were developed for surface extraction and mesh healing. The VHP-Female v.3.0 is distributed in the form of 3D CAD objects. Each 3D CAD object is a triangular surface mesh as consisting of triangle vertices, *t*, and vertex coordinates, *P*. Fig. 4.2 shows an example of the intestine object with triangular mesh.
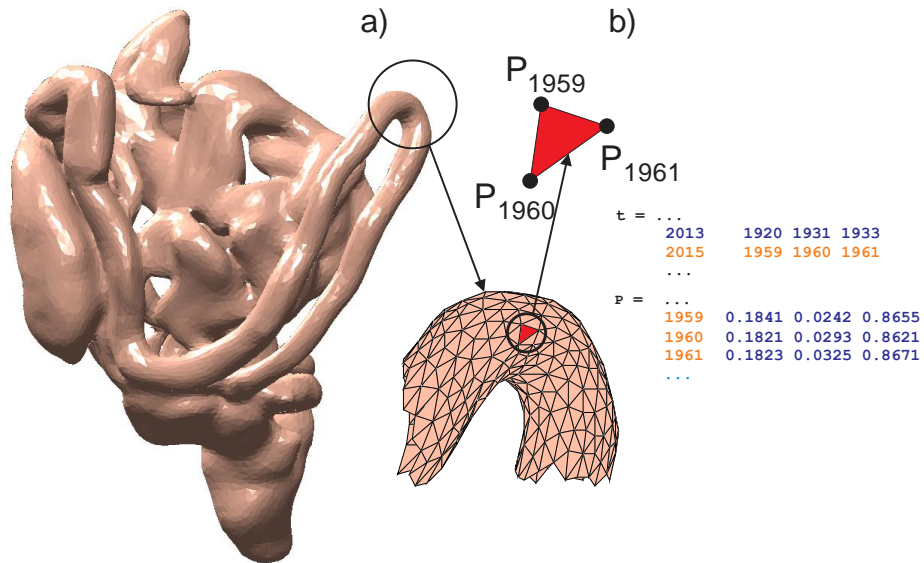
Figure 4. 2. Small and large intestines, and the rectum – derived from Visible Human Project® Female.

4.1.3. Model anatomy and materials

The model was assigned tissue properties from the IT'IS database, which covers the frequency band from 10 Hz to 100 GHz [7].

4.1.4. Three previous model versions

A. VHP-Female version 3.0 BASE (DOI:10.20298/VHP-Female-V.3.0-BASE)

VHP-Female 3.0 Base is the *master* model version, which serves as the base for the creation of the two previous versions. It has 26 individual tissues and 216 separate tissue parts with approximately 160,000 triangular facets in total.

B. VHP-Female version 3.0 SMOOTH (DOI:10.20298/VHP-Female-V.3.0-SMOOTH)

The SMOOTH version is similar to the BASE version but all tissue objects were smoothed to avoid sharp edges and reduce file size. The total number of triangular facets is about 600,000. This version is primarily intended for more accurate SAR calculations.

The differences between BASE and SMOOTH models are demonstrated in Fig. 4.3. It shows the same view of the cardiovascular system for BASE and SMOOTH models, respectively. Fig. 4.4 shows thorax of the SMOOTH model.

C. VHP-Female version 2.2 (DOI:10.20298/VHP-Female-V.2.2)

This is VHP-Female version 3.0 BASE with all individual muscles removed (except for the cardiac muscle). The average body object may now be assigned muscle properties.
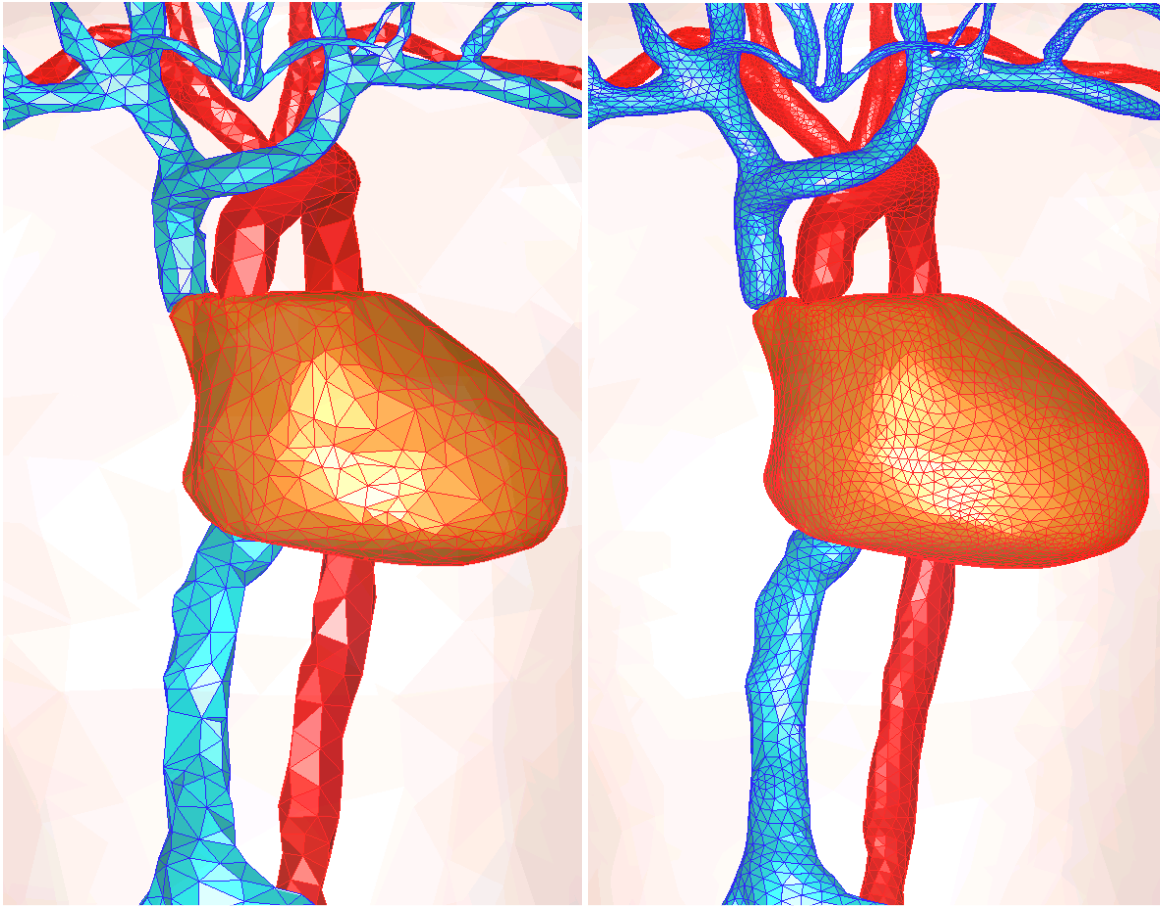
Figure 4. 3. Right –BASE model v. 3.0; Left –the SMOOTH model v.3.0.
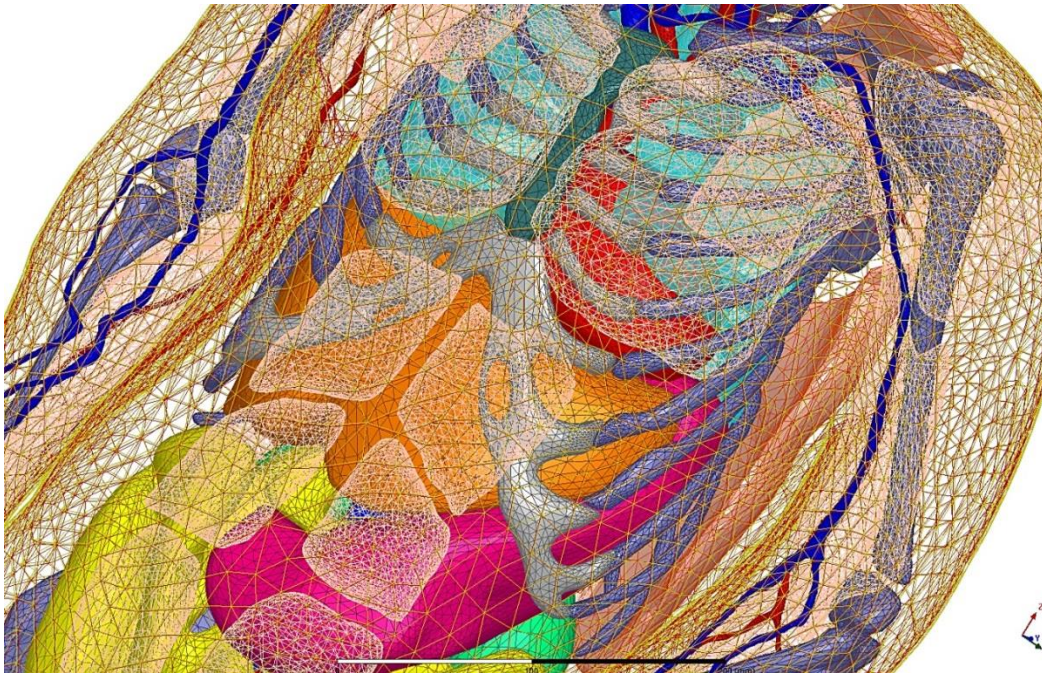
Figure 4. 4. Thorax of the SMOOTH model V. 3.0 with some objects removed for clarity.
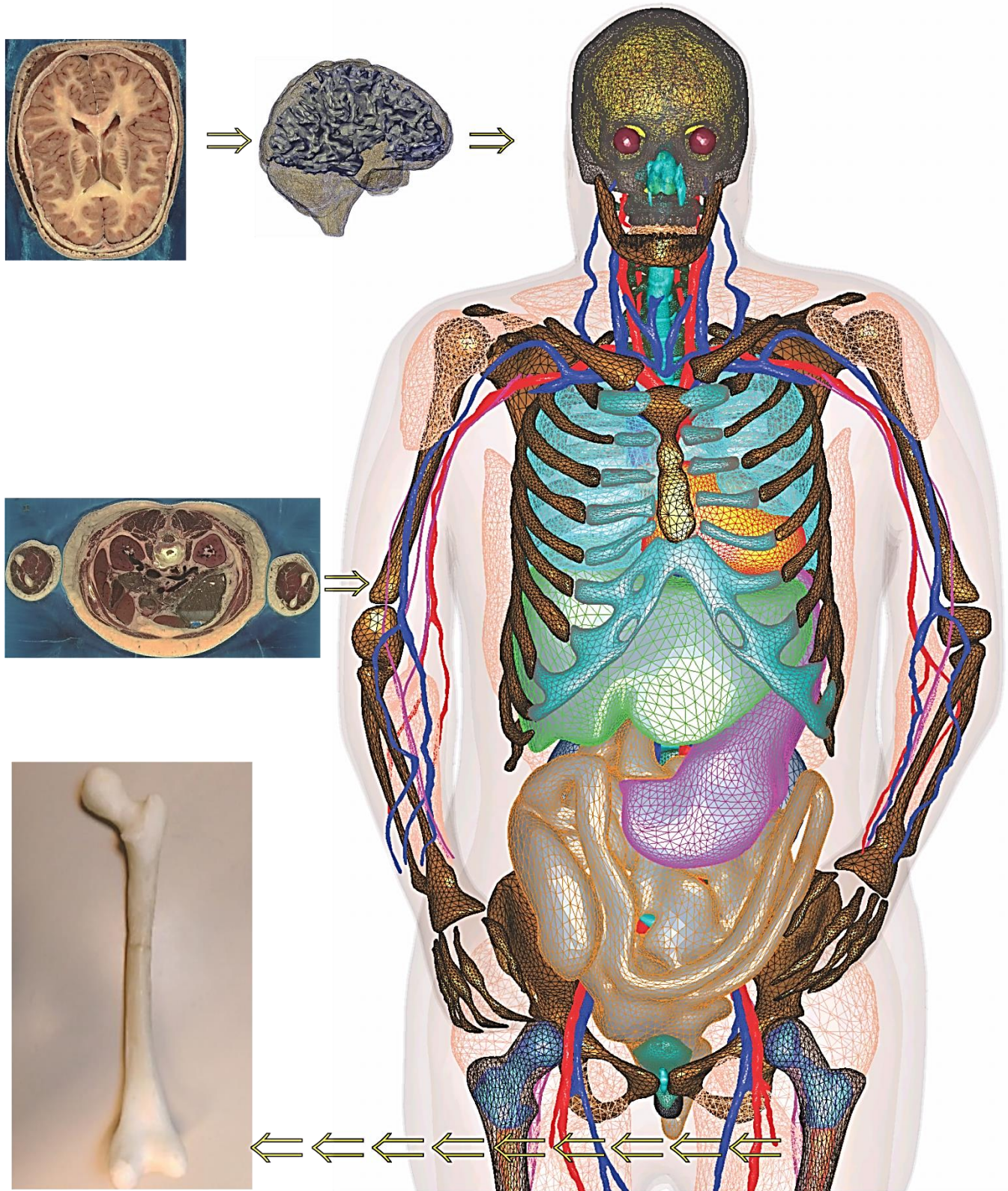
Figure 4. 5. Anterior view of the VHP-Female computational smooth model v. 3.1 within MATLAB. Illustration of tissue object development from cryosection images and 3D printed femur model is shown. Fat and some muscles are not shown.

52
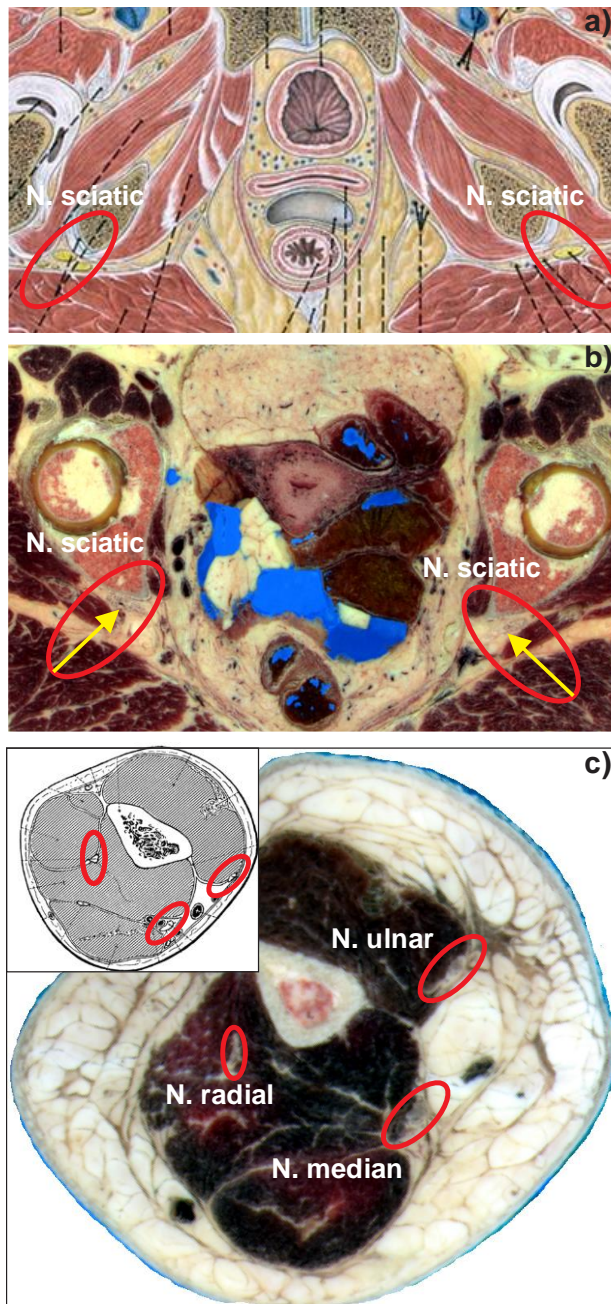
Figure 4. 6. A) Graphical representation of sciatic nerves in the lower pelvic region; b) Illustration of sciatic nerves in the cryosection image from the original dataset; c) Graphical and realistic illustration of radial, median, and ulnar nerves in the forearm region.

4.2. VHP 3.1

4.2.1. Peripheral Nervous System and Cardiovascular System

Version VHP-Female v.3.1 includes a state of the art peripheral nervous system which currently comprises of the radial, median, ulnar (brachial plexus) and sciatic nerves (sacral plexus) segmented over large lengths. The inclusion of cauda equina along with the brachial plexus and sacral plexus is the first of its kind, a unique feature of the VHP-F v3.1 model. This peripheral nervous system can be used to study and model various electrical stimulation systems operating on peripheral nerves. Note that a high-resolution human-head model developed in [8] has the detailed representation of twelve cranial nerves.

Figure 4.6 shows the graphical cartoon representations of the sciatic nerve, as well as radial, median, and ulnar nerves along with the corresponding anatomical counterparts identified in the cryosection images from the VHP-Female image dataset. The nerve tissues were registered and segmented via segmentation algorithms developed in MATLAB. The final triangular tissue mesh structure was built upon the existing point cloud, in the form of connective cylindrical elements with varying radii and lengths. The traditional ball-pivoting algorithm is not effective; it generated results with a low mesh quality since the nerves are extremely thin as compared to other tissues. Table 4.1 summarizes the existing nervous tissues and other nervous tissues considered for future development.

Table 4. 1. Nerves in VHP-Female v. 3.1.

Tissue Name	Triangle Size	Mesh Quality	Minimum Edge Length
Median Nerve left	520	7.14E-7	0.39E-3
Median Nerve Right	492	5.24E-4	0.64
Sciatic Nerve left	946	9.63E-4	0.60
Sciatic Nerve right	932	7.23E-2	0.59
Radial Nerve left	762	8.24E-6	0.03
Radial Nerve right	476	2.66E-3	0.34
Ulnar Nerve left	476	1.90E-2	0.57
Ulnar Nerve right	540	4.51E-6	0.02
Peroneal Nerve left Peroneal Nerve right Femoral Nerve left Femoral Nerve left Saphenous Nerve left Saphenous Nerve right	Under development		

4.2.2. Muscular System

Along with the heart muscle, the muscular system includes major skeletal muscles (32 in total) in the form of separate objects. All muscle objects are contained within the average body object depicted in Fig. 4.5. Table 4.2 summarizes the muscle and other enhancements in VHP-Female v.3.1 as compared to VHP-Female v.2.0.

Table 4. 2. Major Improvements in VHP-Female v. 3.1

	List of improvements
1	Development of sciatic nerve, ulnar nerve, radial nerve, and median nerve for the right and left sides, respectively (state-of-the-art). Addition of spinal cord cauda 55quine.
2	Expansion of systemic venous system.
3	Addition of erector spinae left and erector spinae right.
4	Addition of pectoralis major left, pectoralis major right, pectoralis minor left, and pectoralis minor right.
5	Addition of abdominals left bottom, abdominals left middle, abdominals left top, abdominals right bottom, abdominals right middle, and abdominals right top.
6	Improvement of ribs left2, ribs left7, ribs left8, ribs left9, ribs right2, ribs right7, ribs right8, and ribs right9 along with respective cartilages. Separation of clavicle left, clavicle right from the rib cage.
7	Development of forearm flexors left and right, respectively.
8	Anatomical accuracy of Grey Matter/White Matter/Intestine/Bladder improved.
9	Development of spleen.
10	Sharp corners have been smoothed, which improves the numerical accuracy.

4.2.3. Smoothed Model and Base Model

The VHP-Female v.3.1 model includes two versions: BASE (~160,000 facets) and SMOOTH (a smoothed version for more accurate SAR calculation with ~640,000 facets). The corresponding doi numbers (related to version 3.0) are as follows:
DOI:10.20298/VHP-Female-V.3.0-BASE
DOI:10.20298/VHP-Female-V.3.0-SMOOTH

The BASE and SMOOTH versions of the model have the same number of tissues and identical topology. The corresponding surface deviation between the two models does not exceed 0.2-1 mm on flat surfaces, but may be as high as 2-7 mm for sharp edges and corners.

During electromagnetic simulations the surface charge density formally becomes singular at any edge (not necessarily sharp) of a triangular mesh with non-planar triangles. For sharper edges and large adjacent triangles, this local (electrostatic) effect becomes quite significant and may lead to non-physical field/current peaks [9]. This deficiency has been addressed in the smooth version.

4.3. Improvements of VHP 3.1 and the creation of VHP Female 4.0

Some tissues were re-segmented to improve anatomical accuracy of the VHP 3.0 model. These tissues were segmented by a group of undergraduate students, led by Dr. Noetscher and Dr. Yanamadala, during Summer 2016. They include: cranium system, 3-piece hand phalanges, more accurate version of ribcage, ulna radius bones, humerus bones, along with nerve system: median nerves, peripheral nerves, radial nerves, and ulnar nerves. These tissues (which include resolving intersections, non-manifold edges, and registering them into correct anatomical positions in VHP model) were post-processed during the course of this thesis with SpaceClaim, MATLAB, and finally validate the new model, VHP Female 4.0, in ANSYS HFSS.

4.3.1. VHP Female 4.0 Accurate

The VHP Female 4.0 Accurate model has 703,804 triangles in total. The list of updated tissues is showed in table 4.3.

Table 4. 3. List of updated triangular surface meshes – version Accurate 4.0

Legend:

Hard tissues	Soft tissues	Individual muscles	Cartilage	Nervous tissues

Mesh no	Tissue name	Triangle size	Mesh quality	Min. Edge Length	Tissue Type
1	VHP Hands1 1Phalange left	336	0.561483603	1.304314414	Bone
2	VHP Hands1 1Phalange right	482	0.378796118	0.638968718	Bone
3	VHP Hands1 2Phalange left	214	0.597801938	0.930630174	Bone
4	VHP Hands1 2Phalange right	488	0.38328069	0.699838528	Bone
5	VHP Hands1 3Phalange left	162	0.223276291	0.919268845	Bone
6	VHP Hands1 3Phalange right	238	0.452761662	0.703404769	Bone
7	VHP Hands2 1Phalange left	554	0.444626812	0.840372777	Bone
8	VHP Hands2 1Phalange right	590	0.239003725	0.458872814	Bone
9	VHP Hands2 2Phalange left	362	0.276602112	0.875563174	Bone
10	VHP Hands2 2Phalange right	392	0.334744462	0.960809081	Bone
11	VHP Hands2 3Phalange left	206	0.35070248	1.017620025	Bone
12	VHP Hands2 3Phalange right	240	0.30728027	0.598465013	Bone
13	VHP Hands3 1Phalange left	446	0.543485609	1.206987764	Bone
14	VHP Hands3 1Phalange right	504	0.272527569	1.003718792	Bone
15	VHP Hands3 2Phalange left	444	0.418436275	0.97062073	Bone
16	VHP Median Nerve left	520	7.14E-07	0.000393	Nerves
17	VHP Median Nerve Right	492	0.000525	0.641692	Nerves

18	VHP Peripheral Nerve left	946	0.000964	0.600546	Nerves
19	VHP Peripheral Nerve Right	942	0.037591	0.592834	Nerves
20	VHP Radial Nerve left	762	8.24E-06	0.028498	Nerves
21	VHP Radial Nerve Right	476	0.002656	0.337785	Nerves
22	VHP Ulnar Nerve left	476	0.019017	0.571556	Nerves
23	VHP Ulnar Nerve right	540	4.51E-06	0.018062	Nerves
24	VHP Humerus left	3312	0.145005	1.220145	Bone
25	VHP Humerus right	1634	0.071029	1.374185	Bone
26	VHP Ulna Radius left	2020	0.029373	1.193387	Bone
27	VHP Ulna Radius right	2168	0.002459	0.847576	Bone

Fig. 4.7 depicts the upper body of VHP Female 4.0 Model
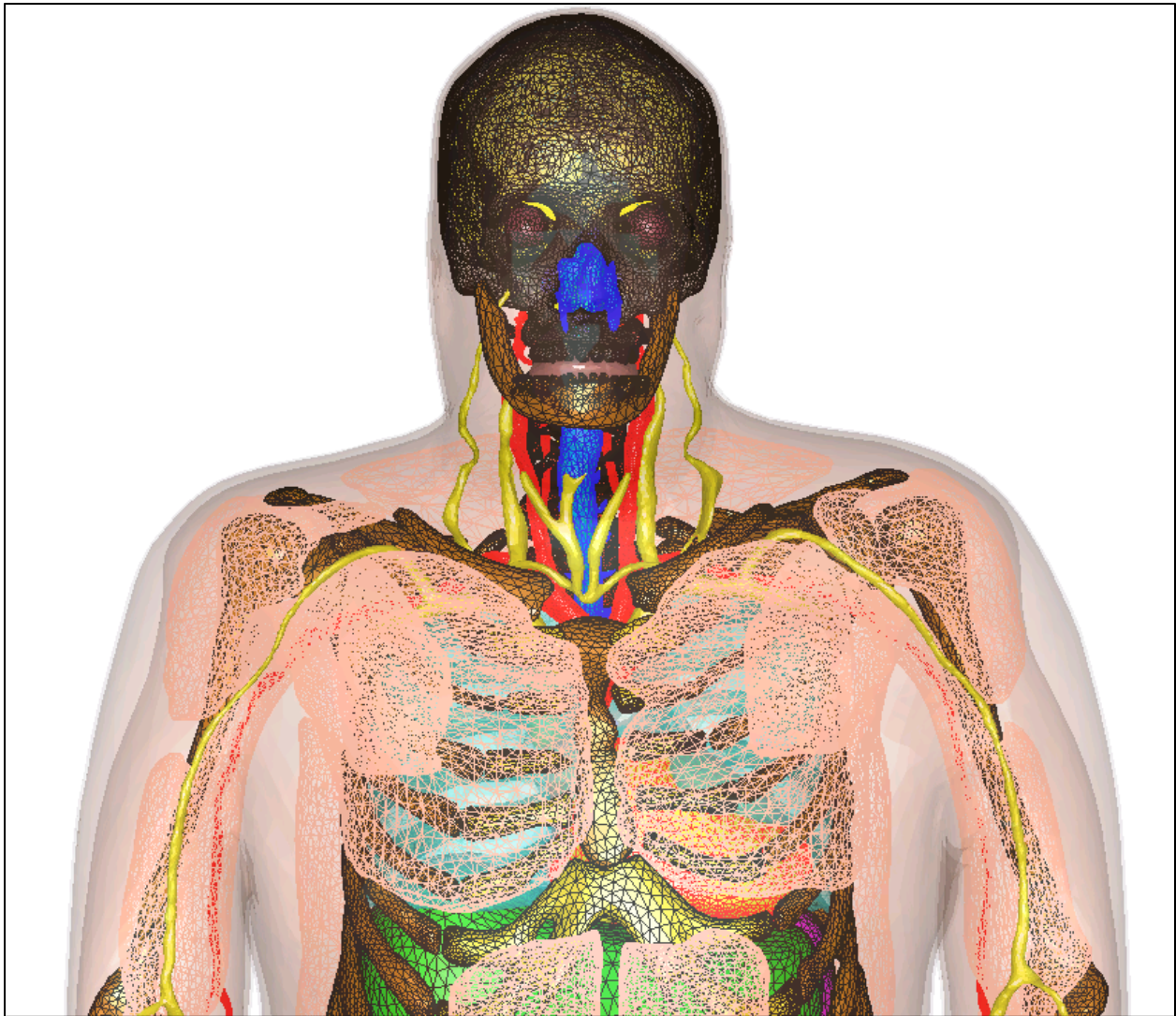
Figure 4. 7. VHP Female 4.0 Accurate Model

4.3.2. VHP Female 4.0 Coarse

The VHP Female 4.0 Coarse model has 239,750 triangles in total. The list of updated tissues is showed in Table 4.4. The nerve system is not smoothed in this model since smoothing creates many non-manifold edges for these particularly thin objects.

Table 4. 4. List of updated triangular surface meshes – version Coarse 4.0

Mesh no	Tissue name	Triangle size	Mesh quality	Min. Edge Length	Tissue Type
1	VHPC Hands1 1Phalange left	84	0.176998	2.690592	Bone
2	VHPC Hands1 1Phalange right	164	0.061583	0.504562	Bone
3	VHPC Hands1 2Phalange left	56	0.302763	1.449309	Bone
4	VHPC Hands1 2Phalange right	122	0.151791	1.311807	Bone
5	VHPC Hands1 3Phalange left	50	0.058059	1.007602	Bone
6	VHPC Hands1 3Phalange right	62	0.207892	1.223796	Bone
7	VHPC Hands2 1Phalange left	148	0.194523	0.857429	Bone
8	VHPC Hands2 1Phalange right	180	0.086288	0.391133	Bone
9	VHPC Hands2 2Phalange left	94	0.035919	1.534043	Bone
10	VHPC Hands2 2Phalange right	108	0.155599	0.98453	Bone
11	VHPC Hands2 3Phalange left	60	0.189769	0.949441	Bone
12	VHPC Hands2 3Phalange right	66	0.096104	1.966641	Bone
13	VHPC Hands3 1Phalange left	114	0.20466	2.057431	Bone
14	VHPC Hands3 1Phalange right	130	0.171594	1.868806	Bone
15	VHPC Hands3 2Phalange left	122	0.116624	1.099541	Bone
16	VHP Median Nerve left	520	7.14E-07	0.000393	Nerve
17	VHP Median Nerve Right	492	0.000525	0.641692	Nerve
18	VHP Peripheral Nerve left	946	0.000964	0.600546	Nerves
19	VHP Peripheral Nerve Right	942	0.037591	0.592834	Nerves
20	VHP Radial Nerve left	762	8.24E-06	0.028498	Nerves
21	VHP Radial Nerve Right	476	0.002656	0.337785	Nerves
22	VHP Ulnar Nerve left	476	0.019017	0.571556	Nerve
23	VHP Ulnar Nerve	540	4.51E-06	0.018062	Nerve

	right				
24	VHPC Humerus left	832	0.090109	1.926172	Bone
25	VHPC Humerus right	542	0.007326	0.785727	Bone
26	VHPC Ulna Radius left	532	0.019598	1.731927	Bone
27	VHPC Ulna Radius right	576	0.032006	1.322617	Bone
28	VHPC Cartilage1 Left	218	0.024327	0.546884	Cartilage
29	VHPC Cartilage1 Right	264	0.00421	0.629753	Cartilage
30	VHPC Cartilage2 Left	212	0.003714	0.268947	Cartilage
31	VHPC Cartilage2 Right	236	0.007489	0.176654	Cartilage
32	VHPC Cartilage3 Left	264	0.007836	0.415367	Cartilage
33	VHPC Cartilage3 Right	298	0.007715	0.566641	Cartilage
34	VHPC Cartilage4 Left	302	0.014622	0.831348	Cartilage
35	VHPC Cartilage4 Right	370	0.015893	0.625633	Cartilage
36	VHPC Cartilage5 Left	410	0.002503	0.625342	Cartilage
37	VHPC Cartilage5 Right	340	0.007511	0.990092	Cartilage
38	VHPC Cartilage6 Left	380	0.003614	0.508669	Cartilage
39	VHPC Cartilage6 Right	305	0.006257	0.258616	Cartilage
40	VHPC Sternum	2622	0.001007	0.124601	Bone

Figure 4.8 shows the VHPC 4.0 model, and Fig. 4.9 demonstrates the 3-piece phalanges hand
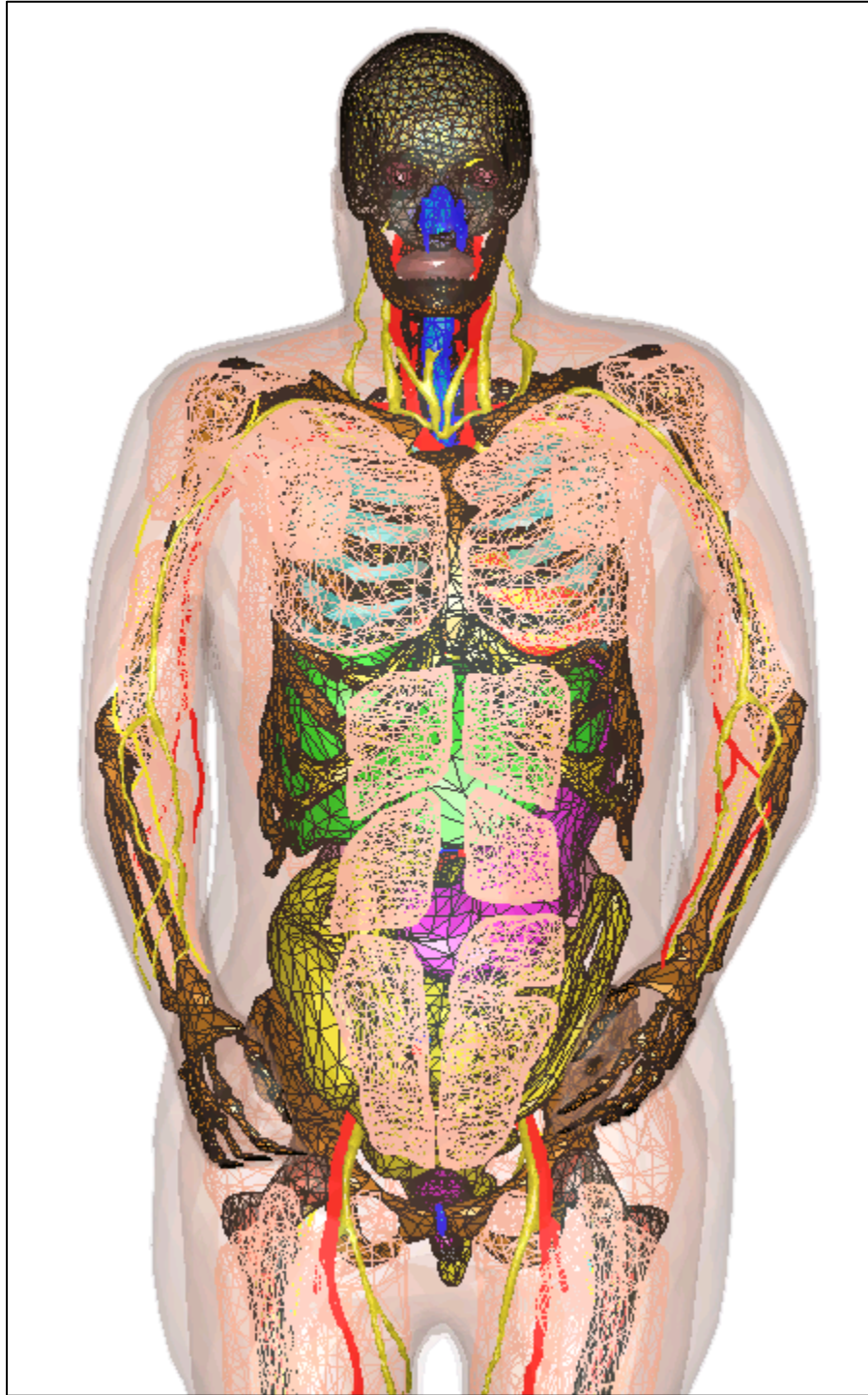
Figure 4. 8. VHP Female Coarse Model
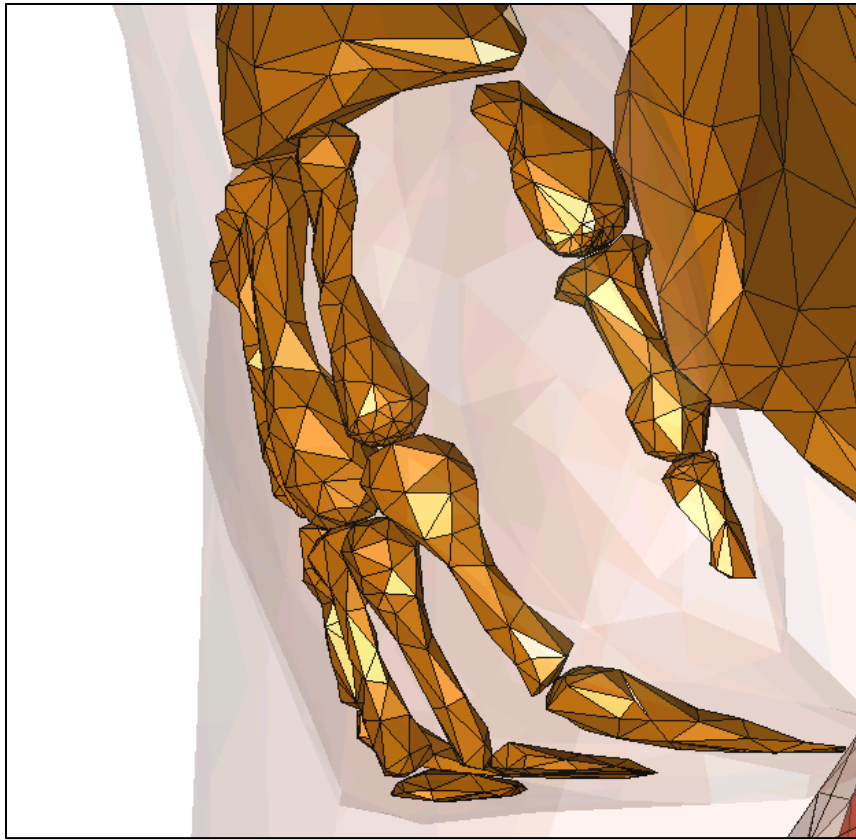
Figure 4. 9. VHPC 4.0 Right hand: 3-piece phalanges

References

[1] U.S. National Library of Medicine. The Visible Human Project®. Online: https://www.nlm.nih.gov/research/visible/visible_human.html

[2] P. A. Yushkevich, P. Joseph, C. Heather, et. Al., "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. **31**(3), pp.1116-28, 2006.

[3] ITK SNAP, P. Yushkevich, University of Pennsylvania. Online: http://www.itksnap.org/pmwiki/pmwiki.php

[4] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics*, vol. **21**(4), pp. 163-169, 1987.

[5] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball pivoting algorithm for surface reconstruction," *IEEE Trans. Vis. And Comp. Graphics*, vol. **5**, no 4, Oct-Dec. 1999.

[6] T. Möller and B. Trumbore, "Fast, Minimum Storage Ray/Triangle Intersection", *Journal of Graphics Tools*, vol. **2**(1), pp. 21-28, 1997.

[7] Hasgall PA, Di Gennaro F, Baumgartner C, Neufeld E, Gosselin MC, Payne D, Klingenböck A, Kuster N, "IT'IS Database for thermal and electromagnetic parameters of biological tissues," Version 2.6, January 13th, 2015. www.itis.ethz.ch/database

[8] M. Iacono *et al.*, "MIDA: A Multimodal Imaging-Based Detailed Anatomical Model of the Human Head and Neck," *PloS ONE*, vol. 10, no 4, online April 22, 2015

[9] S. N. Makarov *et al.*, "Preliminary Upper Estimate of Peak Currents in Transcranial Magnetic Stimulation at Distant Locations from a TMS Coil," *IEEE Trans. Biomedical Engineering*, vol. 64, accepted. Early access:

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7352318&filter%3DAND%28p_IS_Number%3A4359967%29

V. Chapter 5: Future VHP-Female Applications

5.1. Extended CAD virtual family

Fig. 5.1a-c shows the concept of the extended virtual family. It is based on NEVA EM *CAD body shell library*.

a) accurate anatomical cranium model: skull, CSF shell grey matter, white matter, cerebellum, ventricles, tonque, jaw, teeth, air cavity, eyes

b) anatomical body shells

c) skin shell, fat shell, and average-body shell

Volume occupied by average body tissue

Volume occupied by average body tissue

Volume occupied by fat tissue

Volume occupied by skin tissue
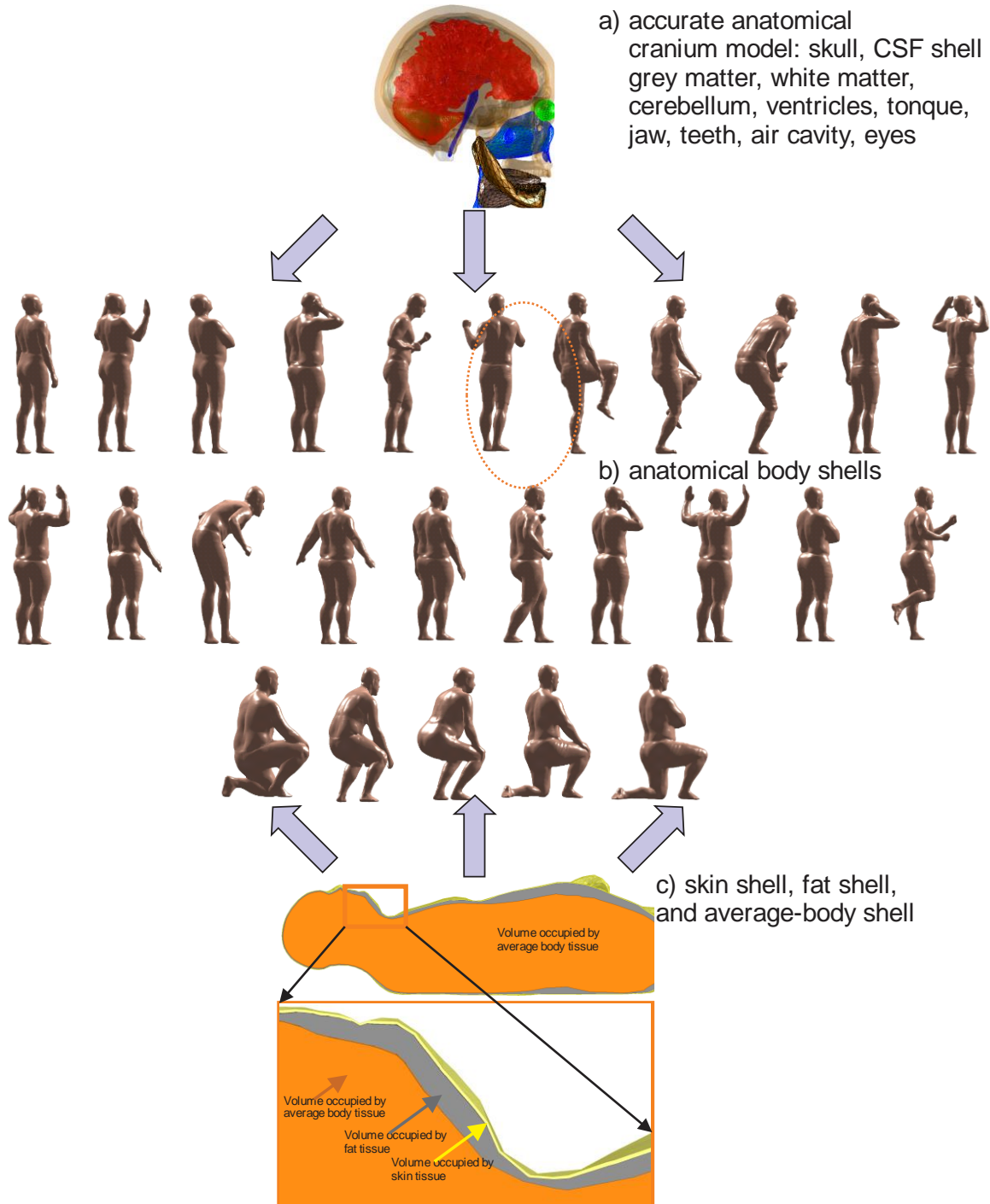
Figure 5. 1. Concept of the extended virtual family.

This anatomical shell library has been created via a 3D full-body laser scanner from CyberWare located at the US Army Natick Soldier Research, Development, and Engineering Center, Natick, MA and then accurately processed in MATLAB. The original scan resolution is 1.0 mm. Assume that a customer manufacturing body area networks needs a model of 30 year old person circled in Fig. 5.1b. Then,

A. Along, with the scanned shell, first create two anatomically-relevant inner body shells as described below and shown in Fig. 5.1c.
B. Next, embed a detailed cranium/cerebral cortex model adopted from either *base anatomical member* of the virtual family into the inner shell via a set of affine transformations as described below and shown in Fig. 5.1a.

5.1.1. Comparison with the existing "poser" approach

This concept is fundamentally different from the standard approach today, which implies using a human model "poser". This approach typically employs only one model, but subject to full-body *free-form* deformations. Examples include advanced posers of ZMT-Zurich MedTech AG (Switzerland) [1] and VariPose of REMCOM [2]. The human-body poser may transform a standing person into a sitting one, indeed. However, it could hardly transform a 60 year old male into a 20 year young female while maintaining a realistic body composition including body fat distribution. In contrast to this, our approach will be significantly more realistic, but it will require more efforts.

5.1.2. Anatomical origin and justification of our approach – cranium/head model

Ongoing discussions with the staff of Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany (Dr. M Kozlov, Dr. N. Weiskopf, and PD Dr. S. Geyer) reveal that the standard (most common) *scalp* thickness from MRI data is approximately 5 mm. This data has been obtained from the analysis of approximately 3,000 T1/T2 MRI head datasets. The further suggestion is to have 2 mm avg body, 2 mm fat, and 1 mm skin (2+2+1 =5 mm total) for three shells around the skull. The same three shells should continue around the entire body, but the fat thickness will indeed be quite different. Using this model, the expected unique skull shape (including frontal bone, two parietal and temporal bones) will be established from the 3D laser head scan surface minus 5 mm. Then, a set of affine and free-form transformations will be applied to the anatomical model of cranium/cerebral cortex to register the model for the given body shell.

5.1.3. Anatomical origin and justification of our approach – fat distribution model

Fat pattering in adult male and female subjects of different ages has been an active area of research for a long time [3]. To construct the unique fat shell, the average data available for four different age groups and at 9 different body locations will be used.

5.1.4. Topology of the entire virtual family

The CAD virtual family will possess the following topological characteristics necessary for cross-platform compatibility:

1. Every original tissue triangular surface mesh will be strictly *2-manifold* [4] or *thin-shell*.
2. No original tissue triangular surface mesh will have any triangular facets *in contact* with other tissue surfaces. In other words, there will always be a (small) *gap of a minimum of 1 mm* between the distinct tissue surfaces. Physically, this gap represents thin membranes separating distinct tissues and is therefore anatomically correct.
3. At the same time, there will exist tissues *fully enclosed* within each other, such as the white matter inside the gray matter, cancellous bone inside a cortical bone shell, etc.
4. Every original tissue triangular surface mesh will have nearly the same segmentation accuracy (resolution) and the *minimum surface triangular mesh size* allowed.
5. Every original tissue triangular surface mesh will possess the *maximum possible average triangle quality* (defined as twice the ratio of incircle radius to circumcircle radius).

6. Every tissue triangular surface mesh will be made available in open-source basic CAD body geometry formats: STL and PLY, as well as in in MATLAB format.

5.1.5. Creating a new member of the family

In order to perform the initial registration of the cranium and intracranial volume with a particular body shell, a complete set of affine transformations will be used: three translations, three rotations, and three uniform scaling operations similar to the methodology of breathing human model. This step will overlap the head of the average-body shell and the skull with the *minimum* surface deviation error.

After that, deformations in the normal directions will be applied with a variable scalar amount of deformation, $d_0(\mathbf{r})$, equal to the normal distance between the skull of the base model and the average-body shell of the model under test, for every small solid angle with the origin at the center of gravity. An amount of deformation will be locally applied to every intracranial tissue (CSF shell, grey matter, white matter, cerebellum, ventricles, etc.) and will decrease as $d(\mathbf{r}) = d_0(\mathbf{r})(\mathbf{r} - \mathbf{r}_0)/\mathbf{r}_0$ where $\mathbf{r}_0$ is the center of gravity of the cranium. Fig. 5.2 shows a final product of such operation for a test case: the cranium accurately registered with a test shell human model under study

Figure 5. 2. Cranium registered with a three-shell human body model. ©NEVA EM.

containing three shells: skin, fat, and average body. In accordance with the discussion given above, the average scalp thickness will always be fixed at 5 mm. It is hardly possible to perform this operation in professional mesh processing software packages such as SpaceClaim® of ANSYS. Center of gravity of an object can be found by:

$$\mathbf{r}_0 = \sum \mathbf{r}_i * A_i / A_i$$

Where r_i is coordinate of the center of triangle I, and A_i is the area of triangle i.

5.1.6. Constructing "onion" full-body shells for a new member of the extended family

To construct the unique average body/fat shells, the average data available for at least four different age groups and at 9 different body locations (cf. [3] and other sources) will be used. Table 5.1 lists average male fat distribution at different ages for different body regions. The most outer body shell (skin) and construct the next (fat) shell will be created first, based on the fixed skin thickness of 1mm. Skin volume is the volume between skin and fat shells. To do so normal deformations in the direction of the inner normal vector will be applied. Areas of special concern are head (lips, ears), underarms, fingers, toes and the reproductive organs. These areas typically need special healing operations. After the second fat shell has been created, deformations in the normal directions,
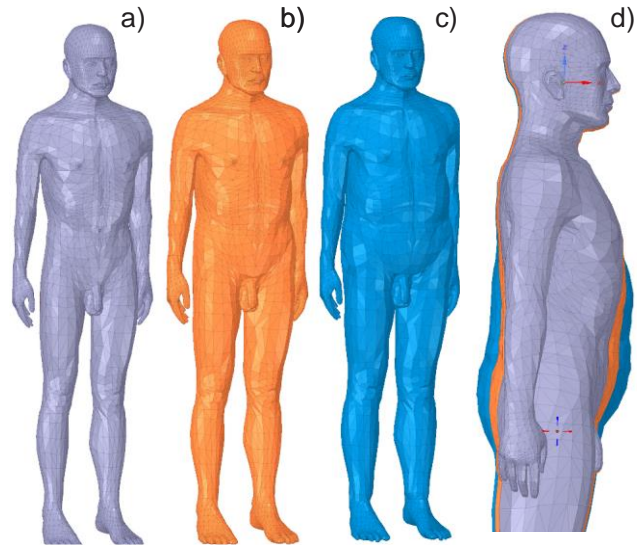
Figure 5. 3. Construction of body shells for a test case.

described in the previous section will be applied with a variable scalar amount of deformation, $d_0(\mathbf{r})$, obtained from interpolation of the data given in Table 5.1 (for a male shell). This will result in an average body shell; the volume between the fat shell and the average body shell will be fat (cf. Fig. 5.1c). Mesh

healing will be again necessary in critical areas listed above. Fig. 5.3 shows a final product of such operations for a test case. The average-body inner shell (Fig. 5.3a) is within the fat shell (Fig. 5.3b or Fig. 5.3c depending on the required fat volume), which, in turn, is within the skin shell separated by 1 mm. The result is shown in Fig. 5.3d. The skin shell cannot be seen due to its small thickness.

Table 5. 1. Average male fat distribution (fat shell thickness in mm for different body locations) [3].

Measurements	20-29	30-39	40-49	50-
Deltoid	14.8	18.6	18.7	18.3
Lateral Arm	4.2	4.7	4.3	5.1
Medial Arm	3.2	3.3	3.4	4.1
Illiac	13.5	17.9	20.8	21.3
Trochanteric	12.3	16.6	17.5	18.4
Posterior leg	6.5	7.3	6.9	8.2
Anterior Leg	2.8	2.8	2.6	2.8
Medial Leg	5.0	6.3	6.2	7.3
Lateral Leg	4.1	5.3	4.5	4.7

5.2. Conclusion and Future Work

In this chapter, the future potential developments of the platform-independent full-body electromagnetic model, the Visible Human Project® (VHP)-Female V.4.0, originated from the U.S. National Library of Medicine, are described. This Visible Human Project® (VHP)-Female V.4.0 has formed a platform to build a family of models, including adult male, adult woman and children with variable BMI. Various 3D custom implant models such as pacemakers/pulse generators, can be registered into the VHP-F model for device modeling studies.

References

[1].   ZMT-Zurich MedTech AG. Available: https://www.zurichmedtech.com/

[2].   REMCOM's VariPose. Available: http://www.remcom.com/varipose

[3].   S.M. Garn and R.V. Harper, "Fat Accumulation and Weight Gain in the Adult Male," *Human Biology*, vol. 27, no. 1, Feb. 1955, pp. 39-49.

[4].   Manifolds online: Dr. Ching-Kuang Shene, http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/model/manifold.html.

[5].   M. Kozlov, G. Noetscher, A. Nazarian, and S. Makarov, "Comparative analysis of different hip implants within a realistic human model located within a 1.5T MRI whole-body coil," *37th Annual Int. Conf. of the IEEE Eng. In Medicine and Biology Society* (EMBC 2015), Milano, Italy, Aug. 25-29, 2015, pp. 7913-7916.

Appendix A: Realization of the Respiratory Cycle for the VHP-Female CAD Model

Note: all coordinates and transformations are reported in millimeter, all rotation are reported in degree.

1. Lung Deformation Sequence

New global coordinate system: Lung_CS. The origin is located at (0, max (Py), max (Pz)) with P being the point cloud of the lungs. The origin coordinates are given by

$$X = 0, Y = 122.8347, Z = -131.3727 \tag{1}$$

Scaling in Lung_CS over N ($N = 11$) **iterations total**:

(**Resulting Parametric Sweep in ANSYS**)

- 10% size increase in the z-direction: lung_scalez $= \left(1 + \frac{0.1}{N}\right)^t$

- 1% size increase in the x-direction: lung_scalex $= \left(1 + \frac{0.01}{N}\right)^t$

- 1% size increase in the y-direction: lung_scaley $= \left(1 + \frac{0.01}{N}\right)^t$

Variable t (sweeping variable, not time!) is running from 0 to N. This will result in the overall volume change from 2.22 L to 2.54 L, i.e. 0.32 L. Other sequences may be constructed in a similar fashion.

Translation in Lung_CS over N **iterations total** ($N = 11$):

Resulting Parametric Sweep in ANSYS

- 3 mm overall in the y-direction: lung_movey $= -\frac{3}{N} * t * 10^{-3}$ (m)

Rotation: None

2. Ribs Deformation Sequence

New global coordinate system: None

Scaling: None

Translation: None

Rotation: Every rib is rotated individually for a particular lung deformation so that there are no intersections between ribs and lungs given the ***minimum*** separation distance. Two rotation angles are used:

- Rotation about a rib axis, which is created by connecting two control points of two adjacent ribs closest to the vertebral column;

- Rotation about the z-axis, in a ***new local CS***, which is obtained by translation the origin of the global CS to the rib control point(s) (individually for every rib)

Control points: Closest points to the vertebral column.

Definition of rotation angles:

θ – Rotation angle about the rib axis, which results in an upward motion of the rib pair

φ – Rotation angle about the local z-axis, which results in an outward motion of the rib pair

Table A1. Ribs Deformation Sequence – Table of Computed Control Points (mm)

Coord. /Rib pair #	1	2	3	4	5	6	7	8	9	10	11	12
X	44.690	43.810	43.42	44.11	39.57	32.07	30.61	34.9	34.31	35.91	46.5	54.13
	-27.49	-19.62	-17.15	-18.25	-13.45	-4.1	-1.296	-0.242	-1.062	-2.352	-10.33	-21.09
Y	58.37	80.38	91.95	104.3	107.7	104.5	106.3	103.3	102.7	106.3	112.5	109.3
	61.14	76.38	93.32	100.7	106.7	102.6	106.3	106.9	102.9	103	108	115.1
Z	-117.2	-131.4	-149.7	-167.3	-187.8	-215.3	-233.8	-260.5	-281.3	-306.4	-334	-367.3
	-121.7	-134	-156.6	-170	-192.2	-217.8	-232.7	-257	-279.9	-304.8	-334	-365.4

Table A2. Ribs Deformation Sequence – Table of Rotation Angles (deg.) extracted from MATLAB

Iter. #/Rib pair #		1	2	3	4	5	6	7	8	9	10	11
$n_R = 1$	θ	0.1667	0.3333	0.6667	0.8333	1.1667	1.5000	2.3333	3.0000	3.6667	4.3333	4.8333
	φ	0.0800	0.1600	0.3200	0.4000	0.5600	0.7200	1.1200	1.4400	1.7600	2.0800	2.3200
$n_R = 2$	θ	0.6667	1.0000	1.3333	1.6667	2.0000	2.3333	2.6667	3.0000	3.3333	3.6667	4.0000
	φ	0.1600	0.2400	0.3200	0.4000	0.4800	0.5600	0.6400	0.7200	0.8000	0.8800	0.9600

$n_R = 3$	θ	1.5000	2.0000	2.5000	3.0000	3.5000	4.0000	4.5000	5.0000	5.5000	6.0000	6.5000
	φ	0.2400	0.3200	0.4000	0.4800	0.5600	0.6400	0.7200	0.8000	0.8800	0.9600	1.0400
$n_R = 4$	θ	0.6667	1.3333	2.0000	2.6667	3.3333	4.0000	4.6667	5.3333	6.0000	6.6667	7.3333
	φ	0.0800	0.1600	0.2400	0.3200	0.4000	0.4800	0.5600	0.6400	0.7200	0.8000	0.8800
$n_R = 5$	θ	0.4167	0.6250	0.8333	1.0417	1.4583	1.8750	2.0833	2.5000	2.9167	3.1250	3.3333
	φ	0.8000	1.2000	1.6000	2.0000	2.8000	3.6000	4.0000	4.8000	5.6000	6.0000	6.4000
$n_R = 6$	θ	0.2500	0.9583	1.6667	2.3750	3.0833	3.7917	4.5000	5.0000	5.2500	5.5000	5.7500
	φ	0.4000	1.5333	2.6667	3.8000	4.9333	6.0667	7.2000	8.0000	8.4000	8.8000	9.2000
$n_R = 7$	θ	0.5833	0.8750	1.1667	1.4583	1.7500	2.0417	2.3333	2.6250	2.9167	3.2083	3.5000
	φ	0.8000	1.2000	1.6000	2.0000	2.4000	2.8000	3.2000	3.6000	4.0000	4.4000	4.8000
$n_R = 8$	θ	0.6667	1.0000	1.3333	1.6667	2.0000	2.3333	2.6667	3.0000	3.3333	3.6667	4.0000
	φ	0.8000	1.2000	1.6000	2.0000	2.4000	2.8000	3.2000	3.6000	4.0000	4.4000	4.8000
$n_R = 9$	θ	0.3750	0.7500	1.1250	1.5000	1.8750	2.2500	2.6250	3.0000	3.3750	3.7500	4.1250
	φ	0.4000	0.8000	1.2000	1.6000	2.0000	2.4000	2.8000	3.2000	3.6000	4.0000	4.4000
$n_R = 10$	θ	0.4167	0.8333	1.2500	1.6667	2.0833	2.5000	2.9167	3.3333	3.7500	4.1667	4.5833
	φ	0.4000	0.8000	1.2000	1.6000	2.0000	2.4000	2.8000	3.2000	3.6000	4.0000	4.4000
$n_R = 11$	θ	0.2292	0.4583	0.6875	0.9167	1.1458	1.3750	1.6042	1.8333	2.0625	3.6667	6.1875
	φ	0.2000	0.4000	0.6000	0.8000	1.0000	1.2000	1.4000	1.6000	1.8000	3.2000	5.4000
$n_R = 12$	Θ	0.5000	1.0000	1.5000	2.0000	2.5000	3.0000	3.5000	4.0000	4.5000	5.0000	5.5000
	φ	0.4000	0.8000	1.2000	1.6000	2.0000	2.4000	2.8000	3.2000	3.6000	4.0000	4.4000

Table A3. Ribs Deformation Sequence – Polynomials of Rotation Angle –Resulting Parametric Sweep in ANSYS

Rib pair #		Polynomials of Rotation Angle (deg)
$n_R = 1$	θ	$5.901e-05*t^6 - 0.002019*t^5 + 0.02502*t^4 - 0.1346*t^3 + 0.3259*t^2 - 0.08703*t + 0.006159$
	φ	$0.0001035*t^6 - 0.00367*t^5 + 0.0489*t^4 - 0.3052*t^3 + 0.9405*t^2 - 1.114*t + 0.5939$
$n_R = 2$	θ	0.3333*t
	φ	0.08*t
$n_R = 3$	θ	0.5*t

	φ	0.08*t
$n_R = 4$	θ	0.6667 *t
	φ	0.08*t
$n_R = 5$	θ	$-1.844e - 05 * t^6 + 0.0004494 * t^5 - 0.003782 * t^4 + 0.01514 * t^3 - 0.03083 * t^2 + 0.4792 * t - 0.001689$
	φ	$-3.54e - 05 * t^6 + 0.0008629 * t^5 - 0.007261 * t^4 + 0.02908 * t^3 - 0.0592 * t^2 + 0.4488 * t - 0.003243$
$n_R = 6$	θ	0.45*t
	φ	0.4*t
$n_R = 7$	θ	0.2917*t
	φ	0.4*t
$n_R = 8$	θ	0.3333*t
	φ	0.4*t
$n_R = 9$	θ	0.375*t
	φ	0.4*t
$n_R = 10$	θ	0.4*t
	φ	0.4167*t
$n_R = 11$	θ	$2.986e - 05 * t^6 - 0.00111 * t^5 + 0.01608 * t^4 - 0.1145 * t^3 + 0.4126 * t^2 - 0.09056 * t - 0.004135$
	φ	$2.156e - 05 * t^6 - 0.000801 * t^5 + 0.01161 * t^4 - 0.08265 * t^3 + 0.2979 * t^2 + 0.01943 * t - 0.002985$
$n_R = 12$	Θ	0.25*t
	φ	$0.5 * t$

3. Sternum Deformation Sequence

New global coordinate system: Sternum_CS. The origin is located at ((3*(min (Px)+max (Px)/5), max (Py), (3*max (Pz)/5)) with P being the point cloud of the sternum. The origin coordinates are given by

$X = 20.78, Y = -28.86, Z = -290.3$

Scaling: None

Translation: None

Rotation in Sternum_CS: one degree about the new global y-axis **over N ($N = 11$) iterations total**.

Resulting Parametric Sweep in ANSYS

$$\text{sternum\_rot} = 0.09091 * t$$

4. Cartilage Deformation Sequence (implemented in MATLAB)

New global coordinate system: None

Scaling: Two movement vectors are determined for every cartilage component at each iteration which will decide its scaling factor as follows:

$$New\ Movement\ Vector = \vec{m}_n$$

$$Old\ Movement\ Vector = \vec{m}_o$$

$$Scaling\ Factor = \frac{|m_n|}{|m_0|}$$

$$Scaling\ Vector = \frac{-m_n}{|m_n|}$$

Translation: A translation vector determines the movement of the cartilage for every iteration, given by:

$$Translation\ Vector = \vec{m}_n - \vec{m}_0$$

Rotation: The rotation axis and the rotation degree is given by:

$$Rotation\ Vector = \cos^{-1}(\vec{m}_n \cdot \vec{m}_0)/(|\vec{m}_n| \times |\vec{m}_0|)$$

$$Rotation\ Axis = \vec{m}_o \times \vec{m}_n$$

Table A4. Cartilage Deformation Sequence – Table of Rotation Angles (radians)

#Iteration no./#cartilage pair	1	2	3	4	5	6	7	8	9	10	11
Left1	0	0	0.01445	0.01388	0.01334	0.03706	0.03272	0.07020	0.01471	0.00713	0.00692
Right1	0	0	0.01445	0.01388	0.01334	0.03706	0.03272	0.07020	0.01471	0.00713	0.00692
Left2	0	0.02217	0.02194	0.02169	0.02143	0.02114	0.02084	0.02053	0.02021	0.01988	0.01954
Right2	0	0.02287	0.02281	0.02272	0.02260	0.02246	0.02230	0.02211	0.02190	0.02167	0.02142
Left3	0	0.03000	0.02975	0.02945	0.02910	0.02870	0.02826	0.02779	0.02728	0.02674	0.02618
Right3	0	0.02900	0.02858	0.02813	0.02763	0.02710	0.02655	0.02597	0.02537	0.02476	0.02413

Left4	0	0.03671	0.03598	0.03517	0.03430	0.03338	0.03242	0.03142	0.03041	0.02939	0.02837
Right4	0	0.03461	0.03470	0.03470	0.03461	0.03443	0.03416	0.03381	0.03338	0.03288	0.03231
Left5	0	0.01392	0.01361	0.01331	0.02113	0.02019	0.01195	0.01883	0.01801	0.01073	0.01049
Right5	0	0.01605	0.01575	0.01544	0.02521	0.02417	0.01399	0.02263	0.02167	0.01262	0.01234
Left6	0	0.00617	0.00610	0.00603	0.00596	0.00590	0.00583	0.00577	0.00571	0.00565	0.00559
Right6	0	0.01445	0.01388	0.01334	0.03706	0.03272	0.07020	0.01471	0.00713	0.00692	0.00672

Table A5. Cartilage Deformation Sequence – Table of Expansion Factor (only the difference)

#Iteration no./#cartilage pair	1	2	3	4	5	6	7	8	9	10	11
Left1	0	0.002864	0.004915	0.003289	0.005533	0.005889	0.013425	0.012064	0.012891	0.013652	0.011283
Right1	0	0.003292	0.005705	0.003689	0.006288	0.006622	0.015209	0.013425	0.014189	0.014884	0.012137
Left2	0	0.005041	0.005525	0.005993	0.006444	0.006876	0.007290	0.007685	0.008061	0.008417	0.008753
Right2	0	0.001314	0.001875	0.002430	0.002978	0.003518	0.004046	0.004562	0.005064	0.005552	0.006023
Left3	0	0.003696	0.004616	0.005512	0.006378	0.007211	0.008008	0.008765	0.009481	0.010153	0.010781
Right3	0	0.006657	0.007496	0.008297	0.009057	0.009774	0.010446	0.011073	0.011653	0.012188	0.012677
Left4	0	0.009183	0.010484	0.011701	0.012829	0.013864	0.014804	0.015651	0.016405	0.017069	0.017648
Right4	0	0.001770	0.000510	0.000766	0.002033	0.003287	0.004517	0.005714	0.006871	0.007980	0.009035
Left5	0	0.010654	0.010767	0.010868	0.023283	0.023299	0.011074	0.023284	0.023185	0.011103	0.011113
Right5	0	0.009079	0.009300	0.009506	0.020776	0.021132	0.010164	0.021585	0.021758	0.010557	0.010633
Left6	0	0.007867	0.007894	0.007918	0.007940	0.007959	0.007977	0.007992	0.008006	0.008017	0.008027
Right6	0	0.019481	0.019443	0.019386	0.062856	0.061347	0.165538	0.035730	0.016683	0.016521	0.016358

5. Muscle Deformation Sequence (Pectoralis Major, Pectoralis Minor, Abdominal Muscles, and Erector Spinae)

New local coordinate systems: A local coordinate system is defined for each muscle using a simple translation. The origins are located at (min(Px), max (Py), max (Pz)) with P being the point cloud of each left muscle and (max(Px), max (Py), max (Pz)) of each right muscle. Abdominal muscles are only transformed with respect to the global coordinate system: the origin at (0, 0, 0)

Table A6. Origin coordinates for the local coordinate systems.

Muscle	Local X (mm)	Local Y (mm)	Local Z (mm)
Pectoralis Minor Left	89.03	27.64	-201.93
Pectoralis Minor Right	-56.28	29.22	-189.24
Pectoralis Major Left	18.83	76.19	-192.42
Pectoralis Major Right	14.82	-31.92	-197.71
Erector Spinae Left	0	146.73	-450.22
Erector Spinae Right	0	146.79	-453.65

Scaling in local CSs: See the following tables for individual muscles

Translation in local CSs: See the following tables for individual muscles

Rotation in local CSs: See the following tables for individual muscles

Table A7. Deformation factors for Pectoralis Minor Left Muscle. All angles are recorded in degrees

Configuration number	Rotation about Z axis (θ)	Movement in Y direction	Scaling in Y direction
1	0.9545	-2.7273	-0.0027
2	1.2727	-3.6364	-0.0036
3	1.9091	-5.4545	-0.0055
4	2.2273	-6.3636	-0.0064
5	2.8636	-8.1818	-0.0082
6	3.1818	-9.0909	-0.0091
7	3.8182	-10.9091	-0.0109
8	4.1364	-11.8182	-0.0118
9	4.4545	-12.7273	-0.0127
10	5.0909	-14.5455	-0.0145
11	5.4091	-15.4545	-0.0155

Table A9. Deformation factors for Pectoralis Minor Right Muscle. All angles are recorded in degrees

Configuration number	Rotation about Z axis (θ)	Movement in Y direction	Scaling in Y direction
1	1.2727	-3.6364	-0.0036
2	1.5909	-4.5455	-0.0045
3	1.9091	-5.4545	-0.0055
4	2.2273	-6.3636	-0.0064
5	2.5455	-7.2727	-0.0073
6	3.1818	-9.0909	-0.0091
7	3.8182	-10.9091	-0.0109
8	4.4545	-12.7273	-0.0127
9	5.0909	-14.5455	-0.0145
10	5.4091	-15.4545	-0.0155
11	6.0455	-17.2727	-0.0173

Table A10. Deformation factors for Pectoralis Major Left Muscle. All angles are recorded in degrees

Configuration number	Movement in Z direction	Movement in Y direction	Scaling in Y direction
1	1.3636	-2.7273	-0.0027
2	1.8182	-3.6364	-0.0036
3	2.2727	-4.5455	-0.0045
4	2.7273	-5.4545	-0.0055
5	3.6364	-7.2727	-0.0073
6	4.0909	-8.1818	-0.0082
7	5.0000	-10.0000	-0.0100
8	5.4545	-10.9091	-0.0109
9	5.9091	-11.8182	-0.0118

10	6.3636	-12.7273	-0.0127
11	7.2727	-14.5455	-0.0145

Table A11. Deformation factors for Pectoralis Major Right Muscle. All angles are recorded in degrees

Configuration number	Movement in Z direction	Movement in Y direction	Scaling in Y direction
1	1.3636	-2.7273	-0.0027
2	1.8182	-3.6364	-0.0036
3	2.2727	-4.5455	-0.0045
4	2.7273	-5.4545	-0.0055
5	3.1818	-6.3636	-0.0064
6	3.6364	-7.2727	-0.0073
7	4.0909	-8.1818	-0.0082
8	5.0000	-10.0000	-0.0100
9	5.4545	-10.9091	-0.0109
10	5.9091	-11.8182	-0.0118
11	6.8182	-13.6364	-0.0136

Table A12. Deformation factors for Erector Spinea Left Muscles. All angles are recorded in degrees.

Configuration number	Scaling in Y direction	Scaling in X direction	Movement in Y direction
1	-0.1600	-0.0800	0.2000
2	-0.2000	-0.1000	0.2500
3	-0.2400	-0.1200	0.3000
4	-0.2800	-0.1400	0.3500
5	-0.3200	-0.1600	0.4000
6	-0.3600	-0.1800	0.4500

7	-0.4000	-0.2000	0.5000
8	-0.4400	-0.2200	0.5500
9	-0.4800	-0.2400	0.6000
10	-0.5200	-0.2600	0.6500
11	-0.5600	-0.2800	0.7000

Table A13. Deformation factors for Erector Spinea Right Muscles. All angles are recorded in degrees.

Configuration number	Scaling in Y direction	Scaling in X direction	Movement in Y direction
1	-0.0200	-0.0200	0.0500
2	-0.0400	-0.0400	0.1000
3	-0.0600	-0.0600	0.1500
4	-0.0800	-0.0800	0.2000
5	-0.1000	-0.1000	0.2500
6	-0.1200	-0.1200	0.3000
7	-0.1400	-0.1400	0.3500
8	-0.1600	-0.1600	0.4000
9	-0.1800	-0.1800	0.4500
10	-0.2000	-0.2000	0.5000
11	-0.2200	-0.2200	0.5500

Table A14. Deformation factors for Abdominal Muscles in the global coordinate system (mm)

Configuration number	Movement in Z direction	Movement in Y direction
1	-0.0909	-0.0909
2	-0.1818	-0.1818
3	-0.2727	-0.2727

4	-0.3636	-0.3636
5	-0.4545	-0.4545
6	-0.5455	-0.5454
7	-0.6364	-0.6363
8	-0.7273	-0.7272
9	-0.8182	-0.8181
10	-0.9091	-0.9090
11	-1.0000	-1.0000

Table A15. Muscles Deform – Polynomials of Deformation Factors

Parametric sweep in ANSYS

Muscles			Polynomials of Deformation Factors (Angles are recorded in deg)
Pectoralis minor (in local CS)	Left	Rot z	$-7.149e-5*t^6 + 0.00252*t^5 - 0.03393*t^4 + 0.2181*t^3 - 0.681*t^2 + 1.406*t + 0.005579$
		Move y	$0.0002042*t^6 - 0.007194*t^5 + 0.09695*t^4 - 0.6231*t^3 + 1.946*t^2 - 4.016*t - 0.01594$
		Scale y	$2.042e-7*t^6 - 7.194e-6*t^5 + 9.695e-5*t^4 - 0.0006*t^3 + 0.001946*t^2 - 0.004*t + 0.999984$
	Right	Rot z	$1.083e-05*t^6 + 8.348e-05*t^5 - 0.00957*t^4 + 0.1344*t^3 - 0.7021*t^2 + 1.774*t + 0.01398$
		Move y	$-3.095e-5*t^6 - 0.0002385*t^5 + 0.02734*t^4 - 0.3841*t^3 + 2.006*t^2 - 5.067*t - 0.03994$
		Scale y	$3.09e-8*t^6 - 2.385e-7*t^5 + 2.73e-5*t^4 - 0.00038*t^3 + 0.002*t^2 - 0.005067*t + 0.99996$
Pectoralis major (in local CS)	Left	Move z	$-0.00013*t^6 + 0.004709*t^5 - 0.0656*t^4 + 0.4366*t^3 - 1.388*t^2 + 2.397*t - 0.005142$
		Move y	$-0.05 + 0.0002525*t^6 - 0.00914*t^5 + 0.1272*t^4 - 0.8445*t^3 + 2.673*t^2 - 4.721*t + 0.01132$
		Scale y	$2.6e-7*t^6 - 9.418e-6*t^5 + 0.0001312*t^4 - 0.00087*t^3 + 0.0028*t^2 - 0.0048*t + 1.00001028$
	Right	Move	$-4.642e-05*t^6 + 0.001553*t^5 - 0.02174*t^4 + 0.1655*t^3 - 0.6858*t^2$

		z	$+1.801 * t + 0.02082$
		Move y	$9.284e-05*t^6 - 0.003107*t^5 + 0.04348*t^4 - 0.331*t^3 + 1.372*t^2$ $-3.602*t - 0.04165$
		Scale y	$9.284e-8*t^6 - 3.107e-6*t^5 + 4.348e-5*t^4 - 0.00033*t^3 + 0.00137*t^2$ $-0.0036*t + 0.99995835$
Erector Spinae (in local CS)	Left	Scale y	$8.987e-06*t^6 - 0.0003339*t^5 + 0.004839*t^4 - 0.03445*t^3 + 0.1242*t^2$ $-0.2474*t + 0.998756$
		Scale x	$4.493e-06*t^6 - 0.0001669*t^5 + 0.00242*t^4 - 0.01723*t^3 + 0.06208*t^2$ $-0.1237*t + 0.9993778$
		Move y	$-1.123e-05*t^6 - 0.0004174*t^5 + 0.006049*t^4 - 0.04307*t^3 + 0.1552*t^2$ $-0.3093*t - 0.3093$
	Right	Scale y	$-0.02*t + 1$
		Scale x	$-0.02*t + 1$
		Move y	$-0.05*t$
Abdominal (in global CS)	Move z		$0.09091*t*10^{-3}$
	Move y		$-0.35*t*10^{-3}$

6. Heart Deformation Sequence

New local coordinate systems: According to literature, the pumping motion of the heart is independent of breathing. As a result, the heart object will only be transformed to avoid intersection with lungs in breathing sequence, with respect to the origin of the global coordinate system (0, 0, 0).

Table A16. Deformation factors for heart.

Configuration number	Movement in Z direction	Movement in Y direction
1	-0.15	-0.05
2	-0.3	-0.1
3	-0.45	-0.15
4	-1.5	-0.5
5	-2.85	-0.95

6	-4.35	-1.45
7	-5.7	-1.9
8	-7.2	-2.4
9	-8.7	-2.9
10	-10.05	-3.35
11	-11.55	-3.85

Table A17. Parametric sweep in ANSYS

Heart		Polynomials of Deformation Factors
Move	Z	$-6.672e-06*t^6 + 0.0008203*t^5 - 0.02038*t^4 + 0.2019*t^3 - 0.917*t^2 + 0.7346*t - 0.03539$
	Y	$2.451e-06*t^6 - 8.201e-05*t^5 + 0.0009*t^4 - 0.0015*t^3 - 0.04611*t^2 + 0.03447*t - 0.0046$

7. Liver Deformation Sequence

New local coordinate systems: The liver object is deformed to avoid intersection with lungs in breathing sequence, with respect to the origin of a local coordinate system: (0, max (Py), max (Pz))

Table A18. Local coordinate system.

	Local X (mm)	Local Y (mm)	Local Z (mm)
Liver	0	120.136	-373.331

Table A19. Deformation factors for liver

Configuration number	Movement in Z direction	Movement in Y direction	Scale in Z direction
1	-0.18	-0.04	-0.001
2	-0.36	-0.8	-0.002
3	-1.44	-0.32	-0.008
4	-2.52	-0.56	-0.014
5	-3.6	-0.8	-0.020
6	-5.04	-1.12	-0.028

7	-6.48	-1.44	-0.036
8	-7.92	-1.76	-0.044
9	-9.36	-2.08	-0.052
10	-10.8	-2.4	-0.060
11	-12.24	-2.72	-0.068

Table A20. Parametric sweep in ANSYS

Liver		Polynomials of Deformation Factors
Move	Z	$-6.672\mathrm{e}-06*t^6 + 0.0008203*t^5 - 0.02038*t^4 + 0.2019*t^3 - 0.917*t^2 + 0.7346*t - 0.03539$
	Y	$2.451e-06*t^6 - 8.2e-05*t^5 + 0.0009106*t^4 - 0.001488*t^3 - 0.0461*t^2 + 0.0344*t - 0.004638$
Scale	Z	$6.1e-8*t^6 - 2.05e-6*t^5 + 2.7e-5*t^4 - 3.7e-5*t^3 - 0.00115*t^2 + 0.00086*t + 0.99989$

8. Skin Shell Deformation

First, the skin shell deformation starts with a generation of N chest objects for each step via non-rigid transformations. This process was accomplished in MATLAB. A deformed chest region is defined as

$$141.3mm < P(:,1) < 173.4mm$$

$$P(:,2) < 64mm$$

$$-330.6mm < P(:,3) < -150.6mm$$

All nodes in the chest region of the skin shell are selected and transformed in the y-direction using the following equation:

$$P(:,2) = Pbase(:,2) - \frac{10}{N} * t * sin(P(:,3) - min(P(:,3)))$$

We chose nodes in the chest region so that $P(:,3) - min(P(:,3))$ goes from 180 to 0. Therefore, nodes that are closer to the upper and lower boundaries of the region will move less than the nodes that are closer to the center. With maximum inhalation, the center node of the chest region will move by 10mm in the Y direction. Thus, only coordinates of nodes belonging to the chest area are changed. Also, the connectivity matrix, t, of the entire skin shell still remains the same. As a result, 11 skin shell objects with different chest regions will be generated.

Second, these new skin shells are subtracted from the original skin shell in HFSS, which results in N smaller deformed chest objects. These chest objects are spaced evenly (400mm in Y direction) in front of the original shell and then united. A moving box is carefully designed so that it covers only one chest object at any time instant t. Then, the intersection is performed. The process is illustrated in Fig A1.

Box original position is given by: -300mm $,(200 - t * 400) * 10^{-3}, -350$mm
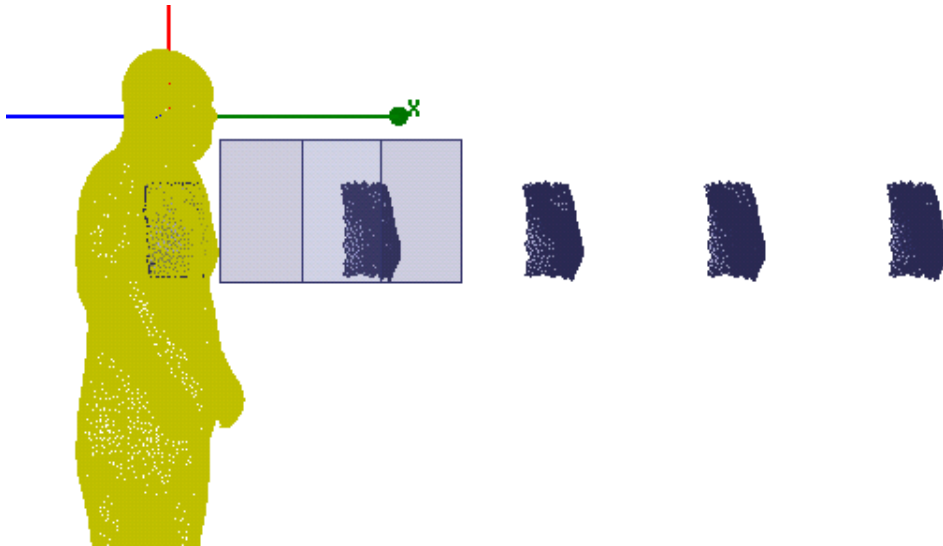
Fig A1. A box is carefully designed so that each iteration covers only one chest object at a time.

An intersection operation is performed with the box and the chest array object, which results in one chest object for a particular time t. Finally, the chest object is moved and a unite operation is performed with the original skin shell (shown in Fig A2).

| Fig A2a. Skin Shell with $t = 0$ | Fig A2b. Skin Shell with $t = 11$ |