Proceedings of the 29th Annual International
Conference of the IEEE EMBS
Cité Internationale, Lyon, France
August 23-26, 2007.

Evaluation of different meshing algorithms in the computation of defibrillation thresholds in children

Jeroen G. Stinstra, Matthew Jolley, Michael Callahan, David Weinstein, Martin Cole, Dana H. Brooks, John Triedman, and Rob S. MacLeod

Abstract— **In this paper we evaluate different meshing schemes to solve for the bioelectric fields that arise in the human body due to the defibrillation shock generated by an Implantable Cardiac Defibrillator, with particular emphasis on implantation in children. For children, the question of relative performance of different electrode locations remains open. Computational simulation is a critical tool to address this question, and mesh design is a critical component of such simulations. We use the SCIRun software package to address this simulation problem because it combines the powerful numeric tools required with interactive flexibility allowing easy comparison of both algorithms and electrode orientation. We describe a pipeline that starts with segmented CT-images and produces clinically useful parameters. Using this framework we report below that a meshing scheme using regularly spaced hexahedral elements which are locally refined around the electrodes constitute a quick and relatively accurate way of solving this problem.**

I. Introduction

The computation of bioelectric fields in the human body commonly relies on the use of a finite element calculation. A common approach is to segment an MR or CT scan into several compartments that each is assumed to have its own (constant) electrical conductivity. With the resolution of such images increasing steadily, the question arises as to what is the best strategy to turn these images into computational meshes for different types of biomedical problems. In particular the best choice of meshing scheme depends on the application and the efficiency of the resulting computation, as the goal is to optimize for the full computational pipeline, *i.e.* from image to clinical parameters. For instance, if the creation of a mesh requires far more time than the actual computation one wants to perform, it might be more beneficial to choose a faster and less efficient scheme for the creation of the mesh and allow for more time to be spent in solving the equations describing the model.

In this paper we describe our use of the open source software framework SCIRun [1], which we are developing to quickly evaluate different modeling schemes before writing targeted applications for specific simulations. The SCIRun framework is a visual programming environment targeted at solving bioelectric problems (among others) and visualizing the results. The advantage of this software framework is that one can quickly evaluate different schemes by connecting alternative modules with a visual interface and that one can immediately visualize the ensuing simulation results as the computational and visual infrastructure are integrated into one software framework. An example of this visual interface is shown in Fig. 1.

Our interest here is in the computation of defibrillation thresholds in children targeted at the evaluation of the efficiency of different implantation electrode locations for Implantable Cardiac Defibrillators (ICDs) [2]. The implantation of ICDs in children is complicated by the fact that one does not always know what the optimal positions are for the defibrillation electrodes and that the anatomical spaces for implantation is often quite limited in children. As a result clinicians may end up implanting the ICDs in non-standard locations. To allow prediction of good locations for the electrodes we are developing a computer model that will predict defibrillation thresholds based on interactively placing and moving the ICD electrodes inside a virtual model of the torso, which is based on a MRI or CT scans. The efficiency of the defibrillation in the this model is judged on the local strength of the induced electric field. The latter is a common way of evaluating defibrillation effectiveness [3,4].

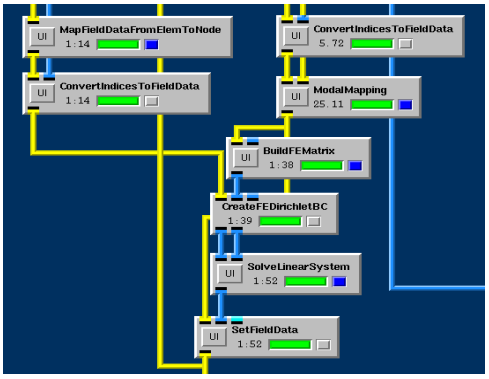


Fig. 1. Example of SCIRun software displaying various modules that are connected visually with dataflow pipelines. The segment shown here is part of the pipeline that transforms segmented data into a clinical parameters.

Manuscript received May 14, 2007. This work was supported in part by the NIH Grant 5P41RR012553-09.

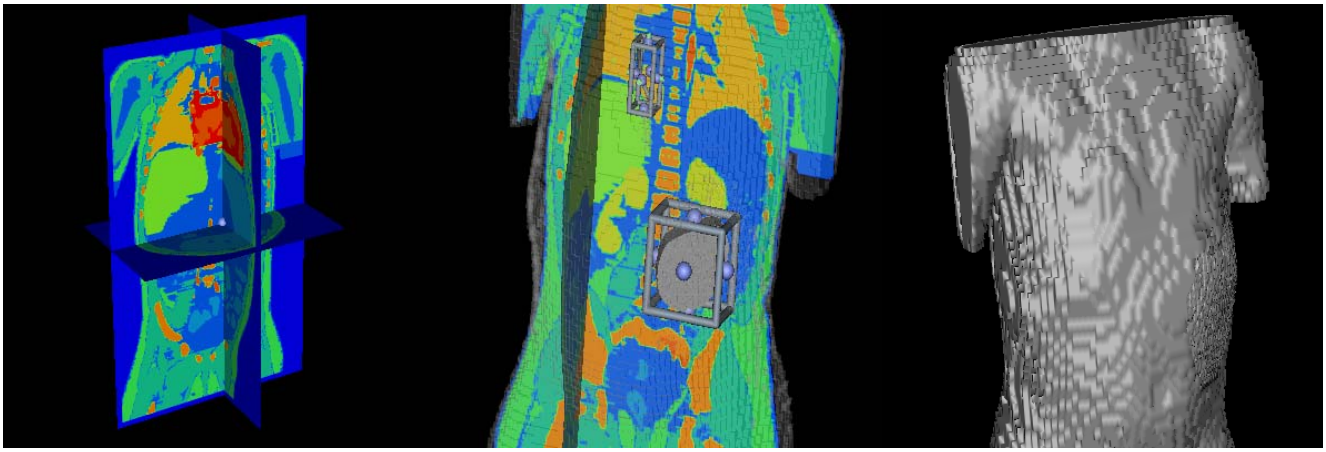
Jeroen G. Stinstra is with the Scientific Computing and Imaging Institute and the CardioVascular Research and Training Institute, University of Utah, Salt Lake City, UT, USA (phone: 801-587-9508 email: jeroen@cvrti.utah.edu)

Matthew Jolley and John Triedman are with the Department of Cardiology, Children's Hospital Boston, Boston, MA, USA

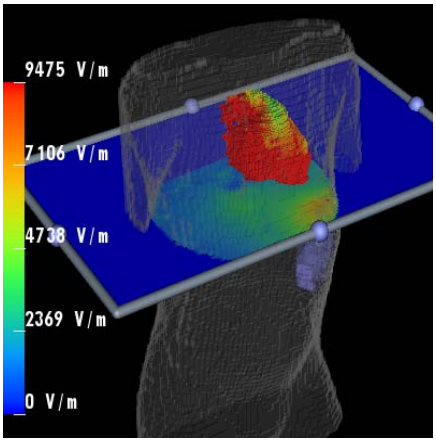
Michael Callahan, Martin Cole, and Dave Weinstein are with the Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, USA

Dana H. Brooks is with Department of Electrical Engineering, Northeastern University, Boston, MA, USA

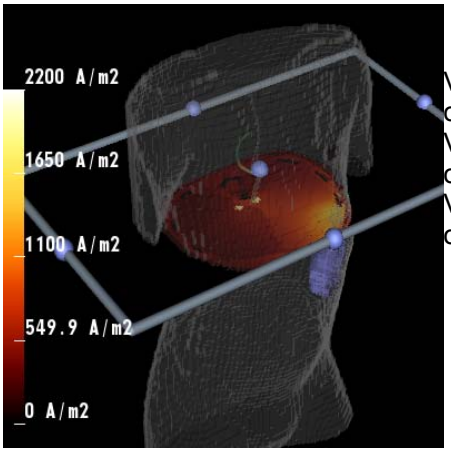
Rob S. MacLeod is with Scientific Computing and Imaging Institute, the CardioVascular Research and Training Institute, and the Department of Bioengineering, University of Utah, Salt Lake City, UT, USA



[A] Segmentation of CT-scan [B] Placement of Electrodes [C] Computational mesh



[D] Electric Field



[E] Current Density

Voltage needed for 90% of volume over 3 V/cm : 100.77V
Voltage needed for 95% of volume over 3 V/cm : 118.39V
Voltage needed for 98% of volume over 3 V/cm : 139.33V

[F] Data analysis

Fig. 2. Creation of a model for the evaluation of defibrillation thresholds. Figure (A) show the segmentation on which the model is based, in (B) electrodes are inserted into the model, in (C) a mesh is created with a refinement on the right, (D) and (E) show the results of the computation projected

II. Methods

The computer model used for evaluating ICD configurations was based on a segmentation of a CT scan of a torso of a 2 year old child. This torso was segmented into different tissue types with different electrical conductivities: heart (0.25S/m), blood (0.7S/m), bone (0.006 S/m), lungs (0.067 S/m), kidney (0.07 S/m), liver (0.15 S/m), muscle (0.25 S/m) and connective tissue (0.22 S/m). These values are based on average values from literature [3,4,5,6].

The segmentation of the torso was visualized using the SCIRun framework, which served as well as a platform for inserting virtual electrodes. In order to insert the electrodes into the model we used visual widgets that could be used to insert models of the ICD can as well as to shape the models of the wire electrodes by visually bending the electrodes. In order to test for a range of clinically relevant configurations, a clinician created a series of clinically relevant locations using this visual interface. These configurations were then used to test the efficiency of different meshing schemes.

The locations of the electrodes and the segmentations were created in the same Cartesian coordinate system and formed the basis for several different meshing schemes: **(1)** The first scheme used a mesh of regularly spaced hexahedral elements that filled the bounding box of the full CT scan. In order to define the conductivity we used a set of 27 regularly spaced sampling points in each element to derive the tissue type from the underlying segmentation and from that we derived the average conductivity of the element. Next we used the same sampling points to check if any of the points was located in one of the electrodes. If so the nodes that bounded the element were set to a preset voltage, the latter tracing served as boundary condition of the finite element approach. **(2)** In the second approach we used the regular grid of the first approach as a start, but to increase accuracy we used a rectangular solid bounding box around the electrodes and refined the elements inside into this box into 27 smaller elements according to a subdivision scheme with three elements in each spatial dimension [7]. **(3)** In a third approach we used the same mesh as the second approach but also refined the elements surrounding the heart. **(4)** In the fourth approach we used a refinement scheme similar to the one in the second approach. However instead of a bounding box, instead we located all the elements that touched the

electrodes and, then dilated this selection to include a region spanning at least 5 neighboring elements. We used a similar scheme in which elements were cut in four per direction using a similar scheme as discussed in [7], but now applied consecutive per direction. As the volume that was refined was not convex the meshing procedure resulted in less regularly shaped elements in the center of the refinement as elements were diced per direction. **(5)** In a fifth approach we not only refined locally around the electrodes but as well around the heart. **(6)** In a sixth approach we tested a two stage approach: first we used the results of the fourth approach to estimate the local gradient of the field, and then we used that result to determine which regions needed to be refined. Similarly to the fourth approach, we selected all elements that had a gradient over a certain threshold and then dilated the area to include the same size neighborhood as in the fourth approach.

In order to be able to compare these methods we generated a solution that was based on the first scheme but was refined in each direction three times, to obtain an underlying high density model. We used each of these seven meshes (the high density mesh and the six described above) to compute the potential throughout the body using a conjugate gradient method to solve the matrix equations. We assumed a boundary condition in which the outside of the can electrode had a potential of 0V and the wire electrode had a nominal potential value of 1V. Based on this computation we derived the strength of the electrical field and evaluated how much the electrical potential difference needed to be scaled to ensure that 95% percent of the cardiac tissue was over a threshold of 3V/cm, which literature suggests is a good defibrillation threshold [3]. In order to derive the volume of tissue over the threshold of 3 V/cm we used a numerical integration that used a sampling scheme of 150 by 150 by 150 sample points in a bounding box around the heart. If the sample point was inside the heart the local potential gradient was evaluated and was added to a histogram of field strengths. Next we estimated how much the histogram needed to be scaled in order to have 95% of tissue over the 3V/cm threshold. The latter was considered to be a good setting for the ICD for that location. Everything except the segmentation of the CT itself was computed within the SCIRun framework, as illustrated in Fig. 2. The segmentation of the CT image was done semi-manually using the Slicer software package [8].

III. RESULTS

The results are summarized in Table 1. We evaluate four metrics: the potential difference needed between the can of the ICD and the wire electrode to have more than 90% of the cardiac tissue over a threshold of 3 V/cm, the potential difference needed to have more than 95% of cardiac tissue over this threshold, the number of elements that were in the resulting mesh and the time needed to compute the results from loading the segmentation until the evaluation of suitable potentials between the electrodes. All the results are shown as a comparison against a non-refined model that consisted out of a grid of 225 by 225 by 225 elements. In order to evaluate the meshing methods we computed the maximum relative error for the two threshold settings for four different electrode locations, where error is measured as the change from full 225x225x225 mesh. The number of elements and the time needed to compute the results is given in relative measures where again 100% is the number of elements or the time needed for the model with the 225x225x225 grid. The reference model itself consisted of 11.3 million elements and it took 20 minutes to setup and solve the equations on an 8-core 2GHz AMD Opteron machine with 16Gb of RAM. We note that of course computation times cannot be taken as absolute measures of achievable results, but we took pains to both program efficiently and equitably across methods so that we believe the relative times we report are valid indications of the resulting complexity.

The results show that refining the coarse model around the electrodes will lead to smaller differences from the finest model compared to the coarse model. All refinement methods show that refining the mesh around the electrodes results in smaller errors while using less time and fewer elements than the fine model. However refining around the heart itself makes a negligible difference. The exceptions to requiring less time are the adaptive refinement methods (the last two methods in table 1), as they take more time in case of the lower threshold which is needed to get to a small relative error. Similarly, refining around the heart does not increase the accuracy of the model and just adds to the

Table 1. Comparison of different meshing methods against the case of a fine completely regular grid.

Meshing Method	**Max relative error in $V_{90\%}$**	**Max relative error in $V_{95\%}$**	**Number of elements**	**Relative time**
No-refinement, grid 75x75x75	14%	16%	4%	5%
No-refinement, grid 150x150x150	5%	5%	29%	30%
Refinement in box around electrodes, grid 75x75x75	2%	3%	17%	65%
Refinement in box around electrodes and heart, grid 75x75x75	2%	2%	19%	162%
Refinement dilated volume around electrodes, grid 75x75x75	2%	3%	3%	8%
Refinement dilated volume around electrodes and heart, grid 75x75x75	2%	2%	10%	40%
Adaptive refinement dilated volume with threshold of 100V/cm	3%	4%	8%	119%
Adaptive refinement dilated volume with threshold of 150V/cm	6%	6%	8%	27%

amount of time needed to compute the elements.

IV. DISCUSSION

One of the major technical limitations in our comparison is that in our software all the numerical algebra is implemented in parallel and that the refinement code is mostly still single threaded. Implemented meshing schemes in parallel is far more complicated than implementing the numerical algebra in parallel. The results shown were computed on a 8 core machine, which is currently state of the art. However multi-core architectures are rapidly becoming more common among desktop machines as well.

A second factor is that for a regular grid one can derive the finite element stiffness matrix with less computation as all the geometric scaling factors in the method are the same. Similarly one does not need to compute tables for specifying the connectivity between elements and hence the refinement schemes are at a computational disadvantage as the meshes are less structured and more bookkeeping overhead is needed for the computation. Hence the benefit gained by the refinement needs to outweigh the additional complexity in dealing with more complex meshes. The results presented here indicate that this is indeed the case for a simple refinement by dilating the volume around the electrodes. In this case the total computational time was no more than double that of the coarse mesh, but we approached the accuracy of the fine mesh, and indeed out performed the intermediate 150 by 150 by 150 mesh.

Besides schemes relying on hexahedral elements, schemes using tetrahedral elements are also a viable option as well. The current SCIRun infrastructure allows for the creation of both kinds of meshes. However our algorithms behind meshing volumes using tetrahedral elements have not yet been optimized for parallel computation, and thus currently require more time than the brute force approach of dividing the volume into very small hexahedral elements. Hence tetrahedral approaches were not taken into account in this study.

The advantage of having all the modeling pieces in one framework allows us to optimize computation for the whole pipeline rather then estimating efficiency of the meshing method based on element quality only.

As the eventual goal of this project is to create a computational infrastructure that is interactive in the sense that when the physician places electrodes in the virtual torso the software almost instantly returns an analysis of defibrillation efficacy, our computation time of about 20 minutes for the fully refined model is still too long. However as the results presented here show, the locally refined mesh takes only a few minutes to evaluate, far closer to the goal of having a fully interactive model than a uniform dense model is.

REFERENCES

[1] Scientific Computing and Imaging Institute. (2007, 3). SCIRun: A Scientific Computing Problem Solving Environment. Available: http://software.sci.utah.edu/scirun.html

[2] Jolley, M., Stinstra, J.G., Weinstein, D., Pieper, S., Estepar, R.S.J, Kindlmann, G., MacLeod, R.S., Brooks, D.H, and, Triedman, J.K., "Open-Source Toolkit for Interactive Finite Element Modeling of Optimal ICD Electrode Placement", in *Proc. Fourth International Conference on Functional Imaging and Modeling of the Heart*, June, 2007.

[3] Aguel, F., Eason, J. C., Trayanova, N. A., Siekas, G., and Fishler, M. G,. "Impact of Transvenous Lead Position on Active-can ICD Defibrillation: A Computer Simulation Study," PACE, vol. 22, pp. 158-164, 1999.

[4] de Jongh, A. L., Entcheva, E. G., Replogle, J. A., Booker, R.S., Kenknight, B.H, Claydon, F.J., "Defibrillation Efficacy of Different Electrode Placements in a Human Thorax Model," PACE, vol. 22, pp. 152-157, 1999.

[5] Geddes, L. A., and, Baker, L. E., "The Specific Resistance of Biological Material--a Compendium of Data for the Biomedical Engineer and Physiologist", Med. Biol. Engin., vol. 5, pp. 271-193, 1967.

[6] Jorgenson, D. B., Haynor, D. R., Bardy, G. H., and Kim, Y., "Computational Studies of Transthoracic and Transvenous Defibrillation in a Detailed 3-D Human Thorax Model," IEEE Trans. Biol. Engin., vol. 42, pp. 172-184, 1995.

[7] Harris, N.J., Benzley, S.E., and, Owen, S.J., "Conformal Refinement of All-Hexahedral Elements meshes based on multiple plane insertion," *in Proc. 13th International Meshing Roundtable, Williamsburg, VA, Sandia National Laboratories,* SAND #2004-3765C, pp. 157-168, September 19-22, 2004

[8] 3D Slicer. Available: http://www.slicer.org/#Downloading