

Finite Element Method Computations of the Acoustics of the Human Head Based on the Projection Based Interpolation Data

Marcin Sieniek, Piotr Gurgul, Maciej Paszyński

Department of Computer Science, AGH University of Science and Technology, Kraków, Poland
Email: maciej.paszynski@agh.edu.pl

Received August 2014

Abstract

In this paper we present the Projection Based Interpolation (PBI) technique for construction of continuous approximation of MRI scan data of the human head. We utilize the result of the PBI algorithm to perform three dimensional (3D) Finite Element Method (FEM) simulations of the acoustics of the human head. The computational problem is a multi-physics problem modeled as acoustics coupled with linear elasticity. The computational grid contains tetrahedral finite elements with the number of equations and polynomial orders of approximation varying locally on finite element edges, faces, and interiors. We utilize our own out-of-core parallel direct solver for the solution of this multi-physics problem. The solver minimizes the memory usage by dumping out all local systems from all nodes of the entire elimination tree during the elimination phase.

Keywords

Finite Element Method, Projection Based Interpolation, Acoustics of the Human Head

1. Introduction

In this paper we present the projection based interpolation technique [1] for construction of continuous approximation of material data representing MRI scan of the human head. The resulting 3D grid is then subject to three dimensional finite element method simulation of propagation of the acoustics waves over the human head, using the model proposed by [2]. We utilize our own out-of-core parallel direct solver for efficient solution of the resulting multi-physics problem [3].

2. Linear Elasticity Coupled with Acoustics

We focus here on numerical simulations of the acoustics of the human head, modeled as concentric spheres with a hole drilled with internal ear model inserted. The tissue, skull and some parts of the internal ear are modeled as linear elasticity, while air and parts of the internal ear filled with fluid are modeled by linear acoustics. We seek

for elastic velocity $\mathbf{u} \in \tilde{\mathbf{u}}_D + \mathbf{V}$ and pressure scalar field $p \in \tilde{p}_D + V$ such that

$$b_{ee}(\mathbf{u}, \mathbf{v}) + b_{ae}(p, \mathbf{v}) = l_e(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}, \quad b_{ea}(\mathbf{u}, q) + b_{aa}(p, q) = l_a(q), \quad \forall q \in V \quad (1)$$

$$b_{ee}(\mathbf{u}, \mathbf{v}) = \int_{\Omega_e} (E_{ijkl} u_{k,l} v_{i,j} - \rho_s \omega^2 u_i v_i) \mathbf{d}\mathbf{x}, \quad b_{ae}(p, \mathbf{v}) = \int_{\Gamma_I} p v_n \mathbf{d}S \quad (2)$$

$$b_{ea}(\mathbf{u}, q) = -\omega^2 \rho_f \int_{\Gamma_I} u_n q \mathbf{d}S, \quad b_{aa}(p, q) = \int_{\Omega_a} (\nabla p \cdot \nabla q - k^2 p q) \mathbf{d}\mathbf{x}, \quad (3)$$

$$l_e(\mathbf{v}) = \int_{\Omega_e} p_{inc} v_i \mathbf{d}\mathbf{x}, \quad l_a(q) = 0 \quad (4)$$

$\tilde{\mathbf{u}}_D = 0$, $\tilde{p}_D \in H^1(\Omega_a)$ is a finite energy lift of pressure on Γ_{D_a} , where Ω_a is occupied by an acoustical fluid, Ω_e is occupied by a linear elastic medium, Γ_I is the interface separating the two sub-domains, Γ_{D_a} is the Dirichlet boundary of the acoustic part. The spaces of test functions are defined as $\mathbf{V} = \{\mathbf{v} \in \mathbf{H}^1(\Omega_e) : \text{trv} = \mathbf{0} \text{ on } \Gamma_{D_e}\}$ and $V = \{q \in H^1(\Omega_a) : \text{tr}q = 0 \text{ on } \Gamma_{D_a}\}$. Here ρ_f is the density of the fluid, ρ_s is the density of the solid, $E_{ijkl} = \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \lambda\delta_{ij}\delta_{kl}$ is the tensor of elasticities, ω is the circular frequency, c denotes the sound speed, $k = \omega/c$ is the acoustic wave number and p_{inc} is the incident wave impinging from the top $p_{inc} = e^{-ike^x}$, $\mathbf{e} = (-1, 0, 0)$.

3. Projection Based Interpolation

The location of material data inside the human head model has been obtained by using the projection based interpolation algorithm that consists the following sub-steps, related to tetrahedral finite element vertices, edges, faces and interiors. For each finite element, we are looking for a_i coefficients in a particular order. The computational mesh can be generated by using a linear computational cost projection based interpolation routine, first proposed by [4]. We start with vertices, since their coefficients are the most straightforward to compute. There is only one function per each of four vertexes with a support on it and the interpolating function is required to be equal to the interpolant which yields:

$$a_{v_i} = \frac{U(v_i)}{\varphi_i(v_i)} \quad \forall i = 1, \dots, 4, \quad (5)$$

On nodes other than vertices, the input function cannot be represented exactly, so instead we are trying to minimize the representation error. First, on each one of the 6 edges of tetrahedral element:

$$\left\| \left(U - \sum_{j=1}^4 u_{v_j} \right) - \sum_{l=1}^{\dim V|_{e_i}} u_{l, e_i} \right\|_{H_0^1(e_i)} \rightarrow \min \forall i = 1, \dots, 6 \quad (6)$$

where $\dim V|_{e_i}$ signifies the number of edge shape functions in space V with supports on edge e_i . Such a problem can be reduced to a linear system and solved with a linear solver, but if we assume the adaptation order $p = 2$ on each node, for each edge there exists only one shape function with a support on it. Not only is this restriction justified performance-wise (one local equation instead of a system), but it also suffices in most cases, according to our experiments. Thus Equation (6) reduces to:

$$\left\| \overbrace{\left(U - \sum_{j=1}^4 u_{v_j} \right)}^{\dot{U}} - u_{0, e_i} \right\|_{H_0^1(e_i)} \rightarrow \min \forall i = 1, \dots, 6, \quad (7)$$

where \dot{U} vanishes on the element's vertices. After rewriting the norm:

$$\sqrt{\int_{e_i} \sum_{k=1}^3 \nabla \cdot (\dot{U} - u_{0, e_i})^2 \mathbf{d}\mathbf{x}} \rightarrow \min \forall i = 1, \dots, 6, \quad (8)$$

$$\int \sum_{e_i, k=1}^3 \left(\frac{dU}{dx_k} - \frac{du_{o,e_i}}{dx_k} \right)^2 dx \rightarrow \min \forall i = 1, \dots, 6 \quad \text{and} \quad u_{o,e_i} = a_{e_i} \varphi_i \quad (9)$$

we have

$$\int \sum_{e_i, k=1}^3 \left(\frac{dU}{dx_k} \right)^2 dx - 2 \int \sum_{e_i, k=1}^3 \frac{dU}{dx_k} \frac{du_{o,e_i}}{dx_k} dx + \int \sum_{e_i, k=1}^3 \left(\frac{du_{o,e_i}}{dx_k} \right)^2 dx \rightarrow \min \forall i = 1, \dots, 6 \quad (10)$$

which leads to

$$\int \sum_{e_i, k=1}^3 \left(\frac{du_{o,e_i}}{dx_k} \right)^2 dx - 2 \int \sum_{e_i, k=1}^3 \frac{dU}{dx_k} \frac{du_{o,e_i}}{dx_k} dx \rightarrow \min \forall i = 1, \dots, 6 \quad (11)$$

since the other term is constant and can be omitted in minimization.

Let be $b(u, v)$ is a bilinear, symmetric form and $l(u)$ is a linear form defined as:

$$b(u, v) = 2 \int \sum_{e_i, k=1}^3 \frac{dU}{dx_k} \frac{du_{o,e_i}}{dx_k} dx, \quad l(v) = \int \sum_{e_i, k=1}^3 \left(\frac{du_{o,e_i}}{dx_k} \right)^2 dx \quad (12)$$

It is proven that minimizing $1/2b(u, v) - l(u)$ is reducible to solving $b(u, v) = l(u)$ for all test functions v . By applying this lemma we obtain

$$\int \sum_{e_i, k=1}^3 \frac{du_{o,e_i}}{dx_k} \frac{dv}{dx_k} dx = 2 \int \sum_{e_i, k=1}^3 \frac{dU}{dx_k} \frac{du_{o,e_i}}{dx_k} dx \rightarrow \min \forall i = 1, \dots, 6 \quad (13)$$

which leads to

$$a_{e_i} = \int \sum_{e_i, k=1}^3 \frac{dU}{dx_k} \frac{d\varphi_{e_i}}{dx_k} dx \Bigg/ \int \sum_{e_i, k=1}^3 \frac{d\varphi_{e_i}}{dx_k} \frac{d\varphi_{e_i}}{dx_k} dx \quad \forall i = 1, \dots, 6 \quad (14)$$

The next step consists in an optimization on four faces of tetrahedral element:

$$\left\| \overbrace{\left(U - \sum_{j=1}^4 u_{v_j} - \sum_{j=1}^6 u_{0,e_j} \right)}^{\ddot{U}} - u_{0,f_i} \right\|_{H_0^1(f_i)} \rightarrow \min \forall i = 1, \dots, 4, \quad (15)$$

where \ddot{U} vanishes on vertices and edges. This leads to:

$$a_{f_i} = \int \sum_{f_i, k=1}^3 \frac{dU}{dx_k} \frac{d\varphi_{f_i}}{dx_k} dx \Bigg/ \int \sum_{f_i, k=1}^3 \frac{d\varphi_{f_i}}{dx_k} \frac{d\varphi_{f_i}}{dx_k} dx \quad \forall i = 1, \dots, 4 \quad (16)$$

Finally, an analogical optimization in the interior of the finite element:

$$\left\| \overbrace{\left(U - \sum_{j=1}^4 u_{v_j} - \sum_{j=1}^6 u_{0,e_j} - \sum_{j=1}^4 u_{0,f_j} \right)}^{\ddot{U}} - u_{0,l} \right\|_{H_0^1(l)} \rightarrow \min, \quad (17)$$

(where \ddot{U} vanishes everywhere except from the interior) yields:

$$a_l = \int \sum_{f_i, k=1}^3 \frac{dU}{dx_k} \frac{d\varphi_l}{dx_k} dx \Bigg/ \int \sum_{f_i, k=1}^3 \frac{d\varphi_l}{dx_k} \frac{d\varphi_l}{dx_k} dx \quad (18)$$

It is worth noting that using this method the global matrix is not constructed at all. Thanks to the $p = 2$ restriction, we have a single equation over each vertex, edge, face and interior. This algorithm requires a computational cost linear with respect to the mesh size, because it involves constant number of operations for each vertex,

edge, face and interior and the number of respective nodes is proportional to n —the number of elements.

4. Parallel Multi-Frontal Solver Algorithm

The multi-frontal solver algorithm decomposes the computational domain recursively into sub-domains. The resulting elimination tree contains the entire mesh in the root node, particular elements at leaves, as well as some patches of the elements in the intermediate nodes. The solver assigns all the processors to the root node, and partition this list of processors recursively when we partition the mesh. This is illustrated on panel (a) in **Figure 1**. There are four processors assigned to the root node 0, 1, 2, 3, the two processors 0,1 assigned to its left son node and the two processors 2, 3 assigned to its right son node. As a result of the partition of the list of processors, the lower level branches have only one processor assigned. The solver generates first the frontal matrices associated with single elements, compare panel (a) in **Figure 1**. The rows and columns in the matrices are related to basis functions assigned to the nodes of the mesh. In the case of the acoustics coupled with linear elasticity we utilize higher order basis functions, and we have several basis functions assigned to each node of the mesh, except the vertices, where we have single basis functions. We refer to [4] for more details on the higher order finite element method basis functions. The solver eliminates rows representing nodes with fully assembled basis functions. Since interior nodes have basis functions defined only over element interiors, we may eliminate them at leaf nodes, compare panel (b) in **Figure 1**. In the next step, we merge the resulting Schur complement matrices (right bottom not eliminated parts of the frontal matrices) at parent node, compare panel (b) in **Figure 1**. The solver may eliminate rows representing nodes with fully assembled basis functions. This time, we can eliminate basis functions associated to common faces. This is illustrated on panel (c) in **Figure 1**. Up to now, all the processors worked over its local branch of the elimination tree. From now on, the processors will performed the elimination of rows working in parallel. At this point, the Schur complement matrices are merged, compare panel (d) in **Figure 1**, and they are processed at parent node by pairs of processors. The processors may eliminate rows associated to nodes with fully assembled basis functions, located again on two common faces one common edge. This is illustrated on panel (e) in **Figure 1**. On the last level, the resulting Schur complement matrices are merged at top node, and the matrix is processed now with all the processors. The matrix has all the rows associated with nodes with fully assembled basis functions, located on the cross-section of the domain. The top problem can be solved, compared panel (f) in **Figure 1**, and it is followed by recursive backward substitutions.

Additionally, in order to minimize the memory usage, we can dump out the Schur complement matrices to the disc, deallocate them, and read them later at parent nodes in order to merge and construct new systems. In order to implement the out of core version of the parallel solver, we need to have a parallel file system on the parallel machine shared between different processors. For more details on the solver algorithm we refer to [5]-[7].

5. Numerical Results

The PBI algorithm has been executed on the 3D MRI scan data. The exemplary slices of the resulting adaptive mesh are presented in **Figure 2**. The material data and the distinguish between acoustic and linear elasticity domain is based on the MRI scan data. The material data for different domain are based on [2].

The out-of-core multi-frontal solver allows for efficient solution of the propagation of acoustic waves over the human head, see **Figure 3**. Left panel in **Figure 4** presents a profile for 64 processors parallel execution of the out-of-core solver for the mesh with 1,058,622 unknowns with basis functions polynomials of the fourth order. This high polynomial order of approximation and tightly coupled 3D ball shape mesh results in a need for processing of 154,304 frontal matrices with size up to 19 GB. The blue line entitled *total* denotes the total execution time for different number of processors. The light pink line entitled *sequential* denotes the time spend by each processors to process their local branch of the elimination tree. The line includes also time spend by each processor on dumping out and in Schur complement on local SCRATCH directory on LONESTAR cluster. The yellow line entitled *dump* denotes the time spend by each processor on dumping out and dumping in matrices with Schur complements to be exchanged between processors through WORK parallel file system on LONESTAR cluster. The blue line entitled *computation* denotes the time spend during processing of the parallel part of the elimination tree by parallel MUMPS solver used to perform partial Schur complements. The dark pink line entitled *other* denotes time spend by processor on other routines including management of nodes connectivity. The resulting speedup and efficiency of the out-of-core multi-frontal solver is limited by the efficiency of the parallel file system, compare right panel in **Figure 4**.

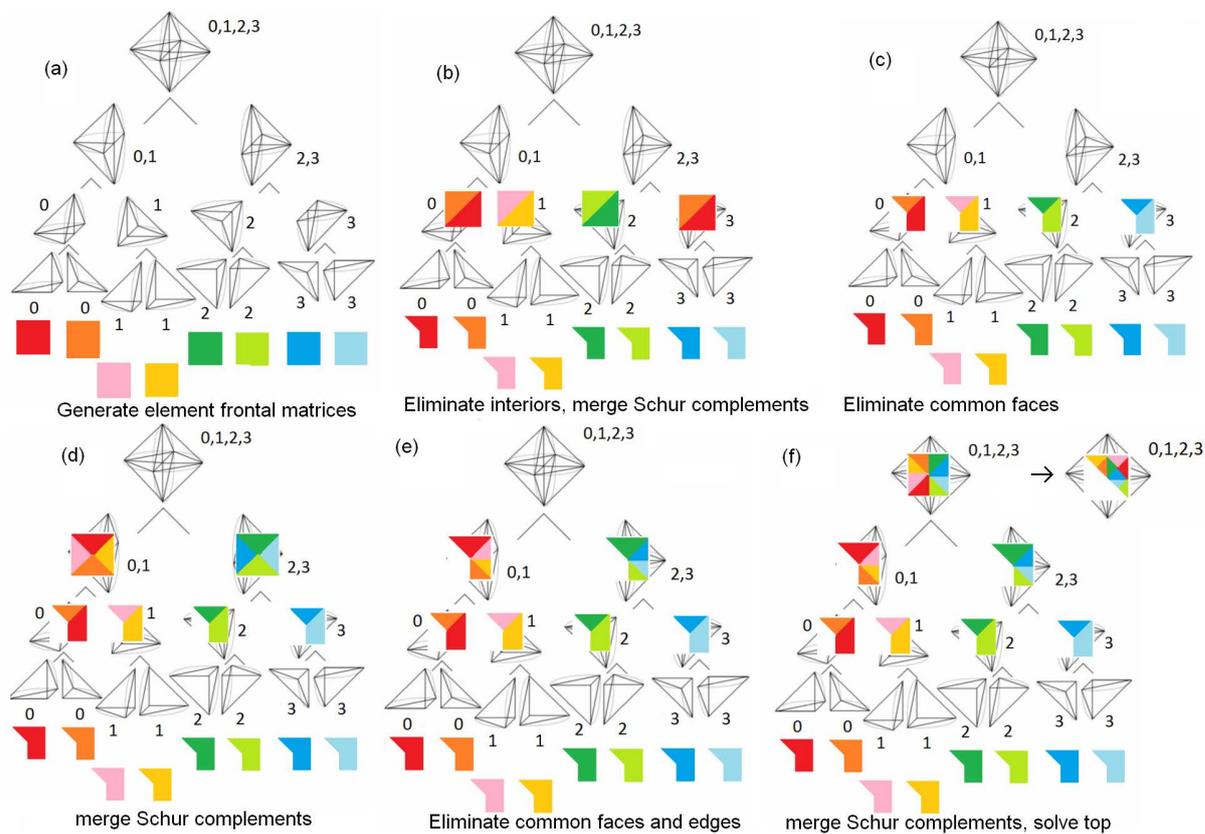


Figure 1. Schematic description of the multi-frontal solver algorithm.

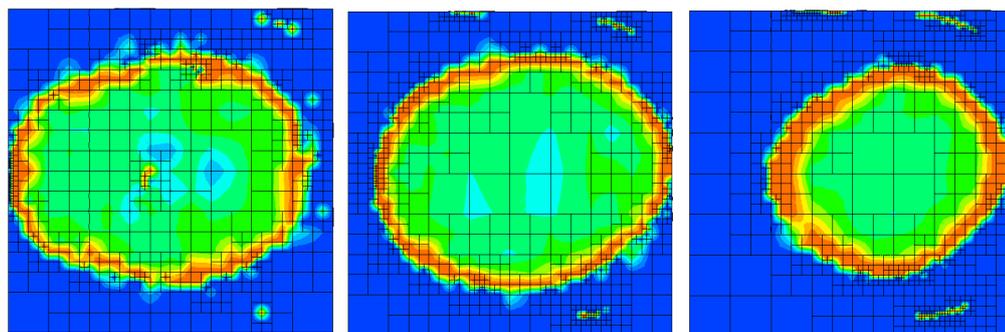


Figure 2. Three slices of the adaptive grid obtained from MRI scan of the human head.

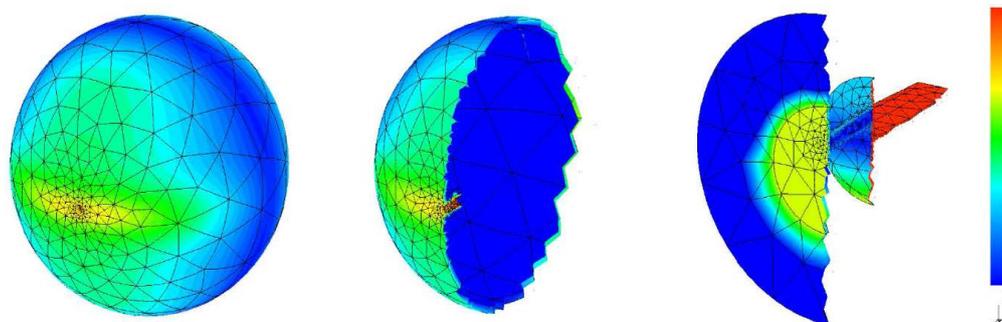


Figure 3. Acoustic pressure on the ball shape domain. Zoom towards membranes of an internal ear.

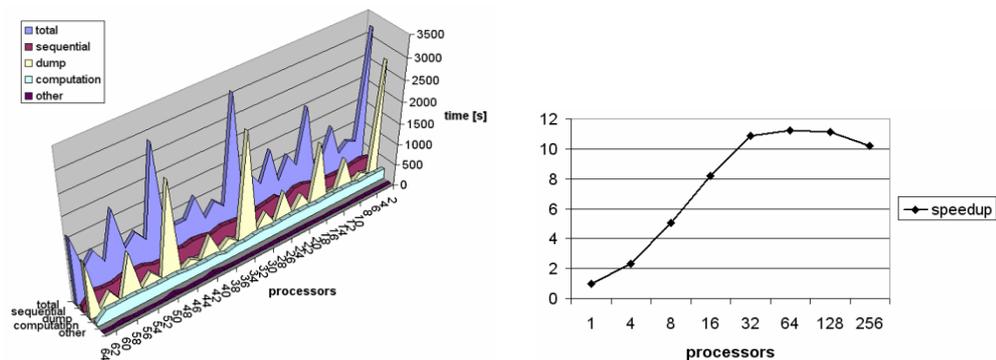


Figure 4. Left panel: Comparison of computations time and dump out/dump in times. Right panel: The efficiency of the solver is limited by the efficiency of the parallel file system.

6. Conclusions

- We presented the parallel algorithm for the out-of-core multi-frontal direct solver dedicated to multi-physics simulations.
- The advantage of our algorithm is that it allows to reduce the memory usage by dumping out all the frontal matrices to the disc.
- The communication in the parallel algorithm occurs through a parallel data storage system.
- The parallel solver scales up to 32 processors, and its scalability is connected with the performance of the parallel file system.
- The solver allowed for efficient solution of the challenging problem of acoustics of the human head.

Acknowledgements

The work of MP presented in this paper has been supported by Polish National Science Center grants no. 2012/06/M/ST1/00363 and 2011/01/B/ST6/00674. The work of PG and MS presented in this paper has been supported by Polish National Science Center grant no. 2011/03/N/ST6/01397.

References

- [1] Sieniek, M. and Paszyński, M. (2014) Subtree Reuse in Multi-Frontal Solvers for Regular Grids in Step-and-Flash Imprint Nanolithography Modeling. *Advanced Engineering Materials*, **16**, 231-240. <http://dx.doi.org/10.1002/adem.201300267>
- [2] Demkowicz, L., Gatto, P., Kurtz, J., Paszynski, M., Rachowicz, W., Bleszynski, E., Bleszynski, M., Hamilton, M. and Champlin, C. (2010) Modeling of Bone Conduction of Sound in the Human Head Using hp Finite Elements I. Code Design and Verification. *Computer Methods in Applied Mechanics and Engineering*, **200**, 1757-1773. <http://dx.doi.org/10.1016/j.cma.2011.02.001>
- [3] Paszyński, M. (2013) Minimizing the Memory Usage with Parallel Out-of-Core Multi-Frontal Direct Solver. *Computer Assisted Mechanics and Engineering Sciences*, **20**, 15-41.
- [4] Demkowicz, L., Kurtz, J., Pardo, D., Paszyński, M., Rachowicz, W. and Zdunek, A. (2007) Computing with hp-Adaptive Finite Elements, vol. II. Three Dimensional Elliptic and Maxwell Problems with Applications. Chapman & Hall, CRC Press.
- [5] Paszyński, M., Pardo, D., Torres-Verdin, C., Demkowicz, L. and Calo, V. (2010) A Parallel Direct Solver for the Self-Adaptive hp Finite Element Method. *Journal of Parallel and Distributed Computing*, **70**, 270-281. <http://dx.doi.org/10.1016/j.jpdc.2009.09.007>
- [6] Paszyński, M. and Schaefer, R. (2010) Graph Grammar Driven Parallel Partial Differential Equation Solver. *Concurrency & Computations, Practise & Experience*, **22**, 1063-1097.
- [7] Paszyński, M., Pardo, D. and Paszyńska, A. (2010) Parallel Multi-Frontal Solver for p Adaptive Finite Element Modeling of Multi-Physics Computational Problems. *Journal of Computational Science*, **1**, 48-54. <http://dx.doi.org/10.1016/j.jocs.2010.03.002>