Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE

Informatyka w Technologii Materiałów

Vol. 13, 2013, No. 1

GRAPH GRAMMAR FOR THREE DIMENSIONAL MULTI-PHYSICS ADAPTIVE FINITE ELEMENT METHOD SIMULATIONS

ANNA PASZYŃSKA[1], MACIEJ PASZYŃSKI[2]*

[1] *Jagiellonian University, Faculty of Physics, Astronomy and Applied Computer Science, Reymonta 4, 30-059, Krakow, Poland*
[2] *AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunication, Department of Computer Science al. A. Mickiewicza 30, 30-059 Krakow, Poland*
**Corresponding author: maciej.paszynski@agh.edu.pl*

Abstract

In this paper we present an application of graph grammar for modelling a three dimensional hp adaptive finite element method for multi-physics simulations. The graph grammar model involves the process of generation of the mesh with tetrahedral finite elements, the process of mesh adaptation by means of h refinement as well as interfacing with the multi-frontal solver algorithm by translating the finite elements into a sequence of frontal matrices, being an input for the multi-frontal solver algorithm. The paper is concluded with numerical results concerning an advanced multi-physics problems, the propagation of an acoustic waves over the model of the human head.

Key words: Graph grammar, *hp*-Adaptive Finite Element Method, multi-physics problems

1. INTRODUCTION

1. INTRODUCTION

The paper presents an application of graph grammar modeling for a three dimensional simulations of multi-physics problems by means of the finite element method (Hughes 2000). Such a simulation consists of three dimensional mesh generation followed by the execution of the multi-frontal direct solver (Amestoy et al. 2000, Amestoy et al. 2001, Paszyński et al. 2010a). In addition, we include either h or p refinement procedure of the computational mesh in order to increase the accuracy of the solution (Demkowicz et al. 2007, Paszyński & Demkowicz 2006). The mesh generation, mesh refinements and translation of the mesh into a sequence of matrices for the multi-frontal solver algorithm execution are expressed by graph grammar productions (Paszyński & Paszyńska 2008, Paszyński 2009a, Paszyńska et al. 2008, Paszyńska et al. 2009, Paszyńska et al. 2012a, Paszyńska et al. 2012b). The mesh generation algorithm is expressed by a sequence of graph grammar productions, where each graph grammar production generates a new single finite element and connects the element to the face of an already existing element. The computational mesh is assumed to contain tetrahedral as well as prism elements, thus, we distinguish several graph grammar productions for generating a new tetrahedral or prism element and adding the newly created element to a face of some other element that can be either tetrahderal or prism element. The mesh h re-

 ISSN 1641-8581

finement algorithm consists of breaking tetrahedral and prism elements. Each tetrahedral element is broken into four smaller tetrahedral elements and two piramid elements. The prism element is broken into eight smaller tetrahedral elements. The mesh generation and refinement is followed by the execution of the multi-frontal direct solver algorithm. The input for the multi-frontal solver is a sequence of element matrices with rows and columns associated with finite element vertices, edges, faces and interiors. We also introduce graph grammar productions for transferring the computational mesh into a sequence of element frontal matrices for the solver execution (Paszyński et al. 2010b, Paszyński et. al. 2011). The paper is concluded with numerical results presenting the generation of a computational mesh and solution procedure for the problem of acoustics of the human head.

2. ACOUSTIC OF THE HUMAN HEAD

We focus here on numerical simulations of the acoustics of 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 (Demkowicz et al. 2011). 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

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

$$b_{ea}(\mathbf{u},q) + b_{aa}(p,q) = l_a(q), \quad \forall q \in V \tag{2}$$

$$b_{ee}(\mathbf{u},\mathbf{v}) = \int_{\Omega_e} \left(E_{ijkl}\, u_{k,l}\, v_{i,j} - \rho_s\, \omega^2 u_i\, v_i \right) d\mathbf{x} \tag{3}$$

$$b_{ae}(p,\mathbf{v}) = \int_{\Gamma_1} p\, v_n\, dS \tag{4}$$

$$b_{ea}(\mathbf{u},q) = -\omega^2 \rho_f \int_{\Gamma_1} u_n\, q\, dS \tag{5}$$

$$b_{aa}(p,q) = \int_{\Omega_a} \left(\nabla p \cdot \nabla q - k^2 p\, q \right) d\mathbf{x} \tag{6}$$

$$l_e(\mathbf{v}) = \int_{\Omega_e} p_{inc}\, v_i\, d\mathbf{x} \tag{7}$$

$$l_a(q) = 0 \tag{9}$$

$\tilde{\mathbf{u}}_D = 0$, $\tilde{p}_D \in H^1(\Omega_a)$ is a finite energy lift of pressure prescribed on Γ_{D_a}, where Ω_a part is occupied by an acoustical fluid, Ω_e part 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

$$\mathbf{V} = \left\{ \mathbf{v} \in \mathbf{H}^1(\Omega_a) : \mathrm{tr}\, \mathbf{v} = \mathbf{0} \text{ on } \Gamma_{D_e} \right\} \tag{10}$$

$$V = \left\{ q \in H^1(\Omega_a) : \mathrm{tr}\, q = 0 \text{ on } \Gamma_{D_a} \right\} \tag{11}$$

Here ρ_f is the density of the fluid, ρ_s is the density of the solid, $E_{ijkl} = \mu\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) + \lambda\delta_{ij}\delta_{kl}$ 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 $p_{inc} = e^{-ik\mathbf{e}x}$ $\mathbf{e} = (-1,0,0)$. For more details we refer to (Demkowicz et al. 2011).

3. GRAPH GRAMMAR PRODUCTIONS FOR MESH GENERATION

The generation of the computational mesh is expressed as a sequence of graph grammar productions. This is illustrated in figure 1, where we start from a graph with a single node attributed by *S* label, and execute (Pinittet) graph grammar production, to generate the structure of the first element. This is followed by executing the sequence of graph grammar productions (Paddtet2) replacing a face of one of previously generated element by a newly generated neighbor. After executing a sequence of eight productions we get the most simple representation of the ball shape domain, approximated with tetrahedral elements. In the real computation we generate a ball shape mesh with 19,288 finite elements, and we artificially drill an ear canal with cochlea in order to simulate an internal ear, see figure 2.

4. GRAPH GRAMMAR PRODUCTIONS FOR MESH ADAPTATION

In order to increase the accuracy of the numerical solution the computational mesh is ofthen refined in the areas with low accuracy, either by performing *h* refinement or *p* refinement. In our case the *h* refinement represents breaking of tetrahedral finite element into smaller elements. This is illustrated in figure 3. As a result we obtain four new tetrahedral elements and two new pyramid elements.

The process of mesh refinement is represented by a sequence of graph grammar productions presented in figures 4,5 and 6. The procedure starts with

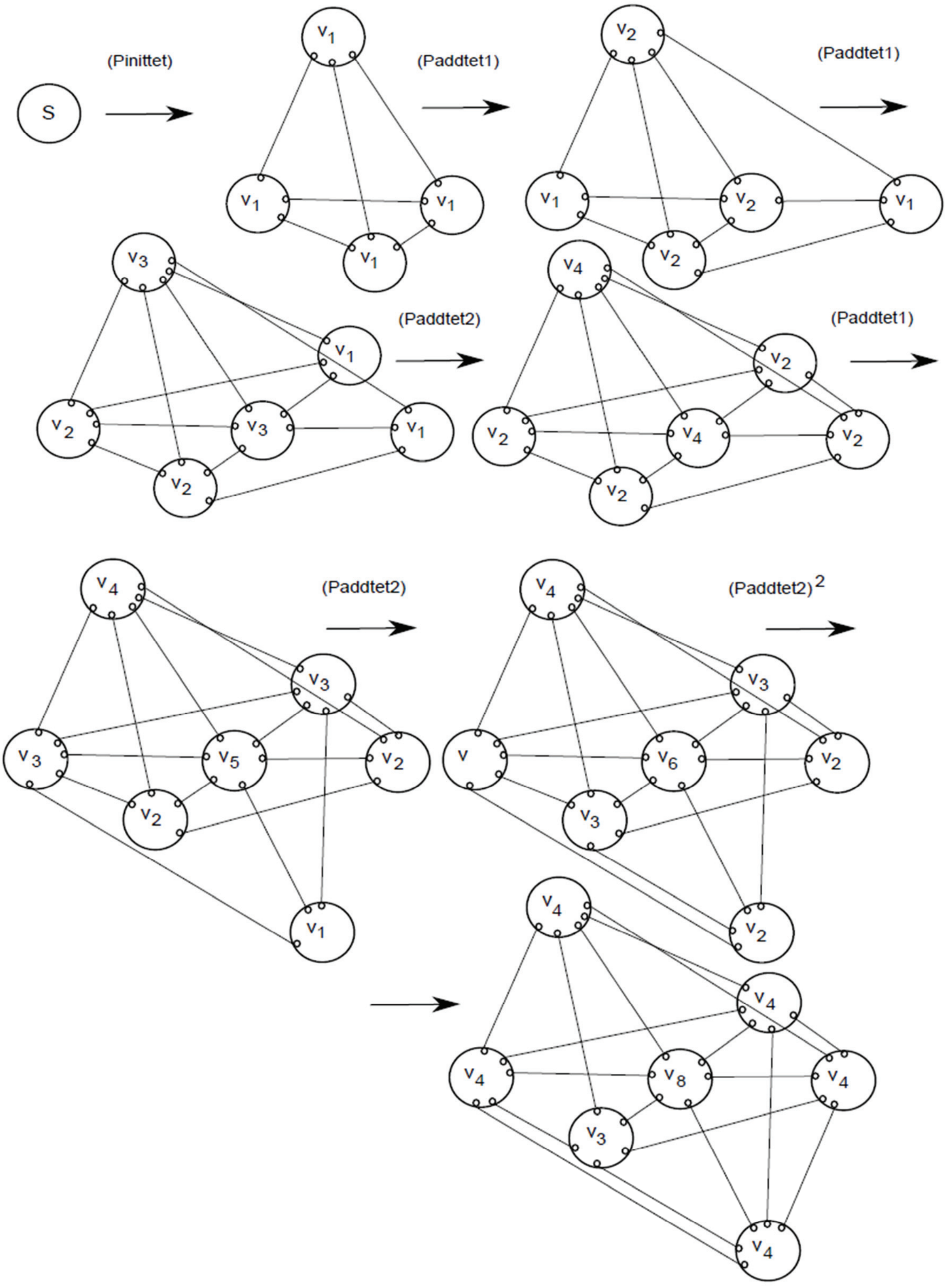

Fig. 1. Sequence of graph grammar productions for generation of the mesh.

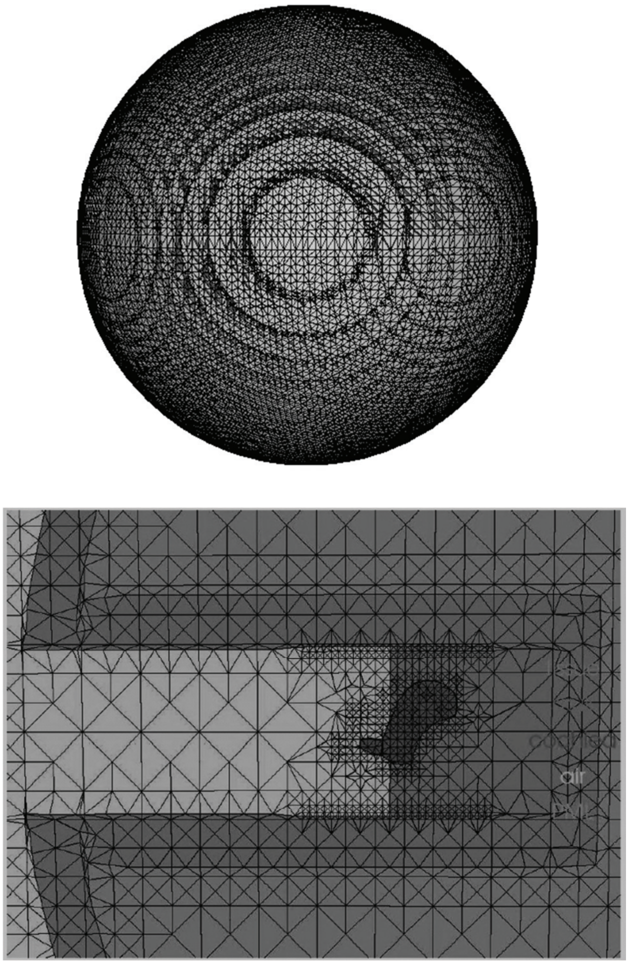

Fig. 2. Left panel: Ball shape domain, Right panel: cross section of the domain with an internal ear and cochlea.

breaking element interior, which is done by executing graph grammar production **(breakint)** presented in figure 6. The production generates interior nodes for four new tetrahedral and two new pyramids, as well as face and edge nodes for faces located on the interfaces between newly generated elements.

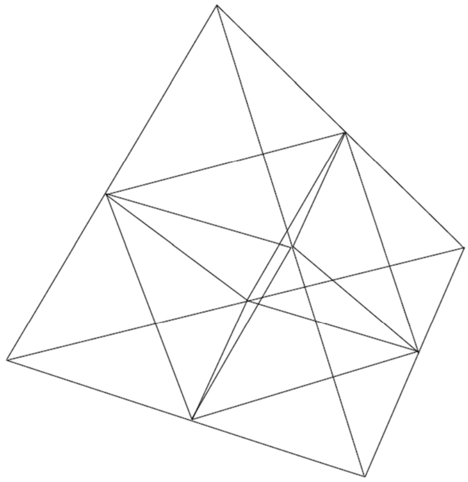

Fig. 3. Breaking of tetrahedral element into 4 tetrahedral element and 2 pyramid elements.

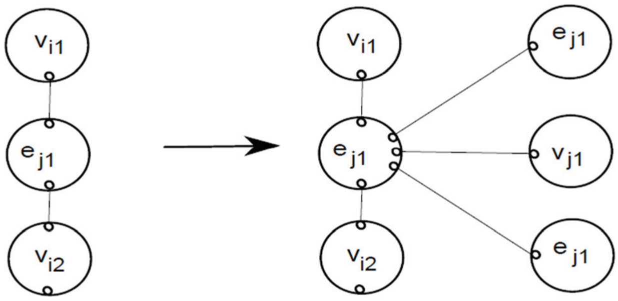

Fig. 4. Graph grammar production ***(breakedge)*** *breaking of an edge into two new edges and one vertex.*

Additionally, the graph grammar production generates the connectivity information. If an element is neighboring some other already broken elements, the corresponding face located between two broken elements is also broken. This is done by executing the graph grammar production **(breakface)** presented in figure 5. The production generates face nodes for three newly generated faces as well as edge

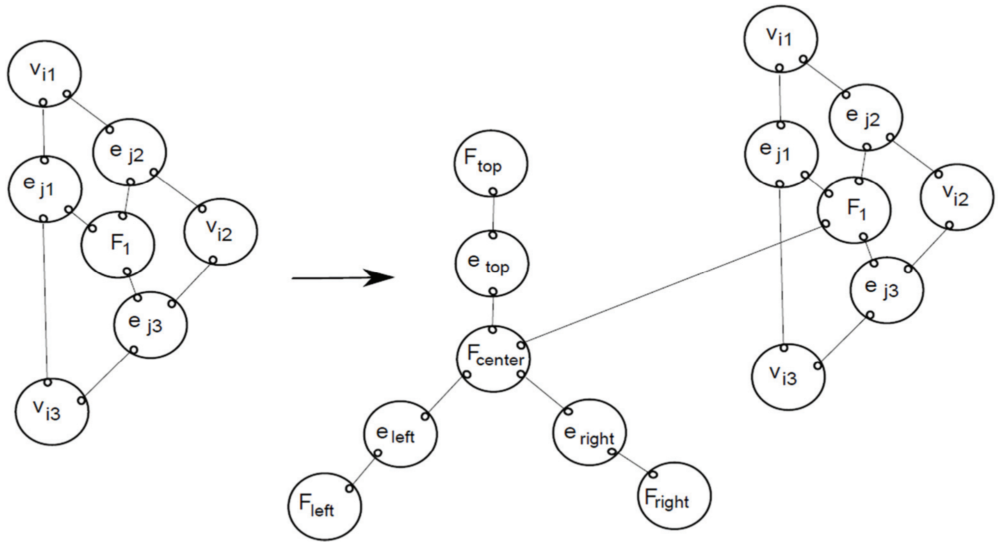

Fig. 5. Graph grammar production ***(breakface)*** *breaking of a face node into three interior nodes, three edge nodes, and a central node.*

nodes and one vertex. Finally, if there are common edges shared between already broken elements, the edges are also broken by executing graph grammar production **(breakedge)** presented in Figure 4.

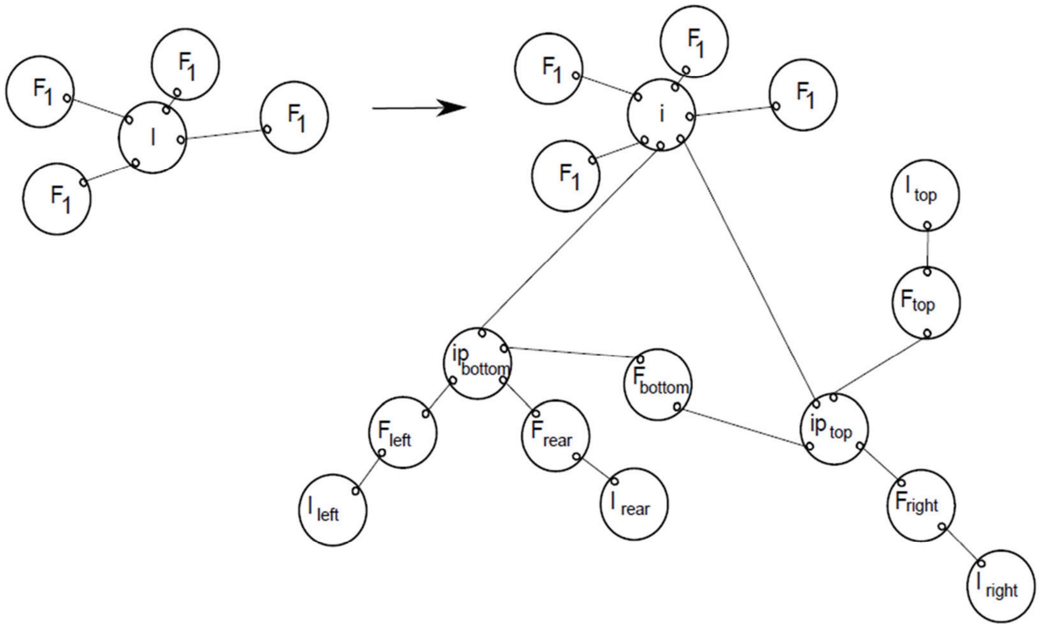

*Fig. 6. Graph grammar production **(breakint)** breaking of a tetrahedral interior into two interior nodes of pyramid elements, three interior nodes of three tetrahedral elements, as well as several interior face nodes, edge nodes.*

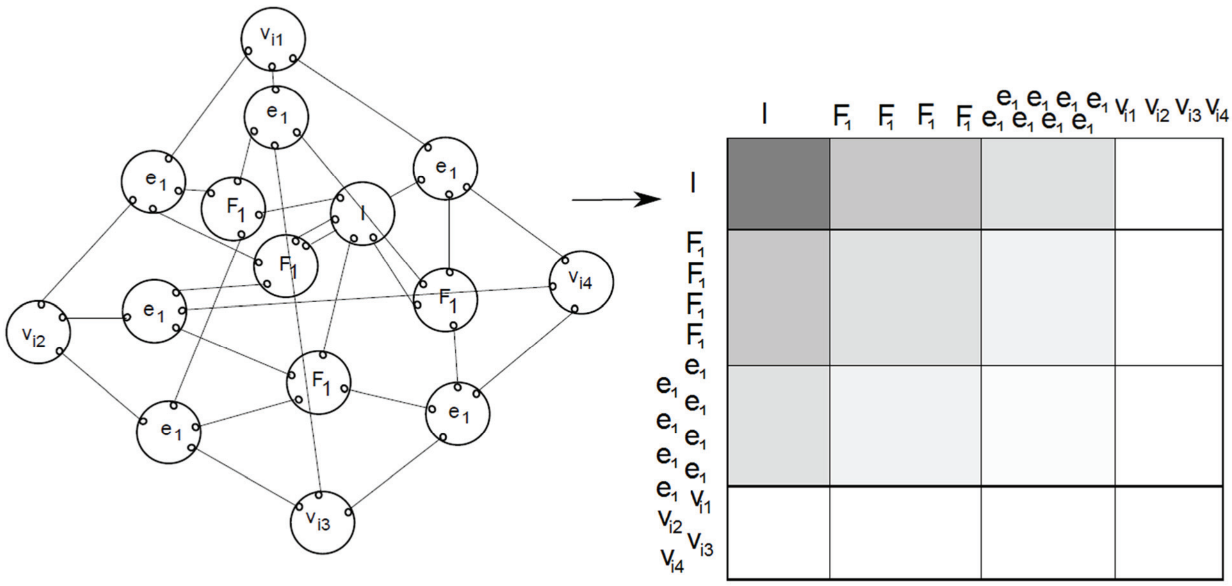

*Fig. 7. Graph grammar production **(generate)** transferring a sub-graph of the graph representation of the mesh into a frontal matrix.*

5. GRAPH GRAMMAR PRODUCTIONS INTERFACING WITH MULTI-FRONTAL SOLVER ALGORITHM

The computational mesh, possibly refined with some *h* adaptations is transformed into a sequence of frontal matrices for the multi-frontal solver algorithm. For the details of the multi-frontal solver algorithm we refer to (Paszyński et al. 2010b, Paszyński et. al. 2011). An input for the multi-frontal solver algorithm is a sequence of frontal matrices with rows and columns labeled by global identifiers of mesh nodes. The identifiers can be generated by global browse of the computational mesh and assigning labels to nodes. The process of generation of the frontal matrix is based on the discretization of the variational formulation, and the details can be found in (Demowicz et al. 2007, Demkowicz et al. 2011). From the point of view of the graph grammar, each sub-graph of the graph representation of the mesh, representing a single finite element is transferred into a frontal matrix. This is illustrated in figure 7, where a sub-graph associated with a single element is transferred into an element matrix, with rows and columns representing interior, face, edge and vertex nodes. This is actually the only place where the multi-physics computations influence the graph grammar model. In other words, the size and the sparsity pattern of the element frontal matrix depends on the multi-physics equations used.

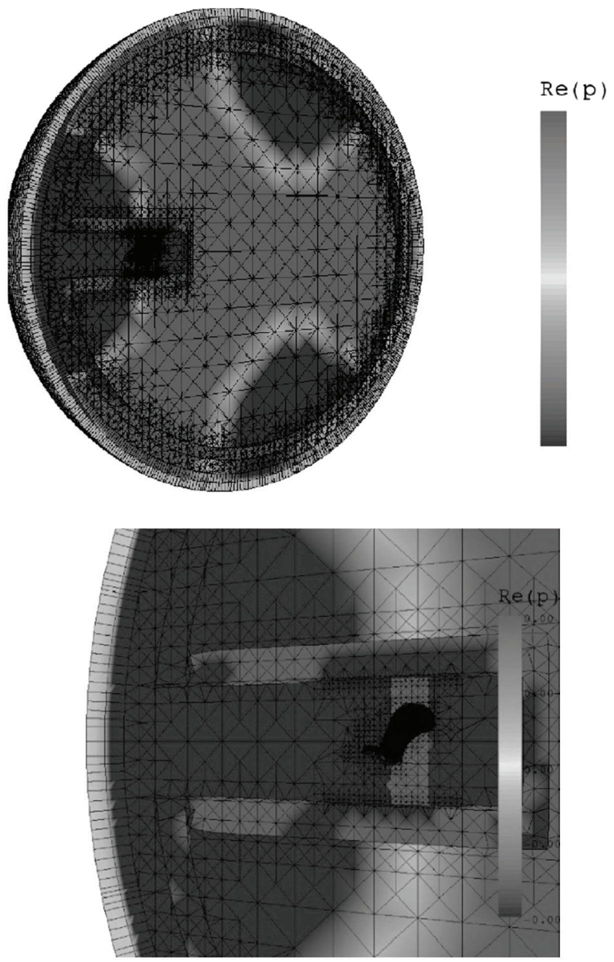

Fig. 8. Real part of the pressure in the domain with zoom to the cochlea part

6. NUMERICAL RESULTS

We conclude the presentation with some numerical results obtained for the propagation of the acoustic waves with the simplified model of the human head with cochlea. The cochlea, presented in figure 2 has been connected with the skull by growing manually an additional bone elements between the skull and the cochlea. In this model, the domain consists of four concentric spheres, with a hole. The most inner sphere is filled with an elastic material with data corresponding to human brain. The first layer is also elastic with constants corresponding to human skull. The graph grammar is a model of data structure transformations executed by the computational code. The process of mesh generation and embedding of the cochlea is controlled by routines setting the order of execution of graph grammar productions. We have solved the computational problem where we assumed that the interior of the human head is filled with water. The results are presented in figures 8, with zoom at the cochlea part. For more details on the numerical formulation and more numerical results we refer to (Demkowicz et al. 2011). The entire process of the computation can be expressed as a sequence of graph grammar productions, starting with productions responsible for the mesh generation, adaptation and interfacing with the solver:

$$\mathbf{(Pinittet)\text{-}(Paddtet2)^{m}\text{-}(Pbreakinterior)^{n}\text{-}(Pbreakface)^{o}\text{-}(Pbreakedge)^{p}\text{-}(Pgenerate)^{r}}$$

7. CONCLUSIONS AND FUTURE WORK

In this paper we discussed a graph grammar model for advanced computational process, namely the three dimensional adaptive finite element method for the multi-physics problems. Expressing of the computational problem by graph grammar productions has a sequence of advantages. First of all, we can understand graph grammar productions as basic undividable tasks. Next, for a given computational problem, we can plot a graph of dependencies representing the partial order of execution of graph grammar productions. We can color the graph, and based on the colors we can schedule the tasks to shared memory parallel machine, in order to obtain best possible parallel implementation of the computational problem. It has been already done for two dimensional grids (Paszyński & Schaefer 2009, Paszyński 2009a, Paszyński 2009b) Another advantage of the graph grammar based representation of the adaptive procedure, is the possibility of constructing Petri nets controlling the order of execution of graph grammar production. Once we do that, we can utilize computer science tools applicable for Petri nets, such as plotting reachability graphs, automatic detecting of deadlock of mesh adaptation procedure, and checking the correctness of the parallel algorithms. This has been already done for two dimensional grids (Szymczak et al 2011)

Acknowledgements. The support of Polish MNiSW grant no NN 519 405 737 and grant no DEC-2011/01/B/ST6/00674 is gratefully acknowledged.

REFERENCES

Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., 2000, Multifrontal parallel distributed symmetric and unsymmetric solvers, *Computer Methods in Applied Mechanics and Engineering*, 184, 501-520.

Amestoy, P. R., Duff, I. S., Koster, J., L'Excellent, J.-Y., 2001, A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM Journal of Matrix Analysis and Applications*, 23, 15-41.

Demkowicz, L., Gatto, P., Kurtz, J., Paszyński, M., Rachowicz, W., Bleszyński, E., Bleszyński, M., Hamilton, M., Champlin, C., Pardo, D., 2011, Modeling of Bone Conduction of Sound in the Human Head Using hp Finite Elements: Code Design and Verification, *Computer Methods in Applied Mechanics and Engineering,* 200, 1757-1773.

Demkowicz, L., Rachowicz, W., Pardo, D., Paszyński, M., Kurtz, J., Zdunek, A., 2007, *Computing with hp-Finite Elements. Volume II*, Chapmann & Hall / CRC Press, Taylor Francis Group, Boca Raton, London, New York.

Hughes, T. J. R. 2000, *The Finite Element Method, Linear Statics and Dynamics Finite Element Method Analysis,* Dover Publications, New York.

Paszyńska, A., Grabska, E., Paszyński, M., 2012a, A Graph Grammar Model of the hp Adaptive Three Dimensional Finite Element Method, Part I, *Fundamenta Informaticae,* 114, 149-182.

Paszyńska A., Grabska, E., Paszyński, M., 2012b, A Graph Grammar Model of the hp Adaptive Three Dimensional Finite Element Method, Part II, *Fundamenta Informaticae,* 114, 183-201.

Paszyńska, A., Paszyński, M., Grabska, E., 2008, Graph Transformations for Modelling hp-Adaptive Finite Element Method with Triangular Elements, *Lecture Notes in Computer Science,* 4967, 604-613.

Paszyńska, A., Paszyński, M., Grabska, E., 2009, Graph Transformations for Modelling hp-Adaptive Finite Element Method with Mixed Triangular and Rectangular Elements, *Lecture Notes in Computer Science,* 5545, 875-884.

Paszyński, M., 2009a, On the Parallelization of Self-Adaptive hp-Finite Element Method, Part I., Composite Programmable Graph Grammar, *Fundamenta Informaticae,* 93, 411-431.

Paszyński, M., 2009b, On the Parallelization of Self-Adaptive hp-Finite Element Method, Part II., Partitioning Communication Agglomeration Mapping Model, *Fundamenta Informaticae,* 93, 435-457.

Paszyński, M., Demkowicz, L., 2006, Parallel Fully Automatic hp-Adaptive 3D Finite Element Package, *Engineering with Computers,* 22, 255-276.

Paszyński, M., Pardo, D., Torres-Verdin, C., Demkowicz, L., Calo, V., 2010a, A Parallel Direct Solver for the Self-Adaptive hp Finite Element Method, *Journal of Parallel and Distributed Computing*, 70, 270-281.

Paszyński, M., Pardo, D., Paszyńska, A., 2010b, Parallel Multi-Frontal Solver for *p* Adaptive Finite Element Modeling of Multi-Physics Computational Problems, *Journal of Computational Science,* 1, 48-54.

Paszyński, M., Pardo, D., Paszyńska, A., Demkowicz, L., 2011, Out-of-core Multi-Frontal Solver for Multi-Physics *hp* Adaptive Problems, *Procedia Computer Science,* 4, 1788-1797.

Paszyński, M., Paszyńska, A., 2008, Graph Transformations for Modelling Parallel hp-Adaptive Finite Element Method, *Lecture Notes in Computer Science,* 4967, 1313-1322.

Paszyński, M., Schaefer R., 2009, Graph Grammar Driven Parallel Partial Differential Equation Solver, *Concurrency & Computation, Practise &* Experience, 11, 1063-1097.

Szymczak, A., Paszyńska, A., Paszyński, M., Pardo, D., 2011, Preventing Deadlock During Anisotropic 2D Mesh Adaptation in *hp* Adaptive FEM, *Journal of Computational Science*, in press.

GRAMATYKA GRAFOWA DLA ADAPTACYJNEJ METODY ELEMENTÓW SKOŃCZONYCH DO SYMULACJI TRÓJWYMIAROWYCH PROBLEMÓW WIELO-FIZYCZNYCH

Streszczenie

W artykule tym przedstawiamy gramatykę grafową do modelowania adaptacyjnej metody elementów skończonych służącej do wykonywania trójwymiarowych symulacji problemów wielo-fizycznych. W szczególności model gramatyki grafowej obejmuje proces generacji siatki obliczeniowej zbudowanej z elementów czworościennych, proces zagęszczania siatki obliczeniowej za pomocą mechanizmu h adaptacji, a także interfejs do solwera wielo-frontalnego uzyskany poprzez generowanie ciągu macierzy frontalnych dla poszczególnych elementów skończonych. Artykuł podsumowują przykładowe wyniki symulacji dotyczące zagadnienia propagacji fal akustycznych na modelu głowy ludzkiej.

Received: October 17, 2012
Received in a revised form: November 9, 2012
Accepted: November 21, 2012