JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
**52**, 1, pp. 71-80, Warsaw 2014

# PARALLEL MESH GENERATOR FOR BIOMECHANICAL PURPOSE

Hubert Hausa, Michał Nowak

*Poznan University of Technology, Institute of Combustion Engines and Transport, Poznań, Polska*

*e-mail: hubert.hausa@put.poznan.pl; michal.nowak@put.poznan.pl*

The analysis of a biological structure with numerical methods based on engineering approach (i.e. Computational Solid Mechanics) is becoming more and more popular nowadays. The examination of complex, well reproduced biological structures (i.e. bone) is impossible to perform with a single workstation. The mesh for Finite Element Method (FEM) of the order of $10^6$ is required for modeling a small piece of trabecular bone. The homogenization techniques could be used to solve this problem, but these methods require several assumptions and simplifications. Hence, effective analysis of a biological structure in a parallel environment is desirable. The software for structure simulation at cluster architecture are available; however, FEM generator is still inaccessible in that environment. The mesh generator for biological applications – Cosmoprojector – developed at Division of Virtual Engineering, Poznan University of Technology has been adapted for the parallel environment. The preliminary results of complex structure generation confirm the correctness of the proposed method. In this paper, the algorithm of computational mesh generation in a parallel environment has been presented. The proposed system has been tested at biological structure.

*Key words:* parallel processing, finite element mesh generation, biomechanics

## 1. Introduction

The high efficiency of software and algorithms is necessary in the case of simulation of large scale objects. This problem is significant in numerical analysis of biological structures. The commercial software for mesh generation is based on geometrical approach; however, the medical data are available in 2D (two-dimensional) images form – microcomputer tomography (micro-CT). Hence, there is a lack in software which could create a model using only such type of data. Additionally, not every software could generate FEM models in a parallel environment. The generation of mesh using that software is limited by size or desired accuracy of the final model. The Simpleware ScanIP software is a good example of the environment which provides an input mesh for FEM analysis based on medical images (Simpleware, 2008). But, the usage of the software is limited due to restriction of ability of parallel generation. Another possibility is to apply the homogenization process (Będziński and Tyndyk, 2003; Nowak *et al.*, 2010; Telega *et al.*, 1999). According to that approach, the bone structure is simplified and it has got averaged mechanical properties in the control volume. It allows one to analyse the whole bone with certain approximation – the accuracy of results depends on assumed material properties. The accurate mesh allows one to control of the whole bone behaviour at the microstructure level only. Hence, the parallel mesh generator is desired.

## 2. The biomimetic mesh generator – Cosmoprojector

The main module of the biomimetic optimization system developed at Division of Virtual Engineering, Poznan University of Technology, is Cosmoprojector – FEM mesh generator. The idea of optimization based on biomimetic remodeling phenomena (Nowak, 2006a,b) assumed the existence of a relation between the trabecular bone structure and applied external load. The constant Strain Energy Density (SED) at the surface of the structure guarantees optimal design in this case (Pedersen, 2003) – the homeostasis state, thus the method of structural modification can be also used in mechanical design as a method of topology optimization. The structure is modified at the surface as an analogy to constant adding and removing a bone in a biological organism. When the SED on the surface is larger than the assumed level, the material is added in order to balance the energy in the structure. Otherwise the material is removed.

The developed optimization procedure requires dedicated mesh generator – Cosmoprojector (Nowak, 2006a). This mesh generator has got the following specific features:

- the input data are a collection of two-dimensional images – the CT digital images,
- the full automatic mesh generation in each optimization iteration – without user interaction,
- the unrestricted evolution of the structure – the FEM model should be able to connect and disconnect in each iteration of the optimization process,
- the strain density energy must be taken into account as a remodeling scenario.

The picture below (Fig. 1) presents the optimization system. The dash line marks mesh generator – Cosmoprojector.

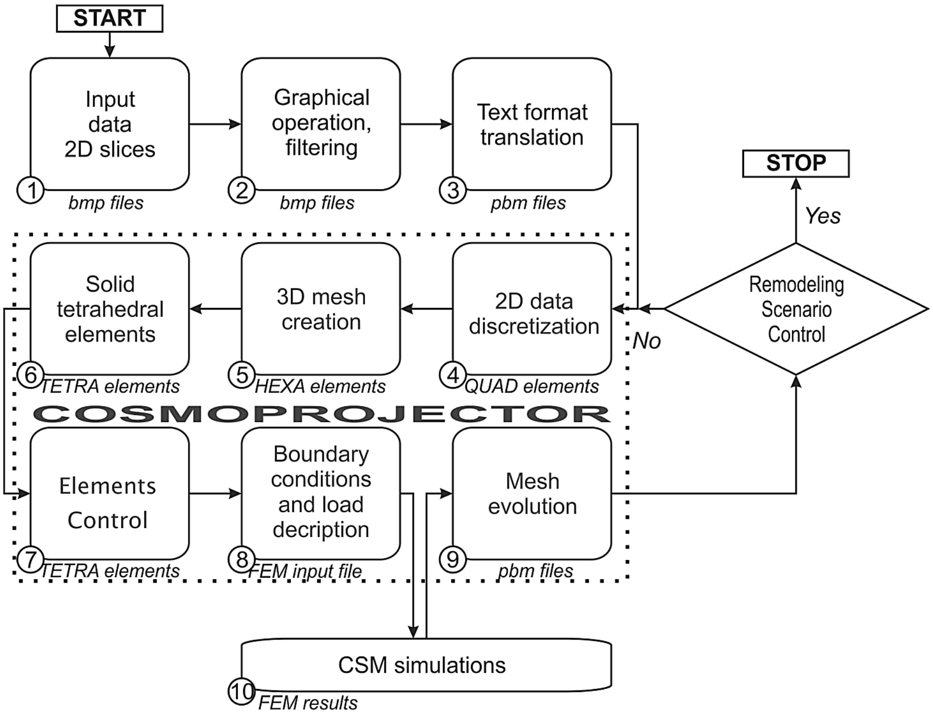
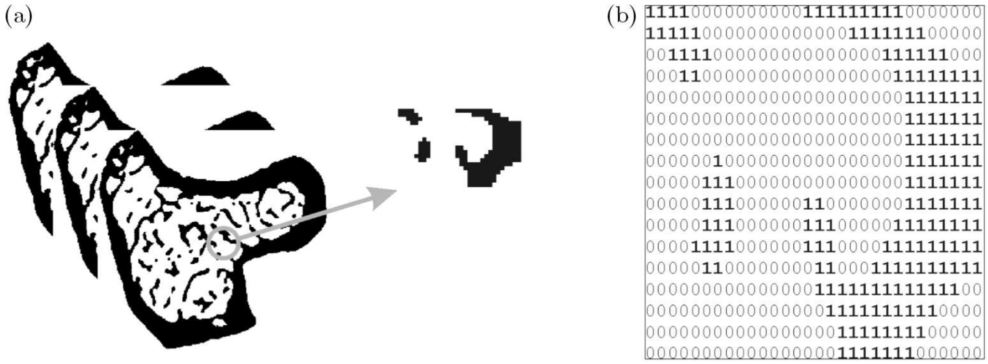


Fig. 1. The scheme of a biomimetic optimization system where the mesh generator – Cosmoprojector – is the essential part (dash line) (Nowak, 2006b)

The necessary steps required for FEM mesh generation based on 2D slices are described below.

The input data are submitted to image processing in the first stage of mesh generation (Fig. 1-①). The aim of this task is to separate bone tissue, and it is only taken into account for further simulation. After that, the data are converted into two-color images, where “1” represents a tissue and “0” a void (Fig. 1-③). The results of these two steps are presented at the picture below (Fig. 2).



```
111100000000001111111110000000
111110000000000000111111100000
001111000000000000000111111000
000110000000000000000011111111
000000000000000000000001111111
000000000000000000000001111111
000000000000000000000001111111
000000100000000000000001111111
000001110000000000000001111111
000001110000001100000001111111
000001110000001110000011111111
000011110000001110000111111111
000001100000000110001111111111
000000000000000111111111111100
000000000000000011111111110000
000000000000000001111111100000
000000000000000001111111000000
```

Fig. 2. The input data for mesh generator – the bone represented by a set of micro-CT images. (a) The set of two-dimensional images with a part of data marked for further demonstration and (b) examplary data converted into the input format

These steps are performed only once, at the beginning of the process with active user interaction. This procedure is not included in automatic mesh generation. Every operation on separated images is automatically applied to the rest of the data.

Next, the automatic mesh generation is started. First (Fig. 1-④), each image is discretized into two-dimensional QUAD elements. It is done by dividing images (slices) in each direction ($x$ and $y$ coordinates) by a constant value and adding the node at the cross-section division. The node is defined in place where the material exits. Two nodes create the edge, four edges give a quadrilateral element. The $z$ direction is defined by the adjacent section – projection of the slices (2D images). The example of this step is presented in Fig. 3a.

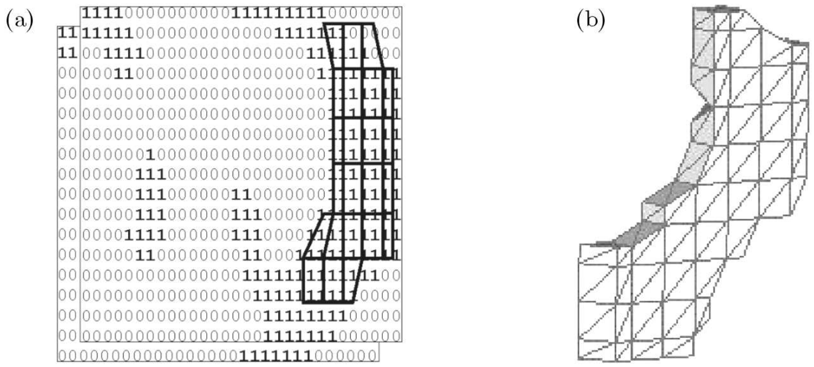


Fig. 3. The discretization of input data: (a) division of images into QUAD elements and (b) the projection of QUAD elements into adjacent slices – generating HEXA elements and their division into TETRA elements

The three-dimensional hexahedron elements are obtained as a result of a projection of each section onto the adjacent one (Fig. 1-⑤). If QUAD element from the first section has got a corresponding element at the second section, then the corners of these elements are connected – the three-dimensional element is generated. Every HEXA element is divided into maximum six tetrahedral elements (Fig. 1-⑥).

The next step involves (Fig. 1-⑦) the mesh control procedure. The degenerated elements with a negative volume are eliminated. Hence, the repaired grid must be renumbered. The example mesh at this stage is presented in Fig. 4. Successively, the boundary conditions are assigned to the mesh, and it is stored with ABAQUS file format.

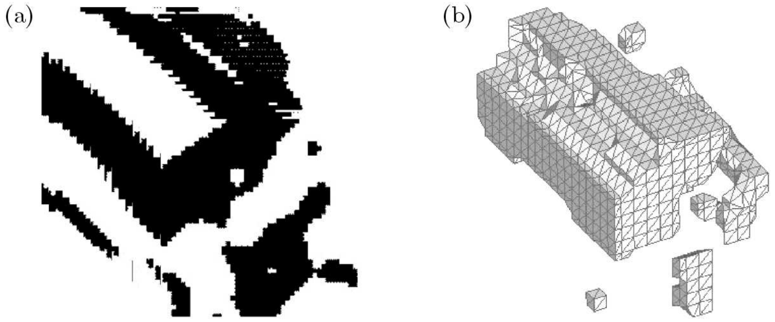
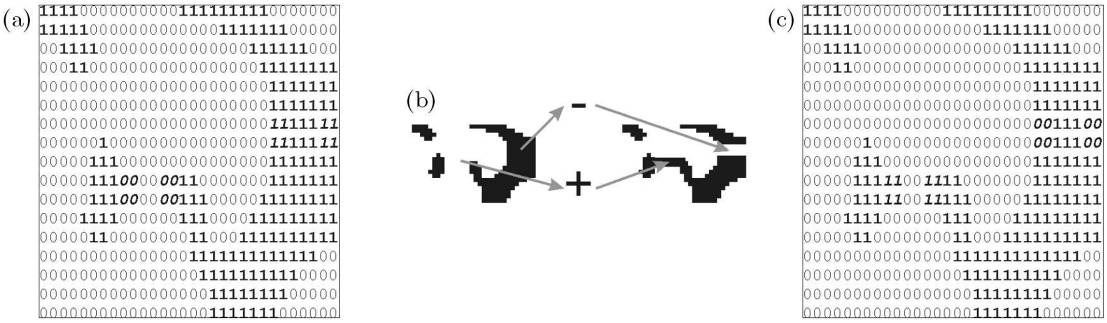


Fig. 4. (a) An exemplary set of tissue data and (b) corresponding FEM grid

The biomimetic optimization procedure requires modification on the structural surface. It is done by the Cosmoprojector at stage 9 (Fig. 1-⑨) by modification of the input data – pixels from "0" to "1" and vice versa (Fig. 5).



(a)

```
111100000000001111111110000000
111110000000000000111111100000
001111000000000000000111111000
000110000000000000000011111111
000000000000000000000001111111
000000000000000000000001111111
000000000000000000000001111111
000000100000000000000001111111
000001110000000000000001111111
000001110000001100000001111111
000001110000001110000011111111
000011110000001110000111111111
000001100000000110001111111111
000000000000000111111111111100
000000000000000011111111110000
000000000000000001111111100000
000000000000000001111111000000
```

(c)

```
111100000000001111111110000000
111110000000000000111111100000
001111000000000000000111111000
000110000000000000000011111111
000000000000000000000001111111
000000000000000000000001111111
000000000000000000000000011100
000000100000000000000000011100
000001110000000000000001111111
000001111100111100000001111111
000001111100111110000011111111
000011110000001110000111111111
000001100000000110001111111111
000000000000000111111111111100
000000000000000011111111110000
000000000000000001111111100000
000000000000000001111111000000
```

Fig. 5. The remodeling scheme applied to 2D images

## 3. Mesh generator in parallel environment – Parallel-Cosmoprojector

The one-core version of Cosmoprojector software allows one to analyse the bone structure with a volume which does not exceed a few cubic millimetres (Nowak, 2006b). The exact representation of the whole bone with trabecular details is impossible at a single workstation. It is necessary to develop the software which is able to use the parallel environment. The fully functional mesh generator – Cosmoprojector – was adapted to work on computer clusters. The new mesh generator was called Parallel-Cosmoprojector.

The idea of data partitioning is based on the slice order. The Cosmoprojector uses the projection method to create a three-dimensional element. Hence, the simplest method of partitioning assumes to run the software on the part of the data (Fig. 6-④) and the collection of all generated subgrids into one global.

The input data is prepared in the same way as it was presented previously. The total number of slices $l$ is divided by computational cores $n$, the total part of that equation gives the number of slices for each domain $C$

$$C = \left[\frac{l}{n}\right] \tag{3.1}$$

The correct decomposition procedure can be done only when $C$ is the total number – each processor is equally loaded, so the generation process is the most efficient – the proportional case. Otherwise, additional slices are added to the last domain. It is possible that the last processors would be overloaded and the generation process is strongly inefficient then. When

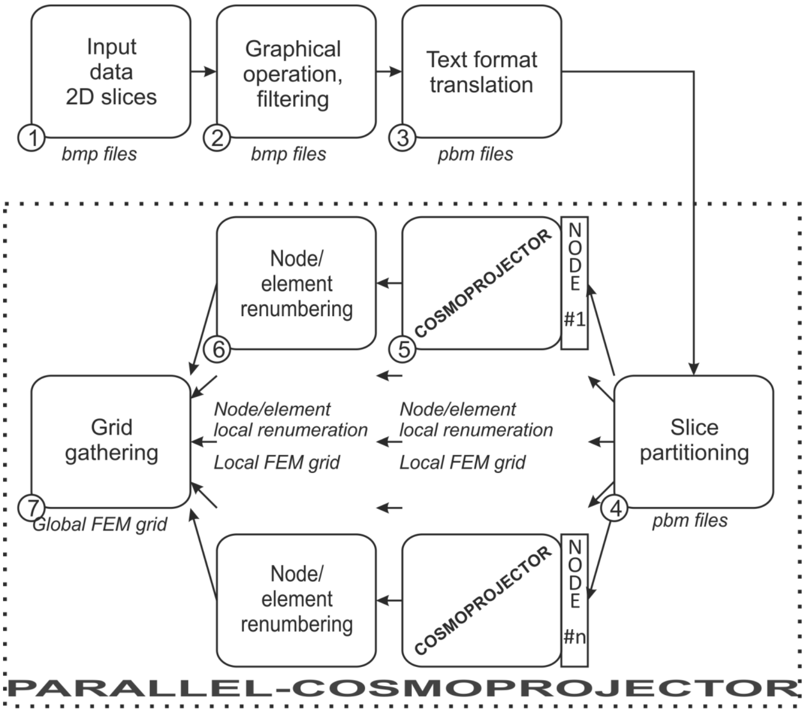


Fig. 6. The parallel mesh generator – Parallel-Cosmoprojector

the above condition is not guaranteed, the correction should be applied in order to ensure proper and efficient mesh generation – the disproportional case.

During the developing process of the software, it was found that it is necessary to add two slices at the beginning and at the end of the domain – the overlaps (Fig. 7). This is the result of applying 2D slices as a base of mesh generation process. The nodes at the first slice are correctly determined when the previous element exists. The overlaps must be removed during the gathering process of all local grids into one global grid.

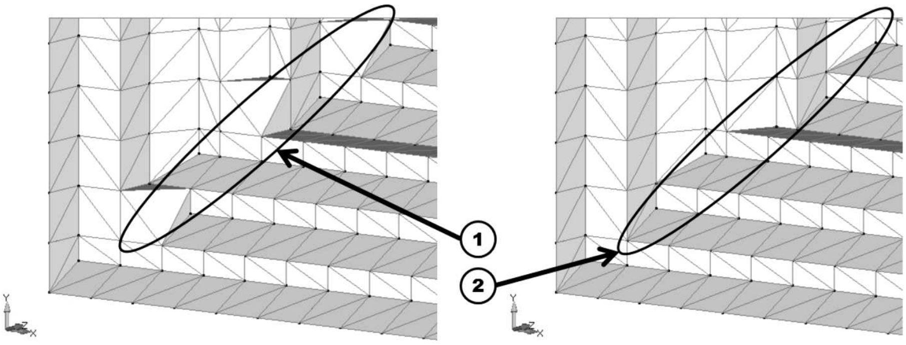


Fig. 7. The grid (1) without overlaps at the boundary of the subdomains causes incorrect geometry representation in opposition to the grid (2) created by assembling of subdomains with the overlaps

Next, the Cosmoprojector is run for each domain parallel (Fig. 6-⑤). There is no communication needed between the processors during the mesh generation for the decomposed space, which is the main advantage of the proposed method. The result of this operation is a set of decomposed domains with local numbering. Additionally, the information about nodes and elements for each domain and overlapping areas is saved at this stage of mesh generation. This data is required for the renumbering procedure (Fig. 6-⑥). Then, all local FEM grids with global numbering are collected into one global FEM grid (in ABAQUS file format). In the figure below (Fig. 8), the adjacent local grids and the result of collecting them in one global mesh are presented (same grid as shown in Fig. 4).

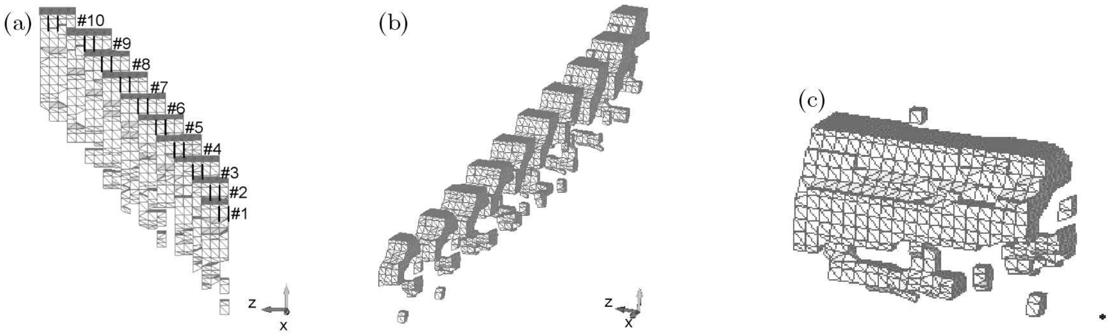


Fig. 8. (a), (b) The local grids with global numbering (marked core of the domains) and (c) the result of grids gathering

## 4. Capabilities of developed software

The Cosmoprojector mesh generator was primarily developed to prepare models for optimization purpose and, therefore, the grid must be suitable for FEM simulations (without degenerated elements). Hence, the output obtained from developed software, Parallel-Cosmoprojector, was used to perform example numerical simulation.

The test model was selected from a rat bone tissue. This data were primarily used in the Miab project (Waarsing, 2004). The information had to be reduced because it was impossible to generate the mesh at one processor for all available data (the restriction for software scalability tests). The resolution of images was fixed at $350 \times 350$ pixels and nodes were generated at every second pixel. The distance between the adjacent slices was fixed at 5 units. The final grid contained 16584963 elements from 160 images. The model contained $3.32 \cdot 10^8$ cubic units (only FEM model, without voids). The resolution of 2D images determined grid size slices which were also spaced at assumed distance. The distance between the slices and nodes on slices should be comparable in order to determine the proper grid.

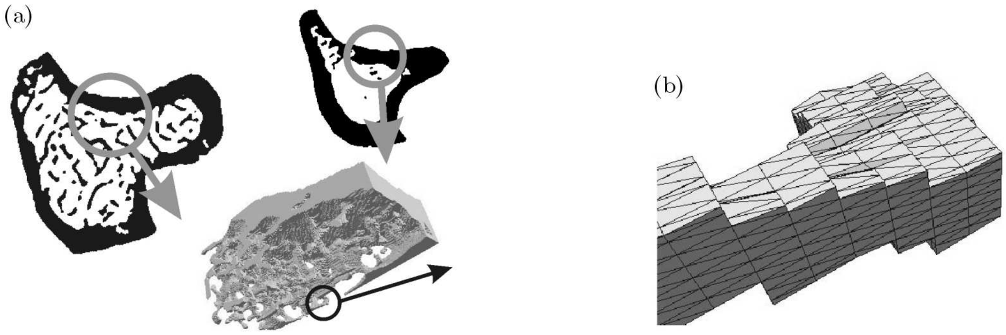


Fig. 9. (a) Rat bone data selected for scalability tests and (b) an enlarged part

For the selected test case, the following boundary condition was applied (Fig. 10a). All nodes from the last plane were fixed and the pressure force was applied to the first plane. Only nodes corresponding to the cortical bone were taken into account. Then the static linear FEM simulation was carried out. The results are presented in Fig. 10b – the Strain Energy Density (SED). The regions with no energy (the value is equal 0) are marked black. It represents the part of the structure which is not submitted to the load or in the second case, it is a trabecular bone not connected to the cortical bone.

This model was used for mesh generation scalability tests.

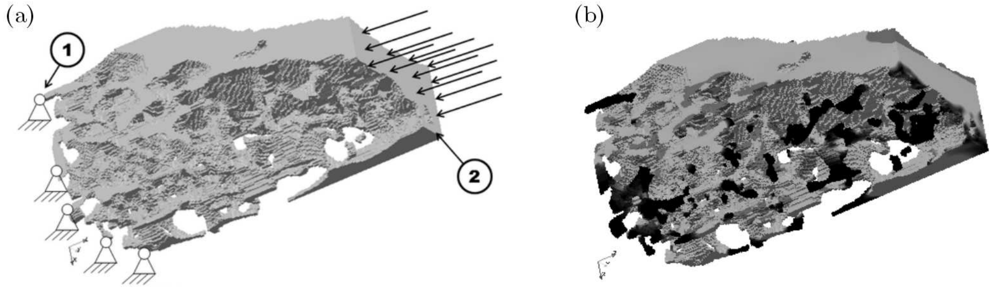


Fig. 10. (a) Boundary conditions applied to the test case (1 – fixed plane, 2 – pressure force applied to cortical bone) and (b) results of FEM simulation (Strain Energy Density distribution: black color – no energy; dark grey – high energy regions)

The FEM grid (Fig. 11) for all available data of the rat bone tissue (166 slices with resolution of $1284 \times 1078$ pixels) contains 68921715 elements with grid spacing of $2 \times 2$ units at slices and 5 units in the slice projection. The mesh was generated with 40 processors.

The application of the Parallel-Cosmoprojector enables FEM simulation of well a reproduced biological structure. The example bone was submitted to the compression load state as shown in Fig. 10a. Hence, the following results have been achieved (Fig. 11). As it was presented in the simple example (Fig. 10), the black color represents regions with no energy and the dark gray with high energy. Such results would be difficult to obtain without the Parallel-Cosmoprojector, so it proved high usability of the developed software.

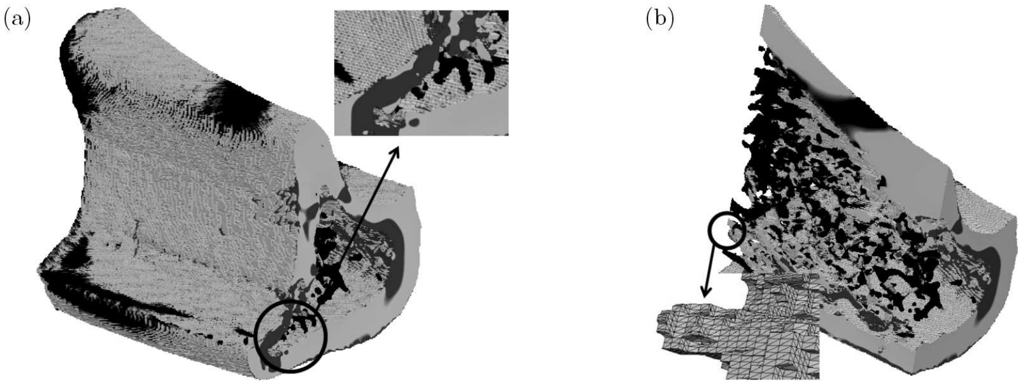


Fig. 11. (a) FEM simulation results of the rat bone model (Strain Energy Density distribution: black color – no energy; dark grey – high energy regions) and (b) the cross-section of the grid with an enlarged part

## 5. Performance of developed software in parallel environment

The performance of the software in the parallel environment is determined by two parameters: speedup – $S_p$ and efficiency – $E_p$. The speedup is defined as (Eager *et al.*, 1989)

$$S_p = \frac{T_1}{T_p} \tag{5.1}$$

and the efficiency (Eager *et al.*, 1989)

$$E_p = \frac{S_p}{p}100\% \tag{5.2}$$

where: $p$ – number of cores (processors) used to generate a grid, $T_1$ – walltime used to carry out the generation at one core, $T_p$ – walltime generation at $p$ processors.

The ideal speed (linear characteristic) is observed when $S_p = p$. The algorithm accelerates twice when the number of processors is doubled. It is considered that the algorithm has a good scalability (Eager *et al.*, 1989).

The efficiency is determined as the proportion between the computation cost spent to solving a problem (generation in this case) and the effort devoted for communication (Eager *et al.*, 1989).

The model for scalability tests was described previously. The computation was carried out at *Zeus*[1] cluster – PL-Grid Infrastructure. The nodes with 2-processors, 4-core each (Intel Xeon L5240 2.5GHz) and 16 GB of RAM were chosen for the tests.

There were two investigations carried out. The first research includes the scalability test for the proportional distribution of slices per domains (with equally loaded domains) and the disproportional distribution (with correction). The second one includes the scalability test for different volumes of the model.

For the first research and for the proportional case, the measurement points were selected at $1, 2, 4, 5, 8, 10, 16, 20, 32, 40, 80$ processors. For the disproportional case, the configuration contains the measurement points (processors) located between the points from the previous case and where the modulus of equation (3.1) is the highest. Therefore, the measurement points are: $1, 3, 7, 9, 15, 18, 23, 27, 33, 54$ processors.

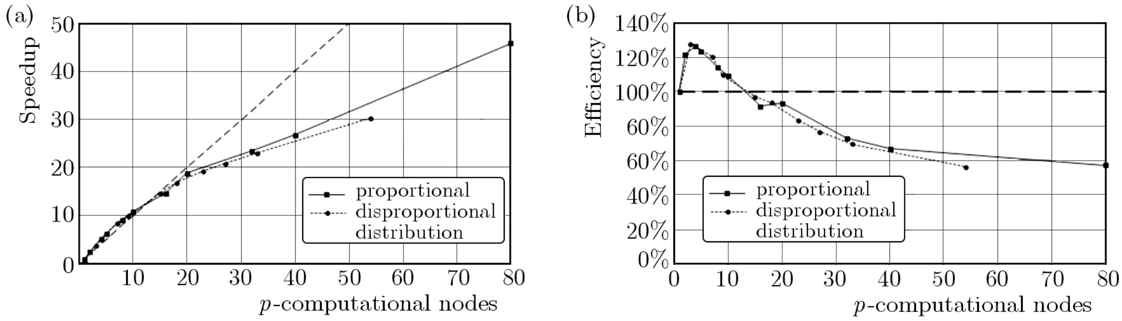


Fig. 12. The results of scalability tests from the first research: (a) speedup and (b) efficiency parameters

The results presented in Figs. 12a and 12b indicate that speedup for selected data reaches maximum at level 50 times. It was achieved for six slices per domain (two slices for the domain-core and four for the overlaps). The software works in the super-linear speedup range when the number of processors used increased from 2 to 10. It is possible due to the proposed method for parallelization software which involves no communication between the processors during generation. For this test case, the efficiency decreases above 10 processors due to the size of subproblems. The time consumed to exchange information during the gathering stage is more distinct than the time spent to generate the local grids. The disproportional distribution yields slightly worse results in comparison with the proportional case. This means that the correction of the partitioning method was correctly applied.

**Table 1.** Grids details for different configuration

| Grid No. | Slices | Element spacing | | | Elements | Volume |
|---|---|---|---|---|---|---|
| | | $x$ | $y$ | $z$ | | |
| 1 | 160 | 2 | 2 | 5 | 16584963 | $3.3170 \cdot 10^8$ |
| 2 | 80 | 4 | 4 | 10 | 2070883 | $3.3134 \cdot 10^8$ |
| 3 | 40 | 8 | 8 | 20 | 251913 | $3.2245 \cdot 10^8$ |

[1] 80th position at Top500 supercomputer list – 104TFlops (June 2011)

For the second research, two additional FEM models were prepared. The grid spacings at slices and between them, as also volume of each model, are presented in Table 1.

The first grid was used in the previous research. The spacing for the rest of configurations was matched at the position which enables one to generate grids with the same volume (without void). The relative error for the first and the second grid equals 0.1% and for the first and the third grid equals 2.7%. The scalability tests was carried out in the proportional distribution of slices per domains. The results are presented in Fig. 13. The greater speedup is achieved for more complex models.

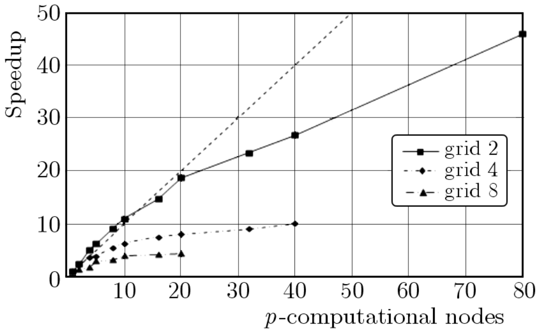


Fig. 13. Mesh generation walltime for the proportional and the disproportional distribution slices per domains

## 6. Summary

In this paper, the finite element mesh generator for biomechanical purposes – Cosmoprojector – has been presented. The method used to parallelization of the software as well as details of grid partitioning were enhanced. The software uses as the input data 2D images which come from micro-CT. The grid prepared by the software is correct for further FEM computations as it has been reported in this article. Hence, Parallel-Cosmoprojector is a universal mesh generator, applicable for biomechanical and technical objects. The necessity of application the overlaps between domains was pointed out in order to obtain a proper mesh. The results of the efficiency test have shown the high scalability of the developed software, which is necessary during testing a complex structure. The research of the worst available partitioning case has not shown a substantial difference in the efficiency, which is an important feature of the software. The Parallel-Cosmoprojector will enable more accurate trabecular bone analysis and optimization of engineering structures and, consequently, it will facilitate the insight into biomechanical problems.

*Acknowledgement*

This work was supported by the Polish National Science Centre under the grant – decision No. DEC-2011/01/B/ST8/06925.

This research was supported in part by PL-Grid Infrastructure.

## References

1. Będziński R., Tyndyk M., 2003, FEM analysis of strain distribution in tiba bone and relationship between strains and adaptation of bone tissue, *Computer Assisted Mechanics and Engineering*, **10**, 3
2. Eager D.L., Zahorjan J., Lazowska E.D., 1989, Speedup versus efficiency in parallel systems, *IEEE Transactions on Computers*, **38**, 408-423
3. Nowak M., 2006a, A generic 3-dimensional system to mimic trabecular bone surface adaptation, *Computer Methods in Biomechanics and Biomechanical Engineering*, **9**, 5, 313-317

4. Nowak M., 2006, Structural optimization system based on trabecular bone surface adaptation, *Journal of Structural and Multidisciplinary Optimization*, **32**, 3, 241-251

5. Nowak M., Morzyński M., Łodygowski T., Wierszycki M., Szajek K., 2010, The hierarchical model of trabecular bone adaptation to simulate the structural evolution of skeleton elements, *37th Solid Mechanics Conference*, 232-232

6. Pedersen P., 2003, *Optimal Design – Structures and Materials – Problems and Tools*, Department of Mechanical Engineering, Solid Mechanics, Denmark

7. Simpleware ltd., 2008, *ScanIP, +ScanFE and +ScanCAD, Tutorial guide*

8. Telega J.J., Gałka A., Tokarzewski S., 1999, Effective moduli of trabecular bone, *Acta of Bioengineering and Biomechanics*, **1**, 1, 53-57

9. Waarsing E., 2004, *Responsible researcher MIAB Project: Mechanical Integrity and Architecture of Bone Relative to Osteoporosis, Ageing and Drug Treatment*, Funding: European Union Fifth Framework Programme, Project Reference: QLK6-CT-1999-02024

*Manuscript received March 5, 2013; accepted for print June 5, 2013*