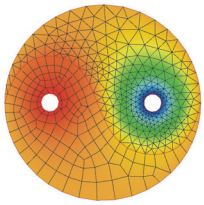
Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE

Informatyka w Technologii Materiałów

Vol. 11, 2011, No. 1



PARALLEL GENETIC ALGORITHM FINDING OPTIMAL INITIAL MESH FOR THREE DIMENSIONAL SELF-ADAPTIVE *hp* FINITE ELEMENT METHOD

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

[1] *Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Reymonta 4, 30-059 Kraków, Poland*
[2] *Department of Computer Science, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Krakow, Poland*
**Corresponding author: anna.paszynska@uj.edu.pl*

Abstract

In this paper we investigate the parallel version of the hierarchical chromosome based genetic algorithm (HCBGA) for finding the optimal initial mesh for self-adaptive hp-Finite Element Method (*hp*-FEM). The HCBGA algorithm solves the global optimization problem of r refinements in order to provide optimal starting mesh for the *hp*-FEM that will fit material data and singularities, and will result in the exponential convergence of the *hp*-FEM. The parallel algorithm is tested on the damaged Step-and-Flash Imprint Lithography problem, modeled by linear elasticity with thermal expansion coefficient.

Key words: genetic algorithms, parallel computing, hp-finite element method

1. INTRODUCTION

The hierarchical chromosome based genetic algorithm (HCBGA) is a genetic algorithm working on the tree structure. Because of this hierarchical representation of the genotypes, the hierarchical genetic operators: hierarchical mutation and hierarchical crossover have to be used to create offspring, in spite of traditional operators. The algorithm, thanks to the hierarchical representation of the objects and hierarchical genetic operators, allows for generating offspring from parents of different size and shape. It allows also for creating offspring with different number of components then their parents. This feature of HCBGA is utilized in many design and engineering problems (Paszyńska & Stożek, 2010; Paszyńska & Paszyński, 2007). We proposed to use the HCBGA for the problem of finding an optimal initial mesh for the self-adaptive *hp*-Finite Element Method (*hp*-FEM) (Paszyński & Demkowicz, 2006, Demkowicz et al., 2007). This procedure is known in the finite element community as *r* refinement and it is a global optimization problem. In this paper the potential for parallelization of HCBGA working with *hp*-FEM is investigated. The case study is the simulation of the Step-and-Flash Imprint Lithography (Colburn et al., 2001; Burns et al., 2004), a modern patterning process utilizing photopolimerization to replicate the template onto the substrate.

ISSN 1641-8581

2. STEP AND FLASH IMPRINT LITHOGRAPHY

The Step and flash imprint lithography (SFIL) is a patterning process utilizing photopolymerization to replicate the topography of a template onto a substrate (Colburn et al., 2001; Burns et al., 2004). The major processing steps of SFIL include (compare left panel in figure 1): depositing a low viscosity, silicon containing, photocurable etch barrier on to a substrate; bringing the template into contact with the etch barrier; curing the etch barrier solution through UV exposure; releasing the template, while leaving high-resolution features behind; a short, halogen break-through etch; and finally an anisotropic oxygen reactive ion etch to yield high aspect ratio, high resolution features.

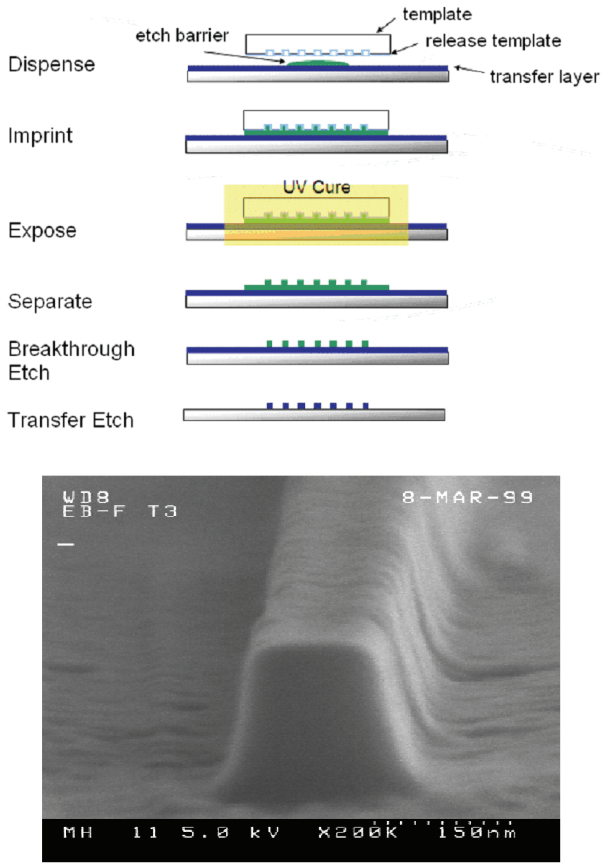


Fig. 1. Processing steps of the Step-and-Flash Imprint Lithography (left panel). Shrinkage of the feature after removal of the template (right panel) (picture obtained from Prof. G. C. Willson from the University of Texas in Austin)

Photopolymerization, however, is often accompanied by densification. The average distance between molecules decreases and causes volumetric contraction. Densification of the SFIL photopolymer (the etch barrier) may affect both the cross sectional shape of the feature and the placement of relief patterns. The exemplary shrinkage of the feature measured after removing the template is presented on right panel in figure 1.

We focus on the simulation of the deformation of the feature after removal of the template. It is assumed that the polymer network has been damaged during the removal of the template, and the interparticle forces are weaker in one part of the 3D domain.

The linear elasticity model with thermal expansion coefficient is used to verify the material response of polymerized networks in cured etch-barrier layers that are formed during the third step of the SFIL process.

Strong formulation

Given $g_i : \Gamma_{D_i} \ni x \to g_i(x) = 0 \in R$, θ and α_{kl} find $u_i : \overline{\Omega} \to R$ the displacement vector field such that

$$\sigma_{ij,j} = 0 \text{ in } \Omega,\ u_i = g_i \text{ in } \Gamma_{D_i} \tag{1}$$

where σ_{ij} is the stress tensor, defined in terms of the generalized Hooke's law

$$\sigma_{ij} = c_{ijkl}\left(\varepsilon_{kl} + \theta\,\alpha_{kl}\right) \tag{2}$$

c_{ijkl} elastic coefficients (known for a given material), ε_{ij}^0 initial strain, θ temperature, α_{kl} thermal expansion coefficients, ε_{ij} is the strain tensor, defined to be $u_{(i,j)}$, the symmetric part of the displacement gradients

$$\varepsilon_{ij} = u_{(i,j)} = \frac{u_{i,j} + u_{j,i}}{2} \tag{3}$$

where u_i displacement vector, $u_{i,j}$ displacement gradients.

Weak formulation

The weak formulation is obtained by multiplying (1) by test functions $w_i \in V_i$ and integrating by parts over Ω.

$$-\int_{\Omega} w_{i,j}\sigma_{ij}\,d\Omega + \int_{\Gamma} w_i \sigma_{ij} n_j\,d\Omega = 0 \tag{4}$$

Since σ_{ij} is symmetric tensor, then $w_{i,j}\sigma_{ij} = w_{(i,j)}\sigma_{ij}$ (compare Hughes, 2000), and $w_i = 0$ on Γ,

$$\int_{\Omega} w_{(i,j)}\sigma_{ij}\,d\Omega = 0 \tag{5}$$

Finally, we substitute (2) into (5) and since $\varepsilon_{ij} = u_{(i,j)}$ we get

$$\int_{\Omega} w_{(i,j)} c_{ijkl} u_{(k,l)}\, d\Omega = -\theta \int_{\Omega} w_{(i,j)} c_{ijkl} \alpha_{kl}\, d\Omega \qquad (6)$$

Abstract index-free notation

For implementation issues, the most convenient is the following form**:** Find $\mathbf{u} \in \mathbf{V}$ such that

$$\mathbf{a}(\mathbf{w},\mathbf{u}) = -\mathbf{A}(\mathbf{w}) \text{ for all } \mathbf{w} \in \mathbf{V} \qquad (7)$$

where

$$\mathbf{a}(\mathbf{w},\mathbf{u}) = \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w})^{\mathrm{T}} \mathbf{D} \boldsymbol{\varepsilon}(\mathbf{u})\, d\Omega \quad \mathbf{A}(\mathbf{w}) = \theta \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w})^{\mathrm{T}} \mathbf{D} \boldsymbol{\alpha}\, d\Omega \qquad (8)$$

with (Hughes, 2000)

$$\boldsymbol{\varepsilon}(\mathbf{z}) = \begin{Bmatrix} z_{1,1} \\ z_{2,2} \\ z_{3,3} \\ z_{2,3} + z_{3,2} \\ z_{1,3} + z_{3,1} \\ z_{1,2} + z_{2,1} \end{Bmatrix}, \qquad \boldsymbol{\alpha} = \begin{Bmatrix} \alpha \\ \alpha \\ \alpha \\ 0 \\ 0 \\ 0 \end{Bmatrix},$$

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \qquad (9)$$

Here $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$, $\mu = \frac{E}{2(1+\nu)}$ are Lame coefficients. Following (Colburn et al., 2001) the Young modulus and Poisson ratio are assumed as E = 1 GPA and ν = 0.3, θ = 1 and α = -0.06115 has been obtained by inverse analysis (Paszyński et al., 2007). The material damage that is supposed to result in slight lean of the feature is modeled by decreasing the value of the Young modulus to E = 0.001 GPA in the subdomain

$$\Omega_d = \{\mathbf{x} = (x,y,z) : x \in [0,0.433],$$
$$y \in [0,1],\ z \in [0.157,0.471]\} \subset \Omega \qquad (10)$$

where the subdomain is prescribed in the relative $\Omega = [0,1]^3$ local system of coordinates. It should be emphasized that the value of the decreased Young modulus has been selected arbitrary, just to test the genetic algorithm, and the future work should involve the investigation of the proper value of the Young modulus in the damaged area.

3. PARALLEL SELF-ADAPTIVE HP FINITE ELEMENT METHOD

Parallel self-adaptive *hp*-Finite Element Method (Paszyński & Demkowicz, 2006) was obtained by applying the domain decomposition paradigm to the sequential *hp*-FEM developed by the group of Leszek Demkowicz (Demkowicz et al., 2007). The parallel *hp*-FEM starts from an initial mesh selected by the user. It generates in fully automatic mode a sequence of meshes, by performing *h*, *p* or *hp* refinements on selected edges, faces and interiors. In the parallel *hp*-FEM the computational mesh is stored in distributed manner, compare figure 2. The selection of refinements is a complicated and computationally expensive process, utilizing the coarse and corresponding fine meshes, which is described in details in Demkowicz et al. (2007). If the initial mesh fits geometry, material data and singularities of the problem, the generated sequence of meshes delivers exponential convergence of the numerical error with respect to the mesh size.

However, if the initial mesh doesn't fit the material data and singularities, e.g. a finite element is too large to capture the local jump of material data, then the *hp*-FEM may not deliver the exponential convergence. When the material data are provided by MRI scan or from some stochastic simulations, it may be difficult to design optimal initial mesh by hand. The fitting of the initial mesh to the material data and singularities is a global optimization problem. Thus it is reasonable to apply a genetic algorithm to find an optimal initial mesh. Table 1 illustrates the lack of exponential convergence for the exemplary damaged SFIL problem when the *hp*-FEM starts from a randomly selected initial mesh, and the presence of the exponential convergence when the *hp*-FEM starts from an optimal initial mesh generated by the genetic algorithm.

The parallel *hp*-FEM works based on the domain decomposition provided by ZOLTAN library (ZOLTAN, 2010). For the SFIL problem it is reasonable to utilize 8 processor, due to number of initial elements in the mesh. The initial mesh can be efficiently processed by using maximum 8 number of processors. However, if we perform *hp* refinements over the mesh, the computational cost changes from one element to the other, compare figure 3, and the uniform load balancing is no longer

Table 1. *Comparison of convergence of the hp-FEM for two initial meshes.*

Poor convergence of the *hp*-FEM from randomly selected initial mesh:					
coarse/fine mesh sizes	729 15625	869 19019	1166 25215	1322 28681	1573 35557
relative error	26.58	19.22	16.16	14.47	14.09
Exponential convergence of the *hp*-FEM from an optimal initial mesh generated by the genetic algorithm:					
coarse/fine mesh sizes	729 15625	932 19399	1306 24839	1817 38211	Relative error formulae
relative error	12.93	9.40	7.47	4.01	$100 * \left\|u_{hp} - U_{hp}\right\|^2_{H^1(\Omega)} \Big/ \left\|U_{hp}\right\|^2_{H^1(\Omega)}$

possible. This is because computational cost of processing interior of elements is of the order of $p_x^3\ p_y^3\ p_z^3$ where p_x, p_y, p_z are the polynomial orders of approximation at element interior in x, y and z directions. For example, in the third iteration there is a single element with $p_x = 3$, $p_y = 4$, $p_z = 5$ so the computational cost of processing this element is of the order of $3^3*4^3*5^3 = 216000$, compare the first panel in figure 3. It implies that the computational cost of processing this single element is several orders of magnitude higher than the cost of processing all other elements, and it makes no sense to use 8 processors. Actually ZOLTAN suggests to utilize 3 processors only, compare third panel in figure 2. Similarly, in the fourth iteration of the parallel *hp*-FEM we can effectively use only 6 and in the fifth iteration only 4 processors.

4. PARALLEL HIERARCHICAL CHROMOSOME BASED GENETIC ALGORITHM FOR INITIAL MESH SELECTION

We propose to use hierarchical chromosome based genetic algorithm to find the optimal initial three dimensional mesh. The chromosome (the tree based structure) which represents an object in our genetic algorithm, codes the structure and the meaning of an object, from the point of view of the designer. Figure 4 presents hierarchical chromosome and an exemplary mesh it codes. Our exemplary mesh (denoted by S1) is described by tree components: X, Y and Z. Components X, Y and Z consist of two, one and tree subcomponents, suitable. Each subcomponent is described by two parameters: the begin and the end limit of the interval, denoted by x_i, y_i and z_i, suitable. These values are coded as the binary strings ($x_i = b_{x0}$, b_{x1}, ...,b_{xm}), ($y_i = b_{y0}$, b_{y1}, ...b_{ym}) and ($z_i = b_{z0}$, b_{z1},...,b_{zm}). The process of finding of the optimal initial three dimensional mesh can be described as follows. First the initial population is randomly created. We assume that each mesh from the initial population has the same number of subcomponents of type X, Y and Z and that the suitable coordinates are the same. But the overall size and shape of the hierarchical chromosomes coding initial meshes can be different. The next step is the evaluation of the newly created initial population. For each individual of the population the computational problem is solved, the global *hp* refinememnt is performed, the problem is solved again, and the relative error estimator for the individual is computed. It is done fully in parallel, all individuals at the same time.

The next step is the generation of offspring, by the means of hierarchical genetic operators. The whole process of evaluation and generation of offspring is repeated, until an optimal initial mesh is not found. For offspring generation the hierarchical genetic operators are used: mutation and crossover. Two kinds of hierarchical mutation are used: the mutation of group of alleles and the mutation of alleles. The hierarchical mutation of group of alleles (coded components) allows to add or to remove "splitting" coordinates from the genotype. This genetic operator is responsible for changing of the genotypes size. The process of removing of splitting coordinates consists in randomly choosing the component and the gene on the third level of the hierarchy tree from this component and removing it from the chromosome. The process of adding the splitting coordinates consists in randomly choosing the component and the splitting coordinate and adding chosen coordinate to the component at the third level of the hierarchy tree (compare left panel of figure 4 – splitting coordinates x_6, x_5 and x_4). If the component

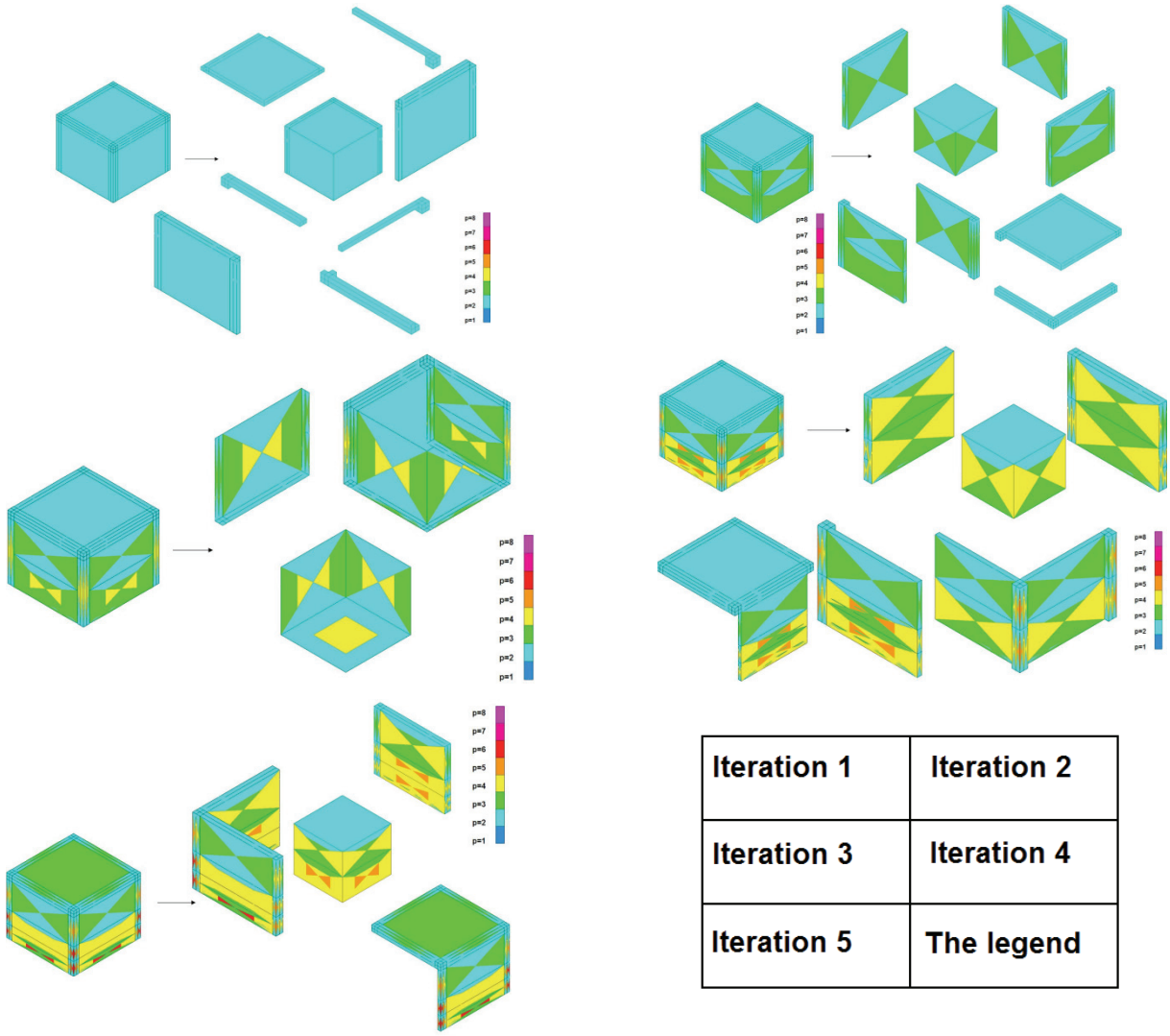


Fig. 2. *Domain decomposition into 8 processors provided by ZOLTAN library in particular generation of the self-adaptive hp-FEM for an exemplary initial mesh.*

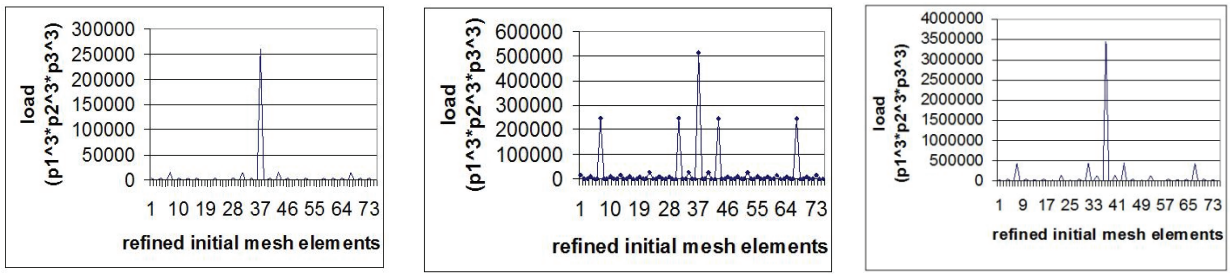


Fig. 3. *Load inbalances for the third, fourth and fifth iterations.*

has not been split yet, the coordinate should be chosen between the beginning and the ending coordinate of the component. If the component has been split before, the coordinate should be chosen between randomly chosen gene in the component and the next gene in the component. The mutation of alleles simply changes the value of randomly chosen allele from the third level of hierarchy tree. The new value for chosen allele must be a value between the neighboring coordinates. The next used genetic operator is the hierarchical crossover. The hierarchical crossover consists in finding the suitable crossover point P in the parent individuals and applying the crossover to the offspring in P. To apply crossover to parents in point P means to copy components from parents to offspring in the following way. First all the components located left from P and their subtrees are copied to the children being generated - respectively, from the first parent to the first child, from the second parent to the second child. After that the components of number P with their subtrees and all primitives located right from P and their sub-

trees are copied to the children - from the first parent to the second child, from the second parent to the first child. Thanks to such a definition of the cross-over, the children being generated can have different size and shape then their parents. We refer to (Paszyńska & Stożek, 2010; Paszyńska & Paszyński, 2007) for more details.

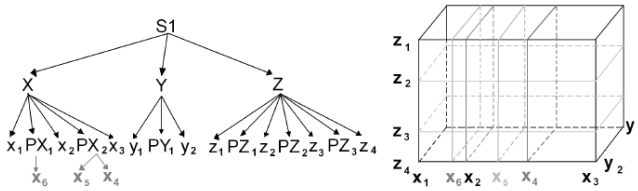


Fig. 4. The hierarchical chromosome and an exemplary mesh that it codes.

5. NUMERICAL RESULTS FOR PARALLEL GENETIC ALGORITHM

The HCBGA algorithm was executed from initial populations with 10 randomly selected individuals and with initial populations with 20 randomly selected individuals. Some of them are presented in figure 5. The fitness function was defined as

$$\text{fitness} = 1 - \text{relative error} = 1 - \frac{\left\| u_{hp} - U_{hp} \right\|^2_{H^1(\Omega)}}{\left\| U_{hp} \right\|^2_{H^1(\Omega)}}$$

where $\left\| u_{hp} - U_{hp} \right\|^2_{H^1(\Omega)} / \left\| U_{hp} \right\|^2_{H^1(\Omega)}$ is the normalized relative error (squared) of the solution over the initial mesh, with respect to the fine mesh (reference mesh) solution, measured in the energy norm.

The resulting convergence of the HCBGA is presented in figure 6. The most expensive part of the HCBGA algorithm is the fitness function estimation, and this can be performed fully in parallel. Each individual (mesh) can be assigned to separate processor. However, there is no sense to use more then 10 individuals, since the convergence of the HCBGA starting from 20 individuals is the same as for HCBGA starting from 10 individuals. It order to utilize more than 10 processors, we need to partition individuals (computational meshes) into subdomains. The fitness function estimations use always the first iteration of the *hp*-FEM, so we can efficiently use 8 processors for each individual, compare figure 2. Thus we can efficiently utilize 80 processors. All meshes from the final fifth iteration have fitness of the order of 0.85-0.87, which implies

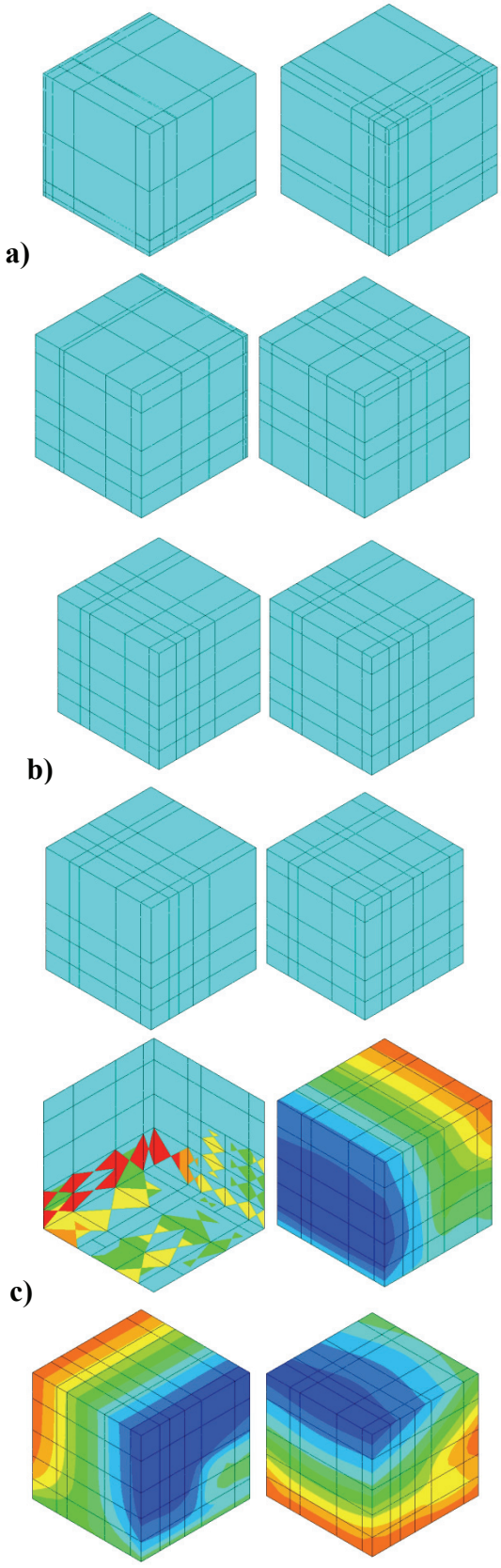


Fig. 5. a) Four exemplary individuals from the first population b) Four exemplary individuals from the fifth population c) hp-refined optimal mesh obtained based on the fifth population (rear view), with solution: x, y, and z components of the displacement vector field.

relative error of the order of 12-14 %. To improve the quality of the solution further, the parallel self-adaptive *hp*-FEM can be executed, as illustrated in figure 2. The *hp*-FEM will provide now the exponential convergence untile the required accuracy of 5% relative error, compare table 1. The exemplary final *hp*-refined mesh and the solution are presented on c) panel in figure 5.

6. CONCLUSIONS

In this paper we discussed the parallel version of HCBGA algorithm finding optimal initial mesh for self-adaptive *hp*-FEM computations. The algorithm was verified on the simulations of the damaged SFIL problem. It has been proved that the optimal mesh found by the parallel HCBGA delivers exponential convergence of *hp*-FEM. The enough number of individuals for the case study damaged SFIL problem was equal to 10, and the fitness function estimations used always the first iteration of the *hp*-FEM, where we can partition each mesh into 8 subdomains. It implies that we can efficiently use maximum 80 processors for the case study problem.

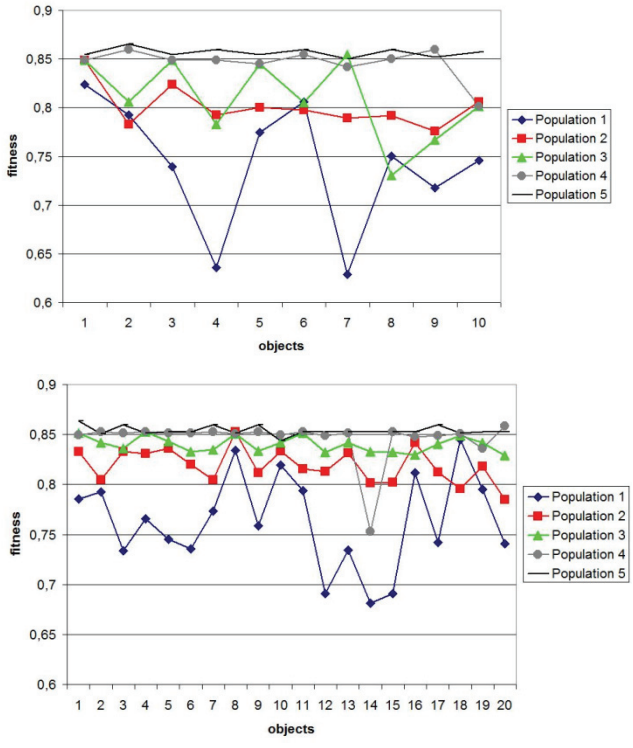


Fig. 6. *Fitness for all objects from 5 consequtive populations starting with 10 individuals (left panel) or 20 individuals (right panel).*

ACKNOWLEDGEMENTS

The support of Polish MNiSW grant no NN 501 120 836 is gratefully acknowledged.

REFERENCES

Paszyńska, A., Stożek, M., 2010, Application of a Hierarchical Chromosome Based Genetic Algorithm to the Problem of Finding Optimal Initial Three Dimensional Meshes for the Self-Adaptive HP-FEM, submitted to *International Conference on Computatioanl Science 2010*, Tsukuba, Japan.

Paszyńska, A., Paszyński, M., 2007, An application of hierarchical chromosome based genetic algorithm to the optimization of the platform shape, *Prace Naukowe, Politechnika Warszawska, Elektronika*, *Proceedings of Evolutionary Computation and Global Optimization* Conference, June 11-13, Będlewo, 225-232.

Paszyński, M., Demkowicz, L., 2006, Parallel Fully Automatic hp-Adaptive 3D Finite Element Package, *Engineering with Computers,* 22, 3-4, 255-276.

Demkowicz, L., Rachowicz, W., Pardo D., Paszyński M., Kurtz J., Zdunek A., 2007, *Computing with hp-Finite Elements. Volume II*, Chapmann & Hall / CRC Press.

Colburn, M. E., Suez, I., Choi, B. J., Meissi, M., Bailey, T., Sreenivasan S. V., Ekerdt J. E., Willson C. G., 2001, Characterization and modeling of volumetric and mechanical properties for SFIL photopolymers, *Journal of Vacuum Science and Technology* B, 19-6, 2685-2689.

Burns, R. L., Johnson, S. C., Schmid, G. M., Kim, E. K., Dickey, D. M. D., Meiring, J., Burns, S. D., Stacey, N. A., Willson, C. G., 2004, Mesoscale modeling for SFIL simulating polymerization kinetics and densification, *Proceeding of SPIE* 5374, 348-360.

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

Paszyński, M., Barabasz, B., Schaefer, R., 2007, Efficient adaptive strategy for solving inverse problems, *Lecture Notes in Computer Science* 4488, 342-349

ZOLTAN, 2010, Data-Management Services for Parallel Applications, http://www.cs.sandia.gov/Zoltan.

RÓWNOLEGŁY ALGORYTM GENETYCZNY ZNAJDUJĄCY OPTYMALNE SIATKI POCZĄTKOWE DLA TRÓJWYMIAROWEJ *hp* ADAPTACYJNEJ METODY ELEMENTÓW SKOŃCZONYCH

Streszczenie

W artykule tym przedstawiono wersję równoległą algorytmu genetycznego bazującego na hierarchicznym chromosomie (AGBHC) służącego do znajdowania optymalnych siatek początkowych dla *hp* adaptacyjnej metody elementów skończonych (*hp*-MES). Zaproponowany algorytm AGBHC rozwiązuje problem *r* adaptacji. Jest to problem globalnej optymalizacji polegający na znalezieniu optymalnej siatki początkowej dla algorytmu automatycznej *hp* adaptacji. Poszukiwana siatka początkowa powinna pasować do przyjętych stałych materiałowych oraz do osobliwości rozwiązania. W efekcie algorytm automatycznej *hp* adaptacji uruchomiony na takiej optymalnej siatce początkowej powinien dostarczyć eksponencjalną zbieżność dokładności rozwiązania względem rozmiaru siatki obliczeniowej. Algorytm równoległy testowany jest na problemie nanolitografii poprzez wyciskanie i naświetlanie, modelowanym z pomocą liniowej sprężystości ze współczynnikiem rozszerzalności cieplnej.

Received: August 14, 2010
Received in a revised form: October 17, 2010
Accepted: November 9, 2010