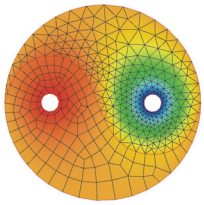
Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE

Informatyka w Technologii Materiałów

Vol. 10, 2010, No. 1



DIFFERENT MESH MANAGEMENT STRATEGIES FOR NON-STATIONARY SELF-ADAPTIVE *hp* FINITE ELEMENT METHOD

MACIEJ PASZYŃSKI[1*], PAWEŁ MATUSZYK[2]

[1] Department of Computer Science, AGH University of Science and Technology Al. Mickiewicza 30, 30-059 Cracow, Poland
[2] Department of Applied Computer Science and Modelling, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Cracow, Poland
**Corresponding Author: paszynsk@agh.edu.pl*

Abstract

In this paper we compare different strategies for solving non-stationary heat and mass transfer problems, utilized to simulate austenite-ferrite phase transformation. The self-adaptive *hp* Finite Element Method (hp-FEM) is utilized to solve the heat and mass transfer problems at every time step. The *hp*-FEM generates a sequence of hp refined meshes delivering exponential convergence of the numerical error with respect to mesh size. To solve the computational problem at every time step, the *hp*-FEM utilizes the numerical solution from previous time step. We compare three different techniques for utilization of the previous time step solution. The first one generates a new mesh for the next time step starting from the regular initial mesh, the second one utilizes previous time step computational mesh, and the third one performs some unrefinements on the mesh from the previous time step before solving the actual time step. The comparison is based on the numerical simulation of the austenite-ferrite phase transition phenomena.

Key words: Finite Element Method, *hp* adaptivity, non-stationary problems

1. THE SELF-ADAPTIVE *hp*-FEM

The non-stationary Finite Element Method (FEM) is widely utilized to solve engineering problems that belongs to the class of parabolic problems. The time discretization utilized within the classical non-stationary FEM is quite simple, since computational meshes from the actual and the previous time steps are the same. Some examples with the phase transition simulations include Zhao and Heinrich (2001), Gandin and Rappaz (1996), Rappaz et al. (1996). In this paper, we consider the adaptive FEM, where the structure of computational meshes change from one time step to the other, and special projections and mesh compatibility techniques must be developed. The following mesh adaptation techniques can be listed:

a) *r adaptation,* where location of finite element nodes is changed, either by a re-meshing procedure or shift of mesh nodes.
b) *Uniform h adaptation*, where all finite elements are uniformly broken into smaller elements.
c) *Uniform p adaptation*, where polynomial order of approximation is increased uniformly over the entire mesh, e.g. by adding bubble shape functions of the higher orders over element edges and interiors.
d) *Non-uniform h adaptation*, where some finite elements are broken into smaller elements, only in parts of the mesh with high numerical error.

ISSN 1641-8581

e) *Non-uniform hp adaptation*, where some finite elements are broken into smaller elements, and polynomial orders of approximation are increased over some finite elements, only in these parts of the mesh where numerical error is high.

The *h, p* and *hp* adaptation techniques start from a selected initial mesh and improve accuracy of the solution by performing a sequence of *h, p* or *hp* refinements. The non-uniform *hp* adaptation, also called the self-adaptive *hp*-FEM, is the most sophisticated strategy (Demkowicz, 2006), in the sense it provides an exponential convergence of the numerical error with respect to the mesh size.

The *r* adaptation is a completely different technique, where mesh quality is improved by some kind of re-meshing technique, without reference to the selected initial mesh. An example of the *r adaptation* technique utilized for the phase transition phenomena simulation can be found in Narski and Picasso (2007). In the *r* adaptation, a computational mesh for the new time step is not compatible with the previous time step mesh, and the projection of the previous time step solution into the new time step mesh must utilized.

The *r* adaptivity re-meshing computational techniques are computationally expensive. Thus, we focus on the *hp* adaptive methodology, starting from a prescribed initial mesh and utilizing computational mesh from the previous time step.

The self-adaptive *hp* Finite Element Method (*hp*-FEM) for two and three dimensional elliptic and Maxwell problems were designed and implemented (Demkowicz, 2006; Demkowicz et al., 2007; Demkowicz et al., 2002; Paszyński et al., 2006). The *hp*-FEM generates a sequence of meshes delivering exponential convergence of the numerical error with respect to the mesh size. Let us focus on the two dimensional code.

The self-adaptive *hp* FEM starts from arbitrary initial mesh, selected by the user. It solves the problem on the initial mesh, called the coarse mesh, and performs global *hp* refinement to obtain so called fine mesh. The global *hp* refinement consists in breaking each finite element from the coarse mesh into 4 new smaller elements, and increasing polynomial order of approximation by one, at every element edge and interior. The problem is solved again on the fine mesh.

The H1 Sobolev space norm (called the "energy norm") of the difference

$$relative\_error = \frac{\left\| u_{hp} - u_{h/2,p+1} \right\|_{H^1}}{\left\| u_{h/2,p+1} \right\|_{H^1}} \qquad (1)$$

between the coarse u_{hp} and fine $u_{h/2,p+1}$ mesh solutions is computed to estimate the relative error of the coarse mesh solution. This estimation is utilized to select optimal refinement of selected coarse mesh elements. Some coarse mesh elements are *h* refined (these elements are broken in either horizontal, vertical or both directions into new smaller elements) or *p* refined (polynomial order of approximation is changed on selected element edges or interiors) or both. Finally, the optimal mesh is obtained by performing selected refinements on the coarse mesh.

The exemplary coarse, fine and optimal meshes from the first iteration are presented in figure 1. The optimal mesh obtained by performing selected *h* or *p* or *hp* refinements becomes the coarse mesh for the next iteration, and the entire process is repeated (Demkowicz, 2006; Demkowicz et al., 2007). The stopping criterion is to obtain required accuracy of the entire coarse mesh solution:

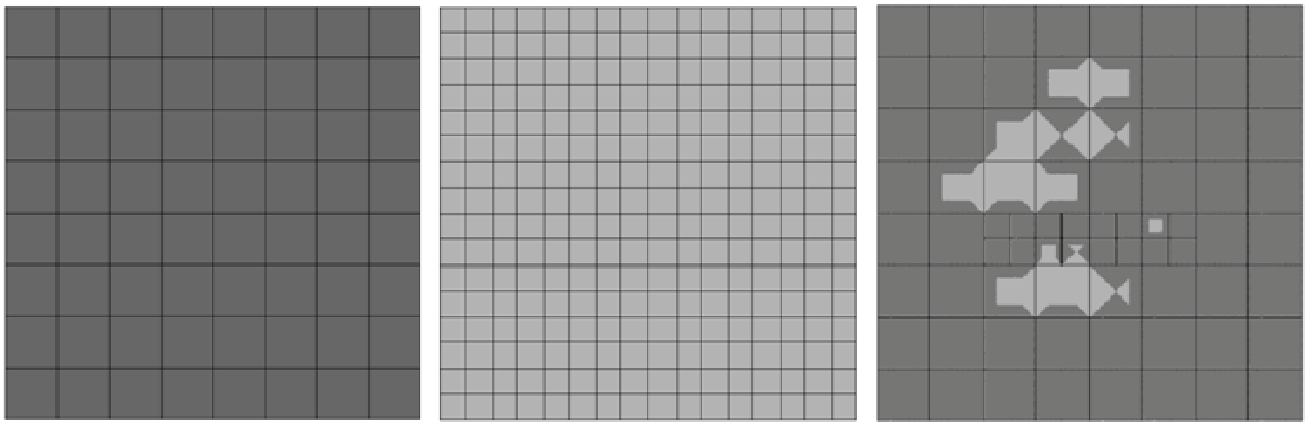


Fig. 1. *First panel: Exemplary regular initial mesh (called the coarse mesh). Second panel: The fine mesh obtained by performing global hp refinement. Third panel: Optimal mesh obtained after the first iteration. Different degrees of gray denote different polynomial orders of approximation utilized on rectangular finite element edges and interiors.*

$$\frac{\left\| u_{h/2^{p+1}} - u_{hp} \right\|_{\Omega,1}}{\left\| u_{h/2^{p+1}} \right\|_{\Omega,1}} < \varepsilon \qquad (2)$$

2. THE SELECTION OF OPTIMAL REFINEMENTS

The fine mesh solution $u_{h/2,p+1}$ is used to mark the optimal refinements for the coarse mesh in order to produce the next optimal mesh. For each element from the coarse mesh we consider different refinement strategies. An element can be either *h* refined in 3 possible ways (broken in horizontal or vertical direction, or both), or *p* refined (the polynomial orders of approximation can be changed on selected element edges as well as in element interior) or both. The fine mesh solution $u_{h/2,p+1}$ is projected onto a nested sequence of meshes that is locally embedded into the fine mesh by using the projection-based interpolation technique (Demkowicz, 2004). For each coarse mesh element the sequence is dynamically constructed by testing possible types of local refinements. The maximum possible refinement that can be considered in a given iteration corresponds to the local restriction of the fine mesh solution (the element broken into four son elements, and all element polynomial orders of approximation are increased by one). The refinement which provides the maximum error decrease rate

$$\text{rate}(w) = \frac{\left\| u_{h/2,p+1} - u_{h,p} \right\|_{1,K} - \left\| u_{h/2,p+1} - w \right\|_{1,K}}{\Delta \text{nrdof}} \qquad (3)$$

is selected for each considered coarse mesh element *K*. In this formula *w* denotes the projection of the fine mesh solution onto the considered coarse mesh refinement, where the projection is computed in H^1 norm on the considered coarse mesh element *K*. The Δnrdof denotes an increase in the number of degrees of freedom or the coarse mesh element, resulting from the execution of the considered mesh refinement strategy. This can be summarized in the following algorithm

```
do loop over coarse mesh elements K
   rate_max = 0
     do loop over considered refinement
       strategies w on element K
       rate_max = max (rate(w),rate_max)
     enddo
     execute strategy w corresponding
     to rate_max
     over element K
enddo
```

3. THE EXTENSION OF THE SELF-ADAPTIVE *hp*-FEM TO NON-STATIONARY PROBLEMS

Recently, the two dimensional strategy has been extended to support non-stationary simulations (Matuszyk & Paszyński, 2007a, 2007b) of heat transfer problem. The non-stationary code has been also interfaced with Cellular Automata (CA) to simulate the austenite-ferrite phase transition phenomena (Paszyński et al., 2008). The phase transition simulation requires implementation of the non-stationary heat and mass transport problems.

There are many phase transition models considered in the literature, some of them focus on the heat transport problem only (Zhao & Heinrich, 2001), the mass transport only (Jacot & Rappaz, 2002), or combined heat and mass transport problems (Zhu & Stefanescu, 2007; Liu et al., 2006). More sophisticated models may also include the fluid flow (Narski & Picasso, 2007). The interface between the phases can be captured by using the front tracking technique (Narski & Picasso, 2007), by decoupling solid and fluid phases and introducing some interface conditions (Zhao & Heinrich, 2001; Jacot & Rappaz, 2002; Zhu & Stefanescu, 2007), or by utilizing the Cellular Automata (CA) technique (Liu et al., 2006). We utilize the CA simulation (Paszyński et al., 2008).

The heat transfer problem is formulated in the following way: Find $R^2 \supset \Omega \ni x \to T(x) \in R$ the temperature scalar field such that

$$\begin{cases} \rho c_p \dfrac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = f & on\ \Omega \times I \\ k\mathbf{n} \cdot \nabla T = \beta (T_N - T) & on\ \partial\Omega \times I \\ T(\mathbf{x},0) = T_o & on\ \Omega \end{cases} \qquad (4)$$

here heat transfer coefficient k, density ρ and specific heat c_p are the material data provided by the CA simulations (Paszyński et al., 2008). Also, the generated heat represented by the right hand side term f is defined by means of CA (Paszyński et al., 2008). The initial condition involves prescribed intial tem-

perature distribution T_0, and the Cauchy boundary is defined on the boundary.

The mass transfer problem is formulated in the following way:

Find $R^2 \supset \Omega \ni x \to c(x) \in R$ the concentration distribution scalar field such that

$$\begin{cases} \dfrac{\partial c}{\partial t} - \nabla \cdot (D \nabla c) = 0 & on\ \Omega \times I \\ \mathbf{n} \cdot \nabla c = 0 & on\ \partial\Omega \times I \\ c(\mathbf{x},0) = c_o & on\ \Omega \end{cases} \quad (5)$$

here the diffusion coefficient D is prescribed by the CA simulations (Paszyński et al., 2008). The initial condition involves prescribed intial concentration distribution field c_0, and the zero Neumann boundary is defined on the boundary. The variational formulations utilized by the non-stationary *hp*-FEM are the following:

Find the temperature distribution $T \in T_D + V$ as well as concentration distribution $c \in c_D + V$ scalar fields satisfying

$$\left(\rho c_p \dot{T}, v\right)_\Omega + \int_\Omega k \nabla T \circ \nabla v \, d\Omega + \int_{\Gamma_N} \beta T v \, d\Gamma_N = \int_\Omega f v \, d\Omega + \int_{\Gamma_N} \beta u_N v \, d\Gamma_N \quad \forall v \in V \quad (6)$$

$$\left(\rho c_p T(0), v\right)_\Omega = \left(\rho c_p T_0, v\right)_\Omega \quad \forall v \in V \quad (7)$$

and

$$\left(\dot{c}, v\right)_\Omega + \int_\Omega D \nabla c \circ \nabla v \, d\Omega = 0 \quad \forall v \in V \quad (8)$$

$$\left(c(0), v\right)_\Omega = \left(c_0, v\right)_\Omega \quad \forall v \in V \quad (9)$$

where $V = H^1(\Omega)$ since the Dirichlet boundary is empty.

The necessary semi-discretization implies the following matrix system for both problems

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}. \quad (10)$$

Applying the trapezoidal rule with respect to the time derivative we obtain

$$(\mathbf{M} + \alpha\delta\mathbf{K})\mathbf{u}^{k+1} = [\mathbf{M} - (1-\alpha)\delta\mathbf{K}]\mathbf{u}^k + \delta\mathbf{f}^k \quad (11)$$

where $\mathbf{M}$ is the mass matrix, δ is the time step, $\alpha \in [0,1]$ gives different time integration schemes.

The general non-stationary *hp*-FEM algorithm for either heat or mass transfer problems can be summarized in the following way:

```
generate initial_mesh with initial
     temperature and concentration
     fields distributions
do time_step = 1, nr_time_steps
     generate coarse_mesh for actual
     time_step
     do iteration of self-adaptive hp-
        FEM for time_step
        project previous time_step
        solution into coarse_mesh
        solve the coarse_mesh problem
        generate the fine_mesh
        project previous time_step
        solution into fine_mesh
        solve the fine_mesh problem
        make decision about optimal
        refinements
        generate optimal_mesh
        coarse_mesh = optimal_mesh
     enddo
enddo
```

The self adaptive *hp*-FEM is utilized at every time step to generate the optimal mesh for the actual time step solution. However, the highly non-uniform optimal mesh generated for a given time step usually is no longer optimal for the next time step. It is necessary to generate a new optimal mesh.

From the discretization scheme presented in Eq. (11) follows that *k+1* time step solution $\mathbf{u}^{k+1}$ requires the previous *k* time step solution $\mathbf{u}^k$. However, the previous time step solution has been obtained on non-compatible finite element mesh, and the problem of projecting the solution from the previous time step into the actual time step mesh must be considered.

In the first time step, the projections of the previous time step solution utilize prescribed initial distributions of the temperature and concentration fields.

The generation of the coarse mesh for every time step can be done in many ways. We can restart computations from initial mesh, reutilize optimal mesh from the previous time step, or perform several unrefinements on the optimal mesh before reruning computations for new time step. In this paper, we investigate such a three mesh reutilization strategies.

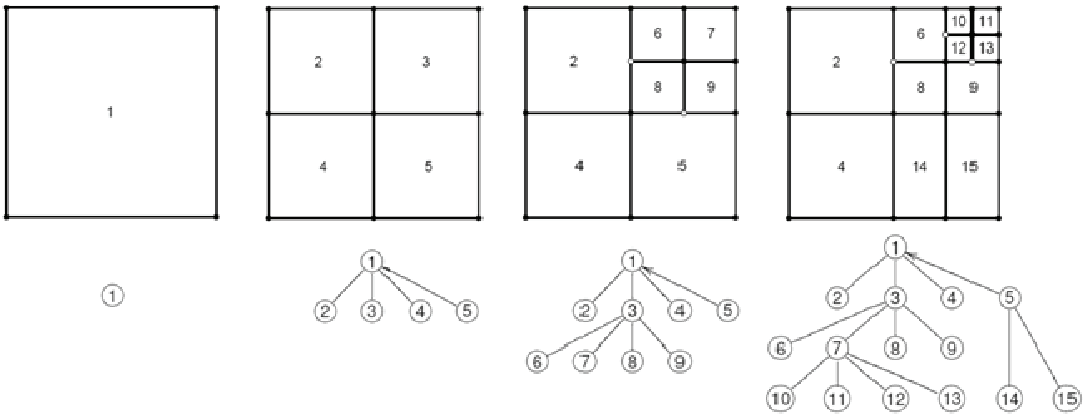


Fig. 2. Top panel: Exemplary sequence of meshes obtained by performing h refinements from the regular single finite element initial mesh. Bottom panel: Tree like structure for storing the history of refinements.

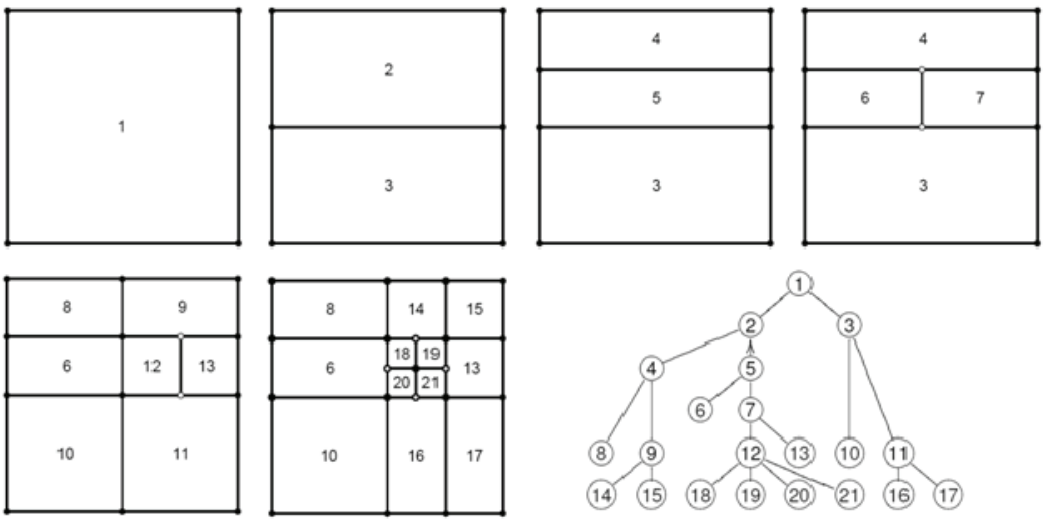


Fig. 3. Exemplary sequence of refinements generating the optimal mesh for the next time step, together with its refinement tree.

4. THE TREE LIKE STRUCTURE FOR STORING COMPUTATIONAL MESHES

The optimal mesh for a given time step is obtained by a sequence of *h* and *p* refinements from the regular initial mesh. The exemplary sequence of meshes is presented on the top panel in figure 2. The computational mesh is stored in the tree like structure presented on the bottom panel in figure 2. Each *h* refinement is expressed by adding new sub-tree to the tree of nodes. The actual active finite elements are represented as leaves in the tree.

Such a data structure is very convenient for recording mesh refinements. It also enables easily to unrefine the mesh, since the entire history of refinements is stored. However, it provides serious problems for non-stationary computations. Let us consider the new optimal mesh, presented in figure 3, obtained within the next time step by performing a sequence of refinements from the regular initial mesh.

The refinement tree for the previous time step presented in figure 2 is not compatible with the refinement tree for the next time step presented in figure 3 for the next time step.

These facts motivated us to consider three different strategies dealing with generation of the next time step optimal mesh as well as the projection of the previous time step solution into the actual mesh.

5. DIFFERENT STRATEGIES FOR MANAGING TIME STEP TRANSITION

We have tested the following three strategies managing with the generation of the new time step optimal mesh as well as the problem of projection of solutions between the previous and current time step

In the first strategy, presented in figure 4, the optimal mesh for the next time step is obtained starting from the same regular initial mesh as for the previous time step solution. There are the following advantages and disadvantages of this strategy:

– The tree of refinements can easily mark the singularities related to the new time step solution, since there are restrictions of the previous time step refinements.
– The whole process of generating the optimal mesh is computationally expensive, since the entire sequence of meshes needs to be considered.
– There is a need to store two computational meshes, since the previous time step solution has been obtained on completely different mesh.

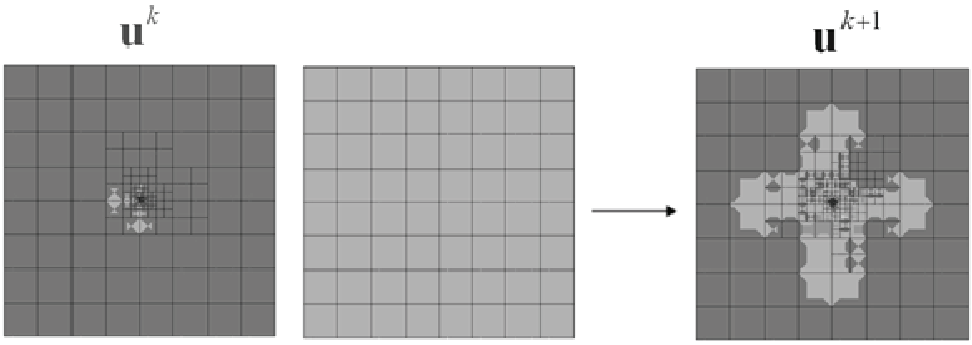


Fig. 4. *The optimal mesh from the previous time step, the regular initial mesh, and the optimal mesh for the next time step obtained from the regular initial mesh. Different degrees of gray denote different polynomial orders of approximation utilized on rectangular finite element edges and interiors.*

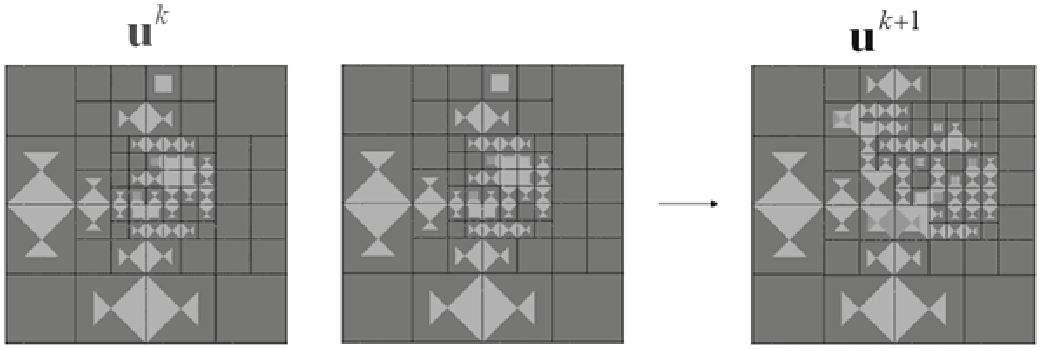


Fig. 5. *The optimal mesh for the next time step is obtained by executing some additional refinements on the optimal mesh from the previous time step. Different degrees of gray denote different polynomial orders of approximation utilized on rectangular finite element edges and interiors.*

The second strategy presented in figure 5 is to utilize the optimal mesh from the previous time step as the initial mesh for the next time step. There are the following advantages / disadvantages of this strategy

– The optimal mesh from the previous time step is no longer optimal for the next time step, and multiple new refinements may be needed in order to obtain the optimal mesh for the next time step solution.
– Since each time step requires some additional new refinements, the size of the problem is growing with furthcoming time steps.
– The solution from the previous time step can be actually stored on the same computational mesh, so there is no need to store two computational meshes.

The third proposed strategy utilizes the idea of unrefinements, based on the tree like data structure. The idea is to utilize the optimal mesh from the previous time step, but with some partial unrefinement, to step back before adjusting the computational mesh to the new time step problem with little different material data and possibly boundary conditions. This procedure is presented in figure 6. It corresponds to cutting leaves in the refinement tree illustrated in figure 2.

There are the following advantages and disadvantages of this strategy

– The quality of the previous time step solution stored on the previous mesh can be decreased after performing proposed unrefinement. This can be overcome by storing a copy of the previous time mesh and projecting the previous time step solution into the actual mesh for the current time step.
– There is no need to generate expensive sequence of meshes starting from initial regular mesh, since we utilize partially unrefned optimal mesh from the previous time step.

– Each time step requires some additional new refinements, however the starting mesh is one tree level smaller than the previous time step optimal mesh, which allows us to minimaze the new problem size.

The number of degrees of freedom over a single *hp* finite element with polynomial orders of approximation (p_1,p_2) presented on the left panel in figure 10, is $(p_1 + 1)(p_2 + 1) = p_1p_2 + p_1 + p_2 + 1$, compare (Paszyński, 2007; Paszyński et al., 2006).

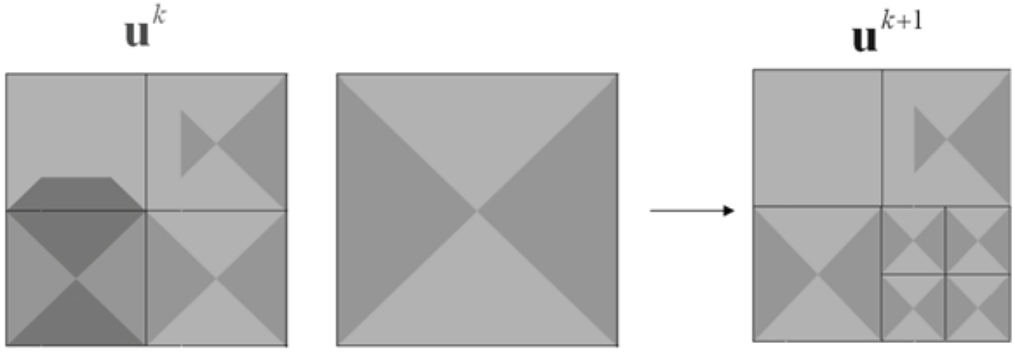


Fig. 6. *The optimal mesh for the next stime step is obtained by executing some unrefinements on the optimal mesh from the previous time step, followed by a sequence of new refinements. Different degrees of gray denote different polynomial orders of approximation utilized on rectangular finite element edges and interiors.*

6. NUMERICAL RESULTS

We conclude the presentation with some numerical results shown in table 1 and figures 7-9, illustrating the computational cost related to several time steps computations with three presented mesh management strategies.

The computational cost related to a single iteration of the self-adaptive *hp*-FEM involves the coarse mesh problem solution, the fine mesh problem solution and the computational cost related to generation of the fine mesh, selection of the optimal refinements and execution of the optimal refinements on the coarse mesh. The most expensive parts of this algorithm are the coarse and fine mesh solution, and we will utilize them as an indicator of the computational cost related to a single iteration of the self-adaptive *hp*-FEM. To estimate the cost, we need the fine mesh problem size.

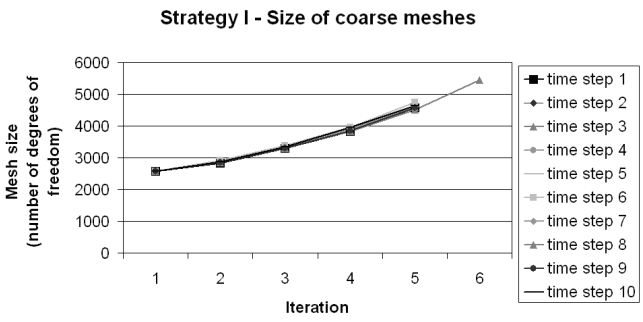


Fig. 7. *Sizes of coarse meshes from the sequences of meshes delivering 1% relative error, generated during ten time steps of the self-adaptive hp-FEM for the first strategy, when the iterations for each new time step are restarted from the initial coarse mesh.*

The fine mesh is obtained by performing global *hp* refinement – thus each finite element is broken into 4 new elements and the polynomial order of approximation is increased by one. The number of degrees of freedom over four new elements is of the order of $4(p_1 + 2)(p_2 + 2) = 4p_1p_2 + 8p_1 + 8p_2 + 16$. Thus, the number of degrees of freedom (d.o.f.) over the refined element is between 4 up to 10 times larger than original element size $N_{fine} = \alpha N_{course}$, where α depends on the polynomial orders of approximation.

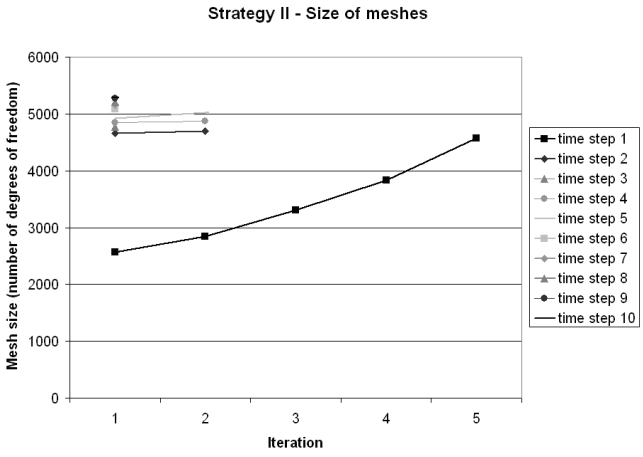


Fig. 8. *Sizes of coarse meshes from the sequences of meshes delivering 1% relative error, generated during ten time steps of the self-adaptive hp-FEM for the second strategy, when the iterations for each new time step are continued from the optimal mesh generated in the previous time step.*

Table 1. *Comparison of convergence of particular time steps executions of the self-adaptive hp-FEM utilizing three different mesh management strategies.*

	Strategy I		Strategy II		Strategy III	
Time step	Coarse mesh nr. of d.o.f.	Relative error	Coarse mesh nr. of d.o.f.	Relative error	Coarse mesh nr. of d.o.f.	Relative error
1	2573 2838 3302 3829 4566	33 32 19 6 1	2573 2838 3302 3829 4566	33 32 19 6 1	2573 2383 3302 3829 4566	33 32 19 6 1
2	2573 2853 3310 3892 4644	34 33 19 6 1	4663 4694	2 1	2527 2571 2913 3508 4539	100 71 21 8 1
3	2573 2853 3312 3859 4616	34 33 20 6 1	4775	1	2886 2983 3350 4356	27 17 6 1
4	2573 2853 3312 3859 4616	34 33 19 6 1	4843 4867	2 1	2548 2617 3054 3801 4750	113 59 18 4 1
5	2573 2863 3348 3889 4594	34 33 20 6 1	4922 5027	2 1	3103 3183 3618 4630	25 15 5 1
6	2573 2910 3373 3954 4745	35 34 19 6 1	5088	1	2544 2614 3064 3825 4725	113 56 18 4 1
7	2573 2869 3349 3909 4599	34 33 20 6 1	5159	1	3104 3230 3956 5076	23 12 3 1
8	2573 2851 3326 3834 4503 5439	34 33 19 6 2 1	5214	1	2563 2627 3148 3918 4860	100 56 17 4 1
9	2573 2851 3303 3848 4573	33 32 19 6 1	5270	1	3111 3208 3766 4797	19 13 3 1
10	2573 2865 3349 3951 4628	34 34 18 6 1	5286	1	2544 2604 2990 3575 4633	112 71 19 6 1

We can estimate the computational cost of a single iteration as:

$$N_{coarse}{}^2 \log N_{coarse} + N_{fine}{}^2 \log N_{fine} = N_{coarse}{}^2 \log N_{coarse} + (\alpha N_{coarse})^2 \log(\alpha N_{coarse}) \quad (11)$$

where $O(N \log N)$ is the estimation of the solver computational cost. We can estimate total computational cost related to each strategy, by assuming average $\alpha = 6$ and utilizing Eq. (11) to estimate total computational cost for all iterations for a given time step.

The comparison of the total computational costs for different mesh management strategies is presented in table 2. From the presented numerical experimants it follows that the second strategy is the best one. The second strategy utilizes the computational mesh from the previous time step. We believe it is related to the fact that in this numerical problem the optimal mesh for the next time step is similar to the optimal mesh from the previous time step. The third strategy utilizing some partial unrefinements doesn't work well in this example. Actually, it requires several new refinements, which can be read from figure 9. It is an open problem if the second strategy would be also the best one for other numerical non-stationary simulations.

Table 2. *Comparison of the total computational costs for different mesh management strategies*.

Time step	Strategy I	Strategy II	Strategy III
1	2.25E+10	2.25E+10	2.25E+10
2	2.30E+10	1.65E+10	1.99E+10
3	2.27E+10	8.62E+9	1.74E+10
4	2.28E+10	1.78E+10	2.20E+10
5	2.29E+10	1.88E+10	2.01E+10
6	2.39E+10	9.84E+9	2.20E+10
7	2.30E+10	1.01E+10	2.29E+10
8	3.37E+10	1.04E+10	2.30E+10
9	2.26E+10	1.06E+10	2.12E+10
10	2.32E+10	1.07E+10	2.07E+10
Total	2.40E+11	1.36E+11	2.12E+11

Finally we present in figure 11 resulting temperature and concentration fields distribution obtained after several iterations of the phase transition simulation, interfaced with the Cellular Automata algorithm Cellular Automata (CA) (Paszyński et al., 2008).

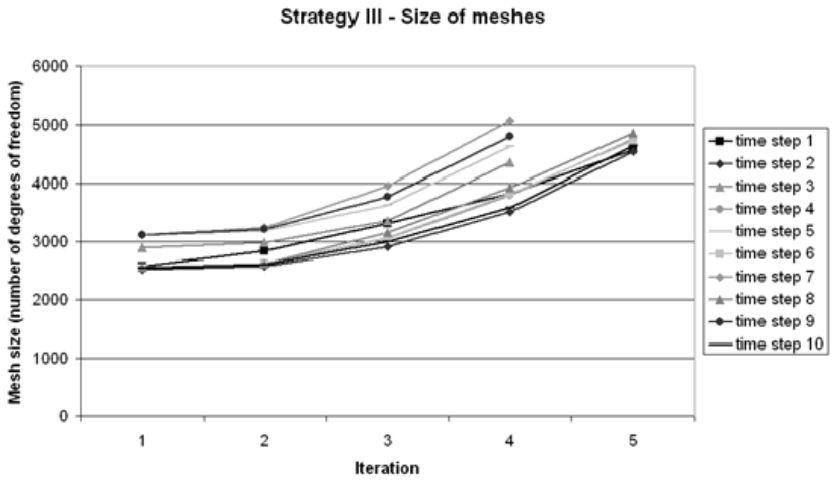


Fig. 9. *Sizes of coarse meshes from the sequences of meshes delivering 1% relative error, generated during ten time steps of the self-adaptive hp-FEM for the third strategy, when the iterations for each new time step are started from the unrefined optimal mesh generated in the previous time step.*

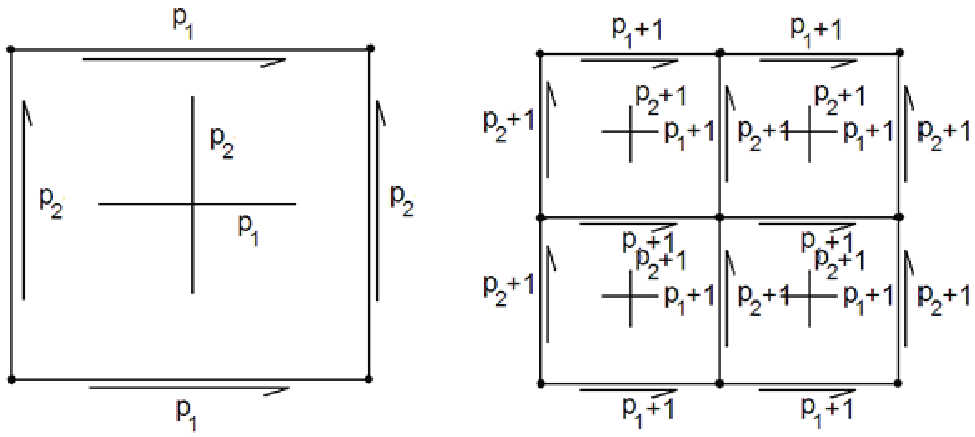


Fig. 10. *Left panel – a single hp finite element with polynomial orders of approximation (p_1,p_2); Right panel – the element from the left panel after hp refinement – there are four new son elements and polynomial orders of approximation are increased by one.*

7. CONCLUSIONS

In this paper we compared three different mesh managing strategies utilized by the non-stationary self-adaptive *hp*-FEM code. In particular, we compared three different techniques for utilization of the computational mesh and the solution from a previous time step solution during computations of an actual time step. The first strategy generated a next time step mesh starting from the regular initial mesh. The second strategy utilized previous time step mesh as a starting point for the next time step iteration. The third strategy performed global unrefinement on the previous time step mesh before utilizing the mesh for the actual time step solution. From the numerical simulations it follows that the second strategy is the most efficient strategy. The future work will involve testing the three strategies on different numerical problems, to check if the conclusions can be generalized.

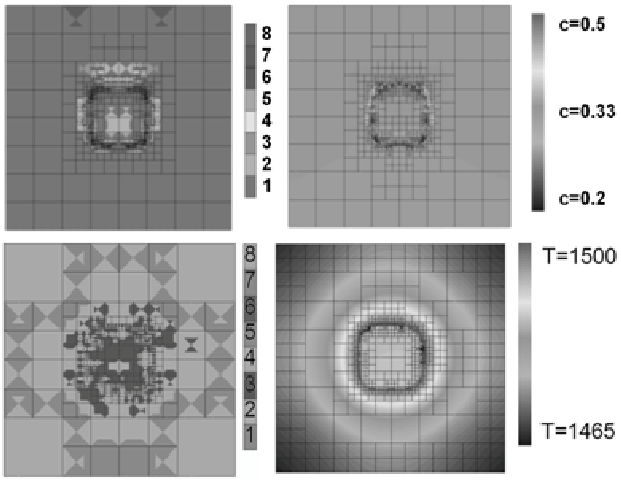


Fig. 11. *The optimal meshes in the last time step, and corresponding concentration and temperature fields distributions.*

ACKNOWLEDGEMENTS

The work reported in this paper has been partially supported by MNiSW grant no. 501 120836.

REFERENCES

Demkowicz, L., 2006, Computing with hp-Adaptive Finite Elements, Vol. I. One and Two Dimensional Elliptic and Maxwell Problems, Chapmann & Hall/Crc Applied Mathematics & Nonlinear Science.

Demkowicz, L., 2004, Projection-based interpolation, *ICES Report* 04-03, The University of Texas in Austin.

Demkowicz, L., Kurtz, J., Pardo, D., Paszyński, M., Rachowicz, W., Zdunek, A., 2007, Computing with hp-Finite Elements. Vol. II. Frontiers Three Dimensional Elliptic and Maxwell Problems with Applications, Chapmann & Hall/Crc Applied Mathematics & Nonlinear Science.

Demkowicz, L., Rachowicz, W., Devloo, Ph., 2002, A Fully Automatic hp-Adaptivity, *Journal of Scientific Computing*, 17, 117-142.

Gandin, Ch.-A., Rappaz, M., 1996, A 3D cellular automaton algorithm for the prediction of dendritic grain growth, *Acta Metallurgica*, 45, 2187-2198.

Jacot, A., Rappaz, M., 2002, A pseudo-front tracking technique for the modelling of solidification microstructures in multi-component alloys, *Acta Materialia*, 50, 8, 1909-1926.

Liu, Y., Xu, Q., Liu, B., 2006, A Modified Cellular Automaton Method for the Modeling of the Dendritic Morphology of Binary Alloys, *Tsinghua Science & Technology*, 11, 5, 495-500.

Matuszyk, P., Paszyński, M., 2007a, Extensions of the 2D fully automatic hp adaptive Finite Element Method for Stokes and non-stationary heat transfer problems, *9 US National Congress on Computational Mechanics, USACM*, San Francisco, USA.

Matuszyk, P., Paszyński, M., 2007b, Fully automatic 2D *hp*-adaptive Finite Element Method for Non-stationary Heat Transfer Problems, *COMPLAS,* Barcelona, Spain.

Narski, J., Picasso, M., 2007, Adaptive finite elements with high aspect ratio for dendritic growth of a binary alloy including fluid flow induced by shrinkage, *Computer Methods in Applied Mechanics and Engineering*, 196, 37-40.

Paszyński, M., 2007, Agents based hierarchical parallelization of complex algorithms on the example of hp Finite Element Method, *Lecture Notes in Computer Science,* 4488, 912-919.

Paszyński, M., Kurtz, J., Demkowicz, L., 2006, Parallel Fully Automatic *hp*-Adaptive 2D Finite Element Package, *Computer Methods in Applied Mechanics and Engineering,* 195, 7-8, 711-741.

Paszyński, M., Matuszyk, P., Gawąd, J., Madej, Ł., 2008, Phase transition modeling with CA merged with hp-adaptive FEM for the heat and mass transport problems, *XV Conference Computer Methods in Material Science*, Korbielów.

Rappaz, M., Gandin, Ch.-A., Desbiolles, J.-L., Thevoz, Ph., 1996, Prediction of grain structures in various solidification process, *Metallurgical and Materials Transactions* A, 27A, 695-704.

Zhao, P., Heinrich, J. C., 2001, Front-tracking finite element method for dendritic solidification, *Journal of Computational Physics*, 173, 765-796.

Zhu, M. F., Stefanescu, D. M., 2007, Virtual front tracking model for the quantitative modeling of dendritic growth in solidification of alloys, *Acta Materialia*, 55, 5, 1741-1755.

RÓŻNE STRATEGIE ZARZĄDZANIA SIATKĄ OBLICZENIOWĄ DLA NIESTACJONARNYCH OBLICZEŃ HP ADAPTACYJNĄ METODĄ ELEMENTÓW SKOŃCZONYCH

Streszczenie

W pracy tej porównujemy różne strategię rozwiązywania niestacjonarnych problemów transportu ciepła i masy, zastosowanych w celu symulacji przemian fazowych austenit-ferryt. Problemy transportu masy i ciepła w każdym kroku czasowym rozwiązywane są z pomocą *hp* adaptacyjnej Metody Elementów Skończonych (*hp*-FEM). Algorytm *hp*-FEM automatycznie generuję ciąg siatek obliczeniowych zawierających elementy różnego rozmiaru oraz różne stopnie aproksymacji wielomianowej, w taki sposób, żeby uzyskać zbieżność eksponencjalną dokładności rozwiązania numerycznego względem rozmiaru siatki. Ze względu na niestacjonarny charakter obliczeń, *hp*-FEM w każdym kroku obliczeniowym wykorzystuję rozwiązanie z poprzedniego kroku czasowego. W pracy porównujemy trzy różne strategię wykorzystania siatki optymalnej z poprzedniego kroku czasowego. Pierwsza strategia polega na uruchamianiu obliczeń od tej samej siatki rzadkiej na początku każdego kroku czasowego. Druga strategia polega na wykorzystaniu siatki optymalnej wygenerowanej w poprzednim kroku czasowym do kontynuowania obliczeń w kolejnym kroku czasowym. Trzecia strategia polega na cofnięciu jednego kroku adaptacji na siatce optymalnej z poprzedniego kroku czasowego przed wykorzystaniem jej w kolejnym kroku czasowym. Porównania dokonano z pomocą przykładowego problemu obliczeniowego – symulacji przemian fazowych austenit-ferryt.

Received: April 7, 2010
Received in a revised form: April 26, 2010
Accepted: April 26, 2010