

## Fast multi-scale simulations of a Step-and-Flash Imprint Lithography

\*M. Sieniek<sup>1</sup>, P. Gurgul<sup>1</sup> and M. Paszyński<sup>1</sup>

<sup>1</sup>AGH University of Science and Technology, Krakow, Poland

\*Corresponding author: msieniek@GMAIL.COM

### Abstract

In this paper we present a graph grammar based multi-frontal direct solvers resulting in 90 percent speedup in a multi-scale simulations of the Step and Flash Imprint Lithography (SFIL) a modern patterning process. The multi-scale simulation involves nano-scale Molecular Statics model coupled with macro-scale linear elasticity with thermal expansion coefficient. The simulations involves the densification of the liquied polymer inside the feature resulting from the photopolimerization, as well as shrinkage of the feature after removal of the template. The macro-scale domain is solved with a new version of multi-frontal direct solver with the graph grammar based mechanism for the reuse of the sub-domains with similar geometries and similar material properties. The graph grammar model enables for automatic localization of the sub-domain that can be reuse in our solver algorithm. We show that the new solver enables for 90 percent speedup of the numerical solution.

**Keywords:** Multi-frontal solver, nanolithography, molecular statics, linear elasticity with thermal expansion coefficient, multi-scale modeling

### Introduction

The paper presents the multi-scale modeling of the Step-and-Flash Imprint Lithography (SFIL), a modern patterning process utilizing photopolymerization in order to replicate a template onto a substrate (Colburn et al. 2001). The SFIL process can be simulated by macro-scale model as linear elasticity with thermal expansion coefficient (Hughes 2000), however in some areas of the domain e.g. on the interface between the feature and the template, the nano-scale model, namely the molecular statics model must be included (Paszynski et al. 2005). The three dimensional finite element method simulations are expensive (Demkowicz et al. 2007), and thus we propose a multi-frontal solver algorithm with the reuse technique. The multi-frontal solvers are most advanced direct solvers used to solve the linear systems of equations (Duff and Reid 1983, Duff and Reid 1984, Geng et al. 2006). In our previous works we already modeled the mesh generation and multi-frontal solvers by graph grammar (Paszynska et al. 2012a, Paszynska et al. 2012b, Paszynska et al. 2008, Paszynski et al. 2009a, Paszynski et al. 2009b, Paszynski et al. 2010, Paszynski and Schaefer 2010). However in this work we introduce a new graph grammar model allowing for efficient reuse of identical sub-branches of the elimination tree.

### Step-and-Flash Imprint Lithography

The major processing steps of SFIL include (compare 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. Photopolymerization, however, is often accompanied by densification (see Fig. 1a). 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.

## Macro scale model

The macro-scale model is based on the adaptive Finite Element Method (FEM) discretization of linear elasticity with thermal expansion coefficient. The FEM model can be summarized as follows: Find  $\mathbf{u} \in \mathbf{V}$  such that

$$\int_{\Omega} w_{(i,j)} c_{ijkl} u_{(k,l)} d\Omega = -\theta \int_{\Omega} w_{(i,j)} c_{ijkl} \alpha_{kl} d\Omega - \int_{\Omega} w_{(i,j)} \sigma_{ij}^0 d\Omega \quad \forall \mathbf{w} \in \mathbf{V} \quad (1)$$

where  $c_{ijkl}$  are elastic coefficients,  $\alpha_{kl}$  thermal expansion coefficients,  $\sigma_{ij}^0$  initial stress.  $\theta$  temperature, and  $u_{(i,j)} = \frac{u_{i,j} + u_{j,i}}{2}$  is the strain tensor.

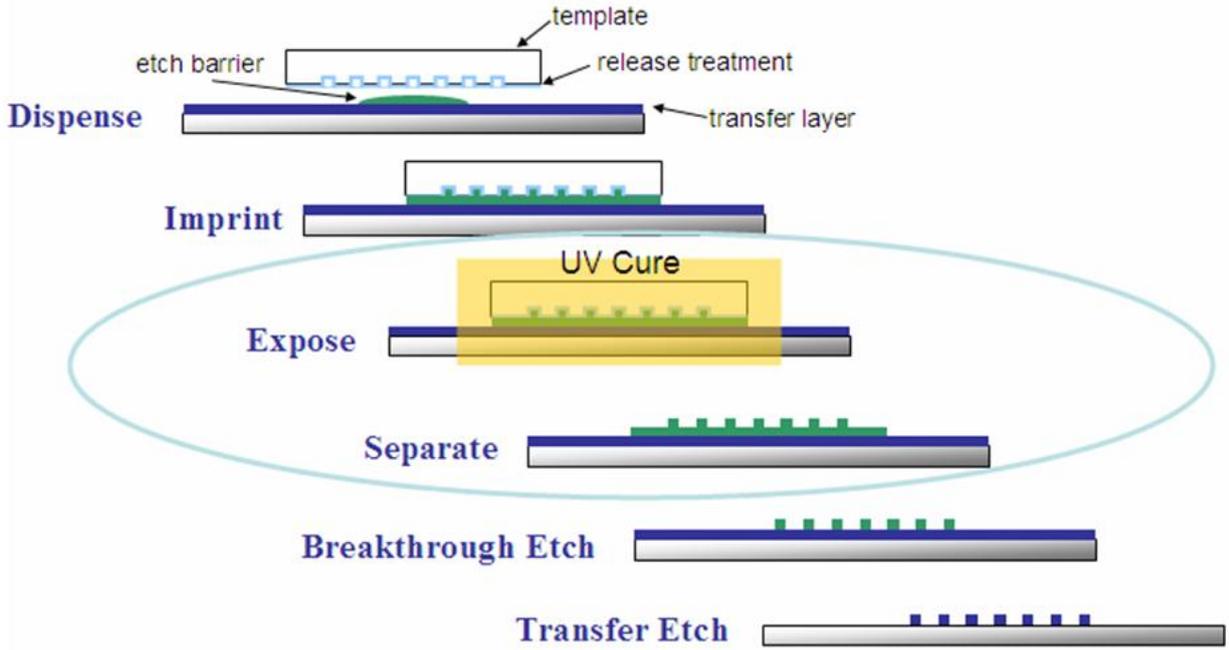


Figure 1. Step and Flash Imprint Lithography

## Molecular statics model

The molecular statics nano-scale model can be summarized as follows:

Find the equilibrium configuration of particles satisfying

$$\sum_{\beta} \mathbf{F}_{\alpha\beta} = \mathbf{0} \quad (2)$$

where

$$\mathbf{F}_{\alpha\beta} = k_{\alpha\beta} \left( r_{\alpha\beta} + \Delta r_{\alpha\beta} - r_{\alpha\beta}^0 \right) \frac{(\mathbf{x}_{\beta} - \mathbf{x}_{\alpha})}{\|\mathbf{x}_{\beta} - \mathbf{x}_{\alpha}\|} \quad (3)$$

is the force between interacting particles  $\alpha$  and  $\beta$ ,  $k_{\alpha\beta}$  is the w spring stiffness coefficient,  $r_{\alpha\beta} + \Delta r_{\alpha\beta} = \|\mathbf{x}_{\beta} - \mathbf{x}_{\alpha}\|$  is the length of the spring in the equilibrium configuration,  $\mathbf{x}_{\alpha}, \mathbf{x}_{\beta}$  represents the (unknown) equilibrium configuration of particles,  $r_{\alpha\beta}^0$  is the length of the unstretched spring. The spring stiffness coefficients are obtained from Monte Carlo simulations concerning the photopolimerization of the feature (Colburn et. al. 2001)

## Coupling between the models

The coupling between the macro-scale and non-scale model is done through identification of particles located on the interface of the nano-scale domain with nodes of the finite element mesh. In such the case over the nano-scale model we solve the molecular statics equations, over the macro-scale model we solve discretized variational formulation for finite element method for linear elasticity with thermal expansion coefficient, however the interface between domains we treat in a special way. The variables from the interface are treated by the nano-scale model as particles represented by their relative change of location  $\Delta\tilde{p}_\alpha = \tilde{x}_\alpha - \tilde{p}_\alpha$ , but from the macro-scale FEM model the variables are treated as degrees of freedom of finite element mesh  $\tilde{u}_\alpha$ . This is equivalent to identification of these two variables

$$\Delta\tilde{p}_\alpha = \tilde{u}_\alpha \quad (4)$$

for each particles (or finite element degrees of freedom)  $\alpha$  located on the interface. In practice it is not necessary to add these new equations to the system, we can just aggregate the nano-scale and macro-scale entries to the same row of the global matrix.

## Graph grammar based solver with reuse technique

In this section we describe a graph grammar based multi-frontal solver with reuse technique.

The first step of the solver algorithm is to generate the computational mesh. It is done by executing a sequence of graph grammar productions, generating a graph structure representing computational mesh. The first graph grammar production is presented on left panel in Figure 2. The productions replaces the starting graph containing only a single vertex **S** with a graph representing a single hexahedral element with eight nodes. The following graph grammar productions replaces some nodes by sub-graphs representing smaller elements. The graph nodes as well as graph grammar productions are attributed by the location over the rectangular domain. The graph grammar production  $(P)^{TNW}$  from right panel in Figure 2 is actually replicated for different locations, for  $\{\mathbf{TNW}, \mathbf{TNE}, \mathbf{TSW}, \mathbf{TSE}, \mathbf{BNW}, \mathbf{BNE}, \mathbf{BSW}, \mathbf{BSE}\}$  where **T** and **B** stands for top and bottom, and **N**, **S**, **W**, **E** stand for north, south, west, east.

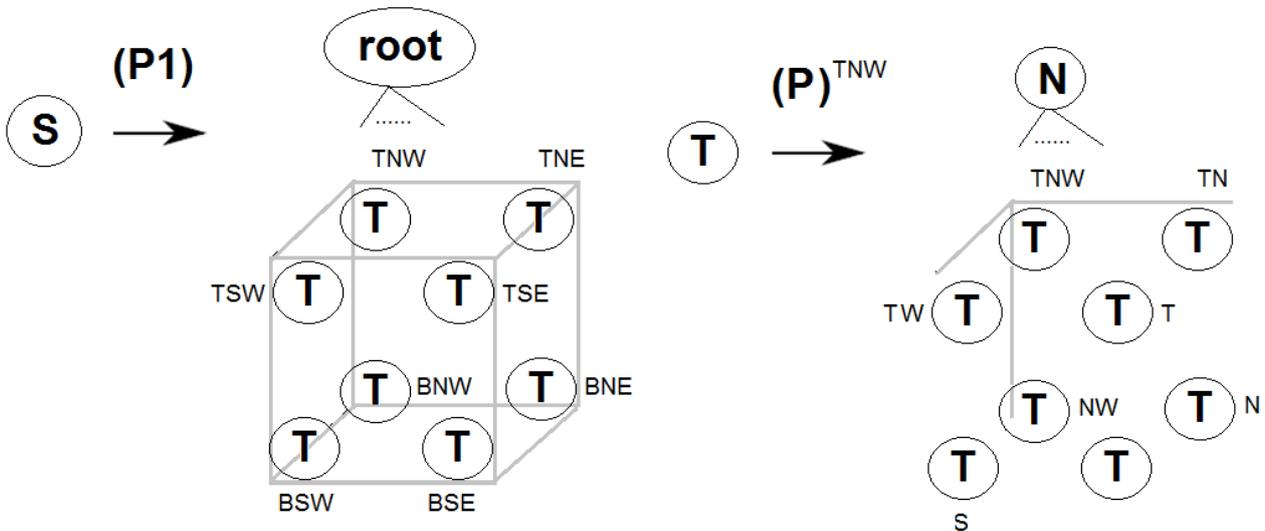
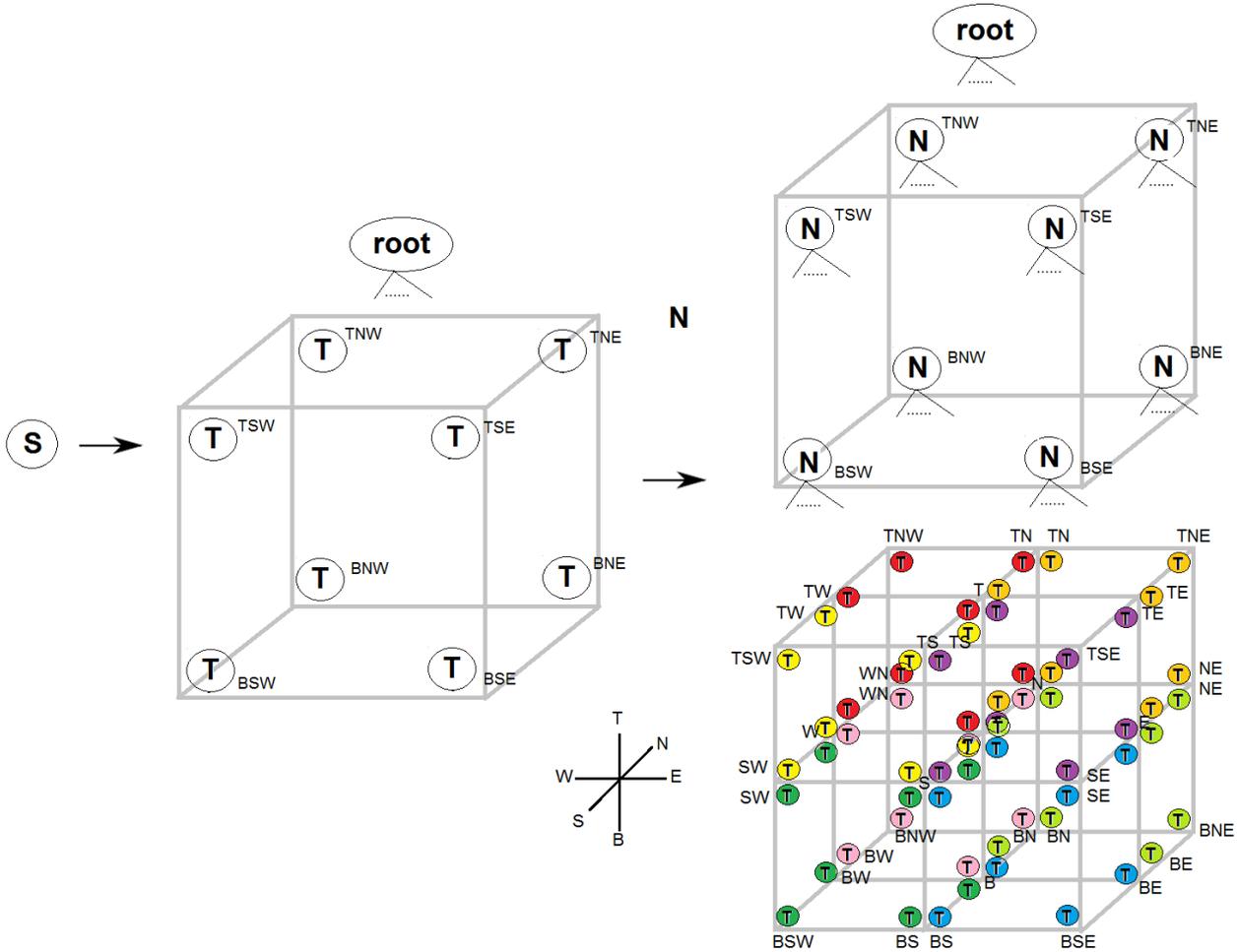
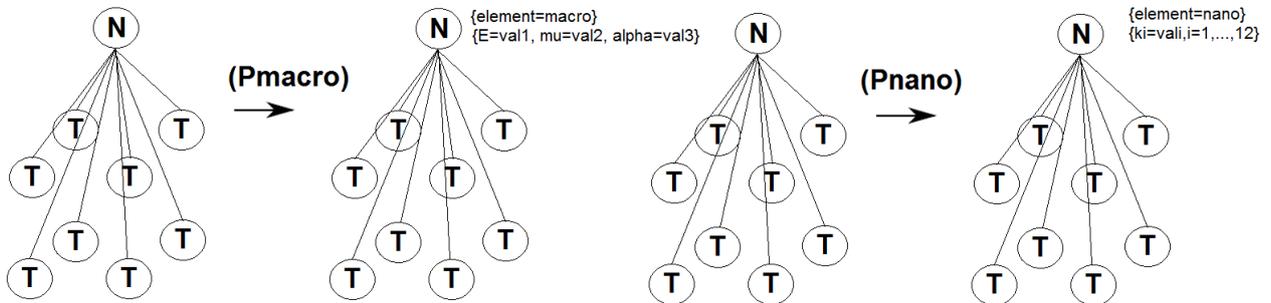


Figure 2. Exemplary graph grammar productions for generating of the structure of the mesh

The exemplary derivation of eight finite element mesh is presented in Figure 3. In the first step of the derivation, production **(P1)** is executed, in the second step, productions **(P)<sup>TNW</sup> - (P)<sup>TNE</sup> - (P)<sup>TSW</sup> - (P)<sup>TSE</sup> - (P)<sup>BNW</sup> - (P)<sup>BNE</sup> - (P)<sup>BSW</sup> - (P)<sup>BSE</sup>** are executed to obtain the eight finite element mesh. The graph representing the mesh has hierarchical tree-like structure storing the history of graph grammar productions derivation. To obtain larger meshes, it is necessary to add graph grammar productions for locations like **{T,B,N,S,W,E,TN,TS,TW,TE,BN,BS,BW,BE,NE,NW,SE,SW}**, compare labels of the left bottom sub-graph at Figure 3.



**Figure 3. Derivation of eight finite element mesh**

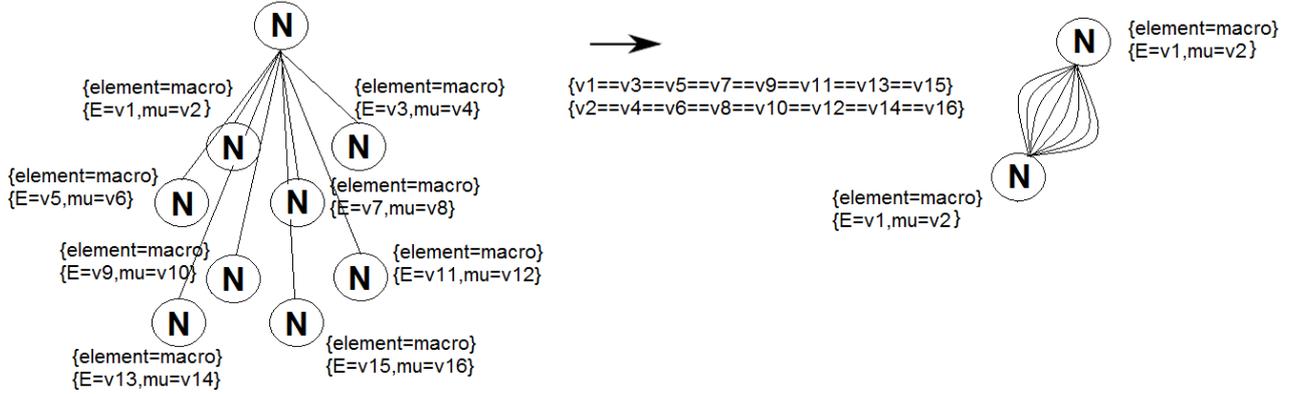


**Figure 4. Graph grammar productions for identification of macro- and nano-scale elements**

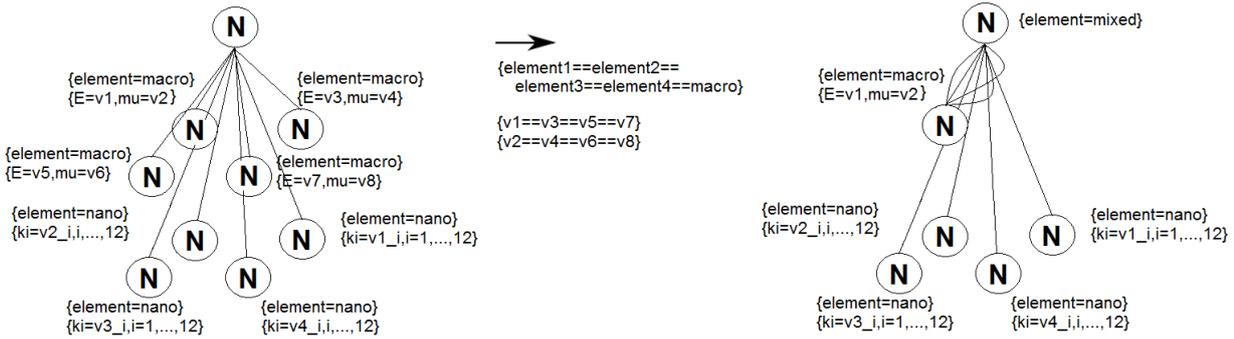
The next step of the solver algorithm is the identification of macro-scale and nano-scale elements. Notice that graph nodes labeled with **N** actually represents particles (over nano-scale elements) or finite element method nodes (over macro-scale elements). Thus, the elements are represented by

patches of eight nodes. In our exemplary mesh presented in Figure 3 we have eight elements denoted by different colors.

This identification is performed by graph grammar production presented in Figure 4. The macro-scale elements are attributed by Young modulus and Poisson ratio values. The nano-scale elements are attributed by parameters of the spring force parameters  $k_{\alpha\beta}$ .



**Figure 5. Exemplary graph grammar production for identification of macro-scale elements with identical material data**



**Figure 6. Exemplary graph grammar production for partial identification of macro-scale elements with identical material data**

The resulting tree structure can be directly utilized by the multi-frontal solver algorithm (Paszynski et al. 2010, Paszynski and Schaefer 2010)

The third step of the solver algorithm is the identification of identical sub-branches of the elimination tree, for the reuse of partially LU factorized matrices. The exemplary graph grammar production for such the identification is presented in Figure 5. Such the graph grammar production checks if all eight son elements are macro-scale elements and if corresponding Young modulus and Poisson ratios are identical. If this is the case, the eight son element nodes are reduced to one representative node, so the LU factorization can be performed only once and father node can merge eight identical matrices from the same representative son node.

Another more complicated case for the identification is presented in Figure 6. In this example only four son elements are macro-scale elements with identical Young modulus and Poisson ratio values. The four identical macro-scale elements are reduced to one representative elements, however the nano-scale elements are stochastic in their nature and cannot be reduce to one representative element.

Finally, on the modified elimination tree we can execute the multi-frontal solver algorithm – namely the forward elimination

```

1 function forward_elimination(node)
2 if new_schur_matrix already computed for the node then
3 return schur_matrix
4 if node is a leaf then
5     generate local system assigned to node
6     excluding boundary conditions
7 else
8     loop through son_nodes
9         schur_matrix = forward_elimination(son_node)
10        merge schur_matrix into new_system
11    end loop
12 end if
13 find fully assembled nodes and eliminate them
14 return new_schur_matrix
15 end function

```

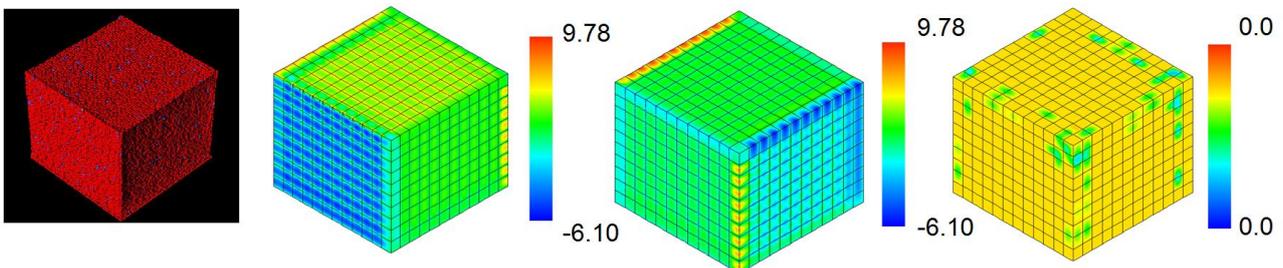
Notice that in case of representative nodes in line 9 we actually call the same node of the elimination tree many times and line 2 prevents from recomputing the identical Schur complement matrices many times. The forward elimination algorithm is followed by analogous backward substitution.

## Numerical results

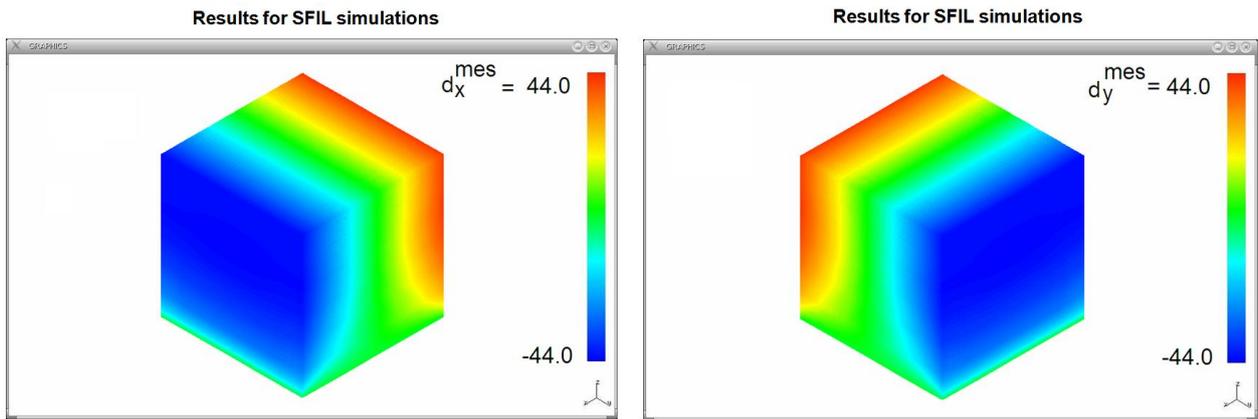
In this paper we consider two simulations. The first simulation concerns the multi-scale simulations of the feature inside the template, where the interior of the domain is modeled by the macro-scale model, but the boundary layers where the interactions between the feature and the template must be well captured the nano-scale model is utilized, compare Figure 7.

The second simulation concerns the macro-scale model for the feature outside the template, compare Figure 8.

In both cases the graph grammar based reuse technique can be utilized. In the second example we are able to reuse all of the matrices at each level of the elimination tree. In the first case, we are also able to reuse all the matrices from the macro-scale domain, however the boundary nano-scale layers must be process at the very end.



**Figure 7. Nano-scale simulations of the feature inside the template**  
**Left panel: Boundary layers modeled by molecular statics**  
**Other panels: X, Y and Z components of the displacement vector field for the interior modeled by linear elasticity with thermal expansion coefficient**

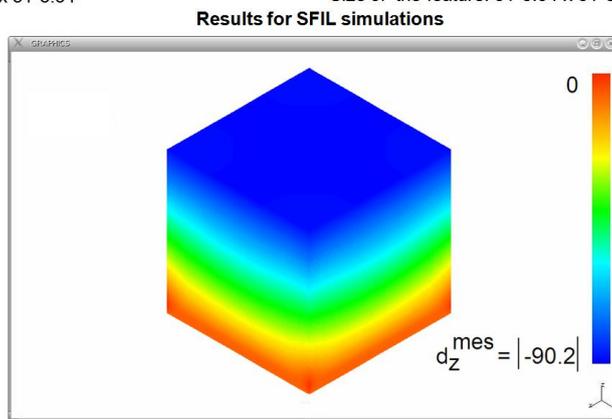


Displacements of the feature in x direction

Size of the feature: 51\*5.34 x 51\*5.34 x 51\*5.34

Displacements of the feature in y direction

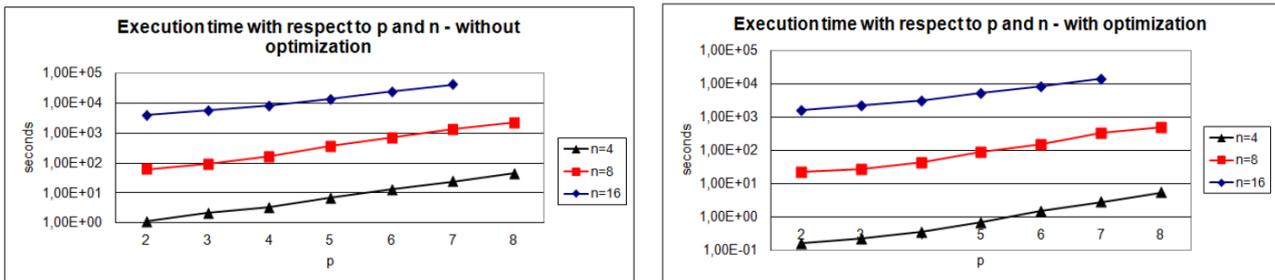
Size of the feature: 51\*5.34 x 51\*5.34 x 51\*5.34



Displacements of the feature in z direction

Size of the feature: 51\*5.34 x 51\*5.34 x 51\*5.34

**Figure 8. Macro-scale simulations of the feature outside the template.**



**Figure 9. Left panel: Execution time of the solver without reuse**

**Right panel: Execution time of the solver with reuse**



**Figure 10. Speedup of the reuse solver**

We conclude the numerical results section with comparison of the execution times of the graph grammar based multi-frontal solver with and without the reuse technique. The results presented in Figure 9 concerns computational grids with different number of elements in each direction ( $n$  parameter) as well as different polynomial orders of approximations utilized over the macro-scale domain ( $p$  parameter). The resulting speedup of the reuse solver algorithm is presented in Figure 10.

## Conclusions

In this paper we presented a fast multi-frontal solver algorithm enabling for speed-up up to 90% of the solution over the macro-scale domain in the multi-scale nanolithography simulations. The direct solver algorithm utilized the graph grammar model and the efficient reuse of identical sub-branches of the elimination tree.

## Acknowledgements

The work of MS was supported by Polish National Science Center grant no. DEC-2011/03/N/ST6/01397. The work of MP was supported by Polish National Science Center grant no. DEC-2012/06/M/ST1/00363.

- Demkowicz L., Kurtz J., Pardo D., Paszynski M., Zdunek A. (2007), Computing with  $hp$ -Adaptive Finite Element Method. Vol. II. Frontiers: Three Dimensional Elliptic and Maxwell Problems. Chapman & Hall / CRC Applied Mathematics and Nonlinear Science
- Duff I. S., Reid J. K. (1984), The multifrontal solution of unsymmetric sets of linear systems, *SIAM Journal of Scientific and Statistical Computing*, vol. 5, pp.633-641.
- Duff I. S., Reid J. K. (1983) The multifrontal solution of indefinite sparse symmetric linear equations, *ACM Transactions on Mathematical Software*, vol. 9, pp. 302-325
- Geng P., Oden T. J., van de Geijn R. A. (2006) A Parallel Multifrontal Algorithm and Its Implementation, *Computer Methods in Applied Mechanics and Engineering*, vol. 149, pp.289-301.
- Paszynska, A., Grabska, E. and Paszynski, M. (2012a). A graph grammar model of the  $hp$  adaptive three dimensional finite element method. part I, *Fundamenta Informaticae*, vol. 114 no., pp. 149–182.
- Paszynska, A., Grabska, E. and Paszynski, M. (2012b). A graph grammar model of the  $hp$  adaptive three dimensional finite element method. part II, *Fundamenta Informaticae* vol. 114 np. 2, 183–201.
- Paszynska, A., Paszynski, M. and Grabska, E. (2008). Graph transformations for modeling  $hp$ -adaptive finite element method with triangular elements, *Lecture Notes in Computer Science*, vol. 5103 pp. 604–613.
- Paszynski, M. (2009a). On the parallelization of self adaptive  $hp$ -finite element methods part I. composite programmable graph grammar model, *Fundamenta Informaticae*, vol. 93 no. 4, pp. 411–434.
- Paszynski, M. (2009b). On the parallelization of self-adaptive  $hp$ -finite element methods part ii. partitioning communication agglomeration mapping (PCAM) analysis, *Fundamenta Informaticae*, vol. 93 no. 4, pp. 435–457.
- Paszynski, M. Pardo, D., Paszynska, A. (2010) Parallel multi-frontal solver for  $p$  adaptive finite element modeling of multi-physics computational problems, *Journal of Computational Science*, vol. 1, no. 1, pp. 48-54.
- Paszynski, M. Schaefer R., (2010) Graph grammar driven partial differential equations solver, *Concurrency and Computations: Practise and Experience*, vol. 22, no. 9, pp.1063-1097.
- Colbum, M. E.Suez, I. Choi, B. J.Meissi, I.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*, vol. 19, pp. 6-20.
- Paszynski, M.Romkes, A.Collister, E.Meiring, J.Demkowicz, L. F.Willson, C. G (2005) On the modeling of Step-and-Flash Imprint Lithography using molecular statics models. *ICES Report*, vol. 05-38
- Hughes T. J. R. (2000) *The Finite Element Method. Linear Statics and Dynamics Finite Element Method Analysis*, Dover, 2000.