Maciej Paszyński*

CONVERGENCE OF ITERATIVE SOLVERS FOR NON-LINEAR STEP-AND-FLASH IMPRINT LITHOGRAPHY SIMULATIONS

The paper presents the analysis of the iterative solvers utilized to solve the non-linear problem of Step-and-Flash Imprint Lithography (SFIL) a modern patterning process. The simulations consists in solving molecular statics problem for the polymer network, with quadratic potentials. The model distinguishes the strong interparticle interactions between particles forming a polymer network, and weak interactions between remaining particles. It also allows for large deformations, which all together implies the non-linear model. To illustrate the convergence of the iterative solvers, we present snapshots of the deformation of the sample being subject to the iterative solution. We claim that the animation is an interesting way of illustrating the convergence of the iterative solvers.

Keywords: *iterative solvers, non-linear problems, Step-and-Flash Imprint Lithography*

ZBIEŻNOŚĆ SOLWERÓW ITERACYJNYCH DLA NIELINIOWYCH SYMULACJI PROCESU NANOLITOGRAFII PRZEZ NAŚWIETLANIE I WYCISKANIE

Artykuł analizuje zbieżność solwerów iteracyjnych dla nieliniowych symulacji procesu nanolitografii przez naświetlanie i wyciskanie. Symulacje polegają na rozwiązaniu zadania statyki cząsteczkowej dla sieci polimerów, w którym przyjęto kwadratowe potencjały międzycząsteczkowe, rozróżniono silniejsze oddziaływania pomiędzy cząstkami tworzącymi łańcuchy polimerów oraz słabsze oddziaływania pomiędzy pozostałymi cząstkami, a także dopuszczono występowanie dużych odkształceń, co implikuje model nieliniowy. W celu ilustracji zbieżności solwerów przedstawiono wizualizacje odkształceń sieci polimerów w kolejnych iteracjach. Taka animacja jest interesującą metodą ilustracji zbieżności solwerów iteracyjnych.

Słowa kluczowe: *solwery iteracyjne, problemy nieliniowe, symulacje nanolitografii*

* AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, IT and Electronics, Department of Computer Science, al. Mickiewicza 30, 30-059 Krakow, Poland, paszynsk@agh.edu.pl

1. Introduction

The paper presents the analysis of the convergence of iterative solver utilized to solve the non-linear molecular statics problem resulting from Step-and-Flash Imprint Lithography (SFIL) simulations. The SFIL is a modern patterning process utilizing the photopoymerization to replicate the topography of a template onto a substrate [1, 3, 4]. The liquid polymer is placed on the substrate, pressed by the template with previosly prepared shape, and the photopolymerization process is forced by the UV expose. The simulation of the SFIL process focuses on the modeling of the densification step, where the polymer network is formed and the interparticle forces arise and result in a shrinkage of the sample. The average distance between the molecules decreases and causes volumetric contraction. Exemplary shrinkage of the sample is presented in Figure 1. Densification of the SFIL photopolymer may affect both the cross sectional shape of the sample and the placement of relief patterns.

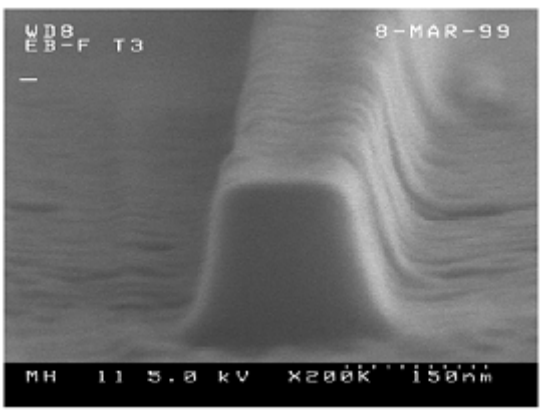

Fig. 1. Shrinkage of the sample after removal of the template (picture obtained from Prof. G. C. Willson from the University of Texas in Austin)

The densification process resulting in the shrinkage of the sample is modeled by the non-linear molecular statics. The input of the algorithm is the polymer network obtained from the Monte-Carlo simulation [2], describing the chemical reactions that take place during the photopolymerization. The resulting molecular structure consists of a lattice of point mass particles with dual, nearest-neighbor, force interactions governed by force potential functions. In the molecular statics model, large particle displacements are allowed and all inter-molecular bonds are governed by quadratic force potential functions.

The resulting model is non-linear and must be solved using the iterative linearization method. The Newton-Raphson method is used for the linearization, and the GMRES iterative solver is used to solve the linearized problems in consequtive iterations. The convergence of the iterative solver is tested on the $50\times50\times50$ nm rep-

resentative sample, and the snapshots of the deformations during particular iterations are presented. The non-linear model is also compared with linear one, assuming small deformations, and the linear model is solved by using PCG solver.

2. Molecular Statics Model

This chapter presents the derivation of the equation governing the equilibrium configurations of the interacting particles forming the polymerized network. The interpartice force parameters were generated by the Monte Carlo simulation of the photo-curing process [2]. The derivation presented here is based on the detailed derivations for linear and non-linear models for the Molecular Statics and Finite Element Method [5] summarized in technical report [6].

We consider an arbitrary pair of molecules in the mesh with indices α and β and given initial position vectors $\mathbf{p}_\alpha = (\hat{x}_\alpha, \hat{y}_\alpha, \hat{z}_\alpha)$ and $\mathbf{p}_\beta$. The unknown equilibrium position vectors of particles α and β are denoted $\boldsymbol{x}_\alpha = (x_\alpha, y_\alpha, z_\alpha)$ and $\mathbf{x}_\beta$, respectively. The displacements from the initial position in the lattice to the equilibrium position are represented by the vectors $\mathbf{u}_\alpha = \mathbf{x}_\alpha - \mathbf{p}_\alpha$ and $\mathbf{u}_\beta = \mathbf{x}_\beta - \mathbf{p}_\beta$. Let $\|\cdot\|$ denote the Eucledean norm in $\mathbb{R}^3$, let $r_{\alpha\beta} = \|\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha\|$ be the distance between particles α and β in equillibrium configuration.

In this paper we consider two formulations of the problem. First formulation assumes small deformations, and thus we formulate the equillibrium equations in the initial configuration of particles. In other words, the interparticle forces $\mathbf{F}_{\alpha\beta}$ act along the vector $\mathbf{p}_\beta - \mathbf{p}_\alpha$. The interparticle force is defined as

$$\mathbf{F}_{\alpha\beta} = -\underbrace{\frac{\partial V(r_{\alpha\beta})}{\partial r_{\alpha\beta}}}_{\text{magnitude}} \underbrace{\frac{\mathbf{p}_\beta - \mathbf{p}_\alpha}{\|\mathbf{p}_\beta - \mathbf{p}_\alpha\|}}_{\text{direction}} \tag{1}$$

where $V(r_{\alpha\beta})$ is the potential function due to the bond between particles α and β. The small deformation assumption, in particular the fact the we formulate the problem in the initial position of particles, implies the linear model.

In the second formulation the large deformations are allowed and that is why we formulate the equillibrium equations in the unknown destination configuration of particles. In other words, the interparticle forces $\mathbf{F}_{\alpha\beta}$ act along the vector $\mathbf{x}_\beta - \mathbf{x}_\alpha$. The interparticle force is defined as

$$\mathbf{F}_{\alpha\beta} = -\underbrace{\frac{\partial V(r_{\alpha\beta})}{\partial r_{\alpha\beta}}}_{\text{magnitude}} \underbrace{\frac{\mathbf{x}_\beta - \mathbf{x}_\alpha}{\|\mathbf{x}_\beta - \mathbf{x}_\alpha\|}}_{\text{direction}} \tag{2}$$

All the force potential functions $V(r_{\alpha\beta})$ are assumed to be quadratic and therefore the magnitude of the intermolecular forces $\mathbf{F}_{\alpha\beta}$ is linearly dependent on $r_{\alpha\beta}$

$$F_{\alpha\beta} = k_{\alpha\beta}\left(r_{\alpha\beta} + \Delta r_{\alpha\beta} - r^0_{\alpha\beta}\right) \tag{3}$$

where $k_{\alpha\beta}$ is the stiffness coefficient of the spring force (provided by the Monte Carlo simulation) $r_{\alpha\beta}+\Delta r_{\alpha\beta}$ is length of the spring in equillibrium configuration of the lattice structure and where $r^0_{\alpha\beta}$ is the unstretched length of the spring (when $\boldsymbol{F}_{\alpha\beta}=\mathbf{0}$).

The equillibrium equations are obtained by summing up all interparticle forces for all particles neighboring the particle α, collected in the set $\mathcal{N}_\alpha$. For the linear formulation assuming small deformations we get

$$\sum_{\beta\in\mathcal{N}_\alpha}\mathbf{F}_{\alpha\beta}=-\sum_{\beta\in\mathcal{N}_\alpha}\frac{\partial V(r_{\alpha\beta})}{\partial r_{\alpha\beta}}\frac{\mathbf{p}_\beta-\mathbf{p}_\alpha}{\|\mathbf{p}_\beta-\mathbf{p}_\alpha\|}=\mathbf{0}. \tag{4}$$

and for the non-linear formulation allowing for large deformations we have

$$\sum_{\beta\in\mathcal{N}_\alpha}\mathbf{F}_{\alpha\beta}=-\sum_{\beta\in\mathcal{N}_\alpha}\frac{\partial V(r_{\alpha\beta})}{\partial r_{\alpha\beta}}\frac{\mathbf{x}_\beta-\mathbf{x}_\alpha}{\|\mathbf{x}_\beta-\mathbf{x}_\alpha\|}=\mathbf{0}. \tag{5}$$

The parameters of the potential functions $\{V(r_{\alpha\beta})\}_{\beta\in\mathcal{N}_\alpha}$ are provided by the Monte Carlo simulation algorithm discussed in [2].

In the non-linear model, the intermolecular forces $\boldsymbol{F}_{\alpha\beta}$ are linear and we allow for large deformations. It implies that directions of the spring forces are aligned along $\boldsymbol{x}_\beta-\boldsymbol{x}_\alpha$. Thus, the non-linearity of the model is a consequence of large deformations.

In this paper we simulate the shrinkage of the sample after removing of the template. The shrinkage is a result of internal forces resulting from the photopolymerization of the sample. That is why there are no external forces included in the equillibrium equations. Moreover, the position of the bottom layer of particles is assumed to be fixed.

3. Linear formulation assuming small deformations

The linear Molecular Statics model is based on two basic assumptions:

1. All the force potential functions $V(r_{\alpha\beta})$ are quadratic and therefore the magnitude of the intermolecular forces $\mathbf{F}_{\alpha\beta}$ is linearly dependent on $r_{\alpha\beta}$.
2. Small deformations are observed. We will show now that this implies the direction of the spring forces along the initial spring alignments $\boldsymbol{p}_\beta-\boldsymbol{p}_\alpha$.

In the following formulae we will write $r_{\alpha\beta}$ as l for simplicity.

$$\Delta l=(l+\Delta l)-l=\frac{(l+\Delta l)^2-l^2}{l+\Delta l+l}=\frac{\|\boldsymbol{x}_\beta-\boldsymbol{x}_\alpha\|^2-l^2}{2l+\Delta l}=$$
$$=\frac{\|(\boldsymbol{p}_\beta-\boldsymbol{p}_\alpha)+(\boldsymbol{u}_\beta-\boldsymbol{u}_\alpha)\|^2-l^2}{2l+\Delta l}$$

where

$$l^2=\frac{\left(\|\boldsymbol{p}_\beta-\boldsymbol{p}_\alpha\|^2\right)^2+(\boldsymbol{p}_\beta-\boldsymbol{p}_\alpha)\cdot(\boldsymbol{u}_\beta-\boldsymbol{u}_\alpha)+\|\boldsymbol{u}_\beta-\boldsymbol{u}_\alpha\|^2-l^2}{2l+\Delta l}=$$

$$= \frac{2(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha) + \|\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha\|^2}{2l + \Delta l}$$

From the assumption of small deformations it follows that $\|\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha\|^2$ is a higher order term that can be neglected, and $\frac{\Delta l}{l} \ll 1$. Thus now

$$\Delta l \approx \frac{1}{l}(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha)$$

which we can rewrite in terms of our unknowns $\boldsymbol{x}_\alpha$, $\boldsymbol{x}_\beta$:

$$\Delta l \approx \frac{1}{l}(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha + \boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) =$$
$$= -\frac{1}{l}\|\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha\|^2 + \frac{1}{l}(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha) =$$
$$= -l + \frac{1}{l}(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha)$$

so

$$l + \Delta l \approx \frac{1}{l}(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha)$$

The magnitude of the force in the spring between particles α and β becomes now

$$\boldsymbol{F}_{\alpha\beta} = k_{\alpha\beta}\left(l + \Delta l - r^0_{\alpha\beta}\right) = \frac{k_{\alpha\beta}}{r_{\alpha\beta}}(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha) - k_{\alpha\beta} r^0_{\alpha\beta}$$

In the initial configuration, the spring is aligned along $\boldsymbol{p}_\alpha - \boldsymbol{p}_\beta$. In the equilibrium configuration it is aligned along $\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha$. The angle between these two alignments is governed by

$$\cos\theta = \frac{(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha)}{\|\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha\|\|\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha\|} = \frac{(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha + \boldsymbol{u}_\beta - \boldsymbol{u}_\alpha)}{l \cdot (l + \Delta l)} =$$
$$= \frac{l^2 + (\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha)}{l \cdot (l + \Delta l)}$$

Since we assumed small deformations, we get

$$\frac{(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha)}{l^2} \leq \frac{\|\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha\|\|\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha\|}{l^2} = \frac{\|\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha\|}{l} \ll 1$$

and $\frac{\Delta l}{l} \ll 1$. Hence $\cos\theta \approx 1$ which implies that $\theta \approx 0$ and $\boldsymbol{F}_{\alpha\beta}$ is approximately aligned along $\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha$, i.e.

$$\boldsymbol{F}_{\alpha\beta} = \left[\frac{k_{\alpha\beta}}{r_{\alpha\beta}}(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha) - k_{\alpha\beta} r^0_{\alpha\beta}\right] \frac{(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha)}{r_{\alpha\beta}}$$

where $\left[\frac{k_{\alpha\beta}}{r_{\alpha\beta}}(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha) - k_{\alpha\beta} r^0_{\alpha\beta}\right]$ is a magnitude and $\frac{(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha)}{r_{\alpha\beta}}$ is a direction.

At this point we are ready to introduce the equillibrium equations for a linear model. At each particle, force equilibrium has to hold. Thus

$$\sum_{\beta} \boldsymbol{F}_{\alpha\beta} = \boldsymbol{0}, \quad \alpha = 1, 2, \ldots, N$$

which can be rewritten as

$$\sum_{\beta} \left(\frac{k_{\alpha\beta}}{r_{\alpha\beta}} (\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) \cdot (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha) \right) \frac{\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha}{r_{\alpha\beta}} = \sum_{\beta} \frac{k_{\alpha\beta} r^0_{\alpha\beta}}{r_{\alpha\beta}} (\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha) = \boldsymbol{0}$$

Note that $r_{\alpha\beta} = \sqrt{(\hat{x}_\beta - \hat{x}_\alpha)^2 + (\hat{y}_\beta - \hat{y}_\alpha)^2 + (\hat{z}_\beta - \hat{z}_\alpha)^2}$. So at each particle we have 3 equations (corresponding to each component of the vector equation).

For example, equations in x direction

$$\sum_{\beta} \frac{k_{\alpha\beta}[(\hat{x}_\beta - \hat{x}_\alpha)(x_\beta - x_\alpha) + (\hat{y}_\beta - \hat{y}_\alpha)(y_\beta - y_\alpha) + (\hat{z}_\beta - \hat{z}_\alpha)(z_\beta - z_\alpha)](\hat{x}_\beta - \hat{x}_\alpha)}{(\hat{x}_\beta - \hat{x}_\alpha)^2 + (\hat{y}_\beta - \hat{y}_\alpha)^2 + (\hat{z}_\beta - \hat{z}_\alpha)^2} =$$

$$= \sum_{\beta} \frac{k_{\alpha\beta} r^0_{\alpha\beta} (\hat{x}_\beta - \hat{x}_\alpha)}{\sqrt{(\hat{x}_\beta - \hat{x}_\alpha)^2 + (\hat{y}_\beta - \hat{y}_\alpha)^2 + (\hat{z}_\beta - \hat{z}_\alpha)^2}} = 0$$

equations in y direction

$$\sum_{\beta} \frac{k_{\alpha\beta} r^0_{\alpha\beta} (\hat{y}_\beta - \hat{y}_\alpha)}{\sqrt{(\hat{x}_\beta - \hat{x}_\alpha)^2 + (\hat{y}_\beta - \hat{y}_\alpha)^2 + (\hat{z}_\beta - \hat{z}_\alpha)^2}} = 0$$

and in z direction

$$\sum_{\beta} \frac{k_{\alpha\beta} r^0_{\alpha\beta} (\hat{z}_\beta - \hat{z}_\alpha)}{\sqrt{(\hat{x}_\beta - \hat{x}_\alpha)^2 + (\hat{y}_\beta - \hat{y}_\alpha)^2 + (\hat{z}_\beta - \hat{z}_\alpha)^2}} = 0$$

4. Netwon-Raphson method for non-linear model allowing large deformations

We start with formulating the equillibrium equations for non-linear model. At each particle α, force equilibrium has to hold. Thus

$$\sum_{\beta} \boldsymbol{F}^{\alpha}_{\alpha\beta} = 0, \quad \alpha = 1, 2, \ldots, N$$

which can be rewritten as

$$\boldsymbol{F}(\boldsymbol{x}) = 0$$

where $\boldsymbol{F}$ is a composition of $\boldsymbol{F}^{\alpha}$ for all $\alpha = 1, 2, \ldots, N$, and $\boldsymbol{x}$ is a composition of $x_1, x_2, \ldots, x_N$.

From Taylor expansion of $\boldsymbol{F}$ around guess $\boldsymbol{x}^i$ we have

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x}^i) + \partial\boldsymbol{F}(\boldsymbol{x}^i, \Delta\boldsymbol{x}^i) + H.O.T.$$

where $\boldsymbol{x} = \boldsymbol{x}^i + \Delta\boldsymbol{x}^i$. Since

$$\boldsymbol{F}(\boldsymbol{x}) \approx \boldsymbol{F}(\boldsymbol{x}^i) + \partial\boldsymbol{F}(\boldsymbol{x}^i, \Delta\boldsymbol{x}^i)$$

then $\Delta\boldsymbol{x}^i$ can be approximated from

$$\partial\boldsymbol{F}(\boldsymbol{x}^i, \Delta\boldsymbol{x}^i) = -\boldsymbol{F}(\boldsymbol{x}^i)$$

The Newton-Raphson algorithm can be summarized in the following way:

1 Select the initial configuration of particles $\boldsymbol{x}^0$
2 Compute the right hand side of $\partial\boldsymbol{F}(\boldsymbol{x}^i, \Delta\boldsymbol{x}^i) = -\boldsymbol{F}(\boldsymbol{x}^i)$
3 Compute the Jacobian for the left hand side of $\partial\boldsymbol{F}(\boldsymbol{x}^i, \Delta\boldsymbol{x}^i) = -\boldsymbol{F}(\boldsymbol{x}^i)$
4 Solve the equation $\partial\boldsymbol{F}(\boldsymbol{x}^i, \Delta\boldsymbol{x}^i) = -\boldsymbol{F}(\boldsymbol{x}^i)$ for $\Delta\boldsymbol{x}^i$
5 Update $\boldsymbol{x}^{i+1} = \boldsymbol{x}^i + \Delta\boldsymbol{x}^i$
6 if $\|\Delta\boldsymbol{x}^i\|$ is not small enough, then $i = i + 1$, go to 2.

It should be emphesized that line 4 of the above algorithm involves the iterations of the GMRES iterative solver. In this version of the Newton-Raphson algorithm we compute directly the Jacobian matrix in line 3. There are some other versions of the Newton-Raphson algorithm where the Jacobian matrix is not computed directly but approximated, however in our case we allow for large deformations and the Jacobian matrix may be very different from one iteration to another, so we decided the compute the Jacobian matrix directly, using the analytical formulae.

Now, we need to specify how to compute the right hand side in line 2 and the Jacobian matrix in line 3.

The right hand side is just equal to

$$\boldsymbol{F}(\boldsymbol{x}^i) = \sum_\beta \boldsymbol{F}_{\alpha\beta}(\boldsymbol{x}^i_\alpha, \boldsymbol{x}^i_\beta) = \sum_\beta k_{\alpha\beta}(\|\boldsymbol{x}^i_\beta - \boldsymbol{x}^i_\alpha\| - r^0_{\alpha\beta}) \cdot \frac{\boldsymbol{x}^i_\beta - \boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta - \boldsymbol{x}^i_\alpha\|}$$

Here $\boldsymbol{x}^i_\alpha = (x^i_\alpha, y^i_\alpha, z^i_\alpha)$ and $\|\boldsymbol{x}^i_\alpha\| = \sqrt{{x^i_\alpha}^2 + y^i_\alpha + {z^i_\alpha}^2}$.

Now we will derive the formula for the Jacobian

$$\partial\boldsymbol{F}(\boldsymbol{x}^i, \Delta\boldsymbol{x}^i) = \sum_\beta \partial\boldsymbol{F}_{\alpha\beta}(\boldsymbol{x}^i_\alpha, \boldsymbol{x}^i_\beta; \Delta\boldsymbol{x}^i_\alpha, \Delta\boldsymbol{x}^i_\beta)$$

We need to compute

$$\begin{aligned}\partial\boldsymbol{F}_{\alpha\beta}(\boldsymbol{x}^i_\alpha, \boldsymbol{x}^i_\beta; \Delta\boldsymbol{x}^i_\alpha, \Delta\boldsymbol{x}^i_\beta) = \\ = \lim_{\theta\to 0} \frac{1}{\theta}\left[\boldsymbol{F}_{\alpha\beta}(\boldsymbol{x}^i_\alpha + \theta\Delta\boldsymbol{x}^i_\alpha, \boldsymbol{x}^i_\beta + \theta\Delta\boldsymbol{x}^i_\beta) - \boldsymbol{F}_{\alpha\beta}(\boldsymbol{x}^i_\alpha, \boldsymbol{x}^i_\beta)\right] =\end{aligned}$$

$$= \lim_{\theta\to 0}\frac{1}{\theta}\left[k_{\alpha\beta}(\|\boldsymbol{x}^i_\beta+\theta\Delta\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha-\theta\Delta\boldsymbol{x}^i_\alpha\|-r^0_{\alpha\beta})\frac{\boldsymbol{x}^i_\beta+\theta\Delta\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha-\theta\Delta\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta+\theta\Delta\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha-\theta\Delta\boldsymbol{x}^i_\alpha\|}-\right.$$
$$\left.k_{\alpha\beta}(\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|-r^0_{\alpha\beta})\frac{\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|}\right]=$$
$$= \lim_{\theta\to 0}\left[k_{\alpha\beta}(\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|-r^0_{\alpha\beta})\right]\frac{\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|}+$$
$$+\lim_{\theta\to 0}\frac{1}{\theta}\left[k_{\alpha\beta}\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|\right]\frac{\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|}$$
$$-\lim_{\theta\to 0}\frac{1}{\theta}k_{\alpha\beta}[\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|]\frac{\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|}$$
$$-\lim_{\theta\to 0}\frac{1}{\theta}k_{\alpha\beta}r^0_{\alpha\beta}\left[\frac{1}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|}-\frac{1}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|}\right](\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha)=$$
$$=k_{\alpha\beta}\left[\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|-r^0_{\alpha\beta}\right]\frac{\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|}$$
$$-\lim_{\theta\to 0}\frac{1}{\theta}k_{\alpha\beta}r^0_{\alpha\beta}\left[\frac{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|^2-\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|^2}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|+\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|}\right]\cdot$$
$$\cdot\frac{\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|\cdot\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|}=$$
$$=k_{\alpha\beta}\left[\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|-r^0_{\alpha\beta}\right]\frac{\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|}$$
$$-\lim_{\theta\to 0}\frac{1}{\theta}k_{\alpha\beta}r^0_{\alpha\beta}\left[\frac{-2\theta\left(\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\right)\cdot\left(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha\right)-\theta^2\|\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha\|^2}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|+\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|}\right]$$
$$\cdot\frac{\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|\cdot\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha+\theta(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)\|}=$$
$$=k_{\alpha\beta}\left[\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|-r^0_{\alpha\beta}\right]\frac{\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|}+$$
$$+k_{\alpha\beta}r^0_{\alpha\beta}\left[\frac{(\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha)\cdot(\Delta\boldsymbol{x}^i_\beta-\Delta\boldsymbol{x}^i_\alpha)}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|^2}\right]\cdot\frac{\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha}{\|\boldsymbol{x}^i_\beta-\boldsymbol{x}^i_\alpha\|}\quad(6)$$

We can reformulate the equations in matrix-vector notation, i.e.

$$\partial\boldsymbol{F}_{\alpha\beta}(\boldsymbol{x}^i;\Delta\boldsymbol{x}^i)=\begin{pmatrix}\boldsymbol{K}^{\alpha\alpha} & \boldsymbol{K}^{\alpha\beta}\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{x}^i_\alpha\\ \Delta\boldsymbol{x}^i_\beta\end{pmatrix}$$

In the following, we use Einstein's notation $a_ib_i=a_1b_1+a_2b_2+a_3b_3$. We define Cartesian base vectors $\boldsymbol{e}_1=(1,0,0)$, $\boldsymbol{e}_2=(0,1,0)$ and $\boldsymbol{e}_3=(0,0,1)$. Hence $\boldsymbol{b}=b_i\boldsymbol{e}_i$

and $\mathbf{a} \cdot \boldsymbol{b} = (a_i \boldsymbol{e}_i) \cdot (b_j \boldsymbol{e}_j) = a_i b_j (\boldsymbol{e}_i \cdot \boldsymbol{e}_j) = a_i b_j \delta_{ij} = a_{ii}$. Now we can write

$$\begin{array}{cc} \Delta \boldsymbol{x}^i_\alpha = \Delta \xi^i_{\alpha,j} \boldsymbol{e}_j & \boldsymbol{x}^i_\alpha = \xi^i_{\alpha,j} \boldsymbol{e}_j \\ \underbrace{\Delta \boldsymbol{x}^i_\beta = \Delta \xi^i_{\beta,j} \boldsymbol{e}_j}_{\text{unknown}} & \underbrace{\boldsymbol{x}^i_\beta = \xi^i_{\beta,j} \boldsymbol{e}_j}_{\text{known}} \end{array}$$

Furthermore, we define

$$\Phi^i_{\alpha\beta} = k_{\alpha\beta} \frac{[\|\boldsymbol{x}^i_\beta - \boldsymbol{x}^i_\alpha\| - r^0_{\alpha\beta}]}{\|\boldsymbol{x}^i_\beta - \boldsymbol{x}^i_\alpha\|}$$

and

$$\varphi^i_{\alpha\beta} = \frac{k_{\alpha\beta} r^0_{\alpha\beta}}{\|\boldsymbol{x}^i_\beta - \boldsymbol{x}^i_\alpha\|^3}$$

Now, we can rewrite equation (6)

$$\begin{aligned} &\partial \boldsymbol{F}_{\alpha\beta}\left(\boldsymbol{x}^i_\alpha, \boldsymbol{x}^i_\beta; \Delta \boldsymbol{x}^i_\alpha, \Delta \boldsymbol{x}^i_\beta\right) = \\ &= \Phi^i_{\alpha\beta}(\Delta \xi^i_{\beta,j} - \Delta \xi^i_{\alpha,j}) \boldsymbol{e}_j + \varphi^i_{\alpha\beta}\left[(\xi^i_{\beta,k} - \xi^i_{\alpha,k}) \boldsymbol{e}_k \cdot (\xi^i_{\beta,l} - \xi^i_{\alpha,l}) \boldsymbol{e}_l\right](\xi^i_{\beta,j} - \xi^i_{\alpha,j}) \boldsymbol{e}_j = \\ &= \Big(\Phi^i_{\alpha\beta}(\Delta \xi^i_{\beta,j} - \Delta \xi^i_{\alpha,j}) + \\ &\qquad + \varphi^i_{\alpha\beta}\left[\xi^i_{\beta,l}\Delta \xi^i_{\beta,l} - \xi^i_{\beta,l}\Delta \xi^i_{\alpha,l} - \xi^i_{\alpha,l}\Delta \xi^i_{\beta,l} + \Delta \xi^i_{\alpha,l}\xi^i_{\alpha,l}\right] \times (\xi^i_{\beta,j} - \xi^i_{\alpha,j})\Big) \boldsymbol{e}_j = \\ &= \Bigg(- \left[\Phi^i_{\alpha\beta} + \varphi^i_{\alpha\beta}(\xi^i_{\beta,j} - \Delta \xi^i_{\alpha,j})^2\right] \Delta \xi^i_{\alpha,j} - \sum_{l \neq j} \varphi^i_{\alpha\beta}(\xi^i_{\beta,l} - \xi^i_{\alpha,l})(\xi^i_{\beta,j} - \xi^i_{\alpha,j}) \Delta \xi^i_{\alpha,l} + \\ &\quad + \left[\Phi^i_{\alpha\beta} + \varphi^i_{\alpha\beta}(\xi^i_{\beta,j} - \xi^i_{\alpha,j})^2\right] \Delta \xi^i_{\beta,j} + \sum_{l \neq j} \varphi^i_{\alpha\beta}(\xi^i_{\beta,l} - \xi^i_{\alpha,l})(\xi^i_{\beta,j} - \xi^i_{\alpha,j}) \Delta \xi^i_{\beta,l} \Bigg) \boldsymbol{e}_j \end{aligned} \quad (7)$$

Thus

$$\boldsymbol{K}^{\alpha\alpha} = \begin{pmatrix} -(\Phi^i_{\alpha\beta} + \varphi^i_{\alpha\beta}(x^i_\beta - x^i_\alpha)^2) & -\varphi^i_{\alpha\beta}(y^i_\beta - y^i_\alpha)(x^i_\beta - x^i_\alpha) & -\varphi^i_{\alpha\beta}(z^i_\beta - z^i_\alpha)(x^i_\beta - x^i_\alpha) \\ -\varphi^i_{\alpha\beta}(x^i_\beta - x^i_\alpha)(y^i_\beta - y^i_\alpha) & -(\Phi^i_{\alpha\beta} + \varphi^i_{\alpha\beta}(y^i_\beta - y^i_\alpha)^2) & -\varphi^i_{\alpha\beta}(y^i_\beta - y^i_\alpha)(z^i_\beta - z^i_\alpha) \\ -\varphi^i_{\alpha\beta}(x^i_\beta - x^i_\alpha)(z^i_\beta - z^i_\alpha) & -\varphi^i_{\alpha\beta}(y^i_\beta - y^i_\alpha)(z^i_\beta - z^i_\alpha) & -(\Phi^i_{\alpha\beta} + \varphi^i_{\alpha\beta}(z^i_\beta - z^i_\alpha)^2) \end{pmatrix}$$

where submatrices are located in α and β rows and columns.

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{K}^{\alpha\alpha} & \boldsymbol{K}^{\alpha\beta} \\ \boldsymbol{K}^{\beta\alpha} & \boldsymbol{K}^{\beta\beta} \end{pmatrix}$$

and $\boldsymbol{K}^{\alpha\beta} = -\boldsymbol{K}^{\alpha\alpha}$, $\boldsymbol{K}^{\beta\beta} = \boldsymbol{K}^{\alpha\alpha}$, $\boldsymbol{K}^{\beta\alpha} = \boldsymbol{K}^{\alpha\beta}$.

5. Numerical experiments

This chapter presents description of the two numerical experiments. The first experiment was performed in order to test the convergence of the Newton-Raphson algorithm with GMRES iterative solver utilized to solve the non-linear formulation allowing for large deformations. The second experiment was performed in order to test the convergence of the PCG iterative solver used to solve the linear formulation assuming smal deformations. Both experiments are performed over the 3D cube of $50 \times 50 \times 50$ [nm] grid with particles, with interparticle forces parameters obtained by using the Monte-Carlo simulation described in [2]. The number of equations is equal to the total number of particles $50^3 = 125\,000$. The interparticle forces between particles forming a polymer network was assumed to be stronger than interparticle forces between another particles. Each particle is assumed to interact with 26 neighbors, in other words the cross-diagonal bounds are also allowed.

5.1. Convergence of the iterative solver for the non-linear formulation

The solution process starts with performing the first linearization – computing the Jacobian matrix for the first time, and then utilizing the GMRES algorithm to solve the linearized problem. We utilize the GMRES algorithm from the SLATEC library [7]. We also tried to utilize the PCG (Preconditioned Concjugate Gradients) solver, but it doesn't converge, since the Jacobain matrix is not positive definite. We also applied the diagonal preconditioner for GMRES. The snapshots of the convergence of this first iteration after 8, 12, 18, 32 and 48 iterations are presented in Figures 2, 3, 4, 5, 6 and 7, respectively. The blue color of some particles denote the beginnig of the polymer network (the initial particles). The accuracy of the GMRES algorithm in these iterations is summarized in Table 1.

Table 1
Convergence of the GMRES alorithm in the first iteration of the Newton Raphson algorithm

Iteration	8	12	18	32	48
GMRES error	0.05	0.04	0.03	0.02	0.01

The obtained shape of the deformed polymer network conforms our expectations about the shrinkage of the sample, however the sample is too high in comparison with the experimental data presented in Figure 1.

The resulting error for the first step of Newton-Raphson algorithm was $\|\Delta \boldsymbol{x}^i\| = 9607.62$ where we consider the Eucledean norm of the displacement vector. This is just the norm of the displacements of all particles from the initial configuration to the first iteration configuration summarized in Figure 8. Notice that the problem is formulated in nm units.

Table 2
Convergence of the Newton-Raphson algorithm

Iteration	1	2	3	4	5	6	7
$\|\Delta \boldsymbol{x}^i\|$	9607.62	12483.78	2496.97	537.48	165.48	37.39	9.22

The following iterations of the Newton-Raphson algorithm adjusts the shape of the sample, as it is illustrated in Figures 8, 9, 10, 11 and 12. The figures present the configuration of the mesh after the final iteration of the GMRES algorithm in corresponding iterations of the Newton-Raphson loop. The corresponding norms of the displacement vectors for the Newton-Raphson algorithm are summarized in Table 2.

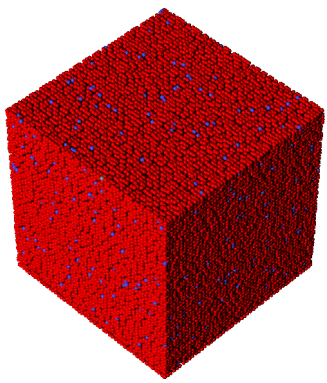

Fig. 2. Grid configuration after 8 iterations of the GMRES algorithm

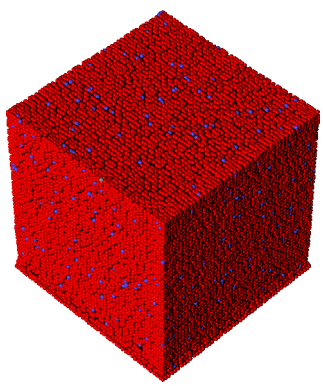

Fig. 3. Grid configuration after 12 iterations of the GMRES algorithm

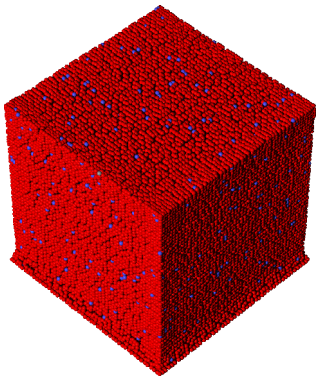

Fig. 4. Grid configuration after 18 iterations of the GMRES algorithm

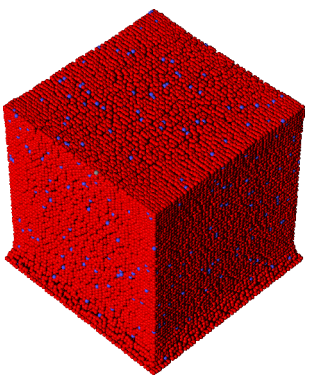

Fig. 5. Grid configuration after 32 iterations of the GMRES algorithm

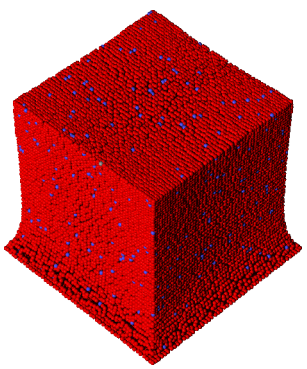

Fig. 6. Grid configuration after 48 iterations of the GMRES algorithm, which is also the configuration after the first Newton-Raphson loop

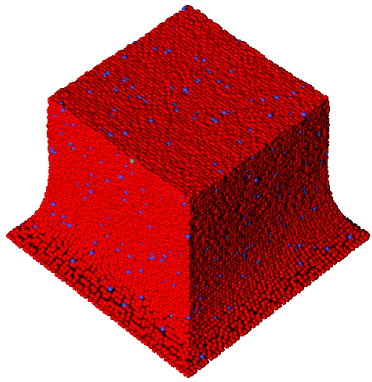

Fig. 7. Grid configuration after second iteration of the Newton-Raphson loop

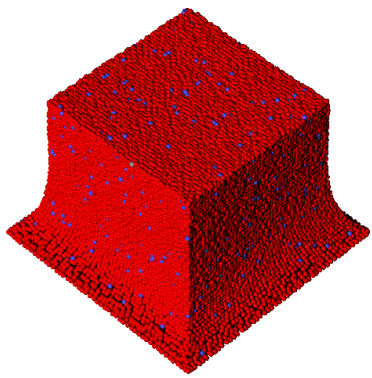

Fig. 8. Grid configuration after third iteration of the Newton-Raphson loop

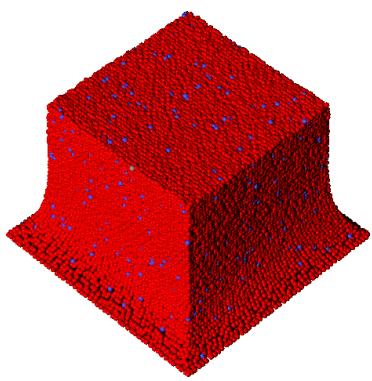

Fig. 9. Grid configuration after fourth iteration of the Newton-Raphson loop

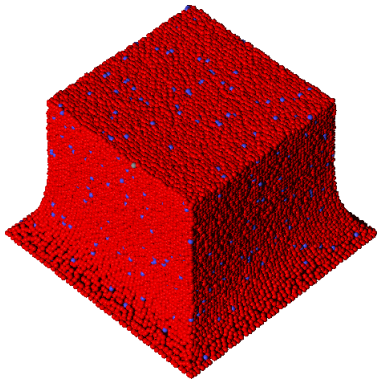

Fig. 10. Grid configuration after fifth iteration of the Newton-Raphson loop

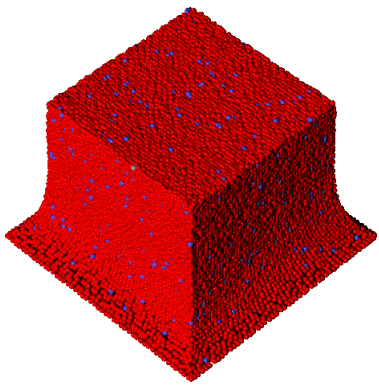

Fig. 11. Grid configuration after sixth iteration of the Newton-Raphson loop

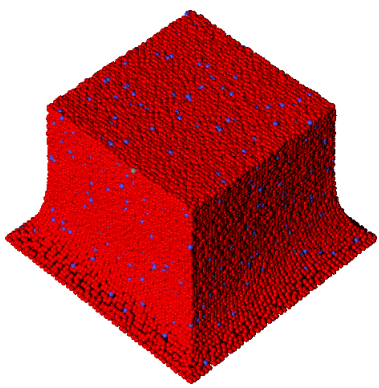

Fig. 12. Grid configuration after seventh iteration of the Newton-Raphson loop

5.2. Convergence of the iterative solver for the linear formulation

In the case of the linear model we just have a single loop perfoming iterations of Precoditioned Conjugate Gradients (PCG) algorithm. We utilize the PCG algorithm from the SLATEC library [7]. The snapshots of the convergence after particular numbers of iterations listed in Table 4 are presented in Figures 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 and 23, respectively. The blue color of some particles denote the beginnig of the polymer network (the initial particles). The accuracy of the PCG algorithm in these iterations is also summarized in Table 4.

Table 3
Convergence of the PCG alorithm

Accuracy	0.5	0.1	0.04	0.03	0.02	0.005	0.01	0.005	0.0025	0.0001
Iterations	4	20	82	134	264	1362	7242	17952	32768	52378

The linear problem solved by means of the PCG algorithm has also provided the shape of the deformed polymer network that conforms our expectations about the shrinkage of the sample. However the price we paid is the number of iterations. We needed 52378 iterations of the PCG algorithm, which took 6063 seconds, while the Newton-Raphson with GMRES for the non-linear model required 583 seconds only.

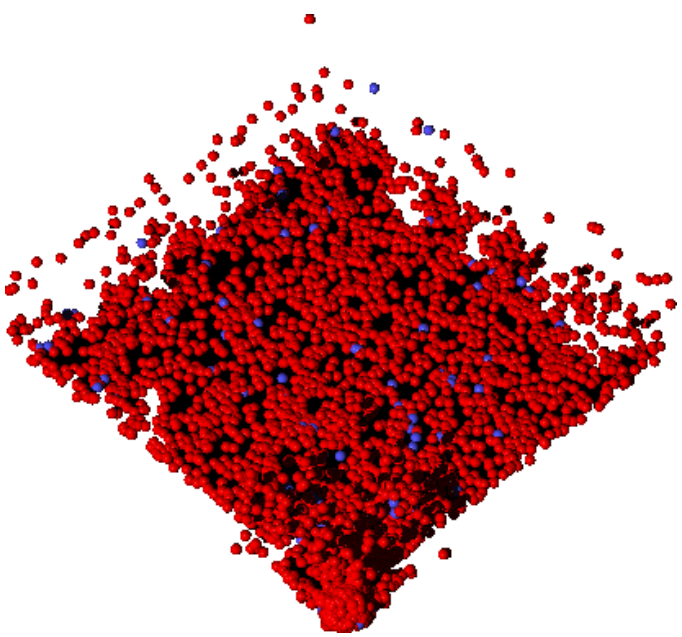

Fig. 13. Grid configuration after 4 iterations of the PCG algorithm

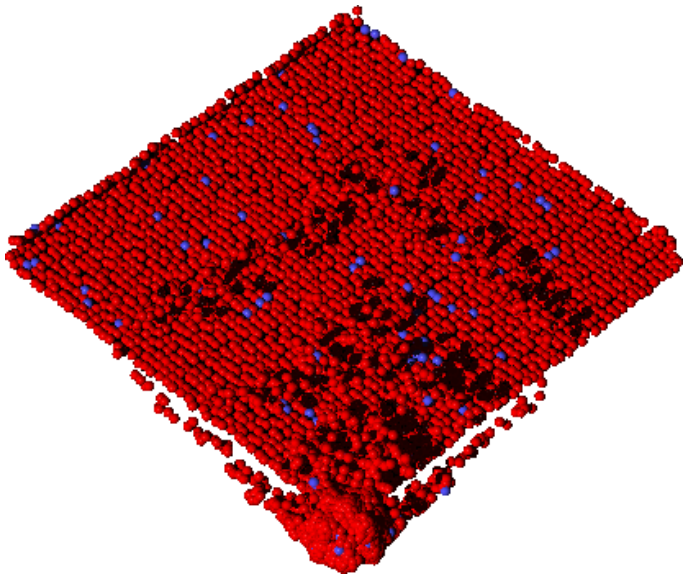

Fig. 14. Grid configuration after 20 iterations of the PCG algorithm

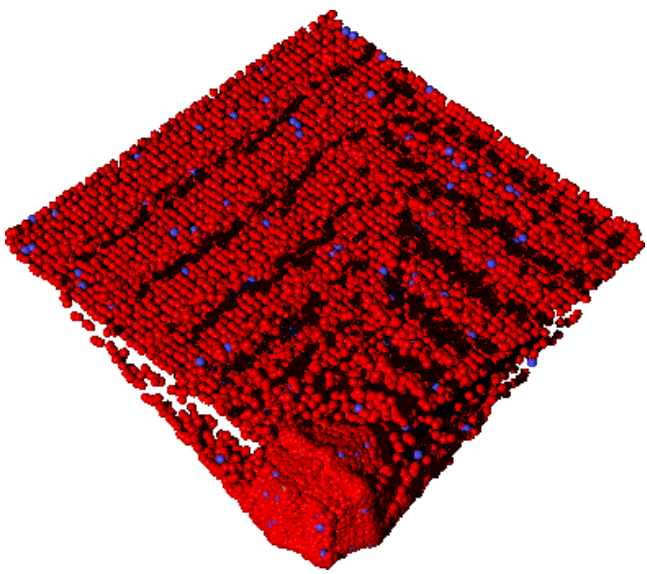

Fig. 15. Grid configuration after 82 iterations of the PCG algorithm

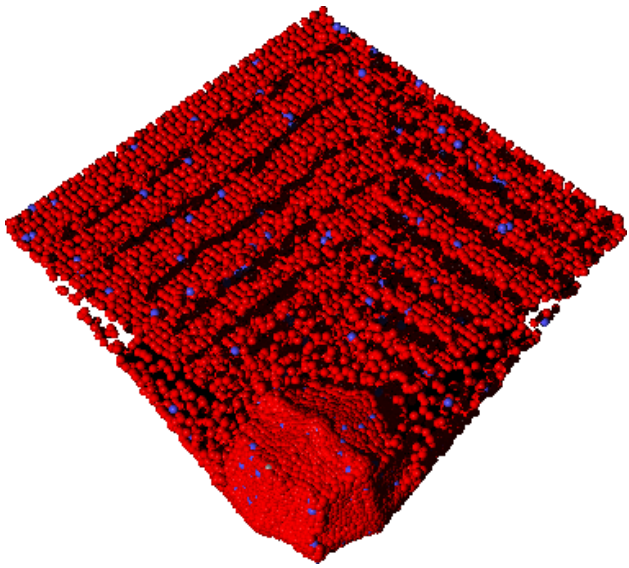

Fig. 16. Grid configuration after 134 iterations of the PCG algorithm

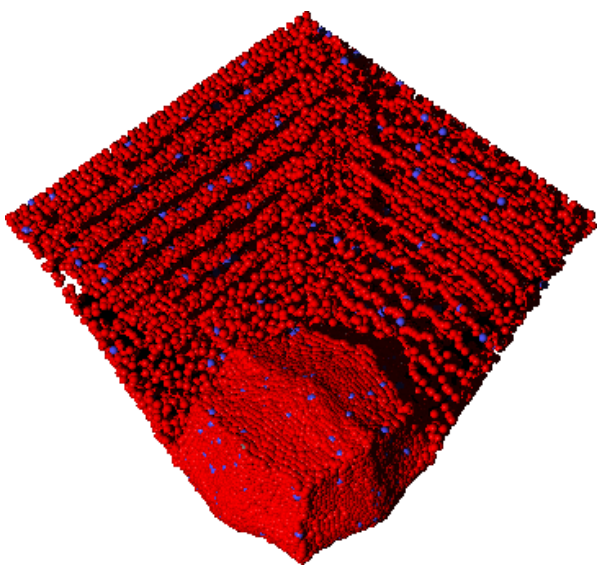

Fig. 17. Grid configuration after 264 iterations of the PCG algorithm

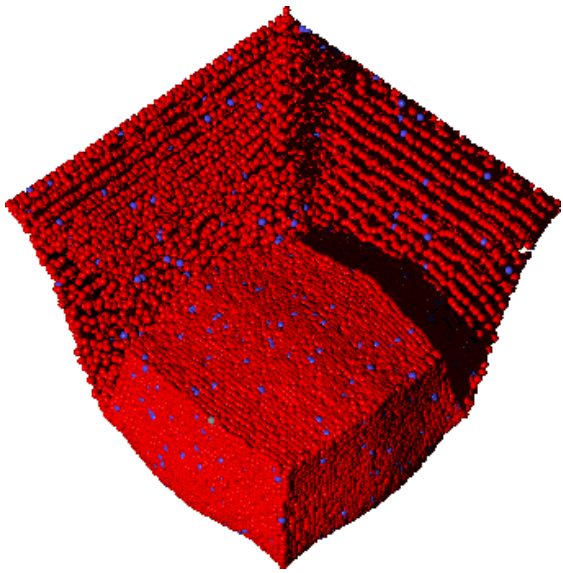

Fig. 18. Grid configuration after 626 iterations of the PCG algorithm

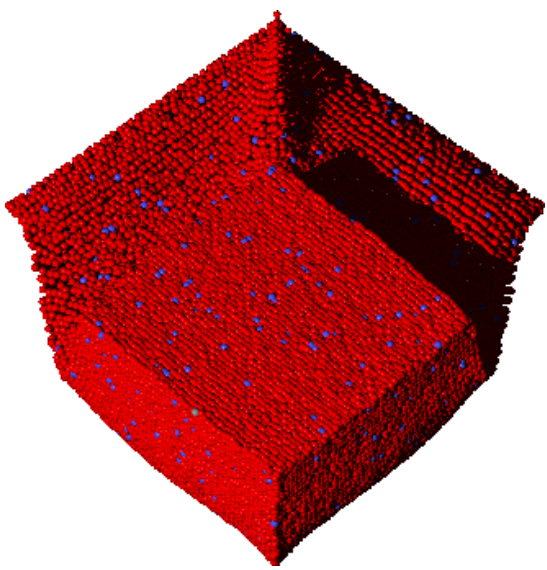

Fig. 19. Grid configuration after 1362 iterations of the PCG algorithm

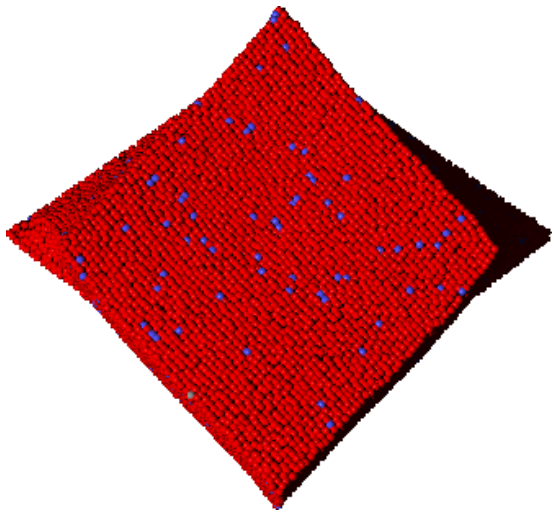

Fig. 20. Grid configuration after 7424 iterations of the PCG algorithm

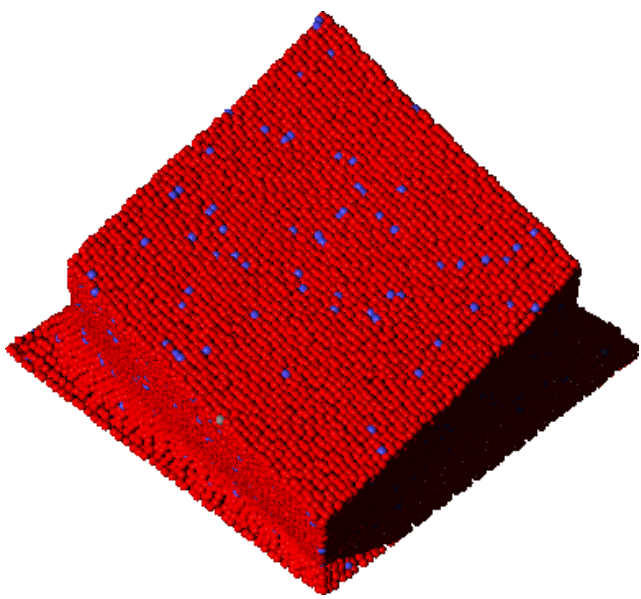

Fig. 21. Grid configuration after 17952 iterations of the PCG algorithm

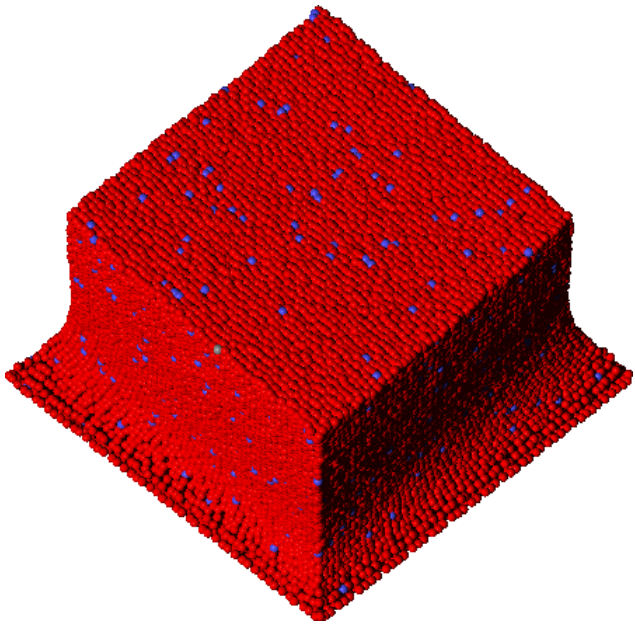

Fig. 22. Grid configuration after 32768 iterations of the PCG algorithm

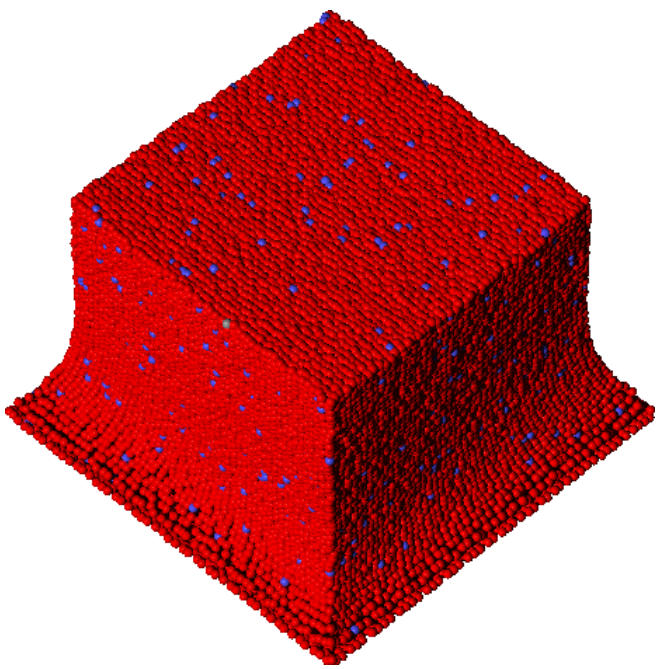

Fig. 23. Grid configuration after 52378 iterations of the PCG algorithm

6. Conclusions

The paper presented the convergence analysis of the Newton-Raphson algorithm interfaced with the iterative GMRES solver used to solve the molecular statics model for the densification of the polymerized network during the Step-and-Flash Imprint Lithography simulations. The convergence was ilustrated with snapshots presenting the deformation of the sample after particular iterations of the GMRES and Newton-Raphson method. We conclude that the algorithm converged well, and that the snapshots are nice way of ilustrating the convergence process for the iterative solvers. We also compared the non-linear algorithm allowing for large deformations with the linear one, assuming small deformations. We concluded that the solution of the linear problem takes one order of magnitude more time than the solution of the non-linear one.

Acknowledgements

The work reported in this paper was supported by the Polish MNiSW grant no. NN 501 120 836.

References

[1] Bailey T. C., Colburn M. E., Choi B. J., Grot A., Ekerdt J. G., Sreenivasan S. V., Willson C. G.: *Step and Flash Imprint Lithography: A Low-Pressure, Room Temperature Nanoimprint Patterning Process'*. Alternative Lithography. Unleashing the Potentials of Nanotechnology, C. Sotomayor Torres, Editor. Elsevier, 2002.

[2] Burns R. L., Johnson, S. C., Schmid G. M., Kim E. K., Dickey M. D., Meiring J., Burns S. D., Stacey N. A., Willson C. G.: *Mesoscale modeling for SFIL simulating polymerization kinetics and densification*. Proc. of SPIE (International Society for Optics and Photonics), vol. 5374, 2004, pp. 348–360.

[3] Colburn M., Suez I., Choi B. J., Meissl M., Bailey T., Sreenivasan S. V., Ekerdt J. G., Willson C. G.: *Characterization and modeling of volumetric and mechanical properties for step and flash imprint lithography photopolymers*. Journal of Vacuum Science and Technology B., vol. 19 issue 6, 2001, pp. 2685–2689.

[4] Bailey T., Smith B., Choi B. J., Colburn M., Meissl M., Sreenivasan S. V., Ekerdt J. G., Willson C. G.: *Step and flash imprint lithography: Defect analysis*. Journal of Vacuum Science and Technology B., vol. 19 issue 6, 2001, p. 2806–2810.

[5] Demkowicz L., Kurtz J., Pardo D., Paszyński M., Rachowicz W., Zdunek A.: *Computing with hp-Adaptive Finite Element Method. Vol. II. Frontiers: Three Dimensional Elliptic and Maxwell Problems*. Chapmann & Hall / CRC Applied Mathematics & Nonlinear Science, 2007.

[6] Paszyński M., Romkes A., Collister E., Meiring J., Demkowicz L., Willson G. C.: *On the Modeling of Step-and-Flash Imprint Lithography using Molecular Statics Models*. ICES Report 05-38, The University of Texas in Austin, 2005.

[7] SLATEC Common Mathematical Library `http://www.netlib.org/slatec`