Advances in Science and Technology Research Journal
Volume 11, Issue 2, June 2017, pages 51–57
DOI: 10.12913/22998624/68730

Research Article

APPLICATION OF AN EQUIVALENT TRUSS MODEL FOR DETERMINING THE STRESS STATE IN MULTI-PHASE MATERIALS WITH CELLULAR AUTOMATA METHOD

Anna Staszczyk[1], Jacek Sawicki[1]

[1] Institute of Materials Science and Engineering, Faculty of Mechanical Engineering, Lodz University of Technology, 1/15 Stefanowskiego Street, 90-924 Łódź, Poland, e-mail: anna.staszczyk@p.lodz.pl

Received: 2017.01.23
Accepted: 2017.03.09
Published: 2017.06.01

ABSTRACT

The Cellular Automata represent a universal method of modelling and simulation. They enable the performance of calculations for even the most complex processes and phenomena. They are also used successfully in mechanical and material engineering. In this paper, the concept of application of the Cellular Automata method for simulating the behaviour of material under stress is presented. The proposed numerical algorithm created performs a number of calculations of local stress states in the structure of precipitation hardened material. The principle of its operation is based on the application of the equivalent truss model, which is often used in the optimisation and design of structures. In this paper, this model was used to simulate a system embodying a section of the material containing various phases with different mechanical properties.

Keywords: Cellular Automata, numerical methods, stress state, strength of materials, precipitation hardening.

INTRODUCTION

The equivalent truss model was applied successfully to optimise the structural topology. Optimisation methods were applied to elements in both macro- and microscopic scale [1]. All the theoretical aspects of modern structural optimization issues were described by Rozvany in his work [2]. In that work, this method is used to solve the stress state in a deformed material. The theoretical model is implemented into the algorithm based on the Cellular Automata method.

The notion of Cellular Automata (abbreviated as CA) was formulated for the first time in 1940s by John von Neumann, a mathematician and computer scientist of Hungarian origin. A cellular automaton is a self-replicating system consisting of a cellular grid where every cell has a finite number of states [3]. Such a grid can have any number of dimensions. Automatons in one, two or three dimensions are those used the most often. For every cell, the so-called neighbourhood is defined, that is, a set of cells that remain in direct contact with it and the states of which are taken into account when establishing the state of the cell. A cellular automaton is a discrete model. Its space, number of possible states and time during which it evolves are finite and countable. The system evolution consists in the cell state being established in every pass on the basis of state of neighbourhood cells and state of the cell concerned during the previous step. The above dependencies are defined via passage rules. The principles of operation of cellular automata have been widely documented [3, 4, 5].

The equivalent truss model is based on the assumption that a solid body, being a continuous centre, can be presented in the form of a truss with a specified geometry [6]. Cellular Automata are a frequently used method of implementing a model into numerical algorithms. Using them for the above mentioned purpose was descried

by Gurdal and Abdalla [7, 8], Kita and Toyoda [9], Gurdla and Tatting [6, 10] and Abdalla and Zakhama [11]. Until now the truss model has not been applied in Cellular Automata for determining the stresses inside multi-phase materials.

NUMERICAL MODEL

The numerical model assumed writing an algorithm that creates the two-dimensional geometry of a multi-phase material and then executes a simulation of the material deformation process and calculates stresses at every point of the Cellular Automaton grid.

Computational domain

The computational domain of an automaton is constituted by a system which is a two-dimensional digital representation of a square section of the material containing a circular or elliptical hardening phase precipitate in its centre. The precipitate geometry is created by the user who introduces its parameters into the software or established on the basis of a microscopic image of the actual particle observed. The types of geometry generated by the software include only circular or elliptical precipitates containing one or two phases of the material.

A cellular grid with a resolution of 200x200 is imposed onto the geometry. Every cell has fixed values of mechanical properties assigned, depending on its affiliation to a given phase of the material. A cell is embedded in the direct neighbourhood of the Moore type, i.e. it has 8 most adjacent neighbours.

The geometry is generated by the software in the form of a bitmap. Every single map pixel is a single automaton cell. The bitmap includes three colours, each of which corresponds to one of the material's phases. Black represents the material's matrix, while light and dark grey represent particular precipitation phases. When reading a pixel's colour, the algorithms assign appropriate material constants to a cell.

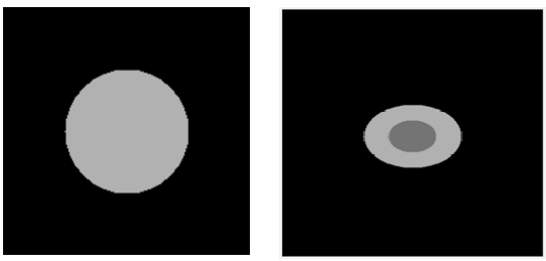

Fig. 1. Examples of geometry generated by the algorithm

The algorithm operates on dimensionless sizes. The size of a single pixel is equivalent to a length of 1. This enables large-scale simulation, without imposing specific dimensions of precipitates and sections of material. In reality, the scale of precipitate sizes in aluminium alloys is very small, which complicates the calculation of actual stresses.

Boundary conditions

A system which is the computational domain of an automaton is subjected to axial tensioning with a fixed force until the moment of achieving the deformation set. In the algorithm described, it was assumed that the lower edge of the system remains immobile, fixed to the substrate. The deformation is applied to the upper edge of the system. The value of this deformation is set by the user and expressed as a percentage. There are no boundary conditions on the side cells, so these are not fixed and are free to move.

Calculations

To make this a problem suitable for Cellular Automata calculation, we first have to change the continuum body into an equivalent truss model. Every body cell is treated as a truss node connected to eight cells of the Moore neighbourhood using rods with specific sections and material constants. There exist forces in these truss members when these are elongated or shortened. This kind of model is easily implemented in a cellular automata scheme. We update every node by positioning it so that the sum of the acting forces is zero. This also coincides with searching for the point of minimal potential energy. After sufficient iterations, the whole body will be very close to its point of minimal potential energy. This means we are converging on the actual solution. Of course this method needs a certain (large) amount of iterations depending on the number of nodes.

As an illustration of this method, we analysed a simple structure. This structure is a flat square made from a homogeneous material with Young's modulus E_1. In the middle, we model the precipitate by adding a ellipsoid area with a different Young's modulus E_2. This is the most basic way we can model an alloy with a certain pre-

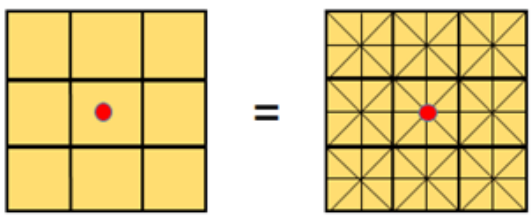

Fig. 2. Diagram of an equivalent truss model

cipitate on a microscopic scale. Of course, with this method, any number of different configurations can be analysed, but this one can be easily checked analytically.

As we are modelling a continuum body as a truss model, we need to relate the thickness of this continuum body and the cross-sectional area of the truss members. These can then be used in the calculation of the stresses and forces in our truss model. We can model this as a superposition of the diagonal and horizontal and vertical members. The members can be interpreted as the springs, with elastic constants corresponding with Young's modulus of the phase material, connecting the nodes being the cell centers [10]. The diagonal members have a cross-sectional area A_d and the horizontal and vertical members have a cross-sectional area A_o. These areas can be calculated by using the equivalence of strain energy between the truss cell and the continuum cell for given nodal displacements.

$$U = \left[\frac{E}{1-\nu^2}\left(\varepsilon_x{}^2 + \varepsilon_y{}^2 + 2\nu\varepsilon_x\varepsilon_y\right) + G\gamma_{xy}{}^2\right]\frac{l^2 t}{2} \quad (1)$$

$$U = \left[A_0\varepsilon_h{}^2 + A_0\varepsilon_v{}^2 + \sqrt{2}A_d\left(\varepsilon_{d_1}{}^2 + \varepsilon_{d_2}{}^2\right)\right]\frac{El}{2} \quad (2)$$

$$A_o = \frac{3}{4}lt \qquad A_d = \frac{3\sqrt{2}}{8}lt \quad (3)$$

Where l and t are the height and the thickness of the cell.

The starting point of this method is the forces acting on a joint in the mesh. These joints are equally spaced. We use each iteration of cellular automata to sequentially position the cell so that the sum of all forces is zero. This gives us a vector equation and a closed expression for the new position of each cell. If we do enough iterations, the resulting positions of each of the joints will be without internal forces, so we will have found the solution to the problem.

Here we are using a Moore neighbourhood, so we have 8 forces coming from 8 members acting on each joint.

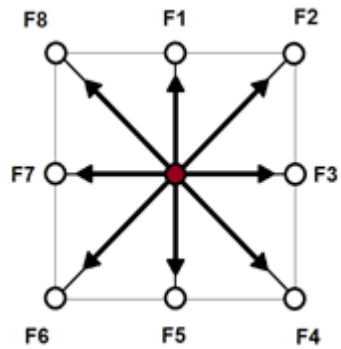

Fig. 3. Forces acting on a truss node

These are used in the force equations to determine the stresses and thus the position where the force is zero. These equations can be rearranged in the x- and y-direction. From these we can directly calculate the x- and y-position of each particle in each iteration.

$$F = E * A * \varepsilon + F_i \quad (4)$$

where: E – Young's modulus, A – member's cross-sectional area, ε – strain in a member, F_i – external force.

$$\Sigma F_x = 0 \quad (5)$$

$$\Sigma F_y = 0 \quad (6)$$

$$F_{xk} = F_k * \cos\theta \qquad k = 1, \dots, 8 \quad (7)$$

$$F_{yk} = F_k * \sin\theta \qquad k = 1, \dots, 8 \quad (8)$$

When the truss nodes change their location, the directional cosines for the constituent forces change their values. This results in the system balance location solution being reduced to a system of complex non-linear equations without an analytical solution. However, due to the fringe conditions adopted and the fact that the analysis is performed for a rigid continuous body, the angle values only change to a very minor extent. The changes in node location for a single algorithm iteration are at the level of 10^{-8}. In view of the above reasons it was assumed that omitting the angle changes would not have any significant impact on the simulation result. The system was reduced to linear equations, where the θ angle is always 45°. This type of solution is probably not going to perform well in the case of deformations of greater magnitude.

The original no-stress heights of the cells are equal to 1 (dimensionless). These can easily be adapted to the grid size. But as all formulas are

dimensionless, this does not matter. From these formulas, we note that the forces and the direction of the forces on the joint are dependent on how much and in which direction the actual position deviates from these no-stress lengths. The strains in the local points of the body are calculated as the deformation of the original no-stress lengths. Since the lengths of the cells are equal to 1 and their positions in the Cartesian coordinate system correspond with the coordinates of bitmap pixels (they differ by one), this deformation can be calculated as the difference between two positions of neighbouring cells and their original, undeformed gap.

$$\varepsilon = position[i,j] - position[i,j-1] - 1 \quad (9)$$

The stress values are later calculated using Hooke's law.

$$\sigma = \varepsilon * E \quad (10)$$

Convergence of the solution

In this method, each cell is dependent on the displacements of its neighbours to calculate its own displacement. For this we can use two different calculation methods; both have their drawbacks as well as benefits. As we are using a whole array of cells, we can calculate all of them in parallel or sequentially. Using parallel computing means we can use the true benefits of cellular automata: the short calculation time, but this requires special code and implementation. We store the old array of cells and calculate the new array directly using the old one. This is called a Jacobi iteration, but is inefficient for any process that converges. If we only have a sequential processor, we can help the rate of convergence by using the updated array of cells to calculate the new positions sequentially. This is called the Gauss-Seidel iteration method. The new values are calculated using the updated values. Smaller numbers of iterations are required to obtain convergence. This however eliminates the possibility of using parallel computing [12].

The convergence criterion has been set at 10^{-6} * MAX (maximum value of the positional change in the last iteration in the system). For the method adopted, the convergence is achieved after about 130 000 iterations.

Table 1. Adopted values of mechanical properties of Al 2024 alloy after processing

Phase of the material	Young's modulus [MPa]	Poisson ratio
Matrix	73 000	0,33
Precipitate	240 000	0,33

RESULTS

The calculations were performed assuming the mechanical properties for the 2024 aluminium alloy containing precipitates after ageing processing. The values of the Young module and Poisson coefficient for a given material are indicated in Table 1. All the simulations were performed for the total deformation in a tensioning direction equal to 0.05%.

The simulation results are presented in the form of a bitmap imposed onto the geometry.

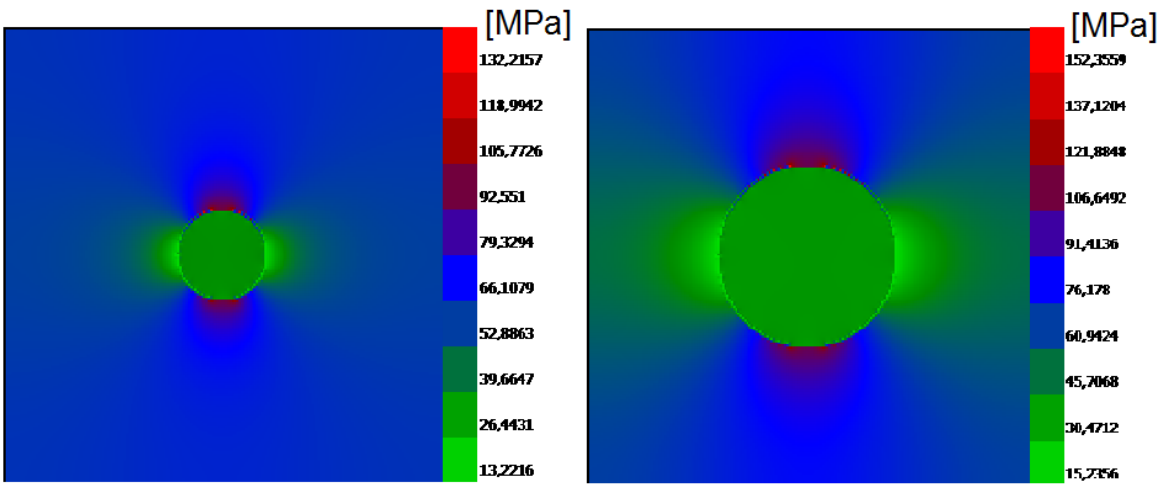

Fig. 4. Maps of σ_y stresses for precipitates with diameters of 40px and 80px

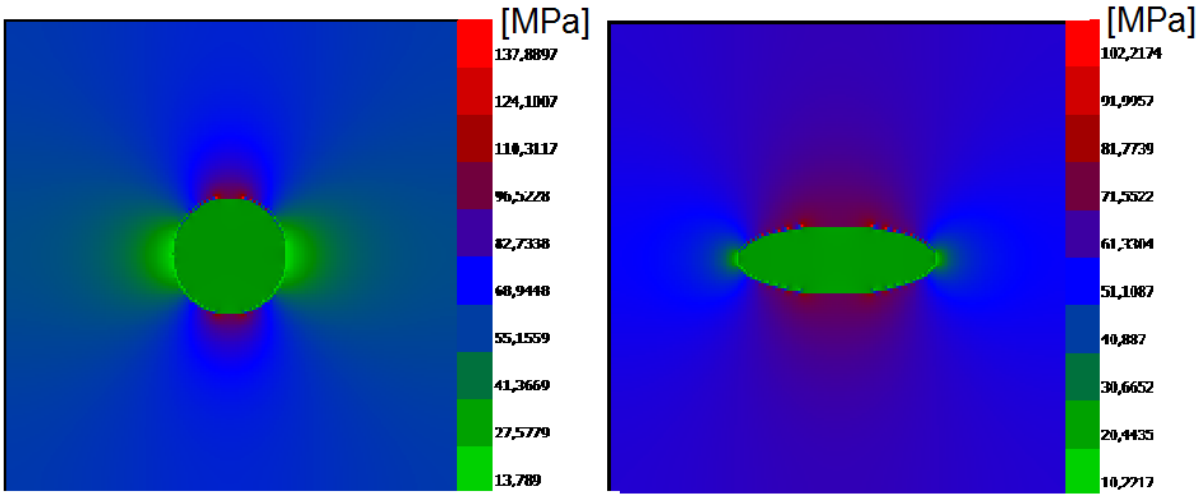

Fig. 5. Maps of σ_y stresses for a circular precipitate with diameter of 50px and an elliptical precipitate with the same section area and aspect ratio of 3

Fig. 6. Profile of directional stresses going through the centre of a precipitate with diameter of 50 px

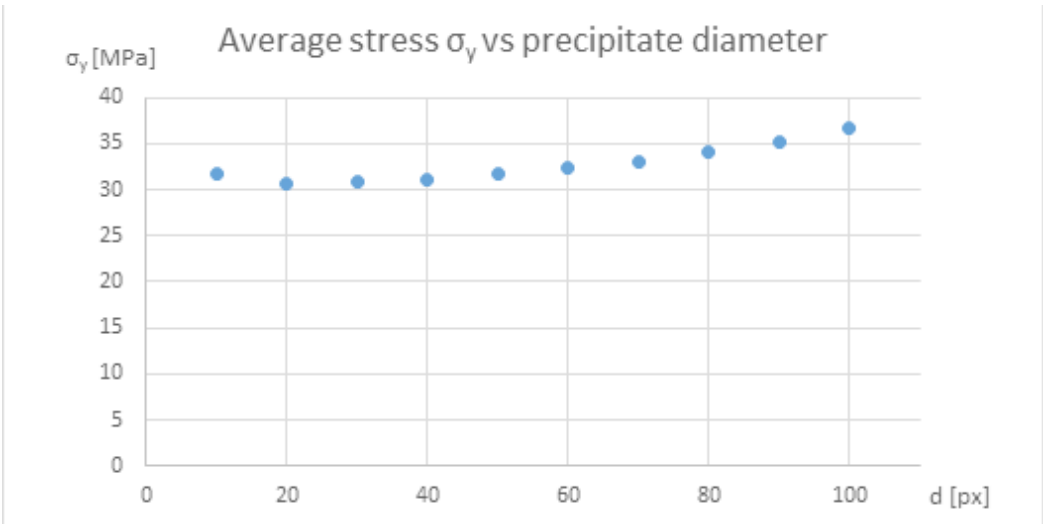

Fig. 7. Directional stress values for different diameters of circular precipitates

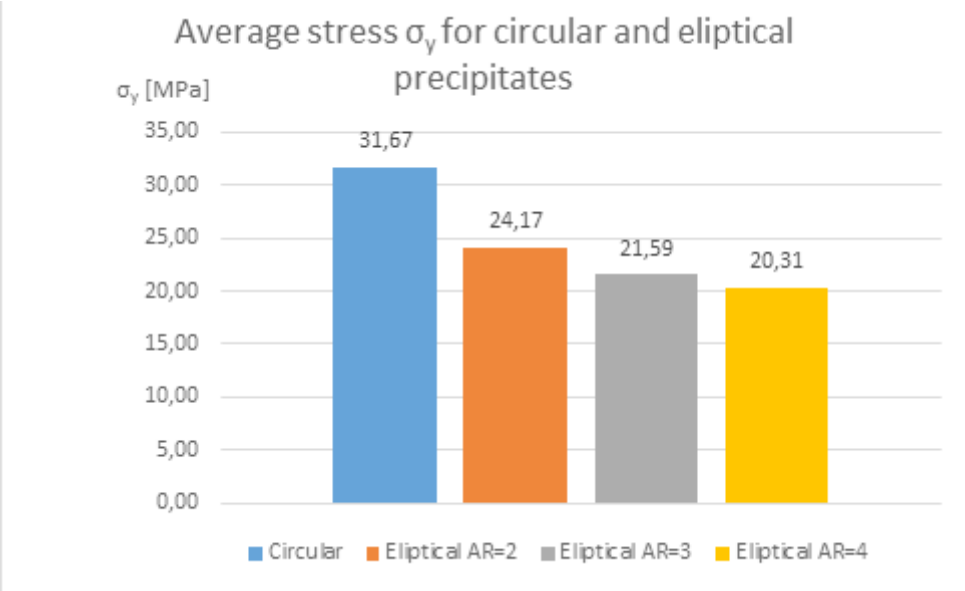

Fig. 8. Stress values for a circular precipitate and elliptical precipitates with differing aspect ratios and identical surface areas

The calculation results for every automaton cell are recorded in text files. By using the coordinates, it is possible to determine the stress profile along the line going through the centre of precipitate. All the coordinates are measured in pixels. From the results obtained, the average values of directional stresses inside the precipitate were calculated. The results were compared among circular precipitates with various diameters in order to determine the relationship between the precipitate size and the stress generated inside of it. Afterwards, the results were compared between a circular precipitate and elliptical precipitates with the same section area and different aspect ratios.

DISCUSSION AND CONCLUSIONS

A computational algorithm was calculated employing the Cellular Automata method, being an iterative method of simulating the stresses in two-phase materials. Simulations of axial tensioning of material containing a single hard hardening phase precipitate on the example of the Al 2024 alloy were performed. In the case of the CA algorithm developed it is possible to introduce our own geometries based on digital representations of microscopic images of actual precipitates. The software is going to be developed within that scope in the future.

On the basis of the calculation results obtained the following conclusions were formulated:

- For identical values of total deformation the values of internal stresses of precipitates depend on their size. The larger the precipitate diameter, the higher the values of the directional stresses.
- The distribution of stresses is influenced to a great extent also by the shape of precipitates. For the precipitates with identical surface area, the stresses were inversely proportional to the ratio of precipitate width to its height.

REFERENCES

1. Kutylowski R.: Optymalizacja topologii kontinuum materialnego. Oficyna Wydawnicza Politechniki Wrocławskiej, 2004.
2. Rozvany G.: Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. Structural and Multidisciplinary Optimization, 21 (2), 2001, 90–108.
3. von Neumann J.: Theory of self-reproducing automata. Urbana: University of Illinois, 1969.
4. Wolfram S.: A New Kind of Science. Champaign: Wolfram Media, 2002.
5. Dewei F.: Numerical computing method Based on Cellular Automata. 3rd International Conference on Computer and Electrical Engineering. 53, 2012, 201–205.
6. Tatting B., Gurdal Z.: Cellular Automata for Design of Two-dimensional Continuum Structures. American Institute of Aeronautics and Astronau-

tics, 8th Symposium on Multidisciplinary Analysis and Optimization, 2000.

7. Gurdal Z., Abdalla M.: Structural Design Using Optimality Based Cellular Automata. American Institute of Aeronautics and Astronautics Paper, 1676, 2002.
8. Abdalla M. M., Gürdal Z.: Structural design using Cellular Automata for eigenvalue problems. Structural and Multidisciplinary Optimization, 25, 2003, 1–9.
9. Kita E., Toyoda T.: Structural design using cellular automata. Structural and Multidisciplinary Optimization, 19 (1), 2000, 64–73.
10. Tatting B., Gurdal Z.: Cellular Automata for Design of Truss Structures with Linear and Nonlinear Response. American Institute of Aeronautics and Astronautics Paper, 2000.
11. Zakhama R., Abdalla M.: Topology design of geometrically nonlinear 2D elastic continua using CA and an equivalent truss model. 11th AIAA/ISSMO Multidiscipinary Analysis and Optimization Conference, 2006, 1–10.
12. Slotta D. J., Tatting B., Watson L. T., Gurdal Z., Missoum S.: Convergence analysis for cellular automata applied to truss design. Engineering Computations, 19 (7/8), 2002, 953–969.