



**AIAA 2000-4832**  
**Cellular Automata**  
**for Design of Two-Dimensional**  
**Continuum Structures**

Tatting Brian  
ADOPTTECH Inc.,  
Blacksburg, Virginia

Zafer Gürdal  
Virginia Polytechnic Institute and State University  
Blacksburg, Virginia

**8th AIAA/USAF/NASA/ISSMO**  
**Symposium on Multidisciplinary Analysis**  
**and Optimization**

September 6-8, 2000  
Long Beach, CA

**CELLULAR AUTOMATA FOR DESIGN OF  
TWO-DIMENSIONAL CONTINUUM STRUCTURES**

Brian Tatting<sup>§</sup>

ADOPTTECH, Inc., Virginia Tech Corporate Research Center, Blacksburg, VA

Zafer Gürdal<sup>‡</sup>

Virginia Polytechnic Institute and State University, Blacksburg, VA

**ABSTRACT**

The implementation of Cellular Automata (CA) techniques for the analysis and design of two-dimensional continuum structures under in-plane loading is presented. The paper summarizes the basic elements of the CA and the specific formulation used for thin isotropic plates. Analysis and design rules are presented that satisfy static equilibrium and fulfill a fully-stressed material failure condition. The update rules are based on the deformation of an equivalent truss structure, and the relation between the thickness of the isotropic plate and the cross-sectional areas of the truss members are derived so that the strain energy due to in-plane deformation is modeled correctly. Example results are presented which verify the accuracy of the modeling technique by a comparison to a closed-form stress functional solution. Additionally, design studies based on topology optimization and thickness sizing are performed, which demonstrate the applicability of the CA environment for efficient design of structures.

**INTRODUCTION**

Present techniques for the optimal design of elastic structures rely on finite element based analysis methods that quickly become prohibitive in terms of computational resources as the models increase in size. The main reason for this is that the complex finite element models that contain a large number of design variables typically require substantial re-analysis of the structure such that rigorous optimization cannot be performed efficiently. An existing alternative to this approach, especially for the case of structures that require computationally expensive nonlinear analysis techniques, is to attack the governing equations in terms of both the field variables and the design variables. This approach is commonly referred to as simultaneous analysis and design<sup>1</sup> (SAND), and only requires the solution of one nonlinear optimization problem without the need for complete structural analysis at every design stage. However, such a nonlinear solution may also be computationally expensive for complex structures.

A non-traditional technique based on biomimetic concepts offers a different approach to this design

problem. In nature, the evolution of complex systems does not follow a “global” design strategy, but rather adapts locally to surrounding conditions in an attempt to improve the functionality of a local process. Modeling complex phenomena through the use of local autonomous units has been termed Cellular Automata (CA), and has been used successfully to study complex natural phenomena. Recently, the CA environment has been applied to the design of elastic structures<sup>2,3</sup>, in which various design update rules were implemented while the analysis of the structure was performed using traditional techniques. The motivation behind these applications was that design can be performed locally according to the level of stress, while the stress state can take the design changes into account simultaneously so that the structure evolves toward a more optimal design. Although such an implementation brings a new level of innovation to the design process, it has little impact on the real computational bottleneck, which is associated with the analysis of the structure itself.

In a recent paper<sup>4</sup>, the authors introduced a novel scheme in which the analysis of the structure is treated as an integral part of the CA scheme, and updating of both the design and state variables is carried out simultaneously. It was shown that this approach was fully capable of accurately modeling the linear and nonlinear response of truss structures. In fact, for some cases, it was demonstrated that the nonlinear analysis required a fewer number of

<sup>§</sup> Senior Researcher, ADOPTTECH, Inc.

<sup>‡</sup> Professor, Departments of Aerospace and Ocean Engineering, and Engineering Science and Mechanics, Associate Fellow AIAA.

Copyright © 2000 by Brian Tatting. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

iterations to converge than the corresponding linear analysis. Computationally, the nonlinear analysis for each time step was only slightly more expensive than the linear case; hence, the total computation time for the nonlinear solution was quite close to the cost of the simpler linear analysis. This situation can rarely be replicated via traditional FEM analyses. The CA truss solution also demonstrated that the simultaneous analysis and design methodology was extremely efficient for combined topology and sizing design. In some cases, it was possible to design a total truss structure in less time than was required to analyze the original configuration. Again, this is an aspect of CA design that is difficult to imitate in any other computational design approach.

Additionally, the CA environment lends itself to extremely efficient algorithms due to its inherent locality and its proven potential within parallel computing environments. Therefore, this paper introduces the CA technique and describes how it can be applied to the design of elastic continuum structures, and provides analysis and design results for typical two-dimensional structures.

### **ELEMENTS OF CELLULAR COMPUTING**

Cellular automata are generally attributed to Ulam<sup>5</sup> and von Neumann<sup>6</sup>, who introduced the concept in the late forties to provide a realistic model for the behavior of complex systems. The literature on the subject is not consistently catalogued in the sense that cellular automata type methods seem to have been implemented for different types of problems under different names. Initially CA techniques were introduced under the name automata networks, which were used to model discrete dynamical systems in time and space. They are used as a finite or infinite graph where each vertex can take on discrete values from a finite set. The state of each vertex can be altered through transition rules that are based on the vertex's current state as well as that of its neighbors in the graph. The entire graph is then updated repetitively either in a synchronous or sequential manner. In the synchronous mode, which is also called parallel mode, all the sites are updated in a discrete time step simultaneously. The sequential update is applicable to only finite networks, and the sites are updated one by one in a prescribed order.

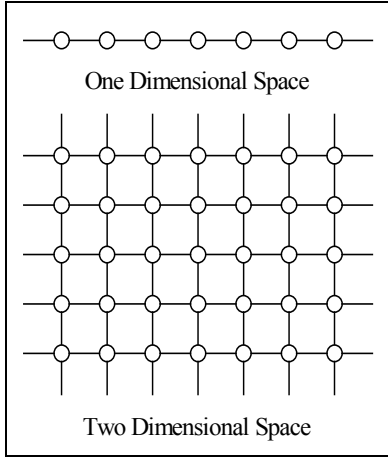
A particular case of automata networks is the cellular automata, in which the graph is a regular lattice and the updating mode is synchronous. Moreover, the update (transition) rules and the neighborhood structure are the same for all sites. A variant of the cellular automata uses continuous lattice site values, sometimes referred to as a coupled map-lattice or cell-dynamic scheme.

One of the fundamental features of cellular automata, which make them highly useful computational tools for large systems, is their inherent parallelism. By assigning a simple processor to every so many cells of a large system of cells, one can increase the detail or the size of the system without increasing the time it takes to update the entire system. There does not seem to be a theoretical limitation or an overhead associated with splitting the problem into small pieces and distributing it. Thus, cellular automata simulations are highly suited for massively parallel computers that possess the proper hardware requirements.

In their modern implementation, cellular automata are viewed as simple mathematical idealizations of natural systems, and are used successfully to represent a variety of phenomena such as diffusion of gaseous systems, solidification and crystal growth in solids, and hydrodynamic flow and turbulence<sup>7</sup>. In most of the previous applications, they are used to represent macroscopic behavior of a system, which are governed by partial differential equations of the continuum under consideration. This is generally accomplished using simple rules that represent the micro-mechanics of the medium. By utilizing a sufficiently large number of cells, it is possible to represent a complex continuum response that is governed by highly nonlinear equations.

This paper proposes an implementation of the methods of cellular automata for the design of structural systems by attempting an integrated solution of the state and design variables. Before detailing the specific ideas behind the proposed implementation, we first provide a brief description of the elements of cellular automata and how they relate to the study of two-dimensional continua.

**CA Lattice:** The form of the cellular space directly reflects the physical dimensions of the problem under consideration. Two sample lattice structures, representing one- and two-dimensional cellular spaces, are shown in Figure 1. A three-dimensional space can be constructed by layering several of the two-dimensional ones, spaced equally so that the distance between them is the same as the distance between the cells in the plane. The lattice structures, however, are not limited to the rectangular ones shown in the figure. Cellular automata based on other lattice systems, such as two-dimensional triangular and hexagonal lattices, are also possible. Wolfram<sup>8</sup>, for example, used a regular two-dimensional lattice of hexagonal cells for a cellular automaton fluid model. The best shape of the lattice is often dictated by the geometry of the problem under study. Since our emphasis here is on two-dimensional plates, a



**Figure 1:** Rectangular Cellular Spaces

rectangular lattice is used so that the edge conditions of the plate can be easily modeled.

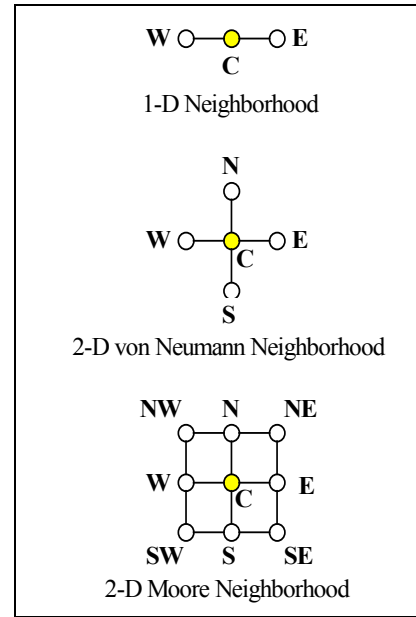
**Cell Definition:** Each lattice site, or cell, has a value or set of values that are updated over the course of time. The cell parameters may be discrete or continuous values allowed to change in a range, or they may be binary (0/1) variables. These values are altered using update rules (explained later), and by monitoring the ever-changing state of the cell lattice a wide variety of natural systems can be simulated. Traditional CA's used binary on/off values for each cell, which modeled such diverse phenomena as traffic flow, snowflake generation, diffusion, and predator-prey ecosystems<sup>7</sup>. For our purposes, the cell definition will include a set of continuous variables that represent the physical system of a two-dimensional continuum. Thus displacements, material properties, and applied forces are all defined for each cell. This can be represented by a list of state variables for each cell,

$$\phi_C = \left\{ \{u, v\}, \{E, \nu, t\}, \{f_x, f_y\} \right\}_C, \quad (1)$$

where the subscript  $C$  is used to indicate a particular cell in the domain. For our formulation, the first two variables represent the in-plane displacements of a cell that are altered according to the analysis update rules (described later). The second set of parameters denotes the material properties of a thin isotropic continuum. The material constants are generally unchanged, while the thickness parameter is the main variable for the design process. Finally, the two forces are optional parameters that only exist for cells that contain an applied force or are restrained from movement. These parameters are used to introduce the loading.

**The Neighborhood:** The neighborhood structure is one of the most important characteristics of a CA lattice. In updating the state of a cell, it is necessary

to consider the local cell parameters as well as the values of the cells in its defined neighborhood. The set of cells that is utilized for the update is highly problem dependent, and relies heavily on the nature of the physical phenomenon that is being modeled. Some common examples of neighborhood structures used in the literature are shown in Figure 2. The cell to be updated is labeled as  $C$ , and the adjacent ones are labeled according to compass directions. Again,



**Figure 2:** Cellular Neighborhoods

these are not the only neighborhood structures. For example, a neighborhood commonly referred to as the “MvonN Neighborhood” combines the nine sites of the Moore neighborhood shown in the figure with four more cells lying two sites away in the north, south, east and west directions. For the present problem involving a two-dimensional continuum, Moore’s neighborhood has been chosen to represent the local sphere of influence of each cell.

**Boundaries:** Since every cell has the same neighborhood structure, cells on the boundary of a physical domain have neighboring cells that are outside the domain. Traditionally, border cells are assumed to have the cells on the opposite boundary as neighbors. For example, for a two-dimensional rectangular domain, a cell on the left border has the cell in the same row on the right border as its left (west) neighbor. With the same update rule applied to all the cells, this produces a periodic boundary condition, which is often representative of an infinite system. Within the classical representation of moving particles, for example, a particle leaving the domain from one side enters the domain from the opposite side in the same row or column. For simulation of

solid mechanics problems, free and constrained boundary representations are needed. For our solution, the boundary conditions are satisfied by defining an extra row or column of cells outside of the domain and stipulating the thickness of these cells to be zero. This effectively creates a free edge on the boundary, and fixed conditions are applied by enforcing the desired displacement for the cells on the boundary edge.

**CA Update Rules:** Computer implementation of update rules is similar to function subroutines that are applied to every cell of the lattice. The arguments for the function subroutine are the relevant parameters of the cells within the neighborhood, and the value returned by the function is the new parameter of the cell at which the function is being applied. For example, for the von Neumann neighborhood the function would require five arguments, **function(C, E, W, N, S)**, which returns the value of the site *C* at the time/iteration (*T*+1) as a function of specified values of the cells within the neighborhood. As alluded to earlier, the update scheme can be carried out for each cell synchronously (which is termed the Jacobi method), or the function can use the most recent values for each cell. This second method corresponds to the Gauss-Seidel iteration scheme. Comparison of the two iterative schemes for the CA has indicated that the Gauss-Seidel method was more efficient for the structures problems under consideration, and thus this update scheme was used for all of the results in this paper.

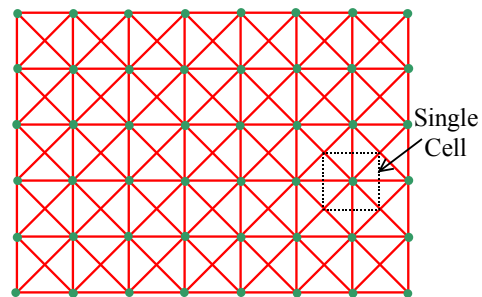
Two update functions were used for this project. The first rule is for the movement of each cell and acts on the first two parameters (the displacements) of equation (1). This rule satisfies equilibrium of the cell for given deformation of the surrounding cells. The second update rule is based on design concerns and operates on the second set of parameters (material properties and geometries). Details of both of the update rules are provided in the next section.

### CA REPRESENTATION OF TWO-DIMENSIONAL CONTINUUM STRUCTURES

In an earlier paper<sup>4</sup>, the authors presented a formulation of the cellular analysis and design of two-dimensional structures represented by pinned trusses. This paper expands the analysis to two-dimensional elastic continuum by formulating an equivalent truss-cell that mimics the properties of an elastic medium. Therefore, the underlying formulation of the two-dimensional truss structure, along with the modifications and update rules for the equivalent continuum structure, are summarized in the following.

### Two-dimensional Cellular Automata

A structural domain that is covered by a regular rectangular lattice of cells is shown in Figure 3. The density of cells within the domain is chosen by the user. The center point of each cell is connected to the neighboring cells via elastic members that extend halfway between the cell centers. In the simplest case, these elastic members are truss elements that undergo only axial deformation, though more complex formulations can be produced by using beam or frame elements. Additional details of the truss CA can be found in an earlier work published by these authors<sup>4</sup>.

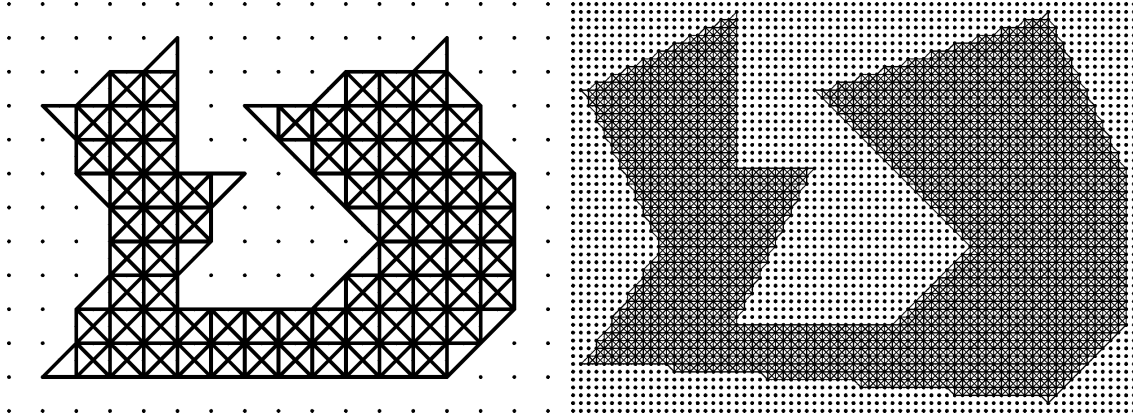


**Figure 3:** Rectangular Cell Lattice

The domain shown in Figure 3 does not limit the designer to rectangular structures. Due to the formulation of the cell definitions, it is possible to describe irregularly shaped structural domains by the selective assignment of cell properties outside the boundaries of a structural domain. For instance, for a truss structure a cell can be “turned off” by assigning a zero value to the cross-sectional area parameter of the truss members. This effectively creates a free edge for the structure. This feature is used within the software to define general shapes based on the domain of a simple closed polygon. An example is shown in Figure 4 for two models with varying cell density. As expected, for domains with diagonal boundaries increased cell density is required to fully capture the geometry of the structure.

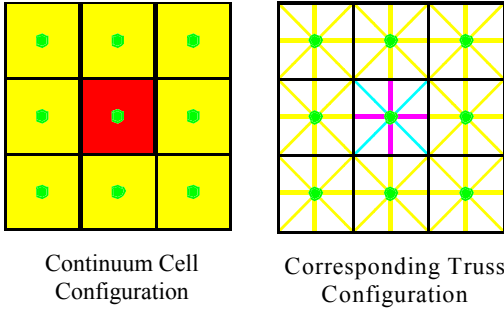
### Continuum Modeling

To represent a two-dimensional continuum, we slightly re-formulate some of the basic elements of the CA program that was based on the discrete truss structure. For instance, the geometry of a two-dimensional continuum structure is more aptly represented by a continuous thickness value for each cell. Therefore, we determine the relations between the apparent thickness of the continuum cell and the cross-sectional areas of the truss members that are used by the CA. The relevant geometries are illustrated in Figure 5, which displays the configuration of a continuum cell with its eight



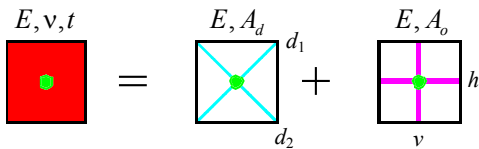
**Figure 4:** Cellular Representation of Polygon

neighboring cells along with the corresponding truss configuration.



**Figure 5:** Continuum Cell and Equivalent Truss Cell

As the different colors used for the diagonal and the cross members indicate, instead of eight independent cross-sectional areas used for the truss structure, there are only two areas in the continuum representation, which are the areas of the diagonal and orthogonal members. Furthermore, the two areas are not independent but rather related to a single quantity, which is the thickness of the continuum cell. In this sense, the continuum cell with a thickness  $t$  and material properties  $E$  and  $\nu$  is represented by a cell with diagonal members that possess areas  $A_d$  and vertical and horizontal members with areas  $A_o$ , all with a material property of  $E$ , as shown in Figure 6. The relation between the truss member areas,  $A_d$  and  $A_o$ , and the thickness  $t$  is computed by using the equivalence of the strain energy between the truss cell and the continuum cell.



**Figure 6:** Continuum/Truss Cell Relation

The strain energy  $U$  for a thin continuum in an  $l \times l$  square domain can be represented in terms of strain as

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

The objective for the analogous truss structure is to generate the same amount of strain energy for a two-dimensional continuum under in-plane loading. The equivalent truss structure is represented by the four continuous members labeled  $\{v, h, d_1, \text{ and } d_2\}$  in Figure 6. The strain energy in these members is computed according to the axial strain experienced by each member, thus

$$U = \left[ A_o \epsilon_h^2 + A_o \epsilon_v^2 + \sqrt{2} A_d (\epsilon_{d_1}^2 + \epsilon_{d_2}^2) \right] \frac{El}{2} \quad (3)$$

where  $\epsilon_i = \Delta_i / L_i$  for each member. Our goal is to choose the correct areas for the truss members so that the strain energies of equations (2) and (3) are identical under the same displacement loading induced by the neighboring cells.

The derivation thus requires a relationship between the in-plane strains of the continuum and the extensional strains of each truss member. This is accomplished by calculating the extensional strain that would exist along the axis of each truss member for a given state of deformation  $(\epsilon_x, \epsilon_y, \gamma_{xy})$ . Basic mechanics generates the relations

$$\begin{aligned} \epsilon_h = \epsilon_x & & \epsilon_{d_1} = \frac{\epsilon_x + \epsilon_y}{2} + \frac{\gamma_{xy}}{2} \\ \epsilon_v = \epsilon_y & & \epsilon_{d_2} = \frac{\epsilon_x + \epsilon_y}{2} - \frac{\gamma_{xy}}{2} \end{aligned} \quad (4)$$

Note that the horizontal and vertical strains are only influenced by the  $x$  and  $y$ -direction strains, while the diagonals are influenced by all the deformations. Insertion of these equations into (3), equating the resulting expression to the continuum strain energy and matching like coefficients yields three

independent equations in terms of the two unknown cross-sectional area quantities:

$$A_o + \frac{A_d}{\sqrt{2}} = \frac{lt}{1-\nu^2} \quad \frac{A_d}{\sqrt{2}} = \frac{\nu lt}{1-\nu^2} \quad \frac{A_d}{\sqrt{2}} = \frac{G}{E} lt \quad (5)$$

This is an over-constrained system and does not have a unique general solution, since the last two equations can only be satisfied simultaneously for a specific value of the Poissons ratio ( $\nu = 1/3$ ). Therefore, for a general continuum, the cross-sectional areas must be chosen such that their values minimize the error associated with the stiffness properties. There are several combinations of choices available, and it was found that the most accurate solution for typical isotropic materials was to assume a material with  $\nu = 1/3$  and choose areas that satisfy the isotropic relation and conform to the modulus of elasticity of the continuum. This results in the following relations for the cross-sectional areas:

$$A_o = \frac{3}{4} lt \quad A_d = \frac{3\sqrt{2}}{8} lt \quad (6)$$

These equations are used to calculate the cross-sectional areas for the truss members given the cell spacing and the continuum thickness  $t$ . Verification of the accuracy of this method is provided in the numerical results.

### **Analysis Rules**

In order to derive the CA update rule for the cell displacement, we consider the equilibrium of a cell at iteration ( $T+1$ ) based on the state values of the neighbors already known at time ( $T$ ). Considering the forces acting on the cell node, two equilibrium equations, one in the horizontal and another in the vertical direction, can be written.

$$\sum_{k=1}^8 F_{xk} + f_x = 0 \quad \sum_{k=1}^8 F_{yk} + f_y = 0 \quad (7)$$

Here  $F_{xk}$  and  $F_{yk}$  are the horizontal and vertical components of the internal forces in each of the eight truss members that are attached to the cell under consideration, and  $f_x$  and  $f_y$  are the applied or reaction forces that exist for constrained cells. The force components  $F_{xk}$  and  $F_{yk}$  can be calculated from the internal force in the member by using the relation for the orientation of the  $k^{\text{th}}$  member at the prescribed angle  $\theta_k$ .

$$F_{xk} = F_k \cos \theta \quad F_{yk} = F_k \sin \theta \quad (8)$$

The internal forces in each member are expressed in terms of the axial extension in the direction of the truss member and the combination of areas for the center and neighboring cell. Since each truss element is actually composed of two half-length members with different cross-sectional areas, the force-

displacement relation is represented as

$$F_k = \frac{2(E_c A_c)(E_k A_k)}{E_c A_c + E_k A_k} \left( \frac{\Delta_k - \Delta_c}{L_k} \right), \quad (9)$$

where  $\Delta$  represents the displacement in the axial direction of the  $k^{\text{th}}$  member (and is related to the cell displacements  $u$  and  $v$ ),  $L_k$  is the length of the truss member, and the subscript  $c$  denotes the values for the cell under consideration. The cross-sectional areas in this equation are calculated from the thickness in each cell according to equation (6).

Substitution of equation (9) into the force equilibrium equations (7) yields a system of two linear equations in terms of the cell displacements  $u$  and  $v$ . This can be solved in closed-form for a general cell, though the equation is lengthy. Several degenerate solutions exist which are required for boundary cells and for cells that have a combination of applied forces and restrained displacements.

Repeated application of the analysis rule for each cell, using values at time iteration ( $T$ ) for the displacements of the neighboring cells, converges to the static equilibrium solution for a two-dimensional continuum. For a given displacement field, average strains and stresses can be extracted from the displacement solution to determine the current stress state on a point-by-point basis. This information can be used to estimate the possibility of failure within the material and can therefore be used in the design of structures.

### **Design Rules**

The displacement update formulae presented above includes the cross-sectional areas of the members connecting each cell to its neighbors, which are based on the thickness of the continuum at the cell locations. The update formula for the thickness parameter used in this initial phase is based on a fully stressed design approach. That is, it is assumed that a cell requires no more material than the minimum needed to carry the internal stresses within the cell. Given the displacement of a cell and its neighbors, the average stresses in the members are easily computed following the equations given earlier. From the individual member stresses, an effective stress measure  $\sigma_e$  can be calculated for each cell. For example, for this investigation the von Mises stress measure was calculated for each cell. Using this calculated value as a comparison for the von Mises failure criterion, stress rationing produces the following rule for the thickness update:

$$t_C^{(T+1)} = t_C^{(T)} \frac{\sigma_e^{(T)}}{\sigma_{allow}} \quad (10)$$

In the early stages of design, this rule must be augmented by move limits so that regions that are initially unaffected by the loading will not develop zero thickness values. Repeated updates using this design rule concurrently with the analysis update produces a solution which evolves toward a statically determinate structure that makes best use of its material to carry the applied loads. This technique can also be used for topology optimization, since the regions that are not needed to carry loads become thinner until they disappear.

### NUMERICAL EXAMPLES

The formulation described in the preceding section was translated into a computer program using the Fortran90 programming language. Example results and comparisons for both analysis and design are provided to illustrate the possibilities of the CA paradigm for structural optimization and design.

#### Analysis Verification

For verification of the CA analysis, a two-dimensional continuum problem with a closed-form solution is used for a comparison. The chosen problem is a rectangular domain supported at the sides and loaded by a sinusoidal stress applied at the top surface. The geometry and loading specifications are shown in Figure 7. The specific loading and boundary conditions for this configuration lend themselves to a straightforward solution using a stress potential function. Closed-form expressions for the stresses, strains, and displacements can be deduced analytically. The equations for the closed-form solution are included in the appendix.

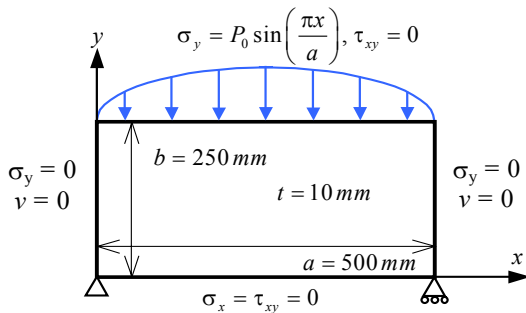


Figure 7: Example Analysis Problem

The corresponding solution within the Cellular Automata environment employs 50 cells in the vertical direction and 100 cells in the x-direction. A comparison of the von Mises stress distribution is shown in Figure 8 for an isotropic material with a Poisson's ratio of one-third. The underlying image and colored contours represent the CA solution, while the black overlays conform to the exact solution. The results show almost perfect correlation,

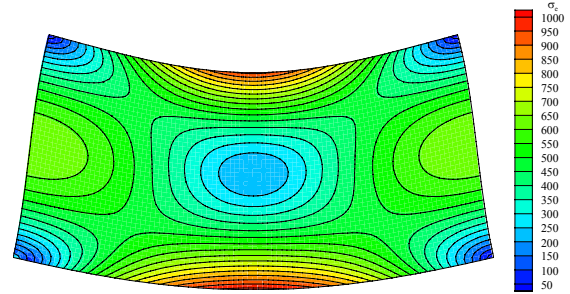


Figure 8: Analytical Comparison

except for some slight errors near the boundaries due to the averaging technique used to calculate the cell stresses. This outcome is slightly surprising since, in essence, we are replacing a continuum with discrete members that only experience axial deformation. However, the results show that the simple equivalent truss structure, when formulated properly, can accurately model a more complex structure. The advantage in using the equivalent truss structure is that it is so simple to implement within the numerical software.

#### Design Example 1: U-shaped Structure

As an illustration of the design/topology possibilities using the truss and continuum models within the CA paradigm, an optimization problem with an atypical ground structure was investigated. The problem consists of a U-shaped domain that is supported on the left side, and the top and bottom corners, and loaded by an inclined force in the upper right corner (see Figure 9). The design process is used to determine the optimal cross-sectional areas to withstand the given loading while achieving minimum weight for different cell densities.

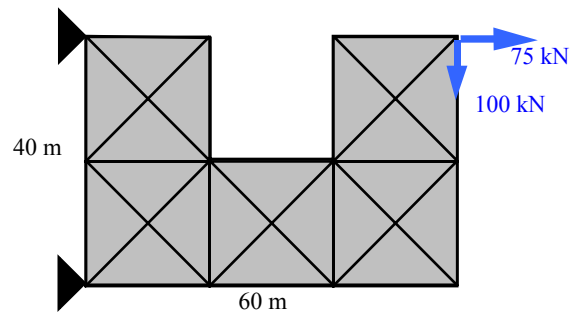
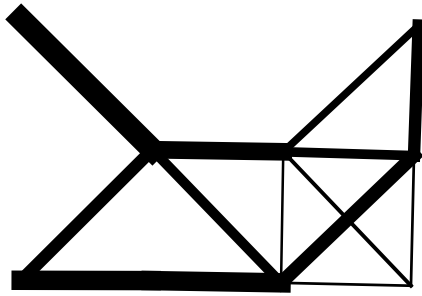


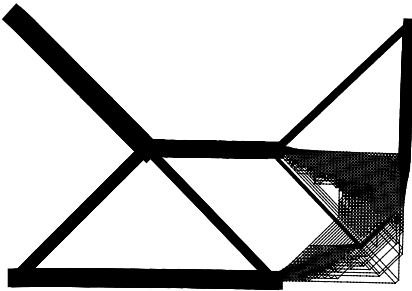
Figure 9: U-shaped Structure

The results for a simple truss solution (a 4x3 cell lattice) are shown in Figure 10, where the line thicknesses represent the relative cross-sectional areas of the truss members. Increasing the density of the truss cells produces a similar structure except in the lower right-hand corner, where the truss members do not converge toward a recognizable configuration.



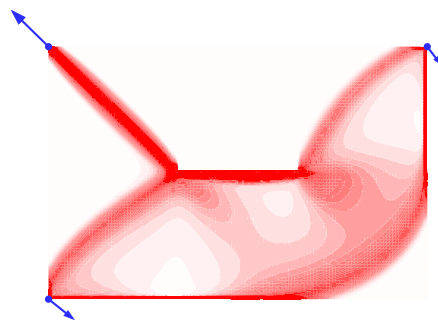
**Figure 10:** U-shaped Truss Solution

For example, the solution using a  $161 \times 121$  cell lattice is shown in Figure 11. Notice how the lower right corner seems to be attempting to transform itself into a curved boundary, but is unable to do so using the discrete truss members as construction elements.



**Figure 11:** Dense U-shaped Truss Solution

Conversely, a corresponding solution using the continuum CA model tends to generate smoother solutions since the eight equivalent trusses are all tied together within each cell through the thickness parameter. The results for the continuum model are shown in Figure 12, where the darkness of the contours indicates the local thickness values of the continuum cells. The continuum design reveals the best topology for the structure under the given loads. Note how it resembles a straight member truss in several regions, though the lower right corner approaches a curved beam, as expected. The simplicity of investigating these alternate structural designs indicates the flexibility of the CA algorithm.

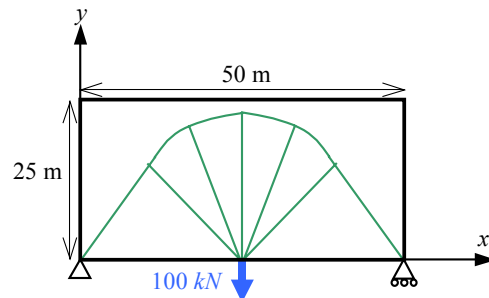


**Figure 12:** U-shaped Continuum Solution

### Design Example 2: Mitchell-type Structure

As a final example, a direct comparison to existing design tools is included. This is meant to illustrate the simplicity of the CA and the ease with which the results were demonstrated. GENESIS<sup>9</sup>, a commercially available finite element design tool, was chosen for the comparison due to its topology and shape/sizing capabilities. However, the comparative results must be tempered by the fact that the numerical studies were performed on a personal computer, which tends to produce slower process times than expected from modern workstations. However, it is significant that the CA can run quite efficiently on such architectures, as the results will indicate.

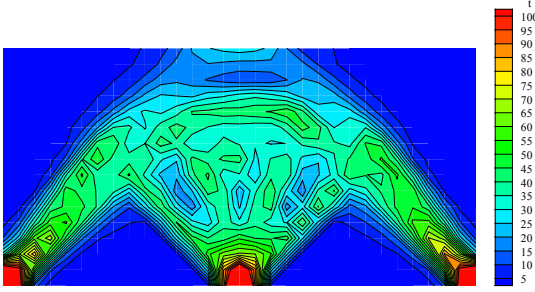
The problem under study is a rectangular ground structure supported at the lower corners, with a downward force applied at the center of the bottom edge (see Figure 13). This problem roughly corresponds to the Mitchell truss, a classical topology optimization problem. The “ideal” solution is shown superimposed in green on the ground structure.



**Figure 13:** Mitchell-type Truss Problem

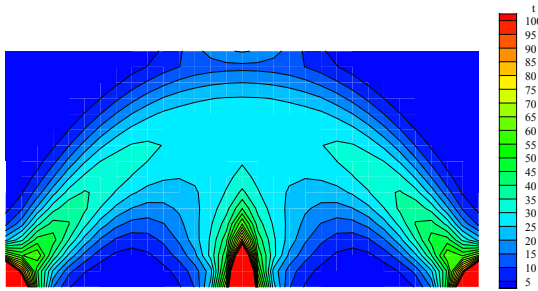
The ground structure geometry was translated into a  $30 \times 15$  finite element model and input into GENESIS. The goal was to find the optimal thickness variation for the given loading conditions while attaining minimum weight. The solution was performed using the shape/sizing capabilities of GENESIS, which do not allow thicknesses to go to zero. That is, the topology aspect cannot be investigated using this tool, and upper and lower limits for the thicknesses needed to be defined. The solution using GENESIS is shown in Figure 14, where the colored contours represent the thickness variation of the material. The model used a grid with  $31 \times 16$  nodes resulting in 992 degrees of freedom. The solution required 5408 CPU seconds for convergence and converged to a final volume of  $28,180 \text{ mm}^3$ .

The same problem was solved using the CA software, with the same number of degrees of freedom ( $31 \times 16$  cells within the given domain) as well as upper and lower limits applied to the thicknesses to mimic the



**Figure 14:** Mitchell Structure, GENESIS Results

exact conditions of the finite element result. Using the same PC, the solution required roughly 10,000 iterations, which ran in 51 CPU seconds. The result is shown in Figure 15. The final volume is 25,641 mm<sup>3</sup>. Note that the CA solution does not fully capture the webbed nature of the solution that is apparent in both the ideal and finite element solutions. However, models with increased cell density do exhibit this webbed nature to a greater degree.



**Figure 15:** Mitchell Structure, CA Results

## CONCLUSIONS

This investigation has demonstrated the formulation of a computer algorithm that implements Cellular Automata computing techniques for the analysis and design of a two-dimensional continuum under in-plane loads. The numerical solution relies on an equivalent truss structure to model the behavior of a thin continuum structure. This technique was found to work quite effectively for the design of planar structures in terms of both topology optimization and re-sizing for a fully stressed design condition. Continued investigation of cellular computing methods for the design of structures is certain to provide a new tool for the development of optimal structures.

## ACKNOWLEDGEMENT

The authors wish to acknowledge the support and guidance of Jarek Sobieski of NASA-Langley Research Center. We would also like to thank Professor Layne Watson of Virginia Tech for his

insightful discussions and guidance with the numerical implementation of the software.

## REFERENCES

1. Haftka, R. T., Kamat, M. P., "Simultaneous Nonlinear Structural Analysis and Design," *Computational Mechanics*, Vol. 4, 1989, pp. 409-416.
2. Hajela, P., Kim, B., "On the Use of Energy Minimization for CA Based Analysis in Elasticity," *Proceedings of the 41<sup>st</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Atlanta, GA, April 3-6, 2000.
3. Kita, E., Toyoda, T., "Structural Design using Cellular Automata," *Structural and Multidisciplinary Optimization*, Vol. 19, 2000, pp. 64-73.
4. Gürdal, Z., Tatting, B., "Cellular Automata for Design of Truss Structures with Linear and Nonlinear Response," *Proceedings of the 41<sup>st</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Atlanta, GA, April 3-6, 2000.
5. Ulam, S., "Random Processes and Transformations," *Proceedings of the International Congress of Mathematics*, Vol. 2, 1952, pp. 85-87.
6. Von Neumann, J., *Theory of Self-Reproducing Automata*, University of Illinois Press, 1966.
7. Gaylord, R. J., Nishidate, K., *Modeling Nature: Cellular Automata Simulations with Mathematica*, Springer-Verlag, Inc., 1996.
8. Wolfram, S., *Cellular Automata and Complexity: Collected Papers*, Addison-Wesley Publishing Company, 1994.
9. *GENESIS User Manual*, VMA Engineering, 1998.

## APPENDIX

The closed-form solution for the problem shown in Figure 7 is given here. The stress functional approach was used to solve for the in-plane stresses:

$$\sigma_x = \frac{P_0 \sin \lambda x}{\sinh^2 \lambda b - (\lambda b)^2} \left[ \lambda b \sinh \lambda b (2 \cosh \lambda y + \lambda y \sinh \lambda y) - (\sinh \lambda b + \lambda b \cosh \lambda b) (\sinh \lambda y + \lambda y \cosh \lambda y) \right]$$

$$\sigma_y = \frac{-P_0 \sin \lambda x}{\sinh^2 \lambda b - (\lambda b)^2} \left[ (\lambda b \sinh \lambda b) (\lambda y \sinh \lambda y) - (\sinh \lambda b + \lambda b \cosh \lambda b) (\sinh \lambda y - \lambda y \cosh \lambda y) \right]$$

$$\tau_{xy} = \frac{-P_0 \cos \lambda x}{\sinh^2 \lambda b - (\lambda b)^2} \left[ (\lambda b \sinh \lambda b) (\sinh \lambda y + \lambda y \cosh \lambda y) - (\sinh \lambda b + \lambda b \cosh \lambda b) (\lambda y \sinh \lambda y) \right]$$

where  $\lambda = \pi/a$ .