Cellular automata modeling of processes on landscape surfaces using triangulation meshes

M A Domasevich¹, A V Pavlova¹, S E Rubtsov¹ and I S Telyatnikov²

¹Department of Mathematical modeling, Kuban State University, 149, Stavropol'skaya str., Krasnodar, 350040, Russia
²Laboratory of Mathematics and Mechanics, Southern Scientific Centre of the Russian Academy of Sciences, 41, Chehova str., Rostov-on-Don, 344006, Russia

E-mail: pavlova@math.kubsu.ru

Abstract. The authors presented an approach to the construction of triangulation grids that approximate real fragments of the terrain. Algorithms for grid generation by height maps have been developed as well as software tools that implement these algorithms. The described approach allows one to construct cellular automata of various processes and test their operation on arbitrary spatial surfaces of a landscape type. As examples authors considered cellular automata (CA) simulating naive diffusion, propagation of fire and spreading of liquid on triangulation meshes. The operation results of CA models given as examples qualitatively correspond to the physical concepts about the course of the processes under consideration.

1. Introduction

Analysis of studies on the modeling of various environmental processes, including those related to the problem of forecasting and eliminating the consequences of those such as fires, floods, landslides, etc., demonstrates the growing interest in the use of discrete cellular automata [1–3, etc.] approaches, as well as hybrid discrete-continuous agent-based models. This interest is not accidental, since discrete mathematical models do not require laborious calculations; they allow for simulation of the situation in real time, are easily parallelized and allow displaying the results in graphical form, which is also their undoubted advantage. In addition, in practice, discrete models, which are less resource-intensive than grid models often, demonstrate more adequate results than simplified analytical ones. Numerous publications are devoted to the development of cellular automata models (CA models) in terms of the possibility for various process simulations. Paper [4] contains a fairly detailed review of Russian authors’ research results.

With all the variety of CA modifications, cellular automata on rectangular grids are the most often used ones, although hexagonal [5] and other polygonal grids are also encountered.

The use of the triangulation meshes as a discrete space for the construction of a cellular automata model makes it possible to take into account the geometric properties of elements when specifying the rules for the CA transition. The application of cellular automata on triangulation grids promises a number of advantages in modeling the phenomena with consideration of terrain. Such phenomena include the fire spread on rough terrain, migration and sedimentation of impurities, etc. The paper presents spatial simulation models of the fire front propagation and a liquid spill on landscape surfaces, discretized in the form of triangulation grids. The examples of modeling demonstrate the qualitative conformity of the results obtained to the existing understanding of the processes.
2. Generation of meshes for landscape surfaces

Triangulation is the process of dividing a finite set of points in space into the simplest subsets, consisting of three elements, possibly partially overlapping. A well-generated grid does not contain degenerate triangles, element overlaps, and holes. At the same time, the size of the elements should provide the required accuracy for the model. Algorithms for the automated generation of triangulation meshes, the most famous of which is, perhaps, the triangulation algorithm based on Delaunay criterion [6–8], are very diverse. The values of the minimum angles of the partition elements are often explicitly involved in the estimates of the approximation quality [7], therefore, we can assume that triangulation using the Delaunay criterion for a certain set of points is in a certain sense optimal.

The coordinates of the surface reference points can be used as the source data. For example, using the constructed digital model of the region's relief [9], including the mountainous and coastal zones of the Krasnodar, one can immediately obtain an array of heights for the selected area (Figure 1).

![Figure 1. Elevation marks for the selected area in the digital model of the region](image)

A triangulation mesh that models a landscape can also be created on the basis of an analytical description of surfaces. Moreover, a graphic image of the landscape in the .PNG format can be used as the initial data for the triangulation grid. Each pixel of such an image is represented by a grey scale, indicating the relative height of the relief at this point (the lighter, the greater the height, and vice versa – the darker, the lower the height), corresponding to a certain point on a real map in a plane coordinate system. And the brightness of the colour gradation of this point contains information about the height of the landscape.

The simplest triangulation algorithm for height maps, which allows you to preserve the specifications of the map, is implemented as follows. The set of all points in the image is divided into subsets of four pixels, forming a square, which in turn is divided into two triangles. The addition of the edge is usually done in the same way for all squares. When dealing with a .PNG image of a landscape, for each pixel, the brightness coefficient is multiplied by its height modifier. To simplify the work...
with location data of grid elements, we use an algorithm for determining and assigning the indices of neighbours for each triangle.

For the development of the graphical interface, we chose the system for creating client Windows applications – WPF, supplied as part of the .NET Framework. MonoGame was used as the main graphical visualization tool. To visualize the triangulation grids, an array of vertices was used, the size of which is a multiple of three. The elements of the array are the radius vectors of the triangles vertices, and the data on the vertices of one triangle correspond to the consecutive elements of the array. Such method of storing information about the triangulation grid nodes was chosen to improve performance, since it makes processing data in blocks possible.

Figure 2. PNG heightmap for grid generation and the resulting grid

We have developed algorithms to change the grid view scale and angle of view.

3. Simulation of spatial processes on a triangulation grid

3.1. CA model of naive diffusion

For the CA description of diffusion, one of two models is usually used: naive diffusion and TM diffusion (with a neighbourhood of Margolus) [10]. The implementation of the latter on a triangulation grid is very problematic, while the naive diffusion algorithm can be transferred from a flat rectangular mesh to a spatial triangulation mesh quite simply, while maintaining the transition rule.

In this paper, we provide an example of the implementation of an asynchronous CA on a generated triangulation grid. We extended the transition rule by balancing the probabilities, i.e. making the probability of choosing an adjacent cell with a short common edge higher than with a long one. In this case, the probabilities of choosing a neighbour are given by the expression:

$$\frac{1}{\text{length}_j} \cdot \frac{1}{\sum_{j=1}^{3} 1/\text{length}_j}$$

where length\(_j\) is the length of the \(j\)-th edge in the triangle. The evolution function implements asynchronous iteration for all elementary automata. During the evolutionary iteration, each cell exchanges its state with a random neighbouring cell, thus simulating the process of naive diffusion. Figure 3 illustrates the CA, on the left there is the initial state of the automata, on the right there is the result after 300 iterations.
3.2. CA model of fire

Paper [11] describes a model of the fire front propagation based on a cellular automata on a triangulation grid. To simulate the dynamics of ignition, a composite cellular automata is used, which is based on naive diffusion, implemented on a triangulation grid. The model takes into account the direction and speed of the wind, specified by a vector in three-dimensional space, the presence of obstacles (non-combustible areas) and the combustibility coefficient of an inhomogeneous underlying surface. The ignition probabilities are determined in accordance with the properties of the surface elements located in adjacent cells. In the CA described in [11], each grid element, except for the cells belonging to the obstacle areas (incombustible elements), can be in one of the states: no fire, combustion, burnt out state. Obstacle cells do not change their state. The burning cell will become completely burnt out within 20–60 iterations (depending on the properties of the underlying surface). If a cell is on fire, then at the next time step, the fire can spread to the nearest neighbouring cells with a probability determined depending on the direction of the wind and the characteristics of flammability. Flammability characteristics are stored in a multidimensional array for the entire grid.

Figure 4 shows the results of the CA operation, for the case when the combustible characteristics of the entire cell array are assumed to be the same. On the left – the initial state of the automata, on the right – the result after 200 iterations. Here, obstacles that are not combustible are highlighted in white, in black – completely burnt out areas, in light grey – cells where fire is possible, dark grey areas – burning areas.

The CA model makes it possible to obtain the shape of a fire and to estimate its area and length of the front. As can be seen from the figure, the ignition propagates more intensively in the direction of the wind.

3.3. CA model of liquid spread over a three-dimensional surface

For the cellular automata description of fluid spread, a synchronous automata on a triangulation grid is used, the transition function is implemented in two steps. At the first cycle, the liquid level is
calculated iteratively for all elementary automata. For each iteration, the change in the liquid level is calculated for the triangle on which it is applied and for its neighbours, respectively, taking into account the lie of the ground. The calculation results are saved in an array, the size of which corresponds to the number of mesh elements. This array has a common indexing with the cellular array of the area under consideration. We calculate local volumes at each iteration, after which they are added with others stored in the array.

Since the model does not consider the possibility of liquid absorption by the surface, its total volume at each step of the model time remains constant. In the second cycle, the calculated changes are applied to elementary automata.

Figure 5 illustrates the operation of the cellular automata of liquid spread. The initial test state of the CA (left) corresponds to the uniform distribution of the liquid over the entire surface defined by the triangulation grid. The colour intensity of each cell depends on the level of liquid in it. As you can see from the figure, the entire surface is coloured evenly. On the right, we see the result of performing 10000 evolutions.

![Figure 5. An example of the CA operation for liquid spread after 10000 evolutions](image)

After 10000 evolutionary changes, the values of the liquid level in the grid cells practically stop changing. We can say that the CA comes to a state of equilibrium.

The implemented functionality of statistic collection allows the evaluation of the operation results for models. Figure 6 (in the upper right part of the application window) shows information about the total volume of liquid on the surface and its highest level at the current step.

![Figure 6. Information about the state of the CA](image)

4. Conclusion

In this work, the authors presented an approach to the construction of triangulation meshes that approximate real fragments of the terrain. Algorithms for grid generation by height maps have been developed as well as software tools that implement these algorithms, development environment – Visual Studio Community, programming language – C#.
The described approach allows one to construct cellular automata of various processes and test their operation on arbitrary spatial surfaces of a landscape type. The operation results of CA models given as examples qualitatively correspond to the physical concepts about the course of the processes under consideration.

In the future, we plan to implement mechanisms for visualization of the results for CA-modeling spatial-temporal processes with reference to GIS data [12], since the integration of GIS and mathematical models can be an effective tool in planning and operational management.

Acknowledgments
Separate fragments of the study were carried out with the support of the Russian Foundation for Basic Research and Administration of Krasnodar Region (Project 19-41-230005).

References
[1] Alexandridis A et al 2008 Appl. Math. and Computation 204(1) 191–201
[2] Bandman O 1999 Lecture Notes in Computer Science 1662 395–9
[3] Freire J G and DaCamara C C 2019 Natural Hazards and Earth Systems Sciences 19 169–79
[4] Matyushkin I V and Zapletina M A 2019 Computer Research and Modeling 11(1) 9–57
[5] Trunfio G A 2004 Lecture Notes in Computer Science 3305 385–94
[6] Skvortsov A V 2002 Computational methods and programme 3 14–39
[7] Delaunay B 1934 Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na 6 793–800
[8] Joe B 1991 Computer Aided Geometric Design 8 123–42
[9] Gladskoy I B, Pavlova A V and Telyatnikov I S 2020 Ecological Bulletin of Research Centers of BSEC 17(2) 29–35
[10] Toffolli T and Margolus N 1987 Cellular Automata Machines (USA: MIT Press) p 279
[11] Pavlova A V, Rubtsov S E and Telyatnikov I S 2020 IOP Conf. Ser.: Earth and Environmental Science 579 012104
[12] Clarke K C, Brad O P and Crane M P 2001 Geographic Information Systems and Environmental Modeling (London: Pearson) p 306