A 2D CELLULAR AUTOMATON MODEL OF LIQUID ABSORPTION INTO PAPER FIBERS WITH HYDROPHOBIC TREATMENT

KŘÍŽ, VÍTĚZSLAV[1]; KŘÍŽOVÁ, HANA[2*]; KOCICH, MARTIN[1] AND DALÍKOVÁ, JOHANA[4]

[1] Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, 612 00 Brno, Královo Pole, Czech Republic
[2] Technical University of Liberec, Institute for Nanomaterials, Advanced Technologies and Innovations, Studentská 1402/2,461 17 Liberec, Czech Republic
[3] Faculty of Matematics and Physics, Charles University, Ke Karlovu 2027/3, 120 00 Prague, Czech Republic

ABSTRACT

In this work, the issue of applying water or a homogeneous aqueous suspension with a uniform size of (nano)particles (e.g., ink) to the surface of SBSK (southern bleached softwood kraft) paper with randomly arranged local hydrophobic treatment is investigated and then simulated. Based on the two investigated models, various simulation approaches were compared, an own simulation model was created, and its validity was subsequently demonstrated on the experiments performed.

KEYWORDS

Cellular automaton; Water absorption simulation; Cellulose fibers; Paper pulp, C++ language.

INTRODUCTION

There are a lot of reasons for studying and simulating the absorption of water into hydrophobically treated paper. Some possible reasons include: to understand the mechanisms by which hydrophobic treatment affects the absorption of liquids into paper; to predict the rate and extent of liquid absorption into different types of paper with different hydrophobic treatments; to optimize the hydrophobic treatment of paper for specific applications, such as in packaging or in sanitary products; to investigate the potential of using hydrophobic treatment as a way to improve the strength, durability, and other properties of paper; to compare the performance of different hydrophobic treatments in terms of their ability to reduce liquid absorption in paper, and more.

The professional literature on simulations using a cellular automaton is mostly focused on the interaction of inks, colors, and pigments with paper fibers [1-4]. These simulations are partially applicable to textiles and composites containing fibers. The creation of other simulation programs describing the behavior of fluids in contact with textiles and fibrous formations is desirable regarding the development of medical materials, hygiene supplies, agrotextiles, geotextiles, etc.

CONCEPTS AND APPROACHES

Paper

Paper is a material made from cellulose fibers, typically derived from wood, rags, or grasses. Paper is made by mechanically or chemically separating the fibers from the raw material, and then forming the fibers into a mat or sheet. The mat is then pressed and dried to remove any remaining moisture, resulting in a sheet of paper. Paper can be treated in various ways to improve its strength, durability, and other properties, such as by adding chemicals or other treatments to make it water-resistant or hydrophobic. Paper is commonly used for writing, printing, and packaging, it can also be used for artistic purposes (painting, origami), to produce hygiene items, disposable dishes for fast food and even for building purposes. The specific conditions of the paper production vary depending on its desired properties and use.

SBSK paper

Southern bleached softwood kraft (SBSK) is a type of paper that is made from pulp obtained from southern bleached softwood trees. The production of SBSK paper involves several steps, including logging, pulping, bleaching, and drying. The pulp obtained from the kraft pulping process is typically bleached using chlorine-based chemicals to produce a bright

* **Corresponding author:** Křížová H., e-mail: *hana.krizova@tul.cz*
Received December 20, 2022; accepted February 24, 2023

white color. The pulp is then formed into sheets of paper, which are dried and pressed to remove excess moisture.

The pressing conditions for SBSK paper typically involve the application of heat, pressure, and time. The specific conditions used can vary depending on the desired properties of the paper and the type of equipment being used. In general, SBSK paper is pressed at temperatures of around 120-180 degrees Celsius [5]. The paper is typically pressed for a period of 5-10 minutes, although longer pressing times may be used for certain applications. The pressure applied to the paper during pressing can range from 1-3 bars [6], depending on the specific equipment being used. These pressing conditions are designed to remove excess moisture from the paper and to compact the fibers, which helps to give the paper its strength and durability. The specific conditions used can be adjusted to achieve the desired properties of the finished paper, such as its thickness, density, and smoothness.

The length and thickness of SBSK paper fibers, as well as their distribution, are important factors in determining the properties of the paper. In general, the fibers used to make SBSK paper are obtained from softwood trees, such as pine or spruce. These fibers are typically long and thin, with an average length of 1-3 millimeters and an average thickness of 0.01-0.02 millimeters [6] [7]. The distribution of fibers in SBSK paper is typically uniform, with the fibers evenly distributed throughout the sheet of paper. This helps to give the paper its strength and durability, and it allows it to withstand high levels of stress without tearing or breaking.

Cellular automaton

A two-dimensional cellular automaton is a type of mathematical model that is made up of a grid of cells (lattice), each of which is represented by a set of numerical parameters or simple state from a set of finite number of states (such as "on" or "off", or "1" or "0") [8].

The neighborhood of a cell is the set of cells that are neighboring to that cell, and it is a key factor in determining how the cell will behave and it can have a significant impact on the overall behavior of the system. For example, a cell's neighbors can include cells that are diagonal to the cell, or cells that are a certain distance away from the cell.

The rules of a cellular automaton specify how each cell should behave at each time step. These rules typically depend on the current state of the cell and the states of its neighbors, and they determine how the state of the cell should change at the next time step. The cells interact with each other based on a set of rules, and the resulting patterns that emerge over time can be used to model studied phenomena.

C++ programming language

C++ is a general-purpose programming language. It was designed to be an extension of the C programming language, and it adds support for object-oriented programming, generic programming, and other features. C++ is often used for building large-scale applications, such as operating systems, web browsers, and games. It is also commonly used in the development of high-performance systems, such as those found in the fields of finance, engineering, and scientific computing [9].

C++ is a compiled language, which means that it needs to be converted into machine code before it can be executed. This typically involves using a compiler to translate the C++ code into an executable program. One of the main advantages of C++ is that it is a statically typed language, which means that most errors can be caught at compile time rather than runtime. This can make it easier to debug and maintain large code bases. Additionally, C++ is a highly efficient language, which makes it well suited for applications that need to run quickly and efficiently, such as those that require real-time performance [10].

General simulation procedure

The simulation process included these main steps [11] [12]:

- Experiments to obtain the data from which the simulation is based.
- Analysis, evaluation, and processing of the conceptual design of two published models to point out the possible risks and pitfalls of simulation process.
- Creation of an abstract model: Formation of a simplified description of the investigated system.
- Creation of a simulation model: Writing an abstract model in the form of a program.
- Verification and validation: Verification of the correctness of the model.
- Simulation: Experimenting with a simulation model.
- Analysis and interpretation of the results: gaining new knowledge about the investigated system.

Description of the applied procedures

In this study, a 2D cellular automaton is used for computer simulation because it allows to see how the system behaves over time [13]. The von Neumann neighborhood was used [14], that means a cell's neighbors are the cells that are directly above, below, to the left, and to the right of the cell.

The program was implemented in the C++20 language using the functions of the standard library without the use of third-party libraries, due to the possibility of a higher degree of customization and optimization of the program. CMake tools were used to translate the source files.

The structure of the program was designed in such a way that it was possible to change the used cell classes, their fields, and the rules for them with minimal intervention in the structure of the program. The principles of designing a cellular automaton were freely taken from [15] but were modified due to the use of the C++ language and own experience with the design of cellular automata.

Theoretical analysis of physical processes influencing the water absorption into paper

Paper is a thin, smooth material made by compacting fibers. The fibers used are cellulose-based, and their length, thickness, and placement usually define the structure and other physical properties of the paper [6]. Liquids deposited on such a surface are subject to several physical processes, which are mostly caused by capillary phenomena [1]. Specifically, these are:

- Soaking of liquids into the surface (influenced by the degree of hydrophobicity)
- Particle movement caused by capillary pressure equalization and non-zero particle velocity
- Sedimentation of solids (for example in the case of fine ink, it is affected by the size of the particles)
- Evaporation (depends on the temperature and relative atmospheric humidity around the paper)
- Capillary diffusion of liquids below the surface of the paper

ANALYSIS OF PUBLISHED MODELS

Model 1

The first tested conceptual model (Fig. 1) was based on a study investigating the application of Japanese Nijimi ink on traditional paper [2]. The paper is described as a 2D lattice of cells, each cell is a quadruplet of parameters $[B, C, W, I]$, where B is the thickness of the paper, C is the maximum height of the trapped water column, W is the number of virtual water particles in that cell, and I is the number of virtual particles of ink in the cell. At each discrete time step, the height difference of neighboring water columns is calculated for all cells and the proportional part of the water column without ink deposition is moved.

The advantage of this model is the structure of the paper, which is simulated by creating random lines representing individual cellulose fibers. These fibers affect the thickness of the paper and the maximum amount of water trapped in the water column.

Problem of the model 1

The model does not work with real quantities, the structure of the paper is not well defined, and the main function of the automaton (spreading water) does not even remotely match the experiments carried out.

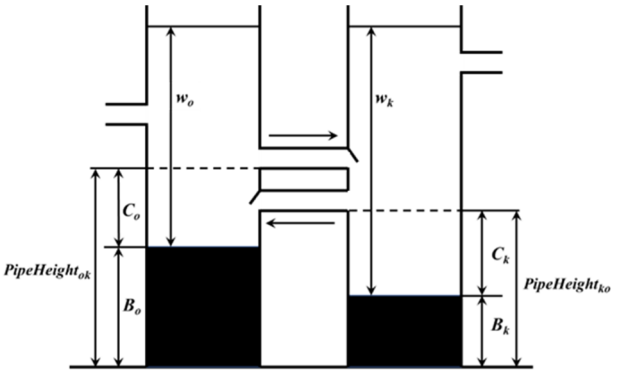

Figure 1. Visualization of the Model 1 [2].

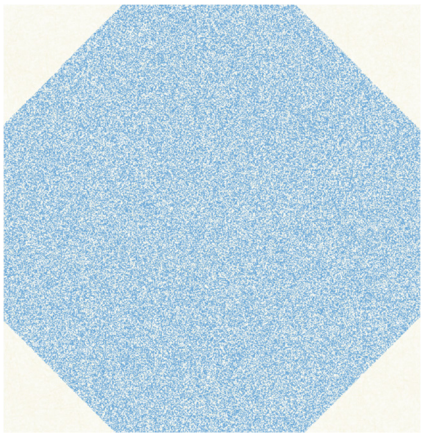

Figure 2. Problem of the Model 1.

When "laying" a circular seed representing a drop of water, after several iterations the state of the automaton is very far from reality (Fig. 2). At the same time, the model does not allow the extension of hydrophobic surface treatment, which was a fundamental problem for our study.

Model 2

The second conceptual model tested [1] was based on a published article on the use of cellular automata for simulating the diffusion of watercolors for computer-generated paintings. In this model, the paper is also described as a 2D lattice of cells, but in contrast to Model 1, each cell is represented as more complex tens $[hx, hy, c, u, v, p, gk, dk, s, m]$, where hx is the direction of paper height between neighboring cells on x-axis, hy is direction of paper height between neighboring cells on y-axis, c is maximum holding capacity of fluid in cell, u is the velocity of fluid flow on x-axis, v is the velocity of fluid flow on y-axis, p is the water pressure in the cell, gk is the amount of pigment in the upper layer of water, dk is the amount of deposited pigment, s is the quantity indicating the saturation level of the paper and m is the mask delimiting the wet part of the paper. Pseudocode describing the behavior of the cellular

```
proc UpdateVelocities(M, u, v, p):
  (u,v) ← (u,v) − h
  Δt ← 1/[max_{i,j} {|u|,|v|}]
  for t ← 0 to 1 by Δt do
    for all cells (i,j) do
      A← u²_{i,j} − u²_{i+1,j} + (uv)_{i+5,j-5} − (uv)_{i+5,j+5}
      B← (u_{i+1.5,j} + u_{i-5,j} + u_{i+5,j+1} + u_{i+5,j-1} − 4u_{i+5,j})
      u´_{i+5,j} ←u_{i+5,j} + Δt(A − μB + p_{i,j} − p_{i+1,j} − κu_{i+5,j})
      A← v²_{i,j} − v²_{i,j+1} + (uv)_{i-5,j+5} − (uv)_{i+5,j+5}
      B← (v_{i+1,j+5} + v_{i-1,j+5} + v_{i,j+1.5} + v_{i,j-5} − 4v_{i,j+5})
      v´_{i,j+5} ← v_{i,j+5} + Δt(A − μB + p_{i,j} − p_{i,j+1} − κv_{i,j+5})
    end for
    (u,v) (u´,v´)
    EnforceBoundaryConditions(M,u,v)
  end for
end proc
```

Figure 4. Pseudocode of the UpdateVelocities procedure.

automaton is described in Fig. 3. (Note: all pseudocodes in this paper are framed.)

The advantage of Model 2 is that the system works with velocities, pressure and constants describing the viscosity of fluids. Due to the complex concept of capillary phenomena, this model is probably very close to reality with a small degree of abstraction.

Problem of the model 2

During the application of the Model 2, several calculation errors were discovered already in the MoveWater procedure, specifically in its UpdateVelocities sub procedure (Fig. 4).

The error lies in the concept of fluid propagation speed with a quadratic dependence of its slope on the current difference of neighboring cells. In addition, the formulas incorrectly favor calculating the pressure in only two quadrants ([+x, +y] and [-x, -y]). This leads to the errors shown in Figures 5 and 6.

```
proc MainLoop(M,u,v,p,g¹,…,gⁿ,d¹,…,dⁿ,s):
  for each time step do:
    MoveWater(M,u,v,p)
    MovePigment(M,u,v,g¹,…,gⁿ)
    TransferPigment(g¹,…,gⁿ,d¹,…,dⁿ)
    SimulateCapillaryFlow(M,s)
  end for
end proc
```

Figure 3. Pseudocode describing the behavior of the Model 2 [1].

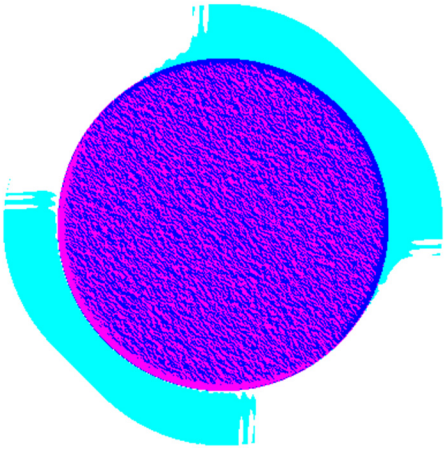

Figure 5. Non-uniform velocity distribution (u = purple, v = blue), state after 10 iterations.

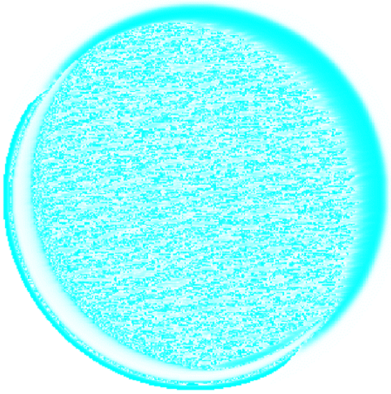

Figure 6. Non-uniform capillary pressure distribution, state after 10 iterations.

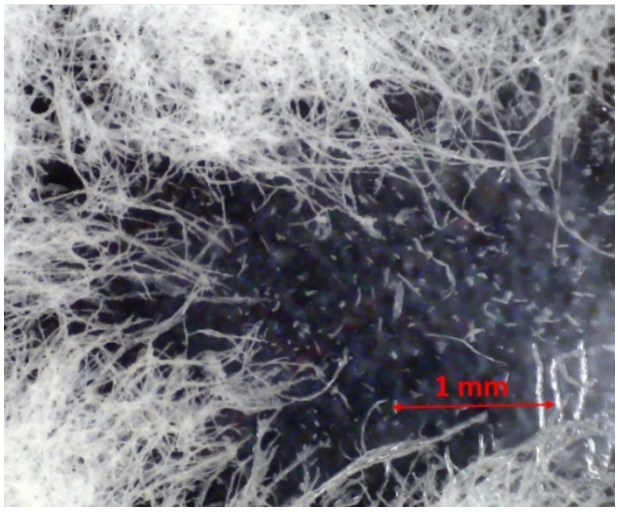

Figure 7. Softwood pulp before papermaking.

CREATION OF ABSTRACT MODEL

Experimental data and observations

The paper used corresponds to the SBSK (southern bleached softwood kraft) type with surface weight 82 g/m2, average fiber length 2.8 mm and, average fiber diameter 20 μm (Fig. 7).

The paper was experimentally coated with hydrophobic spots of mean size of 300 μm. The surface was hydrophobized using polypropylene dissolved in toluene. This gel solution was applied by spraying, and after evaporation of the solvent and thermal fixation, randomly distributed hydrophobic porous structures resembling aerogel were formed. Water soaks into the paper through these spots at a minimum, but after a certain time the upper wet layer connects with the paper base without hydrophobization. As soon as this happens, the canals formed in this way subsequently absorb more water thanks to capillary phenomena and thus effectively bypass the hydrophobic protection in the upper layer of the paper. At the same time, the water spreads in the bottom layer of the paper due to capillary phenomena and this causes further water seepage.

An aqueous suspension of pink ink with a uniform particle size of approximately 165 nm was used to visualize the spreading (absorption) of a drop of water into paper with randomly distributed hydrophobic spots (Fig. 8).

Hypothesis: after flowing through the hydrophobic layer and creating continuous canals through which the water seeps into the paper, the entire area under the hydrophobic spot gets wet.

Paper model design

The paper represents the basic state of a 2D cellular automaton. The cells are arranged in a square grid, where the size of one cell corresponds to 20 μm. This size was chosen based on [2] to accurately create the corresponding paper structure. The second component simulating the structure of the paper are surface irregularities, which simply describe the geometric curvature of the surface.

Bresenham's algorithm [16] for rasterized drawing of lines was used to generate paper fibers. The length of the angle is chosen randomly; the length of the lines corresponds to 2 mm of the real size. In the cells through which the fiber passes, its virtual thickness is increased.

The geometric curvature is like fiber generation simplified by a random process as a depth function of 2D Perlin noise [17]. This depth is an abstract quantity in the range <0, 1> and determines the maximum amount of water trapped in the cell. Real values are greatly simplified by this concept. A graphical visualization of the paper surface is shown in Figure 9.

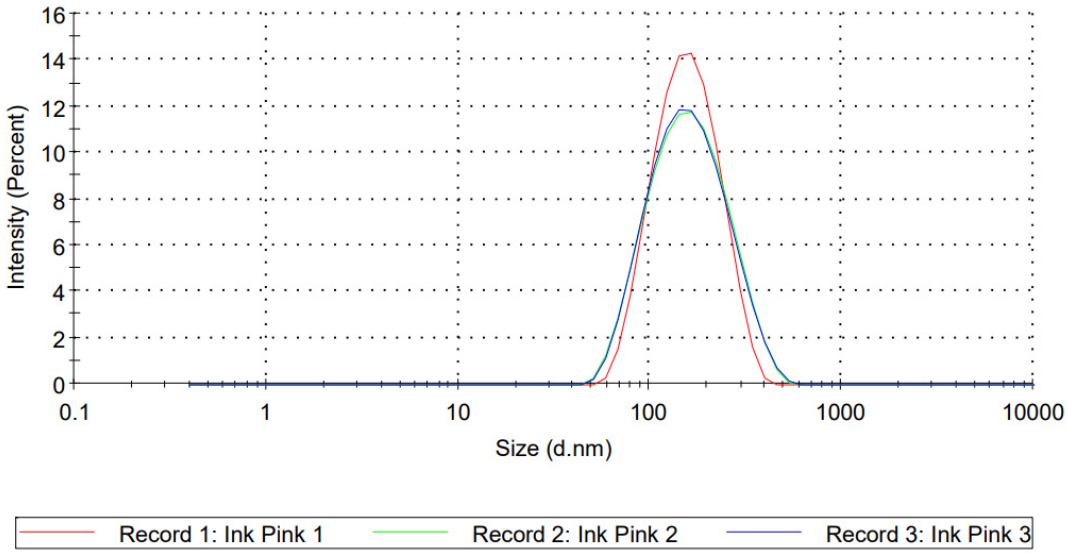

Figure 8. Particle size distribution of the test pink ink as measured by the Zetasizer.

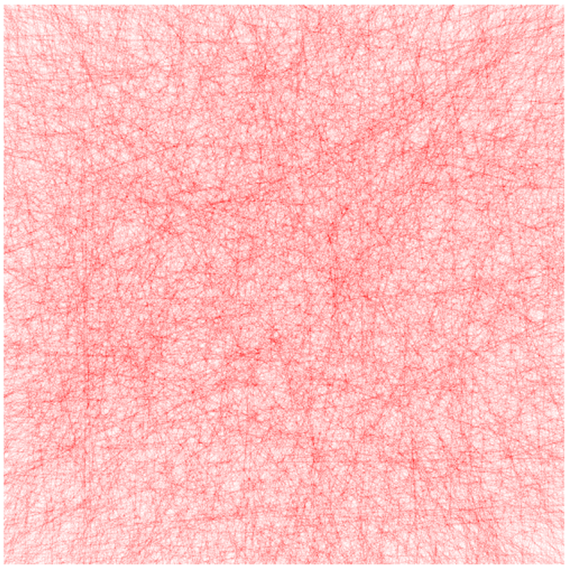

Figure 9. Graphic representation of the *h* parameter on the generated paper

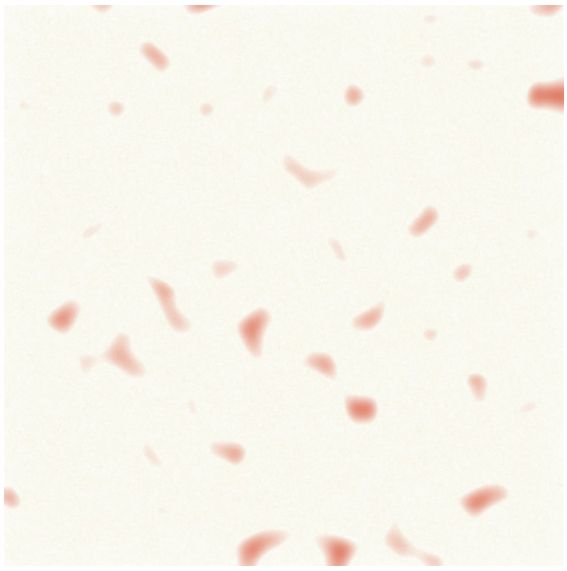

Figure 10. Graphic representation of hydrophobic spots added to paper.

Hydrophobic paper treatment

For the random distribution of hydrophobic spots, the depth function of 2D Perlin noise [17] was again used. As hydrophobically marked islands, noise values with a value of 0.5 were marked and subsequently scaled so that their size corresponded to the mean value of the spots from the measurement (Fig. 10).

Applying a drop of water

The final step in cell preparation was to place an initial drop of water on the paper. Since the applied droplet has a spherical shape, the initial amount of water in the water columns of the cells was determined as the depth of the spherical canopy of the virtual sphere placed below the surface of the paper. For a sphere centered at the point (x_0, y_0) this function has the prescription (1):

$$w = \sqrt{r^2 - (x - x_0)^2 - (y - y_0)^2} - \frac{r}{2} \quad (1)$$

This simplification was created by the authors of the work using an analytical method.

Cell definition

With respect to the parameters specified above, the cell in this model is described as a quintuple [*h, a, p, w_s, w_a*], where:

h is the "height" of the paper, describing its surface structure and the maximum amount of captured water (dimensionless), a is the mask bounding the wet part of the paper, *p* is the degree of hydrophobicity belonging to the given cell (dimensionless), w_s is the amount of water in the upper part of the paper (μg) and w_a is the amount of water absorbed in the paper (μg).

Transition function

In each discrete time step $\boldsymbol{\Delta t}$, the following transition function (2) is performed for all active cells *o* to state $\boldsymbol{S_o}$:

$$\boldsymbol{S_o(t + \Delta t) = f(S_o(t), S_k(t) \mid k \in Neighbour)} \quad (2)$$

Where $\boldsymbol{S_o(t)}$ is the state of the cell at time *t*, given as (3):

$$\boldsymbol{S_o(t) = (activateNearby \circ absorbWater \circ spreadWater \circ flowWater \circ vaporizeWater)(S_o, S_k)} \quad (3)$$

Step 1: activation of neighboring cells

To save computing time, processes are calculated only in active cells. In the initial state, the active cells are under a drop of water, see Fig. 11.

Step 2: absorption of water into the paper

Following capillary phenomena, the water absorption process was approximated by a curve composed of two graphs, as can be seen in Fig. 12. The graph in Fig. 12 is realized by pseudocode shown in Fig. 13.

Coefficients [A, B, C] were chosen based on experiments as [50,0.05,0.001]. This function is intended to simulate the slow "onset" of absorbed water, which accelerates exponentially with its amount, dampens at higher saturation, and at the same time the overall process is delayed due to the hydrophobicity value in the given cell.

```
proc activateNearby(a):
    for cell in neighbours:
        a ← a or a_cell
    end for
end proc
```

Figure 11. Pseudocode extending the active simulation zone to neighbouring cells.

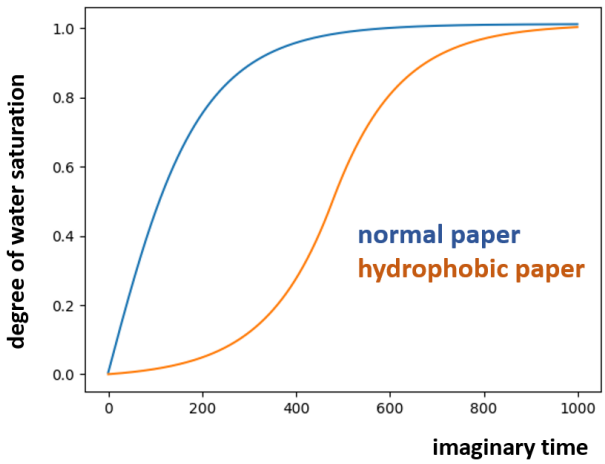

Figure 12. Approximation of the process of absorption of water into paper in time.

$$
\begin{aligned}
&\textbf{proc } absorbWater(\boldsymbol{h}, \boldsymbol{p}, \boldsymbol{w_s}, \boldsymbol{w_a}):\\
&\quad abs\_force \leftarrow 0.01 + \frac{w_s}{A}\\
&\quad \textbf{if } (w_a < p) \textbf{ then}\\
&\qquad absorbe \leftarrow min(B\\
&\qquad\qquad \cdot \sin\left(\frac{\pi}{2} \cdot \frac{w_a}{h}\right) + C, w_s)\\
&\qquad\qquad \cdot abs\_force\\
&\quad \textbf{else then}\\
&\qquad absorbe \leftarrow min(B\\
&\qquad\qquad \cdot \sin\left(\frac{\pi}{2} \cdot \left(h - \frac{w_a}{h}\right)\right)\\
&\qquad\qquad + C, w_s) \cdot abs\_force\\
&\quad \textbf{end if}\\
&\quad w_s \leftarrow w_s - absorbe\\
&\quad w_a \leftarrow w_a + absorbe\\
&\textbf{end proc}
\end{aligned}
$$

Figure 13. Pseudocode for calculation of the amount of absorbed water.

$$
\begin{aligned}
&\textbf{proc } spreadWater(\boldsymbol{h}, \boldsymbol{w_a}):\\
&\quad saturation \leftarrow \frac{w_a}{h}\\
&\quad w_{out} \leftarrow 0\\
&\quad \textbf{for } cell \textbf{ in } neighbours:\\
&\qquad saturation_{cell} \leftarrow \frac{w_{a\,cell}}{h_{cell}}\\
&\qquad w_{out}\\
&\qquad \leftarrow w_{out}\\
&\qquad + \left(\frac{saturation - saturation_{cell}}{2}\right.\\
&\qquad \left. - saturation_{cell}\right) \cdot h \cdot D\\
&\quad \textbf{end for}\\
&\quad w_a \leftarrow w_a - w_{out}\\
&\textbf{end proc}
\end{aligned}
$$

Figure 14. Pseudocode for the calculation of the amount of water transferred to other cells by capillary diffusion.

$$
\begin{aligned}
&\textbf{proc } flowWater(\boldsymbol{h}, \boldsymbol{w_s}):\\
&\quad w_{out} \leftarrow 0\\
&\quad \textbf{for } cell \textbf{ in } neighbours:\\
&\qquad flow_{cell} \leftarrow \frac{w_s - w_{s\,cell}}{2} \cdot E \cdot (1\\
&\qquad\qquad + (p - p_{cell}))\\
&\qquad w_{out} \leftarrow w_{out} + flow_{cell}\\
&\quad \textbf{end for}\\
&\quad w_s \leftarrow w_s - w_{out}\\
&\textbf{end proc}
\end{aligned}
$$

Figure 15. Pseudocode for the calculation of the amount of water transferred to other cells based on hydrophobicity and water column balancing.

Step 3: water diffusion in the bottom layer

The diffusion of water in the lower layer, which is otherwise caused by capillary attractiveness, was simplified in the proposed model to the process of gradually equalizing the degree of saturation of the lower layer with other cells, see Fig. 14.

The amount of water that moves between the cells through this process is not affected by the surface treatment of the paper because it no longer extends to this depth.

Based on testing, the value of 0.1 was chosen as the D coefficient.

Step 4: water diffusion in the top layer

This step in which the error was found in Model 2 is essentially the same as in Model 1. The amount of water moved in each step corresponds to equalizing the water columns of neighboring cells, as described in the pseudocode in Fig. 15.

During this process, the hydrophobicity of the cells is considered (water prefers to flow into a cell with a lower p-value). Based on testing, the coefficient E was set to 0.25.

Step 5: water evaporation

This process removes water from the system with the assumption that the water in the upper layers evaporates faster than the absorbed water. Based on testing, the values [0.001,0.01] were chosen for the [F, G] coefficients, see Fig. 16.

IMPLEMENTATION DETAILS

Multithreading

Computer programs are composed of instructions, which are run sequentially. A thread is a flow of execution of such instructions. However, modern computers can run multiple threads (hence the term Multithreading) at the same time, which allows for a faster parallel execution.

The simulation is divided into horizontal rectangles. Each rectangle is processed by one thread. During processing, the edge of the image (2 cells) and

```
proc vaporizeWater(w_a, w_s):
    v ← F
    if (w_s > v) then
        w_s ← w_s − v
        v ← 0
    else then
        w_s ← 0
        v ← v − w_s
    end if
    v ← v · G
    if (w_a > v) then
        w_a ← w_a − v
    else then
        w_a ← 0
    end if
end proc
```

Figure 16. Pseudocode for the calculation of the amount of evaporated water.

inactive cells are omitted to maintain fast processing and correct memory access. At each time step, logical functions that provide the cell simulation logic will execute for all cells in the cellular automaton and wait for all threads to complete. This way, the program can utilize the full processing power of given program environment and thus causing the simulation to run quickly even for sizes of millions of cells.

Implementation

The cellular automaton uses the 2D field abstraction for simulation. In the source code, the 2D field is implemented as a 1D field indexed by [x + y * W], where W is the width of the field. The field itself does not contain specific elements, but pointers to the given cells to enable caching.

The simulation itself uses two of these fields, which it switches between during the simulation. The 1st field is read-only and contains the old state, which is used to calculate the new state. The 2nd field is intended for storing the new state of the simulation. Two functions are implemented to manipulate these fields: the flip function, which swaps pointers to these fields, and the mirror function, which creates a deep copy of the new field into the old one.

Pointer swapping is fast but cannot be used in all cases. If only parts of the model are updated (for example speed of water movement, amount of water after transfer), data inconsistency may arise. Therefore, this method should only be used after completing the entire simulation step.

Creating a deep copy provides a safe way to manipulate data without possible inconsistency but creating a copy of the data itself is very time consuming compared to changing two pointers.

The implementation of individual models is separated into individual components according to the given model.

VERIFICATION AND VALIDATION OF THE MODEL

The validity of the model could be verified visually by comparison with laboratory results. The output of the simulation is an animated .gif file, which shows the time course of water spreading in the bottom layer of paper. The key points of this simulation are shown in Figure 17.

We managed to verify the main researched hypothesis through simulation. Figure 11 shows how the water gradually spread to the lower layer through the created channels and within a few seconds of their introduction, it was fully saturated. This result, after the optimal determination of the parameters A-G, corresponds to the physical results with which the output of the model was compared (Fig. 18).

CONCLUSION

We managed to detect errors and inaccuracies in two published simulation models and successfully design our own simplified model, thanks to which it is possible to parameterize and visualize water absorption on specially treated paper.

Such a simulation can be used, for example, for artistic needs, but also for simulating the absorption behavior of various liquids and the rate of paper seepage depending on the change of some parameters, especially the thickness of the paper (parameter h), the distribution of hydrophobic areas (parameter p) or specific behavior of the liquid (parameters w), etc.

As non-woven fabrics, for example, have a similar structure and fiber arrangement to paper, this model is also suitable for textile structures where wetting processes, liquid distribution, etc. are very important, e.g., for hygiene products.

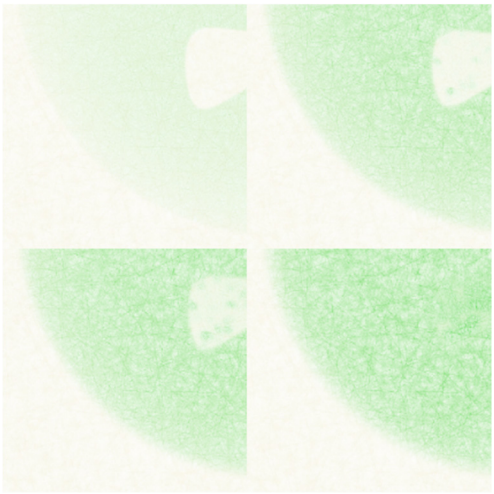

Figure 17. Degree of paper saturation with water at times 3 s, 12 s, 18 s, 26 s.

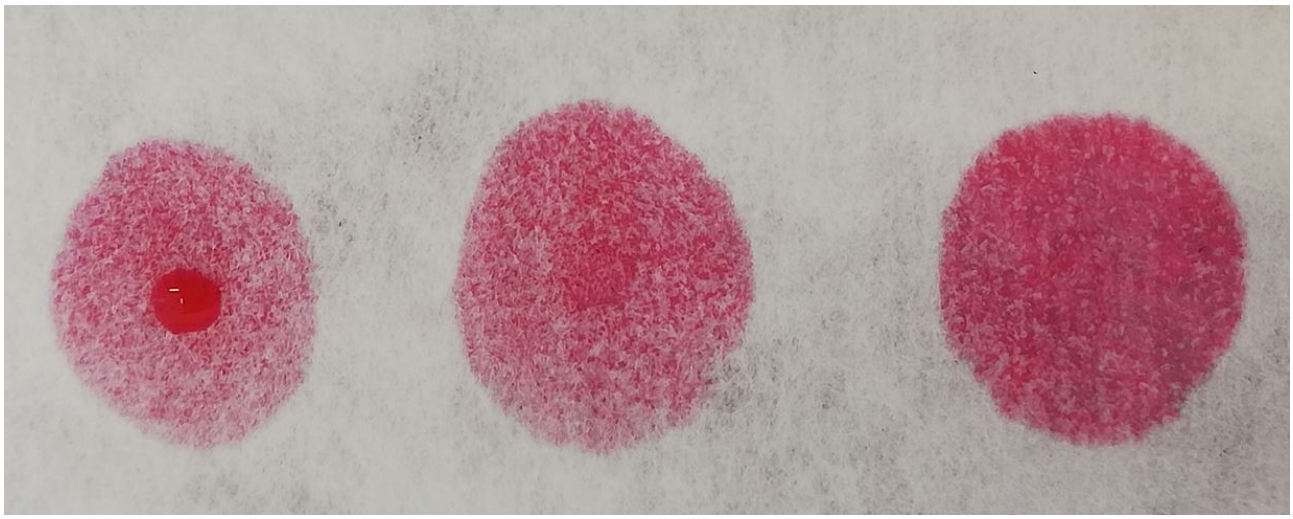

Figure 18. Three samples made at a time interval of 10 s on paper with surface hydrophobization using a spray-applied hydrophobic varnish: left with incipient surface absorption, middle with visible hydrophobic defects, right almost absorbed.

Acknowledgement: *This work was supported by Iceland, Liechtenstein and Norway through the EEA Grants and the Technology Agency of the Czech Republic within the framework of project "inherently Flexible Aerogels for energy effiCient structurES (i-FACES)" (Grant number TO01000311).*

REFERENCES

1. Curtis C.J., et al.: Computer-generated watercolor. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, 1997, pp. 421-430
2. Zhang Q., et al.: Simple cellular automaton-based simulation of ink behaviour and its application to Suibokuga-like 3D rendering of trees, The Journal of Visualization and Computer Animation, 10(1), 1999, pp. 27-37. https://doi.org/10.1002/(SICI)1099-1778(199901/03)10:1<27::AID-VIS194>3.0.CO;2-C
3. Small, D.: Simulating watercolor by modeling diffusion, pigment, and paper fibers. In: Image Handling and Reproduction Systems Integration. SPIE, 1460, 1991, pp. 140-146. https://doi.org/10.1117/12.44417
4. Ren, Y., et al.: Ink diffusion simulation in Chinese calligraphy using navier–stokes equation. In: Simulations, Serious Games and Their Applications. Springer, Singapore, 2014, pp. 15-29. https://doi.org/10.1007/978-981-4560-32-0_2
5. Roberts J. C.: The chemistry of paper, Royal Society of Chemistry, 2007, 204 p.
6. De Assis T., et al. : Comparison of wood and non-wood market pulps for tissue paper application, BioResources, 14(3), 2019, pp. 6781-6810. https://doi.org/10.15376/biores.14.3.6781-6810
7. Zambrano F., et al.: Upcycling strategies for old corrugated containerboard to attain high-performance tissue paper: A viable answer to the packaging waste generation dilemma, Resources, Conservation and Recycling, 175, 2021, 105854. https://doi.org/10.1016/j.resconrec.2021.105854
8. Codd E. F., Ashenhurst R. L.(ed.): Cellular automata, Academic press, 2014, 132 P.
9. Stroustrup B.: An overview of the C++ programming language. Handbook of object technology, CRC Press LLC, Boca Raton, 1999, 23 p.
10. Soulié, J.: C++ Language tutorial, 2008, online: http://www.cplusplus.com/doc/tutorial [cit. 14.12.2022]
11. Zeigler B.P, et al.: Theory of modelling and simulation, 3rd edition, Elsevier, 2018, 435 p.
12. Fishwick P.A.: Simulation model design and execution: Building Digital Worlds, Prentice-Hall, 1995, 448 p.
13. Chopard B., Droz M.: Cellular automata. Modelling of physical, Cambridge University Press, 1998, 341 p.
14. Wolnik B., et al.: Number-conserving cellular automata with a von Neumann neighborhood of range one. Journal of Physics A: Mathematical and Theoretical, 50(43), 435101. https://doi.org.10.1088/1751-8121/aa89cf
15. Shiffman D.: Chapter 7: Cellular automata. Nature of Code, 2012.
16. Koopman P.: Bresenham line-drawing algorithm. Forth Dimensions, 8(6), 1987, pp. 12-16.
17. Hart J. C.: Perlin noise pixel shaders. In: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware, 2001, pp. 87-94.2022.