

# Simulation of diffusion using a modular cell dynamic simulation system

Christoph Leberecht<sup>a,b</sup>, Florian Heinke<sup>a,c</sup> and Dirk Labudde<sup>a,\*</sup>

<sup>a</sup>*Faculty of Applied Computer Sciences and Biosciences, University of Applied Sciences Mittweida, Technikumplatz 17, Mittweida, Germany*

<sup>b</sup>*Biotechnology Center (BIOTEC), TU Dresden, Tatzberg 47-49, Dresden, Germany*

<sup>c</sup>*Faculty of Chemistry and Physics, TU Bergakademie Freiberg, Akademiestrasse 6, Freiberg, Germany*

Received 19 January 2017

Accepted 13 April 2017

**Abstract.** A variety of mathematical models is used to describe and simulate the multitude of natural processes examined in life sciences. In this paper we present a scalable and adjustable foundation for the simulation of natural systems. Based on neighborhood relations in graphs and the complex interactions in cellular automata, the model uses recurrence relations to simulate changes on a mesoscopic scale. This implicit definition allows for the manipulation of every aspect of the model even during simulation. The definition of value rules  $\omega$  facilitates the accumulation of change during time steps. Those changes may result from different physical, chemical or biological phenomena. Value rules can be combined into modules, which in turn can be used to create baseline models. Exemplarily, a value rule for the diffusion of chemical substances was designed and its applicability is demonstrated. Finally, the stability and accuracy of the solutions is analyzed.

**Keywords:** Cellular automaton, graph, diffusion, cell biology, systems biology

## 1. Introduction

The simulation of complex physical, chemical, and biological phenomena has become an important tool for the research in life sciences. Models that mimic the behavior of natural systems support the understanding and analysis of laboratory experiments [1]. In systems and cell biology such models are valuable for inferring the life-sustaining chemical transformations in cells [2], the self-organization of the cytoskeleton synthesis [3], and phenotypes of whole organisms [4]. Quantitative models are able to give actual predictions about the quantities in a system. Most models are generally systems of ordinary or partial differential equations (PDE). Such systems are composed of multiple equations, where each equation represents the concentration of a chemical compound. It is of high importance to model

these equations as accurately as possible to correctly capture the phenomena observed on the biological level. Even small differences in the initial conditions may produce large discrepancies between the simulation and the experimental outcome [5]. Hence, each experimental model has to be addressed differently, and many parameters have to be measured and set to describe the phenomenon observed under specific conditions. While helpful for a small group of scientists, those models can rarely be adapted for problems that are the result of a similar underlying phenomenon [6]. Scientists can only observe consequences of physical and chemical laws that are entangled in complex systems such as cells and organisms. So it is no surprise, that most models do not describe the “real shape” of the problem and only characterize the guise of the phenomenon. For an illustration of the problem, consider a medical doctor who tries to cure an illness. Only by approaching and treating the underlying causes, he can hope to cure the patient. Treating the symptoms is only viable in the most severe cases. Similarly, if we are able

---

\*Corresponding author: Dirk Labudde, Bldg. 8: Richard-Stuecklen Bau, Office 8-105, Am Schwanenteich 6b, 09648 Mittweida, Germany. Tel.: +49 3727 58 1469; Fax: +49 3727 58 21469; E-mail: dirk.labudde@hs-mittweida.de.

to characterize the processes in a cell and know their interactions, it is possible to modify the system and obtain the behavior that is required. The creation of new fundamental laws is a large step in the development of systems biology [7]. Modeling those fundamental processes and combining them as system-theoretic concepts is still one of the greatest challenges in the field of systems biology [8, 9].

One technique to explore the general traits of interacting systems and spatio-temporal patterns are cellular automata (CA). Originated in the 1940s, CA are simulations discrete in time, space, and state. By applying a simple set of rules – which depend on automata elements, so-called cells, and their respective associations between each other – at each time-step, remarkably complex behaviors emerge from such automata [10, 11]. This discrete character to CA is tailored for the computation using traditional computers that operate at finite precision by design. Dynamic Cellular Automata, as an agent-based adaptation of CA, have been shown to describe a multitude of different cellular or biochemical processes [12]. Although CA have been used in many fields of biology [13], they played a minor role in recent years, because powerful tools such as ECell [14] and VCell [15] started to dominate the research field. Both tools and many others alike are based on PDEs that are composed and specified by the user of the software. With all reactions and processes defined, a numerical solution of the system of PDE is generated and calculated. The user is able to inspect the results and adjust parameters to fine-tune the model to fit the results gained via laboratory experiments.

Space is one of the key components when modeling the processes in cells. In the last decade simulations in systems biology have been moving from temporal-only models to spatio-temporal models [16]. An established practice for simulations with a spatial component is to partition the simulation space into smaller compartments that are able to exchange components or information. This holds true for CA, which are traditionally defined to handle spatial neighborhood relations on two- or three dimensional geometric tilings. The neighbors of each cell are defined by adjacent edges or corners.

Another approach to model natural systems are stochastic simulation algorithms (SSA) such as Chemical Master Equations (CME), which can be necessary when biological phenomena depend on stochastic fluctuations [17]. CME are able to model diffusion using particle position distributions – which

are either compartment-based or reliant on Smoluchowski equations – and chemical reactions, based on probabilities of molecular reactions during a defined time period. Assuming a spatially homogeneous system the changes in concentration can be simulated.

PDEs, SSA and CA represent different techniques to explore the behavior of dynamic systems, all designed to define the change that occurs in events or time steps. In PDEs this change is explicitly defined from the starting point and for all following steps in time. Therefore it is possible to poll the state of the system by solving a single equation. In comparison, using CA, numerical methods, or SSE the next state of the system is implicitly defined by its current state, thus adjusting parameters or even states can be accomplished readily at every time step, but every state in between the current and the required time step has to be calculated [18, 19]. This leads to a consideration of the advantages and disadvantages of implicit definitions in opposition to explicit definitions. The updating procedure of explicit systems depends on the value of the previous time step, and may even be analytically formulated as a closed form solution. Therefore the determination of the updates is straightforward and computationally inexpensive. As a compromise, explicit methods are only conditionally stable, meaning errors at any stage of the computation or divergence in the starting conditions are amplified and not attenuated as the simulation progresses. Hence, explicit solutions require very small time steps, especially when there are major changes in a short time period. Implicit systems can include values of the current and previous time steps. The calculations performed in each time step are generally more difficult, resulting in a higher time complexity of the simulation. Nevertheless, implicit schemes are stable and can tolerate larger time steps than explicitly defined systems [20]. This also includes a robustness to unexpected changes. Especially in biological environments the ability to respond and adjust to spontaneous disturbances, such as molecular signals from other cells or changes in temperature is vital in order to create a realistic and robust model [21].

In 2001 TOMITA proposed that systems biology should rise to a grand challenge of the 21st century [22]. The construction of a mathematical model of the whole cell is a key aspect in this challenge. In order to understand the complex system of the cell, the fundamental processes that govern its behavior have to be understood. A modular system that allows to easily add and remove certain aspects of a model is a possible approach to analyze the forces that drive

biological systems. A seed model can be used as a basis to experiment with different pathways, reaction types, and parameter sets, without the need to modify the entire system due to adding a single component. To evaluate the possibilities that CA offer for modularized spatio-temporal modeling of natural systems, we define a graph-oriented CA and design a function to simulate diffusion processes. The diffusion of six different molecules within a rectangular shaped space is simulated with the model proposed in this paper and validated with a numerical approach. We discuss the stability of the solutions gained concerning different time and space resolutions and suggest a general guideline to select those parameters. In systems that simulate multiple chemical components and/or multiple phenomena, it can be difficult to select the best size for the time step. In numerical analyses different algorithms are available to estimate a stable step width, such as Ruge-Kutta methods [19]. With the proposed model it is possible to determine the time and spatial step size, without additional estimation methods. The user of the model has the freedom to weigh and choose between simulation accuracy, complexity, and stability before performing the simulation of the system.

## 2. Methods

The concept of *cellular graph automata* has been pioneered by Angela Wu [23] in the late 1970s. The research with graph automata was mainly focused on investigating the properties of graphs and applications to graph grammars. Nevertheless, the dynamic definition of neighborhood in CA is a powerful aspect to capture geometries that cannot be defined easily by a static geometric definition.

### 2.1. Modeling cellular graph automata

The definition of *cellular d-graph automata* proposed by Wu and Rosenfeld [23] has been adapted for this work. For reference purposes we will present a short informal definition. A *d-graph*  $\Gamma$  is a graph with labeled nodes and numbered edges. A distinguished label element  $\#$  exists – if a node is not labeled with  $\#$  it has exactly  $d$  edges emanating from it. Further a *finite-state automaton*  $M$  on d-graphs is defined as a double  $(Q, \delta)$ , where  $Q$  is a nonempty set of states and  $\delta$  the transition function, which maps a new state to a node depending on the current state, the edge numbers and neighbor states. Further, the cellular d-graph automaton  $\mathcal{M}$  is a triple  $(\Gamma, M, H)$  with a d-graph  $\Gamma$

and  $M$  as defined before and  $H$  the *neighbor vector* of a node  $n$ . An automaton  $M$  is placed at each node  $n$  of  $\Gamma$ . The neighbor vector  $H$  supplies the required information about the neighbors.

For the adaption of *cellular graph automata* (CGA) presented here, we exchanged the definition of d-graphs with *finite graphs*, that are triplets  $(X, U, \alpha)$  where  $X$  and  $U$  are two finite disjoint sets and  $\alpha$  is a function [24]. This implies that nodes and edges are no longer labeled and do not rely on a bounded degree  $d$ . Additionally, the states  $Q$  is introduced at the level of CGA, because they are globally defined.

**Definition 1** (Cellular Graph Automaton). Let a cellular graph automaton be a quadruple  $CGA = (V, E, Q, \alpha)$ , for which:

- $V$  is a finite set of graph automaton cells,
- $E$  is a finite set,
- $Q$  is a finite set, and
- $\alpha : E \mapsto V \cup V$  is a partial function.

$V$ ,  $E$ , and  $Q$  are disjoint sets. The set  $V$  contains the nodes of the graph with elements denoted as  $v_1, v_2, \dots, v_n$  and each element of this set is a graph automaton cell (see Definition 2). Elements  $e_1, e_2, \dots, e_u$  originate from the set of edges  $E$ . The set  $Q = \{q_1, q_2, \dots, q_r\}$  is the set of possible states. The number of elements in each set is given as  $n = |V|$ ,  $u = |E|$ , and  $r = |Q|$ . The function  $\alpha(e_i) = (v_j, v_k)$  assigns the relation between a pair of nodes  $v_j, v_k \in V$  and an edge  $e_i \in E$ . If such a relation exists between two nodes  $v_j, v_k$  they are called neighbors.

A *graph automaton cell* (GAC) is a modified version of the finite state automaton  $M$ , where we keep the essence of the transition function  $\delta$ .

**Definition 2** (Graph automaton cell). Let a graph automaton cell be a triple  $GAC = (p, \delta, q)$ , for which

- $p \in \mathbb{R}^g$  is the position
- $\delta : Q^* \mapsto Q_{CGA}$  is a total function (the state rule), and
- $q \in Q_{CGA}$  is the current state.

The position of a node in a space of dimension  $g$  is represented by the position  $p \in \mathbb{R}^g$ . Each node is enclosed by the Voronoi region of  $p$ , which represents the surface or volume a node is responsible for during the simulation. Formally the Voronoi region  $R_{v_k}$  of a node  $v_k$  can be described by all points in a space  $x \in \mathbb{R}^g$  with a metric  $d_R$ , such that

$$R_{v_k} = \{x \in \mathbb{R}^g \mid d_R(x, p_{v_k}) \leq d_R(x, p_{v_j}), \forall j \neq k\},$$

as described by Aurenhammer in [25]. In the context of CGA the metric  $d_R$  is called *geometric distance*.

CGA are composed of nodes, and unlike CA, their neighborhood relations are based upon the connections represented by edges. Let  $d_N$  be the *neighborhood distance* of two nodes. In other words:  $d_N$  defines the number of edges on the shortest path between two nodes. Therefore the distance  $d_N$  between two nodes  $v_k, v_j \in V$ , is  $d_N(v_k, v_j) = 1$ , if and only if there exists an edge  $e \in E$  such that  $\alpha(e) = (v_k, v_j)$ . Further, the distance  $d_N$  between two distant nodes  $v_1, v_n \in V$  can be defined by the shortest path between them. Hence let  $(v_1, v_2, \dots, v_n) \in V \times V \times \dots \times V$  be the shortest path between two nodes, such that  $v_i$  is neighbored to  $v_{i+1}$  for  $1 \leq i < n$ . The resulting distance will be  $d_N(v_1, v_n) = n - 1$ . In contrast to CA, CGA do not define specific boundary conditions. Different pseudo boundaries can be created by adding new edges that specify the desired interactions.

While CA operate in a predefined tiling, GAC act depending on their neighbors, defined by  $\alpha$  and represented by a multiset  $N$ . Let the multiset  $N$ , called *neighbor states*, be a double  $N = (Q, m)$ , for which  $Q$  is the finite set of states and  $m : Q \mapsto \mathbb{N}$  is a function. The function  $m(q)$  maps a state  $q \in Q$  to a natural number. This is the number of times a neighboring node is in the state  $q \in Q$ . If  $Q \subset Q_{CGA}$ , then the multiplicity  $m$  of the elements  $Q_{CGA} \setminus Q$  is 0. Consequently, the multiset  $N_{v_k}$  of a node  $v_k$  contains the states  $q$  of nodes  $v_i$  that are connected to  $v_k$  with an edge:

$$N_{v_k} = \{q_{v_i} \in Q_{CGA} | d_N(v_k, v_i) = 1\}$$

The cardinality of the multiset is the sum of all multiplicity functions and equal to the degree  $deg$  of the node  $v_k$ :

$$|N_{v_k}| = \sum_{q \in Q} m(q) = deg(v_k)$$

As an example consider a node  $v_k$  in state  $q_1$  with three neighboring nodes in  $q_3$  and one neighbor also in state  $q_1$ . The available states consist of  $Q_{CGA} = \{q_1, q_2, q_3\}$ . The resulting multiset of neighbor states is  $N_{v_k} = \{q_1, q_1, q_1, q_3\}$  and the appropriate functions are  $m(q_1) = 3$ ,  $m(q_2) = 0$  and  $m(q_3) = 1$ .

The transition of a node from one state  $q(t)$  at time  $t$  to the next state  $q(t + \Delta t)$  is defined by the state rule  $\delta$ , which can be either deterministic or stochastic. The step size  $\Delta t$  is uniform. The domain  $Q^*$  of  $\delta$  is a multiset of states from  $Q_{CGA}$  that consists of

the neighbor states  $N_{v_k}$ , in addition to the state of node  $v_k$  itself and maps to the next state  $q(t + \Delta t)$  of  $v_k$ . An exemplary deterministic transition function would be:

$$\delta(Q_{v_k}^*) = \begin{cases} q_1 & \text{if } m(q_1) > \frac{|N|}{2}, \\ q_2 & \text{if } m(q_2) > \frac{|N|}{2}, \\ q_3 & \text{else.} \end{cases}$$

This deterministic transition function determines the next state of a node, by the number of neighboring nodes in a certain state. If the number of nodes in state  $q_1$  is larger than the number of nodes in state  $q_2$  the next state of this node will be  $q_1$  and vice versa. The state  $q_3$  would be assigned if the number of nodes in  $q_1$  and  $q_2$  are exactly even.

With these definitions the state of a node can be set. Considering the initial representation of nodes as a segment of space inside of a biological system, possible states could be comprised of the environment or compartment a node is enclosing. In this context, the state of a node may be any organelle, such as chloroplast, endoplasmic reticulum, or Golgi apparatus. The states or objects can be allowed to move through the cell with rules adapted from the String definition in [26] or remain static. In order to handle continuous values, such as concentrations of species, the previous definition is augmented to *Extended Graph Automaton Cells* (EGAC), which are used to enrich the concept of CGA. Here, each node is able to accommodate a set of entities that represent its content.

The description *species* is used to generally refer to sets of atoms, molecules, molecular fragments, ions and similar entities, subject to a chemical process or to a measurement. Each species  $s \in S$  represents such a chemical compound. A value function  $f(s) \in \mathbb{R}$  maps a value to the species  $s$  that represents any quantitative or physicochemical property. The first quantitative property defined for the purpose of this paper is the concentration function  $f_c(s)$  that maps the concentration of a species  $s$  to any node  $v_k$ . Additionally, the function  $f_d(s)$  maps a species to its diffusivity. This function is independent from any node, since the diffusivity of a species is considered constant regardless of its position in the CGA. Other functions can readily be implemented, and the set of all value functions is referred to as  $\mathcal{F}$ .

**Definition 3** (Extended graph automaton cell). An extended graph automaton cell is a quintuple  $EGAC = (\delta, \Omega, q, W)$ , for which

- $p \in \mathbb{R}^s$  is the position
- $\delta : Q^* \mapsto Q_{CGA}$  is a total function (the state rule),
- $\Omega$  is a set of functions  $\omega : W^* \times s \mapsto W$  (the value rules),
- $q \in Q_{CGA}$  is the current state, and
- $W$  is the current value set.

The Definition 2 of GAC also applies to the multiset  $Q^*$ , the transition function  $\delta$  and the current state  $q$ . The value functions  $\Omega$  and the current value set  $W$  are new components to this definition. The value set  $W$  contains the sets of values defined by each value function  $f \in \mathcal{F}$  and for each species  $s \in S$ . For example, consider two species  $s_1, s_2 \in S$  and two value functions  $f_c$  mapping the concentration and  $f_d$  mapping the diffusivity of a node  $v_k$ . The resulting value set could look like  $W_{v_k} = \{f_c(s_1) = 0.3, f_c(s_2) = 0.4\}, \{f_d(s_1) = 1.2, f_d(s_2) = 2.7\}$ .

The domain  $W^*$  of each value rule  $\omega$  consists again of the neighboring value sets. Each value set  $W_{v_i}$  of the nodes  $v_i$  that share an edge with the node  $v_k$  is included in the  $M_{v_k}$  set.

$$M_{v_k} = \{W_{v_i} | d_N(v_k, v_i) = 1\}$$

The set of value rules  $\Omega$  contains all value rules that are to be applied to the value set of the node  $v_k$ . A single value rule  $\omega$  maps from the set of multiple value sets  $W^*$ , including neighbor value sets  $M_{v_k}$  and the node  $v_k$  itself, to the new value set  $W$ . A rule  $\omega$  is used to calculate the change of any  $f_i(t, v, s)$  of a species  $s$ , at node  $v$  and time  $t$ . For example, the arithmetic mean of the concentrations in a node can be assigned with:

$$\omega_{f_c}^{Avg}(s_i) = \Delta f_c(v_k, s_i) = \frac{1}{|W_{v_k}^*|} \cdot \sum_{f_c(s_i) \in W_{v_k}^*} f_c(s_i).$$

$\omega_{f_c}^{Avg}(s_i)$  denotes the value rule for averaging the concentration  $f_c$  of any species  $s_i$ ,  $\Delta f_c(v_k, s_i)$  describes the change in concentration at node  $v_k$  of the species  $s_i$  and  $|W_{v_k}^*|$  the number of neighbors of  $v_k$ . A visualization of this definition and its interpretation is depicted in Fig. 1.

The idea behind this definition is the creation of multiple value rules that represent natural phenomena inside of the cell. It is possible to define multiple value rules that modify the same value in a value set. For example, consider the diffusion of a species into neighboring compartments in addition to the conversion into another species during a chemical reaction. Both events would modify the concentration of the

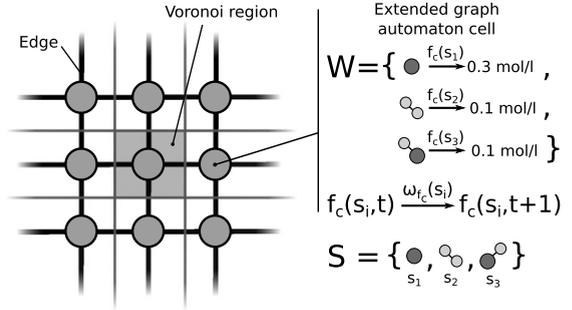


Fig. 1. A section of a cellular graph automaton illustrating nodes, edges, and Voronoi regions. A set of species  $S$  is globally defined and a set of attributes can be assigned to every individual species. Each extended graph automaton cell (see Definition 3) contains the current concentration  $f_c(s_i)$  of a species  $s_i$ . The species and values depicted were only chosen as an example. The value function  $\omega_{f_c}(s_i)$  defines a rule for the change in concentration for each species.

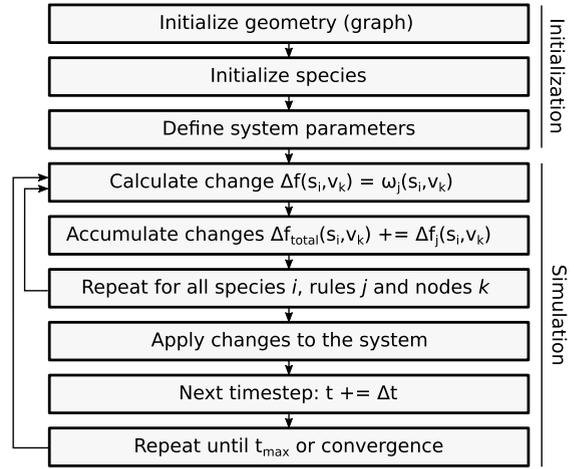


Fig. 2. The flowchart describes the general sequence of steps during the initialization process and actual simulation of CGA.

same species. In this case the sum of the changes for each species and value function will be applied to the node.

The size of the time step is uniform and can be parametrized as described in Section 2.3. This enables the simulation of multiple time scales depending on the phenomenon that is to be addressed. Additionally, the size of the time step provides the possibility to choose between a more accurate prediction (using numerous small time steps) or a faster simulation (using fewer large time steps). A schematic representation of the sequence of operations that are needed to setup and perform a simulation is depicted in Fig. 2.

To evaluate the proposed model, the process of diffusion has been implemented as a value rule  $\omega^{Dif}$ . Diffusion is one of the fundamental processes that governs the behavior of all natural systems on a molecular level. It is possible to derive an implicit value rule that is able to describe the flow of matter between the nodes in CGA. The derivation of this rule is presented in the following section.

## 2.2. Modeling diffusion

Diffusion is a natural physical process in which molecules disperse from an area of high concentration proportional to a concentration gradient. This phenomenon is describes the random motion of all particles in solutions, called Brownian motion. Fick proposed the Second Law of Diffusion, in which the flux of matter in a liquid is proportional to the gradient of its concentration  $c$  scaled by a factor  $D$  [27]:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}. \quad (1)$$

This equation needs to be transformed into a recurrence relation, which then enables the implementation into CGA. Using the finite differences method as shown in [20], the infinitesimal small steps in time  $\partial t$  and space  $\partial x$  can be formulated using finitely small steps  $\Delta t$  and  $\Delta x$ . The concentration was rewritten to  $c(t, x)$ , to indicate its dependence on time  $t$  and the position  $x$ :

$$\underbrace{\frac{1}{\Delta t} (c(t + \Delta t, x) - c(t, x))}_{\text{change of } c \text{ per } \Delta t} = D \frac{\partial}{\partial x} \cdot \underbrace{\frac{1}{\Delta x} (c(t, x + \Delta x) - c(t, x))}_{\text{change of } c \text{ per } \Delta x}.$$

By calculating the limit with  $\Delta x, \Delta t \rightarrow 0$  of this function, it is possible to reformulate Fick's second law in a discrete manner. The reformulation of the second space-derivative is performed specifically for CGA. Since the introduction of a neighborhood-oriented approach is required, it is necessary to distribute the spatial differences as uniformly as possible around the point of interest  $x$ , so that the isotropy of the geometric space is conserved. Subsequently, it is necessary to define the neighborhood of the current position  $x$ . Thus, let  $H = \{h_1, \dots, h_d\}$  be the set of nodes that are adjacent to a node  $v_k$ . This relationship is derived by collecting all nodes where

$d_N(v_k, h_i) = 1$  (see Definition 1). In conclusion, position  $x$  will now be represented as a node  $v_n$  and  $h_i \in H$  are nodes, that have a distance of  $\Delta x = d_N$  to  $v_k$ . Since there is now a finite amount of neighbors, the concentration of each neighboring node needs to be considered:

$$\begin{aligned} \frac{1}{\Delta t} (c(t + \Delta t, v_k) - c(t, v_k)) &= D \frac{1}{d_N^2} \cdot \\ & \left( (c(t, h_1) - c(t, v_k)) + \dots + (c(t, h_d) - c(t, v_k)) \right) \\ &= D \frac{1}{d_N^2} \cdot \sum_{h_i \in H} (c(t, h_i) - c(t, v_k)). \end{aligned}$$

Since the time scale of CGA is uniform and homogenous, it is possible to assume  $\Delta t = 1$  and  $\Delta d_N = 1$  without loss of generality if the nodes in the graph are arranged uniformly. When applying these terms, the previous equation results in:

$$c(t + 1, v_k) - c(t, v_k) = D \cdot \sum_{h_i \in H} (c(t, h_i) - c(t, v_k)).$$

and by rearranging further the final equation is

$$\begin{aligned} c(t + 1, v_k) &= D \cdot \underbrace{\sum_{h_i \in H} c(t, h_i)}_{\text{amount entering}} - \\ & \underbrace{D \cdot d \cdot c(t, v_k)}_{\text{amount leaving}} + \underbrace{c(t, v_k)}_{\text{current amount}}. \end{aligned}$$

Now it is possible to translate the equation to a value rule usable in CGA. The new concentration of a species  $s$  can be calculated with:

$$\begin{aligned} \omega_{f_c}^{Dif}(s_i) = \Delta f_c(v_k, s_i) &= f_d(s_i) \cdot \sum_{f_c(s_i) \in M_{v_k}} f_c(s_i) - \\ & f_d(s_i) \cdot |M_{v_k}| \cdot f_c(v_k, s_i) \quad (2) \end{aligned}$$

## 2.3. Parameterizing and scaling

As mentioned in the definition of CGA, the spatial and time scales of CGA are dimensionless and uniform. The scaling factors that result from the true dimensionality of the system (time step size and node distance) have to be included in the value or transition rules, in order to affect the model system.

The diffusion coefficient  $D(s)$  represented by  $f_d(s)$  is the only constant in the diffusion value rule  $\omega(s)$

that connects the model system with specific characteristics of nature, and therefore provides an opportunity to scale the diffusion value rule using node distance and the size of the time step.  $D$  is measured as a unit of  $\text{area}^2 \cdot \text{time}^{-1}$  and can be retrieved from literature or estimated from the molecular weight or volume. HAYDUK and LAUDIE compiled a list of the diffusivities of 89 small molecules and developed a reliable correlation [28]. YOUNG did the same for 143 proteins [29]. Using these correlations it is also possible to estimate the diffusion coefficient and in turn parametrize the simulation. In case of molecules the coefficient is often given in  $\text{cm}^2 \cdot \text{s}^{-1}$ .

The distance between two nodes  $\Delta d$  and the length of a time step  $\Delta t$  has to be specified in order to simulate a defined environment (see Fig. 2). If these parameters are given, the diffusion constant  $D$  can be scaled to the size of the system. However, this requires the arrangement of nodes to be uniform across the covered space, so that  $d_G(v_j, v_k) \sim d_N(v_j, v_k)$ . We therefore propose the setup work flow depicted in Fig. 3. Ideally, the geometric distance  $d_G$  is equal for all pairs of neighboring nodes and can therefore be used as  $\Delta d$ . The time step can be chosen depending on the environmental configuration. The final value of  $D$ , which is used and applied in CGA, is rescaled to represent  $\Delta t$  and  $\Delta d$  correctly. The scaled diffusivity  $\bar{D}$  is calculated using

$$\bar{D} = f_d(s) = D(s) \cdot \frac{\Delta t}{\Delta d^2}. \quad (3)$$

#### 2.4. Verification

The viability of the simulated diffusion process, calculated by the implementation of the suggested model, is assessed in comparison to a numerical solution of the two-dimensional representation of the diffusion equation [30]. To compare the different approaches, the half-life  $\tau_h$  was measured. In this context,  $\tau_h$  is the time required for half of the concentration of the steady state to accumulate at a specified position. The half-life was used to contrast both approaches, since this measure is highly delicate and important to approximate the speed of the diffusion process [31]. Therefore, a good consensus between the predicted half-lives indicates a good overall compliance.

The sample system to be simulated was constructed as a square with a side length of 2500 nm. This square is partitioned into two rectangles with a width of 1250 nm (half of the system) and a height of 2500 nm.

One of the rectangles is set up to have a starting concentration of  $1.0 \text{ mol} \cdot \text{l}^{-1}$  and the other rectangle is set to a concentration of  $0.0 \text{ mol} \cdot \text{l}^{-1}$ . The system's initial setup and progression is depicted in Fig. 4. Multiple species have been considered to assess the viability of the system. We have chosen some of the smallest molecules that may be of interest in cellular biology. Those species should be the most critical in the system because large quantities are moving between nodes. The diffusivity covers ranges from  $4.40 \cdot 10^{-5} \text{ cm}^2 \cdot \text{s}^{-1}$  for hydrogen gas up to  $6.40 \cdot 10^{-6} \text{ cm}^2 \cdot \text{s}^{-1}$  for ethane-1,2-diol (see Table 1) [28].

Initially, a numerical approach was used to simulate the diffusion process in a closed system. This computation was performed using Octave [32]. The simulation uses an explicit finite difference method with a first order upwind in time and a second order central difference in space scheme [30, 33]. To specify the square shape of the simulation space, four reflecting Neumann boundaries were set. After specifying the systems boundaries, the kinematic viscosity of the system is set to exhibit the properties of the desired species  $\nu = D(s)$ . In the simulation, the concentrations of the far right side are observed and the system is simulated until it approaches the steady state  $c = 0.5 \text{ mol} \cdot \text{l}^{-1}$ . Finally, the simulated half-life  $\tau_h$  is measured.

Additionally, using the definition presented in this paper, a CGA is designed to simulate the same rectangular system (see Fig. 4B). The number of nodes at either side of the grid  $z$  is adjusted and the total number of nodes in the graph is  $|V| = z^2$ . The global size of the system (2500 nm side length) will stay the same, only the distance between nodes  $\Delta d$  has to be recalculated for the scaling of the diffusion coefficient. Different automata were designed to assess the influence of node distance  $\Delta d$  and length of time steps  $\Delta t$ . Therefore 682 different systems have been simulated ranging from 10 to 70 nodes per side (using 31 samples in this range) and time steps of 10 ns to 1500 ns (using 22 samples in this range). Furthermore, these simulations have been performed for all mentioned species.

### 3. Results and discussion

A mathematical concept was developed that allows the simulation of multiple chemical species, subject to natural phenomena described by so-called value rules  $\omega$ . The geometry of the model system

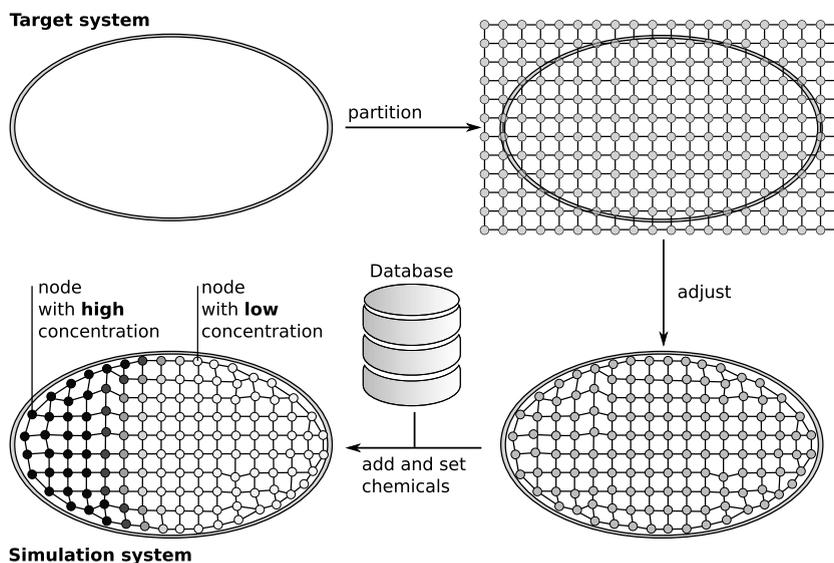


Fig. 3. The first step to setup a simulation with cellular graph automata is to define the desired geometry of the system that one wants to simulate. Afterwards, the system is partitioned by covering it with a net-like graph. Each node is responsible for a subsection of space. Further adjustment methods can be applied to truncate and rearrange nodes, so that the true geometry is represented accurately. After the CGA has been adjusted to cover the simulation space, chemical species can be added to the system. Species and their attributes can be retrieved from databases such as ChEBI [35] or Pubchem [34]. For every node in the system the concentration can be set individually, a gradient qualitatively indicates nodes with a high concentration of a particular species in darker color and low concentrations in light color.

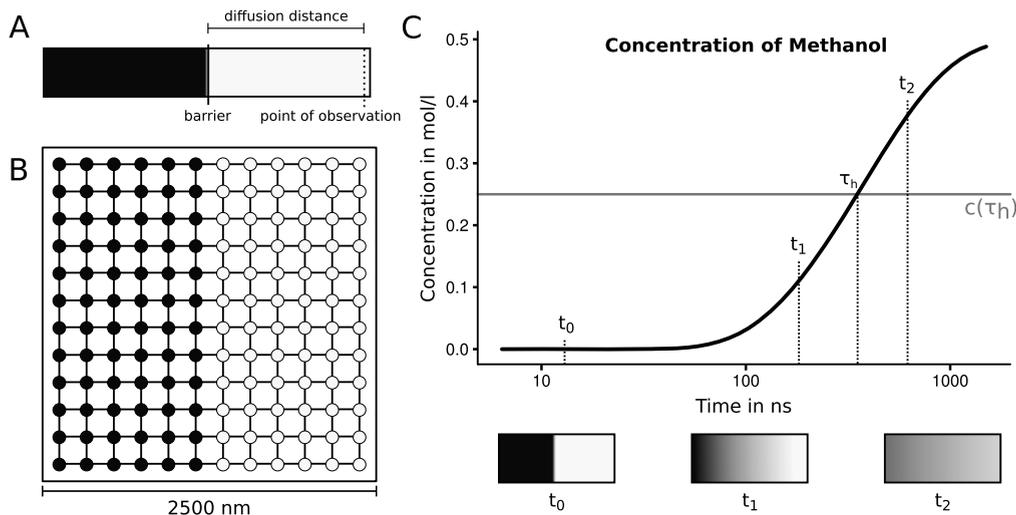


Fig. 4. A: The initial setup of the proposed experiment. A tube is filled with two solutions, separated by a central barrier. One half (indicated with light color) contains a high concentration of the proposed substance, the other half does not contain the solved substance. The distance between the barrier and the point of observation is the diffusion distance, for which the half life will be measured. To start the experiment the barrier is removed. Now the time is measured until half of the equilibrium concentration is reached at the point of observation. B: The initial setup of the rectangular graph automaton with twelve nodes side length. The nodes contain concentrations of  $1.0 \text{ mol} \cdot \text{l}^{-1}$  (indicated with dark color) and  $0.0 \text{ mol} \cdot \text{l}^{-1}$  (indicated with light color). C: A schematic depiction of the progression of the diffusion process of methanol at the point of observation. The half life is measured when the concentration reaches half of the equilibrium. This concentration of  $0.25 \text{ mol} \cdot \text{l}^{-1}$  is indicated with a grey line. The coloring shown in the three boxes illustrate the state of the system at three points in time based on the experimental progression.

is divided into multiple smaller subspaces that are simulated individually (see Fig. 3). Value rules are used to change the concentration of species inside of

subspaces of the modeled system. The model itself is scale free, only the value rules have to be adjusted to the actual time and space regime of the simulation.

Table 1  
Comparison of the simulation results

Species	$D(s) (\frac{cm^2}{s})$	$\tau_h^{CGA}$ (ns)	$\tau_h^{Num}$ (ns)	Difference (ns / %)
Hydrogen gas	$4.40 \cdot 10^{-5}$	135.9	131.8	4.1 / 3.07
Ammonia	$2.28 \cdot 10^{-5}$	262.2	254.3	-7.9 / 3.09
Methanol	$1.66 \cdot 10^{-5}$	360.1	349.3	-10.8 / 3.09
Benzene	$1.09 \cdot 10^{-5}$	548.4	532.0	-16.4 / 3.08
Succinic acid	$8.60 \cdot 10^{-6}$	695.1	674.3	-20.8 / 3.07
Ethane-1,2-diol	$6.40 \cdot 10^{-6}$	934.0	906.1	-27.9 / 3.08

The graph automata simulation has been performed with 5 ns time steps and 40,000 nodes (200 x 200). The values for the diffusivity have been experimentally determined and are presented in [28].

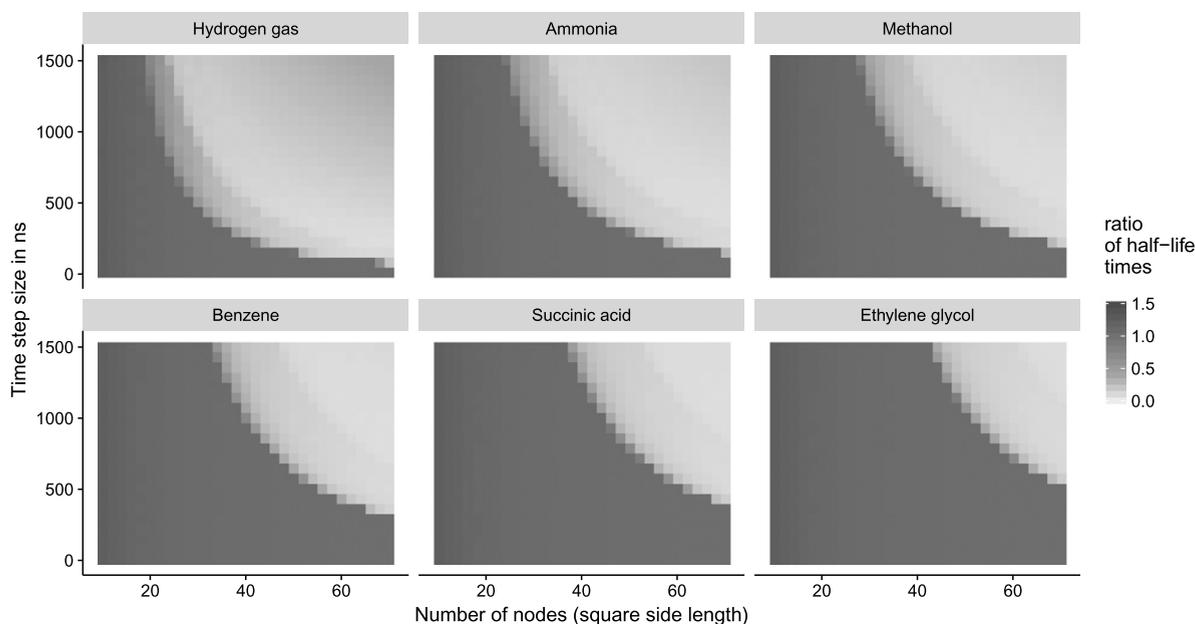


Fig. 5. Simulations for six different substances. The number of nodes (side length of the square graph) ranges from 10 to 70 (implicitly changing the node distance  $\Delta d$  to maintain the overall simulation space) and the time step size  $\Delta t$  ranges from 10 ns to 1500 ns. Each dot corresponds to a simulation performed with different parameters for  $\Delta t$  and  $\Delta d$ . The ratio of the half life  $\tau_h^{CGA} : \tau_h^{Num}$  is represented by the brightness of the dots. Dim gray sections with values near 1.0 corresponds to predictions with a low deviation when compared to numerical methods. Half lives that occurred later than expected (values bigger than 1.0) are indicated in darker gray and light gray regions report models that overestimate  $\tau_h^{CGA}$  (values smaller than 1.0).

After the underlying geometry of the simulation is defined, chemical components can be added to each node of the system (see Fig. 3). Different attributes can be assigned to the chemical components that may be subject to change during the progression of the simulation. The diffusion requires the concentration and the diffusivity of each species. The concentration is node specific and will be changed during the simulation depending on the concentration of the surrounding cells, whereas the diffusivity is species specific and remains constant over time. The two global parameters that determine the scaling of the system are species and node independent. Those are the node distance  $\Delta D$  (spatial step size) and the time step size  $\Delta t$ . Following the initializa-

tion of all required parameters, the simulation can be started. In each time-step the transition rules are applied and new values are set for the next iteration. A step-by-step progress of the setup of such a system is reenacted in Appendix A and another example for the implementation of value rules in the case of first order reactions is given in Appendix B.

The simulation system and model were implemented in order to verify their applicability. We developed the application programming interface Simulation of Natural Systems using Graph Automata (SiNGA) in the Java programming language. The source code is provided on GitHub under GNU General Public License Version 3 (see [github.com/cleberecht/singa](https://github.com/cleberecht/singa)).

The results of an exemplary system (see Fig. 5) indicate, that the simulation using CGA with an increasing number of nodes and a decreasing time step size converge to the half life calculated with the numerical simulation. However, the graphs also show that the system is unstable when a large number of nodes (equivalent to small node distances  $\Delta d$ ) is used in conjunction with large time steps. This is to be expected if the combined amount of concentration to be distributed among the neighbors in one time step is larger than the concentration a node currently contains. Hence, the time scale is underestimated and negative values ensue. Consequently, the initial setup of the system has to be done carefully. The dimensions of the system have to match the speed of the species with the highest diffusivity  $D(s)$ . We derived the following threshold

$$\frac{\Delta d^2}{\Delta t} > \max_{v \in V}(\deg(v)) \cdot \max_{s \in S}(\overline{D}(s)) - \max_{v \in V}(\Delta c(v_i, v_j)), \quad (4)$$

that traces the arches that can be seen in Fig. 5. Here, let  $\max_{v \in V}(\Delta c(v_i, v_j))$  be the maximal difference in concentration between two neighboring nodes. Additionally, the node with the maximal degree and the species with the highest diffusivity are taken into account. If the parameters of a CGA do not satisfy the inequality, the system is unlikely to reach a meaningful result. As a result, this inequality approximates the stability of the global system, leading to a rather pessimistic prediction that resembles the worst case scenario of a possible configuration. By inserting the actual values of a proposed CGA simulation into the inequality, adjustments can be made to create a stable system. The geometric setup of the system can be changed in two different ways. On the one hand, it is possible to reduce the globally highest node degree. On the other hand, species with high diffusivity may be removed, or steep concentration gradients can be smoothed across multiple nodes. If the system setup is to be taken as fixed, the environmental parameters  $\Delta d$  and  $\Delta t$  can be adjusted to increase stability. Smaller time steps entail more calculations in intermediate steps to stop values from changing too rapidly. Increasing the number of nodes has the same effect. When the size of the time step approaches zero and the number of nodes reaches positive infinity, the predicted half life of calculated with CGA approaches the result gained by classical numerical methods. The system gets more accurate when small time steps are used in conjunction with

small node distances. Simultaneously, the runtime of the simulation increases.

As depicted in the flowchart in Fig. 5 for every time step every node needs to be addressed at least once. The complexity of each time step is  $\mathcal{O}(n^2)$ , dominated by the number of nodes  $|V|$  and number of species  $|S|$ . Conclusively, the end user needs to find a trade-off between accuracy and time complexity depending on the requirements for the simulation. To gain a brief overview of the system's behavior an inaccurate setup might suffice. If the diffusion process needs to be determined accurately, more nodes and small time steps are required. Finally, the half life  $\tau_h$  calculated by classical numerical simulations and those from CGA are compared in Table 1. Remarkably the difference between both methods amounts to about three percent across all species.

#### 4. Conclusion

PDEs are an established strategy to simulate a variety of natural systems. In systems that have predictable behaviors and outcomes PDEs flourish to their full potential. In biology however, one often encounters systems that are very complex and influenced by multiple aspects. In these cases PDEs reach the limits of their applicability. Using implicit systems such as CA or CGA, it may be possible to simulate different systems that cannot be tackled efficiently using PDE.

With the proposed definition of CGA we have shown that it is possible to simulate diffusion of different chemical species to a high degree of accuracy. Diffusion is one of the most fundamental processes that many biological systems rely on. The system shown here is able to easily adjust to different time and space scales with little adjustments to the configuration of the system. Using the underlying structure of graphs it is possible to define different topologies. It is possible to add plenty of species to a simulation. Furthermore, the user is able to simulate and observe the system in order to improve the understanding of its current behavior. It is even possible to alter the concentration of a species at a specific node, or the structure of the graph at any point during the simulation, due to the implicit definition. Finally, not much mathematical understanding that is needed to set the system up (see setup process in Appendix A). As shown in Fig. 2, the user has to initialize the graph first, than choose the required species and finally define the systems parameters  $\Delta d$  and  $\Delta t$ . To guide

the definition process of the parameters, the threshold (Equation 4) can be consulted. A trade-off can be chosen between accuracy and time complexity of the simulation. Hence, the end user can decide whether qualitative predictions are sufficient or precise simulations are required. Databases such as PubChem, ChEBI, or UniProt allow the automatic extraction of features for different species that are relevant for the estimation of critical parameters (such as the diffusivity) [34–36].

Computer simulations are able to act as a bridge between theory and experiment [37]. In the past the gap between theory and experiment for molecular behavior has been overcome by molecular dynamic simulations [38, 39]. However there is still a lot to be done to bring theory and experiment together when it comes to the dynamics of cellular processes. First steps were done by many researchers over the past centuries. Nevertheless, the gap between generalizable simulations and experimental outcome still needs to be closed. We hope that the different fields of science will work together to make a similar approach to “Cell Dynamic Simulations” possible.

In the near future, research will be directed towards the creation of value rules for different chemical reaction types and membrane transport. The setup process for simulations will be further simplified by providing possibilities to extract information from relevant databases and by offering visual feedback using the graphical user interface. Additionally the SiNGA application programming interface (available on GitHub [github.com/cleberecht/singa](https://github.com/cleberecht/singa)) is under constant development.

## Acknowledgments

This work was supported by the European Social Fund (ESF) and the Free State of Saxony, Germany. We would like to thank our colleagues, especially Sebastian Bittrich and Florian Kaiser, for thought-provoking impulses and fruitful discussions. Additionally we thank Hannah Siewerts for proofreading and feedback on the manuscript.

## Author contributions

All authors designed the research. C.L. designed and implemented the model and simulation strategy, and evaluated the performance. C.L. and F.H. wrote the paper.

## References

- [1] B.D. Ventura, C. Lemerle, K. Michalodimitrakis and L. Serrano. From *in vivo* to *in silico* biology and back, *Nature* **443**(7111) (2006), 527–533.
- [2] C. Chassagnole, N. Noisommit-Rizzi, J.W. Schmid, K. Mauch and M. Reuss. Dynamic modeling of the central carbon metabolism of *Escherichia coli*, *Biotechnology and Bioengineering* **79**(1) (2002), 53–73.
- [3] F. Nédélec, T. Surrey and E. Karsenti. Self-organisation and forces in the micro-tubule cytoskeleton, *Current Opinion in Cell Biology* **15**(1) (2003), 118–124.
- [4] J.R. Karr, J.C. Sanghvi, D.N. Macklin, M.V. Gutschow, J.M. Jacobs, B. Bolival, N. Assad-Garcia, J.I. Glass and M.W. Covert. A whole-cell computational model predicts phenotype from genotype, *Cell* **150**(2) (2012), 389–401.
- [5] H. Poincaré. *Science and method*. Courier Corporation, 2013.
- [6] M.L. Shuler, S. Leung and C.C. Dick. A mathematical model for the growth of a single bacterial cell, *Annals of the New York Academy of Sciences* **326**(1) (1979), 35–52.
- [7] O. Wolkenhauer and M. Mesarović. Feed-back dynamics and cell function: Why systems biology is called systems biology, *Molecular BioSystems* **1**(1) (2005), 14–16.
- [8] M.D. Mesarović. Systems theory and biology view of a theoretician, In *Systems Theory and Biology*, pages 59–87. Springer, 1968.
- [9] A.F. Villaverde and J.R. Banga. Reverse engineering and identification in systems biology: Strategies, perspectives and challenges, *Journal of The Royal Society Interface* **11**(91) (2014), 20130505.
- [10] C. Adami. *Introduction to artificial life*, volume 1. Springer Science & Business Media, 1998.
- [11] S. Wolfram. *A new kind of science*, volume 5. Wolfram Media Champaign, 2002.
- [12] D.S. Wishart, R. Yang, D. Arndt, P. Tang and J. Cruz. Dynamic cellular automata: An alternative approach to cellular simulation, *In Silico Biology* **5**(2) (2005), 139–161.
- [13] G.B. Ermentrout and L. Edelstein-Keshet. Cellular automata approaches to biological modeling, *Journal of Theoretical Biology* **160**(1) (1993), 97–133.
- [14] M. Tomita, K. Hashimoto, K. Taka-hashii, T.S. Shimizu, Y. Matsuzaki, F. Miyoshi, K. Saito, S. Tanida, K. Yugi, J.C. Venter, et al. E-cell: Software environment for whole-cell simulation, *Bioinformatics* **15**(1) (1999), 72–84.
- [15] B.M. Slepchenko, J.C. Schaff, I. Macara and L.M. Loew. Quantitative cell biology with the virtual cell, *Trends in Cell Biology* **13**(11) (2003), 570–576.
- [16] C. Lemerle, B.D. Ventura and L. Serrano. Space as the final frontier in stochastic simulations of biological systems, *FEBS Letters* **579**(8) (2005), 1789–1794.
- [17] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions, *J Phys Chem* **81**(25) (1977), 2340–2361.
- [18] M. Brin and G. Stuck. *Introduction to dynamical systems*. Cambridge University Press, 2002.
- [19] R.M.M. Mattheij, S.W. Rienstra and J.H.M.T. Thije Boonkamp, *Partial differential equations: Modeling, Analysis, Computation*. Siam, 2005.
- [20] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Courier Corporation, 2012.
- [21] M.E. Csete and J.C. Doyle. Reverse engineering of biological complexity, *Science* **295**(5560) (2002), 1664–1669.

- [22] M. Tomita. Whole-cell simulation: A grand challenge of the 21st century, *Trends in Biotechnology* **19**(6) (2001), 205–210.
- [23] A. Wu and A. Rosenfeld. Cellular graph automata. i. basic concepts, graph property measurement, closure properties, *Information and Control* **42**(3) (1979), 305–329.
- [24] P. Rosenstiehl, J.R. Fiksel and A. Holliger. *Graph theory and computing*, chapter Intelligent Graphs: Networks of Finite Automata Capable of Solving Graph Problems, pages 219–265. Academic Press, 1972.
- [25] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure, *ACM Computing Surveys (CSUR)* **23**(3) (1991), 345–405.
- [26] B. Chopard. A cellular automata model of large-scale moving objects, *Journal of Physics A: Mathematical and General* **23**(10) (1990), 1671.
- [27] A. Fick. Ueber diffusion, *Annalen der Physik* **170**(1) (1855), 59–86.
- [28] W. Hayduk and H. Laudie. Prediction of diffusion coefficients for nonelectrolytes in dilute aqueous solutions, *AIChE Journal* **20**(3) (1974), 611–615.
- [29] M.E. Young, P.A. Carroad and R.L. Bell. Estimation of diffusion coefficients of proteins, *Biotechnology and Bioengineering* **22**(5) (1980), 947–955.
- [30] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, *Advances in Computational Mathematics* **6**(1) (1996), 207–226.
- [31] D. Axelrod, D.E. Koppel, J. Schlessinger, E. Elson and W.W. Webb. Mobility measurement by analysis of fluorescence photobleaching recovery kinetics, *Biophysical Journal* **16**(9) (1976), 1055.
- [32] J.W. Eaton, D. Bateman, S. Hauberg and R. Wehbring. *GNU Octave version 4.0.0 manual: A high-level interactive language for numerical computations*. 2015.
- [33] R.J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.
- [34] S. Kim, P.A. Thiessen, E.E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B.A. Shoemaker, et al. Pubchem substance and compound databases, *Nucleic Acids Research*, page gkv951, 2015.
- [35] J. Hastings, P.D. Matos, A. Dekker, M. Ennis, B. Harsha, N. Kale, V. Muthukrishnan, G. Owen, S. Turner, M. Williams, et al. The chebi reference database and ontology for biologically relevant chemistry: Enhancements for 2013, *Nucleic Acids Research* **41**(D1) (2013), D456–D463.
- [36] UniProt Consortium et al. Uniprot: A hub for protein information, *Nucleic Acids Research*, page gku989, 2014.
- [37] M.P. Allen. Introduction to molecular dynamics simulation. *Computational Soft Matter: From Synthetic Polymers to Proteins* **23** (2004), 1–28.
- [38] M. Karplus. Development of multiscale models for complex chemical systems: from h+ h2 to biomolecules (nobel lecture), *Angewandte Chemie International Edition* **53**(38) (2014), 9992–10005.
- [39] M. Levitt. Birth and future of multiscale modeling for macromolecular systems (nobel lecture), *Angewandte Chemie International Edition* **53**(38) (2014), 10006–10018.
- [40] S. Lloyd. Least squares quantization in pcm, *IEEE Transactions on Information Theory* **28**(2) (1982), 129–137.

## Appendix A. Simulation setup

*Step 1: Definition of the target system.* The layout of the underlying graph of the CGA is the first step in the setup of a simulation. The geometry and dimensionality of the target system has to be respected and eventually assumptions have to be made. A one dimensional system can be simulated by using a line of nodes, where each node is only connected to the next node. By also connecting the first and the last node of the system a periodic boundary condition in terms of regular cellular automata can be introduced. A two dimensional system can be composed in different ways, for instance by arranging four nodes around a central node (see example system in Fig. 4), a classical cellular automaton with a squared tiling can be recreated. It would also be possible to create a hexagonal tiling by introducing six neighbors to each node. Again, by connecting opposite sides of the system periodic boundary conditions are introduced. In theory the modeling of arbitrary dimensionalities should be possible depending only on the connection of nodes in the graph.

*Example:* For the purpose of this guide, a two dimensional system is chosen without periodic boundary conditions.

*Step 2: Placement of the corresponding nodes.* If the CGA should represent the actual geometry of a cell, a microscopic image can be used as a guideline to place the nodes of the graph. The number of nodes that are supplied in this step determine the accuracy of the system to a large extend. Each node is responsible for the simulation of a small region of space during the simulation. More nodes increase accuracy as well as runtime. Since the nodes need to be placed uniformly in the desired space a relaxation algorithms such as Lloyds Algorithm [40] should be applied and afterwards the nodes should be connected using a stringent distance cutoff. After this step the graph should uniformly span across the simulation space. Now the average distance between neighboring nodes is used to parametrize  $\Delta d$ .

*Example:* A simple square simulation space with an area of  $100 \text{ mm}^2$  is filled with 100 nodes. The large square is partitioned in 100 small squares, and at the center of each square a node is placed. Now the space is uniformly covered. Afterwards, neighboring nodes are connected and the distance between two nodes should be uniformly the same with  $\Delta d = 0.9 \text{ mm}$ . The distance might seem surprising but is a result of

the nodes being placed inside of the square and not on the border.

*Step 3: Species selection and concentration assignment.* Now the species of interest need to be specified and the concentrations have to be set. Each node requires a starting concentration.

*Example:* We might choose the molecule Benzene that has a concentration of  $0.5 \text{ mol} \cdot \text{l}^{-1}$  in the two leftmost columns. Hence we set the value function  $f_c(\text{Benzene}) = 0.5$  for the corresponding nodes and  $f_c(\text{Benzene}) = 0.1$  for all remaining nodes.

*Step 4: Choosing  $\Delta t$  and determining  $\bar{D}$ .* If we want to simulate the diffusion of Benzene its diffusivity has to be set as well. To scale the diffusivity for the proposed setup system, the size of the time step is required. To choose the time step size most suitable for ones requirements, the proposed threshold (Equation 4) will be consulted, with the already determined values inserted.

*Example:* For the system that was introduced up to this point, the following threshold can be derived:

$$\frac{0.9^2}{\Delta t} > 4 \cdot \bar{D}(\text{Benzene}) - 0.4$$

A node is neighbored to a maximum of four neighbors, the node distance has been set to  $0.9 \text{ mm}$ , and the maximal difference in concentration is  $0.4 \text{ mol} \cdot \text{l}^{-1}$ . Now the following three time steps will be considered:  $1 \text{ s}$ ,  $50 \text{ s}$  and  $100 \text{ s}$ . To scale the experimentally determined diffusivity  $D(\text{Benzene}) = 1.09 \cdot 10^{-5} \text{ cm}^2 \cdot \text{s}^{-1}$  [28] to  $\bar{D}(\text{Benzene})$ , Equation 3 is calculated for each proposed  $\Delta t$  and inserted into the threshold ( $D$  is brought to the same unit). This insertion results in the following inequalities:  $0.81 > -0.394$  for  $1 \text{ s}$ ,  $0.162 > -0.131$  for  $50 \text{ s}$  and  $0.0081 \not> 0.136$  for  $100 \text{ s}$ . The time steps  $1 \text{ s}$  and  $50 \text{ s}$  are sufficient for a stable system, but a time step of  $100 \text{ s}$  would be too large. The larger the absolute difference between both sides of the inequality, the more accurate the resulting simulation.

*Simulation.* After the graph is constructed and the the time step size  $\Delta t$ , node distance  $\Delta d$ , and scaled diffusivity of all desired species are set, the simulation can begin. During simulation, the change for each node is calculated first using the the value rules  $\omega$ . After all changes have been calculated, they are added to the previous values and the time step is increased.

**Algorithm 1:** Simulation of CGA

---

```

1 while  $t < t_{max}$  do
2   foreach  $v_k$  in  $V$  do
3     foreach  $s_i$  in  $S$  do
4        $\Delta f_c(v_k, s_i) \leftarrow 0.0$ 
5       foreach  $\omega_j$  in  $\Omega$  do
6          $\Delta f_c(v_k, s_i) \leftarrow \Delta f_c(v_k, s_i) + \omega_j(v_k, s_i)$ 
7     foreach  $v_k$  in  $V$  do
8       foreach  $s_i$  in  $S$  do
9          $f_c(v_k, s_i) \leftarrow f_c(v_k, s_i) + \Delta f_c(v_k, s_i)$ 
10     $t \leftarrow t + \Delta t$ 

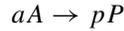
```

---

**Appendix B. Implementing value rules**

The implementation of additional value rules is dependent on their requirement for neighboring values. If the value rule does not require information from neighboring values, its implementation is straight forward. As an example a simple case of chemical kinetics shall be demonstrated:

The general case of a first order reaction with two chemical species  $A$  and  $P$  and their stoichiometric coefficients  $a$  and  $p$  can be denoted in the form:



The reaction rate of a first-order reaction  $A \rightarrow P$  can be described by:

$$v = -\frac{1}{a} \frac{dc(A)}{dt} = \frac{1}{p} \frac{dc(P)}{dt} = kc(A)$$

for which:

- $v$  is the reaction rate (in  $\text{mol} \cdot \text{l}^{-1} \cdot \text{s}^{-1}$ ),
- $c(S)$  is the concentration of a species  $s$ ,
- $k$  is the rate coefficient, and
- $t$  is the time.

The differentials will be transformed to discrete changes, to transition to CGA.

$$v = \frac{\Delta c(S)}{\Delta t} = \frac{1}{p} \frac{\Delta c(P)}{\Delta t} = -\frac{1}{a} \frac{\Delta c(A)}{\Delta t} = kc(A)$$

Further, the concentration will be rewritten to  $c(s, t)$ , to describe  $c$  as a function depending on species and time:

$$\begin{aligned} \frac{1}{\Delta t} \Delta c(s, t + \Delta t) &= \frac{1}{p} \frac{1}{\Delta t} \Delta c(P, t + \Delta t) = \\ &= -\frac{1}{a} \frac{1}{\Delta t} \Delta c(A, t + \Delta t) = kc(A, t) \end{aligned}$$

Because of the uniformity of the time steps  $\Delta t$  can be set to 1:

$$\begin{aligned} \Delta c(s, t + 1) &= \frac{1}{p} \Delta c(P, t + 1) = \\ &= -\frac{1}{a} \Delta c(A, t + 1) = kc(A, t) \end{aligned}$$

Therefore the recurrence relation for the new concentration of a species  $S$  at  $t + 1$  in a first-order reaction can be calculated with:

$$\begin{aligned} c(s, t + 1) &= \\ &= \begin{cases} c(s, t) + k \cdot u(s) \cdot c(A, t) & \text{for products,} \\ c(s, t) - k \cdot u(s) \cdot c(A, t) & \text{for substrates.} \end{cases} \end{aligned}$$

In this equation  $u(s)$  denotes the stoichiometric coefficient in the reaction that is to be simulated. The equation can be translated to a value rule in CGA notation, for which only the change is relevant:

$$\begin{aligned} \omega_{f_c}^{Kin}(s_i) &= \\ &= \begin{cases} f_k(s_i) \cdot u(s_i) \cdot f_c(s_A) & \text{for products,} \\ -f_k(s_i) \cdot u(s_i) \cdot f_c(s_A) & \text{for substrates.} \end{cases} \end{aligned}$$

Note that the rate of change is always calculated in dependence of the substrate concentration of the reaction  $s_A$ , since if there is no actual substrate no reaction can occur. The rate coefficient  $k$  from literature or databases needs to be scaled to take the actual time step into account, that is to be applied in the concrete CGA with  $\bar{k} = f_k(s) = k(s) \cdot \Delta t$ .