

Comparison of discrete and continuous time implementations of a simple spatially explicit individual-based model

Nabil Mabrouk, Guillaume Deffuant

► **To cite this version:**

Nabil Mabrouk, Guillaume Deffuant. Comparison of discrete and continuous time implementations of a simple spatially explicit individual-based model. 2011. <hal-00604341>

HAL Id: hal-00604341

<https://hal.archives-ouvertes.fr/hal-00604341>

Submitted on 30 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparison of discrete and continuous time implementations of a simple spatially explicit individual-based model

N. Mabrouk and G. Deffuant

June 28, 2011

Abstract

We propose to compare discrete and continuous time implementations of a simple birth-death individual-based model (IBM) to analytically derived expression of the mean and variance of the population size. Our results show that the discrete time implementation underestimate the variance, especially when increasing the size of the timestep.

1 Introduction

Over the last two decades, the individual-based approach has been established as an interesting technique for modelling ecological systems [9][6][10][5][3] [8]. With the increase of desktop computing power, it is quiet feasible today to simulate a population by simulating its constitutive individuals as they move across a landscape, interact, reproduce and die [2]. Such simulations have shown to be a valuable aid in studying many complex ecological systems and in reproducing realistic spatial and temporal patterns observed in real systems [12][14][15][13]. However and before inferring any conclusion from these simulations, the modeler should ensure that the computer code producing these results faithfully translates the formal rules as defined in the individual-based model and that design decision relative to temporal and spatial discretization, events execution order and state updating mode are not introducing simulation artifacts. This step is of critical importance and yet it is one of the most difficult as most IBMs have no reference solution (i.e. analytical solution) against which the simulation results can be checked.

While the use of modern compilers and the adoption of good programming practices (i.e. unit testing) can be very effective to prevent both syntically and non-syntically programming faults, the effect of the design choices made by the modeler to put the individual-based rules into a computer program is rarely assessed during IBM developpment. In this work we are concerned with the artifacts that may result from the temporal representation in individual-based models. Time is represented in IBMs used either a discrete or a continuous models. Most IBMs adopt a discrete time model where the time is subdivided into constant timesteps. The processes are then executed at each timestep based on the state of the system at the previous timestep and the state of

the system is updated synchronously at the end of the timestep [?]. This mode of time representation is very common [10][?] due to the simplicity of its implementation. In the continuous representation of time, the model time no longer proceeds in time steps but from event to event. Each event (i.e. a birth of a new individual or death of an existing one), has an exact execution time and the computer simply executes the next action in a queue that is ordered by time [5]. This way of proceeding is also called dynamic scheduling [5] or discrete event model. Contrary to the discrete time model where the states of all the individuals are updated synchronously at the end of the time step, in the continuous time model the state of an individual is usually updated immediately after the event relative to that individual is executed. Hence, for the continuous time model the updating scheme is said asynchronous.

Artifacts due to the choice of the time model has been addressed by several authors mainly from the perspective of comparing synchronous and asynchronous updating modes as synchronous updating mode is associated with the discrete time model and the asynchronous updating scheme with the continuous time model. The comparison of the synchronous and asynchronous modes has been performed for cellular automaton [16][?] and individual-based models [1]. These comparisons showed the time model mode can influence the simulation results. The synchronous update of the state of the individuals has been criticized for its non-realism as many natural systems typically exhibit asynchronous dynamics [7][?][?]. Additionally the synchronous mode requires from the modeler to decide about the order of execution of the event during a time step. Axtell [?] showed that this may introduce additional artifacts and suggested that the order of execution should be randomized.

In order to assess the effect of time representation on the simulation result of an IBM, we use a simple IBM in which the individuals are subject to birth and death events. The IBM is sufficiently simple to allow analytical development. We propose to compare the IBM simulation results to the analytically developed solution and analyze how time representation and individuals state updating scheme can affect the IBM simulation results. Birth-death IBMs are common in ecology as these two processes are encountered in many ecological systems and are involved in the dynamic of the population size and the spatial distribution of the individuals. In a birth-death IBM, an initial population of individuals evolve in time due to individuals births and deaths. Each individual has proper birth and death probabilities per unit of time. these probabilities can be constant or may vary depending the local environment of the individual. For simplicity we consider only the case of constant density independent birth and death rates.

The paper is organized in three parts: in the first part we provide a detailed description of the IBM and the underlying algorithms representing time as discrete and continuous. In the second part we present the results of the comparison to the analytical solution and the comparison the spatial patterns. The third part is dedicated to the discussion of these results and concluding remarks.

2 Methods

We propose to derive a simple IBM for a (virtual) population of individuals living in a two-dimensional domain and subject to stochastic constant-rate birth and death processes. Such IBMs are valuable in showing the degree of variability that we might expect to observe in practice [?]. In this work, however, we use this IBM to illustrate how the choice of the time model (e.g continuous time versus discrete time) may affect the individual-based simulation results. The simplicity of this IBM allows the derivation of analytical expressions for the mean and variance of the number of individuals $N(t)$ at the instant t . We propose to compare the individual-based simulations results to the analytical results.

2.1 Individual-based model description

We consider a two-dimensional spatial domain containing a population of individuals. The individuals are considered as point particles characterized by their location $x = (x_1, x_2)$ in this plane discretized with a constant space step dx . We suppose that the individuals are subject to stochastic birth and death processes. An individual in location x produces a new individual in location x' with a probability $B(x, x')$ and can be removed from the system with a probability $D(x)$. We suppose that the birth and death rates of the individuals are constant and independent from their location (though we keep the general notation where B and D are functions of the location x). The expression of $B(x, x')$ is given by:

$$B(x, x') = bK \left(\frac{\|x - x'\|}{w_b} \right) \quad (1)$$

where b is the constant density-independent probability per unit of time that an individual gives birth to a new individual. $K \left(\frac{\|x - x'\|}{w_b} \right)$ is a birth kernel describing the dispersion of the newly formed individuals around the location x of the parent individual. The death rate of the individuals is given by:

$$D(x) = d \quad (2)$$

where d is a density-independent death probability per unit of time. Because of the stochasticity of the births and deaths of the individuals each individual-based simulation represent a realisation of a stochastic process. The number of individuals at an instant t is a stochastic variable which mean and variance can be derived analytically and are given by [?]:

$$\text{mean : } M[N(t)] = N_0 e^{(b-d)t} \quad (3)$$

$$\text{variance : } V[N(t)] = 2N_0 b t \quad (4)$$

Note that for the particular case where $b = d$, the average population size remains constant $M[N(t)] = N_0$ while the population size grows exponentially for $b > d$ and goes to extinction for $b < d$. In this work we assume that $b = d$.

2.2 Continuous time implementation

In the continuous time model, the time evolution of the system is defined by calculating the waiting time to the next event to occur. The time scale is then progressed with calculating the waiting time which is a real value. Gillespie [4] proposed a Monte Carlo procedure for implementing such time model. the procedure iterates over the following steps:

1. Set time to $t = 0$
2. Calculate the sums $r_b = \sum_{i=1}^{N(t)} b_i = bN(t)$ and $r_d = \sum_{i=1}^{N(t)} d_i = dN(t)$. The total rate at which events occur (births or deaths) is given by $r(t) = r_b(t) + r_d(t)$
3. Choose a waiting time τ for the next event to occur according to $\tau = -\frac{1}{r} \ln \lambda$, where $0 < \lambda \leq 1$ is a uniformly distributed random number
4. Choose a birth or death event with a probability r_b/r and r_d/r respectively
5. Choose an individual k with a probability b_k/r_b (if the event is a birth) or d_k/r_d if the event is a death, where $b_k = b$ and $d_k = d$ are respectively the birth and death rates of the individual k
6. Perform the selected event on the individual k
7. Update time according to $t = t + \tau$
8. Update the number of the individuals $N(t)$
9. Continue from step 2 until $t < t_{end}$

One of the advantages of the continuous time formulation is that designing actions concerning the order of execution of the individuals and the updating mode of the state of the system are implicitly solved in this algorithm. The events are scheduled in a randomized order and the update of the states of the individuals is asynchronous.

2.3 Discrete time implementation

The discrete time model can be implemented in different ways all having in common that time is subdivided into discrete time steps (not necessary with a constant size, but for simplicity we assume that $\Delta t = constant$). The different implementations differ in the order of execution of the different processes (birth, death and update of the individual states). Figure 1 shows 4 examples with different orders of execution of the processes. For each process we iterate over the individuals according to the following generic algorithm :

1. Set time to $t = 0$
2. iterate over the individuals and execute process 1
3. iterate over the individuals and execute process 2
4. ..

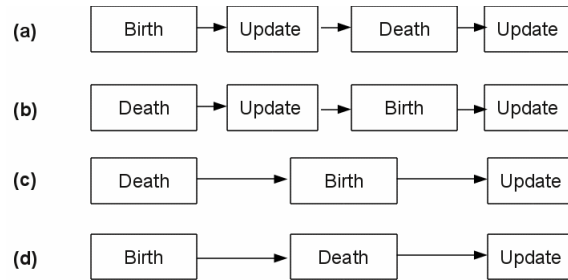


Figure 1: Discrete time model: different scheduling possibilities

5. Update time according to $t = t + \Delta t$
6. Continue from step 2 until $t < t_{end}$

One of the difficulties that raises the discrete time model is that the order of execution of the processes can have an impact on the simulated results. This point will be illustrated by comparing the 4 configurations presented in figure 1.

3 Simulation results

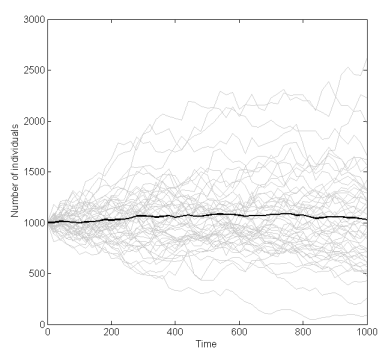
We compare the IBM simulated results using the continuous and the discrete time representation to the analytical solution provided by equations 3 and 4. We proceed by replicating each individual-based simulation a number of times and then calculate the average and variance of $N(t)$. The number of replicates is determined experimentally by testing 50, 150 and 250 replicates. It is expected that, in the absence of artifacts the increase of the number of replicates should yield to better matching between the analytical and the simulated results.

3.1 Comparison of the continuous time model and the analytical solution

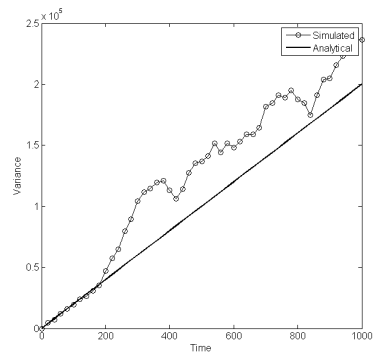
Figure 2 compares the continuous time IBM simulation results to the analytical solution for 50, 150 and 250 replicates of the IBM simulation. The results show that with the increase of the number of replicates the continuous time model converges successfully to the analytical solution.

3.2 Comparison of the discrete time model and the analytical solution

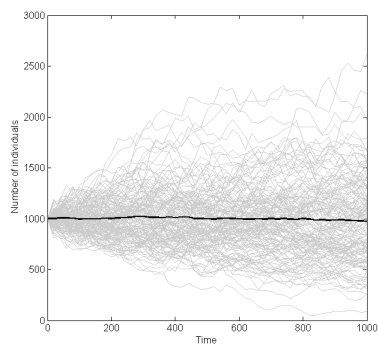
Before comparing the discrete time IBM results to the analytical equations we assess whether the order of execution of the processes has an impact on the simulated results. Figure 3 shows the average number of individuals calculated over 250 runs of the discrete time IBM using the configuration (a) and (c) (see figure ??). The results clearly show that the order of execution of the processes can have an important impact on the simulated results. When death process is executed first the IBM predicts the rapid extinction of the population. This



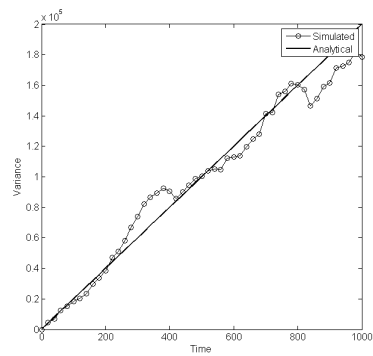
(a) *runs* = 50



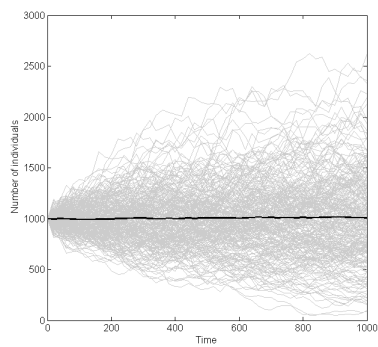
(b) *runs* = 50



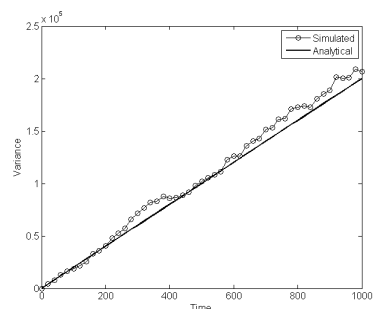
(c) *runs* = 150



(d) *runs* = 150



(e) *runs* = 250



(f) *runs* = 250

Figure 2: Continuous time simulation. Number of individuals and variance over 50, 150 and 250 of the individual-based model

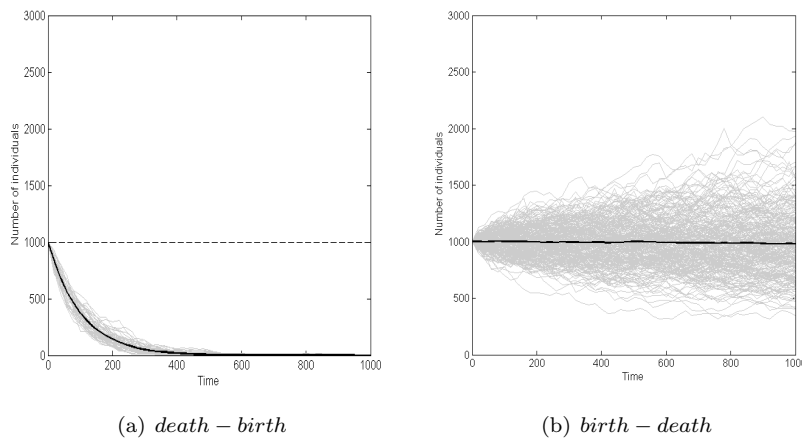


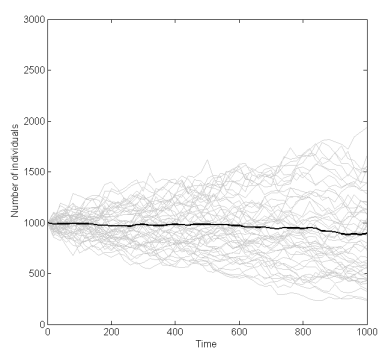
Figure 3: Number of individuals versus time simulated using two algorithms of the discrete time implementation. (a) death events are executed first, (b) birth events are executed first

results is due to a bias induced by the order of execution of the birth, death and updating process. The individuals that dies are removed from the system and has no chance to give birth to new individuals during the time step Δt . Only the remaining individuals can give birth, but as their number is reduced by the number of deaths, the number of birth does not compensate over the long run the number of deaths which yields to the extinction of the population. This artifact is corrected by executing the birth and death processes and storing the changes in a temporary list of individuals. the state of the system is updated only after all the individuals have been tested for birth and for death (configurations (c) and (d) in figure ??).

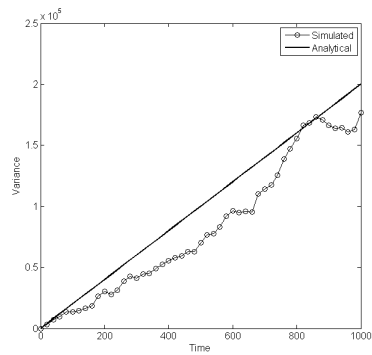
We proceed to the comparison of the discrete time IBM simulation (configuration (c)) to the analytical expression of the mean and variance of the population size. The simulations are replicated 50, 150 and 250 times and the results are reported in figure 4. While the mean number of individuals convergence rapidly to the analytical solution, the variance calculated by the individual-based simulation seems to be slightly lower than the analytical solution. This result is confirmed by taking a larger time steps ($\Delta t = 5$) as can be viewed from figure 5. Hence it seems that a large timestep induces an underestimation of the variance but has no effect (for this simple IBM) on the mean population size.

3.3 Comparison of the spatial patterns simulated with continuous and discrete time implementations

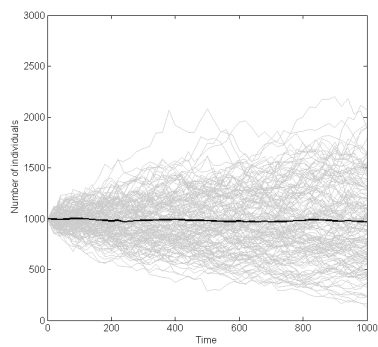
We assessed the effect of the time model on the spatial patterns simulated using the individual-based model. We use the radial pair correlation to characterise the spatial pattern. The spatial pattern in this simple birth death depend to a large extent on their properties of the birth kernel. If the size of the birth kernel is small in comparison to the size of the domain, than the newborn individuals



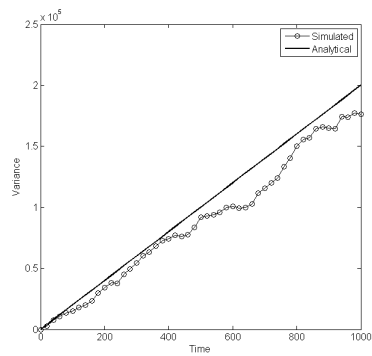
(a) *runs* = 50



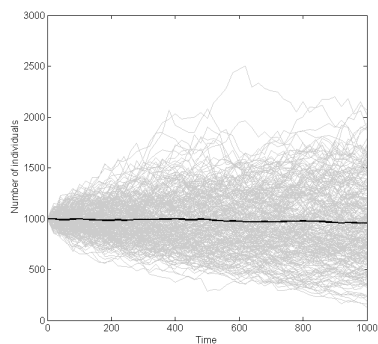
(b) *runs* = 50



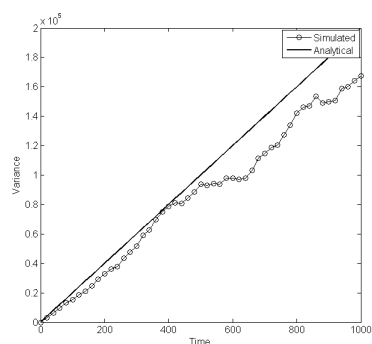
(c) *runs* = 150



(d) *runs* = 150



(e) *runs* = 250



(f) *runs* = 250

Figure 4: Continuous time simulation. Number of individuals and variance over 50, 150 and 250 replicates

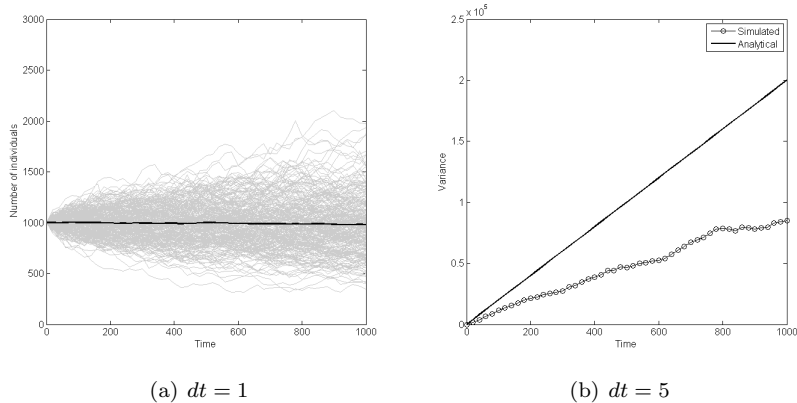


Figure 5: Variance of the simulated number of individuals (over 250 replicates) for two different time steps of the discrete-time algorithm

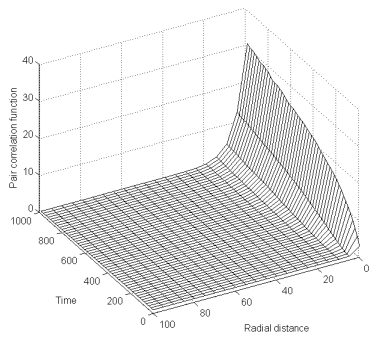
tend to accumulate around the parent individuals yielding the formation of small groups of individuals. This pattern tend towards a uniform distribution of the individuals as the size of the birth kernel is increased. Such patterns can be well characterized using pair correlation functions which measures the average density of individual at different distance from a focal individuals. The pair-correlation function is normalized such that a value of 1 is obtained for a uniform distribution. Value higher than 1 at a distance r means that the average number of neighbors located at a distance r is higher than what would be obtained if the individuals are distributed uniformly.

As for the variance, the discrete time model tends to underestimate the aggregation of the individuals measured with the pair correlation function (figure 6). The increase of the time step reinforces this tendency confirming that the pattern is sensitive to the choice of the timestep in the discrete time IBM.

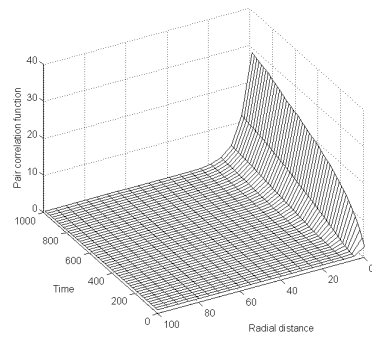
4 Discussion

While a great effort has been devoted during the last years towards the development of improved software libraries for the implementation of IBMs [11] [17][?] many researcher agree that there is an increasing need to develop a good practice that guide the IBM developer during all the phases of the model construction and implementation. Often IBM developer are faced with several design decisions relative to space and time representations (discrete versus continuous), events scheduling and individuals state updating processes (synchronous versus asynchronous). These decisions can in many situations induce artifacts that, due to the IBM complexity and the absence of a reference solution, are difficult to assess.

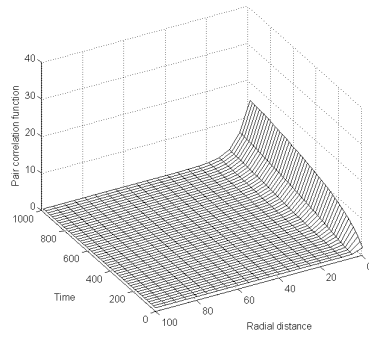
Time representation is a recurrent design decision in individual-based models. Our work shows that the choice of a discrete time model can introduce artifacts, mainly by underestimating the variance. The choice of the discrete time rep-



(a) Continuous time



(b) $dt = 1$



(c) $dt = 5$

Figure 6: Pair correlation function

resentation requires also from the modeler to specify the order in which the processes are executed. This step can introduce additional artifacts. We show that by executing death events systematically before birth events the simulated population size vanishes to zero. Similar artifacts have been reported in [?].

While the discrete time model is easier to implement and in many situations can reduce simulation time, much care should be considered in the choice of the time step in particular for IBMs where different time scales are considered. In many situations the computational advantage of the discrete time model is lost if the time step is set considering the rapid processes.

Acknowledgement

We thank Claude Lobry, Alain Rapaport and Fabien Campillo for their helpful suggestions. This work is supported by the SYCOM program, a grant from the French National Agency of Research (DISCO, program ANR)

References

- [1] Geoffrey Caron-Lormier, Roger W. Humphry, David A. Bohan, Cathy Hawes, and Pernille Thorbek. Asynchronous and synchronous updating in individual-based models. *Ecological Modelling*, 212(3-4):522–527, April 2008.
- [2] U. Dieckmann, R. Law, and J.A.J Metz. *The geometry of ecological interactions: simplifying spatial complexity*. Cambridge University Press, 2000.
- [3] J. Ferrer, C. Prats, and D. Lopez. Individual-based modelling: An essential tool for microbiology. *J Biol Phys*, 34:19–37, 2008.
- [4] T. Gillespie, D. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 1976.
- [5] Volker Grimm and Steven F. Railsback. *Individual-based Modeling and Ecology*. Princeton University Press, 2005.
- [6] Volker Grimm, Tomasz Wyszomirski, David Aikman, and Janusz Uchmanski. Individual-based modelling and ecological theory: synthesis of a workshop. *Ecological Modelling*, 115(2-3):275–282, February 1999.
- [7] Zaiyi Guo and Joc Cing Tay. Multi-timescale event-scheduling in multi-agent immune simulation models. *Biosystems*, 91(1):126–145, January 2008.
- [8] F.L. Hellweger and V. Bucci. A bunch of tiny individuals - individual-based modeling for microbes. *Ecological Modelling*, 220:8–22, 2009.
- [9] O.P. Judson. The rise of the individual-based model in ecology. *Trends in Ecology and Evolution*, 9(1):9–14, 1994.

- [10] Jan-Ulrich Kreft, Cristian Picioreanu, Julian W.T. Wimpenny, and Marck C.M. van Loosdrecht. Individual-based modelling of biofilms. *Microbiology*, 147:2897–2912, 2001.
- [11] S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan. Mason : A java multi-agent simulation toolkit. *Proceedings of the 2004 SwarmFest Workshop*, 2004.
- [12] C. Picioreanu, D.J. Batstone, and M.C.M. Van Loosdrecht. Multidimensional modelling of anaerobic granules. *Water Science and Technology*, 53:501–507, 2005.
- [13] W.C. Pitt, P.W. Bow, and F.F. Knowlton. An individual-based model of canid populations: modelling territoriality and social structure. *Ecological Modelling*, 166(1-2):109–121, 2003.
- [14] S.F. Railsback and B.C. Harvey. Analysis of habitat selection rules using an individual-based model. *Ecology*, 83:1817–1830, 2002.
- [15] Timothy D Scheibe and Marshall C Richmond. Fish individual-based numerical simulator (fins): a particle-based model of juvenile salmonid movement and dissolved gas exposure history in the columbia river basin. *Ecological Modelling*, 147(3):233–252, January 2002.
- [16] B. Schoenfish and A. de Roos. Synchronous and asynchronous updating in cellular automata. *BioSystems*, 51:123–143, 1999.
- [17] Joao B. Xavier, Cristian Picioreanu, and Mark C. M. van Loosdrecht. A framework for multidimensional modelling of activity and structure of multispecies biofilms. *Environmental Microbiology*, 7:1085–1103, 2005.