

# Coexistence-via-Chemical-Interactions: Language and Design Choices, Verification, and new Results

---

## Authors:

- Darius Russell Kish<sup>1,2\*</sup>, [russeldk@bc.edu](mailto:russeldk@bc.edu)
- Yuezhen Chen<sup>1</sup>
- Jessica Fong Ng<sup>1</sup>
- Matthew Uy<sup>1</sup>

(\*) indicates corresponding author

(1) Department of Computer Science, Boston College

(2) Department of Biology, Boston College

## Abstract

## Introduction

Mathematical modeling of population dynamics is ubiquitous in the sciences. From their applications in the presently relevant field of epidemics to population genetics, chemical kinetics, economics, and systems biology, discrete time-based simulations are used to solve systems of differential equations with no analytical solution. Often these systems are highly dependent on their input parameters and stochastic processes, so multiple replicates of the simulation are used to provide more reliable statistics of the models than one iteration provides.

Discrete time-based simulations are however costly due to difficulties in parallelizing the underlying simulation. While techniques for parallelizing along simulation time have arisen, they are not easily generalized to all problems of this type. The replicates are, however, independent simulations and are a prime example of an Embarrassingly Parallel problem. This allows for linear speedups by number of processors in nearly all cases.

Many of these simulation models are designed and written by scientists without a programming background, and thus suffer from inefficiencies surrounding memory allocation, inefficient re-writing of language-implemented algorithms, avoidance of language-specific features and parallel libraries. For example, many modern languages allow for complex array indexing and operations along its dimensions, which might be overlooked for their conceptually easier but bloated explicit implementations. Design for memory efficiency may be overlooked due to language-specifics and complications surrounding the underlying operations masked by high level syntax. Parallelization is often times overlooked due to its complicated nature at the language-level, especially surrounding

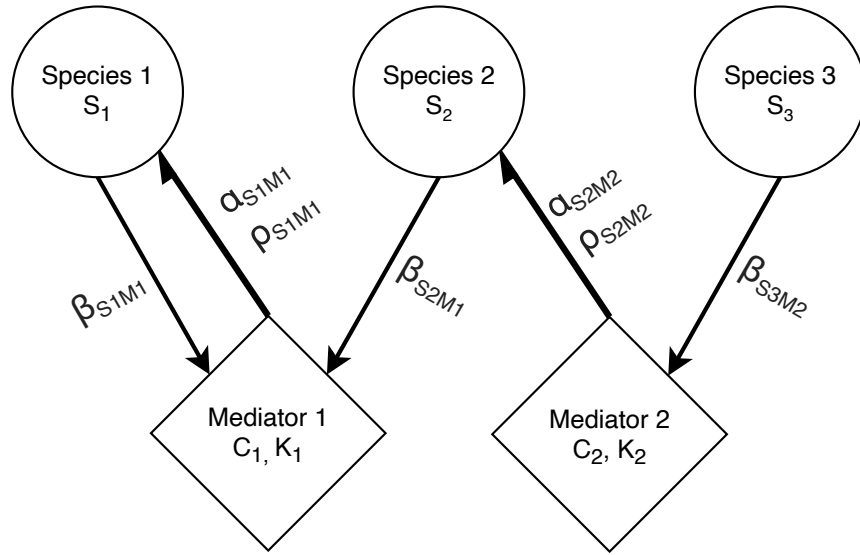
Random Number Generators (RNGs). These language features are costly to learn for researchers focused on the conceptual design of these simulations and their results. Thus, many simulation projects that do not employ dedicated software engineers suffer from sub-optimal performance.

We chose a simulation in the systems biology category from Professor Babak Momeni's lab at Boston College. Its mathematical derivation can be found in *Momeni et al., 2017* and its implementation in Matlab and characterization in *Niehaus et al., 2019*, though we will provide a summary of its background, design, and characteristics here.

Microbial communities naively consist of microbes, small single-celled organisms that can form colonies consisting of cells from the same organism clustered together. While they are widely depicted as isolated colonies on petri dishes, these microbes do not always exist isolated in nature. Communities of microbial species have been found and characterized, sometimes displaying functionality that is not present in its basal components. Understanding the dynamics of microbial communities is vital to harnessing their functionality.

At a very high level, the Well-Mixed model represents a completely homogenous community. All cells can access all mediators in the same concentrations. This is representative of cultures grown in liquid media, which are generally shaken during incubation to ensure constant mixing. This simplifies the model greatly to exclude spatial distribution of species and mediators, needing diffusion constants for both in addition to the inclusion of boundary conditions. Choosing a Well-Mixed approach allows this model to be represented and simulated simply.

The model can be represented as a dynamic graph with two classifications of Nodes, and two classifications of Edges. There are two node types, a species node and a mediator node. The species node has an attribute that tracks the number of cells of the species in the simulation,  $S$ . A mediator is some non-species component that can be produced or consumed by species. When a species consumes a mediator, it induces either a positive or negative effect on the fitness of the species. A mediator node has an attribute that tracks its concentration,  $C$ , and  $K$  corresponding to its saturation level<sup>1</sup>. The two edge types thus represent consumption and production. The production edge has an attribute corresponding to the amount of mediator produced by that species per time,  $\beta$ . A consumption edge has two attributes, the consumption rate,  $\alpha$ , and its fitness effect,  $\rho$ .



**Figure 1:** An example graphical representation of a model. There are three species and two mediators. There are 3 production edges with corresponding  $\beta$  values, and two consumption edges with corresponding  $\alpha$  and  $\rho$  values. Edges are shown with direction to show the flow of mediator, though the actual network topology is undirected.

These communities can be modeled using relatively simple rules and parameters. In its most simple form, the model is driven by two differential equations:

1.  $\frac{dS_i}{dt} = [r_{i0} + \sum_l (\rho_{il}^{pos} \frac{C_l}{C_l + K_{il}} - \rho_{il}^{neg} \frac{C_l}{K_{il}})] S_i$
2.  $\frac{dC_l}{dt} = \sum_i (\beta_{li} S_i - \alpha_{li} \frac{C_l}{C_l + K_{il}} S_i)$

The full model takes into account dilutions and generations, claiming stability is often reached by 200 generations of growth, however these are not represented in the basic equations. Once the total sum of all cells is above the dilution threshold (in real life generally measured using optical density), all cells are diluted proportionally back to an initial value. Species who possessed a greater fraction of total cells before dilution possess the same fraction after dilution. This is done to simulate real dilutions necessary to avoid lag phase and cell death seen in experimental incubation. Additionally, the formula  $\log_2(\frac{dilTh}{nInitCells})$  is used to calculate the number of generations grown in one round of propagation, where  $dilTh$  is the dilution threshold in number of cells, and  $nInitCells$  is the number of initial cells, in number of cells. An additional maximum time is imposed in the numerical implementation of 250 hours. If the dilution threshold is not reached by 250 hours, then it is automatically diluted and a new round of propagation is started.

---

1. This is perhaps the most challenging aspect of this model to understand for non-biologists. For the case of inhibition, the model uses the formula:  $r(C_{inh}) = r_0 - r_{inh} \frac{C_{inh}}{K_{inh}}$ , where  $K_{inh}$  is the corresponding  $k$  value in  $K$ . The amplitude of effect on growth rate is controlled by  $K$  for inhibition. For positive interactions, or facilitation, the model uses  $r(C_{fac}) = r_0 + r_{fac} \frac{C_{fac}}{C_{fac} + K_{inh}}$ , a form of the Monod equation. Here,  $k$  determines at what concentration of  $C$  that  $\frac{r_{fac}}{2}$  will be reached, and  $r_{fac}$  is the saturated effect on the growth constant by the mediator. These equations are determined from experimental data studying growth curves. For more information see *Merchuk and Asenjo, 1995* and *Konak, 1974*. [↩](#)