12-14-2015

A Survey of the Kinetic Monte Carlo Algorithm as Applied to a Multicellular System

Michael Richard Laughlin
University of South Carolina - Columbia

A Survey of the Kinetic Monte Carlo Algorithm as Applied to a Multicellular System

by

Michael Richard Laughlin

Bachelor of Science
Auburn University 2012

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Arts in

Mathematics

College of Arts and Sciences

University of South Carolina

2015

Accepted by:

Yi Sun, Director of Thesis

Peter Binev, Reader

Lacy Ford, Senior Vice Provost and Dean of Graduate Studies

Acknowledgments

For my parents, who raised me to be the best I could be, and for all my family and friends that supported me in my darkest times and encouraged me to follow my passions.

Abstract

We explore the origins and implementation of the Kinetic Monte Carlo method on a system of cells suspended in a liquid media. The situation presented herein has applications in the emerging field of biofabrication, which may have lasting impacts in medical science. The theory behind the method is explained in detail, starting with its emergence in the 1960s, and two major improvements to the scaling of the method are presented, along with a restriction to a special case. Finally, we give the results of several simulations.

Table of Contents

List of Figures

Chapter 1

Background

In biology, morphogenesis refers to the process by which an organism achieves its form in embryonic development. For multicellular organisms, morphogenesis determines the types of cells which will comprise the fully formed organism, where those cells are located within the body, the structures those cells will form, and their relative sizes. Components of this process include organogenesis, the formation of organs, and histogenesis, the process by which cells differentiate themselves.

One of the prevailing means of explaining how morphogenesis comes to pass is through the idea of cellular self-organization, forming highly ordered systems of cells out of predominantly disordered collections. This phenomenon is not unique to biological systems. Examples in chemistry have been well-studied and applied in fields such as nanotechnology [10]. Three biological self-organizational processes are described in *Cytosystems dynamics in self-organization of tissue architecture*, by Yoshiki Sasai [6]. The second process mentioned therein is self-patterning, whereby cells spontaneously form localized structures of homogeneous cell type, and the third is self-driven morphogenesis, where localized control over the mechanics of the cells themselves drive the formation of new structures. Our study will focus on the first process, cellular self-assembly, which is the automatic striation of cells into a patterned structure. Mechanisms for all of the aformentioned self-organizational processes exist in Sasai's paper, for the curious reader.

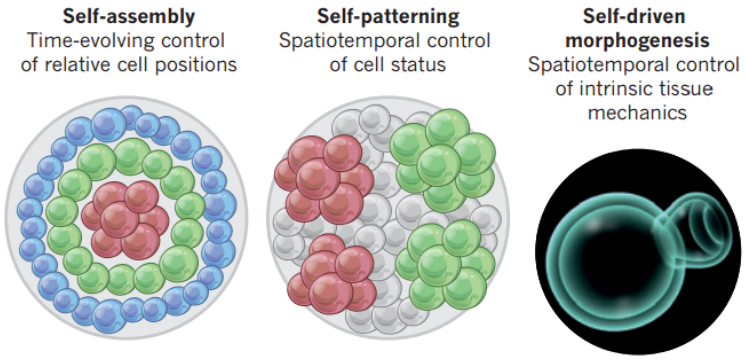


Figure 1.1 Yoshiki Sasai, Cytosystems dynamics in self-organization of tissue architecture, *Nature*, 493(7432):318-326, 2013/01/17, Figure 1.

For cellular self-assembly to occur, there must exist forces between cells driving their movement through the extracellular matrix, and they must have enough motility to move past one another semi-freely. The Differential Adhesion Hypothesis, proposed by Malcolm Steinberg in 1963 [8], states that cells may be treated similarly to molecules of a viscoelastic liquid, forming bonds with one another, but ultimately free to move past each other given enough force acting on them. We expect that the resultant configuration will be a striated pattern in which the cells with the least surface adhesion capabilities will engulf those with stronger adhesions, minimizing the total energy in the system. These adhesion abilities depend on energy differences between the cell types, where similar energies allow for stronger intercellular bonding. This idea is confirmed, for instance, in the structure of our own organs, which are layered with different cell types.

It is the objective to apply these ideas in the field of biofabrication, in which collections of cells will give rise to a desireable morphological structure, say an organ ready for transplant. A development dating back to 2006 [1], bioprinting involves the placing of cellular aggregates layer by layer onto a scaffold in a manner similar to that of a 3D printer. The placed aggregates are balls consisting of various cell types and hydrogels, and after successful deposition, the structure is allowed to proceed

with organogenesis unhindered in a liquid medium. It is of interest to the researcher, therefore, to have an idea how the system will alter in time.

In order to accurately model how the Differential Adhesion Hypothesis will disturb the initial placing of cells, we will use the Kinetic Monte Carlo method. For simplicity, the model will assume that the cells are placed on a cubic grid, that each cell neither dies nor reproduces, and that each cell has interactions only with the cells immediately surrounding it. For every time step, each cell will have a determinable probability of swapping positions with one of its neighbor cells. The striation will depend on the surface tension parameters of each of the involved cell types and the hydrogels, and KMC will account for this in the form of differing swapping probabilities of placed cells, based on their own type and that of their neighbors.

In Chapter 2, we will discuss the history of the Kinetic Monte Carlo method, and we will formulate the algorithm as it will be applied to the multicellular systems thus far discussed. In Chapter 3, the manner in which KMC immediately applies to this particular problem will be made more explicit, and we will present some results of simulations.

Chapter 2

Development of Kinetic Monte Carlo

2.1 Introduction

Kinetic Monte Carlo is an algorithm used to simulate the evolution of a discrete network of objects that can interact with one another in any of a finite number of probabilistically set ways. Though not the originator of the variable-time Monte Carlo simulation, Daniel T. Gillespie provided an in-depth exploration of the topic in 1977 [5]. In his paper, Gillespie stochastically described the evolution of systems of chemicals. It is within this context that we will survey the development of Kinetic Monte Carlo.

The first major improvement to the Gillespie algorithm we will discuss was published by Michael A. Gibson and Jehoshua Bruck in 2000 [4]. By classifying the potential reactions in the network according to an "indexed priority queue", Gibson and Bruck were able to improve the efficiency of the algorithm from linear to logarithmic time. Further improvements were made less than a decade later, by Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton, in 2008 [7], through a reaction propensity grouping procedure, the computation time was further reduced to constant-time.

However, the development of KMC discussed in Sections 2.2 through 2.4 is more general than we require, as one of the assumptions we are making on the multicellular system is that the cells occupy positions on a lattice. The lattice structure of the cells simplifies things to an extent; in fact, the method for handling this situation was

published two years prior to Gillespie's paper, by Bortz, Kalos, and Lebowitz [2]. In Section 2.5, we will explore how KMC applies to the equivalent case of an Ising spin system.

2.2 Gillespie KMC

Suppose we have a chemical system having a set $X = \{X^{(i)}\}_{i=1}^{M}$ denoting the number of molecules of possible chemicals and a set $R = \{R^{(i)}\}_{i=1}^{N}$ of possible reactions of the form

$$\sum_{i=1}^{r_n} R_i^{(n)} \to \sum_{i=1}^{p_n} P_i^{(n)}, \tag{2.1}$$

where $\{R_1^{(n)}, R_2^{(n)}, \ldots, R_{r_n}^{(n)}\}$ are the reactants in the reaction $R^{(n)}$, and $\{P_1^{(n)}, P_2^{(n)}, \ldots, P_{p_n}^{(n)}\}$ are the products. Note that not all the chemicals that can possibly exist in the system are necessarily present at the time of initialization; however, it must be assumed that we may determine all possible reactants and products in the system we seek to describe.

In order to simplify the proceedings, we make the assumption that a given reaction *will* occur if all the reactant molecules collide.

Proposition 2.1. *For any two-molecule reaction $R^{(n)}$, there exists a constant C_n, depending on the total volume of the system, the temperature (the velocity of the molecules), and the radii of all the involved molecules, such that the probability of a $R^{(n)}$ reaction occurring within the next small time interval δt is $C_n \delta t$.*

Proof. Without loss of generality, suppose that the reaction label is $R^{(1)}$. For now, also assume that there is exactly one copy of each reactant in the system. Suppose that the reactant m_1 - of type $R_1^{(1)}$ - has radius r_1 and that the reactant m_2 - of type $R_2^{(1)}$ - has radius r_2, then a collision will occur if the distance between the molecule centers is less than or equal to $r_1 + r_2$.

Let v be the velocity of m_1 relative to m_2. Since a collision is equivalent to the edge of m_2 being within the cylindrical, semisphere-capped path traced by m_1 over the next time interval, it suffices to find the probability of a randomly placed point being within the space traced by a sphere of radius $r_1 + r_2$ in the system. If the volume of the whole system is V, then the probability is therefore

$$C_1 \delta t = \frac{\pi (r_1 + r_2)^2 v \delta t}{V}. \tag{2.2}$$

Now, suppose that there are X_1 copies of reactant $R_1^{(1)}$ and X_2 copies of reactant $R_2^{(1)}$. If the average probability of some fixed pair of these molecules reacting over the next time interval is $K_1 \delta t$, then the total probability of an instance of reaction $R^{(1)}$ occurring in the next time interval is $C_1 = X_1 X_2 K_1 \delta t$. If $R_1^{(1)} = R_2^{(1)}$, and there are X copies of the reactant in the space, then we find the probability of an $R^{(1)}$ reaction to be $C_1 = X(X-1) K_1 \delta t$. □

Of course, there is nothing unique about the aforementioned situations, and we may find such probabilities for all potential reactions in the space in a similar manner.

Theorem 2.2. *For each reaction $R^{(n)}$, there exists a constant C_n, depending on the total volume of the system, the temperature, and the radii of all the involved molecules, such that the probability of a $R^{(n)}$ reaction occurring within the next small time interval δt is $C_n \delta t$.*

Given that we have the means to find all these constants $\{C_i\}_{i=1}^N$, define for each reaction $R^{(n)}$ a time-dependent and molecular-distribution-dependent probability function

$$P_t(\tau, n) \delta t, \tag{2.3}$$

which yields the probability that, given current time t and current molecular distribution $\{X^{(i)}\}_{i=1}^M$, the next reaction in the system will be of type $R^{(n)}$ and will happen

within the time interval $(t+\tau, t+\tau+\delta t)$. Additionally, we define

$$P_t(\tau, \varnothing), \tag{2.4}$$

to be the probability that, given current time t and current molecular distribution $\{X^{(i)}\}_{i=1}^{M}$, no reaction occurs within the time interval $(t, t+\tau]$.

Using these definitions, we note that $P_t(\tau, n)\delta t$ may be represented as the product of the probability that no reaction occurs in the first τ time units and the probability that a $R^{(n)}$ reaction occurs within δt time units:

$$P_t(\tau, n)\delta t = P_t(\tau, \varnothing)C_n\delta t. \tag{2.5}$$

Since $P_t(\tau, n)\delta t$ is expressible in this way, we will seek a more easily usable expression for it by way of $P_t(\tau, \varnothing)$.

Theorem 2.3. $P_t(\tau, n)\delta t = C_n\delta t \exp(-\tau \sum_n C_n)$

Proof. Consider the probability that no reaction occurs within the time interval $(t, t+\tau+\delta t]$, and observe that this is equal to the probability that no reaction occurs in $(t, t+\tau]$ times the probability that no reaction occurs in $(t+\tau, t+\tau+\delta t]$:

$$P_t(\tau+\delta t, \varnothing) = P_t(\tau, \varnothing)P_{t+\tau}(\delta t, \varnothing) \tag{2.6}$$

Now, note the probability that no reaction will occur within the next δt time units is the product of the probabilities that no $R^{(n)}$ reaction will occur for each n.

$$\begin{aligned} P_{t+\tau}(\delta t, \varnothing) &= \prod_n (1 - C_n\delta t) \\ &= 1 - \sum_n C_n\delta t + O\left((\delta t)^2\right) \end{aligned} \tag{2.7}$$

Hence,

$$\begin{aligned} P_t(\tau+\delta t, \varnothing) &= P_t(\tau, \varnothing)\left(1 - \sum_n C_n\delta t + O\left((\delta t)^2\right)\right) \\ &= P_t(\tau, \varnothing) + P_t(\tau, \varnothing)\delta t\left(-\sum_n C_n + O\left(\delta t\right)\right). \end{aligned} \tag{2.8}$$

Therefore,

$$\frac{P_t(\tau+\delta t,\varnothing)-P_t(\tau,\varnothing)}{\delta t}=P_t(\tau,\varnothing)\left(-\sum_n C_n+O\left(\delta t\right)\right). \tag{2.9}$$

Taking the limit as δt approaches zero,

$$P_t'(\tau,\varnothing)=-P_t(\tau,\varnothing)\sum_n C_n. \tag{2.10}$$

Finally, integrating with respect to τ, we obtain

$$P_t(\tau,\varnothing)=\exp(-\tau\sum_n C_n). \tag{2.11}$$

Hence, the result holds. □

Now that we have a general expression for the probability of a given reaction in hand, we will devise the scheme by which we will simulate the evolution of the chemical system. The Monte Carlo method will be the framework, but in the interest of efficiency, it would be ideal if there were no wasted steps in the algorithm. Due to this consideration, we seek time intervals that will dynamically change so that precisely one reaction will take place per time step. To accomplish this, our time intervals and the reactions that will be conducted will both be generated according to the probability distribution that has been constructed.

We precondition the probability function as

$$P_t(\tau,n)\delta t=P_t^1(\tau)\delta t\cdot P_t^2(n|\tau), \tag{2.12}$$

i.e. the probability that the next reaction is of type $R^{(n)}$ and takes place in the time interval $(t+\tau,t+\tau+\delta t)$ is equal to the probability that the next reaction of any type takes place in the interval $(t+\tau,t+\tau+\delta t)$ times the probability that the reaction that takes place is of type $R^{(n)}$, given that the reaction is in the interval.

We can find $P_t^1(\tau)\delta t$ easily enough, as this will be the sum of the probabilities for each individual reaction taking place in the interval

$$P_t^1(\tau)\delta t=\sum_n P_t(\tau,n)\delta t. \tag{2.13}$$

Then, we find $P_t^2(n|\tau)$ by combining (2.12) and (2.13)

$$\begin{aligned} P_t^2(n|\tau) &= \frac{P_t(\tau, n)\delta t}{P_t^1(\tau)\delta t} \\ &= \frac{P_t(\tau, n)\delta t}{\sum_n P_t(\tau, n)\delta t}. \end{aligned} \tag{2.14}$$

Definition 2.4. For a fixed probability distribution $P(x)$, we define the cumulative probability $F(x) = \int_{-\infty}^{x} P(\nu)\, d\nu$. Observe that $0 \leq F(x) \leq 1$. We say that a random value y is chosen with respect to $P(x)$ if for a uniformly distributed random number r in $[0, 1]$, we have $y = F^{-1}(r)$. Observe further that this inverse is always uniquely defineable, as $F(x)$ increases uniformly over all of $\mathbb{R}$.

Now, to pick τ and n for use in a time step, we need only pick them according to their respective probability distributions.

Theorem 2.5. *For uniformly distributed random numbers r_1 and r_2 in $[0, 1]$, τ picked with respect to $P_t^1(\tau)$ has the value $\frac{1}{\sum_k C_k} \log\left(\frac{1}{r_1}\right)$ and n picked with respect to $P_t^2(n|\tau)$ has the value of n for which $\sum_{k=1}^{n-1} C_k < (\sum_k C_k)\, r_2 \leq \sum_{k=1}^{n} C_k$, where $\sum_{k=1}^{0} C_k$ is defined to be 0.*

Proof. Let $F_1(\tau)$ be the cumulative probability distribution of $P_t^1(\tau)$ and let $F_2(n)$ be the cumulative probability distribution of $P_t^2(n|\tau)$.

Applying Theorem 2.3, (2.13), and the fact that $P_t^1(\tau) = 0$ for $\tau < 0$ (since P_t^1 is

predictive, but not historical), we have

$$\begin{aligned} F_1(\tau) &= \int_{-\infty}^{\tau} P_t^1(\hat{\tau})\, d\hat{\tau} \\ &= \int_0^{\tau} \sum_k P_t(\hat{\tau}, k)\, d\hat{\tau} \\ &= \int_0^{\tau} \sum_k C_k \exp(-\hat{\tau} \sum_k C_k)\, d\hat{\tau} \\ &= \left(\sum_k C_k\right) \int_0^{\tau} \exp(-\hat{\tau} \sum_k C_k)\, d\hat{\tau} \\ &= -\exp(-\hat{\tau} \sum_k C_k)\Big|_0^{\tau} \\ &= 1 - \exp(-\tau \sum_k C_k) \end{aligned} \tag{2.15}$$

Hence, for uniformly distributed r in $[0, 1]$,

$$\begin{aligned} F_1^{-1}(r) &= \frac{1}{\sum_k C_k} \log\left(\frac{1}{1-r}\right) \\ &= \frac{1}{\sum_k C_k} \log\left(\frac{1}{r_1}\right) \end{aligned} \tag{2.16}$$

for another uniformly distributed random number $r_1 = 1 - r$ in $[0, 1]$.

Now, suppose that r_2 is uniformly distributed in $[0, 1]$. Applying Theorem 2.3 and

(2.14), we have

$$\begin{aligned} F_2(n) &= \int_{-\infty}^{n} P_t^2(\hat{n}|\tau)\, d\hat{n} \\ &= \int_{-\infty}^{n} \frac{P_t(\tau,\hat{n})}{\sum_k P_t(\tau,k)}\, d\hat{n} \\ &= \int_{-\infty}^{n} \frac{C_{\hat{n}} \exp(-\tau \sum_k C_k)}{\sum_k C_k \exp(-\tau \sum_k C_k)}\, d\hat{n} \\ &= \int_{-\infty}^{n} \frac{C_{\hat{n}}}{\sum_k C_k}\, d\hat{n} \\ &= \frac{1}{\sum_k C_k} \int_{-\infty}^{n} C_{\hat{n}}\, d\hat{n} \end{aligned} \tag{2.17}$$

Since the C_ks are a discrete collection, this leads to the conclusion

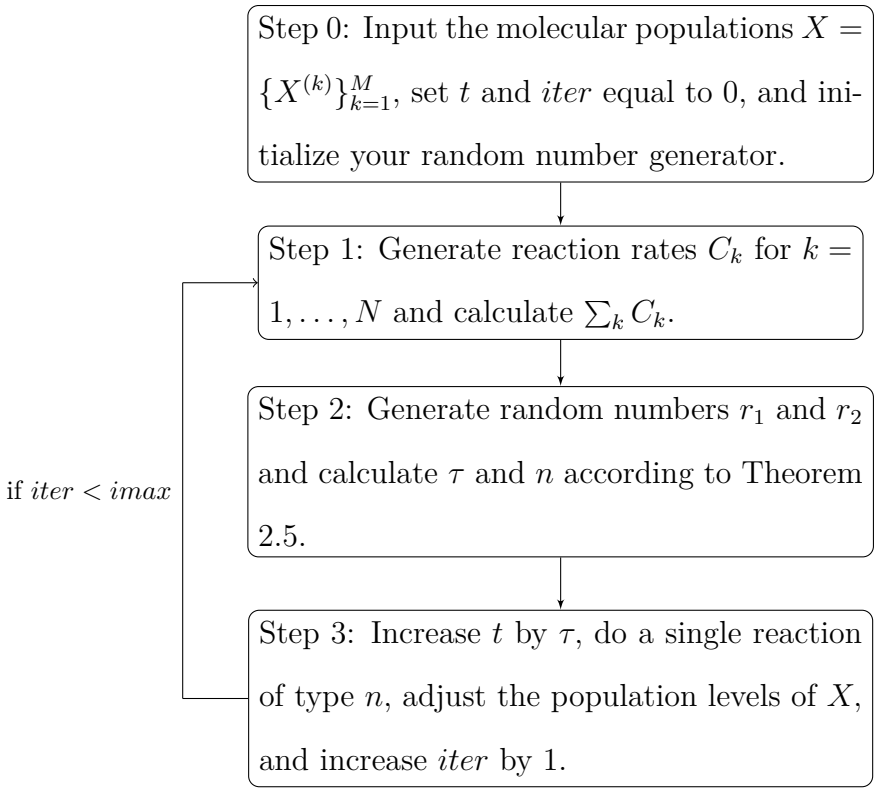
$$F_2(n) = \frac{1}{\sum_k C_k} \sum_{k=1}^{n} C_k \tag{2.18}$$

Hence, if $F_2^{-1}(r_2) \approx n$, then $r_2 \approx F_2(n) = \frac{1}{\sum_k C_k} \sum_{k=1}^{n} C_k$, and therefore, taking $\sum_{k=1}^{0} C_k = 0$,

$$\frac{1}{\sum_k C_k} \sum_{k=1}^{n-1} C_k < r_2 \leq \frac{1}{\sum_k C_k} \sum_{k=1}^{n} C_k \tag{2.19}$$

as desired. □

Having obtained a method for picking τ and n dynamically according to the current state of the system, we may now proceed in outlining the simulation algorithm, wherein we shall use *iter* to denote the number of iterations of the algorithm we have made, and *imax* to denote the maximum number we wish to conduct.



Proposition 2.6. *The Gillespie simulation algorithm scales in $O(N)$ time.*

Proof. Recall that N is the total number of potential reactions in the system. Since it must be done on an element-by-element basis, computing C_k for each k takes $O(N)$ operations. Similarly, computing the sum $\sum_k C_k$ takes $N - 1 \sim O(N)$ operations.

Calculating τ is $O(1)$ time, since it is just the result of a single random number generation combined with one floating-point operation. On the other hand, computing n takes $O(N)$ time, as we must scan through the list of partial sums to find the first value m for which the sum $\sum_{k=1}^{m} C_k$ exceeds the generated random number r_2 times the total sum. This may happen quickly, or it may not occur until $m = N$.

Actually performing the reaction scales as $O(1)$, since the degree to which the size of molecular populations can be effected by a single reaction is bounded from above by the reaction in $\{R^{(k)}\}_{k=1}^{N}$ having the largest combined number of reactants and products. □

While $O(N)$ computation time isn't ideal, Gillespie's method provided the Monte Carlo algorithm with the significant improvement of removing wasted computation steps. Since Monte Carlo used fixed time steps, some of them would result in no actual change in the system. With the variable-time scheme proposed in Gillespie's research, this was no longer an issue. Each time interval contains precisely one reaction. However, the computation time of the algorithm has seen marked reductions since its inception. Gibson and Bruck provide us with our first.

2.3 Gibson-Bruck KMC

One particularly inefficient piece of Gillespie's original algorithm is in the update step for the collection of reaction propensities $\{C_k\}_{k=1}^{N}$. Since not every C_k will necessarily be affected by the occurrence of a given reaction, we seek to alleviate this problem by categorizing the reactions by those other reactions that they influence.

Definition 2.7. We define the dependency graph of the system to be the directed graph whose vertex set consists of all the reactions $\{R^{(i)}\}_{i=1}^{N}$. We will place a directed edge from the vertex $R^{(m)}$ to the vertex $R^{(n)}$ if and only if the set of reactants and products in $R^{(m)}$, namely $\{R_1^{(m)}, \ldots, R_{r_m}^{(m)}\} \bigcup \{P_1^{(m)}, \ldots, P_{p_m}^{(m)}\}$, shares a nontrivial intersection with the set of reactants in $R^{(n)}$, $\{R_1^{(n)}, \ldots, R_{r_n}^{(n)}\}$.

Assuming that such a dependency graph D can be constructed, we may now update the reaction propensities in constant time, bounded from above by the degree of the highest degree vertex $R^{(v)}$ in D. For, suppose that a reaction of type $R^{(v)}$ happens, then for each reaction $R^{(i)}$ not in the neighborhood of $R^{(v)}$, none of the reactant molecules in $R^{(i)}$ will be altered by the occurrence of $R^{(v)}$; hence, the probability that those molecules collide and react in the space will also be unaltered. On the other hand, if $R^{(i)}$ *is* in the neighborhood of $R^{(v)}$, then at least one of the molecules that

are required for $R^{(i)}$ has seen its concentration in the space perturbed; therefore, the reaction rate is subject to change.

So, performing an update on the values $\{C_k\}_{k=1}^N$ only requires a limited number of actual updates. Because of the sparsity of this step, recalculating the sum $\sum_k C_k$ will also be reduced to constant time. Suppose without loss of generality that the values $C_1, \ldots, C_L$ were updated to their new values $C'_1, \ldots, C'_L$, and the rest of the values remain static $C_{L+1} = C'_{L+1}, \ldots, C_N = C'_N$. Then the sum may be recalulated as

$$\begin{aligned}
\sum_{k=1}^{N} C'_k &= \sum_{k=L+1}^{N} C'_k + \sum_{k=1}^{L} C'_k \\
&= \sum_{k=L+1}^{N} C_k + \sum_{k=1}^{L} C_k + \sum_{k=1}^{L} C'_k - \sum_{k=1}^{L} C_k \qquad (2.20) \\
&= \sum_{k=1}^{N} C_k + \sum_{k=1}^{L} (C'_k - C_k),
\end{aligned}$$

with the sum $\sum_{k=1}^{L} (C'_k - C_k)$ taking only constant time due to the bounds on the size of L.

With this, all steps of the algorithm take constant time with the exception of calculating n, our randomly determined selection for which reaction will take place in a given time step. In order to improve the calculation time here, we will construct another graph.

Definition 2.8. We define the indexed priority queue of the system to be a binary tree with weighted vertices whose leaves have weights equal to the reaction propensities $\{C_k\}_{k=1}^N$, and whose non-leaf nodes have weights equal to the sum of the weights of the two vertices beneath them on the tree (Figure 2.1).

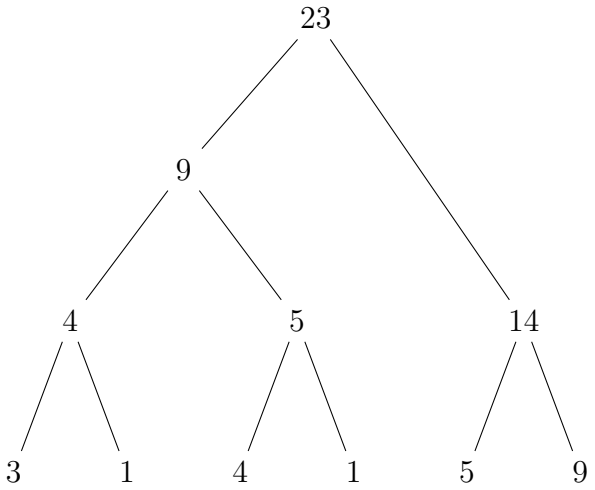


Figure 2.1 Demonstration of the form of an indexed priority queue.

Suppose we are at the step of Gillespie's algorithm in which we are tasked with finding n, the index of the reaction which will be conducted. We proceed as follows.

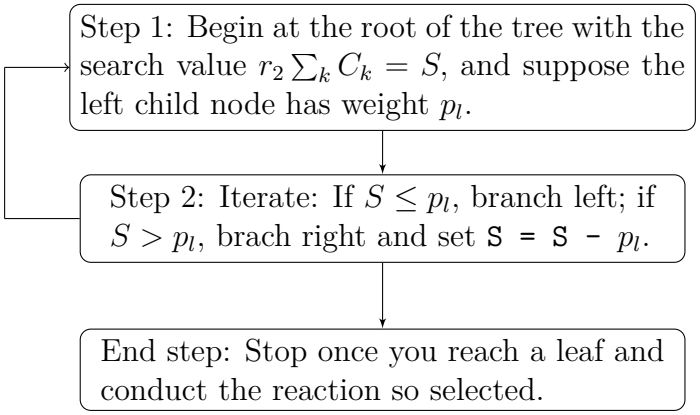
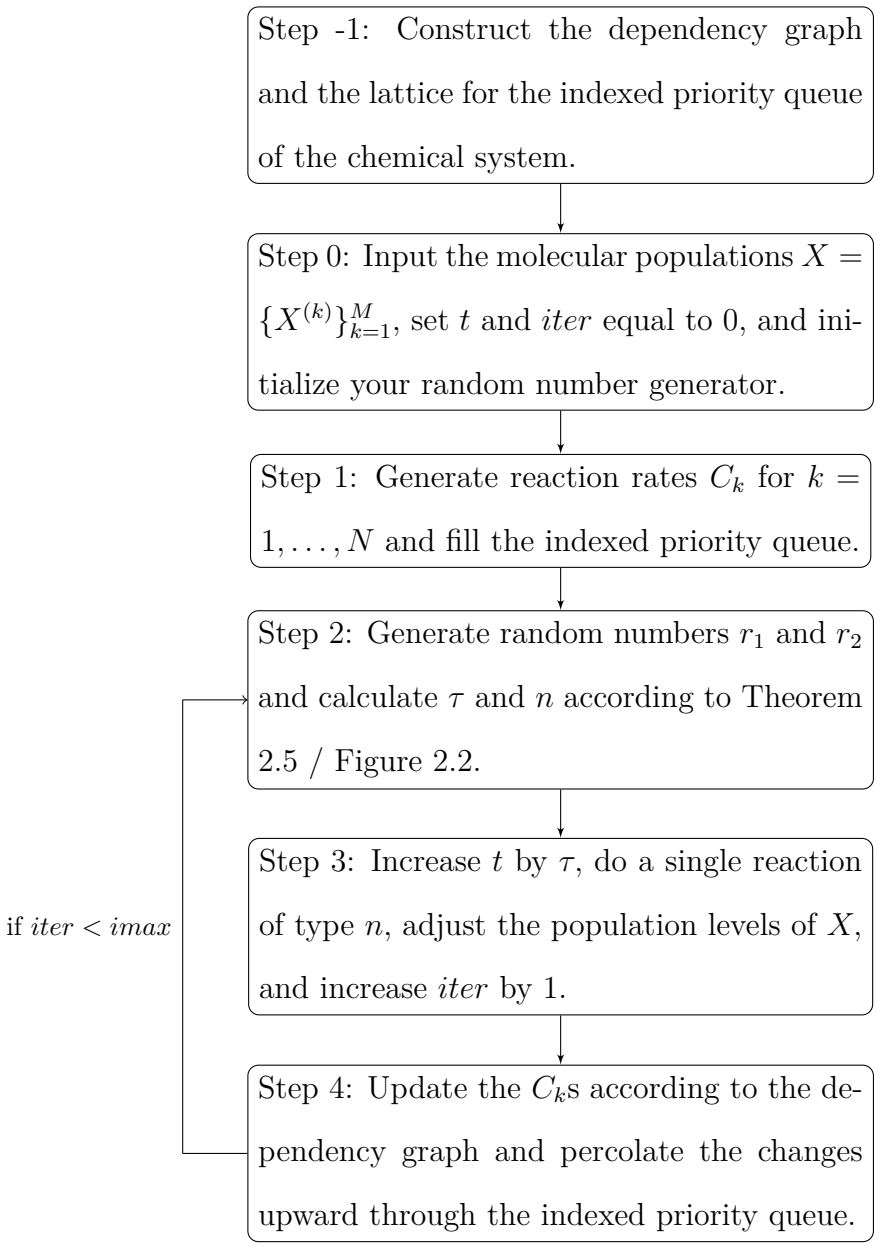


Figure 2.2 Selection method in an indexed priority queue.

We present the following claim without proof.

Theorem 2.9. *The reaction selected by the procedure outlined in Figure 2.2 is the same n as would be chosen in the original Gillepsie algorithm.*

Now, once the appropriate reaction has been selected, we alter the reaction propensities according to the dependency graph, and we percolate the changes upward through the indexed priority queue by recomputing partial sums as in (2.20). This leads to our updated Gibson-Bruck version of the Kinetic Monte Carlo method.



Proposition 2.10. *The Gibson-Bruck simulation algorithm scales in* $O(\log_2(N))$ *time.*

Proof. Since calculation of τ and performing the reaction are operations that have both remained unchanged from Gillespie, each scales as $O(1)$, as they did previously. With the addition of the dependency graph, updating C_k values is also $O(1)$, since only a bounded number of now identifiable reactions can be affected in one iteration of the method.

Taking the place of updating the sum $\sum_k C_k$ is updating the indexed priority queue, requiring one sum update for each tier. By applying (2.20), each update is $O(1)$, and since there are N leaves in the binary graph, there are $O(\log_2(N))$ tiers, assuming roughly equal distribution of vertices. Hence, the sum update is $O(\log_2(N))$. Similarly, since calculating n is done via binary search through the tiers of the indexed priority queue, this too will be a $O(\log_2(N))$ operation. □

The Gibson-Bruck algorithm is a marked improvement over Gillespie, reducing linear scaling to logarithmic while still making the same selections. In this way, it is a pure improvement, requiring only a minimal amount of additional theory to provide structure, leading to not just probabalistically, but precisely the same result. The final step we make in improving the computational behavior of KMC doesn't leave the original algorithm quite so unperturbed, but STP KMC has the great advantage of having constant scaling.

2.4 Constant-Time KMC

To further reduce the scaling behavior of Kinetic Monte Carlo to constant time, we adopt what we shall refer to as the rejection scheme. At its core, the idea of the scheme is to assemble all the potential reactions in the system into a bar graph, so that the x-axis consists of the discrete and numbered reactions while the y-axis represents the propensity for each corresponding reaction to occur in the next time interval (Figure 2.3)

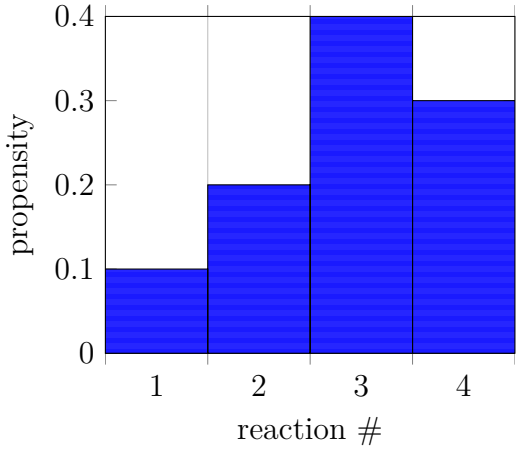


Figure 2.3 Demonstration of a rejection scheme reaction propensity graph.

In order to choose a reaction to conduct, we pick two random numbers, an integer r_{int} over $\{1, \ldots, N\}$ and a random number r_{float} in $[0, \max_k(C_k)]$. We then check the coordinate (r_{int}, r_{float}). If it is inside the bar representing the propensity of reaction $R^{(r_{int})}$, then we conduct the reaction. If not, then we reject the attempt and try again.

Proposition 2.11. *A reaction picked by the rejection scheme is done so with the same probability as in the Gillespie scheme.*

Proof. Since coordinates falling outside of bars in the graph are rejected outright, it suffices to check that a reaction is picked with the same probability given that the random number pair was accepted. For that, we note that by construction, picking n in the original Gillespie scheme was identical to sequentially ordering the reactions in $[0, \sum_k C_k]$, where each reaction $R^{(k)}$ takes its own interval of size C_k in $[0, \sum_k C_k]$, and then picking a random number $r_2 \sum_k C_k$ in $[0, \sum_k C_k]$. If the random number so picked fell into the interval representing reaction n, this was the reaction we chose to conduct, and the probability is $\frac{C_n}{\sum_k C_k}$. Now, in the composition and rejection scheme, if we assume that the point we pick falls into a bar somewhere, then it falls into the

bar representing reaction $R^{(n)}$ with probability $\frac{C_n}{\sum_k C_k}$, since the total sum of all the bar lengths is $\sum_k C_k$. □

The inherent problem with this method is that the reaction propensities need not be prevalent in the area bounding the bar graph. It is conceivable, then, that we could make many rejected attempts before one is successful. Fortunately, since the rejections do not affect the probability of picking a given reaction, we may reduce the number of rejections simply by carefully constructing our bar graph to have an acceptably small blank space.

We make the definitions: $C_{\min}$ is the minimum expected nonzero C_k value (less than or equal to $\min_k C_k$) and $C_{\max}$ is the maximum expected C_k value (greater than or equal to $\max_k C_k$). Then, we collect reactions together into groups $1, 2, \ldots, \Gamma$, where reaction $R^{(k)}$ is in group j if and only if $2^{j-1}C_{\min} \leq C_k < 2^j C_{\min}$. Suppose that the sum of all the reaction propensities in a given group G is defined to be $\sum_G C_k$. Now, we add an additional step to our scheme, wherein we generate a random number r_g in $[0, \sum_k C_k]$ and we pick the group $\widetilde{G}$ as our operating space if r_g falls into the collection of reaction propensities in $\widetilde{G}$:

$$\sum_{G=1}^{\widetilde{G}-1} \sum_{G} C_k < r_g \leq \sum_{G=1}^{\widetilde{G}} \sum_{G} C_k. \tag{2.21}$$

Once we have found $\widetilde{G}$, then we proceed as before with two additional random variables to pick out the specific reaction within group $\widetilde{G}$, using a y-axis bounded above by $2^{\widetilde{G}}C_{\min}$ (Figure 2.4). We will refer to this modified version of the rejection scheme as the composition and rejection (C&R) scheme.

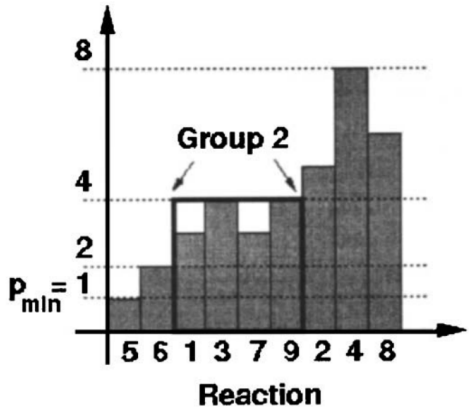


Figure 2.4 Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton, A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction network,. The Journal of Chemical Physics, 128(20):, 2008, Figure 3.

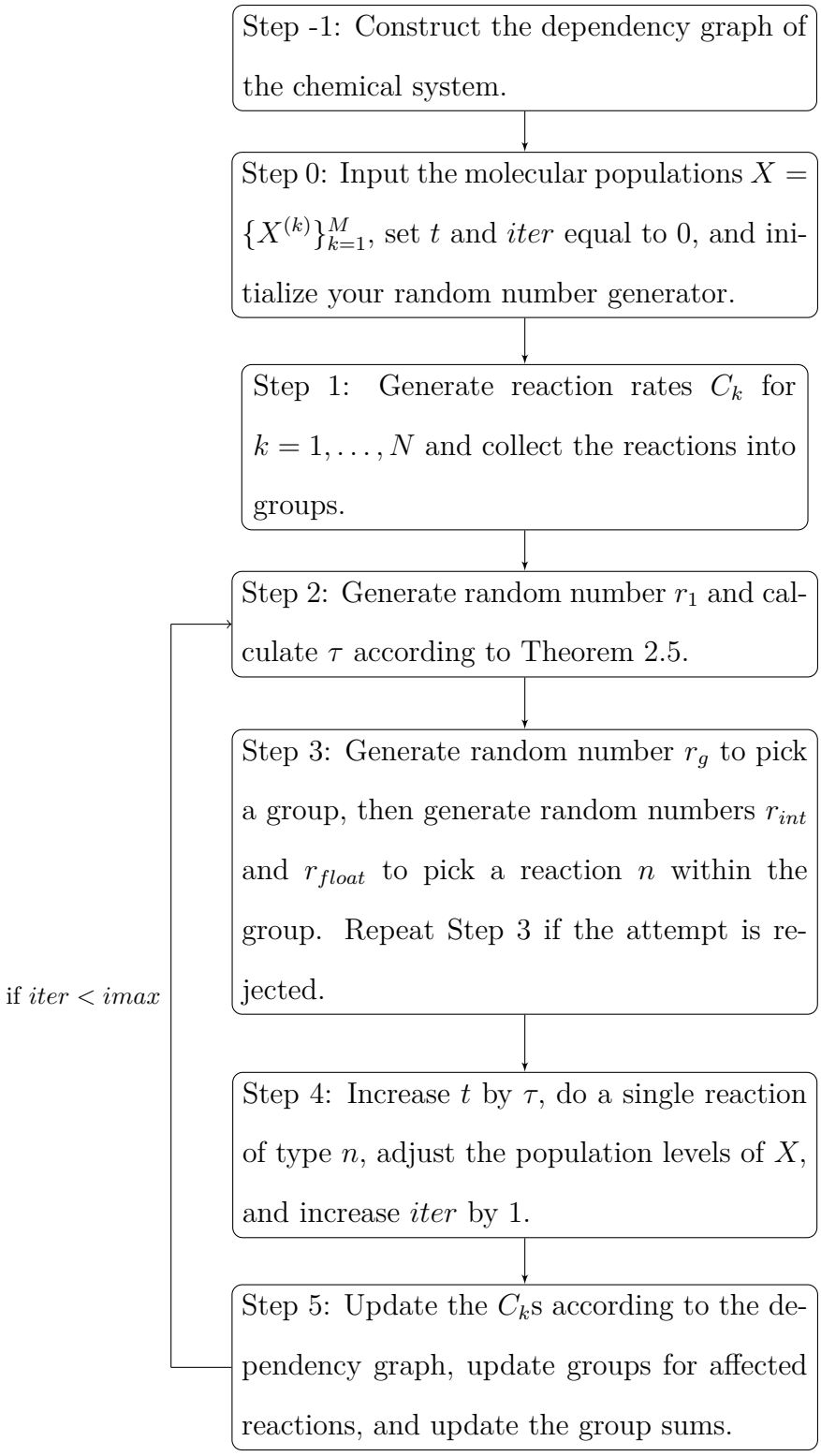
The following propositions are easily verified.

Proposition 2.12. *A reaction picked by the C&R scheme is done so with the same probability as in the Gillespie scheme.*

Proposition 2.13. *By grouping the reactions as described, we reduce the expected number of attempts to at most* 2*. Also, the number of groups is independent of* N*.*

The disadvantage of this scheme is the necessity of tracking to which group a given reaction currently belongs, and maintaining the proper group sums. However, as was shown in (2.20), recomputing sums is a bounded procedure, and so long as we leave the thresholds for group membership static, then determining to which group a reaction belongs is simple. Recall also that the dependency graph of the system will inform us as to which reactions have the potential to change propensities and likewise to change groups.

Now, we are prepared to show STP KMC.



Proposition 2.14. *The STP KMC simulation algorithm scales in $O(1)$ time.*

Proof. Calculation of τ and performing the reaction are still unchanged from Gillespie,

so each scales as $O(1)$, as they did previously. Updating C_k values is $O(1)$ due to the dependency graph, though we now have the additional consideration of tracking groups for the reactions and group sums. However, this too scales as $O(1)$, since only a bounded number of reactions require updated C_k values; therefore, only a bounded number of reactions change groups. The altered group sums can be calculated in $O(1)$ time, as shown in (2.20), and the number of groups requiring such updates is independent of N, as stated in Proposition 2.13.

Also by Proposition 2.13, picking n is now an $O(1)$ operation, as it is the result of three random number generations with an expected number of attempts less than two. □

With this, we have now constructed an efficient kinetic time algorithm for the evolution of a system of chemicals. In the next section, we will take a step back and we will apply the algorithm to a lattice of objects interacting with one another in what is known as an Ising spin system.

2.5 KMC on Ising Spin Systems

An Ising spin system is simply a d-dimensional lattice in which each node has the property of being either "spin up" or "spin down". A lattice point can interact with its neighbors by reversing its own spin, or by interchanging its spin with that of its neighbor. The probability of any such alteration occurring at a node p depends only on the spins of the nodes p_j in the neighborhood of p, on the spins of the nodes in the neighborhoods of each p_j (in the event that the coming action is an interchange), and on the spin of p itself. The neighborhood of a node is dependent on the problem at hand. In Chapter 3, we will use the spin of a lattice point to indicate whether the node currently holds a cell or a medium element. Since we will also assume that cells neither die nor reproduce, it can be said that in our problem, the probability of a node reversing its spin alone is always zero, and there will only be spin interchanges.

Further, in our 3-dimensional lattice, the neighborhood of a node p will be all nodes contained within the 3×3 cube of nodes surrounding p.

For now, we aim to determine an upper bound on how many possible reaction types there are in an Ising spin system with known node neighborhoods.

Theorem 2.15. *In an Ising spin system in which each node has k neighbors, there are at most $2(1+k)(1+k^2)$ possible reaction types.*

Proof. Fix a node p. Since we are only aiming for an upper bound, we assume for simplicity that the neighbor sets of the neighbors of p are disjoint. First, there are 2 ways to pick a spin for p itself. From here, we break up the potential reactions into self-spin reversals, which depend only on neighbors of p, and spin interchanges, which also depend on the neighbors of the potential interchange target. If m_1 is the number of potential configurations of the neighbors of p, and m_2 is the number of ways to pick an interchange target and arrange its and p's neighbors, then the solution is $2[m_1 + m_2]$.

To find m_1, we note that it doesn't matter precisely *where* the nodes with a given spin are located, just so long as they are in the neighborhood of p. Therefore, it suffices to count the number of neighbors with an up spin. There are $k+1$ such configurations, namely 0 neighbors, 1 neighbor, ..., or k neighbors with up spin.

To find m_2, we observe that given we have fixed the configuration of the neighbors of p as before, there are k ways to pick a neighbor p_n - whose spin has already been fixed, and then, removing p as a neighbor of p_n - since the spin of p has already been fixed as well - k ways to configure the neighbors of p_n. Note that this will double count some configurations, but we are aiming only for an upper bound; hence, $m_2 \leq k^2(k+1)$.

Therefore, there are at most $2\left[(k+1) + k^2(k+1)\right] = 2(1+k)(1+k^2)$ possible reaction types in the Ising spin system. □

Corollary 2.16. *If we are working in an Ising spin system in which each node has k neighbors and it is possible to ignore spin interchanges, then the number of potential reaction types is precisely equal to* $2(1+k)$.

Proof. Observe that the assumptions that contributed to the inequality in the proof of the theorem were all made on the neighbors' neighbors or in calculating the bound on the number of interchange reactions. □

Corollary 2.17. *If we are working in an Ising spin system in which each node has k neighbors and it is possible to ignore self-spin reversals, then the number of potential reaction types is at most* $2k^2(1+k)$.

Suppose that there are K reaction types and that we can find probabilities $P_1, \ldots, P_K$ for the potential reactions. Suppose further that we wish to conduct a reaction in the system corresponding to the probabilities we have for each element in the set of all possible events $E = \{e_1, \ldots, e_N\}$, namely every possible self-spin reversal and every node-neighbor swap. Note that if the lattice is static, then the set of possible events is also, though the propensity for any given event may alter as the system evolves. Let E_{P_i} denote the set of events whose current probability is P_i. Then the total reaction propensity - denoted by $\sum_k C_k$ in the previous sections - is given by

$$\begin{aligned} \sum_{e \in E} \text{prob}(e) &= \sum_{i=1}^{K} \sum_{e \in E_{P_i}} \text{prob}(e) \\ &= \sum_{i=1}^{K} \left| E_{P_i} \right| P_i. \end{aligned} \tag{2.22}$$

We organize the potential events in the system in a list-based manner wherein each event can be identified not only by its type, but also by its position as a member of the set of events sharing the same reaction probability. This can be done by creating a table in which the row corresponds to an event's probability and the column

corresponds to its membership index, or one can utilize more creative methods, an example of which is discussed in the paper by Bortz, Kalos, and Lebowitz. We will assume the use of a tabular method for the remainder of this section.

To conduct a reaction, we calculate two random variables r_1 and r_p in $[0, 1]$. Finding the reaction time is done precisely the same way as in Gillespie, so

$$\tau = \frac{1}{\sum_i \left|E_{P_i}\right| P_i} \log\left(\frac{1}{r_1}\right). \tag{2.23}$$

To find the event to conduct, we take two steps. First we pick the reaction type by finding n such that

$$\frac{\sum_{i=1}^{n-1} \left|E_{P_i}\right| P_i}{\sum_i \left|E_{P_i}\right| P_i} < r_p \leq \frac{\sum_{i=1}^{n} \left|E_{P_i}\right| P_i}{\sum_i \left|E_{P_i}\right| P_i}. \tag{2.24}$$

Then, we choose the particular event by taking a random integer $r_n \in \left\{1, \ldots, \left|E_{P_n}\right|\right\}$.

In the update step, we check all the nodes that are either neighbors or neighbors' neighbors to the altered node(s), and we change the reaction propensities accordingly. For example, in the two dimensional planar case where two nodes are considered neighbors if and only if they share an edge, if we interchange the spins of a pair of nodes, then there are a total of 8 self-spin reversal events and 23 interchange events that require updated probabilities. The dependencies of given events on other events may be tracked via a dependency graph as in the case of a chemical system, though it bears mentioning that in the case of an Ising spin system, the number of dependencies scales with the number of neighbors for each node, as opposed to the number of chemical species.

Once those probabilities have been updated, we may relocate events to new rows if necessary, and we may reindex the elements of the rows so affected, updating their row sums as in (2.20). Hence, our algorithm for KMC on an Ising spin system may be stated as follows.

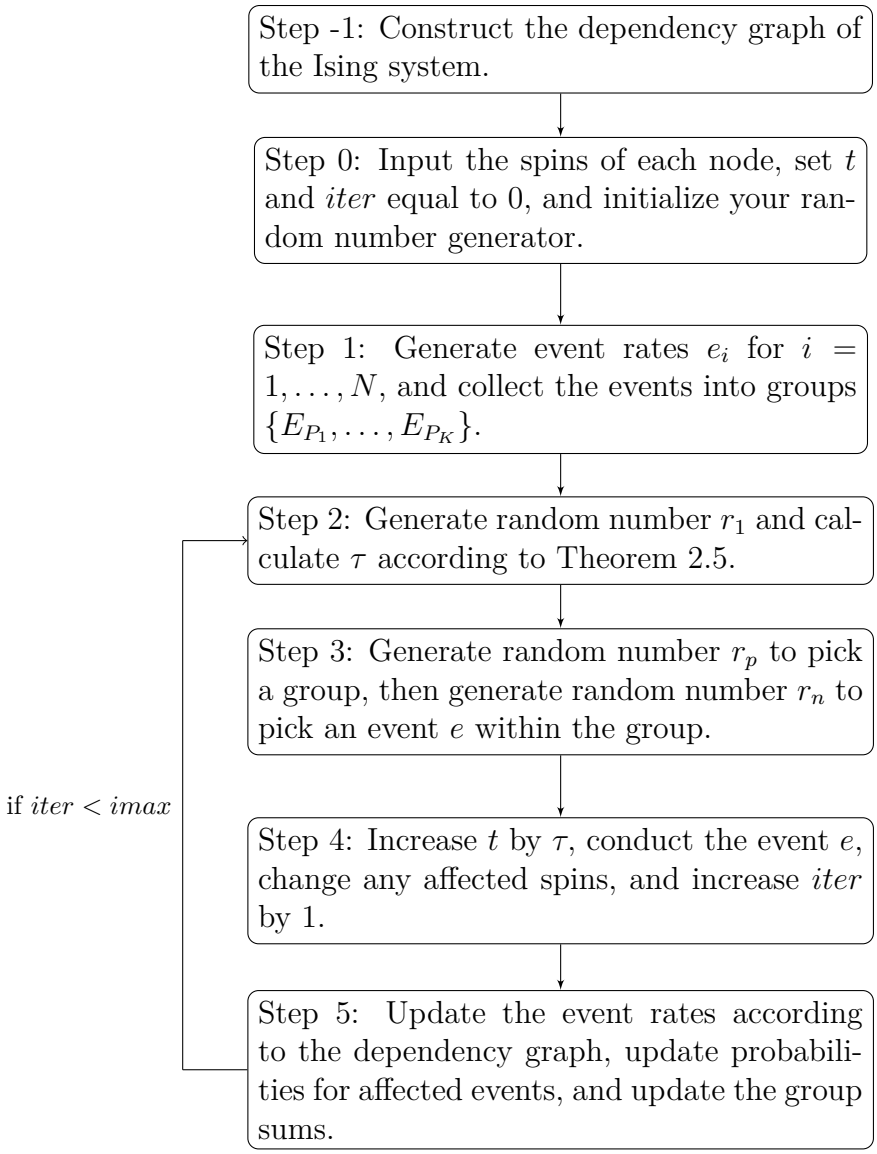


Figure 2.5 KMC on an Ising spin system.

Proposition 2.18. *For an Ising spin system, the scaling of the KMC algorithm as described is $O(K)$.*

Proof. Calculation of τ, performing the reaction, and updating the probabilities for each event e_i are $O(1)$ by Proposition 2.14, with the group sums of STP KMC having been replaced by event list sums. Again, since only a bounded number of events are affected by the conduction of another event, updating the event lists and event sums

is an $O(1)$ operation.

However, to find which reaction type to conduct, we must now find n satisfying (2.24). As shown in Proposition 2.6, this scales as the upper limit of the sum, namely K. Finding the particular reaction to enact afterwards is $O(1)$, as we need only check the number of elements in E_{P_n} and take a random integer. □

So long as the number of reaction types isn't too large, the setback in scaling will be manageable. In the next chapter, we will discuss our assumptions for our Ising-based model, and we will provide the method used to calculate the event probabilities. After, results of several different simulations will be provided.

Chapter 3

Applications to Multicellular Systems

3.1 Explanation of Reaction Rates

As stated in the previous section, we will consider the space in which the cells lie to be a three-dimensional Ising spin system with periodic boundary conditions where cell nodes have an up-spin and medium nodes have a down-spin. Further, we will consider two nodes to be neighbors if they occupy a common 3x3x3 cube of nodes [9], and we forbid self-spin reversals, i.e. removals or additions of cells. For a given node $\mathbf{r} = (\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z)$, we define

$$\sigma_{\mathbf{r}} = \begin{cases} 1, & \text{if } \mathbf{r} \text{ holds a cell} \\ 0, & \text{if } \mathbf{r} \text{ holds a medium element} \end{cases} . \tag{3.1}$$

Let $E(\sigma_{\mathbf{r}}, \sigma_{\mathbf{s}})$ denote the interaction energy - the strength of the bond - between neighbor sites $\mathbf{r}$ and $\mathbf{s}$, dependent only on the elements the nodes are currently holding. Then there are three possible pairwise interaction energies in the system, which we will define as follows:

$$\begin{aligned} E(0,0) &=: -E_{mm} \\ E(0,1) = E(1,0) &=: -E_{cm} \\ E(1,1) &=: -E_{cc}. \end{aligned} \tag{3.2}$$

Define N_m to be the total number of medium elements in the system, N_c to be the total number of cells in the system, and N_{cm} to be the total number of cell-medium neighbor pairs in the system.

Theorem 3.1. *The total energy E in the system is given by $E = \left(\frac{E_{cc}+E_{mm}}{2} - E_{cm}\right) N_{cm} - 13N_cE_{cc} - 13N_mE_{mm}$.*

Proof. First, note that each node in the Ising system described has 26 neighbors. We calculate,

$$\begin{aligned} E &= \sum_{\substack{\mathbf{r} \text{ and } \mathbf{s} \\ \text{neighbors}}} E(\sigma_{\mathbf{r}}, \sigma_{\mathbf{s}}) \\ &= -\sum_{mm \text{ pairs}} E_{mm} - \sum_{cc \text{ pairs}} E_{cc} - \sum_{cm \text{ pairs}} E_{cm}. \end{aligned} \tag{3.3}$$

We claim that $-\sum\limits_{mm \text{ pairs}} E_{mm} = -E_{mm}\Big|\{mm \ pairs\}\Big|$ is given by $-E_{mm}\Big(\frac{1}{2}\Big((26)N_m - N_{cm}\Big)\Big)$. Indeed, consider a single medium element in the space. We may count all the bonds around it and subtract off those bonds that are between it and a cell. If we do this for every medium element, we get a total of $(26)N_m - N_{cm}$; however, this double counts every bond. Hence, dividing by 2 produces the appropriate number of medium-medium bonds, and $-\sum\limits_{mm \text{ pairs}} E_{mm} = -E_{mm}\Big(\frac{1}{2}\Big((26)N_m - N_{cm}\Big)\Big)$. Similarly, $-\sum\limits_{cc \text{ pairs}} E_{cc} = -E_{cc}\Big(\frac{1}{2}\Big((26)N_c - N_{cm}\Big)\Big)$.

Observe that $-\sum\limits_{cm \text{ pairs}} E_{cm} = -E_{cm}N_{cm}$ by the definition of N_{cm}. Therefore,

$$\begin{aligned} E &= -\sum_{mm \text{ pairs}} E_{mm} - \sum_{cc \text{ pairs}} E_{cc} - \sum_{cm \text{ pairs}} E_{cm} \\ &= -E_{mm}\Big(\frac{1}{2}\Big((26)N_m - N_{cm}\Big)\Big) - E_{cc}\Big(\frac{1}{2}\Big((26)N_c - N_{cm}\Big)\Big) - E_{cm}N_{cm} \\ &= \left(\frac{E_{cc}+E_{mm}}{2} - E_{cm}\right) N_{cm} - 13N_cE_{cc} - 13N_mE_{mm} \end{aligned} \tag{3.4}$$

□

We define the constant $\gamma_{cm} := \frac{E_{cc}+E_{mm}}{2} - E_{cm}$. If we discount the possibilities of cell death and cell birth, so that the number of medium and cell nodes stays constant, then we have, for some constant C,

$$E = \gamma_{cm}N_{cm} + C. \tag{3.5}$$

Since the propensity of node interchanges to occur should depend on the change in E, it suffices to consider the modified energy term $\overline{E} = \gamma_{cm} N_{cm}$ in future calculations.

As given by the Arrhenius relation [3], we may take the swap rate of two neighbor nodes $\mathbf{r}$ and $\mathbf{s}$ to be

$$r(\mathbf{r}, \mathbf{s}) = \frac{1}{\tau_0} \exp\left(-\frac{\Delta E}{2}\right) = \frac{1}{\tau_0} \exp\left(-\frac{\Delta \overline{E}}{2}\right) \tag{3.6}$$

where τ_0 is the relaxation time of the system and the change terms are given by the change in the system energy that would occur in the hypothetical swap of the two nodes in question. Observe that $\Delta \overline{E} = \gamma_{cm} (\Delta N_{cm})$. Consider two neighbor nodes. If both are cells or both are medium elements, then we have $\Delta N_{cm} = 0$. Suppose the site $\mathbf{c}$ contains a cell and the site $\mathbf{m}$ contains a medium element. Then the change in N_{cm} is given by the change in the number of medium-type neighbors of the cell currently located at $\mathbf{c}$ after its swap, plus the number of cell-type neighbors of the medium element currently located at $\mathbf{m}$ after its swap.

Suppose $n_m(\mathbf{x})$ denotes the number of medium-type neighbors of site $\mathbf{x}$, and $n_c(\mathbf{x})$ denotes the number of cell-type neighbors of site $\mathbf{x}$. We have

$$\begin{aligned} \Delta N_{cm} &= \Big(n_m(\mathbf{m}) + 1 - n_m(\mathbf{c})\Big) + \Big(n_c(\mathbf{c}) + 1 - n_c(\mathbf{m})\Big) \\ &= \Big(26 - n_c(\mathbf{m}) + 1 - n_m(\mathbf{c})\Big) + \Big(26 - n_m(\mathbf{c}) + 1 - n_c(\mathbf{m})\Big) \\ &= 54 - 2n_c(\mathbf{m}) - 2n_m(\mathbf{c}). \end{aligned} \tag{3.7}$$

Therefore,

$$\begin{aligned} r(\mathbf{r}, \mathbf{s}) &= \frac{1}{\tau_0} \exp\left(-\frac{\Delta \overline{E}}{2}\right) \\ &= \frac{1}{\tau_0} \exp\left(-\frac{\gamma_{cm} \Delta N_{cm}}{2}\right) \\ &= \frac{1}{\tau_0} \exp\left(-\frac{\gamma_{cm}\Big(54 - 2n_c(\mathbf{m}) - 2n_m(\mathbf{c})\Big)}{2}\right) \\ &= \frac{1}{\tau_0} \exp\Big(-\gamma_{cm}\Big(27 - n_c(\mathbf{m}) - n_m(\mathbf{c})\Big)\Big). \end{aligned} \tag{3.8}$$

From here, we may proceed as in section 2.5. For each pair of neighbor points, it suffices to track the number of non-matching neighbors of each node in the pair. We will calculate the rate as shown in (3.8), and proceed on with the algorithm outlined in Figure 2.5. In the final section, we will show the results of several simulations using the Kinetic Monte Carlo algorithm that has been built up.

3.2 Simulation Results

The most basic simulation for the merging of multicellular aggregates would, naturally, be just two aggregates placed side by side. Using constants informed by biological testing, we have the following result.

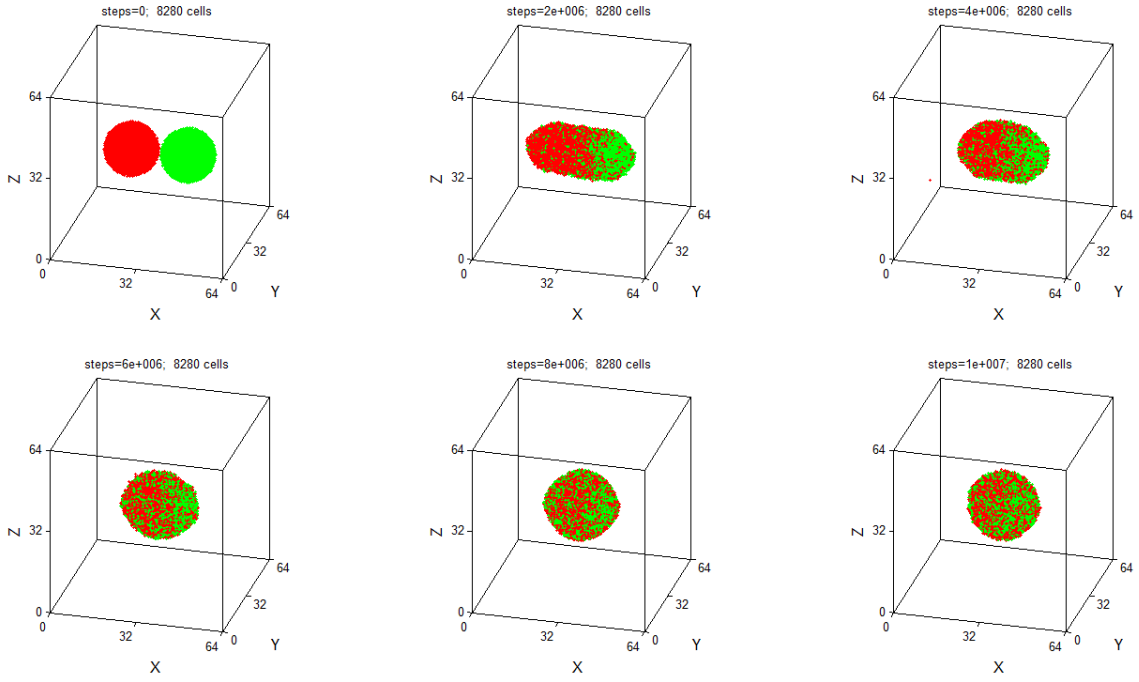


Figure 3.1 2 aggregates, $1 \cdot 10^7$ steps, aggregate radius 10, and aggregate distance 1.

In sectional view,

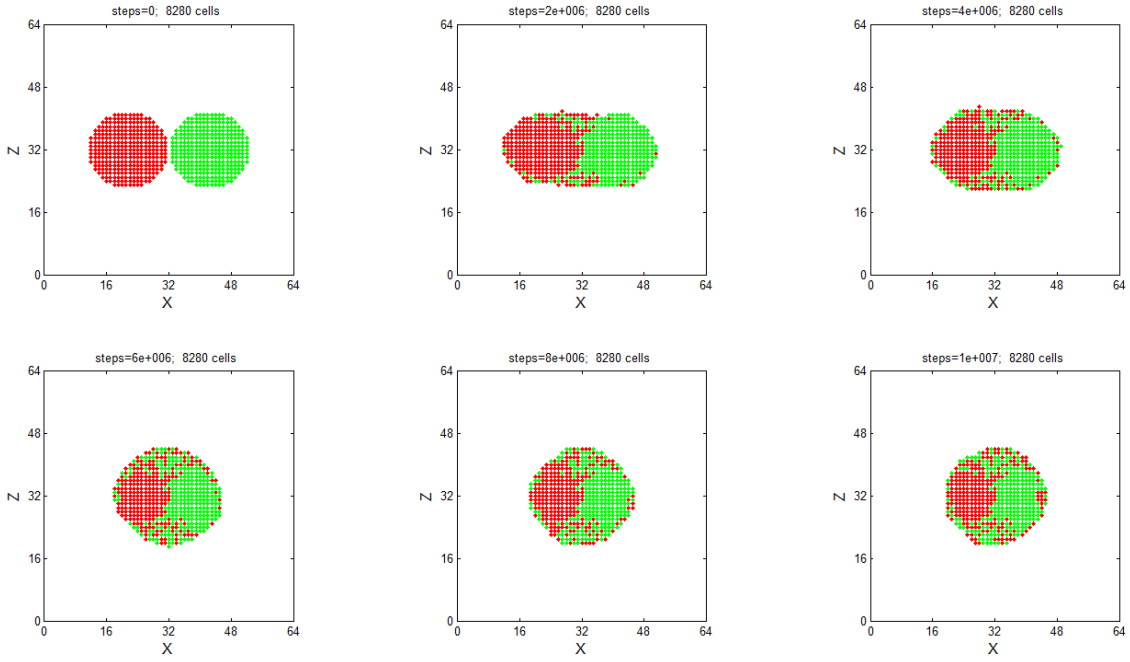


Figure 3.2 2 aggregates, $1 \cdot 10^7$ steps, aggregate radius 10, and aggregate distance 1. Sectional view.

Observe that the aggregates merge and the surface cells distribute themselves somewhat evenly, but the interior cells remain segregated. In the next figure, we can see how the free energy in the system, measured by the number of cell-medium bonds, generally decreases as the system evolves.

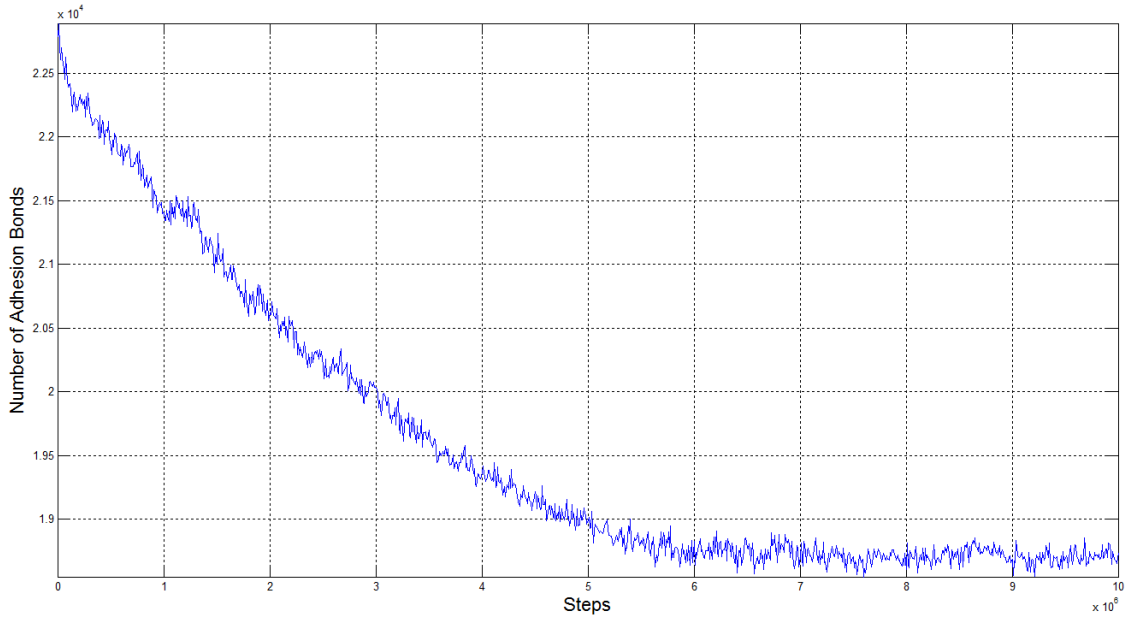


Figure 3.3 2 aggregates, $1 \cdot 10^7$ steps, aggregate radius 10, and aggregate distance 1. Energy estimate.

Additional simulations follow.

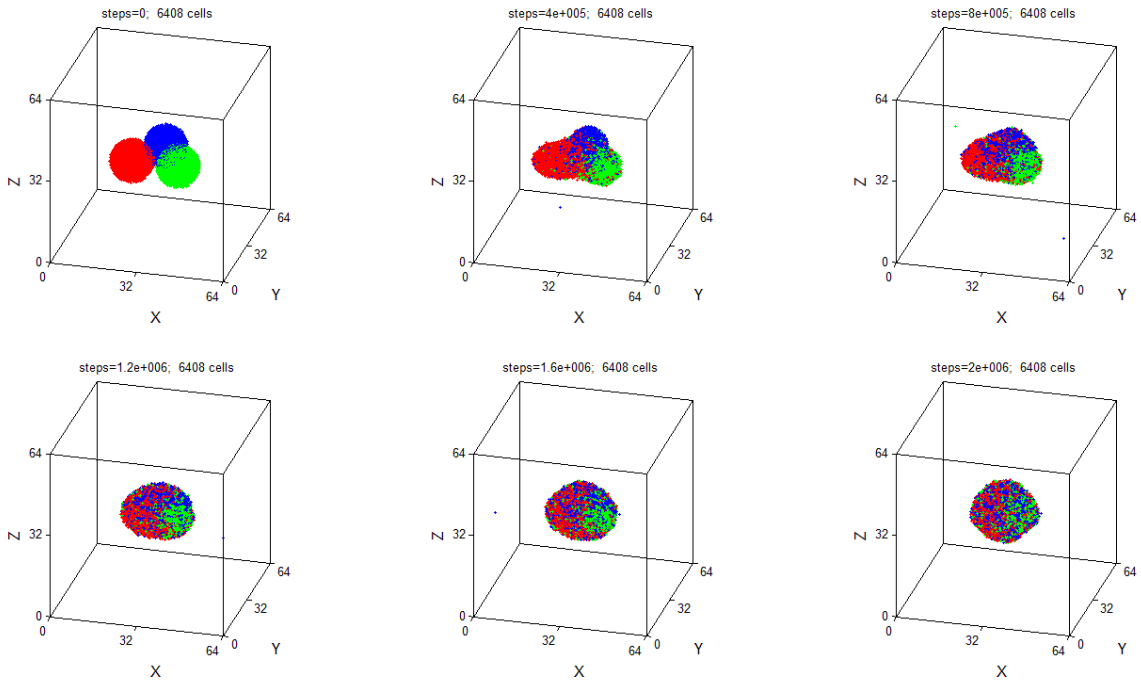


Figure 3.4 3 aggregates in triangular formation, $2 \cdot 10^6$ steps, aggregate radius 8, and aggregate distance 1.

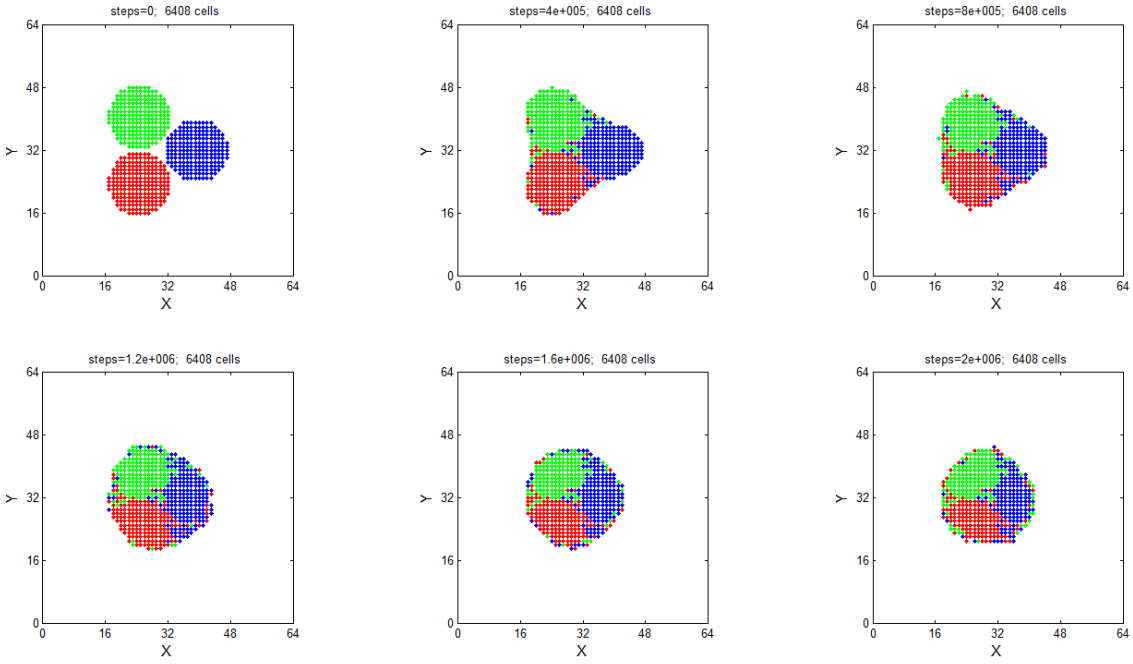


Figure 3.5 3 aggregates in triangular formation, $2 \cdot 10^6$ steps, aggregate radius 8, and aggregate distance 1. Sectional view.

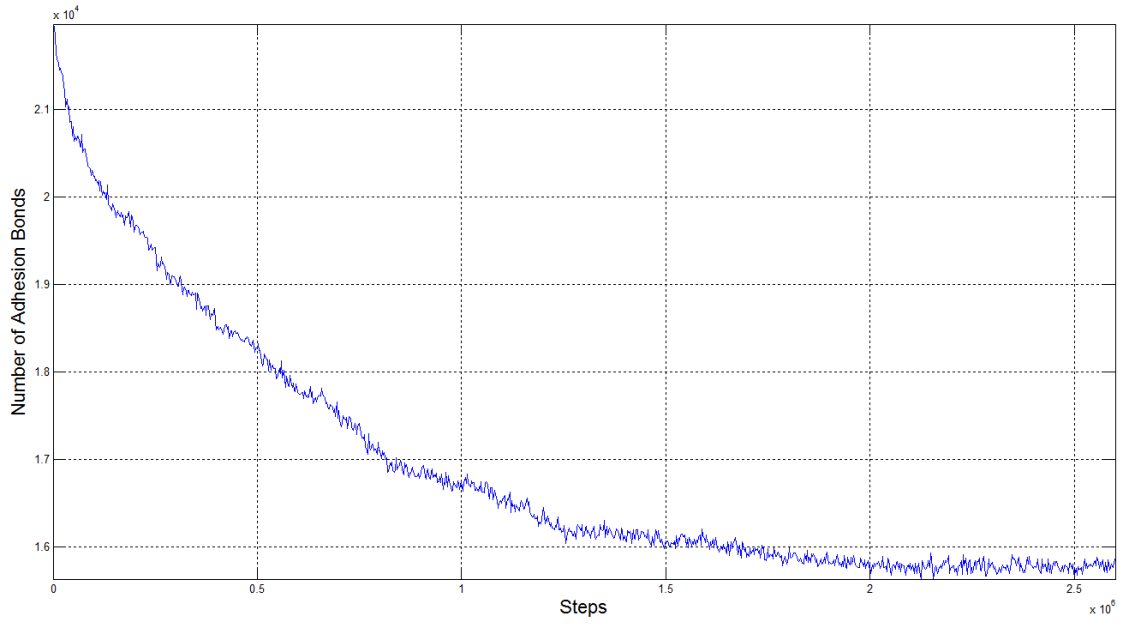


Figure 3.6 3 aggregates in triangular formation, $2.6 \cdot 10^6$ steps, aggregate radius 8, and aggregate distance 1. Energy estimate.

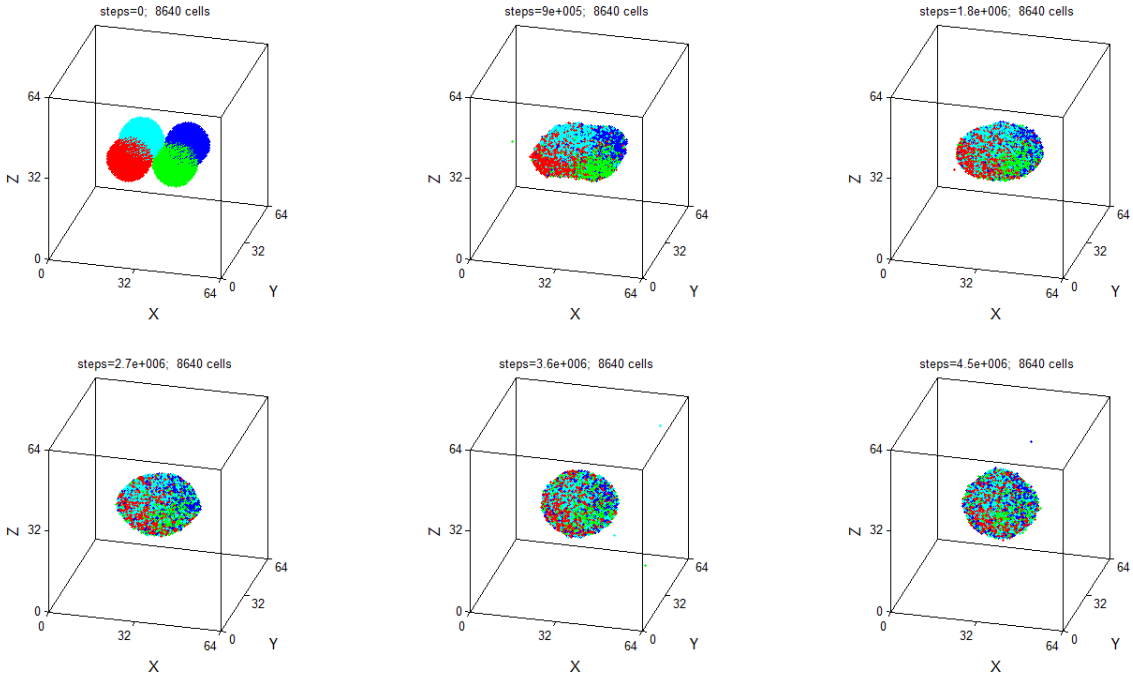


Figure 3.7 4 aggregates in square formation, $4.5 \cdot 10^6$ steps, aggregate radius 8, and aggregate distance 1.

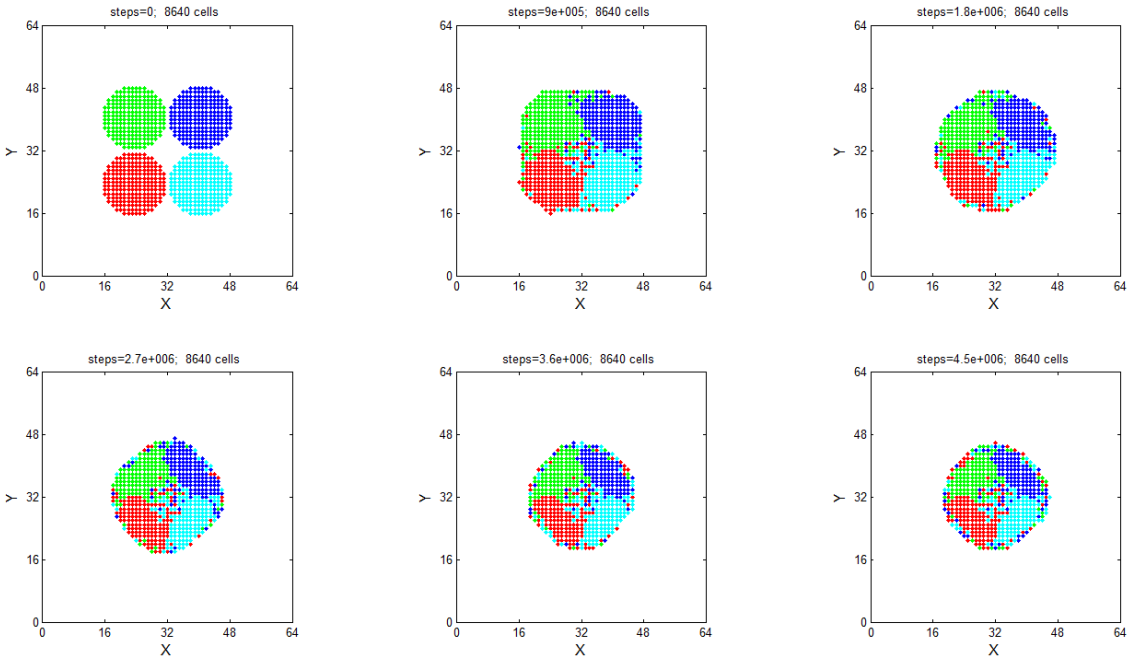


Figure 3.8 4 aggregates in square formation, $4.5 \cdot 10^6$ steps, aggregate radius 8, and aggregate distance 1. Sectional view.

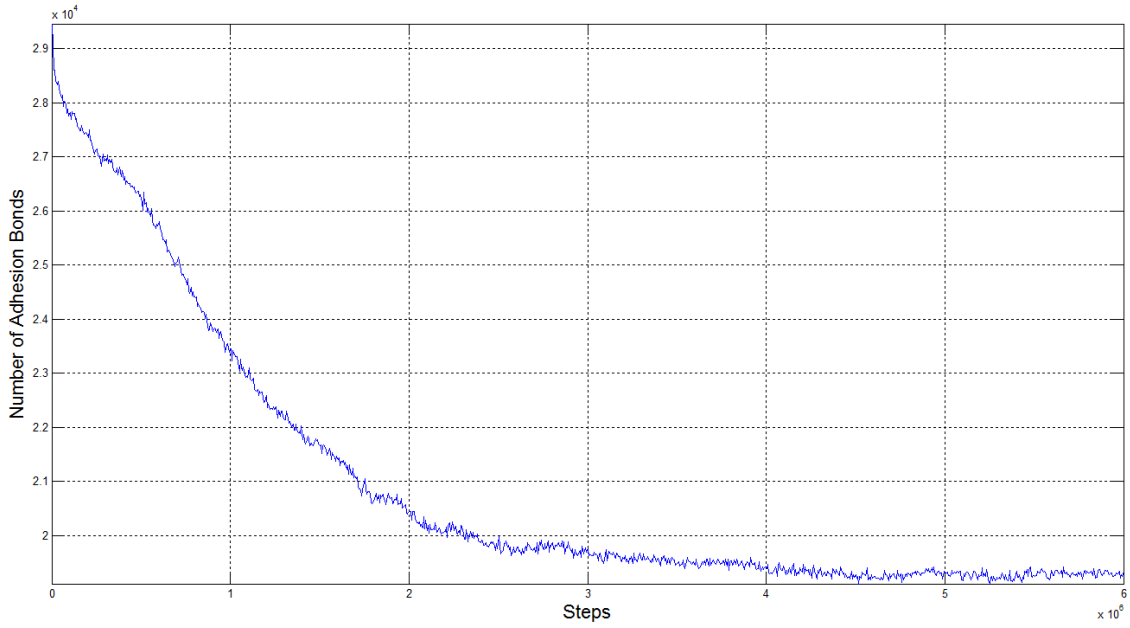


Figure 3.9 4 aggregates in square formation, $6 \cdot 10^6$ steps, aggregate radius 8, and aggregate distance 1. Energy estimate.

In the following tower simulation, note how the center of the mass doesn't fill initially, but eventually overcomes the tangential forces of the neighbor cells along the outside of the unfilled sphere formed by the aggregates. You can see two major drops in cell-medium bonds in Figure 3.12. The first is caused by the initial merger of the cells on the outside of the sphere. The second is caused when the empty center of the sphere collapses.

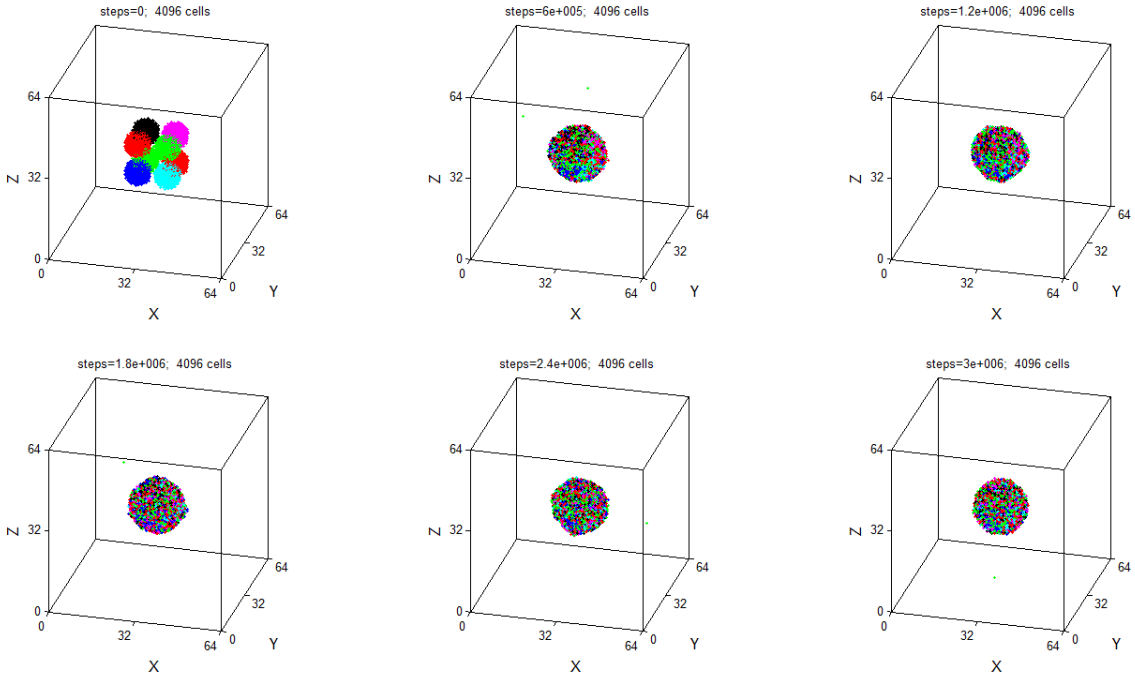


Figure 3.10 8 aggregates in 2-cube formation, $3 \cdot 10^6$ steps, aggregate radius 5, and aggregate distance 1.

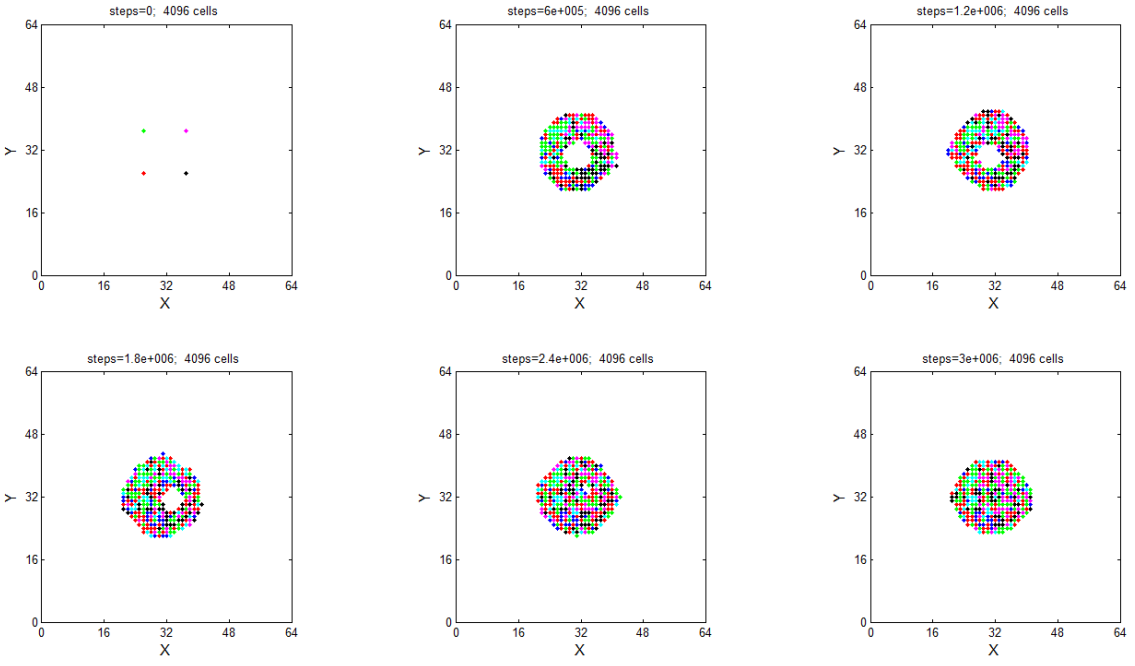


Figure 3.11 8 aggregates in 2-cube formation, $3 \cdot 10^6$ steps, aggregate radius 5, and aggregate distance 1. Sectional view.

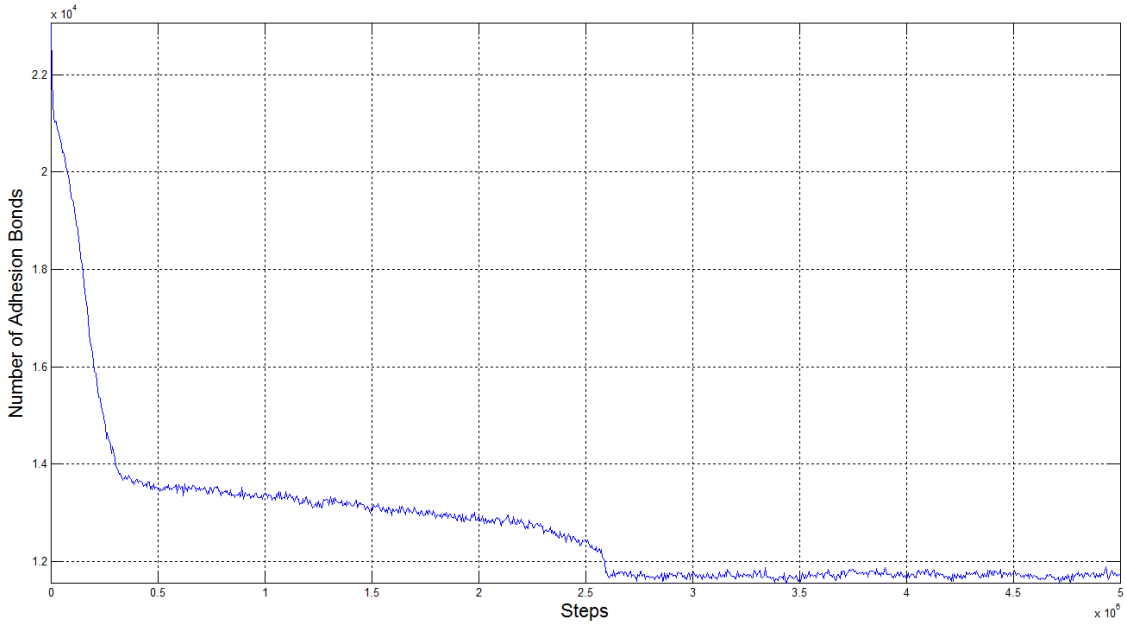


Figure 3.12 8 aggregates in 2-cube formation, $5 \cdot 10^6$ steps, aggregate radius 5, and aggregate distance 1. Energy estimate.

The next simulation shows the evolution of a cube of aggregates. They are arranged as follows: the first, the third, and the fifth layers have all cells of type 1 (red), the second and fourth layers are arranged as illustrated in Figure 3.13. Observe how the cells are pulled toward the outside of the mass before eventually falling inward to form a stable ball. Several different angles are provided.

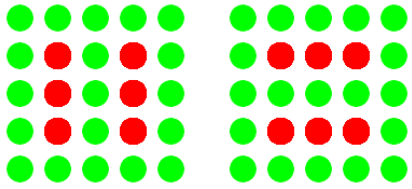


Figure 3.13 Illustration of layers 2 (on the left) and 4 (on the right) in the five-cube simulations.

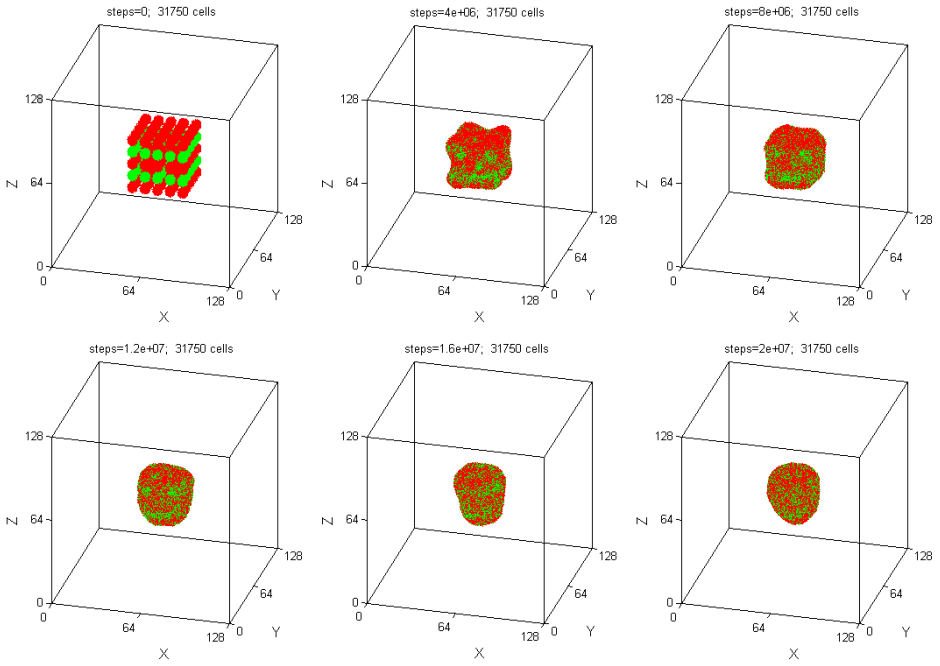


Figure 3.14 125 aggregates in 5-cube formation, $2 \cdot 10^7$ steps, aggregate radius 4, and aggregate distance 1.

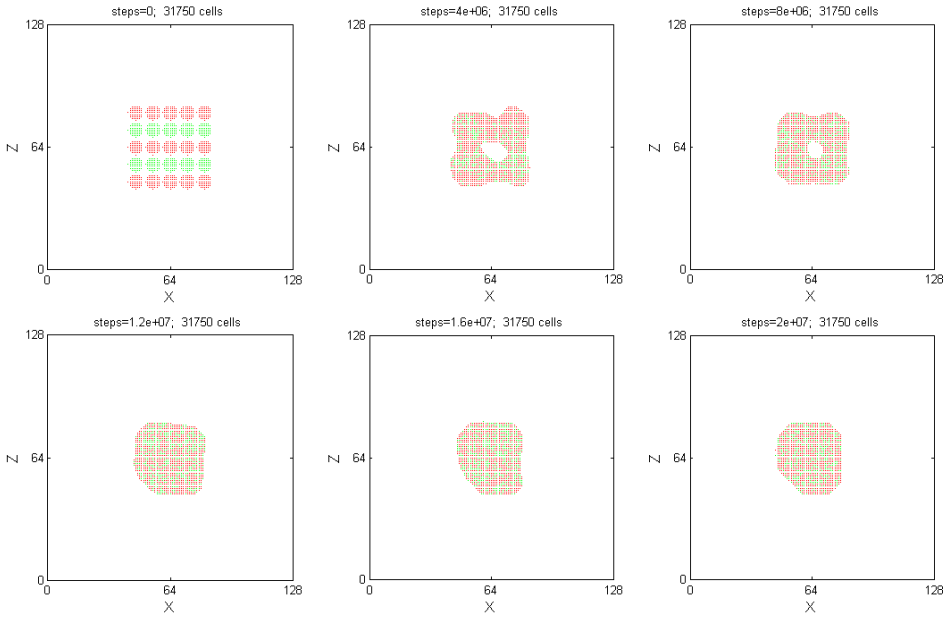


Figure 3.15 125 aggregates in 5-cube formation, $2 \cdot 10^7$ steps, aggregate radius 4, and aggregate distance 1. Side view.

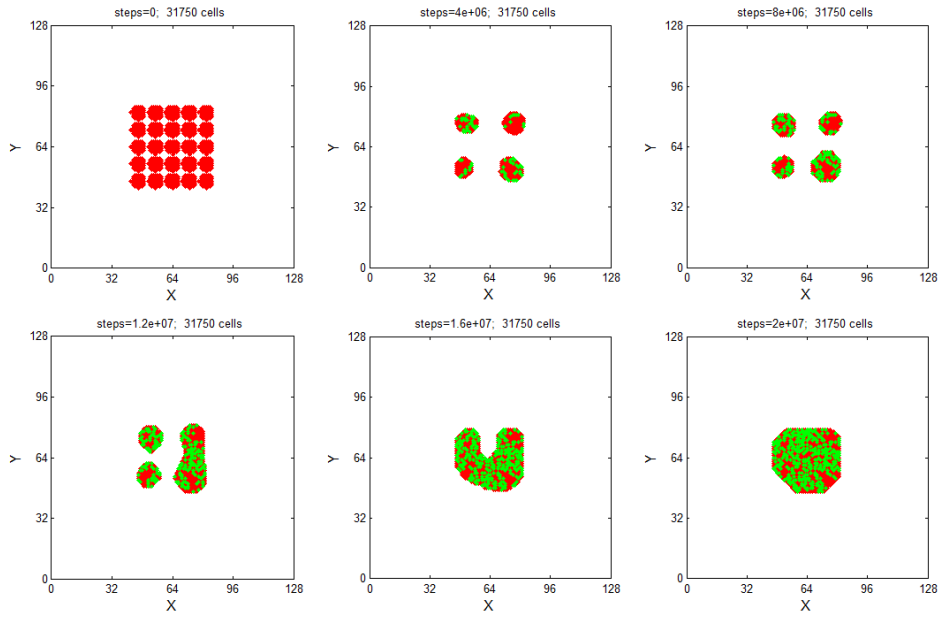


Figure 3.16 125 aggregates in 5-cube formation, $2 \cdot 10^7$ steps, aggregate radius 4, and aggregate distance 1. Horizontal cut.

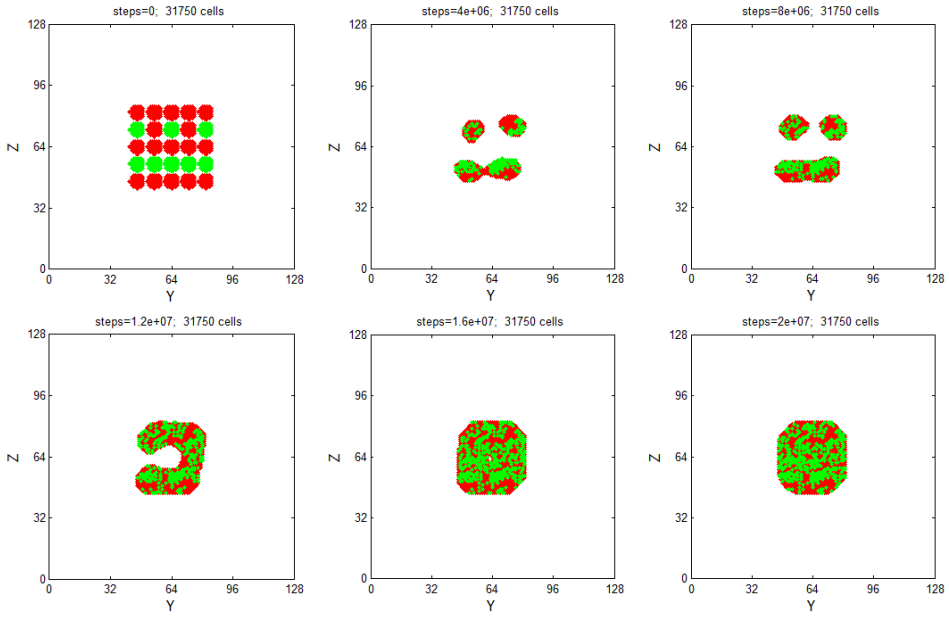


Figure 3.17 125 aggregates in 5-cube formation, $2 \cdot 10^7$ steps, aggregate radius 4, and aggregate distance 1. Vertical cut parallel to the ZY plane.

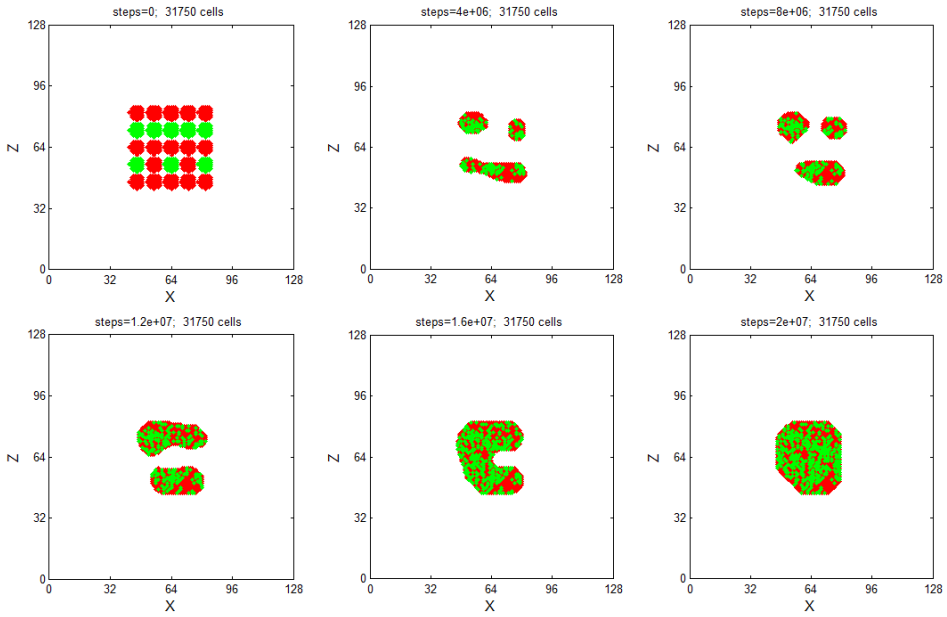


Figure 3.18 125 aggregates in 5-cube formation, $2 \cdot 10^7$ steps, aggregate radius 4, and aggregate distance 1. Vertical cut parallel to the ZX plane.

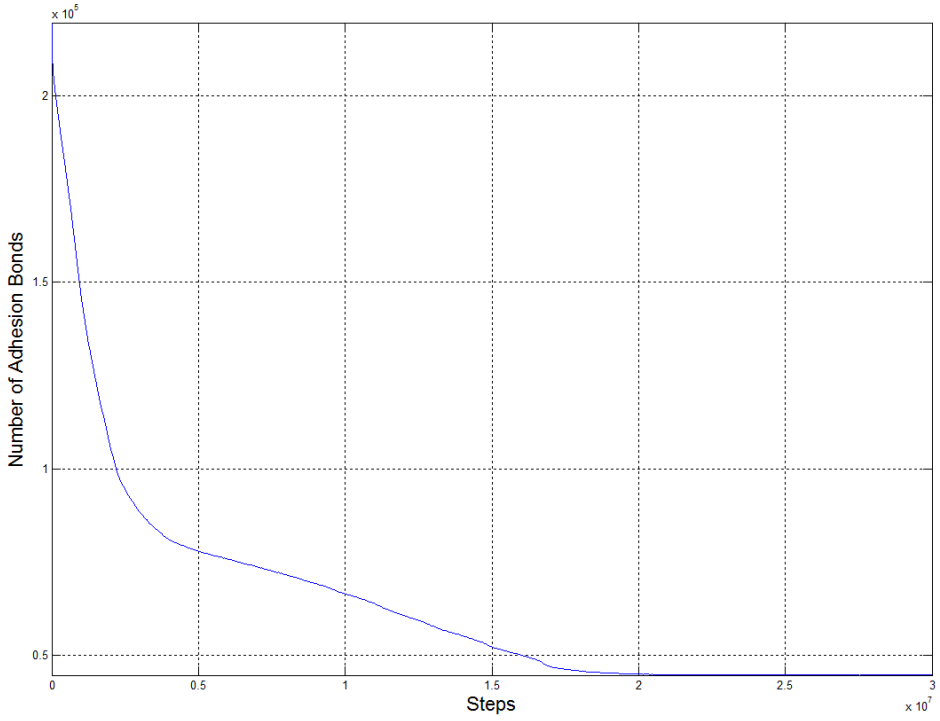


Figure 3.19 125 aggregates in 5-cube formation, $3 \cdot 10^7$ steps, aggregate radius 4, and aggregate distance 1. Energy estimate.

Bibliography

[1] Thomas Boland, William Crisp Wilson Jr., and Tao Xu. Ink-jet printing of viable cells.

[2] A.B. Bortz, M.H. Kalos, and J.L. Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. *Journal of Computational Physics*, 17(1):10 – 18, 1975.

[3] Elijah Flenner, Lorant Janosi, Bogdan Barz, Adrian Neagu, Gabor Forgacs, and Ioan Kosztin. Kinetic monte carlo and cellular particle dynamics simulations of multicellular systems. *Phys. Rev. E*, 85:031907, Mar 2012.

[4] Michael A. Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889, 2000.

[5] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[6] Yoshiki Sasai. Cytosystems dynamics in self-organization of tissue architecture. *Nature*, 493(7432):318–326, 2013/01/17.

[7] Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton. A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction networks. *The Journal of Chemical Physics*, 128(20):–, 2008.

[8] Malcolm S. Steinberg. Reconstruction of tissues by dissociated cells. *Science*, 141(3579):401–408, 1963.

[9] Yi Sun and Qi Wang. Modeling and simulations of multicellular aggregate self-assembly in biofabrication using kinetic monte carlo methods. *Soft Matter*, 9:2172–2186, 2013.

[10] George M. Whitesides and Mila Boncheva. Beyond molecules: Self-assembly of mesoscopic and macroscopic components. *Proceedings of the National Academy of Sciences*, 99(8):4769–4774, 2002.