UNIVERSITY OF CRETE
FACULTY OF SCIENCES AND ENGINEERING
POSTGRADUATE PROGRAM "MATHEMATICS AND ITS APPLICATIONS"
OF MATHEMATICS AND APPLIED MATHEMATICS DEPARTMENTS

Master Thesis

Spatio-Temporal Acceleration of Kinetic Monte Carlo Methods

Giorgos T. Arampatzis
Supervisor: Markos Katsoulakis

HERAKLION
2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
«ΜΑΘΗΜΑΤΙΚΑ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΤΟΥΣ»

Μεταπτυχιακή εργασία

Χωρο-χρονική επιτάχυνση μεθόδων Kinetic Monte Carlo

Γιώργος Θ. Αραμπατζής
Επιβλέπων καθηγητής: Μάρκος Κατσουλάκης

ΗΡΑΚΛΕΙΟ
2011

committee
Katsoulakis Markos
Kosioris Georgios
Makridakis Charalampos

Περίληψη

Το κεντρικό θέμα της εργασίας είναι η μελέτη μεθόδων επιτάχυνσης μοριακών προσομοιώσεων τύπου Kinetic Monte Carlo, σε προβλήματα Θεωρίας Υλικών και Χημικής Μηχανικής. Η βασική μαθηματική και αλγοριθμική δυσκολία σε αυτά τα προβλήματα προέρχεται τόσο από τον τεράστιο αριθμό σωματιδίων, όσο και από τους μακροσκοπικούς χρόνους που πρέπει να προσομοιωθούν και στους οποίους παρατηρούνται ενδιαφέρουσες μεσοσκοπικές και μακροσκοπικές μορφολογίες τεχνολογικού ενδιαφέροντος όπως η συμπύκνωση νανοσωματιδίων, η δημιουργία προτύπων (patterns), και φαινόμενα μετα-ευστάθειας διεπιφανειών.

Το πρώτο τμήμα της εργασίας επικεντρώνεται στην χρονική επιτάχυνση μεθόδων Monte Carlo μέσω της πολύ πρόσφατα προταθείσας μεθόδου τ-leap απο τον Gillespie. Μελετήσαμε για πρώτη φορά στην βιβλιογραφία την μέθοδο σε συστήματα που έχουν αλλαγές φάσεις και δείξαμε τόσο αριθμητικά άλλα και αναλυτικά σε συγκεκριμένα παραδείγματα ότι η μέθοδος τ-leap είναι ιδιαίτερα ευαίσθητη στην επιλογή του χρονικού παράθυρου που περιγράφεται από την παράμετρο τ και σε προβλήματα με αλλαγές φάσεις μπορεί να δώσει λανθασμένες προβλέψεις.

Στο δεύτερο τμήμα της εργασίας αναπτύσσεται μια νεοτεριστική μέθοδος για την παραλληλοποίηση αλγορίθμων Kinetic Monte Carlo. Η προτεινόμενη μέθοδος βασίζεται στην ανάπτυξη μιας ιεραρχικής αναπαράστασης της γεννήτριας που αντιστοιχεί στον αλγόριθμο, η οποία επιτρέπει την συστηματική και ισόρροπη διαμέριση (load balancing) του υπολογιστικού έργου σε ανεξάρτητους υπολογιστικούς επεξεργαστές. Η μέθοδος υλοποιήθηκε σε προβλήματα στατιστικής φυσικής τα οποία έχουν ακριβείς, αναλυτικές λύσεις (πχ μοντέλο Ising σε μία και δύο διαστάσεις) που επέτρεψαν την αυστηρή πιστοποίηση του αλγορίθμου.

Η επιτάχυνση σε σχέση με αντίστοιχους σειριακούς αλγορίθμους μπορεί να φτάσει σε πάνω από τέσσερις τάξεις μεγέθους, σε υλοποιήσεις σε κάρτες γραφικών (GPU). Κατά συνέπεια αναμένουμε ότι η νέα αυτή μέθοδος, που ονομάσαμε Fractional Step Kinetic Monte Carlo (FS-KMC), να δώσει τη δυνατότητα για πρώτη φορά για προσομοίωση μοριακών μοντέλων στην ετερογενή κατάλυση σε ρεαλιστικές διαστάσεις αντιδραστήρα τάξης μεγέθους mm. Η μέθοδος FS-KMC επίσης επιτρέπει την παράλληλη προσομοίωση μοριακών συστημάτων με πολλαπλούς μικροσκοπικούς μηχανισμούς πολλαπλών χρονικών κλιμάκων όπως πχ μοριακά συστήματα διάχυσης/αντίδρασης.

Abstract

The main subject of this thesis is the study of molecular simulation accelerations methods of Kinetic Monte Carlo type in Material Science and Chemical Engineering problems. The main mathematical and algorithmic difficulty in these problems comes from the large number of particles as well as from the macroscopic times that must be simulated and in which interesting mesoscopic and microscopic phenomena take part like nanoparticles concentration, pattern formation and interface metastability phenomena.

In the first part of the thesis we study the temporal acceleration of Monte Carlo methods through the, lately proposed by Gillespie, τ-leap method. For the first time in the bibliography we studied this method in phase transition systems and we showed analytically and numerically in specific examples that this method is particularly sensitive in the selection of time window described by the τ parameter.

In the second part a new method is developed for the parallelization of Kinetic Monte Carlo algorithms. The proposed method is based on the development of a hierarchical representation of the generator which allows a systematic and equal work load balance into independent processing units. The algorithm is implemented in statistical physics problems that have analytical solutions (e.g. Ising model in 1 and 2 dimensions) that allowed the rigorous certification of the algorithm.

The acceleration compared with equivalent serial algorithms can reach in over four orders of magnitude, in implementations on graphics cards (GPU). Therefore we expect that this new method, called Fractional Step Kinetic Monte Carlo (FS-KMC), to give the opportunity, for the first time, of the simulation of molecular models in heterogeneous catalysis under realistic reactor dimensions of magnitude mm. The FS-KMC method also allows the parallel simulation of molecular systems with multiple microscopic mechanisms of multiple time scales e.g. molecular systems of diffusion/reaction.

Contents

1 Introduction

The main purpose of this work is the presentation and study of models that describe many body interacting particle systems. These models find applications in material science, where the particles may be atoms of a magnetic material that interact under the presence of an external field, or in chemistry where the particles may be molecules of different species that react under specific chemical equations.

The models are divided in three categories and the criterion is the scale under which they observe the physical system. The *microscopic* models include the biggest amount of information because they observe the system in the atomistic level. The main disadvantage is that the numerical simulation is extremely slow. In the limiting situation, where the number of particles tends to infinity, we get *deterministic* models for observables such as coverage. The solution of these models is much easier than the microscopic models but a large amount of information is lost. In between of these two classes of models we find the *mesoscopic* models. One part of these models is the same with the deterministic models and the other part is a stochastic term that mimics the randomness of the microscopic models. The difficulty in their simulation, as well as the quality of the information they give us, is in between the two first.

We will present results from all the above models, but particular emphasis will give to the first category. These models, as we will see in more details later, describe the system through high dimensional Ordinary Differential Equations. A typical example is the *Master Equation*

$$\frac{\partial}{\partial t}P(\sigma,t)=\sum_{\sigma'\neq\sigma}a(\sigma,\sigma')P(\sigma',t)-a(\sigma',\sigma)P(\sigma,t), \tag{1}$$

where $P(\sigma,t)$ is the probability density function that describe the probability of the system to be at discrete state σ in time t [7]. The dimension of the above PDEs, that is equal to the number of all possible states of the system, as well as the complexity of the right hand side, make impossible not only the the analytical but also the numerical solution of these.

One possible treatment is to construct a Continuous Time Markov Chain, a stochastic process with specific properties, that for large enough times produces samples from the probability distribution function (1). Having enough samples and applying appropriate statistical methods we are able to reproduce P. One of the advantages of this approach is that it is independent of the system's dimension. Another advantage is that it gives us information not only for the equilibrium of the system but for its dynamical evolution as well. The main disadvantage is that we must have a lot of samples in order to study the statistical properties of the system.

The Monte Carlo (MC) methods is a family of computational algorithms for the simulation of stochastic systems. The first MC algorithm, the Metropolis algorithm, first described in a 1953 paper by Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and

Edward Teller, was cited in "Computing in Science and Engineering" as being among the top 10 algorithms having the "greatest influence on the development and practice of science and engineering in the 20th century" [19]. When we are interested not only in the equilibrium states of the system but for its dynamic evolution as well, a subcategory of the above methods is used, known as kinetic Monte Carlo (kMC).

Gillespie, in 1976, was the first that described an algorithm for the simulation of well mixed chemical reaction systems [8]. This algorithm is known as the Gillespie algorithm or Stochastic Simulation Algorithm (SSA). In order to describe the algorithm we have to give some details on the model describing such systems. For a system being in state σ, we define the rate, the probability per unit time, by which the system jumps from state σ to state σ'. We will denote this function by $a(\sigma, \sigma')$ and its the same found in equation (1). Starting from state σ we compute $r_{\sigma'} = a(\sigma, \sigma')$ for every σ' in the set of all possible states starting from σ and

$$R_{\sigma'} = \sum_{\sigma < \sigma'} r_\sigma,$$

setting $R_0 = \sum_\sigma r_\sigma$. Here we assumed that the states of the system can be put in order, which is true in our case since we stude systems with finite states. In order to find the next state of the system we produce a random number distributed uniformly in $[0, 1]$, $u \sim \mathcal{U}([0, 1])$, and search for the state σ that satisfies

$$\frac{R_{\sigma-1}}{R_0} \leq u < \frac{R_\sigma}{R_0}.$$

The time spent the system in state x is a random variable following exponential distribution with mean value $\frac{1}{R_0}$.

Even though the Gillespie algorithm is exact its main disadvantage is that it is extremely slow for large systems. The problems are 1) to find the in which interval u belongs and b) if R_0 is large enough, the time spent by the system in every state may be relatively small. This means that the algorithm needs a lot of steps in order to reach a final time $T >> 1$.

One solution to the first problem is the algorithm developed by Bortz, Kalos and Lebovitz and is know as the BKL algorithm (or N-fold) [2]. This algorithm sorts the states in equivalent classes having the same rate creating this way less intervals and so less search time. The problem still remains when the system has a lot of classes (e.g. in the case of long range interaction systems).

The τ-leap algorithm, given by Gillespie, is a new algorithm proposed to give a solution to the second problem [9]. It uses a deterministic time step τ, which must be larger than the mean stochastic time increment $\frac{1}{R_0}$. Then, in a way we will discuss in section 3.2, the algorithm finds approximately the state of the system after time τ. The difference with the other algorithms is that this is not an exact algorithm and in many cases the constraints in the choice of τ are so stiff that the criterion $\tau > \frac{1}{R_0}$ cannot be satisfied.

Many works followed the first article of Gillespie on τ-leap. Some of them deal with the issue of the selection of τ and particularly with the adaptive fitting [14]. Interesting

works have been done in the way of selecting the state of the system after time τ because the classic algorithms many times leads to unaccepted state (e.g. negative concentrations in chemical reaction models) [4]. Tiejun Li proved [23], by writing this method as a classical Euler method, that as τ goes to zero the error of the method goes to zero. But as mentioned before, τ should be larger than the mean stochastic time step and Li's result does not give us useful information in that sense. On the other hand Anderson et al. showed that under a suitable rescaling of the system length scales, τ can be selected in such a way such that not only convergence is satisfied but the demand $\tau > \frac{1}{R_0}$ as well [?].

During the past two decades, in step with the rapid growth of the computational systems, big importance was given in the development of parallel algorithms for the simulation of these models. One first approach to the parallel treatment of Kinetic Monte Carlo methods is to run every algorithm in a different processor, producing this way many realizations of the same experiment and increasing the size of the statistical data. Although this approach is extremely useful, it does not give a solution to the problem of simulating a model for large times. In this case one must split the problem into smaller parts, distribute these parts into several processors, solve the small problem locally and then reproduce the solution by collecting the parts from every node.

Here we will present the most important parallel Kinetic Monte Carlo algorithms, from Lubachevsky's first idea until today. The algorithms will be explained on the Ising model: on a lattice $\Lambda_N \subset \mathbb{N}^2$ we define a set of possible states $\Sigma \subset \{0,1\}^N$, where N is the size of the lattice. The system goes from the state σ to the state σ^x with transition rate $c(x,\sigma)$, where $\sigma^x(y) = \sigma(y), y \neq x$ and $\sigma^x(x) = 1 - \sigma(x)$, and remains in this state time amount that follows exponential distribution with mean

$$\frac{1}{\lambda(\sigma)} = \frac{1}{\sum_{x\in\Lambda_N} c(x,\sigma)} \tag{2}$$

The transition from σ to σ^x is done with probability

$$p(\sigma,\sigma^x) = \frac{c(x,\sigma)}{\lambda(\sigma)} \tag{3}$$

The difficulty in parallelizing this model is that the transition clock depends on the whole lattice.

In 1987, Lubachevsky proposed an algorithm for the solution of this problem. The idea is that the lattice is split into smaller parts and every processor simulates only one part [18]. Every part is divided into boundary and interior cells. When an action is to happen in the interior it executes instantly but when it is to happen in the boundary cells then the processor must communicate with its neighbor processor and be sure they are synchronized at the same time. This must be done in order to preserve causality in the system. In this algorithm constant rates are assumed through the whole lattice in order every processor to be in the same time window so that the synchronization waits to be as

small as possible. In [16] this idea was efficiently implemented using the N-fold algorithm as the kernel for every processor instead of the SSA.

A generalization, in the case of non constant rates, was proposed in [20] using the idea of uniformization. A constant rate $\lambda^* = \max_{\sigma\in\Sigma} \lambda(\sigma)$ is chosen and the system goes from state σ to state σ^x with probability $\frac{\lambda(\sigma)}{\lambda^*} p(\sigma, \sigma^x)$ and stays at the same state with probability $1 - \frac{\lambda(\sigma)}{\lambda^*}$. This approach introduces rejections to the algorithm reducing its efficiency.

The above algorithms are known as *asynchronous* because every processor advances its subsystem in a time horizon different than the other processors. The two subcategories are the *conservative* and the *optimistic* algorithms. In the first category when a boundary event is to be carried out the processor waits for its neighbors in order to be at the same time, while in the other case the processor advances the system until a fixed time horizon and then, after the communication, goes back in time and resolves any possible conflicts (rollbacks). A more detailed description can be found in [20].

The other category of parallel algorithms is known as *synchronous*. In this case every processor has some ghost cells in order to keep some of the information from its neighbors. Then advances the system until a fixed time T, and after that they communicate in order to keep the ghost cells informed [21]. These algorithms introduce errors due to inconsistencies at the boundary cells but due to the flexibility in choosing T this error can be as small as we want.

Here we propose a formulation under which the synchronous methods can be proved that converge to the true solution. The idea comes from the Operator Splitting theory that has been successfully applied to the solution of Partial Differential Equations [13]. Our methodology relies on first developing a spatio-temporal decomposition for the Markov operator underlying the Kinetic Monte Carlo algorithm, into a hierarchy of operators, corresponding to the processor architecture. Based on this operator decomposition, we formulate Fractional Step Approximation schemes by employing the Trotter product formula. The idea will be explained in more details in the last section and very promising results will be presented [1].

2 Models and applications

2.1 The Ising model

2.1.1 Description

The Ising type models are microscopic stochastic models defined on a periodic cubic d-dimensional lattice of size $|\mathcal{L}| = N = n^d$, which we denote by $\mathcal{L} = \{x \in \mathbb{Z}^d; 0 \leq x_i \leq n-1\}$. At each lattice site $x \in \mathcal{L}$ an order parameter is allowed to take the values 0 and 1 describing vacant and occupied sites respectively. We refer to the order parameter as spin [10]. A spin configuration σ is an element of the configuration space $\Sigma = \{0,1\}^{|\mathcal{L}|}$ and we write $\sigma = \{\sigma(x) : x \in \mathcal{L}\}$, denoting by $\sigma(x)$ the spin at x. The energy H of the system, evaluated at σ, is given by the Hamiltonian

$$H(\sigma) = -\frac{1}{2}\sum_{x\in\mathcal{L}}\sum_{y\neq x} J(x,y)\sigma(x)\sigma(y) + \sum_{x\in\mathcal{L}} h\sigma(x) \tag{4}$$

where h is the external field and J is the inter-particle potential.

The stochastic process $\{\sigma_t\}_{t\geq 0}$ is a continuous time jump Markov process on $L^\infty(\Sigma, \mathbb{R})$ with generator [15]

$$Lf(\sigma) = \sum_{x\in\mathcal{L}} c(x,\sigma)(f(\sigma^x) - f(\sigma)) \tag{5}$$

for ay test function $f \in L^\infty(\Sigma, \mathbb{R})$. See also appendix B for the relation between this operator and the master equation [7]. Here $c(x,\sigma)$ denotes the rate of a spin flip at x for the configuration σ and σ^x signifies the configuration after a flip at x,

$$\sigma^x = \begin{cases} 1-\sigma(x), & \text{if } y = x \\ \sigma(y), & \text{if } y \neq x \end{cases} \tag{6}$$

In this model we implement spin flip Arrhenius dynamics. Under this type of mechanism the simulation is driven based on the energy barrier a particle has to overcome in flipping from one state to another site given by (9). The Arrhenius spin flip rate $c(x,\sigma)$ at lattice site x is given by

$$c(x,\sigma) = \begin{cases} c_d e^{-\beta U(x)}, & \text{when } \sigma(x) = 0 \\ c_a, & \text{when } \sigma(x) = 1 \end{cases} \tag{7}$$

with adsorption/desorption constants

$$c_a = c_d = \frac{1}{\tau_1} \tag{8}$$

and τ_1 denoting the characteristic time of the stochastic process. By β we denote the quantity $\frac{1}{kT}$, where T is the temperature and k the Boltzmann constant. Here U is the

total energy contribution from the particle interactions with the particle located at the site $x \in \mathcal{L}$

$$U(x) = \sum_{y \neq x} J(x,y)\sigma(y) - h \tag{9}$$

Another possible choice for the rates are the Glauber dynamics [5]. In this case the rate is defined by

$$c(x,\sigma) = \frac{e^{-\beta U(x)\sigma(x)}}{e^{-\beta U(x)} + e^{\beta U(x)}} = \frac{1}{2}\Big(1 - \sigma(x)\tanh(\beta U(x))\Big) \tag{10}$$

2.1.2 Canonical Gibbs measures and reversibility

The flip rates (7) and (10) have built in invariance of the Gibbs measure (see [10])

$$\mu(d\sigma) = \frac{1}{Z} e^{-\beta H(\sigma)} P_N(d\sigma), \tag{11}$$

since they satisfy the detailed balance condition [10],

$$c(x,\sigma)e^{-\beta H(\sigma)} = c(x,\sigma^x)e^{-\beta H(\sigma^x)} \tag{12}$$

Here P_N denotes the product prior distribution on $\mathcal{L}$ (the distribution for infinite temperature),

$$P_N(d\sigma) = \prod_{x \in \mathcal{L}} \rho(d\sigma(x)) = \left(\frac{1}{2}\right)^N, \tag{13}$$

and $\rho(\sigma(x) = 0) = \rho(\sigma(x) = 1) = \frac{1}{2}$. Here Z is the partition function,

$$Z = \sum_{\sigma \in \Sigma} e^{-\beta H(\sigma)} \tag{14}$$

guaranteeing $\mu(\Sigma) = 1$.

2.2 Nearest Neighbour Ising Model

In this case the inter-particle potential is defined as

$$J(x,y) = J(|x-y|) = \begin{cases} J_0, & \text{when } |x-y| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Then the Hamiltonian of the system is simplified into

$$H(\sigma) = -\frac{1}{2}\sum_{x \in \mathcal{L}} J_0\Big(\sigma(x-1) + \sigma(x+1)\Big)\sigma(x) + \sum_{x \in \mathcal{L}} h\sigma(x) \tag{16}$$

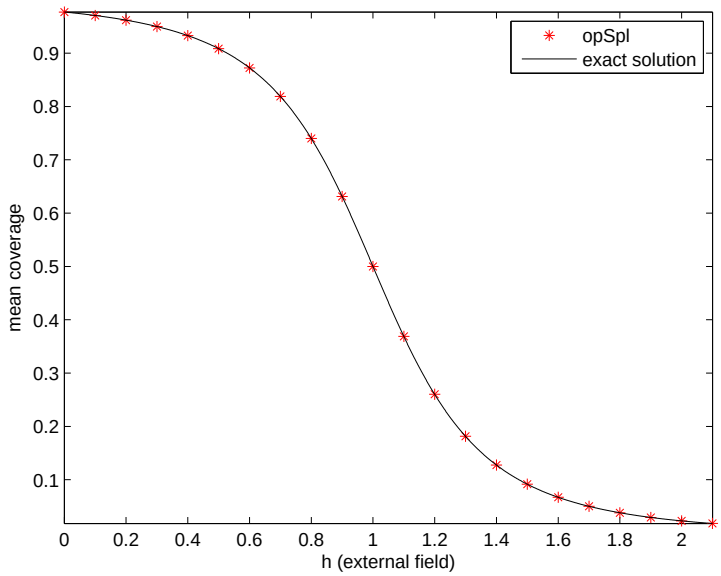

Figure 1: *Comparison of the 1D exact solution for the mean magnetization for the Ising nearest neighbour model with the solution obtained from the Fractional Step algorithm with* $\Delta t = 1$*. Here (b,J)=(2,1), N=1000, T=1000*

2.2.1 Exact solutions

Exactly solvable models of statistical mechanics provide a test bed for sampling algorithms applied to interacting particle systems. In [6] one can find the exact solutions for the mean magnetization which is defined as

$$< m >= \int_{\Sigma} m(\sigma)\mu(d\sigma) \tag{17}$$

where m is the coverage or the magnetization of the lattice,

$$m(\sigma) = \frac{1}{N}\sum_{x\in\mathcal{L}} \sigma(x). \tag{18}$$

for a system with Hamiltonian,

$$H(\sigma) = -\frac{J}{2}\sum_{x\in\mathcal{L}}\sum_{y\in N_x} \sigma(y)\sigma(x) - h\sum_{x\in\mathcal{L}} \sigma(x), \tag{19}$$

and $\sigma \in \{-1, 1\}$. You can see appendix A how to transform the Hamiltonian from $\{-1, 1\}$ to $\{0, 1\}$.

The mean magnetization for the one dimensional model is given by

$$m = \frac{\sinh(\beta h)}{\sqrt{\sinh^2(\beta h) + e^{-4\beta h}}} \tag{20}$$

and for the two dimensional case, known as the Onsager solution, when $h = 0$

$$m = \begin{cases} 0, & \beta < \beta_c \\ \left(1 - \sinh^{-4}(2\beta J)\right)^{1/8}, & \beta > \beta_c \end{cases} \tag{21}$$

where β_c is the critical value for β in which phase transition occur and its value is given by the solution of the equation

$$\sinh(\beta_c J) = 1. \tag{22}$$

In figures 1 and 2 we see the mean magnetization for the 1d and 2d model respectively. Note that in the 1d model no phase transition effects take part, while in the 2d case for $\beta = \beta_c$ you can observe the non smooth transition in the magnetization. However when the range of the inner particle potential is increased phase transition is observed in the 1d model, as we will see later.

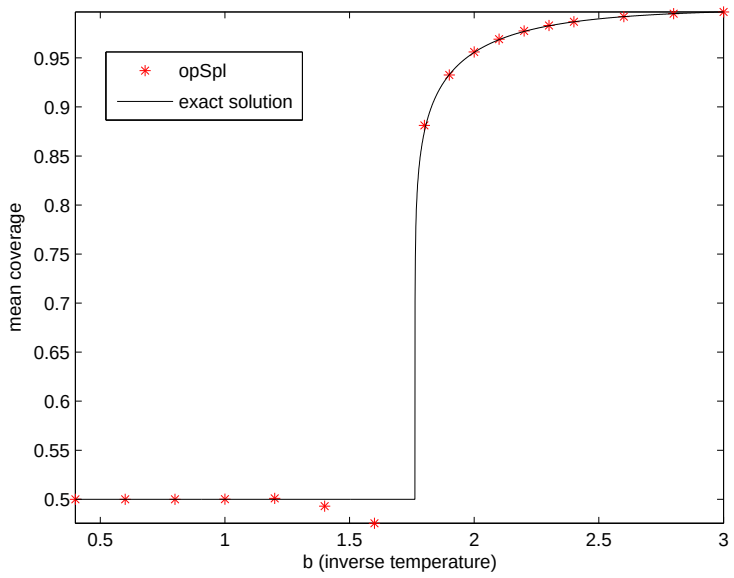

Figure 2: *Comparison of the 2d exact solution for the mean magnetization for the Ising nearest neighbor model with the solution obtained from the Fractional Step algorithm with* $\Delta t = 1$*. Here (J,h)=(1,2),* N*=*100^2*,* T*=100*

Finally, in one dimension, the exact solution for the spatial autocorrelation function (for free boundary conditions), defined by

$$G(x, y) = \mathbb{E}\Big[\Big(\sigma(x) - \mathbb{E}\sigma(x)\Big)\Big(\sigma(y) - \mathbb{E}\sigma(y)\Big)\Big] = \mathbb{E}[\sigma(x)\sigma(y)] \tag{23}$$

is known for $h = 0$ and is given by [10]

$$G(x, y) = \tanh(\beta J)^{|x-y|} \tag{24}$$

Although this formula is exact for free boundary conditions, it can be used for large systems with periodic boundary conditions since the partition functions for these models coincide (for more details on this see [10] §3.14).

Note that for $T \neq \infty$ we have that $\tanh(\beta J) < 1$ and $\coth(\beta J) > 1$ thus $\log \coth(\beta J) > 0$. Therefore, for $y > 0$ we can write

$$G(x, x+y) = e^{-y \log \coth(\beta J)} \tag{25}$$

showing that correlations decay exponentially.

2.3 The Curie Weiss model

2.3.1 Description

The Curie-Weiss model is a special case of the Ising model where the inter-particle potential $J(x, y)$ is constant and equal to $\frac{J_0}{N}$.

The microscopic Hamiltonian (4) can be written in terms of the new variable $\eta = m(\sigma)$ (see eq. (18)) as,

$$\begin{aligned} H(\sigma) &= -\frac{1}{2}\sum_{x\in\mathcal{L}} \sigma(x)\frac{J_0}{N}\left(\sum_{y\neq x}\sigma(y) - \sigma(x)\right) + Nh\eta \\ &= -\frac{J_0}{2}\sum_{x\in\mathcal{L}} \sigma(x)\frac{1}{N}\left(N\eta - \sigma(x)\right) + Nh\eta \\ &= -\frac{J_0}{2}\left(\eta\sum_{x\in\mathcal{L}}\sigma(x) - \frac{1}{N}\sum_{x\in\mathcal{L}}\sigma^2(x)\right) + Nh\eta \\ &= \frac{J_0}{2}\left(N\eta^2 - \frac{1}{N}\sum_{x\in\mathcal{L}}\sigma^2(x)\right) + Nh\eta. \end{aligned}$$

Since $\sigma \in \{0,1\}^N$, we get $\sum_{x\in\mathcal{L}}\sigma^2(x) = \sum_{x\in\mathcal{L}}\sigma(x) = \eta$ and finally the coarse grained Hamiltonian is given by

$$\tilde{H}(\eta) = N\eta\left(\frac{J_0}{2}(\eta - \frac{1}{N}) + h\right) = N\bar{H}(\eta). \tag{26}$$

Now the spin flip rates (7) reads as,

$$c(x,\sigma) = (1-\sigma(x)) + \sigma(x)e^{-\beta\tilde{U}(\eta)} \tag{27}$$

where $\tilde{U}(\eta) = J_0(\eta - \frac{1}{N}) - h$.

We can rewrite the generator (5), in terms of the variable η, by introducing the test function

$$f(\sigma) := g(F(\sigma)) = g(\eta), \quad F(\sigma) = \frac{1}{N}\sum_{x\in\mathcal{L}}\sigma(x) \tag{28}$$

for any test function $g \in L^\infty(\Sigma, \mathbb{R})$. Then

$$\begin{aligned} F(\sigma^x) &= \frac{1}{N}\sum_{y\in\mathcal{L}} \sigma^x(y) = \frac{1}{N}\left(\sum_{y\in\mathcal{L}} \sigma(y) + 1 - 2\sigma(x)\right) \\ &= \begin{cases} F(\sigma) + \frac{1}{N}, & \text{when } \sigma(x) = 0 \\ F(\sigma) - \frac{1}{N}, & \text{when } \sigma(x) = 1 \end{cases} \end{aligned}$$

and

$$f(\sigma^x) = g(F(\sigma^x)) = \begin{cases} g(\eta + \frac{1}{N}), & \text{when } \sigma(x) = 0 \\ g(\eta - \frac{1}{N}), & \text{when } \sigma(x) = 1 \end{cases} \tag{29}$$

Then the second term in the sum in equation (4) becomes

$$\begin{aligned} f(\sigma^x) - f(\sigma) &= g(\eta + \frac{1}{N})(1 - \sigma(x)) + g(\eta - \frac{1}{N})(1 - \sigma(x)) - g(\eta) \\ &= \left(g(\eta + \frac{1}{N}) - g(\eta)\right)(1 - \sigma(x)) + \left(g(\eta - \frac{1}{N}) - g(\eta)\right)\sigma(x). \end{aligned} \tag{30}$$

Using these equations we can write the coarse grained generator

$$\tilde{L}g(\eta) = c_a(\eta)\left(g(\eta + \frac{1}{N}) - g(\eta)\right) + c_d(\eta)\left(g(\eta - \frac{1}{N}) - g(\eta)\right) \tag{31}$$

where $c_a(\eta) = N(1 - \eta)$ and $c_d = \eta N e^{-b\tilde{U}(\eta)}$.

2.3.2 Invariant measure and asymptotics

Let $\tilde{\mu}$ denote the invariant measure for the coarse grained dynamics. Then the detailed balance for this measure dictates

$$c_d(\eta)\tilde{\mu}(\eta) = c_a(\eta - \frac{1}{N})\tilde{\mu}(\eta - \frac{1}{N}) \tag{32}$$

or

$$\tilde{\mu}(\eta) = \tilde{\mu}(\eta - \frac{1}{N})\frac{c_a(\eta - \frac{1}{N})}{c_d(\eta)} = ... = \tilde{\mu}(0)\prod_{i=1}^{\eta N}\frac{c_a(\frac{i-1}{N})}{c_d(\frac{i}{N})} \tag{33}$$

By doing all the calculations we get

$$\tilde{\mu}(\eta) = \frac{1}{\tilde{Z}}e^{-\beta\tilde{H}(\eta)}\tilde{P}(\eta) \tag{34}$$

where

$$\tilde{P}(\eta = \frac{k}{N}) = \binom{N}{k}\left(\frac{1}{2}\right)^N = \frac{N!}{k!(N-k)!}\left(\frac{1}{2}\right)^N \tag{35}$$

and $\tilde{Z}$ is the partition function for the coarse grained measure. We can use Stirling's formula [3] to write

$$\binom{N}{k} = e^{\log\binom{N}{k}} = e^{-N(\eta\log\eta+(1-\eta)\log(1-\eta))+o(N)} \tag{36}$$

where $k = \eta N$.Then equation (34) transforms into

$$\tilde{\mu}(\eta) = \frac{1}{\tilde{Z}} e^{-N\left(\beta\bar{H}(\eta)+\eta\log\eta+(1-\eta)\log(1-\eta)\right)+o(N)} \tag{37}$$

2.3.3 Exact solutions

For this model, with Hamiltonian

$$\tilde{H}(\sigma) = -\frac{\tilde{J}}{N-1}\sum_{x\in\mathcal{L}}\sum_{y\neq x}\tilde{\sigma}(x)\tilde{\sigma}(y) - \tilde{h}\sum_{x\in\mathcal{L}}\tilde{\sigma}(x) \tag{38}$$

and $\sigma(x) \in \{-1, 1\}$, the mean magnetization as $N \to \infty$, is given by [6]

$$\tilde{m} = \tanh(2\beta\tilde{J}\tilde{m} + \beta\tilde{h}) \tag{39}$$

or, solving for the external field

$$\tilde{h} = -2\tilde{J}\tilde{m} + \frac{1}{2\beta}\ln\frac{1+\tilde{m}}{1-\tilde{m}}. \tag{40}$$

In order to transform the above equation for our system, with Hamiltonian given by (4) and $\sigma(x) \in \{0, 1\}$, we have to substitute $\tilde{J} = J/8$, $\tilde{h} = J/4 - h/2$ and $\tilde{m} = 2m - 1$ (see appendix A for more information about this transformation). Then the external field in terms of the coverage, is given by

$$h = Jm - \frac{1}{\beta}\ln\frac{m}{1-m}. \tag{41}$$

As mentioned in previous section, the Ising model with nearest neighbor interactions has no phase transitions. In the C-W model the interactions cover the whole lattice. In this case, for $\beta J > 4$ phase transition occur. In figure (3) you can see how the mean coverage looks like for different values of βJ.

2.4 The Chemical Reactions Model

2.4.1 Description

The time evolution of a spatially homogeneous mixture of chemically reacting molecules is usually calculated by solving a set of coupled ordinary differential equations [8]. If there

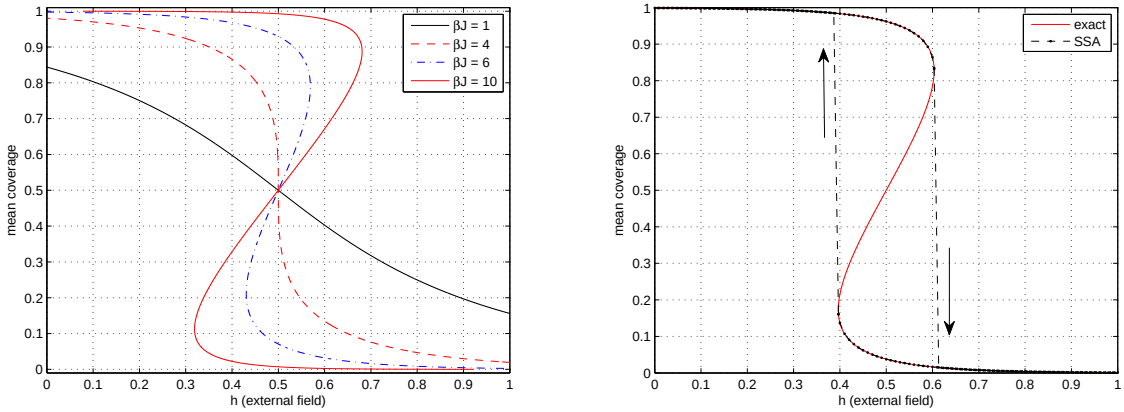

Figure 3: *Mean coverage for the mean field Curie-Weiss model. You can observe phase transitions for $\beta J > 4$ (left). The results from the SSA algorithm are in good agreement with the exact solution and the hysteresis phenomenon is clearly observed for $b = 7,\ J = 1\ N = 10^6$ (right).*

are N chemically active molecular species present, there will be N differential equations in the set. Another approach to the chemical kinetics of such systems is the stochastic formulation. Here the reaction rates are viewed as reaction "probabilities per unit time" and the temporal behavior of the system takes the form of a Markovian random walk on the N-dimensional space of the molecular populations of the N species.

The general problem may be formulated as follows: In a given volume V there are molecules of N chemically active species S_i and possibly species of inert molecules as well. By $X_i(t)$ we denote the current number of molecules of the chemical species S_i, $i = 1, ..., N$ at time t. Moreover these N species react through M chemical reactions R_i, each characterized by a numerical reaction parameter c_i. We will only examine reactions of the following general types

$$
\begin{aligned}
* &\rightarrow \text{reaction products} && \text{(42a)}\\
S_i &\rightarrow \text{reaction products} && \text{(42b)}\\
S_i + S_j &\rightarrow \text{reaction products } (i \neq j) && \text{(42c)}\\
2S_i &\rightarrow \text{reaction products} && \text{(42d)}\\
S_i + S_j + S_k &\rightarrow \text{reaction products } (i \neq j \neq k) && \text{(42e)}\\
S_i + 2S_j &\rightarrow \text{reaction products } (i \neq j) && \text{(42f)}\\
3S_i &\rightarrow \text{reaction products} && \text{(42g)}
\end{aligned}
$$

The first reaction type denotes a spontaneous creation of one or more of the species S_i. The reaction products may contain none, one or more of the chemical species S_i.

For every reaction we define the *propensity function*

$$a_i(\mathbf{x})dt \quad := \quad \text{the probability given } \mathbf{X}(\mathrm{t}) = \mathbf{x} \text{ that one } R_i$$

$$\begin{aligned} &\text{reaction will occur somewhere inside } V \text{ in the} \\ &\text{next infinitesimal time interval } [t, t+dt],\ i = 1, ..., M \end{aligned} \tag{43}$$

where $\mathbf{X}(t) = (X_1(t), ..., X_N(t))$. Also let us define the *state-change vector* $\boldsymbol{\nu}_i$, whose jth component is defined by

$$\begin{aligned} \nu_{ij} \quad = \quad &\text{the change in the number of } S_j \text{ molecules} \\ &\text{produced by an } R_i \text{ reaction} \end{aligned} \tag{44}$$

and the matrix $\boldsymbol{\nu} = (\boldsymbol{\nu}_1, ..., \boldsymbol{\nu}_M)$. The above definitions completely determine the chemical model. For every reaction type in (42) the propensity function is

$$a(\mathbf{x}) = c \tag{45a}$$
$$a(\mathbf{x}) = cX_i \tag{45b}$$
$$a(\mathbf{x}) = cX_iX_j \tag{45c}$$
$$a(\mathbf{x}) = cX_i(X_i - 1)/2 \tag{45d}$$
$$a(\mathbf{x}) = cX_iX_jX_k \tag{45e}$$
$$a(\mathbf{x}) = cX_iX_j(X_j - 1)/2 \tag{45f}$$
$$a(\mathbf{x}) = cX_iX_j(X_j - 1)(X_j - 2)/6 \tag{45g}$$
$$\tag{45h}$$

where c is the reaction parameter.

Having these in mind we are interested in the following probability

$$P(\mathbf{x}, t|\mathbf{x}_0, t_0) = \text{probability that } \mathbf{X}(t) \text{ will equal } \mathbf{x} \tag{46}$$
$$\text{given that } \mathbf{X}(t_0) = \mathbf{x}_0 \tag{47}$$

The above probability density function obeys the following differential equation, known as *chemical master equation*

$$\frac{\partial}{\partial t}P(\mathbf{x}, t|\mathbf{x}_0, t_0) = \sum_{j=1}^{M} a_j(\mathbf{x} - \boldsymbol{\nu}_j)P(\boldsymbol{x} - \boldsymbol{\nu}_j, t|\mathbf{x}_0, t_0) - a_j(\mathbf{x})P(\mathbf{x}, t|\mathbf{x}_0, t_0) \tag{48}$$

with initial condition,

$$P(\mathbf{x}, t_0|\mathbf{x}_0, t_0) = \delta(\mathbf{x} - \mathbf{x}_0) = \begin{cases} 1, & \mathbf{x} = \mathbf{x}_0 \\ 0, & \mathbf{x} \neq \mathbf{x}_0 \end{cases} \tag{49}$$

2.4.2 Example : The Schlögl model

The Schögl model is a simple autocatalytic, trimolecular reaction scheme. It is very interesting because for specific values of the parameters exhibits bistability. Due to its

simplicity it is widely used as a test problem. This model was first analyzed by Schlögl in 1972 (see [11],[24]). The reactions are

$$\begin{aligned} A + 2X &\underset{c_2}{\overset{c_1}{\rightleftarrows}} 3X \\ B &\underset{c_4}{\overset{c_3}{\rightleftarrows}} X \end{aligned}$$

We assume that the number of A and B molecules are maintained constant and we track only the X molecules. Using eq. (45) we get the propensities vector,

$$\begin{pmatrix} a_1(x) \\ a_2(x) \\ a_3(x) \\ a_4(x) \end{pmatrix} = \begin{pmatrix} c_1 a x(x-1)/2 \\ c_2 x(x-1)(x-2)/6 \\ c_3 b \\ c_4 x \end{pmatrix} \tag{50}$$

where a, b and x are the numbers of molecules of A, B and X respectively. The state change matrix is

$$\boldsymbol{\nu} = [\,1 \quad -1 \quad 1 \quad -1\,] \tag{51}$$

3 Temporal acceleration

In this section we will describe the algorithms we will use to solve the models of the previous chapter. First we will see in detail the two basic algorithms for the solution of the microscopic models, the Stochastic Simulation Algorithm (SSA) and the tau-leap ([9], [4]). Then we will give a short description of the Euler-Maruyama scheme for the solution of the SDE approximation of the above models. Finally the mean field approximation of these will be examined. We will check the validity of these algorithms by comparing the empirical histograms with the exact measures, obtained in the previous section.

3.1 Stochastic Simulation Algorithm (SSA)

For the master equation (48) define the quantity

$$a_0(\boldsymbol{x}) = \sum_{j=1}^{M} a_j(\boldsymbol{x}) \tag{52}$$

Then $a_0(\boldsymbol{x})dt$ is the probability that some reaction will occur in the next dt and $e^{a_0(\boldsymbol{x})dt}$ is the probability that a time dt will elapse without any reaction firing. Having this and definition (43) it can be shown that that the *next reaction density function*

$$\begin{aligned} p(\tau, j|\boldsymbol{x}, d\tau) \quad := \quad & \text{probability that, given } \boldsymbol{X}(t) = \boldsymbol{x} \text{ the next reaction in } \Omega \\ & \text{will occur in the time interval } [t+\tau, t+\tau+d\tau] \text{ and will} \\ & \text{be an } R_j \text{ reaction} \end{aligned} \tag{53}$$

is equal to

$$p(\tau, j|\boldsymbol{x}, t) = a_j(\boldsymbol{x}) e^{a_0(\boldsymbol{x})\tau}. \tag{54}$$

Algorithm 1 SSA

Require: $T, \boldsymbol{x}_0$
  $\boldsymbol{x} = \boldsymbol{x}_0$
  while $t \leq T$ **do**
    sample r from $\mathcal{U}([0,1])$
    $j \leftarrow \min\{k | \sum_{i=1}^{k} a_i(\boldsymbol{x}) > r a_0(\boldsymbol{x})\}$
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{\nu}_j$
    sample dt from $\mathcal{E}(a_0(\boldsymbol{x}))$
    $t \leftarrow t + dt$
  end while

As you can see, algorithm (1) is a very simple procedure. It requires only two samples from continuous density functions (one from the uniform $\mathcal{U}$ and one from the exponential

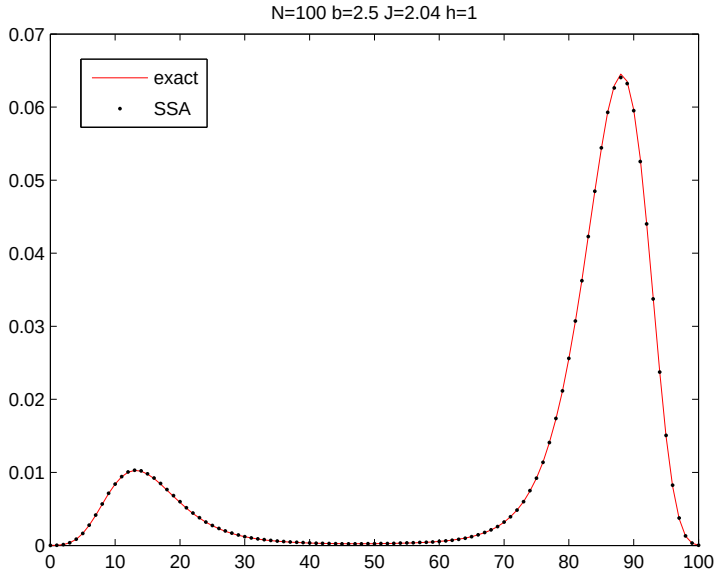

Figure 4: *Comparison of the exact measure with the empirical histogram obtained with the SSA algorithm for the Curie-Weiss model.*

$\mathcal{E}$). We will see later that sampling from continuous pdfs is much less expensive than that of discrete. The only problem is that the mean time interval is given by $< dt >= 1/a_0(x)$, which means that if the propensities functions are relatively large then the waiting times will be small and the algorithm will need a lot of steps in order to reach T. This problem comes the tau-leap method to overcome.

In figure (4) you can see the empirical histogram obtained with the above algorithm in comparison with the exact measure (34) for the Curie-Weiss model. Also in figure (3) you can see how the hysteresis effect is captured by SSA for $\beta J = 7$ and $N = 1000$.

3.2 Tau-leap algorithm

Although SSA is an exact algorithm its big problem is that in some cases may be extremely inefficient. As stated before if the propensities functions are large then the stochastic time steps will be small and a large number of iterations will be needed in order to reach a specific final time. The Tau-leap method is not exact; it uses a deterministic time increment τ, which may change during the execution, and updates the system population by approximating the execution number of every reaction.

Instead of updating the population at random time increments, as the SSA does, we can think as follows [9] : choose a constant time increment τ and ask the question "how many times every reaction fires during τ ?". Then the population can be updated

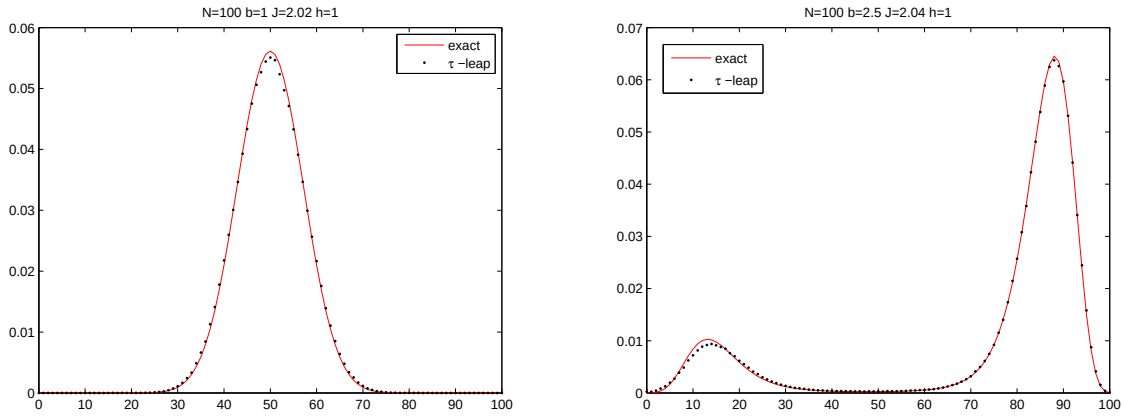

Figure 5: *Comparison of the exact measure with the empirical histogram obtained with the tau-leap algorithm for the Curie-Weiss model.*

appropriately. In order to present this idea formally we have to introduce some new definitions.

The functions that answers the above question is the following,

$$K_j(\tau; \boldsymbol{x}, t) \quad = \quad \text{the number of times that reaction } R_j \text{ will fire} \\ \text{in the time interval } [t, t+\tau], \text{ given } \boldsymbol{X}(t) = \boldsymbol{x}. \tag{55}$$

Of course solving for K_j is as difficult as solving the master equation. But if we incorporate the next condition the above quantity can be approximated.

Leap condition Choose τ to be "small enough", so that the propensity functions will remain about "constant" during the time interval $[t, t+\tau]$.

Then it can be shown that the random variable (55) follows a Poisson distribution with mean $a_j(\boldsymbol{x})\tau$.

Algorithm 2 tau-leap

Require: $T, \boldsymbol{x}_0, \tau$
  $\boldsymbol{x} = \boldsymbol{x}_0$
  while $t \leq T$ **do**
    sample p_j from $\mathcal{P}(a_j(\boldsymbol{x})\tau),\ j = 1, ..., M$
    $\boldsymbol{p} = (p_1, ..., p_M)^T$
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{\nu p}$
    $t \leftarrow t + \tau$
  end while

In order algorithm (2) to be more efficient than SSA, the time step τ must be chosen large enough. As mentioned earlier, sampling from a continuous pdf is much more efficient

than sampling from a discrete one. Some measurements done in matlab, showed that sampling from the Poisson distribution is about 300 times slower than sampling from the nomral distribution. This observation dictates that $\tau > \tau_c = 300/a_0(\boldsymbol{x})$, where $1/a_0(\boldsymbol{x})$ is the mean of the random time increment for the SSA.

This observation creates a conflict with the *leap condition*. If τ is very large then the propensity functions may not be constant during the time interval $[t, t+\tau]$, and the tau-leap method is not valid any more.

The empirical histograms for the Curie-Weiss model, shown in figure (5), were created for values of τ less than the critical value τ_c. Otherwise the *leap condition* is violated.

3.3 Invariant measure

In this section we will compare the invariant measure of τ-leap with the exact invariant measure of the Curie-Weiss models. In every time step we pick two Poisson random variables and take their difference. The first has mean value $a_1(x)\tau$ and the second $a_2(x)\tau$, where a_1, a_2 are the adsorption and desorption rates respectively. Their difference follows a distribution, named Skellam [22],

$$f_s(x; \mu_1, \mu_2) = e^{-(\mu_1+\mu_2)}\Big(\frac{\mu_1}{\mu_2}\Big)^{x/2} I_{|x|}(2\sqrt{\mu_1\mu 2}) \tag{56}$$

where $\mu_1 = a_1(x)\tau$, $\mu_2 = a_2(x)\tau$ and $I_{|x|}$ is the modified Bessel function of first kind [3].

Now we can compute the one step transition matrix, which depends on τ

$$P_\tau(x, y) = \mathbb{P}(X^{t+\tau} = y | X^t = x) = f_s(x; a_1(x)\tau, a_2(x)\tau) \tag{57}$$

The invariant measure π, solves the linear system

$$\pi = P_\tau \pi \tag{58}$$

or

$$\tilde{\pi} = \lim_{n\to\infty} P_\tau^n \tag{59}$$

where $\tilde{\pi}$ has all rows the same and π is equal to one of its rows. The invariant measure matrix was computed in Matlab. The matrix exponential is computed by diagonalization and n is chosen such that

$$||\tilde{\pi}_i - \tilde{\pi}_j||_\infty < tol \tag{60}$$

where $\tilde{\pi}_i$ is the ith row of $\tilde{\pi}$ and tol a small parameter (we chose tol as the machine epsilon). In our computation we take the difference of the first with last row. No difference was observed if we take other pairs.

In figure 6 you can see a comparison between the two invariant measures. For this parameters the mean value of the stochastic time increment is greater or equal to 0.001. For this value of τ good agreement between the two pdfs is observed. But for larger τ the left mode of the distribution is absent.

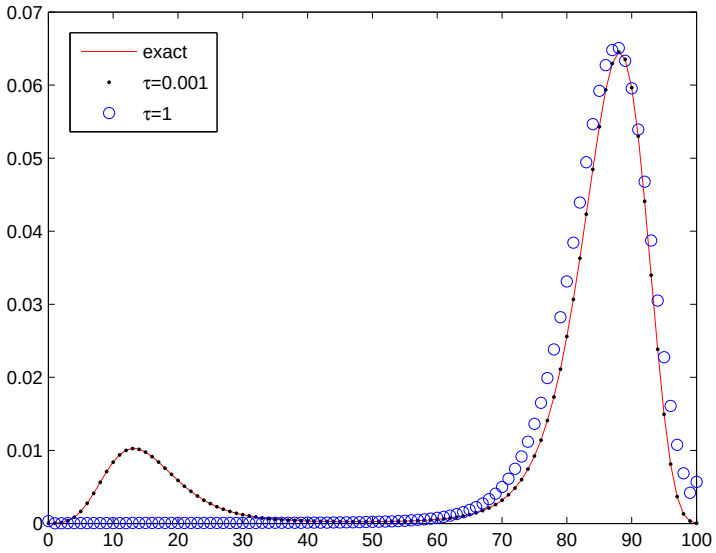

Figure 6: *Comparison of the exact invariant measure of the C-W model with the invariant measure of τ-leap. Here $b = 2.5$, $J = 2.04$, $h = 1$ and $N = 100$.*

3.4 Numerical Solution of SDEs

3.4.1 Fokker-Planck approximation of the Master equation

The general form of the Master equation

$$\frac{dP(x,t)}{dt} = \sum_{y \neq x} a(x,y)P(y,t) - a(y,x)P(x,t) \tag{61}$$

can be approximated by a stochastic differential equation , known as the Fokker-Planck equation. This non rigorous derivation was first given by Kramers and then improved by Moyal and is known as the Kramer-Moyal expansion (see [7]).

In the above equation, we substitute $u = x - y$ in the first term and $u = y - x$ in the second term. Defining

$$t(u,x) = a(x+u,x)$$

and substituting in the Master equation, we get,

$$\begin{aligned} \frac{dP(x,t)}{dt} &= \int \Big(a(x,x-u)P(x-u,t) - a(x+u,x)P(x,t)\, du \Big) \\ &= \int \Big(t(u,x-u)P(x-u,t) - t(u,x)P(x,t)\, du \Big) \end{aligned}$$

Then we expand the first term in the integral in power series,

$$\begin{aligned}\frac{dP(x,t)}{dt} &= \int \sum_{n=1}^{\infty} \frac{-u^n}{n!} \frac{\partial^n}{\partial u^n}(t(u,x)P(x,t))\, du \\ &= \sum_{n=1}^{\infty} \frac{-1^n}{n!} \int u^n \frac{\partial^n}{\partial x^n}(t(u,x)P(x,t))\, du \\ &= \sum_{n=1}^{\infty} \frac{-1^n}{n!} \frac{\partial^n}{\partial x^n} b_n(x)P(x,t) \end{aligned} \tag{62}$$

where

$$b_n(x) = \int u^n t(u,x)\, du = \int (y-x)^n a(y,x) dy$$

Note that the integral in the above equations can be substituted by sums in case the state space is discrete. If we take only the first two terms in the above series, we obtain the Fokker-Planck equation,

$$\frac{\partial}{\partial t}P(x,t) = -\frac{\partial}{\partial x}[b_1(x)P(x,t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[b_2(x)P(x,t)] \tag{63}$$

3.4.2 Equivalence of FPE and SDE

There in a connection between the Fokker-Planck equation, that describes the probability distribution function of a stochastic process, and the Stochastic Differential Equation that describes the individual paths of the same process. Consider the stochastic equation,

$$dX(t) = b_1(X,t)dt + \sqrt{b_2(X,t)}dW(t). \tag{64}$$

In order to show the equivalence between these two equations we need the Ito's formula for change of variables. For an arbitrary function f of $X(t)$, it states that,

$$df(X(t)) = \Big(b_1(X,t)f'(X) + \frac{1}{2}b_2(X,t)f''(X)\Big)dt + \sqrt{b_2}(X,t)f'(X)dW(t) \tag{65}$$

We now consider the time development of f,

$$\begin{aligned}\frac{\mathbb{E}df(X(t))}{dt} &= \mathbb{E}\frac{df(X(t))}{dt} \\ &= \frac{d}{dt}\mathbb{E}f(X(t)) \\ &= \mathbb{E}[b_1(X,t)f_X(X) + \frac{1}{2}b_2(X,t)f_{XX}(X)] \end{aligned}$$

Assuming that $X(t)$ has probability density $P(x,t)$, the above equation can be rewritten,

$$\begin{aligned}\frac{d}{dt}\mathbb{E}f(X(t)) &= \int f(X)\frac{\partial}{\partial t}P(X,t)dX \\ &= \int \Big(b_1(X,t)f_X(X) + \frac{1}{2}b_2(X,t)f_{XX}(X)\Big)P(X,t)dX \end{aligned}$$

By integrating by parts and discarding surface terms we get,

$$\int f(X)\frac{\partial}{\partial t}P(X,t)dx = \int f(X)\Big(-\frac{\partial}{\partial X}[b_1(X)P(X,t)] + \frac{1}{2}\frac{\partial^2}{\partial X^2}[b_2(X)P(X,t)]\Big)dX \quad (66)$$

and since f was arbitrary,

$$\frac{\partial}{\partial t}P(X,t) = -\frac{\partial}{\partial X}[b_1(X)P(X,t)] + \frac{1}{2}\frac{\partial^2}{\partial X^2}[b_2(X)P(X,t)] \quad (67)$$

which is the same as the FPE equation (63).

3.4.3 An example: Curie-Weiss approximation

As an example, consider the Curie-Weiss model by changing the variable $\eta = xN$ so the transition rates for the corresponding Master equation can be written as,

$$a(x,y) = \delta_{x,y+1}c_a(y) + \delta_{x,y-1}c_d(y). \quad (68)$$

In this case the first two terms in the Kramer-Moyal expansion are

$$\begin{aligned} b_1(x) &= \sum_y (y-x)a(y,x) \\ &= (x+1-x)a(x+1,x) + (x+1-x)a(x-1,x) \\ &= c_a(x) - c_d(x) \end{aligned}$$

and

$$\begin{aligned} b_2(x) &= \sum_y (y-x)^2 a(y,x) \\ &= (x+1-x)^2 a(x+1,x) + (x+1-x)^2 a(x-1,x) \\ &= c_a(x) + c_d(x) \end{aligned}$$

Finally the corresponding SDE is,

$$dx(t) = (c_a(x) - c_d(x))dt + \sqrt{c_a(x) + c_d(x)}dW(t) \quad (69)$$

3.4.4 Numerical Solution of SDEs

A general SDE can be written in integral form

$$X(t) = X_0 + \int_0^t f(X(s))ds + \int_0^t g(X(s))dW(s), \quad 0 \le t \le T \quad (70)$$

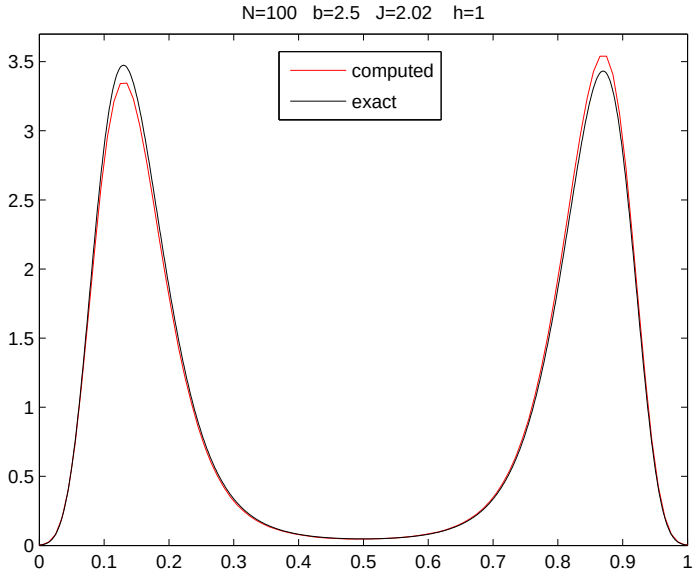

Figure 7: *Comparison of the exact measure with the empirical measure of the SDE that approximates the Curie-Weiss model. For the numerical solution of the SDE the Euler-Maruyama scheme is used.*

where f, g are scalar functions and the initial condition X_0 is a random variable. The second integral is to be taken with respect to Brownian motion [12]. We can can also write the previous equation in differential equation form,

$$dX(t) = f(X(t))dt + g(X(t))dW(t), \quad X(0) = X_0 \quad 0 \leq t \leq T \tag{71}$$

Let N be a positive integer, $\Delta t = \frac{T}{N}$ the time time step and define $t_j = j\Delta t$. Then the Euler-Maruyama method takes the form

$$X_j = X_{j-1} + f(X_{j-1})\Delta t + g(X_{j-1})(W(t_j) - W(t_{j-1})), \quad j = 2, ..., N \tag{72}$$

where X_j is the approximation of $X(t_j)$. As mentioned before $W(t)$ is a Brownian motion so the random variable $W(t) - W(s)$ is normally distributed with mean zero and variance $t - s$. Then we write the above difference $W(t_j) - W(t_{j-1})$ as

$$dW_j = W(t_j) - W(t_{j-1}) = \sqrt{(\Delta t)}r, \quad r \sim \mathcal{N}(0, 1) \tag{73}$$

We can refine the above difference by considering a refinement of the interval $[t_{j-1}, t_j] = [t_{j-1} = t_{j-1}^0, ..., t_{j-1}^M = t_j]$ where $t_{j-1}^i = t_{j-1} + i\delta t,\ i = 0, ..., M$ and $\delta t = \frac{\Delta t}{M}$.

$$dW_j = W(t_j) - W(t_{j-1}) = \sum_{i=1}^{M} W(t_{j-1}^i) - W(t_{j-1}^{i-1}) = \sqrt{\delta t}\sum_{i=1}^{M} r_i, \quad r_i \sim \mathcal{N}(0, 1) \tag{74}$$

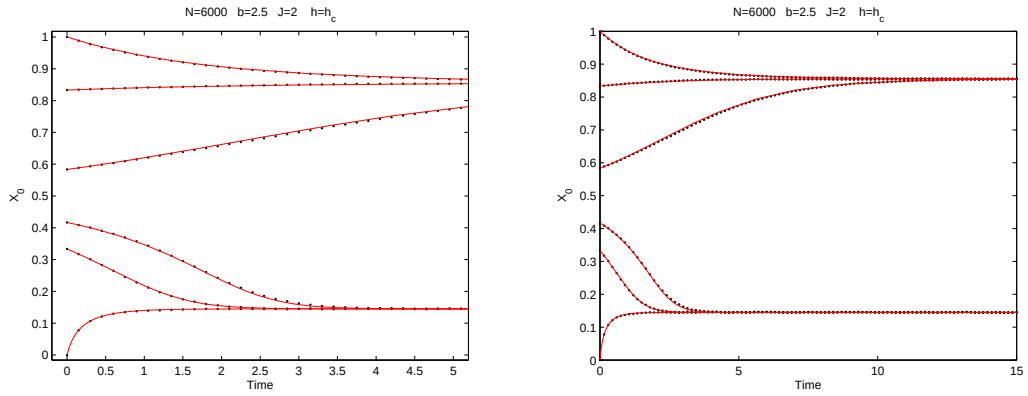

Figure 8: *The continuous line is the solution of the ODE describing the mean field solution for the Curie-Weiss model (parameters?). The dotted line is the mean solution of the SSA algorithm over 100 realizations.*

3.5 Mean Field approximation

In the Kramers-Moyal expansion (62) keeping only the first order term and following the procedure described in the previous section, we get the ordinary differential equation,

$$dX(t) = b_1(X,t)dt. \tag{75}$$

In figure (8) you can see a comparison of the Curie-Weiss model and its mean field approximation, which can be derived from equation (69) by neglecting the stochastic term. For large values of N good approximation is observed.

Another example we can see is the Schlögl model described in the first section. We can view this model as a jump process with rates

$$c_a(x) = c_1 ax(x-1) + c_3 b$$
$$c_d(x) = c_2 x(x-1)(x-2) + c_4 x$$

and the ODE approximation is the same as in the Curie-Weiss model. However there is no parameter we can send to infinity to obtain the deterministic limit. The difference with the Curie Weiss model is that that the first is obtained directly from the microscopic equations by doing appropriate approximations. In this case we have to artificially add a volume parameter, analogous to the number of particles N in the C-W model.

The new rates in the stochastic model $\hat{c}_i$ are related to the old rates by a factor of V which depends on the number of reactants involved in the ith reaction [24]. Fore a reaction involving m reactants, the relation will be $c_i = \hat{c}_i/V^{m-1}$. Thus, the new birth and death rates are written in terms of the old parameters as,

$$\hat{c}_a(x) = \frac{\hat{c}_1}{V} ax(x-1) + \hat{c}_3 V b$$
$$\hat{c}_d(x) = \frac{\hat{c}_2}{V^2} x(x-1)(x-2) + \hat{c}_4 x$$

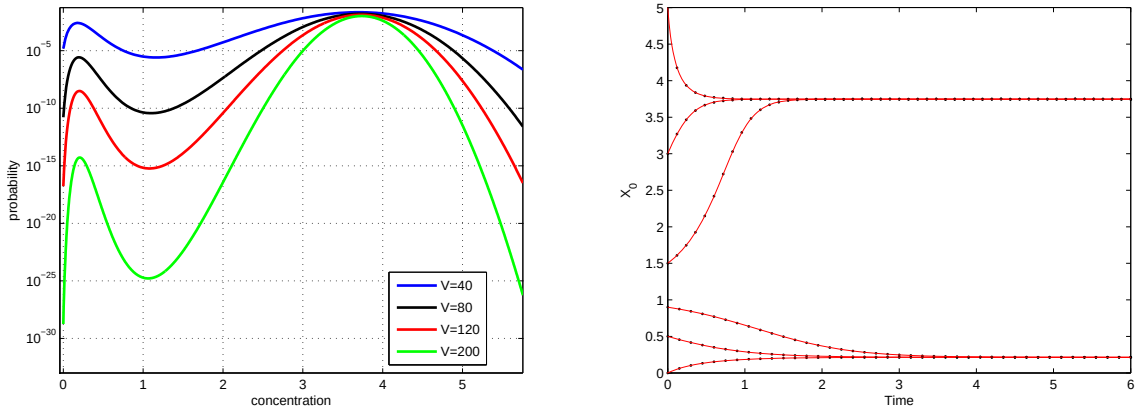

Figure 9: *Probability distribution function for the Schlögl model for different values of V (left). Comparison of the ODE approximation and the stochastic model for $V = 10^4$ (right). The parameters are $\hat{c}_1 = 3, \hat{c}_2 = 0.6, \hat{c}_3 = 0.25, \hat{c}_4 = 2.95, a = 1, b = 2$*

In figure (9) you can see the comparison of the pure jump process and its approximation. Looking on the left figure, we can observe that as V increases the probability of the first mode become smaller and the valley between the two modes is steeper. This means that the system gets trapped in one of the modes and the probability of jumping to the other state is effectively zero.

4 Hierarchical Fractional Step Parallelization Algorithms

In this section, the new algorithm "Hierarchical Parallelization Algorithm" will be presented. This method comes as a solution to the parallelization of kinetic Monte Carlo algorithms. It is based on the tested idea of Operator Splitting, coming from the solution of partial differential equations. First the theoretical derivation will be presented, then some numerical results concerning the validation of the method and finally a presentation and a conversation on some issues towards an optimized implementation.

4.1 Description

In order to allow the modeling of adsorption, desorption, diffusion and reactions mechanisms we consider the following mechanism at each configuration $\sigma \in \Sigma$: we denote by,

$$\begin{aligned} \sigma^{(x,\omega)} &= \text{the new configuration which is a one step} \\ &\quad \text{update of } \sigma \text{ at a neighborhood of each site } x \in \mathcal{L} \end{aligned} \tag{76}$$

Here $\omega \in \Omega_x$, where Ω_x is the index set of all possible configurations that corresponds to an update at neighborhood of site x. For instance, in the case of a diffusion process, in a single species system, $\Omega_x = \{y \in \mathcal{L} : \ |x - y| = 1\}$. You can see table (1) for a complete list of all models used here.

The transition rates of the Markov process for updating σ to $\sigma^(x, \omega)$ are denoted by $c(x, \omega; \sigma)$. The corresponding generator is

$$Lf(\sigma) = \sum_{x \in \mathcal{L}} \sum_{\omega \in \Omega_x} c(x, \omega; \sigma)(f(\sigma^{(x,\omega)}) - f(\sigma)) \tag{77}$$

The main idea behind the development of the Hierarchical Parallelization Algorithm for KMC relies on decomposition of the lattice $\mathcal{L}$ into non-overlapping sets $C_k, \ k = 1, ..., M$ so that,

$$\mathcal{L} = \bigcup_{k=1}^{M} C_k, \quad C_i \cap C_j = \emptyset, \ i \neq j \tag{78}$$

The nature of the above decomposition will be specified in the process. Our strategy on developing a parallel algorithm involves the following inter dependent steps.

Step 1: Lattice decomposition given by (78)

Step 2: Generator decomposition of (77) based on (78)

Model	Σ	Ω_x	$c(x,\omega;\sigma)$	$\sigma^{(x,\omega)}$
M1	$\{0,1\}$	$\{\emptyset\}$	$c(x;\sigma)$	$\sigma^x(y) = \begin{cases} 1-\sigma(x), & y = x \\ \sigma(y), & y \neq x \end{cases}$
M2	$\{1\}$	$\{y \in \mathcal{L} : \ \lvert x-y\rvert = 1\}$	$c(x,y;\sigma)$	$\sigma^{(x,y)}(z) = \begin{cases} \sigma(z), & z \neq x,y \\ \sigma(x), & z = y \\ \sigma(y), & z = x \end{cases}$
M3	$\{0,...,k\}$	$\{0,...,k\}$	$c(x,k;\sigma), k \in \Sigma$	$\sigma^{(x,k)}(z) = \begin{cases} \sigma(z), & z \neq x \\ k, & z = y \end{cases}$
M4	$\{0,...,k\}$	$\{\omega = (y,k,l) : \lvert y-x\rvert = 1, y \in \mathcal{L}, k,l \in \Sigma\}$	$c(x,y,k,l;\sigma), k,l \in \Sigma$	$\sigma^{(x,y,k,l)}(z) = \begin{cases} \sigma(z), & z \neq x,y \\ k, & z = y \\ l, & z = x \end{cases}$

Table 1: *Various information for different models. The abbreviations M1,M2,M3 and M4 stands for diffusion (single species), adsorption/desorption, multicomponent reactions and multicomponent reactions involving neighboring sites respectively.*

$$\begin{aligned} Lf(\sigma) &= \sum_{x\in\mathcal{L}}\sum_{\omega\in\Omega_x} c(x,\omega;\sigma)(f(\sigma^{(x,\omega)}) - f(\sigma)) \\ &= \sum_{k=1}^{M}\sum_{x\in C_k}\sum_{\omega\in\Omega_x} c(x,\omega;\sigma)(f(\sigma^{(x,\omega)}) - f(\sigma)) \\ &:= \sum_{k=1}^{M} L_k f(\sigma) \end{aligned} \tag{79}$$

Here the generators $L_k f(\sigma),\ k = 1,...,M$, define the new Markov processes $\{\sigma_t^k\}_{t\geq 0}$ on the entire Λ_N.

Remarks

Typically interactions are of nearest neighbor type, hence the update rates $c(x,\omega;\sigma)$ depend on the configuration σ only through $\sigma(x)$ and $\sigma(y)$ where $|x-y| = 1$. Similarly the new configuration $\sigma^{(x,\omega)}$ may involve changes not only at $\sigma(x)$, but also at $\sigma(y),\ |x-y| = 1$ (See examples 2-4). Thus the generator L_k in (79) would update at most the lattice sites $y \in \bar{C}_k := \{z : |x-z| = 1,\ x \in C_k\}$.

Therefore, the processes $\{\sigma_t^k\}_{t\geq 0}$ and $\{\sigma_t^{k'}\}_{t\geq 0}$ corresponding to L_k and $L_{k'}$ respectively are independent provided $\bar{C}_k \cup \bar{C}_{k'} = \emptyset$ and they also have independent initial data.

Step 3: Rearranging the operator sum into sums of independent processes.

Typically due to short range interactions, the processes $\{\sigma_t^k\}_{t\geq 0}$ and $\{\sigma_t^{k'}\}_{t\geq 0}$ are independent provided $|k-k'| > 1$, C_k and $C_{k'}$ are not nearest neighbors. Hence they can be simulated independently on separate processors.

Thus we want to group the sets $\{C_k\}_{k=1}^M$ in such a way that in each grouping all resulting all resulting processes are independent from each other and can be simulated in parallel. For instance in figure (10) we can consider the groupings of all odd strips

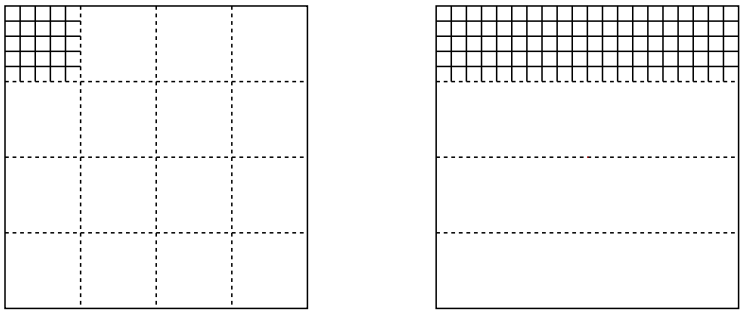

Figure 10: *Typical decompositions of the 2D lattice Λ_N, into coarse cells (right) and strips (left).*

$O = \{C_{2k-1} : k = 1, ..., \frac{M+1}{2}\}$ and even strips $E = \{C_{2k} : k = 1, ..., \frac{M}{2}\}$ we rearrange $Lf(\sigma)$ as

$$\begin{aligned} Lf(\sigma) &= L^E f(\sigma) + L^O f(\sigma) \\ &:= \sum_{k=1}^{\frac{M}{2}} L_{2k} f(\sigma) + \sum_{k=1}^{\frac{M+1}{2}} L_{2k-1} f(\sigma) \end{aligned} \tag{80}$$

Due to short range interactions $\{\sigma_t^{2k}\}_{t\geq 0}$ and $\{\sigma_t^{2k'}\}_{t\geq 0}$, $k \neq k'$ are independent and similarly the processes corresponding to odd indexed sets.

Step 4: Operator Splitting and communication

Based on the decomposition (80), we apply the Trotter formula as follows: first we recall that the corresponding semigroup (see [17]) e^{Lt} to the operator (generator) L has the following meaning,

$$< e^{Lt}\mu_0, f >= \mathbb{E}_{\mu_0} f(\sigma_t) \tag{81}$$

where μ_0 is the initial data distribution for the process $\{\sigma_t\}_{t\geq 0}$.

The Trotter product formula, formally (rigorously under suitably conditions on the operators) gives,

$$e^{Lt} = \lim_{n\to\infty} [e^{L^E \frac{t}{n}} e^{L^O \frac{t}{n}}]^n \tag{82}$$

The parallel scheme we propose here is based on the operator splitting approach stemming from (82) : to reach time T we use n deterministic steps of length $\Delta t = \frac{T}{n}$, where the solver corresponding to L^E and L^O are used in an alternating fashion.

The use of either L^E and L^O can be distributed on the processors in a parallel manner due to Step 3.

Remarks

1. Similarly to the Lie approximation

$$e^{L\Delta t} \approx e^{L^E \Delta t} e^{L^O \Delta t} \tag{83}$$

we may use the Strang splitting

$$e^{L\Delta t} \approx e^{L^E \frac{\Delta t}{2}} e^{L^O \Delta t} e^{L^E \frac{\Delta t}{2}} \tag{84}$$

Spatial corrections and communications take place when we switch from the L^E to L^0 and vice versa.

2. A key point in our method is that we do not rely on heuristic arguments for the communication between processors. Instead our splitting scheme algorithm relies on a Functional Analysis and Numerical Analysis perspective based on generators. As a result we have a a lot of flexibility in the decompositions which are based on selecting a convenient hierarchy $\{L_k f\}_{k=1}^{M}$. However our algorithm is not exact but approximate in Δt.

3. Our hierarchy of generators and domains can have additional levels in a tree-like structure, to account for hardware architecture such as multiple GPUs etc.

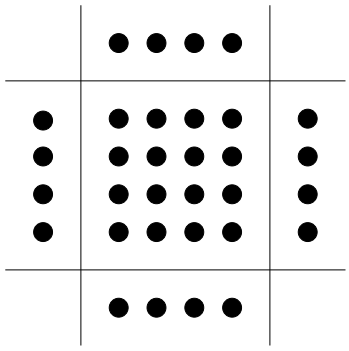

Figure 11: *The domain for one processor with its ghost cells in the case of nearest neighbor interactions.*

4.1.1 Choice of Δt

The choice of Δt is crucial not only because it is related to the error but also because it is related to the communication of the processors. On the one hand we must keep it sufficient small, in order to have meaningful computations, and on the other hand we must maximize its value so we can have less communication and so fast simulations.

As a first approach the choice is driven by the mean value of the stochastic δt. We take this value for one cell and since the distribution of the particles is uniform on the lattice this is approximately the mean jump time for every cell. Then we choose $\Delta t = f\bar{\delta t}$, where $\bar{\delta t}$ is the mean of δt . Here the factor f has the meaning of the mean number of jumps simulated on every cell.

In models were the particles concentration is not uniform (e.g. reaction models), this method has no meaning, since every cell has different mean jump times. In this case we have to decompose the domain again so every cell will have about the same mean jump time. This technique is explained in the section about work load balance.

4.2 Test cases

4.2.1 Exact solution comparison

In figure (1) you can see the comparison of the exact solution (20) with the Operator Splitting algorithm. Since this formula is obtained for $T \to \infty$ and $N \to \infty$ we chose large values of these variables for the simulation, $T = 1000$ and $N = 1000$. Then we average the mean magnetization in a single Markov chain that is already in equilibrium. In this simulation Δt is chosen to be equal to 1 and even for this large value you can see good agreement with the reference solution.

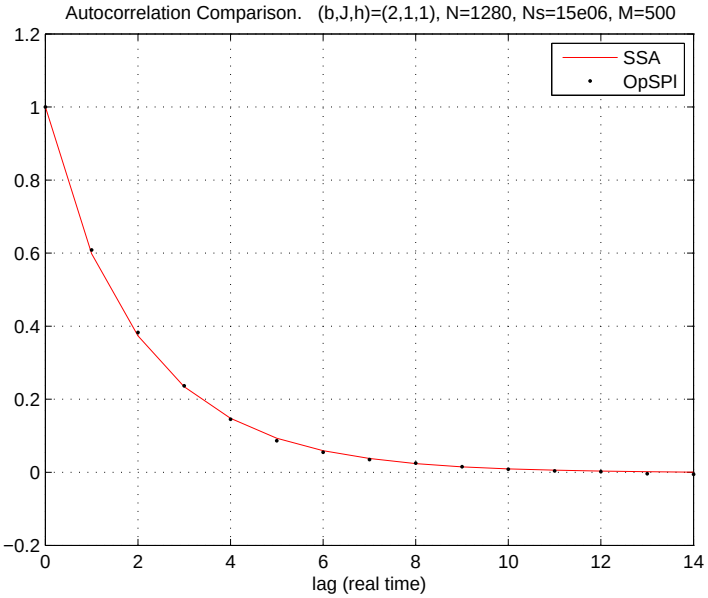

Figure 12: *Comparison of the autocorrelation function with the exact simulation algorithm and the solution obtained from the Operator Splitting algorithm with* $\Delta t = 1$.

4.2.2 Autocorrelation comparison

In this test we compute the autocorrelation function

$$\gamma(\tau) = \frac{\mathbb{E}[(X_{t+\tau} - \mu)(X_t - \mu)]}{\mathbb{E}[(X_t - \mu)^2]} \tag{85}$$

where $\mu = \mathbb{E}[X_t]$ and X_t is the mean coverage. Then we compare this function for the SSA and the Operator Splitting algorithm. This test gives us an estimate of the dynamics of the algorithm. We compute $15 \cdot 10^6$ samples of one Markov chain and then we compute the autocorrelation function. Finally we average over 500 different Markov chains in order to cancel the noise effects. Note that in this simulation the lag (τ) is real time and not number of samples. Again the results are in good agreement with the exact algorithm.

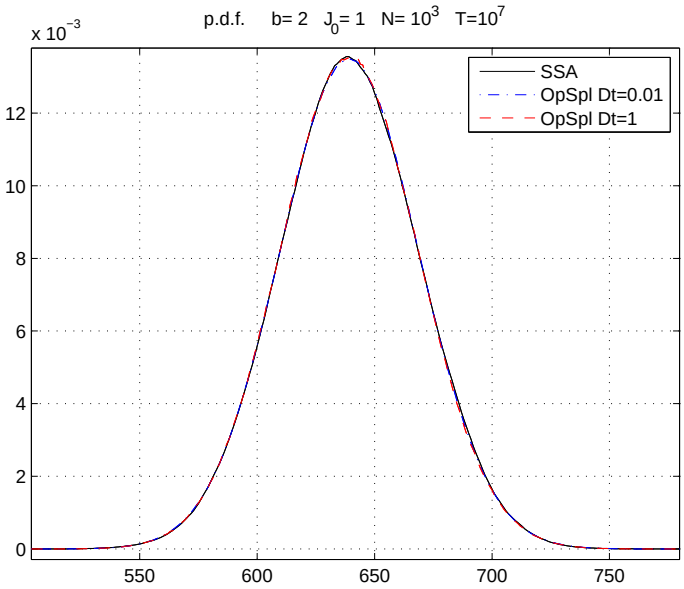

Figure 13: *Comparison of the probability distribution function with the exact simulation algorithm and the solution obtained from the Operator Splitting algorithm with* $\Delta t = 1$ *and* $\Delta t = 0.01$.

4.2.3 Probability distribution function comparison

In this final test we compare the experimental histogram, an approximation of the p.d.f., for the exact and the proposed algorithm. As you can see in figure (13) there is good agreement between the two algorithms. By these test we verify that the algorithm produces samples that have the same mean and variance with the samples obtained from SSA.

APPENDIX

A Hamiltonian transformation

Most solutions for the Ising model in bibliography, are provided for $\sigma \in \{-1, 1\}$. Here we will present the transormation of the Hamiltonian from $\sigma \in \{-1, 1\}$ to $\sigma \in \{0, 1\}$. We will consider the case of finite range potential and we will denote by N_x the set of all points in the latice that are considered neighboutrs of x.

Let $\tilde{\sigma} \in \{-1, 1\}$ and $\tilde{H}$ the coresponding Hamiltonian,

$$\tilde{H} = -\tilde{J} \sum_{x \in \mathcal{L}} \sum_{y \in N_x} \tilde{\sigma}(y)\tilde{\sigma}(x) - \tilde{h} \sum_{x \in \mathcal{L}} \tilde{\sigma}(x) \tag{86}$$

By substituting the relation

$$\tilde{\sigma}(x) = 2\sigma(x) - 1, \quad \sigma \in \{0, 1\} \tag{87}$$

we get

$$\begin{aligned}
\tilde{H} &= -\tilde{J} \sum_{x \in \mathcal{L}} \sum_{y \in N_x} (2\sigma(y) - 1)(2\sigma(x) - 1) - \tilde{h} \sum_{x \in \mathcal{L}} (2\sigma(x) - 1) && (88)\\
&= -4\tilde{J} \sum_{x \in \mathcal{L}} \sum_{y \in N_x} \sigma(x)\sigma(y) + 2\tilde{J} \sum_{x \in \mathcal{L}} \sum_{y \in N_x} \sigma(y) + \sigma(x) - 2\tilde{h} \sum_{x \in \mathcal{L}} \sigma(x) + C && (89)\\
&= -4\tilde{J} \sum_{x \in \mathcal{L}} \sum_{y \in N_x} \sigma(x)\sigma(y) - (2\tilde{h} - 4\tilde{J}|N_x|) \sum_{x \in \mathcal{L}} \sigma(x) + C && (90)
\end{aligned}$$

B Duality between Master Equation and Generator

Some times it is more convenient to consider a different form of the Master equation

$$\frac{dP(x,t)}{dt} = \sum_{y \neq x} a(x,y)P(y,t) - a(y,x)P(x,t) \tag{91}$$

where the probability density function is absent and is replaced by the mean value of an observable.

Consider a test function f in $\mathbb{R}$ that takes values in the configuration space. Then

$$\begin{aligned}
\frac{d\mathbb{E}[f(x)]}{dt} &= \sum_x f(x) \frac{dP(x,t)}{dt} \\
&= \sum_x \sum_{y \neq x} f(x)a(x,y)P(y,t) - \sum_x \sum_{y \neq x} f(x)a(y,x)P(x,t)
\end{aligned}$$

Taking the first part of the right hand side,

$$\begin{aligned}
\sum_x \sum_{y \neq x} f(x)a(x,y)P(y,t) &= \sum_x \left[\sum_y f(x)a(x,y)P(y,t) \right] - f(x)a(x,x)P(x) \\
&= \sum_x \sum_y f(x)a(x,y)P(y,t) - \sum_x f(x)a(x,x)P(x) \\
&= \sum_y \sum_x f(x)a(x,y)P(y,t) - \sum_x f(x)a(x,x)P(x) \\
&= \sum_x \sum_y f(y)a(y,x)P(x,t) - \sum_x f(x)a(x,x)P(x) \\
&= \sum_x \sum_{y \neq x} f(y)a(y,x)P(x,t)
\end{aligned}$$

Finally the above ode becomes,

$$\begin{aligned}
\frac{d\mathbb{E}[f(x)]}{dt} &= \sum_x \sum_{y \neq x} a(y,x)(f(y) - f(x))P(x,t) \\
&= \mathbb{E}[\sum_{y \neq x} a(y,x)\,(f(y) - f(x))].
\end{aligned}$$

References

[1] G. Arampatzis, M. Katsoulakis, P. Plechac, and L. Xu. Hierarchical fractional-step approximations and parallel kinetic monte carlo algorithms. *To be submitted*.

[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] I. Bronshtein, K. Semendyayev, G. Musiol, and H. Muehlig. *Handbook of Mathematics*. Springer, 4th edition, 2003.

[4] A. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis. Binomial distribution based τ-leap accelerated stochastic simulation. *Journal of Chemicall Physics*, 122, 2005.

[5] A. DeMassi and E. Presutti. *Mathematical Methods for Hydrodynamic Limits*. Springer, 1991.

[6] T. C. Dorlas. *Statistical Mechanics: Fundamentals and Model Solutions*. Taylor & Francis, 1999.

[7] C. Gardiner. *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*. Springer, 3rd edition, 2004.

[8] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 1976.

[9] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115(4):1716–1733, 2002.

[10] N. Goldenfeld. *Lectures on Phase Transition and the Renormalization Group*. Addison-Wesley, 1992.

[11] P. Hanggi, H. Grabert, P. Talkner, and H. Thomas. Bistable systems: Master equation versus fokker-planck modeling. *Physical Review A*, 29(1):371–378, 1984.

[12] D. J.Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):552–546, 2001.

[13] K. H. Karlsen and N. H. Risebro. An operator splitting method for nonlinear convection-diffusion equations. *Numer. Math.*, 77:365–382.

[14] J. Karlsson and R. Tempone. Towards automatic global error control: Computable weak error expansion for the tau-leap method. preprint.

[15] M. A. Katsoulakis and A. Szepessy. Stochastic hydrodynamical limits of particle systems. *Comm. Math. Sci.*, 4(3):513–549, 2006.

[16] G. Korniss, M. A. Novotny, and P. A. Rikvold. Parallelization of a dynamic monte carlo algorithm: A partially rejection-free conservative approach. *Journal of Computational Physics*, 153(2):488–508, 1999.

[17] T. M. Liggett. *Interacting Particle Systems*. Springer, 2004.

[18] B. D. Lubachevsky. Efficient parallel simulations of asynchronous cellular arrays. *Complex Systems 1*, 1(6):1099–1123, 1987.

[19] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Chemical Physics*, 21(6):1087–1092, 1953.

[20] D. M. Nicol and P. Heidelberger. A comparative study of parallel algorithms for simulating continuous time markov chains. *ACM Trans. Model. Comput. Simul.*, 5:326–354, 1995.

[21] Y. Shim and J. G. Amar. Semirigorous synchronous sublattice algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Physical Review B*, 71(12):125432+, 2005.

[22] J. G. Skellam. The frequency distribution of the difference between two poisson variates belonging to different populations. *Journal of the Royal Statistical Society*, 109(3):296, 1946.

[23] L. Tiejun. Analysis of explicit tau-leaping schemes for simulating chemically reacting systems. *Multiscale Model. Simul.*, 6(2):417–436, 2007.

[24] M. Vellela and H. Qian. Stochastic dynamics and non-equilibrium thermodynamics of a bistable chemical system: The schlögl model revisited. *Journal of the Royal Society Interface*, 6(39):925–940, 2009.