UNIVERSITY OF CRETE
FACULTY OF SCIENCES AND ENGINEERING
MATHEMATICS AND APPLIED MATHEMATICS DEPARTMENT

PhD Thesis

Parallelization and Uncertainty Quantification of Spatially Extended Kinetic Monte Carlo Methods

Georgios Arampatzis

Supervisor: Markos Katsoulakis

Heraklion, Crete
February 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ

Διδακτορική Διατριβή

Παραλληλοποίηση και Ανάλυση Ευαισθησίας για χωρικές μεθόδους Kinetic Monte Carlo

---

Γεώργιος Αραμπατζής

Επιβλέπων καθηγητής: Μάρκος Κατσουλάκης

Ηράκλειο, Κρήτη
Φεβρουάριος 2014

Thesis Committee

Evangelos Harmandaris
Markos Katsoulakis
Charalampos Makridakis

Thesis Defence Committee

Evangelos Harmandaris
Markos Katsoulakis
Georgios Kosioris
Charalampos Makridakis
Dimitrios Tsagkarogiannis
Chrysoula Tsogka
Zouraris Georgios

Abstract

In the first part of the thesis we present a mathematical framework for constructing and analyzing parallel algorithms for lattice kinetic Monte Carlo (kMC) simulations. The resulting algorithms, Fractional Step kinetic Monte Carlo algorithms (FS–kMC), have the capacity to simulate a wide range of spatio–temporal scales in spatially distributed, non–equilibrium physiochemical processes with complex chemistry and transport micro–mechanisms. The algorithms can be tailored to specific hierarchical parallel architectures such as multi-core processors or clusters of Graphical Processing Units (GPUs). The proposed parallel algorithms are controlled-error approximations of kinetic Monte Carlo algorithms, departing from the predominant paradigm of creating parallel kMC algorithms with exactly the same master equation as the serial one. We carry out a detailed benchmarking of the parallel kMC schemes using available exact solutions, for example, in Ising-type systems and we demonstrate the capabilities of the method to simulate complex spatially distributed reactions at very large scales on GPUs.

In the second part we study from a numerical analysis perspective the algorithms proposed in the first part of the work. FS-kMC are fractional step algorithms with a time-stepping window Δt, and as such they are inherently partially asynchronous since there is no processor communication during the period Δt. In this contribution we primarily focus on the error analysis of FS-kMC algorithms as approximations of conventional, serial kinetic Monte Carlo. A key aspect of the presented analysis relies on emphasizing a goal–oriented approach for suitably defined macroscopic observables (e.g., density, energy, correlations, surface roughness), rather than focusing on strong topology estimates for individual trajectories.

In the third part we propose a new class of coupling methods for the sensitivity analysis of high dimensional stochastic systems and in particular for lattice kinetic Monte Carlo. Sensitivity analysis for stochastic systems is typically based on approximating continuous derivatives with respect to model parameters by the mean value of samples from a finite difference scheme. Instead of using independent samples the proposed algorithm reduces the variance of the estimator by developing a strongly correlated-"coupled"- stochastic process for both the perturbed and unperturbed stochastic processes, defined in a common state space. The novelty of our construction is that the new coupled process depends on the targeted observables, e.g. coverage, Hamiltonian, spatial correlations, surface roughness, etc., hence we refer to the proposed method as goal-oriented sensitivity analysis. We demonstrate in several examples including adsorption, desorption and diffusion kinetic Monte Carlo that for the same confidence interval and observable, the proposed goal–oriented algorithm can be two orders of magnitude faster than existing coupling algorithms for spatial kMC such as the Common Random Number approach

Keywords. kinetic Monte Carlo, parallel, GPU, Fractional Step, goal–oriented, error analysis, sensitivity analysis, uncertainty quantification, variance reduction, stochastic coupling

Περίληψη

Στο πρώτο μέρος της εργασίας παρουσιάζουμε ένα μαθηματικό πλαίσιο για την κατασκευή και την ανάλυση παράλληλων αλγορίθμων kinetic Monte Carlo. Οι συγκεκριμένοι αλγόριθμοι, Fractional Step kinetic Monte Carlo algorithms (FS–kMC), έχουν τη δυνατότητα να προσομοιώνουν ένα μεγάλο εύρος χωρικών και χρονικών κλιμάκων σε φυσικο–χημικές διαδικασίες με πολύπλοκους μηχανισμούς. Οι αλγόριθμοι έχουν τη δυνατότητα να υλοποιηθούν σε ιεραρχικά παράλληλα συστήματα, π.χ. επεξεργαστές με πολλούς πυρήνες ή συστοιχίες από κάρτες γραφικών (Graphical Processing Units, GPUs). Οι εν λόγω αλγόριθμοι είναι αλγόριθμοι ελεγχόμενου σφάλματος και έχουν ως αφετηρία τους ήδη υπάρχοντες αλγορίθμους που ικανοποιούν ακριβώς την Master Equation. Ακόμη, γίνεται εκτεταμένη σύγκριση των αποτελεσμάτων των αλγορίθμων με ακριβείς λύσεις, π.χ. συστήματα τύπου Ising, και δείχνουμε την ικανότητα των αλγορίθμων να προσομοιώνουν με μεγάλη απόδοση μεγάλα χωρικά συστήματα χημικών αντιδράσεων.

Στο δεύτερο μέρος γίνεται η μελέτη απο την σκοπιά της αριθμητικής ανάλυσης των αλγορίθμων που παρουσιάστηκαν στο πρώτο μέρος. Στους FS–kMC αλγόριθμους υπάρχει ένα χρονικό βήμα Δt κατα το οποίο οι επεξεργαστές δεν επικοινωνούν μεταξύ τους κι έτσι η μέθοδος χαρακτηρίζεται ως μερικώς ασύγχρονη. Στη μελέτη αυτή επικεντρωνόμαστε στην ανάλυση του σφάλματος αυτών των μεθόδων ως προσεγγίσεις σειριακών μεθόδων kMC καθώς το μέγεθος του χρονικού βήματος πηγαίνει στο μηδέν. Το κεντρικό σημείο της ανάλυσης είναι ότι δείχνουμε ασθενή σύγκλιση, σύγκλιση δηλαδή μακροσκοπικών παρατηρήσιμων ποσοτήτων, αντί για σύγκλιση σε ισχυρή τοπολογία.

Στο τρίτο μέρος προτείνουμε μια νέα κλάση μεθόδων για την ανάλυση ευαισθησίας μεγάλης διάστασης στοχαστικών συστημάτων και συγκεκριμένα για χωρικούς αλγορίθμους kMC. Η ανάλυση ευαισθησίας στηρίζεται στην προσέγγιση παραγώγων των λύσεων ως προς μια παράμετρο με την μέση τιμή δειγμάτων από ένα σχήμα πεπερασμένων διαφορών. Αντί να χρησιμοποιεί ανεξάρτητα δείγματα που αυξάνουν την διασπορά του εκτιμητή, η προτεινόμενη μέθοδος χρησιμοποιεί δείγματα με μεγάλη ετεροσυσχέτιση κάνοντας χρήση της μεθόδου coupling. Η καινοτομία της δικής μας κατασκευής είναι ότι η κατασκευή της νέας, συζευγμένης, διαδικασίας εξαρτάται από την εκάστοτε παρατηρήσιμη ποσότητα, π.χ. Χαμιλτονιανή, πυκνότητα, κτλ. Τέλος, δείχνουμε σε διάφορα παραδείγματα ότι η προτεινόμενη μέθοδος μπορεί να δώσει μέχρι δύο τάξεις μεγέθους χρονική επιτάχυνση σε σχέση με τις υπάρχουσες μεθόδους για ανάλυση ευαισθησίας.

Λέξεις Κλειδιά. kinetic Monte Carlo, GPU, Fractional Step, stochastic coupling, παράλληλη επεξεργασία, ανάλυση σφάλματος, ανάλυση ευαισθησίας, μείωση διασποράς

Contents

Introduction

The purpose of this work is twofold; in the first part we provide a new parallel algorithm, along with the numerical analysis convergence proof, for the efficient simulation of high dimensional models described by Markov jump–type processes, while in the second part we propose a new algorithm for the sensitivity analysis of such processes. The last algorithm has the property that it has lower variance compared to all known methods for sensitivity analysis of such high dimensional models.

The time evolution of the Probability Density Function (PDF) of the stochastic models, under consideration in this work, is described through a high dimensional system of 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 PDF that describe the probability of the system to be in discrete state σ at time t, see [4] and $a(\sigma,\sigma')$ is the rate at which the system being at state σ jumps to state σ'. The dimension of the above system of ODEs, 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 that satisfies (1). Having enough samples and applying appropriate statistical methods we are able to reproduce the empirical probability distribution relative to P or approximate moments of 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 generate a large number 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" [11]. 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, [5]. 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 $a(\sigma, \sigma')$ for every σ' in the set of all possible states starting from σ and

$$R_{\sigma'} = \sum_{\sigma < \sigma'} a(\sigma, \sigma'),$$

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 draw 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 σ 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 a) 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 $T \gg 1$.

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), see [2]. This algorithm sorts the states in equivalence 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).

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 or simulating a model with many degrees of freedom that exceed the local memory of a single CPU node. In these cases one must use domain decomposition by splitting 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 starting 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 in [10] proposed an algorithm for the solution of this problem. The idea is that the lattice is split into smaller sublattices and every processor simulates only one part. 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 [9] 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 [12] 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 [12].

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. The time interval is divided into equal subintervals of length Δt. Then the algorithm advances the system until the fixed time step Δt and at the end of this step they communicate in order to keep the ghost cells informed, see [15]. These algorithms introduce errors due to inconsistencies at the boundary cells but due to the flexibility in choosing Δt this error can be as small as we want.

In Chapter 1 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 [7]. 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. Moreover, we present detailed benchmarking of the proposed parallel algorithms using analytically available exact solutions, for instance, in Ising-type systems and demonstrate the capabilities of the method to simulate complex spatially distributed molecular systems, such as CO oxidation on a catalytic surface.

In Chapter 2 we provide rigorous *goal-oriented* numerical analysis for the identified class of statistically relevant *macroscopic observables* that include all classical examples of statistical mechanics, e.g., coverage, spatial correlation, surface roughness. Moreover, we prove that the weak error of the numerical method goes to zero as Δt goes to zero, that the order (first or second) of the upper bound of the error in the fractional time step Δt depends on the parallel FS-KMC scheme, e.g., based on Lie, Strang and Random operator splittings, and this upper bound of the error is independent of the system (lattice) size for macroscopic observables. In addition, we employ these results in a practical fashion to investigate and assess the *processor communication* induced by different parallel schemes (Lie, Strang, Random). First, the bigger is the allowable Δt, within a desired error tolerance, the fewer processor communication steps are required, while processor communication itself is mathematically captured by the commutators of the operators involved in FS-KMC. Thus, for macroscopic observables, our error analysis allows us to estimate the asynchrony in the parallelization for such observables in terms of the greatest allowable Δt, within a given tolerance. Finally, our analysis mathematically justifies a class of existing domain decomposition methods for KMC [14, 15, 16], as well as a well established parallel KMC solver [13], giving rise to controlled errors and demonstrating and quantifying their partial asynchrony.

In Chapter 3 we propose a new class of coupling methods for the sensitivity analysis of high dimensional stochastic systems and in particular for lattice Kinetic Monte Carlo. Sensitivity analysis for stochastic systems is typically based on approximating continuous derivatives with respect to model parameters by the mean value of samples from a finite difference scheme. Instead of using independent samples the proposed algorithm reduces the variance of the estimator by developing a strongly correlated-"coupled"- stochastic process for both the perturbed and unperturbed stochastic processes, defined in a common state space. The novelty of our construction is that the new coupled process depends on the targeted observables, e.g. coverage, Hamiltonian, spatial correlations, surface roughness, etc., hence we refer to the proposed method as *goal-oriented* sensitivity analysis. In particular, the rates of the coupled Continuous Time Markov Chain are obtained as solutions to a goal-oriented optimization problem, depending on the observable of interest, by considering the minimization functional of the corresponding variance. We show that this functional can be used as a diagnostic tool for the design and evaluation of different classes of couplings. Furthermore, the resulting KMC sensitivity algorithm has an easy implementation that is based on the Bortz–Kalos–Lebowitz algorithm's philosophy, where here events are divided in classes depending on level sets of the observable of interest. Finally, we demonstrate in several examples including adsorption, desorption and diffusion Kinetic Monte Carlo that for the same confidence interval and observable, the proposed goal-oriented algorithm can be two orders of magnitude faster than existing coupling algorithms for spatial KMC such as the Common Random Number approach.

Bibliography

[1] D. F. Anderson, A. Ganguly, and T. G. Kurtz. Error analysis of tau-leap simulation methods. *Ann. Appl. Probab.*, 21(6):2226–2262, 2011.

[2] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for Monte Carlo simulation of Ising spin systems. *J. Comp. Phys.*, 17(1):10–18, 1975.

[3] Abhijit Chatterjee, Dionissios G. Vlachos, and Markos A. Katsoulakis. Binomial distribution based τ-leap accelerated stochastic simulation. *Journal of Chemicall Physics*, 122, 2005.

[4] C. Gardiner. *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences.* Springer, 4th edition, 2009.

[5] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[6] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115(4):1716–1733, JUL 22 2001.

[7] Kenneth Hvistendahl Karlsen and Nils Henrik Risebro. An operator splitting method for nonlinear convection-diffusion equations. *Numer. Math.*, 77:365–382, 1997.

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

[9] G. Korniss, M. A. Novotny, and P. A. Rikvold. Parallelization of a dynamic Monte Carlo algorithm: A partially rejection-free conservative approach. *J. Comp. Phys.*, 153(2):488–508, 1999.

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

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

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

[13] S. Plimpton, C. Battaile, M. Chandross, L. Holm, A. Thompson, V. Tikare, G. Wagner, E. Webb, X. Zhou, C. Garcia Cardona, and A. Slepoy. Crossing the Mesoscale No-Man's Land via Parallel Kinetic Monte Carlo. Technical Report SAND2009-6226, Sandia National Laboratory, 2009.

[14] Y. Shim and J. G. Amar. Rigorous synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Phys. Rev. B*, 71(11):115436, Mar 2005.

[15] Y. Shim and J. G. Amar. Semirigorous synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Phys. Rev. B*, 71(12):125432, Mar 2005.

[16] Y. Shim and J. G. Amar. Hybrid asynchronous algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *J. Comput. Phys.*, 212(1):305–317, 2006.

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

Chapter 1

Hierarchical fractional-step approximations and parallel kinetic Monte Carlo algorithms

We present a mathematical framework for constructing and analyzing parallel algorithms for lattice Kinetic Monte Carlo (KMC) simulations. The resulting algorithms have the capacity to simulate a wide range of spatio-temporal scales in spatially distributed, non-equilibrium physiochemical processes with complex chemistry and transport micro-mechanisms. The algorithms can be tailored to specific hierarchical parallel architectures such as multi-core processors or clusters of Graphical Processing Units (GPUs). The proposed parallel algorithms are controlled-error approximations of kinetic Monte Carlo algorithms, departing from the predominant paradigm of creating parallel KMC algorithms with exactly the same master equation as the serial one.

Our methodology relies on a spatial decomposition of the Markov operator underlying the KMC algorithm into a hierarchy of operators corresponding to the processors' structure in the parallel architecture. Based on this operator decomposition, we formulate *Fractional Step Approximation schemes* by employing the Trotter Theorem and its random variants; these schemes, (a) determine the *communication schedule* between processors, and (b) are run independently on each processor through a serial KMC simulation, called a *kernel*, on each fractional step time-window.

Furthermore, the proposed mathematical framework allows us to rigorously justify the numerical and statistical consistency of the proposed algorithms, showing the convergence of our approximating schemes to the original serial KMC. The approach also provides a systematic evaluation of different processor communicating schedules. We carry out a detailed benchmarking of the parallel KMC schemes using available exact solutions, for example, in Ising-type systems and we demonstrate the capabilities of the method to simulate complex spatially distributed reactions at very large scales on GPUs. Finally, we discuss *work load balancing* between processors and propose a re-balancing scheme based on probabilistic mass transport methods.

1.1 Introduction

Kinetic Monte Carlo algorithms have proved to be an important tool for the simulation of out-of-equilibrium, spatially distributed processes. Such models arise in physiochemical applications ranging from materials

science and catalysis, to complex biological processes. Typically the simulated models involve chemistry and/or transport micro-mechanisms for atoms and molecules, e.g., reactions, adsorption, desorption processes and diffusion on surfaces and through complex media, [18, 3, 6]. Furthermore, mathematically similar mechanisms and corresponding Kinetic Monte Carlo simulations arise in agent-based, evolutionary games problems in epidemiology, ecology and traffic networks, [36].

The simulation of stochastic lattice systems using Kinetic Monte Carlo (KMC) methods relies on the direct numerical simulation of the underlying Continuous Time Markov Chain (CTMC). Since such stochastic processes are set on a lattice (square, hexagonal, etc.) Λ_N with N sites, they have a discrete, albeit high-dimensional, configuration space Σ and necessarily have to be of jump type describing transitions between different configurations $\sigma \in \Sigma$. Mathematically, CTMC are defined in terms of the transition rates $c(x, \omega; \sigma)$ which correspond to an updating micro-mechanism that describes completely the evolution of the stochastic process as a transition from a current configuration σ of the system to a new configuration $\sigma^{x,\omega}$ by performing an update in a neighborhood of the site $x \in \Lambda_N$. In other words the probability of a transition over an infinitesimal time interval δt is $\mathbb{P}\left(S_{t_{\delta t}} = \sigma^{x,\omega} \,|\, S_t = \sigma\right) = c(x, \omega; \sigma)\delta t + \mathrm{o}(\delta t^2)$. In turn, the transition rates define the total rate

$$\lambda(\sigma) = \sum_{x \in \Lambda_N} \sum_{\omega \in \mathcal{S}_x} c(x, \omega; \sigma)\,, \tag{1.1}$$

which is the intensity of the exponential waiting time for a jump to be performed when the system is currently at the state σ. Here $\omega \in \mathcal{S}_x$, where $\mathcal{S}_x$ is the set of all possible configurations that correspond to an update at a neighborhood Ω_x of the site x. Once this exponential "clock" signals a jump, then the system transitions from the state σ to a new configuration $\sigma^{x,\omega}$ with probability

$$p(\sigma, \sigma^{x,\omega}) = \frac{c(x, \omega; \sigma)}{\lambda(\sigma)}\,. \tag{1.2}$$

Thus the full stochastic evolution is completely defined. We refer to the discussion in Section 1.2 for a complete mathematical description of the KMC method. The implementation of this method is based on efficient calculation of (1.1) and (1.2), and was first developed in [5], known as a BKL Algorithm, for stochastic lattice Ising models, and in [10] known as Stochastic Simulation Algorithm (SSA) for reaction systems. However, as it is evident from formulas (1.1) and (1.2), the algorithms are inherently serial as updates are done at one site $x \in \Lambda_N$ at a time, while on the other hand the calculation of (1.1) depends on information from the entire spatial domain Λ_N. For these reasons it seems, at first glance, that KMC algorithms cannot be parallelized easily.

However, Lubachevsky, in [21], proposed an *asynchronous* approach for parallel KMC simulation in the context of Ising systems, in the sense that different processors simulate independently parts of the physical domain, while inconsistencies at the boundaries are corrected with a series of suitable rollbacks. This method relies on uniformization of the total rates over each processor, see also [12] for the use of uniformization in the parallel simulation of general CTMC. Thus the approach yields a *null-event* algorithm, [18], which includes rejected moves over the entire domain of each processor. Furthermore, Lubachevsky proposed a modification in order to incorporate the BKL Algorithm in his parallelization method, which was implemented and tested in [16]. This is a partially rejection-free (still asynchronous) algorithm, where BKL-type rejection-free simulations are carried out in the interior of each processor, while uniform rates were used at the boundary, reducing rejections over just the boundary set. However, in spite of the proposed improvements, these asynchronous algorithms may still have a high number of rejections for boundary events and rollbacks, which considerably reduce the parallel efficiency, [34]. Advancing processors in time in a synchronous manner over a fixed time-window can provide a way to mitigate the excessive number of

boundary inconsistencies between processors and ensuing rejections and rollbacks in earlier methods. Such *synchronous* parallel KMC algorithms were proposed and extensively studied in [8, 34, 24, 27]. However, several costly global communications are required at each cycle between all processors, whenever a boundary event occurs in any one of them, in order to avoid errors in the inter-processor communication and rollbacks, [27].

As we will discuss further in this paper, many of the challenges in parallel KMC can be addressed by abandoning the earlier perspective on creating a parallel KMC algorithm with the exactly same rates (and hence the generator and master equation) as the serial algorithm, see [23] for a discussion on exact algorithms. This is a very natural idea in the numerical analysis of continuum models such as Ordinary and Partial Differential Equations (ODE/PDE). First, in [35] the authors propose an *approximate* algorithm, in order to create a parallelization scheme for KMC. It was recently demonstrated [27, 30], that this method is very promising: boundary inconsistencies are resolved in a straightforward fashion, while there is an absence of global communications in contrast to synchronous relaxation schemes discussed earlier. Finally, we note that, among the parallel algorithms tested in [27], the approximate algorithm had the highest parallel efficiency.

Here we develop a general mathematical framework for *parallelizable approximations* of the KMC algorithm. Our approach relies on first developing a spatial decomposition of the Markov operator, that defines the Kinetic Monte Carlo algorithm, into a hierarchy of operators. The decomposition is tailored to the processor architecture. Based on this operator decomposition, we formulate *Fractional Step Approximation* schemes by employing the Trotter product formula. In turn these approximating schemes determine *Communication Schedule* between processors through the sequential application of the operators in the decomposition, and the time step employed in the particular fractional step scheme. Here we discuss deterministic schedules resulting from Lie- and Strang-type fractional step schemes, as well as random schedules derived by the Random Trotter Theorem, [17].

We show that the scheme in [35] is a particular case of a random schedule and can be mathematically analyzed within the proposed framework. We recall that the deterministic Trotter Theorem was first proved in [37] for the approximation of semigroups corresponding to operator sums, and it has found wide application in the numerical ODE/PDE analysis, e.g., [11].

In Section 1.2 we show that the Fractional Step KMC schemes allow us to run independently on each processor a serial KMC simulation (called a *kernel*) on each fractional time-step window. Furthermore, processor communication is straightforward at the end of each fractional time-step while no global communications or rollbacks are involved. In Section 1.5 we show that the hierarchical structure of our methodology can be easily implemented for very general physiochemical processes modeled by lattice systems, allowing users to input as the algorithm's KMC kernel their preferred serial algorithm. This flexibility and hierarchical structure are key advantages for tailoring our framework to particular parallel architectures with complex memory and processor hierarchies, e.g., clusters of GPUs.

The proposed mathematical framework allows us to rigorously prove the numerical and statistical consistency of the proposed algorithms, while on the other hand it provides a systematic evaluation of different processor communication schedules. Indeed, in Section 1.3 the numerical and statistical consistency of the proposed algorithms is rigorously justified by the Trotter Theorem, [37], [11] showing the convergence of our approximating schemes to the original serial KMC algorithm, interpreted as convergence to the underlying Markov operator. Using the Random Trotter Theorem [17] we show that the approximation schemes with a random schedule, including the one in [35] as a special case, are numerically consistent in the approximation limit; that is, as the time step in the fractional step scheme converges to zero, it converges to a

continuous time Markov Chain that has the same master equation and generator as the original serial KMC. In Section 1.4 we show that the proposed mathematical framework can allow the study of controlled-error approximation properties of Fractional Step KMC schemes, as well as the systematic evaluation of different processor communicating schedules, comparing for instance the scheme in [35] to the Lie scheme (1.12).

Finally, in Section 1.6 we discuss work-load balancing between processors and propose a re-balancing scheme based on probabilistic mass transport methods, [9], which is particularly well-suited for the proposed fractional step KMC methods. In Section 1.7 we present detailed benchmarking of the proposed parallel algorithms using analytically available exact solutions, for instance, in Ising-type systems and demonstrate the capabilities of the method to simulate complex spatially distributed molecular systems, such as CO oxidation on a catalytic surface.

1.2 Fractional Step Kinetic Monte Carlo Algorithms

We first present the mathematical background of KMC in a more abstract way in order to demonstrate the generality and the applicability of the proposed method. We consider a d-dimensional lattice Λ_N with N lattice sites. We restrict our discussion to lattice gas models where the order parameter or the spin variable takes value in a finite countable set $\Sigma = \{0, 1, \ldots, K\}$. At each lattice site $x \in \Lambda_N$ an order parameter (a spin variable) $\sigma(x) \in \Sigma$ is defined. The states in Σ correspond to occupation of the site $x \in \Lambda_N$ by different species. For example, if $\Sigma = \{0, 1\}$ the order parameter models the classical lattice gas with a single species occupying the site x when $\sigma(x) = 1$ and with the site being vacant if $\sigma(x) = 0$. We denote $\{\sigma_t\}_{t\geq 0}$ the stochastic process with values in the configuration space $\mathcal{S} = \Sigma^{\Lambda_N}$.

Our primary focus is on modeling the basic processes of adsorption, desorption, diffusion and reactions between different species. Thus the local dynamics is described by a collection of the transition rates $c(x, \omega; \sigma)$ and by an updating mechanism such that the configuration σ of the system changes into a new configuration $\sigma^{x,\omega}$ by an update in a neighborhood of the site $x \in \Lambda_N$. Here $\omega \in \mathcal{S}_x$, where $\mathcal{S}_x$ is the set of all possible configurations that correspond to an update at a neighborhood Ω_x of the site x. For example, if the modeled process is a diffusion of the classical lattice gas a particle at x, i.e., $\sigma(x)$ can move to any nearest neighbor of x, i.e., $\Omega_x = \{y \in \Lambda_N \,|\, |x - y| = 1\}$ and $\mathcal{S}_x$ is the set of all possible configurations $\mathcal{S}_x = \Sigma^{\Omega_x}$. In other words the collection of measures $c(x, \omega; \sigma)$ defines the transition probability from σ to $\sigma^{x,\omega}$ over an infinitesimal time interval δt. More precisely, the evolution of the system is described by a continuous time Markov jump process with the generator $\mathcal{L} : C_b(\mathcal{S}) \to C_b(\mathcal{S})$ acting on continuous bounded test functions $f \in C_b(\mathcal{S})$ according to

$$\mathcal{L}f(\sigma) = \sum_{x\in\Lambda_N} \sum_{\omega\in\mathcal{S}_x} c(x, \omega; \sigma)[f(\sigma^{x,\omega}) - f(\sigma)]\,. \tag{1.3}$$

We recall that the evolution of the expected value for an arbitrary observable $f \in C_b(\mathcal{S})$ is given by the action of the Markov semigroup $e^{t\mathcal{L}}$ associated with the generator $\mathcal{L}$ and the process $\{S_t\}_{t\geq 0}$

$$\langle e^{t\mathcal{L}}\mu_0, f\rangle = \mathbb{E}_{S_0}[f(S_t)]\,, \tag{1.4}$$

where μ_0 is the initial distribution of the process, i.e. of the random variable S_0, [19]. Practically, the sample paths $\{\sigma_t\}_{t\geq 0}$ are constructed via KMC, that is through the procedure described in (1.1) and (1.2).

To elucidate the introduced notation we give a few examples relevant to the processes modeled here. We refer, for instance, to [18, 3, 6] for a complete discussion of the physical processes.

Examples.

1. *Adsorption/Desorption for single species particles.* In this case spins take values in $\sigma(x) \in \Sigma = \{0, 1\}$, $\Omega_x = \{x\}$, $\mathcal{S}_x = \{0, 1\}$ and the update represents a spin flip at the site x, i.e., for $z \in \Lambda_N$

$$\sigma^{x,\omega}(z) \equiv \sigma^x(z) = \begin{cases} \sigma(z) & \text{if } z \neq x, \\ 1 - \sigma(x) & \text{if } z = x. \end{cases}$$

2. *Diffusion for single species particles.* The state space for spins is $\sigma(x) \in \Sigma = \{0, 1\}$, $\Omega_x = \{y \in \Lambda_N \,|\, |x - y| = 1\}$ includes all nearest neighbors of the site x to which a particle can move. Thus the new configuration $\sigma^{x,\omega} = \sigma^{(x,y)}$ is obtained by updating the configuration $S_t = \sigma$ from the set of possible local configuration changes $\{0, 1\}^{\Omega_x}$ using the specific rule, also known as spin exchange, which involves changes at two sites x and $y \in \Omega_x$

$$\sigma^{x,\omega}(z) \equiv \sigma^{(x,y)}(z) = \begin{cases} \sigma(z) & \text{if } z \neq x, y, \\ \sigma(x) & \text{if } z = y, \\ \sigma(y) & \text{if } z = x. \end{cases}$$

   The transition rate is then written as $c(x, \omega; \sigma) = c(x, y; \sigma)$. The resulting process $\{\sigma_t\}_{t \geq 0}$ defines dynamics with the total number of particles ($\sum_{x \in \Lambda_N} \sigma(x)$) conserved, sometimes referred to as Kawasaki dynamics.

3. *Multicomponent reactions.* Reactions that involves K species of particles are easily described by enlarging the spin space to $\Sigma = \{0, 1, \ldots, K\}$. If the reactions occur only at a single site x, the local configuration space $\mathcal{S}_x = \Sigma$ and the update is indexed by $k \in \Sigma$ with the rule

$$\sigma^{x,\omega}(z) \equiv \sigma^{(x,k)}(z) = \begin{cases} \sigma(z) & \text{if } z \neq x, y, \\ k & \text{if } z = x. \end{cases}$$

   The rates $c(x, \omega; \sigma) \equiv c(x, k; \sigma)$ define probability of a transition $\sigma(x)$ to species $k = 1, \ldots, K$ or vacating a site, i.e., $k = 0$, over δt.

4. *Reactions involving particles with internal degrees of freedom.* Typically a reaction involves particles with internal degrees of freedom, and in this case several neighboring lattice sites may be updated at the same time, corresponding to the degrees of freedom of the particles involved in the reaction. For example, in a case such as CO oxidation on a catalytic surface, [20], when only particles at a nearest-neighbor distance can react we set $\sigma(x) \in \Sigma = \{0, 1, \ldots, K\}$, $\Omega_x = \{y \in \Lambda_N \,|\, |x - y| = 1\}$ and the set of local updates $\mathcal{S}_x = \Sigma^{\Omega_x}$. Such $\mathcal{S}_x$ contains all possible reactions in a neighborhood of x. When reactions involve only pairs of species, the rates can be indexed by k, $l \in \Sigma$, or equivalently $\mathcal{S}_x = \Sigma \times \Sigma$. Then the reaction rate $c(x, \omega; \sigma) = c(x, y, k, l; \sigma)$ describes the probability per unit time of $\sigma(x) \to k$ at the site x and $\sigma(y) \to l$ at y, i.e., the updating mechanism

$$\sigma^{x,\omega}(z) \equiv \sigma^{(x,y,k,l)}(z) = \begin{cases} \sigma(z) & \text{if } z \neq x, y, \\ k & \text{if } z = x, \\ l & \text{if } z = y, \end{cases}$$

   where $|x - y| = 1$.

1.2.1 Hierarchical structure of the generator1

The generator of the Markov process $\{\sigma_t\}_{t\geq 0}$ given in a general form in (1.3) is our starting point for the development of parallel algorithms based on geometric partitioning of the lattice. The lattice Λ_N is decomposed into non-overlapping cells C_m, $m = 1, \ldots, M$ such that

$$\Lambda_N = \bigcup_{m=1}^{M} C_m\,, \quad C_m \cap C_n = \emptyset\,,\ m \neq n\,. \tag{1.5}$$

With each set C_m a larger set $\bar{C}_m$ is associated by adding sites to C_m which are connected with sites in C_m by interactions or the updating mechanism, see Figure 1.1a. More precisely, we define the range of interactions L for the set C_m and the closure of this set

$$\bar{C}_m = \{z \in \Lambda_N \,|\, |z - x| \leq L\,, x \in C_m\}\,, \ \text{ where } L = \max_{x\in C_m}\{\operatorname{diam}\Omega_x\}\,.$$

In many models the value of L is independent of x due to translational invariance of the model. The boundary of C_m is then defined as $\partial C_m = \bar{C}_m \cap C_m$. This geometric partitioning induces a decomposition of the generator (1.3)

$$\mathcal{L}f(\sigma) = \sum_{x\in\Lambda_N}\sum_{\omega\in\mathcal{S}_x} c(x,\omega;\sigma)[f(\sigma^{x,\omega}) - f(\sigma)] \tag{1.6}$$

$$= \sum_{m=1}^{M}\sum_{x\in C_m}\sum_{\omega\in\mathcal{S}_x} c(x,\omega;\sigma)[f(\sigma^{x,\omega}) - f(\sigma)] \tag{1.7}$$

$$= \sum_{m=1}^{M} \mathcal{L}_m f(\sigma)\,. \tag{1.8}$$

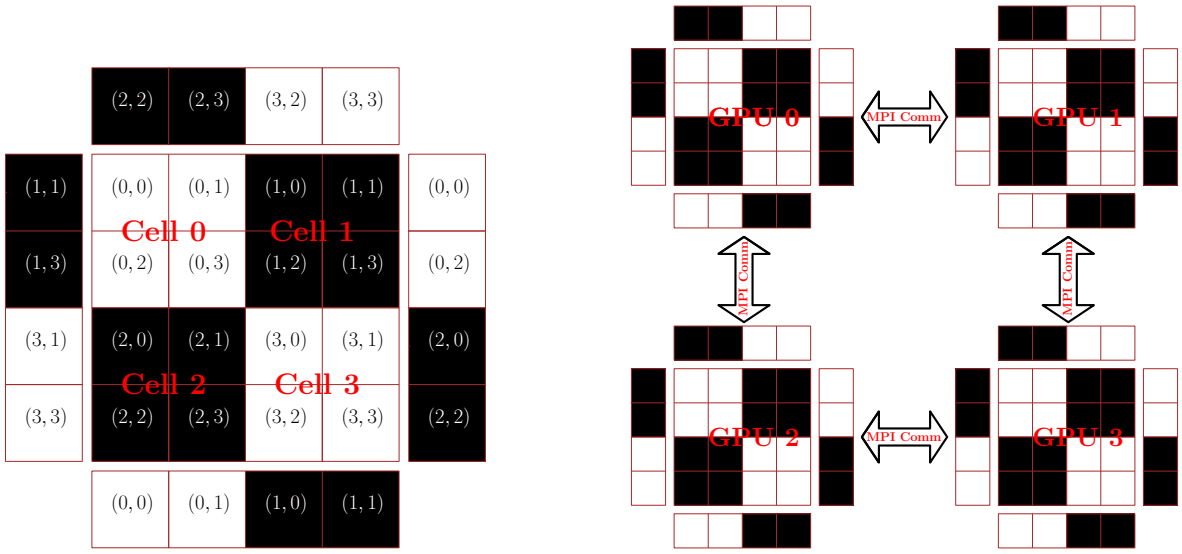
The generators $\mathcal{L}_m$ define new Markov processes $\{S_t^m\}_{t\geq 0}$ on the *entire* lattice Λ_N.

Remark: In many models the interactions between particles are of the two-body type with the nearest-neighbor range and therefore the transition rates $c(x,\omega;\sigma)$ depend on the configuration σ only through $\sigma(x)$ and $\sigma(y)$ with $|x - y| = 1$. Similarly the new configuration $\sigma^{x,\omega}$ involve changes only at the sites in this neighborhood. Thus the generator $\mathcal{L}_m$ updates the lattice sites at most in the set $\bar{C}_m = \{z \,|\, |x - z| = 1\,, x \in C_m\}$, see Figure 1.1a. Consequently the processes $\{S_t^m\}_{t\geq 0}$ and $\{S_t^{m'}\}_{t\geq 0}$ corresponding to $\mathcal{L}_m$ and $\mathcal{L}_{m'}$ are independent provided $\bar{C}_m \cap \bar{C}_{m'} = \emptyset$.

Therefore, splitting (1.6) allows us to define independent processes which yields an algorithm suitable for parallel implementation, in particular, in the case of short-range interactions when the communication overhead can be handled efficiently. If the lattice Λ_N is partitioned into subsets C_m such that the diameter $\operatorname{diam} C_m > L$, where L is the range of interactions, we can group the sets $\{C_m\}_{m=1}^{M}$ in such a way that there is no interaction between sites in the sets C_m that belong to the same group. For the sake of simplicity we assume that the lattice is divided into two *sub-lattices* described by the index sets $\mathcal{I}^1$ and $\mathcal{I}^2$, (black vs. white in Fig. 1.1a), hence we have

$$\Lambda_N = \Lambda_N^E \cup \Lambda_N^O := \bigcup_{m\in\mathcal{I}^1} C_m^E \cup \bigcup_{m\in\mathcal{I}^2} C_m^O\,. \tag{1.9}$$

Other lattice partitionings are also possible and may be more suitable for specific micro-mechanisms in the KMC or the computer architecture. Choice of the partitioning scheme can reduce communication overhead, see for instance [35]. For the sake of simplicity in the presentation, here we consider the partitioning depicted



(a) *(b)*

Figure 1.1: (a) Lattice decomposition in (1.9) using the checkerboard scheme mapped onto a single multi-threading processing unit (e.g., GPU). The integer cell coordinates also indicate communication through boundary buffer regions. In practice other partitionings may result in a lower communication overhead. (b) Hierarchical lattice partitioning on a cluster of processing units.

in (1.9) and Fig. 1.1a, although our mathematical framework applies to any other sublattice decomposition. Returning to (1.9), the sub-lattices *induce* a corresponding splitting of the generator:

$$\mathcal{L} = \mathcal{L}^E + \mathcal{L}^O := \sum_{m\in\mathcal{I}^1} \mathcal{L}^E_m + \sum_{m\in\mathcal{I}^2} \mathcal{L}^O_m \,. \tag{1.10}$$

This simple observation has key consequences for simulating the process $\{S_t\}_{t\geq 0}$ in parallel, as well as formulating different related algorithms: the processes $\{S^m_t\}_{t\geq 0}$ corresponding to the generators $\mathcal{L}^E_m$ are mutually independent for different $m \in \mathcal{I}^1$, and thus can be simulated in parallel; similarly we can handle the processes belonging to the group indexed by $\mathcal{I}^2$. However, there is still communication between these two groups as there is non-empty overlap between the groups due to interactions and updates in the sets ∂C_m, $\partial C'_m$ when $m \in \mathcal{I}^1$ and $m' \in \mathcal{I}^2$ and the cells are within the interaction range L. To handle this communication we next introduce a Fractional Step approximation of the Markov semigroup $e^{t\mathcal{L}}$ associated with the process $\{S_t\}_{t\geq 0}$.

1.2.2 Fractional Step Kinetic Monte Carlo Algorithms

The deterministic Trotter Theorem was first proved in [37] for the approximation of semigroups corresponding to operator sums, and it has found wide application in the numerical ODE/PDE analysis, e.g., [11]. Similarly, the key tool for our analysis is a deterministic as well as a stochastic version of the Trotter formula, [17], applied to the operator $\mathcal{L} = \mathcal{L}^E + \mathcal{L}^O$

$$e^{t\mathcal{L}} = \lim_{n\to\infty} \left[e^{\frac{t}{n}\mathcal{L}^E} e^{\frac{t}{n}\mathcal{L}^O} \right]^n . \tag{1.11}$$

The proposed parallel scheme uses the fact that the action of the operator $\mathcal{L}^E$ (and similarly of $\mathcal{L}^O$) can be distributed onto independent processing units. Thus to reach a time T we define a time step $\Delta t = \frac{T}{n}$ for

a fixed value of n and alternate the evolution by $\mathcal{L}^E$ and $\mathcal{L}^O$. More precisely, (1.11) gives rise to the *Lie* splitting approximation for $n \gg 1$:

$$e^{T\mathcal{L}} \approx \left[e^{\frac{T}{n}\mathcal{L}^E} e^{\frac{T}{n}\mathcal{L}^O} \right]^n . \tag{1.12}$$

Since the simulated systems exhibit short-range interactions, the generators $\mathcal{L}_k^E, \mathcal{L}_l^E$ *commute* for $k, l \in \mathcal{I}^1$, $k \neq l$:

$$\mathcal{L}_k^E \mathcal{L}_l^E - \mathcal{L}_l^E \mathcal{L}_k^E = 0 , \quad \text{for all } k, l \in \mathcal{I}^1,\ k \neq l .$$

Hence, [37], we have the exact formula

$$e^{\Delta t \mathcal{L}^E} e^{\Delta t \mathcal{L}^O} = \prod_{m \in \mathcal{I}^1} e^{\Delta t \mathcal{L}_m^E} \prod_{m \in \mathcal{I}^2} e^{\Delta t \mathcal{L}_m^O} . \tag{1.13}$$

Then the expression (1.13) implies that the KMC solvers corresponding to the semigroup $e^{\Delta t \mathcal{L}^E}$ (resp. $e^{\Delta t \mathcal{L}^O}$) can be simulated *exactly* by breaking down the task into separate processors/threads for each $m \in \mathcal{I}^1$ (resp. $m \in \mathcal{I}^2$). Therefore, this scheme allows us to run independently on each fractional time-step window Δt, and on every processor, a serial KMC simulation, called a *kernel*. The resulting computational framework consisting of the hierarchical decomposition (1.10) and (1.12) permits to input as the algorithm's kernel any preferred optimized serial KMC algorithm.

A single time step of the parallel algorithm is thus easily described in the following stages:

Step 1–Evolution by $\mathcal{L}^E$: Simulate independent Markov processes $\{S_t^m\}_{t \geq 0}$, $m \in \mathcal{I}^1$ by a kinetic Monte Carlo kernel running on non-communicating processors that correspond to each C_m for time Δt.

Step 2–Local Synchronization: communicate configurations σ^E from overlapping domains $\bar{C}_m^E \cap \bar{C}_n^O$ in order to update configurations σ^O.

Step 3–Evolution by $\mathcal{L}^O$: Simulate independent Markov processes $\{S_t^m\}_{t \geq 0}$, $m \in \mathcal{I}^2$ by a KMC *kernel* on non-communicating processors that correspond to each C_m for time Δt.

We emphasize that the resulting process $\{\tilde{S}_t\}_{t \geq 0}$ is an *approximation* of the process $\{S_t\}_{t \geq 0}$ and we discuss it's features and properties in the next two sections.

1.3 Processor Communication Schedule and Random Trotter Products

A key feature of the fractional step methods is the *Processor Communication Schedule* (PCS) that dictates the order with which the hierarchy of operators in (1.6) are applied and for how long. For instance, in (1.12) the processors corresponding to $\mathcal{L}^E$ (resp. $\mathcal{L}^O$) do not communicate, hence the processor communication within the algorithm occurs *only* each time we have to apply $e^{\frac{T}{2n}\mathcal{L}^E}$ or $e^{\frac{T}{2n}\mathcal{L}^O}$. Therefore we can define as the PCS the (deterministic) jump process $X = X(t)$, $t \in [0, T]$, where $[0, T]$ is the simulated time window and taking values in the set $\mathcal{X} = \{1, 2\}$, where we assign the value 1 (resp. 2) to O (resp. E):

$$X(t) = 1 , \quad \frac{2kT}{n} \leq t < \frac{(2k+1)T}{n} , \tag{1.14}$$

$$X(t) = 2 , \quad \frac{(2k+1)T}{n} \leq t < \frac{(2k+2)T}{n} . \tag{1.15}$$

for all $k = 0, \ldots, n-1$. Processor communication occurs at jump times, while in the remaining time the processors operate independently and do not communicate. In an analogous way we can define the PCS for the *Strang* splitting scheme (1.16),

$$e^{T\mathcal{L}} \approx \left[e^{\frac{T}{2n}\mathcal{L}^E} e^{\frac{T}{n}\mathcal{L}^O} e^{\frac{T}{2n}\mathcal{L}^E} \right]^n , \tag{1.16}$$

with the scheduling process

$$X(t) = 1\,, \quad \frac{2kT}{2n} \le t < \frac{(2k+1)T}{2n}\,, \tag{1.17}$$

$$X(t) = 2\,, \quad \frac{(2k+1)T}{2n} \le t < \frac{(2k+3)T}{2n}\,, \tag{1.18}$$

$$X(t) = 1\,, \quad \frac{(2k+3)T}{2n} \le t < \frac{(2k+4)T}{2n}\,, \tag{1.19}$$

for all $k = 0, \ldots, n-1$.

1.3.1 Random Fractional Step Methods

In both cases above (1.12) and (1.16), the communication schedule is fully deterministic, relying on the Trotter Theorem (1.11). On the other hand, we can construct stochastic PCS based on the *Random Trotter Product* Theorem, and as we show below the sub-lattice algorithm proposed in [35] is a fractional step algorithm with stochastic PCS.

The Random Trotter Product Theorem, [17], extends (1.11) as follows: We consider a sequence of semigroups $e^{T\mathcal{L}_\xi}$ with corresponding operators $\mathcal{L}_\xi$ where ξ is in the index set $\mathcal{X}$, assuming for simplicity $\mathcal{X}$ is finite, although a much more general setting is possible, (1.25). Consider also a stochastic jump process $X = X(t)$ with $\mathcal{X}$ as its state space. For each of its trajectories we denote by $\xi_0, \xi_1, ...\xi_n$ the (typically random) sequence of states visited by the stochastic process $X(t)$ and $\tau_0, \tau_1, \ldots, \tau_n$ the corresponding (also typically random) jump times

$$X(t) = \xi_0\,, \quad 0 \le t < \tau_0\,, \tag{1.20}$$

$$X(t) = \xi_1\,, \quad \tau_0 \le t < \tau_1\,, \tag{1.21}$$

$$... \tag{1.22}$$

$$X(t) = \xi_k\,, \quad \tau_{k-1} \le t < \tau_k\,. \tag{1.23}$$

We additionally define as $N(t)$ the number of jumps up to time t. We assume that $X(t)$ is selected so that it has an ergodic behavior, i.e., there is a probability measure $\mu(d\xi)$ such that for all bounded functions g we have that

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t g(X(s))\, ds = \int g(\xi)\, \mu(d\xi)\,. \tag{1.24}$$

For example, if $X(t)$ is a Markov process then under suitable conditions, (1.24) will hold, where μ will be the stationary distribution of $X(t)$, [19]. Conversely, it is well-known that for a given μ we can construct in a non-unique way Markov processes $X(t)$ which satisfy the condition (1.24), [19]. Now we can state the Random Trotter Product Theorem, [17], in analogy to (1.11):

$$e^{T\bar{\mathcal{L}}} = \lim_{n\to\infty} \left[e^{\frac{\tau_0}{n}\mathcal{L}_{\xi_0}} e^{\frac{\tau_1-\tau_0}{n}\mathcal{L}_{\xi_1}} \ldots e^{\frac{nT-\tau_{N(nt)}}{n}\mathcal{L}_{\xi_{N(nt)}}} \right] , \tag{1.25}$$

where the operator $\bar{\mathcal{L}}$ is defined on any bounded function as

$$\bar{\mathcal{L}} g = \int \mathcal{L}_\xi \mu(d\xi)\,. \tag{1.26}$$

It is clear that (1.12) is a special case of (1.25) when $\tau_k - \tau_{k-1} = 1$ and $\xi_{2k} = 1$, $\xi_{2k+1} = 2$ for all k. Similarly, we can also view (1.16) as a deterministic analogue of (1.25).

On the other hand, in the context of the parallel fractional step algorithms for KMC introduced here, the random process (1.20) can be interpreted as a stochastic PCS. For example, the sub-lattice (SL) parallelization algorithm for KMC, introduced in [35], is a fractional step algorithm with stochastic PCS: indeed, in this method the lattice is divided into sub-lattices, for instance as in (1.9), $\Lambda_N = \Lambda_N^E \cup \Lambda_N^O$. Each sub-lattice is selected *at random* and advanced by KMC over a fixed time window Δt. Then a new random selection is made and again the sub-lattice is advanced by Δt, and so on. The procedure is parallelizable as cells C_m^E, C_m^O within each sub-lattice do not communicate. This algorithm is easily recast as a fractional step approximation, when in (1.20) we select deterministic jump times τ_k and random variables ξ_k:

$$\frac{\tau_k - \tau_{k-1}}{n} = \Delta t\,, \quad \text{and} \quad P(\xi_k = 1) = P(\xi_k = 2) = \frac{1}{2}\,. \tag{1.27}$$

As in (1.14), here we assign the value 1 (resp. 2) to the O (resp. E) sub-lattice. Furthermore, we can easily calculate (1.26) to obtain

$$\bar{\mathcal{L}} g = \frac{1}{2}\left(\mathcal{L}^E + \mathcal{L}^O\right)\,,$$

which is just a time rescaling of the original operator $\mathcal{L}$. Thus the SL algorithm is rewritten as the fractional step approximation with the stochastic PCS (1.27) as

$$e^{T\bar{\mathcal{L}}} \approx e^{\frac{\tau_0}{n}\mathcal{L}_{\xi_0}} e^{\frac{\tau_1-\tau_0}{n}\mathcal{L}_{\xi_1}} \dots e^{\frac{nT-\tau_{N(nt)}}{n}\mathcal{L}_{\xi_{N(nt)}}}\,. \tag{1.28}$$

From the numerical analysis viewpoint, our re-interpretation of the SL algorithm in [35] as a fractional step scheme allows us to also provide a mathematically rigorous justification that it is a *consistent* estimator of the serial KMC algorithm, due to the Random Trotter Theorem (1.25). That is, as the time step in the fractional step scheme converges to zero, it converges to the continuous time Markov Chain that has the same master equation and generator as the original serial KMC. Finally, the (deterministic) Trotter Theorem (1.11) also implies that the Lie and the Strang schemes are, in the numerical analysis sense, consistent approximations of the serial KMC algorithm.

1.4 Controlled Error Approximations of KMC

In this section we present a formal argument for the error analysis of the fractional step approximations for KMC, which suggests the order of convergence of the schemes, as well as the restrictions on the Fractional Step KMC time step Δt.In the decomposition (1.10) the operators are linear operators on the high, but finite-dimensional configuration space $\mathcal{S}$, hence by the standard error analysis of splitting schemes, see [11], we have

$$e^{\Delta t\mathcal{L}} - e^{\Delta t\mathcal{L}^E} e^{\Delta t\mathcal{L}^O} = [\mathcal{L}^E\mathcal{L}^O - \mathcal{L}^O\mathcal{L}^E]\frac{(\Delta t)^2}{2} + \mathcal{O}(\Delta t^3)\,, \tag{1.29}$$

where we readily see that the term $[\mathcal{L}^E, \mathcal{L}^O] := \mathcal{L}^E\mathcal{L}^O - \mathcal{L}^O\mathcal{L}^E$ is the Lie bracket (commutator) of the operators $\mathcal{L}^E$, $\mathcal{L}^O$. This Lie bracket captures the effect of the boundary regions $\bar{C}_m^E \cap \bar{C}_n^O$ through which we have processor communication: if there was no communication the Lie bracket would be exactly zero.

Furthermore, instead of (1.12) we can consider the *Strang-type* splitting (1.16). As in the ODE case, [11], this is expected to yield a higher order error term $\mathcal{O}(\Delta t^3)$ instead of the second order approximation

in (1.29), in the following sense:

$$e^{\Delta t\mathcal{L}} - e^{\frac{\Delta t}{2}\mathcal{L}^E} e^{\Delta t\mathcal{L}^O} e^{\frac{\Delta t}{2}\mathcal{L}^E} = \Big\{ \frac{1}{12}[\mathcal{L}^O,[\mathcal{L}^O,\mathcal{L}^E]] - \frac{1}{24}[\mathcal{L}^E,[\mathcal{L}^E,\mathcal{L}^O]] \Big\}(\Delta t)^3 + \mathcal{O}(\Delta t^4)\,. \tag{1.30}$$

Such calculations suggest that the Strang splitting leads to a more accurate scheme, which is balanced by more complicated boundary local communication in the same time window Δt, as is evident when comparing (1.12) and (1.16).

Next, we briefly comment on the error estimation suggested by the calculation (1.29) and return to the rigorous numerical analysis in [1]. In order to obtain an estimate in the right-hand side of (1.29) which is independent of the system size N, it is essential to obtain an upper bound on the total number of jumps up to the time T. This is a key point related to the *extensivity* of the system and to the fact that the weak error analysis is restricted (as it should be physically) to mesoscopic observables satisfying (1.44). We observe the dependence of the error on mesoscopic observables in the following subsection. In the context of coarse-graining, in [14] an analogous estimate was shown rigorously using a Bernstein-type argument applied to the discrete derivatives, in the spirit of (1.44), of the solutions to the backward Kolmogorov equation. We refer to such bounds as "Bernstein-like" due to their similarity to gradient estimates for linear and nonlinear parabolic PDEs.

1.4.1 Error Analysis and comparison between random and deterministic PCS

In this section we further demonstrate the use of the operator splitting formulation as a numerical analysis tool by comparing the time-step of Δt the random PCS introduced in [35] to the deterministic Lie PCS introduced in (1.12). A similar comparison can be made for the Strang scheme (1.16). A detailed discussion including rigorous error estimates for mesoscopic observables such as (1.44), which are independent of the lattice size N will be discussed in [1].

Here we focus on the example of adsorption/desorption discussed in Section 1.2. The generator in the one space dimension is decomposed as in (1.10)

$$\mathcal{L}^E f(\sigma) = \sum_{x\in\Lambda} c^E(x,\sigma)\Big(f(\sigma^x) - f(\sigma)\Big)\,,$$

and

$$\mathcal{L}^O f(\sigma) = \sum_{x\in\Lambda} c^O(x,\sigma)\Big(f(\sigma^x) - f(\sigma)\Big)\,,$$

where

$$c^E(x,\sigma) = \begin{cases} c(x,\sigma), & x\in\Lambda_N^E \\ 0, & \text{otherwise} \end{cases} \qquad c^O(x,\sigma) = \begin{cases} c(x,\sigma), & x\in\Lambda_N^O \\ 0, & \text{otherwise} \end{cases}$$

and the sub-lattices Λ_N^E, Λ_N^O are defined in (1.9). The rates $c(x,\sigma)$ of the corresponding generator (1.3) for the case of Arrhenius adsorption/desorption are given by

$$c(x,\sigma) = c_a(1-\sigma(x)) + c_d\sigma(x)\exp\big(-\beta U(x,\sigma)\big)\,, \tag{1.31}$$

where c_a and c_d are the adsorption and desorption constants respectively, [6]. The desorption potential $U = U(x,\sigma)$ is defined as

$$U(x,\sigma) = \sum_{y\neq x} J(x-y)\sigma(y)\,, \tag{1.32}$$

where $J = J(x-y)$ is the lateral interaction potential; for simplicity we assume that the range of interactions is L, while in typical simplified nearest neighbor models $L = 1$. Similarly we define diffusion dynamics with Arrhenius dynamics, [13].

First we discuss the error analysis for the Lie splitting scheme. For given finite lattice size N, in the decomposition (1.10) the operators are linear operators on the high, but finite-dimensional configuration space $\mathcal{S}$, hence by the standard error analysis of Lie splitting schemes, we obtain (1.29). A more careful study of the commutator reveals that the generator decomposition (1.10) induces significant cancellations in the evaluation of the generator: indeed, we define

$$C_m^o = C_m \setminus \partial C_m \,, \qquad C_m = C_m^o \cup C_m^\partial \,,$$

where in Section 1.2 we introduced $\partial C_m = \bar{C}_m \cap C_m$ and $\bar{C}_m = \{z \in \Lambda_N \,|\, |z-x| \le L \,, x \in C_m\}$. Thus, in (1.10) we obtain the further decomposition

$$\mathcal{L}^E = \mathcal{L}^{E,o} + \mathcal{L}^{E,\partial} := \sum_{m \in \mathcal{I}^1} \mathcal{L}_m^{E,o} + \mathcal{L}_m^{E,\partial} \,, \tag{1.33}$$

where $\mathcal{L}_m^{E,o}, \mathcal{L}_m^{E,\partial}$ is the restriction of $\mathcal{L}^E$ on C_m^o and C_m^∂ respectively. Analogously we define $\mathcal{L}^O = \mathcal{L}^{O,o} + \mathcal{L}^{O,\partial}$. We now return to the evaluation of the commutator

$$[\mathcal{L}^E, \mathcal{L}^O] = [\mathcal{L}^{E,\partial}, \mathcal{L}^{O,\partial}] + [\mathcal{L}^{E,o}, \mathcal{L}^{O,o}] + [\mathcal{L}^{E,\partial}, \mathcal{L}^{O,o}] + [\mathcal{L}^{E,o}, \mathcal{L}^{O,\partial}] \,. \tag{1.34}$$

However, due to the lack of communication between generators beyond the interaction range, we have that

$$[\mathcal{L}^{E,o}, \mathcal{L}^{O,o}] = 0 \,, \quad [\mathcal{L}^{E,\partial}, \mathcal{L}^{O,o}] = 0 \,, \quad [\mathcal{L}^{E,o}, \mathcal{L}^{O,\partial}] = 0 \,,$$

thus we readily get

$$[\mathcal{L}^E, \mathcal{L}^O] = [\mathcal{L}^{E,\partial}, \mathcal{L}^{O,\partial}] = \sum_{m \in \mathcal{I}^1} \sum_{\substack{l \in \mathcal{I}^2 \\ |l-m|=1}} [\mathcal{L}_m^{E,\partial}, \mathcal{L}_l^{O,\partial}] \mathcal{L}_k^{E,\partial}, \mathcal{L}_{k+1}^{O,\partial}] \,. \tag{1.35}$$

The formula (1.35) captures the processor communication between boundary regions of $\bar{C}_m^E$, $\bar{C}_n^O$. But more importantly, when combined with (1.29), it suggests the limitations on the time window Δt of the Lie scheme (1.12), denoted for differentiation by Δt_{Lie}, in order to obtain a given error tolerance TOL. In that sense it is useful to obtain an upper bound on (1.35). Indeed, we readily obtain:

$$\begin{aligned}
[\mathcal{L}^E, \mathcal{L}^O] f(\sigma) = \sum_{\substack{m \in \mathcal{I}^1, l \in \mathcal{I}^2 \\ |l-m|=1}} \sum_{x,y} & \Big[c^E(x,\sigma) c^O(y,\sigma^x) - c^E(x,\sigma^y) c^O(y,\sigma) \Big] f\big((\sigma^x)^y\big) \\
& - \sum_{x,y} c^E(x,\sigma) \Big[c^O(y,\sigma^x) - c^O(y,\sigma) \Big] f(\sigma^x) \\
& - \sum_{x,y} c^O(y,\sigma) \Big[c^E(x,\sigma) - c^E(x,\sigma^y) \Big] f(\sigma^y)
\end{aligned} \tag{1.36}$$

where all summations are over $x \in C_m^{E,\partial}, y \in C_l^{O,\partial}$. For *mesoscopic observables*, such as the mean coverage $f(\sigma) = \frac{1}{N} \sum_{x \in \Lambda} \sigma(x)$ we obtain

$$\begin{aligned}
[\mathcal{L}^E, \mathcal{L}^O] f(\sigma) = \sum_{\substack{m \in \mathcal{I}^1, l \in \mathcal{I}^2 \\ |l-m|=1}} \sum_{x,y} & c^O(y,\sigma) \Big[c^E(x,\sigma) - c^E(x,\sigma^y) \Big] \frac{1-2\sigma(x)}{N} \\
& + \sum_{x,y} c^E(x,\sigma) \Big[c^O(y,\sigma^x) - c^O(y,\sigma) \Big] \frac{1-2\sigma(y)}{N} \,,
\end{aligned} \tag{1.37}$$

where all summations are over $x \in C_m^{E,\partial}, y \in C_l^{O,\partial}$. Therefore, due to the *cancellation* of all interior components $\mathcal{L}^{E,o}, \mathcal{L}^{O,o}$ in (1.35), we obtain the bound for the case of the interaction range $L = 1$,

$$|[\mathcal{L}^E, \mathcal{L}^O]f(\sigma)| \sim O\Big(\frac{M \cdot L}{N}\Big) = O\Big(\frac{1}{q}\Big)\,, \tag{1.38}$$

where q is the size of each cell C_m, and $O(1)$ depends on the physical parameters in the rate (1.31). The local error analysis in (1.29), (1.38) can be propagated up to a prescribed time $T = N_{\text{Lie}}\Delta t_{\text{Lie}}$ Therefore, for the simulation of the mesoscopic observable f up to the time T within a given error tolerance TOL, (1.29) and (1.38) give the *observable*-dependent relation for the Lie time step

$$\text{TOL} \sim T \cdot |[\mathcal{L}^E, \mathcal{L}^O]f(\sigma)|\Delta t_{\text{Lie}} \sim T \cdot O\Big(\frac{1}{q}\Big)\Delta t_{\text{Lie}} \tag{1.39}$$

Next, using the fractional step formulation, we analyze in the same spirit as for the Lie scheme, the random PCS (1.27) proposed in [35]. For notational simplicity we set $A_1 = \mathcal{L}^O$, $A_2 = \mathcal{L}^E$. Then the local error operator $E^{\Delta t}$ can also be calculated as in (1.29):

$$\begin{aligned}\text{Local Error} = E^{\Delta t} :=& e^{\Delta t A_{\xi_1}} e^{\Delta t A_{\xi_2}} - e^{\Delta t(A_1+A_2)} \\ =& \Big(I + (A_{\xi_1} + A_{\xi_2})\Delta t + \frac{1}{2}(A_{\xi_1}^2 + 2A_{\xi_1}A_{\xi_2} + A_{\xi_2}^2)\Delta t^2\Big) - \\ & \Big(I + (A_1 + A_2)\Delta t + \frac{1}{2}(A_1 + A_2)^2\Delta t^2\Big) + O(\Delta t^3)\end{aligned} \tag{1.40}$$

The mean value of the error over the sequence of independent random variables $\xi = (\xi_i\,, i = 1, ..., n)$ of the PCS (1.27) on an observable $f = f(\sigma)\,, s \in \mathcal{S}$ can be explicitly evaluated:

$$\mathbb{E}_\xi[E^{\Delta t} f] = \frac{1}{4}(A_1 - A_2)^2 f\Delta t^2 + O(\Delta t^3) \quad = \frac{1}{4}(\mathcal{L}^E - \mathcal{L}^O)^2 f\Delta t^2 + O(\Delta t^3)\,.$$

As in (1.38), for the mesoscopic observable $f(\sigma) = \frac{1}{N}\sum_{x\in\Lambda}\sigma(x)$, we obtain, after disregarding the higher order local error $O(\Delta t^3)$,

$$(\mathcal{L}^E - \mathcal{L}^O)^2 f(\sigma) \sim O(1)\,, \tag{1.41}$$

where $O(1)$ depends on the physical parameters in the rate (1.31). Similarly to (1.39), for the simulation of the mesoscopic observable f up to the same prescribed time $T = N_{\text{Random}}\Delta t_{\text{Random}}$, within the same error tolerance TOL, (1.29) and (1.41) give the *observable*-dependent relation for the random PCS time step

$$\text{TOL} \sim T \cdot |(\mathcal{L}^E - \mathcal{L}^O)^2 f(\sigma)|\Delta t_{\text{Random}} \sim T \cdot O(1)\Delta t_{\text{Random}} \tag{1.42}$$

Comparing the random and the Lie PCS through (1.39) and (1.42) implies that in order the two schemes to conform (in the mean) to the same tolerance TOL, their respective time steps should be selected so that

$$\Delta t_{\text{Lie}} \sim O(q)\Delta t_{\text{Random}} \tag{1.43}$$

This relation in turn suggests that the Lie scheme (1.12) is expected to parallelize better than the random PCS (1.27) since it allows a q-times larger time step Δt for the same accuracy, keeping in mind that during each time step processors do not communicate.

A similar analysis is possible for general mesoscopic observables $f = f(\sigma)\,, s \in \mathcal{S}$, e.g., spatial correlations, that satisfy

$$\sum_{x\in\Lambda_N} |f(\sigma^x) - f(\sigma)| \le C \tag{1.44}$$

where C is a constant independent of N, see the formulation and estimates for coarse-grained stochastic systems in [14]. We revisit this issue, as well as the rigorous derivation of N-independent error bounds in place of the expansions (1.29), (1.30) in the upcoming publication [1]. Such estimates can also allow a detailed analysis on the balance between accuracy and local processor communication for PCS such as (1.12), (1.16) and (1.27).

1.5 Hierarchical structure of Fractional Step algorithms and implementation on GPUs

The fractional step framework allows a hierarchical structure to be easily formulated and implemented, which is a key advantage for simulating in parallel architectures with complex memory hierarchies and processing units. The Graphical Processing Unit (GPU) architecture is inherently different from a traditional CPU architecture. GPUs are massively parallel multi-threaded devices capable of executing a large number of active threads concurrently. A GPU consists of multiple streaming multiprocessors (MP), each of which contains multiple scalar processor cores. For example, NVIDIA's C2050 GPU architecture contains 14 such multiprocessors, each of which contains 32 cores, for a total of 448 cores which can handle up to 24k active threads in parallel. A GPU has several types of memory which are differently organized compared to the traditional hierarchical CPU memory, most notably the main device memory (global memory) shared between all the multiprocessors and the on-chip memory shared between all cores of a single multiprocessor (shared memory). The memory sizes and access speeds depend on the type of GPU. For instance, the memory size of the NVIDIA C2050 GPU is 3GB while the memory size of the NVIDIA C2070 GPU is 6GB.

From the perspective of a GPU programmer writing a code for NVIDIA GPU's, the GPU is treated as a co-processor to the main CPU. Programs are written in C and linked to the CUDA libraries. A function that executes on the GPU, called a GPU kernel, consists of multiple threads executing code in a single instruction, multiple data (SIMD) fashion. That is, each thread in a GPU kernel executes the same code, but on different data. Further, threads can be grouped into thread blocks. This abstraction takes advantage of the fact that threads executing on the same multiprocessor can share data via on-chip shared memory, allowing some degree of cooperation between threads in the same block [33]. A major drawback in GPU programming is the slow communication between GPU global memory and the main memory of the CPU, compared to the communication within a GPU. Programmers address this problem by maximizing the amount of arithmetic intensive computations performed on GPU, minimizing the communication between CPU and GPU, and allowing the communication latency to be hidden by overlapping with execution. Communication among GPUs, although costly, is enabled by APIs such as OpenMP and features available in CUDA 2.2+ such as portable pinned memory, when the communication is among GPUs connected to the same shared-memory computer node. When the communication takes place among GPUs across nodes of a cluster, message passing paradigms such as MPI can serve the same scope.

In our parallelization of the KMC method, we redefine the data structures to represent lattice sites in the simulation so that the whole simulated system is cut into equal-sized black and white coarse cells like a chessboard in (1.9). For instance, Fig. 1.1a shows a simple example in which we map a 4×4 lattice sites into 2×2 cells, each cell containing 2×2 sites. One GPU thread is assigned to one cell. Coverage information of the whole lattice is stored in an array located in the GPU global memory so that all the threads can access the information related to their neighboring sites across MPs. The GPU kernel performing the KMC simulation over the whole lattice by using the Lie scheme (1.12) and the decomposition (1.10), is sequentially launched twice for each synchronization time step Δt to work on the black and white cells respectively. The

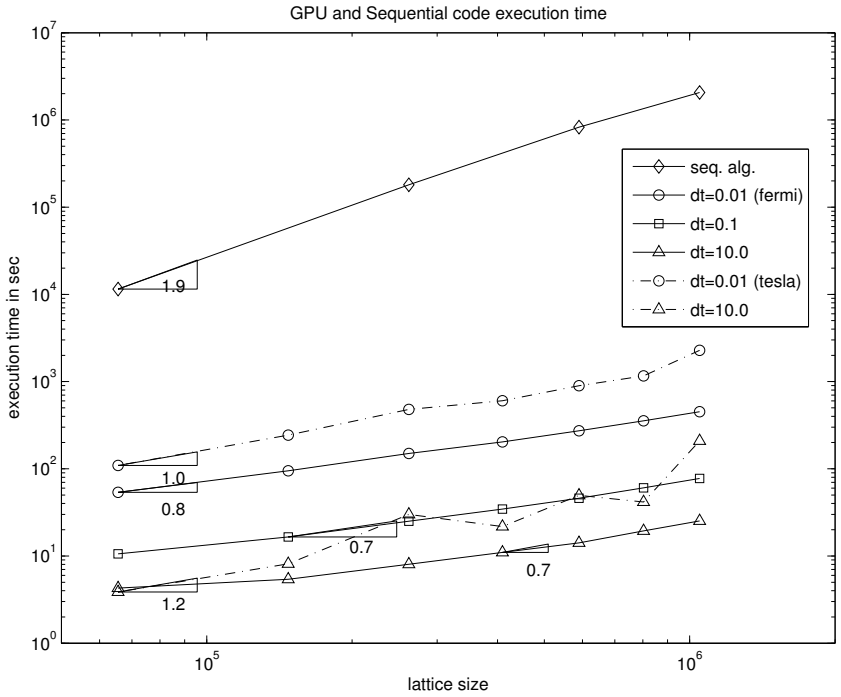


Figure 1.2: Execution time of the fractional step KMC for lattices of different sizes. The comparison with the sequential algorithm (top curve) is based on the same SSA KMC implementation which, however, does not have the optimal complexity of the BKL algorithm. The simpler implementation of the SSA algorithm was used. The simple implementation has the complexity $\mathcal{O}(N^2)$, where N is the total number of lattice sites. This complexity is reflected in the indicated scaling (the slope in the log-log plot). Note that the due to partitioning of the lattice in the fractional step algorithm the same KMC kernel will scale as $\mathcal{O}(N)$ only, which is in agreement with the observed slope in the plots.

execution times for lattices of different sizes are compared in Fig 1.2, where we take as a reference a sequential KMC-kernel, which is a direct numerical implementation of (1.1) and (1.2). The same kernel is then used for the implementation on GPUs where we compare times for different choices of Δt. We remark that the KMC kernel is not optimized by techniques such as the BKL algorithm, [5, 18], which is also manifested in the scaling with respect to the size of the lattice N. However, the same kernel is used in the Fractional Step algorithm thus here we present comparisons between the same KMC algorithms, one serial and one parallelized by the Fractional Step approach. Clearly any optimized KMC kernel can be used without difficulty in our framework.

The size of lattices that can be simulated on a single GPU is limited by memory, thus in order to simulate large systems it will be necessary to employ a cluster of GPUs communicating, for instance, through an MPI protocol. We will demonstrate next how Fractional Step KMC algorithms can be tailored to an architecture that involves multiple GPUs. We return to the formulation in (1.10), and consider the sub-lattice decomposition (1.9). In this formulation each one of the coarse-cells C_m^E or C_m^O are simulated on a single GPU. Within each one of the GPUs we have the same lattice decomposition as in (1.9), see Figure 1.1b, namely

$$C_m^E = C_m^{EE} \cup C_m^{EO} := \bigcup_{l=1}^{L} D_{ml}^{EE} \cup \bigcup_{l=1}^{L} D_{ml}^{EO} \,, \tag{1.45}$$

and similarly we define a decomposition for C_m^O. Each one of the (sub-)sub-lattices D_{ml}^{EE} and D_{ml}^{EE} corresponds to individual threads within the GPU. Next, (1.9) and (1.45) define *nested sub-lattices*, which yield a hierarchical decomposition of the operator $\mathcal{L}$ into (1.10) and

$$\mathcal{L}_m^E = \mathcal{L}_m^{EE} + \mathcal{L}_m^{EO} := \sum_{l=1}^{L} \mathcal{L}_{ml}^{EE} + \sum_{l=1}^{L} \mathcal{L}_{ml}^{EO} \,, \tag{1.46}$$

and similarly we also define the decomposition for $\mathcal{L}_m^O$. Finally, schemes such as (1.12) and (1.16) give rise to Fractional Step algorithms based on the nested decompositions (1.10) and (1.46). In this case, boundary communication, see Fig. 1.1b, plays a key role in the parallelization of our algorithm when multiple GPUs are required. As we discussed earlier, this scenario happens when the lattice size grows to the point that the lattice data structures no longer fit into a single GPU global memory. In turn, this threshold depends on the type of GPU used, e.g., for a NVIDIA's C2050 GPU the maximum lattice size is currently $8,182 \times 8,182$ cells. To simulate larger systems, we can decompose the domain into regular sub-domains and distribute both the sub-domain cells and associated computation among multiple GPUs, as discussed in (1.46). Boundary communication between two adjacent sub-domains are exchanged between GPUs, see Fig. 1.1b, and supported by either MPI or OpenMP, depending on the fact that the GPUs are located on the same cluster node or across nodes. Thus, the multi-GPU parallel KMC algorithm is based on and benefits from the hierarchical structure of the Fractional Step KMC algorithms discussed in (1.46). At the same time, it can enable the scalability of our simulations to lattice sizes beyond the ones accessible with a single GPU e.g., $8,182 \times 8,182$ sites in a C2050 GPU. The study of performance and scalability of our multi-GPU algorithm and code for different lattice sizes and types of GPU clusters is beyond the scope of this paper.

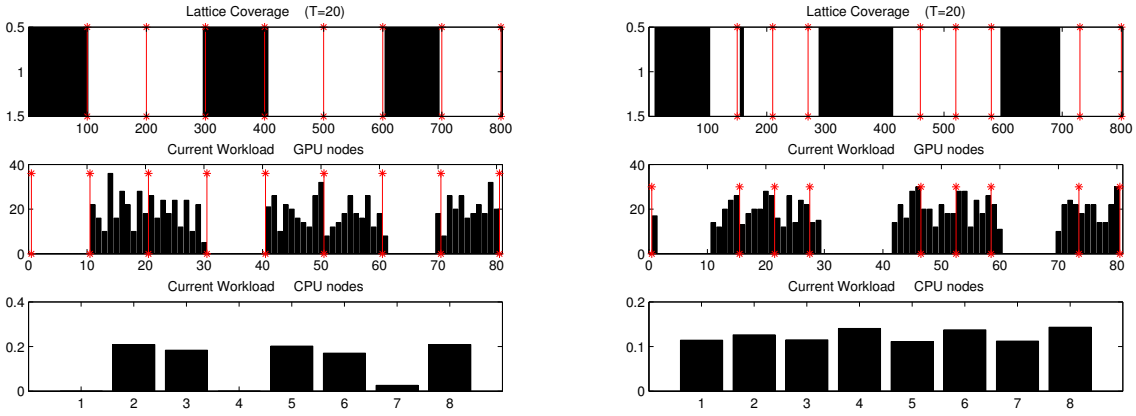
1.6 Mass Transport and Dynamic Workload Balancing

Due to the spatially distributed nature of KMC simulations and the dependence of jump rates on local coverage, (1.1), fractional step algorithms may have an imbalance in the number of operations/jumps performed in each coarse cell C_m in (1.9), as well as on the corresponding processors. In fact, formulas (1.1) and (1.2), and the very structure of the fractional step algorithms (1.10), allow us to define the *workload* $W_{n\Delta t}(\sigma) = W_{n\Delta t}(m;\sigma)$, $1 \le m \le M$ as

$$W_{n\Delta t}(m) = \#\text{ jumps in } C_m \text{ during } [(n-1)\Delta t, n\Delta t] \,, \tag{1.47}$$

when the configuration at time $(n-1)\Delta t$ is σ. We also renormalize $W_{n\Delta t}$ (and still denote it with the same symbol) in order to obtain a histogram, i.e., a probability density. Since different coarse cells C_m in the fractional step algorithms such as (1.12) or (1.16) do not communicate during intervals of length Δt the quantities (1.47) are easy to keep track on-the-fly during the simulations. The possibility of workload imbalance is depicted in Figure 1.3, where many more jumps are performed in the processors corresponding to cells of low coverage, while the other processors remain idle.

In this Section we introduce a probabilistic strategy to re-balance the workload $W_{n\Delta t}$ dynamically during the simulation based on the following idea from *Mass Transport* methods, e.g., [9]. One wants to transport the "imbalanced" density $W_{n\Delta t}$ into an almost uniform density over the number of processors used, in order to ensure that they remain as uniformly active as possible. The mass transport connection and terminology refers to the mapping of a given probability measure into a desirable probability measure. Typically, [9], this problem is posed as an optimization over a suitable cost functional and is known as the Monge-Kantorovich problem. In our context the cost functional could reflect constraints related to various parallel architectures.

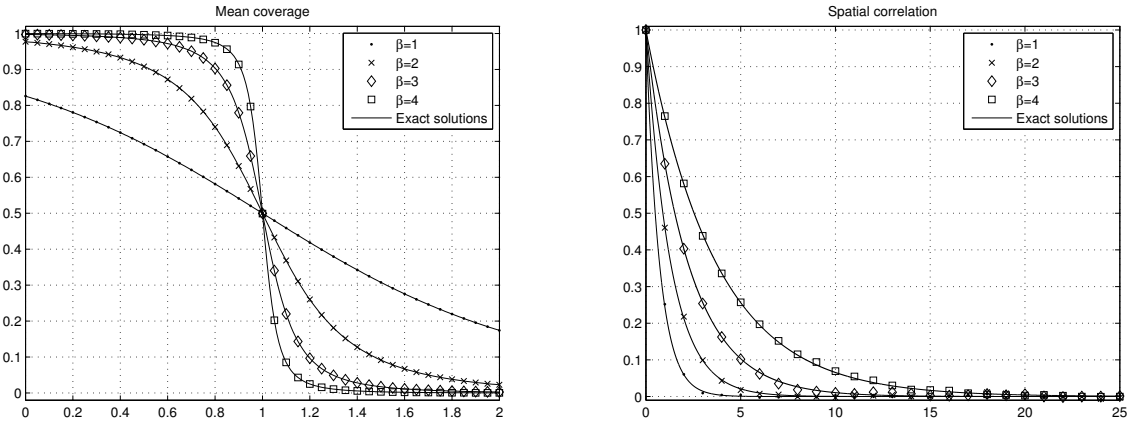


(a) *(b)*

Figure 1.3: (a) Workload imbalance in 1D unimolecular reaction system: the top figure depicts local coverage, the bottom figure workload distribution; (b) Workload redistribution in Figure (a) using the mass transport for re-balancing.

We can formulate and implement this strategy in several different ways: probably the simplest approach, that serves mostly as an illustration, is to assume that we have a number of processors P, where $P \ll M$; during the interval $[(n-1)\Delta t, n\Delta t]$ a number of coarse cells C_m, $1 \leq m \leq M$, which are simulated independently in a fractional step algorithm, are allocated to each processor. By the end of the simulation time $n\Delta t$ the workload on all processors is described similarly to (1.47), by a histogram $R_{n\Delta t}(\sigma) = R_{n\Delta t}(l;\sigma)$, $1 \leq l \leq P$. One wants to map (1.47) onto a histogram $R_{n\Delta t}$ which is almost uniform in $1 \leq l \leq P$. One such function can be constructed by mapping the mass corresponding to each value of the cumulative distribution function (cdf) of (1.47), onto an equal mass on the uniform distribution over the P processors. In another implementation of the mass transport method we can adjust the size of the coarse cells C_m according to the workload redistribution strategy discussed earlier, see Figure 1.3. This is effectively a one-dimensional example of an adsorption/desorption process where the mass transport procedure is carried out by mapping (1.47) into a new histogram $R_{n\Delta t}(\sigma) = R_{n\Delta t}(l;\sigma)$ corresponding to a new set of variable size coarse cells C_l, $1 \leq l \leq M'$. The cell size adjustment ensures the uniformity of the new histogram by defining $R_{n\Delta t}$ as a mapping of the cdf corresponding to (1.47).

The mass transport mappings discussed above are not expected to be carried out at every time step $n\Delta t$ in order to reduce computational and communication cost, but instead they should follow a rationally designed coarser-in-time schedule, in analogy to processor communication scheduling, e.g., (1.20).The overall implementation appears rather simple since here we demonstrated the methodology in a one-dimensional example. However, in higher dimensions, adjusting the size and shape of coarse cells C_m can be much harder. Nevertheless the structure of re-balancing procedure can remain one-dimensional even in higher dimensional lattices if we pick a sub-lattice decomposition (1.5) into strips C_m. We note that the mapping we constructed using cdf's did not take into account the processor architecture and a suitable cost functional formulation for the mass transport to a uniform distribution, as in the Monge-Kantorovich problem, [9], may be more appropriate. We will revisit such issues in a future publication.



(a) Coverage *(b) Two-point spatial correlations*

Figure 1.4: (a) Comparison of the exact solution (1.51) (solid line) for the total coverage $c_\beta(K,h)$, $K=1$, with the mean coverage obtained in simulations on the one-dimensional lattice with $N=2^{15}$ and $\Delta t=1.0$. (b) Two-point spatial correlation function estimated at $h=1$ on the same lattice and $\Delta t=1.0$ compared to the exact solution.

1.7 Parallel Simulations: Benchmarks and Applications

Exactly solvable models of statistical mechanics provide a test bed for sampling algorithms applied to interacting particle systems. We present benchmarks for two important cases: (a) sampling of equilibrium distributions, i.e., long time behavior of the simulated Markov process, and (b) weak approximations of the dynamics. In the first set of tests we work with the classical Ising model on one and two dimensional lattices where spins interact through a nearest-neighbor potential. Thus the Hamiltonian of the system is

$$H(\sigma) = -\frac{K}{2}\sum_{x\in\Lambda_N}\sum_{|y-x|=1}\sigma(x)\sigma(y) + h\sum_{x\in\Lambda_N}\sigma(x)\,,$$

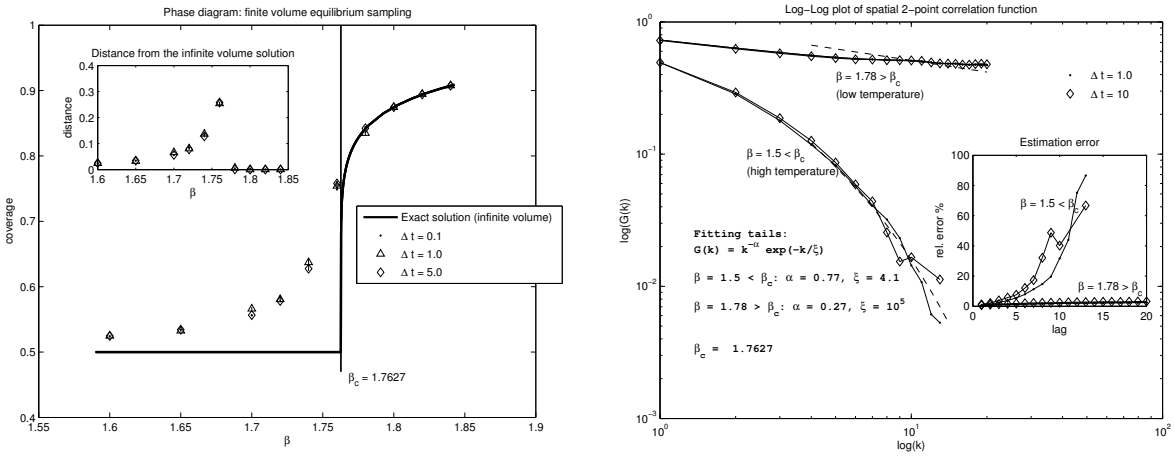
where K is a real parameter that defines the strength of the interaction and h the external field. We work with the spin-flip Arrhenius dynamics with the rates defined in the nearest-neighbor set $\Omega_x=\{z\,|\,|z-x|=1\}$ and the updates in $\mathcal{S}_x=\{0,1\}$.

$$c(x,\sigma) = c_1(1-\sigma(x)) + c_2\sigma(x)e^{-\beta U(x)}\,, \tag{1.48}$$

$$U(x) = K\sum_{y\in\Omega_x}\sigma(x+y) + h\,, \tag{1.49}$$

with β is a given inverse temperature. The generator of (1.48) is a self-adjoint operator on the space $L^2(\mathcal{S},\mu_N)$ where $\mu_N(d\sigma)=Z^{-1}e^{-\beta H(\sigma)}\,d\sigma$ is the canonical Gibbs measure of the system at the constant inverse temperature β. Consequently the dynamics is reversible and the measure μ_t of the process $\{S_t\}_{t\geq 0}$ converges to the Gibbs measure μ_N as $t\to\infty$. Thus the dynamics (1.48) can be used for computing expected values $\mathbb{E}_{\mu_N}[f]$ by invoking ergodicity and averaging on a single trajectory

$$\mathbb{E}_{\mu_N}[f] \equiv \int_{\mathcal{S}} f(\sigma)\,\mu_N(d\sigma) = \lim_{T\to\infty}\frac{1}{T}\int_0^T f(S_t)\,dt\,.$$



(a) Coverage *(b) Two-point spatial correlations*

Figure 1.5: (a) Comparison of the exact solution (1.54) (solid line) for the total coverage $c_\beta(K,h)$, $h=2$, with mean coverage obtained in simulations on the one-dimensional lattice with $N=128$ and various Δt's. (b) Spatial two-point correlation function in the two-dimensional Ising model simulated on the lattice $N=512^2$ at a sub-critical temperature $\beta>\beta_c$ and supercritical regime $\beta<\beta_c$. The simulation confirms the behavior obtained from the infinite volume exact solution: at high temperatures the decay is exponential while at temperatures below the critical temperature the decay is algebraic. The dashed line represents the fitted function of the form $k^{-\alpha}e^{-k/\xi}$.

In the simulations we estimate two observables:

$$\text{mean coverage: } \bar{c}_t = \frac{1}{|\Lambda_N|}\mathbb{E}[\sum_{x\in\Lambda_N}\sigma_t(x)]\,,$$
$$\text{2-point correlation function: } \bar{\lambda}_t(x,y) = \mathbb{E}[\sigma_t(x)\sigma_t(x+y)]\,.$$

Due to translational invariance the function $\bar{\lambda}_k(x,y)$ depends on the distance $|x-y|$ only. For exactly solvable one and two dimensional Ising models we have explicit formulas which we summarize here for the spins in $\Sigma=\{0,1\}$.

1D Ising model: The one-dimensional Ising model does not exhibit a phase transition and thus presents a simple benchmark for accuracy. Working with lattice gas models requires a simple transformation of the well-known exact solution, [4], which for the Hamiltonian of the system given on the periodic lattice

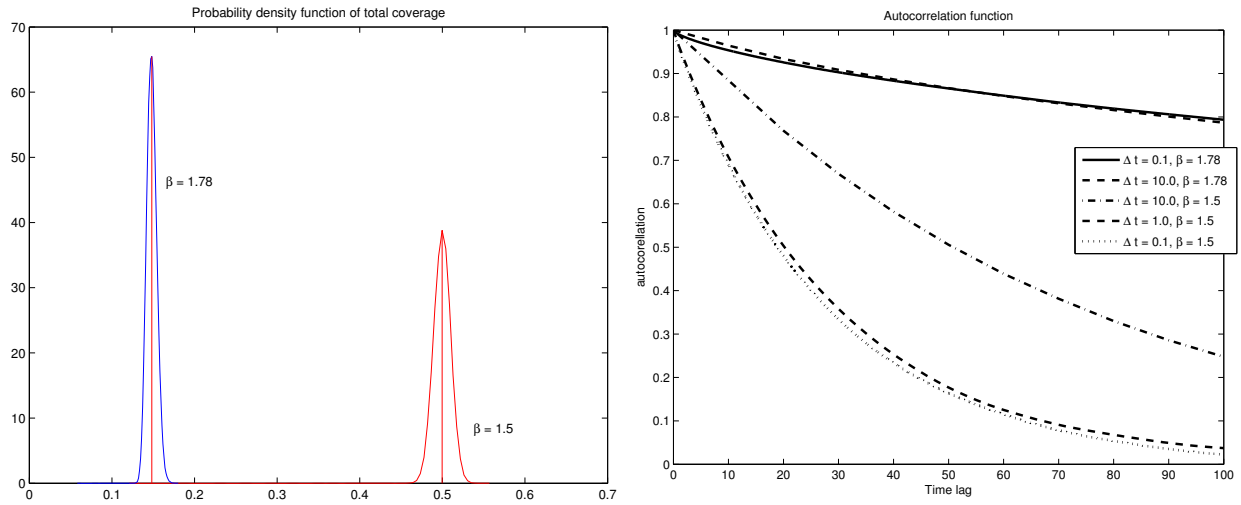
$$H(\sigma) = -K\sum_{x=1}^{N}\sigma(x)\sigma(x+1) + h\sum_{x=1}^{N}\sigma(x)\,,$$

yields the equilibrium mean coverage and the 2-point correlation function

$$\bar{c}(h,\beta) = \frac{1}{2}\left(1+\frac{\sinh(h')}{\left(\sinh^2(h')+e^{-4K'}\right)^{1/2}}\right)\,, \tag{1.50}$$

$$\bar{\lambda}(x,y) = \frac{1}{4}\left(1+e^{4K'}\sinh^2(h')\right)\times \tag{1.51}$$

$$\left[\frac{e^{K'}\cosh(h')-e^{-K'}\left(1+e^{4K'}\sinh^2(h')\right)^{1/2}}{e^{K'}\cosh(h')+e^{-K'}\left(1+e^{4K'}\sinh^2(h')\right)^{1/2}}\right]^{(x-y)}\,,\ y\geq x\,, \tag{1.52}$$



(a) *(b)*

Figure 1.6: (a) Estimated equilibrium distributions of the coverage process at the two temperatures simulated in Fig. 1.5b. (b) Autocorrelation functions for the coverage process in the two-dimensional Ising model simulated at $\beta = 1.5$ (high temperature above the critical temperature β_c and at $\beta = 1.78 > \beta_c$ (low temperature), see parameters in Fig. 1.5b.

where

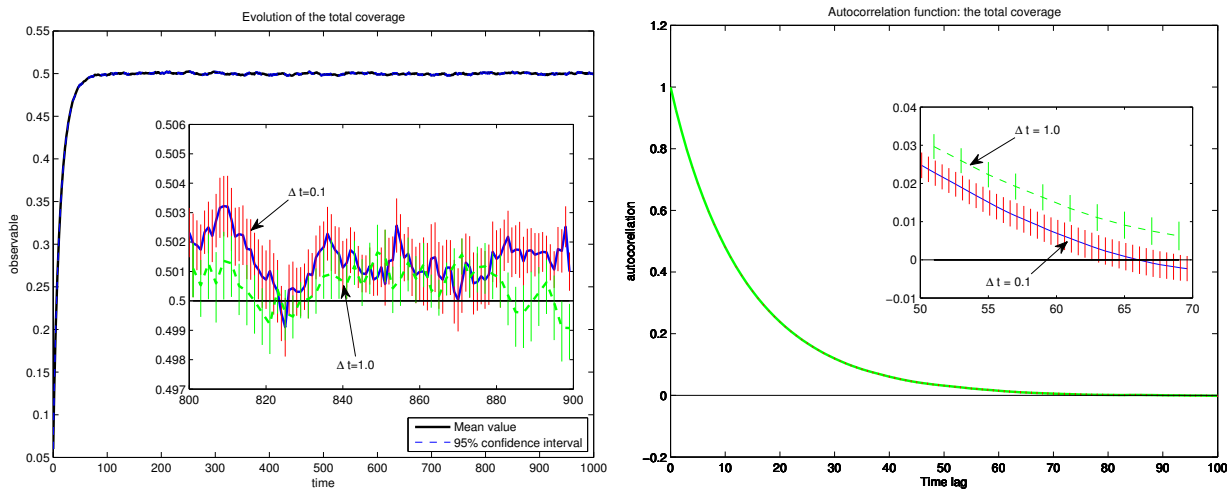
$$K' = \frac{1}{4}\beta K\,, \quad \text{and} \quad h' = \frac{1}{2}\beta(h - K)\,. \tag{1.53}$$

Since the one-dimensional Ising model does not exhibit a phase transition it allows us to assess the accuracy of the approximation for the phase diagram calculation. The phase diagram depicting dependence of the coverage on the external field for different values of β is shown in Figure 1.4a. In this simulation a rather conservative $\Delta t = 1.0$ was chosen. The statistical errors (confidence intervals) are below the resolution of the graph. As seen in the figure the isotherms for the average equilibrium coverage are thus obtained with a good accuracy. As a global observable the total coverage is less sensitive to statistical errors therefore we also monitor the 2-point correlation function and its agreement with the exact solution (1.52). The results for different values of β in Figure 1.4b demonstrate good accuracy.

2D Ising model: The phase transition that occurs in two-dimensional Ising model presents a more challenging test case. However, the celebrated exact solution due to Onsager for spins $\Sigma = \{-1, 1\}$, [28], in the case with the zero external field and further refinements yield closed formulas for the mean coverage and two point correlation functions. We restrict our tests to the isotropic case, i.e., on the two-dimensional periodic lattice we have the Hamiltonian

$$\begin{aligned} H(\sigma) = &- K \sum_{x=(x_1,x_2)\in\Lambda_N} (\sigma(x_1, x_2)\sigma(x_1, x_2 + 1) + \\ &\sigma(x_1, x_2)\sigma(x_1 + 1, x_2)) + h \sum_{x\in\Lambda_N} \sigma(x)\,. \end{aligned}$$

Transforming the exact solutions for the spins $\Sigma = \{0, 1\}$ we obtain the equivalent to the zero external field the value $h = 2K$ at which value the critical inverse temperature solves $\sinh(\frac{1}{2}\beta_c K) = 1$. The exact solution



(a) Sample path *(b) Autocorrelation function*

Figure 1.7: (a) A sample path of the total coverage process $\{S_t\}$ simulated at $\Delta t = 1.0$ and $\Delta t = 0.1$ on the one-dimensional lattice with $N = 2^{15}$ and $\Delta t = 1.0$. (b) Autocorrelation function of the coverage process. The means were obtained from $M = 1000$ independent realizations of the process at $\beta = 4$ and $h = 1$. The inset shows error bars for the empirical mean estimator.

for the mean coverage has the form

$$\bar{c}(\beta) = \begin{cases} \frac{1}{2}\left(1 + \left[1 - (\sinh(\frac{1}{2}\beta K))^{-4}\right]^{1/8}\right), & \beta > \beta_c\,, \\ \frac{1}{2}\,, & \beta < \beta_c\,. \end{cases} \tag{1.54}$$

The exact solution for the 2-point correlation is available in [38], however, we use only the asymptotics in $|x-y|$, [4]. Introducing $\kappa = (\sinh(\frac{1}{2}\beta K))^{-2}$ we have

$$\bar{\lambda}(x,y) = \begin{cases} (1-\kappa^2)^{1/4} + \mathcal{O}(\kappa^{|x-y|})\,, & \beta > \beta_c\,, \\ \mathcal{O}(\kappa^{-|x-y|/2})\,, & \beta < \beta_c\,. \end{cases} \tag{1.55}$$

The phase diagram is computed at $h = 2$ which for $K = 1$ corresponds to the regime when the second-order phase transition occurs at the critical temperature $\sinh(\frac{1}{2}K\beta_c) = 1$. Sampling the coverage exhibits well-known difficulties close to the critical point β_c which are not cured by the fractional step algorithm. Instead, we demonstrate in Figure 1.5a that for wide range of choices Δt the phase diagram is constructed accurately for β outside a neighborhood of β_c. Close to the critical point the algorithm provides approximations that are in agreement with other Monte Carlo sampling approach. The finite-size effects are pronounced at the neighborhood of the critical point due to algebraic decay of correlations. Thus it is not expected that a good agreement with the infinite volume exact solution will be observed in the finite size simulations. Nonetheless, the presence of the second-order phase transition is indicated in the computed phase diagram. Furthermore, the proposed algorithm provides an efficient implementation that allows for simulations on large lattice. It is shown in Figure 1.5b that algebraic decay of the 2-point correlation function is well approximated in the low-temperature (sub-critical) regime, while at super-critical temperatures the exponential decay is observed. Overall, we note that such long-time sampling of the simulated CTMC is a particularly challenging task since in principle, errors from any approximation may accumulate at long times and contaminate the simulation.

Table 1.1: An Event in Ω_x, $x^{nn} \in \Omega_x$ is a randomly selected site from the nearest-neighbor set of x, and $r_2(x) = \frac{1}{4}(1-\sigma(x)^2)\nu_0^x$, $r_3(x) = \frac{1}{8}\sigma(x)(1+\sigma(x))\nu_{-1}^x$, $r_4(x) = \frac{1}{8}\sigma(x)(\sigma(x)-1)\nu_1^x$, where ν_k^x is the number of nearest neighbors (n.n.) of x that are equal to k.

ω	site	$\sigma(x)$	σ^x	Rate $c(x,\omega;\sigma)$	Comment
1	vacant	0	$0 \to 1$	$k_1(1-(\sigma(x))^2)$	CO adsorb
2	vacant	0	$0 \to -1$	$(1-k_1)r_2(x)$	O_2 adsorb
			$0 \to -1$, x^{nn}		
3	CO	1	$1 \to 0$	$k_2 r_3(x)$	CO + O and desorb
			$-1 \to 0$, x^{nn}		
4	O	-1	$-1 \to 0$	$k_2 r_4(x)$	CO + O and desorb
			$1 \to 0$, x^{nn}		

Studying approximation properties of the stochastic dynamics poses a more difficult task due to the lack of an exact solution for the evolution of observables. Certain guidance can be obtained from mean-field approximations, however, those do not give sufficiently good approximation for Ising model in low dimensions. Therefore we compare the evolution of the coverage obtained from the traditional SSA algorithm with approximations generated by the proposed fractional time step algorithm with different choices Δt. In Figure 1.7a we compare the expected value and variance of the total coverage process $C_t = \frac{1}{|\Lambda_N|}\sum_{\Lambda_N} S_t(x)$. Furthermore, it is also shown that the auto-correlation function for the process C_t is well-approximated and approximations converge as $\Delta t \to 0$, see Figures 1.7b and 1.6b.

1.7.1 Examples from Catalysis and Reaction Engineering

In order to demonstrate the applicability of the proposed parallelization methodology in systems exhibiting complex spatio-temporal morphologies at mesoscopic length scales,e.g., islands, spirals, rings, etc., we implement a KMC algorithm arising in the modeling of chemical reaction dynamics on a catalytic surface. Here we focus on CO oxidation, which is a prototypical example for molecular-level reaction-diffusion mechanisms between adsorbates on a surface. We note that molecular dynamics simulations have also been employed to understand micro-mechanisms on surfaces such as reaction paths [29]. However, reaction kinetics for mesoscale adsorbate structures cannot be simulated by using molecular dynamics because of spatio-temporal scale limitations of such methods, while KMC methods, have the ability to simulate much larger scales [22].

In KMC models for CO oxidation on a catalytic surface spatial resolution is a critical ingredient of the modeling since in-homogeneously adsorbed O and CO react on the catalytic surface only where the corresponding phases meet. Sophisticated KMC models for CO oxidation on catalytic surfaces, where kinetic parameters are estimated by *ab initio* density functional theory (DFT), [15], were recently developed in [31] and later in [26], [20]. Such KMC models yield a remarkable agreement with experiments, see also the review articles [25] and [7].

Next we demonstrate the performance of parallel Fractional Step algorithms for KMC simulation to heterogeneous catalysis. We implement a simplified CO oxidation model known as the Ziff-Gulari-Barshad (ZGB) model, [32], which was one of the first attempts towards a spatially distributed KMC modeling in reaction systems. Although a simplified model compared to the *ab initio* KMC models described earlier, it incorporates the basic mechanisms for the dynamics of adsorbate structures during CO oxidation on catalytic surfaces: single site updates (adsorption/desorption) and multi-site updates (specifically, reactions with two sites being involved). The spins take values $\sigma(x) = 0$ denoting a vacant site $x \in \Lambda_N$, $\sigma(x) = -1$ for a

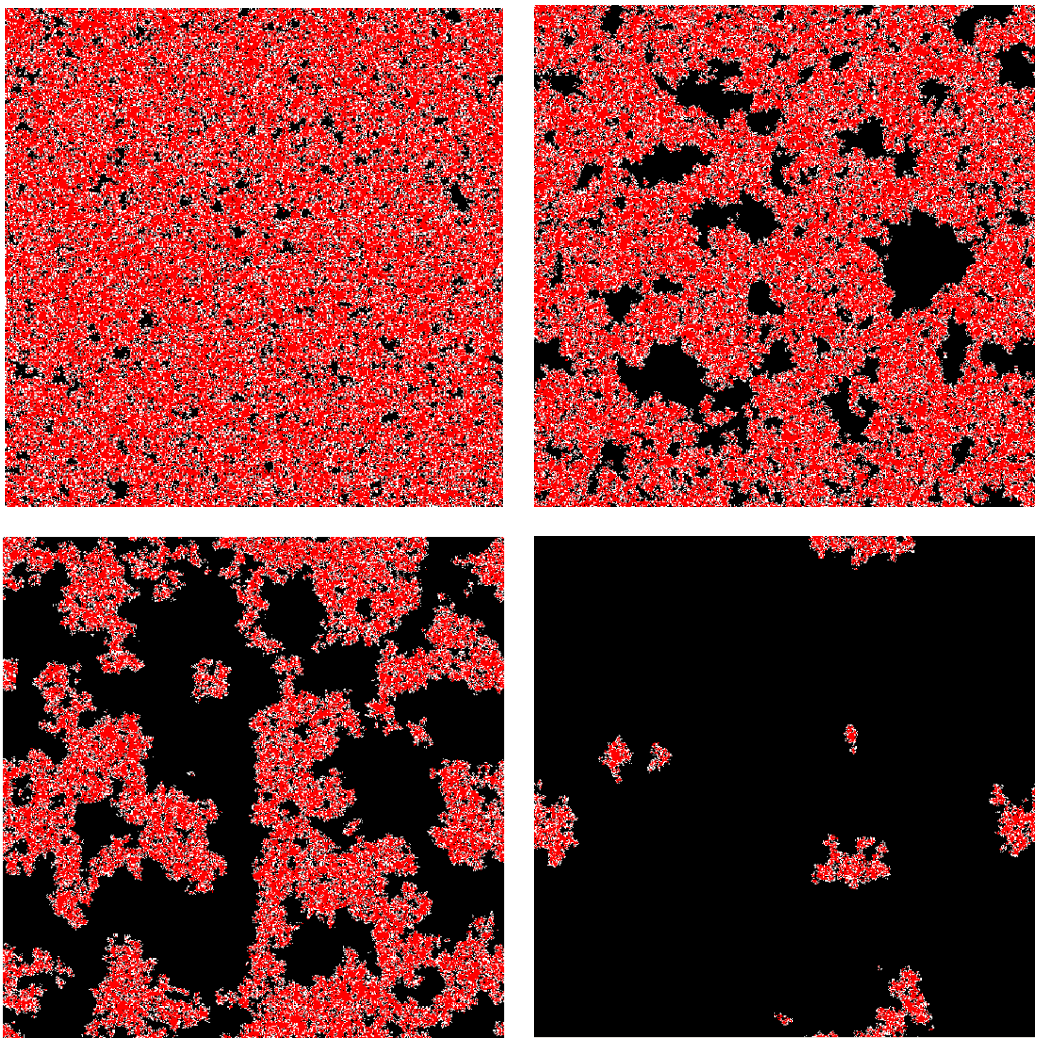


Figure 1.8: Snapshot at different simulation times for the CO *oxidation process, on a two-dimensional lattice* $N = 1024^2$.

molecule CO at x, and $\sigma(x) = 1$ representing a O_2 molecule. Depending on the local configurations of the nearest neighbors in $\Sigma_x = \{y \,|\, |y - x| = 1\}$ the events in Table 1.1 are executed. The rates of individual events depend on the states in Ω_x which are enumerated by $\omega = \{1, 2, 3, 4\}$ and are summarized in Table 1.1.

The execution times for lattices of different sizes are compared in Figure 1.2, while a snapshot of the spatial morphology is depicted in Figure 1.8. Here we take as a reference the sequential KMC-BKL kernel. The same kernel is then used for the implementation on GPUs where we compare times for different choices of Δt. We remark that the KMC kernel is not optimized by techniques such as the BKL algorithm, [18], which is manifested in the scaling with respect to the size of the lattice N. However, the *same kernel* is used in the fractional step algorithm thus we present fair comparisons between serial and parallel solvers, noting that any optimized serial KMC algorithm can be used as a kernel in our Fractional Step framework. It is worth noting that by partitioning of the problem into the subproblems the $\mathcal{O}(N^2)$ complexity of the simple implementation for the SSA algorithm is reduced, which is also demonstrated in Figure 1.2 where the slope of lines for simulations using GPUs suggest the reduced complexity of order $\mathcal{O}(N)$. Hence the proposed approach also offers a simple but efficient implementation of KMC simulators.

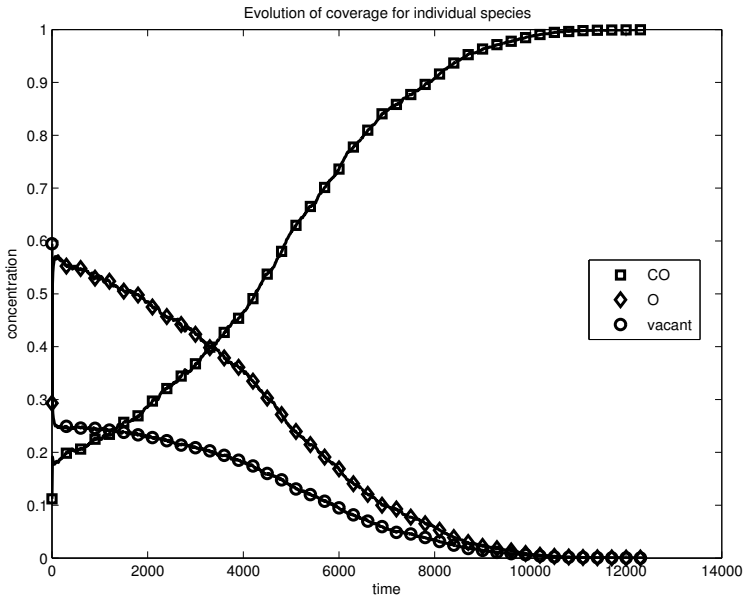


*Figure 1.9: Evolution of the mean coverage for species in the oxidation process (*CO, O_2*, and vacant sites).*

Finally, in our implementation (as well as in the original ZGB model) we did not implement the fast diffusion mechanism of O adsorbates on the surface, [20]. However, the scheme (1.16) can allow us to easily implement within our parallelization framework schemes with disparate time-scales which turn out to be important for the long-time adsorbate dynamics.

1.8 Conclusions

In this paper we proposed a new framework for constructing parallel algorithms for lattice KMC simulations. Our approach relies on a spatial decomposition of the Markov generator underlying the KMC algorithm, into a hierarchy of operators corresponding to processors' structure in the parallel architecture. Based on this operator decomposition, we can formulate Fractional Step Approximation schemes by employing the Trotter product formula; these schemes allow us to run independently on each processor a serial KMC simulation on each fractional time-step window. Furthermore, the schemes incorporate the Communication Schedule between processors through the sequential application of the operators in the decomposition, as well as the time step employed in the particular fractional step scheme. Here we discussed deterministic schedules resulting from Lie- and Strang-type fractional step schemes, as well as random schedules derived by the Random Trotter Theorem, [17]. We demonstrated that the latter category includes the algorithm [35] as one particular example.

Some of the key features of the proposed framework and possible future directions include: The hierarchical structure can be easily derived and implemented for very general physiochemical processes modeled by lattice systems, allowing users to input as the KMC kernel their preferred serial algorithm. This flexibility and hierarchical structure allow for tailoring our framework to particular parallel architectures with complex memory and processor hierarchies, e.g., clusters of GPUs communicating, for instance, through an MPI protocol, and using the nested generator decomposition (1.46). Moreover, multi-scale Trotter algorithms for systems with fast and slow processes are widely used in Molecular Dynamics, e.g., [11], and they can be

recast along with the proposed methods into a spatio-temporal hierarchy of operators that allow computational tasks to be hierarchically decomposed in space/time. The numerical consistency of the proposed algorithms is rigorously justified by Trotter Theorems, [37, 17] showing the convergence of our approximating schemes to the original serial KMC algorithm. Related numerical estimates are expected to provide insights on the design and the relative advantages of various communication schedules and architectures. We discussed work load balancing between processors through a re-balancing scheme based on probabilistic mass transport methods that is particularly well-suited for the proposed fractional step KMC methods. We carried out detailed benchmarking using analytically available exact solutions from statistical mechanics and applied the method to simulate complex spatially distributed molecular systems, such as reaction-diffusion processes on catalytic surfaces. Finally, we studied the performance and scalability of our algorithm (1.46) and the resulting code for different lattice sizes and types of GPUs.

Concluding we note that there are some interesting conceptual analogies between the parallelization and coarse-graining algorithms of KMC such as the Coarse-Grained Monte Carlo (CGMC) method e.g., [13, 2]. In both methods we decompose the particle system in components communicating minimally, e.g., (1.10), (1.12), or trivially as in coarse-graining methods, thus, local information is represented by collective (coarse) variables, or computed on separate processors within a parallel architecture. An early work towards parallelizing CGMC [13] in problems with locally well-mixed particle interactions is [39], while further progress towards understanding and exploiting the analogies and the complementarity of CGMC and parallel KMC has the potential to give efficient KMC algorithms capable of simulating complex systems at mesoscopic length scales.

Acknowledgments: The research of M.A.K. was partially supported by the National Science Foundation under the grant NSF-DMS-071512, by the Office of Advanced Scientific Computing Research, U.S. Department of Energy under DE-SC0002339 and the European Commission FP7-REGPOT-2009-1 Award No 245749. The research of P.P. was partially supported by the National Science Foundation under the grant NSF-DMS-0813893 and by the Office of Advanced Scientific Computing Research, U.S. Department of Energy under DE-SC0001340; the work was partly done at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725. The research of G.A. was partially supported by the National Science Foundation under the grant NSF-DMS-071512 and NSF-CMMI-0835582. The research of M. T. was partially supported by the AFOSR STTR Program, the Army Research Office under the grant 54723-CS, and the NVIDIA University Professor Partnership Program. The research of L.X. was supported by the National Science Foundation under the grant NSF-DMS-0813893.

Bibliography

[1] Giorgos Arampatzis, Markos A. Katsoulakis, Petr Plecháč, and Luc Rey-Bellet. Error analysis for parallel kinetic monte carlo algorithms: accuracy and processor communication. preprint, 2011.

[2] Sasanka Are, Markos A. Katsoulakis, Petr Plecháč, and Luc Rey-Bellet. Multibody interactions in coarse-graining schemes for extended systems. *SIAM J. Sci. Comput.*, 31(2):987–1015, 2008.

[3] S. M. Auerbach. Theory and simulation of jump dynamics, diffusion and phase equilibrium in nanopores. *Int. Rev. Phys. Chem.*, 19(155), 2000.

[4] Rodney J. Baxter. *Exactly Solved Models in Statistical Mechanics*. Academic Press, 3rd edition, 1989.

[5] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for Monte Carlo simulation of Ising spin systems. *J. Comp. Phys.*, 17(1):10–18, 1975.

[6] A. Chatterjee and D. G. Vlachos. An overview of spatial microscopic and accelerated kinetic Monte Carlo methods. *J. Comput.-Aided Mater. Design*, 14:253–308, 2007. 10.1007/s10820-006-9042-9.

[7] C. H. Christensen and J. K. Norskov. A molecular view of heterogeneous catalysis. *Journal of Chemical Physics*, 128(18), 2008.

[8] S. G. Eick, A. G. Greenberg, B. D. Lubachevsky, and A. Weiss. Synchronous relaxation for parallel simulations with applications to circuit-switched networks. *ACM Trans. Model. Comput. Simul.*, 3(4):287–314, 1993.

[9] Lawrence C. Evans. Partial differential equations and Monge-Kantorovich mass transfer. In *Current developments in mathematics, 1997 (Cambridge, MA)*, pages 65–126. Int. Press, Boston, MA, 1999.

[10] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[11] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[12] P. Heidelberger and D. M. Nicol. Conservative parallel simulation of continuous time Markov chains using uniformization. *IEEE Trans. Parallel Distrib. Syst.*, 4:906–921, August 1993.

[13] M. A. Katsoulakis, A. Majda, and D. Vlachos. Coarse-grained stochastic processes for microscopic lattice systems. *Proc. Natl. Acad. Sci*, 100(3):782–782, 2003.

[14] M. A. Katsoulakis, P. Plecháč, and A. Sopasakis. Error analysis of coarse-graining for stochastic lattice dynamics. *SIAM J. Numer. Anal.*, 44(6):2270–2296, 2006.

[15] W. Kohn. Nobel Lecture: Electronic structure of matter-wave functions and density functionals. *Reviews of Modern Physics*, 71(5):1253–1266, 1999.

[16] G. Korniss, M. A. Novotny, and P. A. Rikvold. Parallelization of a dynamic Monte Carlo algorithm: A partially rejection-free conservative approach. *J. Comp. Phys.*, 153(2):488–508, 1999.

[17] T. G. Kurtz. A random Trotter product formula. *Proc. Amer. Math. Soc.*, 35:147–154, 1972.

[18] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, Cambridge, 2000.

[19] Thomas M. Liggett. *Interacting Particle Systems*, volume 276 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1985.

[20] Da-Jiang Liu and J. W. Evans. Atomistic and multiscale modeling of CO-oxidation on Pd(100) and Rh(100): From nanoscale fluctuations to mesoscale reaction fronts. *Surf. Science*, 603:1706–1716, 2009.

[21] B. D. Lubachevsky. Efficient parallel simulations of dynamic Ising spin systems. *J. Comput. Phys.*, 75(1):103–122, 1988.

[22] J. J. Lukkien, J. P. L. Segers, P. A. J. Hilbers, R. J. Gelten, and A. P. J. Jansen. Efficient Monte Carlo methods for the simulation of catalytic surface reactions. *Physical Review E*, 58(2, Part B):2598–2610, AUG 1998.

[23] E. Martínez, J. Marian, M. H. Kalos, and J. M. Perlado. Synchronous parallel kinetic Monte Carlo for continuum diffusion-reaction systems. *J. Comput. Phys.*, 227(8):3804–3823, 2008.

[24] M. Merrick and K. A. Fichthorn. Synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Phys. Rev. E*, 75(1):011606, Jan 2007.

[25] H. Metiu. Preface to special topic: A survey of some new developments in heterogeneous catalysis. *Journal of Chemical Physics*, 128(18), 2008.

[26] Masanari Nagasaka, Hiroshi Kondoh, Ikuyo Nakai, and Toshiaki Ohta. Co oxidation reaction on pt(111) studied by the dynamic monte carlo method including lateral interactions of adsorbates. *J. Chem. Phys.*, 126:044704–7, 2007.

[27] G. Nandipati, Y. Shim, J. G. Amar, A. Karim, A. Kara, T. S. Rahman, and O. Trushin. Parallel kinetic Monte Carlo simulations of Ag(111) island coarsening using a large database. *Journal of Physics Condensed Matter*, 21:084214, 2009.

[28] Lars Onsager. Crystal statistics. I. A two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65(3-4):117–149, Feb 1944.

[29] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos. Iterative minimization techniques for *ab initio* total-energy calculations: molecular dynamics and conjugate gradients. *Rev. Mod. Phys.*, 64:1045–1097, Oct 1992.

[30] S. Plimpton, C. Battaile, M. Chandross, L. Holm, A. Thompson, V. Tikare, G. Wagner, E. Webb, X. Zhou, C. Garcia Cardona, and A. Slepoy. Crossing the Mesoscale No-Man's Land via Parallel Kinetic Monte Carlo. Technical Report SAND2009-6226, Sandia National Laboratory, 2009.

[31] K Reuter, D Frenkel, and M Scheffler. The steady state of heterogeneous catalysis, studied by first-principles statistical mechanics. *Physical Review Letters*, 93(11), SEP 10 2004.

[32] Z. M. Robert, E. Gulari, and Y. Barshad. Kinetic phase transitions in an irreversible surface-reaction model. *Phys. Rev. Lett.*, 56:2553–2556, 1986.

[33] J. Sanders and E. Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional, Cambridge, 2010.

[34] Y. Shim and J. G. Amar. Rigorous synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Phys. Rev. B*, 71(11):115436, Mar 2005.

[35] Y. Shim and J. G. Amar. Semirigorous synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Phys. Rev. B*, 71(12):125432, Mar 2005.

[36] G. Szabo and G. Fath. Evolutionary games on graphs. *Physics Reports*, 446(4-6):97–216, JUL 2007.

[37] H. F. Trotter. On the product of semi-groups of operators. *Proc. Amer. Math. Soc.*, 10:545–551, 1959.

[38] Tai Tsun Wu, Barry M. McCoy, Craig A. Tracy, and Eytan Barouch. Spin-spin correlation functions for the two-dimensional Ising model: Exact theory in the scaling region. *Phys. Rev. B*, 13(1):316–374, 1976.

[39] Lifan Xu, M. Taufer, S. Collins, and D.G. Vlachos. Parallelization of tau-leap coarse-grained monte carlo simulations on gpus. In *Parallel Distributed Processing (IPDPS), 2010 IEEE International Symposium on*, pages 1–9, 2010.

Chapter 2

Parallelization, processor communication and error analysis in lattice kinetic Monte Carlo

In this paper we study from a numerical analysis perspective the Fractional Step Kinetic Monte Carlo (FS-KMC) algorithms proposed in [1] for the parallel simulation of spatially distributed particle systems on a lattice. FS-KMC are fractional step algorithms with a time-stepping window Δt, and as such they are inherently *partially asynchronous* since there is no processor communication during the period Δt. In this contribution we primarily focus on the error analysis of FS-KMC algorithms as approximations of conventional, serial kinetic Monte Carlo (KMC). A key aspect of the presented analysis relies on emphasizing a goal-oriented approach for suitably defined macroscopic observables (e.g., density, energy, correlations, surface roughness), rather than focusing on strong topology estimates for individual trajectories. The presented error analysis allows us to compare different parallelization strategies and their processor communications by relating the algorithm partial asynchrony to the time step Δt and a prescribed error tolerance. Finally, the presented results show that previously developed KMC algorithms based on domain decomposition principles, [25, 26, 21, 22], also allow for simulations with controlled errors for macroscopic of observables, while their partially asynchrony can be also demonstrated and quantified.

2.1 Introduction

The simulation of stochastic lattice systems using kinetic Monte Carlo (KMC) methods relies on the direct numerical simulation of the underlying Continuous Time Markov Chain (CTMC). In [1] we proposed and tested a mathematical and computational framework for constructing parallel algorithms for KMC simulations that relies on domain decomposition of the underlying lattice. Resulting parallel algorithms rely on first developing a spatio-temporal decomposition of the Markov operator for the underlying CTMC–based on a lattice domain decomposition–into a hierarchy of operators corresponding to the particular parallel architecture. Based on this operator decomposition, we formulate *Fractional Step Approximation schemes* by employing the Trotter product formula, which in turn determines the processor *communication schedule*. The fractional step framework allows for a hierarchical structure to be easily formulated and implemented, offering a key advantage for simulating on modern parallel architectures with elaborate memory and proces-

sor hierarchies. The resulting parallel algorithms are inherently *partially asynchronous* as processors do not communicate during the fractional time step window Δt.

The presented parallel algorithm for KMC simulations of spatially distributed stochastic systems addresses limitations caused by the size of the simulated systems, set on lattices. More specifically, memory requirements for realistic simulations exceed the available resources of a single computer node, making a domain decomposition of the lattice necessary. For instance, in a three-dimensional simulation, the local memory is not sufficient to keep all the needed information, we refer to [22], p. 65, Table III; in this simulation the dimension of the system is $403 \times 403 \times 95$, i.e., $N = \mathcal{O}(10^7)$, while in order to resolve a domain of size $1\mu m^3$, a lattice of the size $N = \mathcal{O}(10^{12})$ is necessary.

Simulations of CTMC are based on the efficient calculation of transition probabilities. The computational implementations have evolved from the algorithm, [2], known as a BKL Algorithm, for Ising models, and [6] known as Stochastic Simulation Algorithm (SSA) for reaction systems. Standard lattice KMC algorithms are *inherently serial* as updates of configurations are done at one lattice site at a time, while on the other hand the total transition rate depends on information from the entire spatial domain. For these reasons it appears that KMC cannot be parallelized easily.

However, Lubachevsky, in [18], proposed an asynchronous approach for parallel KMC simulation in the context of Ising systems, in the sense that different processors simulate independently parts of the physical domain, while inconsistencies at the boundaries are corrected with a series of suitable rollbacks. This method relies on the uniformization of the total transition rate; thus the approach yields a null-event algorithm, [15], which includes rejected moves over the entire spatial sub-domain that corresponds to each processor, see also [8]. A modification in order to incorporate the BKL Algorithm was proposed in [18], and tested in [13] leading to an asynchronous algorithm with rejections on the subdomain boundaries. However, these asynchronous algorithms may still have a high number of rejections for boundary events and rollbacks, which considerably reduce the parallel efficiency, [24]. *Synchronous* parallel KMC algorithms that advance processes over a fixed time-window were proposed in [3], [24], [20], [21]. However, the efficiency is plagued by several costly *global* communications, [21]. In the case of spatially homogeneous stochastic systems a different direction has been pursued for parallelization of KMC simulations, for example, applying ideas from time parallel algorithms (parareal methods) to simulations of well-mixed, stochastic chemical kinetics, [4].

In this paper we focus on spatially distributed systems and we adopt an approach that rests on developing a parallel KMC algorithm which *approximates* the underlying continuous time Markov chain of the serial algorithm instead of reproducing its master equation exactly. Earlier, in [25] the authors also proposed an *approximate* algorithm, in order to create a parallelization scheme for KMC. It was demonstrated in [21, 22], that boundary inconsistencies are resolved in a straightforward fashion, while there is an absence of global communications. Finally, among the parallel algorithms tested in [21], the one in [25] had the highest parallel efficiency. We demonstrated that the approximate algorithm in [25] is a special case of the *Fractional Step Approximation* schemes introduced in [1]. We also demonstrated, using the Random Trotter Theorem, [14], that the algorithm in [25] is *consistent* in the approximation limit, i.e., as the time step in the fractional step scheme converges to zero, it converges to a Markov Chain that has the same master equation and generator as the original serial KMC. The presented FS-KMC framework and our error analysis encompass various parallel KMC implementations, including the open source SPPARKS parallel Kinetic Monte Carlo simulator, [22].

Overall, in [1], we proposed the mathematical framework for the asynchronous parallel KMC algorithm, i.e., how the algorithm is connected with domain decomposition and the semigroup theory. However the numerical analysis arguments therein were heuristic and were only applicable to the local error analysis,

that is for a single fractional time step Δt. Furthermore, in [1] we focused on implementation and practical aspects and we demonstrated the algorithm on a single Graphical Processing Unit (GPU), as well as a cluster of GPUs. We also tested the algorithm, using as benchmarks exact (analytically available) solutions and reference solutions from serial KMC, in parameter regimes that included phase transitions regimes and critical phenomena, e.g. the onset of phase transitions.

In the present paper, (a) we provide rigorous *goal-oriented* numerical analysis for the identified class of statistically relevant *macroscopic observables* that include all classical examples of statistical mechanics, e.g., coverage, spatial correlation, surface roughness, etc (Section 5); (b) we prove that the weak error of the numerical method goes to zero as Δt goes to zero (Section 4), that the order (first or second) of the upper bound of the error in the fractional time step Δt depends on the parallel FS-KMC scheme, e.g., based on Lie, Strang and Random operator splittings, and this upper bound of the error is independent of the system (lattice) size for macroscopic observables (Section 5); (c) we employ these results in a practical fashion to investigate and assess the *processor communication* induced by different parallel schemes (Lie, Strang, Random). First, the bigger is the allowable Δt, within a desired error tolerance, the fewer processor communication steps are required, while processor communication itself is mathematically captured by the commutators of the operators involved in FS-KMC (Section 5). Thus, for macroscopic observables, our error analysis allows us to estimate the asynchrony in the parallelization for such observables in terms of the greatest allowable Δt, within a given tolerance (Section 6). Finally, our analysis mathematically justifies a class of existing domain decomposition methods for KMC [24, 25, 26], as well as a well an established parallel KMC solver [22], giving rise to controlled errors and demonstrating and quantifying their partial asynchrony.

2.2 Background

We consider an interacting particle system defined on a d-dimensional lattice Λ_N. Naturally, the simulations are performed on a finite lattice of the size N, however, given the size of real molecular systems it is either necessary to treat the case $N \to \infty$, e.g., $\Lambda = \mathbb{Z}^d$, or alternatively any numerical estimates we obtain need to be *independent of the system size* N. We restrict our discussion to lattice gas models where the order parameter or the spin variable takes values in a compact set, in most cases the set is finite $\Sigma = \{0, 1, \ldots, K\}$. At each lattice site $x \in \Lambda_N$ an order parameter (a spin variable) $\sigma(x) \in \Sigma$ is defined. The states in Σ correspond to occupation of the site $x \in \Lambda_N$ by different species. For example, if $\Sigma = \{0, 1\}$ the order parameter models the classical lattice gas with a single species occupying the site x when $\sigma(x) = 1$ and with the site being vacant if $\sigma(x) = 0$. We denote $\{\sigma_t\}_{t \geq 0}$ the stochastic process with values in the countable configuration space $\mathcal{S} = \Sigma^{\Lambda_N}$. Microscopic dynamics is described by transitions (changes) of spin variables at different sites. We study systems in which the transitions are localized and involve only finite number of sites at each transition step. Mathematically, the microscopic dynamics, i.e., the continuous time Markov chain (CTMC) is a stochastic process $\{\sigma_t\}$ defined completely in terms of the local transition rates $c(\sigma, \sigma')$ which determine the updates (jumps) from any current state $\sigma_t = \sigma$ to a (random) new state σ'. In the context of the spatially distributed applications in which we are interested here, the local transition rates will be denoted as

$$c(\sigma, \sigma') = c(x, \omega; \sigma)\,, \tag{2.1}$$

Thus the *local* dynamics is described by an updating mechanism and corresponding transition rates $c(x, \omega; \sigma)$ in (2.1), such that the configuration at time t, $\sigma_t = \sigma$ changes into a new configuration $\sigma^{x,\omega}$ by an update in a neighborhood of the site $x \in \Lambda_N$. Here $\omega \in \mathcal{S}_x$, where $\mathcal{S}_x$ is the set of all possible configurations that

correspond to an update at a neighborhood Ω_x of the site x. For example, if the modeled process is a diffusion of the classical lattice gas a particle at x, i.e., $\sigma(x)$ can move to any unoccupied nearest neighbor y of x, i.e., $\Omega_x = \{y \in \Lambda_N \,|\, |x-y| = 1\}$ and $\mathcal{S}_x$ is the set of all possible configurations $\mathcal{S}_x = \Sigma^{\Omega_x}$, see [1] for more examples modeling physicochemical processes..

Realizations of the process are constructed from the embedded discrete time Markov chain $S_n = \sigma_{t_n}$ (see [12]), corresponding to jump times t_n. The local transition rates (2.1) define the total rate

$$\lambda(\sigma) = \sum_{x \in \Lambda_N} \sum_{\omega \in \mathcal{S}_x} c(x, \omega; \sigma)\,, \tag{2.2}$$

which is the intensity of the exponential waiting time for a jump from the state σ. The transition probabilities for the embedded Markov chain $\{S_n\}_{n \geq 0}$ are

$$p(\sigma, \sigma^{x,\omega}) = \frac{c(x, \omega; \sigma)}{\lambda(\sigma)}\,. \tag{2.3}$$

In other words once the exponential "clock" signals a jump, the system transitions from the state σ to a new configuration $\sigma^{x,\omega}$ with the probability $p(\sigma, \sigma^{x,\omega})$. On the other hand, the evolution of the entire system at any time t is described by the transition probabilities $P(\sigma, t; \zeta) := \mathbb{P}\left(\sigma_t = \sigma \,|\, \sigma_0 = \zeta\right)$ where $\zeta \in \mathcal{S}$ is an initial configuration. The transition probabilities corresponding to the local rates (2.1) satisfy the Forward Kolmogorov Equation (Master Equation), [16, 5].

In [1] we proposed a mathematical framework for *parallelizable approximations* of the KMC algorithm. Rather than focusing on exactly constructing stochastic trajectories in (2.2) and (2.3), we proposed to approximate the evolution of *observables* $f = f(\sigma) \in C_b(\mathcal{S})$, i.e., of bounded continuous functions on the configuration space $\mathcal{S}$. The space of bounded continuous functions, $C_b(\mathcal{S})$, is regarded as a Banach space with the norm $\| f \|_\infty = \sup_{\sigma \in \mathcal{S}} |f(\sigma)|$. Here we consider observables/functions $f(\sigma)$ depending on large number of variables $\sigma(x)$, $x \in \Lambda_N$, such as coverage, surface roughness, correlations, etc., see for instance the examples in Section 2.5.

Typically in KMC we need to compute expected values of such observables, that is quantities such as

$$u(\zeta, t) := \mathbb{E}^\zeta[f(\sigma_t)] = \sum_\sigma f(\sigma) P(\sigma, t; \zeta)\,, \tag{2.4}$$

conditioned on the initial data $\sigma_0 = \zeta$. By a straightforward calculation we obtain that the expected observable (2.4) satisfies the initial value problem

$$\partial_t u(\zeta, t) = \mathcal{L} u(\zeta, t)\,, \qquad u(\zeta, 0) = f(\zeta)\,, \tag{2.5}$$

where the operator $\mathcal{L} : C_b(\mathcal{S}) \to C_b(\mathcal{S})$ is known as the *generator* of the continuous time Markov chain, [16], and in the case of (2.1) it is

$$\mathcal{L} f(\sigma) = \sum_{\sigma'} c(\sigma, \sigma')[f(\sigma') - f(\sigma)] = \sum_{x \in \Lambda_N} \sum_{\omega \in \mathcal{S}_x} c(x, \omega; \sigma)[f(\sigma^{x,\omega}) - f(\sigma)]\,. \tag{2.6}$$

We then write (2.4), as the the action of the Markov semi-group $e^{t\mathcal{L}}$ associated with the generator $\mathcal{L}$ and the process $\{\sigma_t\}_{t \geq 0}$, [16], on the observable f

$$u(\zeta, t) = \mathbb{E}^\zeta[f(\sigma_t)] = e^{t\mathcal{L}} f(\zeta)\,, \tag{2.7}$$

where $\mathbb{E}^\zeta$ denotes the expected value with respect to the law of the process $\{\sigma_t\}_{t \geq 0}$ conditioned on the initial configuration ζ.

We define a difference operator $\delta_x f$ as an analogue of a derivative. Higher-order derivative analogues are defined in Section 2.5 when needed in the error analysis. We define a corresponding function space, which is necessary in order to set up the semigroup $P = e^{t\mathcal{L}}$ when we consider the infinite lattice $\Lambda = \mathbb{Z}^d$ or to obtain estimates which are independent of the system size N when considering the lattice Λ_N in Section 2.5.

Definition 2.2.1. *Let $f \in C_b(\mathcal{S})$ then for any $x \in \Lambda_N$ we define*

$$\delta_{x,\omega} f(\sigma) = f(\sigma^{x,\omega}) - f(\sigma)\,.$$

We define the norm $\|f\|_1 \equiv \sum_{x,\omega} \| \delta_{x,\omega} f \|_\infty$ and the space of functions on $\mathcal{S} = \Sigma^{\Lambda_N}$

$$C^1(\mathcal{S}) = \{ f \in C_b(\mathcal{S}) \,|\, \|f\|_1 \le C_f \text{ where } C_f \text{ is independent of } N \}\,.$$

Similarly we define the space of functions on $\mathcal{S} = \Sigma^{\Lambda}$ associated with the infinite lattice $\Lambda = \mathbb{Z}^d$

$$C^1(\mathcal{S}) = \{ f \in C_b(\mathcal{S}) \,|\, \|f\|_1 < \infty \}\,.$$

Because of the estimates in Section 2.5, see (2.29) and (2.31) in Theorem 2.5.1, we will later employ spaces with higher discrete derivatives that will be defined in Section 2.5. On the infinite lattice Λ macroscopic observables are all $f \in C^1(\mathcal{S})$. In the case of Λ_N, macroscopic observables are all $f = f(\sigma)$ such that $\|f\|_1$ is independent of the system size N; such typical examples are discussed in Section 2.5.

Typically, the evolution of the particle system on the infinite lattice $\Lambda = \mathbb{Z}^d$ is well-defined, as demonstrated in the next propositions.

Proposition 2.2.1. *For any $f \in C^1(\mathcal{S})$ we have that the series*

$$\mathcal{L} f(\sigma) = \sum_{x \in \Lambda} \sum_{\omega \in \mathcal{S}_x} c(x,\omega;\sigma)[f(\sigma^{x,\omega}) - f(\sigma)]\,,$$

converges uniformly and defines a function in $C_b(\mathcal{S})$, provided $\sup_{x,\omega,\sigma} c(x,\omega;\sigma) < \infty$. Furthermore,

$$\| \mathcal{L} f \|_\infty \le \sup_{x,\omega,\sigma} c(x,\omega;\sigma) \|f\|_1\,.$$

Under the boundedness assumptions on the rates, the closure of the operator $\mathcal{L}$ defines a Markov generator for a Markov semigroup $P \equiv e^{t\mathcal{L}}$, such that for $f \in C^1(\mathcal{S})$, $Pf \in C^1(\mathcal{S})$ and

$$\|e^{t\mathcal{L}} f\|_1 \le e^{\Gamma t} \|f\|_1\,,$$

where Γ is a constant depending on the rates $c(x,\omega;\sigma)$.

Proof. See [16, Theorem 3.9, pp 27]. □

Clearly the same results hold for the finite lattice Λ_N and the corresponding high-dimensional configuration space $\mathcal{S}$, where all constants are independent of the size N.

2.3 Fractional time step kinetic Monte Carlo algorithms

In practice, the sample paths $\{\sigma_t\}_{t \ge 0}$ are constructed by the kinetic Monte Carlo algorithm, that is by simulating the embedded Markov chain defined by (2.2) and (2.3) and advancing the tine by random time-steps from the exponential distribution.

In [1] we proposed a class of parallel KMC algorithms that are based on operator splitting of the Markov generator $\mathcal{L}$ which is based on a geometric decomposition of the lattice Λ_N.

Definition 2.3.1. *The lattice Λ_N is decomposed into non-overlapping coarse cells C_m, $m = 1, \ldots, M$ such that, $|C_m| = Q = q^d$, where d is the dimension,*

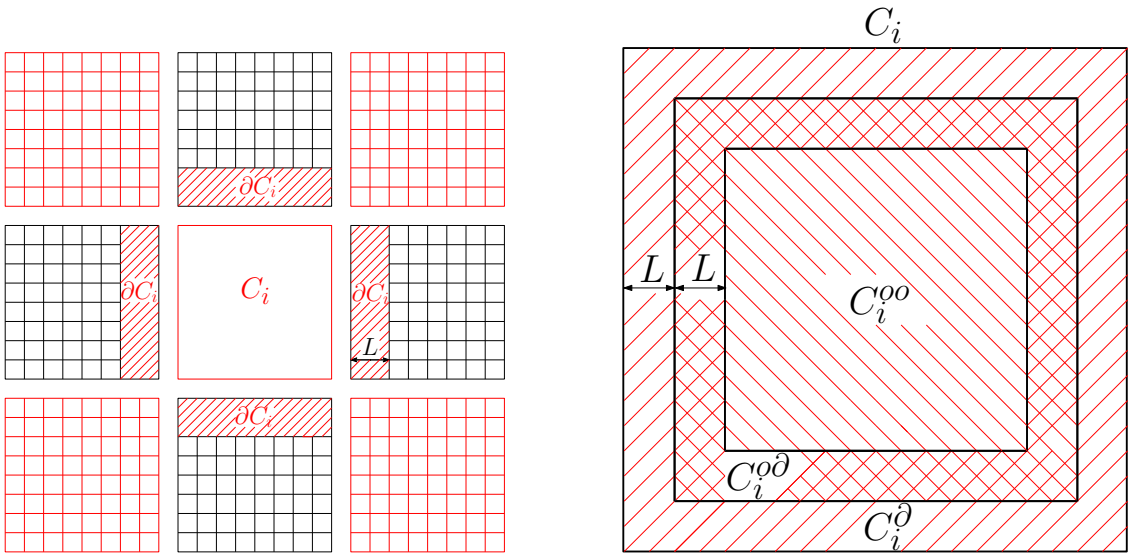
$$\Lambda_N = \bigcup_{m=1}^{M} C_m\,, \quad C_m \cap C_n = \emptyset\,,\ m \neq n\,,\ N = MQ\,. \tag{2.8}$$

The range of interactions is defined as $L = \max_{x\in C_m}\{\operatorname{diam}\Omega_x\}$. For a coarse cell C_m the closure of this set is $\bar{C}_m = \{z \in \Lambda_N \,|\, |z - x| \leq L\,, x \in C_m\}$. The boundary of C_m is then defined as $\partial C_m = \bar{C}_m \setminus C_m$.

The closure $\bar{C}_m$ thus includes all sites of C_m and all "boundary" lattice sites ∂C_m which are connected with sites in C_m through particle interactions in the updating mechanism, see Figure 2.1a. In many models the value of the interaction range L is independent of x due to the translational invariance of the model. This geometric partitioning induces a decomposition of (2.6)

$$\mathcal{L}f(\sigma) = \sum_{m=1}^{M} \mathcal{L}_m f(\sigma)\,, \qquad \mathcal{L}_m f(\sigma) = \sum_{x\in C_m}\sum_{x,\omega\in\mathcal{S}_x} c(\omega;\sigma)[f(\sigma^{x,\omega}) - f(\sigma)]\,. \tag{2.9}$$

The generators $\mathcal{L}_m$ define a new Markov process $\{\sigma_t^m\}_{t\geq 0}$ on the *entire* lattice Λ_N. In many models such as in catalysis, the interactions between particles are short-range, [23, 17], and therefore the transition rates $c(x, \omega; \sigma)$ depend on the configuration σ only through $\sigma(x)$ and $\sigma(y)$ with $|x - y| \leq L$, where L is small (typically one). Similarly the new configuration $\sigma^{x,\omega}$ involves changes only at the sites in this neighborhood. Thus the generator $\mathcal{L}_m$ updates the lattice sites at most in the set $\bar{C}_m = \{z \,|\, |x - z| \leq L\,, x \in C_m\}$. Consequently the processes $\{\sigma_t^m\}_{t\geq 0}$ and $\{\sigma_t^{m'}\}_{t\geq 0}$ corresponding to $\mathcal{L}_m$ and $\mathcal{L}_{m'}$ are *independent provided* $\bar{C}_m \cap \bar{C}_{m'} = \emptyset$.



(a) Lattice partitioning in (2.10). *(b) Sub-lattice partitioning.*

Figure 2.1: Lattice and sub-lattice partitioning. Note we use the notation ∂C_i to denote $\bar{C} \setminus C$,i.e., the interior boundary of C_i.

The operator decomposition yields an algorithm suitable for parallel implementation, in particular, in the case of short-range interactions when the communication overhead can be handled efficiently: if the lattice Λ_N is partitioned into subsets C_m such that $\operatorname{diam} C_m > L$, we can group the sets $\{C_m\}_{m=1}^{M}$ so that there is

no interaction between sites in C_m that belong to the same group. For the sake of simplicity we assume that the lattice is divided into two *sub-lattices* described by the index sets $\mathcal{I}^1$ and $\mathcal{I}^2$ (black/red in each block in Fig. 2.1a), which in turn *induce* a corresponding splitting of the generator:

$$\Lambda_N = \Lambda_N^1 \cup \Lambda_N^2 := \bigcup_{m\in\mathcal{I}^1} C_m \cup \bigcup_{m\in\mathcal{I}^2} C_m \quad \text{and}$$
$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 := \sum_{m\in\mathcal{I}^1} \mathcal{L}_{1,m} + \sum_{m\in\mathcal{I}^2} \mathcal{L}_{2,m}\,. \tag{2.10}$$

The decomposition (2.10) has key consequences for simulating the process $\{\sigma_t\}_{t\geq 0}$ in parallel, as well as formulating different related algorithms. The processes $\{\sigma_t^m\}_{t\geq 0}$ corresponding to the generators $\mathcal{L}_{1,m}$ are mutually independent for different $m \in \mathcal{I}^1$, and thus can be simulated in parallel. Similarly we can handle the processes belonging to the group indexed by $\mathcal{I}^2$. However, there is still *local communication/synchronization* between these two groups as there is non-empty overlap between the groups due to interactions and updates in the sets $\bar{C}_m \cap \bar{C}_{m'}$ when $m \in \mathcal{I}^1$ and $m' \in \mathcal{I}^2$ and the cells are within the interaction range L. Mathematically, we can describe all that through a fractional step approximation of the Markov semigroup $P \equiv e^{t\mathcal{L}}$ of the process $\{\sigma_t\}_{t\geq 0}$. The operator splitting or equivalently the fractional step approximation can be also viewed as an alternating dimension approximation since we solve the evolution of $u(\sigma, t)$ given as solution of (2.5) by alternating between evolution of σ's in the dimensions corresponding to $\mathcal{I}^1$ and $\mathcal{I}^2$.

The key tool for our analysis are different versions of the Trotter formula, [27, 14], $e^{T\mathcal{L}} = \lim_{n\to\infty} \left[e^{\frac{T}{n}\mathcal{L}_1} e^{\frac{T}{n}\mathcal{L}_2}\right]^n$ when applied to the operator $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$ in (2.10). Thus to reach a time T we define a time step $\Delta t = h = \frac{T}{n}$ for a fixed value of n and alternate the evolution by $\mathcal{L}_1$ and $\mathcal{L}_2$, giving rise to the *Lie* splitting approximation for $n \gg 1$:

$$e^{T\mathcal{L}} \approx P_L := \left[e^{\Delta t\mathcal{L}_1} e^{\Delta t\mathcal{L}_2}\right]^n, \quad \text{where} \quad \Delta t = \tfrac{T}{n}\,. \tag{2.11}$$

To develop a parallelizable scheme we use the fact that the action of the operator $\mathcal{L}_1$ (and similarly of $\mathcal{L}_2$) can be distributed onto independent processing units, indexed by m in (2.10),

$$e^{\Delta t\mathcal{L}_1} = \prod_{m\in\mathcal{I}^1} e^{\Delta t\mathcal{L}_{1,m}}\,, \qquad e^{\Delta t\mathcal{L}_2} = \prod_{m\in\mathcal{I}^2} e^{\Delta t\mathcal{L}_{2,m}}\,.$$

Analogously we have the *Strang* splitting scheme

$$e^{T\mathcal{L}} \approx P_S := \left[e^{\frac{\Delta t}{2}\mathcal{L}_1} e^{\Delta t\mathcal{L}_2} e^{\frac{\Delta t}{2}\mathcal{L}_1}\right]^n, \quad \text{where} \quad \Delta t = \tfrac{T}{n}\,. \tag{2.12}$$

From now on, for the notational convenience, we shall also use h to symbolize Δt.

While operator splitting has been exploited in many classical numerical methods, e.g.,[7], in our context it offers a rigorous framework for extending simple (deterministic) alternating strategies associated with, for example, traditional *Lie or Strang* splittings to more elaborate and randomized *Processor Communication Schedules*, we refer to Section 2.6 for a complete discussion.

We characterize the FS-KMC (Fractional Step KMC) algorithm (2.11) as *partially asynchronous* since there is no processor communication during the period $\Delta t \equiv h$. Furthermore, at every h we have only local synchronization between processors, i.e., between the sets $\bar{C}_m \cap \bar{C}_{m'}$ when $m \in \mathcal{I}^1$ and $m' \in \mathcal{I}^2$. Hence, the bigger the allowable h in (2.11) or in (2.12) the less processor communication we have, in which case the error in the approximation (2.11) or (2.12) worsens. This *balance between accuracy and processor communication* in algorithms is one of the themes of this article.

2.4 Local and global error analysis

The FS-KMC algorithm approximates the evolution of observables $u(\sigma, t)$ given by the original semigroup P. We present an error analysis which focuses on classes of observables such as (2.4) instead of estimating an approximation of the probability distribution of the process solving Forward Kolmogorov equation. This perspective is also relevant to practical simulations, where the estimated quantity is linked to specific observables, and is simulated by the FS-KMC algorithm.

We first analyze the error for the two cases of deterministic PCS: the Lie splitting defines a new semigroup (2.11) that we denote P_L and similarly P_S denotes the semigroup (2.12) obtained by the Strang splitting. The local error analysis can be treated in a similar way as it is done for the finite dimensional case when working on the lattice Λ_N by using the property proved in [9]. The estimates for local and global error follow standard steps and are presented next for completeness. However, for *macroscopic observables* that typically arise in the simulation of extended KMC systems, we prove estimates which are system-size independent in Section 2.5.

Lemma 2.4.1. *Let $\mathcal{L}$ be the generator of a strongly continuous contraction semigroup $\{e^{t\mathcal{L}}\}_{t\geq 0}$ on the Banach space C_b. Then the operators*

$$\mathcal{D}_m(t\mathcal{L}) = e^{t\mathcal{L}} - \sum_{k=0}^{m-1} \frac{t^k}{k!}\mathcal{L}^k\,, \qquad m \in \mathbb{N}^+ \tag{2.13}$$

satisfy the bound

$$\| \mathcal{D}_m(t\mathcal{L})v \|_\infty \leq \frac{t^m}{m!} \| \mathcal{L}^m v \|_\infty, \quad \forall v \in C_b \tag{2.14}$$

Proof. see Jahnke, [9]. □

Lemma 2.4.2 (Local Error)**.** *Let $P_L(t)$ and $P_S(t)$ be the schemes (2.11) and (2.12) associated with the Lie and Strang splittings respectively, and let $u(h) = P(h)f$ be the solution of (2.5). Then the local error for the Lie splitting is*

$$\| P_L(h)f - u(h) \|_\infty \leq c_1 \| [\mathcal{L}_1, \mathcal{L}_2]f \|_\infty h^2 + c_2 \sum_{|m|=3} \| \mathcal{L}_1^{m_1}\mathcal{L}_2^{m_2} f \|_\infty h^3\,, \tag{2.15}$$

and for the Strang splitting scheme

$$\begin{aligned} \| P_S(h)f - u(h) \|_\infty \leq c_3 \| [\mathcal{L}_1, [\mathcal{L}_1, \mathcal{L}_2]]f - 2[\mathcal{L}_2, [\mathcal{L}_2, \mathcal{L}_1]]f \|_\infty h^3 \\ + c_4 \sum_{|m|=4} \| \mathcal{L}_1^{m_1}\mathcal{L}_2^{m_2}\mathcal{L}_1^{m_3} f \|_\infty h^4 \end{aligned} \tag{2.16}$$

where $[\mathcal{L}_1, \mathcal{L}_2] = \mathcal{L}_1\mathcal{L}_2 - \mathcal{L}_2\mathcal{L}_1$ denotes the commutator of $\mathcal{L}_1$ and $\mathcal{L}_2$ and $c_i,\ i = 1, ..., 4$ are positive constants with $c_i < 1$.

Proof. Using Lemma 2.4.1 the proof follows the standard finite dimensional approach based on the expansion of the operator exponential, see, e.g., [9]. For the sake of convenience we present the calculations in Appendix 2.A. □

Remark 2.4.1. The previous Lemma demonstrates that the commutators between the operators in the FS-KMC method, capture to leading order the error during processor communication occurring at each time instance $\Delta t = h$. It is clear that within a given error tolerance, the commutator controls what is the maximum allowable h, since the larger the time increment is the less processor communication we have. We return to this issue in Section 2.6.

After establishing the local truncation error it is straightforward to obtain the global error estimate.

Theorem 2.4.1 (Global error)**.** *Let $P_L(t)$ and $P_S(t)$ be the the schemes (2.11) and (2.12) associated with the Lie and Strang splittings respectively and let $u(t_n) = P(t_n)f$ be the exact solution of (2.5). Then the global error at the time $T = t_n = nh$, for the Lie splitting is bounded by*

$$\| P_L(t_n)u(0) - u(t_n) \|_\infty \leq C_1 \max_{k=0,\dots,n} \| [\mathcal{L}_1, \mathcal{L}_2]u(t_k) \|_\infty h + \mathcal{R}_L(u)h^2 \,, \tag{2.17}$$

where the remainder is given by

$$\mathcal{R}_L(u) \equiv \mathcal{R}_L(u; n, h) = C_2 \max_{k=0,\dots,n} \sum_{|m|=3} \| \mathcal{L}_1^{m_1} \mathcal{L}_2^{m_2} u(t_k) \|_\infty \,. \tag{2.18}$$

and for the Strang scheme

$$\begin{aligned} \| P_S(t_n)u(0) - u(t_n) \|_\infty \leq & C_3 \max_{k=0,\dots,n} \| \Big([\mathcal{L}_1, [\mathcal{L}_1, \mathcal{L}_2]] - 2[\mathcal{L}_2, [\mathcal{L}_2, \mathcal{L}_1]] \Big) u(t_k) \|_\infty h^2 \\ & + \mathcal{R}_S(u)h^3 \,, \end{aligned} \tag{2.19}$$

where

$$\mathcal{R}_S(u) = \mathcal{R}_S(u; n, h) = C_4 \max_{k=0,\dots,n} \sum_{|m|=4} \| \mathcal{L}_1^{m_1} \mathcal{L}_2^{m_2} \mathcal{L}_1^{m_3} u(t_k) \|_\infty \,, \tag{2.20}$$

and C_1, C_2, C_3 and C_4 are constants, depending only on T.

Proof. It can be shown by induction that

$$e_n = \tilde{P}^n(h)u(0) - u(t_n) = \sum_{k=0}^{n-1} \tilde{P}^k(h) \Big(\tilde{P}(h) - P(h) \Big) P^{(n-k-1)}(h)u(0) \,.$$

where $\tilde{P}$ denotes either P_L or P_S. By the assumptions, the operators $\mathcal{L}_1$ and $\mathcal{L}_2$ generate strongly continuous contraction semigroups and thus $\| \tilde{P}^k \|_\infty \leq 1$, the global error is bounded by

$$\begin{aligned} \| e_n \|_\infty &\leq \sum_{k=0}^{n-1} \| \Big(\tilde{P}(h) - P(h) \Big) u(t_{n-k-1}) \|_\infty \\ &\leq n \max_{k=0,\dots,n} \| \Big(\tilde{P}(h) - P(h) \Big) u(t_k) \|_\infty \,. \end{aligned}$$

Using Lemma 2.4.2, for $\tilde{P} = P_L$ and $\tilde{P} = P_S$, to estimate the local error and the fact that $nh = T$ we obtain the estimates (2.17) and (2.19) for the Lie and the Strang scheme respectively. □

2.5 Estimates for macroscopic observables

A primary novelty of the presented error analysis rests on a goal-oriented error approach for suitably defined *macroscopic observables*. On one hand, in Theorem 2.4.1 we have shown that the proposed splitting schemes are convergent as the time step h tends to zero. Here we show that if we restrict the initial data of the problem (2.5) to a special class of functions/observables, then it is possible to show that the error terms are independent of the system size, in this case of the lattice size, N. It turns out that this is a broad class containing some of the most common observables in KMC simulations, such as mean coverage or spatial correlations, we refer to Section 2.5.1 below. In Section 2.6 we will see the implications of this independence of the error estimates from the system size, on the selection of the time step $\Delta t = h$ in the FS-KMC

and therefore on the processor communication in the parallel algorithm. Macroscopic observables were also considered in the study of coarse-graining of stochastic interacting particle systems in [11] and [10].

In order to simplify the notation we suppress the dependence of the discrete derivative operator $\delta_{x,\omega}$ on ω in Definition 2.2.1.

Definition 2.5.1. *For $\mathbf{x} = (x_1, \ldots, x_m) \in \Lambda_N^m$ we introduce the notation*

$$\delta_{\mathbf{x}} f(\sigma) = \delta_{x_1} \ldots \delta_{x_m} f(\sigma) = \delta_{x_1 \ldots x_m} f(\sigma)\,,$$

and we refer to it as the discrete derivative of f with respect to $\mathbf{x}$. For example if $\mathbf{x} = (x, y)$ then

$$\delta_{xy} f(\sigma) = \delta_x \delta_y f(\sigma) = f(\sigma^{xy}) - f(\sigma^x) - f(\sigma^y) + f(\sigma)\,.$$

Definition 2.5.2. *Let $\mathbf{x} = (x_1, \ldots, x_m) \in \Lambda_N^m$ and $f \in C_b(\mathcal{S})$. Then we define the norm*

$$\|f\|_m = \sum_{x_1 \in \Lambda_N} \cdots \sum_{x_m \in \Lambda_N} \|\delta_{\mathbf{x}} f\|_\infty\,,$$

and the function space

$$C^m(\mathcal{S}) = \{f \in C_b(\mathcal{S}) \,|\, \sum_{k=1}^{m} \|f\|_k \le C_f \ \text{ where } \ C_f \ \text{ is independent of } \ N\}\,, \forall m \in \mathbb{N}\,.$$

We refer to elements of $C^m(\mathcal{S})$ as *macroscopic observables* and we will discuss examples in Section 2.5.1. We now present the main theorem of this paper, showing that for such macroscopic observables, or equivalently under smoothness conditions on the initial data, the global error estimates for the Lie and the Strang schemes are *independent* of the dimension of the system, i.e., of the system size N. The proof of this theorem is presented in the next two subsections.

Theorem 2.5.1. *(a) Let $u(t)$ be the solution of*

$$\partial_t u(\zeta, t) = \mathcal{L} u(\zeta, t)\,, \qquad u(\zeta, 0) = f(\zeta)\,, \tag{2.21}$$

with $\mathcal{L} : C_b(\mathcal{S}) \to C_b(\mathcal{S})$ and $u(0) = f \in C^3(\mathcal{S})$. Then the global error estimate of Theorem 2.4.1 for the Lie scheme (2.11) becomes

$$\| P_L(t_n) u(0) - u(t_n) \|_\infty \le C_1 \max_{k=0,\ldots,n} \| [\mathcal{L}_1, \mathcal{L}_2] u(t_k) \|_\infty h + \mathcal{R}_L(u) h^2\,,$$

with

$$\| [\mathcal{L}_1, \mathcal{L}_2] u(t_k) \|_\infty < C\,, \quad \text{and } \mathcal{R}_L(u) < \tilde{C}\,,$$

where both constants C and $\tilde{C}$ are independent of the system size N. Moreover, if $u(0) = f \in C^4(\mathcal{S})$ then for the global error of the Strang scheme

$$\| \Big([\mathcal{L}_1, [\mathcal{L}_1, \mathcal{L}_2]] - 2[\mathcal{L}_2, [\mathcal{L}_2, \mathcal{L}_1]] \Big) u(t_k) \|_\infty < C\,, \quad \text{and } \mathcal{R}_S(u) < \tilde{C}\,,$$

where the constants C and $\tilde{C}$ are independent of the system size N.
(b) If $u(0) = f \in C^m(\mathcal{S})$ and it also satisfies the inequality

$$\max_{z \in \Lambda_N} \|\delta_z u(0, \cdot)\|_\infty + \max_{x,y \in \Lambda_N} \| \delta_{xy} u(0, \cdot) \|_\infty + \max_{x,y,z \in \Lambda_N} \| \delta_{xyz} u(0, \cdot) \|_\infty \le \frac{C}{N}\,, \tag{2.22}$$

then the bounds for the commutators become

$$\|[\mathcal{L}_1, \mathcal{L}_2]u(t,\cdot)\|_\infty \leq C\,\frac{L^{d+1}}{q}\,, \tag{2.23}$$

and

$$\|\,\Big([\mathcal{L}_1,[\mathcal{L}_1,\mathcal{L}_2]] - 2[\mathcal{L}_2,[\mathcal{L}_2,\mathcal{L}_1]]\Big)u(t_k)\,\|_\infty \leq C\,\frac{L^{2d+1}}{q}\,, \tag{2.24}$$

where $\frac{N}{M} = Q = q^d$ *and the constant* C *is independent of* N*. The parameters* L*,* M*,* N*,* q *are defined in Definition 2.3.1, and* d *is the dimension of the lattice* $\Lambda_N \subset \Lambda = \mathbb{Z}^d$*.*

Remark 2.5.1. We note that the constants in the bounds of Theorem 2.5.1 may depend on parameters of the rates (temperature, external field, interaction strength, etc.) as close inspection of the proofs reveal. The constants in the bounds are obtained from $\max_x \|c(x,\cdot)\|_\infty$ and $\max_x \|\delta_x c(x,\cdot)\|_\infty$. For example, in the case of the spin-flip Arrhenius dynamics used in Example 2.6.1 we have the rate

$$c(x,\sigma) = c_d(1-\sigma(x)) + c_a\sigma(x)e^{-\beta U(x,\sigma)}\,,$$

with $U(x,\sigma) = \sum_y J(y-x)\sigma(y) + \bar{h}$, and $J(z)$ a given summable function and $\sigma^x(y) = 1-\sigma(y)$ if $y = x$ and $\sigma^x(y) = \sigma(y)$ otherwise. Thus we can compute explicitly $\delta_x c(y,\sigma) = \delta_x\sigma(y) + e^{-\beta(U(y,\sigma)}[\delta_x\sigma(y)e^{-\beta\delta_x U(y,\sigma)} + \sigma(y)e^{-\beta\delta_x U(y,\sigma)} - \sigma(y)]$, $\delta_x\sigma(y) = 1-2\sigma(x)$ for $y = x$ and $\delta_x\sigma(y) = 0$ otherwise, and similarly $\delta_x U(y,\sigma) = J(y-x)(1-2\sigma(x))$. Consequently the bounds depend on the sign of the interaction potential, and they can be estimated from $\max_{x,\sigma} e^{-\beta U(x,\sigma)}$. For example, in the case of the nonnegative interactions $J(z) \geq 0$ we have $\max_x \|c(x,\cdot)\|_\infty \leq 1$, on the other hand for the nearest-neighbor interactions with the negative strength $J < 0$ we have $\max_x \|c(x,\cdot)\|_\infty \leq e^{2\beta|J|}$ which can become arbitrarily large as $\beta \to \infty$ (i.e., in low temperatures). This dependence will impact sharpness of the error estimates in different parts of the phase diagram, however, it is a natural consequence of the error estimation technique based on a priori estimates. The fact that the error is independent of the system size, and more specifically that the commutator bounds (2.23) and (2.24) are independent of N, see also Remark 2.4.1, is employed in Section 2.6, as means of determining processor communication and asynchrony in the parallelization of Lie and Strang FS-KMC algorithms, as well as comparing them also to serial (SSA) KMC simulations, e.g., (2.51).

Remark 2.5.2. We note that the presented numerical analysis yields error control on a finite time interval only. The important question of long-time approximations and thus of ergodicity for the approximation scheme is out of scope of this contribution. However, as shown in [19] the finite time estimates may imply similar long time estimates under suitable ergodicity assumptions. However, mathematical techniques of [19], so far, apply to finite dimensional stochastic differential equations equations. Extending these methods to high-dimensional jump processes studied here is currently an open problem.

The proof of Theorem 2.5.1 is given in Section 2.5.4, while the supporting results are proved earlier in Sections 2.5.2 and 2.5.3. Next, we discuss typical examples of macroscopic observables f which are used in KMC simulations and also satisfy the assumptions of Theorem 2.5.1. Examples of observables that satisfy (2.22) are shown next.

2.5.1 Examples of observables

There is a wide class of macroscopic observable functions in $C^m(\mathcal{S})$, that satisfy

$$\delta_x f(\sigma) := \frac{1}{N}\phi\big(\sigma(x+k_1),\cdots,\sigma(x+k_\ell)\big)\,.k_i \in \Lambda_N\,, \forall x \in \Lambda_N\,, \tag{2.25}$$

A class of functions that satisfies (2.25), or more generally (2.22), includes the coverage, spatial correlations, Hamiltonians and more generally observables of the type

$$f(\sigma) = \frac{1}{N} \sum_{y \in \Lambda_N} U\big(\sigma(y+k_1), \dots, \sigma(y+k_\ell)\big)\,, k_i \in \Lambda_N\,.$$

These functions have the property that their discrete derivatives depend only on a fixed number of points on the lattice that does not scale with N. Here we show for specific examples from this class of functions that they belong in $C^m(\mathcal{S})$.

Example 2.5.1 (Coverage)**.** Let $f(\sigma) = \bar{\sigma} = \frac{1}{N}\sum_{x\in\Lambda_N}\sigma(x)$, the observable that measures the mean coverage of the lattice Λ_N. Then

$$\delta_x f(\sigma) = \frac{1}{N}(\sigma^x(x) - \sigma(x))\,,$$

and in the case $\sigma(x) \in \{0,1\}$ it takes the simple form $\delta_x f(\sigma) = \frac{1}{N}(1-2\sigma(x))$. The *local average* over a percentage of the domain, defined as $f(\sigma) = \frac{1}{N}\sum_{x\in A\subset\Lambda_N}\sigma(x)$. is also in the same class.

Example 2.5.2 (Spatial correlations)**.** Let $f(\sigma;k) = \frac{1}{N}\sum_{x\in\Lambda_N}\sigma(x)\sigma(x+k)$, the mean spatial correlation of length k. Then, when $\sigma(x) \in \{0,1\}$ it takes the form

$$\delta_x f(\sigma) = \frac{1}{N}\big(1-2\sigma(x)\big)\big(\sigma(x+k)+\sigma(x-k)\big)\,.$$

In these examples it is obvious that $f \in C^1(\mathcal{S})$. To such functions we can apply Lemma 2.5.1 and easily conclude that they belong to $C^m(\mathcal{S})$ for $m \le m_0$, where m_0 depends on the form of the observable.

Example 2.5.3. Let f be an observable of type (2.25) with $\ell = 1$ and $k_1 = 0$, then

$$\delta_x\delta_y f(\sigma) = \delta_x \frac{1}{N}\phi(\sigma(y)) = \frac{1}{N}\phi(\sigma^x(y)) - \frac{1}{N}\phi(\sigma(y)) = 0\,, \quad |x-y| > 1\,,$$

hence $f \in C^2(\mathcal{S})$. An analogous result holds when $\ell \ge 1$ and $k_i \neq 0$ with $|x-y| > c(\ell)$, where the constant depends on ℓ but not on N.

Finally, there are macroscopic observables that are not of the type (2.25) but still satisfy (2.22):

Example 2.5.4 (Variance)**.** Let $f(\sigma) = \frac{1}{N}\sum_{x\in\Lambda_N}(\sigma(x)-\bar{\sigma})^2 = \bar{\sigma} - \bar{\sigma}^2$. Then

$$\delta_x f(\sigma) = \frac{1}{N}\big(1-2\sigma(x)\big)\Big(1-2\bar{\sigma}+\frac{2\sigma(x)-1}{N}\Big)\,.$$

It is easy to verify that variance is in $C^2(\mathcal{S})$ and satisfies (2.22).

2.5.2 Bounds on the remainder

In order to establish that the remainders $\mathcal{R}_L(u)$, (2.18), or $\mathcal{R}_S(u)$, (2.20) in Theorem 2.4.1, are bounded by constants independent of N we derive estimates for powers of the operators $\mathcal{L}_1$, $\mathcal{L}_2$ and their compositions such as $\mathcal{L}_1^2\mathcal{L}_2$. The idea for such estimates is an extension of estimates on $\mathcal{L}^2$ acting on the solution of (2.5), which we present next. First, we prove that $\mathcal{L}^2 u$ is bounded by the sum of first and second derivatives of u. Before stating and proving the estimate we show a property of the rate function.

Lemma 2.5.1. *Let c be a rate function with interactions of range L*

$$c(a,\sigma) = \tilde{c}\big(\sigma(a-L),\dots,\sigma(a+L)\big), \quad a \in \Lambda_N\,,$$

then

$$\delta_x c(a,\sigma) = 0, \quad \forall x \in \Lambda_N \ \text{ with } \ |x-a| > L\,,$$

and

$$\delta_{xy} c(a,\sigma) = 0, \quad \forall x,y \in \Lambda_N \ \text{ with } \ |x-y| > 2L+1\,.$$

Moreover, for all higher derivatives holds that

$$\delta_{x_1}\delta_{x_2}\dots\ \delta_{x_n} f(\sigma) \equiv \prod_{k=1}^{n} \delta_{x_k} f(\sigma) = 0\,, \quad |x_i - x_j| > 2L+1\,,\ i \neq j\,.$$

Proof. For the first discrete derivative it is sufficient to observe that if $x \neq y$ then $\sigma^y(x) = \sigma(x)$. Thus when a has distance from x greater than L the rate function $c(a,\sigma)$ is equal to $c(a,\sigma^x)$ and the first derivative is zero.

For the second derivative, based on the calculation for the first derivative, we have

$$\delta_x\Big(\delta_y c(a,\sigma)\Big) = 0\,, \quad |y-a| > L\,,$$

or, if we interchange x and y,

$$\delta_y\Big(\delta_x c(a,\sigma)\Big) = 0\,, \quad |x-a| > L\,.$$

Finally, the second derivative is always zero when $|x-y| > 2L+1$.

For the general case, the proof follows from the fact that $\delta_x\delta_y c(a,\sigma) = \delta_y\delta_x c(a,\sigma)$ and from the following observation

$$\prod_{\substack{k=1\\k\neq i,j}}^{n} \delta_{x_k}\Big(\delta_{x_i}\delta_{x_j} c(a,\sigma)\Big) = 0\,, \quad |x_i - x_j| > 2L+1\,, \quad i \neq j\,,$$

which is true by the result for the second derivative. □

Lemma 2.5.2. *Let u be the solution of equation (2.5). Then for the operator $\mathcal{L}^2$ the following bound holds*

$$\|\mathcal{L}^2 u(t,\cdot)\|_\infty \leq c_1 \|u(t,\cdot)\|_1 + c_2 \|u(t,\cdot)\|_2\,. \tag{2.26}$$

where c_1 and c_2 are independent of N. We denote $\|u(t,\cdot)\|_1 \equiv \sum_{x\in\Lambda_N} \|\delta_x u(t,\cdot)\|_\infty$, and $\|u(t,\cdot)\|_2 \equiv \sum_{x,y\in\Lambda_N} \|\delta_{xy} u(t,\cdot)\|_\infty$.

Proof. By a straightforward calculation

$$\mathcal{L}^2 u(t,\sigma) = \sum_{x,y\in\Lambda_N} c(x,\sigma) c(y,\sigma^x) \delta_{xy} u(t,\sigma) - \sum_{x,y\in\Lambda_N} c(x,\sigma)\delta_x c(y,\sigma)\delta_y u(t,\sigma)\,,$$

and by taking norms on both sides

$$\begin{aligned}
\|\mathcal{L}^2 u(t,\cdot)\|_\infty &\leq \| \sum_{x\in\Lambda_N} \sum_{|x-y|\leq L} c(x,\cdot)\delta_y u(t,\cdot)\|_\infty + c_2 \sum_{x,y\in\Lambda_N} \|\delta_{xy} u(t,\cdot)\|_\infty \\
&\leq c_1 \sum_{x\in\Lambda_N} \|\delta_y u(t,\cdot)\|_\infty + c_2 \sum_{x,y\in\Lambda_N} \|\delta_{xy} u(t,\cdot)\|_\infty\,,
\end{aligned}$$

where the first inequality follows from the boundedness of the rates and the fact that $\delta_x c(y,\sigma) = 0$ when $|x-y| > L$, see Lemma 2.5.1, where we show that the derivatives of the rate functions have compact support that depends only on the length of the interaction L. □

Proposition 2.5.1. *Let $u(t,\sigma)$ be the solution of the equation (2.5) with initial data in $C^2(\mathcal{S})$. Then the operator $\mathcal{L}^2$ satisfies the bounds,*

$$\|\mathcal{L}^2 u(t,\cdot)\|_\infty \le C\,, \tag{2.27}$$

and

$$\|u(t,\cdot)\|_1 + \|u(t,\cdot)\|_2 \le C_1\|u(0,\cdot)\|_1 + C_2\|u(0,\cdot)\|_2\,,$$

where C, C_1 and C_2 are constants independent of N.

Proof. We will bound the right hand side of the equation (2.26) thus we need estimates on the first and the second derivatives of u. For the sake of brevity we use a vectorial notation $\mathcal{L}f = \mathbf{c}(\sigma)\cdot\nabla_\sigma f(\sigma) \equiv \sum_x c(x,\sigma)\delta_x f(\sigma)$. The governing equations for u, $v_1 \equiv \delta_x u$, $v_2 \equiv \delta_y u$ and $w \equiv \delta_x\delta_y u$ are

$$\begin{aligned}
\partial_t u &= \mathbf{c}(\sigma)\cdot\nabla_\sigma u \\
\partial_t v_1 &= \mathbf{c}(\sigma)\cdot\nabla_\sigma v_1 + \delta_x\mathbf{c}(\sigma)\cdot\nabla_\sigma u(\sigma^x) \\
\partial_t v_2 &= \mathbf{c}(\sigma)\cdot\nabla_\sigma v_2 + \delta_y\mathbf{c}(\sigma)\cdot\nabla_\sigma u(\sigma^y) \\
\partial_t w &= \mathbf{c}(\sigma)\cdot\nabla_\sigma w + \delta_y\mathbf{c}(\sigma)\cdot\nabla_\sigma v_1(\sigma^y) + \delta_x\mathbf{c}(\sigma)\cdot\nabla_\sigma v_2(\sigma^x) + \delta_{xy}\mathbf{c}(\sigma)\cdot\nabla_\sigma u(\sigma^{xy})\,.
\end{aligned}$$

First, we bound the first derivative writing the solution for $v(t,\sigma)$

$$\delta_x u(t,\sigma) = e^{\mathcal{L}t}u(0,\sigma) + \int_0^t e^{(t-s)\mathcal{L}} \sum_{|y-x|\le N} \delta_x c(y,\sigma)\delta_y u(s,\sigma^x)\,ds\,. \tag{2.28}$$

By taking the norms and summing over all $x\in\Lambda_N$ we have,

$$\sum_{x\in\Lambda_N} \|\,\delta_x u(t,\cdot)\,\|_\infty \;\le\; \sum_{x\in\Lambda_N} \|\,\delta_x u(0,\cdot)\,\|_\infty + c_1\int_0^t \sum_{x\in\Lambda_N}\sum_{|y-x|\le L} \|\,\delta_y u(s,\cdot)\,\|_\infty\,ds\,.$$

Setting

$$\varphi(t) = \|u(t,\cdot)\|_1 = \sum_{x\in\Lambda_N} \|\,\delta_x u(t,\cdot)\,\|_\infty\,, \tag{2.29}$$

we obtain

$$\varphi(t) \le \varphi(0) + \bar{c}_1\int_0^t \varphi(s)\,ds\,.$$

Similarly, for the second derivatives we have, by using Lemma 2.5.1,

$$\begin{aligned}
\partial_t\delta_{xy}u(t,\sigma) = \mathcal{L}\delta_{xy}u(t,\sigma) \;&+\; \sum_{|z-y|\le L}\delta_y c(z,\sigma)\delta_{xz}u(t,\sigma^y) + \sum_{|z-x|\le L}\delta_x c(z,\sigma)\partial_{yz}u(t,\sigma^x) \\
&+\; \sum_{\substack{|z-x|\le L\\ |z-y|\le L}}\delta_{xy}c(z,\sigma)\delta_z u(t,\sigma^{xy})\chi_{C_{2L}}(x,y)\,,
\end{aligned}$$

where $\chi_{C_{2L}}$ is the characteristic function and $C_{2L} = \{(x,y)\in\Lambda_N^2\,|\,|x-y|<2L\}$. The solution of the above equation is expressed as

$$\begin{aligned}
\delta_{xy}u(t,\sigma) = e^{t\mathcal{L}}\delta_{xy}u(0,\sigma) \;&+\; \int_0^t e^{(t-s)\mathcal{L}}\Bigg[\sum_{|z-x|\le L}\delta_y c(z,\sigma)\delta_{xz}u(s,\sigma^y) \\
&+\; \sum_{|z-y|\le L}\delta_x c(z,\sigma)\partial_{yz}u(s,\sigma^x) \\
&+\; \sum_{\substack{|z-x|\le L\\ |z-y|\le L}}\delta_{xy}c(z,\sigma)\partial_z u(s,\sigma^{xy})\chi_{C_{2L}}(x,y)\Bigg]\,ds\,.
\end{aligned}$$

Thus, by using the contraction property of the semigroup and the fact that the discrete derivatives of the rates are bounded functions, we have the estimate

$$\begin{aligned} \| \delta_{xy} u(t,\cdot) \|_\infty \le \| \delta_{xy} u(0,\cdot) \|_\infty &+ c_1 \int_0^t \sum_{|z-x|\le L} \| \delta_{xz} u(s,\cdot) \|_\infty ds \\ &+ c_2 \int_0^t \sum_{|z-y|\le L} \| \delta_{yz} u(s,\cdot) \|_\infty \, ds \\ &+ c_3 \int_0^t \sum_{\substack{|z-x|\le L \\ |z-y|\le L}} \| \delta_z u(s,\cdot) \|_\infty \chi_{C_{2L}}(x,y) \, ds \, . \end{aligned} \tag{2.30}$$

By summing over all $x, y \in \Lambda_N$ and setting

$$\vartheta(t) = \|u(t,\cdot)\|_2 = \sum_{x,y\in\Lambda_N} \| \delta_{xy} u(t,\cdot) \|_\infty \, , \tag{2.31}$$

we obtain

$$\vartheta(t) \le \vartheta(0) + \bar{c}_2 \int_0^t \vartheta(s) ds + \bar{c}_3 \int_0^t \varphi(s) \, ds \, ,$$

where both $\bar{c}_2$ and $\bar{c}_3$ depend on L but not on N. However, using the standard arguments based on Gronwall inequality extended also for systems of evolution equations we have

$$\varphi(t) \le \tilde{c}_1 \varphi(0) = \tilde{c}_1 \|u(0,\cdot)\|_1 < C_1 \, ,$$

where the last inequality follows from the assumption that $u(0,\sigma) \in C^1(\mathcal{S})$. Similarly we obtain

$$\vartheta(t) \le \tilde{c}_2 \vartheta(0) + \tilde{c}_3 \varphi(0) = \tilde{c}_2 \|u(0,\cdot)\|_2 + \tilde{c}_3 \|u(0,\cdot)\|_1 < C_2 \, ,$$

where the bound on $\vartheta(0)$ follows since the initial data are in $C^2(\mathcal{S})$. Finally, we obtain from Lemma 2.5.2,

$$\|\mathcal{L}^2 u(t,\cdot)\|_\infty \le C_1 \varphi(t) + C_2 \vartheta(t) \le C \, .$$

□

Remark 2.5.3. The same result can be obtained if we notice that the function $v(t,\sigma) = \mathcal{L}^2 u(t,\sigma)$ satisfies the equation (2.5). Then the solution can be written as $v(t,\sigma) = e^{t\mathcal{L}^2} v(0,\sigma)$ and by taking the norm on both sides we get the estimate

$$\|\mathcal{L}^2 u(t,\cdot)\|_\infty \le \| \, e^{t\mathcal{L}^2} u(0,\cdot) \, \|_\infty \le \| \, u(0,\cdot) \, \|_\infty \le C \, ,$$

where the second inequality follows from the fact that $\mathcal{L}^2$ generates a contraction semigroup. However, in order to get bounds for quantities like $\mathcal{L}_1 \mathcal{L}_2 u$, it is sufficient to observe from Lemma 2.5.2 that

$$\begin{aligned} \|\mathcal{L}_1 \mathcal{L}_2 u(t,\cdot)\|_\infty &\le c_1 \sum_{\substack{x\in\Lambda_N^1 \\ y\in\Lambda_N^2}} \|\delta_{xy} u(t,\cdot)\|_\infty + c_2 \sum_{x\in\Lambda_N^1} \|\delta_x u(t,\cdot)\|_\infty \\ &\le \|u(t,\cdot)\|_1 + \|u(t,\cdot)\|_2 \end{aligned}$$

and the norms on the right hand side are bounded from Proposition 2.5.1.

Our last goal for this section is to prove that the remainders in the Lie and the Strang scheme, (2.18) and (2.20) respectively, are independent of the size of the lattice. To achieve this, we first have to bound third and fourth powers of combinations of the operators $\mathcal{L}_1$ and $\mathcal{L}_2$ arising in (2.18) and (2.20). Then, as in Remark 2.5.3, using a more general form of Lemma 2.5.2 it is easy to prove that all relevant combinations of $\mathcal{L}_1$ and $\mathcal{L}_2$ are also bounded by constants independent of N.

2.5.3 Bounds on the commutators

The constants in the local error estimate derived in Lemma 2.4.2 involve bounds on the commutators of the splitting operators $\mathcal{L}_1$ and $\mathcal{L}_2$. We prove that these commutators are bounded operators on the spaces $C^m(\mathcal{S})$, independently of the system size N. The error analysis quantifies the intuitive link of the approximation error to the commutator $[\mathcal{L}_1, \mathcal{L}_2]$ of the operators $\mathcal{L}_1$ and $\mathcal{L}_2$, see Remark 2.4.1. As we show next, the commutator is also directly related to the geometric decomposition and the range of particle interactions.

The error estimates in Lemma 2.4.2 link the local error to the commutator of the operators $\mathcal{L}_1$ and $\mathcal{L}_2$. In principle the commutator can be computed explicitly in terms of the rates $c(x, \omega; \sigma)$ although general formulae quickly become complicated. Therefore we give an example for a specific example of single site events, i.e., $\omega = \{x\}$. The example also demonstrates a procedure that is used for more involved cases. First we evaluate the commutators associated with the decomposition of the lattice into disjoint sub-lattices, Definition 2.3.1.

Lemma 2.5.3. *Let $\mathcal{L}_1, \mathcal{L}_2$ be two operators defined by*

$$\mathcal{L}_1 f(\sigma) = \sum_{x \in C_1} c(x, \sigma)[f(\sigma^x) - f(\sigma)], \quad \textit{and} \quad \mathcal{L}_2 f(\sigma) = \sum_{x \in C_2} c(x, \sigma)[f(\sigma^x) - f(\sigma)],$$

and $C_1, C_2 \subset \Lambda_N$ with $\operatorname{dist}(C_1, C_2) > L$. Then $\mathcal{L}_1$ and $\mathcal{L}_2$ commute, i.e., $[\mathcal{L}_1, \mathcal{L}_2] = 0$.

Proof. The proof follows from the straightforward calculation based on the fact that $c(x, \sigma^y) = c(x, \sigma)$ when $x \in C_1$ and $y \in C_2$ or vice versa and $f(\sigma^{xy}) = f(\sigma^{yx})$. By a direct calculation we get

$$\begin{aligned}
\mathcal{L}_1\mathcal{L}_2 f(\sigma) &= \sum_{x \in C_1} c(x, \sigma)\Big[\mathcal{L}_2 f(\sigma^x) - \mathcal{L}_2 f(\sigma)\Big] \\
&= \sum_{y \in C_2} c(y, \sigma)\Big(\sum_{x \in C_1} c(x, \sigma^y)[f(\sigma^{yx}) - f(\sigma^y)] - \sum_{x \in C_1} c(x, \sigma)[f(\sigma^x) - f(\sigma)]\Big) \\
&= \mathcal{L}_2\mathcal{L}_1 f(\sigma).
\end{aligned}$$

□

Lemma 2.5.4. *Let C_1 and C_2 be such that $C_i = C_i^o \cup C_i^\partial$, where $C_i^o := \{x \in C_i \mid \operatorname{dist}(x, (C_i)^c) > L\}$, where A^c is the complement of set A. With further decomposition $C_i^o = C_i^{oo} + C_i^{o\partial}$ where $C_i^{oo} := \{x \in C_i \mid \operatorname{dist}(x, (C_i)^c) > 2L\}$ (see Figure 2.1b). Let $\mathcal{L}_i = \mathcal{L}_i^o + \mathcal{L}_i^\partial$ and $\mathcal{L}_i^o = \mathcal{L}_i^{oo} + \mathcal{L}_i^{o\partial}$, $i = 1, 2$ be the corresponding decomposition of the generator $\mathcal{L}$, then*

$$[\mathcal{L}_1, \mathcal{L}_2] = [\mathcal{L}_1^\partial, \mathcal{L}_2^\partial], \quad \textit{and} \quad [\mathcal{L}_1, [\mathcal{L}_1, \mathcal{L}_2]] = [[\mathcal{L}_1^{o\partial}, \mathcal{L}_1^\partial], \mathcal{L}_2^\partial] + [\mathcal{L}_1^\partial, [\mathcal{L}_1^\partial, \mathcal{L}_2^\partial]].$$

Proof. The proof of the first statement follows directly from Lemma 2.5.3 by observing that $\operatorname{dist}(C_1^o, C_2^o) = 2L$ and $\operatorname{dist}(C_1^o, C_2^\partial) = \operatorname{dist}(C_1^\partial, C_2^o) = L$. For the second statement, using the same lemma, we compute

$$[\mathcal{L}_1, [\mathcal{L}_1, \mathcal{L}_2]] = [\mathcal{L}_1^o, \mathcal{L}_1^\partial\mathcal{L}_2^\partial] - [\mathcal{L}_1^o, \mathcal{L}_2^\partial\mathcal{L}_1^\partial] + [\mathcal{L}_1^\partial, [\mathcal{L}_1^\partial, \mathcal{L}_2^\partial]].$$

The first term on the right hand side can be further simplified where we used the fact that $\mathcal{L}_2^\partial\mathcal{L}_1^o = \mathcal{L}_1^o\mathcal{L}_2^\partial$ and $[\mathcal{L}_1^{oo}, \mathcal{L}_1^\partial] = 0$. The same procedure leads to simplifying the second term but the third cannot be simplified further. Combining all these steps we obtain the result of the proposition. □

The estimation of the commutator in Theorem 2.5.1 requires *local estimates* on the first and second discrete derivatives of the solution to (2.5) by the discrete derivatives of the initial data:

Lemma 2.5.5. *The solution of the equation*

$$\partial_t u = \mathcal{L}u\,, \quad t \in (0,T]\,, \quad u(0,\sigma) = f(\sigma)\,, \tag{2.32}$$

satisfies the bounds

$$\max_{x\in\Lambda_N} \|\delta_x u(t,\cdot)\|_\infty \leq C \max_{x\in\Lambda_N} \|\delta_x u(0,\cdot)\|_\infty \tag{2.33}$$

and

$$\max_{x,y\in\Lambda_N} \|\,\delta_{xy}u(t,\cdot)\,\|_\infty \leq C\big[\max_{x\in\Lambda_N} \|\delta_x u(0,\cdot)\|_\infty + \max_{x,y\in\Lambda_N} \|\,\delta_{xy}u(0,\cdot)\,\|_\infty\big]\,, \tag{2.34}$$

where C is a constant independent of N, however, it may depend exponentially on t.

Proof. Using (2.28) and Lemma 2.5.1, we have

$$\begin{aligned}\|\delta_x u(t,\cdot)\|_\infty \leq \|\delta_x u(0,\cdot)\|_\infty \quad &+ \quad \mathcal{O}(1)\int_0^t \|\delta_x u(s,\cdot)\|_\infty\, ds \\ &+ \quad \mathcal{O}(\frac{1}{L})\int_0^t \sum_{|x-y|\leq L} \|\delta_y u(s,\cdot)\|_\infty\, ds\,.\end{aligned} \tag{2.35}$$

Here the symbol $\mathcal{O}$ is asymptotic in the size of the system $N\to\infty$. Setting $\gamma(t) = \max_{x\in\Lambda_N}\|\delta_x u(t,\cdot)\|_\infty$ we have

$$\gamma(t) \leq \gamma(0) + \mathcal{O}(1)\int_0^t \gamma(s)ds\,.$$

Applying Gronwall's inequality we conclude the proof and obtain the bound $\gamma(t)\leq e^{ct}\gamma(0)$. The inequality (2.34) follows similarly from (2.30) and from Gronwall's inequality. □

The commutator, as shown in Lemma 2.5.4, is a localized quantity that depends only on the boundary sites of the decomposed sub-lattices. Thus the localized estimate in Lemma 2.5.5 gives us a tool in order to reveal the scaling of the commutator when acting on macroscopic observables.

2.5.4 Proof of Theorem 2.5.1

By Lemma 2.5.4, the commutator can be written as $[\mathcal{L}_1^\partial, \mathcal{L}_2^\partial]$, which due to Lemma 2.5.3 is expanded to

$$\begin{aligned}[\mathcal{L}_1^\partial, \mathcal{L}_2^\partial]u(t,\sigma) = \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial \\ |x-y|\leq L}} \quad & c_1(x,\sigma)c_2(y,\sigma^x)\delta_y u(\sigma^x,t) - c_1(x,\sigma)c_2(y,\sigma)\delta_y u(\sigma,t) \\ - \quad & c_1(x,\sigma^y)c_2(y,\sigma)\delta_x u(\sigma^x,t) + c_1(x,\sigma)c_2(y,\sigma)\delta_x u(\sigma,t)\,.\end{aligned}$$

On the other hand, by a straightforward calculation, we have

$$\begin{aligned}\mathcal{L}_1^\partial\mathcal{L}_2^\partial u(t,\sigma) = \quad & \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial \\ |x-y|\leq L}} c_1(x,\sigma)c_2(y,\sigma^x)\delta_{xy}u(t,\sigma) \\ & - \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial \\ |x-y|\leq L}} c_1(x,\sigma)\delta_x c_2(y,\sigma)\delta_y u(t,\sigma)\,.\end{aligned}$$

Taking norms on both sides similarly to Lemma 2.5.2 and using the fact that the rates are bounded functions on $\Lambda_N\times\Sigma$,

$$\|[\mathcal{L}_1,\mathcal{L}_2]u(t,\cdot)\|_\infty \quad \leq \quad C \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial \\ |x-y|\leq L}} \|\delta_{xy}u(t,\cdot)\|_\infty + \|\delta_y u(t,\cdot)\|_\infty \leq C\,, \tag{2.36}$$

where the second inequality follows from Proposition 2.5.1, using the fact that the initial data are *macroscopic observables*, i.e., belong to $C^2(\mathcal{S})$. Similarly, we obtain the commutator estimate for the Strang scheme.

Next, we turn our attention to (2.23). Many observables are in $C^2(\mathcal{S})$, but also satisfy the local bound (2.22) as one can see in Section 2.5.1. Under this assumption, we obtain from (2.36) the bound for the commutator

$$\begin{aligned}\|[\mathcal{L}_1,\mathcal{L}_2]u(t,\cdot)\|_\infty &\le C \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial \\ |x-y|\le L}} \|\delta_{xy}u(t,\cdot)\|_\infty + \|\delta_y u(t,\cdot)\|_\infty \\ &\le C\big[\max_{x,y\in\Lambda_N}\|\delta_{xy}u(0,\cdot)\|_\infty + \max_{y\in\Lambda_N}\|\delta_y u(0,\cdot)\|_\infty\big] \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial \\ |x-y|\le L}} 1\,, \quad (2.37)\end{aligned}$$

where the second inequality follows from Lemma 2.5.5. Using the fact that the initial data belong to $C^2(\mathcal{S})$ and satisfy (2.22), as well as that $|C_m^\partial| = c(d)Lq^{d-1}$, where d is the dimension, we deduce that

$$\|[\mathcal{L}_1,\mathcal{L}_2]u(t,\cdot)\|_\infty \le \frac{\tilde{C}}{N} \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial \\ |x-y|\le L}} 1 \le \frac{\tilde{C}}{N}\times M\times c(d)Lq^{d-1}\times L^d = C\frac{L^{d+1}}{q}\,, \quad (2.38)$$

where we used the fact that $\frac{N}{M} = Q = q^d$. We note that for more general, non-square lattices, the estimate is modified accordingly as the structure of neighbors in the calculation of $|C_m^\partial|$ will evidently change. Finally, the proof of (2.24) follows along the same lines, noting that the the summation in (2.38) is now replaced by summations such as

$$\sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial, z\in\Lambda_1^\partial \\ |x-y|\le L, |x-z|\le L}} 1 \le M\times c(d)Lq^{d-1}\times L^d\times L^d\,.$$

2.6 Processor communication and error analysis

In this section we examine the balance between accuracy and processor communication in the parallel Fractional Step KMC algorithms. Our analysis is based on the local and global error analysis tools we have developed in this article.

A key feature of the fractional step methods is what we define as the Processor Communication Schedule (PCS), which dictates the order with which the hierarchy of operators in (2.10) are applied and for how long. For instance, for the Lie scheme (2.11) the processors corresponding to $\mathcal{L}_1$ (resp. $\mathcal{L}_2$) do not communicate, hence the processor communication within the algorithm occurs *only* each time we have to apply $e^{\Delta t\mathcal{L}_1}$ or $e^{\Delta t\mathcal{L}_2}$. For this reason, we characterize the FS-KMC algorithms (2.11), (2.12) as *partially asynchronous* since there is no processor communication during the period Δt. Furthermore, at every Δt we have only local synchronization between processors, i.e., between the sets $\bar{C}_m \cap \bar{C}_{m'}$ when $m \in \mathcal{I}^1$ and $m' \in \mathcal{I}^2$. Hence, the bigger the allowable Δt in (2.11) or in (2.12) the less processor communication we have, in which case the error in the approximation (2.11) or (2.12) worsens.

In both schemes (2.11), and (2.12), the communication schedule is fully deterministic, relying on the Trotter Theorem. On the other hand, we can construct general randomized PCS based on the *Random Trotter Product* Theorem, [14]. Indeed, the sub-lattice parallelization algorithm for KMC, introduced in [25], is a particular example of a fractional step algorithm with stochastic PCS. In [25, 22] each sub-lattice is selected at random, independently and advanced by KMC over a fixed time window Δt, subsequently a new

random selection is made and again the sub-lattice is advanced by Δt, etc. This algorithm is easily recast as a fractional step approximation, [1].

Here we compare the deterministic and randomized PCS from the point of view of processor communication and error analysis: we specify the same error tolerance TOL for all PCS, which by means of our error analysis selects in each case a possibly different time windows Δt. Larger time windows Δt give rise to algorithms that have less processor communication for the same error tolerance.

2.6.1 Randomized processor communication schedules

A generalization by Kurtz, [14], of the Trotter Theorem suggests numerically consistent schemes in which evolutions are applied not in a deterministic, prescribed, order but as a random composition of individual propagators resulting in a random evolution. Given a pure jump process $X(t)$, with stationary measure $\mu(d\xi)$, and given the infinitesimal generators $\mathcal{L}_k$ we define a random evolution by

$$\mathcal{T}_n(t)f = e^{\tau_0/n\mathcal{L}_{\xi_0}} e^{\tau_1/n\mathcal{L}_{\xi_1}} \dots e^{\tau_{N(nt)}/n\mathcal{L}_{\xi_{N(nt)}}} f\,,$$

where $N(t)$ is the number of jumps up to time t and τ_k are the sojourn (waiting) times at the visited states $(\xi_0, \dots, \xi_{N(t)})$. The random Trotter product theorem yields the expectation semigroup

$$\lim_{n\to\infty} \mathcal{T}_n(t)f = e^{t\bar{\mathcal{L}}} f\,, \quad \text{a.s.} \tag{2.39}$$

with the generator $\bar{\mathcal{L}}$ characterized explicitly

$$\bar{\mathcal{L}} f = \int \mathcal{L}_\xi f \, \mu(d\xi)\,. \tag{2.40}$$

We present the construction in a simpler case of the independent identically distributed random variables that index the individual generators $\mathcal{L}_\xi$. We analyze the randomized Lie scheme for the operator splitting given by $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$. In the context of the parallel FS-KMC the random process $X(t)$ can be interpreted as a *stochastic* PCS. In [1] we demonstrated that the sub-lattice parallelization algorithm for KMC, introduced in [25], is a particular example of a fractional step algorithm with stochastic PCS. From the numerical analysis viewpoint, our re-interpretation of the algorithm in [25] as (2.39) allows us to provide a rigorous justification that it is a *consistent* estimator of the serial KMC algorithm. Next we present the local error analysis of randomized PCS and in analogy to Lemma 2.4.2, we estimate the mean (weak) local error of the approximating γ-process.

Definition 2.6.1 (Random Lie splitting)**.** *Let $P_i(t)$, $i = 1, 2$, be two Markov semigroups with the infinitesimal generators $\mathcal{L}_i$ and the transition probability kernels $p_i(t; \gamma, \gamma')$. Assume $\{\xi_1, \xi_2, \dots\}$ be a sequence of i.i.d. Bernoulli random variables with values $\xi \in \{1, 2\}$. We define the random evolution as the process $\{\gamma_{kh}\}_{k=0}^n$ by setting for $h > 0$, $k = 0, 1, 2, \dots, n$, and ξ_{2k}, ξ_{2k-1} independent of $\gamma_0, \gamma_h, \dots, \gamma_{(k-1)h}$*

$$\mathbb{E}[f(\gamma_{kh}) \,|\, \gamma_{(k-1)h}] := P_{\xi_{2k-1}}(h) P_{\xi_{2k}}(h) f(\gamma_{(2k-1)h})\,, \tag{2.41}$$

where the transition probability kernel is

$$[P_{\xi_1}(h) P_{\xi_2}(h) f](\eta) = \sum_{\gamma'} \sum_{\gamma''} p_{\xi_1}(h; \eta, \gamma') p_{\xi_2}(h; \gamma', \gamma'') f(\gamma'')\,.$$

For a given $f \in C_b(\mathcal{S})$ we estimate the quantity $\mathbb{E}^\sigma[f(\sigma_{kh})]$ and $\mathbb{E}^\gamma[f(\gamma_{kh})]$ where the expected values are computed on the corresponding probability spaces associated with each process and conditioned on the

initial states $\sigma_0 = \sigma$ and $\gamma_0 = \gamma$ respectively. We denote the initial states by different letters in order to distinguish between these two different probability path measures, however, the initial state is assumed to be same for both $\{\sigma_t\}_{t\geq 0}$ and $\{\gamma_{kh}\}_{k=0}^n$.

Theorem 2.6.1 (Local Error)**.** *Assume* $\mathbb{P}(\xi_k = 1) = \mathbb{P}(\xi_k = 2) = \frac{1}{2}$*, for the approximating process* $\{\gamma_{kh}\}_{k=0}^n$ *of Definition 2.6.1. Then for any* $f \in C_b(\mathcal{S})$ *and given* $\Delta t = h > 0$*, the exact process* $\{\sigma_t\}_{t\geq 0}$ *with* $\sigma_0 = \gamma_0 = \gamma$ *corresponding to the generator* $\frac{1}{2}\mathcal{L}$ *satisfies*

$$\begin{aligned}\mathbb{E}^\gamma[f(\gamma_h)] - \mathbb{E}^\sigma[f(\sigma_h)] &= \mathbb{E}^\xi\left[(P_{\xi_1}(h)P_{\xi_2}(h)f(\gamma) - u(\gamma,h))\right] \\ &= \frac{h^2}{2}\mathbb{E}^\xi\left[\mathcal{L}_{\xi_1}^2 + \mathcal{L}_{\xi_2}^2 + 2\mathcal{L}_{\xi_1}\mathcal{L}_{\xi_2} - \frac{1}{4}\mathcal{L}^2\right]f(\gamma) + \mathcal{O}(h^3)\,.\end{aligned}$$

where $u(\gamma, h) = P(h)f(\gamma)$ *is the solution of the rescaled, by* $1/2$*, equation (2.5)*

$$\partial_t u(\zeta,t) = \frac{1}{2}\mathcal{L}u(\zeta,t)\,, \qquad u(\zeta,0) = f(\zeta)\,. \tag{2.42}$$

Proof. We estimate the local truncation error following similar steps as in the deterministic case. From the definition of the γ-process we have

$$\mathbb{E}^\gamma[f(\gamma_h)] = \mathbb{E}^\xi[P_{\xi_1}(h)P_{\xi_2}(h)f(\gamma)]\,,$$

and similarly, using the fact that the initial states are same, $\sigma_0 = \gamma_0 = \gamma$,

$$\mathbb{E}^\sigma[f(\sigma_h)] = P(h)f(\gamma) = u(\gamma,h)\,.$$

Hence we obtain a representation of the mean local error

$$\mathbb{E}^\gamma[f(\gamma_h)] - \mathbb{E}^\sigma[f(\sigma_h)] = \mathbb{E}^\xi\left[(P_{\xi_1}(h)P_{\xi_2}(h) - P(h))\,f(\gamma)\right]\,. \tag{2.43}$$

Now for given realizations of ξ_1, ξ_2 we have the expansion of $P_{\xi_1}(h)P_{\xi_2}(h) - P(h)$ as in the deterministic case, thus obtaining

$$\begin{aligned}&[P_{\xi_1}(h)P_{\xi_2}(h) - P(h)]f = \\ &h[\mathcal{L}_{\xi_1} + \mathcal{L}_{\xi_2} - \frac{1}{2}\mathcal{L}]f + \frac{h^2}{2}[\mathcal{L}_{\xi_1}^2 + \mathcal{L}_{\xi_2}^2 + 2\mathcal{L}_{\xi_1}\mathcal{L}_{\xi_2} - \frac{1}{4}\mathcal{L}^2]f + \mathcal{O}(h^3)\,.\end{aligned} \tag{2.44}$$

Note that $\frac{1}{2}\mathcal{L} = \frac{1}{2}\mathcal{L}_1 + \frac{1}{2}\mathcal{L}_2$ is associated with the process $\{\sigma_t\}_{t\geq 0}$. We have that the leading term of the local truncation error is $\mathbb{E}^\xi[\mathcal{L}_{\xi_1} + \mathcal{L}_{\xi_2} - \frac{1}{2}\mathcal{L}]$ and thus this term vanishes whenever $\frac{1}{2}\mathcal{L} = \mathbb{E}^\xi[\mathcal{L}_{\xi_1} + \mathcal{L}_{\xi_2}]$, which holds true when $P(\xi_k = 1) = P(\xi_k = 2) = \frac{1}{2}$. □

Remark 2.6.1. This calculation also shows that if we want to obtain the generator $\mathcal{L}$ instead of $\frac{1}{2}\mathcal{L}$ in Lemma 2.6.1, then in order to evolve the process σ by the time step h, each semigroup P_{ξ_1}, P_{ξ_2} needs to be applied with the time step $2h$, giving rise to the approximating process γ_h. In this case we have the local error representation

$$\begin{aligned}\mathbb{E}^\gamma[f(\gamma_h)] - \mathbb{E}^\sigma[f(\sigma_h)]: &= \mathbb{E}^\xi\left[(P_{\xi_1}(2h)P_{\xi_2}(2h)f(\gamma) - u(\gamma,h))\right] \\ &= \frac{h^2}{2}\mathbb{E}^\xi\left[4\mathcal{L}_{\xi_1}^2 + 4\mathcal{L}_{\xi_2}^2 + 8\mathcal{L}_{\xi_1}\mathcal{L}_{\xi_2} - \mathcal{L}^2\right]f(\gamma) + \mathcal{O}(h^3)\,,\end{aligned} \tag{2.45}$$

where $u(\gamma, h) = P(h)f(\gamma)$ is the solution of (2.5).

2.6.2 Comparison of deterministic and random schedules

The presented error analysis allows us to evaluate and compare deterministic (Lie and Strang) PCS introduced in [1], as well as randomized PCS such as the one in Lemma 2.42, introduced earlier in [25]. We compare the deterministic and randomized PCS from the point of view of processor communication and error analysis by specifying the same error tolerance TOL for all PCS which, by means of our error analysis, selects in each case a possibly different time window Δt. Larger time windows give rise to algorithms that have less processor communication for the same error tolerance. We start with the Lie and Strang schemes.

We fix the same error tolerance level TOL in the Lie and Strang global errors (2.17) and (2.19) respectively. We also fix the same time window $T = n_L \Delta t_L$ and $T = n_S \Delta_S$ where Δt_L and Δt_S are the respective time steps of the Lie and the Strang schemes that will ensure the same tolerance level TOL up to time T. Based on Theorems 2.4.1 and 2.5.1 we have that the leading errors are governed by the commutators

$$\text{TOL} \sim C_{\text{Lie}}(T)\Delta t_{\text{Lie}}\,, \quad C_{\text{Lie}}(T) = \max_{k=0,\ldots,n} \| \,[\mathcal{L}_1, \mathcal{L}_2]u(t_k)\,\|_\infty\,, \tag{2.46}$$

and

$$\begin{aligned} \text{TOL} &\sim C_{\text{Strang}}(T)\Delta t_{\text{Strang}}^2\,, \\ C_{\text{Strang}}(T) &= \max_{k=0,\ldots,n} \| \Big([\mathcal{L}_1,[\mathcal{L}_1,\mathcal{L}_2]] - 2[\mathcal{L}_2,[\mathcal{L}_2,\mathcal{L}_1]]\Big)u(t_k)\,\|_\infty\,, \end{aligned} \tag{2.47}$$

where $u = u(t)$ solves (2.5). Furthermore, due to (2.23) and (2.24) we have that

$$\text{TOL} \sim \mathcal{O}\Big(\frac{L^{d+1}}{q}\Big)\Delta t_{\text{Lie}}\,, \quad \text{TOL} \sim \mathcal{O}(\frac{L^{2d+1}}{q}\Big)\Delta t_{\text{Strang}}^2\,. \tag{2.48}$$

In the case of the randomized PCS the same reasoning as in Theorem 2.4.1 allows us to iterate the mean local error (2.45) to obtain

$$\begin{aligned} \text{TOL} &\sim C_{\text{Random}}(T)\Delta t_{\text{Random}}\,, \\ C_{\text{Random}}(T) &= \max_{k=0,\ldots,n} \mathbb{E}^\xi\left[4\mathcal{L}_{\xi_1}^2 + 4\mathcal{L}_{\xi_2}^2 + 8\mathcal{L}_{\xi_1}\mathcal{L}_{\xi_2} - \mathcal{L}^2\right]u(t_k)\,, \end{aligned} \tag{2.49}$$

where $u = u(t)$ solves (2.5). We now easily obtain that

$$\mathbb{E}^\xi\left[4\mathcal{L}_{\xi_1}^2 + 4\mathcal{L}_{\xi_2}^2 + 8\mathcal{L}_{\xi_1}\mathcal{L}_{\xi_2} - \mathcal{L}^2\right]u(t) = \left[4\mathcal{L}_1^2 + 4\mathcal{L}_2^2 + \mathcal{L}^2\right]u(t)\,.$$

Thus, due to the rigorous remainder bounds in Section 2.5 on the solution of (2.5) such as Lemma 2.5.1, we have that the term $\| \left[4\mathcal{L}_1^2 + 4\mathcal{L}_2^2 + \mathcal{L}^2\right] u \|_\infty$ is of order $\mathcal{O}(1)$ in the system size N, and we have

$$\text{TOL} \sim \mathcal{O}(1)\Delta t_{\text{Random}}\,. \tag{2.50}$$

In order to achieve the same error tolerance TOL, (2.48) and (2.50) imply the following relation between the respective time steps

$$\delta t_{\text{SSA}} \ll \Delta t_{\text{Random}} \sim \frac{L^{d+1}}{q}\Delta t_{\text{Lie}} < \Delta t_{\text{Lie}} \sim L^d \Delta t_{\text{Strang}}^2 < \Delta t_{\text{Strang}}\,. \tag{2.51}$$

Here q is the diameter of each of the cells C_k in Figure 2.1a, and $\delta t_{\text{SSA}} = \mathcal{O}(1/N)$ is the stochastic time step (the waiting time) of the SSA algorithm [6], which is exponentially distributed according to (2.2).

The relation (2.51) has several practical implications:

- The selection of the time window Δt in each PCS is intrinsically goal-oriented in the sense that it depends directly on the macroscopic observable $f(\sigma)$ through the commutator estimates of the solution to (2.5);
- The random and deterministic PCS studied here are rigorously partially asynchronous as their respective time windows are much larger than the SSA time step δt_{SSA} for a given error tolerance;
- The Lie scheme (2.11) is expected to parallelize better than the randomized PCS in [25] when $L^{d+1} \ll q$, since it allows a q-times larger time step Δt for the same accuracy. This outcome is also demonstrated in Figure 2.2.
- Finally, among the PCS we studied, the Strang PCS yields parallel schemes with the least processor communication, at least when $L \sim \mathcal{O}(1)$, due to its higher order accuracy and the commutator estimate (2.24).

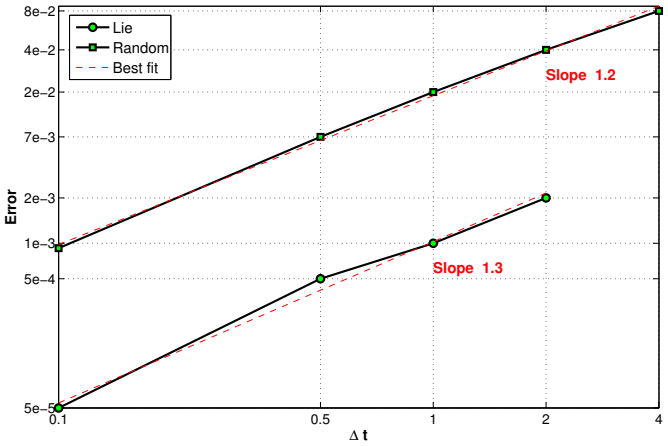


Figure 2.2: Convergence of the weak error for deterministic and randomized Lie splitting.

Example 2.6.1. We demonstrate this comparison in a computational example in which a jump process defined by Arrhenius spin-flip dynamics on a one-dimensional lattice was simulated. Simulations of two-dimensional Ising models were conducted in [1] including detailed benchmarks at long times as well as close to criticality. Numerous higher dimensional simulations are also provided by SPPARKS, [22].

The simulated system corresponds to the Ising model with nearest-neighbor interactions and spins taking values in $\{0, 1\}$. The rate of the process is give by

$$c(x, \sigma) = c_d(1 - \sigma(x)) + c_a \sigma(x) e^{-\beta U(x)} ,$$

where $U(x) = J(\sigma(x-1) + \sigma(x+1)) + \bar{h}$, and c_d, c_a, β, J, h are the parameters of the model.

We verified the theoretical order of convergence by computing the error

$$\int_0^T |\mathbb{E}[C(t)] - \mathbb{E}[\tilde{C}(t)]| \, dt$$

where $C(t)$ and $\tilde{C}(t)$ are the reference KMC and the FS-KMC solution, respectively, obtained by averaging the spatial mean coverage process $C(t) = \sum_{x \in \Lambda_N} \sigma_t(x)$ of the system over K independent realizations. For

the reference solution, the classical stochastic simulation algorithm (SSA) was used. In order to eliminate the impact of the statistical averaging error $K = 10^5$ independent samples were used. The error bars are below resolution of the graph depicted in Figure 2.2. In Figure 2.2 the error behavior is compared for different values of the splitting time step $h \equiv \Delta t$ for the randomized PCS and the Lie splitting. The lattice size is $N = 800$ and the parameters of the system are $\beta = 15$, $J = 0.37$, $h = 0.5$ and $c_a = c_d = 1$. For the fractional step algorithm four processors were used, thus the size of the sub-lattice is $q = 100$. The final time is chosen to be $T = 4$.

Example 2.6.2. In this example we investigate the dependence of the weak error, as defined in the previous example, on the sub-lattice parameter q. The model we used to run the simulation is Ising model, as described in Example 2.6.1. The parameters for the model are $\beta = 5$, $J = 1$, $h = 0.5$, and $c_a = c_d = 1$. The final time is chosen to be $T = 5$ and the dimension of the lattice $N = 480$. For the FS-KMC algorithm a constant, and rather large, time step parameter $\Delta t = 5$ was used. For the FS-KMC algorithm we used $K = 10^4$ samples to compute the mean value of the solution on the interval $[0, T]$ and for the reference solution, which was obtained with the SSA algorithm, $K = 10^5$ samples were used. In Figure 2.3 we can

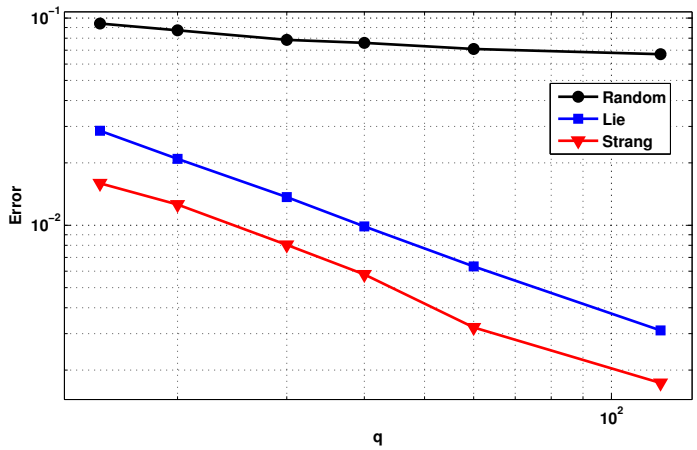


Figure 2.3: Dependence of the weak error on the sub-lattice size parameter q, see also (2.51).

observe that the deterministic schedules of Lie and Strang give better results than those of the random PCS. Also the Strang scheme has lower error than the Lie scheme as expected from the theoretical analysis. Finally, the dependence of the error on $\frac{1}{q}$ is also revealed, which in logarithmic scale is shown as a straight line.

2.7 Conclusions

In this paper we presented numerical analysis for the computational framework introduced in [1] as a tool for constructing partially asynchronous parallel KMC algorithms using the Fractional Step Kinetic Monte Carlo method. This class of parallel KMC algorithms, based on domain decomposition, approximates the underlying continuous time Markov chain. The presented numerical analysis demonstrates that for a particular set of observables the weak error (i.e., error in expected values) is controlled by the fractional time step Δt independently of the system size. The class of observables includes statistically relevant macroscopic observables for lattice systems of interacting particles, e.g., coverage, spatial correlations, surface roughness

etc. The conditions on such observables are explicit and can be checked a priori. The analysis also shows that the studied algorithms based either on deterministic (Lie, Strang) or random operator splitting are consistent and that the weak error for the identified class of observables converges to zero with a given rate as $\Delta t \to 0$.

The numerical analysis carried out here allows us, through the direct use of the convergence rate, to address systematically the processor communication of different parallelization strategies by demonstrating and comparing their (partial) asynchrony. Such comparison is made possible by relations such as (2.51) obtained when the approximation error is fixed within a prescribed tolerance. Furthermore, the presented results show that previously developed KMC algorithms based on domain decomposition principles, [25, 26, 21], as well as a well-established parallel KMC solver, [22], allow for simulations with controlled errors for a well-defined class of observables and their implementation is also partially asynchronous. Finally, moving beyond the parallelization problems discussed here, it appears that the proposed goal-oriented methodology, introduced in Section 2.5, can be generally applicable in the development and study of numerical approximations of molecular and other extended systems.

Bibliography

[1] G. Arampatzis, M. A. Katsoulakis, P. Plecháč, Michela Taufer, and Lifan Xu. Hierarchical fractional-step approximations and parallel kinetic monte carlo algorithms. *Journal of Computational Physics*, 231(23):7795–7814, 2012.

[2] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for Monte Carlo simulation of Ising spin systems. *J. Comp. Phys.*, 17(1):10–18, 1975.

[3] S. G. Eick, A. G. Greenberg, B. D. Lubachevsky, and A. Weiss. Synchronous relaxation for parallel simulations with applications to circuit-switched networks. *ACM Trans. Model. Comput. Simul.*, 3(4):287–314, 1993.

[4] S. Engblom. Parallel in time simulation of multiscale stochastic kinetics. *Multiscale Model. Simul.*, 8(1):46–68, 2009.

[5] C. Gardiner. *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*. Springer, 4th edition, 2009.

[6] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[7] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[8] P. Heidelberger and D. M. Nicol. Conservative parallel simulation of continuous time Markov chains using uniformization. *IEEE Trans. Parallel Distrib. Syst.*, 4:906–921, August 1993.

[9] T. Jahnke and D. Altıntan. Efficient simulation of discrete stochastic reaction systems with a splitting method. *BIT*, 50(4):797–822, 2010.

[10] M. A. Katsoulakis, P. Plecháč, and A. Sopasakis. Error analysis of coarse-graining for stochastic lattice dynamics. *SIAM J. Numer. Anal.*, 44(6):2270–2296, 2006.

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

[12] C. Kipnis and C. Landim. *Scaling Limits of Interacting Particle Systems*. Springer-Verlag, 1999.

[13] G. Korniss, M. A. Novotny, and P. A. Rikvold. Parallelization of a dynamic Monte Carlo algorithm: A partially rejection-free conservative approach. *J. Comp. Phys.*, 153(2):488–508, 1999.

[14] T. G. Kurtz. A random Trotter product formula. *Proc. Amer. Math. Soc.*, 35:147–154, 1972.

[15] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, Cambridge, 2000.

[16] Thomas M. Liggett. *Interacting Particle Systems*, volume 276 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1985.

[17] Da-Jiang Liu and J. W. Evans. Atomistic and multiscale modeling of CO-oxidation on Pd(100) and Rh(100): From nanoscale fluctuations to mesoscale reaction fronts. *Surf. Science*, 603:1706–1716, 2009.

[18] B. D. Lubachevsky. Efficient parallel simulations of dynamic Ising spin systems. *J. Comput. Phys.*, 75(1):103–122, 1988.

[19] Jonathan C. Mattingly, Andrew M. Stuart, and M. V. Tretyakov. Convergence of numerical time-averaging and stationary measures via Poisson equations. *SIAM J. Numer. Anal.*, 48(2):552–577, 2010.

[20] M. Merrick and K. A. Fichthorn. Synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Phys. Rev. E*, 75(1):011606, Jan 2007.

[21] G. Nandipati, Y. Shim, J. G. Amar, A. Karim, A. Kara, T. S. Rahman, and O. Trushin. Parallel kinetic Monte Carlo simulations of Ag(111) island coarsening using a large database. *Journal of Physics Condensed Matter*, 21:084214, 2009.

[22] S. Plimpton, C. Battaile, M. Chandross, L. Holm, A. Thompson, V. Tikare, G. Wagner, E. Webb, X. Zhou, C. Garcia Cardona, and A. Slepoy. Crossing the Mesoscale No-Man's Land via Parallel Kinetic Monte Carlo. Technical Report SAND2009-6226, Sandia National Laboratory, 2009.

[23] K Reuter, D Frenkel, and M Scheffler. The steady state of heterogeneous catalysis, studied by first-principles statistical mechanics. *Physical Review Letters*, 93(11), SEP 10 2004.

[24] Y. Shim and J. G. Amar. Rigorous synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Phys. Rev. B*, 71(11):115436, Mar 2005.

[25] Y. Shim and J. G. Amar. Semirigorous synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *Phys. Rev. B*, 71(12):125432, Mar 2005.

[26] Y. Shim and J. G. Amar. Hybrid asynchronous algorithm for parallel kinetic Monte Carlo simulations of thin film growth. *J. Comput. Phys.*, 212(1):305–317, 2006.

[27] H. F. Trotter. On the product of semi-groups of operators. *Proc. Amer. Math. Soc.*, 10:545–551, 1959.

Appendix 2.A Proof of Lemma 2.4.2

Proof. In order to simplify the notation, we introduce

$$a_{k,N}(h) = \begin{cases} \frac{h^k}{k!}\mathcal{L}_1^k & \text{if } k < N\,, \\ \mathcal{D}_k(h\mathcal{L}_1) & \text{if } k = N > 0\,, \\ e^{h\mathcal{L}_1} & \text{if } k = N = 0\,, \end{cases} \quad b_{k,N}(h) = \begin{cases} \frac{h^k}{k!}\mathcal{L}_2^k & \text{if } k < N\,, \\ \mathcal{D}_k(h\mathcal{L}_2) & \text{if } k = N > 0\,, \\ e^{h\mathcal{L}_2} & \text{if } k = N = 0\,. \end{cases} \tag{2.52}$$

Now the semigroup for the Lie splitting, at $t = h$, can be written as

$$\begin{aligned} e^{h\mathcal{L}_1}e^{h\mathcal{L}_2}f &= \sum_{i+j\le 3} a_{i,3-j}(h)b_{j,3}(h)f \\ &= \Big(I + h(\mathcal{L}_1+\mathcal{L}_2) + \frac{h^2}{2}(\mathcal{L}_1+\mathcal{L}_2)^2\Big)f \\ &+ h^2[\mathcal{L}_1,\mathcal{L}_2]f + \sum_{i+j=3} a_{i,3-j}(h)b_{j,3}(h)f\,. \end{aligned}$$

Comparing with

$$e^{h(\mathcal{L}_1+\mathcal{L}_2)}f = \Big(I + h(\mathcal{L}_1+\mathcal{L}_2) + \frac{h^2}{2}(\mathcal{L}_1+\mathcal{L}_2)^2\Big)f + \mathcal{D}_3(h(\mathcal{L}_1+\mathcal{L}_2))f\,,$$

we get the estimate for the local error

$$\begin{aligned} \|\, e^{h\mathcal{L}_1}e^{h\mathcal{L}_2}f - e^{h(\mathcal{L}_1+\mathcal{L}_2)}f \,\|_\infty &\le h^2 \|\, [\mathcal{L}_1,\mathcal{L}_2]f \,\|_\infty \\ &+ \|\, \mathcal{D}_3(h(\mathcal{L}_1+\mathcal{L}_2))f \,\|_\infty + \|\, \sum_{i+j=3} a_{i,3-j}(h)b_{j,3}(h)f \,\|_\infty\,. \end{aligned}$$

The second term in the above inequality is bounded by Lemma 2.4.1 and the third term is bounded by

$$\|\, \sum_{i+j=3} a_{i,3-j}(h)b_{j,3}(h)f \,\|_\infty \le ch^3\Big(\|\, \mathcal{L}_1^3 f \,\|_\infty + \|\, \mathcal{L}_1^2\mathcal{L}_2 f \,\|_\infty + \|\, \mathcal{L}_1\mathcal{L}_2^2 f \,\|_\infty + \|\, \mathcal{L}_2^3 f \,\|_\infty\Big)\,,$$

which follows from the definitions of a_k and b_k. The last step completes the proof for the local error in the Lie case. For the Strang scheme the proof follows the same idea, we only have to take one more term in the expansion,

$$\begin{aligned} e^{\frac{h}{2}\mathcal{L}_1}e^{h\mathcal{L}_2}e^{\frac{h}{2}\mathcal{L}_1}f &= \sum_{i+j+k\le 4} a_{k,4-i-j}(\frac{h}{2})b_{j,4-i}(h)a_{i,4}(\frac{h}{2})f \\ &= \Big(I + h(\mathcal{L}_1+\mathcal{L}_2) + \frac{h^2}{2}(\mathcal{L}_1+\mathcal{L}_2)^2 + \frac{h^3}{6}(\mathcal{L}_1+\mathcal{L}_2)^3\Big)f \\ &+ \frac{h^3}{24}\Big([\mathcal{L}_1,[\mathcal{L}_1,\mathcal{L}_2]] - 2[\mathcal{L}_2,[\mathcal{L}_2,\mathcal{L}_1]]\Big)f \\ &+ \sum_{i+j+k=4} a_{k,4-i-j}(\frac{h}{2})b_{j,4-i}(h)a_{i,4}(\frac{h}{2})f\,. \end{aligned}$$

Comparing with

$$\begin{aligned} e^{h(\mathcal{L}_1+\mathcal{L}_2)}f &= \Big(I + h(\mathcal{L}_1+\mathcal{L}_2) + \frac{h^2}{2}(\mathcal{L}_1+\mathcal{L}_2)^2 \\ &+ \frac{h^3}{6}(\mathcal{L}_1+\mathcal{L}_2)^3\Big)f + \mathcal{D}_4(h(\mathcal{L}_1+\mathcal{L}_2))f\,, \end{aligned}$$

the estimate for the local error follows

$$\| e^{\frac{h}{2}\mathcal{L}_1} e^{h\mathcal{L}_2} e^{\frac{h}{2}\mathcal{L}_1} f - e^{h(\mathcal{L}_1+\mathcal{L}_2)} f \|_\infty \leq ch^3 \| [\mathcal{L}_1,[\mathcal{L}_1,\mathcal{L}_2]]f - 2[\mathcal{L}_2,[\mathcal{L}_2,\mathcal{L}_1]]f \|_\infty$$
$$+ \| \sum_{i+j+k=4} a_{k,4-i-j}(\frac{h}{2}) b_{j,4-i}(h) a_{i,4}(\frac{h}{2}) f \|_\infty + \| \mathcal{D}_4(h(\mathcal{L}_1+\mathcal{L}_2)) f \|_\infty .$$

The second term is bounded by Lemma 2.4.1 and the third term is bounded by

$$\| \sum_{i+j+k=4} a_{k,4-i-j}(\frac{h}{2}) b_{j,4-i}(h) a_{i,4}(\frac{h}{2}) f \|_\infty \leq c_4 h^4 \sum_{|m|=4} \| \mathcal{L}_1^{m_1} \mathcal{L}_2^{m_2} \mathcal{L}_1^{m_3} f \|_\infty ,$$

which again follows from (2.52). □

Appendix 2.B A general form of Gronwall's inequality

For the sake of completeness we prove a variant of Gronwall's lemma for a particular case that appears in the proof of Proposition 2.5.5. We prove it in the presence of two equations, but the result can be easily generalized for a system of equations.

Lemma 2.B.1 (Gronwall's inequality)**.** *Let ϑ and φ satisfy the following inequalities*

$$\begin{aligned} \varphi(t) &\leq \varphi(0) + \int_0^t \varphi(s)\, ds \\ \vartheta(t) &\leq \vartheta(0) + \int_0^t \vartheta(s)\, ds + \int_0^t \varphi(s)\, ds \end{aligned}$$

then

$$\varphi(t) \leq e^t \varphi(0) \tag{2.53}$$
$$\vartheta(t) \leq e^t \vartheta(0) + (e^t + te^t - 1)\varphi(0) \tag{2.54}$$

Proof. The first estimate follows directly from Gronwall's inequality. By integrating this inequality on $[0,t]$

$$\int_0^t \varphi(s) \leq (e^t - 1)\varphi(0) ,$$

and by substituting this to the second inequality we obtain

$$\vartheta(t) \leq \vartheta(0) + (e^t - 1)\varphi(0) + \int_0^t \vartheta(s)\, ds .$$

If we multiply by e^{-t} and integrate on $[0,t]$ we have

$$\int_0^t \Big[e^{-r} \int_0^r \vartheta(r) \Big]' dr \leq (1 - e^{-t})\vartheta(0) + (e^t + te^t - 1)\varphi(0) ,$$

and after straightforward calculations

$$\vartheta(t) \leq e^t \vartheta(0) + (e^t + te^t - 1)\varphi(0) .$$

□

Remark 2.B.1. Let $\Phi(t) = (\varphi_1(t), \ldots, \varphi_n(t))$ satisfying

$$\Phi(t) \leq \Phi(0) + \int_0^t A\Phi(s)\, ds$$

where A is a constant lower triangular matrix and the inequality has the meaning that it is true componentwise, then

$$\Phi(t) \leq B(t)\Phi(0)\,,$$

where B is a lower triangular matrix with elements exponentially depending on t.

Chapter 3

Goal-oriented sensitivity analysis for lattice kinetic Monte Carlo simulations

In this paper we propose a new class of coupling methods for the sensitivity analysis of high dimensional stochastic systems and in particular for lattice Kinetic Monte Carlo. Sensitivity analysis for stochastic systems is typically based on approximating continuous derivatives with respect to model parameters by the mean value of samples from a finite difference scheme. Instead of using independent samples the proposed algorithm reduces the variance of the estimator by developing a strongly correlated-"coupled"- stochastic process for both the perturbed and unperturbed stochastic processes, defined in a common state space. The novelty of our construction is that the new coupled process depends on the targeted observables, e.g. coverage, Hamiltonian, spatial correlations, surface roughness, etc., hence we refer to the proposed method as *goal-oriented* sensitivity analysis. In particular, the rates of the coupled Continuous Time Markov Chain are obtained as solutions to a goal-oriented optimization problem, depending on the observable of interest, by considering the minimization functional of the corresponding variance. We show that this functional can be used as a diagnostic tool for the design and evaluation of different classes of couplings. Furthermore, the resulting KMC sensitivity algorithm has an easy implementation that is based on the Bortz–Kalos–Lebowitz algorithm's philosophy, where here events are divided in classes depending on level sets of the observable of interest. Finally, we demonstrate in several examples including adsorption, desorption and diffusion Kinetic Monte Carlo that for the same confidence interval and observable, the proposed goal-oriented algorithm can be two orders of magnitude faster than existing coupling algorithms for spatial KMC such as the Common Random Number approach.

3.1 Introduction

Recently there has been significant progress in developing sensitivity analysis tools for stochastic processes modeling well-mixed chemical reactions and biological networks. Some of the mathematical tools include log-likelihood methods and Girsanov transformations [15, 28, 31], polynomial chaos [20], finite difference methods and their variants [32, 2] and pathwise sensitivity methods [35]; a somewhat parallel literature, facing related challenges, exists also in mathematical finance [8] and operations research [6, 12, 5]. However,

existing sensitivity analysis approaches can have an overwhelming computational cost in high dimensions, such as lattice Kinetic Monte Carlo (KMC), either due to high variance in the gradient estimators, or in models with a high-dimensional parameter space. Such issues and comparisons between methods are discussed, for instance, in recent literature [26, 36, 30], see also the demonstration in Figure 3.3.

Estimating the sensitivity of a stochastic process $\{\sigma_t = \sigma_t(\theta),\, t \geq 0\}$ with respect to perturbations in the model parameters $\theta = (\theta_1, ..., \theta_k)$ can be viewed either at the level of the process' probability distribution function (PDF) or as a response of specific averaged observables $f = f(\sigma_t)$ in some time interval $t \in [0, T]$, namely the quantity

$$u(t, \rho; \theta) = \mathbb{E}_\rho[f(\sigma_t(\theta))]\,,$$

to parameter perturbations, where ρ denotes the initial configuration of the stochastic process σ_t and $\mathbb{E}_\rho$ the corresponding expected value. We focus on the latter perspective, which we also refer to as a *goal-oriented* approach, since the focus is on observables of interest. In this case, we quantify the sensitivity of specific observables by estimating gradients of the type $\partial_{\theta_l} u(t, \rho; \theta)$, where $l \in \{1, ..., k\}$ and $\theta = (\theta_1, ..., \theta_k)$.

In turn, it is commonplace to evaluate such gradients for parametric sensitivity of the observable $u(t, \rho; \theta)$ by using finite difference (FD) approximations, e.g., for first order derivatives

$$\partial_{\theta_l} u(t, \rho; \theta) \approx \Delta := \frac{1}{h}(u(t, \rho; \theta + he_l) - u(t, \rho; \theta))\,, \tag{3.1}$$

where $h \in \mathbb{R}$ is a small parameter and e_l a unit vector in $\mathbb{R}^k$ with $e_{l,j} = 1$ if $l = j$ and zero otherwise. While such simulation approach appears straightforward, it suffers from a well-known problem arising from the high variance of the estimator for the finite-difference Δ [32], at least if we naively pick independent samples as we further explain in (3.23) below.

Overall, developing methods of reduced variance is a *critical computational task* for carrying out sensitivity analysis in high-dimensional complex stochastic models such as the spatial Kinetic Monte Carlo algorithms studied here. The high computational cost of individual realizations of the stochastic process renders prohibitive the generation of a large number of samples for reliable ensemble averaging. Hence, reducing the variance by orders of magnitude, e.g. Figure 3.3, will result in an analogous reduction in the number of samples needed for the ensemble average.

The variance of finite difference sensitivity estimators is usually reduced by employing *coupling methods*, i.e., by constructing highly correlated paths for the processes $\sigma_t(\theta)$ and $\sigma_t(\theta + he_l)$. The simplest coupling is to run the two processes using the same stream of random number, known as Common Random Number (CRN) method, which for spatially extended systems the induced correlation is not enough to reduce the variance. Improvements of this method is the Common Random Path algorithm [32] (CRP) and the Coupled Finite Difference (CFD) method [2]. In fact, it is shown [36] that among these coupling methods CFD performs better, at least for relatively simple, low-dimensional reaction networks. In general, such couplings are suitable only for low-dimensional, well-mixed systems as variance estimates depend on system size [2] and more importantly it was not clear, up to now, how to extend them in an efficient manner to spatially distributed models Kinetic Monte Carlo (KMC) models. As we see in Figure 3.3 depicting a lattice KMC simulation of a spatially extended adsorption/desorption model, the variance of such coupled estimators remains very high even if CRN is implemented.

The novelty of the approach we propose in this direction relies on developing a different concept of stochastic coupling which (a) is suitable for spatially extended systems such as lattice KMC, and (b) is designed for specific observables $\mathbb{E}_\rho[f(\sigma_t)]$ hence we call it a goal-oriented coupling method. Our proposed method relies on defining a new coupled continuous time Markov Chain through a suitable generator that acts on observables of the involved high-dimensional stochastic processes associated with parameters θ and

$\theta + \epsilon$, where ϵ denotes any k-dimensional perturbation in parameter space, namely

$$\epsilon = he_l, \quad l = 1, \ldots, k\,.$$

Indeed, we define the coupling $\zeta_t = (\sigma_t, \eta_t)$ of two stochastic processes $\sigma_t = \sigma_t(\theta)$ and $\eta_t = \sigma_t(\theta + \epsilon)$, i.e. we couple the dynamics, setting them in a common, product probability space, but at the same time we respect the marginal distributions of each one of them. Clearly we have freedom on how to select these coupling rates, however our goal is to minimize the variance of estimators of the finite difference of specific observables f, while keeping the computational cost of coupling low. The rates of the coupled stochastic processes $\zeta_t = (\sigma_t, \eta_t)$ are obtained by solving an optimization problem associated with minimizing the variance between the coupled stochastic dynamics. Furthermore, the optimization functional is also a measure of the "tightness" of the coupling, allowing us to compare various coupling schemes and systematically assess their effectiveness in variance reduction. The algorithmic implementation of the proposed coupled method is a Bortz-Kalos-Lebowitz (BKL) – type algorithm in the sense that events are categorized into pre-defined sets. In the classical BKL algorithm [4] events are divided into classes of equal rates, i.e. according to level sets of the rates. However, here the events are divided into classes depending on the observable's level sets, since we are interested in tight couplings of time series of specific observables. Numerical examples of spatial KMC, e.g. adsorption/desorption/diffusion processes, are presented throughout the paper and demonstrate that the variance can be improved by two orders of magnitude compared to coupling methods used up to now, such as the CRN method. At the same time the computational overhead of computing the coupled rates is two times slower than that of the CRN method leading to an overall speed up factor of two orders of magnitude. Furthermore, numerical experiments also demonstrate that the variance-related optimization functional indeed constitutes a diagnostic tool for the design and evaluation of different couplings.

The paper is structured as follows: in Section 2 we provide background and notation for spatial KMC methods and discuss earlier work on coupling methods. In Section 3 we introduce coupling methods for spatial KMC algorithms and demonstrate the resulting variance reduction in several examples. In Section 4 we introduce improved coupling algorithms which do not attempt to couple the time series configurations of the entire stochastic process, but instead they are designed to couple only time series of specific observables; the resulting algorithms are constructed in the spirit of the BKL algorithms for KMC and we demonstrate variance reduction up to two orders of magnitude. In Section 5, we discuss the limitations, as well as the potential applicability of the proposed coupling to systems with a very high-dimensional parameter space. Finally, in Appendix we give detailed examples of complex reaction-diffusion models and a description of the implementation of the proposed sensitivity analysis method.

3.2 Background

3.2.1 Markov Chains and kinetic Monte Carlo

In this work we present the proposed sensitivity analysis methods in the context of spatial Kinetic Monte Carlo (KMC) methods, although similar challenges and ideas are relevant to all other molecular simulation methods. The resulting stochastic processes studied in this work are set on a discrete, albeit high-dimensional, configuration space $\mathcal{S}$ and necessarily have to be of jump type describing transitions between different configurations $\sigma \in \mathcal{S}$. Mathematically, such a Continuous Time Markov Chain (CTMC) is a stochastic process $\{\sigma_t\}_{t\geq 0}$ defined completely in terms of the local transition rates $c(\sigma, \sigma')$ which determine the updates (jumps) from any current state $\sigma_t = \sigma$ to a (random) new state σ'. Usually, simulated paths of the process

are constructed via Kinetic Monte Carlo (KMC), that is through the procedure described in (3.2) and (3.3) below.

Realizations of the process are constructed from the embedded discrete time Markov chain [21] $S_n = \sigma_{t_n}$ with jump times t_n from the exponential distribution: In the context of the spatially distributed problems (in which we are interested here), the local transition rates will be denoted as $c(\sigma, \sigma'; \theta)$ where $\theta \in \mathbb{R}^k$ is a vector of the model parameters, describing transitions from the configuration at time t, $\sigma_t = \sigma$ into a new configuration σ' . The local transition rates c, define the total rate

$$\lambda(\sigma; \theta) = \sum_{\sigma'} c(\sigma, \sigma'; \theta)\,, \tag{3.2}$$

which is the intensity of the exponential waiting time for a jump to be performed when the system is currently at the state σ. The transition probabilities for the embedded Markov chain $\{S_n\}_{n\geq 0}$ are

$$p(\sigma, \sigma'; \theta) = \frac{c(\sigma, \sigma'; \theta)}{\lambda(\sigma; \theta)}\,. \tag{3.3}$$

In other words once the exponential "clock" signals a jump, the system transitions from the state σ to a new configuration σ' with probability $p(\sigma, \sigma'; \theta)$. On the other hand, the evolution of the entire system at any time t is described by the transition probabilities $P(\sigma, t; \sigma_0; \theta) := \mathbb{P}\left(\sigma_t = \sigma \,|\, \sigma_0 = \sigma_0\right)$ where $\sigma_0 \in \mathcal{S}$ is any initial configuration. The transition probabilities, corresponding to the local rates c, satisfy the Forward Kolmogorov Equation [13] (Master Equation) ,

$$\partial_t P(\sigma, t; \sigma_0; \theta) := \sum_{\sigma' \neq \sigma} c(\sigma', \sigma; \theta) P(\sigma', t; \sigma_0; \theta) - c(\sigma, \sigma'; \theta) P(\sigma, t; \sigma_0; \theta)\,, \tag{3.4}$$

where $P(\sigma, 0; \sigma_0; \theta) = \delta(\sigma - \sigma_0)$ and $\delta(\sigma - \sigma_0) = 1$ if $\sigma = \sigma_0$ and zero otherwise.

Generators for CTMC. Typically in KMC we need to compute expected values of such observables, that is quantities defined as

$$u(\sigma_0, t) := \mathbb{E}_{\sigma_0}[f(\sigma_t)] = \sum_{\sigma'} f(\sigma') P(\sigma', t; \sigma_0)\ . \tag{3.5}$$

Here $\mathbb{E}_{\sigma_0}$ denotes the expected value with respect to the law of the process $\{\sigma_t\}$ conditioned on the initial configuration σ_0. By a straightforward calculation [13] using (3.4) we obtain that the observable (3.5) satisfies the initial value problem

$$\partial_t u(\sigma_0, t) = \mathcal{L} u(\sigma_0, t)\,, \qquad u(\sigma_0, 0) = f(\sigma_0)\,, \tag{3.6}$$

where the operator $\mathcal{L}$ is known as the *generator* of the CTMC [23]

$$\mathcal{L} f(\sigma) = \sum_{\sigma'} c(\sigma, \sigma')[f(\sigma') - f(\sigma)]\,. \tag{3.7}$$

The generator fully determines the process σ_t while (3.7) can viewed as the dual [23] of (3.4). Although in order to describe KMC algorithms (and in general any discrete space continuous time Markov process) is not necessary to use generators, we have found that here they allow us to systematically construct and assess couplings of stochastic processes for the purpose of providing low variance finite difference estimators. For this reason we introduce the concept, hopefully in a self contained manner, and exploit it in Sections 3.3.2 and 3.4.2. Using generator notation [23] we then can rewrite (3.5), as the the action of the Markov semi-group propagator $e^{t\mathcal{L}}$ associated with the generator $\mathcal{L}$ and the process $\{\sigma_t\}_{t\geq 0}$ acting on the observable f:

$$u(\sigma_0, t) = \mathbb{E}_{\sigma_0}[f(\sigma_t)] := e^{t\mathcal{L}} f(\sigma_0)\ . \tag{3.8}$$

In fact, in the case where the state space is finite but high dimensional, as in spatial KMC processes considered here, the operator $\mathcal{L}$ is essentially a matrix (bounded), that is

$$\| \mathcal{L}f \|_\infty \leq c \| f \|_\infty \tag{3.9}$$

where $\| f \|_\infty = \max_{\sigma \in \mathcal{S}} |f(\sigma)|$ and c is a constant independent of f. Due to the boundness of $\mathcal{L}$, $e^{t\mathcal{L}}$ can be also defined as an infinite series. For $t = \delta t \ll 1$ the semigroup (3.8) can be approximated by series truncation as

$$e^{\delta t \mathcal{L}} = I + \delta t \mathcal{L} + \mathcal{O}(\delta t^2)\,, \tag{3.10}$$

allowing us to write the solution [21] of (3.6) in terms of the operator $\mathcal{L}$ and δt,

$$u(\delta t; \sigma) = u(0; \sigma) + \delta t \mathcal{L} u(0; \sigma) + \mathcal{O}(\delta t^2)\,. \tag{3.11}$$

Rigorous statements with less stringent conditions on the types of such processes and corresponding generators can be found in literature [23].

3.2.2 KMC on a Lattice and Benchmark Examples

We consider interacting particle system defined on a d-dimensional lattice Λ_N of any type (square, hexagonal, Bravais, etc), where N is the size of the lattice. As a result, we model the dynamics of the configuration space with a Continuous Time Markov Chain jump process. We restrict our discussion to lattice gas models where the order parameter or the spin variable takes values in a finite set $\Sigma = \{0, 1, \ldots, K\}$. At each lattice site $x \in \Lambda_N$ an order parameter (or a spin variable in Ising systems) $\sigma(x) \in \Sigma$ is defined. The elements in Σ correspond to occupation of the site $x \in \Lambda_N$ by different species. The stochastic process $\{\sigma_t\}_{t \geq 0}$ takes values in the configuration space

$$\mathcal{S} = \Big\{ \big(\sigma(x_1), \ldots, \sigma(x_N)\big) \mid x_i \in \Sigma,\, i = 1, \ldots, N \Big\}\,.$$

Microscopic dynamics are described by changes (transitions) of spin variables at different sites . We study systems in which the transitions are localized and involve only a finite number of sites at each transition step. First, the *local* dynamics are described by an updating mechanism and corresponding transition rates $c(x, \sigma)$, such that the configuration at time t, $\sigma_t = \sigma$ changes into a new configuration $\sigma' = \sigma^x$ by an update of σ at the site $x \in \Lambda_N$. For example, if $\Sigma = \{0, 1\}$ the order parameter models the classical lattice gas with a single species occupying the site x when $\sigma(x) = 1$ and with the site being vacant if $\sigma(x) = 0$. The system can jump from state σ to the new configuration σ^x (adsorption $0 \to 1$, desorption $1 \to 0$), where[19, 23]

$$\sigma^x(y) = \begin{cases} 1 - \sigma(x), & \text{if } x = y \\ \sigma(y), & \text{if } x \neq y\,, \end{cases} \tag{3.12}$$

and the generator of the process is,

$$\mathcal{L}f(\sigma) = \sum_{x \in \Lambda_N} c(x, \sigma) \big[f(\sigma^x) - f(\sigma) \big]\,. \tag{3.13}$$

The Ising model, with adsorption/desorption dynamics and spins in $\{0, 1\}$ has rates,

$$c_I(x, \sigma) = \begin{cases} c_a, & \text{if } \sigma(x) = 0 \\ c_d e^{-\beta \left(J(\sigma(x-1) + \sigma(x+1)) - h \right)}, & \text{if } \sigma(x) = 1 \end{cases} \tag{3.14}$$

where β, J and h are the inverse temperature, inter-particle potential and external field respectively and c_a, c_d are the adsorption and desorption constants respectively. Using the same setup as in the Ising model, the simple diffusion process can be described by the transition from the configuration σ the new configuration state $\sigma' = \sigma^{x,y}$, where the particle exchanges position with an empty site,

$$\sigma^{x,y}(z) = \begin{cases} \sigma(x), & \text{if } z = y \\ \sigma(y), & \text{if } z = x \\ \sigma(z), & \text{if } z \neq x, y\,, \end{cases} \tag{3.15}$$

with rate,

$$c_D(x, y, \sigma) = \begin{cases} c_{diff}, & \text{if } |x - y| = 1 \\ 0, & \text{if } |x - y| \neq 1\,. \end{cases} \tag{3.16}$$

Modelling interacting diffusions is also straightforward [7]. Combining simple mechanisms one can build more complex models. For example the generator that describes an adsorption/desorption type model with diffusion is given by the sum of the generator of the two basic processes,

$$\mathcal{L}f(\sigma) = \sum_{x \in \Lambda_N} c_I(x, \sigma)\big[f(\sigma^x) - f(\sigma)\big] + \sum_{x \in \Lambda_N} \sum_{y \in \Lambda_N} c_D(x, y, \sigma)\big[f(\sigma^{x,y}) - f(\sigma)\big]\,.3 \tag{3.17}$$

A general formulation for the description of more complex lattice systems and mechanisms that include all of the above is presented in Section 3.4.1, see also Appendix 3.A. These models describe a wide range of applications ranging from crystal growth, to catalysis, to biology [7]. For instance, in catalysis, KMC describes microscopic events, such as adsorption of species on a catalyst and its reverse (desorption to the fluid), surface diffusion, and surface reactions; typically these mechanisms occur concurrently. In such KMC models the transition rate constants were determined in an ad hoc or a semi-empirical manner. However, more recently *first-principles* KMC methods were developed, where kinetic parameters of micro mechanisms are estimated by *ab initio* density functional theory [22] (DFT) see for instance the recent works [16, 33, 25, 40]. Contrary to earlier work, more qualitative descriptions, such first-principles models yield a remarkable agreement with experiments [37, 27, 10]. In view of the significant role of DFT based parameter fitting the role of sensitivity analysis is a crucial step in building reliable predictive KMC algorithms.

3.2.3 Finite Difference sensitivity and Coupling Methods

We first consider the following family or generators parametrized by the parameter vector $\theta \in \mathbb{R}^k$,

$$\mathcal{L}^\theta f(\sigma) = \sum_{x \in \Lambda_N} c(x, \sigma; \theta)\big[f(\sigma^x) - f(\sigma)\big]\,, \tag{3.18}$$

while we address general lattice KMC in section 3.4.1. Our goal is to assess the sensitivity of u^θ, i.e. the solution of (3.6) with generator $\mathcal{L}^\theta$, in perturbations in the parameter θ of the rates function c. Therefore we consider the quantity,

$$D_\epsilon(t; \sigma, \eta) = u^\theta(t, \sigma) - u^{\theta+\epsilon}(t, \eta)\,. \tag{3.19}$$

From now on, in order to keep the generality and lighten the notation, we define $u_A := u^\theta$ and $u_B := u^{\theta+\epsilon}$. The subscript A and B will also be used in rates and generators to correspond to θ and $\theta + \epsilon$. In order to estimate the quantity (3.19), one must simulate many realizations of the stochastic model, $\sigma_t^{[i]}$, and then take ensemble averages to compute u_A and u_B. This can be written as

$$D_\epsilon(t; \sigma, \eta) \approx D_{N,\epsilon}(t; \sigma, \eta) = \sum_{i=1}^{N} \frac{f(\sigma_t^{[i]}) - f(\eta_t^{[i]})}{N}, \tag{3.20}$$

where $\sigma_t^{[i]}$ and $\eta_t^{[i]}$ are the $i-$th path at time t, generated from $\mathcal{L}_A$ and $\mathcal{L}_B$ respectively.

Next we focus on calculating the variance of estimator (3.20). First, we write D_ϵ in terms of the probability distributions as,

$$\begin{aligned} D_\epsilon(t;\sigma,\eta) &= u^A(t,\sigma_0) - u^B(t,\eta_0) \\ &= \mathbb{E}_{\sigma_0} f(\sigma_t) - \mathbb{E}_{\eta_0} f(\eta_t) \\ &= \sum_{\sigma'} f(\sigma') p^A(\sigma', t; \sigma_0) - \sum_{\eta'} f(\eta') p^B(\eta', t; \eta_0) \\ &= \sum_{\sigma',\eta'} \Big[f(\sigma') - f(\eta') \Big] \bar{p}(\sigma', \eta', t; \sigma_0, \eta_0)\,, \end{aligned}$$

where $p^A(\cdot, t; \sigma_0)$ and $p^B(\cdot, t; \eta_0)$ are the transition probability measures at time t given that the initial state is σ_0 and η_0, according to the dynamics of the rate functions c^A and c^B respectively. Here the summation is considered over the entire configuration space, while $\bar{p}$ denotes the joint probability of the processes $\{\sigma_t\}_{t\geq 0}$ and $\{\eta_t\}_{t\geq 0}$ defined as

$$\bar{p}(\sigma, \eta, t; \sigma_0, \eta_0) = \mathbb{P}(\sigma_t = \sigma, \eta_t = \eta | \sigma_0 = \sigma_0, \eta_0 = \eta_0)\,. \tag{3.21}$$

Thus, the variance of estimator (3.20) is,

$$\begin{aligned} 0 \leq \mathrm{Var}_{(\sigma_0,\eta_0)}\big[D_{N,\epsilon}(t;\sigma,\eta)\big] &= \mathrm{Var}_{(\sigma_0,\eta_0)}\big[f(\sigma_t) - f(\eta_t)\big] \\ = \sum_{\sigma',\eta'} \Big[\big(f(\sigma') - \mathbb{E}_{\sigma_0} f(\sigma_t)\big) - \big(f(\eta') - \mathbb{E}_{\eta_0} f(\eta_t)\big) \Big]^2 \bar{p}(\sigma', \eta', t; \sigma_0, \eta_0) \\ = \sum_{\sigma'} \Big[f(\sigma') - \mathbb{E}_{\sigma_0} f(\sigma_t) \Big]^2 p^A(\sigma', t; \sigma_0) + \sum_{\eta'} \Big[f(\eta') - \mathbb{E}_{\eta_0} f(\eta_t)) \Big]^2 p^B(\eta', t; \eta_0) - \\ - 2 \sum_{\sigma',\eta'} \big(f(\sigma') - \mathbb{E}_{\sigma_0} f(\sigma_t)\big)\big(f(\eta') - \mathbb{E}_{\eta_0} f(\eta_t)\big) \bar{p}(\sigma', \eta', t; \sigma_0, \eta_0)\,. \end{aligned} \tag{3.22}$$

If these processes are independent then

$$\bar{p}(\sigma', \eta', t; \sigma_0, \eta_0) = p^A(\sigma', t; \sigma_0) p^B(\eta', t; \eta_0) \tag{3.23}$$

and the last term in equation (3.22) is 0. Hence a strong coupling of the processes, i.e. when the processes σ and η are correlated, is expected to reduce the variance of the estimator (3.20). Therefore, based on (3.22), our goal is to maximize

$$\sum_{\sigma',\eta'} \big(f(\sigma') - \mathbb{E}_{\sigma_0} f(\sigma_t)\big)\big(f(\eta') - \mathbb{E}_{\eta_0} f(\eta_t)\big) \bar{p}(\sigma', \eta', t; \sigma_0, \eta_0) \tag{3.24}$$

although note that inequality (3.22) implies that any coupling has a theoretical upper bound.

As shown above, highly correlated processes will reduce the variance of estimator (3.20). Earlier work on coupling focuses on reactions networks which model well mixed systems, i.e. when the system is spatially homogeneous. The simplest coupling is to run the two processes using the same stream of random numbers, known as Common Random Number (CRN). One improvement of this method is the Common Random Path algorithm [35] (CRP). This method uses separate streams of random numbers for every reaction, introducing non zero correlation between the two processes thus leading to more coupled processes. In systems where the dynamics are high dimensional this coupling is intractable: the number of reactions is proportional to the lattice size which means that in order to implement the CRP method one has to store as many different random number streams or keep as many different seeds and bring the random number generator to the

previous state for every reaction. The former needs extremely large amount of memory, while the later is using up too much computational time.

A different approach was introduced recently where the two processes are coupled using a generator acting on both processes [2]. The same idea was used by Liggett [23], in a different context, to obtain theoretical results for the monotonicity and ergodicity of interacting particle systems by Markov Chains. Our work uses these ideas as a starting point and extends them to spatial systems with complex dynamics and a high dimensional state space. More importantly our approach incorporates properties of macroscopic observables leading to a much more efficient goal-oriented method. Finally we also refer to Lindvall [24] for the mathematical framework of coupling for random variables and general stochastic processes.

3.3 Formulation of Coupling and Variance Reduction for Spatially Extended Systems

In this Section we first present couplings of stochastic processes based on correlating the time series of the KMC evolutions defined on the entire configuration space. Subsequently, in Section 4 we present a class of more efficient couplings for variance reduction, which are nevertheless constructed building on the couplings of Section 3, and are based on correlating just the time series of the targeted observable.

In the same spirit of earlier coupling methods, in order to reduce the variance of the estimator (3.20), one can couple the two generators and produce highly correlated paths. The novelty of our approach is that we introduce couplings that are efficient for very high-dimensional state spaces arising in spatially extended systems such as reaction-diffusion KMC. We will present a general and systematic formulation for the coupled generator and the rates of this generator will be obtained by solving an optimization problem that is going to be presented subsequently, see Section 3.3.3 below.

We first define a coupling of stochastic process, $\zeta_t = (\sigma_t, \eta_t)$, constructed so that (a) ζ_t is a continuous time Markov chain, hence it is easy to implement using KMC methods and (b) σ_t (respectively η_t) is a Markov process with generator $\mathcal{L}_A$ (respectively $\mathcal{L}_B$). We demonstrate the construction of such a process next. These processes σ_t, η_t need to be strongly correlated as suggested by (3.22). In order to achieve this we will define a generator for ζ_t, see Section 3.2.1, acting on an observable function on both processes, that will couple the dynamics of both processes but at the same time it will respect the dynamics of each one. More specifically, let $g(\sigma, \eta)$ be an observable on two processes. The corresponding coupled generator $\bar{\mathcal{L}}$ should be related to generators $\mathcal{L}_A$ and $\mathcal{L}_B$ and must satisfy the following property

$$\begin{aligned} \bar{\mathcal{L}} g(\sigma,\eta) &= \mathcal{L}_A f(\sigma)\,, \quad \text{if} \quad g(\sigma,\eta) = f(\sigma) \ \text{ and} \\ \bar{\mathcal{L}} g(\sigma,\eta) &= \mathcal{L}_B f(\eta)\,, \quad \text{if} \quad g(\sigma,\eta) = f(\eta)\,. \end{aligned} \tag{3.25}$$

That is $\bar{\mathcal{L}}$ reduces to $\mathcal{L}_A$ (respectively $\mathcal{L}_B$) for observables depending only on σ (respectively η). Indeed, it can be proved (see [23] Ch.3, Theorem 1.3) that if property (3.25) holds then,

$$\begin{aligned} \mathbb{E}_{(\sigma_0,\eta_0)} g(\sigma_t,\eta_t) &= \mathbb{E}_{\sigma_0} f(\sigma_t)\,, \quad \text{if} \quad g(\sigma,\eta) = f(\sigma) \ \text{ and} \\ \mathbb{E}_{(\sigma_0,\eta_0)} g(\sigma_t,\eta_t) &= \mathbb{E}_{\eta_0} f(\eta_t)\,, \quad \text{if} \quad g(\sigma,\eta) = f(\eta)\,, \end{aligned} \tag{3.26}$$

where $\mathbb{E}_{(\sigma_0,\eta_0)} g(\sigma_t, \eta_t)$ denotes the mean value of the observable function g of the state variables (σ_t, η_t) with respect to the law imposed by the initial data (σ_0, η_0). Heuristically this fact can also be seen, at least for short times, by combining (3.8), (3.10) and (3.25). These relations imply that all averages of two coupled processes, and thus all observables f, coincide with the averaged observables generated by the uncoupled generators $\mathcal{L}_A, \mathcal{L}_B$.

3.3.1 Variance reduction via coupling

We are interested in the evolution of the variance of an observable on two processes (σ_t, η_t), imposed from the dynamics of c_A and c_B. This pair of processes can be simulated using $\mathcal{L}_A$ and $\mathcal{L}_B$ or the coupled generator $\bar{\mathcal{L}}$. Variance in (3.22) can be written as

$$\operatorname{Var}\big[f(\sigma_t) - f(\eta_t)\big] = \operatorname{Var} f^2(\sigma_t) + \operatorname{Var} f^2(\eta_t) + 2\mathbb{E}f(\sigma_t)\mathbb{E}f(\eta_t) - 2\mathbb{E}\big[f(\sigma_t)f(\eta_t)\big]\,, \tag{3.27}$$

where the mean and variance in the above equation is assumed with respect to initial data (σ_0, η_0). From now on we will adopt this lighter notation unless otherwise is stated. The first three terms of the above equation cannot be controlled from coupling as they depend on the mean values of single paths, which by (3.26) are independent of coupling. However, the last term is the mean solution of a coupled process $\zeta_t = (\sigma_t, \eta_t)$ with generator $\bar{\mathcal{L}}$. Thus, if we pick $g(\sigma, \eta) = f(\sigma)f(\eta)$ and define the mean value of g with respect to initial data (σ_0, η_0),

$$u(t; \sigma_0, \eta_0) = \mathbb{E}_{(\sigma_0,\eta_0)} g(\sigma_t, \eta_t) = \mathbb{E}_{(\sigma_0,\eta_0)} f(\sigma_t)f(\eta_t) \overset{(3.8)}{=} e^{t\bar{\mathcal{L}}} f(\sigma_0)f(\eta_0)\,, \tag{3.28}$$

then u satisfies the following differential equation,

$$\partial_t u(t; \sigma_0, \eta_0) = \bar{\mathcal{L}} u(t; \sigma_0, \eta_0), \quad u(0; \sigma_0, \eta_0) = g(\sigma_0, \eta_0)\,. \tag{3.29}$$

The coupled generator $\bar{\mathcal{L}}$ will be chosen in a way such that it maximizes the quantity in (3.28) leading to a variance minimization in (3.27).

3.3.2 Microscopic coupling

For concreteness, we will first present the idea of coupling using a simple Ising type model, see also (3.14). We call this coupling a *microscopic coupling* because it couples the time series of the entire configuration, i.e. it pairs every site x of the σ process with the same site x of the η process, see Figure 3.8. Further examples of typically more efficient couplings which are designed for specific observables will be presented in Section 3.4, see also Figure 3.8. Returning to the introduction of microscopic couplings, the following coupling was introduced by Liggett [23] to study monotonicity and ergodic properties of interacting particle systems,

$$\begin{aligned} \bar{\mathcal{L}} g(\sigma, \eta) = &\sum_{x \in \Lambda_N} c(x; \sigma, \eta)\Big[g(\sigma^x, \eta^x) - g(\sigma, \eta)\Big] \\ &+ \big(c_A(x; \sigma) - c(x; \sigma, \eta)\big)\Big[g(\sigma^x, \eta) - g(\sigma, \eta)\Big] \\ &+ \big(c_B(x; \eta) - c(x; \sigma, \eta)\big)\Big[g(\sigma, \eta^x) - g(\sigma, \eta)\Big]\,, \end{aligned} \tag{3.30}$$

for $\sigma(x), \eta(x) \in \Sigma = \{0, 1\}$ where σ^x is a flip of $\sigma(x)$ to $1 - \sigma(x)$. Figure 3.1 demonstrates this coupling by visualizing the possible transitions and demonstrates that the coupled process ζ_t can be simulated like any standard KMC mechanism. Note that the coupled generator (3.30) can be written in terms of the original generators $\mathcal{L}_A$ and $\mathcal{L}_B$ as,

$$\begin{aligned} \bar{\mathcal{L}} g(\sigma, \eta) = &\mathcal{L}_A g(\sigma, \eta) + \mathcal{L}_B g(\sigma, \eta) \\ &+ \sum_{x \in \Lambda_N} c(x; \sigma, \eta)\big[g(\sigma^x, \eta^x) - g(\sigma^x, \eta) - g(\sigma, \eta^x) + g(\sigma, \eta)\big]\,. \end{aligned} \tag{3.31}$$

It is straightforward to verify that for all possible rates $c = c(x; \sigma, \eta)$ the above coupling satisfies property (3.25). Since ζ_t is Markovian, the rates in $\bar{\mathcal{L}}$ (see also Figure 3.1) must be non-negative, i.e., we have the condition

$$0 \le c(x; \sigma, \eta) \le \min\{c_A(x; \sigma), c_B(x; \eta)\}\,. \tag{3.32}$$

The form of the rate function c will be obtained next by solving an optimization problem that involves (3.32) as a constraint while the variance is optimized.

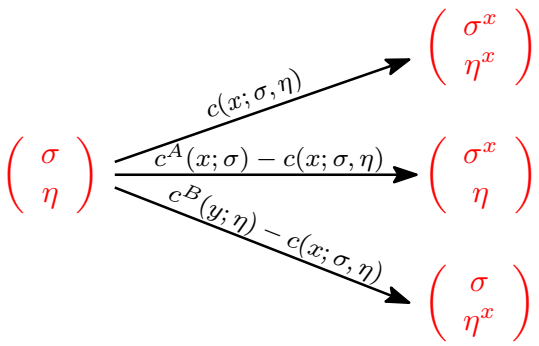


Figure 3.1: The process being at state (σ,η) can move to three different states for every $x\in\Lambda_N$ according to the coupled generator (3.30). Each state has a different probability, $p_1(x;\sigma,\eta)=c(x;\sigma,\eta)/c_0(x;\sigma,\eta)$, $p_2(x,\sigma,\eta)=(c^A(x;\sigma)-c(x;\sigma,\eta))/c_0(x;\sigma,\eta)$, $p_3(y,\sigma,\eta)=(c^B(y;\eta)-c(x;\sigma,\eta))/c_0(x;\sigma,\eta)$ and $c_0=c+c^A-c+c^B-c=c^A+c^B-c$.

In order to have a comparison basis, we define a new generator,

$$\mathcal{L}^0 g(\sigma,\eta)=\mathcal{L}_A g(\sigma,\eta)+\mathcal{L}_B g(\sigma,\eta)\,, \tag{3.33}$$

which corresponds to the case $c\equiv 0$ in (3.31). We call it the *trivial coupling* because, as suggested by the construction in Figure 3.1, there is no attempt to correlate the two time series σ_t,η_t, while (3.25) is obviously satisfied. As indicated by the simulations in Section 3.4.4 this generator leads to same variance as with using two uncoupled processes, constructed by independent KMC.

We will compute the difference of the variance between the general coupled generator $\bar{\mathcal{L}}$ that respects (3.25) and the trivially coupled generator $\mathcal{L}^0$, showing that the variance of estimator (3.20) using the coupled generator is always less or equal than that of using the trivially coupled generator. The computation is done locally in time, meaning that we examine the behavior of the variance for $t\in[0,\delta t]$ for $\delta t\ll 1$. Although the computations below follow these constraints, the numerical results in Section 3.3.4 indicate that the theoretical result is also true for large times. The presentation of a more general theoretical result is beyond the scope of this paper and can be considered by exploiting the mathematical tools presented in recent works [3, 18].

In (3.27) we see that the variance of the coupled generator depends only on the quantity (3.28). Thus the difference of the variance will depend on the same quantity,

$$\left[e^{\bar{\mathcal{L}}\delta t}-e^{\mathcal{L}^0\delta t}\right]g(\sigma_0,\eta_0)=\delta t\big(\bar{\mathcal{L}}-\mathcal{L}^0\big)g(\sigma_0,\eta_0)+\mathcal{O}(\delta t^2)\,, \tag{3.34}$$

where we used (3.11) since both generators are bounded due to (3.9). The above expansion shows that the difference in variance, locally in time, is controlled by the difference of the two generators. By comparing $\mathcal{L}^0$ with $\bar{\mathcal{L}}$ from equation (3.31) and noting that $g(\sigma,\eta)=f(\sigma)f(\eta)$, the difference becomes,

$$\begin{aligned}&\big(\bar{\mathcal{L}}-\mathcal{L}^0\big)f(\sigma)f(\eta)=\\ &=\sum_{x\in\Lambda_N}c(x;\sigma,\eta)\big[f(\sigma^x)f(\eta^x)-f(\sigma^x)f(\eta)-f(\sigma)f(\eta^x)+f(\sigma)f(\eta)\big]\\ &=\sum_{x\in\Lambda_N}c(x;\sigma,\eta)\big[f(\sigma^x)-f(\sigma)\big]\big[f(\eta^x)-f(\eta)\big]\,.\end{aligned} \tag{3.35}$$

Remark 3.3.1. *If we set $\sigma_0 = \eta_0$ in (3.35), which corresponds to the case where the two trajectories have the same initial data, then,*

$$\big(\bar{\mathcal{L}} - \mathcal{L}^0\big) f(\sigma_0) f(\sigma_0) = \sum_{x \in \Lambda_N} c(x; \sigma_0, \sigma_0) \big(f(\sigma_0^x) - f(\sigma_0)\big)^2 \geq 0\,. \tag{3.36}$$

This last relation, combined with (3.27),(3.28) and (3.34), clearly shows that the variance of $\bar{\mathcal{L}}$ is always less or equal than that of the $\mathcal{L}^0$ for processes that start from the same state.

Next we show that we can obtain such variance reduction for processes starting at different initial configurations and determine the form of the coupled rates c in (3.30) by defining and solving an appropriate optimization problem.

3.3.3 Goal Oriented Optimization as a Coupling Design Tool

Indeed, in order to find the form and hopefully the optimal choice of the rate function $c(x; \sigma, \eta)$ in (3.30), see also Figure 3.1, we can write the variance reduction problem as an optimization problem motivated by (3.27). Note that minimization of variance is equivalent of maximizing (3.28). Thus for $t = \delta t \ll 1$, equation (3.28) can be rewritten as,

$$\begin{aligned}\mathbb{E}\big[f(\sigma_{\delta t}) f(\eta_{\delta t})\big] &= e^{\bar{\mathcal{L}}\delta t} f(\sigma_0) f(\eta_0) \\ &= f(\sigma_0) f(\eta_0) + \delta t \bar{\mathcal{L}} f(\sigma_0) f(\eta_0) + \mathcal{O}\big(\delta t^2\big)\,.\end{aligned}$$

Using equation (3.31) we have:

$$\begin{aligned}\mathbb{E}\big[f(\sigma_{\delta t}) f(\eta_{\delta t})\big] =& f(\sigma_0) f(\eta_0) + \delta t \big[\mathcal{L}_A + \mathcal{L}_B\big] f(\sigma_0) f(\eta_0) \\ &+ \delta t \sum_{x \in \Lambda_N} c(x; \sigma_0, \sigma_0) \big[f(\sigma_0^x) - f(\sigma_0)\big] \big[f(\eta_0^x) - f(\eta_0)\big] + \mathcal{O}(\delta t^2)\,.\end{aligned} \tag{3.37}$$

Because the generator $\mathcal{L}_A$ acts only on the process σ_t and $\mathcal{L}_B$ acts only on η_t the above relation yields,

$$\begin{aligned}\mathbb{E}\big[f(\sigma_{\delta t}) f(\eta_{\delta t})\big] =& f(\sigma_0) f(\eta_0) + \delta t f(\eta_0) \mathcal{L}_A f(\sigma_0) + \delta t f(\sigma_0) \mathcal{L}_B f(\eta_0) \\ &+ \delta t \sum_{x \in \Lambda_N} c(x; \sigma_0, \sigma_0) \big[f(\sigma_0^x) - f(\sigma_0)\big] \big[f(\eta_0^x) - f(\eta_0)\big] + \mathcal{O}(\delta t^2)\,.\end{aligned} \tag{3.38}$$

As can be seen from the above equation, the only term we can control to reduce variance is the last term in last equation involving c. In order to minimize the variance in one time step δt, we maximize the leading order of (3.38) under the constraints (3.32):

Optimization Problem 1. *Given the rate functions $c_A(x, \sigma)$ and $c_B(x, \eta)$ and an observable function f, the rates $x(x; \sigma, \eta)$ of the coupled generator (3.30) that minimize the variance of estimator (3.20) are given by the solution of the optimization problem,*

$$\max_c \mathcal{F}[c; f] = \max_c \sum_{x \in \Lambda_N} c(x; \sigma, \eta) \big[f(\sigma^x) - f(\sigma)\big] \big[f(\eta^x) - f(\eta)\big] \tag{3.39a}$$

under the constraint

$$0 \leq c(x; \sigma, \eta) \leq \min\{c_A(x; \sigma), c_B(x; \eta)\}\,. \tag{3.39b}$$

The constraint is needed to ensure positive rates in generator (3.30), see also Figure 3.1.

The functional $\mathcal{F}$ will be used below as a diagnostic tool to design and evaluate different couplings for variance reduction. One obvious, possibly suboptimal, choice for the coupled rates, satisfying the constraint in (3.39b), is

$$c_0(x;\sigma,\eta) = \min\{c_A(x;\sigma), c_B(x;\eta)\}, \tag{3.40}$$

which we will show next that it is not the optimal choice in this class of microscopic couplings. The maximization problem (3.39) depends clearly on the choice of the observable function f: if we choose, for example, as observable the coverage on the lattice, i.e.

$$f(\sigma) = \frac{1}{N}\sum_{x\in\Lambda_N}\sigma(x)\,, \tag{3.41}$$

then it is straightforward to verify that

$$\operatorname{sign}\Big(\big[f(\sigma^x)-f(\sigma)\big]\big[f(\eta^x)-f(\eta)\big]\Big) = \begin{cases} +1, & \text{if } \sigma(x)=\eta(x) \\ -1, & \text{if } \sigma(x)\neq\eta(x) \end{cases}. \tag{3.42}$$

Thus the solution to the maximization problem (3.39) for observable (3.41) is the function

$$c_1(x;\sigma,\eta) = \begin{cases} \min\{c_A(x;\sigma), c_B(x;\eta)\}, & \text{if } \sigma(x)=\eta(x) \\ 0, & \text{if } \sigma(x)\neq\eta(x) \end{cases}. \tag{3.43}$$

which is exactly the generator proposed by Liggett [23]. Notice that both rates, (3.40) and (3.43), define microscopic couplings due to the fact that they couple the same site x for the two processes through the mechanism depicted in Figure 3.1.

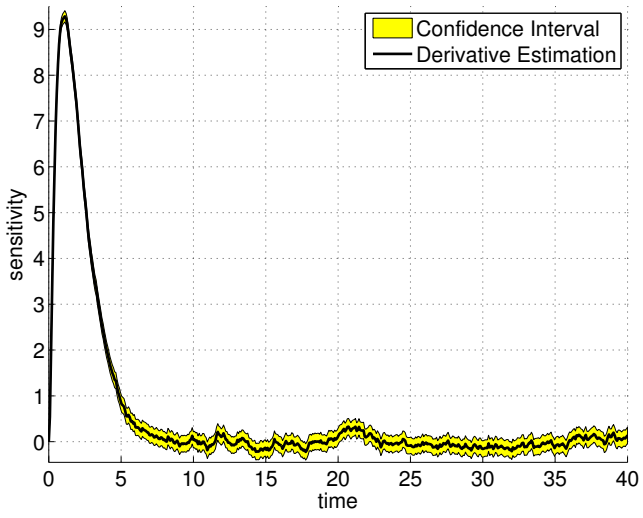


Figure 3.2: Estimation of derivative and confidence interval for the example discussed in Section 3.3.4

Finally for the functional (3.39a) holds that

$$\mathcal{F}[c_0;f] \le \mathcal{F}[c_1;f]\,. \tag{3.44}$$

Note also that from (3.31) the rates $c \equiv 0$ is the case of the trivially coupled generator (3.33).

$$0 = \mathcal{F}[0; f] \leq \mathcal{F}[c_1; f]\,. \tag{3.45}$$

Therefore the "tightest" coupling is expected to be given by (3.43). Here, the functional $\mathcal{F}$ is used as a diagnostic tool to evaluate and design different couplings obtaining the one with the maximum variance reduction. We explore this issue next with a specific example.

3.3.4 Numerical results

As a first example we will investigate the behavior of the nearest neighbor Ising model, with an adsorption/desorption mechanism described by the generator (3.13). The states, σ^x, that the system can move to are given by (3.12) and the transition rate from σ to σ^x is given by (3.14). For the simulation presented in Figure 3.2 and 3.3 the parameters are $\beta = J = h = 1$, the size of the lattice is $N = 100$ and the final time is $T = 40$. In order to estimate the derivative with respect to β we choose $\epsilon = 0.1$ and the mean is computed over 4×10^4 sample paths.

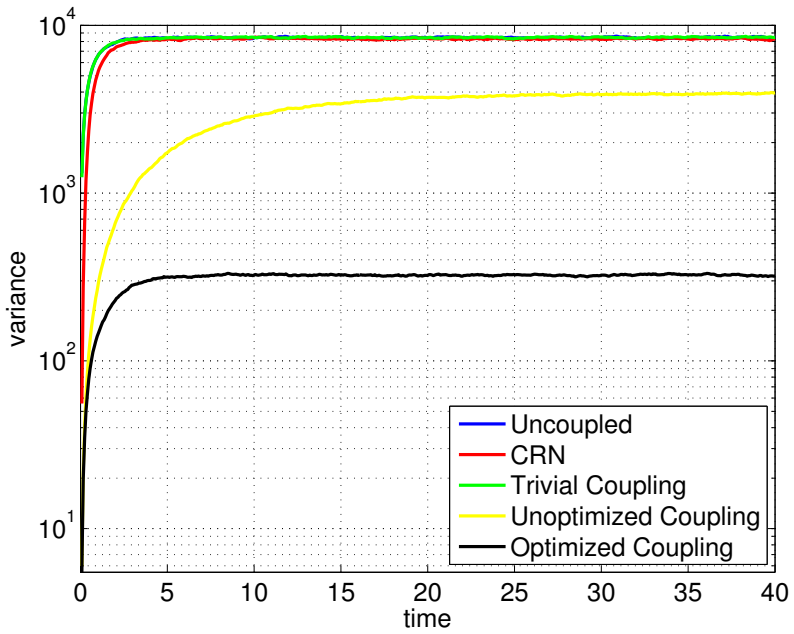


Figure 3.3: Comparison of the variance for various estimation methods in Section 3.3.3. The results are in agreement with inequality (3.44) and (3.45).

In Figure 3.2 the estimated derivative with respect to β, using the estimator (3.20) and coverage (3.41) as observable, is presented. The confidence interval of 99% is also presented as an indicator that the calculated quantity is not far from the mean. In Figure 3.3 the variance of various estimators for the simulation in Figure 3.2 is presented. To be more precise, we compute the variance using the uncoupled process, the CRN method, the trivial coupling (3.33), the unoptimized coupling (3.40) and the optimized coupling (3.43). The first three schemes have variance of the same order which is about 10^4. The unoptimized coupling and the optimized coupling gives about an order and two orders variance reduction respectively. This calculation is in qualitative agreement with the theory presented in the previous section and specifically with inequality

(3.63). Moreover, the results presented in this Section are all reproducible by Matlab scripts[1]. For a short description of the implementation see Appendix 3.C.

Note that in this simulation the sampling of the finite difference estimator (3.20) using the optimal coupling (3.43) requires 200 times fewer samples than the CRN or the uncoupled methods. At the same time the computational overhead of using the specific coupling slows down the KMC only by a factor of two leading to a scheme that is 100 times faster than schemes used up to now such as CRN.

3.4 Coupling Macroscopic Observables

Microscopic couplings defined in Section 3.3, at least heuristically, appear to be too restrictive because the same site of the first lattice is coupled with the same site of the second lattice, see Figure 3.7. It is reasonable to conjecture that a more flexible approach could be to couple certain observables of the system: for example, if we are interested in the concentration of certain species it would be preferable to couple the two systems in such a way that they will lead to correlated concentrations rather than trying to correlate the *entire* configurations of the two systems. We call this class of couplings *macroscopic couplings* due to the fact that they depend on macroscopic quantities that depend on the entire lattice, e.g. coverage, Hamiltonian, spatial correlations, surface roughness, etc, in contrast to microscopic couplings that depend on local spin configurations.

First, we will present the formulation of the generator corresponding to a general lattice KMC model, extending the discussion of the previous Section; subsequently we will present the concept of macroscopic coupling. A general optimization problem will be defined in Section 3.4.2 giving the ability to compare this new more general coupling with the microscopic coupling already defined in Section 3.3.2. Finally, we will embed all these couplings within a hierarchy of couplings (see Figure 3.8) defined on a hierarchy of *mesoscopic* geometric decompositions of the lattice, see also Figure 3.8.

3.4.1 General form of the Generator

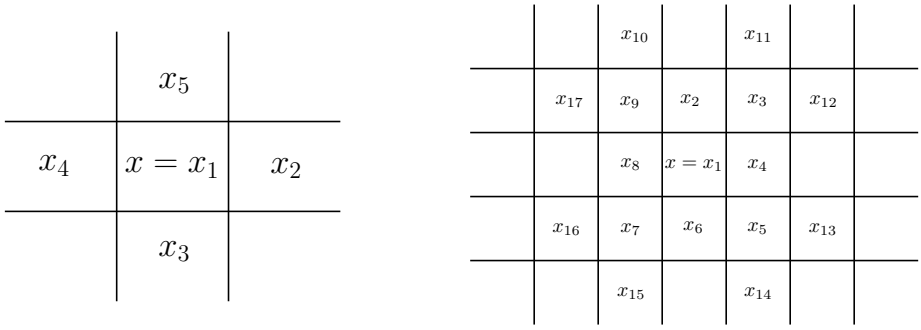


Figure 3.4: Numbering of the neighborhood N_x for (left) the Ising or the ZGB [34] model and (right) a larger neighborhood of a more complex model of catalytic CO oxidation (see [39]) (right), see also Appendix 3.A.

In order to describe a lattice KMC model in great generality one needs to define and enumerate the set of all involved species, Σ, and how these species interact with each other. The interactions are defined through the rate functions, $c(\sigma, \sigma')$, see (3.2),(3.3),(3.7). These functions give the rate at which the system at given state σ, will move to a different state σ' according to (3.2),(3.3). These changes are generally local, meaning that affect only the sites in the neighborhood of a given site $x \in \Lambda_N$, and thus the rate functions

are also local functions. In order to make clear this idea we will introduce some notation and then present the generator specifics.

Definition 3.4.1. *For a given $x \in \Lambda_N$ define $N_x = (x_1 = x, x_2, \dots, x_k)$, an ordering of* ***the neighborhood of site*** *x which describes the local geometry, see Figure 3.4. The state variable evaluated at the neighborhood sites will be denoted by $\sigma_i = \sigma(x_i)$, $i = 1, \dots, k$ The neighborhood is fully determined by the dynamics of the selected model.*

Definition 3.4.2. *Given a neighborhood N_x of size k we define the* ***local configuration*** *as a vector $\omega = (\omega_0, \dots, \omega_k) \in \Sigma^k$, where Σ^k is the set of k-tuples with elements in Σ. The* ***global updated configuration*** *affected by the local configuration ω at site x is defined as,*

$$\sigma^{x,\omega}(y) = \begin{cases} \omega_i, & y = x_i \in N_x,\ i = 1, \dots, k \\ \sigma(y), & y \notin N_x \end{cases}. \tag{3.46}$$

and the rate going from σ to $\sigma^{x,\omega}$ is $c(x; \sigma, \omega)$. Moreover we define the ***set of all accessible local configuration*** *for the given dynamics of the system, i.e., the set of all local configurations such that for a given configuration σ there exists at least one $x \in \Lambda_N$ with non zero rate,*

$$\Omega(\sigma) := \{\omega \in \Sigma^k \mid \ \text{exists } x \in \Lambda_N \text{ such that } c(x, \omega; \sigma) \neq 0\}. \tag{3.47}$$

The dependence of Ω on the configuration σ will be omitted in the sake of a lighter notation.

Using these definitions the generator can be written in the following way,

$$\mathcal{L}f(\sigma) = \sum_{x \in \Lambda_N} \sum_{\omega \in \Omega} c(x, \omega; \sigma) \big[f(\sigma^{x,\omega}) - f(\sigma) \big], \tag{3.48}$$

which will allow us to give the formulation for a general coupling. All spatial KMC models that have mechanisms such as adsorption, desorption, diffusion and reaction between species, see Section 3.2.2 and Appendix 3.A, and much more complex KMC models [38], where molecules have internal degrees of freedom and may occupy a neighborhood on a lattice, can be described by the generator (3.48), or combinations of such generators.

The starting point before we describe the macroscopic coupling will be to present the microscopic coupled generator in its full generality as it corresponds to (3.48). In analogy to (3.30) we couple every pair (x, ω) of the first process, σ_t, with every pair (y, ω') of the second process, η_t. The following generator describes this concept,

$$\begin{aligned} \bar{\mathcal{L}} g(\sigma, \eta) = & \sum_{x \in \Lambda_N} \sum_{y \in \Lambda_N} \sum_{\omega \in \Omega} \sum_{\omega' \in \Omega} c(x, y, \omega, \omega'; \sigma, \eta) \big[g(\sigma^{x,\omega}, \eta^{y,\omega'}) - g(\sigma, \eta) \big] \\ & + \sum_{x \in \Lambda_N} \sum_{\omega \in \Omega} \Big[c_A(x, \omega; \sigma) - \sum_{y \in \Lambda_N} \sum_{\omega' \in \Omega} c(x, y, \omega, \omega'; \sigma, \eta) \Big] \big[g(\sigma^{x,\omega}, \eta) - g(\sigma, \eta) \big] \\ & + \sum_{y \in \Lambda_N} \sum_{\omega' \in \Omega} \Big[c_B(y, \omega'; \eta) - \sum_{x \in \Lambda_N} \sum_{\omega \in \Omega} c(x, y, \omega, \omega'; \sigma, \eta) \Big] \big[g(\sigma^{x,\omega}, \eta^{y,\omega'}) - g(\sigma, \eta) \big], \end{aligned} \tag{3.49}$$

where the coupling rates $c(x, y, \omega, \omega'; \sigma, \eta)$ should be defined in a way such that all rates are positive (see the constraints in (3.59)). The microscopic coupling (3.48) is a generalization of the microscopic coupling in Figure 3.1.

3.4.2 Macroscopic coupling

In this section we will describe a different approach for coupling two processes (σ_t, η_t) that is based on partitioning the range of an observable function. Instead of coupling the events that happen on each lattice site, as in the microscopic coupling (3.30), we choose to couple events that belong to the same predefined class determined by level sets of the targeted observable f. This is natural choice since we are interested in the sensitivity analysis of a specific observable, hence we need to couple only the time series of that observable in (3.20). In turn, we expect that such a perspective can allow us more flexibility in the choice of coupling methods in order to minimize the variance of the sensitivity estimators. Indeed, as we demonstrate below, this goal-oriented approach will lead to greater variance reduction compared to the microscopic coupling.

First, we define a partition of the range of the discrete derivatives, $f(\sigma^{x,\omega}) - f(\sigma)$, into a finite number of classes, for example all level sets of $f(\sigma^{x,\omega}) - f(\sigma) = k$. The reason is that the macroscopic coupling will be based on coupling pairs of (x, ω) with (y, ω') that belong to the same class. We are interested in correlating time series of the observable f, hence we need to keep track of changes in f. This partition depends on the choice of the observable function f.

$$\sigma \xrightarrow{c(x,\omega;\sigma)} \sigma^{x,\omega} \quad \forall (x,\omega) \in \Lambda \times \mathcal{S}$$

$$\sigma \xrightarrow{\lambda_k(\sigma)} \boxed{S_k \text{ class selection}} \xrightarrow{\frac{c(x,\omega;\sigma)}{\lambda_k(\sigma)}} \sigma^{x,\omega} \quad \forall k \in I, \ \forall (x,\omega) \in S_k(\sigma)$$

Figure 3.5: Two different implementations of KMC for the generator (3.48) or equivalently (3.53). In the top case the choice is being done in a single step with the appropriate rate, see (3.48). In the bottom case the transition is done in two steps: first choose one of the predefined sets S_k and then choose a state from that set using a new normalized rate, see (3.53).

Definition 3.4.3. *We decompose the range of an observable function f in disjoint sets, i.e., let $J_i \subset \mathbb{R}$, $i \in I$ with $J_i \cap J_j = \emptyset$ when $i \neq j$ and $\cup_{i \in I} J_i = \mathbb{R}$. We now define the sets $S_k(\sigma)$ containing all possible events (x, ω) such that the value of $f(\sigma^{x,\omega}) - f(\sigma)$ belongs in J_k,*

$$S_k(\sigma) = S_k(\sigma; f) = \left\{ (x,\omega) \in \Lambda_N \times \Omega \ \mid \ c(x,\omega;\sigma) \neq 0 \ \text{ and } \ f(\sigma^{x,\omega}) - f(\sigma) \in J_k \right\}. \tag{3.50}$$

For example, one choice of partition are the sets

$$J_1 = (-\infty, 0), \quad J_2 = \{0\}, \quad J_3 = (0, \infty). \tag{3.51}$$

which separate the values of $f(\sigma^{x,\omega}) - f(\sigma)$ according to their sign only. We also define the corresponding rate,

$$\lambda_k(\sigma) = \sum_{(x,\omega) \in S_k(\sigma)} c(x,\omega;\sigma), \tag{3.52}$$

which is the total rate in which the set $S_k(\sigma)$ can be selected, see Figure 3.5. Thus the probability of selecting a pair (x, ω) from $S_k(\sigma)$ is $c(x, \omega; \sigma)/\lambda_k(\sigma)$. Using these definitions we can rewrite the generator for a single

process (3.48), as

$$\begin{aligned}\mathcal{L}f(\sigma) &= \sum_{x\in\Lambda_N}\sum_{\omega\in\Omega} c(x,\omega;\sigma)\big[f(\sigma^{x,\omega}) - f(\sigma)\big] \\ &= \sum_{k\in I}\sum_{(x,\omega)\in S_k(\sigma)} c(x,\omega;\sigma)\big[f(\sigma^{x,\omega}) - f(\sigma)\big] \\ &= \sum_{k\in I}\lambda_k(\sigma)\sum_{(x,\omega)\in S_k(\sigma)}\frac{c(x,\omega;\sigma)}{\lambda_k(\sigma)}\big[f(\sigma^{x,\omega}) - f(\sigma)\big]\,.\end{aligned} \tag{3.53}$$

Note that in the straightforward implementation (top of Figure 3.5, corresponding to first line of (3.53)) only one transition is needed. In our case (bottom Figure 3.5, bottom line of (3.53)) we first choose one of the predefined sets $S_k(\sigma)$ and then choose a pair (x,ω) from this set. Obviously these two methods are equivalent because they give rise to the same stochastic process since they have the same generator [23].

Remark 3.4.1. *The Bortz-Kalos-Lebowitz algorithm [4], also known as the 10-fold algorithm, can be written in the form of equation (3.53), see also Figure 3.6. In the BKL algorithm the lattice sites of the Ising model (3.14) are grouped in ten sets of equal rate. Thus, the "classes" $S_k(\sigma)$ are defined through*

$$S_k(\sigma) = \{x\in\Lambda_N \mid c(x;\sigma) = c_k\},\quad k = 1,\dots,10\,,$$

where $c_k,\ k = 1,\dots,5$ is the rate at a site with spin 0 having 0 to 4 neighbours and $k = 6,\dots,10$ is the rate at a site with spin 1 having 0 to 4 neighbours. The generator of the Ising process is written as,

$$\mathcal{L}f(\sigma) = \sum_k c_k \sum_{x\in S_k(\sigma)}\frac{c(x;\sigma)}{c_k}\big[f(\sigma^x) - f(\sigma)\big]\,.$$

Macroscopic Coupling. Equipped with the equivalent representation (3.53), we turn our attention to macroscopic coupling. In analogy with definition (3.52) we define the rates for the two processes σ and η,

$$\lambda_k^A(\sigma) = \sum_{(x,\omega)\in S_k(\sigma)} c_A(x,\omega;\sigma) \quad\text{and}\quad \lambda_k^B(\eta) = \sum_{(y,\omega')\in S_k(\eta)} c_B(y,\omega';\eta)\,. \tag{3.54}$$

Instead of coupling every lattice site x as in Figure 3.1, in this approach the coupling is being done in the selection of $S_k(\sigma)$ and $S_k(\eta)$ and then a state from each of these sets is selected independently from each other. Thus the rate at which the set $S_k(\sigma)$ and/or $S_k(\eta)$ is being chosen has three steps:

1. with the minimum of $\lambda_k^A(\sigma)$ and $\lambda_k^B(\eta)$ we choose the sets $S_k(\sigma)$ and $S_k(\eta)$, i.e., do a transition in both processes, σ and η,

2. with $\lambda_k^A(\sigma)$ subtracted the minimum from step 1. we choose the set $S_k(\sigma)$, i.e., move only the σ process and

3. with $\lambda_k^B(\eta)$ subtracted the minimum from step 1. we choose the set $S_k(\eta)$, i.e., move only the η process.

According to this mechanism, after selecting the sets $S_k(\sigma)$ and/or $S_k(\eta)$, a pair (x,ω) from $S_k(\sigma)$ is selected independently from a pair (y,ω') from $S_k(\eta)$.

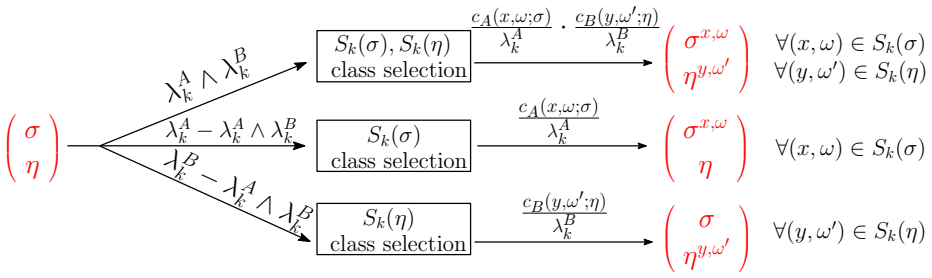


Figure 3.6: Schematic representation of the macroscopically coupled generator (3.55). The coupling involves only the selection of the set $S_k(\sigma)$, see Definition 3.4.3, and/or $S_k(\eta)$ (first transition) and then a state from each one of these sets is selected independently (second transition). Compare to microscopic coupling in Figure 3.1.

In Figure 3.6 a schematic representation of this procedure is presented. The generator that describes this coupling in the spirit of (3.53) is:

$$\begin{aligned}\bar{\mathcal{L}}g(\sigma,\eta) = &\sum_{k\in I} \min\{\lambda_k^A(\sigma), \lambda_k^B(\eta)\} \quad \dots \\ &\dots \sum_{(x,\omega)\in S_k(\sigma)} \frac{c_A(x,\omega;\sigma)}{\lambda_k^A(\sigma)} \sum_{(y,\omega')\in S_k(\eta)} \frac{c_B(y,\omega';\eta)}{\lambda_k^B(\eta)} \big[g(\sigma^{x,\omega},\eta^{y,\omega'}) - g(\sigma,\eta)\big] \\ &+ \sum_{k\in I} \Big[\lambda_k^A(\sigma) - \min\{\lambda_k^A(\sigma), \lambda_k^B(\eta)\}\Big] \sum_{(x,\omega)\in S_k(\sigma)} \frac{c_A(x,\omega;\sigma)}{\lambda_k^A(\sigma)} \big[g(\sigma^{x,\omega},\eta) - g(\sigma,\eta)\big] \\ &+ \sum_{k\in I} \Big[\lambda_k^B(\eta) - \min\{\lambda_k^A(\sigma), \lambda_k^B(\eta)\}\Big] \sum_{(y,\omega')\in S_k(\eta)} \frac{c_B(y,\omega';\eta)}{\lambda_k^B(\eta)} \big[g(\sigma,\eta^{y,\omega'}) - g(\sigma,\eta)\big]\,.\end{aligned} \tag{3.55}$$

A straightforward calculation shows that property (3.25) of a coupling generator is satisfied. Moreover, all rates in (3.55), depicted also in Figure 3.6, are positive functions. Furthermore, reordering the terms in (3.55) we can also write the total coupled rate for the general coupled generator (3.49),

$$\begin{aligned}&c_N(x,y,\omega,\omega';\sigma,\eta) = \\ &\qquad = \sum_{k\in I} \min\{\lambda_k^A(\sigma), \lambda_k^B(\eta)\} \frac{c_A(x,\omega;\sigma)}{\lambda_k^A(\sigma)} \frac{c_B(y,\omega';\eta)}{\lambda_k^B(\eta)} \chi_{S_k(\sigma)}(x,\omega)\chi_{S_k(\eta)}(y,\omega')\,,\end{aligned} \tag{3.56}$$

where

$$\chi_A(a) = \begin{cases} 1, & \text{if } a \in A \\ 0, & \text{otherwise}\,, \end{cases} \tag{3.57}$$

is the *characteristic function* of the set A. We will use this expression later in order to compare the gain in variance reduction of the macroscopic against the microscopic coupling.

In Figure 3.7 a pathwise comparison of the microscopic coupling (3.30) versus the macroscopic coupling (3.55) is presented for the example discussed in Section 3.4.4. Note that in the first snapshot there is almost perfect pointwise agreement between the two simulated configurations while in the second snapshot the distributions of particles is different between the two systems in both couplings depicted in Figure 3.7. However, the value of the observable function, which is coverage in this example, is 0.5 for the σ and 0.52 for the η process in both depicted couplings in Figure 3.7. This fact will be discussed in more details in the next section where we show that the values of the observable function are tighter coupled for the macroscopic coupling (3.56), leading to a greater variance reduction, see Figure 3.9.

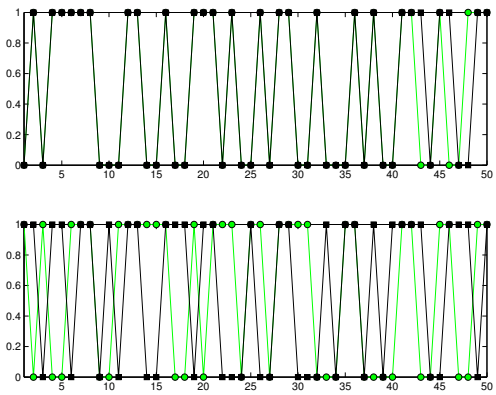


Figure 3.7: Microscopic vs macroscopic snapshots at time $T = 100$ from the coupled Ising model using the generator (3.30) (top) and the generator (3.55) (bottom). The black circles and the green squares represent the two different systems for different parameters. Note that although in the first case there is very good pointwise agreement between the two configurations while for the second this is not true, the observable used for this simulation (coverage) has values that are very close for both couplings. However, the corresponding variances are different, see Figure 3.9.

3.4.3 Comparison of Microscopic and Macroscopic Couplings

In the same spirit as in Section 3.3.3 we can formulate the maximization problem (3.39) for the general coupled generator (3.49). The functional $\mathcal{F}$ is obtained by the same procedure as the one presented in Section 3.3.3. The analogue of problem (3.39) is stated bellow.

Optimization Problem 2. *Given a collection of rate functions $c_A(x;\omega,\sigma)$ and $c_B(x;\omega,\eta)$ and an observable function f, we define*

$$\mathcal{F}[c;f] = \sum_{x\in\Lambda_N}\sum_{y\in\Lambda_N}\sum_{\omega\in\Omega}\sum_{\omega'\in\Omega} c(x,y,\omega,\omega',\sigma,\eta)[f(\sigma^{x,\omega}) - f(\sigma)][f(\sigma^{y,\omega'}) - f(\sigma)]\,. \tag{3.58}$$

Then the rates $c(x,y,\omega,\omega',\sigma,\eta)$ of the coupled generator (3.49) that minimize the variance of estimator (3.20) are given by the solution of the optimization problem,

$$\max_{c} \mathcal{F}[c;f]\,, \tag{3.59a}$$

$$\text{under the constraints}\quad 0 \le \sum_{y\in\Lambda_N}\sum_{\omega'\in S} c(x,y,\omega,\omega';\sigma,\eta) \le c_A(x,\omega;\sigma)\,, \tag{3.59b}$$

$$0 \le \sum_{x\in\Lambda_N}\sum_{\omega\in S} c(x,y,\omega,\omega';\sigma,\eta) \le c_B(y,\omega';\eta)\,. \tag{3.59c}$$

The constraints stem from the requirement that rates in generator (3.49) are positive.

It is straightforward to verify that rates (3.56) of the macroscopically coupled generator (3.55) satisfy the constraints of optimization problem (3.39). Thus, this coupling is a possibly suboptimal solution of the optimization problem (3.59).

Our next goal is to compare the macroscopic coupling (3.56) with the microscopic couplings (3.40),(3.43) presented in Section 3.3.3. We use the maximization functional (3.58) in order to compare different couplings, following along the lines of the simpler setting in section 3.3.3. In fact, it is easy to see that these

microscopically coupled generators can be written in the general form of equation (3.49). For example, the rates for the unoptimized microscopic coupling (3.40) using coverage as an observable are,

$$c_0(x,y,\omega,\omega';\sigma,\eta) = \min\{c_A(x,\omega;\sigma), c_B(y,\omega';\eta)\}\delta(x-y)\,, \tag{3.60}$$

and for the optimized microscopic coupling (3.43) ,

$$\begin{aligned} &c_1(x,y,\omega,\omega';\sigma,\eta) = \\ &\quad = \min\{c_A(x,\omega;\sigma), c_B(y,\omega';\eta)\}\delta(x-y)\sum_{k\in I}\chi_{S_k(\sigma)}(x,\omega)\chi_{S_k(\eta)}(y,\omega')\,, \end{aligned} \tag{3.61}$$

where $J_k,\ k=1,2,3$ are the sets defined in (3.51) and χ the characteristic function (3.57). As we already know from (3.44),

$$\mathcal{F}[c_0;f] \le \mathcal{F}[c_1;f]\,, \tag{3.62}$$

where f has been chosen to be the coverage defined by (3.41),

$$f(\sigma) = \frac{1}{N}\sum_{x\in\Lambda_N}\sigma(x)\,.$$

If we compare the expressions of c_1 and c_N (3.56) we can deduce that

$$\mathcal{F}[c_1;f] \le \mathcal{F}[c_N;f]\,, \tag{3.63}$$

showing that the macroscopic coupling (3.56) may not be an optimal solution since we did not optimize the functional (3.58) but reduces the variance compared to the previous microscopic couplings (3.40),(3.43). This inequality is a special case of inequality (3.67) which will be proved in Appendix 3.B. In Section 3.4.4 a numerical demonstration of this inequality is presented, see Figure 3.10.

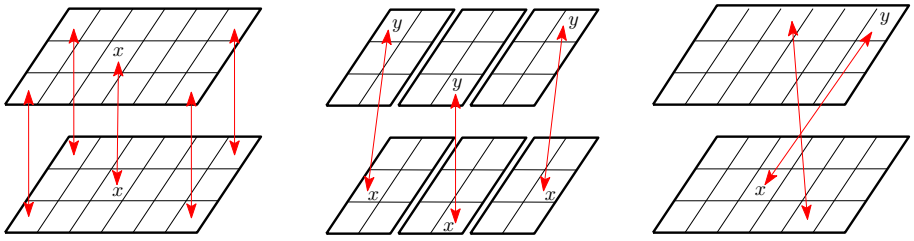


Figure 3.8: Schematic representation of the couplings, microscopic (left), coarse grained (middle) and macroscopic (right).

Hierarchy of Couplings. It is natural to ask the question if there are any other couplings between the finest microscopic and the coarsest macroscopic level. Indeed, such a coupling is a natural construction coming from coarse graining ideas [17]: the lattice is split into M cells, denoted $C_i,\ i=1,\dots,M$ and the size of every cell is $|C_i| = q$. The idea is that instead of coupling the configurations $(x,\omega),(y,\omega')$ for x,y in the whole lattice, as in the macroscopic coupling, we couple only the configurations for x,y that belong in the same cell indexed by i bellow, see Figure 3.8. The coupled rates in this case take the form,

$$\begin{aligned} &c_q(x,y,\omega,\omega';\sigma,\eta) = \\ &= \sum_{i=1}^{M}\sum_{k\in I}\min\{\lambda^A_{k,i}(\sigma),\lambda^B_{k,i}(\eta)\}\frac{c_A(x,\omega;\sigma)}{\lambda^A_{k,i}(\sigma)}\frac{c_B(y,\omega';\eta)}{\lambda^B_{k,i}(\eta)}\chi_{S_{k,i}(\sigma)}(x,\omega)\chi_{S_{k,i}(\eta)}(y,\omega')\,, \end{aligned} \tag{3.64}$$

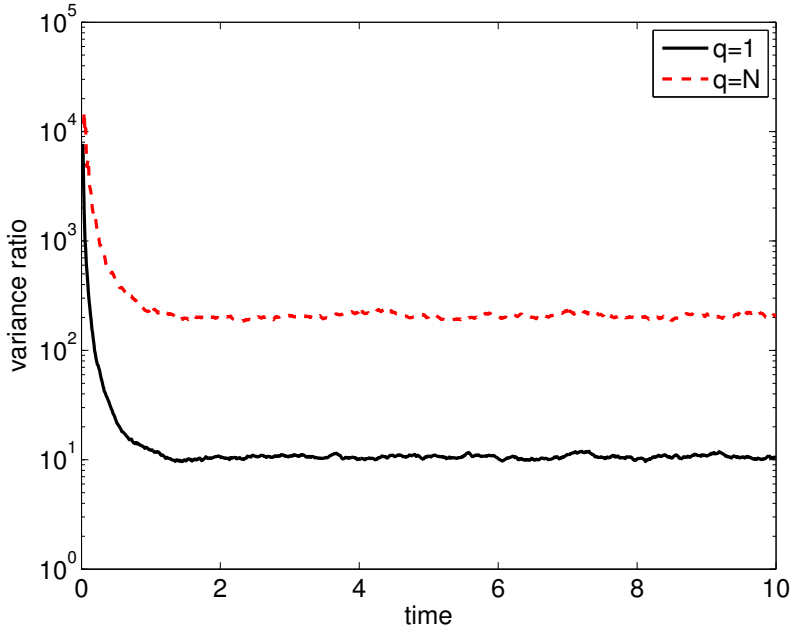


Figure 3.9: Comparison of the variance ratio between the microscopic coupling (3.30) ($q = 1$) and the macroscopic coupling (3.55)($q = N$) for the adsorption/desorption/diffusion model discussed in Section 3.4.4.

where

$$S_{k,i}(\sigma; f) = \left\{(x, \omega) \in C_i \times \Omega \;\middle|\; c(x, \omega; \sigma) \neq 0 \text{ and } f(\sigma^{x,\omega}) - f(\sigma) \in J_k\right\}, \tag{3.65}$$

and

$$\lambda^A_{k,i}(\sigma) = \sum_{(x,\omega)\in S_{k,i}(\sigma)} c_A(x, \omega; \sigma) \quad \text{and} \quad \lambda^B_{k,i}(\eta) = \sum_{(y,\omega')\in S_{k,i}(\eta)} c_B(y, \omega'; \eta)\,. \tag{3.66}$$

with the convention that $S_{k,N} = S_k$. Note that when $M = 1$, i.e. $q = N$, then the above rates are the same as the rates in (3.56) and when $M = N$, i.e., $q = 1$ then the above rates are equal to the rates of equation (3.61). We can use the functional (3.58) to compare these types of couplings. Indeed, for f defined by (3.41) we can show that (see Appendix 3.B),

$$\mathcal{F}[c_0; f] \leq \mathcal{F}[c_1; f] \leq \mathcal{F}[c_q; f] \leq \mathcal{F}[c_N; f], \quad 1 < q < N\,. \tag{3.67}$$

Note that, as in inequality (3.63), the functional $\mathcal{F}$ serves, not only as the objective function of the optimization problem, but as measure of the quality of the coupling as well. The numerical experiments presented in Figure 3.10 is in perfect agreement with the hierarchy of inequalities (3.67).

3.4.4 Numerical results

In this section we will discuss numerical results that compare the various couplings presented in the previous sections. As a reference model we use a combination of adsorption/desorption (3.12),(3.14) and diffusion (3.15),(3.16) mechanisms, see (3.17). As observable we used the coverage defined by (3.41). The parameters used in this model are $\beta = 0.1, J = 1, h = 0, c_a = c_d = c_{diff} = 1, N = 100$ and the final time $T = 10$. The perturbation is done in the β parameter with $\epsilon = 10^{-3}$. The estimator (3.20) is computed using 2000 samples.

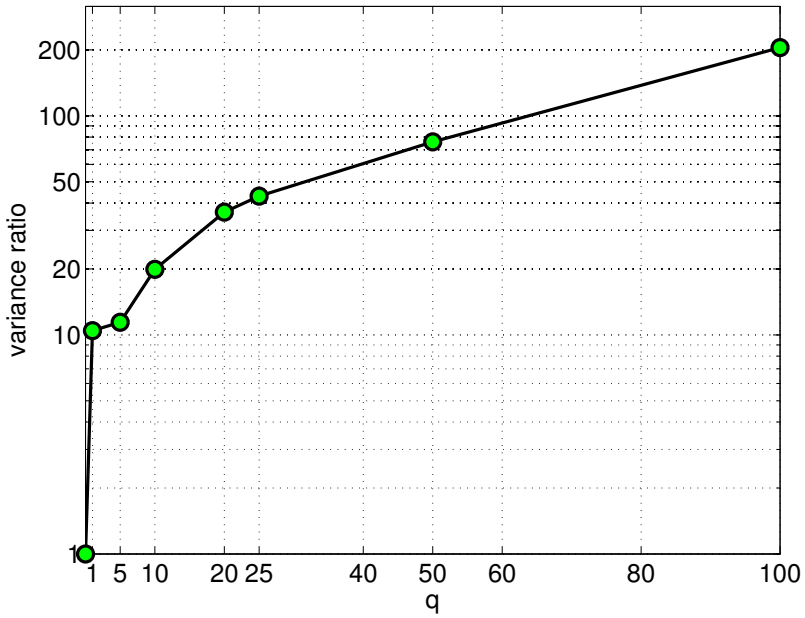


Figure 3.10: Numerical verification of inequality (3.67) where coverage (3.41) is used as the observable. The first point in the graph ($q = 0$) corresponds to the uncoupled generator (3.33). Note that the optimal choice of q is obtained for $q = N$, at least for the observable (3.41).

As suggested by inequality (3.67) in Section 3.4.2 the variance reduction using the macroscopic generator (3.55), corresponding to $q = N$, is 20 times more than that obtained using the microscopic generator (3.30), corresponding to $q = 1$, as seen in Figure 3.9. On the other hand, simulating the coupled algorithm for $q = 1$ and $q = N$ is 3.8 and 2.4 times slower, respectively, than simulating two uncoupled processes or the CRN method, see Table 3.1. Moreover, this simulation shows that the variance reduction is still obtained for large times even though the design of the algorithm is based on the assumption that $t \ll 1$, see discussion in Section 3.3.2.

In Figure 3.10 a numerical verification of inequality (3.67) is presented showing that the maximum variance reduction is obtained for $q = N$, at least for the coverage used as the observable in the calculation in Appendix 3.B. The first point in the graph, $q = 0$, corresponds to the variance ratio of the uncoupled process (3.33) and is shown here only for comparison purposes. The overall variance reduction for the macroscopic coupling, $q = N$, is 200 times more than using two uncoupled processes or the CRN method. Thus, in order to get the same variance with the macroscopic coupling, the uncoupled algorithm needs about 80 times more computational time than the macroscopically coupled process. This result combined with the execution time comparisons of Table 3.1 shows that the optimal choice for q is when $q = N$. The results presented in this section can be reproduced by Matlab scripts found in[1], see also Appendix 3.C.

q	1	2	4	5	10	20	25	50	100
$\frac{\text{coupled execution time}}{\text{uncoupled execution time}}$	3.8	3.1	2.8	2.7	2.5	2.4	2.4	2.4	2.4

Table 3.1: Ratio of coupled to uncoupled averaged execution time for the model presented in Section 3.4.4.

3.5 Observable-based coupling methods and gradient-free information theoretic tools.

The coupling sensitivity analysis methods presented in earlier sections focused on the quantification of parameter sensitivities for specific observables of the KMC process. However, in many complex spatial reaction-diffusion KMC algorithms there is a combinatorial explosion in the number of parameters [9]. The high-dimensional parameter space creates a seemingly intractable challenge for any finite-difference, *gradient-based*, sensitivity analysis method, including the low-variance coupling methods we develop here, due to the fact that all partial derivatives (3.1) need to be computed.

On the other hand in [30] the authors developed a computationally tractable, *gradient-free* sensitive analysis method suitable for complex stochastic dynamics, including spatial KMC algorithms, as well as systems with a very high-dimensional parameter space such as biochemical reaction networks with over 200 parameters [29]. The method is based on developing *computable* information theory-based sensitivity metrics such as relative entropy, relative entropy rate and Fisher Information Matrix at the path-space level, i.e., studying the sensitivity of the entire stationary time-series. Overall, in physicochemical and biological models, we focus on specific observables such as coverage, populations, spatial correlations, population variance, while important observables such as autocorrelations and exit times depend on the entire time series. Therefore, it is plausible to attempt to connect the parameter sensitivities of observables to the gradient-free, e.g. information-theoretic methods proposed [30]. Indeed, relative entropy can provide an upper bound for a large family of observable functions through the Csiszar-Kullback-Pinsker inequality [11]. More precisely, for any bounded observable function f, the Pinsker inequality states that

$$|\mathbb{E}_{Q^\theta}[f] - \mathbb{E}_{Q^{\theta+\epsilon}}[f]| \leq \max_\sigma |f(\sigma)| \sqrt{2\mathcal{R}\left(Q^\theta \,|\, Q^{\theta+\epsilon}\right)}\,, \tag{3.68}$$

and $\mathcal{R}\left(Q^\theta \,|\, Q^{\theta+\epsilon}\right)$ is the relative entropy between the *path space distributions* Q^θ and $Q^{\theta+\epsilon}$ of the time series with parameters θ and $\theta+\epsilon$ respectively. Practically, the relative entropy $\mathcal{R}\left(Q^\theta \,|\, Q^{\theta+\epsilon}\right)$ quantifies the loss of information in the time-series distribution due to a perturbation from θ to $\theta+\epsilon$. In [30] it is shown that it is a computable observable and that it can be approximated by a corresponding Fisher Information Matrix (FIM) defined on path space:

$$\mathcal{R}\left(Q^\theta \,|\, Q^{\theta+\epsilon}\right) = \frac{1}{2}\epsilon^T \mathbf{F}_{\mathcal{H}}(Q^\theta)\epsilon + O(|\epsilon|^3) \tag{3.69}$$

where $\mathbf{F}_{\mathcal{H}}(Q^\theta)$ is a $k \times k$ matrix–k is the dimension of the parameter vector θ–and can be considered as a path-wise analogue for the classical Fisher Information Matrix (FIM). Moreover, we have an explicit formula for the path-wise FIM involving only the transition rates hence it is numerically computable as an observable of the process [30]. As in the classical FIM for parametrized distributions [11], (3.69) is the Hessian of the RER which geometrically corresponds to the curvature around the minimum value of the relative entropy. Therefore, the spectral analysis of the FIM yields a derivative-free sensitivity analysis method, characterized by the the eigenvalues and eigenvectors of the FIM: the eigenvectors correspond to the (hidden) sensitivity directions of the system, and they are ordered in terms of the eigenvalues from most (higher eigenvalues) to least (lower eigenvalues) sensitive.

An outcome of (3.68) is that if the (pseudo-)distance between two distributions defined by $\mathcal{R}\left(Q^\theta \,|\, Q^{\theta+\epsilon}\right)$ is controlled, then the error between the two distributions is also controlled for any bounded observable. In the context of sensitivity analysis, inequality (3.68) implies that if the relative entropy is small, i.e., insensitive in a particular parameter direction, then, any bounded observable f is also expected to be insensitive towards the same direction. Thus, (3.68), combined with our current work on coupling suggest the following

strategy for sensitivity analysis: (a) first, the upper bound in (3.68) constitutes a theoretical indicator that relative entropy is a reliable tool for sensitivity analysis; (b) from a practical perspective, (3.68) can rule out insensitive directions in parameter space, given by eigenvectors of the FIM in (3.69); in turn, this fact provides a significant advantage in the study of models with a very large number of parameters, where the calculation of all gradients, even using coupling would be impractical; (c) finally, the remaining sensitive directions can be explored using our proposed coupling methods.

3.6 Conclusions

In this paper we proposed a coupling method for variance reduction of finite difference-based sensitivity analysis for lattice kinetic Monte Carlo algorithms. Variance reduction in sensitivity analysis is a challenge of particular importance in high-dimensional stochastic dynamics, such as the ones addressed here, since the high computational cost of individual realizations of the stochastic process renders prohibitive the generation of a large number of samples for reliable ensemble averaging. Our proposed method relies on defining a new coupled continuous time Markov Chain through a suitable generator that acts on observables of the involved high-dimensional stochastic processes associated with parameters θ and $\theta + \epsilon$. The rates of the coupled generator–and the corresponding coupled stochastic processes–are obtained by solving an optimization problem associated with minimizing the variance between the coupled stochastic dynamics. Moreover, the optimization problem depends on the particular observable quantity of interest. Thus, the form of the coupled rates depends directly on the choice of the observable function, hence it is a "goal-oriented" method. The implementation of the coupled method is a BKL–type algorithm in the sense that events are categorized into pre-defined sets. In the classical BKL algorithm [4] the events are divided into classes of equal rates, i.e. level sets of the rates. However here the events are divided into classes depending on the targeted observable's level sets, since we are interested in tight couplings of time series of specific observables. The division of the state space into classes is being done automatically by the provided code, overcoming the problem of predefining the classes. Numerical examples of spatial KMC, e.g. adsorption/desorption/diffusion processes, presented throughout the paper show that the variance reduction can be improved by two orders of magnitude compared to typical methods used up to now, such as the Common Random Number method. At the same time the computational overhead of constructing the coupled rates is two times slower than that of the CRN method leading to an overall speed up factor of two orders of magnitude. Furthermore, numerical experiments also demonstrate that the optimization functional (3.58) indeed constitutes a diagnostic tool for the design and evaluation of different couplings, e.g., see (3.67) and Figure 3.10. Finally, we noted that even the proposed coupling methods for sensitivity analysis are not an efficient approach for systems with a very high dimensional parameter space, due to the fact that coupling is a gradient method requiring the calculation of all (discrete) partial derivatives. Moreover, the results presented in this paper are all reproducible by Matlab scripts[1]. We expect that a combination of the proposed low variance coupling methods with gradient-free methods, such as the Relative Entropy Rate method [30], see (3.68), can also provide a realistic and systematic approach towards the sensitivity analysis of systems with a high-dimensional parameter space.

Acknowledgements

The work of GA was supported by the Office of Advanced Scientific Computing Research, U.S. Department of Energy under Contract No. DE-SC0002339 and by the European Union (European Social Fund) and Greece

(National Strategic Reference Framework), under the THALES Program, grant AMOSICSS. The work of MAK was supported in part by the Office of Advanced Scientific Computing Research, U.S. Department of Energy under Contracts No. DE-SC0002339 and No. DE-SC0010723.

Bibliography

[1] Matlab source code. `https://www.researchgate.net/publication/259070387_Goal-oriented_Sensitivity_Matlab_code?ev=prf_pub`.

[2] D.F. Anderson. An efficient finite difference method for parameter sensitivities of continuous-time Markov chains. *SIAM J. Numerical Analysis*, 50(5):2237–2258, 2012.

[3] G. Arampatzis, M. A. Katsoulakis, and P. Plechac. Parallelization, processor communication and error analysis in lattice kinetic monte carlo. 2012.

[4] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for Monte Carlo simulation of Ising spin systems. *J. Comp. Phys.*, 17(1):10–18, 1975.

[5] P. Bremaud and L. Massoulie. Sensitivity analysis over a random horizon: the maximal coupling rate perturbation analysis. In *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, pages 783–788 vol.1, 1992.

[6] Xi-Ren Cao and Han-Fu Chen. Perturbation realization, potentials, and sensitivity analysis of markov processes. *Automatic Control, IEEE Transactions on*, 42(10):1382–1393, 1997.

[7] A. Chatterjee and D. G. Vlachos. An overview of spatial microscopic and accelerated kinetic Monte Carlo methods. *J. Comput.-Aided Mater. Design*, 14:253–308, 2007. 10.1007/s10820-006-9042-9.

[8] N. Chen and P. Glasserman. Malliavin greeks without malliavin calculus. *Stochastic Processes and their Applications*, 117(11):1689–1723, 2007.

[9] Y. Chen, M. Salciccioli, and D. G. Vlachos. An efficient reaction pathway search method applied to the conversion of biomass derivatives on platinum. *J. Phys. Chem. C*, 115(38):18707–18720, 2011.

[10] C. H. Christensen and J. K. Norskov. A molecular view of heterogeneous catalysis. *Journal of Chemical Physics*, 128(18), 2008.

[11] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[12] L. Dai. Perturbation analysis via coupling. *Automatic Control, IEEE Transactions on*, 45(4):614–628, 2000.

[13] C. Gardiner. *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*. Springer, 4th edition, 2009.

[14] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[15] P.W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

[16] E. Hansen and M. Neurock. Predicting lateral surface interactions through density functional theory: application to oxygen on Rh(100). *Surface Science*, 441(2-3):410–424, 1999.

[17] M. A. Katsoulakis, A. Majda, and D. Vlachos. Coarse-grained stochastic processes for microscopic lattice systems. *Proc. Natl. Acad. Sci*, 100(3):782–782, 2003.

[18] M. A. Katsoulakis, P. Plecháč, and A. Sopasakis. Error analysis of coarse-graining for stochastic lattice dynamics. *SIAM J. Numer. Anal.*, 44(6):2270–2296, 2006.

[19] M. A. Katsoulakis and D. G. Vlachos. Coarse-grained stochastic processes and kinetic monte carlo simulators for the diffusion of interacting particles. *The Journal of Chemical Physics*, 119(18):9412–9427, 2003.

[20] D. Kim, B.J. Debusschere, and H.N. Najm. Spectral methods for parametric sensitivity in stochastic dynamical systems. *Biophysical Journal*, 92:379–393, 2007.

[21] C. Kipnis and C. Landim. *Scaling Limits of Interacting Particle Systems*. Springer-Verlag, 1999.

[22] W. Kohn. Nobel Lecture: Electronic structure of matter-wave functions and density functionals. *Reviews of Modern Physics*, 71(5):1253–1266, 1999.

[23] Thomas M. Liggett. *Interacting Particle Systems*, volume 276 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1985.

[24] T. Lindvall. *Lectures on the Coupling Method*. Dover, 1992.

[25] Da-Jiang Liu and J. W. Evans. Atomistic and multiscale modeling of CO-oxidation on Pd(100) and Rh(100): From nanoscale fluctuations to mesoscale reaction fronts. *Surf. Science*, 603:1706–1716, 2009.

[26] J. A. McGill, B. A. Ogunnaike, and D. G. Vlachos. Efficient gradient estimation using finite differencing and likelihood ratios for kinetic Monte Carlo simulations. *J. Comp. Phys.*, 231(21):7170–7186, 2012.

[27] H. Metiu. Preface to special topic: A survey of some new developments in heterogeneous catalysis. *Journal of Chemical Physics*, 128(18), 2008.

[28] M. Nakayama, A. Goyal, and P. W. Glynn. Likelihood ratio sensitivity analysis for Markovian models of highly dependable systems. *Stochastic Models*, 10:701–717, 1994.

[29] Y. Pantazis, M. Katsoulakis, and D. Vlachos. Parametric sensitivity analysis for biochemical reaction networks based on pathwise information theory. *BMC Bioinformatics*, 14(1), 2013.

[30] Y. Pantazis and M. A. Katsoulakis. A relative entropy rate method for path space sensitivity analysis of stationary complex stochastic dynamics. *The Journal of Chemical Physics*, 138(5):054115, 2013.

[31] S. Plyasunov and A. P. Arkin. Efficient stochastic sensitivity analysis of discrete event systems. *J. Comp. Phys.*, 221:724–738, 2007.

[32] M. Rathinam, P. W. Sheppard, and M. Khammash. Efficient computation of parameter sensitivities of discrete stochastic chemical reaction networks. *J. Chem. Phys.*, 132:034103–(1–13), 2010.

[33] K Reuter, D Frenkel, and M Scheffler. The steady state of heterogeneous catalysis, studied by first-principles statistical mechanics. *Physical Review Letters*, 93(11), SEP 10 2004.

[34] Z. M. Robert, E. Gulari, and Y. Barshad. Kinetic phase transitions in an irreversible surface-reaction model. *Phys. Rev. Lett.*, 56:2553–2556, 1986.

[35] P.W. Sheppard, M. Rathinam, and M. Khammash. A pathwise derivative approach to the computation of parameter sensitivities in discrete stochastic chemical systems. *J. Chem. Phys.*, 136(3):034115, 2012.

[36] R. Srivastava, D. F. Anderson, and J. B. Rawlings. Comparison of finite difference based methods to obtain sensitivities of stochastic chemical kinetic models. *J. Chem. Phys.*, 138(7):074110, 2013.

[37] M. Stamatakis and D. G. Vlachos. Unraveling the Complexity of Catalytic Reactions via Kinetic Monte Carlo Simulation: Current Status and Frontiers. *ACS Catalysis*, 2(12):2648–2663, 2012.

[38] Michail Stamatakis and Dionisios G. Vlachos. A graph-theoretical kinetic Monte Carlo framework for on-lattice chemical kinetics. *Journal of Chemical Physics*, 134(21), JUN 7 2011.

[39] Yu. Suchorski, J. Beben, E. W. James, J. W. Evans, and R. Imbihl. Fluctuation-induced transitions in a bistable surface reaction: Catalytic co oxidation on a pt field emitter tip. *Phys. Rev. Lett.*, 82:1907–1910, 1999.

[40] C. Wu, D. J. Schmidt, C. Wolverton, and W. F. Schneider. Accurate coverage-dependence incorporated into first-principles kinetic models: Catalytic NO oxidation on Pt (111). *J. Catalysis*, pages 88–94, 2012.

Appendix 3.A Examples of lattice models

In order to lighten the notation in the following examples, let us give a helpful definition.

Definition 3.A.1. *Let A be a set. Then,*

$$A[\textbf{\textit{condition}}] = \begin{cases} A, & \textit{if } \textbf{\textit{condition}} \textit{ is satisfied,} \\ \emptyset, & \textit{else} \end{cases} . \tag{3.70}$$

Example 1 The ZGB model [34] is a simplified atomistic lattice-gas model of CO oxidation and consists of three reactions,

$$\begin{aligned} &\text{CO adsorption on an empty site} \\ &\text{O}_2 \text{ adsorption on two neigbor empty sites} \\ &\text{CO} + \text{O} \longrightarrow \text{CO}_2 \text{ and instantaneous desorption leaving two empty neigbor sites}\,. \end{aligned}$$

In this case $\Sigma = \{-1, 0, 1\}$, where $-1, 0$ and 1 correspond to CO, vacant site and O respectively. The neighborhood Ω_x is described in Figure 3.4. Let us also define $\sigma_i := \sigma(x_i)$. Then the possible transitions are given through the rate function,

$$c(x;\sigma,\omega) = \begin{cases} c_a, & \text{if } \omega \in \Omega_1(x,\sigma), \\ 1 - c_a, & \text{if } \omega \in \Omega_2(x,\sigma), \\ c_r, & \text{if } \omega \in \Omega_3(x,\sigma), \\ 0, & \text{else}\,, \end{cases} \tag{3.71}$$

where using notation (3.70) we write,

$$\begin{aligned} \Omega_1(x,\sigma) &= \Big\{(-1,\sigma_2,\ldots,\sigma_5)\Big\}\Big[\sigma_1 = 0\Big] \\ \Omega_2(x,\sigma) &= \bigcup_{i=1}^{4}\Big\{(1,\sigma_2,\ldots,\omega_i,\ldots,\sigma_5) \text{ with } \omega_i = 1\Big\}\Big[\sigma_1 = 0 \text{ and } \sigma_i = 0\Big] \\ \Omega_3(x,\sigma) &= \bigcup_{i=1}^{4}\Big\{(0,\sigma_2,\ldots,\omega_i,\ldots,\sigma_5) \text{ with } \omega_i = 0\Big\}\Big[\sigma_1 = \pm 1 \text{ and } \sigma_i = \mp 1\Big]\,. \end{aligned}$$

Example 2 A more refined model of CO oxidation is presented in [39]. This model has the same reactions with the ZGB model with the addition of CO diffusion. In order to take into account the strong O-O repulsions, the O_2 desorption is done at diagonal nearest-neighbor sites with the additional constraint that the six sites adjacent to these are empty. The neighborhood of the model is presented in Figure 3.4.

In order to describe the sets of possible local configurations let us first define the set

$$D_i = \{\text{the indices of the adjacent neighbors at site } x_i \in \Omega_x\},\ i = 3, 5, 7, 9\,,$$

e.g. $D_3 = \{2, 4, 11, 12\}$, see Figure 3.4. Then the possible transitions are given through the rate function,

$$c(x;\sigma,\omega) = \begin{cases} c_a, & \text{if } \omega \in \Omega_1(x,\sigma), \\ 1 - c_a, & \text{if } \omega \in \Omega_2(x,\sigma), \\ c_{diff}, & \text{if } \omega \in \Omega_3(x,\sigma), \\ c_d, & \text{if } \omega \in \Omega_4(x,\sigma), \\ c_r, & \text{if } \omega \in \Omega_5(x,\sigma), \\ 0, & \text{else}\,, \end{cases} \tag{3.72}$$

where using notation (3.70) we write,

$$\begin{aligned}
\Omega_1(x,\sigma) &= \Big\{(-1,\sigma_2,\ldots,\sigma_{17})\Big\}\Big[\sigma_1=0\Big]\\
\Omega_2(x,\sigma) &= \bigcup_{i=3,5,7,9}\Big\{(1,\sigma_2,\ldots,\omega_i,\ldots,\sigma_{17}) \text{ with } \omega_i=1\Big\}\cdots\\
&\qquad\qquad\cdots\Big[\sigma_1=0 \text{ and } \sigma_i=0 \text{ and } \sigma_j=0,\ \forall j\in D_i\Big]\\
\Omega_3(x,\sigma) &= \bigcup_{i=2,4,6,8}\Big\{(0,\sigma_2,\ldots,\omega_i,\ldots,\sigma_{17}) \text{ with } \omega_i=-1\Big\}\Big[\sigma_1=-1 \text{ and } \sigma_i=0\Big]\\
\Omega_4(x,\sigma) &= \Big\{(0,\sigma_2,\ldots,\sigma_{17})\Big\}\Big[\sigma_1=-1\Big]\\
\Omega_5(x,\sigma) &= \bigcup_{i=2,4,6,8}\Big\{(0,\sigma_2,\ldots,\omega_i,\ldots,\sigma_{17}) \text{ with } \omega_i=0\Big\}\Big[\sigma_1=\pm1 \text{ and } \sigma_i=\mp1\Big]\,.
\end{aligned}$$

Appendix 3.B Proof of inequality (3.67)

In order to prove inequality (3.63) we use the coverage (3.41) as an observable and the sets (3.51) as the partition sets. Thus the values of the $f(\sigma^{x,\omega})-f(\sigma)$ in (3.58) are $-1,0,1$ for $k=1,2,3$ respectively. Using the rates (3.64) into the objective function (3.58) of the optimization problem (3.59), we have:

$$\begin{aligned}
\mathcal{F}[c_q;f] &= \sum_{j=1}^{M}\sum_{j'=1}^{M}\sum_{k=1,3}\sum_{k'=1,3}\sum_{(x,\omega)\in S_{k,j}(\sigma)}\sum_{(y,\omega')\in S_{k',j'}(\eta)}\cdots\\
&\qquad\ldots\min\{\lambda^A_{k,j}(\sigma),\lambda^B_{k',j'}(\eta)\}\frac{c_A(x,\omega;\sigma)}{\lambda^A_{k,j}(\sigma)}\frac{c_B(y,\omega';\eta)}{\lambda^B_{k',j'}(\eta)}\delta(k-k')\delta(j-j')\\
&= \sum_{k=1,3}\sum_{j=1}^{M}\min\{\lambda^A_{k,j}(\sigma),\lambda^B_{k,j}(\eta)\}\Big[\sum_{(x,\omega)\in S_{k,j}}\sum_{(y,\omega')\in S_{k,j}}\frac{c_A(x,\omega;\sigma)}{\lambda^A_{k,i}(\sigma)}\frac{c_B(y,\omega';\eta)}{\lambda^B_{k,i}(\eta)}\Big]\\
&= \sum_{k=1,3}\sum_{j=1}^{M}\min\{\lambda^A_{k,j}(\sigma),\lambda^B_{k,j}(\eta)\}\,,
\end{aligned}\tag{3.73}$$

where $\delta(x)=1$ if $x=0$ and zero otherwise. The absence of $k=2$ is justified by the fact that $[f(\sigma^{x,\omega})-f(\sigma)][f(\sigma^{y,\omega'})-f(\sigma)]$ is zero for $(x,\omega)\in S_{k,j}(\sigma)$ and $(y,\omega')\in S_{k,j}(\eta)$ while when $k=1,3$ the same product equals to 1. By fixing k in the summation of the last part of (3.73) we have:

$$\begin{aligned}
&\sum_{j=1}^{M}\min\{\lambda^A_{k,j}(\sigma),\lambda^B_{k,j}(\eta)\} =\\
&\qquad= \sum_{j=1}^{M}\min\{\sum_{(x,\omega)\in S_{k,j}(\sigma)}c_A(x,\omega;\sigma),\sum_{(y,\omega')\in S_{k,j}(\eta)}c_B(y,\omega';\eta)\}\\
&\qquad\le \min\{\sum_{j=1}^{M}\sum_{(x,\omega)\in S_{k,j}(\sigma)}c_A(x,\omega;\sigma),\sum_{j=1}^{M}\sum_{(y,\omega')\in S_{k,j}(\eta)}c_B(y,\omega';\eta)\}\\
&\qquad= \min\{\sum_{(x,\omega)\in S_k(\sigma)}c_A(x,\omega;\sigma),\sum_{(y,\omega')\in S_k(\eta)}c_B(y,\omega';\eta)\}\\
&\qquad= \min\{\lambda^A_k(\sigma),\lambda^B_k(\eta)\}\,.
\end{aligned}\tag{3.74}$$

Finally, by summation over k in both sides of (3.74) and that fact that

$$\mathcal{F}[c_N; f] = \sum_{k=1,3} \min\{\lambda_k^A(\sigma), \lambda_k^B(\eta)\}\,, \tag{3.75}$$

we get that

$$\mathcal{F}[c_q; f] \leq \mathcal{F}[c_N; f], \quad 1 \leq q < N. \tag{3.76}$$

Appendix 3.C Implementation

The goal-oriented sensitivity analysis method proposed in this work was implemented in Matlab and the source code can be downloaded from here [1]. The code was written in a way such that the user can easily specify and run different models: the neighborhood (see definition (3.4.1)), the set of all accessible local configurations (see definition (3.4.2)) and the corresponding rates (see also Appendix 3.A) are described in two files with a specific pre-defined format. In a different file the preferred observable function can be defined by the user. The microscopic (3.60),(3.61), macroscopic (3.56) and all intermediate couplings (3.64) can be executed by changing the value of a variable. Moreover, the code can easily handle 1D as well as 2D models.

The code is split into multiple functions so that new methods can be easily integrated. For example, the current version of the code includes only the the Stochastic Simulation Algorithm [14] but a different method for the time advancement of the stochastic system can be included. Finally, all examples and figures presented in Sections 3.3.4 and 3.4.4 can be reproduced by the included Matlab scripts [1].