2014

Parallelization, Processor Communication and Error Analysis in Lattice Kinetic Monte Carlo

Giorgos Arampatzis
Markos Katsoulakis, *University of Massachusetts - Amherst*
Petr Plechac

PARALLELIZATION, PROCESSOR COMMUNICATION AND ERROR ANALYSIS IN LATTICE KINETIC MONTE CARLO[*]

GIORGOS ARAMPATZIS[†], MARKOS A. KATSOULAKIS[‡], AND PETR PLECHÁČ[§]

Abstract. 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 our 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.

One of the key implications of our error analysis is that it allows us to address systematically the processor communication of different parallelization strategies for KMC by comparing their (partial) asynchrony, which in turn is measured by their respective fractional time step Δt for a prescribed error tolerance.

Key words. Kinetic Monte Carlo method, parallel algorithms, Markov semigroups, operator splitting, partially asynchronous algorithms, Graphical Processing Unit (GPU)

AMS subject classifications. 65C05, 65C20, 82C20, 82C26

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 a new mathematical and computational framework for constructing parallel algorithms for KMC simulations.The parallel algorithms in [1] are controlled approximations of Kinetic Monte Carlo algorithms, and rely on first developing a spatio-temporal decomposition of the Markov operator for the underlying CTMC into a hierarchy of operators corresponding to the particular parallel architecture. Based on this operator decomposition, we formulated *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 processor hierarchies. The resulting parallel algorithms are inherently *partially asynchronous* as processors do not communicate during the fractional time step window Δt.

Earlier, in [23] the authors also proposed an *approximate* algorithm, in order to create a parallelization scheme for KMC. It was demonstrated in [19, 20], that

[*] 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.A.K. was partially supported 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.

[†] Department of Applied Mathematics, University of Crete and Foundation of Research and Technology-Hellas, Greece, garab@math.uoc.gr

[‡] Department of Mathematics and Statistics, University of Massachusetts, Amherst, MA 01003–9305, USA, markos@math.umass.edu

[§] Department of Mathematical Sciences, University of Delaware, Newark, DE 19716, USA, plechac@math.udel.edu

boundary inconsistencies are resolved in a straightforward fashion, while there is an absence of global communications. Finally, among the parallel algorithms tested in [19], the one in [23] had the highest parallel efficiency. In [1], we demonstrated that the approximate algorithm in [23] is a special case of the Fractional Step Approximation schemes introduced in [1]. There we also demonstrated, using the Random Trotter Theorem [13], that the algorithm in [23] is *numerically 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 open source SPPARKS parallel Kinetic Monte Carlo simulator, [20], can also be formulated as a Fractional Step approximation. In this article, the convergence, reliability and efficiency of all such Fractional Step KMC parallelization methods is systematically explored by rigorous numerical analysis which relies on controlled-error approximations in transient regimes relevant to the simulation of extended systems.

A key aspect of the presented analysis relies on emphasizing a *goal-oriented* error approach for suitably defined macroscopic observables, e.g., density, energy, correlations, surface roughness , giving rise to estimates which are independent of the (very large) system size of the particle system. Besides the obtained numerical consistency and reliability of the approximating CTMC obtained from FS-KMC there is an additional key practical point: the bigger is the allowable Δt, within a desired error tolerance, the less processor communication is required. From a broader mathematical perspective, and driven from parallel computing challenges, the developed mathematical and numerical analysis attempts to balance between controlled error approximations and processor communication. The same methods could also prove useful for developing and evaluating parallel numerical schemes for other molecular and extended systems.

2. Background. Kinetic Monte Carlo (KMC) algorithms have proved to be an important tool for the simulation of *non-equilibrium, spatially distributed* chemical processes arising in applications ranging from materials science, catalysis and reaction engineering, 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 porous media, [14, 2, 4]. Furthermore, mathematically similar mechanisms and corresponding KMC simulations arise in *agent-based* models in epidemiology, ecology and traffic networks, [24].

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 $\Sigma = \{0, 1, \dots, K\}$. 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. First, 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., $\Omega_x = \{y \in \Lambda_N \mid |x-y| = 1\}$ and $\mathcal{S}_x$ is the set of all possible configurations $\mathcal{S}_x = \Sigma^{\Omega_x}$. Computationally the sample paths $\{\sigma_t\}_{t\geq 0}$ are constructed via KMC, that is through the procedure described in (2.2) and (2.3) below.

The studied stochastic processes are set on a lattice (square, hexagonal, etc.) Λ_N with N sites, they have 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, a 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

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

which correspond to an updating micro-mechanism from a current configuration $\sigma_t = \sigma$ of the system to a new configuration $\sigma^{x,\omega}$ by performing an update in a neighborhood of each site $x \in \Lambda_N$. Here ω is an index for all possible configurations $\mathcal{S}_x$ that correspond to an update at a neighborhood Ω_x of the site x; we refer to the end of the section for specific examples.

The probability of a transition over an infinitesimal time interval δt is

$$\mathbb{P}\left(\sigma_{t+\delta t} = \sigma^{x,\omega} \mid \sigma_t = \sigma\right) = c(x,\omega;\sigma)\delta t + \mathrm{o}(\delta t)\,.$$

Realizations of the process are constructed from the embedded discrete time Markov chain $S_n = \sigma_{t_n}$ (see [11]), 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 $P(\sigma,t;\zeta) := \mathbb{P}\left(\sigma_t = \sigma \mid \sigma_0 = \zeta\right)$ 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), [15, 6],

$$\partial_t P(\sigma,t;\zeta) := \sum_{\sigma',\sigma'\neq\sigma} c(\sigma',\sigma)P(\sigma',t;\zeta) - \lambda(\sigma)P(\sigma,t;\zeta)\,, \tag{2.4}$$

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

In [1] we developed a general 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 continous 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 5. Alternatively, we may consider observables depending on infinitely many variables $\sigma(x)$, $x \in \Lambda = \mathbb{Z}^d$, to stress the necessity of working with the infinite volume limit.

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

$$u(\zeta, t) := \mathbb{E}^\zeta[f(\sigma_t)] = \sum_\sigma f(\sigma) P(\sigma, t; \zeta) \,, \tag{2.5}$$

conditioned on the initial data $\sigma_0 = \zeta$. By a straightforward calculation using (2.4) we obtain that the observable (2.5) satisfies the initial value problem

$$\partial_t u(\zeta, t) = \mathcal{L} u(\zeta, t) \,, \qquad u(\zeta, 0) = f(\zeta) \,, \tag{2.6}$$

where the operator $\mathcal{L} : C_b(\mathcal{S}) \to C_b(\mathcal{S})$ is known as the *generator* of the continous time Markov chain, [15], and in the case of (2.1) it is

$$\mathcal{L} f(\sigma) = \sum_{\sigma'} c(\sigma, \sigma')[f(\sigma') - f(\sigma)] = \sum_{x \in \Lambda_N} \sum_{\omega \in \mathcal{S}_x} c(x, \omega; \sigma)[f(\sigma^{x,\omega}) - f(\sigma)] \,. \tag{2.7}$$

We then write (2.5), as the the action of the Markov semi-group $e^{t\mathcal{L}}$ associated with the generator $\mathcal{L}$ and the process $\{\sigma_t\}_{t \geq 0}$, [15], on the observable f

$$u(\zeta, t) = \mathbb{E}^\zeta[f(\sigma_t)] = e^{t\mathcal{L}} f(\zeta) \,, \tag{2.8}$$

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 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 5.

Definition 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 $\mathcal{S} = \Sigma^{\Lambda_N}$

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

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 5, see (5.8) and (5.10) in Theorem 5.6, we will employ spaces with higher discrete derivatives that will be defined in Section 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 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. *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,*

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

Proof. Follows directly from (2.7) and the definition of $C^1(\mathcal{S})$. □

PROPOSITION 2.3. *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*

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

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

Proof. See [15, 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.

EXAMPLES.

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 $z \in \Lambda_N$

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

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 $\sigma_t = \sigma$ 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, [4].

Multicomponent reactions. Reactions that involve K species of particles are easily described by enlarging the spin space to $\Sigma = \{0, 1, \dots, K\}$. 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 $k = 1, \dots, K$ or vacating a site, i.e., $k = 0$, over δt.

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, [16], when only particles at a nearest-neighbor distance can react we set $\sigma(x) \in \Sigma = \{0, 1, \dots, K\}$, $\Omega_x = \{y \in \Lambda_N \,|\, |x - y| = 1\}$ 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 $|x - y| = 1$.

3. Towards parallel kinetic Monte Carlo algorithms. In practice, the sample paths $\{\sigma_t\}_{t\geq 0}$ 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. Implementations are based on the efficient calculation of transition probabilities, e.g., [3] for Ising models, known as a BKL Algorithm, and in [7] known as Stochastic Simulation Algorithm (SSA) for reaction systems.

It is evident from formulas (2.2) and (2.3), that KMC algorithms are *inherently serial* as updates are done at one site $x \in \Lambda_N$ at a time, while on the other hand (2.2) depends on information from the entire spatial domain Λ_N. For these reasons it appears that KMC cannot be parallelized easily. However, Lubachevsky, in [17], 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 (2.2); thus the approach yields a null-event algorithm, [14], which includes rejected moves over the entire spatial sub-domain that corresponds to each processor, see also [9]. A modification in order to incorporate the BKL Algorithm was proposed in [17], which was implemented and tested in [12]. This is a still asynchronous algorithm, where BKL-type rejection-free simulations are carried out in the interior of each sub-domain (processor), while uniform rates are used at the boundaries, reducing rejections to just the boundary set. However, these asynchronous algorithms may still have a high number of rejections for boundary events and rollbacks, which considerably reduce the parallel efficiency, [22]. 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 in [5], [22], [18], [19]. 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 processor communication, [19]. As we will discuss in the sequel, many of these issues with parallel KMC can be addressed by abandoning the earlier perspective on creating a parallel KMC algorithm with exactly the same rates $c(x, \omega; \sigma)$ in (2.7) as the serial algorithm.

Indeed, in [1], we adopted the approach of creating a parallel KMC algorithm which *approximates* the underlying continuous time Markov chain of the serial algorithm instead of reproducing its master equation exactly. We proposed a spatio-temporal decomposition for the Markov operator underlying the KMC algorithm into a hierarchy of operators corresponding to the processor architecture. Based on this operator decomposition we can formulate *Fractional Step KMC Approximation* schemes by employing the Trotter product formula. In turn these approximating schemes determine 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. Earlier, in [23] the authors also proposed an *approximate* algorithm, in order to create a parallelization scheme for KMC. It was demonstrated in [19, 20], that boundary inconsistencies are resolved in a straightforward fashion, while there is an absence of global communications. Finally, among the parallel algorithms tested in [19], the one in [23] had the highest parallel efficiency. In [1], we demonstrated that the approximate algorithm in [23] is a special case of the *Fractional Step Approximation* schemes introduced in [1]. We also demonstrated, using the Random Trotter Theorem, [13], that the algorithm in [23] is *numerically 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. Finally, the open source SPPARKS parallel Kinetic Monte Carlo simulator, [20], also relies on such Fractional Step approximations.

In this article, the convergence, reliability and efficiency of parallel algorithms, that fit the Fractional Step KMC approximation framework, are systematically explored by rigorous numerical analysis which relies on controlled-error approximations in transient regimes relevant to the simulation of extended systems.

3.1. Fractional time step kinetic Monte Carlo algorithms. 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 3.1. *The lattice Λ_N is decomposed into non-overlapping coarse cells C_m, $m = 1, \dots, M$ such that, $|C_m| = Q = q^d$, where d is the dimension,*

$$\Lambda_N = \bigcup_{m=1}^{M} C_m \,, \quad C_m \cap C_n = \emptyset \,,\ m \neq n \,,\ N = MQ \,. \tag{3.1}$$

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 3.1. 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 (2.7)

$$\mathcal{L}f(\sigma) = \sum_{m=1}^{M} \mathcal{L}_m f(\sigma) \,, \qquad \mathcal{L}_m f(\sigma) = \sum_{x \in C_m} \sum_{x,\omega \in \mathcal{S}_x} c(\omega;\sigma)[f(\sigma^{x,\omega}) - f(\sigma)] \,. \tag{3.2}$$

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

Remark 3.1. In many models such as in catalysis the interactions between particles are short-range, [21, 16], 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* $\bar{C}_m \cap \bar{C}_{m'} = \emptyset$. 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. 3.1), which in turn *induce* a corresponding splitting of the generator:

$$\begin{aligned} \Lambda_N = \Lambda_N^1 \cup \Lambda_N^2 &:= \bigcup_{m \in \mathcal{I}^1} C_m \cup \bigcup_{m \in \mathcal{I}^2} C_m \quad \text{and} \\ \mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 &:= \sum_{m \in \mathcal{I}^1} \mathcal{L}_{1,m} + \sum_{m \in \mathcal{I}^2} \mathcal{L}_{2,m} \,. \end{aligned} \tag{3.3}$$

The decomposition (3.3) 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

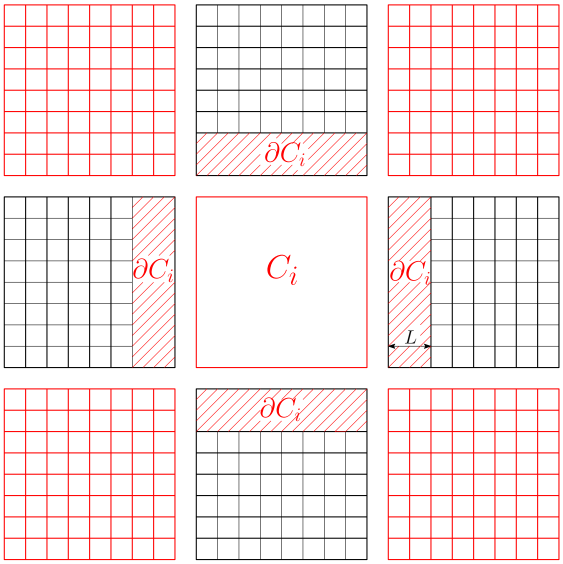

FIG. 3.1. *Lattice partitioning in (3.3). Note that later we use the notation C_i^{∂} to denote the interior boundary of C_i, see Figure 5.1*

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 (2.6) by alternating between evolution of σ's in the dimensions corresponding to $\mathcal{I}^1$ and $\mathcal{I}^2$.

Indeed, the key tool for our analysis are different versions of the Trotter formula, [25, 13],

$$e^{T\mathcal{L}} = \lim_{n\to\infty} \left[e^{\frac{T}{n}\mathcal{L}_1} e^{\frac{T}{n}\mathcal{L}_2} \right]^n \tag{3.4}$$

when applied to the operator $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$ in (3.3). 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 $n \gg 1$:

$$e^{T\mathcal{L}} \approx P_L := \left[e^{\Delta t\mathcal{L}_1} e^{\Delta t\mathcal{L}_2} \right]^n, \quad \text{where} \quad \Delta t = \tfrac{T}{n}\,. \tag{3.5}$$

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 (3.3),

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

Analogously we have the *Strang* splitting scheme

$$e^{T\mathcal{L}} \approx P_S := \left[e^{\frac{\Delta t}{2}\mathcal{L}_1} e^{\Delta t\mathcal{L}_2} e^{\frac{\Delta t}{2}\mathcal{L}_1} \right]^n, \quad \text{where} \quad \Delta t = \tfrac{T}{n}\,. \tag{3.6}$$

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.,[8], 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 6 for a complete discussion.

We characterize the FS-KMC (Fractional Step KMC) algorithm (3.5) as *partially asynchronous* since there is no processor communication during the period 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 (3.5) or in (3.6) the less processor communication we have, in which case the error in the approximation (3.5) or (3.6) worsens. This *balance between accuracy and processor communication* in algorithms is one of the themes of this article.

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 (2.5) instead on estimating an approximation of the probability distribution of the process solving (2.4). 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.

To understand the error of this approximation we first analyze the error for the two cases of deterministic PCS: the Lie splitting defines a new semigroup (3.5) that we denote P_L and similarly P_S denotes the semigroup (3.6) 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 [10]. 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 5. Finally, in Section 7 we present, as a complementing theoretical perspective, the same estimates on the infinite lattice $\Lambda = \mathbb{Z}^d$, in which case the involved generators are necessarily unbounded.

Lemma 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*

$$\mathcal{D}_m(t\mathcal{L}) = e^{t\mathcal{L}} - \sum_{k=0}^{m-1} \frac{t^k}{k!}\mathcal{L}^k \,, \qquad m \in \mathbb{N}^+ \tag{4.1}$$

satisfy the bound

$$\| \mathcal{D}_m(t\mathcal{L}) v \|_\infty \leq \frac{t^k}{k!} \| \mathcal{L}^k v \|_\infty, \quad \forall v \in C_b \tag{4.2}$$

Proof. see Jahnke, [10]. □

Lemma 4.2 (Local Error). *Let $P_L(t)$ and $P_S(t)$ be the schemes (3.5) and (3.6) associated with the Lie and Strang splittings respectively, and let $u(h) = P(h)f$ be the solution of (2.6). Then the local error for the Lie splitting is*

$$\| P_L(h)f - u(h) \|_\infty \leq c_1 \| [\mathcal{L}_1, \mathcal{L}_2] f \|_\infty h^2 + c_2 \sum_{|m|=3} \| \mathcal{L}_1^{m_1} \mathcal{L}_2^{m_2} f \|_\infty h^3 \,, \tag{4.3}$$

and for the Strang splitting scheme

$$\begin{aligned} \| P_S(h)f - u(h) \|_\infty \le c_3 \| [\mathcal{L}_1,[\mathcal{L}_1,\mathcal{L}_2]]f - 2[\mathcal{L}_2,[\mathcal{L}_2,\mathcal{L}_1]]f \|_\infty h^3 \\ + c_4 \sum_{|m|=4} \| \mathcal{L}_1^{m_1}\mathcal{L}_2^{m_2}\mathcal{L}_1^{m_3} f \|_\infty h^4 \end{aligned} \tag{4.4}$$

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 4.1 the proof follows the standard finite dimensional approach based on the expansion of the operator exponential. We present the calculations here for the sake of completeness. In order to simplify the notation, we introduce

$$a_{k,N}(h) = \begin{cases} \frac{h^k}{k!}\mathcal{L}_1^k & \text{if } k < N\,, \\ \mathcal{D}_k(h\mathcal{L}_1) & \text{if } k = N > 0\,, \\ e^{h\mathcal{L}_1} & \text{if } k = N = 0\,, \end{cases} \quad b_{k,N}(h) = \begin{cases} \frac{h^k}{k!}\mathcal{L}_2^k & \text{if } k < N\,, \\ \mathcal{D}_k(h\mathcal{L}_2) & \text{if } k = N > 0\,, \\ e^{h\mathcal{L}_2} & \text{if } k = N = 0\,. \end{cases} \tag{4.5}$$

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

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

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 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,

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

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

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

The second term is bounded by Lemma 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 (4.5). □

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

Theorem 4.3 (Global error). *Let $P_L(t)$ and $P_S(t)$ be the the schemes (3.5) and (3.6) associated with the Lie and Strang splittings respectively and let $u(t_n) = P(t_n)f$ be the exact solution of (2.6). Then the global error at the time $T = t_n = nh$, for the Lie splitting is bounded by*

$$\| P_L(t_n)u(0) - u(t_n) \|_\infty \leq C_1 \max_{k=0,\dots,n} \| [\mathcal{L}_1,\mathcal{L}_2]u(t_k) \|_\infty h + \mathcal{R}_L(u)h^2\,, \tag{4.6}$$

where the remainder is given by

$$\mathcal{R}_L(u) \equiv \mathcal{R}_L(u;n,h) = C_2 \max_{k=0,\dots,n} \sum_{|m|=3} \| \mathcal{L}_1^{m_1}\mathcal{L}_2^{m_2} u(t_k) \|_\infty\,. \tag{4.7}$$

and for the Strang scheme

$$\begin{aligned} \| P_S(t_n)u(0) - u(t_n) \|_\infty \leq &C_3 \max_{k=0,\dots,n} \| \Big([\mathcal{L}_1,[\mathcal{L}_1,\mathcal{L}_2]] - 2[\mathcal{L}_2,[\mathcal{L}_2,\mathcal{L}_1]]\Big)u(t_k) \|_\infty h^2 \\ &+\mathcal{R}_S(u)h^3\,, \end{aligned} \tag{4.8}$$

where

$$\mathcal{R}_S(u) = \mathcal{R}_S(u;n,h) = C_4 \max_{k=0,\dots,n} \sum_{|m|=4} \| \mathcal{L}_1^{m_1}\mathcal{L}_2^{m_2}\mathcal{L}_1^{m_3} u(t_k) \|_\infty \,, \tag{4.9}$$

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

Proof. It can be shown by induction that

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

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

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

Using Lemma 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 (4.6) and (4.8) for the Lie and the Strang scheme respectively.

□

5. Estimates for macroscopic observables. In Theorem 4.3 we have shown that the proposed splitting schemes are convergent as the time step h tends to zero. The main idea of the proposed scheme is to control an error for observables, in other words we estimate the weak error by analyzing solutions of (2.6). If we restrict the initial data of the problem (2.6) to a special class of functions, then it is possible to show that the error terms are independent of the size of the lattice, N. It turns out that this is a wide class of function containing some of the most common observables in KMC simulations, such as mean coverage or spatial correlations, we refer to Section 5.1.

In order to simplify the notation we suppress the dependence of the discrete derivative operator $\delta_{x,\omega}$ on ω in Definition 2.1.

Definition 5.1. *For* $\mathbf{x} = (x_1, \dots, x_m) \in \Lambda_N^m$ *we introduce the notation*

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

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

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

and the function space

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

We refer to elements of $C^m(\mathcal{S})$ as *macroscopic observables* and we will discuss examples in Section 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. The proof of this theorem is contained in the next two subsections.

THEOREM 5.3. *(a) Let $u(t)$ be the solution of (2.6) with $u(0) = f \in C^3(\mathcal{S})$. Then for the global error estimate of Theorem 4.3 on the Lie scheme (3.5) we have*

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

where

$$\| [\mathcal{L}_1, \mathcal{L}_2] u(t_k) \|_\infty < C$$

and

$$\mathcal{R}_L(u) < \tilde{C} \,,$$

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

$$\| \Big([\mathcal{L}_1, [\mathcal{L}_1, \mathcal{L}_2]] - 2[\mathcal{L}_2, [\mathcal{L}_2, \mathcal{L}_1]] \Big) u(t_k) \|_\infty < C \,,$$

and

$$\mathcal{R}_S(u) < \tilde{C} \,,$$

where the constants C and $\tilde{C}$ are independent of the system size N.
(b) Many macroscopic observables $u(0) = f$ are not just in $C^3(\mathcal{S})$ but also satisfy a local bound such as

$$\max_{z \in \Lambda_N} \|\delta_z u(0, \cdot)\|_\infty + \max_{x,y \in \Lambda_N} \| \delta_{xy} u(0, \cdot) \|_\infty + \max_{x,y,z \in \Lambda_N} \| \delta_{xyz} u(0, \cdot) \|_\infty \leq \frac{C}{N} \,. \tag{5.1}$$

Then the bounds for the commutators become

$$\| [\mathcal{L}_1, \mathcal{L}_2] u(t, \cdot) \|_\infty \leq C \, \frac{L^{d+1}}{q} \,, \tag{5.2}$$

and

$$\| \Big([\mathcal{L}_1, [\mathcal{L}_1, \mathcal{L}_2]] - 2[\mathcal{L}_2, [\mathcal{L}_2, \mathcal{L}_1]] \Big) u(t_k) \|_\infty \leq C \, \frac{L^{2d+1}}{q} \,, \tag{5.3}$$

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 3.1, and d is the dimension of the lattice $\Lambda_N \subset \Lambda = \mathbb{Z}^d$.

The proof of this theorem is given in Section 5.4, while the supporting results are proved earlier in Sections 5.2 and 5.3. Next, we discuss typical examples of macroscopic observables f which are used in KMC simulations and also satisfy the assumptions of Theorem 5.3.

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

$$\delta_x f(\sigma) := \frac{1}{N}\phi\big(\sigma(x+k_1),\cdots,\sigma(x+k_\ell)\big)\,.k_i \in \Lambda_N\,, \forall x \in \Lambda_N\,, \tag{5.4}$$

A class of functions that satisfies (5.4), or more generally (5.1), includes the coverage, spatial correlations, Hamiltonians and more generally observables of the type

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

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. In this section we will show that this class of function belong in $C^m(\mathcal{S})$, $\forall m \in \mathbb{N}^+$.

Example 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 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 5.5 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 5.3. *Let f be an observable of type (5.4) with $\ell = 1$ and $k_1 = 0$, then*

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

An analogous result holds when $\ell \ge 1$ and $k_i \ne 0$ 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 (5.4) but still satisfy (5.1)

Example 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 (5.1).

5.2. Bounds on the remainder. In order to establish that the remainders $\mathcal{R}_L(u)$, (4.7), or $\mathcal{R}_S(u)$, (4.9), Theorem 4.3, 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 easy extension of estimates on $\mathcal{L}^2$ acting on the solution of (2.6), which we present next. First, we prove that $\mathcal{L}^2 u$ is bounded by the sum of first and second derivatives of u.

Lemma 5.4. *Let u be the solution of equation (2.6). Then for the operator $\mathcal{L}^2$ the following bound holds*

$$\begin{aligned}\|\mathcal{L}^2 u(t,\cdot)\|_\infty &\le c_1 \sum_{x\in\Lambda_N} \|\delta_x u(t,\cdot)\|_\infty + c_2 \sum_{x,y\in\Lambda_N} \|\delta_{xy} u(t,\cdot)\|_\infty \\ &= c_1 \|u(t,\cdot)\|_1 + c_2 \|u(t,\cdot)\|_2 \,. \end{aligned} \tag{5.5}$$

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

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

where the first inequality follows from the fact that $\delta_x c(y,\sigma)=0$ when $|x-y|>L$, see Lemma 5.5, where we show that the derivatives of the rate functions have compact support that depends only on the length of the interaction L. □

Lemma 5.5. *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

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

Proof. For the first discrete derivative it is sufficient to observe that if $x\ne y$ 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. □

Proposition 5.6. *Let $u(t,\sigma)$ be the solution of the equation (2.6) with initial data in $C^2(\mathcal{S})$. Then the operator $\mathcal{L}^2$ satisfies the bounds,*

$$\|\mathcal{L}^2 u(t,\cdot)\|_\infty \le C\,, \tag{5.6}$$

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 (5.5) 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, $v_1 \equiv \delta_x u$, $v_2 \equiv \delta_x u$ 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 $v(t,\sigma)$

$$\delta_x u(t,\sigma) = e^{\mathcal{L}t}u(0,\sigma) + \int_0^t e^{(t-s)\mathcal{L}} \sum_{|y-x|\le N} \delta_x c(y,\sigma)\delta_y u(s,\sigma^x)\,ds\,. \tag{5.7}$$

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

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

By setting

$$\varphi(t) = \|u(t,\cdot)\|_1 = \sum_{x\in\Lambda_N}\|\,\delta_x u(t,\cdot)\,\|_\infty\,, \tag{5.8}$$

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 5.5,

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

where $\chi_{C_{2L}}$ is the characteristic function and $C_{2L} = \{(x,y)\in\Lambda_N^2 \mid |x-y| < 2L\}$. 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

$$\begin{aligned}\|\,\delta_{xy}u(t,\cdot)\,\|_\infty \le \|\,\delta_{xy}u(0,\cdot)\,\|_\infty &+ c_1\int_0^t\sum_{|z-x|\le L}\|\,\delta_{xz}u(s,\cdot)\,\|_\infty ds\\ &+ c_2\int_0^t\sum_{|z-y|\le L}\|\,\delta_{yz}u(s,\cdot)\,\|_\infty\,ds\\ &+ c_3\int_0^t\sum_{\substack{|z-x|\le L\\|z-y|\le L}}\|\,\delta_z u(s,\cdot)\,\|_\infty\chi_{C_{2L}}(x,y)\,ds\,.\end{aligned}\tag{5.9}$$

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

$$\vartheta(t) = \|u(t,\cdot)\|_2 = \sum_{x,y\in\Lambda_N}\|\,\delta_{xy}u(t,\cdot)\,\|_\infty\,,\tag{5.10}$$

we obtain

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

where both $\bar{c}_2$ and $\bar{c}_3$ depend on L but not on N. However, from Lemma A.1 (see Appendix) we have

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

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

$$\vartheta(t) \le \tilde{c}_2\vartheta(0) + \tilde{c}_3\varphi(0)\,,$$

and the second term is bounded from the previous argument whereas the first term is equal to

$$\vartheta(0) = \|u(0,\cdot)\|_2 < C\,,$$

which is true because the initial data are in $C^2(\mathcal{S})$. Finally, we obtain from Lemma 5.4,

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

□

Remark 5.1. The same result can be obtained if we notice that the function $v(t,\sigma) = \mathcal{L}^2 u(t,\sigma)$ satisfies the equation (2.6). 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 5.4 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 5.6.

Our last goal for this section is to prove that the remainders in the Lie and the Strang scheme, (4.7) and (4.9) 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 (4.7) and (4.9). Then, as in Remark 5.1, using a more general form of Lemma 5.4 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. We will present the general idea of the proof, rather than showing all the technical details that anyhow follow the same idea as in Proposition 5.6. First we give a definition of the discrete derivatives that generalizes Definition 5.1.

Definition 5.7. *Let $\mathbf{x} = (x_1,\dots,x_m) \in \Lambda_N^m$ and $\forall k\in\mathbb{N}\,, k\le m$ define $a\in\Lambda_N^k$, the k-dimensional multi-index of k-tuples of $\mathbf{x}$. As in Definition 2.1, the discrete derivative with respect to a is*

$$\delta_a f(\sigma) = \delta_{a_1\dots a_k} f(\sigma)\,,$$

and we define $\delta_{-a} f(\sigma)$ the derivative with respect to all variables in $\mathbf{x} = (x_1,\dots,x_m)$ that are not contained in a.

Using this definition we are able to write a general form of the governing equation for the m-th discrete derivative,

$$\begin{aligned}\partial_t \delta_{\mathbf{x}} u(t,\sigma) &= \sum_{0\le|\alpha|\le m}{}' \delta_\alpha c(\sigma)\cdot\nabla_\sigma \delta_{-\alpha} u(t,\sigma^\alpha) \\ &= \mathcal{L}\delta_{\mathbf{x}} u(t,\sigma) + \sum_{1\le|\alpha|\le m}{}' \delta_\alpha c(\sigma)\cdot\nabla_\sigma \delta_{-\alpha} u(t,\sigma^\alpha)\,, \qquad (5.11)\end{aligned}$$

where the prime in the summation symbol means that we sum over tuples without distinguishing the order of the variables, e.g., $\delta_{xy} = \delta_{yx}$. Using this representation and the Remark A.1 we can apply the same idea as in Proposition 5.6 and prove bounds for the operator $\mathcal{L}^k, k\in\mathbb{N}^+$, with initial data in $C^k(\mathcal{S})$.

5.3. Bounds on the commutators. The constants in the local error estimate derived in Lemma 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 $\mathcal{L}_2$. The commutator is directly related to the geometric decomposition and the range of particle interactions. In order to demonstrate this relation more specifically we discuss an example of Ising-type interacting system in which the events (updates) occur only at a single site $x\in\Lambda_N$.

The error estimates in Lemma 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 become too complicated and impractical. 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 3.1).

Lemma 5.8. *Let $\mathcal{L}_1,\mathcal{L}_2$ be two operators defined by*

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

and $C_1,C_2\subset\Lambda_N$ with $\mathrm{dist}(C_1,C_2)>L$. *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

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

□

Lemma 5.9. *Let C_1 and C_2 be such that $C_i=C_i^o\cup C_i^\partial$, where $C_i^o:=\{x\in C_i\,|\,$*$\mathrm{dist}\,(x,(C_i)^c)>L\}$*, where A^c is the complement of set A. With further decomposition $C_i^o=C_i^{oo}+C_i^{o\partial}$ where $C_i^{oo}:=\{x\in C_i\,|\,$*$\mathrm{dist}\,(x,(C_i)^c)>2L\}$ *(see Figure 5.1). 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*

$$[\mathcal{L}_1,\mathcal{L}_2]=[\mathcal{L}_1^\partial,\mathcal{L}_2^\partial]$$

and

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

Proof. The proof of the first statement follows directly from Lemma 5.8 by observing that $\mathrm{dist}\,(C_1^o, C_2^o) = 2L$ and

$$\mathrm{dist}\,(C_1^o, C_2^\partial) = \mathrm{dist}(C_1^\partial, C_2^o) = L\,.$$

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

$$\begin{aligned}
[\mathcal{L}_1^o,\mathcal{L}_1^\partial\mathcal{L}_2^\partial] &= \mathcal{L}_1^o\mathcal{L}_1^\partial\mathcal{L}_2^\partial - \mathcal{L}_1^\partial\mathcal{L}_2^\partial\mathcal{L}_1^o = \mathcal{L}_1^o\mathcal{L}_1^\partial\mathcal{L}_2^\partial - \mathcal{L}_1^\partial\mathcal{L}_1^o\mathcal{L}_2^\partial \\
&= [\mathcal{L}_1^o,\mathcal{L}_1^\partial]\mathcal{L}_2^\partial = [\mathcal{L}_1^{oo} + \mathcal{L}_1^{o\partial},\mathcal{L}_1^\partial]\mathcal{L}_2^\partial \\
&= [\mathcal{L}_1^{o\partial},\mathcal{L}_1^\partial]\mathcal{L}_2^\partial\,,
\end{aligned}$$

where in the second equation we used the fact that $\mathcal{L}_2^\partial\mathcal{L}_1^o = \mathcal{L}_1^o\mathcal{L}_2^\partial$ and in the last equation $[\mathcal{L}_1^{oo},\mathcal{L}_1^\partial] = 0$. 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. □

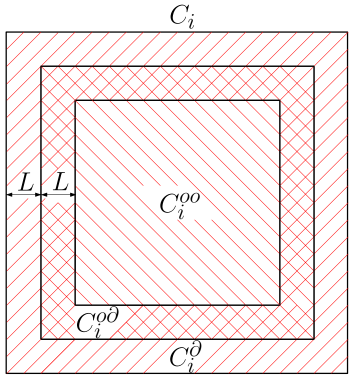

FIG. 5.1. *Sub-lattice partitioning. Note that we use the notation ∂C_i to denote $\bar{C} \setminus C$, see also Figure 3.1.*

The estimation of the commutator in Theorem 5.3 requires *local estimates* on the first and second discrete derivatives of the solution to the backward Kolmogorov equation by the discrete derivatives of the initial data

LEMMA 5.10. *The solution of the equation*

$$\partial_t u = \mathcal{L}u\,, \quad t \in (0,T]\,, \quad u(0,\sigma) = f(\sigma)\,, \tag{5.12}$$

satisfies the bounds

$$\max_{x\in\Lambda_N} \|\delta_x u(t,\cdot)\|_\infty \leq C \max_{x\in\Lambda_N} \|\delta_x u(0,\cdot)\|_\infty \tag{5.13}$$

and

$$\max_{x,y\in\Lambda_N} \|\,\delta_{xy}u(t,\cdot)\,\|_\infty \le C\big[\max_{x\in\Lambda_N}\|\delta_x u(0,\cdot)\|_\infty + \max_{x,y\in\Lambda_N}\|\,\delta_{xy}u(0,\cdot)\,\|_\infty\big]\,, \qquad (5.14)$$

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

Proof. Using (5.7) and Lemma 5.5, we have

$$\|\delta_x u(t,\cdot)\|_\infty \le \|\delta_x u(0,\cdot)\|_\infty + \mathcal{O}(1)\int_0^t \|\delta_x u(s,\cdot)\|_\infty\,ds \\ + \mathcal{O}(\frac{1}{L})\int_0^t \sum_{|x-y|\le L}\|\delta_y u(s,\cdot)\|_\infty\,ds\,. \qquad (5.15)$$

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

$$\|\delta_x u(t,\cdot)\|_\infty \le \gamma(0) + \mathcal{O}(1)\int_0^t\gamma(s)ds + \mathcal{O}(\frac{1}{L})L\int_0^t\gamma(s)\,ds\,,$$

or

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

Applying Gronwall's inequality we conclude the proof and obtain the bound

$$\gamma(t)\le e^{ct}\gamma(0)\,.$$

The inequality (5.14) follows similarly from (5.9) and from Lemma A.1. □

The commutator, as shown in Lemma 5.9, is a localized quantity that depends only on the boundary sites of the decomposed sub-lattices. Thus the localized estimate in Lemma 5.10 gives us a tool in order to reveal the scaling of the commutator when acting on macroscopic observables.

5.4. Proof of Theorem 5.3. By Lemma 5.9, the commutator can be written as $[\mathcal{L}_1^\partial,\mathcal{L}_2^\partial]$, which due to Lemma 5.8 is expanded to

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

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

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

Taking norms on both sides similarly to Lemma 5.4 and using the fact that the rates are bounded functions on $\Lambda_N\times\Sigma$,

$$\|[\mathcal{L}_1,\mathcal{L}_2]u(t,\cdot)\|_\infty \le C\sum_{\substack{x\in\Lambda_1^\partial,y\in\Lambda_2^\partial\\|x-y|\le L}}\|\delta_{xy}u(t,\cdot)\|_\infty + \|\delta_y u(t,\cdot)\|_\infty \le C\,, \qquad (5.16)$$

where the second inequality follows from Proposition 5.6, 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 (5.2). Many observables are in $C^2(\mathcal{S})$, but also satisfy the local bound (5.1) as one can see in Section 5.1. Under this assumption, we obtain from (5.16) the bound for the commutator

$$\begin{aligned}\|[\mathcal{L}_1,\mathcal{L}_2]u(t,\cdot)\|_\infty &\le C \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial\\ |x-y|\le L}} \|\delta_{xy}u(t,\cdot)\|_\infty + \|\delta_y u(t,\cdot)\|_\infty \\ &\le C\big[\max_{x,y\in\Lambda_N}\|\delta_{xy}u(0,\cdot)\|_\infty + \max_{y\in\Lambda_N}\|\delta_y u(0,\cdot)\|_\infty\big] \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial\\ |x-y|\le L}} 1\,,\end{aligned} \tag{5.17}$$

where the second inequality follows from Lemma 5.10. Using the fact that the initial data belong to $C^2(\mathcal{S})$ and satisfy (5.1), as well as that $|C_m^\partial| = c(d)Lq^{d-1}$, where d is the dimension, we deduce that

$$\|[\mathcal{L}_1,\mathcal{L}_2]u(t,\cdot)\|_\infty \le \frac{\tilde{C}}{N} \sum_{\substack{x\in\Lambda_1^\partial, y\in\Lambda_2^\partial\\ |x-y|\le L}} 1 \le \frac{\tilde{C}}{N} \times M \times c(d)Lq^{d-1} \times L^d = C\frac{L^{d+1}}{q}\,, \tag{5.18}$$

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 (5.3) follows along the same lines, noting that the the summation in (5.18) is now replaced by summations such as

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

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 (3.3) are applied and for how long. For instance, for the Lie scheme (3.5) 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 (3.5), (3.6) 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 (3.5) or in (3.6) the less processor communication we have, in which case the error in the approximation (3.5) or (3.6) worsens.

In both schemes (3.5), and (3.6), 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, [13]. Indeed, the sub-lattice parallelization algorithm for KMC, introduced in [23], is a particular example of a fractional step algorithm with stochastic PCS. In [23, 20] 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.

6.1. Randomized processor communication schedules. A generalization by Kurtz, [13], 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

$$\lim_{n\to\infty} \mathcal{T}_n(t)f = e^{t\bar{\mathcal{L}}}f\,, \quad \text{a.s.} \tag{6.1}$$

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

$$\bar{\mathcal{L}}f = \int \mathcal{L}_\xi f\, \mu(d\xi)\,. \tag{6.2}$$

While the random Trotter formula serves as a motivation for constructing schemes in which the evolution of the system, i.e., the process $\{\sigma_t\}_{t\geq 0}$, is *approximated* by a process obtained from a random composition of propagators $e^{\Delta t\mathcal{L}_k}$, the error analysis in the spirit of Theorem 5.3 requires more careful inspection of the approximating process $\{\gamma_{kh}\}_{k=0}^n$ on the interval $[0,T]$ with $T = n\Delta t$.

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 [23], is a particular example of a fractional step algorithm with stochastic PCS. In [23] each sub-lattice is selected at random, independently and advanced by KMC over a fixed time window $\Delta t = h$, subsequently a new random selection is made and again the sub-lattice is advanced by h, etc. This algorithm is easily recast as a fractional step approximation, where we can show that $\bar{\mathcal{L}} = \frac{1}{2}\left(\mathcal{L}_1 + \mathcal{L}_2\right)$ which is a time-rescaling of the original operator $\mathcal{L}$. From the numerical analysis viewpoint, our re-interpretation of the algorithm in [23] as (6.1) 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 4.2, we estimate the mean (weak) local error of the approximating γ-process.

Definition 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 $\gamma_0,\gamma_h,\dots,\gamma_{(k-1)h}$*

$$\mathbb{E}[f(\gamma_{kh})\,|\,\gamma_{(k-1)h}]:=P_{\xi_{2k-1}}(h)P_{\xi_{2k}}(h)f(\gamma_{(2k-1)h})\,, \tag{6.3}$$

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 6.2 (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 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 (2.6)

$$\partial_t u(\zeta,t)=\frac{1}{2}\mathcal{L}u(\zeta,t)\,,\qquad u(\zeta,0)=f(\zeta)\,. \tag{6.4}$$

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

$$\mathbb{E}^\gamma[f(\gamma_h)]-\mathbb{E}^\sigma[f(\sigma_h)]=\mathbb{E}^\xi\left[(P_{\xi_1}(h)P_{\xi_2}(h)-P(h))\,f(\gamma)\right]\,. \tag{6.5}$$

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

$$\begin{aligned}&[P_{\xi_1}(h)P_{\xi_2}(h)-P(h)]f=\\&h[\mathcal{L}_{\xi_1}+\mathcal{L}_{\xi_2}-\frac{1}{2}\mathcal{L}]f+\frac{h^2}{2}[\mathcal{L}_{\xi_1}^2+\mathcal{L}_{\xi_2}^2+2\mathcal{L}_{\xi_1}\mathcal{L}_{\xi_2}-\frac{1}{4}\mathcal{L}^2]f+\mathcal{O}(h^3)\,.\end{aligned} \tag{6.6}$$

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 6.1. For Bernoulli variables ξ_i with probabilities p this means that the γ-process approximates a process with the generator $p\mathcal{L}_1 + (1-p)\mathcal{L}_2$ instead of $\frac{1}{2}\mathcal{L} = \frac{1}{2}\mathcal{L}_1 + \frac{1}{2}\mathcal{L}_2$. This indicates that the usual order of the Lie splitting is achieved by properly weighing the time steps, i.e., applying $P_1(h_1)$ and $P_2(h_2)$ with different time steps h_1 and h_2 respectively. This calculation also shows that if we want to obtain the generator $\mathcal{L}$ instead of $\frac{1}{2}\mathcal{L}$ in Lemma 6.2, 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

$$\begin{aligned}\mathbb{E}^\gamma[f(\gamma_h)] - \mathbb{E}^\sigma[f(\sigma_h)] :&= \mathbb{E}^\xi\left[(P_{\xi_1}(2h)P_{\xi_2}(2h)f(\gamma) - u(\gamma,h))\right]\\ &= \frac{h^2}{2}\mathbb{E}^\xi\left[4\mathcal{L}_{\xi_1}^2 + 4\mathcal{L}_{\xi_2}^2 + 8\mathcal{L}_{\xi_1}\mathcal{L}_{\xi_2} - \mathcal{L}^2\right]f(\gamma) \\ &\quad + \mathcal{O}(h^3)\,,\end{aligned} \tag{6.7}$$

where $u(\gamma,h) = P(h)f(\gamma)$ is the solution of (2.6).

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 6.4, introduced earlier in [23]. 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 (4.6) and (4.8) 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 4.3 and 5.3 we have that the leading errors are governed by the commutators

$$\mathrm{TOL} \sim C_{\mathrm{Lie}}(T)\Delta t_{\mathrm{Lie}}\,, \qquad C_{\mathrm{Lie}}(T) = \max_{k=0,\ldots,n}\|\,[\mathcal{L}_1,\mathcal{L}_2]u(t_k)\,\|_\infty\,, \tag{6.8}$$

and

$$\begin{aligned}&\mathrm{TOL} \sim C_{\mathrm{Strang}}(T)\Delta t_{\mathrm{Strang}}^2\,,\\ &C_{\mathrm{Strang}}(T) = \max_{k=0,\ldots,n}\|\Big([\mathcal{L}_1,[\mathcal{L}_1,\mathcal{L}_2]] - 2[\mathcal{L}_2,[\mathcal{L}_2,\mathcal{L}_1]]\Big)u(t_k)\,\|_\infty\,,\end{aligned} \tag{6.9}$$

where $u = u(t)$ solves (2.6). Furthermore, due to (5.2) and (5.3) we have that

$$\mathrm{TOL} \sim \mathcal{O}\Big(\frac{L^{d+1}}{q}\Big)\Delta t_{\mathrm{Lie}}\,, \quad \mathrm{TOL} \sim \mathcal{O}(\frac{L^{2d+1}}{q}\Big)\Delta t_{\mathrm{Strang}}^2\,. \tag{6.10}$$

In the case of the randomized PCS the same reasoning as in Theorem 4.3 allows us to iterate the mean local error (6.7) to obtain

$$\begin{aligned} \mathrm{TOL} &\sim C_{\mathrm{Random}}(T)\Delta t_{\mathrm{Random}}\ , \\ C_{\mathrm{Random}}(T) &= \max_{k=0,\dots,n} \mathbb{E}^{\xi}\left[4\mathcal{L}_{\xi_1}^2 + 4\mathcal{L}_{\xi_2}^2 + 8\mathcal{L}_{\xi_1}\mathcal{L}_{\xi_2} - \mathcal{L}^2\right] u(t_k)\,, \end{aligned} \tag{6.11}$$

where $u = u(t)$ solves (2.6). 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 5 on the solution of (2.6) such as Lemma 5.6, 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

$$\mathrm{TOL} \sim \mathcal{O}(1)\Delta t_{\mathrm{Random}}\,. \tag{6.12}$$

In order to achieve the same error tolerance TOL, (6.10) and (6.12) imply the following relation between the respective time steps

$$\delta t_{\mathrm{SSA}} \ll \Delta t_{\mathrm{Random}} \sim \frac{L^{d+1}}{q}\Delta t_{\mathrm{Lie}} < \Delta t_{\mathrm{Lie}} \sim L^d \Delta t_{\mathrm{Strang}}^2 < \Delta t_{\mathrm{Strang}}\,. \tag{6.13}$$

Here q is the diameter of each of the cells C_k in Figure 3.1, and $\delta t_{\mathrm{SSA}} = \mathcal{O}(1/N)$ is the stochastic time step (the waiting time) of the SSA algorithm [7], which is exponentially distributed according to (2.2).

The relation (6.13) has several practical implications.

(i) 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 (2.6).

(ii) 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.

(iii) The Lie scheme (3.5) is expected to parallelize better than the randomized PCS in [23] 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 6.1.

(iv) 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 (5.3).

Example 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. 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 6.1. In Figure 6.1 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 $T = 4$.

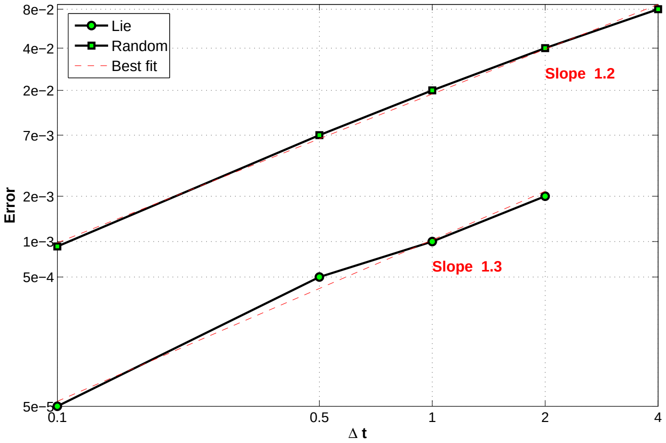

Fig. 6.1. *Convergence of the weak error for deterministic and randomized Lie splitting.*

Example 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 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 6.2 we can 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.

7. The infinite volume limit.

In this paper we considered interacting particle systems defined on a d-dimensional lattice Λ_N, as the numerical analysis and simulations for the parallel fractional step Kinetic Monte Carlo are performed on a finite lattice of size N. However, given the size of real molecular systems it is necessary that numerical estimates are independent of the system size N as we showed in Section 5. Alternatively we can consider the case $N \to \infty$, e.g., by setting up our analysis on the infinite lattice $\Lambda = \mathbb{Z}^d$. We outline the latter approach here for completeness of our

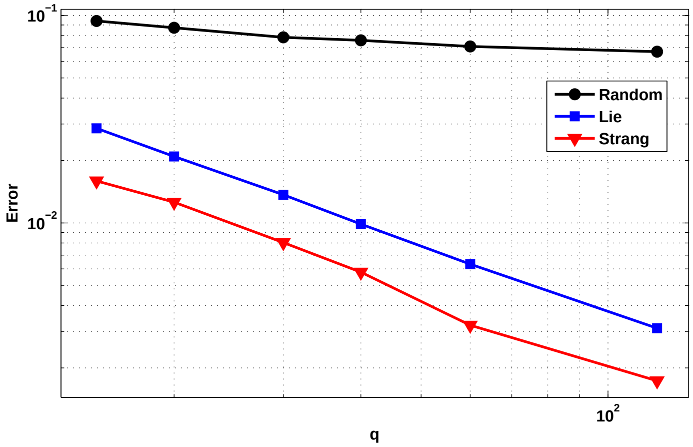

FIG. 6.2. *Dependence of the weak error on the sub-lattice size parameter q, see also (6.13).*

analysis. We refer to [15] for a comprehensive study of interacting particle systems set on infinite lattices.

First, we consider the configuration space $\mathcal{S} = \Sigma^{\Lambda}$, where $\Lambda = \mathbb{Z}^d$ and the space of bounded continuous functions $C_b(\mathcal{S})$. Then the generator (2.7) is defined on a suitable domain $\mathcal{D}(\mathcal{L})$,

$$\mathcal{L} : D(\mathcal{L}) \subset C_b(\mathcal{S}) \mapsto C_b(\mathcal{S})\,.$$

In this case, Theorem 4.3 is restated similarly to Theorem 3 in [10], provided the solution $u = u(t)$ of (2.6) satisfies $u(t_k) \in \mathcal{D}(\mathcal{L}_1^{m_1}\mathcal{L}_2^{m_2})$ for $|m| \leq 3$ and $k = 0, \ldots, n$. As it was also pointed out in [10] this is in principle an uncheckable hypothesis. However, this is not the case here: due to the results of Section 5 we have that if $f \in C^2(\mathcal{S})$ is a macroscopic observable, where

$$C^m(\mathcal{S}) := \{f \in C_b(\mathcal{S}) \,|\, \sum_{k=1}^{m} \|f\|_k < \infty\}\,, \forall m \in \mathbb{N}\,,$$

then $u(t) \in \mathcal{D}(\mathcal{L}_1^{m_1}\mathcal{L}_2^{m_2})$ due to Theorem 5.3 and Remark 5.1. Therefore, all estimates of Section 5 hold also true in the infinite lattice Λ, which is certainly not unexpected since all previous results in Λ_N were independent of the system size N.

8. Conclusions.

In this paper, we derived numerical error estimates for the Fractional Step Kinetic Monte Carlo (FS-KMC) algorithms proposed in [1] for the parallel simulation of interacting particle systems on a lattice. These algorithms have the capacity to simulate a wide range of spatio-temporal scales of spatially distributed, non-equilibrium physiochemical processes with complex chemistry and transport micro-mechanisms, while they can be tailored to specific hierarchical parallel architectures such as clusters of Graphical Processing Units. A key aspect of our approach relies on emphasizing a *goal-oriented* error analysis for macroscopic observables (e.g., density, energy, correlations, surface roughness), rather than focusing on strong topology estimates for individual trajectories or estimating probability distributions solving the Master Equation (Forward Kolmogorov Equation). Our analysis also addresses earlier work on parallel KMC algorithms [23, 20] that fit into the FS-KMC framework. Furthermore, moving beyond the parallelization problems discussed here, it appears that these methodologies, introduced in Section 5, can be generally useful in the development and study of numerical approximations of molecular and other extended systems. Our error analysis allows us to address systematically the processor communication of different parallelization strategies for KMC by comparing their (partial) asynchrony, which in turn is measured by their respective fractional step time-step for a prescribed error tolerance.

REFERENCES

[1] G. Arampatzis, M. A. Katsoulakis, P. Plecháč, M. Taufer, and L. Xu, *Hierarchical fractional-step approximations and parallel kinetic Monte Carlo algorithms*, J. Comp. Phys., (2012), `doi: 10.1016/j.jcp.2012.07.017`.

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

[3] 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 (1975), pp. 10–18.

[4] A. Chatterjee and D. G. Vlachos, *An overview of spatial microscopic and accelerated kinetic Monte Carlo methods*, J. Comput.-Aided Mater. Design, 14 (2007), pp. 253–308.

[5] 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 (1993), pp. 287–314.

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

[7] D. T. Gillespie, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, J. of Comp. Phys., 22 (1976), pp. 403–434.

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

[9] P. Heidelberger and D. M. Nicol, *Conservative parallel simulation of continuous time markov chains using uniformization*, IEEE Trans. Parallel Distrib. Syst., 4 (1993), pp. 906–921.

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

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

[12] 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 (1999), pp. 488–508.

[13] T. G. Kurtz, *A random Trotter product formula*, Proc. Amer. Math. Soc., 35 (1972), pp. 147–154.

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

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

[16] 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 (2009), pp. 1706–1716.

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

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

[19] 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 (2009), p. 084214.

[20] 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*, Tech. Report SAND2009-6226, Sandia

National Laboratory, 2009.

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

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

[23] ———, *Semirigorous synchronous relaxation algorithm for parallel kinetic Monte Carlo simulations of thin film growth*, Phys. Rev. B, 71 (2005), p. 125432.

[24] G. Szabo and G. Fath, *Evolutionary games on graphs*, Physics Reports, 446 (2007), pp. 97–216.

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

APPENDIX

Appendix A. 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 5.10. We prove it in the presence of two equations, but the result can be easily generalized for a system of equations.

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

$$\varphi(t) \le \varphi(0) + \int_0^t \varphi(s)\,ds$$
$$\vartheta(t) \le \vartheta(0) + \int_0^t \vartheta(s)\,ds + \int_0^t \varphi(s)\,ds$$

then

$$\varphi(t) \le e^t \varphi(0) \tag{A.1}$$
$$\vartheta(t) \le e^t \vartheta(0) + (e^t + te^t - 1)\varphi(0) \tag{A.2}$$

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

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

and by substituting this to the second inequality we obtain

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

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

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

and after straightforward calculations

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

□

Remark A.1. Let $\Phi(t) = (\varphi_1(t), \dots, \varphi_n(t))$ satisfying

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

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

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

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