Parallel Kinetic Monte Carlo for Crystal Surface Diffusion

Anya Katsevich and Terrence Alsup

May 20, 2019

Crystal Surface Jump Process

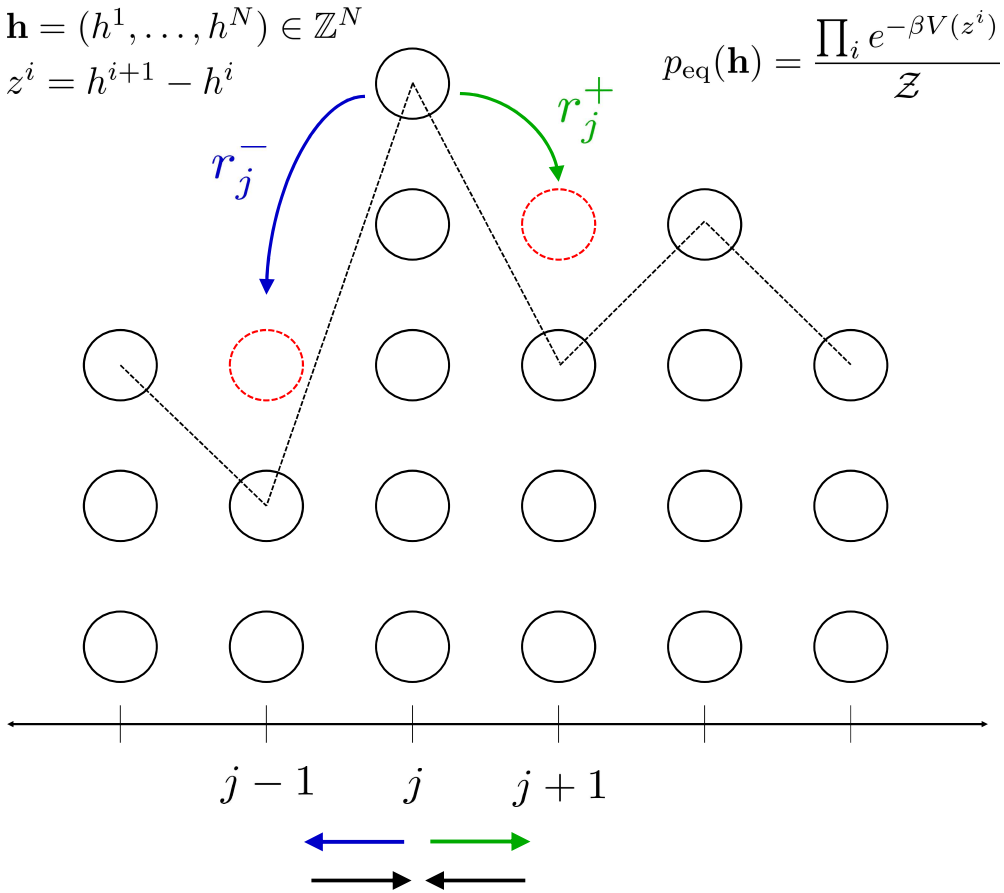


Figure: Surface diffusion on 1D lattice; an event occurs at site j when the topmost atom jumps left or right, at a rate of $r_j^{\pm}$.

Scaling Limit of Process

Re-scale the height, domain, and time as

$$h(x,t) = \frac{1}{L} h_{\lfloor Lx \rfloor}(L^4 t).$$

As $L \to \infty$ the continuous height profile $h(x,t)$ will satisfy the following PDE describing the evolution of h:

$$\partial_t h(t,x) = -\frac{K}{2} \partial_{xxx} \left[\sigma_F \left(\partial_x h(t,x) \right) \right],$$

where σ_F is the free energy of the surface [1].

Serial KMC Description

Input: Initial lattice state $\vec{h}$ and time $t = 0$

While $t < T$ do

1. For the given state $\vec{h}$, compute the current values of all rates, r_i, of events E_i.
2. Draw a waiting time $\Delta t \sim \mathrm{Exp}(R)$ for the next event to occur $(R = \sum_i r_i)$ and update the time $t \mapsto t + \Delta t$.
3. Draw an event E from the discrete distribution $\mathbb{P}(E_i) = \frac{r_i}{R}$ and update the state.

Parallel KMC Description

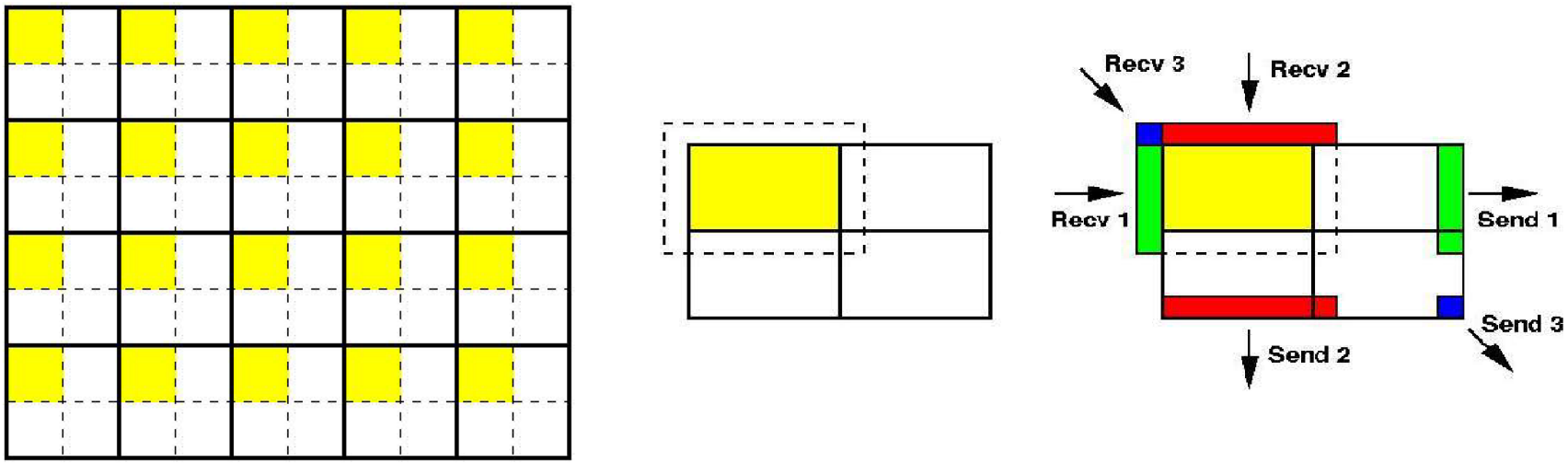


Figure: A processor works on the yellow section of its block, then sends and receives boundary information from other processors before moving on to work on the next section [2]

Parallel KMC Description

MPI Implementation of [2], [3]

Input: current physical time t, current values of $R_{\max}^{0,1}$, t_{stop}, and current crystal state $\vec{h}_k(t)$ known to processor k.

1. Run **Serial KMC** until time $t = t + t_{\text{stop}}$ in the left section of the block (updating rates after each event).
2. Send left bdy info and receive right bdy info (`MPI_Send`, `MPI_Recv`)
3. Compute the new value of $R_{\max}^{1}$ (using `MPI_Allreduce`) and t_{stop}.
4. Repeat Steps 1-3 for the right section.

Performance: Accuracy

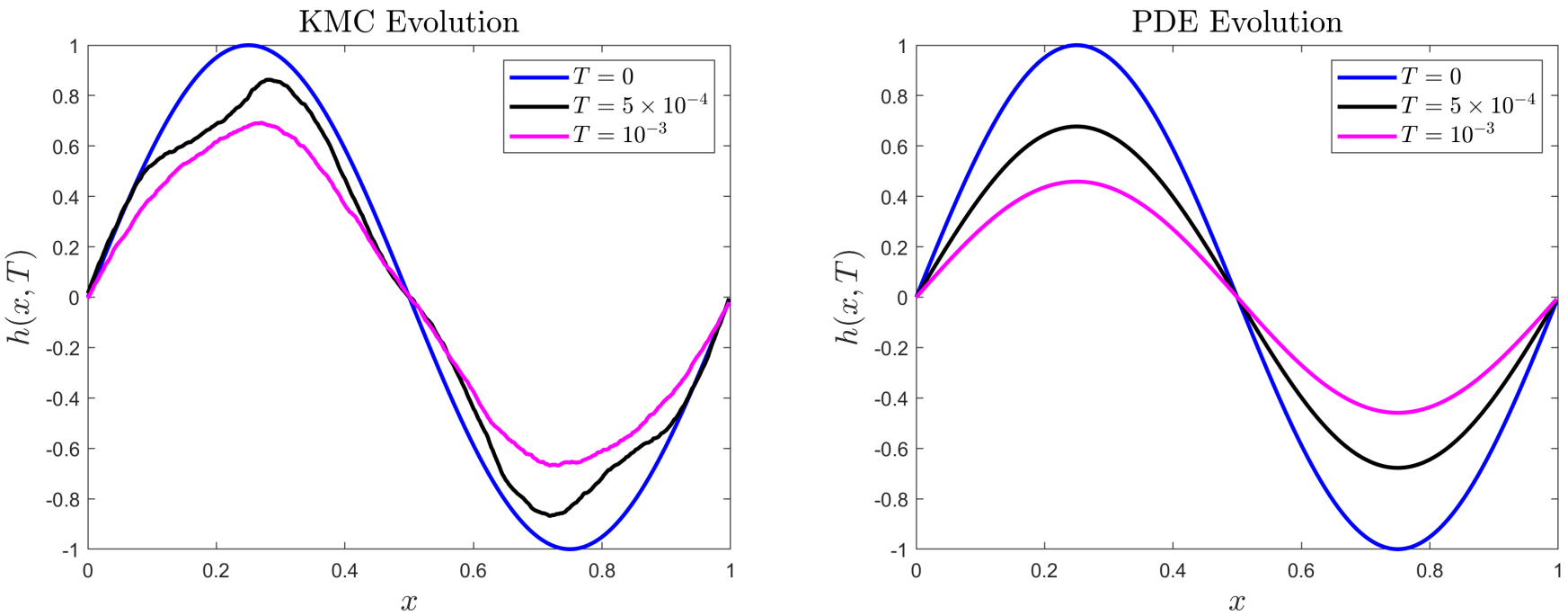


Figure: Evolution of crystal surface simulated with parallel KMC simulation (left) compared to evolution of PDE solution (right) for initial condition $h(x) = \sin(2\pi x)$.

Performance: Accuracy

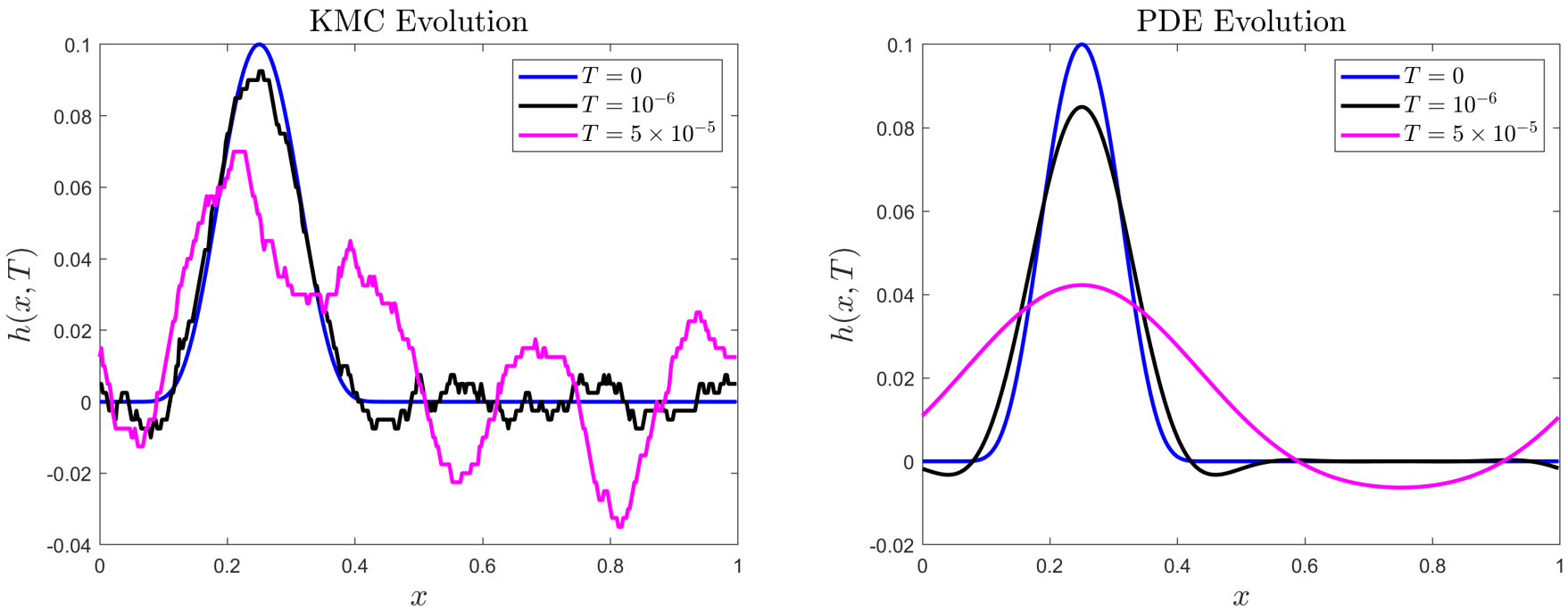


Figure: Evolution of crystal surface simulated with parallel KMC simulation (left) compared to evolution of PDE solution (right) for initial condition with compact support.

Performance: Theoretical Analysis

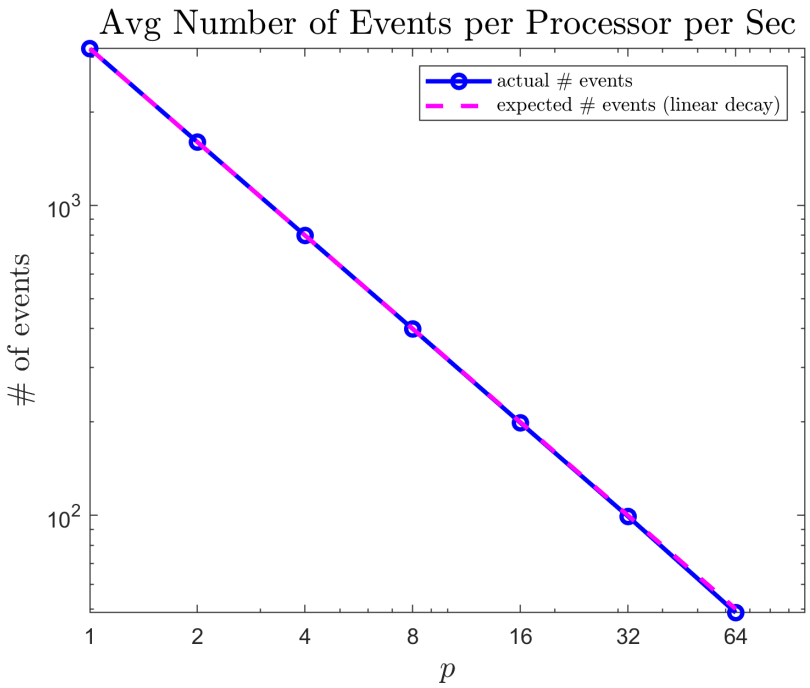
Strong Scaling

	Complexity
1 processor:	
Computational effort / event	$O(L)$
Total computational effort	$O(L) \times$ # Events.
P processors:	
# Events/ processor	$\frac{1}{P} \times$ # Events
Computational effort / event	$O(L/P)$
Total computational effort / processor	$\frac{1}{P^2} \times O(L) \times$ # Events

Table

Weak Scaling: Poor because of $T \sim P^4$ scaling.

Performance: Experiments



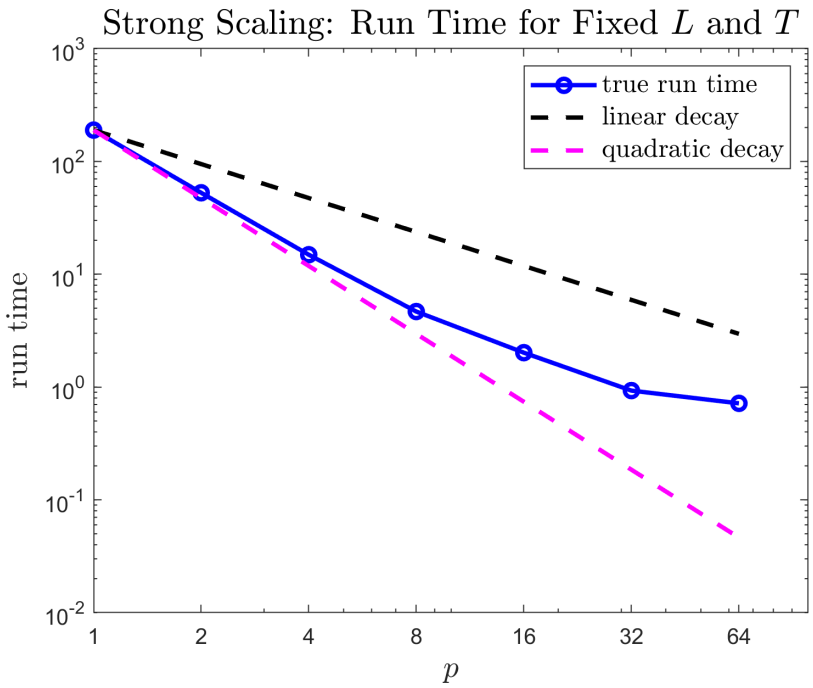


Figure: Strong scaling algorithm performance (left), and number of events per processor per second (right). and. Note that the run time decreases faster than linearly as the number of processors increases. This is because both the number of events and the computational effort per event decreases linearly as the number of processors increases.

Conclusion

- Serial KMC has same behavior as PDE only for large L, and parallel algorithm is approximation to serial one.
- No known error bounds for parallel approximation; unclear if parallel simulation reliable for small L.
- Good strong scaling (P^2 speedup) but poor weak scaling.


J. Marzuola and J. Weare. *The relaxation of a family of broken bond crystal surface models.* Phys Rev E 88 (2013), 032403.


S. Plimpton, C. Battaile, et.al. *Crossing the Mesoscale No-Man's Land via Parallel Kinetic Monte Carlo*, SAND20096226 Unlimited Release, October 2009


I. Martin-Bragado, J. Abujas, P.L. Galindo, J. Pizarro *Synchronous parallel Kinetic Monte Carlo: Implementation and results for object and lattice approaches*, Nuclear Instruments and Methods in Physics Research B, 352 (2015) 2730