Fast, Hierarchical, and Adaptive Algorithm for Metropolis Monte Carlo Simulations of Long-Range Interacting Systems
MCS-Janke Algorithm ⌣̈

Fabio Müller[1], Henrik Christiansen[2], Stefan Schnabel[1], and Wolfhard Janke[1]

[1]Leipzig University [2]NEC Laboratories Europe, Germany

CompPhys 23
19 December, 2023

Why Monte Carlo Simulations of Long-Range Systems?

- Long-range (LR) pair potentials ubiquitous in nature

$$H_{LR} = \sum_{i}^{N} \sum_{j<i}^{N} V(\mathbf{r}_{i,j}), \qquad (\propto N^2 \text{ terms})$$

- Many real-world potentials (Coulomb, dipole-dipole, Lennard-Jones, etc.) are power-law:

$$V(\mathbf{r}_{i,j}) \propto r^{-\alpha}$$

- Energy difference ΔE associated with update of single component requires summation of $\sim N$ terms

  $\Rightarrow$ Large system sizes prohibitive due to $O(N^2)$ scaling

- Physics of true LR systems[1] not recovered with cut-offs[2].

$\Rightarrow$ Aiming at a versatile and fast algorithm for **Metropolis** simulation of general LR interacting systems

[1] H. Christiansen, S. Majumder, and W. Janke, Phys. Rev. E **99**, 011301(R) (2019).
[2] J. Gundh, A. Singh, and R. K. B. Singh, PloS One **10**, e0141463 (2015)

Established Approaches for (approximate) Energy Calculation

- Ewald summation[3]:
  - Split sum into **r**- and **k**-space
  - Short-range contributions converge fast in **r**
  - LR part converges fast in **k**

  $\Rightarrow$ $O(N^{3/2})$ scaling, PBC only
- Barnes-Hut method[4]:
  - Subdivide simulation box in quad-tree
  - Interactions with distant particles in boxes treated collectively
  - Boxes grow with distance

  $\Rightarrow$ $O(N \log N)$ scaling, hard to control errors

Both methods rarely adopted in Monte Carlo (MC). Existing nonequil. MC studies mostly based on toy models or smart mappings.

[3] J. W. Perram, H. G. Petersen, and S. W. De Leeuw, Mol. Phys. **65**, 875 (1988).

[4] J. Barnes and P. Hut, Nature (London) **324**, 446 (1986).

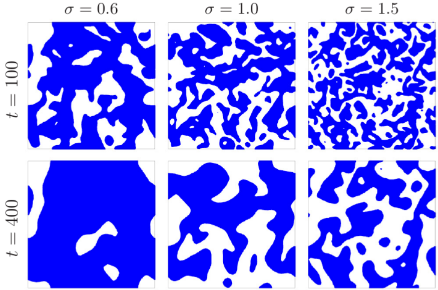
Nonconserved Order Parameter:

- Could be investigated at low T_q with effective field trick.[1]:

$$\Delta E = \sum_{j \neq i} s_i^{\mathrm{old}} J_{ij} s_j - \sum_{j \neq i} s_i^{\mathrm{new}} J_{ij} s_j = -\Delta s_i F_i,$$

$$\text{where } \Delta s_i = s_i^{\mathrm{new}} - s_i^{\mathrm{old}} \text{ and } F_i \equiv \sum_{j \neq i} J_{ij} s_j$$

- F_j updated only after accepted update of s_i
- A fraction $P_{\mathrm{acc}} \approx 10^{-3}$ of computational effort needed
- Domains can move without the need of thermal excitations



[1] H. Christiansen, S. Majumder, and W. Janke, Phys. Rev. E **99**, 011301(R) (2019).

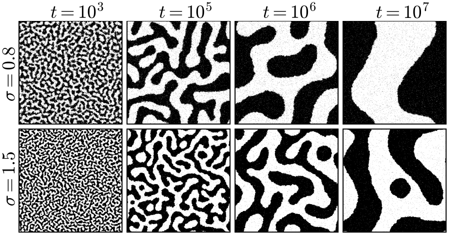
Conserved Order Parameter (Phase Separation):[5]

- Spins behave like particles (or holes) which can hop only onto empty positions
- Jumps need activation energy

$\Rightarrow$ At low T_q frozen dynamics

- For viable T_q rather large acceptance rates
- $P_{\mathrm{acc}} > 0.5$ for $T_q = 0.5\,T_c$

$\Rightarrow$ Effective field not efficient



[5] F. Müller, H. Christiansen, and W. Janke, Phys. Rev. Lett. **129**, 240601 (2022).

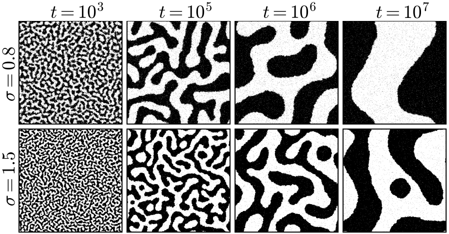
Conserved Order Parameter (Phase Separation):[5]

- Spins behave like particles (or holes) which can hop only onto empty positions
- Jumps need activation energy

$\Rightarrow$ At low T_q frozen dynamics

- For viable T_q rather large acceptance rates
- $P_{\mathrm{acc}} > 0.5$ for $T_q = 0.5\,T_c$

$\Rightarrow$ Effective field not efficient



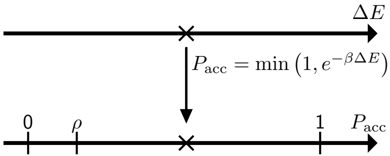
We need a new algorithm!

[5]F. Müller, H. Christiansen, and W. Janke, Phys. Rev. Lett. **129**, 240601 (2022).

Inverted Metropolis Criterion[6]

Traditional Metropolis:

1. Propose update
2. Calculate ΔE
3. $P_{\mathrm{acc}} = \min\left(1, e^{-\beta \Delta E}\right)$
4. Draw random number $\rho \in [0, 1)$
5. Accept if $\rho \leq P_{\mathrm{acc}}$



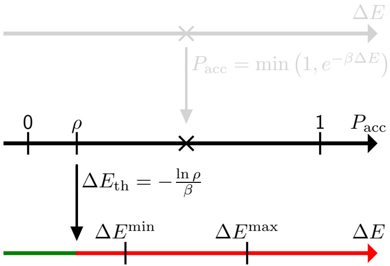
[6] S. Schnabel and W. Janke, Comput. Phys. Commun. **256**,107414 (2020).

Inverted Metropolis Criterion[6]

Inverted Metropolis:

1. Propose update
2. calculate ΔE
3. $P_{\mathrm{acc}} = \min\left(1, e^{-\beta \Delta E}\right)$
2. Draw random number $\rho \in [0, 1)$
5. Accept if $\rho \leq P_{\mathrm{acc}}$
3. Calculate $\Delta E_{\mathrm{th}} = -\frac{\ln \rho}{\beta}$
4. Decide upon strict lower and upper bounds $\Delta E^{\mathrm{min/max}}$

   Accept if $\Delta E^{\mathrm{max}} < \Delta E_{\mathrm{th}}$ or reject if $\Delta E^{\mathrm{min}} > \Delta E_{\mathrm{th}}$



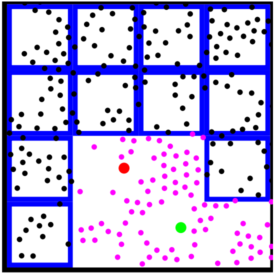
[6] S. Schnabel and W. Janke, Comput. Phys. Commun. **256**,107414 (2020).

Example Decision

- Relies on a decomposition $\mathcal{D}$ of simulation domain into boxes
- Energy difference due to pink particles is calculated exactly as pair interactions

$$\Delta E_B = \sum_{j \in B} V_{ij}$$

- Majority of interactions (with black particles) bounded as particle-box interactions $\Delta E_B^{\mathrm{min/max}}$
- Total energy difference give by $\Delta E^{\mathrm{min/max}} = \sum_{B \in \mathcal{D}} \Delta E_B^{\mathrm{min/max}}$

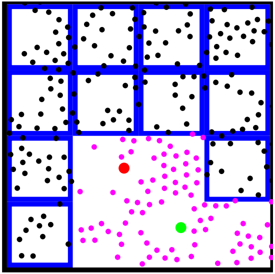


Example Decision

- Relies on a decomposition $\mathcal{D}$ of simulation domain into boxes
- Energy difference due to pink particles is calculated exactly as pair interactions

$$\Delta E_B = \sum_{j \in B} V_{ij}$$

- Majority of interactions (with black particles) bounded as particle-box interactions $\Delta E_B^{\mathrm{min/max}}$
- Total energy difference give by $\Delta E^{\mathrm{min/max}} = \sum_{B \in \mathcal{D}} \Delta E_B^{\mathrm{min/max}}$



How do we get there?

Bounds for component-box interaction for LJ system

In general for a particle-box interaction:

$$\Delta E_B = \sum_{j \in B} V_{ij}^{\mathrm{new}} - \sum_{j \in B} V_{ij}^{\mathrm{old}}$$

Lower bound:

$$\Delta E_B^{\mathrm{min}} \equiv N_B \left(V_{ij}^{\mathrm{new,min}} - V_{ij}^{\mathrm{old,max}} \right) < \Delta E_B$$

Upper bound:

$$\Delta E_B^{\mathrm{max}} \equiv N_B \left(V_{ij}^{\mathrm{new,max}} - V_{ij}^{\mathrm{old,min}} \right) > \Delta E_B$$

The bounds come along with an uncertainty of the particle-box interaction:

$$\Delta_B \equiv \Delta E_B^{\mathrm{max}} - \Delta E_B^{\mathrm{min}}$$

The integer part of the logarithmic uncertainty can be take to store spin-box interaction in linear data structure

$$\lfloor \log \Delta_B \rfloor$$

Bounds for component-box interaction for LJ system

- Potential very short-ranged

$$V_{\mathrm{LJ}}(r) = 4\epsilon\left[\left(\frac{r}{\sigma}\right)^{-12} - \left(\frac{r}{\sigma}\right)^{-6}\right]$$

- Typical cutoff $r_c = 2.5\sigma$
- Still discussion about artifacts due to truncation

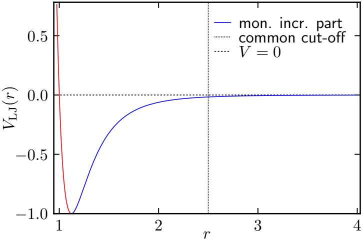
⇒ Particle-mesh Ewald for LJ available

- Only treat mon. growing part (blue) collectively

⇒

$$V_{ij}^{\max/\min} = V_{\mathrm{LJ}}\left(r_{ij}^{\max/\min}\right)$$

⇒ Distances $r_{ij}^{\max/\min}$ can easily be calculated in constant time

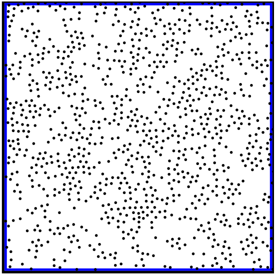


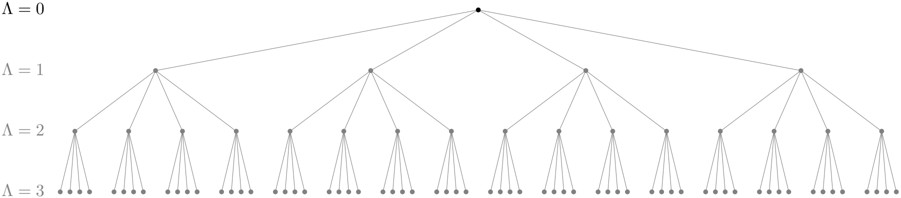
The Hierarchical Decomposition

- $\mathcal{T}$: complete quad-tree of depth $m + 1$ which persists over whole simulation
- Each level Λ contains 4^{Λ} nodes

The Hierarchical Decomposition

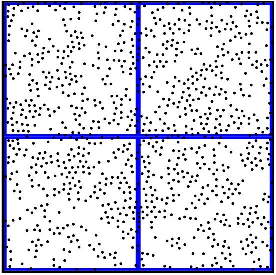
- $\mathcal{T}$: complete quad-tree of depth $m+1$ which persists over whole simulation
- Each level Λ contains 4^Λ nodes

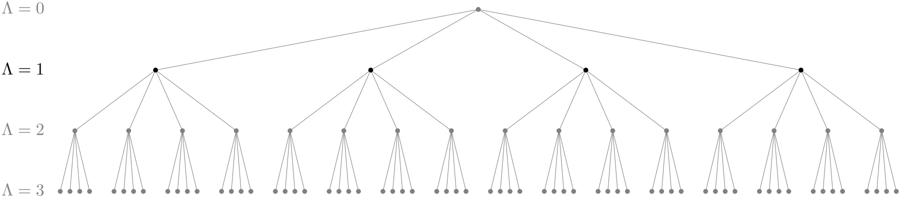




The Hierarchical Decomposition

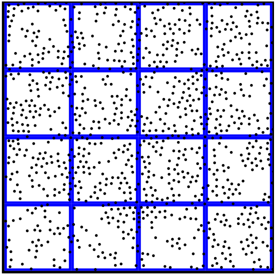
- $\mathcal{T}$: complete quad-tree of depth $m + 1$ which persists over whole simulation
- Each level Λ contains 4^{Λ} nodes

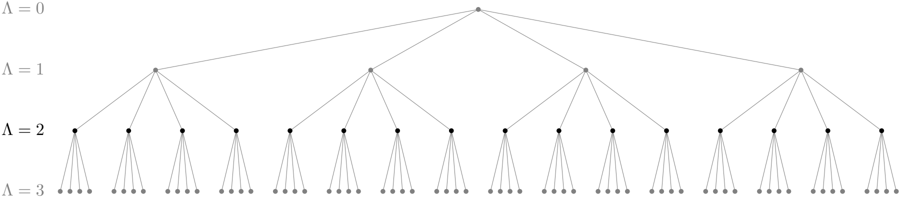




The Hierarchical Decomposition

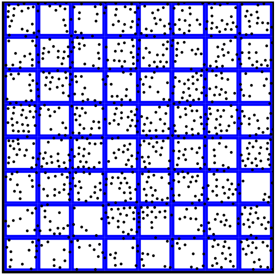
- $\mathcal{T}$: complete quad-tree of depth $m+1$ which persists over whole simulation
- Each level Λ contains 4^{Λ} nodes

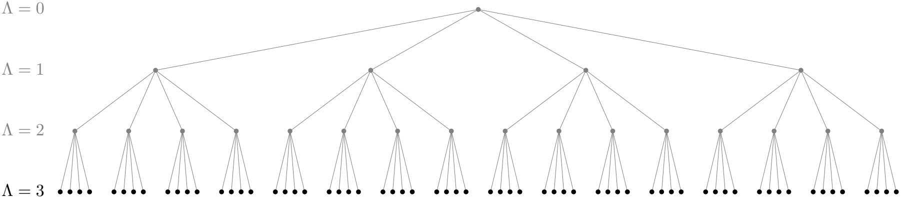




The Hierarchical Decomposition

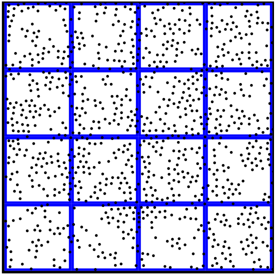
- $\mathcal{T}$: complete quad-tree of depth $m+1$ which persists over whole simulation
- Each level Λ contains 4^{Λ} nodes

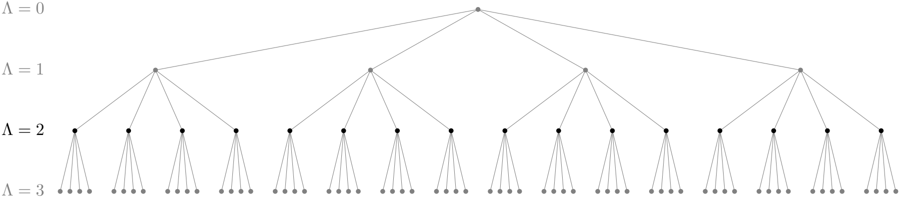




The Hierarchical Decomposition

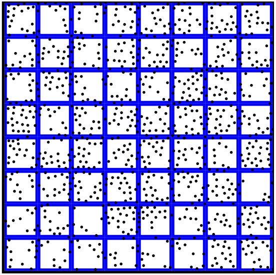
- $\mathcal{T}$: complete quad-tree of depth $m+1$ which persists over whole simulation
- Each level Λ contains 4^Λ nodes
- Inner nodes contain collective information for calculating bounds

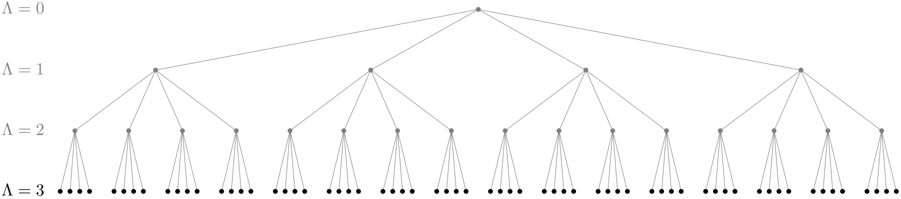




The Hierarchical Decomposition

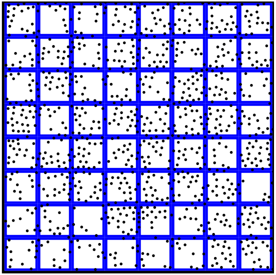
- $\mathcal{T}$: complete quad-tree of depth $m + 1$ which persists over whole simulation
- Each level Λ contains 4^{Λ} nodes
- Inner nodes contain collective information for calculating bounds
- Leafs, additionally, pointers to contained components

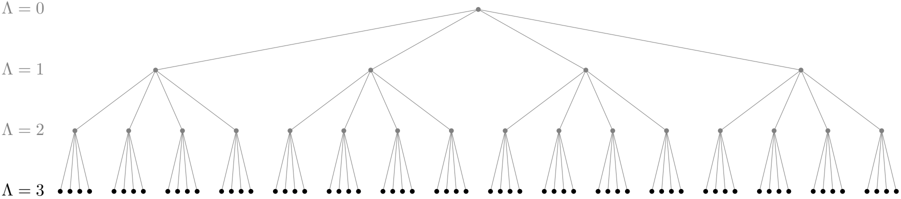




The Hierarchical Decomposition

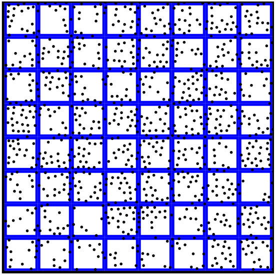
- $\mathcal{T}$: complete quad-tree of depth $m + 1$ which persists over whole simulation
- Each level Λ contains 4^Λ nodes
- Inner nodes contain collective information for calculating bounds
- Leafs, additionally, pointers to contained components
- Updating one component $\propto \log N$ operations

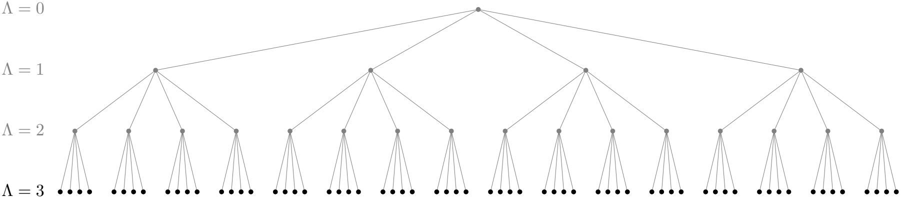




The Hierarchical Decomposition

- $\mathcal{T}$: complete quad-tree of depth $m+1$ which persists over whole simulation
- Each level Λ contains 4^{Λ} nodes
- Inner nodes contain collective information for calculating bounds
- Leafs, additionally, pointers to contained components
- Updating one component $\propto \log N$ operations



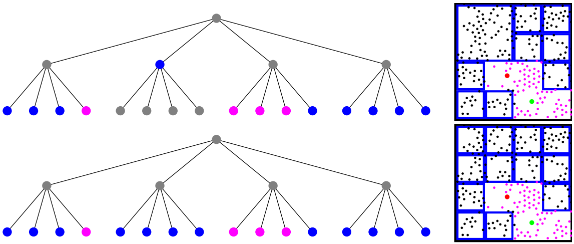


Alternative: incomplete (pointer based) tree

Adaptive refinement of the decomposition

- The decomposition can be refined by

i) Splitting inner nodes into their child nodes

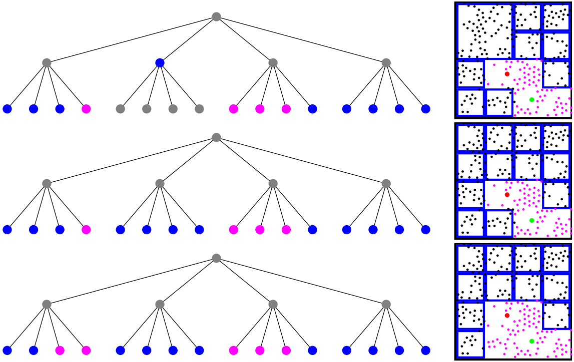


Adaptive refinement of the decomposition

- The decomposition can be refined by

i) Splitting inner nodes into their child nodes

ii) Replacing the energy bounds of a leaf by the exact sum over all contained particles



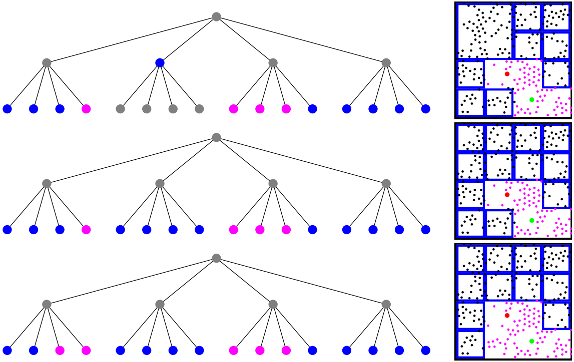
Adaptive refinement of the decomposition

- The decomposition can be refined by

i) Splitting inner nodes into their child nodes

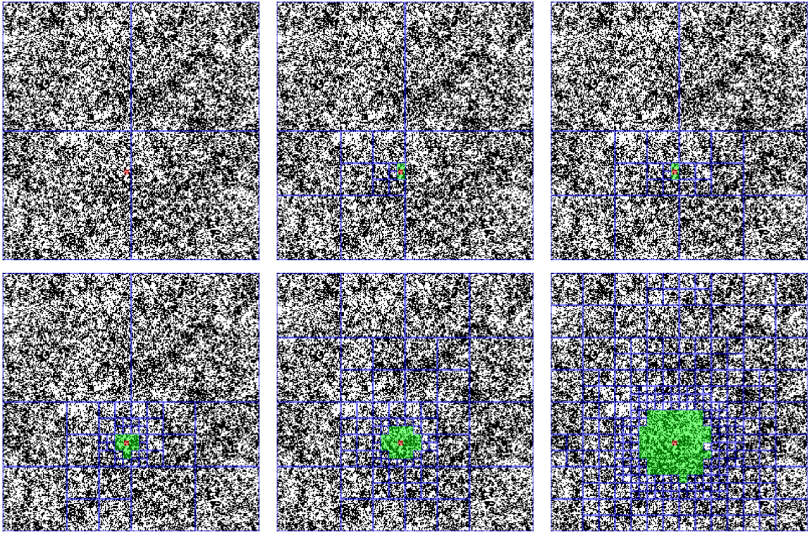
ii) Replacing the energy bounds of a leaf by the exact sum over all contained particles

- Nodes with relatively large uncertainty Δ_B picked first



Adaptive refinement of the decomposition

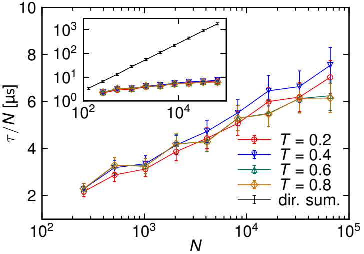
Larger example:

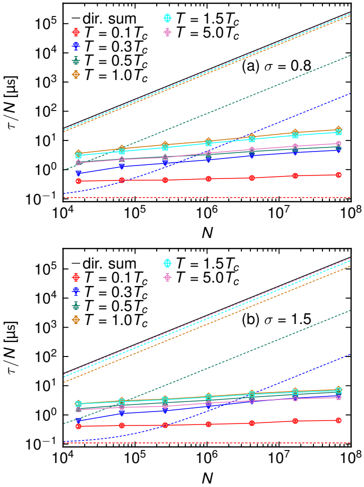


- Runtime scaling compatible with $N \log N$ for both LRIM and LJ-system

$\Rightarrow$ Works for both lattice and off-lattice

- Exponent of potential matters for performance

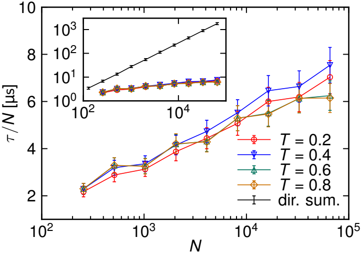


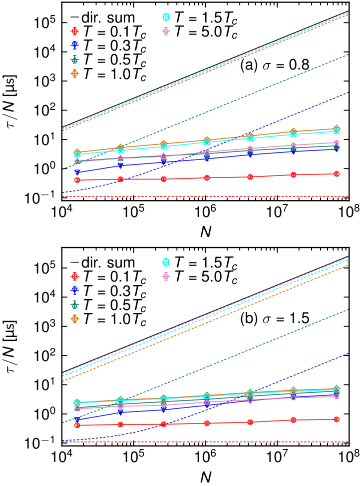


- Runtime scaling compatible with $N \log N$ for both LRIM and LJ-system

$\Rightarrow$ Works for both lattice and off-lattice

- Exponent of potential matters for performance





"Break even point" reached at small system sizes for realistic temperatures.

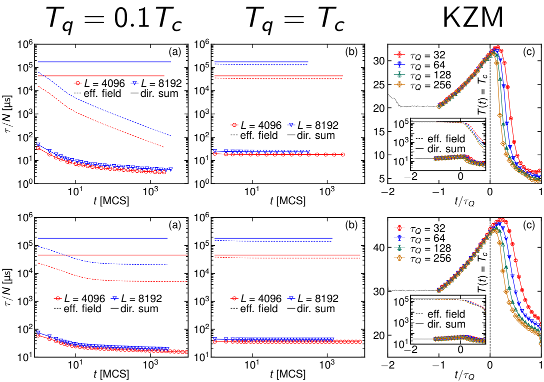
Fast in Nonequilibrium for O(n) models

For the LR Ising model ($n = 1$, upper row) and the LR XY model ($n = 2$, lower row):

(a) Quenches from random to $T_q = 0.1\,T_c$

(b) Quenches from random to $T_q = T_c$

(c) A Kibble-Zurek protocol with $T(t) = t/\tau_Q$



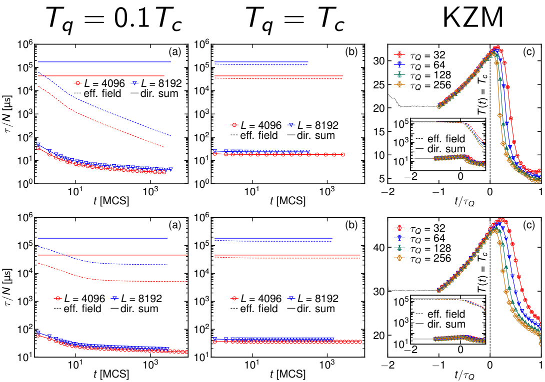
Fast in Nonequilibrium for $\mathrm{O}(n)$ models

For the LR Ising model ($n = 1$, upper row) and the LR XY model ($n = 2$, lower row):

(a) Quenches from random to $T_q = 0.1\,T_c$

(b) Quenches from random to $T_q = T_c$

(c) A Kibble-Zurek protocol with $T(t) = t/\tau_Q$



For these lattice sizes which are realistic in the investigation of the nonequilibrium behavior of LR $\mathrm{O}(n)$ models we find up to more than 10000-fold speedups.

Conclusion

- New algorithm for **general** ensembles of LR interacting particles

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$
- Very small prefactors (measured speedup of $\approx \mathbf{10000}$)

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$
- Very small prefactors (measured speedup of $\approx$ **10000**)
- Serves as **plug-in replacement** for the exact energy calculation

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$
- Very small prefactors (measured speedup of $\approx$ **10000**)
- Serves as **plug-in replacement** for the exact energy calculation
- Reproduces (with same randomness) exactly same trajectories

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$
- Very small prefactors (measured speedup of $\approx$ **10000**)
- Serves as **plug-in replacement** for the exact energy calculation
- Reproduces (with same randomness) exactly same trajectories

  $\Rightarrow$ Very helpful for debugging ;-)

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$
- Very small prefactors (measured speedup of $\approx$ **10000**)
- Serves as **plug-in replacement** for the exact energy calculation
- Reproduces (with same randomness) exactly same trajectories

  $\Rightarrow$ Very helpful for debugging ;-)

  $\Rightarrow$ All knowledge and experience can be **carried over**!

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$
- Very small prefactors (measured speedup of $\approx$ **10000**)
- Serves as **plug-in replacement** for the exact energy calculation
- Reproduces (with same randomness) exactly same trajectories

  $\Rightarrow$ Very helpful for debugging ;-)

  $\Rightarrow$ All knowledge and experience can be **carried over**!

  $\Rightarrow$ Especially useful for **nonequilibrium** studies

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$
- Very small prefactors (measured speedup of $\approx$ **10000**)
- Serves as **plug-in replacement** for the exact energy calculation
- Reproduces (with same randomness) exactly same trajectories

  $\Rightarrow$ Very helpful for debugging ;-)

  $\Rightarrow$ All knowledge and experience can be **carried over**!

  $\Rightarrow$ Especially useful for **nonequilibrium** studies
- Simulate in few days what would have taken centuries before

Conclusion

- New algorithm for **general** ensembles of LR interacting particles
- Improved algorithmic scaling compatible with $N \log N$
- Very small prefactors (measured speedup of $\approx$ **10000**)
- Serves as **plug-in replacement** for the exact energy calculation
- Reproduces (with same randomness) exactly same trajectories

  $\Rightarrow$ Very helpful for debugging ;-)

  $\Rightarrow$ All knowledge and experience can be **carried over**!

  $\Rightarrow$ Especially useful for **nonequilibrium** studies
- Simulate in few days what would have taken centuries before

Thank you for your attention!