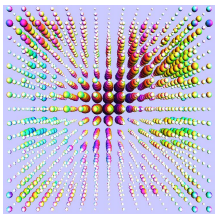
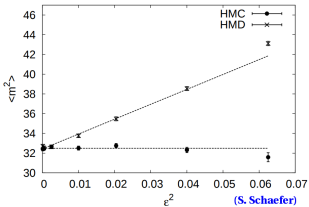
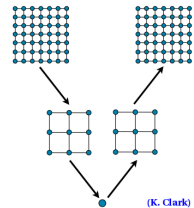
Lattice Field Theory Algorithms — Part 1

David Schaich (University of Liverpool)







Methods of Effective Field Theory and Lattice Field Theory

Bad Honnef Physics School, 14 July 2021

david.schaich@liverpool.ac.uk www.davidschaich.net

Overview and plan

The capabilities and reliability of lattice calculations
depend on the algorithms available

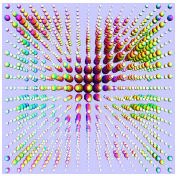
Importance sampling and Markov chains

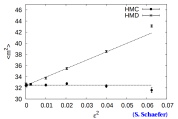
Hybrid Monte Carlo and friends

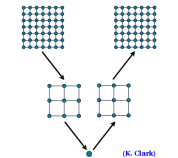
Iterative algorithms for fermions

[Time permitting] Alternatives to address sign problems

Conceptual focus, from big ideas to more detailed aspects
that can be practiced through exercises

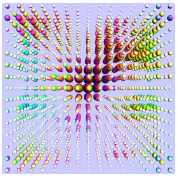






Overview and plan

The capabilities and reliability of lattice calculations depend on the algorithms available

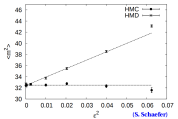


Importance sampling and Markov chains

Hybrid Monte Carlo and friends

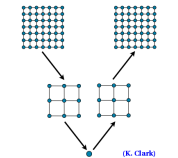
Iterative algorithms for fermions

[Time permitting] Alternatives to address sign problems



Interaction encouraged

"It's better to uncover a little than to cover a lot" (V. Weisskopf)



Further resources

Algorithms are a broad domain overlapping with other sessions

There are many excellent resources — here is a small sample:

- **Textbooks**
  Newman & Barkema, "Monte Carlo Methods in Statistical Physics" (1999)
  DeGrand & DeTar, "Lattice Methods for Quantum Chromodynamics" (2006)
  Gattringer & Lang, "Quantum Chromodynamics on the Lattice" (2010)
  Knechtli, Günther & Peardon, "Lattice Quantum Chromodynamics" (2017)

- **Lecture notes**
  Les Houches, "Modern perspectives in lattice QCD" (2009)
  Kennedy, "Algorithms for Dynamical Fermions" (2012)
  Joseph, "Markov Chain Monte Carlo Methods in Quantum Field Theories" (2020)

- "Algorithms" track at the annual Lattice conference

Big picture: Monte Carlo integration

Goal: Predict observable expectation values

$$\langle \mathcal{O} \rangle \;=\; \frac{1}{\mathcal{Z}} \int \mathcal{D}\Phi \;\; \mathcal{O}(\Phi) \;\; e^{-S[\Phi]} \;=\; \frac{\int \mathcal{D}\Phi \;\; \mathcal{O}(\Phi) \;\; e^{-S[\Phi]}}{\int \mathcal{D}\Phi \;\; e^{-S[\Phi]}}$$

Lattice regularization $\longrightarrow$ path integral is finite-dimensional

but still analytically intractable

Big picture: Monte Carlo integration

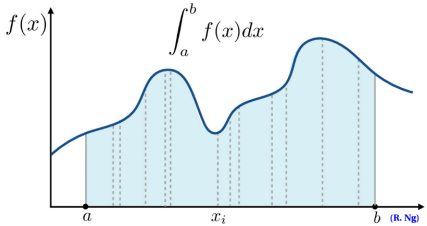
Lattice regularization $\longrightarrow$ path integral is finite-dimensional

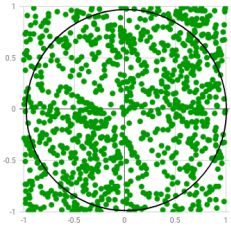
but still analytically intractable

Monte Carlo integration

Stochastically sample points in integration domain

$\longrightarrow$ most powerful for high-dimensional domains





Monte Carlo path integration

$$\langle \mathcal{O} \rangle \;=\; \frac{1}{\mathcal{Z}} \int \mathcal{D}\Phi \;\; \mathcal{O}(\Phi) \;\; e^{-S[\Phi]}$$

Stochastically sample **lattice field configurations**

$\longrightarrow$ plenty of dimensions in integration domain

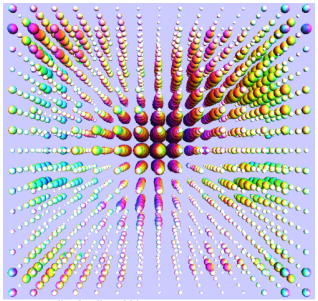
SU(N) gauge fields on $L^3 \times N_T$ lattice:

$$\sharp_G = (N^2 - 1) \times 4 \times L^3 \times N_T \;\sim\; 10^9$$

for typical $N = 3,\; L = 64,\; N_T = 128$

Similarly, for $N_F = 2$ quark flavors:

$$\sharp_F = 4N \times N_F \times L^3 \times N_T \;\sim\; 10^9$$



(Image credit: Claudio Rebbi)

Importance sampling Monte Carlo

Typical lattice calculations sample $\lesssim 10^3$ independent field configurations to estimate each $\sim 10^9$-dimensional integral

This succeeds thanks to using **importance sampling** algorithms

By sampling field configurations with (unknown!) probability $\frac{1}{\mathcal{Z}} e^{-S[\Phi]}$

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathcal{D}\Phi \;\; \mathcal{O}(\Phi) \;\; e^{-S[\Phi]}$$

$$\longrightarrow \;\; \frac{1}{N} \sum_{i=1}^{N} \mathcal{O}(\Phi_i) \;\; \text{with statistical uncertainty} \;\; \propto \frac{1}{\sqrt{N}}$$

Importance sampling Monte Carlo

By sampling field configurations with (unknown!) probability $\frac{1}{\mathcal{Z}} e^{-S[\Phi]}$

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathcal{D}\Phi \;\; \mathcal{O}(\Phi) \;\; e^{-S[\Phi]}$$

$$\longrightarrow \;\; \frac{1}{N} \sum_{i=1}^{N} \mathcal{O}(\Phi_i) \;\; \text{with statistical uncertainty} \;\; \propto \frac{1}{\sqrt{N}}$$

For now let's assume $S[\Phi]$ is real and positive

so that $\frac{1}{\mathcal{Z}} e^{-S[\Phi]}$ can be considered a probability

Markov-chain Monte Carlo

How to sample configurations according to unknown probability $p_\Phi \propto e^{-S[\Phi]}$?

Markov chains

Generate each new sample based on the current field configuration,

so $P(A \to B)$ depends on no configurations besides A and B

Design the generation such that samples are correctly distributed

$$\Phi_0 \longrightarrow \Phi_1 \longrightarrow \Phi_2 \longrightarrow \Phi_3 \longrightarrow \cdots \longrightarrow \Phi_N$$

Markov chain conditions for correctness

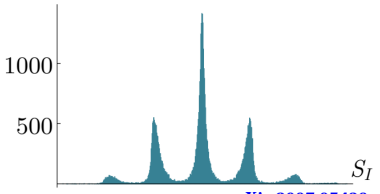
Ergodicity

The chain must be able to reach any allowed configuration

In practice, can be challenging. . .

. . . to correctly sample different topological sectors with small lattice spacings

. . . to correctly sample different Lefschetz thimbles . . . etc.



arXiv:2007.05436

Markov chain conditions for correctness

Ergodicity

The chain must be able to reach any allowed configuration

Balance

No sources or sinks of probability:

$$\sum_A p_A \, P(A \to B) = \sum_A p_B \, P(B \to A)$$

(sums include $A = B$)

Typical algorithms satisfy stronger **detailed** balance condition:

$$p_A \, P(A \to B) = p_B \, P(B \to A)$$

Example: Metropolis–Rosenbluth–Rosenbluth–Teller–Teller

The paradigmatic Markov chain algorithm

1) Modify current configuration (A) to propose new configuration (B)

(flip a spin, rotate a gauge link, . . .)

2) Compute the resulting change in lattice action, $\Delta S = S_B - S_A$

3) Accept proposed modification with probability

$$P_{\text{acc}}(A \to B) = \min\left\{1, e^{-\Delta S}\right\} = \min\left\{1, e^{-(S_B - S_A)}\right\}$$

Otherwise **repeat** current configuration A in chain

Metropolis–Rosenbluth–Rosenbluth–Teller–Teller illustration

Transition probability combines both generating and accepting proposed update:

$$P(A \to B) = P_{\text{gen}}(A \to B) \times P_{\text{acc}}(A \to B)$$

Simplifying with symmetric generation probability $P_{\text{gen}}(A \to B) = P_{\text{gen}}(B \to A)$,

$$\frac{P(A \to B)}{P(B \to A)} = \frac{P_{\text{acc}}(A \to B)}{P_{\text{acc}}(B \to A)} = \frac{\min\left\{1, e^{-(S_B - S_A)}\right\}}{\min\left\{1, e^{-(S_A - S_B)}\right\}} = \frac{e^{-S_B}}{e^{-S_A}}$$

$$\implies \text{detailed balance,} \quad e^{-S_A}\, P(A \to B) = e^{-S_B}\, P(B \to A)$$

Initial configuration and equilibration

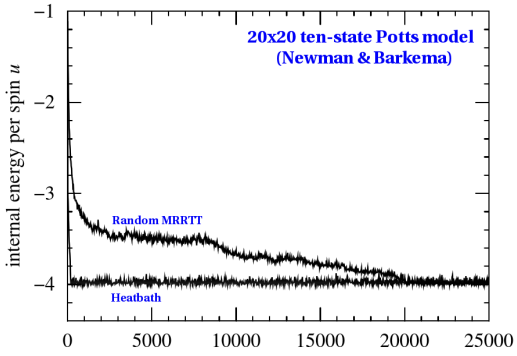
An ergodic & balanced Markov chain can start from **any** initial configuration

Eventually it will reach the important configurations with large probabilities

Equilibration

Depends on update algorithm

Can be a bottleneck in practice



('Heatbath' updates craft proposals

to maximize acceptance rate)

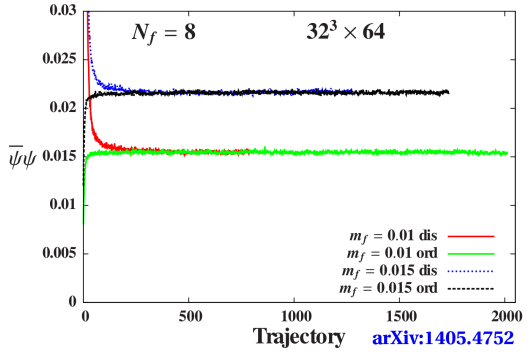
Initial configuration and equilibration

An ergodic & balanced Markov chain can start from **any** initial configuration

Eventually it will reach the important configurations with large probabilities

Comparing different initial configurations can help estimate equilibration time

Example: Convergence of hot- vs. cold-start calculations



Auto-correlations

A more general challenge

Equilibration is one aspect of 'auto'-correlations
between subsequent configurations sampled in the chain

$$\Phi_0 \longrightarrow \Phi_1 \longrightarrow \Phi_2 \longrightarrow \Phi_3 \longrightarrow \cdots \longrightarrow \Phi_N$$

Generating $\Phi_i \longrightarrow \Phi_{i+1} \implies \Phi_{i+1}$ correlated with Φ_i

Correlations decrease for configurations more distantly separated in chain,
which eventually become statistically independent

Auto-correlation function

$$\Phi_0 \longrightarrow \Phi_1 \longrightarrow \Phi_2 \longrightarrow \Phi_3 \longrightarrow \cdots \longrightarrow \Phi_N$$

Correlations decrease for configurations more distantly separated in chain, which eventually become statistically independent

Quantify by auto-correlation function $\rho_{\mathcal{O}}(t) = \dfrac{C_{\mathcal{O}}(t)}{C_{\mathcal{O}}(0)}$ where

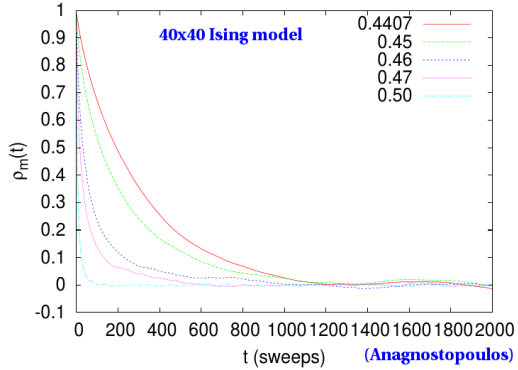
$$C_{\mathcal{O}}(t) = \langle \{\mathcal{O}_i - \langle \mathcal{O} \rangle\} \cdot \{\mathcal{O}_{i+t} - \langle \mathcal{O} \rangle\} \rangle = \langle \mathcal{O}_i \mathcal{O}_{i+t} \rangle - \langle \mathcal{O} \rangle^2 \qquad \mathcal{O}_i \equiv \mathcal{O}(\Phi_i)$$

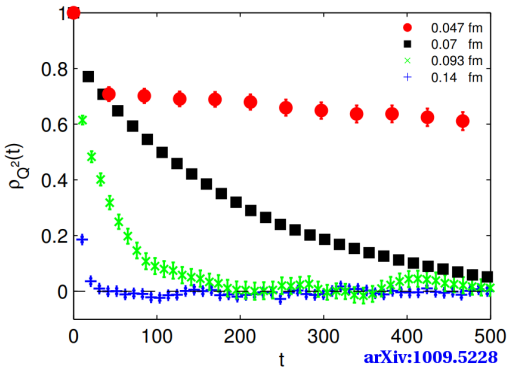
Auto-correlations depend **both** on updating method and on observable $\mathcal{O}$

Auto-correlation function

Quantify by auto-correlation function $\rho_{\mathcal{O}}(t) = \dfrac{C_{\mathcal{O}}(t)}{C_{\mathcal{O}}(0)}$ where

$$C_{\mathcal{O}}(t) = \langle \{\mathcal{O}_i - \langle\mathcal{O}\rangle\} \cdot \{\mathcal{O}_{i+t} - \langle\mathcal{O}\rangle\} \rangle = \langle \mathcal{O}_i \mathcal{O}_{i+t} \rangle - \langle\mathcal{O}\rangle^2 \qquad \mathcal{O}_i \equiv \mathcal{O}(\Phi_i)$$





Auto-correlation time

Auto-correlation functions typically decrease exponentially, $\rho(t) \simeq \exp\left[-\frac{t}{\tau_{\text{exp}}}\right]$

$\longrightarrow$ Simpler quantification by "auto-correlation time"

('time' counting samples in chain, not physical dimension)

In practice **integrated auto-correlation time** more robust

$$\tau_{\text{int}} = \int_0^\infty \rho(t)\, dt \approx \frac{1}{2} + \sum_{t=1}^{W} \rho(t)$$

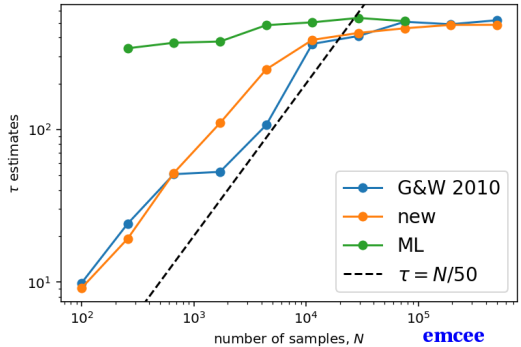
Adjustable "window" $\tau_{\text{int}} \ll W \ll N$ to control contamination by large-t noise

Integrated auto-correlation time

$$\tau_{\text{int}} \approx \frac{1}{2} + \sum_{t=1}^{W} \rho(t)$$

Adjustable "window" $\tau_{\text{int}} \ll W \ll N$ to control contamination by large-t noise

Can be quite delicate $\longrightarrow$ best to use established package [UWerr; UNEW; emcee]



Statistically independent samples

Earlier we claimed importance sampling $\longrightarrow$ statistical uncertainties $\propto \frac{1}{\sqrt{N}}$

This is true only for uncorrelated samples

Auto-correlation time $\tau_{\mathcal{O}}$ for observable $\mathcal{O}$ $\longrightarrow$ statistical uncertainty $\propto \sqrt{\frac{2\tau_{\mathcal{O}}}{N}}$

Equivalently, have only $N_{\text{indep.}} = \frac{N}{2\tau_{\mathcal{O}}}$ statistically independent samples

See Gattringer & Lang eq. 4.63 for factor of 2,

or note $\rho(t) = 0 \longrightarrow \tau \approx \frac{1}{2} + \sum \rho(t) = \frac{1}{2}$

Auto-correlated data analysis

Have only $N_{\text{indep.}} = \dfrac{N}{2\tau_{\mathcal{O}}}$ statistically independent samples

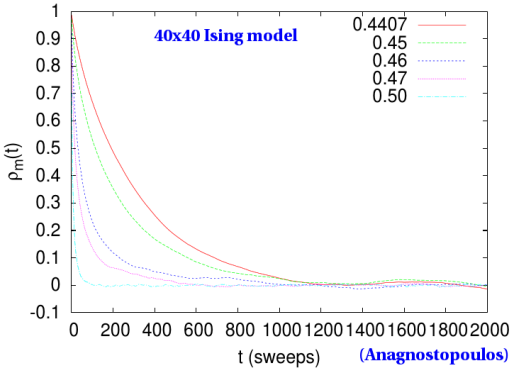
Common technique — 'blocking' a.k.a. 'binning'

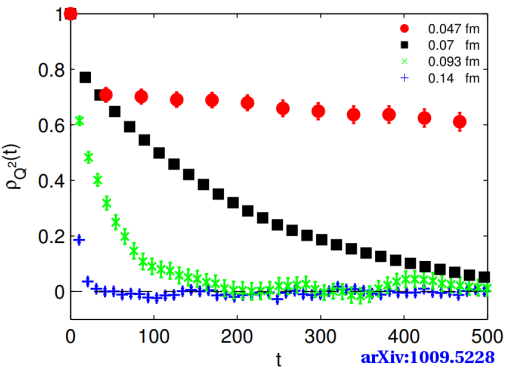
Define $\dfrac{N}{B}$ statistically independent data points

by averaging each 'block' of $B \gtrsim 2 \max\{\tau_{\mathcal{O}}\}$ raw measurements

Feed blocked data into standard jackknife or bootstrap analyses

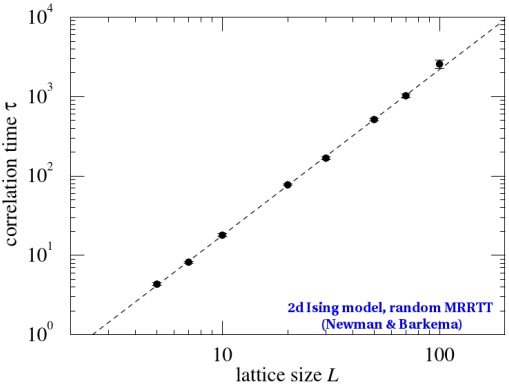
Critical slowing down

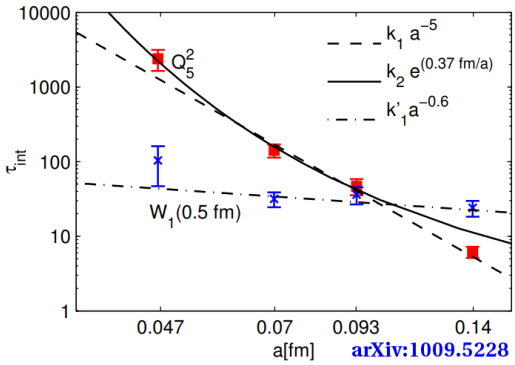




Auto-correlations typically get (much) worse near critical points,
including continuum limit of lattice gauge theories

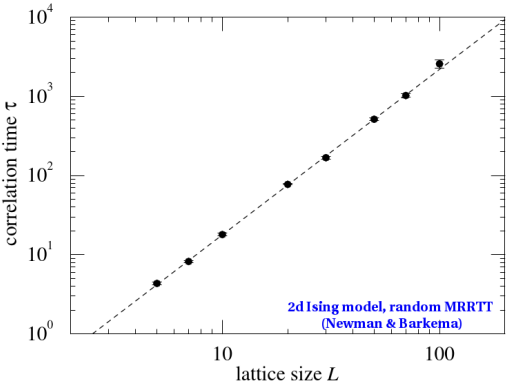
Critical slowing down

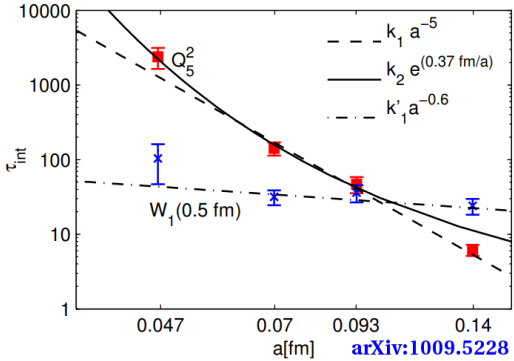




Auto-correlations typically get (much) worse near critical points,
including continuum limit of lattice gauge theories

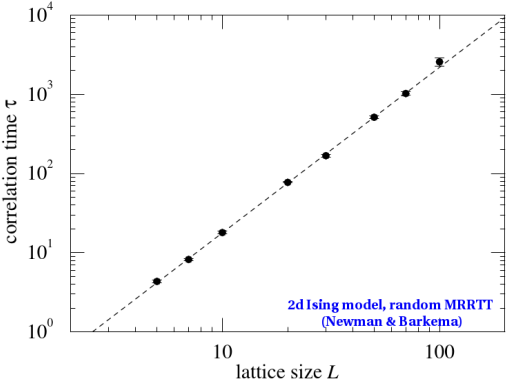
Critical slowing down

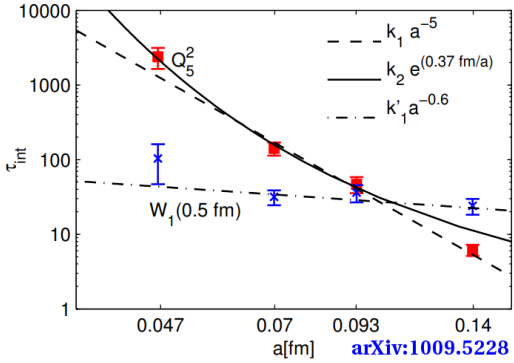




Critical fluctuations over all length scales cut off by lattice size $L \longrightarrow \tau_{\mathcal{O}} \propto L^z$
with algorithm-dependent dynamical critical exponent $z \geq 0$

Critical slowing down





$\tau_{\mathcal{O}} \propto L^z \implies$ total computational costs $\propto L^{d+z}$ in d dimensions

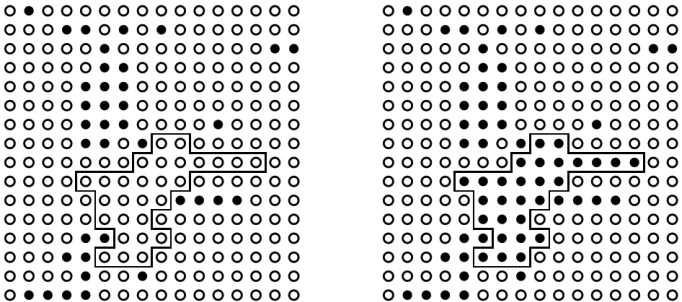
Aside: Cluster algorithms

For spin systems and scalar field theories,

cluster algorithms dramatically reduce critical slowing down

(2d Ising $z \approx 2 \longrightarrow 0.25$)

Incorporate critical fluctuations over all length scales



(Newman & Barkema)

Aside: Cluster algorithms

For spin systems and scalar field theories,
cluster algorithms dramatically reduce critical slowing down
(2d Ising $z \approx 2 \longrightarrow 0.25$)

Incorporate critical fluctuations over all length scales

No **efficient** cluster algorithm for gauge theories

'Quantum link model' formulation is related work
now contributing to quantum simulations

Logistics of lattice calculations

Markov chain equilibration and decorrelation

$\longrightarrow$ divide large-scale lattice calculations into two stages

Configuration generation

Configuration analysis

Depending on analyses, computational costs may be comparable

Stages typically require different **parallelization** strategies

Logistics of lattice calculations

Configuration generation

Configurations obtained only one at a time in each Markov chain

efficient **data-parallel** updating algorithms crucial

Save ensemble of configurations for subsequent analyses,

typically subset (e.g., every 10th) to allow partial decorrelation

Analyses of saved configurations

Each configuration can be analyzed independently

$\longrightarrow$ **ideal parallelization** in addition to data parallelism

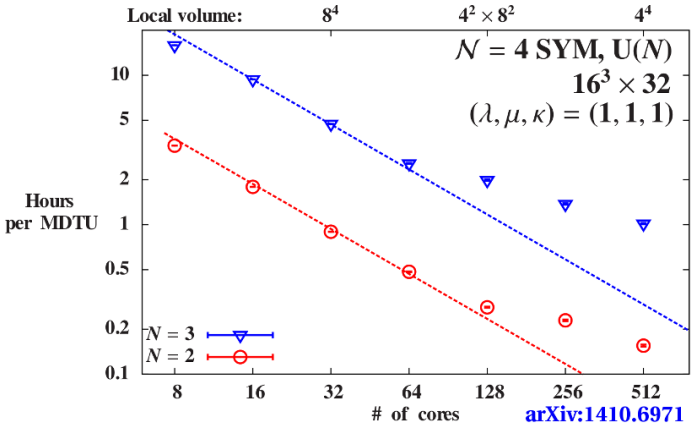
Data parallelism and strong scaling

Divide lattice volume among computing units / processes / threads
that need to communicate to treat boundaries consistently

Strong scaling

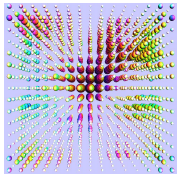
"How much speed-up from data parallelization?"

Communications eventually slow things down



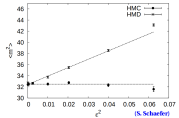
Checkpoint

The capabilities and reliability of lattice calculations
depend on the algorithms available



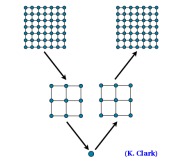
✓ Importance sampling and Markov chains — Questions?

Hybrid Monte Carlo and friends



Iterative algorithms for fermions

[Time permitting] Alternatives to address sign problems



Interaction encouraged

"It's better to uncover a little than to cover a lot" (V. Weisskopf)

Specialize to gauge–fermion lattice field theories

$$\mathcal{Z} = \int \mathcal{D}\Phi \;\; e^{-S[\Phi]} \;\longrightarrow\; \int \mathcal{D}U\mathcal{D}\overline{\Psi}\mathcal{D}\Psi \;\; e^{-S_G[U]-S_F[U,\overline{\Psi},\Psi]}$$

SU(N) gauge links U_μ are $N \times N$ complex matrices

Gauge action $S_G[U]$ built from plaquettes, optionally larger loops of links

Fermions $\{\overline{\Psi}, \Psi\}$ in fundamental rep. are $4N$-component vectors
of anti-commuting Grassmann variables

Fermion action $S_F[U, \overline{\Psi}, \Psi] = \overline{\Psi}\, M[U]\, \Psi$
where Dirac operator $M[U] = D[U] + m\mathbb{I}_{\sharp_F}$ is **large** $\sharp_F \times \sharp_F$ matrix

Fermion determinant from Grassmann integration

Bilinear fermion action $\Longrightarrow$

$$\mathcal{Z} = \int \mathcal{D}U \;\; e^{-S_G[U]} \int \mathcal{D}\overline{\Psi}\mathcal{D}\Psi \;\; e^{-\overline{\Psi}\, M[U]\, \Psi} = \int \mathcal{D}U \;\; e^{-S_G[U]} \;\; \det\left(M[U]\right)$$

$\Longrightarrow$ only need to integrate over bosonic fields

(Auxiliary fields can transform more general fermion actions to bilinear form)

Can ensure positivity by considering even number of identical fermion flavors N_F

$$\int \mathcal{D}\overline{\Psi}_1\mathcal{D}\Psi_1\mathcal{D}\overline{\Psi}_2\mathcal{D}\Psi_2 \;\; e^{-\overline{\Psi}_1 M\Psi_1 - \overline{\Psi}_2 M\Psi_2} = (\det M)^2 = \det\left[M^\dagger M\right]$$

using γ_5-hermiticity $\longrightarrow$ $\det[M] = \det\left[\gamma_5 M^\dagger \gamma_5\right] = \det\left[M^\dagger\right]$

Determinantal Monte Carlo

$$\text{sampling probability} \;\propto\; e^{-S_G[U]} \;\det\left[M^\dagger M\right]$$

Fermion determinant is demanding proposition

$\sim$ billion $\times$ billion matrix impractical to store in memory

Cost of evaluating $\propto \sharp_F^3 \propto V^3$

'Global' quantity $\longrightarrow$ needs to be recomputed if even a single link changes

Re-evaluating for each of $\mathcal{O}(V)$ gauge link updates $\longrightarrow$ costs $\propto V^4$

'Determinantal' Monte Carlo is used (with some tricks) in condensed matter

(Algorithms for Lattice Fermions, arXiv:2012.11914)

Pseudofermions

Exploit similarity between Grassmann and gaussian integration

$$\det\left[M^\dagger M\right] \propto \int \mathcal{D}\Phi^\dagger \mathcal{D}\Phi \;\; e^{-\Phi^\dagger \left(M^\dagger M\right)^{-1} \Phi}$$

with $\left\{\Phi^\dagger, \Phi\right\}$ commuting complex 'pseudofermions'

$$\mathcal{Z} = \int \mathcal{D}U \mathcal{D}\Phi^\dagger \mathcal{D}\Phi \;\; e^{-S_G[U] - \Phi^\dagger \left(M^\dagger M\right)^{-1} \Phi}$$

Fermion mass $m > 0$ ensures well-defined inverse

Inversions dominate costs, but **much** cheaper than determinant
thanks to iterative algorithms to be discussed next time

Avoiding inversions

$$\mathcal{Z} = \int \mathcal{D}U \mathcal{D}\Phi^\dagger \mathcal{D}\Phi \;\; e^{-S_G[U] - \Phi^\dagger \left(M^\dagger M\right)^{-1} \Phi}$$

Inversion still needs to be recomputed if even a single link changes

$\longrightarrow$ re-evaluating for each of $\mathcal{O}(V)$ gauge link updates remains impractical

Want to update many links between accept/reject tests

Doing so naively leads to large $\Delta S \longrightarrow$ low acceptance

Trick is to use micro-canonical ('energy-conserving') updates

$\longrightarrow$ **molecular dynamics**

Molecular dynamics (MD)

Add conjugate momentum P for each bosonic variable U

[traceless hermitian matrix for SU(N) gauge link]

$\longrightarrow$ effective hamiltonian (a.k.a. 'action' S)

$$H_{\text{eff}}(P, U) = \frac{1}{2}\sum P^2 + S_G[U] + \Phi^\dagger \left(M^\dagger M\right)^{-1} \Phi$$

Evolve in 'MD time' τ via Hamilton's equations

$$\frac{dU}{d\tau} = \frac{dH}{dP} \qquad \frac{dP}{d\tau} = -\frac{dH}{dU}$$

Molecular dynamics (MD)

$$H_{\text{eff}}(P, U) = \frac{1}{2}\sum P^2 + S_G[U] + \Phi^\dagger \left(M^\dagger M\right)^{-1} \Phi$$

Evolve in 'MD time' τ via Hamilton's equations

$$\frac{dU}{d\tau} = \frac{dH}{dP} = P$$
$$\frac{dP}{d\tau} = -\frac{dH}{dU} = -\frac{dS_G}{dU} - \left[(M^\dagger M)^{-1}\Phi\right]^\dagger \frac{d\left(M^\dagger M\right)}{dU}\left[(M^\dagger M)^{-1}\Phi\right]$$

(exercise)

Requires more inversions, but all links updated in between inversions ✓

Hybrid molecular dynamics (HMD)

$$H_{\text{eff}}(P, U) = \frac{1}{2} \sum P^2 + S_G[U] + \Phi^\dagger \left(M^\dagger M\right)^{-1} \Phi$$

MD evolution (in principle) conserves H_{eff}, exchanging energy between P and U

Improve by regularly refreshing momenta after each "trajectory" τ

- $\longrightarrow$ draw stochastic P from gaussian distribution ("heat-bath")
- $\longrightarrow$ explore different constant-energy hyper-surfaces in (P, U) phase space
- $\longrightarrow$ "**hybrid**" combination of momentum heat-bath and MD evolution

Hybrid molecular dynamics (HMD)

$$H_{\text{eff}}(P,U) = \frac{1}{2}\sum P^2 + S_G[U] + \Phi^\dagger \left(M^\dagger M\right)^{-1} \Phi$$

Regularly refresh momenta after each trajectory

stochastically drawing P from gaussian distribution

Even easier for pseudofermions!

Set up gaussian random complex vector R at start of each trajectory

Set $\Phi = M^\dagger \cdot R \longrightarrow$ desired U-dependent distribution for $R^\dagger R = \Phi^\dagger \left(M^\dagger M\right)^{-1} \Phi$

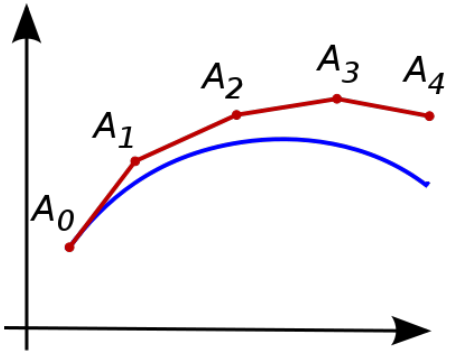
$\left\{\Phi^\dagger, \Phi\right\}$ unchanged during MD evolution

Molecular dynamics integration

Numerical MD evolution requires **integrator** algorithm

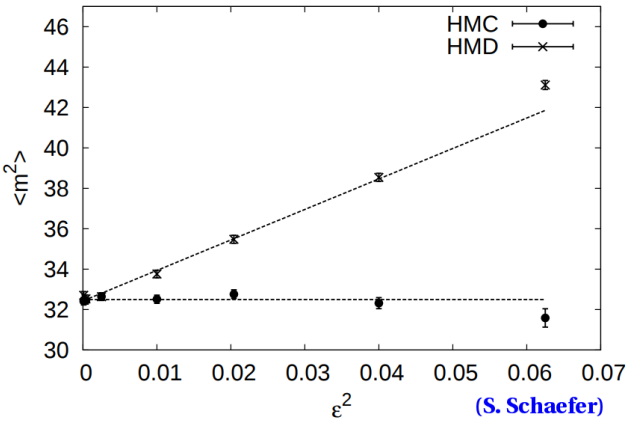
Updates use discrete ‘steps’ of non-zero size ε

$\longrightarrow$ discretization errors accumulate over $N_S = \tau/\varepsilon$ steps



Hybrid Monte Carlo (HMC)

Step size errors $\longrightarrow$ $\Delta H \neq 0$
and contaminate all $\langle \mathcal{O} \rangle$



Solution: Accept/reject test on ΔH after each trajectory

$P_{\text{acc}} = \min\left\{1, e^{-\Delta H}\right\}$ $\longrightarrow$ exact **hybrid Monte Carlo** algorithm

Validity of HMC

Recall Markov chain conditions for correctness

Ergodicity needs to be checked for theory of interest

Detailed balance guaranteed if MD integrator is both **symmetric** and **symplectic**

Symmetry $\longrightarrow$ MD evolution is reversible (to machine precision)

Simple to check: 1) Evolve $(P, U) \longrightarrow (P', U')$

2) Negate $P' \rightarrow -P'$

3) Evolve $(-P', U') \longrightarrow (-P, U)$

and confirm starting configuration recovered

Validity of HMC

Recall Markov chain conditions for correctness

Ergodicity needs to be checked for theory of interest

Detailed balance guaranteed if MD integrator is both **symmetric** and **symplectic**

A **symplectic** integrator preserves the phase-space 'volume' $\mathcal{D}P\mathcal{D}U$

$$\longrightarrow \text{jacobian} = \det\left[\frac{\partial(P', U')}{\partial(P, U)}\right] = 1$$

Can check via 'Creutz equality' $\left\langle e^{-\Delta H}\right\rangle = 1$ (exercise)

Symmetric & symplectic MD integrators

Build from two elementary updates ($\varepsilon \ll 1$):

$$\mathcal{T}_U(\varepsilon) \;:\; (P, U) \longrightarrow (\quad P, \quad U + \varepsilon \nabla_P H)$$

$$\mathcal{T}_P(\varepsilon) \;:\; (P, U) \longrightarrow (P - \varepsilon \nabla_U H, \quad U \quad)$$

where 'force' $\nabla_U H$ requires expensive inversion $\left(M^\dagger M\right)^{-1}$

Compose to evolve over full trajectory of length $\tau = N_S \times \varepsilon$

Example: Euler integrator is $T_\varepsilon(\tau) = [\mathcal{T}_P(\varepsilon)\, \mathcal{T}_U(\varepsilon)]^{N_S}$

(notation: operator that first updates U then P)

Higher-order MD integrators

Euler integrator $T_\varepsilon(\tau) = [\mathcal{T}_P(\varepsilon)\, \mathcal{T}_U(\varepsilon)]^{N_S} \longrightarrow \left\langle \Delta H^2 \right\rangle^{1/2} \propto \varepsilon$

$\Longrightarrow$ need small step size ε or else low acceptance $\longrightarrow$ longer auto-correlations

Easy to do better with same number of inversions ($\mathcal{T}_P$):

$$T_\varepsilon(\tau) = [\mathcal{T}_U(\varepsilon/2)\, \mathcal{T}_P(\varepsilon)\, \mathcal{T}_U(\varepsilon/2)]^{N_S} \longrightarrow \left\langle \Delta H^2 \right\rangle^{1/2} \propto \varepsilon^2$$

$\longrightarrow$ second-order Verlet ("leap-frog") integrator

[can fuse $\mathcal{T}_U(\varepsilon/2)\, \mathcal{T}_U(\varepsilon/2) = \mathcal{T}_U(\varepsilon)$ for inner steps]

Higher-order MD integrators with $\left\langle \Delta H^2 \right\rangle^{1/2} \propto \varepsilon^4$, $\left\langle \Delta H^2 \right\rangle^{1/2} \propto \varepsilon^6$, etc.

require more inversions for each full step

Exercises (1)

You will write and test HMC code for ϕ^4 theory in $D = 3$ dimensions

[optional template code available]

Break up into several small exercises:

- Generate gaussian-distributed momenta π_x
- Implement $H = \sum_x \left[\frac{1}{2}\pi_x^2 - 2\kappa \sum_{\mu=0}^{D-1} \phi_x \phi_{x+\hat{\mu}} + \phi_x^2 + \lambda \left(\phi_x^2 - 1\right)^2 \right]$
- Derive the 'force' $-\nabla_\phi H$ needed for the elementary update $\mathcal{T}_\pi$
- Implement the elementary updates $\mathcal{T}_\phi$ and $\mathcal{T}_\pi$
- Combine the elementary updates into a leap-frog integrator with N_S steps
- Implement ΔH and set up accept/reject test

Exercises (2)

You will write and test HMC code for ϕ^4 theory in $D = 3$ dimensions

[optional template code available — model solution code to come later in the school]

Break up into several small exercises:

- Check the reversibility of the integrator [6^3 lattice should suffice]

  [$\lambda = 1.1689$ and $\kappa = 0.185825$ are reasonable values near the critical line]

- Check $\left\langle \Delta H^2 \right\rangle^{1/2} \propto \varepsilon^2$ and the Creutz equality $\left\langle e^{-\Delta H} \right\rangle = 1$

  [$\sim 10^3$ trajectories with $\tau = 1$ should suffice]

- Combine the elementary updates into an Omelyan–Mryglod–Folk integrator and compare $\left\langle \Delta H^2 \right\rangle^{1/2}$ vs. the leap-frog integrator

Exercises (3)

You will write and test HMC code for ϕ^4 theory in $D = 3$ dimensions

[optional template code available — model solution code to come later in the school]

Break up into several small exercises:

- Measure the 'magnetization' $m = \dfrac{M}{V} = \dfrac{1}{V} \sum_x \phi_x$

  $\langle m \rangle = 0$ by symmetry $\longrightarrow$ interesting observables are magnetic susceptibility and Binder cumulant

$$\chi = \frac{1}{V} \left\langle M^2 \right\rangle = V \left\langle m^2 \right\rangle \qquad\qquad U = \frac{\left\langle m^4 \right\rangle}{\left\langle m^2 \right\rangle^2}$$

Exercises (4)

You will write and test HMC code for ϕ^4 theory in $D = 3$ dimensions

[optional template code available — model solution code to come later in the school]

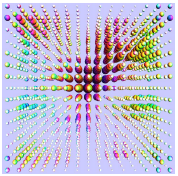
Break up into several small exercises:

- Monitor m for $\sim 10^3$ trajectories
  to check equilibration for $0.15 \leq \kappa \leq 0.2$ with $\lambda = 1.145$
- Collect magnetization data for $\sim 10^6$ **equilibrated** trajectories
  and estimate its auto-correlation time for $0.15 \leq \kappa \leq 0.2$ with $\lambda = 1.145$
- Experiment with other values of L, λ, κ, etc., as curiosity drives you

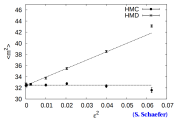
[hep-lat/9806012 might provide inspiration]

Next time

The capabilities and reliability of lattice calculations depend on the algorithms available



- ✓ Importance sampling and Markov chains
- Hybrid Monte Carlo and friends — wrap up
- Iterative algorithms for fermions
- [Time permitting] Alternatives to address sign problems



Interaction encouraged

"It's better to uncover a little than to cover a lot" (V. Weisskopf)