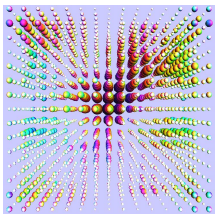
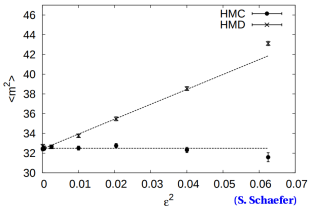
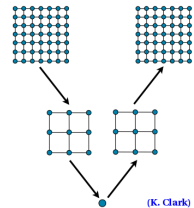
Lattice Field Theory Algorithms — Part 2

David Schaich (University of Liverpool)







Methods of Effective Field Theory and Lattice Field Theory

Bad Honnef Physics School, 16 July 2021

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

Any questions about last time?

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

✓ Importance sampling and Markov chains

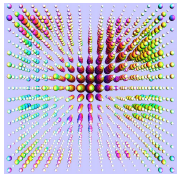
Hybrid Monte Carlo and friends — wrap up

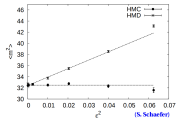
Iterative algorithms for fermions

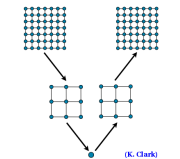
[Time permitting] Alternatives to address sign problems

Interaction encouraged

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







Hybrid Monte Carlo recap

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

1) Stochastic conjugate momenta P and pseudofermions $\left\{\Phi^\dagger, \Phi\right\}$

2) Inexact molecular dynamics (MD) evolution
along trajectory $\tau = N_S \times \varepsilon$ in unphysical 'MD time'

3) Accept/reject test on energy non-conservation ΔH from $\varepsilon > 0$

Costs dominated by U-dependent $\left(M^\dagger M\right)^{-1}$ inversions
to evaluate H and update P ($\sim$ 'force') in MD evolution

MD integrator considerations

Standard second-order Verlet ("leap-frog") integrator:

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

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

Small step size $\varepsilon \longrightarrow$ lots of inversions $\longrightarrow$ large costs

Large step size $\varepsilon \longrightarrow$ low acceptance $\langle P_{\text{acc}} \rangle = \operatorname{erfc}\left(\frac{1}{8} \left\langle \Delta H^2 \right\rangle^{1/2}\right)$
$\longrightarrow$ long auto-correlations $\longrightarrow$ large costs

Higher-order integrators $\left\langle \Delta H^2 \right\rangle^{1/2} \propto \varepsilon^n \longrightarrow$ more inversions $\longrightarrow$ large costs

MD integrators and optimization

Worthwhile to improve **coefficient** of $\left\langle \Delta H^2 \right\rangle^{1/2} \propto \varepsilon^2$

Omelyan–Mryglod–Folk integrator:

$$\mathcal{T}_\varepsilon(\tau) = \left[\mathcal{T}_U(\xi\varepsilon)\ \mathcal{T}_P(\varepsilon/2)\ \mathcal{T}_U\left((1-2\xi)\varepsilon\right)\ \mathcal{T}_P(\varepsilon/2)\ \mathcal{T}_U(\xi\varepsilon)\right]^{N_S}$$

with tunable parameter $0 \leq \xi \leq 0.5$ (typically $\xi \approx 0.19$)

Takaishi & de Forcrand, hep-lat/0505020

$2\times$ inversions per step vs. $\sim 3\times$ larger steps for fixed acceptance rate

$\longrightarrow \sim 50\%$ speedup

Optimal $P_{\text{acc}} = \exp\left[-1/n\right]$ for nth-order MD integrator

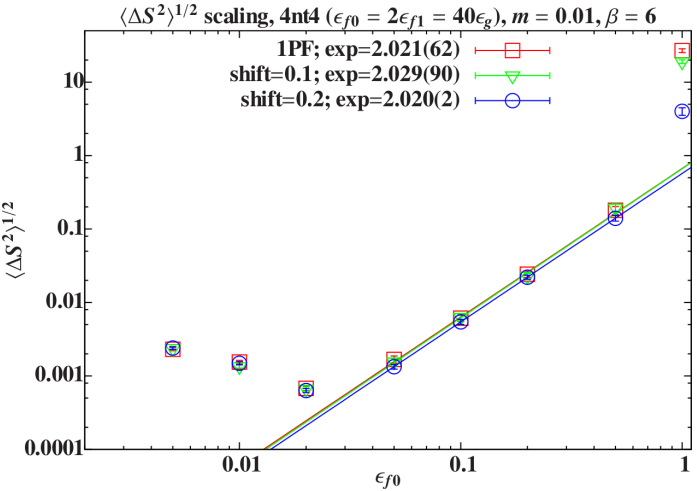
[$\sim 60\%$ for $n = 2$; $\sim 80\%$ for $n = 4$]

MD integrators and debugging

$$\left\langle \Delta H^2 \right\rangle^{1/2} \propto \varepsilon^2$$

Non-trivial debugging check

Any errors must match between action and forces



We'll come back to this plot

HMC cost scaling

Computational cost dependence on volume $V = L^3 \times N_T$

Simple factors $\propto V$ for updating $\mathcal{O}(V)$ links and momenta

Inversions also $\propto V$ (next section)

Fix acceptance rate $\longleftrightarrow$ fix $\left\langle \Delta H^2 \right\rangle^{1/2} \propto V^{1/2}$ [hep-lat/0505020 or think random walk]

Second-order integrator $\longrightarrow \left\langle \Delta H^2 \right\rangle^{1/2} \propto V^{1/2} \varepsilon^2 = \text{const.}$

$\Longrightarrow$ smaller step size $\varepsilon \propto 1/V^{1/4} \longrightarrow$ additional cost $\propto V^{1/4}$

All together, HMC costs $\propto V^{5/4}$ — **much** better than determinant cost $\propto V^3$

HMC cost scaling

Simple factors $\propto V$ for updating $\mathcal{O}(V)$ links and momenta

Inversions also $\propto V$ (next section)

Fix acceptance rate $\longleftrightarrow$ fix $\left\langle \Delta H^2 \right\rangle^{1/2} \propto V^{1/2}$ [hep-lat/0505020 or think random walk]

Second-order integrator $\longrightarrow \left\langle \Delta H^2 \right\rangle^{1/2} \propto V^{1/2} \varepsilon^2 = \text{const.}$

$\Longrightarrow$ smaller step size $\varepsilon \propto 1/V^{1/4} \longrightarrow$ additional cost $\propto V^{1/4}$

All together, HMC costs $\propto V^{5/4}$ — **much** better than determinant cost $\propto V^3$

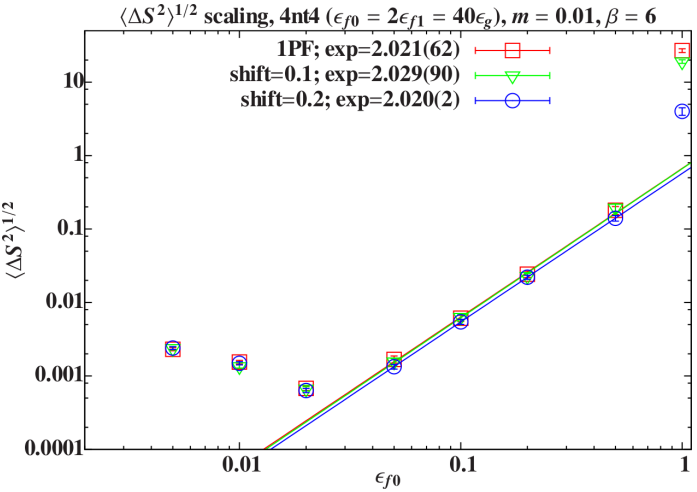
SU(N) gauge theories $\longrightarrow$ costs $\propto N^3$ from $N \times N$ matrix multiplication

Integrator instabilities

Coming back to this plot...

Small step size ε

Inexact inversion (later)
$\longrightarrow$ errors accumulate
over $\tau/\varepsilon \gg 1$ steps



Large steps $\longrightarrow$ integrator instabilities from large forces $\longrightarrow \left\langle \Delta H^2 \right\rangle^{1/2} \gtrsim 5$

Forces and integrator instabilities

Single large fluctuation in force at single lattice site can destabilize integrator
$\longrightarrow$ expect more difficulties as lattice volume increases

In practice problems arise from expensive fermion forces

"Smeared" lattice fermion actions can help
$\longrightarrow$ fermions couple to 'smoothed' gauge fields to reduce UV fluctuations

Smaller fluctuations $\longrightarrow$ higher acceptance for larger ε $\longrightarrow$ reduced costs

For given lattice action, "preconditioning" can also help

Mass preconditioning

Idea: Add an extra set of **heavy** pseudofermions $\left\{\Phi_H^\dagger, \Phi_H\right\}$:

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

Correct distribution from gaussian complex $\{R, R_H\}$ with

$$\Phi_H = M_H^\dagger \cdot R_H \qquad \Phi = M^\dagger M_H^{-1} \cdot R = M^\dagger \left(M_H^\dagger M_H\right)^{-1} M_H^\dagger \cdot R$$

Mass preconditioning

Idea: Add an extra set of **heavy** pseudofermions $\left\{\Phi_H^\dagger, \Phi_H\right\}$:

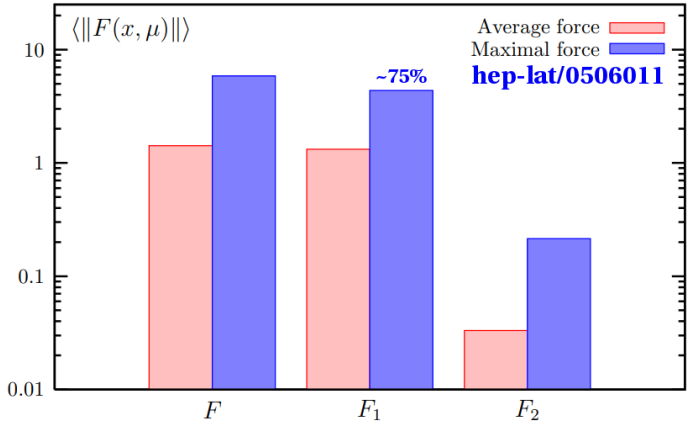
$$\det\left[M^\dagger M\right] = \frac{\det\left[M^\dagger M\right]}{\det\left[M_H^\dagger M_H\right]} \det\left[M_H^\dagger M_H\right]$$
$$\propto \int \mathcal{D}\Phi^\dagger \mathcal{D}\Phi \mathcal{D}\Phi_H^\dagger \mathcal{D}\Phi_H \;\; e^{-\Phi^\dagger M_H^\dagger \left(M^\dagger M\right)^{-1}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

Heavier mass $\longrightarrow$ small forces & cheaper inversions $\left(M_H^\dagger M_H\right)^{-1}$ (next section)

'Preconditioned' combination $M_H^\dagger \left(M^\dagger M\right)^{-1} M_H \longrightarrow$ reduced force fluctuations

Mass preconditioning

'Preconditioned' combination $M_H^\dagger \left(M^\dagger M\right)^{-1} M_H \longrightarrow$ reduced force fluctuations



Mass preconditioning and multiple step sizes

$$\int \mathcal{D}\Phi^\dagger \mathcal{D}\Phi \mathcal{D}\Phi_H^\dagger \mathcal{D}\Phi_H \; e^{-\Phi^\dagger M_H^\dagger \left(M^\dagger M\right)^{-1} M_H \Phi - \Phi_H^\dagger \left(M_H^\dagger M_H\right)^{-1} \Phi_H}$$

Can generalize MD integrator to use different step sizes

- Largest ε_0 for expensive $\left(M^\dagger M\right)^{-1}$
- Smaller ε_H for cheaper $\left(M_H^\dagger M_H\right)^{-1}$ (tuning # and masses)
- Smallest ε_G for cheapest $\frac{dS_G}{dU}$

Years turn into months

Depending on quark mass, leap-frog $\longrightarrow$ preconditioned multi-ε OMF can reduce costs by $\sim 10\times$

The nth-root trick

Rather than tuning masses of extra heavy pseudofermions

take n identical copies with nth root reducing fluctuations

$$\det\left[M^\dagger M\right] = \left(\det\left[M^\dagger M\right]^{1/n}\right)^n \propto \int \prod_{j=1}^{n} \mathcal{D}\Phi_j^\dagger \mathcal{D}\Phi_j \;\; e^{-\Phi_j^\dagger \left(M^\dagger M\right)^{-1/n} \Phi_j}$$

Clark & Kennedy, hep-lat/0608015

For fermion mass $0 < m \ll 1$, [roughly]

optimal $n_{\text{opt}} \sim \log \frac{1}{m} \longrightarrow$ reduces HMC costs $\sim m \left(\log \frac{1}{m}\right)^2$

The nth-root trick

Rather than tuning masses of extra heavy pseudofermions

take n identical copies with nth root reducing fluctuations

$$\det\left[M^\dagger M\right] = \left(\det\left[M^\dagger M\right]^{1/n}\right)^n \propto \int \prod_{j=1}^{n} \mathcal{D}\Phi_j^\dagger \mathcal{D}\Phi_j \;\; e^{-\Phi_j^\dagger \left(M^\dagger M\right)^{-1/n} \Phi_j}$$

Can generalize to consider single quark flavor:

$$\det M = \det\left[M^\dagger M\right]^{1/2} \propto \int \mathcal{D}\Phi^\dagger \mathcal{D}\Phi \;\; e^{-\Phi^\dagger \left(M^\dagger M\right)^{-1/2} \Phi}$$

Can generalize to consider Majorana fermions (e.g., supersymmetric gluinos):

$$\mathrm{pf}\, M = \det\left[M^\dagger M\right]^{1/4} \propto \int \mathcal{D}\Phi^\dagger \mathcal{D}\Phi \;\; e^{-\Phi^\dagger \left(M^\dagger M\right)^{-1/4} \Phi}$$

Rational hybrid Monte Carlo (RHMC)

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

How to take nth root of $\sim$ billion $\times$ billion matrix?

Rational approximation from series of p partial fractions:

$$\left(M^\dagger M\right)^{-1/n} = \alpha_0 + \sum_{j=1}^{p} \alpha_j \left(M^\dagger M + \beta_j \mathbb{I}_{\sharp_F}\right)^{-1}$$

'multi-shift' algorithm does all inversions simultaneously

Rational hybrid Monte Carlo (RHMC)

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

Rational approximation from series of p partial fractions:

$$\left(M^\dagger M\right)^{-1/n} = \alpha_0 + \sum_{j=1}^{p} \alpha_j \left(M^\dagger M + \beta_j \mathbb{I}_{\sharp_F}\right)^{-1}$$

'multi-shift' algorithm does all inversions simultaneously

Remez algorithm $\longrightarrow$ $\{\alpha_j, \beta_j\}$ that minimize approximation error given p and 'spectral range' (extremal eigenvalues)

Rational hybrid Monte Carlo (RHMC)

Rational approximation from series of p partial fractions:

$$\left(M^\dagger M\right)^{-1/n} = \alpha_0 + \sum_{j=1}^{p} \alpha_j \left(M^\dagger M + \beta_j \mathbb{I}_{\sharp_F}\right)^{-1}$$

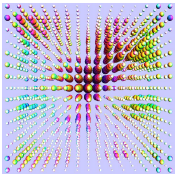
Resulting rational hybrid Monte Carlo (RHMC) algorithm just like HMC with rational approximation in $H_{\text{eff}}(P, U)$ and fermion forces

$$-\sum_{j=1}^{p} \alpha_j \left[(M^\dagger M + \beta_j \mathbb{I}_{\sharp_F})^{-1} \Phi\right]^\dagger \frac{d\left(M^\dagger M\right)}{dU} \left[(M^\dagger M + \beta_j \mathbb{I}_{\sharp_F})^{-1} \Phi\right]$$

[Can use different step size ε_j for each β_j]

Checkpoint

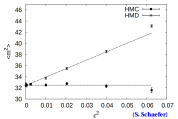
The capabilities and reliability of lattice calculations
depend on the algorithms available



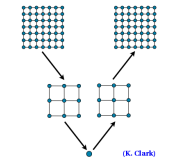
✓ Importance sampling and Markov chains

✓ Hybrid Monte Carlo and friends — Questions?

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)

Sparse matrix inversion

Exact $\sharp_F \times \sharp_F$ matrix inversion costs $\propto \sharp_F^3$ [with some caveats]

$\longrightarrow$ looks as bad as fermion determinant $\det\left[M^\dagger M\right]$

We don't need the full inverse — only $x = \left(M^\dagger M\right)^{-1} \Phi$ acting on 'source' vector

For **sparse** matrices with most elements zero

efficient iterative algorithms solve $\left(M^\dagger M\right) x = \Phi$ to given precision

'Efficient' $\longrightarrow$ costs $\propto \zeta\sqrt{\kappa}$ [ref]

$\zeta \propto \sharp_F \ll \sharp_F^2$ is number of non-zero matrix elements

κ is **condition number** — ratio of largest / smallest eigenvalues

Condition number

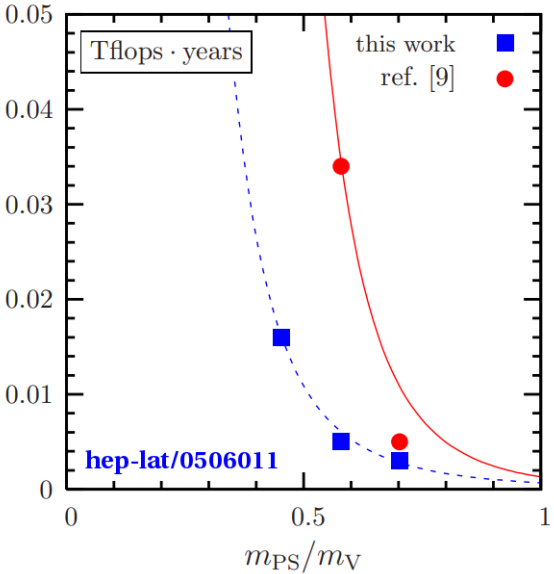
Cost $\propto \sharp_F \sqrt{\kappa}$ to solve $\left(M^\dagger M\right) x = \Phi$

Largest eig$\left(M^\dagger M\right) \sim \frac{1}{a^2} \equiv 1$

Smallest eig$\left(M^\dagger M\right) \sim m^2 \ll 1$

$\Longrightarrow$ costs $\sim \frac{V}{m}$ per solve

Many solves $\longrightarrow$ dominant cost of lattice calculations



Conjugate gradient algorithm — idea

Solving $Ax = \Phi$ where $A = M^{\dagger}M$ is positive-definite hermitian

Construct sequence x_k using **residual vector** $r_k = \Phi - Ax_k$ to choose next x_{k+1}

Demand $r_i^T \cdot r_j = 0 \ \forall \ i \neq j \longrightarrow$ 'never wrong in the same way twice'

Use r_k to generate next 'search direction' p_{k+1} with $p_i^T \cdot A \cdot p_j = 0 \ \forall \ i \neq j$ and

$$x_n = x_0 + \sum_{k=0}^{n-1} \alpha_k p_k$$

Repeat until residual $||r_n||$ smaller than given **target** ε

Conjugate gradient algorithm — details

Solving $Ax = \Phi$ where $A = M^\dagger M$ is positive-definite hermitian

Initialize x_0; $r_0 = \Phi - Ax_0$; $p_0 = r_0$

for $k = 0 \to \sharp_F$ **do**

$$\alpha_k = \frac{r_k^T \cdot r_k}{p_k^T \cdot A \cdot p_k}; \qquad x_{k+1} = x_k + \alpha_k p_k; \qquad r_{k+1} = r_k - \alpha_k A \cdot p_k$$

if $||r_{k+1}|| < \varepsilon$ **then declare victory**

$$\beta_k = \frac{r_{k+1}^T \cdot r_{k+1}}{r_k^T \cdot r_k}; \qquad p_{k+1} = r_{k+1} + \beta_k p_k$$

end for

Conjugate gradient algorithm — observations

Solving $Ax = \Phi$ where $A = M^\dagger M$ is positive-definite hermitian

Main computational step is $A \cdot p_k \sim A^k \cdot \Phi$
matrix-vector operator ('matvec') must be efficiently data-parallel

Builds **Krylov subspaces** $\mathcal{K}_k(A, \Phi) \equiv \text{span}\left\{\Phi,\ A \cdot \Phi,\ A^2 \cdot \Phi,\ \cdots,\ A^{k-1}\Phi\right\}$

Each x_k is true solution x projected onto $\mathcal{K}_k$
$\longrightarrow$ when $k = \sharp_F$ guaranteed to find exact solution assuming exact arithmetic

Aside (1): Stabilized biconjugate gradient algorithm ('BiCGstab')
generalizes to non-symmetric matrices, competitive with $M^{-1} = \left(M^\dagger M\right)^{-1} M^\dagger$

Conjugate gradient algorithm — observations

Solving $Ax = \Phi$ where $A = M^\dagger M$ is positive-definite hermitian

Main computational step is $A \cdot p_k \sim A^k \cdot \Phi$
matrix-vector operator ('matvec') must be efficiently data-parallel

Builds **Krylov subspaces** $\mathcal{K}_k(A, \Phi) \equiv \text{span}\left\{\Phi,\ A \cdot \Phi,\ A^2 \cdot \Phi,\ \cdots,\ A^{k-1}\Phi\right\}$

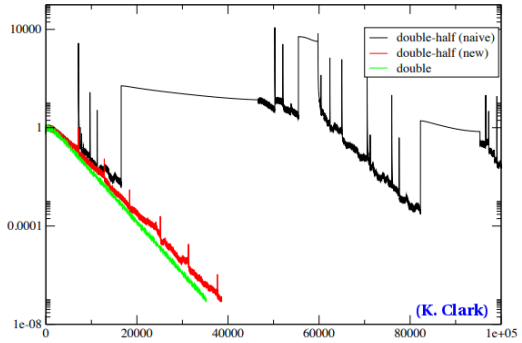
Each x_k is true solution x projected onto $\mathcal{K}_k$
$\longrightarrow$ when $k = \sharp_F$ guaranteed to find exact solution assuming exact arithmetic

Aside (2): Local $M^\dagger M \longrightarrow \mathcal{O}(L)$ iterations needed
for information to propagate across L^D lattice

Conjugate gradient algorithm — convergence

When $k = \sharp_F$ guaranteed to find exact solution assuming exact arithmetic

Inexact computer arithmetic $\longrightarrow$ round-off errors accumulate
when summing $\sharp_F$ small numbers to compute $||r_k||$

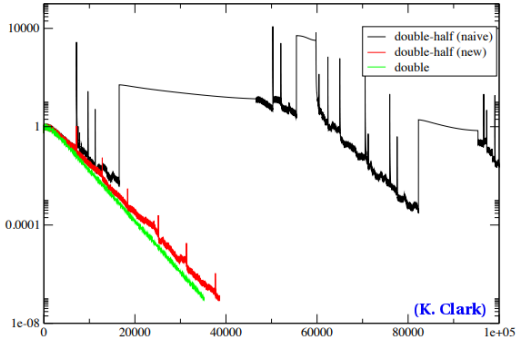


Conjugate gradient algorithm — convergence

Occasionally restarting algorithm can improve convergence

[arXiv:0911.3191]

Even better, doing matvec in lower (32- or 16-bit) precision $\longrightarrow$ $\sim$3–5$\times$ speed-up



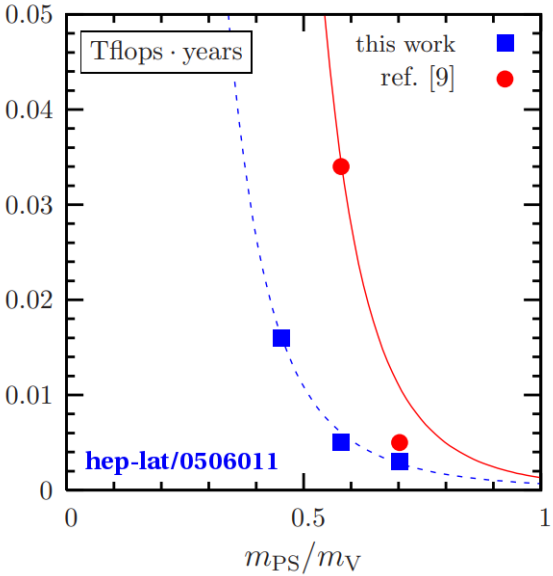
Chiral slowing down

$\sim$3–5$\times$ speed-up
has limited impact as $m \to 0$

Increasing condition number
analogous to critical fluctuations

Analog of cluster algorithm
$\longrightarrow$ **multigrid** inverter

Solve different energy ranges
using different (smaller!) lattices



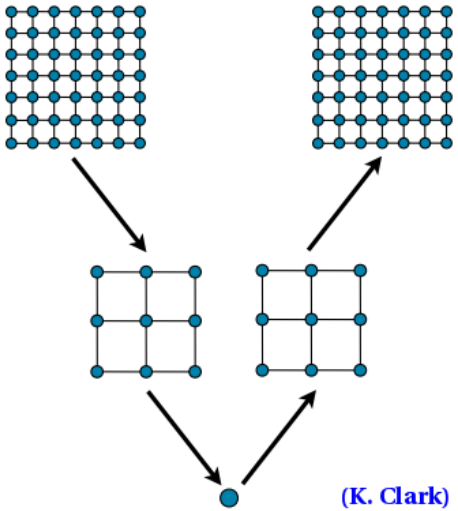
Chiral slowing down

$\sim$3–5$\times$ speed-up
has limited impact as $m \to 0$

Increasing condition number
analogous to critical fluctuations

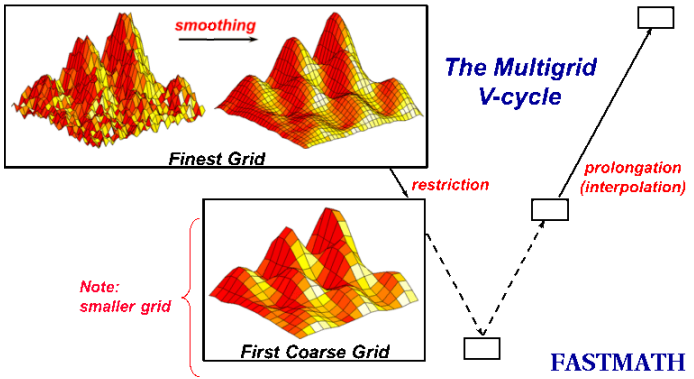
Analog of cluster algorithm
$\longrightarrow$ **multigrid** inverter

Solve different energy ranges
using different (smaller!) lattices



Multigrid inversion algorithms

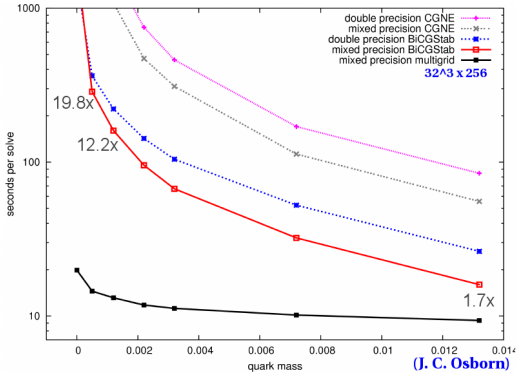
Non-trivial procedure requires ‘adaptive’ projection 1991–2007
depending on low-energy eigenvectors (‘low modes’)



Caveats: Sensitive to fermion discretization; substantial setup costs

Multigrid inversion algorithms

Non-trivial procedure requires ‘adaptive’ projection 1991–2007
depending on low-energy eigenvectors (‘low modes’)



Caveats: Sensitive to fermion discretization; substantial setup costs

Aside: Iterative eigensolvers

Full diagonalization again costs $\propto \sharp_F^3$

Often interested only in extremal eigenvalues and eigenvectors

- Low modes for adaptive multigrid [or 'deflation' algorithms]
- Extremal eigenvalues to verify RHMC rational approximation
- Physics!

Iterative Krylov-subspace methods [Lanczos, Arnoldi]
and non-Krylov methods [Davidson] on the market

PReconditioned Iterative MultiMethod Eigensolver (PRIMME)
is extensive, efficient, established library

Exercises (5)

You will write and test a conjugate gradient inverter

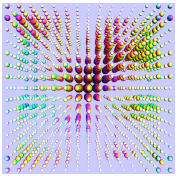
[optional template code available — model solution code to come later in the school]

Several small exercises:

- Monitor residual magnitude to check convergence
- Check convergence dependence on stopping condition
- Check convergence dependence on different volumes and geometries

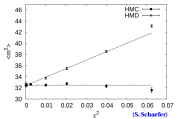
Checkpoint

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



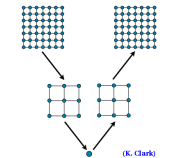
- ✓ Importance sampling and Markov chains
- ✓ Hybrid Monte Carlo and friends
- ✓ Iterative algorithms for fermions — Questions?

[Time permitting] Alternatives to address sign problems



Interaction encouraged

“It’s better to uncover a little than to cover a lot” (V. Weisskopf)



Sign problems and phase reweighting

Suppose we have a complex fermion determinant $\det M = e^{i\alpha} |\det M|$

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathcal{D}U \;\; \mathcal{O} \;\; e^{-S_G} \det M = \frac{\int \mathcal{D}U \;\; \mathcal{O} \; e^{i\alpha} \;\; e^{-S_G} \, |\det M|}{\int \mathcal{D}U \; e^{i\alpha} \;\; e^{-S_G} \, |\det M|} = \frac{\langle \mathcal{O} \; e^{i\alpha} \rangle_{\text{pq}}}{\langle e^{i\alpha} \rangle_{\text{pq}}}$$

$\langle \cdot \rangle_{\text{pq}}$ is computed by importance sampling

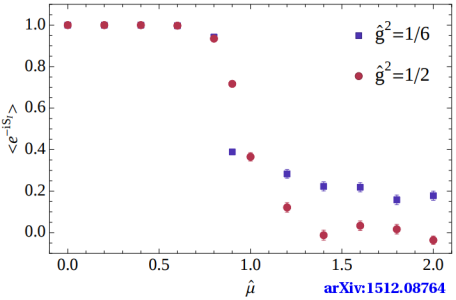
using well-behaved 'phase-quenched' $e^{-S_G} \, |\det M|$

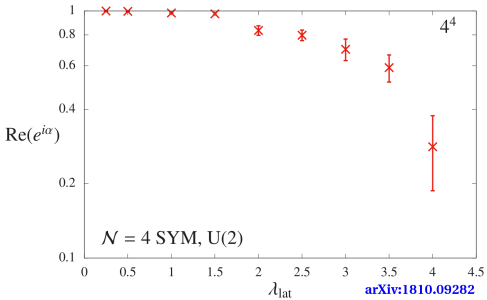
$\langle e^{i\alpha} \rangle_{\text{pq}} = \dfrac{\mathcal{Z}}{\mathcal{Z}_{\text{pq}}}$ quantifies severity of sign problem

and viability of this 'phase reweighting'

Sign problems and phase reweighting

$\left\langle e^{i\alpha} \right\rangle_{\text{pq}} = \dfrac{\mathcal{Z}}{\mathcal{Z}_{\text{pq}}}$ quantifies severity of sign problem & viability of phase reweighting



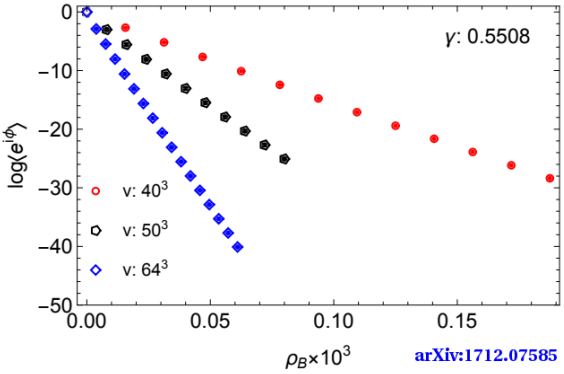


Reweighting is more general procedure that often suffers from 'overlap problem'

$\longrightarrow$ instabilities due to large fluctuations in weights

Expect sign problems to be exponentially severe

$$\left\langle e^{i\alpha} \right\rangle_{\mathrm{pq}} = \frac{\mathcal{Z}}{\mathcal{Z}_{\mathrm{pq}}} = \exp\left[-(f - f_{\mathrm{pq}})\, V\right] \propto e^{-V} \quad \text{where free energy density} \quad f = -\tfrac{1}{V} \log \mathcal{Z}$$



$\longrightarrow$ let's consider (some) alternatives to importance sampling

Complex Langevin

For real action S, stochastic process from Langevin equation
proven to produce correct $\langle \mathcal{O} \rangle$ — "stochastic quantization"

$$\frac{d\Phi}{d\tau} = -\frac{\partial S}{\partial \Phi} + \eta(\tau)$$

τ is 'Langevin time' and $\eta(\tau)$ is gaussian white noise $\langle \eta(\tau_1)\eta(\tau_2) \rangle = 2\delta(\tau_1 - \tau_2)$

Proposal: Complexify integration variables $z = \Phi + i\Theta$ [$\eta(\tau)$ still real]
with complex action $S_C(z) = S_R(z) + iS_I(z)$

$$\frac{dz}{d\tau} = -\frac{\partial S_C}{\partial z} + \eta(\tau)$$

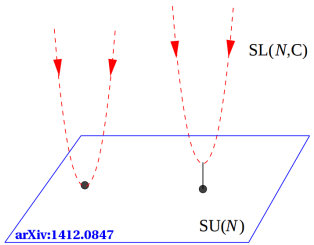
Proof of correctness no longer holds, but method may yet work

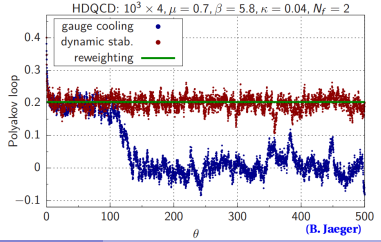
Complex Langevin

Complex integration variables $z = \Phi + i\Theta$ and action $S_C(z) = S_R(z) + iS_I(z)$
enlarges $\mathrm{SU}(N) \longrightarrow \mathrm{SL}(N, \mathbb{C})$ gauge invariance

Historically, complex Langevin could happily converge to wrong results

Found to be related to system drifting away from SU(N)
$\longrightarrow$ ongoing development of diagnostics and stabilization procedures



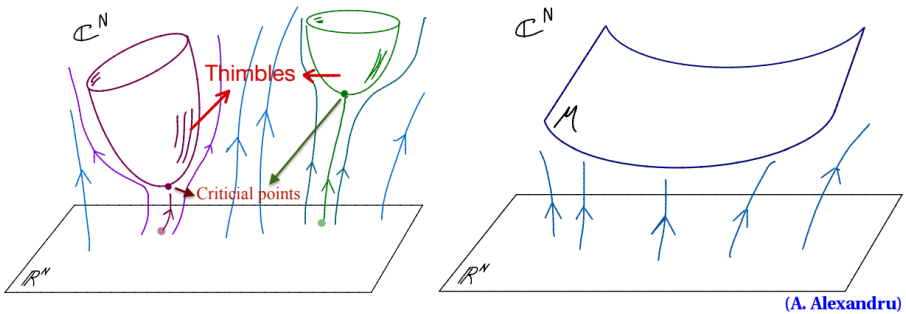


Lefschetz thimbles

Keep complex z and $S_C(z)$, but not stochastic process

Instead restrict integration to $\sharp_F$-dim. manifold in $2\sharp_F$-dim. complex hyper-plane

Steepest descent $\dfrac{dz}{d\tau} = -\overline{\dfrac{\partial S_C}{\partial z}}$ from $\dfrac{dS_C}{dz} = 0$ critical points $\longrightarrow$ **thimbles**

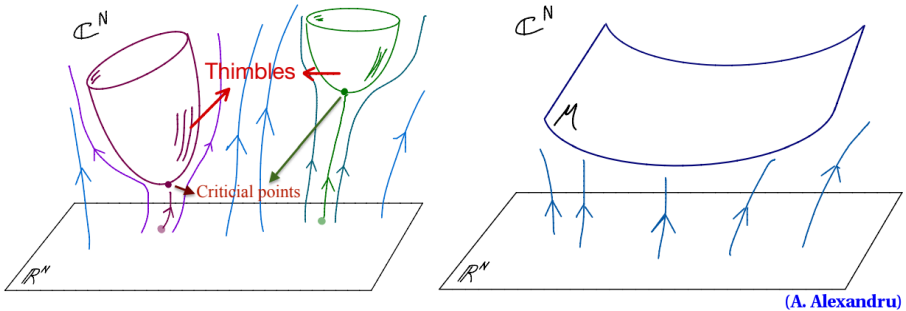


(A. Alexandru)

Lefschetz thimbles

Steepest descent $\frac{dz}{d\tau} = -\overline{\frac{\partial S_C}{\partial z}} \longrightarrow$ constant S_I allows importance sampling

Hard to correctly combine multiple thimbles, or find them in the first place



Generalized integration manifolds

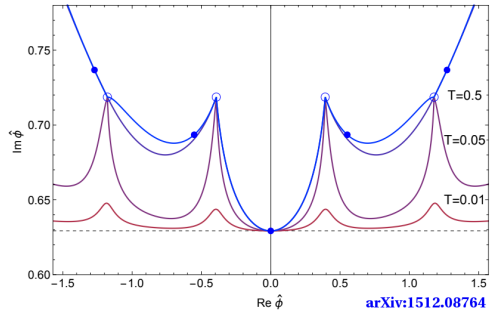
Hard to correctly combine multiple thimbles, or find them in the first place

Thimbles found by $\tau \to \infty$ 'flow'

$$\frac{dz}{d\tau} = +\overline{\frac{\partial S_C}{\partial z}}$$

Small-τ flow already stabilizes $\left\langle e^{i\alpha} \right\rangle_{\text{pq}}$

$\longrightarrow$ phase reweighting



Other methods (e.g., machine learning) can search for 'sign-optimized' manifolds

Density of states

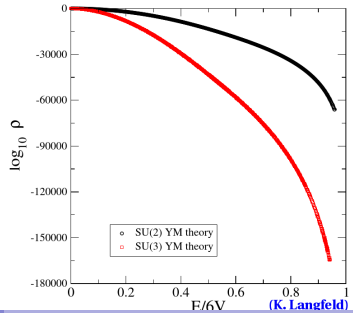
$$\langle \mathcal{O} \rangle \;=\; \frac{1}{\mathcal{Z}} \int \mathcal{D}\Phi \;\; \mathcal{O}(\Phi) \;\; e^{-S[\Phi]} \;=\; \frac{1}{\mathcal{Z}} \int dE \; \mathcal{O}(E) \; \rho(E) \;\; e^{-E}$$

One-dim. integral involving **density of states** $\rho(E) = \int \mathcal{D}\Phi \;\; \delta\left(S[\Phi] - E\right)$

$\sim$ volume of constant-E hyper-surface

Need ρ across **many** orders of magnitude

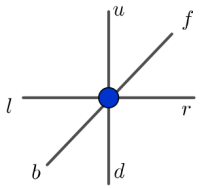
Exponential error suppression
from LLR algorithm



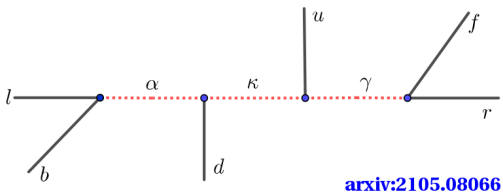
Tensor networks

Construct states in Hilbert space as contractions over many few-index tensors

Partition function as tensor trace $\mathcal{Z} = \mathrm{tTr}\left(\prod_{j=1}^{V} T^{(j)}_{lrudfb}\right) \longrightarrow$ expectation values



T_{lrudfb}



$A_{lb\alpha}B_{\alpha d\kappa}C_{\kappa u\gamma}D_{\gamma fr}$

arxiv:2105.08066

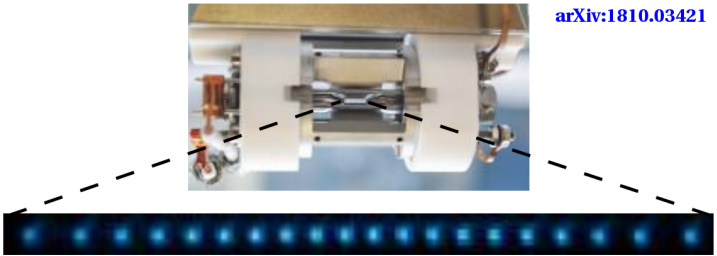
Many algorithms, applications, challenges — tensors.net, tensornetwork.org, etc.

Quantum computing

‘Real-time’ quantum dynamics built into computing device itself
[either digital or analog]

Quantum error correction will be key to scaling up
$\longrightarrow$ requires many high-quality physical qubits

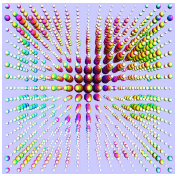
Much more coming later in this school…



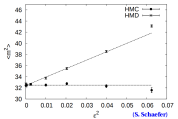
arXiv:1810.03421

Wrap up

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

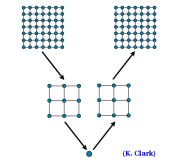


- ✓ Importance sampling and Markov chains
- ✓ Hybrid Monte Carlo and friends
- ✓ Iterative algorithms for fermions
- ✓ Alternatives to address sign problem — Questions?



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