# Numerical Methods for Lattice QCD I

Stefan Schaefer

John von Neumann Institute for Computing, DESY

VII Parma International School of Theoretical Physics

# Numerical Methods for Lattice QCD

**Definition**

Using a discrete space-time QCD can be defined.

→ Del Debbio's lectures

**Computational method**

Path integral → high dimensional integral

Computation of integrals by Monte Carlo.

E.g. hadron masses, decay constants, . . .

**Algorithms**

Many calculations computationally very expensive.

Massively parallel computers employed.

Need to work on simulation setup, algorithms,. . .

Many choices.

Physics understanding ↔ Numerical setup.

# Algorithms in Lattice QCD

**Field update**

Integral over field variables → **sum** over field configurations

How to efficiently generate these fields.

Transformation of action into form amenable
to numerical treatment.

Numerical solution of differential eq.

**Solution of Dirac equation**

Needed in field update and fermionic observables.

Out-of-the-box algorithms, e.g., the conjugate gradient, perform badly
→ Need to take physics into account

# Outline

**Monday**

Introduction

Markov Chain Monte Carlo

The HMC algorithm

**Tuesday**

MD integrators

Fermions in QCD simulations

Mass preconditioning

Other fermion methods (RHMC, DD-HMC)

# Outline

**Wednesday**

Solution of the Dirac equation

Preconditioning

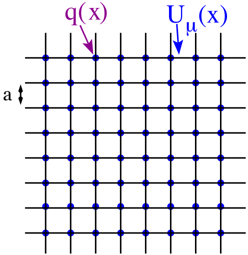
Local deflation

**Thursday**
Methods to compute hadron observables

Autocorrelations

**General reading**
M. Lüscher
Computational strategies in lattice QCD
Les Houches 2009, `arXiv:1002.4232`

# Goal



**Computation of path integral**

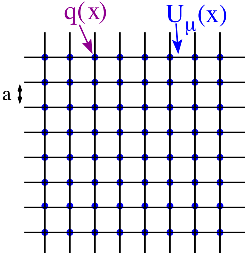
$$\langle A \rangle = \frac{1}{Z} \int \prod_{x,\mu} dU_{x,\mu} e^{-S[U]}\, A[U]$$

with

$$Z = \prod_{x,\mu} dU_{x,\mu} e^{-S[U]}$$

One SU$(3)$ integration variable for each link.

# Goal



$$\langle A \rangle = \frac{1}{Z} \int \prod_{x,\mu} dU_{x,\mu} e^{-S[U]}\, A[U]$$

One $\mathrm{SU}(3)$ integration variable for each link.

$128 \times 64^3$ lattice → $1.3 \cdot 10^8$ links

Classical numerical quadrature would need $N^{\text{\#variables}}$ function evaluations

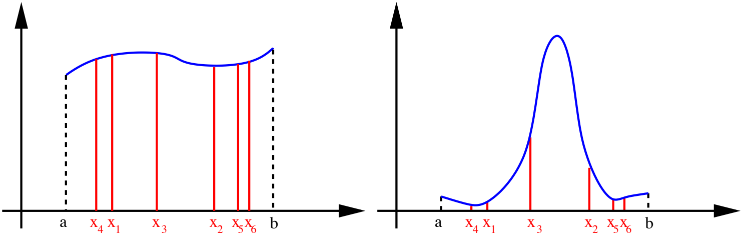
# Monte Carlo

General idea of Monte-Carlo integration

$$\frac{1}{b-a}\int_a^b \mathrm{d}x\, f(x) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

with randomly chosen points $x_i$ in the integration region

- good idea, if $f(x)$ approximately constant
  $\Rightarrow$ small fluctuations in $f(x_i)$.
- For a given realization of the $N$ points $x_i$, this is an **unbiased estimator** of the integral



# Error of a MC simulation

$$\tilde{F}_j = \frac{1}{N}\sum_{i=1}^{N} f(x_i^j)$$

Index $j$ labels the repetition of the "experiment".

Unbiased = gives correct result *on average*

$$F = \langle\!\langle \tilde{F} \rangle\!\rangle$$

$\langle\!\langle \cdot \rangle\!\rangle$ average over realizations of the $x_i$

Typical deviation →variance *of this estimator*.

$$\begin{aligned}\langle\!\langle (\tilde{F} - F)^2 \rangle\!\rangle &= \frac{1}{N^2}\sum_{ik} \langle\!\langle \tilde{F}_i \tilde{F}_k \rangle\!\rangle - F^2 \\ &= \frac{1}{N}(\sum_i \langle\!\langle \tilde{F}_i^2 \rangle\!\rangle - F^2) = \mathrm{var}(F)/N\end{aligned}$$

Error decreases as $1/\sqrt{N}$

# Note of caution

This theoretical analysis assumes knowlege of two quanitites

$$\bar{F} = \int dx\, f(x) \qquad \text{and} \qquad \mathrm{var(f)} = \int dx\, (f(x) - \bar{F})^2$$

What you get from the Monte Carlo are **estimators** of these quantities.

The analysis is correct for $N \to \infty$; $\infty$ is a large number.

These estimators might have significant errors, which are hard to get from the MC. You might also just have been unlucky.

**To a certain extend, practical Monte Carlo is an art and requires careful inspection of the results.**

# Importance sampling

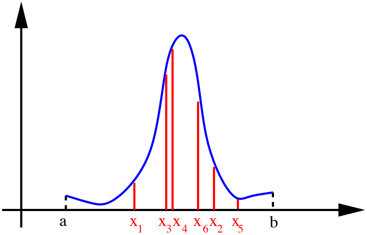
Estimator correct up to $\sqrt{\mathrm{var}(f)/N}$ → **reduce variance**.

$$\frac{1}{\Delta}\int_a^b \mathrm{d}x\, f(x) = \frac{1}{\Delta}\int_a^b P(x)\mathrm{d}x\,\frac{f(x)}{P(x)} = \left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{P(x_i)}\right](1+\mathcal{O}(N^{-1/2}))$$

with points $x_i$ **chosen according to** $P$. $\Delta = b - a$

Choose points accoring to probablity distribution similar to function to be integrated

Optimal, if distribution $\propto |f(x)|$ . . . need $\int dx|f(x)|$



# Markov Chain Monte Carlo

Problem of "straight" Monte Carlo is to find a *normalized* probability density $P(x)$

**Solution**

Use a method which only needs **relative** probabilites

Construct a sequence of points

$$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \cdots \rightarrow x_N$$

using a transition probability $T(x_{i+1} \leftarrow x_i)$

Analysis using arguments of a $N$ sets of such chains.

# Properties of $T$

For any given pair of points $x_1$ and $x_2$

$$T(x_2 \leftarrow x_1)$$

with the following properties

(A) **Stability**

$P(x') = \int dx\, T(x' \leftarrow x)\, P(x)$

(B) **Normalization**

$\int dx'\, T(x' \leftarrow x) = 1$

(C) **Ergodicity**

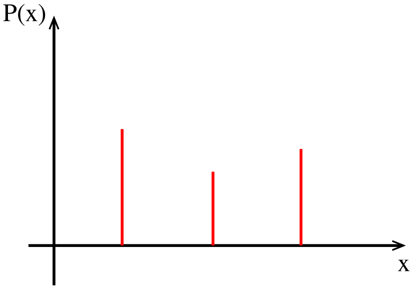
$T(x' \leftarrow x) > 0$ for each pair $x, x'$

**No reference to absolute normalization of $P$**

# Analysis of MCMC

**For the sake of simplicity, consider discrete state space.**
Integrals → sums.

Example for a single variable with three possible values:



$T(x' \leftarrow x)$ is a matrix acting in the space of states.
For the example a $3 \times 3$ matrix, because $x$ can take 3 values.

A probability distribution is a normalized vector in this space.

# Analysis of MCMC

Imagine having an ensemble of points $x_i$ distributed according to $P(x)$.

Condition (A) reads

$$P = T P$$

→ $P$ is eigenvector of $T$ with eigenvalue 1.

---

**Theorem of Frobenius–Perron**

For a matrix with the properties A–C the following holds

- There is exactly one eigenvalue $\lambda = 1$.
- All eigenvalues $\lambda$ have $|\lambda| \leq 1$.

# Convergence of Markov Process

Given any starting distribution $P_0$ repeated application of $T$ leads to exponential convergence to desired distribution

$$\begin{aligned} T^n P_0 &= \sum_{i=0}^{N} \lambda_i^n \left(\Psi_i, P_0\right) \Psi_i \\ &= \sum_{i=0}^{N} e^{\log\,|\lambda_i|\,n} \left(\Psi_i, P_0\right) \frac{\lambda_i}{|\lambda_i|} \Psi_i \\ &\propto P + \mathrm{O}(e^{-n/\tau_1}) \end{aligned}$$

$\tau_i = 1/\log|\lambda_i|$ are the exponential autocorrelation times

A nicer interpretation in terms of single exponentials can be given with detailed balance, see later.

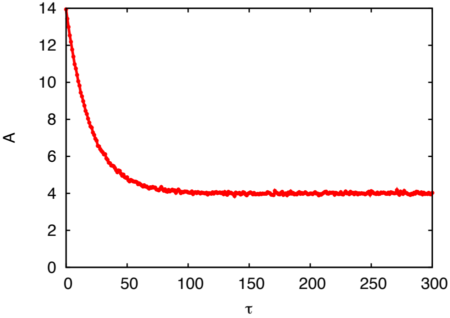
# Practical MCMC

Start with one (or a few) points

**Distribution/evolution of $x$ $\rightarrow$ distribution/evolution of $f(x)$**

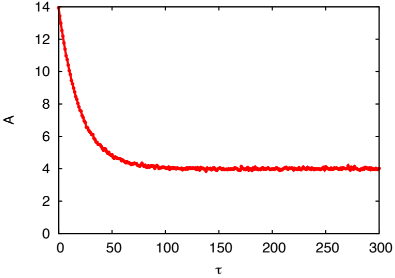
Averages of Monte Carlo time

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$



Cut away first *thermalization* phase.

# Time evolution



$$\langle f \rangle = \frac{1}{N} \sum_{t=1}^{N} f(x_i)(1 + \mathcal{O}(1/\sqrt{N}))$$

At the beginning large contributions from $\Psi_{i>0}$.

Formally $1/N$, but large coefficient, best discarded.

A valid simulation must have $n \gg 1/\tau_1$
needs to be determined from the simulation

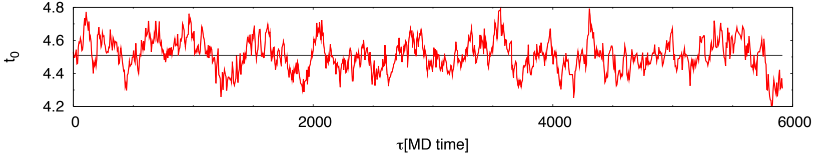
# Autocorrelations

There is a price to pay:
Subsequent points $x_i$/configurations are in general not independent.

$$\langle\langle f(x_i) f(x_{i+1})\rangle\rangle \neq 0$$

This needs to be taken into account in the *analysis* of the data.

It also can make simulations exceedingly expensive.



# Construction of $T$

Constructing a valid $T$ seems a daunting task.
**Metropolis–Hastings method** Metropolis et al'53

- Symmetric proposal (trans. prob. w/ $P_0(x)$ =const)

$$W(x' \leftarrow x) = W(x \leftarrow x')$$

- Acceptance step

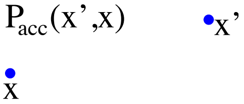
$$P_{\rm acc}(x', x) = \min[1, \frac{P(x')}{P(x)}]$$

Next point is $x'$ with propability $P_{\rm acc}$, else $x$

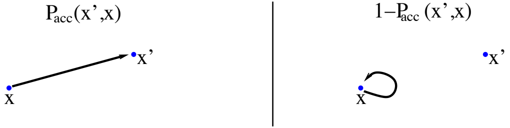
# Metropolis

**Start at point $x$**

**Pick new point with probability $W(x', x)$**



**Compute acceptance probability of proposal $x'$**
**Make a decision where to continue**



$$T(x' \leftarrow x) = W(x' \leftarrow x)P_{\mathrm{acc}}(x', x) + \delta_{xx'} \sum_{\tilde{x}} (1 - P_{\mathrm{acc}}(\tilde{x}, x))W(\tilde{x} \leftarrow x)$$

# Proof

$$T(x' \leftarrow x) = W(x' \leftarrow x)P_{\text{acc}}(x', x) + \delta_{xx'}[\sum_{\tilde{x}}(1 - P_{\text{acc}}(\tilde{x}, x))W(\tilde{x} \leftarrow x)]$$

(A) **Stability**

$$\begin{aligned}
&\sum_x P(x)\, T(x' \leftarrow x) \\
&= \sum_x W(x' \leftarrow x)P(x)P_{\text{acc}}(x', x) + P(x') \sum_{\tilde{x}} \big([1 - P_{\text{acc}}(\tilde{x}, x)]W(\tilde{x} \leftarrow x)\big) \\
&= \sum_x W(x \leftarrow x')P(x')P_{\text{acc}}(x, x') + P(x')[1 - \sum_{\tilde{x}} P_{\text{acc}}(\tilde{x}, x)]W(\tilde{x} \leftarrow x) = P(x')
\end{aligned}$$

(B) **Normalization**

$$\sum_{x'} T(x' \leftarrow x) = 1$$

# Practical implementation

We want to generate the sequence

$$x_1 \to x_2 \to x_3 \to \cdots$$

and arrived at $x_j$

- Make a proposal $y$ according to probability $W(y \leftarrow x_j)$

- Compute $P_{\mathrm{acc}}$ and draw random number $0 \leq r < 1$

$$x_{j+1} = \begin{cases} y & \text{if } \; r < P_{\mathrm{acc}} \\ x_j & \text{else} \end{cases}$$

Good proposal

- Easy to generate
- High acceptance propability

# Generic issues of the method

The step size can be rather limited

**Small steps**

$P(x') \approx P(x) \rightarrow P_{\mathrm{acc}}$ high
But it takes many steps to sample the whole integration space
Large autocorrelations

**Large steps**

Small autocorrelations
Feasible only if an update leading to reasonable $P_{\mathrm{acc}}$ can be found
Can be difficult to compute proposal.

In high dimensional spaces, it can be very difficult to argue, what “large step” and “small step” even mean.

# Summary of part I

Monte Carlo is a method to numerically compute high-dimensional integrals.

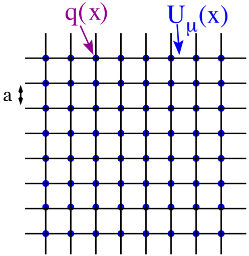
Integral → average over sample points.

Sample points need to be chosen in region of high probability.

Problem of constructing a normalized distribution
→ Markov Chain Monte Carlo

$+$ No normalized distribution needed.
$-$ Deal with autocorrelations.

# Back to QCD



One point $x$
→ one value for each link variable
→ one field configuration

Instead of $x$ we will therefore now use $U$ which are in SU$(3)$.

# QCD

$$\langle A \rangle = \frac{1}{Z} \int \prod_{x,\mu} dU_{x,\mu} e^{-S[U]} \, A[U]$$

with

$$Z = \prod_{x,\mu} dU_{x,\mu} e^{-S[U]}$$

Normalized probability density

$$P[U] = \frac{1}{Z} e^{-S[U]}$$

Ratios in probabilities
$\rightarrow$
Need to evaluate *differences* in the action $S[U]$
**While this choice of $P[U]$ seems natural, it is not unique**

# QCD

Basically two types of algorithms

**Single link updates**

Of the $4V$ links, only one is changed at a time.

In each step $S[U] - S[U']$
Possible if this is an $\mathrm{O}(1)$ operation.

With dynamical fermions, estimating change in action is a global $\mathrm{O}(V)$ operation

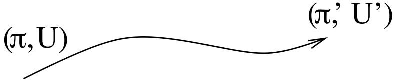
Method of choice in pure gauge theory.
No systematic study.

# QCD

Basically two types of algorithms

**Updates based on Molecular Dynamics**

Based on ideas from classical mechanics.



Field configuration position
Introduce momenta → equations of motion.

Updates keep propbability constant (micro-canonical)

Solves problem of finding a good proposal in Metropolis-Hastings procedure

# Hybrid Monte Carlo

**Extended field space**

$$Z = \int [dU][d\pi]\, e^{-\frac{1}{2}(\pi,\pi)-S[U]}$$

Expectation values of observables $A[U]$ remain the same.

Momenta $\pi = \pi^a T^a \in \mathbf{su}(N)$, $\pi^a \in \mathbb{R}$

$$(\pi,\pi) = \sum_{x,\mu} \pi^a_{x,\mu}\pi^a_{x,\mu}$$

**Updates**

Make updates in this extended phase space.
$\Rightarrow$ updates for $U$ fields.

# Molecular dynamics

Essential update step for the gauge fields: $(\pi, U) \to (\pi', U')$

Simulation time $\tau$.

## Hamilton's equations of motion

Hamiltonian

$$H[\pi, U] = \frac{1}{2}(\pi, \pi) + S[U]$$

E.o.m.

$$\dot{U}_{x,\mu} = \pi_{x,\mu} U_{x,\mu}$$

$$\dot{\pi}_{x,\mu} = -F_{x,\mu} \ , \qquad F^a_{x,\mu} = \frac{\partial S(e^\omega U)}{\partial \omega^a(x,\mu)}$$

# Molecular dynamics

By **Liouville's theorem**, the classical dynamics

$$(\pi, U) \to (\pi', U')$$

maps areas of equal likelihood into eachother.

*Energy conservation*

$$\frac{d}{d\tau} H = 0$$

Boltzmann factor $e^{-H}$ is constant.

*Conservation of phase space*

**An exact solution of the E.o.m is a valid update.**

Fundamentally different from Metropolis–Hastings

In practice: integration errors

# HMC

## Momentum Heatbath

Refresh momenta $\pi$ (Gaussian random numbers)

## Molecular Dynamics

Solve numerically MD equations for some MC time $\tau$ (trajectory) deriving from Hamiltonian $H = \frac{1}{2}(\pi, \pi) + S[U]$.

## Acceptance Step

Correcting for inaccuracies in integration.

# Metropolis

Different viewpoint:

The molecular dynamics as the symmetric proposal in Metropolis.

This solves the problem of the inexact integration.

Need a **symplectic integrator**, i.e. area conserving and reversible.

Reversibility haunts computer implementations.
No good theory for this.

# Metropolis

## Acceptance step

Molecular dynamics $(\pi, U) \to (\bar{\pi}, \bar{U})$

$$P_{\text{acc}} = \min(1, e^{-(H(\bar{\pi},\bar{U})-H(\pi,U))})$$

Exact solution of MD equations has $\Delta H = 0$
$\Rightarrow$ always accepted

$\Delta H$ needs to be $\mathrm{O}(1)$ for good acceptance.
Difficult to achieve on large volume.

# Updates

$$Z = \int [dU][d\pi]\, e^{-\frac{1}{2}(\pi,\pi)-S[U]}$$

## Momenta: Heatbath

- $(\pi,\pi) = \sum_{x,\mu} |\pi^a_{x,\mu}|^2$
  $\pi^a_{x,\mu}$ are Gaussian random numbers.
- Normalization is known.

Just for $\pi$ this is the optimal update
no correlation to previous config.
Also true for combined system? (Kramers’ rule,. . . )

# Comments

**Momentum heat-bath is the only source of randomness.**

Makes algorithm ergodic.

Problems with ergodicity from $S = \infty$ surfaces.

---

The original hope was that a trajectory consititutes a macroscopic update.

Free field theory

$$\ddot{U} = \dot{\pi} = \frac{\delta S}{\delta U}$$

$\rightarrow$ Monte Carlo Time $\tau \propto 1/a$

In the classic algorithm trajectory length is scaled with $1/a$.

In number of updates, autocorrelations should stay the same.

M.Lüscher, S.S.' 11

This does not apply in interacting theory

Virotta,Sommer, S.S.' 10

Still, rather long trajectories are a good idea, $\tau \sim 2$.

---

For exercises in $\phi^4$ theory:

S.S., Les Houches 2009, available at NIC@DESY website.

# Summary

Markov Chain Monte Carlo provides general purpose framework for importance sampling—no normalization needed in distribution

Choice of transition probablity

Metropolis = (symmetric proposal) × (acceptance step)

Molecular dynamics “swiss army knife” for continuous variables

Link updates perform better(?)

# Numerical Methods for Lattice QCD II

Stefan Schaefer

John von Neumann Institute for Computing, DESY

VII Parma International School of Theoretical Physics

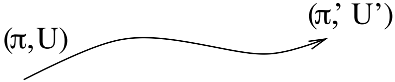
# HMC

### Momentum Heatbath

Refresh momenta $\pi$ (Gaussian random numbers)

### Molecular Dynamics

Solve numerically MD equations for some MC time $\tau$ (trajectory) deriving from Hamiltonian $H = \frac{1}{2}(\pi,\pi) + S[U]$.



### Acceptance Step

Correcting for inaccuracies in integration.

# Numerical integration

$$H = \frac{1}{2}(\pi, \pi) + S[U] = T + S$$

$$\dot{U}_{x,\mu} = \pi_{x,\mu} U_{x,\mu} \qquad\qquad \dot{\pi}_{x,\mu} = -F_{x,\mu}$$

**Splitting methods**

Eom for each part $T$, $S$ can be solved exactly
$\rightarrow$ symplectic

$T$ defines $T_U$

$$U_{x,\mu}(\tau) = e^{\pi\tau} U_{x,\mu}(0), \quad \pi(\tau) = \pi(0)$$

$S$ defines $T_p$

$$U_{x,\mu}(\tau) = U_{x,\mu}(0), \quad \pi(\tau) = \pi(0) - \tau F$$

# Splitting methods

$$
\begin{aligned}
T_U = e^{\epsilon\hat{T}}: &\qquad U_{x,\mu}(\epsilon) = e^{\pi\epsilon}U_{x,\mu}(0), &\qquad \pi(\epsilon) = \pi(0)\\
T_p = e^{\epsilon\hat{S}}: &\qquad U_{x,\mu}(\epsilon) = U_{x,\mu}(0), &\qquad \pi(\epsilon) = \pi(0) - \epsilon F
\end{aligned}
$$

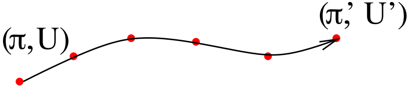
Can be put together in any order.

**Legal integrator**
**Time steps of $T_U$ and $T_p$ sum up to $1$.**

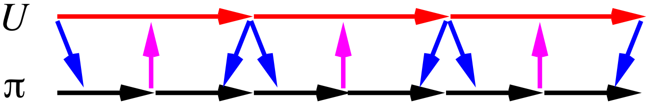
**Symmetric integrator**
$\rightarrow$**Integration error automatically $\mathrm{O}(\epsilon^2)$**

# Example: Leapfrog / Verlet integrator

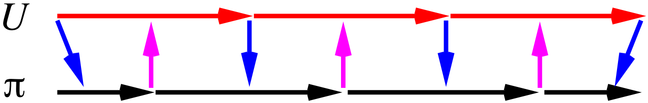
Divide trajectory in $N$ steps of size $\epsilon = \tau/N$



$$T = (T_U(\epsilon/2)T_p(\epsilon)T_U(\epsilon/2))^N$$



You can combine the half steps → save computation



# Omelyan & Co

Leapfrog has been long-time workhorse
Robust, but in general not optimal.

Easy improvement without detailed knowledge of physics system.

Seminal paper `Omelyan, Mrygold, Folk, 2003`

Introduce *reduandant* parameters and optimize

$$T = [T_p(\epsilon\lambda)T_U(\epsilon/2)T_p(\epsilon(1-2\lambda)T_U(\epsilon/2)T_p(\epsilon\lambda)]^{N/2}$$

$\lambda = 0.19$ performs roughly $2\times$ better than leapfrog.

The paper contains O(100) integrators.

# Optimizing integrators

Exact time evolution operator

$$e^{\tau\frac{d}{dt}} = e^{\tau\hat{H}} \qquad \text{with} \qquad \hat{H} = -\frac{\delta S}{\delta U}\frac{\partial}{\partial U} - \frac{\delta T}{\delta\pi}\frac{\partial}{\partial\pi} = \hat{S} + \hat{T}$$

with $T(\pi) = (\pi, \pi)$ and $S[U]$ the action.
$\hat{H}$ is the Hamiltonian vector field.

Leap-frog integrator

$$\begin{aligned}
&[e^{\epsilon/2\,\hat{S}}e^{\epsilon\,\hat{T}}e^{\epsilon/2\,\hat{S}}]^{\tau/\epsilon} \\
&= \exp\{(\hat{S}+\hat{T})\epsilon - \frac{\epsilon^3}{24}([\hat{S},[\hat{S},\hat{T}]] + 2[\hat{T},[\hat{S},\hat{T}]])\}^{\tau/\epsilon} \\
&= \exp\{(\hat{S}+\hat{T})\tau - \frac{\tau\epsilon^2}{24}([\hat{S},[\hat{S},\hat{T}]] + 2[\hat{T},[\hat{S},\hat{T}]])\}
\end{aligned}$$

Baker-Campbell-Hausdorff formula has been used.
see series of paper by Clark and Kennedy

# Shadow Hamiltonian

For each symplectic integrator, there is the conserved **shadow Hamiltonian**

Can be constructed by a power series
Commutators → Poisson brackets

$$\begin{aligned}\tilde{H} &= H + \epsilon^2 (c_1 \{S, \{S, T\}\} + c_2 \{T, \{S, T\}\}) \\ &= H + \epsilon^2 (c_1 \, \partial_a S \partial_a S - c_2 \, \pi_a \pi_b \partial_a \partial_b S) \ldots\end{aligned}$$

Convergence of the series?

$c_1$ and $c_2$ depend only on the integrator

For a long time, it has been believed that what matters is the *size* of

$$\delta H = \tilde{H} - H$$

# Optimizing integrators

At the beginning of the trajectory

$$H_1 = \frac{1}{2}(\pi_1, \pi_1) + S[U_1] \qquad \tilde{H}_1 = H_1 + \delta H_1 \tag{1}$$

At the end of the trajectory

$$H_2 = \frac{1}{2}(\pi_2, \pi_2) + S[U_2] \qquad \tilde{H}_2 = H_2 + \delta H_2 \tag{2}$$

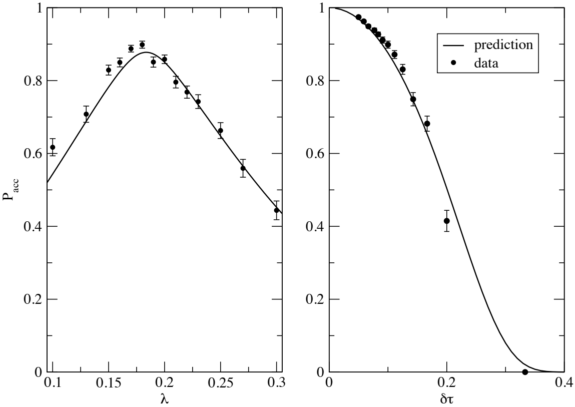
During the trajectory, $\tilde{H}$ is conserved. $\tilde{H}$=$\tilde{H}_1$=$\tilde{H}_2$

$$\Delta H = H_2 - H_1 = (H_2 - \tilde{H}) - (H_2 - \tilde{H}) = \delta H_2 - \delta H_1$$

**What matters is the fluctuation of $\delta H$.**

# Example

From Clark, Joo, Kennedy, Silva, `1108.1828`



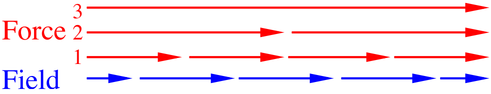
# Multiple time scales

In the HMC, different forces have vastly different size.

$$F_g \gg F_{\mathrm{ferm},UV} \gg F_{\mathrm{ferm},IR}$$

This is the oppostite ordering of the cost of their computation.

Multiple time scale integrators have been proposed.



The idea is to integrate “large forces” on a finer time scale — exacter.

# Multiple time scales

$$
\begin{aligned}
&T_\pi(\epsilon/2)T_U(\epsilon)T_\pi(\epsilon/2)\\
&\rightarrow T_{\pi,1}(\epsilon/2)\,[T_{\pi,2}(\epsilon/2m)\,T_U(\epsilon/m)\,T_{\pi,2}(\epsilon/2m)]^m\,T_{\pi,1}(\epsilon/2)
\end{aligned}
$$

Experimental finding: it never works as well as expected.

Can be understood by Shadow Hamiltonian

$$
\begin{aligned}
\tilde{H} = H + \epsilon^2[&c_1(F_1,F_1) + c_2\pi^a\pi^b S_1^{(ab)} + c_2(F_1,F_2)\\
&+\frac{1}{m^2}(c_2(F_2,F_2) + c_2\pi^a\pi^b S_2^{(ab)})]
\end{aligned}
$$

Interference term between “large” and “small” force not suppressed by relative times scale $m$.

# Summary: Integrators

Integrators have contributed to improvement in algorithms.

Typical gains are factor two.
No miracles to be expected.

Difficulty separating IR from UV.

Optimization by measurement is possible.

Fermions
Formulation of the theory

# Fermions

Textbook verions contains Grassmann fields $\psi$ and $\bar{\psi}$

$$Z = \int \prod_i d\psi_i d\bar{\psi} \prod_{i,\mu} dU_{i,\mu} e^{-S_g - \sum_f \bar{\psi}_f D(m_f) \psi_f}$$

We integrate out the fermions and get the quark determinant

$$Z = \int \prod_{i,\mu} dU_{i,\mu} \prod_f \det D(m_f) \, e^{-S_g}$$

Determinant not usable in large volume situation
→ too complicated/expensive to compute

# Fermions in simulations

Ideally, we would want to use

$$S_{\text{ferm}} = -\sum_{i=1}^{N_f} \text{tr}\,\log D(m_i) = -\sum_{i=1}^{N_f} \log\,\det D(m_i)$$

Unfortunately, the determinant of a $N \times N$ matrix is virtually impossible to compute for large $N$.

Need $\mathrm{O}(N^3)$ operations.

Large memory requirement.

Is numerically extremely unstable.

$\Rightarrow$
Need algorithm with is based on solutions of linear equations.

# Pseudofermions

## Pseudofermions

PETCHER, WEINGARTEN'81

$$\det Q^2 \propto \int [\mathrm{d}\phi][\mathrm{d}\phi^\dagger]\, e^{-(\phi,\, Q^{-2}\phi)} \,, \qquad Q = \gamma_5 D$$

Gaussian integral → apply transformation $\phi = Q\,\eta$

$$\int [\mathrm{d}\eta][\mathrm{d}\eta^\dagger]\, e^{-(\eta,\,\eta)} = \int [\mathrm{d}\phi][\mathrm{d}\phi^\dagger]\, \det{}^{-2} Q\, e^{-(\phi,\, Q^{-2}\phi)}$$

Determinant is the **Jacobian** of this transformation.

$Q$ is Hermitian

---

Generate Gaussian complex-valued quark field $\eta$

$$P[\eta] \propto e^{-(\eta,\eta)}$$

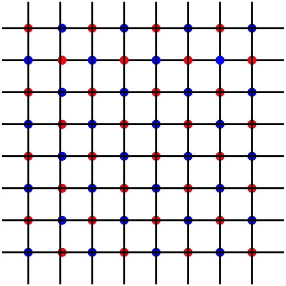
Multiply with $Q$

$$\phi = Q\eta$$

# Even–odd preconditioning

The Wilson Dirac operator connects only neighboring sites.

Label them “even” and “odd”.



$$D = \begin{pmatrix} D_{ee} & D_{eo} \\ D_{oe} & D_{oo} \end{pmatrix}$$

$D_{oo}$ and $D_{ee}$ are site-diagonal matrices.

# Even–odd preconditioning

Matrix identity

$$\begin{pmatrix} D_{ee} & D_{eo} \\ D_{oe} & D_{oo} \end{pmatrix} =$$
$$\begin{pmatrix} 1 & D_{eo}D_{oo}^{-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} (D_{ee} - D_{eo}D_{oo}^{-1}D_{oe}) & 0 \\ 0 & D_{oo} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ D_{oo}^{-1}D_{oe} & 1 \end{pmatrix}$$

For the determinant this means

$$\det D = \det D_{oo} \; \det(D_{ee} - D_{eo}D_{oo}^{-1}D_{oe}) \equiv \det D_{oo} \det \hat{D}$$

with $\hat{D}$ the Schur complement.

In the following, I will mostly write $D$ or $Q = \gamma_5 D$.
In practice, this frequently means $\hat{D}$ or $\hat{Q}$.

# Partition function

Include pseudofermions in path integral.

$$Z = \int [dU][d\pi][d\phi][d\phi^\dagger]\, e^{-\frac{1}{2}(\pi,\pi) - S_g[U] - (\phi, \frac{1}{Q^2}\phi) + 2\log \det Q_{oo}}$$

$S_g$: gauge action

effective fermion action for $N_f = 2$.

$$S_{f,\mathit{eff}} = (\phi, \frac{1}{\hat{Q}^2}\phi) - 2\log\det Q_{oo}$$

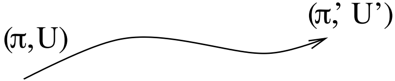
# HMC

### Momentum and pseudofermion Heatbath

Refresh momenta $\pi$
Refresh pseudofermions $\phi \to$ kept fixed during trajectory

### Molecular Dynamics

Solve numerically MD equations for some MC time $\tau$ (trajectory) deriving from Hamiltonian $H = \frac{1}{2}(\pi, \pi) + S[U]$.



### Acceptance Step

Correcting for inaccuracies in integration.

# Problems

## Pseudofermions

PETCHER, WEINGARTEN'81

$$\det \boldsymbol{Q}^2 \propto \int \mathbf{d}\phi^\dagger \mathbf{d}\phi \, e^{-(\phi,\, \boldsymbol{Q}^{-2}\phi)}$$

- Works only for pairs of degenerate flavors
  Solution: take square root → PHMC, RHMC
- Force evaluation expensive: 2 solutions of Dirac eq.

$$\boldsymbol{F}_{\rm pf} = -(\phi,\, \boldsymbol{Q}^{-2}\, \delta\boldsymbol{Q}\, \boldsymbol{Q}^{-1}\, \phi) + \text{h.c.}$$

- Seems somewhat unnatural
  Start with manifestly local action
  → quite non-local expression

# Berlin Wall

Status 2000 Quarks 16× heavier than in nature.
No perspective even with 2010 computers.

Coarse lattices $a \approx 0.1$fm
(the typical length scale is 1fm)

## Cost of a simulation (Ukawa Lattice 2001)

$$\mathrm{Cost} = C \left[\frac{\#conf}{1000}\right] \cdot \left[\frac{m_q}{16 m_{\mathrm{phys}}}\right]^{-3} \cdot \left[\frac{L}{3\mathrm{fm}}\right]^{5} \cdot \left[\frac{a}{0.1\mathrm{fm}}\right]^{-7}$$

$C \approx 2.8$ Tflops year

# Fermions

## Pseudofermions

Petcher, Weingarten'81

$$\det Q^2 \propto \int [\mathrm{d}\phi][\mathrm{d}\phi^\dagger]\, e^{-(\phi,\, Q^{-2}\phi)}$$

- **HMC + single pseudofermion action not successful**
- Compare

$$F_{\mathrm{pf}} = \delta(\phi,\, Q^{-2}\phi) \qquad \text{and} \qquad F_{\mathrm{ex}} = -\delta \mathrm{tr}\,\log\, Q^2$$

- $F_{\mathrm{pf}}$ is "stochastic estimate" of $F_{\mathrm{ex}}$
  At beginning of the trajectory $\langle F_{\mathrm{pf}}\rangle_\phi = F_{\mathrm{ex}}$
- Very large fluctuations in $F_{\mathrm{pf}}$

$$|F_{\mathrm{pf}}| \gg |F_{\mathrm{ex}}|$$

Fermions
Modifications

# Determinant Splitting

## Insight

- Need better estimate of determinant.
- Frequency splitting.

## Mass preconditioning

Hasenbusch'01, Hasenbusch,Jansen'03

$$\det Q^2 = \det \frac{Q^2}{Q^2 + \mu^2} \det(Q^2 + \mu^2)$$

- Each determinant represented by pseudo-fermion
- “Pauli-Villars” for fermion force
- more intermediate $\mu \rightarrow$ Noise reduction in force.
- success depends on choice of $\mu$. Urbach et al'04

# Numerical examples

## Action

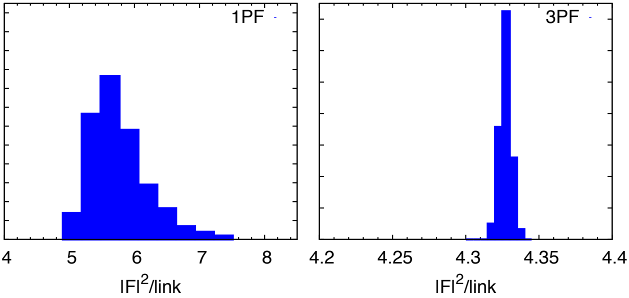
- $N_\mathrm{f} = 2 + 1$ NP improved Wilson fermions
- Iwasaki gauge action
- $64 \times 32^3$ lattice with $a = 0.09$fm
- studied extensively by PACS-CS AOKI ET AL'09,'10
- $m_\pi = 200$MeV
- $m_\pi L = 3$

## Algorithm M. LÜSCHER, S.S.'12

- Reweighting to avoid stability problems.
- Generated with public `openQCD` code.
  `http://cern.ch/luscher/openQCD`

# Effect of determinant factorization

Forces for light quark, 20 configurations. $\mu_1 = 0.05$, $\mu_2 = 0.5$



- Fluctuations of force not much reduced.
- Fluctuations in **norm** squared of force:
  Spread reduced by more than factor 100.
  (Different scale!)

# Understanding the improvement

**Framework** CLARK, JOO, KENNEDY, SILVA'11

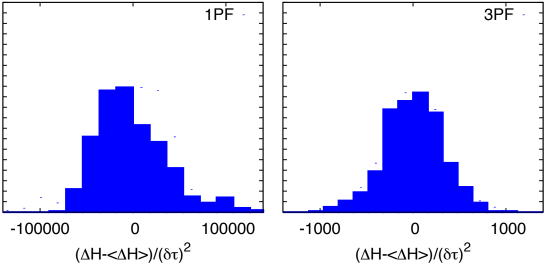
- Shadow Hamiltonian of symplectic integrators

$$\tilde{H} = H + (c_1\, \partial_a S \partial_a S - c_2\, \pi_a \pi_b \partial_a \partial_b S)\epsilon^2 + \dots$$

- Large cancellation between the two terms
  → **potential for optimization.**

- 2nd order minimum norm integrators:
  minimum of $c_1^2 + c_2^2$ Omelyan, Mrygold, Folk'03
- **Symplectic integrators profit from reduced fluctuations in norm of force.**

# Numerical examples



- $\Delta H = \tilde{H} - H$, fermions only.
- Second order min. norm Omelyan integrator.
- Much larger step-size possible.

# Factorizations

**Hasenbusch** Hasenbusch'03

$$\det Q^2 = \det \frac{Q^2}{Q^2+\mu_1^2}\;\det \frac{Q^2+\mu_1^2}{Q^2+\mu_2^2}\cdots \det(Q^2+\mu_N^2)$$

**RHMC** Kennedy, Horvath, Sint'99, Clark, Kennedy'07

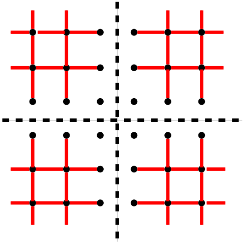
$$\det Q^2 = \prod_{i=1}^{N} \det \sqrt[N]{Q^2}$$

**Domain decomposition** Lüscher '04

$$\det Q = \det Q_{\rm block} \det R$$

# Domain decomposition

Lüscher'04



- Domain decomposition
  → Divide the lattice in blocks

$$\det D = \det D_{\text{block}} \cdot \det D_{\text{R}}$$

- Do not update links connecting blocks
  → longer autocorrelations
- Good for slow communication.

# Summary

Fermion action $-\mathbf{tr}\,\mathbf{log}D$ cannot be simulated directly.

Use pseudofermions *together* with matrix factorization

Several factorizations lead to working setups.

Need of solving the Dirac equation in each force evaluation.

# Numerical Methods for Lattice QCD III

Stefan Schaefer

John von Neumann Institute for Computing, DESY

VII Parma International School of Theoretical Physics

# Solving the Dirac equation

The solution of the Dirac equation

$$\boldsymbol{D}\,\psi = \phi$$

is the most costly part of lattice simulations including dynamical fermions.

Needed for **forces in the MD evolution**
**hadronic observables**

Dirac operator $\boldsymbol{D}$ can be viewed as a matrix acting in $\mathbb{C}^{12V}$

Large literature on **iterative solvers** for sparse linear systems

# Wilson Dirac operator

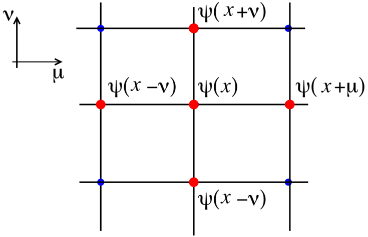
$$D_{\mathrm{W}}(m_0) = \frac{1}{2}\sum_{\mu=0}^{3}\{\gamma_\mu(\nabla_\mu^* + \nabla_\mu) - a\nabla_\mu^*\nabla_\mu\} + m_0$$

with

$$(\nabla_\mu\psi)(x) = \frac{1}{a}\left[U(x,\mu)\psi(x+a\hat{\mu}) - \psi(x)\right]$$

**Spinor fields**

$\psi_{c,d}(x) \quad \rightarrow$ `complex psi[i], i=1...12V`



$$\begin{pmatrix} \times & \times & 0 & 0 & 0 & \times \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ \times & 0 & 0 & 0 & \times & \times \end{pmatrix}$$

# The Dirac operator

Dirac operator $D$ can be viewed as a matrix acting in $\mathbb{C}^{12V}$

For **Wilson, staggered** and **domain wall** fermions this matrix is sparse.

Application of $D$ on vector scales $\propto V$.

---

Methods to solve Dirac equation based on

Matrix $\times$ vector

Itererative methods, huge applied math literature.

# Krylov space

The **Krylov space** $\mathcal{K}_n$ of order $n$ generated by a starting vector $\phi$ and a matrix $D$ is

$$\mathcal{K}_n(D,\phi) = \mathrm{span}\{\phi, D\phi, \ldots, D^{n-1}\phi\}$$

## Cayley-Hamilton theorem

Any $N \times N$ matrix $A$ satisfies its charact. polynomial $p(x)$.

$$p(x) = \det(x 1_{N\times N} - A) \qquad \Rightarrow \qquad p(A) = 0$$

Any power $A^M$ with $M > N - 1$ can be expressed trough polynomial of degree $N - 1$.

The value of a function (also the inverse) of an $N \times N$ matrix can be constructed from its powers up to $N - 1$.

$\rightarrow$ **Solution of Dirac equation is in $\mathcal{K}_{N-1}$.**

This theorem is not of much practical help if $N$ is in the millions.

# Krylov space solvers

**Use the Krylov space to iteratively construct the solution of the Dirac equation to a specified precision.**

$$\boldsymbol{D}\,\psi = \phi$$

Polynomial approx. with **coefficients depending on** $D$.

The basic problem is that it is not practical to safe the Krylov space.

- too much memory would be needed
- Only up to $\mathcal{O}(20)$ vectors can be stored

Different algorithms make different choices on how may vectors to compute before **restarting**.

$$\mathcal{K}_n(\boldsymbol{D}, \phi) \to \rho_0 \to \mathcal{K}_n(\boldsymbol{D}, \rho_0) \to \rho_1 \to \cdots$$

# Iterative improvement

$$\boldsymbol{D}\,\psi = \phi$$

For a certain approximation $\tilde{\psi}$ you can define the **residue**

$$\rho = \phi - \boldsymbol{D}\tilde{\psi}$$

This $\rho$ can be used as a new right hand side

$$\boldsymbol{D}(\psi - \tilde{\psi}) \equiv \boldsymbol{D}\psi' = \rho \quad \rightarrow \psi = \tilde{\psi} + \psi'$$

Used in

- Restarting iterative solvers
- Single precision acceleration
- Chronologically predicted solutions

# Convergence criteria and accuracy

$$\boldsymbol{D}\,\psi = \phi$$

The convergence of the algorithm will be tested using

$$|\rho| = |\phi - \boldsymbol{D}\tilde{\psi}| < \epsilon|\phi|$$

This deviates from the exact solution by at most

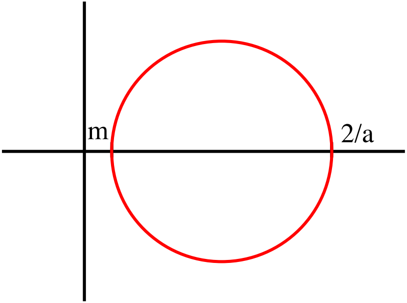
$$|\tilde{\psi} - \psi| = |\boldsymbol{D}^{-1}\boldsymbol{D}(\tilde{\psi} - \psi)| = |\boldsymbol{D}^{-1}\rho| < \epsilon|\boldsymbol{D}^{-1}||\phi| < \epsilon\kappa(\boldsymbol{D})|\psi|$$

with $\kappa(\boldsymbol{D})$ the **condition number**

$$\kappa(\boldsymbol{D}) = |\boldsymbol{D}||\boldsymbol{D}^{-1}|$$

Note that $\kappa(\boldsymbol{D})$ can be large
→ criterion on residue can mean very little for the solution
(and even less for single elements).

# Condition number



For $\lambda_{\min}^2$, $\lambda_{\max}^2$ the smallest/largest EV of $D^\dagger D$

$$\lambda_{\min} \propto$$
$$\lambda_{\max} \propto 1/a$$

Condition number $\kappa(D) \propto (am)^{-1}$.

# The GCR algorithm

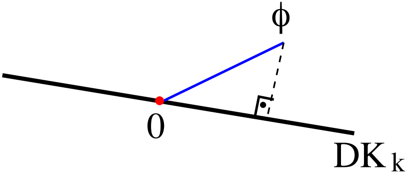
The Generalized Conjugate Residue algorithm constructs for each $k$ the solution of the Dirac equation

$$\psi_1, \psi_2, \psi_3, \dots \psi_k, \dots$$

that minimizes the residue

$$\rho_k = \phi - \boldsymbol{D}\,\psi_k$$

This minimum is attained where $\boldsymbol{D}\,\psi_k$ is the orthogonal projection of $\phi$ to $D\mathcal{K}_k$.



# The GCR algorithm

**Issue: How to represent the Krylov space**

This orthogonal projection is simplified by computing an **orthonormalized basis** $\chi_i$

$$D\mathcal{K}_k = \mathrm{span}\{\chi_i\}.$$

Computation of the new residue = **orthogonal projection**

$$\rho_k = \phi - \sum_{l=0}^{k-1} c_l \, \chi_l \qquad \text{with} \qquad c_l = (\chi_l, \phi)$$

# The GCR algorithm

At some point in the algorithm one thus has the orthonormal basis of $D\mathcal{K}_{k-1}$ and the residues

$$\{\chi_0, \chi_1, \ldots, \chi_{k-1}\} \qquad \{\rho_0, \rho_1, \ldots, \rho_{k-1}\}$$

one constructs the current residue

$$\rho_k = \phi - \sum_{l=0}^{k-1} c_l\, \chi_l \qquad \text{with} \qquad c_l = (\chi_l, \phi)$$

and adds to the basis $\chi_k$, the contribution of $D\rho_k$.

Compute the representation of the $\chi$ in terms of the $D\rho$.

$$\chi_k = \sum_{j=0}^{k} a_{kj} D\rho_j \qquad \text{with } \rho_0 = \phi$$

# Computing the current solution

$$\chi_k = \sum_{j=0}^{k} a_{kj} D \rho_j$$

and

$$\rho_k = \phi - \sum_{l=0}^{k-1} c_l \, \chi_l \qquad \text{with} \qquad c_l = (\chi_l, \phi)$$

Putting everything together, we have

$$D\psi_k = \sum_{l=0}^{k-1} c_l \, \chi_l = \sum_{l=0}^{k-1} c_l \sum_{j=0}^{l} a_{lj} D \rho_j$$

Now we can divide by $D$ and get the solution $\psi_k$ in terms of the already computed residues.

$$\psi_k = \sum_{l=0}^{k-1} c_l \sum_{j=0}^{l} a_{lj} \rho_j$$

# Convergence rate

The solution is constructed in the Krylov space
$\rightarrow$ the solution is a polynomial of $D$ times source.

$$\rho_k = \phi - D\tilde{p}_k(D)\phi = p_k(D)\phi$$

The GCR algorithm minimizes the residue

$$|\rho_k| = \mathbf{min}_p |p_k(D)\phi| \leq \mathbf{min}_p |p_k(D)||\phi|$$

Take diagonaliziable $D \rightarrow D = V\Lambda V^{-1}$, with $\Lambda$ diagonal

$$|p_k(D)| = |V p_k(\Lambda) V^{-1}| \leq \kappa(V)|p_k(\Lambda)|$$

This leads to the inequality for any $p_k$ w/ $p_k(0) = 1$

$$|\rho_k| \leq \kappa(V)\mathbf{max}_{\lambda\in\mathbb{D}}|p_k(\lambda)||\phi|$$

Assumption is that spectrum is contained in a disk $\mathbb{D}$ right of origin.

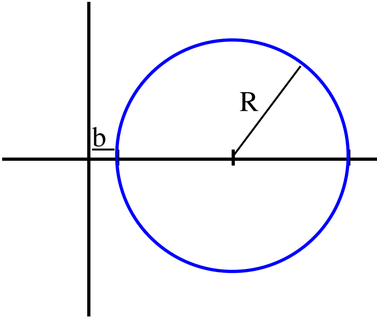
# Convergence rate

$$|\rho_k| \leq \kappa(V)\mathbf{min}_p\mathbf{max}_{\lambda\in\mathbb{D}}|p_k(\lambda)||\phi|$$

Theory of polynomial approximations:
The optimal polynomial, for which the extremum is attained is

$$p_k(\lambda) = (1 - \frac{\lambda}{R+b})^k$$

$R$ is radius of the disk,
$b$ the distance from origin.



$$|\rho_k| \leq \kappa(V)(1+\frac{b}{R})^{-k}|\phi|$$

For $b \ll R$ this decays roughly exponentially $e^{-k\frac{b}{R}}$

# GCR: Summary

Above, one iteration of the GCR has been described.

It requires to order $k$

- $2k$ spinor fields of storage.
- $k$ matrix vector products.

Typical is the computation 20–40 vectors.

Then perform a restart.

# Preconditioning

At small quark masses, the condition number of $D$ becomes large.

Many matrix-vector multiplications needed for solution.

**Preconditioning**

$$LDR\psi' = L\phi \qquad \to \psi = R\,\psi'$$

with $L$ and $R$ chosen such that $L\,D\,R$ is well conditioned.

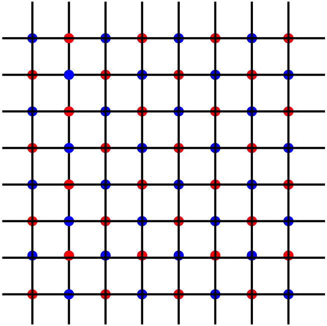
Many variants of preconditioning.

A good preconditioning will work on low-mode part of spectrum.

# Even-odd preconditioning

If the Dirac operator only connects nearest-neighbors, the lattice can be split in “even” and “odd” sites.

$$D = \begin{pmatrix} D_{\mathrm{ee}} & D_{\mathrm{eo}} \\ D_{\mathrm{oe}} & D_{\mathrm{oo}} \end{pmatrix}$$



# Even-odd preconditioning

$$D = \begin{pmatrix} D_{\text{ee}} & D_{\text{eo}} \\ D_{\text{oe}} & D_{\text{oo}} \end{pmatrix}$$

We can use the matrices $L$ and $R$

$$L = \begin{pmatrix} 1 & -D_{\text{eo}} D_{\text{oo}}^{-1} \\ 0 & 1 \end{pmatrix} \qquad R = \begin{pmatrix} 1 & 0 \\ -D_{\text{oo}}^{-1} D_{\text{oe}} & 1 \end{pmatrix}$$

to get

$$LDR = \begin{pmatrix} \hat{D} & 0 \\ 0 & D_{\text{oo}} \end{pmatrix} \qquad \text{with} \qquad \hat{D} = D_{\text{oo}} - D_{\text{oe}} D_{\text{ee}}^{-1} D_{\text{eo}}$$

$\hat{D}$ is the Schur complement
**$\hat{D}$ has less than half the condition number of $D$.**

# Preconditioning

Even-odd is the prime example of “classical” preconditioning.
It is used in all simulations with next-nearest operators only.

Another example is SSOR.
Complicated to implement → not used so frequently.

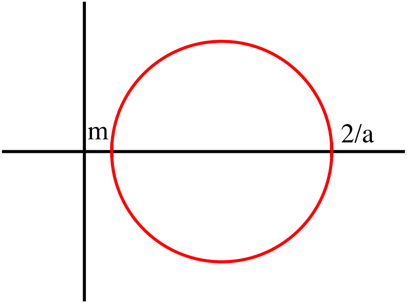
Their benefit is limited: they do not know about the physics.

For large quark masses, standard Krylov-space solvers + eo work fine.

Small quark masses: condition number $\propto (am)^{-1}$.

Need to take the IR physics into consideration to get more significant speed-up.

# Deflation



When quark mass gets small

$$\kappa \propto 1/(am)$$

due to small eigenvalues with $\mathrm{Re}\lambda \approx m$.

# Exact deflation with eigenvectors

Elimitate these eigenmodes from the Dirac equation.

$$D\,\psi_i = \lambda_i\,\psi_i$$

Projector on small eigenmodes $\psi_i$

$$P = \sum_{i=1}^{N_s} \psi_i\,\psi_i^\dagger$$

Using it, we can split the Dirac equation in two

$$D = \begin{pmatrix} P\,D\,P & 0 \\ 0 & (1-P)D(1-P) \end{pmatrix}$$

$$\Rightarrow D^{-1} = \begin{pmatrix} \sum_{i=1}^{N_s} \frac{1}{\lambda_i}\psi_i\,\psi_i^\dagger & 0 \\ 0 & [(1-P)D(1-P)]^{-1} \end{pmatrix}$$

# Discussion

$$D^{-1} = \begin{pmatrix} \sum_{i=1}^{N_s} \frac{1}{\lambda_i} \psi_i \, \psi_i^\dagger & 0 \\ 0 & [(1-P)D(1-P)]^{-1} \end{pmatrix}$$

**Reduction of condition number** $\kappa \to |\frac{\lambda_1}{\lambda_{N_s+1}}|\kappa$

Efficient if small number of very small eigenvalues e.g., $\epsilon$-regime calculations

Need to compute eigenvectors (can be set up with approximate vectors)

**For constant effect need** $N_s \propto V$ **vectors.**

In large volume, computation of a single eigenvector exceedingly expensive.

# Local deflation

The problem of "classical" deflation is the scaling with the volume.
Need $N_s \propto V$ modes w/ cost/mode at least $\propto V$.

**Local coherence**

Lüscher'07

Experimental fact:
*Locally* eigenvectors with $\lambda < 100$ MeV can be constructed from very few components.

Procedure:
Take $N_0$ lowest eigenmodes.

Decompose the lattice in small blocks $\Lambda_i$, e.g., $(0.3\,\mathrm{fm})^4$

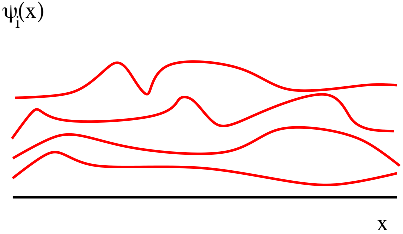
Consider space spanned by block projected vectors.

$$\mathcal{R} = \mathrm{span}\{P_{\Lambda_i}\psi_j |\ i = 1, \ldots, N_{\mathrm{block}},\ j = 1, \ldots, N_0\}$$

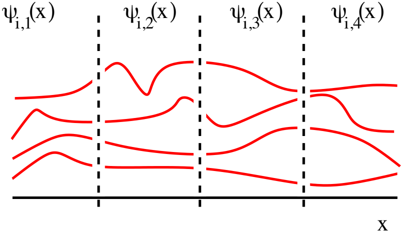
# Deflation subspace

$$\mathcal{R} = \mathbf{span}\{P_{\Lambda_i}\psi_j |\, i = 1, \ldots, N_{\text{block}},\, j = 1, \ldots, N_0\}$$

Start with global modes



And chop them apart



# Deflation subspace

$$\mathcal{R} = \mathrm{span}\{P_{\Lambda_i}\psi_j |\, i = 1, \ldots, N_{\mathrm{block}},\, j = 1, \ldots, N_0\}$$

Define deficit

$$\epsilon = |P_{\mathcal{R}}\psi_i - \psi_i|$$

with $P_{\mathcal{R}}$ the orthonormal projector to $\mathcal{R}$.

Experimental finding:

**The deficit for eigenvectors $\psi_i$ with eigenvalue $\leq 100$ MeV is small, $N_0 \sim 12$.**

$\epsilon \approx$ O(few %)

# Local coherence

This result can be interpreted as consequence of local coherence.

**In each point, the IR fields are aligned**.

However, the vectors in $\mathcal{R}$ are quite discontinuous.

They can only be decent approximations to the eigenvectors in the centers of the blocks.

---

Big advantage is that size of deflation space $\propto$ volume.

Eigenvectors do not need to be very exact.
A few inverse iterations suffice.

# Free field theory

Eigenmodes are plane waves

$$\psi_p(x) = u_p \, e^{i \, px}$$

$12$ lowest modes are constant

**How well can the higher modes be constructed by constant block modes?**

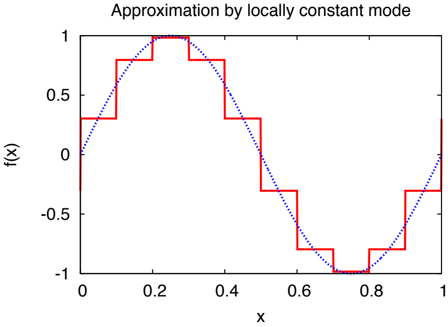
# Free field theory

Eigenmodes are plane waves

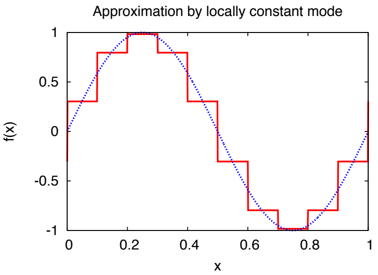
$$\psi_p(x) = u_p \, e^{i\,px}$$

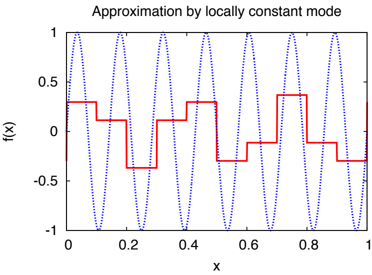
$12$ lowest modes are constant

**How well can the higher modes be constructed by constant block modes?**



# Free field theory





Efficiency is expected to decrease for eigenmodes with larger eigenvalues.

In the UV the Krylov solvers work very well.

Use as a preconditioner.

# Implementation in a solver

Decomposition of the Dirac operator

$$D = \begin{pmatrix} (1-P_{\mathcal{R}})\,D\,(1-P_{\mathcal{R}}) & (1-P_{\mathcal{R}})DP_{\mathcal{R}} \\ P_{\mathcal{R}}\,D\,(1-P_{\mathcal{R}}) & P_{\mathcal{R}}\,D\,P_{\mathcal{R}} \end{pmatrix}$$

with the "little Dirac operator"

$$D_{LL} = P_{\mathcal{R}}\,D\,P_{\mathcal{R}}$$

This is a $(N_s N_{\mathrm{block}})^2$ matrix.
Using the usual Schur complement trick

$$\begin{pmatrix} D_{HH} & D_{HL} \\ D_{LH} & D_{LL} \end{pmatrix}^{-1} =$$

$$\begin{pmatrix} 1 & 0 \\ -D_{LL}^{-1}D_{LH} & 1 \end{pmatrix} \begin{pmatrix} (D_{HH} - D_{HL}D_{LL}^{-1}D_{LH})^{-1} & 0 \\ 0 & D_{LL}^{-1} \end{pmatrix} \begin{pmatrix} 1 & -D_{HL}D_{LL}^{-1} \\ 0 & 1 \end{pmatrix}$$

# Deflating the Dirac equation

The Schur complement trick reduces the problem to the solution of

$$D_{LL}\psi_{\|} = \phi_{\|}$$
$$(D_{HH} - D_{HL}D_{LL}^{-1}D_{LH})\psi_{\perp} = \phi_{\perp}$$

The condition number of the matrix in the second equation is significantly reduced.

Rewrite 2nd eq. in form of preconditioning

$$(1 - DP_{\mathcal{R}}(P_{\mathcal{R}}DP_{\mathcal{R}})^{-1}P_{R})\,D\,\psi_{\perp} = \phi_{\perp}$$

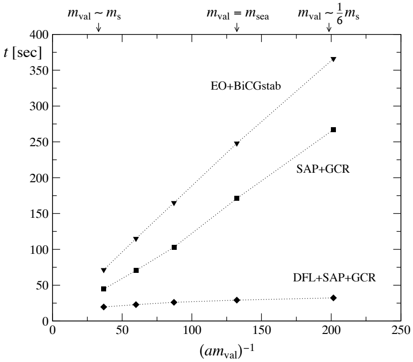
Can be solved with a GCR, but this is still expensive due to solution of the little system.

Still need a good preconditioner to make it feasible
Needs to be effective in the UV
→ Schwarz alternating procedure

# Performance of the deflated GCR

Plot from original paper

M. Lüscher, Local coherence and deflation of the low quark modes in lattice QCD, JHEP0707:081,2007

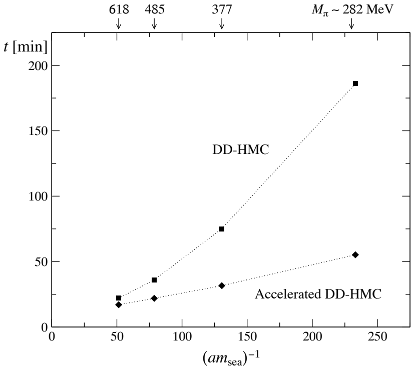


# Deflation and the HMC

The construction of the deflation subspace is not cheap.

The more solutions of the Dirac equation, the more it pays.
→ good for Hasenbusch decomposition

Low-modes evolve slowly in MC time
→ take subspace in several consecutive time step.



# HMC

## Momentum and pseudofermion Heatbath

Refresh momenta $\pi$
Refresh pseudofermions $\phi \rightarrow$ kept fixed during trajectory
Initialization of deflation subspace

## Molecular Dynamics

Solve numerically MD equations for some MC time $\tau$.

Repeated refresh of deflation subspace.

## Acceptance Step

Correcting for inaccuracies in integration.

Need to be careful for violation of reversibility.

# Summary

The solution to the Dirac equation is essential part of lattice computations with fermions.

Methods take advantage of the spasity of the Dirac matrix.
Matrix-Vector multiplications essential operation, is O$(V)$.

Cost can be high for small quark masses
→ shrinking gap to origin
→ problem with exceptionally small eigenvalues

Need to find a *physical* solution to problem.

# Numerical Methods for Lattice QCD IV

Stefan Schaefer

John von Neumann Institute for Computing, DESY

VII Parma International School of Theoretical Physics

# Measuring hadronic observables

The goal is to compute hadronic correlation functions on a set of gauge configurations, e.g.

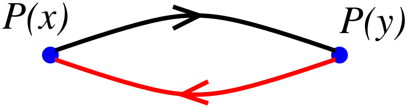
$$\langle P^a(x)\, P^b(y) \rangle$$

with

$$P^a = \overline{\psi} \frac{1}{2} \tau^a \gamma_5 \psi \qquad \text{and} \qquad \psi = \begin{pmatrix} u \\ d \end{pmatrix}$$

Use Wick's theorem to eliminate the Grassmann fields

$$\langle P^a(x)\, P^b(y) \rangle = -\frac{1}{2} \delta^{ab} \left\langle \mathrm{tr}_{c,d} \left\{ S_{m_u}(x,y)\, \gamma_5\, S_{m_d}(y,x)\, \gamma_5 \right\} \right\rangle$$



# Measuring hadronic observables

$$\langle P^a(x)\,P^b(y)\rangle = -\frac{1}{2}\delta^{ab}\,\langle\,\mathrm{tr}\,\{S_{m_u}(x,y)\,\gamma_5\,S_{m_d}(y,x)\,\gamma_5\,\}\rangle$$

With the propagator

$S(x,y) = D^{-1}(x,y)$ and $S_{\mathrm{H}}(x,y) = Q^{-1}(x,y) = S(x,y)\gamma_5$

we arrive for degenerate quarks at

$$\langle P^a(x)\,P^b(y)\rangle = \delta^{ab}\,\langle\,\mathrm{tr}\,\{S_{\mathrm{H}}(x,y)\,S_{\mathrm{H}}(y,x)\,\}\rangle$$

# Practical computation

Traditional method:

$$\langle \mathrm{tr}_{d,c} S_{\mathrm{H}}(x,y) S_{\mathrm{H}}(y;x)\rangle = \sum_{c,d,c',d'} \langle S_{H}(x,y)_{c,d;c',d'} S_{\mathrm{H}}(y;x)_{c',d';c,d}\rangle$$

For this one space-time column of the propagator is needed.

$$S_{\mathrm{H}}(x,y)_{c,d;c',d'} = (S_{\mathrm{H}}\,\eta^{(y,c',d')})(x)_{c,d}$$

with a point source

$$\eta^{(x_0,c_0,d_0)}_{c,d}(x) = \delta_{x,x_0}\delta_{c,c_0}\delta_{d,d_0}$$

Solve Dirac equation for the $4 \times 3$ Dirac–color index combinations

$$\boldsymbol{Q}\,\phi = \eta^{(y,c',d')}$$

Get propagator from one point to *all* other points.

# Volume average

Pion propagator projected on zero momentum

$$f_P(x_0 - y_0) = -\frac{1}{L^3} \sum_{\vec{x}} \sum_{\vec{y}} \langle \mathbf{tr}_{d,c} S_H(x,y) S_H(y;x) \rangle$$

Using point sources, the sum over $y$ is difficult to do, would need $12V$ solutions of the Dirac equation.

Translational invariance helps, need sum only at one end.

$$f_P(x_0) = -\sum_{\vec{x}} \langle \mathbf{tr}_{d,c} S_H(x,\mathbf{0}) S_H(\mathbf{0};x) \rangle$$

Lose information from different source positions.

Use a stochastic estimate for the traces.

# Noise sources

Insert additional complex scalar fields into your partition function.
Here just for one time slice $y_0$; "one-end trick".

$$Z_\eta = \int [d\eta][d\eta^\dagger]\, e^{-(\eta,\eta)}$$

Each lattice point, Dirac and color index has an independent Gaussian random number

$$\langle \eta_{c,d}(\vec{x})\, \eta^\dagger_{c',d'}(\vec{y})\rangle_\eta = \delta_{\vec{x},\vec{y}}\, \delta_{d,d'}\, \delta_{c,c'}$$

Insert in correlation function

$$f_P(x_0 - y_0) = -\frac{1}{V}\sum_{\vec{x}} \langle \mathrm{tr}_{d,c} S_{\mathrm{H}}(x,\cdot)\eta\eta^\dagger S_{\mathrm{H}}(\cdot;x)^\dagger\rangle$$

here the $\langle\cdot\rangle$ includes average over $\eta$ fields.

# Stochastic estimate

$$f_P(x_0 - y_0) = -\frac{1}{V}\sum_{\vec{x}}\langle \mathbf{tr}_{d,c} S_\mathrm{H}(x,\cdot)\eta\eta^\dagger S_\mathrm{H}(\cdot;x)^\dagger\rangle$$

As always in Monte Carlo, we replace integrals by a sum over a number of field realizations.

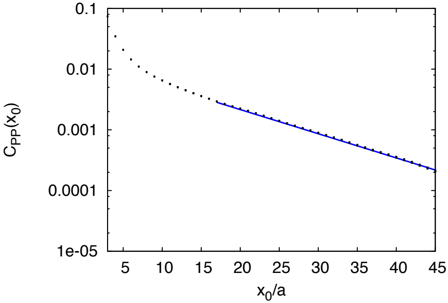
$$-\frac{1}{V}\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{\vec{x}}\langle \mathbf{tr}_{d,c} S_\mathrm{H}(x,\cdot)\eta_i\eta_i^\dagger S_\mathrm{H}(\cdot;x)^\dagger\rangle$$

Unbiased estimator, no need to take $N_s$ large.
Also $N_s = 1$ is correct, but take new source on each configuration. Integrals commute.

Need to solve one Dirac equation per source.
For pions $\mathrm{O}(10)$ give a very good signal.

For mesons, no need to use more sources as $V \to \infty$.

# Pion propagator



Exponential fall-off for $x_0 \to \infty$

$$C_{PP}(x_0) = \sum_n A_n e^{-m_n x_0} \to A_0 e^{-m_\pi x_0}$$

Source couples to all states with given quantum numbers.
Excited states clearly visible at small $x_0$.

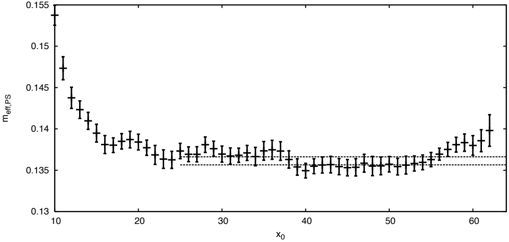
# Effective mass

Since exponential fall-off is difficult to judge, one typically looks at quantities that show a plateau.

$$\frac{C(x_0)}{C(x_0+a)} = \frac{A\,e^{-mx_0}}{A\,e^{-m(x_0+a)}} = e^{am}$$

Effective mass

$$am_{\text{eff}} = \log\frac{C(x_0)}{C(x_0+a)}$$



# Statistical error

Reminder:
The square of the error of a measuremnt is proportional to the variance of the observable

$$\sigma^2(A) = \langle A^2 \rangle - \langle A \rangle^2$$

Parisi'83

The variance is a physical observable, the exponential fall-off can be predicted.

$$\underset{\propto e^{-E_\pi |x-y|}}{\langle A \rangle = \langle P^a(x) P^a(y) \rangle} \quad \rightarrow \quad \sigma^2(A) \quad \underset{\propto e^{-E_{2\pi} |x-y|}}{= \langle P^a(x) P^b(x) P^a(y) P^b(y) \rangle - \langle A \rangle^2}$$

In large volume, $E_{2\pi} = 2m_\pi = 2E_\pi$
Constant signal-to-noise ratio

$$\frac{\langle A \rangle}{\sigma(A)} \propto \frac{e^{-m_\pi |x-y|}}{\sqrt{e^{-2m_\pi |x-y|}}} = \mathrm{const.}$$

# Signal-to-noise problem

For the nucleon, one considers

$$\langle A\rangle = \Gamma_{\alpha\beta}\langle N_\alpha(x)\bar{N}_\beta(y)\rangle \propto e^{-E_N|x-y|}$$

Variance

$$\langle A^2\rangle - \langle A\rangle^2 \text{``} = \text{``}\langle N(x)\bar{N}(x)\, N(y)\bar{N}(y)\rangle - (\langle N(x)\,\bar{N}(y))^2\rangle \propto e^{-E_{3\pi}|x-y|}$$

Matches quantum numbers of three pions and therefore the signal-to-noise ratio is

$$\frac{\langle A\rangle}{\sigma(A)} = e^{-(m_N - \frac{3}{2}m_\pi)|x-y|}$$

Exponential reduction once $m_N > \frac{3}{2}m_\pi$.

Makes calculations of proton properties exceedingly difficult.

# Summary fermions

Most effort goes into fermions.

Deflation of Dirac equation brought great progress.
Is there even more possible?

Computation of PS meson two-point functions well-established.

Significant challenges in baryon sector.

# Back to simulations

**What happened so far**

Methods for Markov Chain Monte Carlo

Sequence of field configuratoins

→MC time series of measurements

Field updates are expensive → limited statistics

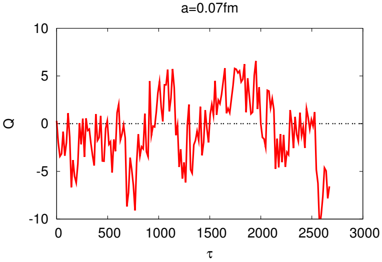
---

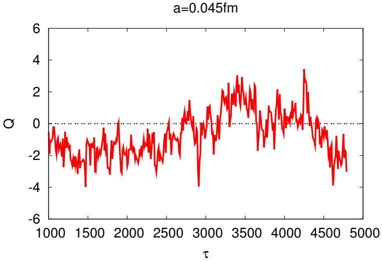
**Outline for today**

Methods to deal with autocorrelations

# Bad start

Topological charge





$a \approx 0.08\text{fm}$

$64 \times 32^3$

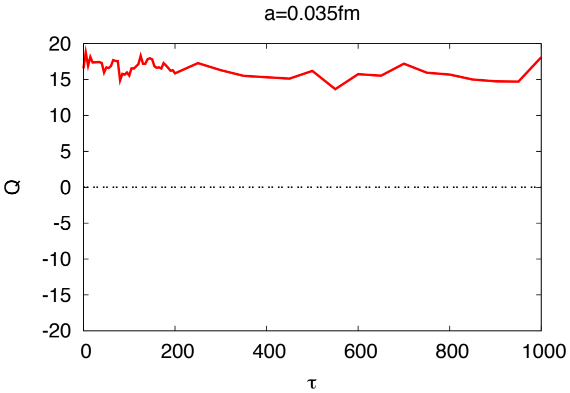
$m_\pi \approx 360\text{MeV}$

$a \approx 0.06\text{fm}$

$64 \times 32^3$

$m_\pi \approx 460\text{MeV}$

# A bad start



$a \approx 0.04$fm

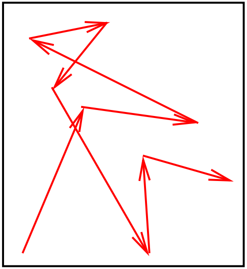
$128 \times 64^3$

$m_\pi \approx 480$MeV

# Markov Chain Monte Carlo

Sequence of field configurations

$$U_1 \rightarrow U_2 \rightarrow U_3 \rightarrow \cdots \rightarrow U_N$$



Generated by a **transition probability density**

$$T(U' \leftarrow U) \geq 0 \quad \text{for all } U, U'$$

Stability

$$\int [dU]\, T(U' \leftarrow U)\, P[U] = P[U']$$

Normalization

$$\int [dU']\, T(U' \leftarrow U) = 1$$

# Autocorrelations

Sequence of field configurations

$$U_1 \to U_2 \to U_3 \to \cdots \to U_N$$

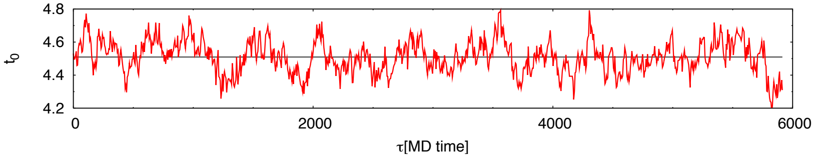
Measurements of observables are correlated

$$A_1 \to A_2 \to A_3 \to \cdots \to A_N$$

Estimates

$$\bar{A} \approx \tilde{A}_N = \frac{1}{N}\sum_{i=1}^{N} A_i$$

How far is this off?



# Autocorrelations

Variance of estimator

$$\langle\!\langle(\tilde{A}_N - \bar{A})^2\rangle\!\rangle = \frac{1}{N^2}\sum_{i,j=1}^{N} \langle\!\langle(A_i - A)(A_j - A)\rangle\!\rangle$$

For $N$ large, this depends only on the difference in simulation time

$$\langle(\tilde{A}_N - \bar{A})^2\rangle = \frac{1}{N}\sum_{t=-\infty}^{\infty} \Gamma_A(t)$$

$$\Gamma_A(t) = \langle(A_0 - \bar{A})(A_t - \bar{A})\rangle$$

Note:
again substitution average over simulations
→ average in simulation time

# Error of the measurement

$$\langle(\tilde{A}-\bar{A})^2\rangle = \frac{1}{N}\sum_{t=-\infty}^{\infty}\Gamma_A(t)\ ; \qquad \Gamma_A(t) \quad = \langle(A_0-A)(A_t-A)\rangle$$

$$= \frac{\mathrm{var}(A)}{N}\sum_{t=-\infty}^{\infty}\rho_A(t)$$

**Integrated autocorrelation time**

$$\tau_{\mathrm{int}}(A) = \frac{1}{2}+\sum_{t=1}^{\infty}\frac{\Gamma_A(t)}{\Gamma_A(0)} \equiv \frac{1}{2}+\sum_{t=0}^{\infty}\rho_A(t)$$

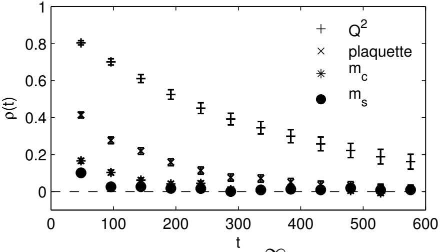
**Error of the measurement**

$$\sigma_A = \sqrt{\frac{\mathrm{var}(A)}{N/(2\tau_{\mathrm{int}}(A))}}$$

Measures efficiency of algorithm.

→ eff. statistics reduced by $2\tau_{\mathrm{int}}$

# Measuring autocorrelations



$$\tau_{\text{int}}(A) = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_A(t)$$

We only have a limited precision estimate of the integrand.
Summing to $t = \infty$ leads to diverging variance.
→ need to cut the summation
→ biased estimate
**Need to find a balance between stat. and syst. error.**

# Error of $\tau_{\mathrm{int}}$

$$\tau_{\mathrm{int}}(A) = \frac{1}{2} + \sum_{t=1}^{W} \frac{\Gamma_A(t)}{\Gamma_A(0)}$$

**Systematic error**

Summation truncated at $W$
→ neglect potentially large tail.
Particular problem in presence of slow modes.

**Statistical error** Madras,Sokal

$$\langle[\tilde{\tau}_{\mathrm{int}}(A,W) - \tau_{\mathrm{int}}(A,W)]^2\rangle \approx \frac{4}{N}(W + \frac{1}{2} - \tau_{\mathrm{int}}(A))\tau_{\mathrm{int}}(A)^2$$

Infinite variance for $W \to \infty$.

# Criteria for $W$

All automatic methods are problematic.

- Cut where $\delta\Gamma > \Gamma$
  $\rightarrow$ large systematic error
- Madras-Sokal criterion
  $\rightarrow$ minimum of sum of systematic and statistical error

$$\frac{\delta\sqrt{\tau}}{\sqrt{\tau}} \propto \min_{W} \left( e^{-W/\tau} + 2\sqrt{W/N} \right)$$

- ALPHA method (2010)
  Estimate $\tau_{\mathrm{exp}}$ from various (slow) observables
  Add tail to all other observables before losing signal $\Gamma_A$

# Analyzing autocorrelation functions

The autocorrelation function depends on

**Underlying theory**

**Algorithm**

**Observable**

---

Analyze it similarly to a corraltion function
Wave functions + masses

# Detailed balance

## Detailed balance

$$T(U' \leftarrow U)\,P[U] = P[U']\,T(U' \leftarrow U)$$

Implies stability

$$\int [dU]T(U' \leftarrow U)P[U] = \int [dU]P[U']T(U' \leftarrow U) = P[U']$$

Elementary steps frequently fulfill this condition.

As a consequence we have a **symmetric matrix** $M$

$$M(U' \leftarrow U) = P[U']^{-1/2}T(U' \leftarrow U)P[U]^{1/2}$$

# Detailed balance

Detailed balance

$$T(U' \leftarrow U)\, P[U] = P[U']\, T(U' \leftarrow U)$$

Associated symmetric matrix $M$

$$M(U' \leftarrow U) = P[U']^{-1/2} T(U' \leftarrow U) P[U]^{1/2}$$

If $\eta$ eigenvector of $T$

$$\xi(U) = P^{-1/2}(U)\eta(U)$$

is eigenvector of $M$ **with the same eigenvalue** $\lambda$.

**Spectral decomposition**

$$M = \sum_i \lambda_i\, \xi_i \xi_i^\dagger$$

# Autocorrelation

Spectral decomposition

$$\begin{aligned}
\Gamma_A(t) &= \langle\, (A_t - \bar{A})\,(A_0 - \bar{A})\,\rangle \\
&= \int [dU][dU'] \delta A(U')\, T^t(U' \leftarrow U)\, \delta A(U)\, P[U] \\
&= \int [dU][dU'] P^{1/2}[U'] \delta A(U') M^t(U' \leftarrow U) \delta A(U) P^{1/2}[U] \\
&= \sum_{n>0} (\lambda_n)^t \, [c_n(A)]^2
\end{aligned}$$

With "matrix elements"

$$c_n(A) = \int [dU] \xi_n(U) [P[U]]^{1/2} (A(U) - \bar{A})$$

# Spectral representation

$$\begin{aligned}\Gamma_A(t) &= \sum_n (\lambda_n)^t \, [c_n(A)]^2 \\ &= \sum_n \operatorname{sign}\lambda_n \, e^{-t/\tau_n} \, [c_n(A)]^2\end{aligned}$$

- $\tau_n = 1/\log|\lambda_n| > 0$
- For the analysis of algorithms it is useful to think of Monte Carlo time $t$ as a fifth dimension.
- Autocorrelation function is a 2pt function.
- time constants $\tau_n \rightarrow$ inverse masses
- Slowest decay $\tau_1 \rightarrow$ exponential AC time

# Comments

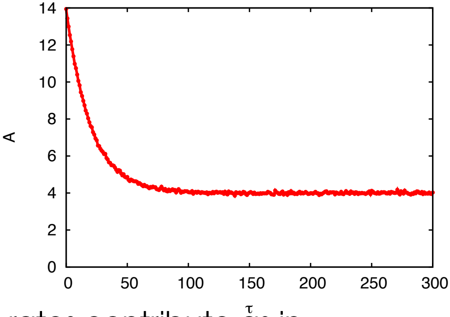
$$\Gamma_A(t) = \sum_n e^{-t/\tau_n} c_n^2(A)$$

- $\tau_n$ depend only on algorithm
- Matrix elements $c_n$ depend on observable.
- All observables affected by slow modes.

## Length of a simulation

- Simulation must have length of at least $\mathrm{O}(100) \times \tau_1$.
- $\tau_{\mathrm{int}}(A)$ can be much smaller than $\tau_1$
- Danger of
  - Incomplete thermalization.
  - Bias.
  - Wrong estimate of autocorrelations.

# Thermalization



Same decay rates contribute as in $\tau_{\text{int}}$
different initial distribution/matrix elements

$$\int [dU][dU'] P^{1/2}[U'] \delta A(U') M^t(U' \leftarrow U) \delta A(U) \frac{P_0[U]}{P^{1/2}[U]}$$

$$= \sum_{n>0} (\lambda_n)^t \, [c_n(A)][c_n^{(0)}(A)]$$

Opportunity to learn about largest $\tau_1$.

# Observables

Need to look at observables with large $c_i(A_k)$, $i$ small.
**Hunt for slow quantities.**

---

Noise can cover up auto-correlations ($\eta$ Gaussian noise)

$$A \to A + c\eta \qquad \Rightarrow \qquad \Gamma(t) \to \Gamma(t) + c^2\delta_{t,0}$$

$$\Rightarrow \qquad \tau_{\text{int}}(A) \to \tau_{\text{int}}(A)\frac{\text{var}(A)}{\text{var}(A) + c^2}$$

**Look at low-noise observables**

---

Take into consideration expected **scaling properties**

$$\tau_{\text{int}} \propto \frac{1}{a^2} \text{ for } a \to 0$$

# Wilson flow

LÜSCHER'10, LÜSCHER&WEISZ'11

- Smoothing with **gradient flow** at fixed flow time $t = t_0$.

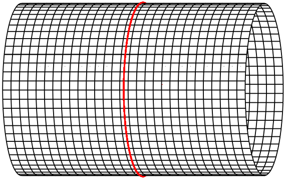
$$\partial_t V_t(x,\mu) = -g_0^2 \left[\partial_{x,\mu} S(V_t)\right] V_t(x,\mu); \quad V_t(x,\mu)|_{t=} = U(x,\mu)$$

- Gaussian smoothing over $r \sim \sqrt{8t}$.
- Renormalized quantities with continuum limit.
- Smooth observables → long autocorrelations.

$$\overline{E} = -\frac{a^3}{2L^3} \sum_{\vec{x}} \mathrm{tr}\, G_{\mu\nu} G_{\mu\nu} \Big|_{x_0 = T/2}$$

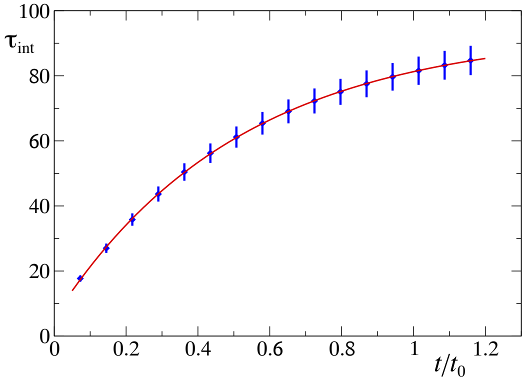
$$\overline{Q} = -\frac{a^3}{32\pi^2} \sum_{\vec{x}} \mathrm{tr}\, \tilde{G}_{\mu\nu} G_{\mu\nu} \Big|_{x_0 = T/2}$$

$$Q = -\frac{a^4}{32\pi^2} \sum_{x} \mathrm{tr}\, \tilde{G}_{\mu\nu} G_{\mu\nu}$$



# Effect of the smoothing

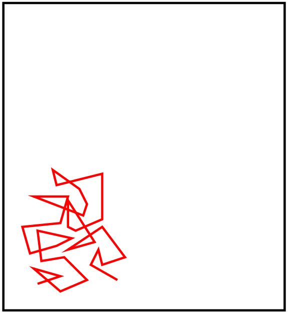
Autocorrelation time of $\bar{E}$ vs smoothing range (a=0.05fm).



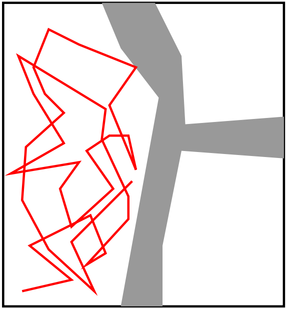
- $\sqrt{8t}$ smoothing radius $\rightarrow t = t_0$ smoothing over $r \approx r_0$
- $\tau_{\rm int}$ saturates with $\tau_{\rm int} = 93 + ae^{-ct}$.

# Dangers

- Algorithm is slow.
- Detectable by measuring autocorrelations.



- There are barriers in field space.
- Hard to detect.



# Topological charge

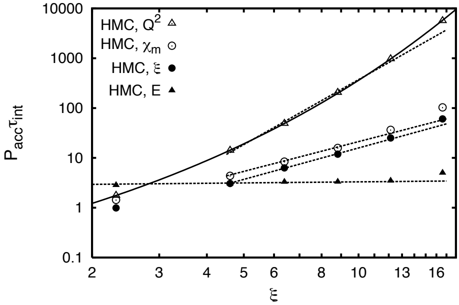
Best recipe: Avoid large autocorrelations $\tau_n$.

Special case: Topological charge

In the continuum, topological sectors form.
Consequence of the periodic boundary conditions.

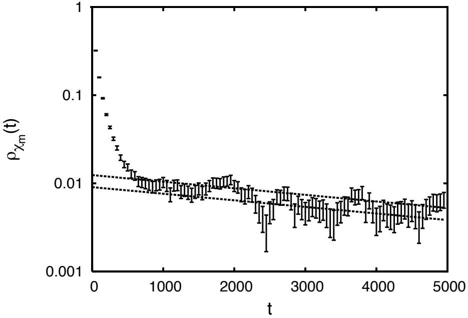
Happens very quickly as $a \to 0$.

Engel, S.'10



# Example from $CP^{N-1}$ model

Topological charge AC dominates other observables.



Solution:
Use setup without topological sectors
→ Open boundary conditions.

# Summary: Autocorrelations

Autocorrelations *always* present in MCMC.

Not sufficient to look at one observable alone.

Noise covers correlations, but does not solve them.

Smoothed out quantities particularly sensitive.

For a reliable result, need a careful estimate of $\tau_{\mathrm{int}}(A)$ and $\tau_1$.

**Work on better algorithms: more statistics, smaller AC.**