Introduction to lattice quantum chromodynamics

CFNS Summer School 2022

Chris Monahan

William & Mary

Center for Frontiers in Nuclear Science

Lecture outline

Lecture 1

- Path integrals and lattice field theory
- Quantum chromodynamics

Lecture 2

- Spin models
- Numerical integration
- Monte Carlo methods

Lecture 3

- Gauge field Monte Carlo
- Correlation functions

Summary: lecture 1

- Lattice field theory provides rigorous definition of nonperturbative QFT
- QCD describes fundamental interactions of the strong nuclear force in terms of quarks and gluons
- Lattice QCD formulated with quarks on the lattice sites and *SU(3)* link variables representing gluons
- Gluon action built from closed loops of link variables, such as plaquettes
- Fermions on the lattice suffer from fermion doubling; intimately connected to existence of chiral fermions in a regulated theory
- Different discretisations possible, but any choice should lead to QCD in the continuum limit of vanishing lattice spacing

Summary: lecture 2

- Numerical evaluation of high-dimensional integrals requires **Markov chain Monte Carlo**
- Markov chain Monte Carlo consists of two basic steps
  - propose local update
  - accept/reject update
- Local-update algorithms suffer large **autocorrelations** – use **global-update** algorithms
- Global-update algorithms work well near phase transitions

3.1

Monte Carlo for gauge theories

From Metropolis to hybrid Monte Carlo

QCD on the lattice

Reminder: QCD in the continuum

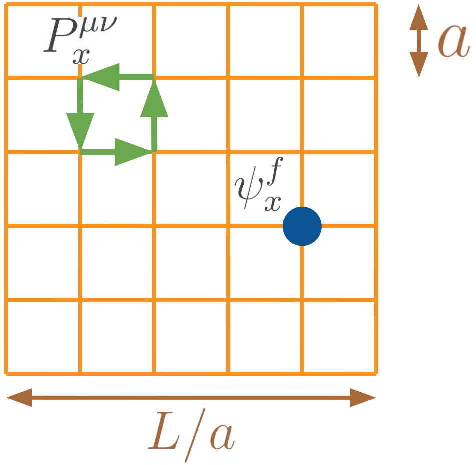
$$\mathcal{L}_{\mathrm{QCD}} = -\frac{1}{4} F_{\mu\nu}(x) F^{\mu\nu}(x) + \sum_f \overline{\psi}_f(x) \left[\gamma_\mu D^\mu + m_f \mathbb{I}\right] \psi_f(x)$$

QCD on the lattice

$$\mathcal{L}_{\mathrm{LQCD}} = \mathcal{L}_g(U_x) + \sum_f \overline{\psi}\,{}^f_y D_{yx}(U)\, \psi^f_x$$

Path integrals are numerical integrals

$$\langle 0|\overline{\psi}\psi|0\rangle = \int \mathcal{D}U\, \overline{\psi}\psi\, \det D(U)\, e^{-S_{\mathrm{LQCD}}[\overline{\psi},\psi,U]}$$



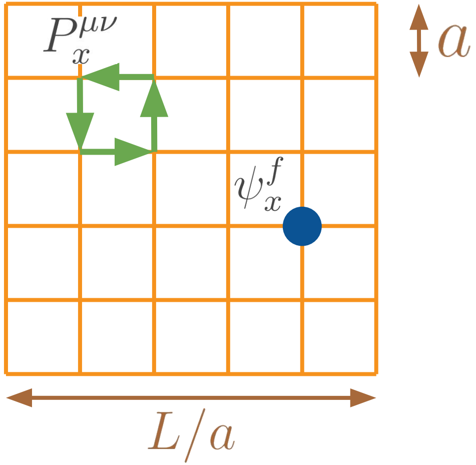
QCD on the lattice

Reminder: QCD in the continuum

$$\mathcal{L}_{\mathrm{QCD}} = -\frac{1}{4}F_{\mu\nu}(x)F^{\mu\nu}(x) + \sum_f \overline{\psi}_f(x)\,[\gamma_\mu D^\mu + m_f\mathbb{I}]\,\psi_f(x)$$

QCD on the lattice

$$\mathcal{L}_{\mathrm{LQCD}} = \mathcal{L}_g(U_x) + \sum_f \overline{\psi}{}^f_y D_{yx}(U)\,\psi^f_x$$



Path integrals are numerical integrals

$$\langle 0|\overline{\psi}\psi|0\rangle = \int \mathcal{D}U\,\overline{\psi}\psi\,\boxed{\det D(U)\,e^{-S_{\mathrm{LQCD}}[\overline{\psi},\psi,U]}}$$

Need to generate *configurations* with this *probability distribution*

Monte Carlo for gauge theories

Need to adapt our Monte Carlo methods for gauge theories

- Ising model updated (by flipping or otherwise) spins

$$\langle f_\beta \rangle = \sum_\sigma f(\sigma) P_\beta(\sigma) = \frac{1}{Z_\beta} \sum_\sigma f(\sigma) e^{-\beta H(\sigma)}$$

- For gauge theories we need to update links (group elements/matrices)

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U \, \mathcal{O}[U] \exp\Big[- S_{\text{Latt.}}[U(x)] \Big]$$

Need to do this in a way that maintains gauge invariance

Formally, the integration measure is the **Haar measure**

Metropolis algorithm

1. Generate a uniformly ~~distributed random number~~ random *SU(3)* matrix
2. Modify the value ~~at a specific lattice site~~ on a specific link

   $U_\mu \to MU_\mu$
3. Calculate the change in the action generated by this modification
4. If

   $\Delta S < 0$

   accept the change and move on to the next site [return to 1.]
5. Else if

   $\Delta S > 0$

   generate a new random number

   $\eta \in (0, 1)$
6. If

   $\exp\left[-\Delta S\right] > \eta$

   accept the change, otherwise reject the change
7. Move on to the next site [return to 1.]

Metropolis algorithm

Comments:

1. Usually the random *SU(3)* matrices are generated beforehand
    - variety of methods
2. Set of approximately 100 random *SU(3)* matrices is sufficient, if:
3. Updates are local, so multiple sites can be updated simultaneously
4. Multiple updates to a single site usually carried out before moving sites
5. Link re-unitarization required to avoid propagation of round-off errors
    - each matrix has an inverse within the set
    - set spans *SU(3)* through matrix multiplication

Molecular dynamics: the idea

Recast the Euclidean path integral as Hamiltonian dynamics

Basic idea:

$$\mathbb{I} = \int \frac{\mathrm{d}p}{\sqrt{2\pi}}\, e^{-p^2/2}$$

1. Insert the identity (field)

$$P(\phi) = \int \mathcal{D}\phi\, e^{-S_{\mathrm{E}}[\phi]} = \int \mathcal{D}\phi \int \mathcal{D}p\, e^{-p^2/2} e^{-S_{\mathrm{E}}[\phi]} = \int \mathcal{D}\phi \int \mathcal{D}p\, e^{-\frac{1}{T}H[\phi,p]}$$

2. Dynamics now captured by Hamilton's equations in **enlarged phase space**

$$\dot{\phi}(x) = -\frac{\partial H[\phi,p]}{\partial p} \qquad \dot{p}(x) = -\frac{\partial H[\phi,p]}{\partial \phi}$$

Fields evolve in a new dimension – **molecular dynamics time**

Molecular dynamics: the formalism

Introduce momenta conjugate to link variables and Hamiltonian

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

that defines classical evolution in **molecular dynamics** time

$$\frac{\partial \pi^a_\mu(x)}{\partial \tau} = -\frac{\partial S[e^\omega U(x)]}{\partial \omega^a_\mu(x)} \qquad \frac{\partial U_\mu(x)}{\partial \tau} = \pi^\mu(x) U_\mu(x)$$

Follows from Liouville theorem

Corresponding set of all trajectories is area preserving, with

$$Z = \int \mathcal{D}\pi\, \mathcal{D}U \; \exp\left[-H[\pi, U]\right]$$

Assuming ergodicity, probability of visiting a specific link variable along a classical trajectory is

$$\exp\left[-S[U(x)]\right]$$

Ensemble averages are replaced by trajectory averages

$$\langle \mathcal{O} \rangle = \frac{1}{\tau}\int_{\tau_0}^{\tau_0+\tau} \mathrm{d}\tau\, \mathcal{O}[U[\tau]]$$

Hybrid Monte Carlo

Exact solutions of the Molecular Dynamics equations are valid updates

Of course, any numerical solution is not exact – introduce **hybrid Monte Carlo** (HMC)

View molecular dynamics evolution as a **Metropolis update proposal**

Apply usual accept/reject step following this update

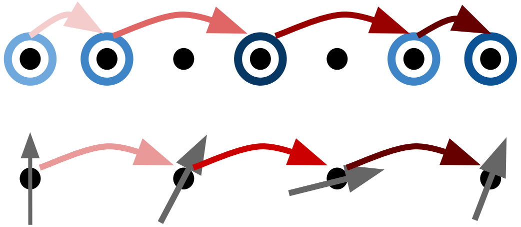
Comments:

- Accept/reject step ensures Molecular Dynamics step satisfies detailed balance even for finite numerical integration step size
- Acceptance rate decreases exponentially with system size
- Higher order integrators do not seem to help
- Sampling efficiency can be improved by drawing new momenta – **“refreshed molecular dynamics”**
- Requires time-reversal invariant and symplectic numerical integration algorithm, e.g **leap-frog algorithm**

Even better are *Omelyan integrators*

Hybrid Monte Carlo algorithm

1. Generate initial configuration of link variables
2. Choose conjugate momenta from a Gaussian distribution
3. Evolve links and momenta via **Molecular Dynamics** step [**leap frog**]:
    a. Evolve momenta by one half time-step
    b. Evolve link variables by one time step
    c. Repeat for desired number of iterations, ending with a.



4. Accept new configuration with usual Metropolis probability

$$P = \min\left\{1, \exp\left[-\Delta H[\pi, U]\right]\right\}$$

5. Return to 2.

Summary: lecture 3.1

- Gauge configuration algorithms are more sophisticated than spin models
- Should maintain gauge invariance and cannot take advantage of cluster algorithms
- Examples:
    - Metropolis, hybrid Monte Carlo
- State-of-the-art algorithms mix methods (i.e. use a lot of jargon)
    - e.g. hybrid Monte Carlo with refreshed molecular dynamics and an Omelyan integrator

3.2 Constructing correlation functions

From correlators to cross-sections

Two point functions on the lattice

Two-point functions correspond to the **propagators** of the theory

i.e. the set of all energy states of the theory

They encode the **spectrum** of the theory

$$\langle O|\hat{\phi}(t,\mathbf{0})\hat{\phi}(0,\mathbf{0})|0\rangle := \langle O|\hat{\phi}(t)\hat{\phi}(0)|0\rangle$$

$$\begin{aligned}\langle O|\hat{\phi}(t)\hat{\phi}(0)|0\rangle &= \langle 0|e^{it\hat{H}}\hat{\phi}(0)e^{-it\hat{H}}\hat{\phi}(0)|0\rangle \\ &= \langle 0|\hat{\phi}(0)e^{-it\hat{H}}\sum_n |n\rangle\langle n|\hat{\phi}(0)|0\rangle \\ &= \sum_n e^{-itE_n}\langle 0|\hat{\phi}(0)|n\rangle\langle n|\hat{\phi}(0)|0\rangle \\ &= \sum_n e^{-itE_n}\left|\langle 0|\hat{\phi}(0)|n\rangle\right|^2\end{aligned}$$

$$\mathbb{I} = \sum_n |n\rangle\langle n| \qquad \hat{H}|n\rangle = E_n|n\rangle$$

In Euclidean spacetime, this becomes

$$\langle O|\hat{\phi}(t)\hat{\phi}(0)|0\rangle = \sum_n e^{-tE_n}\left|\langle 0|\hat{\phi}(0)|n\rangle\right|^2 \overset{t\to\infty}{\longrightarrow} e^{-tE_1}\left|\langle 0|\hat{\phi}(0)|1\rangle\right|^2$$

Hadrons on the lattice I

Lattice QCD is a nonperturbative formulation of QCD – observable states are those of QCD!

Need to construct **interpolating operators** – operators that create hadrons of specific quantum numbers

Interpolating operators create *all* states with the same quantum numbers

Example: the pion

Construct a pseudoscalar quark-antiquark pair

$$\mathcal{O}^{\pi}_{x} = \overline{\psi}^{\,u/d}_{x} \gamma_5 \, \psi^{u/d}_{x}$$

then the two-point function is

$$\langle 0 | \sum_{\mathbf{y}} \mathcal{O}^{\pi}_{\mathbf{y},t} \cdot \mathcal{O}^{\pi}_{\mathbf{x},0} | 0 \rangle \overset{t\to\infty}{\to} e^{-m_\pi t} \left| \langle 0 | \mathcal{O}^{\pi}_{\mathbf{x},0} | \pi \rangle \right|^2$$

sum over space projects out pion at rest

In the absence of electromagnetism, up and down quarks are only distinguished by their mass. But relative to the other scales of QCD, their masses can treated as **degenerate**. High-precision results (~0.5%) require **isospin breaking corrections** to account for effects of electromagnetism and quark mass splitting.

Hadrons on the lattice II

$$\int \mathcal{D}\overline{\psi}\mathcal{D}\psi \, \exp\left[-\overline{\psi}D(U)\psi\right] = \det D(U)$$

recall: Gaussian integral

To do this in practice, we need some more **Grassmann integrals**

$$\int \mathcal{D}\overline{\psi}\mathcal{D}\psi \, \psi_y\overline{\psi}_x \exp\left[-\overline{\psi}D(U)\psi\right] = D^{-1}_{yx}\det D(U)$$

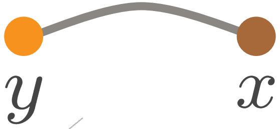
$$D_{yx}\psi^f_x = 0$$

recall: lattice Dirac operator

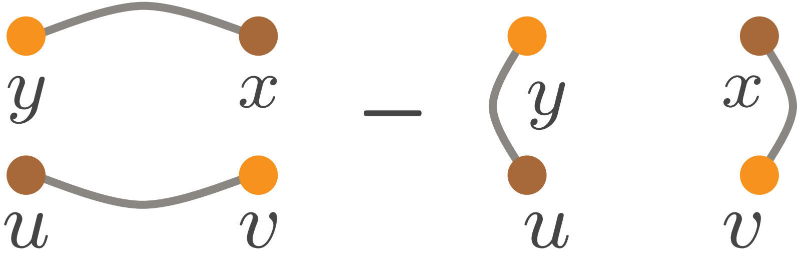
which we can generalise to larger numbers of fermion fields

$$\int \mathcal{D}\overline{\psi}\mathcal{D}\psi \, \psi_y\overline{\psi}_x\psi_v\overline{\psi}_u \exp\left[-\overline{\psi}D(U)\psi\right] = \left[D^{-1}_{yx}D^{-1}_{vu} - D^{-1}_{yu}D^{-1}_{vx}\right]\det D(U)$$

This is basically Wick's theorem, but our inverse Dirac operators correspond to fermion propagators



two-point function



four-point function

Mesons on the lattice

Example: the pion two-point function

$$\mathcal{O}^{\pi}_{x} = \overline{\psi}^{\,u/d}_{x} \gamma_5 \, \psi^{u/d}_{x}$$

Construct pseudoscalar quark-antiquark pairs at two different time slices

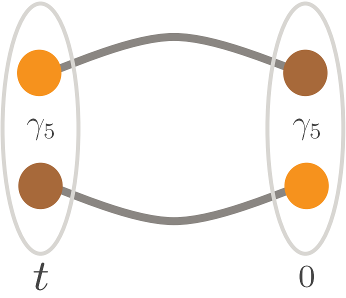
$$\langle 0 | \sum_{\mathbf{y}} \mathcal{O}^{\pi}_{\mathbf{y},t} \cdot \mathcal{O}^{\pi}_{\mathbf{x},0} | 0 \rangle = \int \mathcal{D}U \mathcal{D}\psi \mathcal{D}\overline{\psi} \sum_{\mathbf{y}} \mathcal{O}^{\pi}_{\mathbf{y},t} \cdot \mathcal{O}^{\pi}_{\mathbf{x},0} \, e^{-\mathcal{S}_g[U]} e^{-\overline{\psi} D[U] \psi}$$

Evaluate the fermion integral exactly

$$\langle 0 | \sum_{\mathbf{y}} \mathcal{O}^{\pi}_{\mathbf{y},t} \cdot \mathcal{O}^{\pi}_{\mathbf{x},0} | 0 \rangle = \int \mathcal{D}U \, \mathrm{Tr} \left[\gamma_5 D^{-1}_{\mathbf{y}t,\mathbf{x}0} \gamma_5 D^{-1}_{\mathbf{x}0,\mathbf{y}t} \right] \, \det D[U] \, e^{-\mathcal{S}_g[U]}$$

And approximate the gauge integral via Markov chain Monte Carlo

$$\langle 0 | \sum_{\mathbf{y}} \mathcal{O}^{\pi}_{\mathbf{y},t} \cdot \mathcal{O}^{\pi}_{\mathbf{x},0} | 0 \rangle \approx \frac{1}{N_{\mathrm{conf}}} \sum_{m=1}^{N_{\mathrm{conf}}} \mathrm{Tr} \left[\gamma_5 D^{-1}_{\mathbf{y}t,\mathbf{x}0} \gamma_5 D^{-1}_{\mathbf{x}0,\mathbf{y}t} \right]$$



Summary: lattice calculation procedure

1. Generate an **ensemble** of gauge **configurations** (sets of gluon link variables) on a **finite volume**, **discretised**, **Euclidean spacetime lattice**

2. Calculate inverse of Dirac matrix (which is a function of the gluon links) on each configuration, at fixed times

3. Calculate appropriate trace (gamma matrix structure given by **interpolating operator**)

4. Average results from all configurations

5. Repeat from 2. at different times

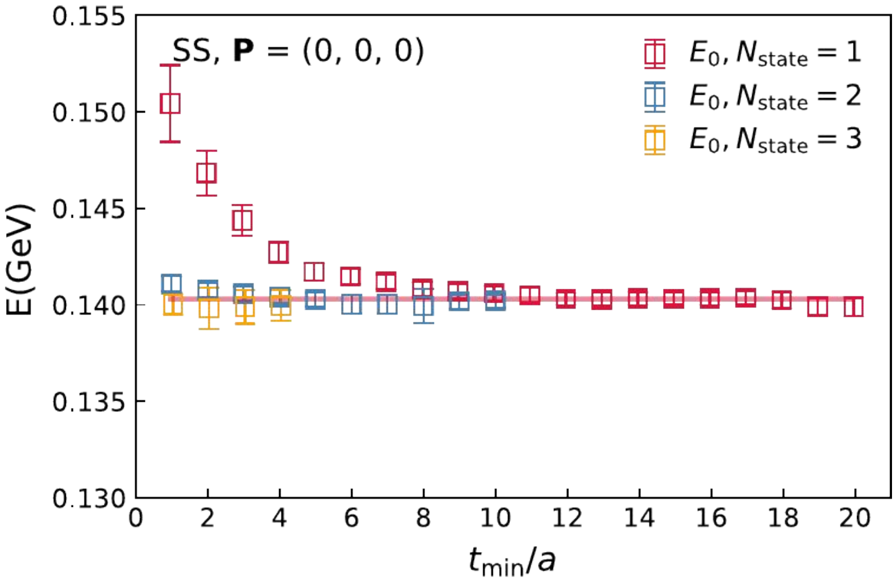
6. Result:

   numerical data (with **statistical uncertainties**) as a function of **Euclidean time**

Sample results

$$\langle 0| \sum_{\mathbf{y}} \mathcal{O}^{\pi}_{\mathbf{y},t} \cdot \mathcal{O}^{\pi}_{\mathbf{x},0} |0\rangle \approx \frac{1}{N_{\text{conf}}} \sum_{m=1}^{N_{\text{conf}}} \text{Tr}\left[\gamma_5 D^{-1}_{\mathbf{y}t,\mathbf{x}0} \gamma_5 D^{-1}_{\mathbf{x}0,\mathbf{y}t}\right]$$

Example: the pion two-point function

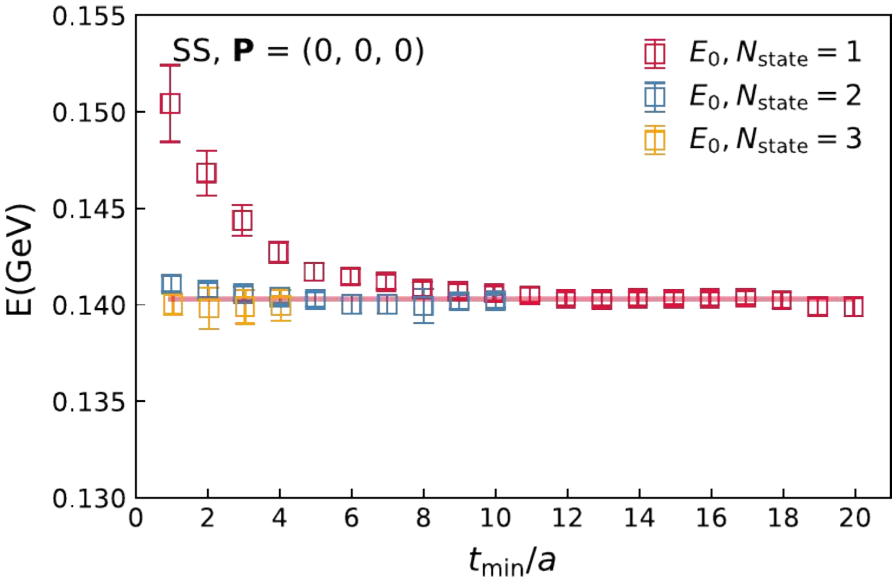


Gao et al., PRD 104 (2021) 114515

Sample results

$$\langle 0|\sum_{\mathbf{y}} \mathcal{O}^{\pi}_{\mathbf{y},t}\cdot\mathcal{O}^{\pi}_{\mathbf{x},0}|0\rangle \approx \frac{1}{N_{\rm conf}}\sum_{m=1}^{N_{\rm conf}} \mathrm{Tr}\left[\gamma_5 D^{-1}_{\mathbf{y}t,\mathbf{x}0}\gamma_5 D^{-1}_{\mathbf{x}0,\mathbf{y}t}\right]$$

Example: the pion two-point function

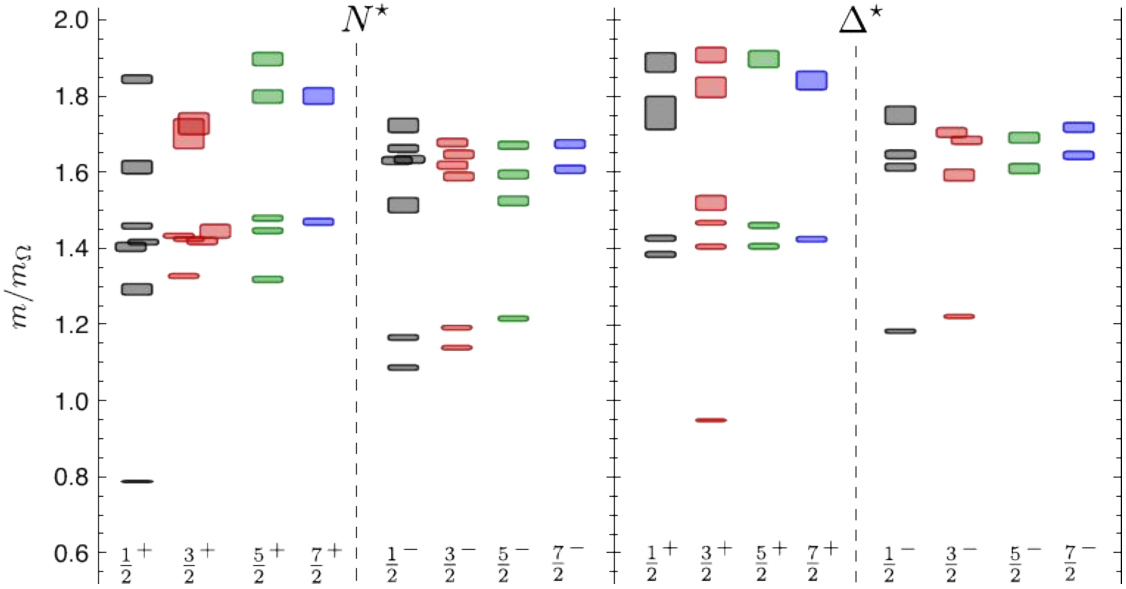


$$\begin{aligned} C_2(t) &= \langle 0|\mathcal{O}^{\pi}_t\cdot\mathcal{O}^{\pi}_0|0\rangle \\ &= \langle 0|e^{tH}\mathcal{O}^{\pi}_0 e^{-tH}\cdot\mathcal{O}^{\pi}_0|0\rangle \\ &= \langle 0|\mathcal{O}^{\pi}_0 e^{-tH}\sum_n |n\rangle\langle n|\mathcal{O}^{\pi}_0|0\rangle \\ &= \sum_n e^{-tE_n}\langle 0|\mathcal{O}^{\pi}_0|n\rangle\langle n|\mathcal{O}^{\pi}_0|0\rangle \\ &= \sum_n e^{-tE_n}|A_n|^2 \overset{t\to\infty}{\longrightarrow} A_0 e^{-m_\pi t} \end{aligned}$$

Gao et al., PRD 104 (2021) 114515

Sample results: two-point functions and the meson spectrum

Example: an excited baryon spectrum



had spec

GLUEX

Edwards et al. (HadSpec), PRD 84 (2011) 074508

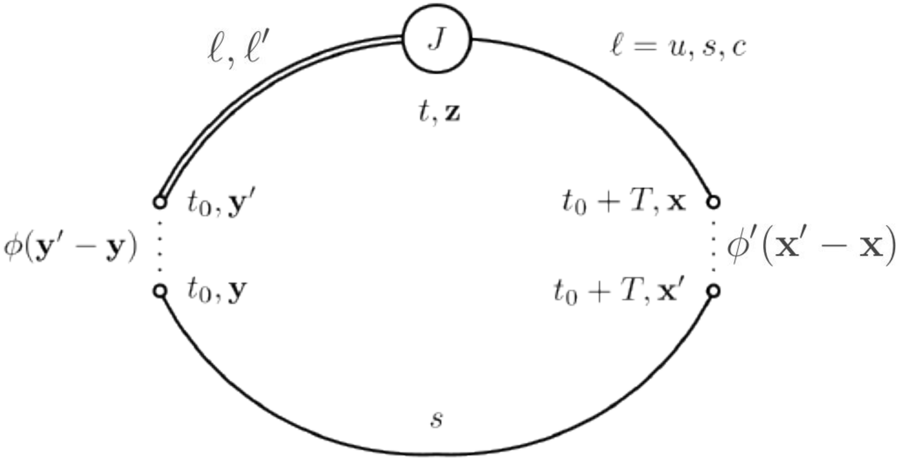
Interacting with hadrons on the lattice

Current *J* can be a simple quark bilinear, like a vector or axial-vector current, or a more complicated object, like a spatially-extended Wilson line.

Two point functions encode a lot of physics and provide rich information on the hadron spectrum

But if we want our hadrons to interact with external forces, we need **three-point functions**

insert a probe operator (or *current*) with the relevant quantum numbers



$$\begin{aligned} C_2(t) &= \langle 0|\mathcal{O}_t^{\pi} \cdot \mathcal{O}_0^{\pi}|0\rangle \\ &= \sum_n e^{-tE_n} \langle 0|\mathcal{O}_0^{\pi}|n\rangle\langle n|\mathcal{O}_0^{\pi}|0\rangle \\ &= \sum_n e^{-tE_n}|A_n|^2 \end{aligned}$$

Spectral analysis of three-point functions more complicated, follows the same logic as two-point function

Physics typically encoded in **matrix elements**, which capture overlap of operator with energy eigenstates

Interacting with hadrons on the lattice

Parton distribution functions (PDF) capture the longitudinal momentum structure of hadrons. In essence, they answer the question: what fraction of the momentum of a fast-moving hadron is carried by an individual quark or gluon?

Example: the unpolarised gluon PDF of the nucleon

Need to calculate the three-point function

$$\langle C_{3pt}(t, t_g)\rangle = \langle 0|T\{O_N(t)\, O_g(t_g)\, \bar{O}_N(0)\}|0\rangle$$

of the gluon operator

$$O_g(z) = G_{ji}(z)\, U(z,0)\, G_{ij}(0)\, U(0,z) - \ G_{ti}(z)\, U(z,0)\, G_{it}(0)\, U(0,z)$$

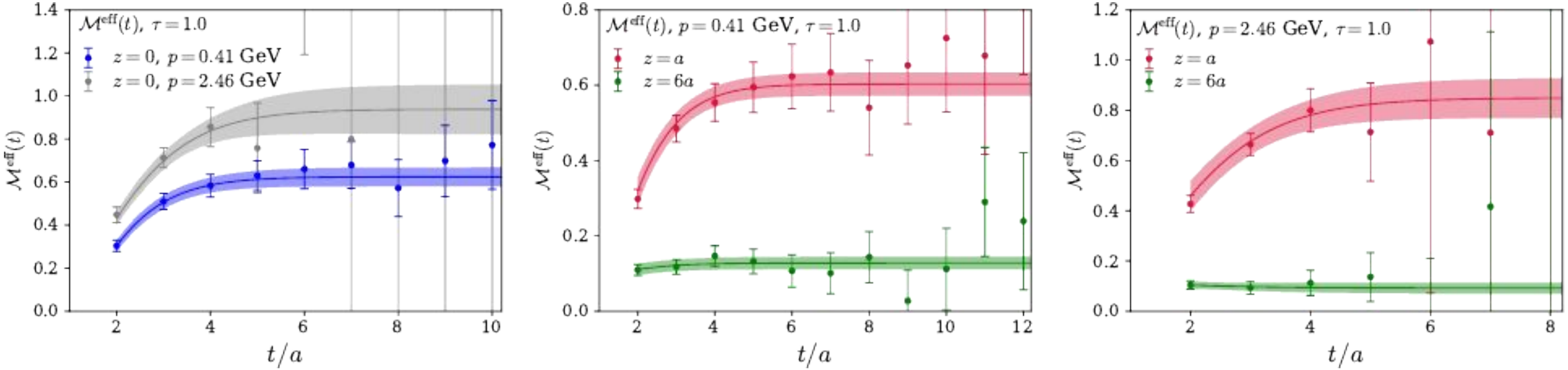
with *U(x,y)* the Wilson line between *x* and *y*, in the adjoint representation.

Khan et al. (HadStruc), PRD 104 (2021) 094516

Interacting with hadrons on the lattice

Parton distribution functions (PDF) capture the longitudinal momentum structure of hadrons. In essence, they answer the question: what fraction of the momentum of a fast-moving hadron is carried by an individual quark or gluon?

Example: the unpolarised gluon PDF of the nucleon

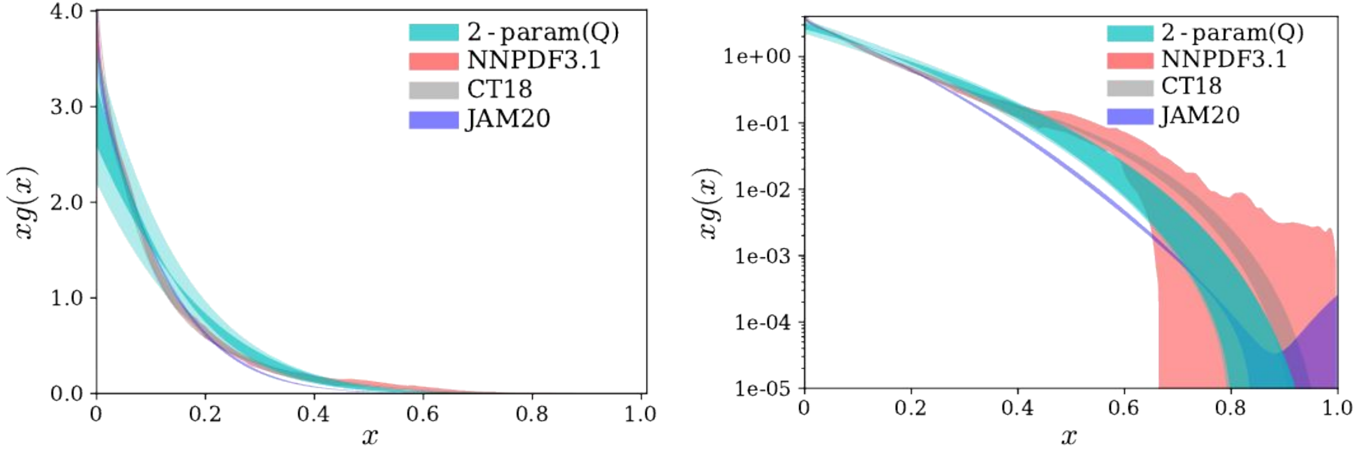


Khan et al. (HadStruc), PRD 104 (2021) 094516

Interacting with hadrons on the lattice

Parton distribution functions (PDF) capture the longitudinal momentum structure of hadrons. In essence, they answer the question: what fraction of the momentum of a fast-moving hadron is carried by an individual quark or gluon?

Example: the unpolarised gluon PDF of the nucleon



Khan et al. (HadStruc), PRD 104 (2021) 094516

Summary: lattice calculation procedure

1. Generate an **ensemble** of gauge **configurations** (sets of gluon link variables) on a **finite volume**, **discretised**, **Euclidean spacetime lattice**
2. Calculate inverse of Dirac matrix (which is a function of the gluon links) on each configuration, at fixed times
3. Calculate appropriate trace (gamma matrix structure given by **interpolating operator**)
4. Average results from all configurations
5. Repeat from 2. at different times
6. Result:

   numerical data (with **statistical uncertainties**) as a function of **Euclidean time**

7. Extract physics from noisy data at large Euclidean times

statistical uncertainties

Summary: lattice systematic uncertainties

finite volume effects

1. Generate an **ensemble** of gauge **configurations** (sets of gluon link variables) on a **finite volume**, **discretised**, **Euclidean spacetime lattice**
2. Calculate inverse of Dirac matrix (which is a function of the gluon links) on each configuration, at fixed times
3. Calculate appropriate trace (gamma matrix structure given by **interpolating operator**)
4. Average results from all configurations
5. Repeat from 2. at different times
6. Result: numerical data (with **statistical uncertainties**) as a function of **Euclidean time**
7. Extract physics from noisy data at large Euclidean times

lattice spacing effects

cannot access real-time or light-cone quantities

signal-to-noise issues

excited state contamination

Summary: lecture 1

- Lattice field theory provides rigorous definition of nonperturbative QFT
- QCD describes fundamental interactions of the strong nuclear force in terms of quarks and gluons
- Lattice QCD formulated with quarks on the lattice sites and *SU(3)* link variables representing gluons
- Gluon action built from closed loops of link variables, such as plaquettes
- Fermions on the lattice suffer from fermion doubling; intimately connected to existence of chiral fermions in a regulated theory
- Different discretisations possible, but any choice should lead to QCD in the continuum limit of vanishing lattice spacing

Summary: lecture 2

- Numerical evaluation of high-dimensional integrals requires **Markov chain Monte Carlo**
- Markov chain Monte Carlo consists of two basic steps
  - propose local update
  - accept/reject update
- Local-update algorithms suffer large **autocorrelations** – use **global-update** algorithms
- Global-update algorithms work well near phase transitions

Summary: lecture 3

- Gauge configuration algorithms
  - Metropolis, hybrid Monte Carlo
- Basic lattice procedure:
  1. Generate configurations
  2. Calculate correlation functions (inverting the Dirac operator)
  3. Average over configurations to provide estimate of path integral
  4. Extract physics from long Euclidean-time limit
- Two-point functions encode hadron spectrum
- Interactions with external forces given by three-point functions
- x-dependent hadron structure requires more complicated operator insertions

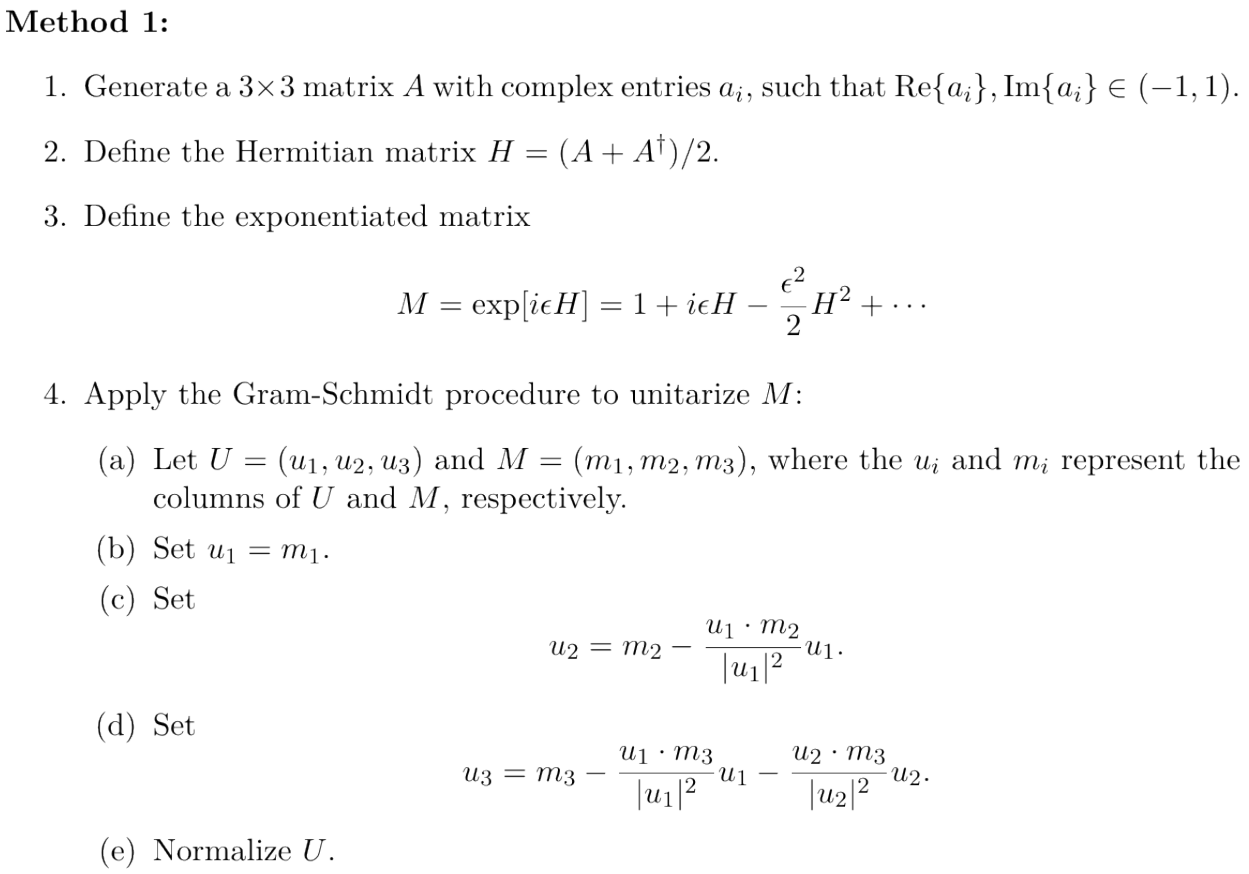
Thank you

Questions?

You can find me at

- cjmonahan@wm.edu
- www.cjmonahan.net

Generating *SU(3)* matrices



Method 1:

1. Generate a 3×3 matrix A with complex entries a_i, such that $\mathrm{Re}\{a_i\}, \mathrm{Im}\{a_i\} \in (-1, 1)$.

2. Define the Hermitian matrix $H = (A + A^\dagger)/2$.

3. Define the exponentiated matrix

$$M = \exp[i\epsilon H] = 1 + i\epsilon H - \frac{\epsilon^2}{2} H^2 + \cdots$$

4. Apply the Gram-Schmidt procedure to unitarize M:

   (a) Let $U = (u_1, u_2, u_3)$ and $M = (m_1, m_2, m_3)$, where the u_i and m_i represent the columns of U and M, respectively.

   (b) Set $u_1 = m_1$.

   (c) Set

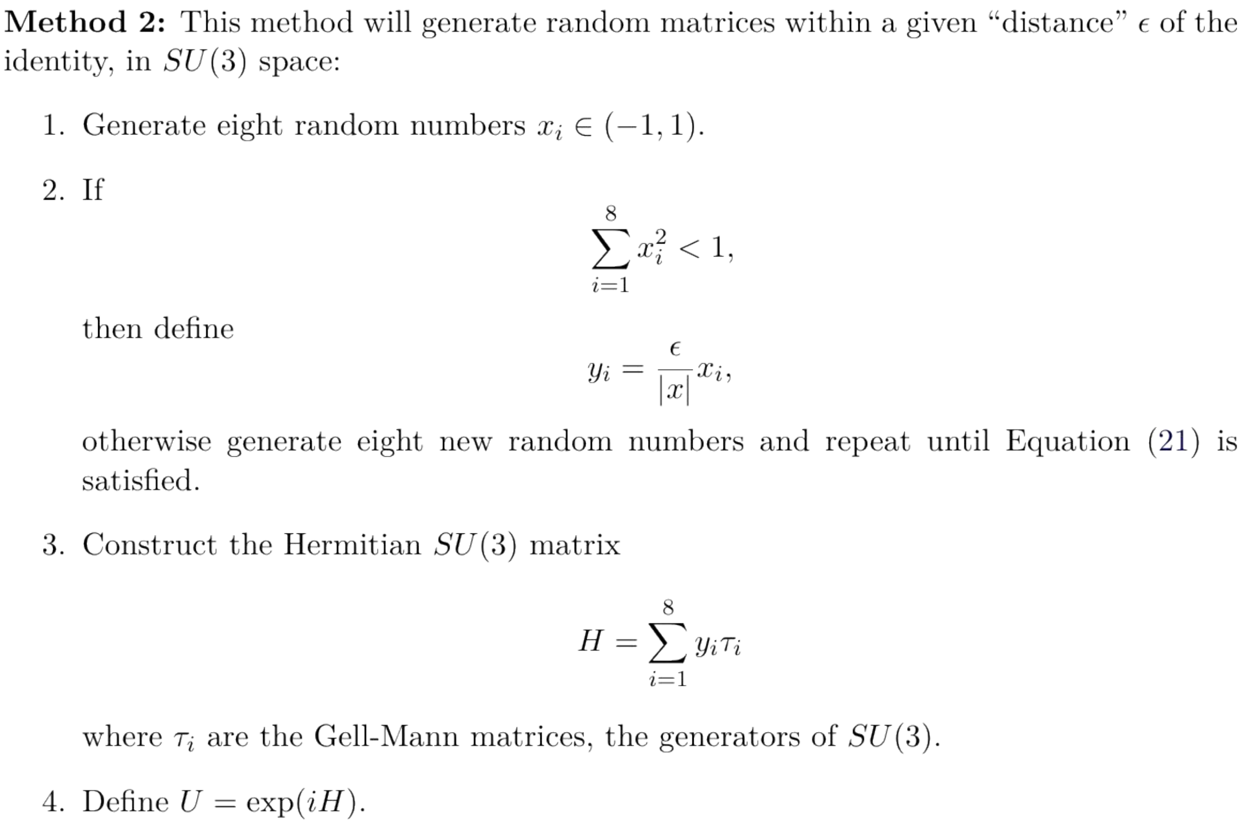
$$u_2 = m_2 - \frac{u_1 \cdot m_2}{|u_1|^2} u_1.$$

   (d) Set

$$u_3 = m_3 - \frac{u_1 \cdot m_3}{|u_1|^2} u_1 - \frac{u_2 \cdot m_3}{|u_2|^2} u_2.$$

   (e) Normalize U.

Generating *SU(3)* matrices



Method 2: This method will generate random matrices within a given "distance" ϵ of the identity, in $SU(3)$ space:

1. Generate eight random numbers $x_i \in (-1, 1)$.
2. If
$$\sum_{i=1}^{8} x_i^2 < 1,$$
then define
$$y_i = \frac{\epsilon}{|x|} x_i,$$
otherwise generate eight new random numbers and repeat until Equation (21) is satisfied.
3. Construct the Hermitian $SU(3)$ matrix
$$H = \sum_{i=1}^{8} y_i \tau_i$$
where τ_i are the Gell-Mann matrices, the generators of $SU(3)$.
4. Define $U = \exp(iH)$.

Heatbath algorithm

Basic procedure:

1. Visit each link
2. Replace variable with probability

$$P(U) \propto \exp\left[\beta \,\mathrm{Re}\,\mathrm{Tr}\, X^\dagger U\right]$$

In essence: approximate local region as a heat bath with thermal distribution governed by the "local action"

For *SU(2)*

- Creutz-Pendleton-Kennedy algorithm
- works very well

Creutz, PRD 21 (1980) 2308
Pendleton & Kennedy, PLB 143 (1984) 459

For *SU(3)*

- algorithm exists but is impractical
- "pseudo-heatbath" preferable
- "overrelaxation" methods even better

Pietarinen, NPB 190 (1981) 349

Cabibbo & Marinari, PLB 119 (1982) 387

Adler, PRD 23 (1981) 2901
de Forcrand & Jahn, hep-lat/0503041

Leapfrog algorithm

Split Hamiltonian

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

Solve time evolution separately

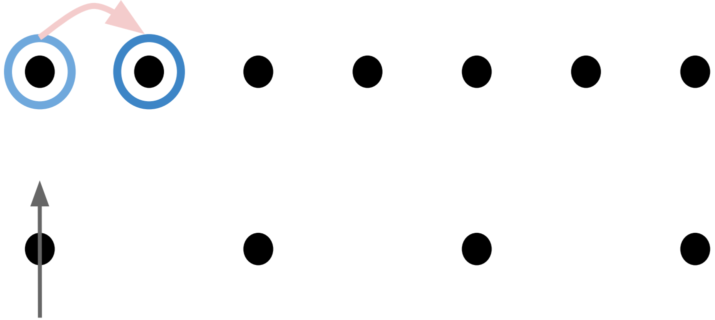
$$T_U = e^{\tau\widehat{T}} \qquad T_\pi = e^{\tau\widehat{S}}$$

$$\begin{aligned} U(x,\tau) &= e^{\pi\tau}U(x,0) \\ \pi(\tau) &= \pi(0) \end{aligned} \qquad \begin{aligned} U(x,\tau) &= U(x,0) \\ \pi(\tau) &= \pi(0) - \tau F \end{aligned}$$

Update steps can be carried out in any order

Leapfrog algorithm

a. Evolve momenta by one half time-step
b. Evolve link variables by one time step
c. Repeat for desired number of iterations, ending with a.



Leapfrog algorithm

Split Hamiltonian

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

Solve time evolution separately

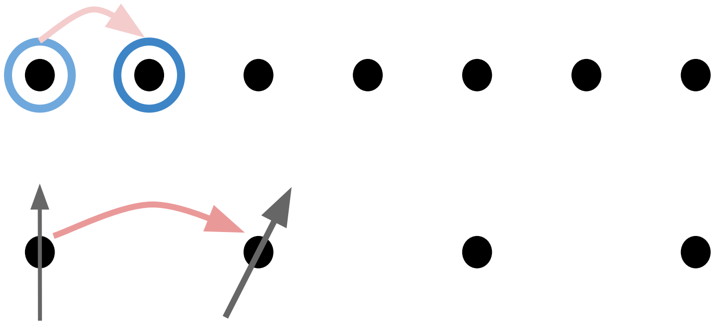
$$T_U = e^{\tau\widehat{T}} \qquad T_\pi = e^{\tau\widehat{S}}$$

$$\begin{aligned} U(x,\tau) &= e^{\pi\tau}U(x,0) \\ \pi(\tau) &= \pi(0) \end{aligned} \qquad \begin{aligned} U(x,\tau) &= U(x,0) \\ \pi(\tau) &= \pi(0) - \tau F \end{aligned}$$

Update steps can be carried out in any order

Leapfrog algorithm

a. Evolve momenta by one half time-step
b. Evolve link variables by one time step
c. Repeat for desired number of iterations, ending with a.



Leapfrog algorithm

Split Hamiltonian

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

Solve time evolution separately

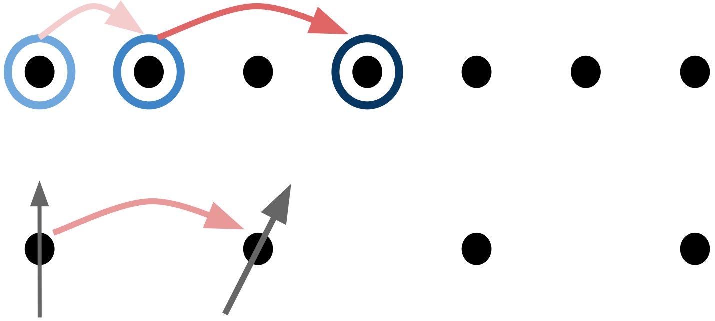
$$T_U = e^{\tau \widehat{T}} \qquad\qquad T_\pi = e^{\tau \widehat{S}}$$

$$\begin{aligned} U(x,\tau) &= e^{\pi\tau}U(x,0) & \qquad U(x,\tau) &= U(x,0) \\ \pi(\tau) &= \pi(0) & \qquad \pi(\tau) &= \pi(0) - \tau F \end{aligned}$$

Update steps can be carried out in any order

Leapfrog algorithm

a. Evolve momenta by one half time-step
b. Evolve link variables by one time step
c. Repeat for desired number of iterations, ending with a.



Leapfrog algorithm

Split Hamiltonian

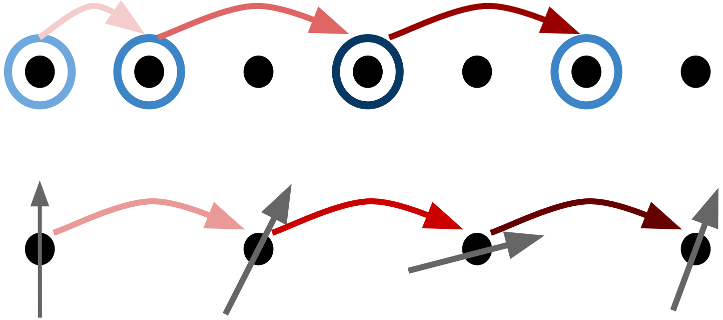
$$\begin{tikzcd}[column sep=small]
& H[\pi, U] = \frac{1}{2}\pi_\mu(x)\pi^\mu(x) + S[U(x)] = T[\pi(x)] + S[U(x)] \arrow[dl, "T_U = e^{\tau\widehat{T}}"] \arrow[dr, "T_\pi = e^{\tau\widehat{S}}"] & \\
\begin{array}{c} U(x,\tau) = e^{\pi\tau}U(x,0) \\ \pi(\tau) = \pi(0) \end{array} & & \begin{array}{c} U(x,\tau) = U(x,0) \\ \pi(\tau) = \pi(0) - \tau F \end{array}
\end{tikzcd}$$

Solve time evolution separately

Update steps can be carried out in any order

Leapfrog algorithm

a. Evolve momenta by one half time-step
b. Evolve link variables by one time step
c. Repeat for desired number of iterations, ending with a.



Leapfrog algorithm

Split Hamiltonian

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

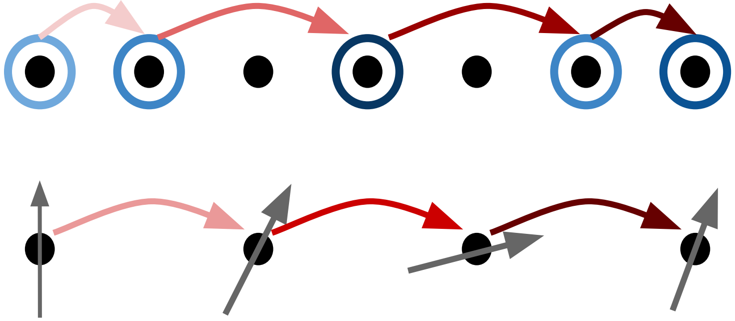
Solve time evolution separately

$$T_U = e^{\tau \widehat{T}} \qquad T_\pi = e^{\tau \widehat{S}}$$

$$\begin{aligned} U(x,\tau) &= e^{\pi\tau}U(x,0) \qquad & U(x,\tau) &= U(x,0) \\ \pi(\tau) &= \pi(0) & \pi(\tau) &= \pi(0) - \tau F \end{aligned}$$

Update steps can be carried out in any order

Leapfrog algorithm



a. Evolve momenta by one half time-step
b. Evolve link variables by one time step
c. Repeat for desired number of iterations, ending with a.

Improved Hybrid Monte Carlo

Leapfrog algorithm is robust but not optimal

Accept/reject rate depends on magnitude of the error in the total energy

Leapfrog time evolution (second order integrator)

$$T = \left[T_U\left(\frac{\epsilon}{2}\right)T_\pi\left(\epsilon\right)T_U\left(\frac{\epsilon}{2}\right)\right]^N$$

Omelyan integrator

$$T = \left[T_\pi\left(\epsilon\lambda\right)T_U\left(\frac{\epsilon}{2}\right)T_\pi\left(\epsilon(1-2\lambda)\right)T_U\left(\frac{\epsilon}{2}\right)T_\pi\left(\epsilon\lambda\right)\right]^{N/2}$$

λ = 0.19 performs about twice as well as the leapfrog algorithm, even though the optimised integrator has twice the computational cost

Actually, the optimal value is apparently

$$\lambda_c = \frac{1}{2} - \frac{(2\sqrt{326}+36)^{1/3}}{12} + \frac{1}{(6\sqrt{326}+36)^{1/3}} \approx 0.1931833275037836$$

See Kennedy, Silva & Clark, PRD 87 (2012) 034511