Introduction to Lattice QCD algorithms

Urs Wenger (ETH Zürich)

LAP06 – 27 November 2006

Functional integrals and QFT

- The expecation value of an operator $\mathcal{O}$ is defined non-perturbatively by the functional integral

$$\langle \mathcal{O} \rangle \equiv \frac{1}{Z} \int (\mathcal{D}\phi)\, \mathrm{e}^{-S[\phi]} \mathcal{O}[\phi],$$

  - normalisation constant Z is chosen such that $\langle 1 \rangle = 1$,
  - $\mathcal{D}\phi$ is the appropriate functional measure,
  - $S[\phi]$ is the action.
- In QFT there is one integration per degree of freedom:
  - we are dealing with an infinite dimensional functional integral,
  - well-defined only in Euclidean space-time using a lattice regularisation and a finite volume.

Monte Carlo methods I

- Lattice regularisation:
  - continuum limit (lattice spacing $a \to 0$) and thermodynamic limit (physical volume $V \to \infty$) are necessary,
  - still dealing with hopelessly many integrations...
- Monte Carlo integration is based on the identification of *probabilities* with *measures*:

$$\Rightarrow \text{importance sampling}$$

  - generate a sequence of random field configurations $\{\phi_1, \phi_2, \ldots \phi_N\}$ chosen from the probability distribution

$$P(\phi_t)\mathcal{D}\phi_t = \frac{1}{Z} e^{-S[\phi_t]},$$

Monte Carlo methods II

- measure the value of $\mathcal{O}$ on each configuration and compute the average
$$\overline{\mathcal{O}} \equiv \frac{1}{N} \sum_{t=1}^{N} \mathcal{O}[\phi_t].$$
- Limit of large numbers guarantees
$$\langle \mathcal{O} \rangle = \lim_{N \to \infty} \overline{\mathcal{O}}.$$
- Central limit theorem guarantees $\langle \mathcal{O} \rangle \sim \overline{\mathcal{O}} + O(\sqrt{\frac{\sigma}{N}})$,
  - where the variance of the distribution of $\mathcal{O}$ is $\sigma \equiv \langle (\mathcal{O} - \langle \mathcal{O} \rangle)^2 \rangle$.

Markov chains I

- Use Markov process to generate the correct probability distribution.
- Consider space of configurations Ω together with (ergodic) stochastic transitions $P' : \Omega \to \Omega$.
- The deterministic evolution of probability distributions $P : Q \to Q$ is a *Markov process*.
- Distribution converges to a unique fixed point:
  - Define a metric on the space of probability distributions,
  - show Markov process is a contraction mapping,
  - the sequence $Q, PQ, P^2Q, P^3Q, \ldots$ is Cauchy,
  - space of probability distributions is complete, so

$$\overline{Q} = \lim_{n \to \infty} P^n Q$$

is the unique fixed point.

Markov chains II

- Suppose ergodic Markov process P with $\overline{Q}$ as its fixed point.
- Use Markov chains to sample from $\overline{Q}$:
  - start with an arbitrary state,
  - iterate the Markov process until it has converged, ('thermalised')
  - thereafter, successive configurations will be distributed according to $\overline{Q}$.
- To construct P we only need *relative probabilitites* of states:
  - we don't know the normalisation of Q,
  - can only compute ratios of integrals,
  - cannot use Markov chains to compute integral directly.

Markov chains III

- How to construct a Markov process with specified $\overline{Q}$?
  - Detailed balance $P(y \leftarrow x)\overline{Q}(x) = P(x \leftarrow y)\overline{Q}(y)$
    - $\Rightarrow$ sufficient but not necessary.
  - Metropolis algorithm $P(x \leftarrow y) = \min\left(1, \frac{\overline{Q}(x)}{\overline{Q}(y)}\right)$
    - $\Rightarrow$ sufficient but not necessary for detailed balance.
  - Other choices are possible, e.g. $P(x \leftarrow y) = \frac{\overline{Q}(x)}{\overline{Q}(y)+\overline{Q}(x)}$.
- Markov steps P_1, P_2 with the same fixed point distribution can be combined $P_1 \circ P_2$
  - $P_1 \circ P_2$ may be ergodic, even if P_1 and P_2 are not.

Hybrid Monte Carlo I

- In order to carry out Monte Carlo computations we want an algorithm which
  - updates the fields globally,
    - $\rightarrow$ since single updates are expensive for non-local actions,
  - takes large steps through configuration space,
    - $\rightarrow$ in order to decorrelate successive configuration
  - does not introduce any systematic errors.

Hybrid Monte Carlo II

- A useful class of algorithms with these properties is the Hybrid Monte Carlo (HMC) method:
  - introduce 'fictitious' momentum p conjugate to each dynamical degree of freedom q,
  - find a Markov chain with fixed point $\propto \exp[-H(p,q)]$ where

$$H(p,q) = \frac{1}{2}p^2 + S(q)$$

    is the 'fictitious' Hamiltonian:
    - the action $S(q)$ of the underlying QFT plays the role of the potential in the 'fictitious' classical mechanics system,
    - the Hamiltonian gives the evolution of the system in a fifth dimension, 'fictitious' or MC time.
  - This generates the desired distribution $\exp[-S(q)]$ if we ignore the momenta p.

Hybrid Monte Carlo III

- The HMC Markov chain alternates two Markov steps.
  - Molecular Dynamics Monte Carlo:
    - exact integration of Hamilton's equations gives a trajectory of constant 'fictitious energy'
      $\rightarrow$ equiprobable fictitious phase space configurations,
    - approximate integration must be *reversible* and *area preserving*,
    - the so produced phase space configurations have to pass a *Metropolis accept/reject step* with acceptance $\min[1, \exp(-\delta H)]$
  - (Partial) Momentum refreshment.
- Both steps have the desired fixed point.
- Together they are ergodic.

Note: the Metropolis test makes the algorithm exact even for approximate integration.

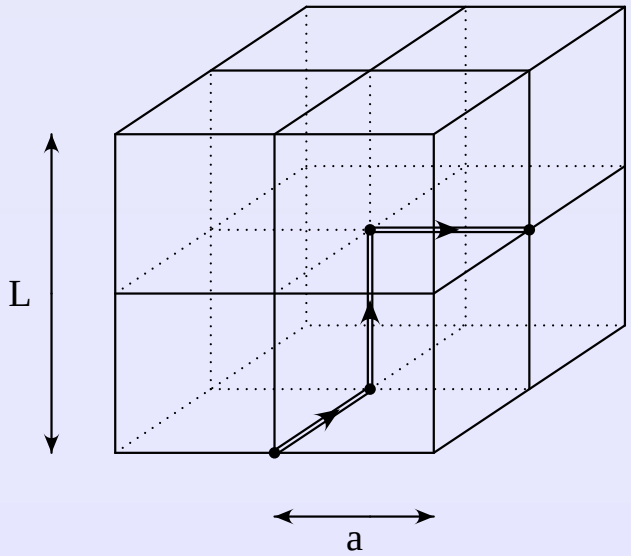
QCD on the Lattice

Quantumchromodynamics is formally described by the Lagrange density:

$$\mathcal{L}_{\mathrm{QCD}} = \bar{\psi}(i\not{D} - m_q)\psi - \frac{1}{4} G_{\mu\nu} G^{\mu\nu}$$

Lattice regularization: discretize Euclidean space-time

- hypercubic L^4-lattice with lattice spacing a
- derivatives $\Rightarrow$ finite differences
- integrals $\Rightarrow$ sums
- gauge potentials A_μ in $G_{\mu\nu} \Rightarrow$ link matrices U_μ ('  ')



QCD on the Lattice II

- Partition function $Z_{\mathrm{QCD}} = \int \left(\mathcal{D}U\mathcal{D}\overline{\psi}\mathcal{D}\psi\right) \mathrm{e}^{-S_{\mathrm{QCD}}[U;\overline{\psi},\psi]}$
- Mathematically well defined theory
- Non-perturbative, gauge invariant regularisation (low energy physics)
- Continuum limit $\Rightarrow a \longrightarrow 0$
  - Poincaré symmetries are restored automatically
- Direct simulation of Grassmann fields is not feasible.
  - The problem is not that of manipulating anticommuting values in a computer.
  - It is that $e^{-S_F} = e^{-\bar{\psi}D\psi}$ is not positive and thus we get poor importance sampling.

QCD on the Lattice III

- We therefore integrate out the fermion fields to obtain the fermion determinant $\int \mathcal{D}\psi\mathcal{D}\bar{\psi}e^{-\bar{\psi}D\psi} \propto \det(D)$:

$$Z = \int (\mathcal{D}U) \det D(U) \mathrm{e}^{-S_G[U]}$$

  - ψ and $\bar{\psi}$ always occur quadratically,
  - the overall sign of the exponent is unimportant.
- Any operator $\mathcal{O}$ can be expressed in terms of the bosonic fields

$$\mathcal{O}'(U) = \mathcal{O}\left(\frac{\delta}{\delta\psi}, \frac{\delta}{\delta\bar{\psi}}; U\right) e^{-\bar{\psi}D\psi}\Big|_{\psi=\bar{\psi}=0}$$

e.g. the fermion propagator is $\langle\psi(x)\bar{\psi}(y)\rangle = D^{-1}(x,y)$.

Dynamical fermions I

- Pseudofermions:
  - Represent the fermion determinant as a bosonic Gaussian integral with a non-local kernel

$$\det D(U) \propto \int \mathcal{D}\phi \mathcal{D}\bar{\phi} e^{-\bar{\phi} D^{-1}(U) \phi}.$$

  - The fermion kernel must be positive definite for the bosonic integral to converge.
  - The new bosonic fields are called *pseudofermions*.
  - It is usually convenient to introduce two flavours of fermions and to write

$$(\det D(U))^2 = \det \left(D(U)^\dagger D(U) \right) \propto \int \mathcal{D}\phi \mathcal{D}\bar{\phi} e^{-\bar{\phi} \left(D(U)^\dagger D(U) \right)^{-1} \phi}$$

Dynamical fermions II

- Introduce Gaussian momenta $P_\mu(x)$ conjugate to $U_\mu(x)$ with action $S_P = 1/2 \sum_{\mu,x} P_\mu(x)^2$.
- The new partition function is now

$$Z = \int \left(\mathcal{D}U\mathcal{D}\overline{\phi}\mathcal{D}\phi\right) \mathrm{e}^{-S_P - S_g[U] - \overline{\phi}\frac{1}{D^\dagger(U)D(U)}\phi}$$

- Generate a sequence of $\{P, U\}$ with the correct probability distribution:
  - update $P_\mu(x)$ using Gaussian random noise,
  - update ϕ using Gaussian random noise via $\phi = D^\dagger \eta$,
  - evolve $\{P, U\}$ according to the Hamiltonian

$$\mathcal{H}[P, U] = \frac{1}{2}P^2 + S_g[U] + S_f[U]$$

  - accept/reject the final configuration $\{P', U'\}$ with probability

$$P_{\text{accept}} = \min\left(1, e^{-\left(\mathcal{H}[P',U'] - \mathcal{H}[P,U]\right)}\right).$$

Dynamical fermions III

- The discrete Hamiltonian equations of motion dictate the following update for U and P,

$$\begin{aligned} T_U(\delta\tau): \quad U &\Longrightarrow e^{i\delta\tau P}U \\ T_P(\delta\tau): \quad P &\Longrightarrow P + \delta\tau \cdot F \end{aligned}$$

  where F is the force due to the variation of the gauge field:

$$F = -\frac{\delta\mathcal{H}}{\delta U}.$$

- Inversions required for the equations of motion need not be exact – integration is approximate anyway.
- Inversion for the Metropolis accept/reject step needs to be 'exact'.

Dynamical fermions IV

- Are HMC trajectories reversible and area preserving?
  - Yes, if a leapfrog integration scheme is used:

$$T(\delta\tau) = T_P(\delta\tau/2)T_U(\delta\tau)T_P(\delta\tau/2).$$

  - Recently, so-called Omelian integrators were suggested which are also reversible and area preserving, but appear to be more efficient.
  - The only fundamental source of irreversibility is the rounding error caused by using finite precision floating point arithmetic.
- The evaluation of the pseudofermion action and the corresponding force requires the solution of a (large) set of linear equations $\left(D^\dagger D\right)^{-1}\phi = \chi$.

Iterative methods I

- Consider a system of linear equations $Ax = b$ and the residual vector $r \equiv b - Ax_i$ for an approximate solution x_i.
- Rewriting the system as

$$(I - (I - A))x = b$$

leads to a basic iteration

$$\begin{aligned} x_i &= b + (I - A)x_{i-1} \\ &= x_{i-1} + r_{i-1} \\ &= x_{i-2} + r_{i-2} + r_{i-1} \\ &\vdots \\ &= x_0 + r_0 + r_1 + \ldots + r_{i-1}. \end{aligned}$$

Iterative methods II

- Multiply $x_i = x_{i-1} + r_{i-1}$ with A from the left

$$Ax_i = Ax_{i-1} + Ar_{i-1}$$

and subtract from b

$$\begin{aligned} b - Ax_i &= b - Ax_{i-1} + Ar_{i-1} \\ r_i &= r_{i-1} - Ar_{i-1} \\ &= (I - A)r_{i-1} \end{aligned}$$

- So finally we have

$$\begin{aligned} x_i &= x_0 + r_0 + (I - A)r_0 + \ldots + (I - A)^{i-1} r_0 \\ &= x_0 + [r_0, Ar_0, A^2 r_0, \ldots A^{i-1} r_0]. \end{aligned}$$

Krylov spaces I

- This linear space defines the Krylov subspace

$$\mathcal{K}_n(A; r_0) \equiv \text{span}(r_0, Ar_0, A^2 r_0, A^3 r_0, \ldots, A^{n-1} r_0).$$

- Iterative methods are based on finding an approximate solution x_n within a Krylov subspace

$$\Rightarrow \text{Krylov subspace methods}$$

- Convergence is measured by the *residual* $r_n = ||b - Ax_n||$.

Krylov spaces II

- Krylov subspace methods are often used as exact methods:
  - they require $O(V)$ iterations to find the solution,
  - they do not give the 'exact' answer in practice due to rounding errors,
  - they are more naturally thought of as methods for solving systems of linear equations in an (almost) ∞-dimensional linear space.

Krylov spaces III

- Approximations obtained from a Krylov subspace method are of the form

$$A^{-1}b \sim x_n = x_0 + P_{n-1}(A)r_0$$

  where P_{n-1} ia a polynomial of degree $n-1$
- For the simple case $x_0 = 0$ we have

$$A^{-1}b \sim P_{n-1}(A)b$$

  i.e. $P_{n-1}(A)$ is a polynomial approximation of $A^{-1}b$.
- All techniques provide the same type of polynomial approximations, but the type of constraints has important effects.

Krylov spaces IV

- More specifically, we seek an approximate solution x_n in $\mathcal{K}_n$ by imposing the Petrov-Galerkin projection condition

$$r_n \equiv b - Ax_n \perp \mathcal{L}_n$$

  where $\mathcal{L}_n$ is another n-dimensional subspace.
- Two broad choices:
  - $\mathcal{L}_n = \mathcal{K}_n(A; r_0)$ or $\mathcal{L}_n = A\mathcal{K}_n(A; r_0) \Leftrightarrow$ orthogonalisation $\rightarrow$ FOM (Arnoldi), GMRES, CG, GCR,...
  - $\mathcal{L}_n = \mathcal{K}_n(A^\dagger; r_0) \Leftrightarrow$ bi-orthogonalisation $\rightarrow$ Lanczos, BCG, QMR, BiCGstab, ...
- A projection method may have different implementations $\rightarrow$ different, but mathematically equivalent algorithms.

Krylov spaces V

- For the choice $\mathcal{L}_n = \mathcal{K}_n$ one can show that $\{r_0, r_1, \ldots, r_{n-1}\}$ form an orthogonal basis of $\mathcal{K}_n(A; r_0)$.
- Assume A hermitian and $r_0, \ldots, r_n$ orthogonal. Then

$$\gamma_n r_{n+1} = A r_n - \alpha_n r_n - \beta_n r_{n-1} - \delta_n . r_{n-2} - \ldots$$

- Requiring $(r_{n+1}, r_{n-2}) = 0$ leads to 3-term recurrence

$$\gamma_n r_{n+1} = A r_n - \alpha_n r_n - \beta_n r_{n-1}.$$

- Requiring $(r_{n+1}, r_{n-1}) = (r_{n+1}, r_n) = 0$ leads to

$$\begin{aligned} \alpha_n &= (A r_n, r_n)/(r_n, r_n), \\ \beta_n &= (A r_n, r_n)/(r_{n-1}, r_{n-1}). \end{aligned}$$

Krylov spaces VI

- In contrast to the above multi-dimensional projection methods, there also exist 1-dimensional projection processes where

$$\mathcal{K} = \text{span}(w) \quad \text{and} \quad \mathcal{L} = \text{span}(v),$$

  i.e. each iteration step is completely independent of the previous one.

  e.g. Minimal Residual (MR) iteration

- The best and most complete reference is

  Yousef Saad, *Iterative Methods for Sparse Linear Systems*, second edition, SIAM, 2003.

Multi-shift solvers

- Often necessary to solve $(A+\sigma)x = b$ for several shifts σ
  - for obtaining propagators at several masses, $(D+\delta m_i)^{-1}b$,
  - for calculating rational matrix functions $R_{n,n}(A) = \frac{P_n(A)}{Q_n(A)} = \sum_{i=1}^{n} \frac{c_i}{A+\sigma_i}$.
- For Krylov subspace solvers this can be achieved by realising that [Jegerlehner, Frommer et al.]

$$\mathcal{K}(A+\sigma;0) = \mathcal{K}(A;0).$$

  - the solution $(A+\sigma)^{-1}b$ can be obtained with little overhead during the construction of $A^{-1}b$,
  - only a single Krylov space is needed.
- Overall convergence usually governed by the worst conditioned $A+\sigma$.

Generalities

- Most (all) inversion methods suffer from slow convergence.
- Preconditioning is the key ingredient for the success of Krylov subspace methods.
- General strategy is to modify the original linear system which makes it easier to solve by iterative methods:
  - search for preconditioner M which approximates A^{-1}, then solve

$$M^{-1}Ax = M^{-1}b,$$

  - M^{-1} should be easy to calculate.
- Polynomial preconditioners, incomplete factorisations, Schwarz alternating procedure, domain decompositions, ...
- No limits, but good preconditioners usually derived from good knowledge of the physical problem.

Even/odd preconditioning I

- Dirac operators containing only nearest-neighbour interactions can be written as

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

  where D_{ee} and D_{oo} are diagonal.
- Perform a LU-decomposition

$$D = \begin{pmatrix} D_{ee} & 0 \\ D_{oe} & 1 \end{pmatrix} \begin{pmatrix} 1 & D_{ee}^{-1} D_{eo} \\ 0 & D_{oo} - D_{oe} D_{ee}^{-1} D_{eo} \end{pmatrix} = L \cdot U \,.$$

Even/odd preconditioning II

- Invert D by inverting each factor separately,

$$L^{-1} = \begin{pmatrix} D_{ee} & 0 \\ D_{oe} & 1 \end{pmatrix}^{-1} = \begin{pmatrix} D_{ee}^{-1} & 0 \\ -D_{oe} & 1 \end{pmatrix},$$

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

  where $\hat{D} = D_{oo} - D_{oe} D_{ee}^{-1} D_{eo}$.

- Note that $\hat{D}$ is better conditioned and hence cheaper to invert than the original D.

Low-mode preconditioning I

- The vector space on which A acts, can be split into two (bi-)orthogonal pieces using the (bi-)orthogonal projectors

$$P = \sum_k r_k l_k^\dagger, \quad P_\perp = 1 - P$$

  - $r_k's$ and $l_k's$ are approximate right and left eigenvectors,
  - form a bi-orthogonal basis, i.e. $l_i^\dagger r_j = \delta_{ij}$.
- Then A yields the following block form

$$A = \begin{pmatrix} PAP & PAP_\perp \\ P_\perp AP & P_\perp AP_\perp \end{pmatrix}.$$

Low-mode preconditioning II

- Perform a LU decomposition of A

$$A = \begin{pmatrix} 1 & 0 \\ P_\perp A P (PAP)^{-1} & 1 \end{pmatrix} \begin{pmatrix} PAP & PAP_\perp \\ 0 & S \end{pmatrix} \equiv L \cdot U$$

  where $S = P_\perp A P_\perp - P_\perp A P (PAP)^{-1} PAP_\perp$ is the Schur complement of A.
- Invert each factor separately,
  - L^{-1} is easy:

$$L^{-1} = \begin{pmatrix} 1 & 0 \\ -P_\perp A P (PAP)^{-1} & 1 \end{pmatrix}$$

  - U^{-1} requires $(PAP)^{-1}$ which is easy (inversion in a small sub-space) and S^{-1}
- Note that S is better conditioned than original matrix A.

Recapitulation

- Recapitulate:
  - represent the fermion determinant as a bosonic integral,

$$\det\left(D(U)D(U)^\dagger\right) = \int \mathcal{D}\phi\mathcal{D}\phi^\dagger e^{-\phi^\dagger (D(U)D(U)^\dagger)^{-1}\phi},$$

  - introduce Gaussian momenta $P_\mu(x)$ conjugate to $U_\mu(x)$ with action $S_P = 1/2\sum_{\mu,x} P_\mu(x)^2$,
  - evolve P, U according to the Hamiltonian

$$\mathcal{H} = \frac{1}{2}P^2 + S_g[U] + S_f[U]$$

yielding the force

$$F = -\frac{\delta\mathcal{H}}{\delta U} = F_g[U] + F_f[U].$$

Multiple time-steps I

- One observes that $F_g[U] >> F_f[U]$.
- Introduce two time steps:
  - a short one associated with the large, but cheap gauge force $F_g[U]$,
  - a long one associated with the small, but expensive fermionic force $F_f[U]$.
- Moreover, the fermionic force itself can be split into two or more pieces, $F_f[U] = F_f^1[U] + F_f^2[U] + \ldots$ and put on different time scales according to their size.
- Split the force such that the most expensive piece contributes the least.

Multiple time-steps II

- Generically, this is achieved by splitting the fermion Dirac operator into a long-range infrared (IR) part and a short-range ultraviolet (UV) part:
  - UV part is large, but cheap,
  - IR part is small, but expensive.
- In practice, split the fermion determinant into different pieces

$$\det(M) = \det(M_1)\det(M_2)\ldots$$

  and use different pseudo-fermion fields on different time scales.
- How do we obtain the desired splitting?

$\Rightarrow$ look at the preconditionings!

Multiple time-steps III

- Mass preconditioning [Hasenbusch, Urbach et al.]

$$\det\left(D(m)^\dagger D(m)\right) = \det\left(\frac{D(m)^\dagger}{D(M_1)^\dagger}\frac{D(m)}{D(M_1)}\right)\det\left(D(M_1)^\dagger D(M_1)\right)$$

  where m is the physical mass, and $M > m$ the preconditioning mass:
  - force from $D(M_1)^\dagger D(M_1)$ is large, but cheap,
  - force from $\frac{1}{D(M_1)^\dagger} D(m)^\dagger D(m) \frac{1}{D(M_1)}$ is small but expensive.
- Tune M (or possibly $M_1, M_2, \ldots$) such that the forces are optimally arranged in order to apply mutiple time-steps.
- Speed-up factors can be up to 10 even in a physically relevant set-up.

Multiple time-steps III

- Polynomial filtering [Peardon et al.]

$$\det\left(D^\dagger D\right) = \det\left(P\left(D^\dagger D\right)\right) \det\left(D^\dagger D \frac{1}{P\left(D^\dagger D\right)}\right)$$

  - $P(x) \approx \frac{1}{x}$ in the interval $[\mu, \lambda_{\max}\left(D^\dagger D\right)]$.
  - The approximation covers the UV part of $D^\dagger D$.
  - Only a low order polynomial is needed, since μ is large.
  - $P(x)$ is easy to invert and yields a large force contribution.
  - Correction term is still hard to invert, but yields a small force.
- Speed-up factors currently still under investigation.

Multiple time-steps IV

- Domain decomposition [Lüscher]. An ultra-local Dirac operator can be decomposed into different domains

$$D = \begin{pmatrix} D_\Omega & D_{\partial\Omega} \\ D_{\partial\Omega'} & D_{\Omega'} \end{pmatrix}.$$

  - The preconditioning matrix is

$$D' = \begin{pmatrix} D_\Omega & 0 \\ 0 & D_{\Omega'} \end{pmatrix}.$$

    and describes the UV physics.
  - D' is easy to invert and yields a large force.
  - The correction term is expensive to invert, but yields a small force.
- Again, speed up factors up to 10 can be achieved in physically relevant situations.

Multiple time-steps V

- Rational Hybrid Monte Carlo [Clark and Kennedy] writes

$$\det\left(D^\dagger D\right) = \det\left[\left(D^\dagger D\right)^{1/n}\right]^n$$

and uses a rational approximation $R(x) \approx x^{1/n}$.

  - Inverse of R is also a rational function.
  - Use multi-shift solver to calculate R and R^{-1}.
  - Smallest shift is expensive, but contributes a small force.
  - Use coarser time scale for the more expensive smaller shifts.

- Improvement factors are again up to 10 in physically relevant simulation set-ups.

Conclusions

- All the different algorithmic improvements in HMC rely on multi-pseudofermion fields and multiple time scales.
- Together with the usual increase in computer time, the new developments push lattice QCD calculations into new regimes.
- Calculations are now possible which were not possible before.
- Lattice QCD is entering exciting times,

$\Rightarrow$ make sure you participate!