# Polynomial Filtered HMC

Taylor Haar
Collaborators: Waseem Kamleh and James Zanotti

CSSM
University of Adelaide

Lattice Hadron Physics V, 2015

# Outline

# Outline

1. Motivation

2. Intro to PFHMC
   - HMC recap
   - Polynomial-filtered HMC (PFHMC)

3. Analysis
   - PFHMC versus Hasenbusch
   - PF-RHMC

# Motivation

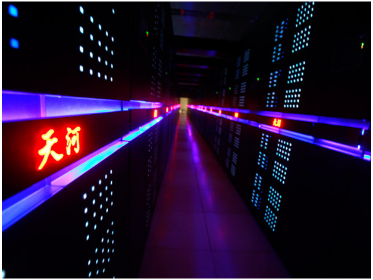
- Lattice QCD is *the* non-perturbative method for determining the behaviour of the strong force
- Progressively faster/better over the years, due in part to advances in computation power but also algorithmic improvements

1974 Cray-I

$\sim$ 100 MFLOPS

$\Rightarrow$

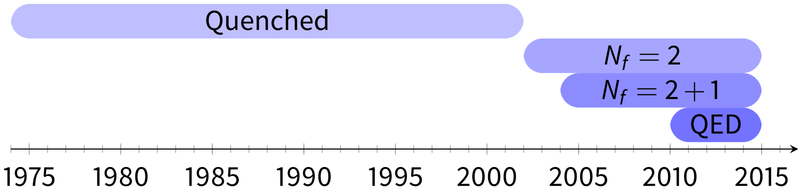
2014 Tianhe-2




33.86 PFLOPS

# Motivation

- As compute power has improved, so has the precision in Lattice QCD measurements




- Search for better algorithms is an active field: simulations still take a long time, so even a 10% speed improvement is great.
- Includes polynomial-filtered Hybrid Monte Carlo (PFHMC).

# Outline

# Expectation values

- In Lattice QCD, we are interested in expectation values

$$\langle O \rangle = \frac{\int \mathscr{D}[\psi, \overline{\psi}, U] e^{-S[\psi, \overline{\psi}, U]} O[U]}{\int \mathscr{D}[\psi, \overline{\psi}, U] e^{-S[\psi, \overline{\psi}, U]}},$$

where

$$S = S_G[U] + S_F'[\psi, \overline{\psi}, U] = S_G + \sum_f \overline{\psi} M^{(f)} \psi$$

is the lattice action and $M^{(f)}$ is the Dirac operator for the $f^{\text{th}}$ fermion flavour.

# Evaluation

- Integration over fermion fields can evaluated via Wick's theorem: e.g.

$$Z \equiv \int \mathscr{D}[\psi, \overline{\psi}, U] e^{-S[\psi,\overline{\psi},U]} = \int \mathscr{D}[U] e^{-S_G[U]} \prod_f \det M^{(f)}$$

- Evaluating $\det M^{(f)}$ directly is computationally infeasible, so we evaluate determinants numerically via pseudo-fermions $\phi$:

$$\det M = \int e^{-S_F[U,\phi,\phi^\dagger]} d\phi d\phi^\dagger, \quad \text{where } S_F = \phi^\dagger M^{-1} \phi.$$

- Have to invert $M$ now, but this is tenable.

# Monte Carlo

- Integration over gauge fields $U$ and pseudo-fermion fields $\phi$ then can be done via Monte Carlo methods.
- We generate $U_i$ and $\phi$ distributed according to the probability distribution

$$\frac{1}{Z} e^{-S_G[U]-S_F[U,\phi,\phi^\dagger]},$$

  then evaluate expectation values as $\langle O \rangle \approx \frac{1}{N} \sum_i O[U_i]$.
- This requires $S_F$ to be real and non-negative: non-trivial as $M$ has a complex spectrum.
- Most common solution is to work with two mass-degenerate quarks, and write

$$\det M^{(u)} \det M^{(d)} = \det M \det M = \det M^\dagger M \equiv \det K$$

$$\text{s.t.} \quad S_F[U] = \phi^\dagger (M^\dagger M)^{-1} \phi$$

# Evaluation

- The target probability distribution then has Boltzmann factor $\exp(-S[U])$ with
$$S[U] = S_G[U] + \phi^\dagger K^{-1} \phi$$
- Generating correctly distributed $\phi$ is easy:
generate $\chi \sim e^{-\chi^\dagger \chi}$ and use $\phi = M^\dagger \chi$.
- Generating correctly distributed $U$ is more involved:
use a Markov chain Monte Carlo method to generate configurations $U_i$.

# Hybrid Monte Carlo

- The method of choice is Hybrid Monte Carlo (HMC).
- Central idea is to extend the action $S[U]$ with conjugate momenta $P$ to a Hamiltonian

$$H[P,U] = \mathrm{Tr}\, P^2 + S[U],$$

then evolve the system according to Hamilton's equations.
- Corresponding update steps are

$$V_T(\delta\tau): \; [P,U] \to [P, e^{iP\delta\tau}U],$$
$$V_S(\delta\tau): \; [P,U] \to [P - F\delta\tau, U].$$

where the force term $F$ is given by

$$F = \frac{dS}{dU} = \frac{dS_G}{dU} - \phi^\dagger K^{-1}\frac{dK}{dU}K^{-1}\phi.$$

# Hybrid Monte Carlo

- Require a reversible, space-preserving integration, e.g. leapfrog:

$$I(\delta\tau) = V_T(\delta\tau/2)V_S(\delta\tau)V_T(\delta\tau/2)$$

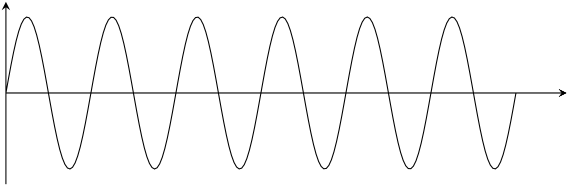
- Repeat $n$ times to generate the next candidate gauge configuration, $[P', U'] = I(\delta\tau)^N[P, U]$.
- We accept the new gauge configuration $U'$ with probability

$$P_{\text{acc}} = \exp\left(H[P', U'] - H[P, U]\right)$$

- This step is necessary to ensure that successive $U_1 \rightarrow U_2 \rightarrow \dots$ approach the required equilibrium distribution.
- Hamilton's equations preserve the Hamiltonian, so we should expect a good acceptance rate.

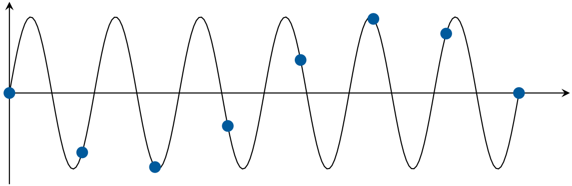
# Implementation

- Main computational expense is in calculating $K^{-1}\phi$, i.e. solving $K\chi = \phi$ for $\chi$, for the force term.
- The linear system is hard to solve due to $K$'s sheer size: about 8 million rows and columns for a $24^3 \times 48$ lattice.
- Mass matrices of interest have high-frequency modes, which mean small step-sizes $\delta t$ are required for numerical stability.



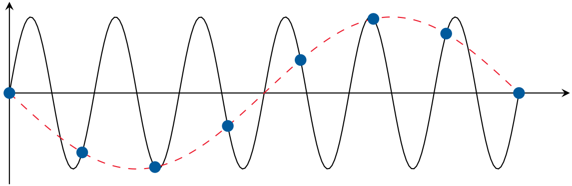
# Implementation

- Main computational expense is in calculating $K^{-1}\phi$, i.e. solving $K\chi = \phi$ for $\chi$, for the force term.
- The linear system is hard to solve due to $K$'s sheer size: about 8 million rows and columns for a $24^3 \times 48$ lattice.
- Mass matrices of interest have high-frequency modes, which mean small step-sizes $\delta t$ are required for numerical stability.



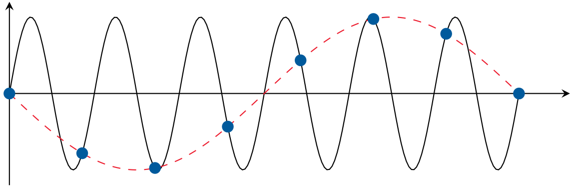
# Implementation

- Main computational expense is in calculating $K^{-1}\phi$, i.e. solving $K\chi = \phi$ for $\chi$, for the force term.
- The linear system is hard to solve due to $K$'s sheer size: about 8 million rows and columns for a $24^3 \times 48$ lattice.
- Mass matrices of interest have high-frequency modes, which mean small step-sizes $\delta t$ are required for numerical stability.



# Implementation

- Main computational expense is in calculating $K^{-1}\phi$, i.e. solving $K\chi = \phi$ for $\chi$, for the force term.
- The linear system is hard to solve due to $K$'s sheer size: about 8 million rows and columns for a $24^3 \times 48$ lattice.
- Mass matrices of interest have high-frequency modes, which mean small step-sizes $\delta t$ are required for numerical stability.



- Hence, plain HMC can take a long time.

# Improving HMC

By factorizing the determinant via

$$\det K = \det L \, \frac{\det K}{\det L},$$

we can use an alternative action with more terms, namely

$$S' = \phi_1^\dagger L^{-1} \phi_1 + \phi_2^\dagger L K^{-1} \phi_2$$

## Idea

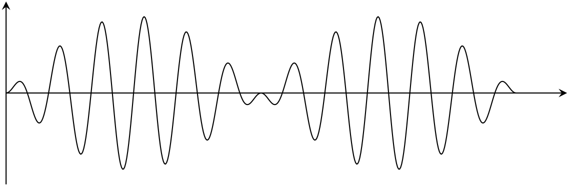
Choose $L$ such that

- $L^{-1}$ captures the UV (high energy) modes of the system
- $L^{-1}\phi$ is cheap to evaluate

This ensures we can place the two terms on different time-scales: the cheap UV term $S_1$ on a finer scale than the expensive IR term $S_2$.

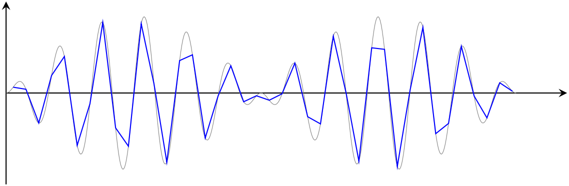
# Filtering

$K^{-1}$

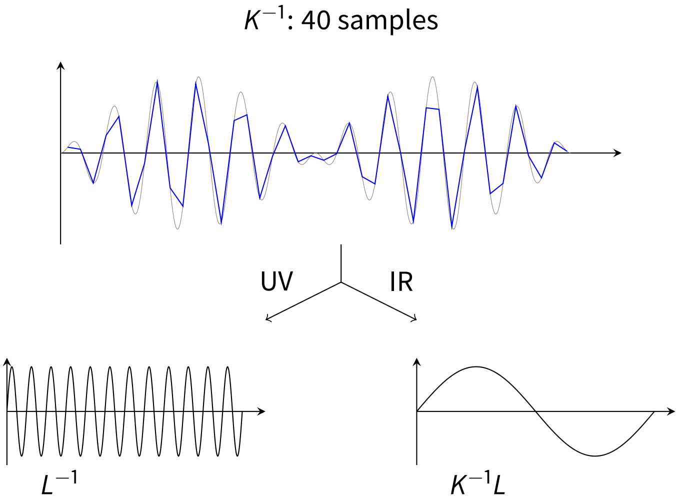


# Filtering

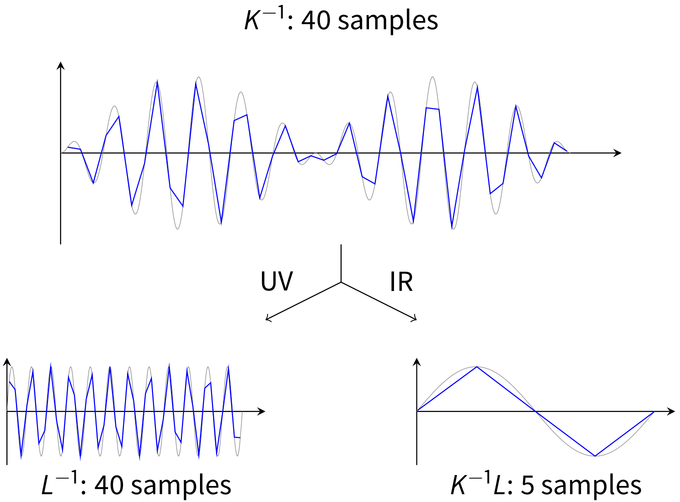
$K^{-1}$: 40 samples



# Filtering



# Filtering



# Polynomial-filtered HMC

## Definition

Polynomial-filtered HMC (PFHMC) chooses $L = 1/P(K)$ where $P$ is a polynomial approximating the inverse $P \approx 1/K$. The action then becomes

$$S = \phi_1^\dagger P(K)\phi_1 + \phi_2^\dagger [P(K)K]^{-1}\phi_2$$

*Kamleh and Peardon, Comp. Phys. Comm. 183, 2011, arXiv:1106.5625*

- By increasing the polynomial order $p$, the polynomial captures more of the action whilst $P(K)K$ gets closer to the identity $I$.

# Polynomial-filtered HMC

- This method can be extended to use several filters; for example, the 2-filter fermion action is

$$S_{2-\text{poly}} = \phi_1^\dagger P_1(K)\phi_1 + \phi_2^\dagger Q(K)\phi_2 + \phi_3^\dagger [P_2(K)K]^{-1}\phi_3.$$

- Here, we have two polynomials $P_1$ and $P_2$ with orders $p_1 < p_2$ and both approximating $1/K$.
- The polynomials are chosen such that $Q \equiv P_2/P_1$ is also a polynomial with order $q = p_2 - p_1$.
- $S_{2-\text{poly}}$ further separates the frequency modes of the fermion matrix, and the terms can be placed on separate scales $n_3 < n_2 < n_1$.

# PFHMC – Implementation

- Modified an existing Lattice QCD program, BQCD, to accommodate polynomial filtering. This Fortran code is used by the QCDSF collaboration.

  *Nakamura and Stüben, PoS Lattice 2010, arXiv:1011.0199*

- Simulations were performed on a $24^3 \times 48$ lattice with a pion mass of $m_\pi \sim 400$ MeV ($\kappa = 0.1362$), along with other lattices that will not be presented here.

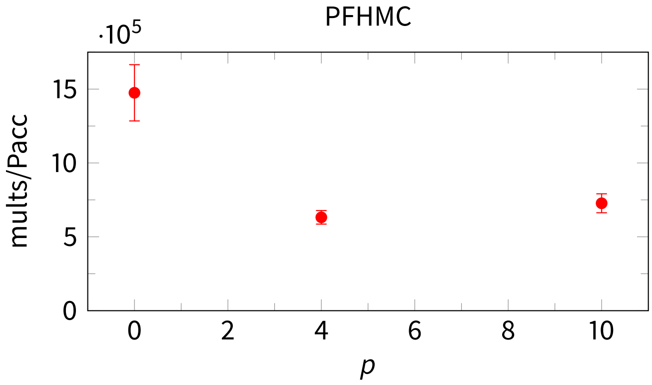
  Simulation parameters

- This is relatively light pion mass, so the tests will give a good idea of how the algorithms will perform close to physical masses.

# A Word of Caution

- Attempts have been made to optimize each possible fermion action by varying their parameters.
- However, the parameter space that can be explored is vast:
  - Polynomial orders $p_1, p_2$
  - Choice of polynomials (Chebyshev, etc.)
  - Number of steps $n_i$ to use for each action term $S_i$, e.g. $n_1 = 560, n_2 = 280, n_3 = 140$
  - Choice of integrator (e.g. leapfrog, $2^{nd}$ order minimal-norm)
- This space hasn't been fully explored, so these results are preliminary.

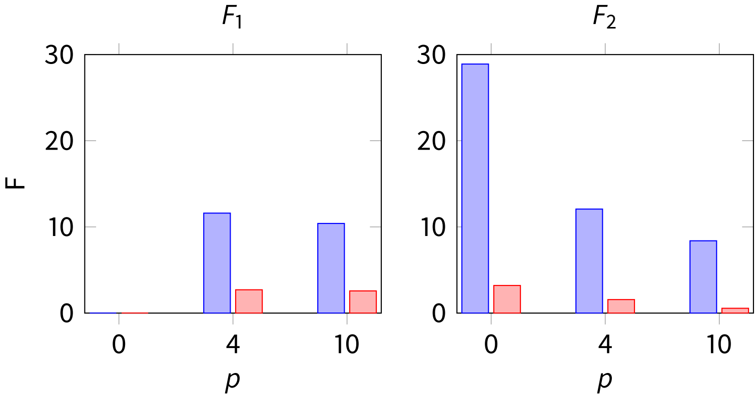
# PFHMC – Cost




- Graph shows the number of matrix ($M^\dagger M$) multiplies per trajectory weighted by the inverse acceptance rate.
- This weighting takes into account the cost of rejected trajectories.

▸ Data table

# PFHMC – Forces




-  shows the maximal forces $\max(F_i)$ and  shows the average forces.
- The stronger the force, the more UV terms the action term is incorporating $\Longrightarrow$ require a finer step-size.

# Outline

# PFHMC versus Hasenbusch

## Definition

Hasenbusch or mass preconditioning filters the fermion action $S_F$ with another mass matrix $K'$, identical to $K$ except that it has hopping parameter $\rho < \kappa$:

$$S_{\text{hasen}} = \phi_1^\dagger K'^{-1} \phi_1 + \phi_2^\dagger K' K^{-1} \phi_2$$

- Hasenbusch preconditioning is rendered more effective via the use of two filters, i.e. using the action

$$S_{2-\text{hasen}} = \phi_1^\dagger K_1^{-1} \phi_1 + \phi_2^\dagger K_1 K_2^{-1} \phi_2 + \phi_3^\dagger K_2 K^{-1} \phi_3$$

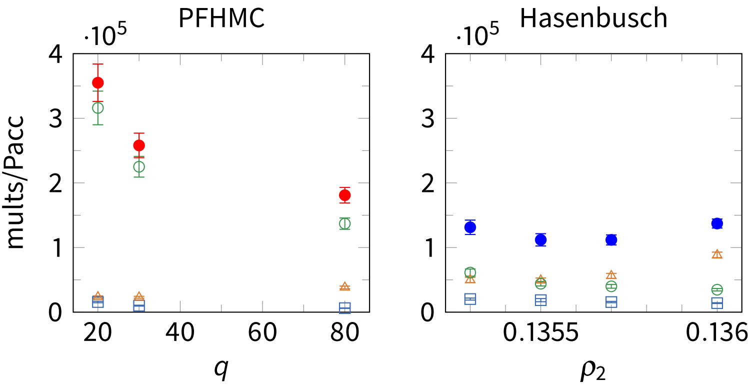
where $\rho_1 < \rho_2 < \kappa$.

# PFHMC versus Hasenbusch – Implementation

- In order to determine the effectiveness of PFHMC, we compared it to the dominant filtering method, Hasenbusch preconditioning.
- Used 2 filters for each method, varying $q = p_2 - p_1$ and $\rho_2$ to find an optimum fit whilst keeping $p_1 = 4$ and $\rho_1 = 0.130$ fixed.
- Recall:

$$S_{2-\text{poly}} = \phi_1^\dagger P_1(K)\phi_1 + \phi_2^\dagger Q(K)\phi_2 + \phi_3^\dagger [P_2(K)K]^{-1}\phi_3$$
$$S_{2-\text{hasen}} = \phi_1^\dagger K_1^{-1}\phi_1 + \phi_2^\dagger K_1 K_2^{-1}\phi_2 + \phi_3^\dagger K_2 K^{-1}\phi_3$$

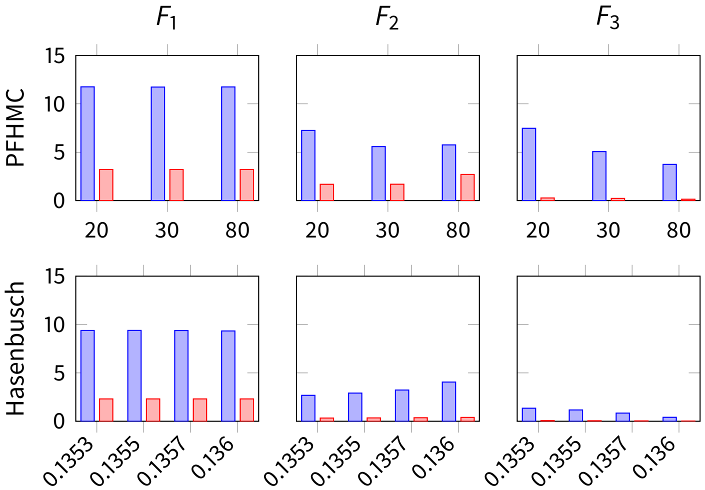
# PFHMC versus Hasenbusch – Cost



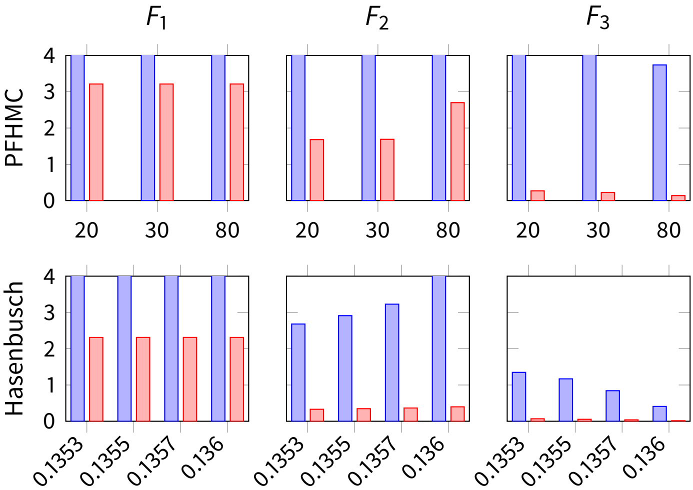

- Number of matrix ($M^\dagger M$) multiplies per trajectory, weighted by the inverse acceptance rate (cf. plain HMC $\approx 15 \times 10^5$).
- Contributions from different terms: $S_1$ —□—, $S_2$ —△—, and $S_3$ —○—.

Data table

# PFHMC versus Hasenbusch – Forces



# PFHMC versus Hasenbusch – Forces

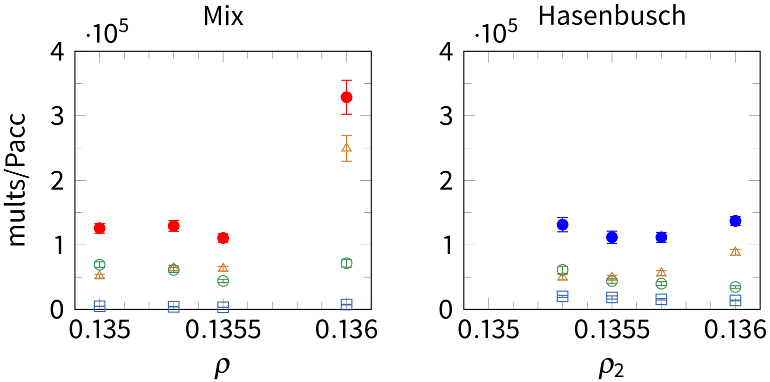


- One can also combine the two methods, using a polynomial filter to capture the high energy modes, and a Hasenbusch intermediate filter:

$$S_{\text{mix}} = \phi_1^\dagger P(K')\phi_1 + \phi_2^\dagger [P(K')K']^{-1}\phi_2 + \phi_3^\dagger K' K^{-1}\phi_3$$

- The motivation is that the polynomial $P(K')$ is much easier to calculate than the inverse of an equivalent heavier mass matrix $K''$, and so may be better suited to capturing the UV modes.

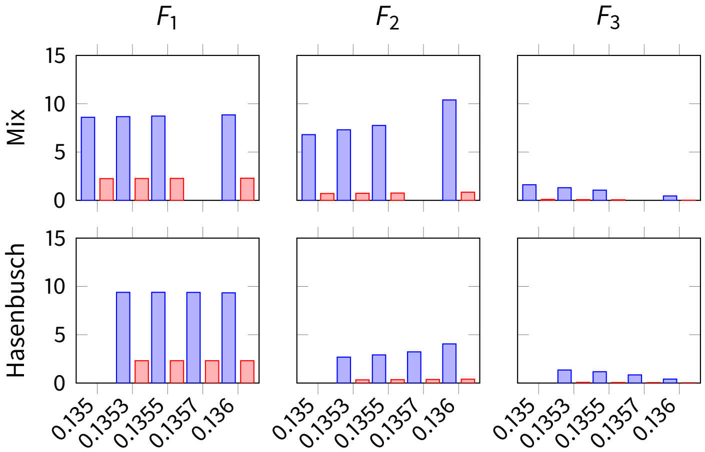
# PFHMC with Hasenbusch – Cost



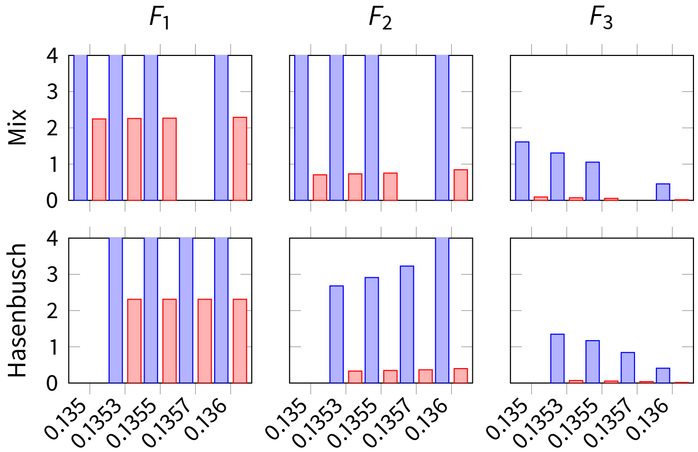

- Left hand plot shows the results for mixed PFHMC/Hasenbusch with polynomial order $p = 4$, whilst the right hand plot is from before.
- $S_1$ —□—, $S_2$ —△—, $S_3$ —○—

Data table

# PFHMC with Hasenbusch – Forces



# PFHMC with Hasenbusch – Forces



# Single Flavours

- Have been using 2 degenerate flavours of quark up to now, with $K = M^\dagger M$, i.e. $S_F = \phi^\dagger M^\dagger M \phi$.
- However, to include Lattice QED effects we must use singleton quarks, as the quark charges discriminate the up and down quarks.
- To simulate just a single flavour, can we just use

$$S_F = \phi^\dagger M \phi?$$

# Single Flavours

- Have been using 2 degenerate flavours of quark up to now, with $K = M^\dagger M$, i.e. $S_F = \phi^\dagger M^\dagger M\phi$.
- However, to include Lattice QED effects we must use singleton quarks, as the quark charges discriminate the up and down quarks.
- To simulate just a single flavour, can we just use

$$S_F = \phi^\dagger M\phi?$$

## Problem

- $e^{-S}$ needs to be interpreted as a probability distribution, so $M$ must be positive-semidefinite (i.e. $S_F \geq 0$).
- The mass matrix $M$ is not positive-semidefinite in general.

# Single Flavours

- Have been using 2 degenerate flavours of quark up to now, with $K = M^\dagger M$, i.e. $S_F = \phi^\dagger M^\dagger M \phi$.
- However, to include Lattice QED effects we must use singleton quarks, as the quark charges discriminate the up and down quarks.
- To simulate just a single flavour, can we just use

$$S_F = \phi^\dagger M \phi?$$

## Problem

- $e^{-S}$ needs to be interpreted as a probability distribution, so $M$ must be positive-semidefinite (i.e. $S_F \geq 0$).
- The mass matrix $M$ is not positive-semidefinite in general.

## Solution

*Replace M by a positive-semidefinite approximation, e.g. Rational HMC*

# Rational HMC

## Definition

Rational HMC (RHMC) approximates the inverse of the fermion matrix $K$ by a rational function $R(Q)$. One then uses the action

$$S_{\text{RHMC}} = \phi^{\dagger} R(Q) \phi$$

- In the singleton quark case, we choose $Q = M^{\dagger} M$ and set $R(Q)$ to approximate $\frac{1}{\sqrt{Q}}$, which is positive-semidefinite. This works as

$$\det Q^{\frac{1}{2}} = \sqrt{\det Q} = \sqrt{\det M \det M} = |\det M|$$

# Polynomial Filtered RHMC

- Polynomial filtering can be applied to RHMC.
- In the case of a single quark flavour, we choose a polynomial $P(Q) \approx 1/\sqrt{Q}$ then use the action

$$S_{\mathrm{PF-RHMC}} = \phi_1^\dagger P(Q)\phi_1 + \phi_2^\dagger R(Q)P^{-1}(Q)\phi_2 .$$

- As before, we can place the UV/first term on a finer time-scale than the IR/second term.
- Tests ongoing for $1+1$ fermion flavours, such that existing $24^3 \times 48$ configurations can be used.

# Summary

- PFHMC is better than plain HMC.
- Hasenbusch + PFHMC works about as well as Hasenbusch + Hasenbusch.
- Tests for PF-RHMC are ongoing.

## Future Work

- Tune relative step-sizes for the mixed case
- 2-filter PF-RHMC
- QED

# Outline

4 Tables

# Tables

| size | $\beta$ | $\kappa$ | $m_\pi$ (MeV) | integrator |
|---|---|---|---|---|
| $24^3 \times 48$ | 5.29 | 0.1362 | 400 | 2MNSTS |

Table: General simulation parameters

- Wilson gauge action
- Clover fermion action, $c_{sw} = 1.9192$
- All runs have $\geq 100$ trajectories

Return to 'PFHMC - Implementation'

# Tables

| p | steps | $P_{acc}$ | matrix ops |
|---|---|---|---|
| 0 | $300 \times 2$ | 0.60(7) | $885{,}000 \pm 11{,}000$ |
| 4 | $160 \times 2 \times 2$ | 0.86(5) | $529{,}800 \pm 7900$ |
| 10 | $110 \times 2 \times 2$ | 0.65(5) | $446{,}700 \pm 5300$ |

Table: 1-filter PFHMC parameters

Return to graph

# Tables

| $p$ | $q$ | steps | $P_{acc}$ | matrix ops |
|---|---|---|---|---|
| 4 | 20 | $80 \times 2 \times 2 \times 2$ | 0.66(5) | $233{,}200 \pm 2700$ |
| 4 | 30 | $55 \times 2 \times 2 \times 2$ | 0.70(5) | $180{,}900 \pm 1600$ |
| 4 | 80 | $35 \times 2 \times 2 \times 2$ | 0.73(4) | $132{,}500 \pm 1300$ |

Table: 2-filter PFHMC parameters

| $\rho_1$ | $\rho_2$ | steps | $P_{acc}$ | matrix ops |
|---|---|---|---|---|
| 0.130 | 0.1353 | $10 \times 2 \times 2 \times 2$ | 0.62(5) | $82{,}160 \pm 580$ |
| 0.130 | 0.1355 | $8 \times 2 \times 2 \times 2$ | 0.67(5) | $75{,}800 \pm 1200$ |
| 0.130 | 0.1357 | $7 \times 2 \times 2 \times 2$ | 0.71(5) | $80{,}330 \pm 450$ |
| 0.130 | 0.136 | $8 \times 2 \times 2 \times 2$ | 0.84(4) | $116{,}270 \pm 780$ |

Table: 2-filter Hasenbusch parameters

Return to graph

# Tables

| $\rho$ | p | steps | $P_{acc}$ | matrix ops |
|---|---|---|---|---|
| 0.135 | 4 | $14 \times 2 \times 2 \times 2$ | 0.74(4) | $89{,}050 \pm 620$ |
| 0.1353 | 4 | $11 \times 2 \times 2 \times 2$ | 0.67(5) | $85{,}100 \pm 540$ |
| 0.1355 | 4 | $10 \times 2 \times 2 \times 2$ | 0.77(4) | $82{,}800 \pm 500$ |
| 0.136 | 4 | $20 \times 2 \times 2 \times 2$ | 0.68(5) | $218{,}500 \pm 1500$ |

Table: Mixed PFHMC/Hasenbusch parameters

Return to graph