Single flavour pseudofermions on the lattice

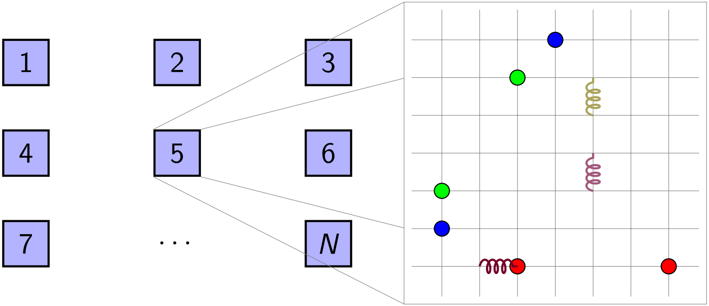
Taylor Haar

Collaborators: Waseem Kamleh, Yoshifumi Nakamura, James Zanotti

University of Adelaide

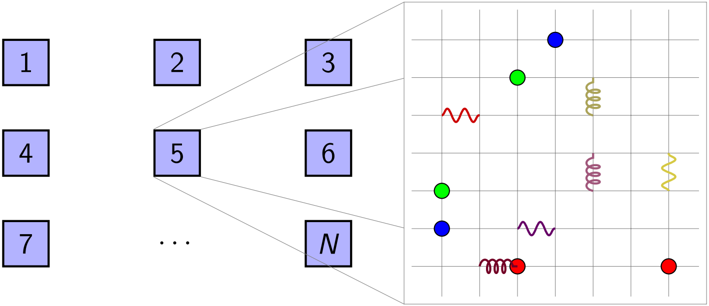
QCD Down Under
13th of July, 2017

Lattice QCD



- In Lattice QCD, we generate configurations with quarks fields $\psi^{(f)}$ on the lattice sites and gluon fields U_μ on the lattice links.
- Many simulations combine the up and the down quark into a single entity $n_f = 2$ for convenience.

Lattice QCD+QED



- Recent lattice simulations add in QED effects, which is done by introducing a photon field A_μ^{QED} .
- But now the up and down quark are distinguished by their charge, so we can no longer use the $n_f = 2$ trick.
- We must use single flavour pseudofermions $n_f = 1$.

Outline

1. Theory: what are single flavour pseudofermions?
2. Results: how fast can we generate them?
   - Filtering methods
   - Multiple filters
   - Characteristic scale tuning

Expectation values

- Lattice gauge theory centres around the evaluation of expectation values of operators O:

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

where the lattice action S is defined as

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

with $M^{(f)}$ being the Dirac operator for fermion flavour f.

Expectation values

- We evaluate expectation values on the lattice by generating configurations $(U_i, \psi_i^{(f)}, \overline{\psi}_i^{(f)})$ distributed according to the Boltzmann distribution $\frac{1}{Z}\exp(-S[U, \psi^{(f)}, \overline{\psi}^{(f)}])$, then taking the average

$$\langle O \rangle \approx \frac{1}{N} \sum_{i=1}^{N} O[U_i, \psi_i^{(f)}, \overline{\psi}_i^{(f)}].$$

- The effort lies in generating appropriate configurations, for which we use Hybrid Monte Carlo (HMC).

Double flavour pseudo-fermions

- HMC requires the fermions $\psi, \overline{\psi}$ to be in a non-Grassmannian form.
- Consider the fermion part of the partition function Z for an action with degenerate up and down quarks

$$\begin{aligned}\int D\psi D\overline{\psi} \exp\left[-\overline{\psi}_u M_u \psi_u - \overline{\psi}_d M_d \psi_d\right] \\ &= \det M_u \det M_d \\ &= \det M^\dagger M \\ &= \int D\phi D\phi^* \exp\left[-\phi^\dagger (M^\dagger M)^{-1} \phi\right]\end{aligned}$$

- This is now a regular integral with pseudofermions ϕ, and hence we can generate appropriate configurations via HMC with the lattice fermion action $S_F = \phi^\dagger (M^\dagger M)^{-1} \phi$.

Skip HMC

Hybrid Monte Carlo

Goal

Generate configurations (U_i, ϕ_i) distributed according to $\frac{1}{Z} e^{-S[U,\phi]}$.

- Generate ϕ from the Gaussian distribution $\exp(-S_F[U, \phi])$
- Introduce a "momentum" field P conjugate to the gauge field U, and construct the Hamiltonian

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

Hybrid Monte Carlo

Goal

Generate configurations (U_i, ϕ_i) distributed according to $\frac{1}{Z}e^{-S[U,\phi]}$.

- Integrate a molecular dynamics trajectory based on Hamilton's equations, i.e. using steps

$$(P, U) \to (P, e^{ihP}U) \quad \text{and} \quad (P, U) \to (P - hF, U),$$

  where $F = \frac{\partial S}{\partial U}$ is the force term. This produces the candidate state (P', U').
- Accept this state with probability $\min(1, \exp[H - H'])$.

Single-flavour pseudo-fermions?

- Consider an action with just the strange quark. The fermion part of the partition function can be written as

$$\det M_s = \int D\phi D\phi^* \exp\left[-\phi^\dagger M_s^{-1} \phi\right]$$

Single-flavour pseudo-fermions?

- Consider an action with just the strange quark. The fermion part of the partition function can be written as

$$\det M_s = \int D\phi D\phi^* \exp\left[-\phi^\dagger M_s^{-1}\phi\right]$$

Problem

The pseudo-fermion integral is not guaranteed to converge, as M_s may have negative eigenvalues.

- This is not a problem with the double-flavour pseudo-fermion because $M^\dagger M$ is positive semi-definite by construction.

Rational Hybrid Monte Carlo

Idea

Replace the matrix M_s^{-1} with a rational approximation $R(K) \approx K^{-1/2}$ where $K = M_s^\dagger M_s$.

- A rational function of $M_s^\dagger M_s$ is positive semi-definite by construction, and so we can safely use HMC with fermion action

$$S_F = \phi^\dagger R(K)\phi.$$

- This technique of replacing the fermion action with a rational approximation is known as Rational Hybrid Monte Carlo (RHMC).

Rational Hybrid Monte Carlo

Idea

Replace the matrix M_s^{-1} with a rational approximation $R(K) \approx K^{-1/2}$ where $K = M_s^\dagger M_s$.

- A rational function of $M_s^\dagger M_s$ is positive semi-definite by construction, and so we can safely use HMC with fermion action

$$S_F = \phi^\dagger R(K)\phi.$$

- This technique of replacing the fermion action with a rational approximation is known as Rational Hybrid Monte Carlo (RHMC).

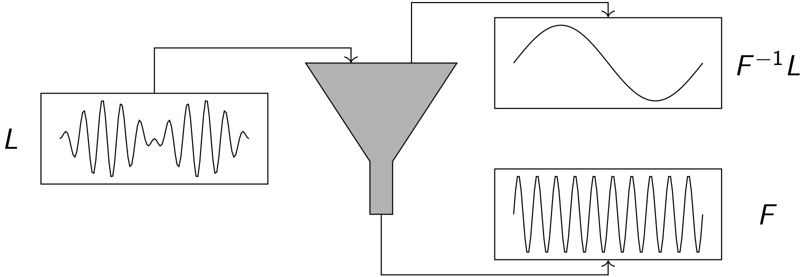
Can we do better?

Filtering methods

- One way to reduce the cost of HMC is to split the action into multiple terms via a filter F on the fermion action kernel L

$$\phi^\dagger L\phi \to \phi_1^\dagger F\phi_1 + \phi_2^\dagger F^{-1} L\phi_2,$$

then integrate each term with a different step-size h_i.



Single flavour filters: Polynomial filtered RHMC

- Consider filtering the RHMC action

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

- One choice of filter is a low order polynomial $P(K)$ that approximates $K^{-1/2}$, giving fermion action:

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

- By choosing a particular class of polynomials $P(K)$, we only need to tune the polynomial order p.

Single flavour filters: tRHMC

- We can write the rational approximation $R(K)$ as

$$R(K) = d_n \prod_{k=1}^{n} \frac{(K + a_k)}{(K + b_k)}$$

where $a_k, b_k, d_n > 0$ and $a_{k+1} \leq a_k, b_{k+1} \leq b_k$.

- Assuming that the truncated ordered product given by

$$R_{i,j}(K) = (d_n)^{\delta_{i1}} \prod_{k=i}^{j} \frac{(K + a_k)}{(K + b_k)}$$

approximates $K^{-1/2}$ whenever $i = 1$, we can use $R_{1,t}(K)$ as a filter:

$$S_{tRHMC} = \phi_1^\dagger R_{1,t}(K)\phi_1 + \phi_2^\dagger R_{t+1,n}(K)\phi_2.$$

Lattice setup

- All runs were performed using BQCD, which has these filtering methods built into the latest release: www.rrz.uni-hamburg.de/bqcd
- The test lattice used in this work was a $16^3 \times 32$ lattice with Wilson gauge $\beta = 5.6$ and two degenerate single-flavour Wilson fermions at $\kappa = 0.15825$.
- This corresponds to lattice spacing $a \sim 0.08$ fm and pion mass $m_\pi \sim 400$ MeV.

Goal

Minimize the cost to generate independent configurations, which we parametrize as

$$\text{Cost} \propto \frac{N_{mat}}{P_{acc}}$$

Tuning the step-sizes

- Our actions have several step-sizes to tune; e.g. the PF-RHMC action

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

has step-sizes $\{h_G, h_1, h_2\}$.

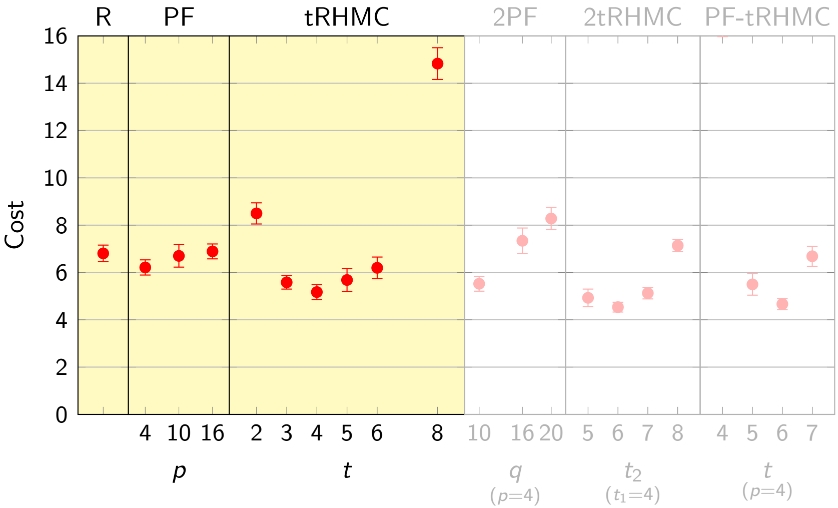
- We tune these by force balancing, where given the forces $F_i = \frac{\partial S_i}{\partial U}$ for each action term, the step-sizes follow

$$F_i h_i \approx \text{constant}$$

- We then tune the only free step-size, the coarsest scale h_n, such that the acceptance rate $P_{acc} \sim 0.75$, which can be performed by fitting P_{acc} to

$$P_{acc} \approx \text{erfc}(h_n^2 c^2).$$

Filtering: Results



Combining filters

We can improve the performance by using two filters.

2PF-RHMC

$$S_{2PF-RHMC} = \phi_1^\dagger P(K)\phi_1 + \phi_2^\dagger Q(K)\phi_2 + \phi_3^\dagger R(K)[P(K)Q(K)]^{-1}\phi_3$$

where $Q(K) \approx P(K)^{-1}K^{-1/2}$ is a polynomial $q > p$

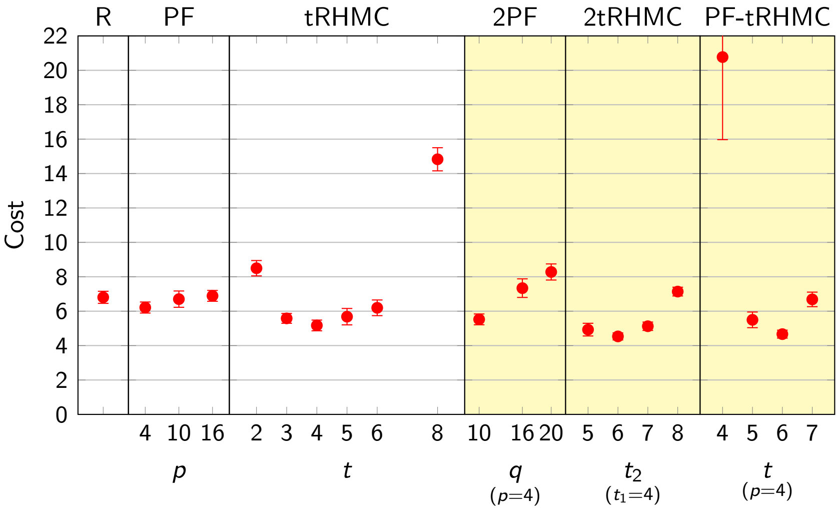
2tRHMC

$$S_{2tRHMC} = \phi_1^\dagger R_{1,t_1}(K)\phi_1 + \phi_2^\dagger R_{t_1+1,t_2}(K)\phi_2 + \phi_3^\dagger R_{t_2+1,n}(K)\phi_3$$

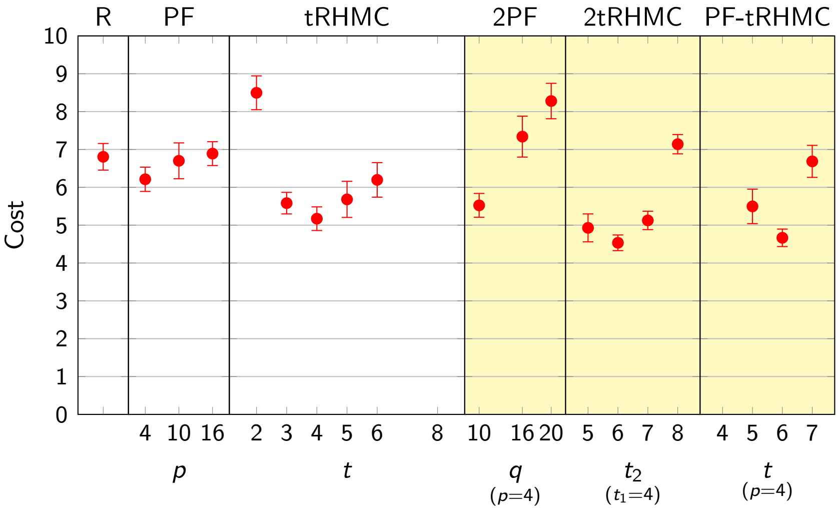
PF-tRHMC

$$S_{PF-tRHMC} = \phi_1^\dagger P(K)\phi_1 + \phi_2^\dagger R_{1,t}(K)P(K)^{-1}\phi_2 + \phi_3^\dagger R_{t+1,n}(K)\phi_3$$

Combining filters: Results



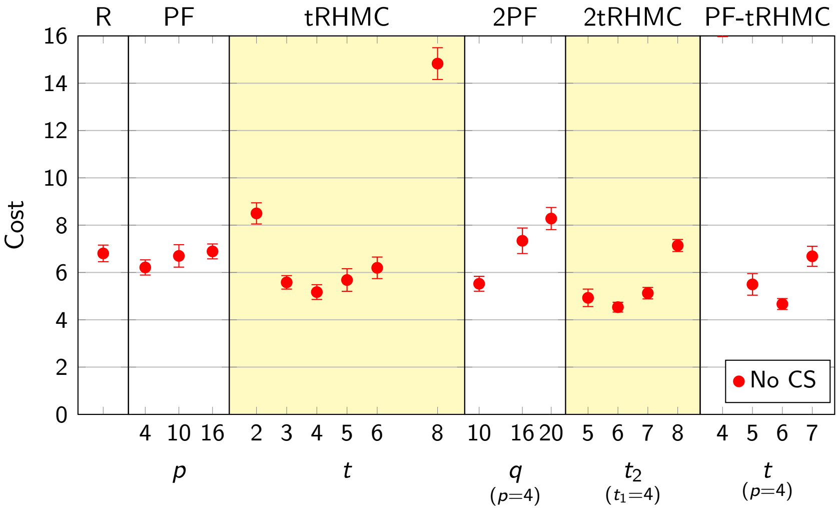
Combining filters: Results



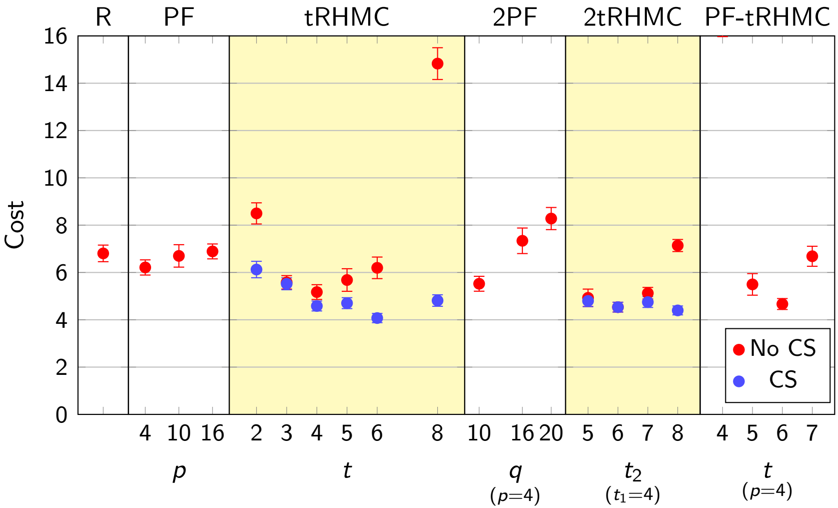
Characteristic scale tuning

- Looking at some of the previous results (e.g. the $t = 2, 8$ points for 1tRHMC), it seems that force balancing does not perform well for all choices of filter.
- The idea of characteristic scale tuning is to tune the coarsest scale via fitting to $\mathrm{erfc}(h_n^2 c^2)$ as before, then tune the second-coarsest scale manually while keeping the remaining step-sizes in sync according to $F_i h_i \approx$ constant.
- This can help offset an unfavourable distribution of forces between action terms.

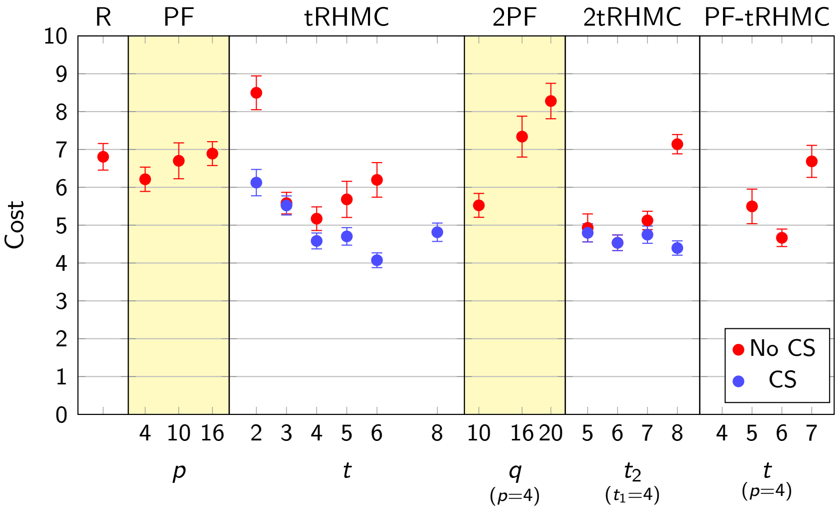
CS-tuning: tRHMC



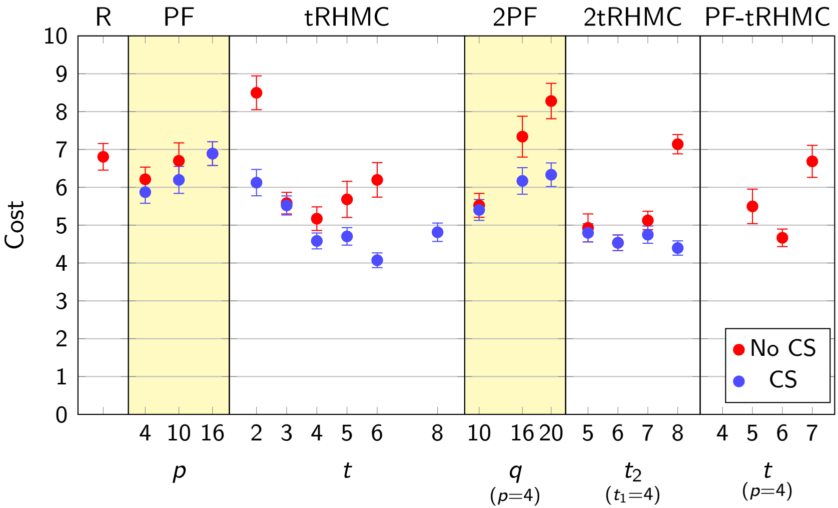
CS-tuning: tRHMC



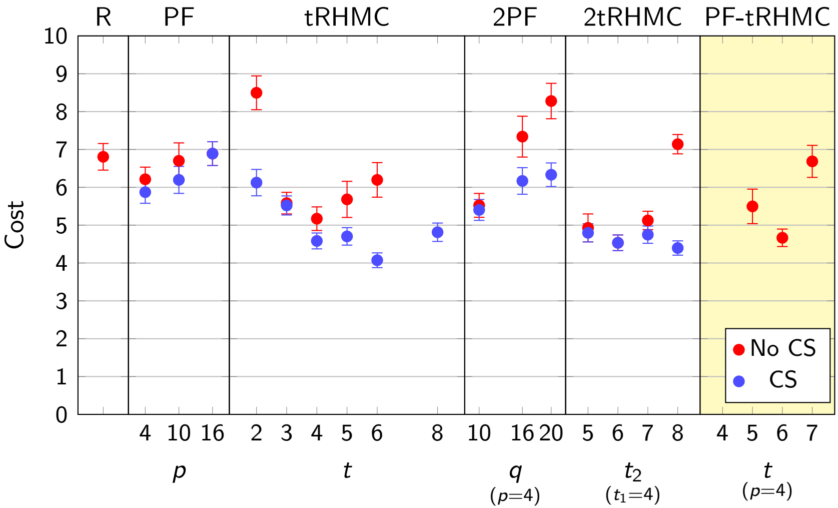
CS-tuning: PF-RHMC



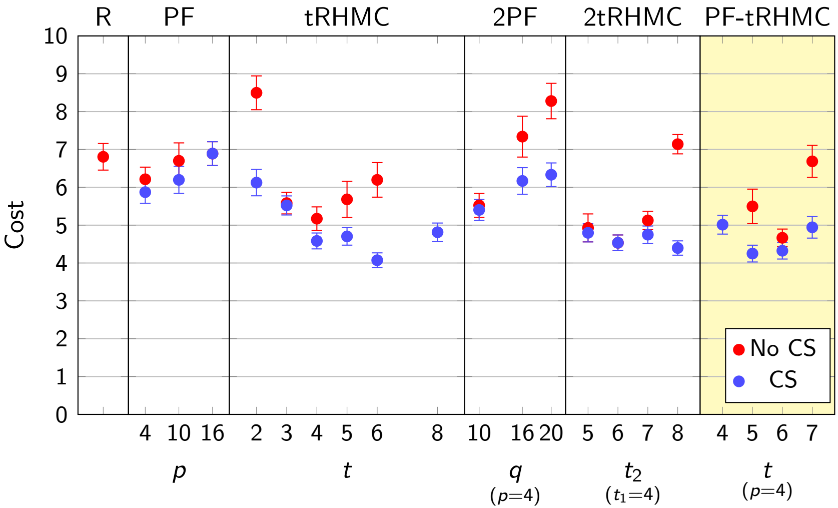
CS-tuning: PF-RHMC



CS-tuning: PF-tRHMC



CS-tuning: PF-tRHMC

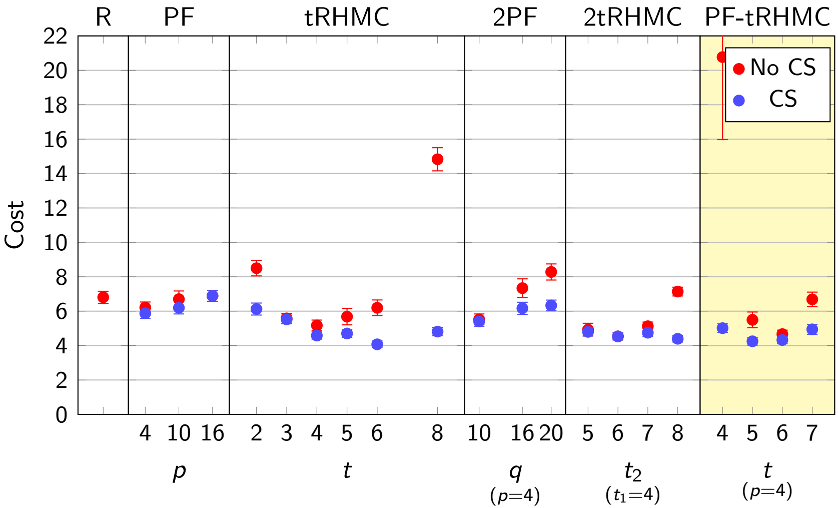


Summary

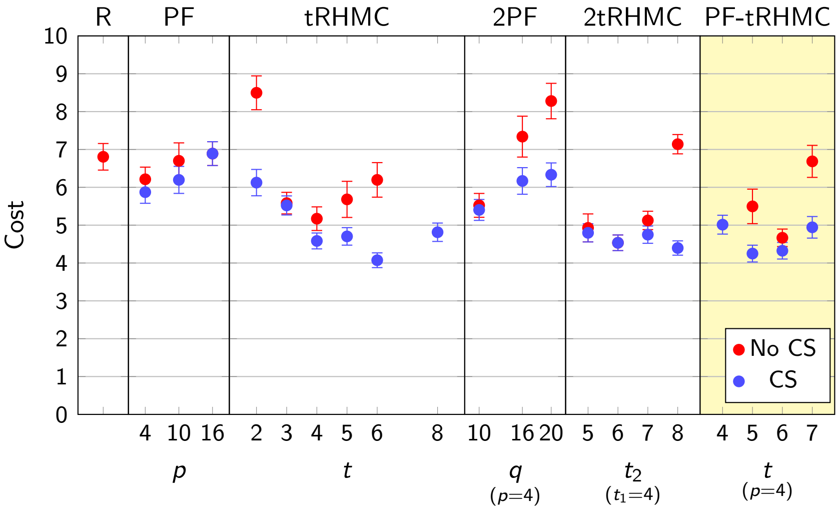
- We tried two different filtering methods to improve the cost of generating single-flavour pseudofermions: polynomial filtering and tRHMC.
- tRHMC performs the best, with about a $\times 1.5$ speedup.
- Characteristic scale tuning makes tRHMC work well across a wide range of truncation orders.
- tRHMC is simple to implement into existing RHMC codes.
- Multiple truncation filters can be used, should prove useful at light quark masses.

Backup slides

Full cost plot



Full cost plot



Cost data: PF-RHMC

p	Steps	N_{mat}	P_{acc}	Cost
0	12-480	47,000(400)	0.69(3)	6.8(4) $\times 10^4$
4	10-18-480	42,200(300)	0.68(3)	62,000(3000)
10	10-26-480	47,200(500)	0.70(4)	67,000(5000)
16	11-35-480	51,700(300)	0.75(3)	69,000(3000)

Table: PF-RHMC config data (force tuned)

Cost data: PF-RHMC

p	Steps	N_{mat}	P_{acc}	Cost
0	12-480	47,000(400)	0.69(3)	6.8(4) $\times 10^4$
4	10-30-480	41,600(300)	0.71(3)	58,700(3000)
10	10-30-480	44,900(300)	0.73(4)	62,000(4000)
16	11-35-480	51,700(300)	0.75(3)	69,000(3000)

Table: PF-RHMC config data (c-scale tuned)

Cost data: tRHMC

t	Steps	N_{mat}	P_{acc}	Cost
2	15-5-480	57,400(500)	0.68(3)	85,000(5000)
3	10-10-480	40,000(400)	0.72(3)	56,000(3000)
4	8-20-480	34,700(200)	0.67(4)	52,000(3000)
5	6-35-480	34,200(300)	0.60(4)	57,000(5000)
6	5-52-480	40,800(200)	0.66(4)	62,000(5000)
8	4-110-480	124,400(300)	0.84(4)	148,000(7000)

Table: tRHMC config data (force tuned)

Cost data: tRHMC

t	Steps	N_{mat}	P_{acc}	Cost
2	11-25-480	43,500(600)	0.71(3)	61,000(3000)
3	10-25-480	41,900(300)	0.76(3)	55,000(3000)
4	8-22-480	34,800(400)	0.76(3)	46,000(2000)
5	6-30-480	32,800(200)	0.70(3)	47,000(2000)
6	5-15-480	29,000(200)	0.71(3)	41,000(2000)
8	4-10-480	32,500(200)	0.68(3)	48,000(2000)

Table: tRHMC config data (c-scale tuned)

Cost data: 2PF-RHMC

p	q	Steps	N_{mat}	P_{acc}	Cost
4	10	7-33-62-480	36,600(300)	0.66(3)	55,000(3000)
	16	8-51-98-480	49,500(300)	0.67(4)	73,000(5000)
	20	9-65-125-480	58,600(300)	0.71(4)	83,000(5000)

Table: 2PF-RHMC config data (force tuned)

Cost data: 2PF-RHMC

p	q	Steps	N_{mat}	P_{acc}	Cost
4	10	7-20-36-480	36,200(200)	0.67(3)	54,000(3000)
	16	8-15-27-480	40,000(200)	0.65(3)	61,700(3000)
	20	9-20-38-480	45,400(300)	0.72(3)	63,300(3000)

Table: 2PF-RHMC config data (c-scale tuned)

Cost data: 2tRHMC

t_1	t_2	Steps	N_{mat}	P_{acc}	Cost
4	5	6-6-29-480	32,960(240)	0.67(4)	49,000(4000)
	6	5-12-40-480	32,730(140)	0.72(3)	45,300(2100)
	7	4-17-50-480	36,180(180)	0.71(3)	51,200(2400)
	8	4-30-80-480	60,460(160)	0.85(3)	71,400(2600)

Table: 2tRHMC config data (force tuned)

Cost data: 2tRHMC

t_1	t_2	Steps	N_{mat}	P_{acc}	Cost
4	5	6-7-36-480	32,660(170)	0.68(4)	48,000(2400)
	6	5-12-40-480	32,730(140)	0.72(3)	45,300(2100)
	7	4-12-35-480	32,770(140)	0.69(3)	47,500(2300)
	8	4-10-30-480	35,080(190)	0.80(3)	44,000(1900)

Table: 2tRHMC config data (c-scale tuned)

Cost data: PF-tRHMC

p	t	Steps	N_{mat}	P_{acc}	Cost
4	4	8-4-27-480	31,000(300)	0.15(3)	210,000(50,000)
	5	6-12-36-480	33,140(270)	0.60(4)	55,000(5000)
	6	5-26-59-480	34,800(180)	0.75(3)	46,700(2400)
	7	4-38-79-480	47,460(180)	0.71(4)	67,000(4000)

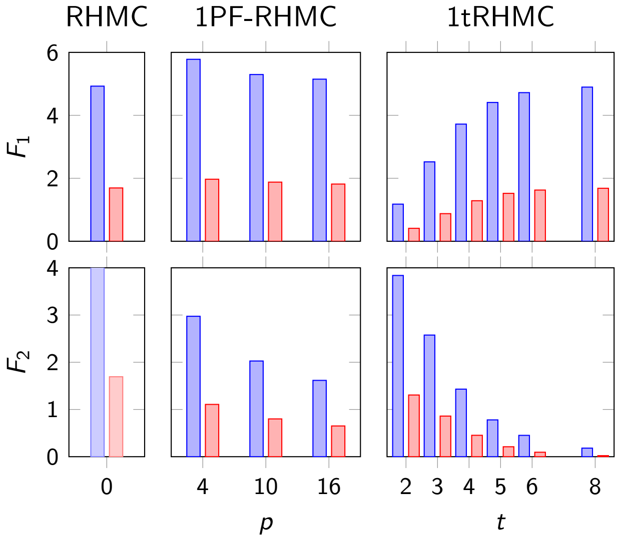
Table: PF-tRHMC config data (force tuned)

Cost data: PF-tRHMC

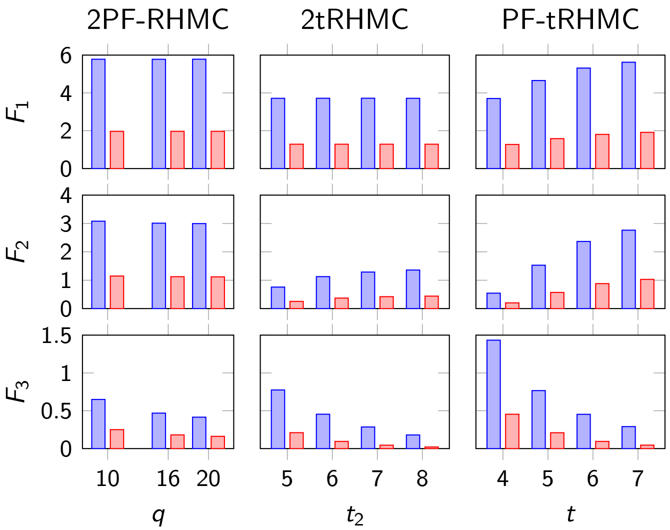
p	t	Steps	N_{mat}	P_{acc}	Cost
4	4	8-8-42-480	35,300(300)	0.74(3)	50,100(2500)
	5	6-10-30-480	27,960(240)	0.68(3)	42,500(2200)
	6	5-12-27-480	28,770(210)	0.69(3)	43,200(2200)
	7	4-16-32-480	32,640(180)	0.69(4)	49,000(3000)

Table: PF-tRHMC config data (c-scale tuned)

Force plots



Force plots


2PF-RHMC
2tRHMC
PF-tRHMC
F_1
F_2
F_3
6
4
2
0
4
3
2
1
0
1.5
1
0.5
0
10
16
20
5
6
7
8
4
5
6
7
q
t_2
t

Force data

	F_1		F_2	
p	avg	max	avg	max
0	–	–	1.69	4.93
4	1.97	5.78	1.11	2.97
10	1.88	5.30	0.80	2.03
16	1.82	5.15	0.65	1.62

Table: PF-RHMC forces

Force data

	F_1		F_2	
t	avg	max	avg	max
2	0.412	1.18	1.307	3.84
3	0.877	2.52	0.860	2.58
4	1.287	3.72	0.453	1.43
5	1.522	4.41	0.211	0.779
6	1.626	4.72	0.0944	0.452
8	1.683	4.90	0.0219	0.182

Table: tRHMC forces

Force data

		F_1		F_2		F_3	
p	q	avg	max	avg	max	avg	max
4	10	1.970	5.78	1.147	3.08	0.2505	0.650
	16	1.970	5.78	1.123	3.01	0.1811	0.469
	20	1.970	5.78	1.118	3.00	0.1617	0.415

Table: 2PF-RHMC forces

Force data

		F_1		F_2		F_3	
t_1	t_2	avg	max	avg	max	avg	max
4	5	1.287	3.72	0.2524	0.758	0.2103	0.775
	6	1.287	3.72	0.3702	1.126	0.0947	0.454
	7	1.287	3.72	0.4193	1.287	0.0447	0.286
	8	1.287	3.72	0.4387	1.358	0.02115	0.180

Table: 2tRHMC forces

Force data

		F_1		F_2		F_3	
p	t	avg	max	avg	max	avg	max
4	4	1.274	3.70	0.2010	0.542	0.4540	1.433
	5	1.582	4.66	0.5680	1.528	0.2094	0.767
	6	1.806	5.31	0.8798	2.366	0.0946	0.453
	7	1.914	5.62	1.028	2.764	0.0455	0.291

Table: PF-tRHMC forces

More lattice parameters

- The rational approximation used was the Zolotarev optimal rational approximation with order $n = 20$ and range $[5 \times 10^{-5}, 3]$.
- The polynomials used were Chebyshev approximations to $K^{-1/2}$ (or $P(K)^{-1} K^{-1/2}$ in the two filter case) with range $[5 \times 10^{-5}, 3]$.