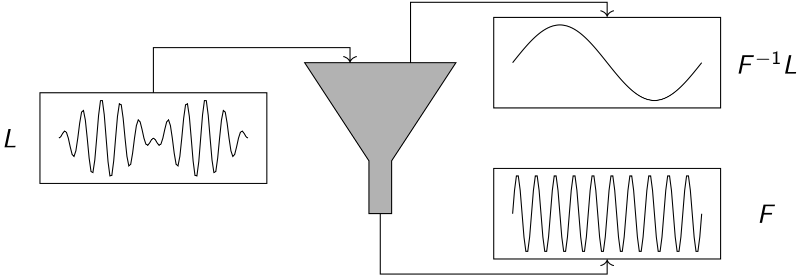
THE UNIVERSITY of ADELAIDE

Single flavour filtering for RHMC in BQCD

Waseem Kamleh

Collaborators

Taylor Haar, Yoshifumi Nakamura, James Zanotti



35th International Symposium on Lattice Field Theory
Granada, Spain, June 18-24, 2017

Lattice QCD+QED

- Lattice QCD precision has advanced to the point that electromagnetic effects are significant
  (e.g. the proton-neutron mass difference)

Lattice QCD+QED

- Lattice QCD precision has advanced to the point that electromagnetic effects are significant (e.g. the proton-neutron mass difference)
- QCD+QED simulations are being performed,

$$\{\psi^{(f)}, U_\mu\} \to \{\psi^{(f)}, U_\mu, A_\mu\}.$$

Lattice QCD+QED

- Lattice QCD precision has advanced to the point that electromagnetic effects are significant
  (e.g. the proton-neutron mass difference)
- QCD+QED simulations are being performed,

$$\{\psi^{(f)}, U_\mu\} \to \{\psi^{(f)}, U_\mu, A_\mu\}.$$

- $q_u \neq q_d, m_d \neq m_s \to$ simulate as $1+1+1$ separate (single) quark flavours.

Lattice QCD+QED

- Lattice QCD precision has advanced to the point that electromagnetic effects are significant (e.g. the proton-neutron mass difference)
- QCD+QED simulations are being performed,

$$\{\psi^{(f)}, U_\mu\} \rightarrow \{\psi^{(f)}, U_\mu, A_\mu\}.$$

- $q_u \neq q_d, m_d \neq m_s \rightarrow$ simulate as $1+1+1$ separate (single) quark flavours.
- Single flavour fermion simulations pose additional computational challenges.

Degenerate two-flavour simulations

- HMC requires the fermions $\psi, \overline{\psi}$ to be in a non-Grassmannian form.

Degenerate two-flavour simulations

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

$$\int D\psi D\overline{\psi} \exp\left[-\overline{\psi}_u M_u \psi_u - \overline{\psi}_d M_d \psi_d\right] = \det M_u \det M_d$$

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

Degenerate two-flavour simulations

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

$$\int D\psi D\overline{\psi} \exp\left[-\overline{\psi}_u M_u \psi_u - \overline{\psi}_d M_d \psi_d\right] = \det M_u \det M_d$$

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

- Effective pseudo-fermion action $S_F = \phi^\dagger (M^\dagger M)^{-1} \phi$.

Single-flavour simulations

- Consider an action with just the strange quark,

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

Single-flavour simulations

- Consider an action with just the strange quark,

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

- **Rational HMC**
  Replace the matrix M_s with a rational approximation

$$R(K) \approx K^{-1/2} \text{ where } K = M_s^\dagger M_s.$$

Single-flavour simulations

- Consider an action with just the strange quark,

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

- **Rational HMC**
  Replace the matrix M_s with a rational approximation

$$R(K) \approx K^{-1/2} \text{ where } K = M_s^\dagger M_s.$$

- Effective pseudo-fermion action $S_F = \phi^\dagger R(K)\phi$.

Single-flavour simulations

- Consider an action with just the strange quark,

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

- **Rational HMC**
  Replace the matrix M_s with a rational approximation

$$R(K) \approx K^{-1/2} \text{ where } K = M_s^\dagger M_s.$$

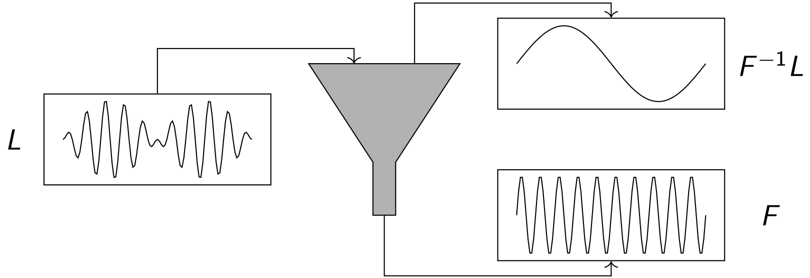
- Effective pseudo-fermion action $S_F = \phi^\dagger R(K)\phi$.
- $M^\dagger M$ is positive semi-definite, correct weight if $\det M > 0$.

Filtering methods

- One way to improve the performance 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.



Single flavour filters: Polynomial filtered RHMC

- Many filtering techniques for two flavour simulations.

Single flavour filters: Polynomial filtered RHMC

- Many filtering techniques for two flavour simulations.
  - Mass preconditioning, polynomial filtering, SAP.

Single flavour filters: Polynomial filtered RHMC

- Many filtering techniques for two flavour simulations.
  - Mass preconditioning, polynomial filtering, SAP.
- Consider filtering the RHMC action

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

Single flavour filters: Polynomial filtered RHMC

- Many filtering techniques for two flavour simulations.
  - Mass preconditioning, polynomial filtering, SAP.
- Consider filtering the RHMC action

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

- Choose low order polynomial $P(K) \simeq K^{-1/2}$, such that

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

Single flavour filters: Polynomial filtered RHMC

- Many filtering techniques for two flavour simulations.
  - Mass preconditioning, polynomial filtering, SAP.
- Consider filtering the RHMC action

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

- Choose low order polynomial $P(K) \simeq K^{-1/2}$, such that

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

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

Single flavour filters: Polynomial filtered RHMC

- Many filtering techniques for two flavour simulations.
  - Mass preconditioning, polynomial filtering, SAP.
- Consider filtering the RHMC action

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

- Choose low order polynomial $P(K) \simeq K^{-1/2}$, such that

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

- Given a class of polynomials $P(K)$, we only need to tune the polynomial order p.
  - e.g. Chebyshev approximations.

Single flavour filters: tRHMC

- The rational polynomial $R(K) \simeq K^{-1/2}$ in ordered product form is
$$R(K) = d_n \prod_{k=1}^{n} \frac{(K + a_k)}{(K + b_k)}$$
where $a_k, b_k, d_n > 0$ and $a_k > a_{k+1}, b_k > b_{k+1}$.

Single flavour filters: tRHMC

- The rational polynomial $R(K) \simeq K^{-1/2}$ in ordered product form is

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

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

- Partition the product into partial factors defined by

$$R_{i,j}(K) = (d_n)^{\delta_{1,i}} \prod_{k=i}^{j} \frac{(K + a_k)}{(K + b_k)}$$

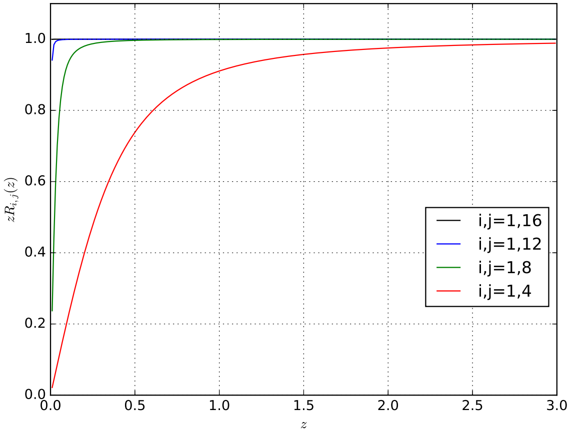
Single flavour filters: tRHMC

- The rational polynomial $R(K) \simeq K^{-1/2}$ in ordered product form is
$$R(K) = d_n \prod_{k=1}^{n} \frac{(K + a_k)}{(K + b_k)}$$
where $a_k, b_k, d_n > 0$ and $a_k > a_{k+1}, b_k > b_{k+1}$.
- Partition the product into partial factors defined by
$$R_{i,j}(K) = (d_n)^{\delta_{1,i}} \prod_{k=i}^{j} \frac{(K + a_k)}{(K + b_k)}$$

  - e.g. Luscher & Schaefer, *Comput.Phys.Commun. 184 (2013)*.

Single flavour filters: tRHMC

- The truncated product $R_{1,t}(K)$ also approximates $K^{-1/2}$.



Single flavour filters: tRHMC

- Truncated ordered product (top) filtering for RHMC.

Single flavour filters: tRHMC

- Truncated ordered product (top) filtering for RHMC.
- Define tRHMC action as

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

Single flavour filters: tRHMC

- Truncated ordered product (top) filtering for RHMC.
- Define tRHMC action as

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

  - Reuse existing HMC code!

Single flavour filters: tRHMC

- Truncated ordered product (top) filtering for RHMC.
- Define tRHMC action as

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

  - Reuse existing HMC code!
  - Distinct from partitioning the sum over poles.

Single flavour filters: tRHMC

- Truncated ordered product (top) filtering for RHMC.
- Define tRHMC action as

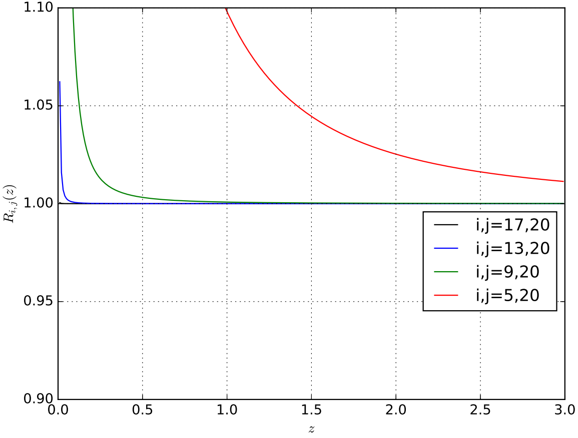
$$S_{tRHMC} = \phi_1^\dagger R_{1,t}(K)\phi_1 + \phi_2^\dagger R_{t+1,n}(K)\phi_2.$$

  - Reuse existing HMC code!
  - Distinct from partitioning the sum over poles.
- After truncating the ordered product, note that for small values of a_k and $b_k, (k > t)$ remainder term behaves like

$$R_{t+1,n}(K) = \prod_{k=t+1}^{n} \frac{(K + a_k)}{(K + b_k)} \simeq 1.$$

Single flavour filters: tRHMC

- The remainder term approximates unity.



BQCD lattice code

- All runs were performed using BQCD [see the poster (Stübern)].
- PF-RHMC and tRHMC filtering methods built into the latest release:
  www.rrz.uni-hamburg.de/services/hpc/bqcd
- Performance tests on $16^3 \times 32$ lattice.
  - Wilson gauge $\beta = 5.6$
  - Lattice spacing $a \sim 0.08$ fm.
  - $1 + 1$ (degenerate) single-flavour Wilson fermions.
  - $\kappa = 0.15825$ yields $m_\pi \sim 400$ MeV.

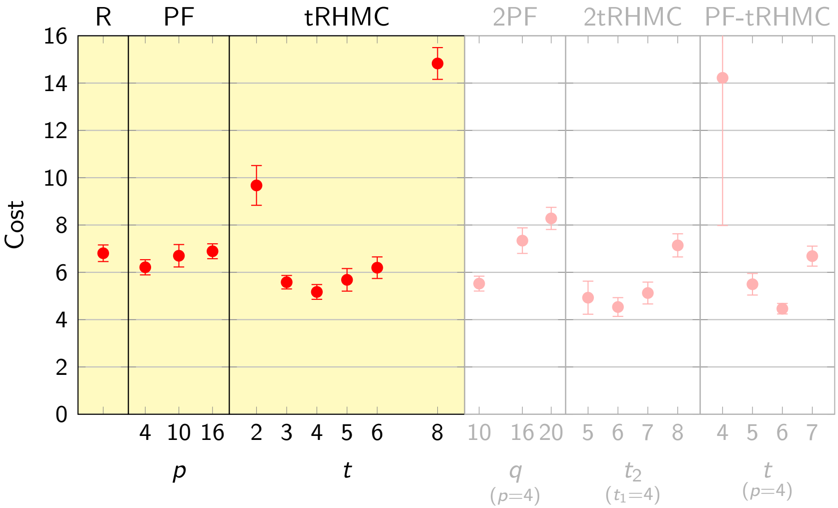
Step-size tuning (force-balanced)

- Step-sizes $h_i = \{h_G, h_1, h_2, \ldots, h_n\}$ for each action term can be tuned by force balancing.
- Given the forces F_i for each action term, the step-sizes follow

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

- F_i can be the average or maximum force.
- We then tune the only free step-size, the coarsest scale h_n, such that the acceptance rate $P_{acc} \sim 0.75$.

Filtering: Results (force-balanced)



Combining filters

- We can improve the performance by using two filters.

Combining filters

- We can improve the performance by using two filters.
- For PF-RHMC, add a higher order polynomial $Q(K)$ $(q > p)$ which approximates $P(K)^{-1}K^{-1/2}$, and use action

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

Combining filters

- We can improve the performance by using two filters.
- For PF-RHMC, add a higher order polynomial $Q(K)$ $(q > p)$ which approximates $P(K)^{-1}K^{-1/2}$, and use action

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

- For tRHMC, add another truncation point t_2:

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

Combining filters

- We can improve the performance by using two filters.
- For PF-RHMC, add a higher order polynomial $Q(K)$ ($q > p$) which approximates $P(K)^{-1}K^{-1/2}$, and use action

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

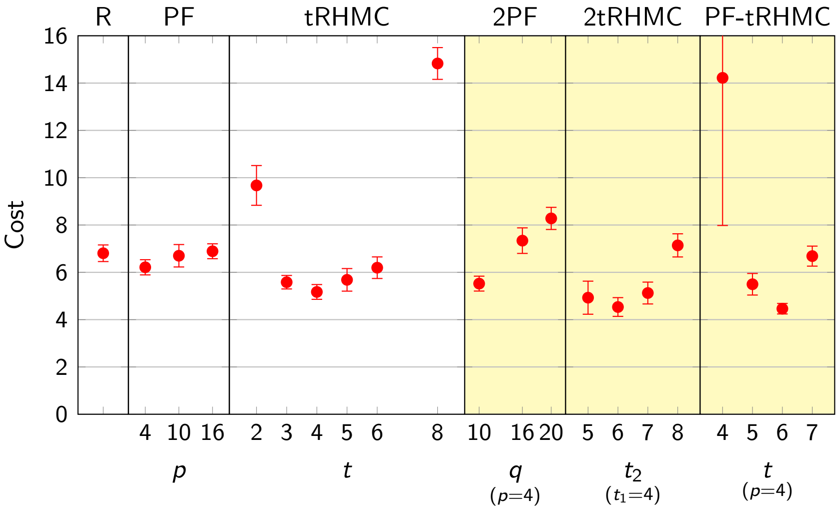
- For tRHMC, add another truncation point t_2:

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

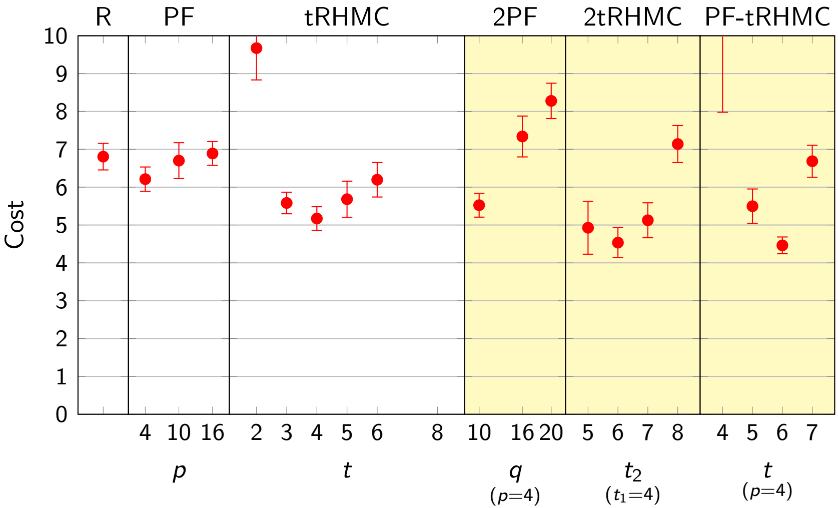
- Can also place a polynomial filter on top of tRHMC:

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

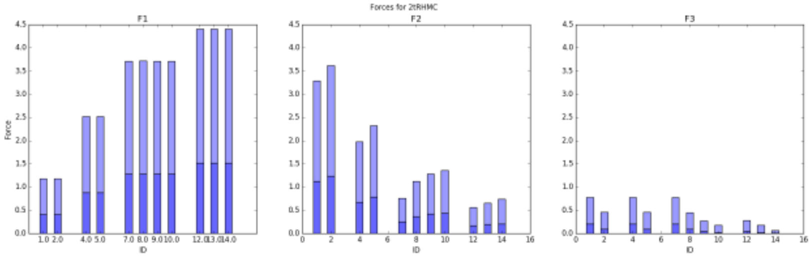
Combining filters: Results (force-balanced)



Combining filters: Results (force-balanced)

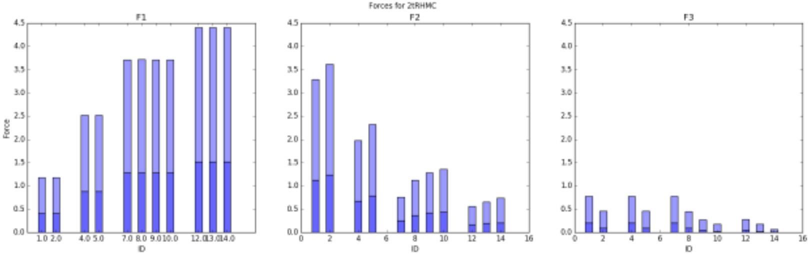


Step-size tuning (characteristic scale)



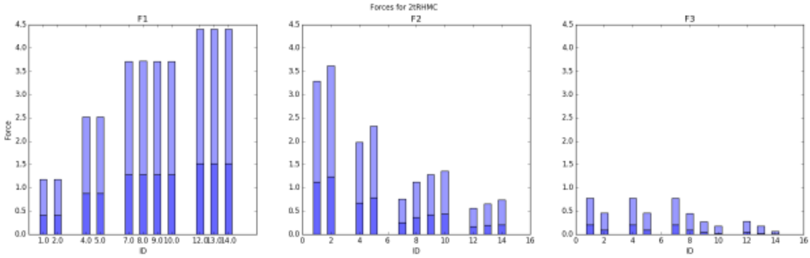
- For higher order truncations, the force due to the remainder term is very small.

Step-size tuning (characteristic scale)



- For higher order truncations, the force due to the remainder term is very small.
- Force-balanced approach to tuning becomes sub-optimal.

Step-size tuning (characteristic scale)



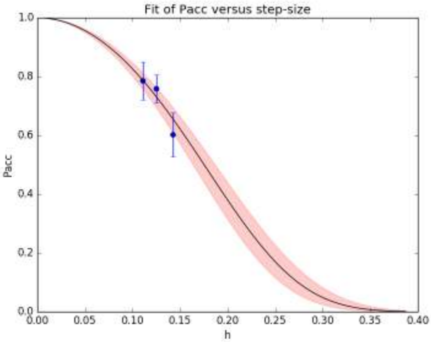
- For higher order truncations, the force due to the remainder term is very small.
- Force-balanced approach to tuning becomes sub-optimal.
- Alternative approach: Note that for filtered actions the acceptance rate is primarily determined by the coarsest scale.

Step-size tuning (characteristic scale)

- Tune the coarsest scale by fitting the acceptance probability with the complementary error function,

$$P_{acc} \approx \mathrm{erfc}(h_n^2 c^2),$$

where the characteristic scale c is the only fit parameter.



Step-size tuning (characteristic scale)

- Tune the coarsest scale by fitting the acceptance probability with the complementary error function,

$$P_{acc} \approx \mathrm{erfc}(h_n^2 c^2),$$

where the characteristic scale c is the only fit parameter.

- Tune the second-coarsest scale manually to achieve target acceptance rate.
- Keep the remaining (finer) step-sizes in sync according to force-balanced approach (relative to the second-coarsest scale)

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

Step-size tuning (characteristic scale)

- Tune the coarsest scale by fitting the acceptance probability with the complementary error function,

$$P_{acc} \approx \mathrm{erfc}(h_n^2 c^2),$$

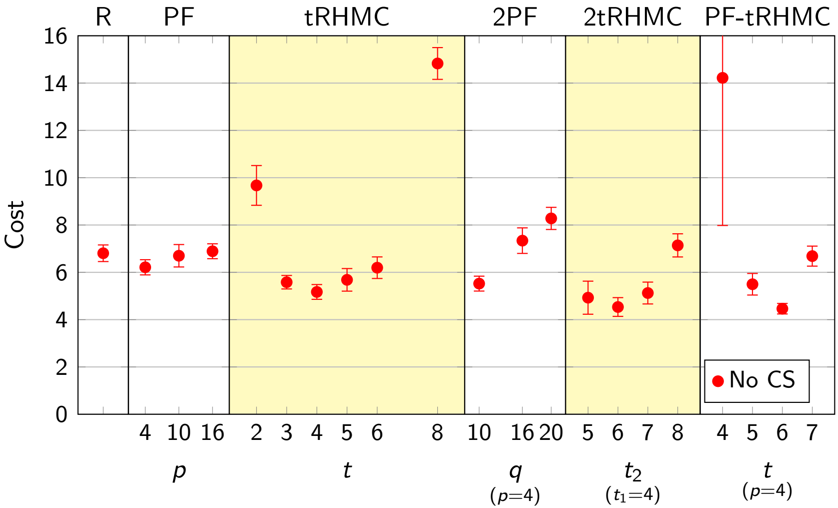
where the characteristic scale c is the only fit parameter.

- Tune the second-coarsest scale manually to achieve target acceptance rate.
- Keep the remaining (finer) step-sizes in sync according to force-balanced approach (relative to the second-coarsest scale)

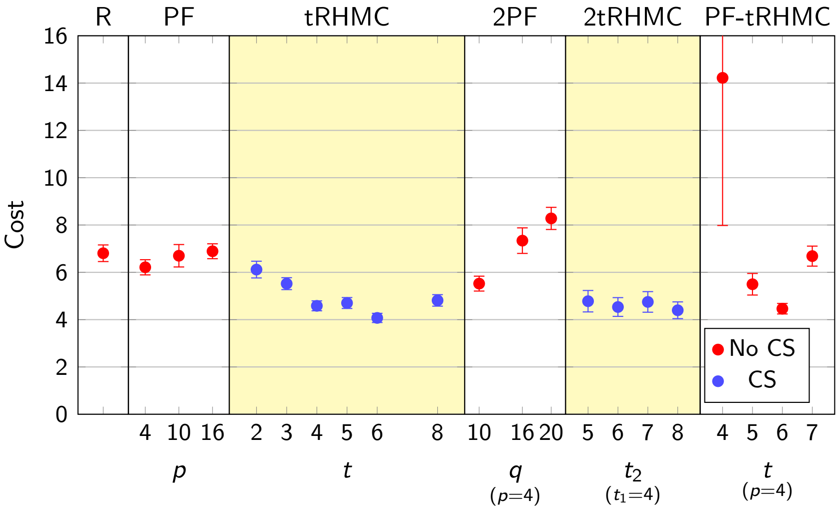
$$F_i h_i \approx \text{constant}$$

- This helps offset an unfavourable distribution of forces between action terms.

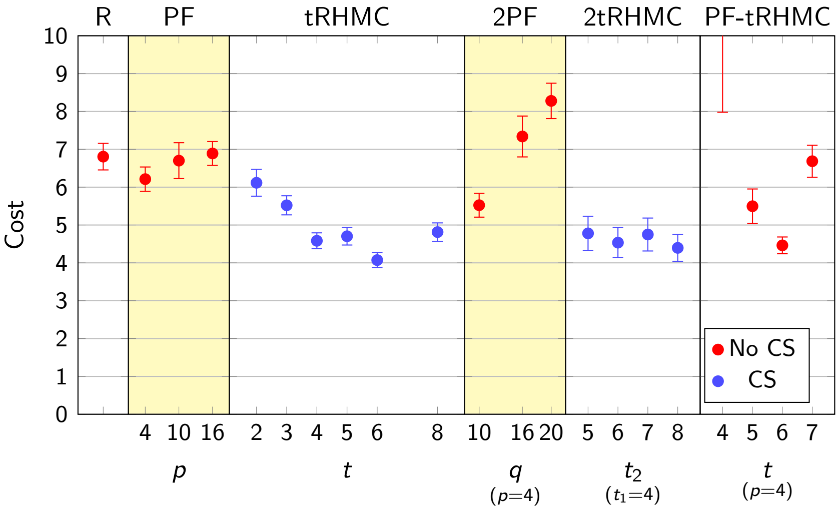
CS-tuning: tRHMC



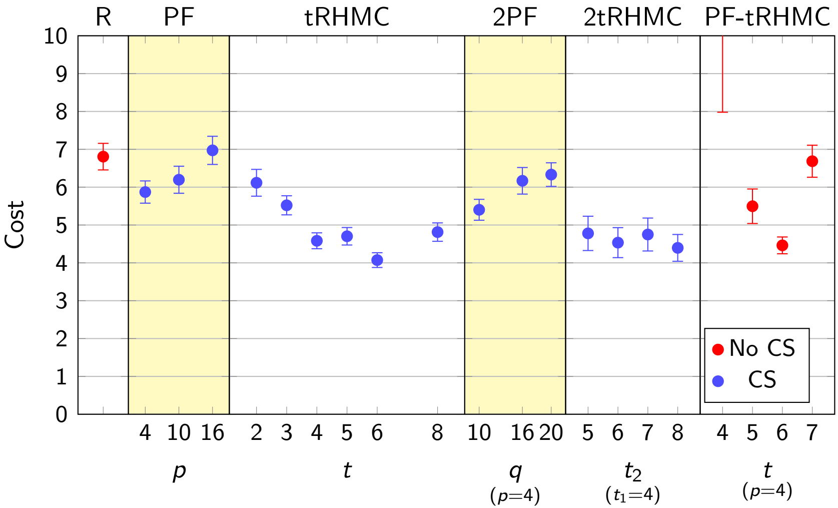
CS-tuning: tRHMC



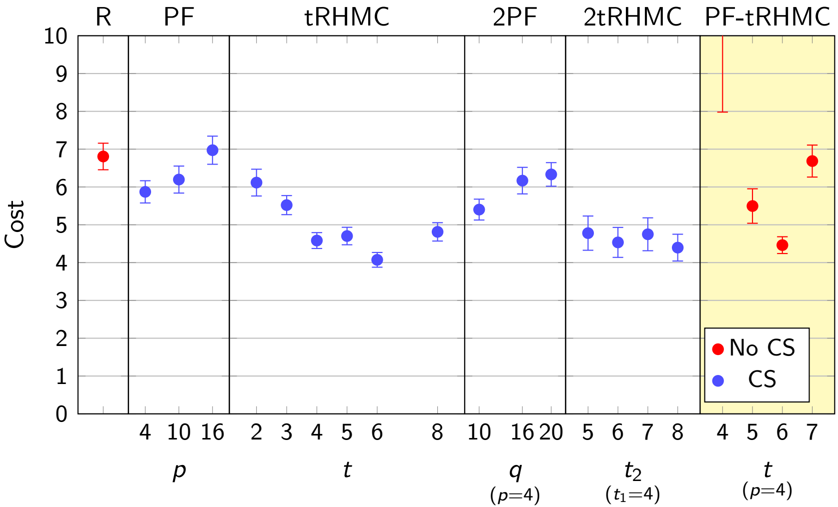
CS-tuning: PF-RHMC



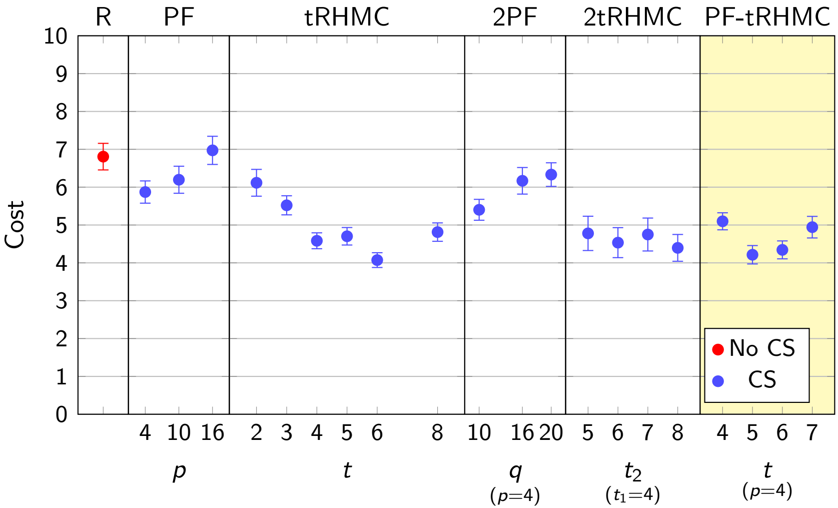
CS-tuning: PF-RHMC



CS-tuning: PF-tRHMC



CS-tuning: PF-tRHMC



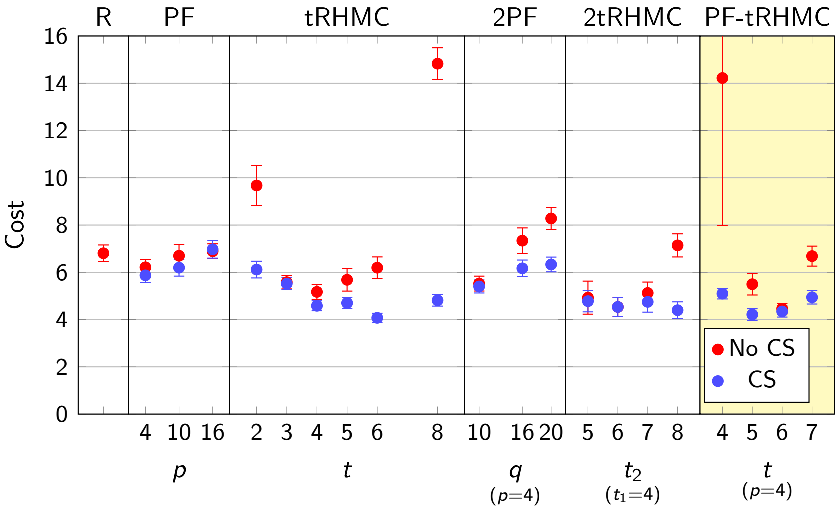
Summary

- Tested different filtering methods to improve the cost of generating single-flavour pseudofermions:
  - Polynomial-filtered RHMC.
  - Truncated-ordered-product filtering (tRHMC).
- tRHMC provides the best performance improvement, roughly factor of 2 speedup.
- Characteristic scale tuning works well for tRHMC across a wide range of truncation orders.
- tRHMC is provided by BQCD, but is also (almost) trivial to implement in existing RHMC codes.
- Multiple truncation filters can be used (should prove to be useful at light quark masses).

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 $[5 \times 10^{-5}, 3]$.

Full cost plot



Full cost plot