An application of the UV-filtering preconditioner to the Polynomial Hybrid Monte Carlo algorithm

Ken-Ichi Ishikawa

for PACS-CS collaboration

S.Aoki, T.Ishikawa, N.Ishizuka, K.Kanaya, Y.Kuramashi, M.Okawa, K.Sasaki, Y.Taniguchi, A.Ukawa, T.Yoshie

1. Introduction

- Dynamical Lattice QCD simulation
  - Needs huge amount of computational power.
  - Improving numerical algorithm is important.

  - Hybrid Monte Carlo algorithm (HMC)
  - Multi Boson algorithm (MBA)
    - Various improvements have been proposed for these algorithms.

Recent improvement program for Dynamical Hybrid Monte Carlo algorithm are based on the combination of

- UV-IR decoupling / Preconditioner

  [de Forcrand, Takaishi, NPB(Proc.Suppl.)53,Lat96]

- Multi time scale Molecular Dynamics (MD) integrator [Sexton-Weingarten, NPB 380(92)]
  - Hasenbusch's heavy mass preconditioner

    [PLB 519 (01),Urbach, Jansen, Shindler & Wenger, CPC 174(06)]
  - Lüscher's domain decomposition SAP preconditioner [CPC 165 (05)]
  - Polynomial preconditioner [Kamleh & Peardon, PoS(LAT05)106; Peardon, this conference]
  - ILU preconditioner [Peardon, hep-lat/0011080]
  - SSOR[Guagnelli & Heitger, CPC130(00)]

- The basic idea is to deform quark part of QCD partition function
  - QCD partition function

$$Z = \int DU \, \det[\, D[U\,]]^{N_f} \, e^{-S_G[U\,]}$$

  - Using a preconditioner $P[U]$

$$Z = \int DU \det[P[U]D[U]]^{N_f} \det[P[U]]^{-N_f} e^{-S_G[U]}$$
$$= \int DU \det[Q[U]]^{N_f} \det[P[U]]^{-N_f} e^{-S_G[U]}$$

  - If $Q[U] \approx 1$ and the cost of *P*[*U*] is cheaper than that of *D*[*U*], we expect that the total cost can be reduced.
  - apply HMC algorithm.

- For MBA, de Forcrand proposed the UV-filter preconditioner.

[de Forcrand, NPB(Proc.Suppl.)73(Lat98);
Alexandrou, de Forcrand, D'Elia & Panagopoulos, PRD61(00)]

Preconditioner P[U]

$$P[U] = \exp(-s(D[U]-1))$$

$$\det[P[U]] = \exp(s\mathrm{Tr}[D[U]-1]) = \exp[S_{UV}[U]]$$

For the Wilson-Dirac operator *D*[*U*], Suv[*U*] is still local action.

- QCD partition function

$$Z = \int DU \det[P[U]D[U]]^{N_f} \det[P[U]]^{-N_f} e^{-S_G[U]}$$

$$= \int DU \det[Q[U]]^{N_f} e^{-S_G[U]-N_f Suv[U]}$$

- Then apply Multi Boson algorithm to the modified partition function.

- In this talk we apply the UV-filter preconditioner to the Polynomial HMC algorithm and investigate the efficiency of the UV-filtering.

2. UV-filtered PHMC algorithm

- We start with the QCD partition function with O(a)-improved Wilson fermion in symmetrically even/odd-site preconditioned form.

$$Z = \int DU \det[D[U]]^{N_f} e^{-S_G[U]-S_{clv}[U]}$$

where

$$S_{clv} = -N_f \mathrm{Tr}[\mathrm{Log}[T]]$$

M_{eo} : Hopping matrix [odd sites $\Rightarrow$ even sites]

(contains hopping parameter κ)

$$D[U] = 1 - T_{ee} M_{eo} T_{oo} M_{oe} = 1 - \hat{M}_{ee}$$

$$T = (1 + c_{sw} \kappa \sigma \cdot F)^{-1}$$

$$\hat{M}_{ee} \approx O(\kappa^2)$$

- The UV-filter preconditioner $P[U]$

$$P[U] = \exp[s\hat{M}_{ee}]$$

s: tunable parameter.

- When s=1, the preconditioned operator Q[U] is 1+O(κ^4).

$$Q[U] = P[U]D[U] = \exp[\hat{M}_{ee}](1-\hat{M}_{ee}) = 1 - \frac{(\hat{M}_{ee})^2}{2} - \frac{(\hat{M}_{ee})^3}{3} - \ldots$$

- The action from the UV-filter P[U] is still local.

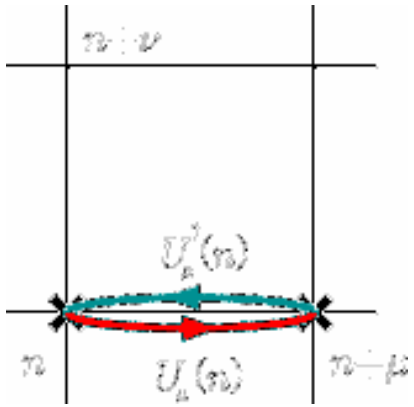
$$\det[P[U]]^{-N_f} = \exp[-N_f s \mathrm{Tr}[\hat{M}_{ee}]] = \exp[-S_{UV}[U]]$$

$$S_{UV}[U] = N_f s \mathrm{Tr}[\hat{M}_{ee}]$$
$$= N_f s \kappa^2 \sum_{n,\mu} \mathrm{tr}_{\mathrm{color,dirac}}[T(n)(1-\gamma_\mu)U_\mu(n)T(n+\mu)(1+\gamma_\mu)U_\mu^+(n)]$$

For unimproved case, Suv[U]=0 and further filtering can be achieved by O(κ^4) filter. [de Forcrand, Lat98, for MBA]

$$\det[\exp[s_1\hat{M}_{ee} + s_2(\hat{M}_{ee})^2]] = \exp\left(s_2 \ \mathrm{Const} \sum_{n,\mu,\nu} \mathrm{Tr}[P_{\mu\nu}(n)] \right)$$

$$(1-\hat{M}_{ee})\exp\left[\hat{M}_{ee} + \frac{1}{2}(\hat{M}_{ee})^2\right] = 1 - \frac{(\hat{M}_{ee})^3}{3} - \ldots$$



- Polynomial HMC (PHMC) partition function (Nf=2)

$$Z = \int DPDUD\Phi^* D\Phi \det[W[U]]^{N_f} e^{-H[P,U,\Phi]}$$

$$H[P,U,\Phi] = \mathrm{Tr}[\mathrm{P}^2] + S_G[U] + S_{clv}[U] + S_{UV}[U] + S_Q[U,\Phi]$$

$$S_Q = | P_{Npoly}[\hat{M}_{ee}]\Phi |^2$$

$$W = P_{Npoly}[\hat{M}_{ee}](1-\hat{M}_{ee})\exp[s\hat{M}_{ee}]$$

$$P_{Npoly} = \sum_{k=0}^{Npoly} c_k (\hat{M}_{ee})^k$$

  - c_k and s are tuned to satisfy $W \approx 1$. When *s*=0, this is the normal PHMC algorithm.
  - The effect of det[*W*] is incorporated by global noisy Metropolis test.
- We investigated the efficiency of this algorithm.

3. Test results

- Lattice parameters
  - Plaquette gauge action

$$16^3 \times 48,\ \beta = 5.2,\ c_{sw} = 2.02,\ \boxed{N_f = 2}$$

	κ	M_{PS}/M_V
Heavy	0.1340	0.8
Light	0.1350	0.7

- Simulation parameters
  - 2-time scale Sexton-Weingarten MD integrator in UPU order.

$$\left[\left(U\left(\frac{\tau}{2N_1N_0}\right)P_{UV}\left(\frac{\tau}{N_1N_0}\right)U\left(\frac{\tau}{2N_1N_0}\right)\right)^{\frac{N_0}{2}} P_{IR}\left(\frac{\tau}{N_1}\right)\left(U\left(\frac{\tau}{2N_1N_0}\right)P_{UV}\left(\frac{\tau}{N_1N_0}\right)U\left(\frac{\tau}{2N_1N_0}\right)\right)^{\frac{N_0}{2}}\right]^{N_1}$$

$U(d\tau)$: update link by $d\tau$

$P(d\tau)$: update momentum by $d\tau$

$P_{UV}(d\tau) : S_G + S_{clv} + S_{UV}$

$P_{IR}(d\tau) : S_Q$

- Choice of the polynomial coefficients
  - Adopted polynomial [de Forcrand, Lat98, for MBA]
    - Minimize

$$R(\vec{c},s)=\left|\left(\sum_{k=0}^{Npoly} c_k(\hat{M}_{ee})^k(1-\hat{M}_{ee})\exp[s\hat{M}_{ee}]-1\right)\eta\right|^2$$

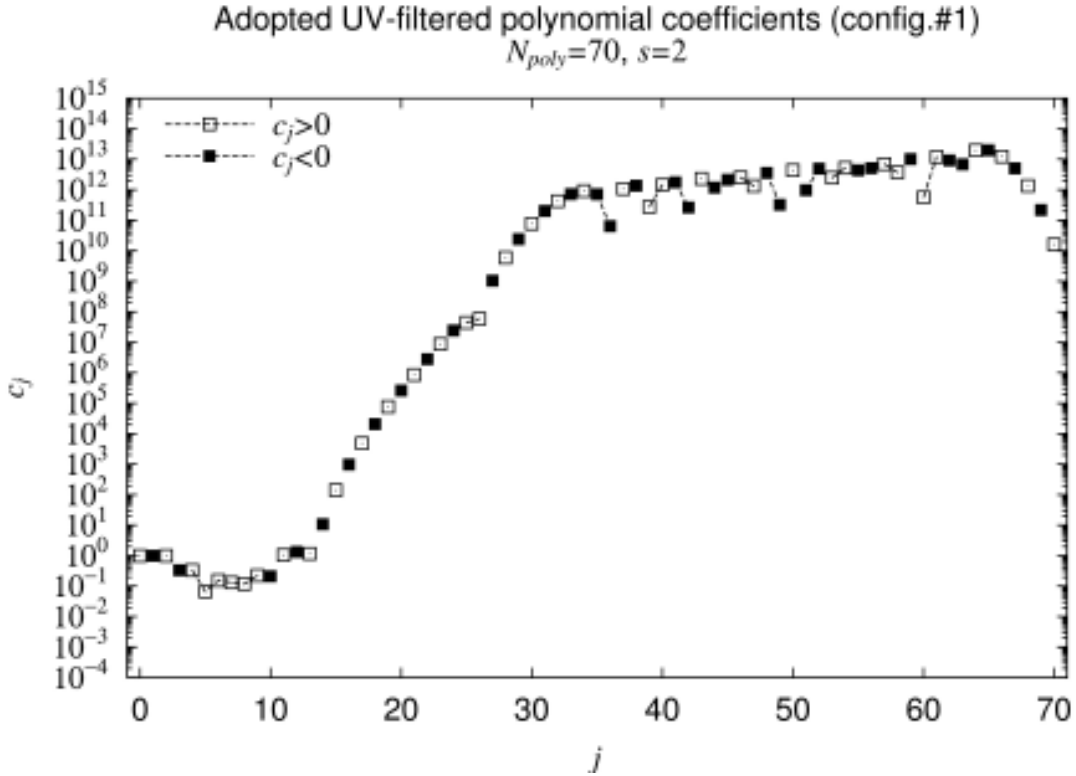
as a function of c_k and s

  - Taylor expansion

$$\sum_{k=0}^{Npoly} c_k(\hat{M}_{ee})^k \approx \left[(1-\hat{M}_{ee})\exp[s\hat{M}_{ee}]\right]^{-1}$$

$$c_k=\sum_{j=0}^{k}\frac{(-s)^j}{j!}$$

Adopted polynomial coefficient



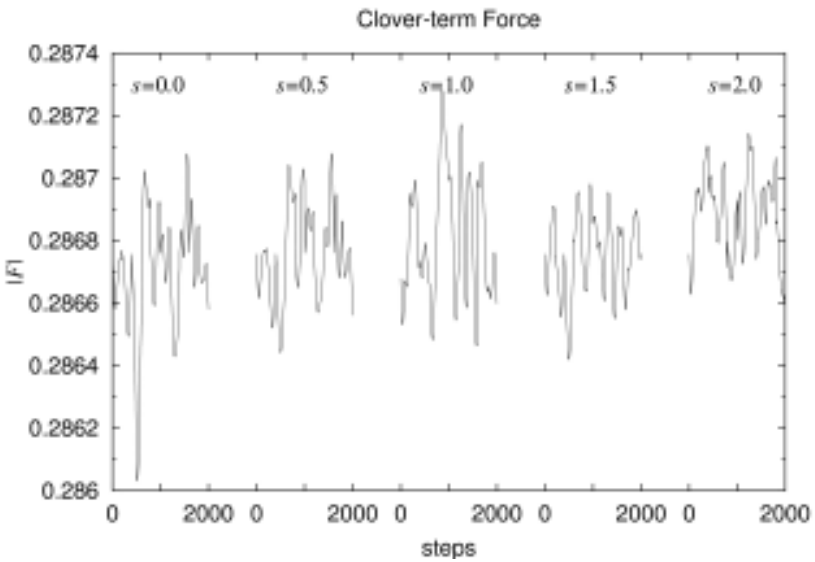
- Ranges O(10^15) → Loss of significant digit.
- We use Taylor expansion coefficient for c_k.

- MD force norm $\boxed{Nf=2 \text{ case}}$
  - We measured the MD force norm for a few trajectories start from a thermalized config.

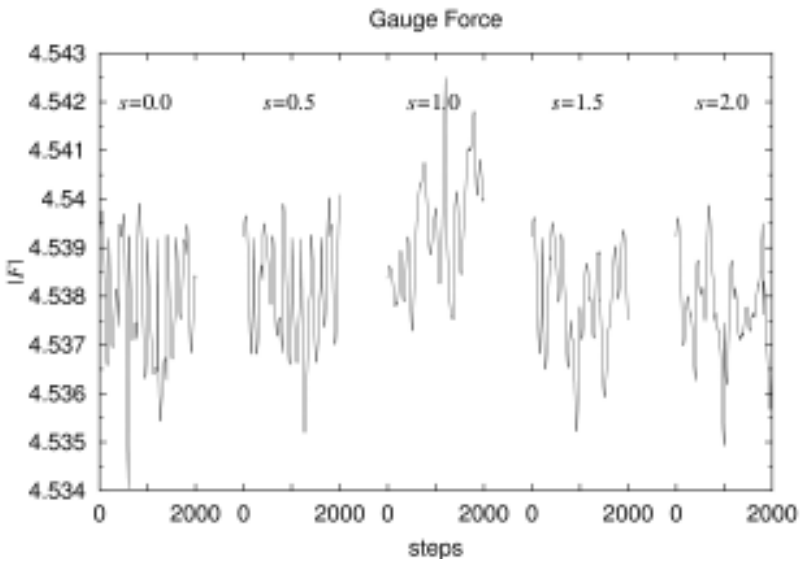
$$|F| \equiv \frac{1}{4L^3T}\sum_n\sum_\mu 2\mathrm{Tr}[F_\mu(n)F_\mu^+(n)]$$

  - $M_\pi/M_\rho \approx 0.7$ case (*Npoly*=160)

Clover-term force



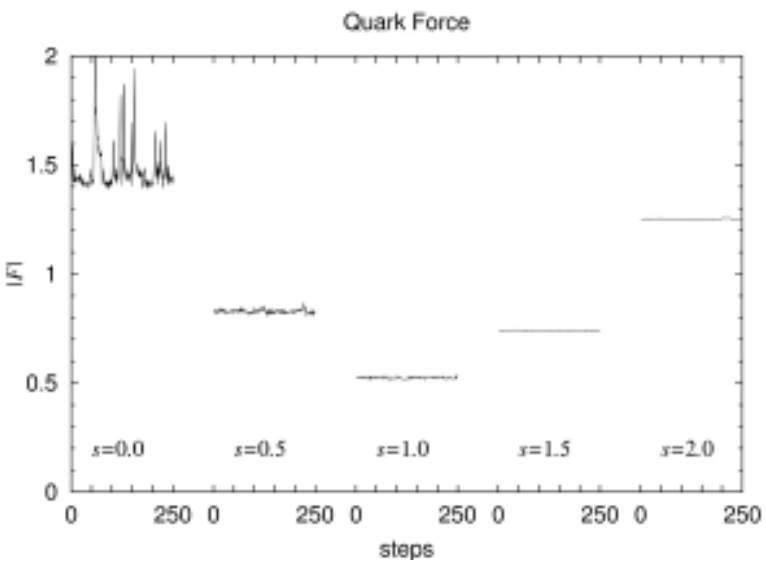
Gauge action force



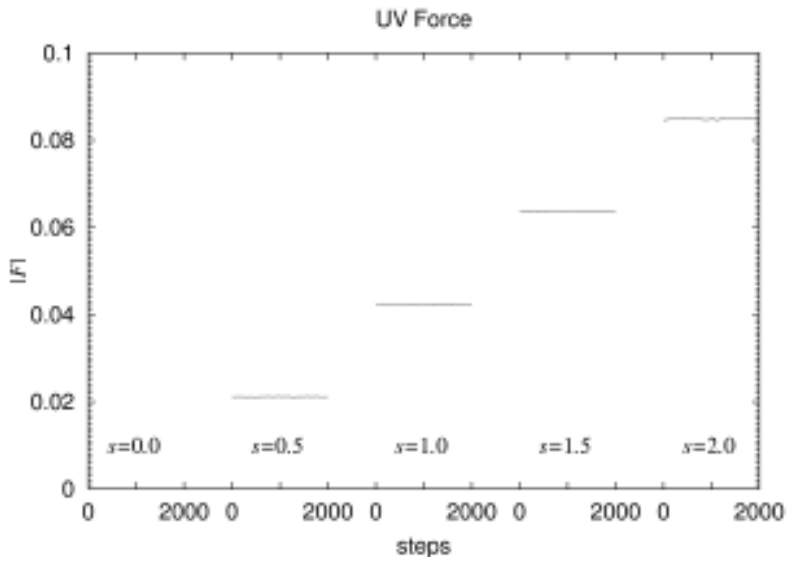
Independent of *s*

- MD force norm (cont'd) $M_\pi / M_\rho \approx 0.7$ $\boxed{Nf\text{=2 case}}$

Pseudo-fermion force



UV-filter force



s=1 factor 3 smaller than s=0

S=0, (UV-filter action) = 0

We expect a factor 3 speed up with multi-time scale MD integrator

- Computational cost comparison

Nf=2 case

  - $M_\pi / M_\rho \approx 0.7$ case (*Npoly*=160)

[τ ,N_0,N_1]	s	Traj.	HMCacc	GMPacc	#Mult/ traj.
[1,25,4]	1.1	1000	0.692(18)	0.868(16)	12640(29)
[1,70]	0 (PHMC)	1100	0.762(16)	0.871(14)	30385(20)
[1,70]	N.A. (HMC)	340	0.80(3)	N.A.	37491(166)

PHMC : s=0, without multi time scale (SW) MD integrator
HMC : even/odd site preconditioned HMC without SW MD integrator

$d\tau = 1/25$: IR part step size

$d\tau = 1/25/4 = 1/100$: UV part step size

~2 speed up from PHMC algorithm without SW MD integrator.

- Computational cost comparison(cont'd) *Nf*=2 case
  - $M_\pi / M_\rho \approx 0.8$ case (*Npoly*=80)

[τ,N_0,N_1]	s	Traj.	HMCacc	GMPacc	#Mult/ traj.
[1,20,6]	1.1	1500	0.742(14)	0.894(10)	5587(13)
[1,56]	0 (PHMC)	1100	0.780(24)	0.895(17)	12456(15)

2.2 speed up from PHMC algorithm without SW MD integrator.

4. Summary

- We have tested the UV-filtered PHMC algorithm (Nf=2).
- With *s*=1 (uv-filter parameter), the contribution from the pseudo-fermion is reduced by a factor 3 in the MD integrator.
- The computational cost is reduced by a factor 2.
  - This algorithm can be applicable to the heavy (preconditioner) part of the Hasenbusch's heavy mass preconditioner and to the Kamleh and Peardon's Polynomial filtering.
  - We will apply this algorithm to the single flavor part of Nf=2+1 simulations. [Kuramashi, LDDHMC+UVPHMC, yesterday]
  - We did not investigated the algorithm with the unimproved Wilson quark action. In this case we can eliminate the term up to $O(\kappa^6)$ with no extra cost.

Backup slides

5. Matrix exponential

- How to evaluate the UV-filter $\exp(s\hat{M}_{ee})$
  - This is required in the beginning of the MD integrator to produce the pseudo-fermion field.
  - This is required in the evaluation of correction term, i.e. Global-noisy Metropolis test.

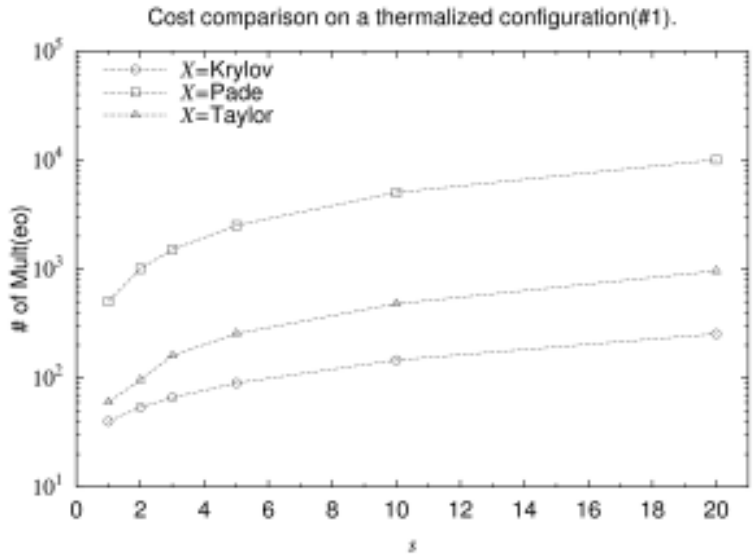
$$W = P_{Npoly}[\hat{M}_{ee}](1-\hat{M}_{ee})\exp(s\hat{M}_{ee})$$

  - There are several ways to do this.
    - Taylor approximation method
    - Pade approximation method
    - Krylov approximation method (EXPOKIT)

5. Matrix exponential (cont'd)

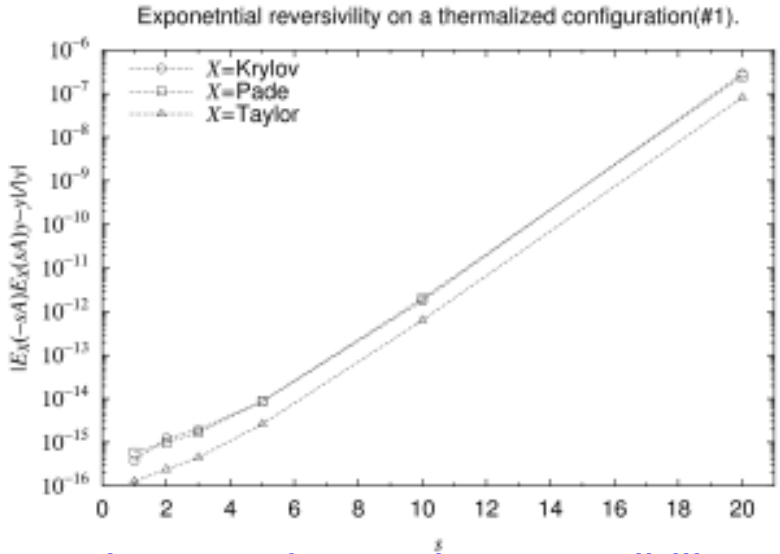
- These methods require the norm of $\hat{M}_{ee}$. We evaluate it from the maximal eigenvalue of $\hat{M}_{ee}(\hat{M}_{ee})^{+}$

Cost comparison (accuracy fixed)



Krylov subspece method is the best.

Exponential Reversibility check (exp(A)exp(-A)=1)



s becoms large, the reversibility becomes worse

6. Single flavor case

- Single flavor case we need the (inverse) square root of correction matrix W.
  - The Taylor approximation method has been used by CP-PACS/JLQCD collaboration in Nf=2+1 simulations.
    - Needs residual check to ensure the accuracy.
    - This doubles the cost.
  - Krylov subspace method is also available.

    [Eiermann & Ernst, preprint 2005]

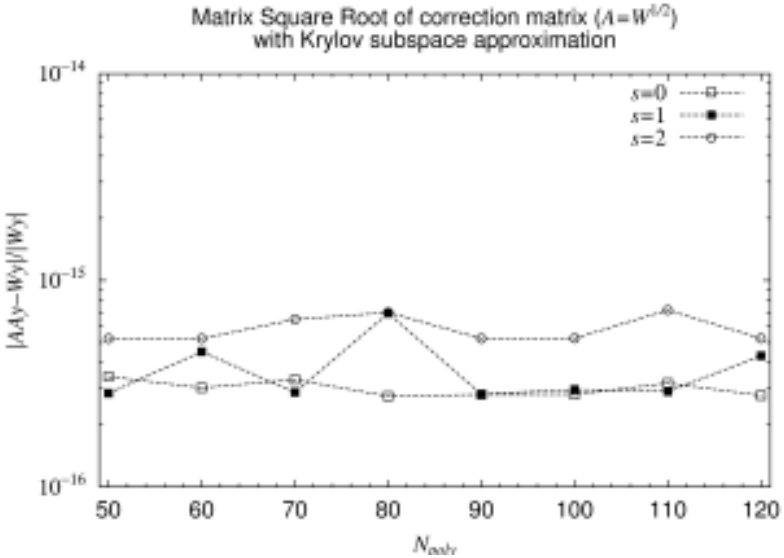
    - This method does not need the explicit computation of the residual.
  - We tested the Krylov method for the sqrt of W.

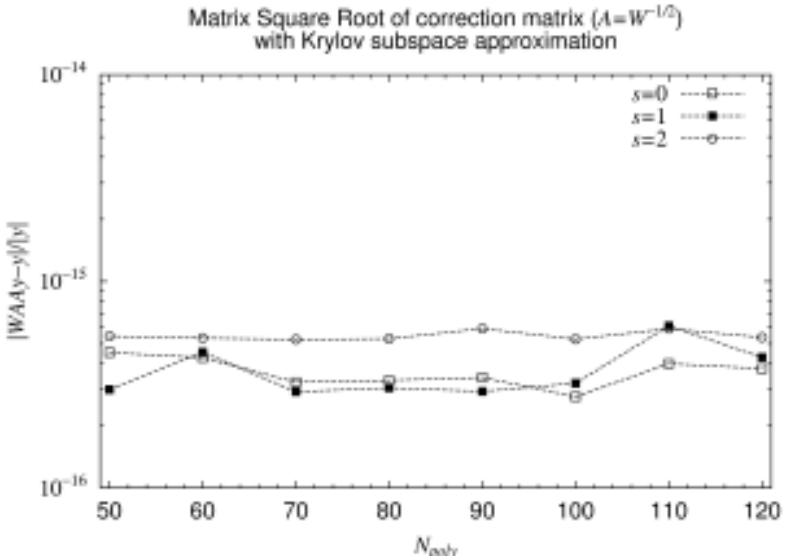
6. Single flavor case (cont'd)

- Residual check

$$\left|(W^{\frac{1}{2}})(W^{\frac{1}{2}})\eta - W\eta\right| \Big/ \left|W\eta\right|$$

$$\left|W(W^{-\frac{1}{2}})(W^{-\frac{1}{2}})\eta - \eta\right| \Big/ \left|\eta\right|$$



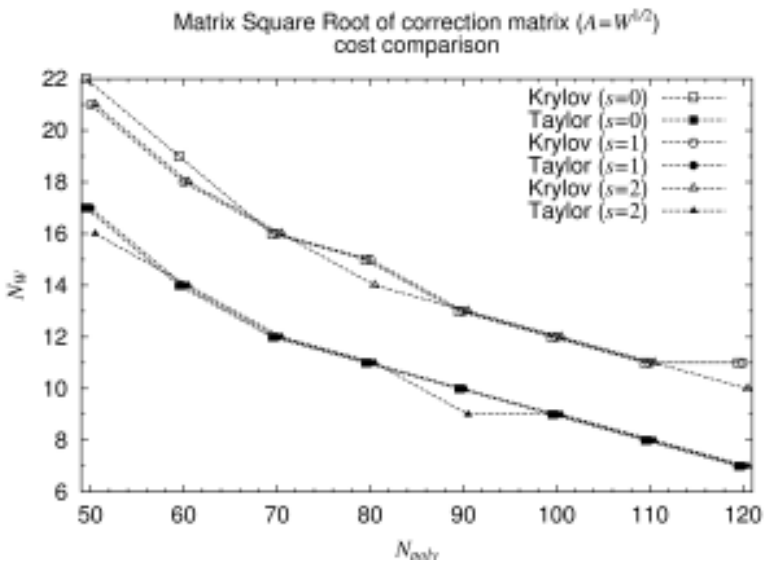


Relative residuals reach double precision accuracy

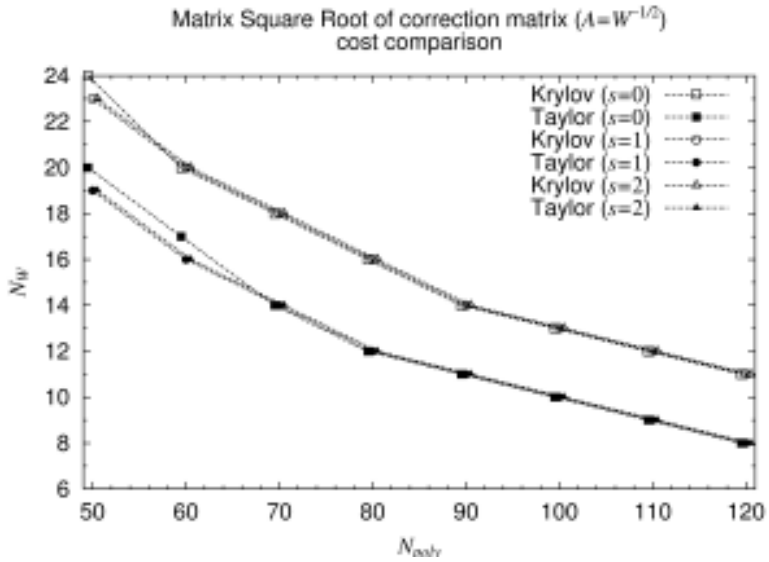
6. Single flavor case (cont'd)

- Cost comparison (Krylov vs Taylor)

$$W^{\frac{1}{2}}$$

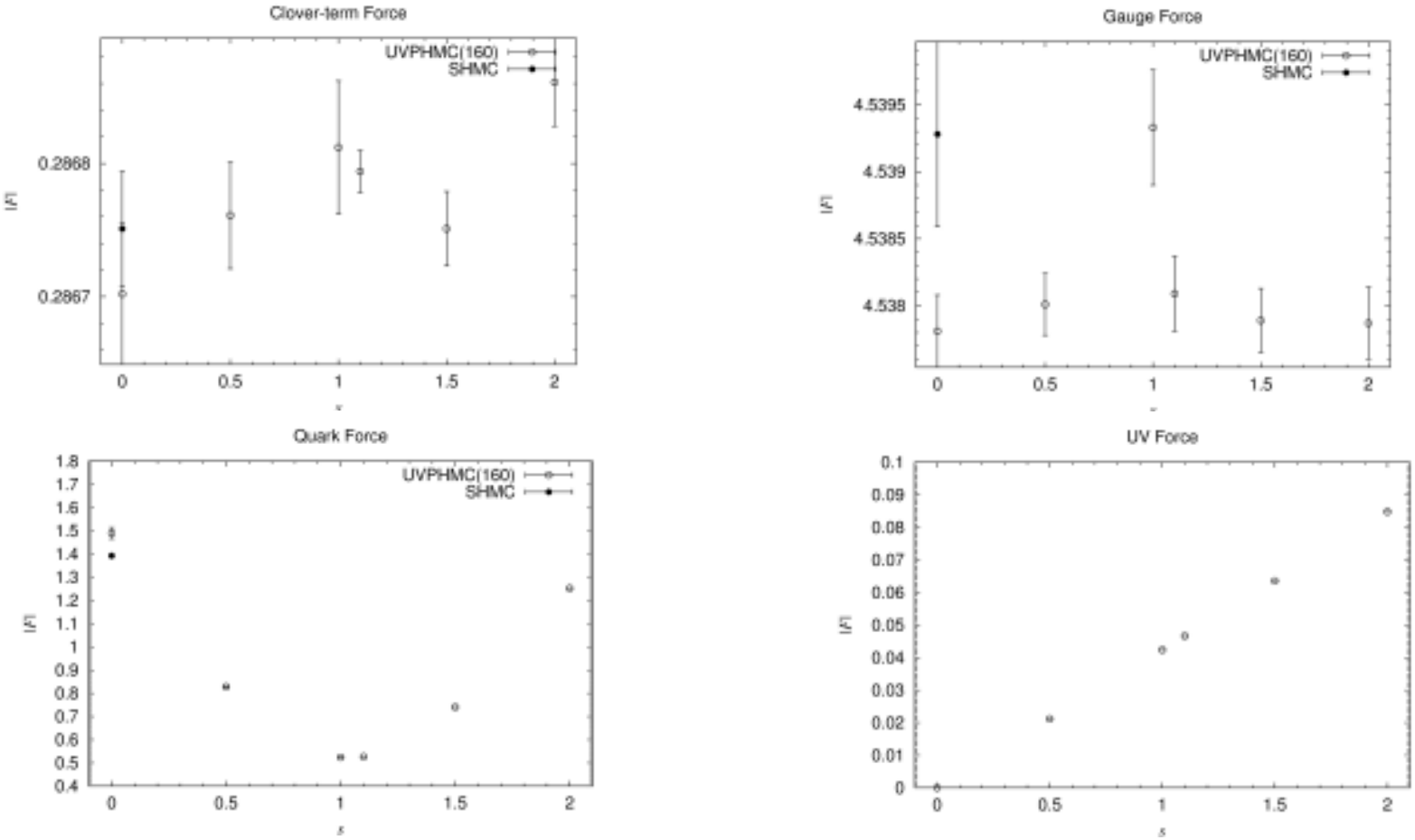


$$W^{-\frac{1}{2}}$$



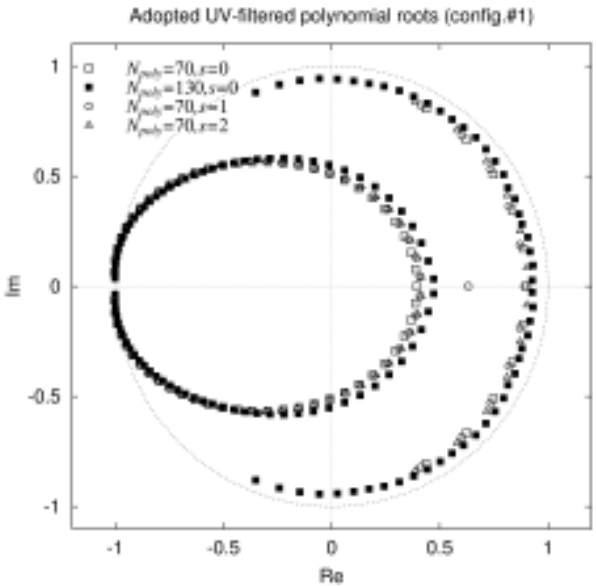
The cost of the Taylor method is smaller than that of the Krylov method.
The Krylov method, however, does not need the explicit residual computation.
The total cost of the Krylov method seems to be cheaper than the Taylor.

- UV-filter parameter *s* dependence of the MD force. (Nf=2)



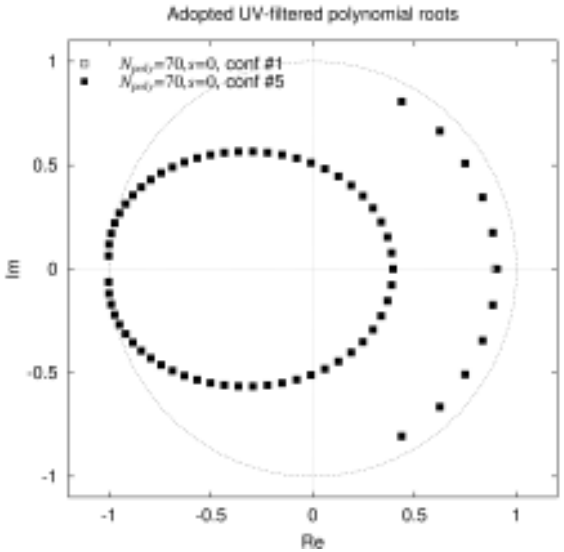
s=1 seems to be the minimum of pseudo-fermion froce.

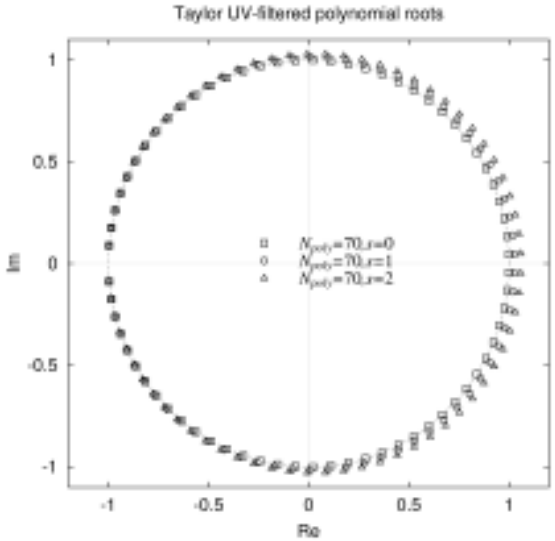
Polynomial roots distribution



- Adopted polynomial

- Adopted poly. Config. dependence





- Taylor (Chebyshev) polynomial