Algorithms

MITP - Summer school

Aug. 1st , 2025

J. Finkenrath

Outline - Part 3

- Part 0
  - Markov Chain Monte Carlo
  - Hybrid Monte Carlo algorithm
- Part 1
  - Linear solvers, Krylov subspace solvers
  - Preconditioners, smoothers and coarse grid
- Part 3
  - Details on multigrid preconditioning
  - Fermions in simulations

Recap: Krylov subspace solver

Iterative solver: define Krylov subspace

$$K_m = \mathrm{span}(r, Ar, A^2 r, \ldots, A^{m-1} r)$$

with

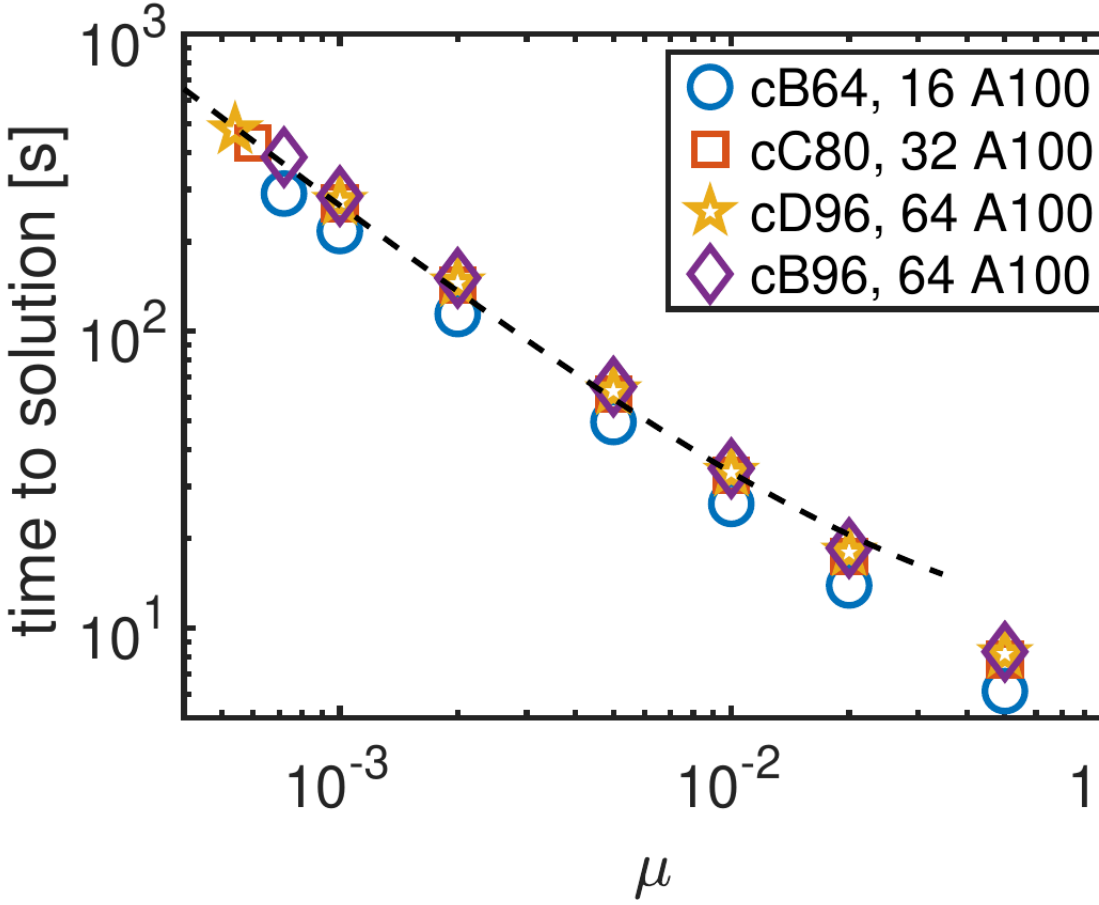
$$\tilde{x} \in x_0 + K_m \qquad \text{and} \qquad r \perp L$$

For **CG** solver follows

$$\mathrm{cost} \approx V \frac{c_0}{\mu}$$

Iteration count proportional to condition number

- depends on the smallest mode
- iteration count increases drastically at physical point



Recap Multigrid

Idea: Preconditioning **IR** and **UV** modes

Error can be split:

$$e = e_{low} + e_{high}$$

- a good smoother eliminates e_{high} high effectively

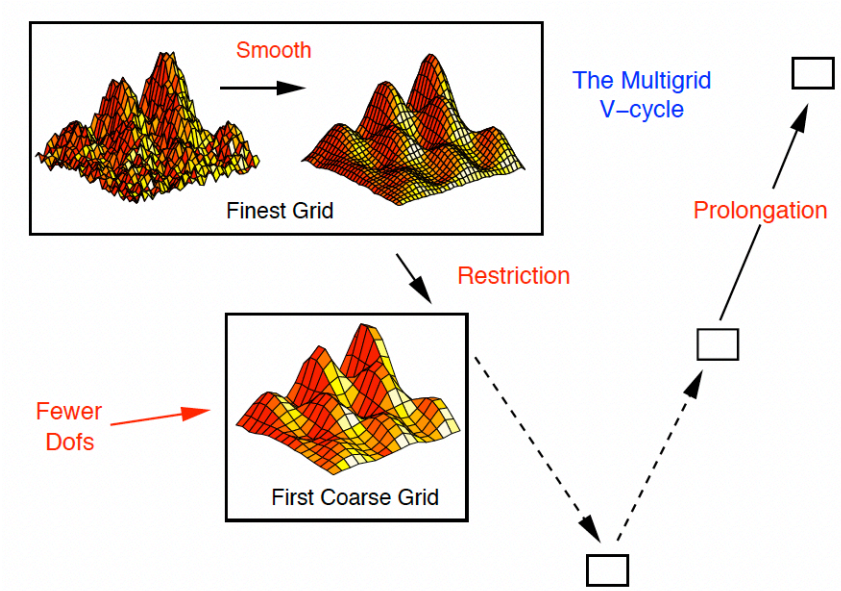
A good choice: SAP which apply block inverses in alternating procedure

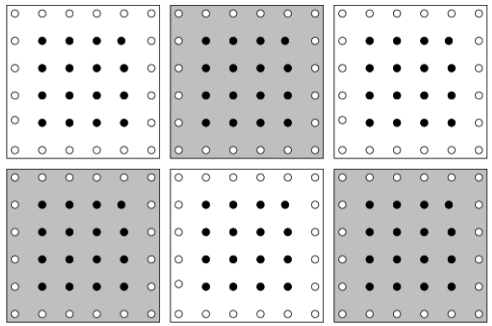
$$x^{(k)} \leftarrow (1 - SA)x^{(k)} + Sb$$

where $S = A_r^{-1} + A_b^{-1} - A_b^{-1} A_{br} A_r^{-1}$

and the error scales

$$e_{SAP} = 1 - SA = (1 - A_b^{-1}A)(1 - A_r^{-1}A)$$





Algebraic multigrid: non-geometric

Two-level AMG

- Petrov-Galerin projection

$$A_c = RDP$$

with R restriction operator, P prolongation operator, A_c coarse grid operator

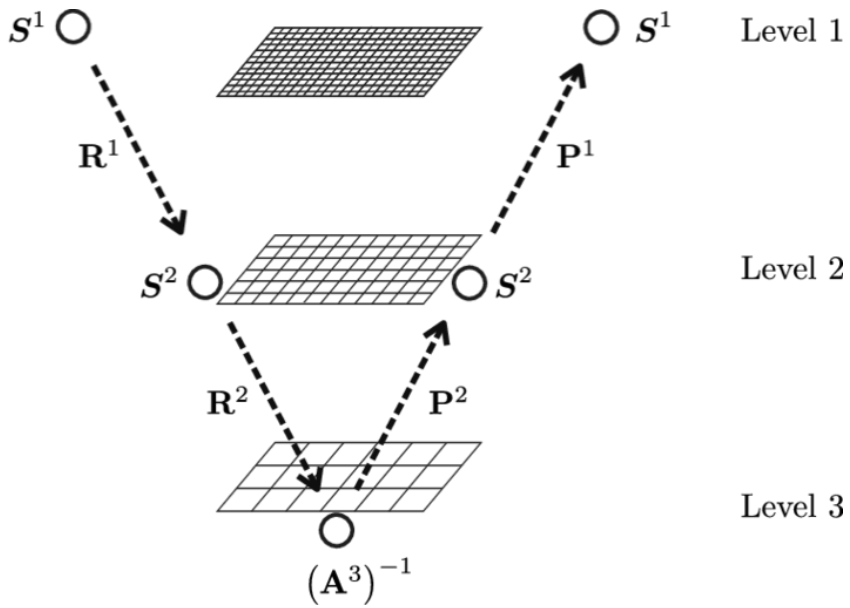
- Coarse-grid correction

$$x^{(k)} \leftarrow x^{(k)} + PA_c^{-1}R \cdot r^{(k)}$$

- Then for the error propagation follows

$$e^{(k)} = I - PD_c^{-1}RA$$

Task: Capture sufficient low modes within P



Aggregation

Construction of P. Lets order the subspaces:

$$L \times S \times C$$

with L spacial lattice, S spin space, C color space

Aggregation blocks:

$$A^{(i,\tau)} = L^{(i)} \times S^{(\tau)} \times C$$

with $(A^{i,\tau} : j = 1, \ldots, n^{(L)}; \tau = 0, 1)$

The **prolongation operator** fullfills:

$$\gamma_5 P = P \gamma_5^c$$

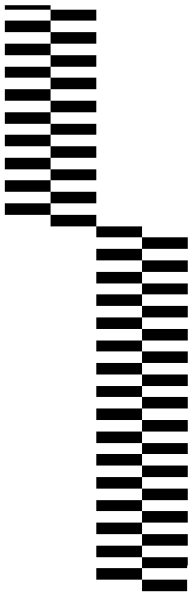
γ_5^c diagonal on the coarse grid. Setting $R = P^\dagger$, we get:

$A_c = P^\dagger D P \qquad \text{and it follows} \qquad \gamma_5^c A_c = A_c^\dagger \gamma_5^c$

Using N test vectors: Split them up into $N \times s$

$$(v_1 \mid \ldots \mid v_N) = \longrightarrow P = \begin{matrix} \mathcal{A}_1 \\ \mathcal{A}_2 \\ \vdots \\ \mathcal{A}_s \end{matrix}$$

Test-vector aggregation additional in spin space:



Building the coarse grid space

Test vectors: use approximate eigenvectors in the construction of P

An ideal approach: Take eigenspace of small eigenvalues

- direct approach to expensive (although iterative procedures for eigenspace calculations exists)
- indirect approach **approximation via A^{-1}. Start with random vector w and apply A^{-1} k-times

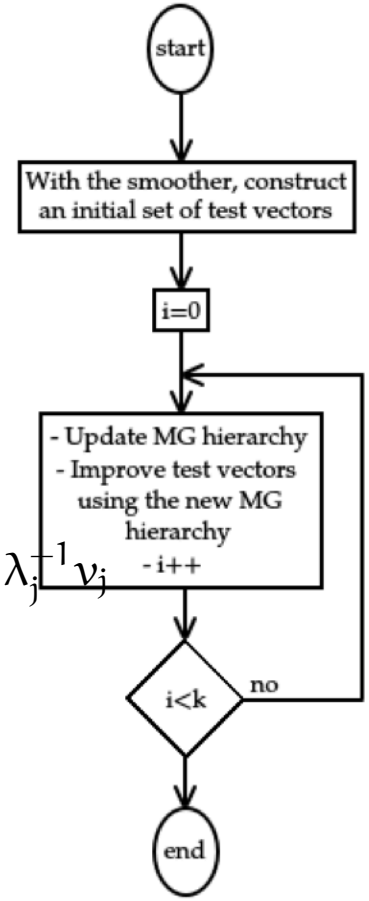
$$w^{(k)} \leftarrow (A^{-1})w^{(0)} \qquad with \quad w^{(1)} = A^{-1}w^{(0)} = \sum_j (v_j, w^{(0)})\lambda_j^{-1} v_j$$

for k large, $w^{(k)}$ good approx. of $v_{smallest\lambda}$ and $A = V\Lambda V^\dagger$ (if $A = A^\dagger$)

- application of k-times A^{-1} for each test vector too expensive

Idea: Use multigrid hirarchy to approx. A^{-1}, update multigrid hirarchy as test vectors improves

$$w^{(k+1)} \leftarrow" A^{-1}(v^{(k)}) " v^{(k)}$$



[Luescher 2007]

Parameters of DDalphaAMG

	parameter		default
setup	number of iterations	n_{inv}	6
	number of test vectors	N	20
	size of lattice-blocks for aggregates on level 1		4^4
	size of lattice-blocks for aggregates on level $\ell > 1$		2^4
	coarse system relative residual tolerance (stopping criterion for the coarse system)$^{(*)}$	ϵ	$5 \cdot 10^{-2}$
solver	restart length of FGMRES	n_{kv}	10
	relative residual tolerance (stopping criterion)	tol	10^{-10}
	number of post-smoothing steps$^{(*)}$	ν	5
	size of lattice-blocks in SAP$^{(*)}$		2^4
	number of Minimal Residual (MR) iterations to solve the local systems in SAP$^{(*)}$		3
K-cycle	maximal length$^{(*)}$		5
	maximal restarts$^{(*)}$		2
	relative residual tolerance (stopping criterion)$^{(*)}$		10^{-1}

$(*)$: same in solver and setup

[https://wwwold.mathematik.tu-dortmund.de/lsix/events/UQ18/talks/Frommer-Dortmund-2018.pdf]

Conclusion and outlook

Lattice libraries

Plenty of solver library exists like

`QUDA` [https://github.com/lattice/quda] : implements a multi-grid solver procedure for various operators on GPUs (Nvidia and AMD)

`openQCD` : implements a 2 lvl-MG procedure optimized for the Wilson-Dirac discretization (currently for CPUs)

`DDalphaAMG` : implements a multi-grid procedure for Wilson-Dirac discretization

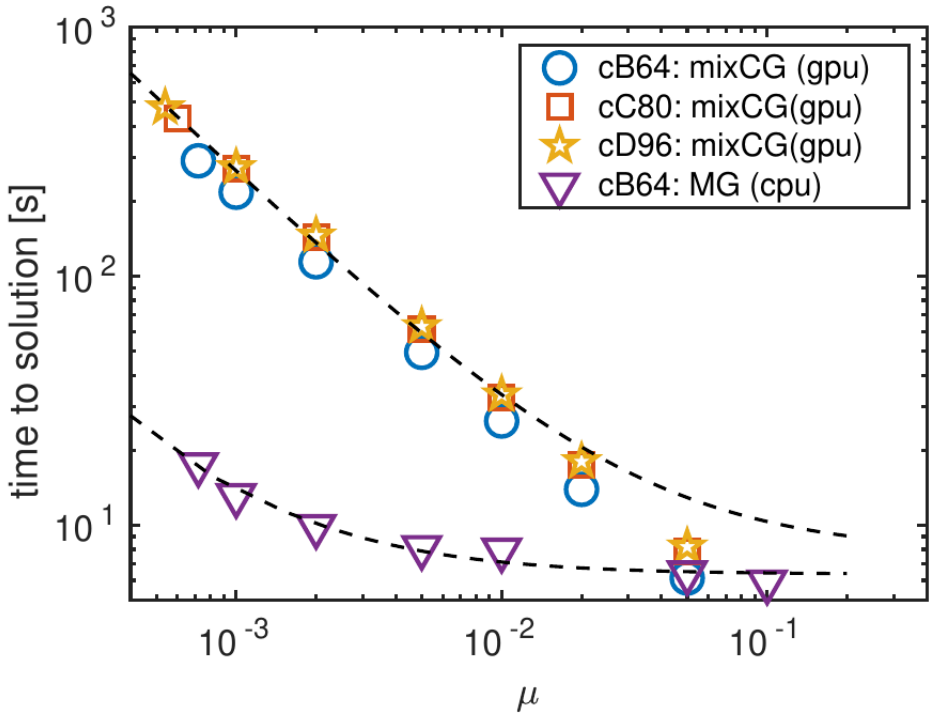
`grid` : implements also multi-grid procedure

multi-grid is computational limited by

- less scalable (coarse grid boundary surface large)
- memory bound

... more ..

Usually main computational challenge:



$$\text{cost} \approx V\left(C_0 + \frac{C_1}{\mu}\right) \qquad \text{with} \qquad C_0 \gg C_1$$

Goal: Evaluation of pathintegral

Recap: Motivation for MCMC

$$< O >= \frac{1}{Z} \int D\phi e^{-S(\phi)} O(\phi) \qquad D\phi = \prod_{i=1}^{M} d\phi_i \quad e.g. \quad \phi = U, \overline{\psi}, \psi$$

- Deterministic integration methods not feasible!

  Current lattice QCD simulations can have $M = O(10^9)$

- Monte Carlo: evaluates integral by sampling the integrand at points selected via probability under the integration measure

Basic idea

1. Generate sequence of field configurations with probability

$$P(\phi^{(t)}) = \frac{1}{Z} e^{-S(\phi^t)}$$

1. Evaluate

$$\overline{O} = \frac{1}{N} \sum_{t=1}^{N} O(\phi^{(t)})$$

Ideal HMC algorithm

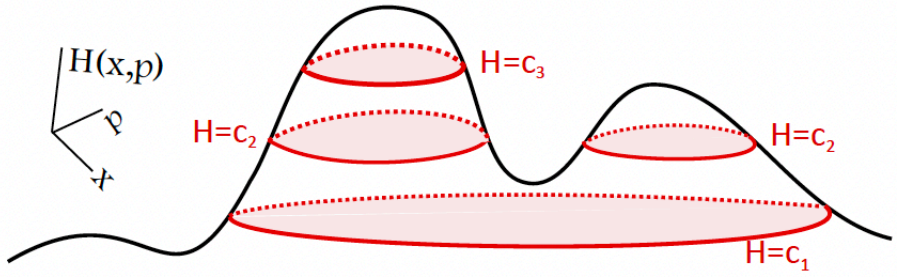
1. Start from a gauge-field $U(0)$
2. Sample a momentum field $P(0)$ from the Gaussian distribution

$$P_\pi = e^{(\pi,\pi)/2}/Z$$

3. Solver Hamilton equations for a time τ

$$(\pi(0), U(0)) \rightarrow (\pi(\tau), U(\tau))$$

4. Repeat 2. and 3. taking $U(0) = U(t)$



Fermions in simulations

Lattice QCD path-integral

$$\langle O \rangle = \frac{1}{Z} \int DU \, D\overline{\psi} \, D\psi \, e^{-S_g[U]} e^{-\overline{\psi} D[U] \psi} O[U, \overline{\psi}, \psi]$$

$$= \frac{1}{Z} \int DU \, e^{-S_g[U]} \det(D[U]) O[U]$$

Fermionic observable via generating function:

$$O[U, \overline{\psi}, \psi] \to O'[U] = O\left[U, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \overline{\eta}}\right] e^{\overline{\eta} D^{-1}[U] \eta}$$

Determinant as observable:

$$\langle O \rangle = \frac{\langle \det(D[U]) O'[U] \rangle}{\langle \det D[U] \rangle}$$

- Calculation of $\det(D[U])$ requires $O(V^3)$ operations
- Overlap problem $\to$ large statistical fluctuations

Fermions in simulations

Lattice QCD path-integral

$$\langle O \rangle = \frac{1}{Z} \int DU \, D\overline{\psi} \, D\psi \, e^{-S_g[U]} e^{-\overline{\psi} D[U] \psi} O[U, \overline{\psi}, \psi]$$

$$= \frac{1}{Z} \int DU \, e^{-S_g[U]} \det(D[U]) O[U]$$

Fermionic observable via generating function

$$O[U, \overline{\psi}, \psi] \to O'[U] = O\left[U, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \overline{\eta}}\right] e^{\overline{\eta} D^{-1}[U] \eta}$$

Determinant in Metropolis

$$P_A \propto e^{-S_g[U'] - S_g[U]} \frac{\det D[U']}{\det D[U]}$$

- for a single link update $\det(D[U + \delta U] D[U]^{-1})$ for a full sweep follows proportional to V^2
- $\det D[U]$ must be real and positiv

Fermions in simulations

Lattice QCD path-integral

$$\langle O \rangle = \frac{1}{Z} \int DU \, D\overline{\psi} \, D\psi \, e^{-S_g[U]} e^{-\overline{\psi} D[U] \psi} O[U, \overline{\psi}, \psi]$$

$$= \frac{1}{Z} \int DU \, e^{-S_g[U]} \det(D[U]) O[U]$$

Fermionic observable

$$O[U, \overline{\psi}, \psi] \to O'[U] = O\left[U, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \overline{\eta}}\right] e^{\overline{\eta} D^{-1}[U] \eta}$$

Determinant as effective action

$$S^{eff} = -\mathrm{tr} \ln(D[U]) \qquad \Rightarrow \qquad F^{eff} = -\mathrm{tr}(D[U]^{-1} \partial D[U])$$

- Difficult and impractical to make the algorithm efficient (and exact)
- $\det D[U]$ must be real and positiv

Fermions in simulations

Pseudofermions *[Weingarten, Petcher 81]*

$$\det(D)^2 = \det(Q^2) \propto \int D\phi^\dagger D\phi e^{-\phi^\dagger Q^{-2}\phi} \qquad [Q = \gamma_5 D = Q^\dagger]$$

- $\det Q^2$ is expressed in terms of a bosonic Gaussian integral with pseudo-fermions interacting non-locally

  $$\det(Q)^* = \det(Q^\dagger) = \det Q \in \mathbb{R}$$

  but for Wilson quarks not necessarily positiv
- positive definite of the fermion kernel is required for the convergence of the integral
- using two degenerate quarks guarantee positive definite operator and allow for an easy pseudo-fermion generation
- more difficult for single quarks

HMC with pseudofermions

Heat-bath

$$P_\eta[\eta] \propto \int D\eta^\dagger D\eta \; e^{-\eta^\dagger \eta} \quad \Rightarrow \quad \phi = Q\eta \quad \Rightarrow P_\phi[\phi]$$

- Generate pseudo-fermions from Gaussian fields between HMC steps
- Averaging results over many trajectories effectively samples the contribution from the pseudofermion integral

Hamiltonian

$$H = \frac{1}{2}(\pi, \pi) + S \qquad S = S_g + S_{pf} \qquad S_{pf} = (Q^{-1}\phi, Q^{-1}\phi)$$

$$\text{with the Dynamics:} \qquad \partial_t U(x,\mu) = \pi(x,\mu)U(x,\mu) \qquad \partial_t \pi(x,\mu) = -F(x,\mu) \qquad \text{with} \qquad F^a(x,\mu) =$$

Idea: pseudofermions are held fixed during the Hamiltonian evolution (not additional momentas needed)

Fermionic forces

$$0 = \partial(M^{-1}M) = \partial M^{-1} \cdot M + M^{-1}\partial M \qquad \Longrightarrow \qquad \partial M^{-1} = -M^{-1}(\partial M)M^{-1}$$

$${F_{pf}}^a{}_{x,\mu} = \partial^a{}_{x,\mu}(\psi,\psi) = -2Re(\chi, (\partial^a{}_{x,\mu}Q)\psi) \qquad \text{with} \qquad \psi = Q^{-1}\phi \quad \text{and} \quad \chi = Q^{-1}\psi$$

Challenges of simulating fermions

Fermionic forces

- Single pseudofermion HMC is not efficient. **Namely**:

$$F_{pf} = (\phi, \partial Q^{-2} \phi) \qquad vs. \quad F_F^{eff} = -2\partial(\mathrm{tr}\ \ln(Q))$$

  F_{pf} is a stochastic estimate of F^{eff} at the start of the trajectory $\langle F_{pf} \rangle = F_F^{eff}$

- F_{pf} has very large fluctuations

$$||F_{pf}|| \gg ||F^{eff}{}_F|| \qquad \Longrightarrow \qquad var(||F_{pf}||) \gg var(F^{eff}{}_F)$$

- **Recall** the shadow Hamiltonian

$$\Delta H_{OMF2} = (c_1 ||F||^2 + c_2 \pi^2 \partial^2 S) h^2 + O(h^4)$$

  large $var(||F_{pf}||) \Rightarrow$ large $var(\Delta H) \Rightarrow$ low $P_A \Rightarrow$ *small* h

- large forces also trigger instabilities more easily $\rightarrow$ limits the step size h

$\Rightarrow$ try/use similar preconditioners idea like for Krylov solvers (seperation of IR and UV modes)

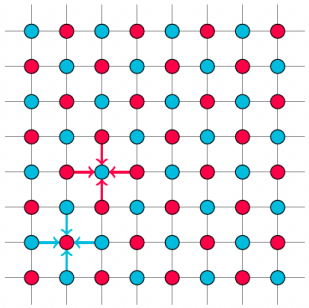
Even-odd reduction/preconditioning

If D only connects nearest-neighboring sites the fermionic problem can be effectively reduced to half of the lattice

Checkerboard decomposition

Even or odd point

Schur decomposition



$$D = \begin{bmatrix} D_{ee} & D_{eo} \\ D_{oe} & D_{oo} \end{bmatrix} \qquad \det D = \det \hat{D} \, \det D_{oo} \qquad \hat{D} = D_{ee} - D_{eo} D^{-1}{}_{oo} D_{oe}$$

Even-odd action

$$S_{pf} \rightarrow S_{pfe} + S_{det} = (\hat{Q}^{-1} \phi_e, \hat{Q}^{-1} \phi_e) - 2\mathrm{tr} \, \ln(Q_{oo})$$

- S_{pfe} involves pseudo-fermions residing only on the even sites of the lattice, helps in speed and reducing fluctuations
- S_{det} and corresponding force can be evaluated exactly, i.e., without introducing pseudofermions
  - leads to a speed-up of ~ 2

Hasenbusch mass-preconditioning

Frequency splitting *[Hasenbusch 01, Hasenbusch, Jansen 03]*

$$\det(Q^2) = \det(Q^2 + \mu^2) \prod_{k=0}^{n-1} \det\left(\frac{Q^2 + \mu_k^2}{Q^2 + \mu^2{}_{k+1}}\right) \qquad 0 = \mu_0 < \ldots < \mu_n$$

Actions and forces

$$S_{pf,n} = (\phi_n, (Q^2 + \mu_n^2)^{-1}\phi_n)$$

$$S_{pf,k} = (\phi_k, (Q^2 + \mu^2{}_{k+1})(Q^2 + \mu^2{}_k)^{-1}\phi_k) \qquad k = 0, \ldots, n-1$$

$$(F_{pf,k})^a(x,\mu) = -2(\mu^2{}_{k+1} - \mu_k^2)\mathrm{Re}(\chi_k, (\partial^a{}_{x,\mu}Q)\psi_k)$$

- A proper tuning of μ can lead to a significant improvement
- smaller $\mathrm{var}(||F_{pf}||) \Rightarrow$ smaller $\mathrm{var}(\Delta H) \Rightarrow$ larger $P_A \Rightarrow$ larger h
- for QCD, at the physical point,

$$\mu_1 \sim m_q; \quad \mu_{k+1} \sim 10\mu_k$$

Typically $||F_{pf,k+1}|| \gg ||F_{pf,k}||$ while opposite for their cost, multiple time-step integration is useful

Rational HMC

Single-quark determinante *[Kennedy et al. 98, Clark, Kennedy 03]*

$$\det(Q) \to \det(\sqrt{Q^2}) = \det(W)\ \det R^{-1} \qquad W = |Q|R$$

where $R = r_b R^{n,\epsilon}(r_b^{-2} Q^2)$, $\epsilon = (r_a/r_b)^2$ and $\lambda(|Q|) \in [r_a, r_b]$

Rational approximation

$$R^{n,\epsilon}(x) = A \frac{(x+a_1)\cdots(x+a_{2n-1})}{(x+a_2)\cdots(x+a_{2n})} \approx \frac{1}{\sqrt{x}} \qquad a_1 > \ldots > a_{2n} > 0$$

- Zolotarev rational function of degree (n,n)
- Guarantees smallest possible $\delta = max_{\epsilon \leqslant x \leqslant 1} |1 - \sqrt{x} R^{n,\epsilon}(x)|$

Frequency splitting (e.g. $n = 10$)

$$\det(R^{-1}) \propto \det({P^{-1}}_{1,4}) \det({P^{-1}}_{5,7}) \det({P^{-1}}_{8,10})$$

$$with \qquad P_{k,l} = \prod_{j=k}^{l} \frac{Q^2 + \nu_j^2}{Q^2 + \mu_j^2} = 1 + \sum_{j=k}^{l} \frac{\rho_j}{Q^2 + \mu_j^2} \qquad \mu_1 > \ldots > \mu_n$$

Rational HMC

Actions and forces

$$S_{pf,k,l} = (\phi_{k,l}, P_{k,l}\,\phi_{k,l}) \qquad (F_{pf,k,l})^a(x,\mu) = \partial^a{}_{x,\mu} S_{pf,k,l}$$

Remarks

- We have to ensure $det(Q) > 0$ this is ensured by measure the spectral range $[r_a, r_b]$ of Q^2

  If that fails, sign flip need to be included as a reweighting factor (on some CLS ensembles thats the case *[Mohler, Schaefer 19]*)

- Choose a large enough n to have a good approximation of R

- For Wilson quarks the RHMC is typically used for heavy quarks, (charm and strange), for staggered it required for taking 4th root

- $det(W)$ can be included in the accept-reject step or in the observable as a reweighting factor. It can be estimated stochastically

$$\langle O \rangle_{|Q|} = \frac{\langle OW \rangle_{R^{-1}}}{\langle W \rangle_{R^{-1}}} \qquad W = \langle e^{-\eta^\dagger [(1+Z)^{-1/2} - 1]\eta} \rangle_P$$

Critical slowing down

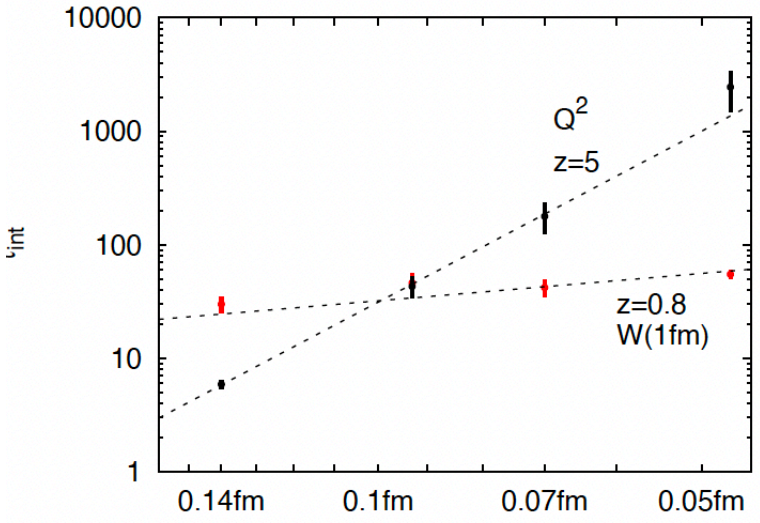
Approaching the continuum limit

$$\tau^{\mathrm{int,O}} \propto a^{-z}$$

where z depends on the algorithms

certain algorithms can be analyzed as QFTs :

- Simulation time is the (D+1)th dimension *[Parisi,Wu 81, Zinn-Justin 86]*
- HMC can not be analyzed this way: empirically $z = 2$
- this might be true if topology issue is absent



[Schaefer, Virota, et al. 2010]

Costs

Approaching the continuum limit at constant physics, the cost to obtain a set of statistical independent configurations scales like

$$C \propto V a^{-4} \qquad V^{1/8} a^{-1/2} \qquad a^{-z} \propto V^{9/8} a^{-4.5-z}$$

where the first term comes from the solver, the second from the integrator and the third from the autocorrelation

- Open boundary conditions: z=2, Periodic boundary conditions z=5
  - note that topological charge need to diffuse into lattice in case of OBC, leads to $\tau^{\mathrm{int}} \approx O(10^2)$ at $a \sim 0.05$ at physical point *[McGlynn,Mawhinney,2014]*

On a GPU machine with nodes each with 4*Mi250 costs are roughly given:

- O(1000) trajectories
- A physical volume of L=7.3 fm and a=0.057 fm
- Nf=2+1+1 twisted mass fermions at physical masses

$$C \approx 0.5 \text{ Mi node hours on Lumi-G}$$

Conclusions

Challenges are at the horizon and they waiting **to be solved** !!!

- topological freezing not sufficient solved (at least at physical pion masses)
- algorithmic challenges, e.g. chiral fermions like overlap fermions, very large lattices
- signal to noise and excited states
- .... and sampling with a sign

Looking forward to your contributions !!!

For HMC there are advance techniques:

- tuning of optimal solver methods, of kernels for GPU systems,...
- higher order integrators, force gradient integrators,..
- domain decomposition, multi-level sampling , master fields ...
- combination with flows ...

See for some current challenges (mentioned in some recent reviews) see:

- Peter Boyle *[Advances in algs for solvers and sim. https://arxiv.org/abs/2401.16620]*
- Gurtej Kanwar*[Flow-based sampling for LFT https://arxiv.org/abs/2401.01297]*
- J.F. *[Future trends in LQCD sim. https://cds.cern.ch/record/2928997]*
- J.F. *[Reviews on Algs for dyn. fermions. https://arxiv.org/abs/2402.11704]*

References

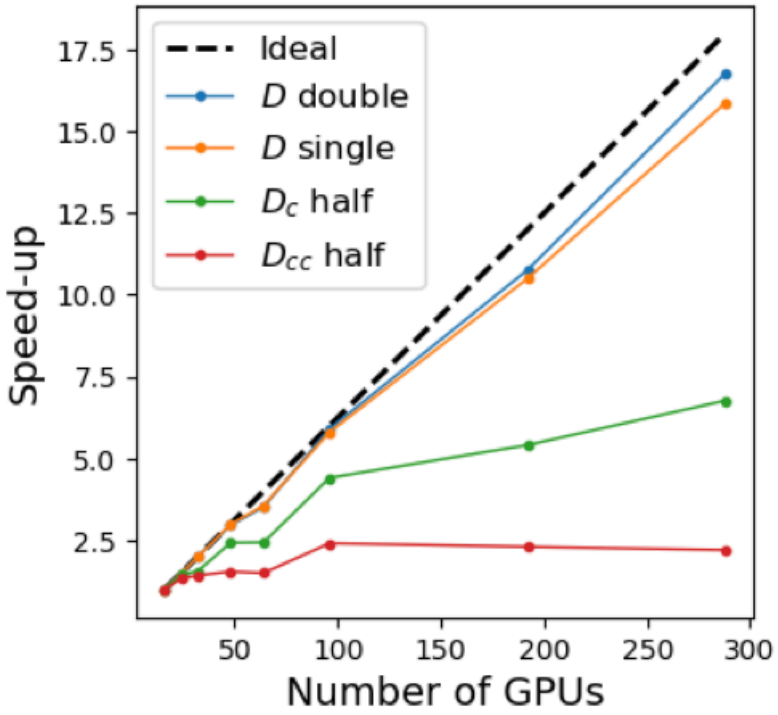
- Some part of the talk is based on Mattia Della Brida's Lattice Practice Contribution of 2021

Other references:

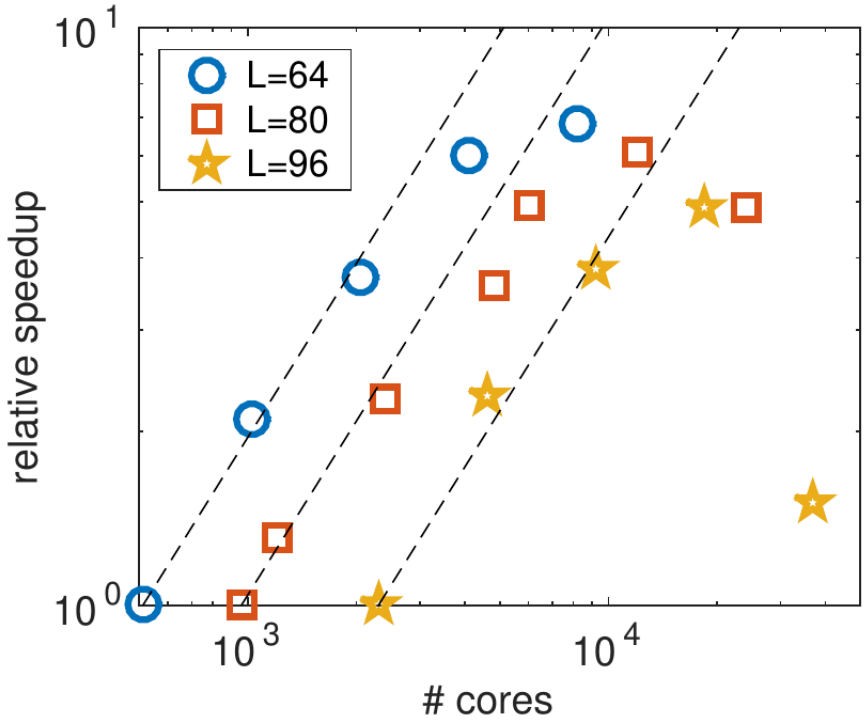
- M. Lüscher, Computational Strategies in Lattice QCD, in Les Houches Summer School: Session 93: Modern perspectives in lattice QCD: Quantum field theory and high performance computing, pp. 331–399, 2, 2010, 1002.4232.
- A. D. Kennedy, Algorithms for dynamical fermions, hep-lat/0607038.
- S. Schaefer, Simulations with the hybrid Monte Carlo algorithm: Implementation and data analysis, in Les Houches Summer School: Session 93: Modern perspectives in lattice QCD: Quantum eld theory and high performance computing, pp. 401–422, 8, 2009.
- S. Schaefer, Simulation Algorithms, Lattice Practices 2018
- S. Schaefer, Status and challenges of simulations with dynamical fermions, PoS LATTICE2012 (2012) 001, [1211.5069].
- S. Duane, A. D. Kennedy, B. J. Pendleton and D. Roweth, Hybrid Monte Carlo, Phys. Lett. B 195 (1987) 216–222.
- A. Barp, F.-X. Briol, A. D. Kennedy and M. Girolami, Geometry and Dynamics for Markov Chain Monte Carlo, 1705.0289
- ALPHA collaboration, U. Wolff , Monte Carlo errors with less errors, ComputPhys. Commun. 156 (2004) 143–153, [hep-lat/0306017].
- ALPHA collaboration, S. Schaefer, R. Sommer and F. Virotta, Critical slowingdown and error analysis in lattice QCD simulations, Nucl. Phys. B 845 (2011) 93–119, [1009.5228].

Scaling of multi-grid

Scaling of operators within the muli-grid procedure (QUDA, Juwels Booster)

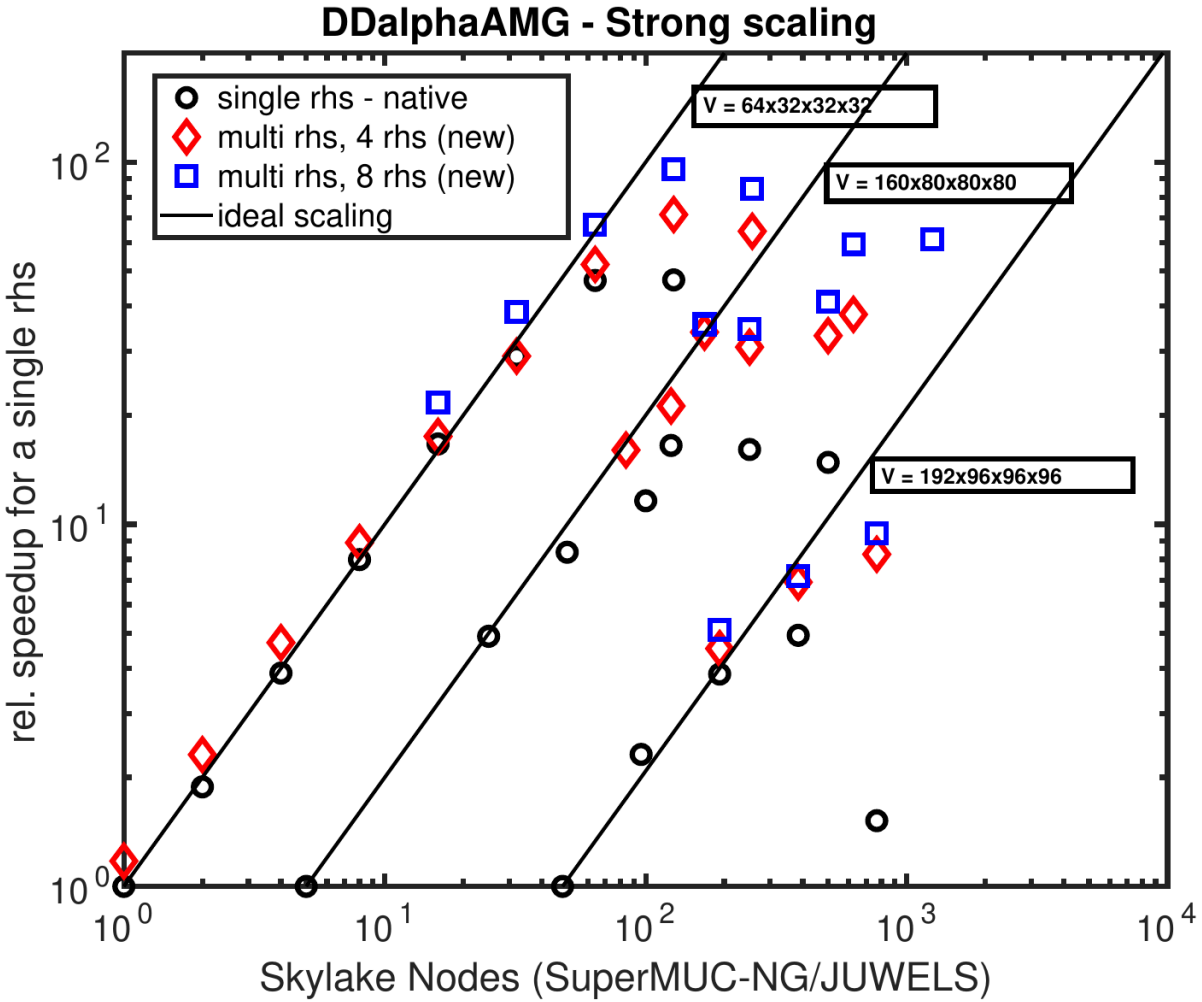


Scaling of the muli-grid solver (DDalphaAMG, SuperMUC-NG)



Remarks

Improve scaling via multi-rhs



Remarks

Parameter example for ETMC cB48 - lattice for QUDA

```
./quda/test/invert_test \
--compute-clover 1 --verbosity verbose --epsilon 0.0 --solve-type direct-pc --dslash-type twisted-
--dim ${XDIM} ${YDIM} ${ZDIM} ${TDIM} --gridsize ${XGRID} ${YGRID} ${ZGRID} ${TGRID} \
--load-gauge conf.1000 --prec double --prec-sloppy single --prec-precondition half \
--prec-null half --recon 12 --recon-sloppy 8 --recon-precondition 8 \
--kappa 0.1394265 --mu 0.00072 --clover-csw 1.69 \
--mass-normalization mass --pipeline 24 --ngcrkrylov 24 \
--niter 1000 --tol 1.0e-9 --inv-multigrid true --inv-type gcr --nsrc 120 \
--mg-levels 3 --mg-mu-factor 2 1.0 \ # 3 level multigrid - fine (0) + intermediate (1) + coarsest
--mg-setup-tol 0 5e-7 --mg-setup-tol 1 5e-7 --mg-setup-maxiter 0 1000 \
--mg-setup-maxiter 1 1000 --mg-setup-inv 0 cg --mg-setup-inv 1 cg \
--mg-setup-iters 0 1 --mg-pre-orth false --mg-post-orth true \
--mg-block-size 0 4 4 4 4 --mg-block-size 1 2 2 2 2 \ # blocking from 0 → 1 and 1 → 2
--mg-nu-pre 0 0 --mg-nu-pre 1 0 --mg-nu-post 0 4 --mg-nu-post 1 2 \
--mg-verbosity 0 silent --mg-verbosity 1 silent --mg-verbosity 2 silent \
--mg-verbosity 3 silent --mg-omega 0.85 --mg-smoother 0 ca-gcr --mg-smoother 1 ca-gcr \
--mg-nvec 0 24 --mg-nvec 1 32 --mg-smoother-tol 0 0.22 --mg-smoother-tol 1 0.46 \
--mg-schwarz-cycle 0 1 --mg-schwarz-cycle 1 1 --mg-coarse-solver 2 ca-gcr --mg-coarse-solver-ca-ba
--mg-coarse-solver-tol 1 0.22 --mg-coarse-solver-tol 2 0.46 --mg-coarse-solver-maxiter 2 50 \
--mg-eig 2 true --mg-eig-type 2 trlm \ #exact deflation on coarse level
--mg-eig-n-ev 2 1000 --mg-nvec 2 1000 \
--mg-eig-require-convergence 2 false --mg-eig-n-kr 2 1500 \
--mg-eig-tol 2 1e-4 --mg-eig-poly-deg 2 100 --mg-eig-amin 2 6e-2 \
--mg-eig-amax 2 8.0 --mg-eig-max-restarts 2 8 \
--mg-eig-use-dagger 2 false --mg-eig-use-normop 2 true \
--mg-eig-preserve-deflation true
```