Algorithms for Lattice QCD III

Stefan Schaefer

NIC, DESY

Lattice Gauge Theory School

Introduction

1 General idea of Markov Chain Monte Carlo

HMC algorithm $\rightarrow$ classical equations of motion

2 Integrators and Fermions

$\mathbf{det} \rightarrow$ pseudofermions
$\rightarrow$ Need solution of Dirac equation

Hasenbusch factorization

Other fermion methods

3 Methods to solve the Dirac equation

Factorizations

Hasenbusch Hasenbusch'03

$$\det Q^2 = \det \frac{Q^2}{Q^2+\mu_1^2} \det \frac{Q^2+\mu_1^2}{Q^2+\mu_2^2} \cdots \det(Q^2+\mu_N^2)$$

RHMC Kennedy, Horvath, Sint'99, Clark, Kennedy'07

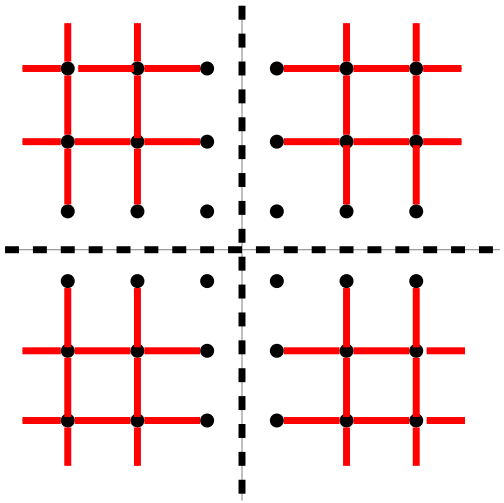
$$\det Q^2 = \prod_{i=1}^{N} \det \sqrt[N]{Q^2}$$

Domain decomposition Lüscher '04

$$\det Q = \det Q_{\text{block}} \det R$$

Domain decomposition

Lüscher'04



- Domain decomposition
  $\rightarrow$ Divide the lattice in blocks

$$\det D = \det D_{\text{block}} \cdot \det D_{\text{R}}$$

- Do not update links connecting blocks
  $\rightarrow$ longer autocorrelations
- Good for slow communication.

RHMC

To simulate non-degenerate flavors

$$\det D = \det(1_e + D_{oo}) \det \hat{D} = \det(1_e + D_{oo}) \cdot W \cdot \det R^{-1}$$

R: rational approximation to $(\hat{D}^\dagger \hat{D})^{-1/2}$

$$W = \det(\hat{D} R)$$

Rational Zolotarov approximation

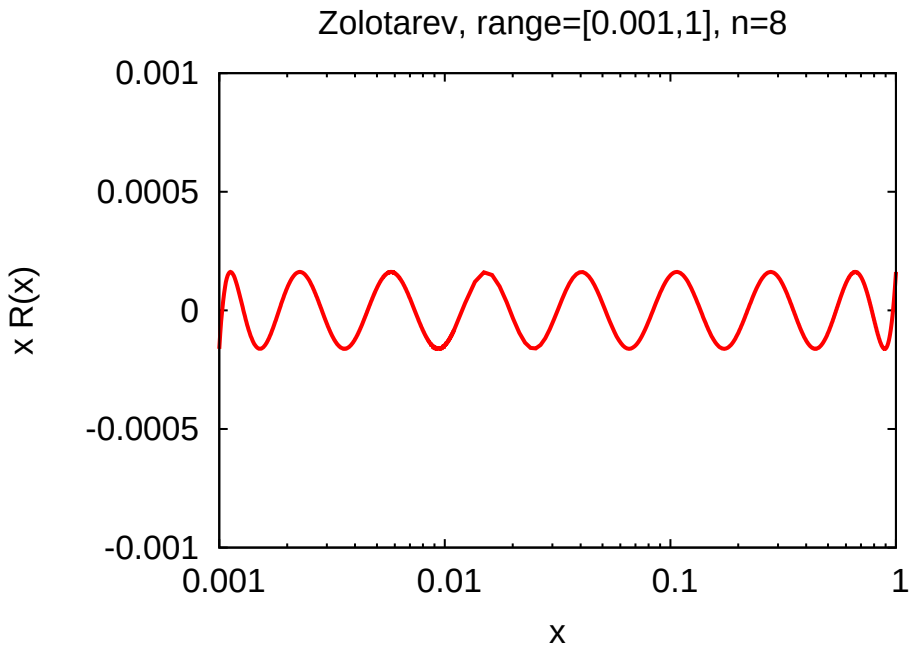
$$R_{n,\epsilon}(y) = A \frac{(y + a_1)(y + a_3) \dots (y + a_{2n-1})}{(y + a_2)(y + a_4) \dots (y + a_{2n})}$$

valid in range $\epsilon \leq y \leq 1$

Simulate with $R = 1 \rightarrow$ include R by reweighting

Zolotarev rational approximation

$$R_{n,\epsilon}(y) = A\frac{(y+a_1)(y+a_3)\dots(y+a_{2n-1})}{(y+a_2)(y+a_4)\dots(y+a_{2n})}$$



Shifts $a_i > 0$ can be quite small, in example $\sim 10^{-6}$.
Small shifts have small contributions.

RHMC II

Split approximation in several factors

$$R_{n,\epsilon}(y) = A\frac{(y+a_1)(y+a_3)\dots(y+a_{2n-1})}{(y+a_2)(y+a_4)\dots(y+a_{2n})}$$
$$= A\frac{(y+a_1)(y+a_3)}{(y+a_2)(y+a_4)}\frac{(y+a_5)\dots(y+a_{2n-1})}{(y+a_5)\dots(y+a_{2n})}$$

Each factor on a single pseudo-fermion, e.g.,

$$S_{\mathrm{RHMC}} = S_1 + S_2 = \phi_1^\dagger \frac{(y+a_2)(y+a_4)}{(y+a_1)(y+a_3)}\phi_1 + \phi_2^\dagger \frac{(y+a_6)(y+a_8)}{(y+a_5)(y+a_7)}\phi_2$$

Why?
Can use different solvers for different PF.
Separate expensive but small contributions
$\rightarrow$ larger time scale

Twisted mass reweighting

Wilson fermions: Dirac operator is not protected from zero eigenvalues.

$$S_f = -\mathrm{tr}\log Q^2$$

$\rightarrow$ action can become infinite
$\rightarrow$ field space separated in sectors

Consequences

- Thermalization difficult
- Problems with ergodicity
- Difficulties with numerical integration (spikes in forces)

Twisted mass reweighting

Solution:

- Run with protected Dirac operator
- Correct in the measurement

Reweighting (Ferrenberg & Swendsen '82)

$$\begin{aligned}
\langle O \rangle &= \frac{1}{Z} \int \prod_{i,\mu} dU_{i,\mu} e^{-S_g - S_{f,\text{eff}}} O[U] \\
&= \frac{Z'}{Z} \times \frac{1}{Z'} \int \prod_{i,\mu} dU_{i,\mu} e^{-S_g - S'_{f,\text{eff}}} \left(O[U] e^{-(S_{f,\text{eff}} - S'_{f,\text{eff}})} \right) \\
&= \frac{\langle O e^{-(S_{f,\text{eff}} - S'_{f,\text{eff}})} \rangle'}{\langle e^{-(S_{f,\text{eff}} - S'_{f,\text{eff}})} \rangle'}
\end{aligned}$$

Twisted mass reweighting

Want a reweighting factor with little fluctuation

Proposal by Lüscher & Palombi

$$\det Q^2 \to \begin{cases} \det\left(Q^2+\mu^2\right) \\ \det \frac{(Q^2+\mu^2)^2}{Q^2+2\mu^2} \end{cases}$$

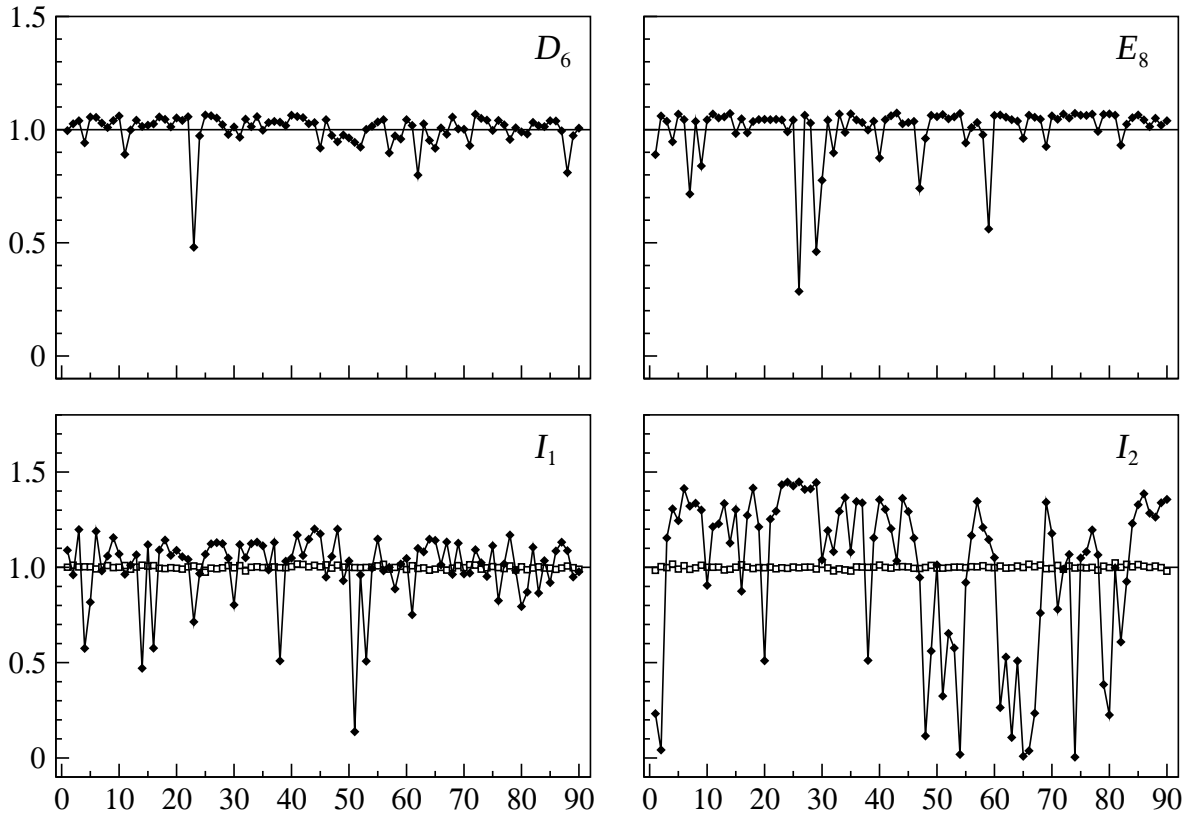
Reweighting factor

$$\det X^{-1} \to \begin{cases} \det\left(1+\frac{\mu^2}{Q^2}\right) & = \prod_\lambda \left(1+\frac{\mu^2}{\lambda^2}\right) \\ \det\left(1+\frac{\mu^2}{Q^2}\frac{(Q^2+\mu^2)}{Q^2+2\mu^2}\right) & = \prod_\lambda \left(1+\frac{\mu^4}{\lambda^4}\right) \end{cases}$$

Second term less sensitive to UV contribution.

Example

Lüscher, Schaefer'12



Summary

Fermion action $-\mathrm{tr}\log D$ cannot be simulated directly.

Use pseudofermions *together* with matrix factorization

Several factorizations lead to working setups.

Need of solving the Dirac equation in each force evaluation.

Solving the Dirac equation

The solution of the Dirac equation

$$\boldsymbol{D}\,\psi = \phi$$

is the most costly part of lattice simulations including dynamical fermions.

Dirac operator $\boldsymbol{D}$ can be viewed as a matrix acting in $\mathbb{C}^{12\boldsymbol{V}}$

For Wilson, staggered and domain wall fermions this matrix is sparse.

$\rightarrow$

Application of $\boldsymbol{D}$ on vector scales $\propto \boldsymbol{V}$.

Krylov space

Cayley-Hamilton theorem

The Krylov space $\mathcal{K}_n$ of order n generated by a starting vector ϕ and a matrix D is

$$\mathcal{K}_n(D, \phi) = \mathbf{span}\{\phi, D\phi, \ldots, D^{n-1}\phi\}$$

Cayley-Hamilton theorem

The value of a function (also the inverse) of an $N \times N$ matrix can be constructed from its powers up to N.

$\rightarrow$ Solution of Dirac equation is in $\mathcal{K}_N$.

This theorem is not of much practical help if N is in the millions.

Krylov space solvers

Use the Krylov space to iteratively construct the solution of the Dirac equation to a specified precision.

$$\boldsymbol{D}\,\psi = \phi$$

The basic problem is that it is not practical to safe the Krylov space.

- too much memory would be needed
- Only up to $\mathcal{O}(20)$ vectors can be stored

Different algorithms make different choices on how may vectors to compute before restarting.

$$\mathcal{K}_n(\boldsymbol{D}, \phi) \to \rho_0 \to \mathcal{K}_n(\boldsymbol{D}, \rho_0) \to \rho_1 \to \cdots$$

Iterative improvement

$$\boldsymbol{D}\,\psi = \phi$$

For a certain approximation $\tilde{\psi}$ you can define the **residue**

$$\rho = \phi - \boldsymbol{D}\tilde{\psi}$$

this ρ can be used as a new right hand side

$$\boldsymbol{D}\psi_1 = \rho \quad \rightarrow \psi = \tilde{\psi} + \psi_1$$

Used in

- Restarting iterative solvers
- Single precision acceleration
- Chronologically predicted solutions

Convergence criteria and accuracy

$$\boldsymbol{D}\,\psi = \phi$$

The convergence of the algorithm will be tested using

$$|\rho| = |\phi - \boldsymbol{D}\tilde{\psi}| < \epsilon|\phi|$$

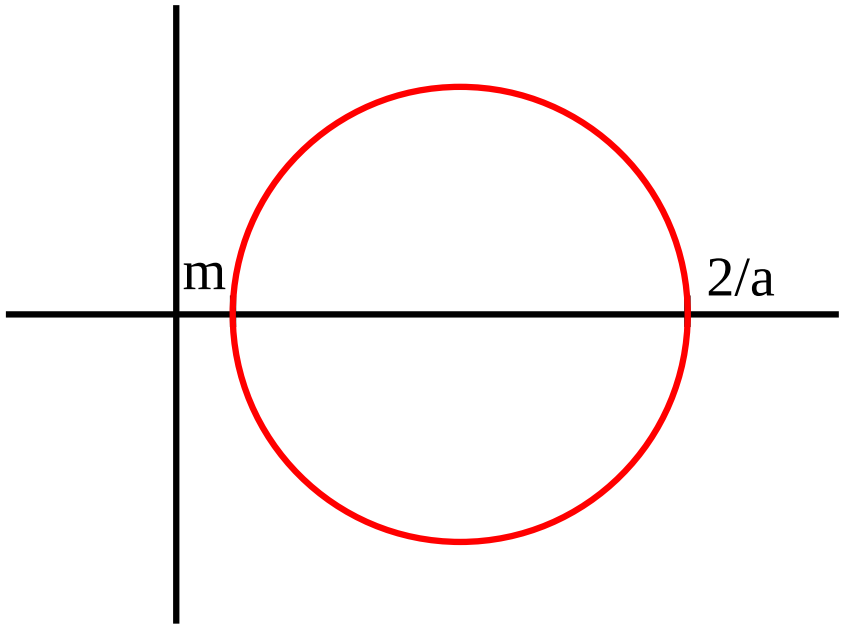
This deviates from the exact solution by at most

$$|\tilde{\psi} - \psi| < \epsilon\kappa(\boldsymbol{D})|\psi|$$

with $\kappa(D)$ the **condition number**

$$\kappa(\boldsymbol{D}) = |\boldsymbol{D}||\boldsymbol{D}^{-1}|$$

Condition number



For λ_{min}, λ_{max} the smallest/largest EV of $D^\dagger D$

$$\lambda_{\text{min}} \propto m$$
$$\lambda_{\text{min}} \propto 1/a$$

Condition number $\kappa(D) \propto (am)^{-1}$.

The GCR algorithm

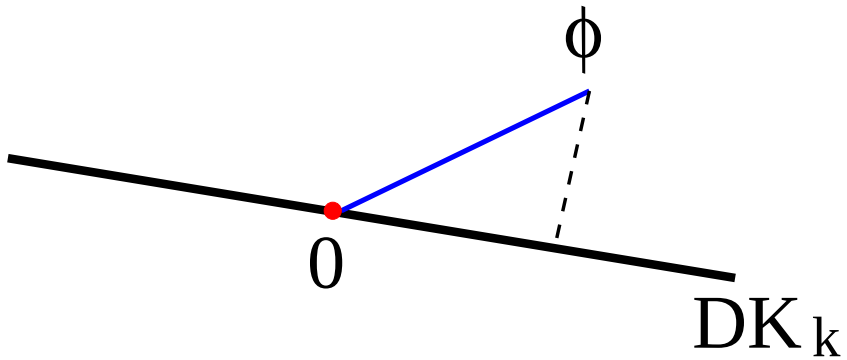
The Generalized Conjugate Residue algorithm constructs for each k the solution of the Dirac equation

$$\psi_1, \psi_2, \psi_3, \ldots \psi_k, \ldots$$

that minimizes the residue

$$\rho_k = \phi - D\,\psi_k$$

This minimum is attained where $D\,\psi_k$ is the orthogonal projection of ϕ to $D\mathcal{K}_k$.



The GCR algorithm

This orthogonal projection is simplified computing an orthonormalized basis χ_i

$$D\mathcal{K}_k = \mathbf{span}\{\chi_i\}.$$

Then the computation of the new residue —the orthogonal projection— is just

$$\rho_k = \eta - \sum_{l=0}^{k-1} c_l \, \chi_l \qquad \text{with} \qquad c_l = (\chi_l, \eta)$$

The GCR algorithm

At some point in the algorithm one thus has
The orthonormal basis of $D\mathcal{K}_{k-1}$ and the residues

$$\{\chi_0, \chi_1, \ldots, \chi_{k-1}\} \qquad \{\rho_0, \rho_1, \ldots, \rho_{k-1}\}$$

one constructs the current residue

$$\rho_k = \eta - \sum_{l=0}^{k-1} c_l \, \chi_l \qquad \text{with} \qquad c_l = (\chi_l, \eta)$$

and adds to the basis χ_k, the contibution of $D\rho_k$.

Compute the representation of the χ in terms of the $D\rho$.

$$\chi_k = \sum_{j=0}^{k} a_{kj} D\rho_j \qquad \text{with } \rho_0 = \phi$$

Computing the current solution

$$\chi_k = \sum_{j=0}^{k} a_{kj} D\rho_j$$

and

$$\rho_k = \eta - \sum_{l=0}^{k-1} c_l \, \chi_l \qquad \text{with} \qquad c_l = (\chi_l, \eta)$$

Putting everything together, we have

$$D\psi_k = \sum_{l=0}^{k-1} c_l \, \chi_l = \sum_{l=0}^{k-1} c_l \sum_{j=0}^{l} a_{lj} D\rho_j$$

Now we can divide by D and get the solution ψ_k in terms of the already computed residues.

$$\psi_k = \sum_{l=0}^{k-1} c_l \sum_{j=0}^{l} a_{lj} \rho_j$$

Convergence rate

The solution is constructed in the Krylov space
$\rightarrow$ the residue is a polynomial of D times source.

$$\rho_k = p_k(D)\phi$$

The GCR algorithm minimizes the residue

$$|\rho_k| = \mathbf{min}_p |p_k(D)\phi| \leq \mathbf{min}_p |p_k(D)||\phi|$$

Take diagonaliziable $D \rightarrow D = V\Lambda V^{-1}$, with Λ diagonal

$$|p_k(D)| = |V p_k(\Lambda) V^{-1}| \leq \kappa(V)|p_k(\Lambda)|$$

This leads to the inequality

$$|\rho_k| \leq \kappa(V) \mathbf{max}_{\lambda \in \mathbb{D}} |p_k(\lambda)||\phi|$$

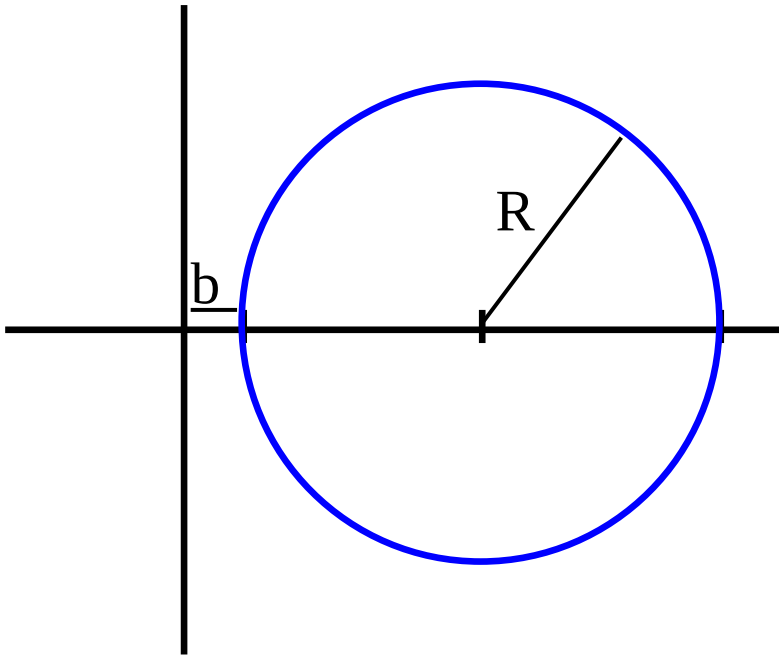
Assumption is that spectrum is contained in a disk $\mathbb{D}$ right of origin.

Convergence rate

$$|\rho_k| \leq \kappa(V)\mathbf{max}_{\lambda \in \mathbb{D}}|p_k(\lambda)||\phi|$$

Theory of polynomial approximations:
The optimal polynomial, for which the maximum is attained is

$$p_k(\lambda) = (1 - \frac{\lambda}{R+b})^k$$



R is radius of the disk,
b the distance from origin.

$$|\rho_k| \leq \kappa(V)(1 - \frac{b}{R})^{-k}|\phi|$$

For $b \ll R$ this decays roughly exponentially $e^{-k\frac{b}{R}}$

GCR: Summary

Above, one iteration of the GCR has been described.

It requires to order k

- $2k$ spinor fields of storage.
- k matrix vector products.

Typical is the computation 20–40 vectors.

Then perform a restart.

Preconditioning

At small quark masses, the condition number of D becomes large.

Many matrix-vector multiplications needed for solution.

Preconditioning

$$LDR\psi' = L\phi \qquad \rightarrow \psi = R^{-1}\psi'$$

with L and R chosen such that $L\,D\,R$ is well conditioned.

Many variants of preconditioning.

A good preconditioning will work on low-mode part of spectrum.

Even-odd preconditioning

If the Dirac operator only connects nearest-neighbors, the lattice can be split in "even" and "odd" sites.

$$D = \begin{pmatrix} D_{\mathrm{ee}} & D_{\mathrm{eo}} \\ D_{\mathrm{oe}} & D_{\mathrm{oo}} \end{pmatrix}$$

and we can use the matrices L and R

$$L = \begin{pmatrix} 1 & -D_{\mathrm{eo}} D_{\mathrm{oo}}^{-1} \\ 0 & 1 \end{pmatrix} \qquad R = \begin{pmatrix} 1 & 0 \\ -D_{\mathrm{oo}}^{-1} D_{\mathrm{oe}} & 1 \end{pmatrix}$$

to get

$$LDR = \begin{pmatrix} \hat{D} & 0 \\ 0 & D_{\mathrm{oo}} \end{pmatrix} \qquad \text{with} \qquad \hat{D} = D_{\mathrm{oo}} - D_{\mathrm{oe}} D_{\mathrm{ee}}^{-1} D_{\mathrm{eo}}$$

$\hat{D}$ is the Schur complement
$\hat{D}$ **has less than half the condition number of** D.

Preconditioning

Even-odd is the prime example of "classical" preconditioning.
It is used in all simulations with next-nearest operators only.

Another example is SSOR.
Complicated to implement $\rightarrow$ not used so frequently.

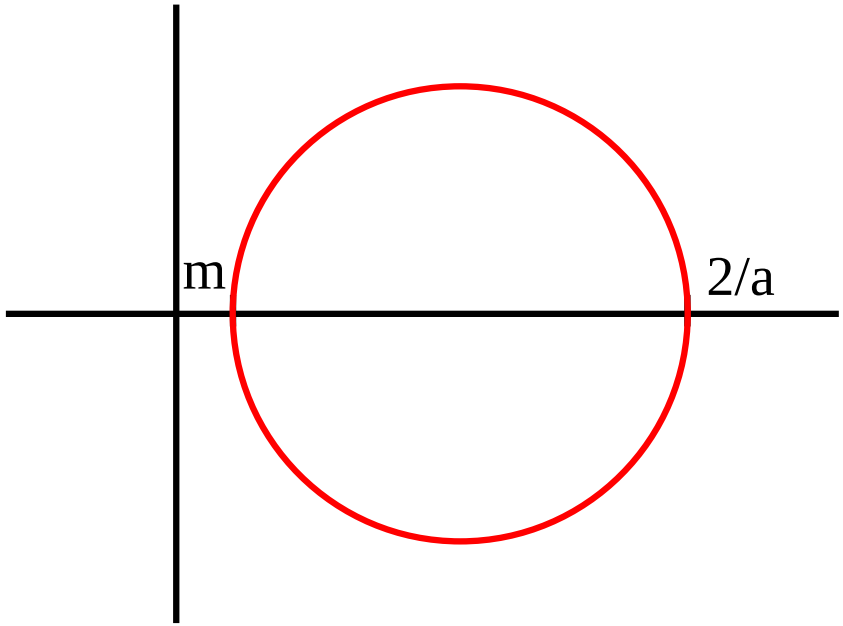
Their benefit is limited: they do not know about the physics.

For large quark masses, standard Krylov-space solvers + eo work fine.

Small quark masses: condition number $\propto am$.

Need to take the IR physics into consideration to get more significant speed-up.

Deflation



When quark mass gets small

$$\kappa \propto 1/(am)$$

due to small eigenvalues with $\mathrm{Re}\lambda \approx m$.

Exact deflation with eigenvectors

Elimitate these eigenmodes from the Dirac equation.

$$D\,\psi_i = \lambda_i\,\psi_i$$

Projector on small eigenmodes ψ_i

$$P = \sum_{i=1}^{N_s} \psi_i\,\psi_i^\dagger$$

Using it, we can split the Dirac equation in two

$$D = \begin{pmatrix} PDP & 0 \\ 0 & (1-P)D(1-P) \end{pmatrix}$$

$$\Rightarrow D^{-1} = \begin{pmatrix} \sum_{i=1}^{N_s} \frac{1}{\lambda_i}\psi_i\,\psi_i^\dagger & 0 \\ 0 & [(1-P)D(1-P)]^{-1} \end{pmatrix}$$

Discussion

$$D^{-1} = \begin{pmatrix} \sum_{i=1}^{N_s} \frac{1}{\lambda_i} \psi_i \, \psi_i^\dagger & 0 \\ 0 & [(1-P)D(1-P)]^{-1} \end{pmatrix}$$

Reduction of condition number $\kappa \to |\frac{\lambda_1}{\lambda_{N_s+1}}|\kappa$

Efficient if small number of very small eigenvalues
ϵ-regime calculations

Need to compute eigenvectors (can be set up with approximate vectors)

For constant effect need $N_s \propto V$ vectors.

In large volume, computation of a single eigenvector exceedingly expensive.

Summary

The solution to the Dirac equation is essential part of lattice computations with fermions.

Methods take advantage of the spasity of the Dirac matrix.
Matrix-Vector multiplications essential operation, is $O(V)$.

Cost can be high for small quark masses
$\rightarrow$ shrinking gap to origin
$\rightarrow$ problem with exceptionally small eigenvalues

Need to find a *physical* solution to problem.