Temporal Preconditioning for Wilson-like Fermions

R. Edwards[1] B. Joó[1] M. Peardon [2]

[1]Thomas Jefferson National Accelerator Facility
Newport News, VA, USA

[2]School of Mathematics
Trinity College, Dublin, Ireland

QCD and Numerical Analysis, 2007

Preconditioning in Lattice QCD

- In lattice QCD we solve large sparse linear systems involving the fermion matrix M
- During configuration generation we solve

$$M^{\dagger} M \, \phi = \chi$$

  - for the pseudofermionic action $\phi^{\dagger} \left(M^{\dagger} M \right)^{-1} \phi$
  - for the computation of the Molecular Dynamics (MD) force:

$$\phi^{\dagger} \left(M^{\dagger} M \right)^{-1} \left[\frac{\partial M^{\dagger}}{\partial U} M + M^{\dagger} \frac{\partial M}{\partial U} \right] \left(M^{\dagger} M \right)^{-1} \phi$$

- For post analysis (propagators,noisy estimators) we solve:

$$M \, \phi = \chi$$

Preconditioning in Lattice QCD

- Preconditioning is essential to reduce cost of solves AND
- Preconditioning also changes the simulation 'action' AND
- Preconditioning changes the MD fermion forces
  - Forces change because the action changes
  - Roughly:

$$F \propto \kappa \left(M^\dagger M\right)^\nu$$

    - can take larger MD steps avoiding the integrator instabilities
    - put fermionic term on a slower timescale
    - cf. Mike Clark's talk.

Example: Schur Style Even-Odd Preconditioning

- colour lattice sites as even and odd (red-black)
- Write M as $\begin{pmatrix} M_{ee} & M_{eo} \\ M_{oe} & M_{oo} \end{pmatrix}$
- Perform a Schur Decomposition:

$$\begin{pmatrix} M_{ee} & M_{eo} \\ M_{oe} & M_{oo} \end{pmatrix} = \overbrace{\begin{pmatrix} 1 & 0 \\ M_{oe}M_{ee}^{-1} & 1 \end{pmatrix}}^{L} \begin{pmatrix} M_{ee} & 0 \\ 0 & \tilde{M} \end{pmatrix} \overbrace{\begin{pmatrix} 1 & M_{ee}^{-1}M_{eo} \\ 0 & 1 \end{pmatrix}}^{U}$$

where

$$\tilde{M} = M_{oo} - M_{oe}M_{ee}^{-1}M_{eo}$$

- Note that: $\det L = \det U = 1$
- Inverses of L and U are trivial (flip sign of off diag. piece)
- M_{ee}^{-1} should be straightforward to apply.

Example: Propagators Computations

- Rewrite propagator system

$$\begin{aligned} M\,\phi &= \chi \\ \Rightarrow L \begin{pmatrix} M_{ee} & 0 \\ 0 & \tilde{M} \end{pmatrix} U\,\phi &= \chi \\ \Rightarrow \begin{pmatrix} M_{ee} & 0 \\ 0 & \tilde{M} \end{pmatrix} \phi' &= \chi' \end{aligned}$$

with $\phi' = U\phi$ and $\chi' = L^{-1}\chi$.
- The hard work solving $\tilde{M}\phi'_o = \chi'_o$ (since M^{-1} is easy)
- At the end $\phi = U^{-1}\phi'$

Example: Schur Even-Odd Preconditioning and HMC

- Want to simulate $\det\left(M^{\dagger}M\right)$.
- From the Schur Decomposition:

$$\det\left(M^{\dagger}M\right) = \det\left(M^{\dagger}_{ee}M_{ee}\right)\det\left(\tilde{M}^{\dagger}\tilde{M}\right)$$

- Can rewrite our action as:

$$\exp\left\{-\phi^{\dagger}\left(M^{\dagger}M\right)\phi\right\} \Rightarrow \exp\left\{\log\det\left(M^{\dagger}_{ee}M_{ee}\right) - \phi'^{\dagger}\left(\tilde{M}^{\dagger}\tilde{M}\right)^{-1}\phi'\right\}$$

- Now try to take advantage of knowledge of M_{ee}:
  - M_{ee} is independent of gauge fields $\Rightarrow$ drop altogether (Wilson Fermions, Domain Wall Fermions)
  - Compute $\log\det\left(M^{\dagger}_{ee}M_{ee}\right)$ directly (Clover Fermions)
- Key Point: Preconditioning modifies simulation action

Example: Schur Even-Odd Preconditioning and HMC

- Two new force terms in MD
  - From $\exp\left\{\log\det\left(M_{ee}^{\dagger}M_{ee}\right)\right\}$
  - From $\exp\left\{-\phi'^{\dagger}\left(\tilde{M}^{\dagger}\tilde{M}\right)^{-1}\phi'\right\}$
- New pseudofermionic force involves $\tilde{M}$ rather than M.
- $\tilde{M}$ has better condition than M
  - we get smaller forces
  - further from integrator step size instabilities
  - Can take bigger steps in MD at same overall cost
  - Larger step-size integrators become useful
  - Fewer inversions for fixed MD trajectory length.
  - All the benefits Mike discussed

Previously Successful Preconditionings

- Even Odd (previous example), Lexicographic SSOR
- Domain Decomposition Combined with HMC (Lüscher)
- Hasenbusch Mass Preconditioning (Hasenbusch et. al)
  - Simulate

$$\frac{\det\left(M_1^\dagger M_1\right)}{\det\left(M_2^\dagger M_2\right)} \det\left(M_2^\dagger M_2\right)$$

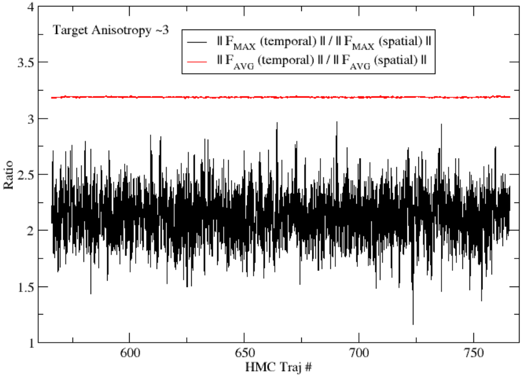
  - Choose $M_2 = M_1 + \delta$
  - M_2 is better conditioned, Ratio is close to $1 + O(\delta)$
- Nth-rootery / Multipseudofermions (Clark et. al):

$$\det\left(M^\dagger M\right) = \left[\det\left(M^\dagger M\right)^{\frac{1}{N}}\right]^N \Rightarrow \prod_{i=1}^{N} e^{\left\{-\phi_i^\dagger \left(M^\dagger M\right)^{-\frac{1}{N}} \phi_i\right\}}$$

  - Now have N terms each with condition number $\kappa^{\frac{1}{N}}$.
  - Win if $N \kappa^{\frac{1}{N}} < \kappa$

Anisotropic Lattices

- Ideal world
  - Want fine lattice spacing (close to continuum)
- Real World:
  - Fine lattice too costly, do as coarse as possible
- Compromise: Make just one dimension (time) fine
  - 2 lattice spacings: a_s (spatial) and a_t (temporal)
  - Typical choice: $a_t << a_s$
  - Important physics applications (eg: spectroscopy)
- Ramifications:
  - Lowest modes of fermion matrix result from fine a_t
  - Largest forces from a_t



Ratios of spatial and Temporal forces in Anisotropic RHMC for $\xi \approx 3$

Motivation for Temporal Preconditioning

- Basic Idea
  - Write $M = M_s + M_t = M_t \left(M_t^{-1} M_s + 1 \right)$
  - The preconditioned matrix is $\tilde{M} = 1 + M_t^{-1} M_s$
  - Deal separately with $\det(M_t)$ in HMC
- Expect
  - To still have even-odd preconditioning spatial dimensions
  - To gain an improvement in condition number $\approx$ anisotropy
  - To gain a reduction in temporal pseudofermion force in HMC

The Wilson Fermion Operator

- Unpreconditioned Wilson Fermion Operator ($r = 1$):

$$
\begin{aligned}
M &= D_s + D_t \\
D_s &= -\sum_{k=1}^{3} P^k_- U_k(x)\delta_{x+\hat{k},y} + P^k_+ U^\dagger_k(x-\hat{k})\delta_{x-\hat{k},y} \\
D_t &= \hat{m} - P_- \tilde{U}_t(x)\delta_{x+\hat{t},y} - P_+ \tilde{U}^\dagger_t(x-\hat{t})\delta_{x-\hat{t},y}
\end{aligned}
$$

with

$$
\begin{aligned}
P^k_\pm &= (1/2)\,(1 \pm \gamma_k) \qquad k = 1,2,3 \\
P_\pm &= (1/2)\,(1 \pm \gamma_4) \\
\tilde{U}(x) &= \frac{\nu}{\xi_0} U(x), \qquad U \in SU(3) \\
\hat{m} &= 1 + (N_d - 1)\frac{\nu}{\xi_0} + M
\end{aligned}
$$

Central Temporal Preconditioning

- Define Matrices:

$$\begin{aligned} T(\vec{x})_{t,t'} &= \hat{m} - \tilde{U}_t(\vec{x},t)\delta_{t+1,t'} \text{ with periodic boundaries in time} \\ C_L^{-1} &= P_+ + P_- T \\ C_R^{-1} &= P_- + P_+ T^\dagger \end{aligned}$$

- Then we have (playing Projector games): $C_L^{-1} C_R^{-1} = D_t$
- Precondition as:

$$\tilde{M} = C_L \; M \; C_R = C_L \; D_s \; C_R + 1$$

- We retain a kind of γ_5 hermiticity:

$$\gamma_5 C_L^{-1} \gamma_5 = \left(C_R^{-1} \right)^\dagger \quad \gamma_5 C_R^{-1} \gamma_5 = \left(C_L^{-1} \right)^\dagger, \quad \gamma_5 \tilde{M} \gamma_5 = \tilde{M}^\dagger$$

Inverting the Preconditioning Matrices

The Sherman Morrison Woodbury Formula

- Consider C_L only (C_R proceeds similarly)

$$C_L^{-1} = P_+ + P_- T \Rightarrow C_L = P_+ + P_- T^{-1}$$

with

$$T = \begin{pmatrix} \hat{m} & -U_t(\vec{x},0) & 0 & \dots & \\ 0 & \hat{m} & -U_t(\vec{x},1) & 0 & \dots \\ \vdots & 0 & \ddots & \ddots & \\ 0 & \dots & 0 & \hat{m} & -U_t(\vec{x},N_t-2) \\ -U_t(\vec{x},N_t-1) & 0 & \dots & 0 & \hat{m} \end{pmatrix}$$

- Write T as $T = T_0 + V\, W^T$ with

$$T_0 = \begin{pmatrix} \hat{m} & -U_t(\vec{x},0) & 0 & \dots & \\ 0 & \hat{m} & -U_t(\vec{x},1) & 0 & \dots \\ \vdots & 0 & \ddots & \ddots & \\ 0 & \dots & 0 & \hat{m} & -U_t(\vec{x},N_t-2) \\ 0 & 0 & \dots & 0 & \hat{m} \end{pmatrix}$$

$$V = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ -U_t(\vec{x},N_t-1) \end{pmatrix} \quad W = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix}$$

- Sherman Morrison Woodbury Formula:

$$T^{-1} = T_0^{-1} - P(1 + W^T P)^{-1} W^T T_0^{-1} \quad \text{with } P = T_0^{-1} V$$

- T_0^{-1} easy to apply with back substitution

- We can compute $P = T_0^{-1} V$ by solving $TP = V$

$$\begin{aligned} P_{N_t-1} &= -\frac{1}{\hat{m}}\, U_t(N_t-1) \\ P_{N_t-2} &= -\frac{1}{\hat{m}^2}\, U_t(N_t-2)U_t(N_t-1) \\ P_i &= -\frac{1}{\hat{m}^{N_t-i}} \prod_{j=N_t-i}^{N_t-1} U_t(j) \\ P_0 &= -\frac{1}{\hat{m}^{N_t}} \prod_{j=0}^{N_t-1} U_t(j) \end{aligned}$$

- We define

$$Q = (1 + W^T P)^{-1} = (1 + P_0)^{-1}$$

and

$$T^{-1} = (1 - PQW^T)\, T_0^{-1}$$

Comments

- Computing P takes N_t $SU(3)$ multiplications per *spatial coordinate* $\vec{x}$ (or 1 SU(3) multiplication per site)
- P_0 is essentially just the Polyakov Loop
- Computing Q takes 1 3x3 complex matrix inversion per *spatial coordinate*. We use LU decomposition.
- Life is made easy if all temporal sites for a spatial coordinate x are kept 'local' to a processor

HMC Considerations

The determinant to simulate

- Determinant of interest is:

$$\det\left(M^\dagger M\right) = \det\left[\left(C_R^{-1}\right)^\dagger C_R^{-1}\right] \times \det\left[\left(C_L^{-1}\right)^\dagger C_L^{-1}\right] \times \det\left[\tilde{M}^\dagger \tilde{M}\right]$$

- Using the γ_5 hermiticity of C_L^{-1} and C_R^{-1}:

$$\begin{aligned}\det\left(M^\dagger M\right) &= \left[\det\left(C_R^{-1}\right)\right]^2 \times \left[\det\left(C_L^{-1}\right)\right]^2 \times \det\left(\tilde{M}^\dagger \tilde{M}\right) \\ &= e^{2\log\det\left(C_R^{-1}\right)} e^{2\log\det\left(C_L^{-1}\right)} \int d\phi^\dagger d\phi \; e^{-\phi^\dagger\left(\tilde{M}^\dagger\tilde{M}\right)^{-1}\phi}\end{aligned}$$

HMC Considerations

$\det\left(C_L^{-1}\right)$ and $\det\left(C_R^{-1}\right)$

- In Dirac Basis:

$$P_+ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad P_- = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and so

$$\begin{aligned} \det\left(C_L^{-1}\right) &= \det\left(P_+ + P_- T\right) = \det\left(T\right)^2 \\ \det\left(C_R^{-1}\right) &= \det\left(P_- + P_- T^\dagger\right) = \det\left(T^\dagger\right)^2 \end{aligned}$$

HMC Considerations

det (T)

- Finally:

$$
\begin{aligned}
T &= T_0 + VW^T \\
&= T_0 \left(1 + T_0^{-1} VW^T\right) \\
&= T_0 \left(1 + PW^T\right)
\end{aligned}
$$

and

$$
1 + PW^T = \begin{pmatrix} 1 + P_0 & 0 & \dots & 0 \\ P_1 & 1 & 0 & \dots \\ \vdots & 0 & \ddots & 0 \\ P_{N_t - 1} & 0 & \dots & 1 \end{pmatrix}
$$

so

$$
\det (T) = \det (T_0) \det (1 + P_0)
$$

- Recall that T_0 is upper diagonal with

$$\mathrm{diag}\left(T_0\right) = \mathrm{diag}\left(\hat{m}I_3, \hat{m}I_3, \ldots\right)$$

so

$$\det\left(T_0\right) = \hat{m}^{3N_t}$$

- We also have

$$1 + P_0 = 1 - \frac{1}{\hat{m}^{N_t}} \prod_{j=0}^{N_t-1} U_t(j)$$

So

$$\det\left(T\left(\vec{x}\right)\right) = \hat{m}^{3N_t} \det\left[1 - \frac{1}{\hat{m}^{N_t}} \prod_{j=0}^{N_t-1} U_t(\vec{x}, j)\right]$$

Clover Fermions

Improved Wilson Fermions

- Clover Fermions: Wilson Fermions $+$ and Improvement ("Clover") Term

$$M = D_s + D_t + A \quad \text{where} \quad A(x) = -\frac{c_{SW}\sigma_{\mu\nu}}{4} F_{\mu\nu}(x)$$

- The clover term A is local and Hermitian
- Precondition with same C_L and C_R as before

$$\begin{aligned} M &= C_L^{-1} C_R^{-1} + D_s + A \\ \tilde{M} &= C_L M C_R = [1 + C_L (D_s + A) C_R] \end{aligned}$$

Even–Odd Preconditioning in Space
Preliminaries

- Want even–odd preconditioning in space together with temporal preconditioning.
- Label sites as even an odd based on *spatial* coordinate $\vec{x}$:

$$-1^{x+y+z} = \begin{cases} +1 & \Rightarrow \quad \text{even} \\ -1 & \Rightarrow \quad \text{odd} \end{cases}$$

- D_t, C_L, C_R and T do not couple neighbours in $\vec{x}$
  - hence they are *diagonal* in even-odd space
- A is also diagonal in even-odd space
- D_s couples nearest neighbours in $\vec{x}$

The Operator in Even–Odd Space

- We write the clover operator as:

$$\tilde{M} = 1 + C_L(D_s + A)C_R = \begin{pmatrix} \tilde{M}_{ee} & \tilde{M}_{eo} \\ \tilde{M}_{oe} & \tilde{M}_{oo} \end{pmatrix} = \begin{pmatrix} 1 + C^e_L A^{ee} C^e_R & C^e_L D^{eo}_s C^o_R \\ C^o_L D^{oe}_s C^e_R & 1 + C^o_L A^{oo} C^o_R \end{pmatrix}$$

- Wilson operator simplifies since $A = 0$.
- We consider 2 spatial preconditionings
  - Schur decomposition based
  - Incomplete LU decomposition

Schur Decomposition

- We perform the Schur Decomposition:

$$\begin{aligned}
\tilde{M} &= L\,\mathcal{D}\,U \\
L &= \begin{pmatrix} 1 & 0 \\ C^o_L D^{oe}_s C^e_R \left(1 + C^e_L A^{ee} C^e_R\right)^{-1} & 1 \end{pmatrix} \\
U &= \begin{pmatrix} 1 & \left(1 + C^e_L A^{ee} C^e_R\right)^{-1} C^e_L D^{eo}_s C^o_R \\ 0 & 1 \end{pmatrix} \\
\mathcal{D} &= \begin{pmatrix} 1 + C^e_L A^{ee} C^e_R & 0 \\ 0 & 1 + C^o_L A^{oo} C^o_R - C^o_L D^{oe}_s C^e_R \left(1 + C^e_L A^{ee} C^e_R\right)^{-1} C^e_L D^{eo}_s C^o_R \end{pmatrix}
\end{aligned}$$

- Note the term:

$$1 + C_L A C_R = C_L (D_t + A) C_R$$

- We rewrite with $C_L(D_t + A)C_R$

$$
\begin{aligned}
L &= \begin{pmatrix} 1 & 0 \\ C_L^o D_s^{oe} (D_t + A)_{ee}^{-1} \, C_L^{-1} & 1 \end{pmatrix} \\
U &= \begin{pmatrix} 1 & \left(C_R^e\right)^{-1} (D_t + A)_{ee}^{-1} D_s^{eo} C_R^o \\ 0 & 1 \end{pmatrix} \\
\mathcal{D} &= \begin{pmatrix} C_L^e \, (D_t + A)_{ee} \, C_R^e & 0 \\ 0 & C_L^o \, (D_t + A)_{oo} \, C_R^o - C_L^o D_s^{oe} (D_t + A)_{ee}^{-1} D_s^{eo} C_R^o \end{pmatrix}
\end{aligned}
$$

- The matrix $D_t + A$ is:

$$\begin{pmatrix} \hat{m} + A(0) & -U(0)P_- & 0 & \dots & -U^\dagger(N_t-1)P_+ \\ -U^\dagger(0)P_+ & \hat{m}+A(1) & -U(1)P_- & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & -U^\dagger(N_t-3)P_+ & \hat{m}+A(N_t-2) & -U(N_t-2)P_- \\ -U(N_t-1)P_- & 0 & \vdots & -U^\dagger(N_t-2)P_+ & \hat{m}+A(N_t-1) \end{pmatrix}$$

- Now the $P_\pm$ enter giving the matrix *spin structure*
- Dimension is increased by a factor of $N_s = 4$
- The matrix is Tridiagonal + Corner pieces.
  - Can still play the Woodbury Game

- Write

$$D_t + A = T + VW^T, \qquad V = \begin{pmatrix} -U^\dagger(N_t-1)P_+ \\ 0 \\ 0 \\ \vdots \\ 0 \\ -U(N_t-1)P_- \end{pmatrix} \qquad W = \begin{pmatrix} P_- \\ 0 \\ 0 \\ \vdots \\ 0 \\ P_+ \end{pmatrix}$$

and

$$T = \begin{pmatrix} \hat{m}+A(0) & -U(0)P_- & 0 & \dots & 0 \\ -U^\dagger(0)P_+ & \hat{m}+A(1) & -U(1)P_- & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & -U^\dagger(N_t-3)P_+ & \hat{m}+A(N_t-2) & -U(N_t-2)P_- \\ 0 & 0 & \vdots & -U^\dagger(N_t-2)P_+ & \hat{m}+A(N_t-1) \end{pmatrix}$$

- At this point, things get a little messy for Clover
  - Inversion of T doable in principle
  - T^{-1} by LDU decomposition builds up continued fractions of $P_{+}A^{-1}P_{-}$.
  - A has spin structure – doesn't commute with $P_{\pm}$.
  - Projectors destroy 6×6 block structure of A
  - Need minimally inversion of 12×12 matrices.
  - Iterative inversion is undesirable (mutliplicative cost?)
- For Wilson Fermions the Schur method is straightforward
  - $(D_t + A)^{-1} \Rightarrow D_t^{-1} = C_L C_R$
  - We can already compute these easily.

Incomplete LU Decomposition

The other way to do even-odd preconditioning

- Recall our Clover Operator:

$$M = D_t + D_s + A = C_L^{-1} C_R^{-1} + D_s + A$$

- A property of C_L^{-1} and C_R^{-1}:

$$C_L^{-1} + C_R^{-1} = P_+ + P_- T + P_- + P_+ T^\dagger = C_L^{-1} C_R^{-1} + 1$$

so

$$M = C_L^{-1} + C_R^{-1} + D_s + A - 1$$

- From the previous page

$$M = C_L^{-1} + C_R^{-1} + D_s + A - 1$$

- Define

$$\mathcal{L}^{-1} = \begin{pmatrix} (C_R^e)^{-1} & 0 \\ D_s^{oe} & (C_R^o)^{-1} \end{pmatrix}$$

$$\mathcal{U}^{-1} = \begin{pmatrix} (C_L^e)^{-1} & D_s^{eo} \\ 0 & (C_L^o)^{-1} \end{pmatrix}$$

- We can write an Incomplete LU decomposition of M as:

$$M = \mathcal{L}^{-1} + \mathcal{U}^{-1} + (A - 1)$$

- Precondition as

$$\tilde{M} = \mathcal{L} M \mathcal{U} = \mathcal{U} + \mathcal{L} + \mathcal{L}(A-1)\mathcal{U}$$

- Precondition as

$$\tilde{M} = \mathcal{L} M \mathcal{U} = \mathcal{U} + \mathcal{L} + \mathcal{L}(A-1)\mathcal{U}$$

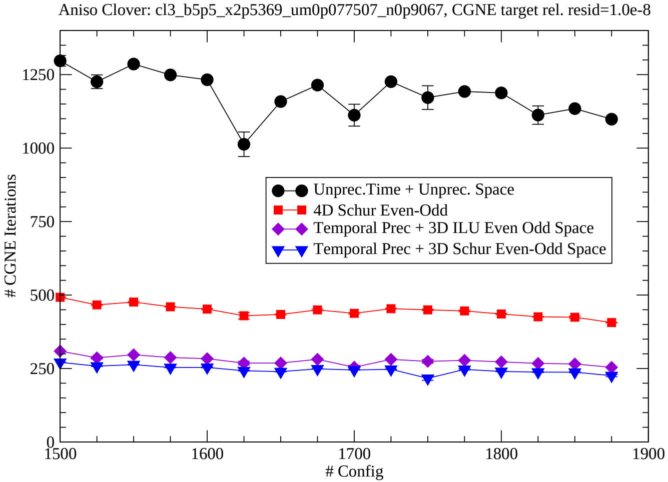
- Can immediately write down $\mathcal{L}$ and $\mathcal{U}$:

$$\mathcal{U} = \begin{pmatrix} C_L^e & -C_L^e D_s^{eo} C_L^o \\ 0 & C_L^o \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} C_R^e & 0 \\ -C_R^o D_s^{oe} C_R^e & C_R^o \end{pmatrix}$$

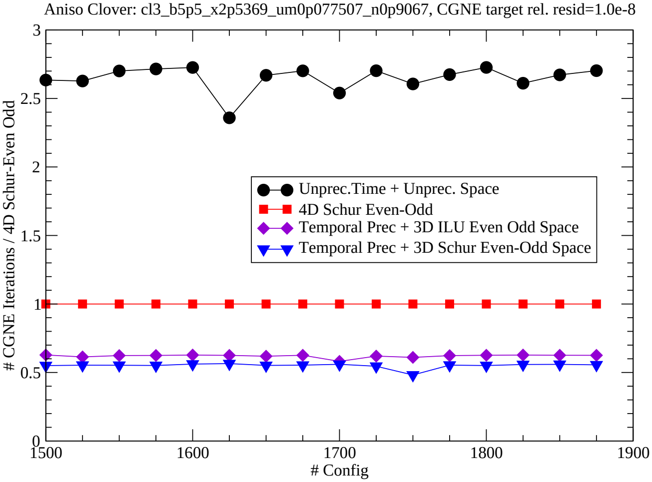
- This preconditioning is very clean.
  - Same C_L and C_R as the spatially unpreconditioned case (just applied to different subsets of sites)
  - No spin structure in the T and $T^\dagger$.
  - I don't even need to compute A^{-1}.

- 16 Configurations from Anisotropic Clover Tuning Run
  - 3 Flavours of Degenerate Clover Quarks (for m_s tuning)
  - $\beta = 5.5$, $m = -0.077507$, $c^R_{SW} = 0.90671$, $c^T_{SW} = 0.62002$, $\xi_0 = 2.5369$, $\nu = 0.90671$
  - 2 Levels of Stout Smearing in the Linear Operator, $\rho = 0.22$. Time dimension not smeared
  - Volume=$16^3 \times 64$, Target Anisotropy: $\xi \approx 3$
  - Trajectories 1500-1875 generated by Rational Hybrid Monte Carlo
    - 3 Timescales: Fermions, Spatial Gauge, Temporal Gauge
    - Integrators: 2nd Order Omelyan, 2nd Order Leapfrod, 2nd Order Leapfrog
    - Relative step sizes: $\frac{1}{7}$, $\frac{1}{3}$, $\frac{1}{2}$
- Computed Propagators(CGNE), Condition Numbers for various Operators

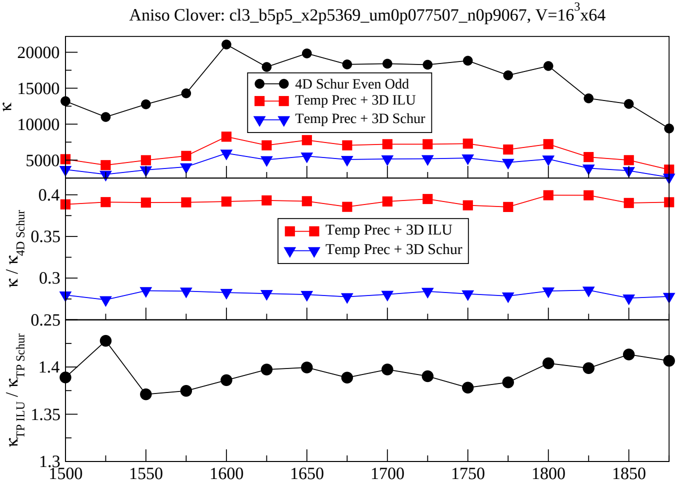
- Configurations were generated on Cray XT3/4 Facilies at
  - NCCS, Oak Ridge National Lab
  - Pittsburgh Supercomputing Center
- Inversions and condition numbers were computed on the USQCD 6n Intel-Infiniband Cluster at JLab
- The temporal preconditioning algorithms were coded with the Chroma software package



Raw Iteration Counts For Unpreconditioned, 4D Schur, Temp. Prec. + 3D ILU, Temp. Prec. + 3D Schur



Iteration Counts Normalized to 4D Schur-Even Odd (the “standard”)



Condition Number Data

Summary

- We presented the motivation for and details of Temporal Preconditioning
- Compared to the standard 4D Even-Odd Preconditioning
  - Gain just under a factor of 2 in CGNE iteration counts
  - Temp Prec. Condition Numbers are about 60-73% lower
  - Schur Style 3D Spatial Preconditioning seems slightly better than ILU (but much more complicated)
- Future Work:
  - Determine and implement MD Force terms
  - Incorporate into current Chroma HMC Structure
  - Consider BiCGStabX where (X is 2, L, etc)
  - Optimized Level 3 software... (???)

Acknowledgements

- We gratefully acknowledge support and computer time from the U. S. Department of Energy through the SciDAC (USQCD) and INCITE programmes.
- We also thank the National Science Foundation (NSF) for grants of computer time at Pittsburgh Supercomputing Center and the Teragrid through project TG-MCA07S017