Testing SSOR preconditioner for Domainwall/Overlap normal fermions

Ken-Ichi Ishikawa（Hiroshima Univ.）

[A04 & A01]

Contents

1. Lattice Chiral fermions and chiral symmetry
2. Even/Odd site preconditioning
3. SSOR preconditioning for normal equations
4. Results
5. Summary

1. Lattice Chiral fermions and chiral symmetry

- Lattice Chiral fermions
  - Lattice chiral symmetry: extends continuum chiral symmetry on the lattice
  - Avoids additive mass renormalization
  - Important phenomenologically and theoretically
    - Low energy QCD/finite temp/chiral condensate,…
- Overlap/Domainwall fermions
- Needs huge computational resources
  - Overlap/Domainwall type >> Wilson type > Staggered type

1. Lattice Chiral fermions and chiral symmetry

- Needs huge computational resources
  - Overlap/Domainwall type >> Wilson type > Staggered type
  - This is due to maintain the lattice chiral symmetry or to maintain accuracy of signum function of Wilson/Dirac operator
  - Overlap operator inversion solver
- In this talk
  - Testing solver improvement for
    - Domainwall fermion inversion (as a 5-D effective form of Overlap fermions)
    - On a small lattice

1. Lattice Chiral fermions and chiral symmetry

- Linear equations for Chiral fermions
  - Overlap operator

$$D_{OV} = \frac{1}{2}\left[(1+m_f) + (1-m_f)\gamma_5 sign(\gamma_5 D_{4DW})\right]$$

  - GW-relation $\gamma_5 D_{OV} + D_{OV}\gamma_5 = \frac{2}{1+m_f}\left[m_f\gamma_5 + D_{OV}\gamma_5 D_{OV}\right]$
  - Linear equation $D_{OV}x = b \rightarrow x = (D_{OV})^{-1}b$
- 5-Dimensional effective form
  - By introducing a signum function approximation(matrix function)

$$b_{5D} = Qb, D_{5DEff}x_{5D} = b_{5D} \rightarrow x_{5D} = (D_{5DEff})^{-1}b_{5D}$$

$$\rightarrow x = Px_{5D} = (D_{OV})^{-1}b$$

  - at a desired accuracy (sign-func).

1. Lattice Chiral fermions and chiral symmetry

- The form of the 5-Dimensional effective operator
  - Overlap fermion solver can be converted into 5-Dimensional effective form (at a accuracy for signmum function)

$$D_{5Deff}{}^{a;b}_{\alpha;\beta}(n,r;m,s) = \sum_{\gamma} X_{\alpha;\gamma}(r;s) D_{4DW}{}^{a,b}_{\gamma;\beta}(n;m) + Y_{\alpha;\beta}(r;s)\delta^{a;b}(n;m)$$

  - n, m : 4-D lattice index, r, s : 5th index, a, b : color, αβγ: spin
  - There are several types for the matrixes X and Y. Based on the approximation type of the signum function.
  - D_{4DW} : 4-D Wilson/Dirac matrix with a negative bare mass

$$D_{4DW}{}^{a,b}_{\alpha;\beta}(n;m) = (4-M)\delta^{a;b}_{\alpha;\beta}(n;m) - \sum_{\mu=1}^{4}\begin{bmatrix} (1-\gamma_\mu)_{\alpha;\beta} U_\mu{}^{a;b}(n)\delta(n+\hat{\mu};m) \\ +(1+\gamma_\mu)_{\alpha;\beta} U_\mu{}^{b;a}(n-\hat{\mu})^* \delta(n-\hat{\mu};m) \end{bmatrix}$$

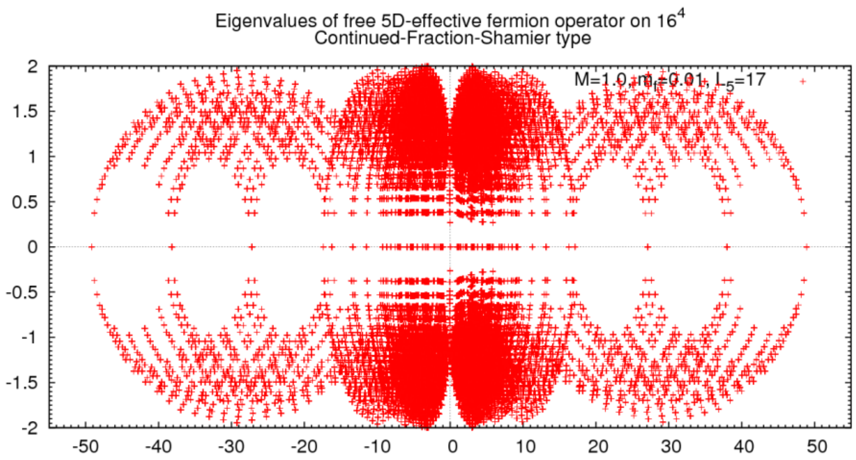
  - D_{5Deff} has sparse matrix structure on 4D lattice site index. Easy to implement.
- improve and speed up to solve $D_{5DEff}\, x_{5D} = b_{5D}$

1. Lattice Chiral fermions and chiral symmetry

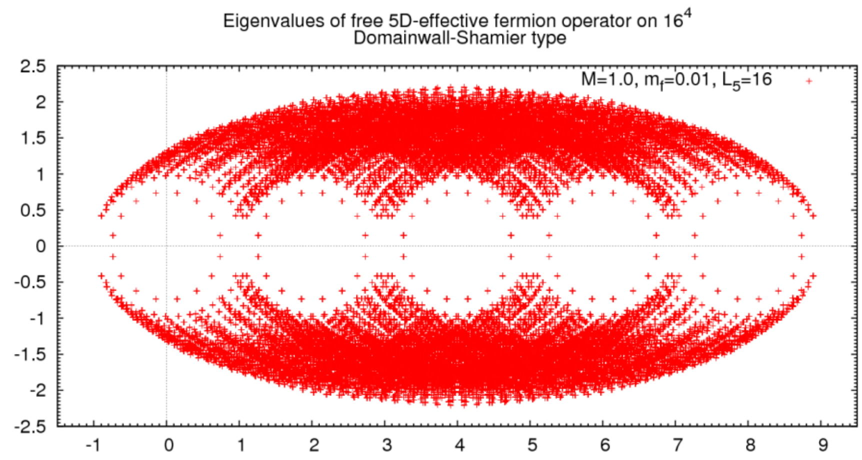
- Properties of the 5-D effective coefficient matrix

$$D_{5Deff}{}^{a;b}_{\alpha;\beta}(n,r;m,s)=\sum_{\gamma}X_{\alpha;\gamma}(r;s)D_{4DW}{}^{a,b}_{\gamma;\beta}(n;m)+Y_{\alpha;\beta}(r;s)\delta^{a;b}(n;m)$$

  – This contains negative real-part eigenvalues. (at least in free cases)



Continuued fraction approximation with Shamier kernel



Domainwall type approximation with Shamier type

  – Linear equation $D_{5DEff}\,x_{5D}=b_{5D}$ is not suitable for iterative solvers.

1. Lattice Chiral fermions and chiral symmetry

- $D_{5Deff}\, x = b$ is not suitable for any iterative solvers as D_{5Deff} contains eigenvalues with negative real-part.
- Usually this is solved by normalizing the equatoin:

$$\left(D_{5Deff}^{\dagger} D_{5Deff}\right) x = D_{5Deff}^{\dagger} b$$

- Or

$$\left(D_{5Deff} D_{5Deff}^{\dagger}\right) z = b, \quad x = D_{5Deff}^{\dagger} z$$

- The coefficient matrix is now non-negative and Hermit. We can solve them with CG iterative algorithm.
- However the convergence is slow as the coefficient matrix is doubled and could have a large condition number.
- Preconditioning for the normal equation is desired to improve the convergence property.

1. Lattice Chiral fermions and chiral symmetry

- In this talk, I tried two types of preconditioning for the Domainwall quark sovler and compared them on a small lattice.
- (1) Even/odd site precidiotioning for $D_{5Deff}\, x = b$
  + then Normalized
- (2) SSOR preconditioner for the normal equaiton:

$$\left(D_{5Deff}{}^{\dagger} D_{5Deff}\right) x = D_{5Deff}{}^{\dagger} b$$

  - No-lattice parallelism. Single computer test.
  - Type(1) preconditioning has been used in the literature.
  - Tyep(2) the direct use of the SSOR for the normal equation is not seen.
- Conclusion from my test:
- Type(2) is not good at the elapse time level even if the reduction of the iteration counts of the CG solver is seen.

2. Even/Odd preconditioning + normalize

- Preconditioning based on Even/Odd (or Red/Black) site ordering
  - 4D lattice index is colored by mod(nx+ny+nz+nt, 2). $D_{5\mathrm{Deff}}$ is Sparse matrix (w.r.t. 4D lattice site index)

$$D_{5\mathrm{Deff}} = \begin{pmatrix} D_{5\mathrm{Deff}\,ee} & D_{5\mathrm{Deff}\,eo} \\ D_{5\mathrm{Deff}\,oe} & D_{5\mathrm{Deff}\,oo} \end{pmatrix}$$

  - The linear equation $D_{5\mathrm{Deff}}\, x = b$ is transformed to

$$\hat{D}_{5\mathrm{Deff}\,ee}\, x_e = \hat{b}_e$$

$$\hat{D}_{5\mathrm{Deff}\,ee} \equiv \left(1 - \left(D_{5\mathrm{Deff}\,ee}\right)^{-1} D_{5\mathrm{Deff}\,eo} \left(D_{5\mathrm{Deff}\,oo}\right)^{-1} D_{5\mathrm{Deff}\,oe}\right)$$

$$\hat{b}_e = \left(D_{5\mathrm{Deff}\,ee}\right)^{-1}\left(b_e - D_{5\mathrm{Deff}\,eo}\left(D_{5\mathrm{Deff}\,oo}\right)^{-1} b_o\right)$$

$$x_o = \left(D_{5\mathrm{Deff}\,oo}\right)^{-1}\left(b_o - D_{5\mathrm{Deff}\,oe}\, x_e\right)$$

  - $\hat{D}_{5\mathrm{Deff}\,ee}$ is still ill conditioned (e-values with negative real part). Normal equation is applied.

$$\left(\hat{D}_{5\mathrm{Deff}\,ee}^{\dagger} \hat{D}_{5\mathrm{Deff}\,ee}\right) x_e = \hat{D}_{5\mathrm{Deff}\,ee}^{\dagger} \hat{b}_e$$

(1)Even/odd-NRCG

3. SSOR preconditioning for Normal equations

- SSOR Preconditioning by the 4D-site index structure:

$$A \equiv D_{5Deff}{}^{\dagger} D_{5Deff} = C + L + U$$

C : Diagonal part

L : Strictly Lower triangular part

U : Strictly Upper triangular part

- SSOR preconditioning by multiplying the inverse of $\left(1 + C^{-1}L\right)$ and $\left(1 + C^{-1}U\right)$

$$\hat{A} \equiv \left(1 + C^{-1}L\right)^{-1} C^{-1} A \left(1 + C^{-1}U\right)^{-1}$$

SSOR preconditioned matrix

- The inversion of $\left(1 + C^{-1}L\right)$ and $\left(1 + C^{-1}U\right)$ is done by forward or backward substitution.

3. SSOR preconditioning for Normal equations

$$D_{5Deff}^{\dagger} D_{5Deff}\, x = D_{5Deff}^{\dagger} b$$

$$\Downarrow$$

$$\left(1+C^{-1}L\right)^{-1} C^{-1}\left(D_{5Deff}^{\dagger} D_{5Deff}\right)\left(1+C^{-1}U\right)^{-1} y = \left(1+C^{-1}L\right)^{-1} C^{-1} D_{5Deff}^{\dagger} b$$

$$\Rightarrow \hat{A} y = c,$$

(B) NRSSOR-CG

$$\hat{A} \equiv \left(1+C^{-1}L\right)^{-1} C^{-1}\left(D_{5Deff}^{\dagger} D_{5Deff}\right)\left(1+C^{-1}U\right)^{-1},$$

SSOR preconditioned matrix

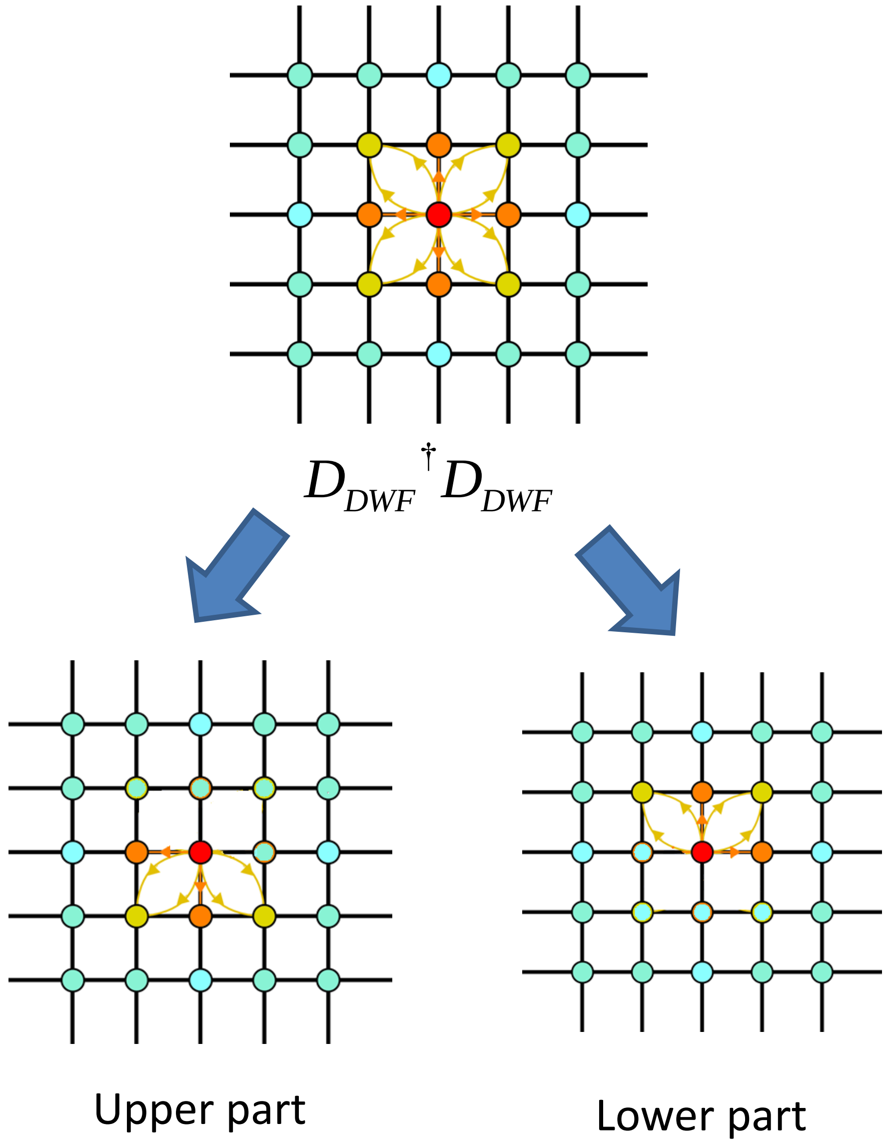
$$c = \left(1+C^{-1}L\right)^{-1} C^{-1} D_{5Deff}^{\dagger} b, \quad x = \left(1+C^{-1}U\right)^{-1} y,$$

- $\hat{A}$ is still non-negative and has a reduced condition number.
- Expected that CG solver for (B) converges faster than original equation.
- However the implementation of the forward and backward solver is difficult. This is because of the extended hopping structure of $A \equiv D_{5Deff}^{\dagger} D_{5Deff}$
- Here we consider
  - Normal site ordering.
  - Domainwall operator.

$$D_{5Deff} = D_{DWF} = K - \frac{1}{2} 1 \otimes M_{hop}$$

Hopping structure of $D_{DWF}^{\dagger} D_{DWF}$.

- 4D lattice site access pattern of $D_{DWF}^{\dagger} D_{DWF}$
- Two-hopping operations（orange and yellow）
- L-type link vars.
- L-type access is tensor.
- Computational cost is much larger if we do not use intermediate vectors.
- SSOR requires hopping access decomposition.



Difficulty of SSOR

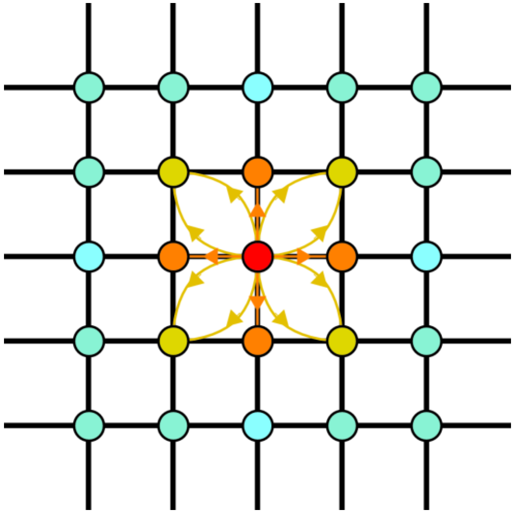
- Two hopping operations in $D_{DWF}{}^{\dagger} D_{DWF} v$

$$\sum_{\mu=1}^{4}\sum_{\nu=1}^{4}(1+\gamma_{\mu})(1-\gamma_{\nu})U_{\mu}(n)U_{\nu}(n+\hat{\mu})v(n+\hat{\mu}+\hat{\nu})$$

$$\sum_{\mu=1}^{4}\sum_{\nu=1}^{4}(1+\gamma_{\mu})(1+\gamma_{\nu})U_{\mu}(n)U_{\nu}{}^{\dagger}(n+\hat{\mu}-\hat{\nu})v(n+\hat{\mu}-\hat{\nu})$$

$$\sum_{\mu=1}^{4}\sum_{\nu=1}^{4}(1-\gamma_{\mu})(1-\gamma_{\nu})U_{\mu}{}^{\dagger}(n-\hat{\mu})U_{\nu}(n-\hat{\mu})v(n-\hat{\mu}+\hat{\nu})$$

$$\sum_{\mu=1}^{4}\sum_{\nu=1}^{4}(1-\gamma_{\mu})(1+\gamma_{\nu})U_{\mu}{}^{\dagger}(n-\hat{\mu})U_{\nu}{}^{\dagger}(n-\hat{\mu}-\hat{\nu})v(n-\hat{\mu}-\hat{\nu})$$



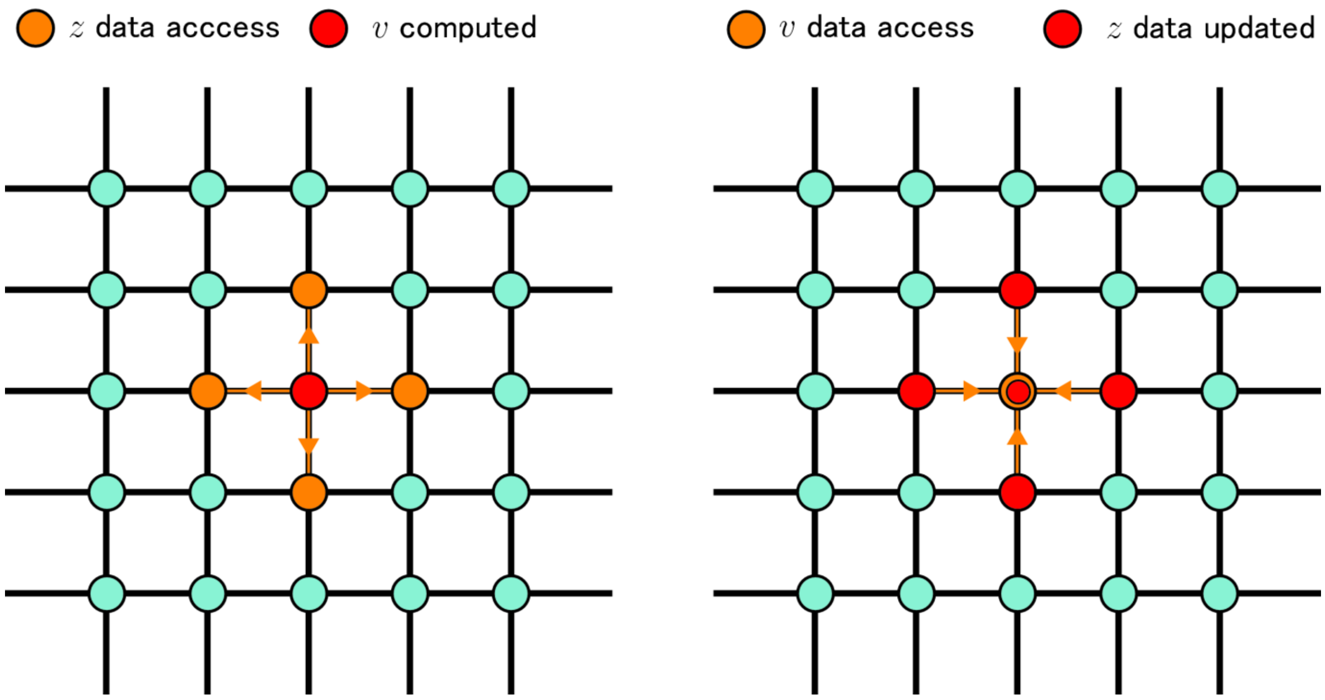
- Tensor summation => Too many computations.
- We must keep intermediate vector site by site as done usualy for

$$w = D_{DWF}{}^{\dagger} D_{DWF} v \Leftrightarrow s = D_{DWF} v, w = D_{DWF}{}^{\dagger} s$$

Textbook:"Iterative Methods for Sparse Linear Systems", Y.Saad, 2nd ed., 2003.SIAM.

SSOR for Normal equation by Saad (NRSSOR)

- Site access pattern for $D_{DWF}^{\dagger} D_{DWF}$



- Keeping working vector z s.t. $z = D_{DWF} x$ holds

3. SSOR preconditioning for Normal equations by Saad

- Solver algorithm for $(1+C^{-1}L)x=b$

$$D_{5\mathrm{Deff}} = D_{DWF} = K - \frac{1}{2} 1 \otimes M_{hop}$$

$x(:)=0; z(:)=0$

for n (4D lattice site, ascending order for L part)

  $v=0$

  for $\mu=1,2,3,4$ $v=(M_{hop}{}^{\dagger}z)(n)$

    $v = v + (1+\gamma_\mu)U_\mu(n)z(n+\hat{\mu}) + (1-\gamma_\mu)U_\mu{}^{\dagger}(n-\hat{\mu})z(n-\hat{\mu})$

  end for

  $v = b(n) - C^{-1}(K^{\dagger}z(n) - \frac{1}{2}v)$

  $x(n) = v$

  $z(n) = z(n) + Kv$

  for $\mu=1,2,3,4$

    $z(n-\hat{\mu}) = z(n-\hat{\mu}) - \frac{1}{2}(1-\gamma_\mu)U_\mu(n-\hat{\mu})v$

    $z(n+\hat{\mu}) = z(n+\hat{\mu}) - \frac{1}{2}(1+\gamma_\mu)U_\mu{}^{\dagger}(n)v$

  end for

end for

Global Working vector z

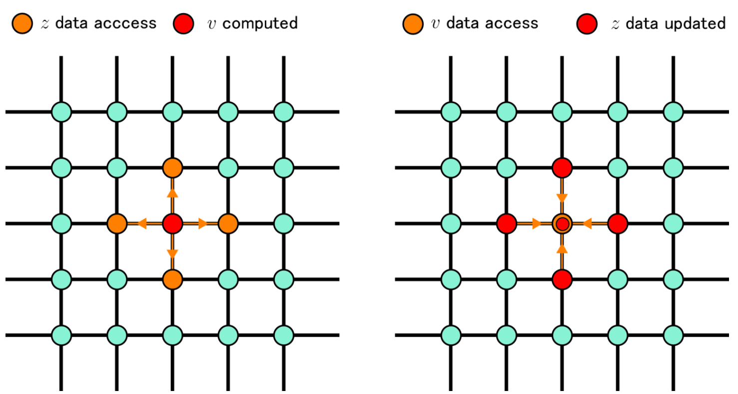
$$z = D_{DWF}x$$

Local update vector v

$$v = b(n) - (C^{-1}D_{DWF}{}^{\dagger}z)(n)$$

Update x at n

Keep the relation between x and z



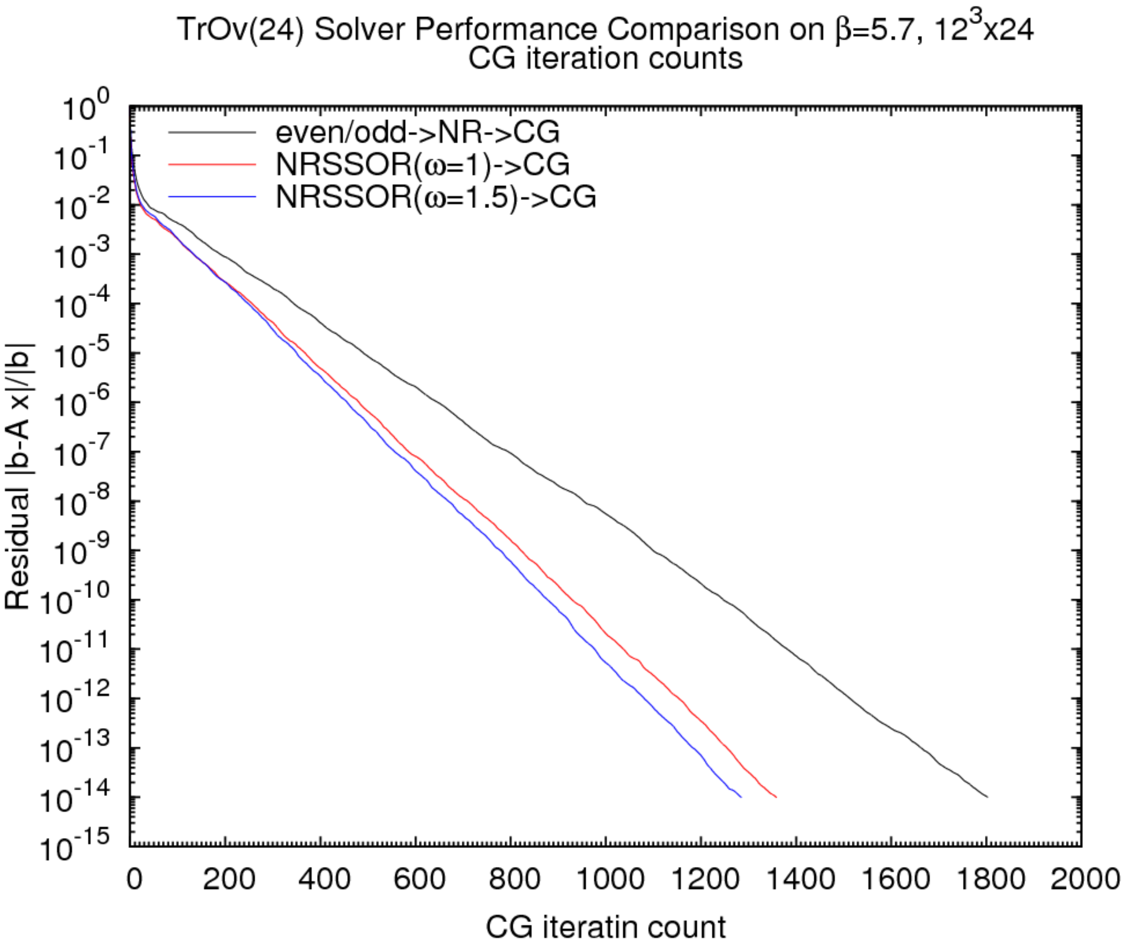
4. Results

- Benchmarking Parameters
  - 12^3x24,β=5.7 Wilson gauge Quenched one config
  - DWF (Borici) Domainwall,
  - DW height M = 1.6
  - mass m_q=0.03
  - SSOR over relaxation parameter omega=1.0, 1.5
  - 5th length N_5=24
  - m_{res} = −0.01655
  - We compare
    - Iteration counts, timing, and Flop counts for CG convergence between

      (1) Even/Odd-NRCG and (2) NRSSOR-CG

4. Results

- CG residual history

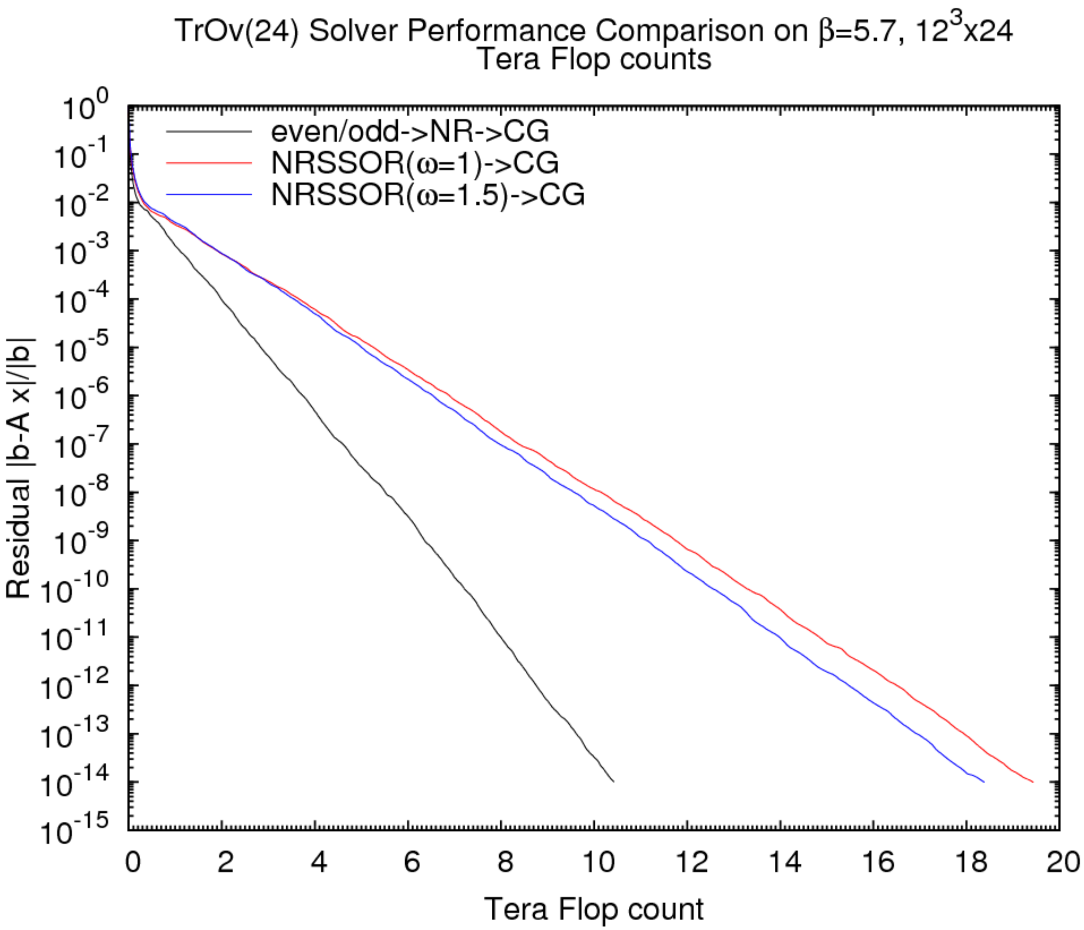


Reduced Iteration counts for NRSSOR-CG By ¾.

NRSSOR actually reduces the condition number

4. Results

- CG Floating point number operation history

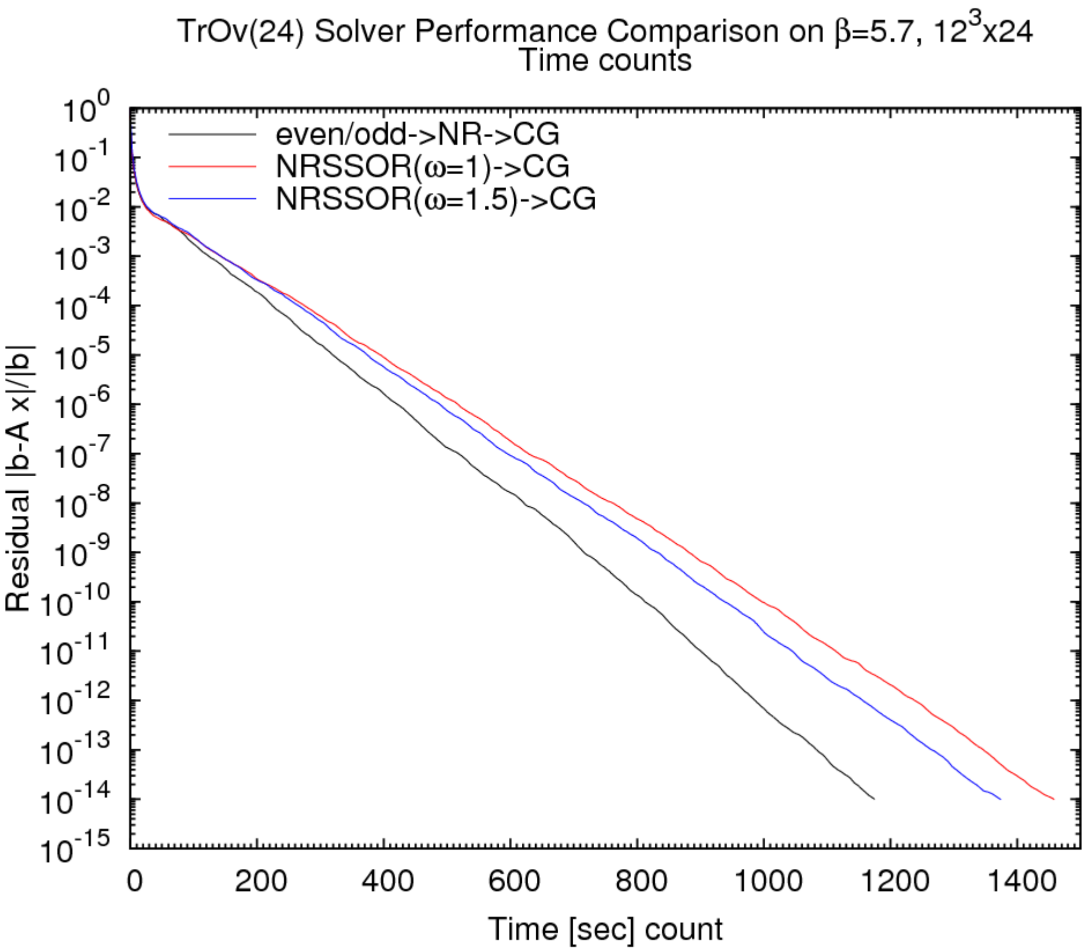


Total Flop count:
NRSSOR-CG is a factor 2 larger than E/O-NRCG

CG one iteration:
NRSSOR-CG requires 1.5x computation than E/O-NRCG during one CG-iteration.

4. Results

- Timing for CG convergence



The convergence time is slightly slower for NRSSOR-CG than E/O-NRCG.

NRSSOR has some benefit from the local update algorithm as it helps cache usage. The good cache property of NRSSOR cures larger computational cost of NRSSOR.

NRSSOR is slow.

5. Summary

- We tested the SSOR preconditioner for the normal equation of the Domainwall fermion.
- We compared the NRSSOR-CG and Even/odd-NRCG.
- The CG iteration count is reduced by a factor 3/4
- The computational cost is 1.5x larger for single CG-iteration.
- The convergence time of the NRSSOR-CG is slower than the Even/Odd-NRCG
- The Origin of the larger computational cost
  - My implementation for the NRSSOR keeps the relation $z = D_{DWF} x$ during the SSOR iteration. This computation contains redundant computation. Some component of z is not reused.