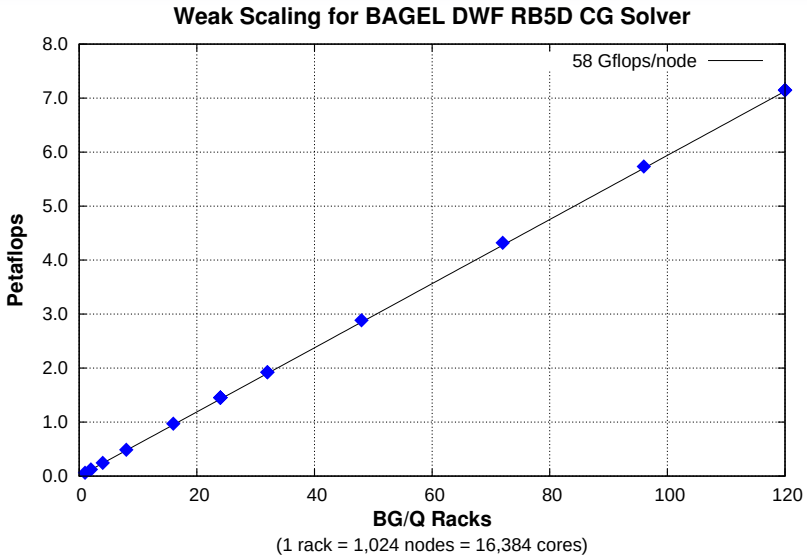
BlueGene/Q simulations and Hierarchically Deflated Conjugate Gradient

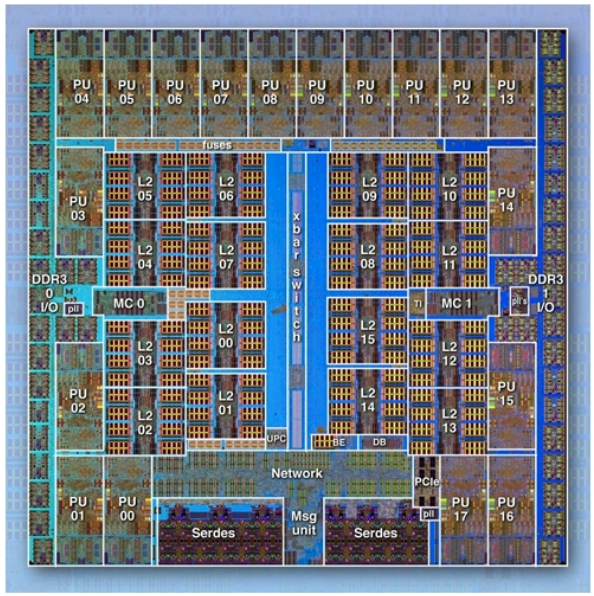
Peter Boyle, University of Edinburgh

July 4, 2013

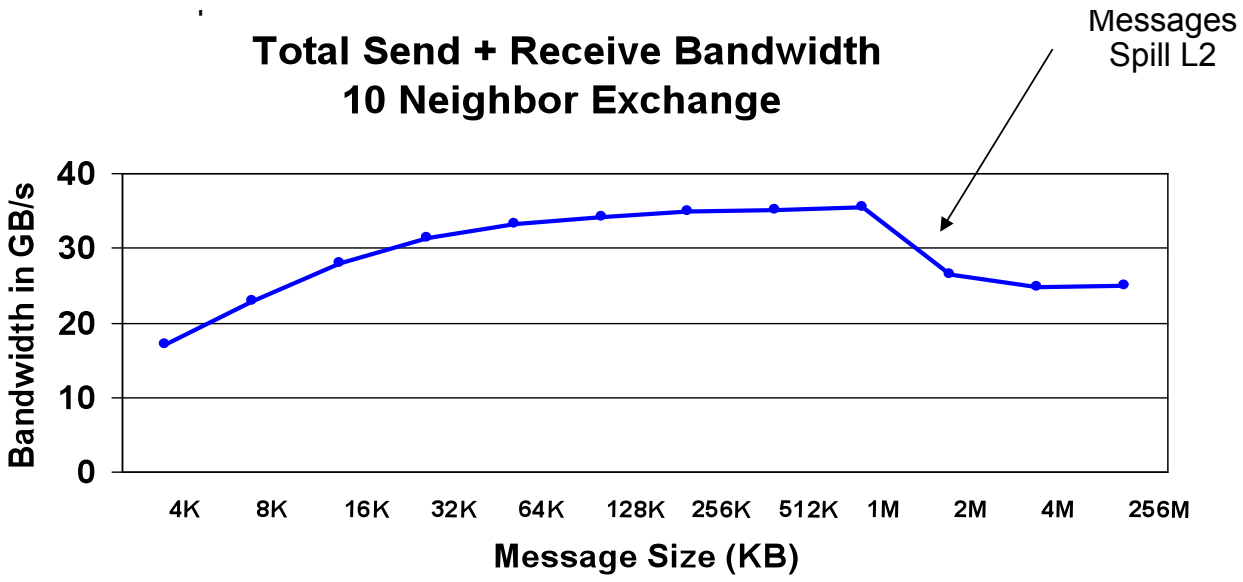
UKQCD resources



- Performance and scalability are good;
  Bagel available as open source.
  Now works with JLQCD's IroIro package
  Used by JLQCD, , RBC, UKQCD, QCDSF
  (KEK, LLNL, Edinburgh, BNL, Argonne)
  Also part of IBM's BG/Q diagnostics system
- 1.26Pflop/s system – codesigned BlueGene/Q with IBM.
- PAB designed the memory prefetch engine for BQC chip.
- Four joint patents in memory system design
  0.1% of IBM's 2012 patent haul.



Why does it scale?



SIMD optimisation

QPX supports paired complex SIMD operations (quad double)

- Develop BAGEL domain specific compiler for BG/Q QPX support
- Remember why SIMD was *easy* on the Connection Machine!
  - Subdivide node volume into smaller *virtual nodes*
  - Spread virtual nodes across SIMD lanes (these were memory banks in CM5)
  - Modifies data layout to align data parallel operations to SIMD hardware
- Data parallel operation on both virtual nodes is now simple
  - Crossing between SIMD lanes restricted to during cshifts between virtual nodes
  - Code to treat N-virtual nodes is identical to scalar code for one, except datum is N fold bigger

$$\underbrace{(A, B, C, D)}_{\text{virtual subnode}} \quad \underbrace{(E, F, G, H)}_{\text{virtual subnode}} \rightarrow \underbrace{(AE, BF, CG, DH)}_{\text{Packed SIMD}}$$

  - CSHIFT involves a CSHIFT of SIMD, and a permute *only* on the surface

$$(AE, BF, CG, DH) \rightarrow \underbrace{(BF, CG, DH, AE)}_{\text{cshift bulk}} \rightarrow \underbrace{(BF, CG, DH, EA)}_{\text{permute face}}$$

SIMD made easy

- Sequence of operations remains the same as on BG/Q after BAGEL layout transformation
- O(100%) SIMD efficiency

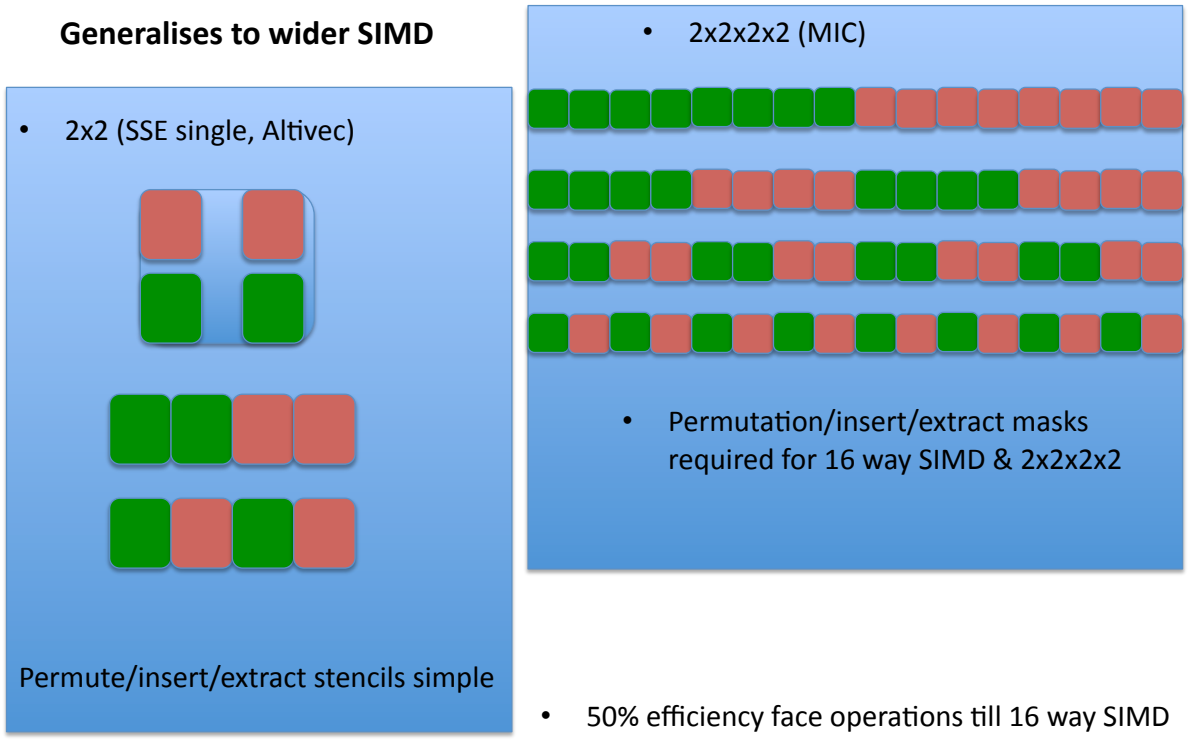
Optimised sequence of operations is *identical* for scalar complex and SIMD operation
BG/L(left, scalar complex) and BG/Q(right vector complex) assembler comparison

```
bt gt,  __lab3
addi.  %r9 , %r13 , 0
__lab3:
fxcxnpma  0 , 30 , 29 , 26
dcbt  %r18,%r9
fxcxnpma  1 , 30 , 22 , 24
stfpdx  9,%r21,%r17
fxcxnpma  2 , 30 , 7 , 23
stfpdx  10,%r22,%r17
fxcxnpma  3 , 30 , 28 , 27
dcbt  %r20,%r9
fxcxnpma  4 , 30 , 21 , 25
stfpdx  11,%r23,%r17
fxcxnpma  5 , 30 , 6 , 31
la   %r16, -1(%r16)
fxpmul  7 , 15 , 0
dcbt  %r22,%r9
fxpmul  6 , 12 , 0
```

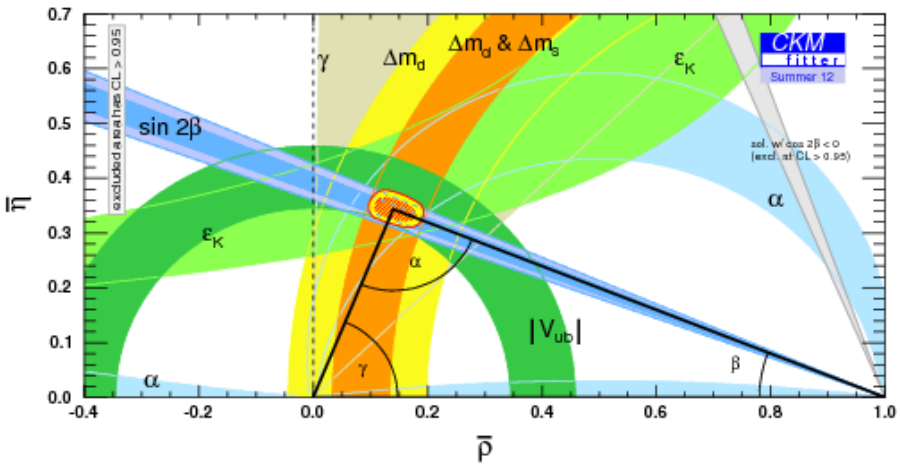
```
        bt gt,  __lab3
         addi  %r9 , %r13 , 0
__lab3:
         qvfxxnpmadd  0 , 29 , 30 , 26
         dcbt    %r18,%r9
         qvfxxnpmadd  1 , 22 , 30 , 24
         qvstfdx  9,%r21,%r17
         qvfxxnpmadd  2 , 7 , 30 , 23
         qvstfdx  10,%r22,%r17
         qvfxxnpmadd  3 , 28 , 30 , 27
         dcbt    %r20,%r9
         qvfxxnpmadd  4 , 21 , 30 , 25
         qvstfdx  11,%r23,%r17
         qvfxxnpmadd  5 , 6 , 30 , 31
         la   %r16,      -1(%r16)
         qvfxmul  7 , 15 , 0
         dcbt    %r22,%r9
         qvfxmul  6 , 12 , 0
```

Path to wider SIMD?

- F90 data parallel compiler with HPF-like distribute extensions controlling *both* SIMD and Thread parallelism could be an exascale killer app
- cmfortran + MPI !



Flavor physics from lattice QCD

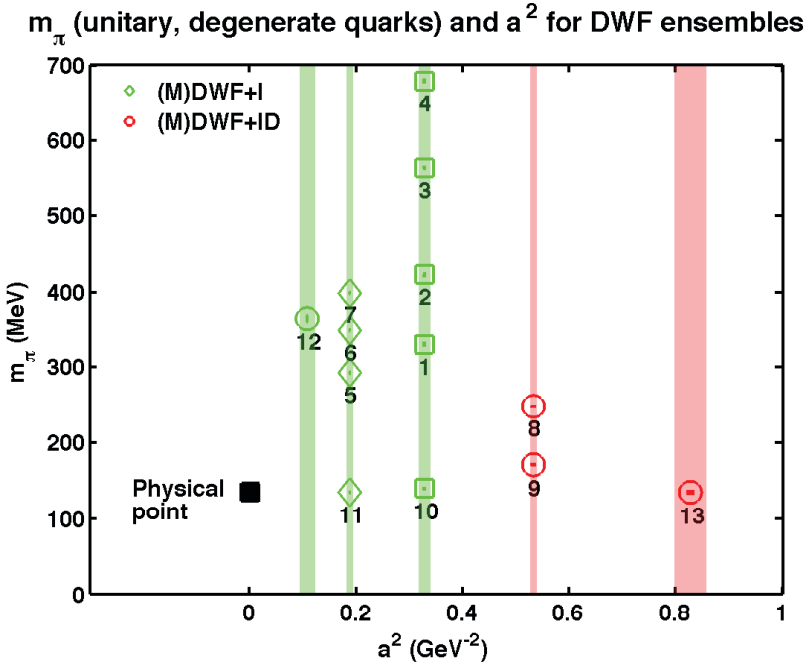


Theoretical parameter	Value and uncertainties	Reference
$f_+(0)$	$0.9632 \pm 0.0028 \pm 0.0051$	Sec. 3
f_K	$156.3 \pm 0.3 \pm 1.9$ MeV	Sec. 3
f_K/f_π	$1.205 \pm 0.001 \pm 0.010$	Sec. 3
f_{D_s}/f_D	$1.186 \pm 0.005 \pm 0.010$	Sec. 3
f_{D_s}	$251.3 \pm 1.2 \pm 4.5$ MeV	Sec. 3
f_{B_s}	$231 \pm 3 \pm 15$ MeV	Sec. 3
f_{B_s}/f_B	$1.209 \pm 0.007 \pm 0.023$	Sec. 3
$\hat{B}_{B_s}/\hat{B}_{B_d}$	$1.01 \pm 0.01 \pm 0.03$	Sec. 3
$\hat{B}_{B_s}$	$1.28 \pm 0.02 \pm 0.03$	Sec. 3
$\hat{B}_K$	$0.730 \pm 0.004 \pm 0.036$	Sec. 3
κ_ϵ	$0.940 \pm 0.013 \pm 0.023$	[10]
$\overline{m}_c(\overline{m}_c)$	$(1.286 \pm 0.013 \pm 0.040)$ GeV	[10]
$\overline{m}_t(\overline{m}_t)$	$(165.017 \pm 1.156 \pm 0.11)$ GeV	[10]
$\alpha_s(M_Z)$	0.1176 ± 0.0020	[4]
η_{cc}	computed from $\overline{m}_c(\overline{m}_c)$ and α_s	[11]
η_{ct}	0.47 ± 0.04	[12]
η_{tt}	0.5765 ± 0.0065	[13]
$\hat{\eta}_B$	0.8393 ± 0.0034	[10]

- A key motivation for lattice field theory is theoretical input to flavour physics
- Cabibbo, Kobayashi, Maskawa flavour induced by Higgs couplings
  Sensitivity too New Particles through loop corrections if these also mix flavours
  Possibly induce non-unitarity of measured CKM matrix
- Relevance for both Energy & Intensity Frontier
  - Lattice calculations increasingly dominant source of theoretical input
  - Range of calculations *increasing* with time
    c.f. RBC-UKQCD $K \to \pi\pi$ work (2012 Wilson Award)
- Focused K_{l3} decays, f_K/f_π (V_{us}) and neutral kaon mixing within and beyond standard model, $K \to \pi\pi$

RBC-UKQCD simulation status

Ensembles



New physical point Mobius (H_T) 2+1f ensembles, $m_{\rm res} \sim 1$ MeV

Iwasaki	$48^3 \times 96 \times 24$	$a^{-1} = 1.75$GeV	1600 Trajectories
Iwasaki	$64^3 \times 128 \times 12$	$a^{-1} = 2.3$GeV	2100 (1000) Trajectories

RBC-UKQCD simulation status

- All-mode-averaging analysis giving 0.1 % scale statistical errors for both ensembles
- EigCG deflation is used in solver
- 50 measurements on 48^3
- 22 measurements on 64^3

Quantity	Physical Value	Simulation Value	Deviation (Sim. - Phys.)/Phys.
m_π/m_K	0.2723	0.2793(6)	2.5%
m_π/m_Ω	0.0807	0.0835(5)	3.3%
m_K/m_Ω	0.2964	0.2989(16)	0.8%

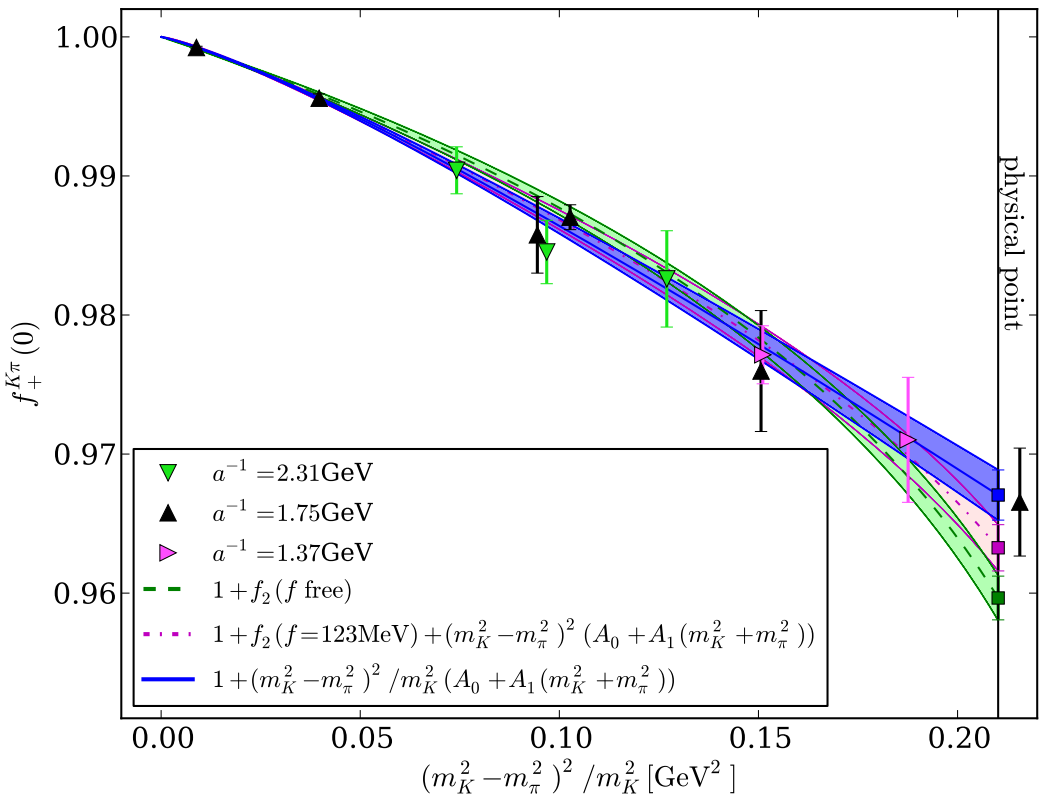
Above the π^0, below the $\pi^{+/-}$.

$$f_K/f_\pi = 1.1914(21)$$

K_{l3} form factor

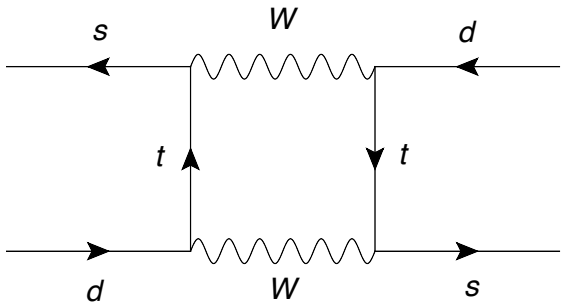
$$\langle \pi(p_\pi)|V_\mu|K(p_K)\rangle = f_+^{K\pi}(q^2)(p_K+p_\pi)_\mu + f_-^{K\pi}(q^2)(p_K-p_\pi)_\mu$$

- Product $G_F V_{us} f_+^{K\pi}(q^2=0)$ determined experimentally
  - Theoretical input required to determine V_{us} (λ in Wolfenstein parametrisation)
  - Chiral perturbation theory, Lattice QCD are the most competitive approaches

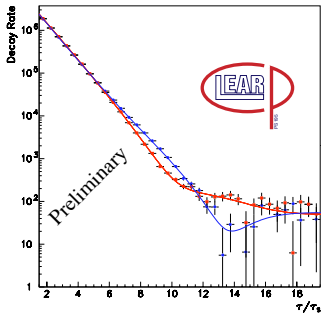


New datapoint eliminates systematic error in mass extrapolation.

Neutral Kaon oscillation and decay



$\Rightarrow$



Induces off diagonal part of Wigner Weisskopf Hamiltonian:

$$M_{12}^{*} = \frac{1}{2m_K}\frac{G_F^2 M_W^2}{16\pi^2}\left[\lambda_c^2 S_0(x_c,x_c) + 2\lambda_c\lambda_t S_0(x_c,x_t) + \lambda_t^2 S_0(x_t,x_t)\right]$$
$$\times \quad \langle K^0|\bar{s}\gamma_\mu(1-\gamma_5)d\bar{s}\gamma_\mu(1-\gamma_5)d|\bar{K}^0\rangle$$

Here,

$$\lambda_t = V_{ts}^{*}V_{td} = -A^2\lambda^5(1-\rho-i\eta)$$

Indirect CP violation from imaginary part of λ_t^2 piece

$$\Rightarrow \eta(1-\rho) = \text{constant}$$

hyperbola ϵ_K constraint
Also obtaining:

1. B_K giving 0.1% statistical error
2. $K \to \pi\pi$ $\Delta_I = \frac{3}{2}$ giving 2% statistical error
   may lead to *new* constraint on $i\eta$

Eigenvector Deflation

Krylov solvers convergence controlled by the condition number

$$\kappa \sim \frac{\lambda_{max}}{\lambda_{min}}$$

- Lattice chiral fermions possess an exact index theorem
- Index theorem $\Rightarrow \exists$ near zero modes separated from zero only by quark mass
- Recent algorithmic progress eliminates low mode subspace from Krylov inversion

EigCG:

- Determine $N_{vec} \sim O(V)$ eigenvectors ϕ_i up to some physical λ
- $48^3 \Rightarrow 600$ vectors, $64^3 \Rightarrow 1500$ vectors
- Significant setup cost & storage cost $\propto V^2$
- Eliminates N_{vec} dimensional subspace $S = \mathrm{sp}\{\phi_i\}$ from problem

$$M = \begin{pmatrix} M_{\bar{s}\bar{s}} & \epsilon \\ \epsilon^\dagger & M_{ss} \end{pmatrix} \quad ; \quad M_{ss}^{-1} = \frac{1}{\lambda_i} |i\rangle\langle i|$$

Where $\epsilon = M_{\bar{s}s}$ is proportional to the error in the eigenvectors
Guess $\phi = \mathrm{diag}\{0\} \oplus \mathrm{diag}\{\frac{1}{\lambda_i}\}\eta$

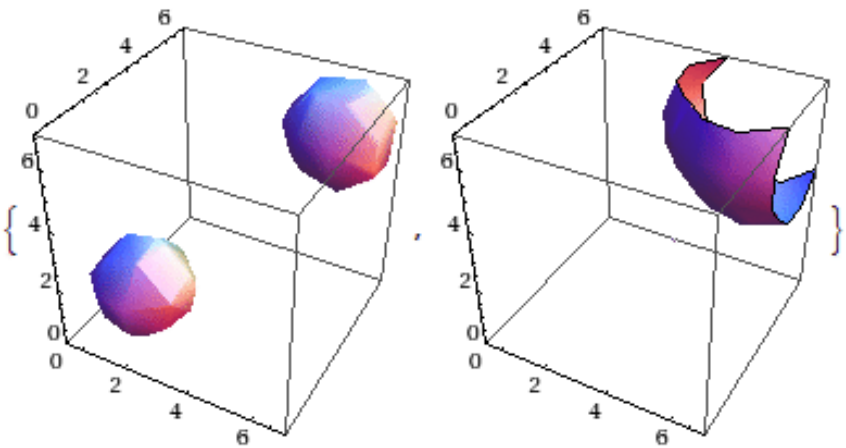
Why can we do better

Luscher's observation: local coherence of low modes

low virtuality solutions of gauge covariant Dirac equation locally similar

Consider N well separated instantons

- N-zero modes look like admixtures of single instanton eigenmodes
- Divide *one* mode into chunks centred on each each instanton
- All N-zero modes described by the span of these chunks



Luscher's inexact deflation

Avoid critical slowing down in Krylov solution of

$$M\psi = \eta$$

- Accelerate sparse matrix inversion by treating a vector subspace $S = \mathrm{span}\{\phi_k\}$ exactly
- If the lowest lying eigenmodes are well contained in S the "rest" of the problem avoids critical slowing down

Setup:

- Must generate subspace vectors ϕ_k that are "rich" in low modes
- Subdividing these vectors into blocks b

$$\phi_k^b(x) = \begin{cases} \phi_k(x) & ; \quad x \in b \\ 0 & ; \quad x \notin b \end{cases}$$

yields a much larger subspace[1] e.g. $48^3 \times 96$ lattice with 4^4 blocks gives a $12^3 \times 24$ coarse grid and $O(10^4)$ bigger deflation space.

[1] This idea was previously used in adaptive multigrid where small *covariant derivative* $\leftrightarrow$ algebraically smooth. Blocks $\leftrightarrow$ aggregates. Luscher, though reinventing ideas used in multigrid, established connection between Krylov deflation such as EigCG and MG

Luscher's setup

Introduce subspace projectors

$$P_S = \sum_{k,b} |\phi_k^b\rangle\langle\phi_k^b| \qquad ; \qquad P_{\bar{S}} = 1 - P_S \tag{1}$$

Compute M_{ss} as

$$M = \begin{pmatrix} M_{\bar{S}\bar{S}} & M_{S\bar{S}} \\ M_{\bar{S}S} & M_{SS} \end{pmatrix} = \begin{pmatrix} P_{\bar{S}} M P_{\bar{S}} & P_S M P_{\bar{S}} \\ P_{\bar{S}} M P_S & P_S M P_S \end{pmatrix}$$

- Can represent matrix M exactly on this subspace by computing its matrix elements, known as the *little Dirac operator* [2]

$$A_{jk}^{ab} = \langle\phi_j^a|M|\phi_k^b\rangle$$

$$(M_{SS}) = A_{ij}^{ab}|\phi_i^a\rangle\langle\phi_j^b|$$

and

$$M_{SS}^{-1} = (A^{-1})_{ij}^{ab}|\phi_i^a\rangle\langle\phi_j^b|$$

A inherits a sparse structure from M

[2] Coarse grid matrix in MG

Subspace Schur decomposition

We can Schur decompose any matrix

$$M = \begin{bmatrix} M_{\bar{s}\bar{s}} & M_{\bar{s}s} \\ M_{s\bar{s}} & M_{ss} \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} 1 & M_{\bar{s}s}M_{ss}^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & M_{ss} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ M_{ss}^{-1}M_{s\bar{s}} & 1 \end{bmatrix} \tag{3}$$

$$= UDL \tag{4}$$

where the Schur complement

$$S = M_{\bar{s}\bar{s}} - M_{\bar{s}s}M_{ss}^{-1}M_{s\bar{s}} \tag{5}$$

and the inverse matrix is

$$M^{-1} = L^{-1}D^{-1}U^{-1} \tag{6}$$

$$= \begin{bmatrix} 1 & 0 \\ -M_{ss}^{-1}M_{s\bar{s}} & 1 \end{bmatrix} \begin{bmatrix} S^{-1} & 0 \\ 0 & M_{ss}^{-1} \end{bmatrix} \begin{bmatrix} 1 & -M_{\bar{s}s}M_{ss}^{-1} \\ 0 & 1 \end{bmatrix} \tag{7}$$

Projector properties

Lower and upper diagonal matrices of Schur decomp. correspond to Luscher's projectors P_L and P_R

$$P_L = P_{\bar{S}} U^{-1} = \begin{pmatrix} 1 & -M_{\bar{S}S} M_{SS}^{-1} \\ 0 & 0 \end{pmatrix}$$

$$P_R = L^{-1} P_{\bar{S}} = \begin{pmatrix} 1 & 0 \\ -M_{SS}^{-1} M_{S\bar{S}} & 0 \end{pmatrix}$$

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & M_{SS}^{-1} \end{pmatrix}$$

Luscher's properties:

$$P_L^2 = P_L \qquad ; \qquad P_R^2 = P_R$$

$$P_L M = M P_R = P_L M P_R = (1-P) D (1-P) = \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}$$

$$P P_L = P_R P = 0$$

$$P_L (1-P) = (1-P) P_R = (1-P) = P_{\bar{S}}$$

Luscher's algorithm

$$M\psi = UDL\psi = \eta$$

Multiply by P_L and $1 - P_L$ obtaining two independent equations:

$$P_L M\psi = P_L \eta$$

$$(1 - P_L)M\psi = M(1 - P_R)\psi = (1 - P_L)\eta$$

The first implies

$$P_L M P_R \psi = P_L \eta$$

and second implies

$$(1 - P_R)\psi \equiv \psi_s + M_{ss}^{-1} M_{s\bar{s}} \psi_{\bar{s}} = M_{ss}^{-1} \eta_s$$

Luscher develops an inversion algorithm for

$$P_L M \chi = P_L \eta$$

and then reconstructs the complete solution

$$\psi = P_R \chi + M_{ss}^{-1} \eta_s$$

Also we have

$$QM = 1 - P_R$$

Inversion of $P_L M$

Luscher suppress little Dirac Operator overhead with Schwarz alternating procedure (SAP)

$$(P_L M) M_{SAP} \phi = P_L \eta$$

$$\psi = M_{SAP} \phi$$

$$P_L = \begin{pmatrix} 1 & -M_{\bar{S}S} M_{SS}^{-1} \\ 0 & 0 \end{pmatrix}$$

- Each step of an outer Krylov solver involves an *inner* Krylov solution of the little Dirac op coarse grid
- This enters the matrix $P_L M$ being inverted and errors propagator into solution
- Luscher tightens the precision during convergence; uses history forgetting *flexible* GCR

Non-hermitian system possible as evalues of D_W live in right half of complex plane:

- Little Dirac operator for D_W is *nearest neighbour*
- Red black preconditioning of Little dirac op possible
- Schwarz alternating procedure possible as D_W does not connect red to red.

Generalisation to 5d Chiral fermions

Krylov solution of Hermitian system necessary (CG-NR, MCR-NR)
Aim to speed up the red-black preconditioned system as this starts better conditioned

$$\mathcal{H} = \left(M_{oo} - M_{oe}M_{ee}^{-1}M_{eo}\right)^{\dagger}\left(M_{oo} - M_{oe}M_{ee}^{-1}M_{eo}\right) = M_{\mathrm{prec}}^{\dagger}M_{\mathrm{prec}}$$

Matrix being deflated is is next-to-next-to-next-to-nearest-neighbour!

Tasks!

- Must find *further suppression of little Dirac operator* overhead as LDop more costly
- Must find a replacement for the Schwarz preconditioner
- Must find appropriate solver: $(P_L M)M_{SAP}$ nonhermitian matrix so unsuitable for CG
- Must ensure system is tolerant to ill convergence of inner Krylov solver(s).

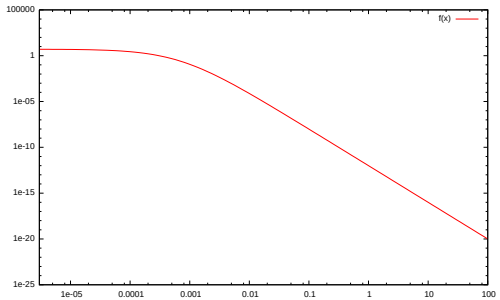
Hermiticity and improved subspace generation

- Hermitian system gains the properties

$$P_L^\dagger = P_R \qquad (P_L M)^\dagger = P_L M$$

- Since we use $\mathcal{H} = M_{\rm prec}^\dagger M_{\rm prec}$ we have a Hermitian Positive (semi) Definite matrix. Generate subspace with rational multi-shift solver applied to Gaussian noise

$$R(\eta^{\rm Gaussian}) = \frac{24\epsilon}{(\mathcal{H}+\epsilon)(\mathcal{H}+2\epsilon)(\mathcal{H}+3\epsilon)(\mathcal{H}+4\epsilon)}$$



- Classic low pass filtering problem – use rational filter
  - Gain $1/x^4$ suppression in single pass *without* inverse iteration
  - $\epsilon \sim 10^{-3}$ adds IR safety to the inversion $O(1000)$ iterations per subspace vector
  - NB Also possible for $\gamma_5 D_W$

SHAPE OF VECTORS...

Little Dirac Operator

4 hop little Dirac operator is painful!

- Limit the stencil of the Little Dirac operator by requiring block $\geq 4^4$
- Mobius fermions M_{ee}^{-1} is non-local in s-direction $\Rightarrow$ blocks stretch full s-direction
- Sparse in 4d with next-to-next-to-next-to-nearest coupling
- Matrix *still* connects to 80 neighbours

$$(\pm\hat{x}),\ (\pm\hat{y}),\ (\pm\hat{z}),\ (\pm\hat{t})$$
$$(\pm\hat{x}\pm\hat{y}),\ (\pm\hat{x}\pm\hat{z}),\ (\pm\hat{x}\pm\hat{t}),\ (\pm\hat{y}\pm\hat{z}),\ (\pm\hat{y}\pm\hat{t}),\ (\pm\hat{z}\pm\hat{t})$$
$$(\pm\hat{x}\pm\hat{y}\pm\hat{z}),\ (\pm\hat{x}\pm\hat{y}\pm\hat{t}),\ (\pm\hat{x}\pm\hat{z}\pm\hat{t}),\ (\pm\hat{y}\pm\hat{z}\pm\hat{t})$$
$$(\pm\hat{x}\pm\hat{y}\pm\hat{z}\pm\hat{t})$$

- Underlying cost at least ten times more than non-Hermitian system
- Reducing to 4d has saved L_s factor *but may require more vectors* to describe 5th dimension

Little Dirac Operator Implementation

- 10×10 matrix multiply reasonably high cache reuse
- Using IBM xlc vector intrinsics gives adequate performance
- 80 small messages of order 1-5 KB
- Programme BG/Q DMA engines directly to eliminate MPI overhead
- Asynchronous send overhead under 10 microseconds with precomputed DMA descriptors.
- 50x faster than MPI calls.
- Single precision accelerated gives around 50 Gflop/s per node in L2 cache
- (re)Discovered bug in L2 cache around 4 months after Argonne

Replacing SAP preconditioner

Since we are deflating the low modes, seek approximate inverse preconditioner for Hermitian system that is accurate for high modes.

- Useful to prototype preconditioner using Chebyshev polynomials
  Can shape spectral response to any desired shape ... at a cost
- Naive left-right preconditioner:

  $$L^\dagger (P_L \mathcal{H}) L\phi = L^\dagger P_L \eta$$

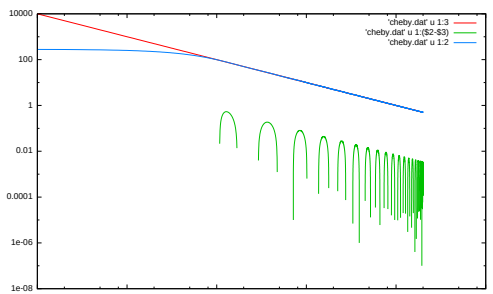
  Use fixed order Chebyshev polynomial preconditioner

  $$L = \mathrm{Cheby}(x^{-\frac{1}{2}}, \mathcal{H}) \quad ; \quad \mathcal{H} = M^\dagger_{\mathrm{prec}} M_{\mathrm{prec}} \quad ; \quad \chi = L\phi$$

  - This is Hermitian and works in CG, but is not a very good preconditioner.
- Better to use preconditioned CG (p 278 Saad) with Hermitian preconditioner M_P

  $$M_P = L^\dagger L = \mathrm{Cheby}(x^{-\frac{1}{2}}, \mathcal{H})\mathrm{Cheby}(x^{-\frac{1}{2}}, \mathcal{H}) \to \mathrm{Cheby}(x^{-1}, \mathcal{H})$$

- Accuracy exponential in Cheby order so better to use single, higher order
- Found it best to restrict range of Chebyshev to be accurate at higher eigenvalues, rely on deflation on lowest modes!



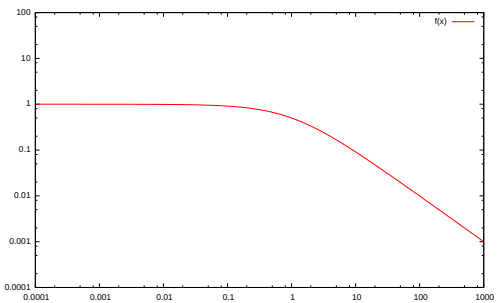
Tuned chebyshev preconditioner spectral response

IR shift preconditioner

- Better to use a Krylov solver
  Data dependent coefficients seek optimal polynomial *for the actual spectrum* of $\mathcal{H}$ under some norm
- Use fixed number of shifted CG iterations as preconditioner (IR shifted preconditioner)

$$M_{IRS} = \frac{1}{\mathcal{H} + \lambda}$$

- λ is an gauge covariant infra-red regulator that shifts the lowest modes
- Keeps the Krylov solver working hard on the high mode region
- Plays similar role to the domain size in SAP
- Does not have locality benefit of SAP[3]



[3] Comms in BG/Q tolerate this, but Additive Schwarz is worth investigating for future machines (suggested by Mike Clark)

Robustness

Two inner Krylov solvers

- Little Dirac operator inversion $Q \equiv M_{SS}^{-1}$
- IR shifted preconditioner inversion $M_{IRS} = \frac{1}{\mathcal{H}+\lambda}$

Curious robustness effects: during solution to 10^{-8} on a 16^3 lattice

M_{SS}^{-1} residual	M_{IRS} residual	Iteration count
10^{-11}	10^{-8}	36
10^{-8}	10^{-8}	Non converge [4]
10^{-11}	10^{-8}	36
10^{-11}	10^{-4}	36
10^{-11}	10^{-2}	36

Although *flexible* CG (Notay 1999) exists better to understand *why* the CG is tolerant to variability in M but not Q

[4] smallest residual is 10^{-7} then diverges. Here Luscher introduced flexible algorithms

Robustness

Consider preconditioned CG with $A = P_L \mathcal{H} = \begin{pmatrix} 1 & -M_{\bar{S}S} M_{SS}^{-1} \\ 0 & 0 \end{pmatrix} \mathcal{H}$

1. $r_0 = b - Ax_0$
2. $z_0 = M_{IRS} r_0$; $p_0 = z_0$
3. for iteration k
4. $\alpha_k = (r_k, z_k)/(p_k, Ap_k)$
5. $x_{k+1} = x_k + \alpha_k p_k$
6. $r_{k+1} = r_k - \alpha_k A p_k$
7. $\mathbf{z_{k+1}} = \mathbf{M_{IRS} r_{k+1}}$
8. $\beta_\mathbf{k} = (\mathbf{r_{k+1}}, \mathbf{z_{k+1}})/(\mathbf{r_k}, \mathbf{z_k})$
9. $\mathbf{p_{k+1}} = \mathbf{z_{k+1}} + \beta_\mathbf{k} \mathbf{p_k}$
10. end for

- Noise in the preconditioner M_{IRS} *only* enters the search direction α_k is based on matrix elements of $P_L \mathcal{H}$.
- Better to use the Little Dirac operator inverse as a preconditioner ...and not separate the solution into subspace and complement

Combining preconditioners

- Have little Dirac operator Q and M_{IRS} representing approximate inverse
  - Q on subspace containing low mode
  - M_{IRS} on high mode space
  - splitting is necessarily *inexact*
- Options for combining these as a preconditioner
  - Additive

$$M_{IRS} + Q$$

  - Consider alternating error reduction steps

$$\begin{aligned} x_{i+1} &= x_i + M_{IRS}[b - \mathcal{H}x_i] \\ x_{i+2} &= x_{i+1} + Q[b - \mathcal{H}x_{i+1}] \\ &= x_i + M_{IRS}[b - \mathcal{H}x_i] + Q[b - \mathcal{H}[x_i + M_{IRS}[b - \mathcal{H}x_i]]] \\ &= x_i + [(1 - Q\mathcal{H})M_{IRS} + Q](b - \mathcal{H}x_i) \\ &= x_i + [P_R M_{IRS} + Q](b - \mathcal{H}x_i) \end{aligned}$$

  - Infer family of preconditioner

Sequence	Preconditioner	Name
additive	$M_{IRS} + Q$	AD
M_{IRS}, Q	$P_R M_{IRS} + Q$	A-DEF2
Q, M_{IRS}	$M_{IRS} P_L + Q$	A-DEF1
Q, M_{IRS}, Q	$P_R M_{IRS} P_L + Q$	Balancing Neumann Neumann (BNN)
Q, M_{IRS}, Q	$M_{IRS} P_L + P_R M_{IRS} + Q - M_{IRS} P_L \mathcal{H} M_{IRS}$	MG Hermitian V(1,1) cycle

Generalised framework for inexact deflation solvers

Extend framework of Tang, Nabben, Vuik, Erlangga (2009) to three levels

Take $Q = \begin{pmatrix} 0 & 0 \\ 0 & M_{SS}^{-1} \end{pmatrix}$ and $M_{IRS} = (\mathcal{H} + \lambda)^{-1}$

Method	V_{start}	M_1	M_2	M_3	V_{end}
PREC	x	M_{IRS}	$\mathbb{1}$	$\mathbb{1}$	x_{k+1}
AD	x	$M_{IRS} + Q$	$\mathbb{1}$	$\mathbb{1}$	x_{k+1}
DEF1	x	M_{IRS}	$\mathbb{1}$	P_L	$Qb + P_R x_{k+1}$
DEF2	$Qb + P_R x$	M_{IRS}	P_R	$\mathbb{1}$	x_{k+1}
A-DEF1	x	$M_{IRS} P_L + Q$	P_R	$\mathbb{1}$	x_{k+1}
A-DEF2	$Qb + P_R x$	$P_R M_{IRS} + Q$	$\mathbb{1}$	$\mathbb{1}$	x_{k+1}
BNN	x	$P_R M_{IRS} P_L + Q$	$\mathbb{1}$	$\mathbb{1}$	x_{k+1}

Observations:

- Remain in deflated Krylov picture
- Luscher's algorithm is DEF1
- A-DEF2 moves the little Dirac operator into the preconditioner M_1
- A-DEF1 looks like V(1,0) multigrid
- A-DEF2 looks like V(0,1) multigrid
- Will make it Heirarchical by deflating the deflation matrix Q

Algorithm

1. x arbitrary
2. $x_0 = V_{\text{start}}$
3. $r_0 = b - \mathcal{H} x_0$
4. $y_0 = M_1 r_0$; $p_0 = M_2 y_0$
5. for iteration k
6. $w_k = M_3 \mathcal{H} p_k$
7. $\alpha_k = (r_k, y_k)/(p_k, w_k)$
8. $x_{k+1} = x_k + \alpha_k p_k$
9. $r_{k+1} = r_k - \alpha_k w_k$
10. $\mathbf{y_k} = \mathbf{M_1 r_k}$
11. $\beta_{\mathbf{k}} = (\mathbf{r_{k+1}}, \mathbf{y_{k+1}})/(\mathbf{r_k}, \mathbf{y_k})$
12. $\mathbf{p_{k+1}} = \mathbf{M_2 y_{k+1}} + \beta_{\mathbf{k}} \mathbf{p_k}$
13. end for
14. $x = V_{\text{end}}$

Why does CG work here?

- Hermiticity of M_1 clear for BNN but not A-DEF1/2
  Theorem: for $V_{\text{start}} = Qb + P_R x$ A-DEF2 is identical to BNN.

- We have from $Q\mathcal{H} = (1 - P_R)$

$$Qr_0 = Q[\mathcal{H}V_{\text{start}} - b] = (1 - P_R)[Q_b + P_R x] - Qb = P_R Q_b = 0$$

$$Q\mathcal{H}p_0 = (1 - P_R)[P_R M P_L + Q]r_0 = 0$$

- get induction steps:

$$Qr_{j+1} = Qr_j - \alpha_j Q\mathcal{H}p_j = 0$$

$$Q\mathcal{H}p_{j+1} = (1 - P_R)[P_R M P_L + Q]r_j + \beta_j Q\mathcal{H}p_j = 0$$

- Can also show $P_L r_0 = 0$ and $P_L \mathcal{H}p_0 = \mathcal{H}p_0$ so that

$$P_L \mathcal{H}p_{j+1} = \mathcal{H}P_R[P_R M P_L + Q]r_j + \beta_j p_j = \mathcal{H}p_{j+1}$$

  and

$$P_L r_{j+1} = P_L r_j - \alpha_j P_L \mathcal{H}p_j = r_j - \alpha_j \mathcal{H}p_j = r_{j+1}$$

> BNN then retains $P_L r_j = r_j$ in exact subspace projection arithmetic
> $\Rightarrow$ BNN iteration ($P_R M P_L r_j$) and A-DEF2 iteration ($P_R M r_j$) equivalent up to rounding

- DEF1(Luscher), DEF2, A-DEF1, A-DEF2, BNN are ALL equivalent up to rounding

> They *differ* hugely in sensitivity to convergence error in Q

Reducing little Dirac operator overhead

- Use A-DEF2 to move the little Dirac operator into preconditioner
  Can relax convergence precision to 10^{-2}
  $\Rightarrow$ eight order of magnitude gain, saving of O(10) in cost
- Deflate the deflation matrix (Heirarchical).
  Computing 128 low modes is cheap and saves another factor of 10.
- Reduces O(2000) little Dirac operator iterations to O(20).

From 48^3 at physical quark masses

Precision	Heirarchical deflation	iterations
10^{-7}	N	4478
10^{-7}	Y	250
10^{-2}	Y	63

100 x reduction in little dirac operator overhead!

HDCG algorithm

Subspace generation

1. Generate N_S vectors ϕ_k from rational (4th order low pass filter)

$$R(\mathcal{H}) = \frac{1}{(\mathcal{H}+\lambda_S)(\mathcal{H}+2\lambda_S)(\mathcal{H}+3\lambda_S)(\mathcal{H}+4\lambda_S)}$$

   applied to Gaussian noise
   Multishift Krylov tolerance $\mathrm{tol}_S \sim 10^{-6}$
   Cutoff $\lambda_S \sim 10^{-3}$ O(1000) fine matrix multiplies for each vector

2. Block these vectors ϕ_k^b (e.g. $4^4 \times L_s$) and compute little Dirac operator
   Need only apply $N_{\mathrm{stencil}} = 80$ matrix multiplies per vector to compute little Dirac operator with a Fourier trick
   Can *detect* stencil from matrix application and generate optimal code for 1,2,4 hop operators

3. Compute second level of deflation heirarchy using inverse iteration on Gaussian noise.

4. Diagonalise this basis to make multiplication cheap

HDCG solver

Use outer CG A-DEF2 solver, DeflCG little dirac solver

Method	V_{start}	M_1	M_2	M_3	V_{end}
A-DEF2	$Qb + P_R x$	$P_R M_{IRS} + Q$	$\mathbb{1}$	$\mathbb{1}$	x_{k+1}
DeflCG	$Qb + P_R x$	$\mathbb{1}$	$\mathbb{1}$	$(1 - P_R)$	x_{k+1}

Where

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & M_{SS}^{-1} \end{pmatrix} \quad ; \quad P_R = \begin{pmatrix} 1 & 0 \\ -M_{SS}^{-1} M_{S\bar{S}} & 0 \end{pmatrix}$$

$$\mathcal{H} = M_{\mathrm{pc}}^{\dagger} M_{\mathrm{pc}} \quad ; \quad M_{IRS} = \left[\mathcal{H} + \lambda_{\mathrm{pc}}\right]^{-1}$$

- Shifted matrix inversion M is solved with CG and fixed iteration count (N=8)
- M_{SS} inversion is itself deflated
- All operations in CG are perfromed in single precision except $\mathcal{H}$ multiply, x_j and r_j updates.

Tunable parameters

Fine N_{vec}	40
Fine blocksize	$4^4 \times L_s$
Fine subspace filter	4th order rational $\lambda_S \sim 10^{-3}$
Fine subspace tolerance	10^{-6}
Coarse N_{vec}	128
Coarse blocksize	full volume
Coarse subspace filter	Inverse iteration (3)
Coarse subspace tolerance	10^{-7}
$\left[M_{\mathrm{pc}}^{\dagger} M_{\mathrm{pc}} + \lambda_{\mathrm{pc}}\right]^{-1}$	8 iterations (tol $\sim 10^{-1}$)
λ_{pc}	1.0
M_{SS}^{-1}	tol 5×10^{-2}

1. x arbitrary
2. $x_0 = V_{\mathrm{start}}$
3. $r_0 = b - \mathcal{H} x_0$
4. $y_0 = M_1 r_0$; $p_0 = M_2 y_0$
5. for iteration k
6. $w_k = M_3 \mathcal{H} p_k$
7. $\alpha_k = (r_k, y_k)/(p_k, w_k)$
8. $x_{k+1} = x_k + \alpha_k p_k$
9. $r_{k+1} = r_k - \alpha_k w_k$
10. $\mathbf{y_k = M_1 r_k}$
11. $\beta_{\mathbf{k}} = (\mathbf{r_{k+1}}, \mathbf{y_{k+1}})/(\mathbf{r_k}, \mathbf{y_k})$
12. $\mathbf{p_{k+1} = M_2 y_{k+1} + \beta_k p_k}$
13. end for
14. $x = V_{\mathrm{end}}$

Performance

Both fine and coarse dirac operators give around 30-50Gflop/s per node on BG/Q.
On $48^3 \times 96 \times 24$, $M_\pi = 140$MeV, $a^{-1} = 1.73$ GeV on 1024 node rack

Algorithm	Tolerance		Matmuls
CGNE (double)	10^{-8}	1270s	16000
CGNE (mixed)			23000
EigCG (mixed)	10^{-8}	320s	11710
EigCG (mixed)	10^{-4}	55s	1400
EigCG (setup)		10h	
EigCG (vectors)		600 vectors	
HDCG (mixed)	10^{-8}	170s	3100
HDCG (mixed)	10^{-4}	9s	200
HDCG (setup)		1h	
HDCG (vectors)		40 vectors	

10^{-4} precision is used for the All-mode-averaging analysis

- Anticipate at least 5x speedup for RBC-UKQCD valence analysis over EigCG

Conclusions

Comparison	Gain
Exact Solve vs CGNE	7.5x
Exact Solve vs EigCG	2x
Inexact Solve vs EigCG	5x
Setup vs EigCG	10x
Footprint vs EigCG	20x

- Developed inexact deflation method to accelerating preconditioned normal equations
  Larger stencil required substantial algorithmic improvements
- Moving little Dirac operator into preconditioner gives more robust solver (10x)
- Heirarchical multi-level deflation (10x)
- IR shifted preconditioner replacement for SAP
- Preconditioned CG is to loose convergence of inner Krylov solver(s).
- No flexible algorithm was required
- Approach based in Krylov space methods, but similarities to multigrid

To do:

- Check for numerical rounding in $P_L r_j = r_j$
- Investigate numerically efficiency of additive Schwarz preconditioning [5]
  Domain decomposed preconditioner should give $2x$ Gflop/s improvement
  Greater locality $\Rightarrow$ candidate exascale algorithm

[5] suggested by Mike Clark