Deflation (Lüscher)

- assume you have a set of N orthonormal spinor fields ϕ_l
- project Dirac operator D on the subspace

$$A_{kl} = \langle \phi_k \; D \; \phi_l \rangle$$

- and construct projectors

$$P_L \psi = \psi - \sum_{k,l} D\phi_k (A^{-1})_{kl} \langle \phi_l | \psi \rangle$$
$$P_R \psi = \psi - \sum_{k,l} \phi_k (A^{-1})_{kl} \langle \phi_l | D\psi \rangle$$

- Now solve $D\psi = \eta$ in steps:
  - solve $P_L D \chi = P_L \eta$ for χ
  - construct

$$\psi = P_R \chi + \sum_{k,l} \phi_k (A^{-1})_{kl} \langle \phi_l , \eta \rangle$$

Deflation (Lüscher)

- how to obtain the subspace?
- the exact eigenvectors would require V^2 effort
- Lüscher: use inexact eigenvectos
  - Divide the lattice into N_b blocks with coordinate $\vec{a}$
  - Generate N_s random vectors ψ_l
  - make them rough eigenvectors applying approximate D^{-1}
  - generate $N_b \cdot N_s$ by restricting the ψ to the blocks

$$\phi_l^{\vec{a}} = \begin{cases} \psi_l^{\vec{b}} & \vec{a} = \vec{b} \\ 0 & \text{else} \end{cases}$$

Deflation: Free Case

- basic algorithm implemented and working (Albert, Karl, Siebren)
- 8^4 lattice, 4^4 blocks, $N_s = 24$, $\mu = 0$

κ	DFLGCR	GCR
0.1	30	37
0.11	36	56
0.12	44	106
0.124	48	158
0.125	49	

- 8^4 lattice, 4^4 blocks, $N_s = 24$, $\kappa = 0.125$

$2\kappa\mu$	DFLGCR	GCR	CG
0.0002	49	175	58

Deflation: Realistic Case

- we see a significant reduction in the GCR iteration numbers compared to CG even/odd
- however, execution time is not yet reduced compared to CG with even/odd
- We need to work on preconditioning (we have polynomial and SAP) and other code improvements (in particular single precision inner solver iterations)
- we want to include it into the HMC
- all of this work in progress, ideas welcome

New Code Features: Online Measurements

- we measure on the fly (source - sink):

$$\langle PP \rangle,\ \langle PA_0 \rangle,\ \langle PV_0 \rangle$$

- using a stochastic time-slice source (all colour, all spin)
- allows to measure:

$$m_{\rm PS},\ f_{\rm PS},\ Z_{\rm V}$$

- relevant input parameters:

```
PerfromOnlineMeasurements = yes|no
OnlineMeasurementsFreq = n
```

New Code Features: Arbitrary n_f

- code supports now arbitrary no. of pseudo fermion “monomials” (chroma)
- a monomial can represent:

$$\det(Q(\kappa)^2 + \mu_q^2),\ \frac{\det(Q(\kappa)^2 + \mu_q^2)}{\det(Q(\kappa')^2 + \mu_q'^2)},\ \det(\ P_{n,\epsilon}(Q_{\mathrm{nd}}(\kappa, \bar{\epsilon}, \bar{\mu}_q)^2)\ )$$

- no. of flavours and preconditioning can be freely combined
- integration on any available timescale possible

New Code Features: Arbitrary n_f

- relevant input parameters (example):

```
BeginMonomial DET
  Timescale = 1
  2KappaMu = 0.
  kappa = 0.125
  AcceptancePrecision =  1.e-20
  ForcePrecision = 1.e-12
  Name = det
  solver   = cg
  CSGHistory = 10
  CSGHistory2 = 10
EndMonomial
```

- available monomials are:
  `DET, DETRATIO, NDPOLY, GAUGE`

ToDo List

- further optimisation for the BG/P
- single precision Dirac operators for various architectures
- ...!?