April 21, 2010

Multi-Shifted Bi-Conjugate Gradient Stabilized

Craig Pelissier

In this note, I describe how to use the multi-shifted bi-conjugate gradient stabilized (BICGStabM) for the inversion of linear systems involving the wilson fermionic matrix for several values of κ.

In the following, we are interested in inverting the linear system

$$\left(\mathbf{1} - \kappa \not{D}\right) \mathbf{x} = \mathbf{b}. \tag{1}$$

We first have to cast it into the shifted form

$$(M + \sigma)\,\mathbf{x} = \mathbf{b} \tag{2}$$

This can be achieved with the following definitions

$$\mathbf{x}' = -\mathbf{x}/\kappa \quad \sigma = -\frac{1}{\kappa} \quad M = \not{D} \tag{3}$$

which written explicitly is

$$\left[\not{D} + (-1/\kappa)\right](-\kappa \mathbf{x}) = \mathbf{b} \tag{4}$$

Therefore, we can solve for the solutions $\mathbf{x}'$ and recover the original solution $\mathbf{x} = -\mathbf{x}'/\kappa$ by a simple rescaling.

To use the BICGStabM, we pass the most singular shifted matrix — the one that takes the most iterations. The algorithm then iterates until *this* system converges to the desired accuracy and exits. Since this is the most singular shifted matrix, all other solutions should have converged. **Note: The algorithm does not check this for you. You need to check this in your code**.

In the following, will work through a concrete example. Let's assume we have a three κ values. shifts $= \{\kappa_0, \kappa_1, \kappa_2\}$ and k_0 is the most singular one. For the wilson fermionic matrix, this corresponds to the largest value of κ. The function call is given by

```
int qcd::bicgstabm_device(
device_wilson_field &src, int noshifts, double shifts[],
device_wilson_field *sol[], double dError,
int maxiters, matmult⟨device_wilson_field⟩& pfncMatMult,
device_random_field& rnd
);
```

- define the matrix multiplication to be pfncMatMult $= \not{D} - 1/\kappa_0$

- define the remaining two shifts as $\sigma_i = 1/\kappa_0 - 1/\kappa_i \quad i = 1, 2$ so that

$$M + \sigma_i = \not{D} - 1/\kappa_0 + 1/\kappa_0 - 1/\kappa_i = \not{D} + (-1/\kappa_i) \tag{5}$$

  which is the desired result (see eq. **??**).

- set shfits = $\{\sigma_1, \sigma_2\}$.
- set noshifts = 2.
- set src = **b**.
- I set dError = 1e-10.
- pass an array of 3 pointers to device_wilson_fields.
- set the maxiters accordingly and pass an instance of a device_random_field for rnd.
- the results are stored in sol where *sol[i] is the solution for the ith κ.
- Lastly rescale the solutions accordingly.