Speeding up Domain Wall Fermion Algorithms using QCDLAB

Artan Boriçi
University of Tirana

http://phys.fshn.edu.al/borici.html

Outline of the talk

- Domain Wall Fermions: from concepts to generalisations
- An implementation issue: can we reduce the noise further?
- Prototyping Domain Wall Fermion algorithms using QCDLAB
  - Introducing QCDLAB 1.0
  - Domain Wall Fermions in QCDLAB 1.1
  - An example calculation

Domain Wall Fermions

- Kaplan's idea: separate chiralities across a 5th dimension
- For finite 5th dimension chiralities overlap:

  $\Rightarrow$ residual chiral symmetry breaking
- Shamir's lattice implementation:
  - Shifted 5D Wilson operator with mass-dependent b.c.:
    $2 - D_W^{(5)}$ and $\psi_{N_5+1} = -m\psi_1, \quad \psi_0 = -m\psi_{N_5}$
  - Light fermion: linear combination of chiral boundary fields

Domain Wall Fermion Operator

$$\mathcal{M} = \begin{pmatrix} \mathbb{1} - D_W & P_+ & & -mP_- \\ P_- & \mathbb{1} - D_W & \ddots & \\ & \ddots & \ddots & P_+ \\ -mP_+ & & P_- & \mathbb{1} - D_W \end{pmatrix}$$

4D Hamiltonian kernel: $\mathcal{H} = \gamma_5 \frac{D_w}{2 - D_W}$

Link with the Neuberger's overlap kernel: $H_W \leftrightarrow \mathcal{H}$

Idea of the Proof

Truncated Overlap Fermions: Domain Wall Fermions with H_W kernel

$$\mathcal{M} = \begin{pmatrix} \mathbb{1} - D_W & (\mathbb{1} + D_W)P_+ & & -m(\mathbb{1} + D_W)P_- \\ (\mathbb{1} + D_W)P_- & \mathbb{1} - D_W & \ddots & \\ & \ddots & \ddots & (\mathbb{1} + D_W)P_+ \\ -m(\mathbb{1} + D_W)P_+ & & (\mathbb{1} + D_W)P_- & \mathbb{1} - D_W \end{pmatrix}$$

- Generalised Hamiltonian kernels for Domain Wall Fermions: Operator-valued Moebius transforms $\frac{aD_W+b}{cD_W+d}$

Can we reduce the noise?

- Theory and practice telles that noise can slow down simulation
- Pseudofermion action of Domain Wall Fermions:

  $$S_{\text{PF}} = \left\|\mathcal{M}^{-1}\mathcal{M}_1\Phi\right\|^2 = \left\|[D^{(N_5)}]^{-1}\chi_1\right\|^2 + \sum_{i=2}^{N_5} \|C_i\chi_i\|^2$$

  - Second term can be large for fixed Φ: $C_i \sim T^i[D^{(N_5)}]^{-1}$
  - Can drive the simulation to the Aoki phase
- Solution: remove the second term by redefining

  $$S_{\text{PF}} = \left\|\varepsilon_1^T P^T \mathcal{M}_1^{-1}\mathcal{M}P\varepsilon_1\phi\right\|^2 = \left\|[D^{(N_5)}]^{-1}\phi\right\|^2 \quad \text{, with}$$

  $$P\varepsilon_1 = (P_+, 0, \ldots, 0, P_-)^T$$

  - Easy to implement! (see Lattice 2005 proceedings)

Prototyping DWF algorithms using QCDLAB

Why QCDLAB?

– Lattice codes are production codes

– eg. MILC, CHROMA, DD-HMC etc.

– Very large and not easy to learn and master

– Lack of flexibility to switch between codes

– Difficult to implement new ideas

– Prevents innovation, obstacle for studends

QCDLAB

QCDLAB is a MATLAB/OCTAVE tool for lattice QCD.

- MATLAB/OCTAVE high level lengauge and environment of technical computing.
- Multidimensional arrays as basic data structure
- Vast Build-in Blas, Lapack, Minpack, etc. libraries
- Dynamically loaded modules from other languages

⇒ QCDLAB serves as a fast *prototyping* tool.

Prototyping with QCDLAB

A prototype code is a minimal possible code which is able to test gross features of the theory and algorithms at shortest possible time and largest acceptable errors on a standard computing platform.

a. A “minimal code” should not exceed a few printed pages.

b. A “short run time” should be measured in minutes rather than hours.

c. A “standard computing platform” should not be exceedingly high.

d. “Large acceptable errors” are those which do not compromise the gross features of the theory.

   – The quenched approximation, for example, should not be considered as an acceptable approximation for light quark physics.

QCDLAB 1.0

Collection of 20 functions:

Autocorel	BiCGg5	BiCGstab	Binning	cdot5
CG	CGNE	Dirac_KS	Dirac_r	Dirac_W
FOM	Force_KS	Force_W	GMRES	HMC_KS
HMC_W	Lanczos	SCG	SUMR	wloop

Where to download: http://phys.fshn.edu.al/qcdlab.html

Licensing:

QCDLAB 1.0 is a free software licensed under the terms of the GNU General Public License.

QCDLAB 1.0

- General functions:
  - Solvers: `BiCGg5, BiCGstab, CG, CGNE, FOM, GMRES, Lanczos, SCG, SUMR`
  - Data processing: `Autocorel, Binning`.
- Specialised functions for the Schwinger model:
  - Simulation: `HMC_W, HMC_KS, Force_W, Force_KS`
  - Operators: `Dirac_KS, Dirac_r, Dirac_W, cdot5`
  - Measurements: `wloop`

Getting started with QCDLAB 1.0

- Get a taste running three projects:
  - For Matlab: `MProject1, MProject2, MProject3`
  - For Octave: `OProject1, OProject2, OProject3`
  - Put scripts in the working directory and type the names
- Project1: Simulation project
- Project2: Linear system and eigenvalue solver project
- Project3: Linear system and eigenvalue solver project for chiral fermions

QCDLAB 1.1

Extended functionality:

4D Wilson operators:

`Initialise_Dirac_W.m Mult_Dirac_W.m Mult_Dirac_W_H.m Dirac4.m`

Gamma5 operators:

`mult_gamma5.m P5plus.m  P5minus.m`

Domain Wall Fermion operators:

`Mult_DWF_H.m Mult_DWF.m`

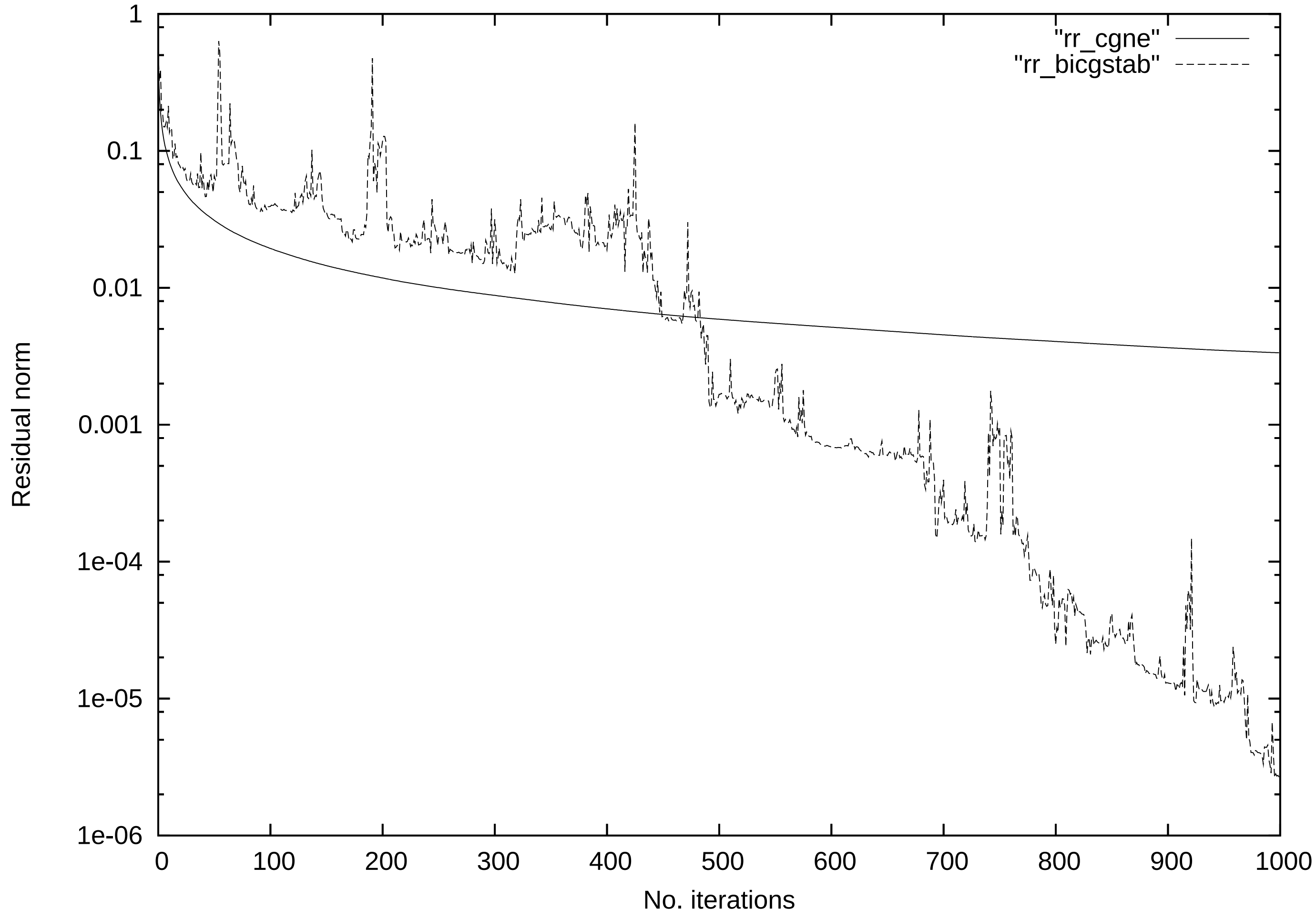
Egenvalue solvers:

`PowerMethod.m InversePower.m`

A Domain Wall Fermion Project using QCDLAB

```
Initialise_Dirac_W.m;
b=rand(12*N*N5,1)+i*rand(12*N*N5,1);
x0=rand(12*N*N5,1)+i*rand(12*N*N5,1);
tol=1e-6; nmax=1000;
[x,rr] = cgne(b,x0,tol,nmax);
```

The Result


"rr_cgne"
"rr_bicgstab"
Residual norm
1
0.1
0.01
0.001
1e-04
1e-05
1e-06
0
100
200
300
400
500
600
700
800
900
1000
No. iterations

The Domain Wall Fermion Operator in QCDLAB

```
function y=Mult_DWF(x,N5);
% Multiplies a vector by the Domain Wall Fermion matrix
global N mass_dwf
x=reshape(x,12*N,N5);
y(:,1)=x(:,1)-Mult_Dirac_W(x(:,1))+P5plus(x(:,2))-mass_dwf*P5minus(x(:,N5));
for j5=2:N5-1;
  y(:,j5)=x(:,j5)-Mult_Dirac_W(x(:,j5))+P5plus(x(:,j5+1))+P5minus(x(:,j5-1)
end
y(:,N5)=x(:,N5)-Mult_Dirac_W(x(:,N5))-mass_dwf*P5plus(x(:,1))+P5minus(x(:,N5
x=reshape(x,12*N*N5,1);
y=reshape(y,12*N*N5,1);
% Copyright (C) 2006-2007 Artan Borici.
% This program is a free software licensed under the terms of the GNU Genera
```

Running QCDLAB in parallel

Options:

- Matlab Distributed Engine → Expensive
- Octave and multi-threading using MPITB
  - Ready to use within the ParallelKnopix bootable DVD
  - Boot the cluster, run, store results and shutdown
  - Downloadable from University of Barcelona
- Octave and dynamically linked scalable BLAS libraries