Fueling BSM Theoretical Development with Lua

Meifeng Lin
Computational Science Center
Brookhaven National Laboratory

Lattice Meets Experiment Workshop
December 5-6, BNL

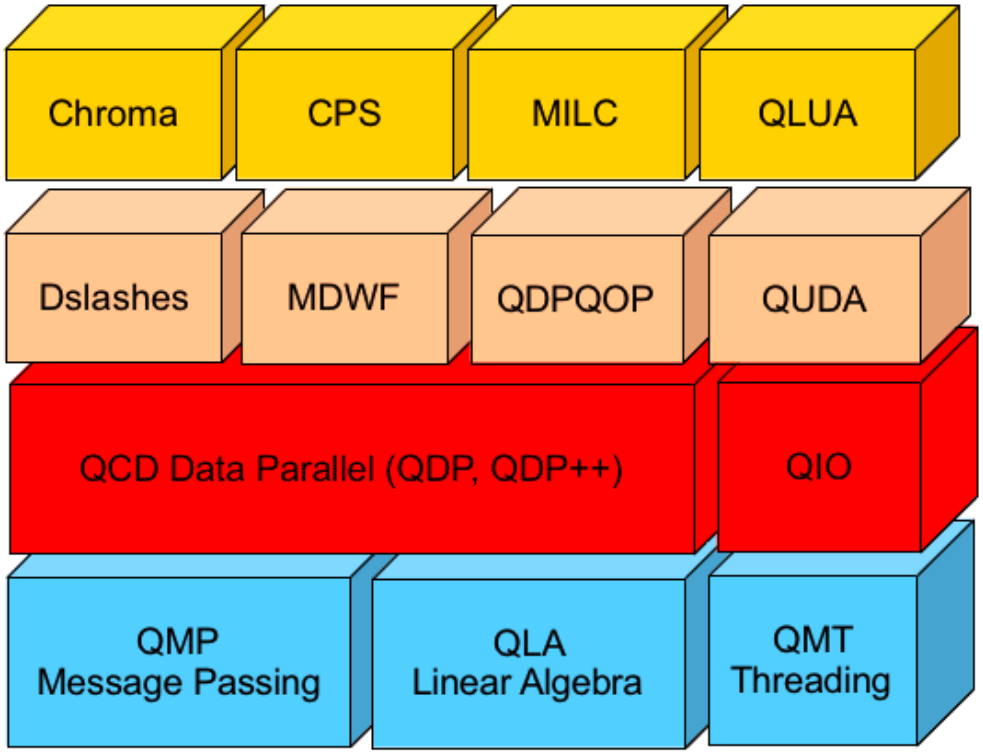
BROOKHAVEN NATIONAL LABORATORY
a passion for discovery

U.S. DEPARTMENT OF ENERGY | Office of Science

Outline

- FUEL – A dedicated software package for BSM lattice simulations
- Other USQCD BSM Software

USQCD SciDAC Software



Applications

Level 3

Level 2

Level 1

Need for dedicated BSM software

- Lattice QCD applications are well developed and tested.
- Most of the applications were written specifically with QCD in mind.
- Not easy to change to accommodate the needs of Lattice BSM simulations.
- Need an application suite that can be easily adopted to test new ideas and aid theoretical development.

- Enters FUEL.
- Funded under SciDAC-3

FUEL Framework for Unified Evolution of Lattices

Lead developer: James Osborn (ALCF)
Contributors: ML (BNL), Evan Weinberg (BU)

- Uses Lua scripts as wrappers for SciDAC Level-3 libraries
- Currently supports qopqdp
- Focused on gauge field generation
- Has basic measurement capabilities
- Designed to be light-weight, portable and extensible

- Under active development. Changes on daily basis.
- Has been used in the USBSM 8-flavor staggered simulations.

The Lua programming language

- A scripting language developed in 1990s.
- “Lua” means “Moon” in Portuguese.
- Designed to be fast, portable, embeddable and extensible.
- Has been used in many industrial applications.
- Is the leading scripting language in games
  - World of Warcraft and Angry Birds, for example.

- Comes pre-packaged with FUEL.

FUEL Status

- Gauge action:
  - Plaquette, plaquette-adjoint
  - Symanzik-improved
  - Iwasaki, DBW2
- Fermion action:
  - Staggered (nHYP, Asqtad, HISQ, Stout)
  - Unimproved Wilson (incl. stout smearing)
  - Clover (solver only, no force term)
- Arbitrary Nc, Nf; fundamental representation only.
- Nc can be set at compile time or runtime.

FUEL Status

- Supports most HMC evolution algorithms
  - Hasenbusch mass preconditioning
  - RHMC for staggered
  - Omelyan integrators
  - Anisotropy
  - Multigrid (in progress, ML)
- HMC integrator, parameters etc. set in Lua scripts.
- Easy to tune the HMC algorithms

Current FUEL Status

- Has simple observables
  - Plaquette,
  - Wilson loop, Wilson flow
  - Polyakov loop
  - Chiral condensate
  - Meson spectrum for Wilson fermions
  - Meson spectrum for staggered ongoing (Evan Weinberg)

SU(Nc) in FUEL

- Compile time: can choose specific optimized code for Nc=1,2,3
- Nc > 3 uses generic implementation.

To build the library for SU(2) simulations, set:

```
COLORLIB = 2
NC = 2
```

To build the library for SU(3) simulations, set:

```
COLORLIB = 3
NC = 3
```

For other SU(N) simulations, set:

```
COLORLIB = n
NC = <N>
```

where *<N>* is a numeric value, e.g., 4.

Example: SU(Nc) in FUEL

- Currently dynamic runtime uses generic Nc implementation.
- Performance not optimal.
- Will hook individual Nc=1,2,3 implementations.
- Also supports multiple gauge fields with different Nc.

```
require 'Lattice'
require 'Action'
require 'Evolver'

-- set a lattice geometry
L = Lattice{4,4,4,8}

-- set a random number seed
L:Seed(987654321)

-- define the gauge group. SU(3) here.
G = L:GaugeField{group="SU",nc=3}

-- set the gauge links to unity
G:Set("unit")

-- or load an existing lattice
-- G:Load("lattice")

-- set the gauge action and coupling
GA = Action{kind="gauge",style="plaquette",beta=6,field=G}

-- get the conjugate momentum
M = G:Momentum()
MA = Action{kind="momentum",momentum=M}

-- set the HMC integrator
I = Evolver{kind="md",
            style="leapfrog",
            action=GA,
            field=G,
            momentum=M,
            tau=1,
            nSteps=40}

-- start the MC Markov chain
E = Evolver{kind="mc",
            markov=I,
            actions={MA,GA},
            fields={G}}

printf("action: %g\n", GA:Action())
E:Run()
printf("action: %g\n", GA:Action())
myprint(E.oldActions,"\n")
myprint(E.newActions,"\n")
myprint(E.mcRand,"\n")
```

FUEL Future Plans

- BSM-related actions (higher representations)
- Clover HMC
- Domain wall fermions
- Integration with other Level-3 libraries
  - QUDA, MDWF, …
- Integration with Qlua?

Other BSM Software

- Chroma
  - Developed mainly for QCD.
  - Can be modified to do Nc !=3 calculations.
- Qlua
  - Uses Lua scripts to wrap SciDAC libraries, similar to FUEL.
  - Supports arbitrary Nc.
- SUSY Lattice
  - Dedicated SUSY lattice code

Chroma

- Builds on QDP++
- Nc = 2, 4 tested and used in production with Wilson fermions
- Supports two- and three-point contractions with Nc=2 and 4.
- Supports Schrodinger functional calculations.
- Fundamental representation only.

- BSM Contributors:

  George Fleming, Ethan Neil, Gennady Voronov

Qlua

Lead developers: Andrew Pochinsky, James Osborn, Sergey Syritsyn

- Uses Lua to provide interface to the SciDAC libraries.
- Each lattice object has its own value of Nc.
- Has checking in place: watches for violations of rules of arithmetic, e.g., it does not allow users to multiply Nc=4 matrix to Nc=7 fermion.
- Users can write their own applications using its interface to qdp, qla and qopqdp libraries.

Courtesy of Andrew Pochinsky

SUSY LATTICE

- Historically based on MILC code.
- Current parallel code focuses on four-dimensional *N=4* Supersymmetric Yang-Mills theory
- Includes an RHMC and various related measurements.
  - Basic gauge quantities (plaquette, Polyakov loop, plaquette determinant and link trace)
  - Wilson loops for the static potential
  - "Standard" supersymmetric observables: the Konishi correlator and supergravity correlator
  - Fermion bilinears to explore SUSY breaking
  - Connected mesons
  - …
- Under active development.

Courtesy of David Schaich

Links

- Source code
  - http://usqcd.jlab.org/usqcd-software/index.html (USQCD Software)
  - http://lattice.bu.edu/~josborn/fuel/ (FUEL)
  - https://usqcd.lns.mit.edu/redmine/projects/qlua (Qlua)
  - https://www.assembla.com/code/smilc/subversion/nodes (SUSY)

Summary

- FUEL is designed to be a flexible framework for BSM lattice simulations.
- Supports arbitrary Nc for SU(Nc) gauge theories.
- Provides an easy way to tune parameters in HMC.
- More features are being constantly added.
- Users/testers are welcome.

- Many thanks to

  Ethan Neil, James Osborn, Andrew Pochinsky, David Schaich, Gennady Voronov