Algorithmic Challenges for Lattice Field Theory in the Multi-scale Era

SciDAC-3 PI Meeting
DOE July 24, 2013
Rich Brower Boston University

*Narrative**

- Algorithms are the critical multi-disciplinary pursuit
  - Physics + Applied Math + Computer Science ➔ Algorithms designers
- Need “Heroic” programming just to keep up
  - MB/Q,Titan, BlueWaters, Stampede ➔ CUDA, JIT, SUPER
- Multi-grid QCD success story: why did it takes so long?
  - MG, DD,, FastMATH, Hyper, Qlua, FUEL
- Future: New theories, New lattices, Much more Multi-level
  - Condensed Matter, Graphene, Radial Quantization, FEM, etc.

Lattice Field Theory Coming of Age

Kenneth G. Wilson "Confinement of quarks"
Phys. Rev. D 10, 2445–2459 (1974)

K. Wilson: "Lecture at Lattice 1989 Capri"

"lattice gauge theory could also require a 10^8 increase in computer power AND spectacular algorithmic advances before useful interactions with experiment ...

- ab initio Chemistry

1. 1930+50 = 1980
2. 0.1 flops ➔ 10 Mflops
3. Gaussian Basis functions

vs

- ab initio QCD

1. 1980 + 50 = 2030?*
2. 10 Mflops ➔ 1000 Tflops
3. Clever Collective Variable?

Sustained Petaflops: 15 years ahead of schedule!

Expanding Physics Goals of Lattice Gauge Theory

- Ab initio QCD for Nuclear/Astrophysics
  - I. Structure and excitation of Nuclei (QCD)
  - II. Quark Gluon Plasma (QCD)

Jefferson Lab

RHIC

CMS

CERN

- Beyond the Standard Model for High Energy Physics :
  - III. QCD for precision tests of Standard Model
    (aka "Intensity Frontier")
  - IV. Explore new Theories (not QCD) at TeV
    (aka "Energy Frontier")

ATLAS EXPERIMENT

Fermilab

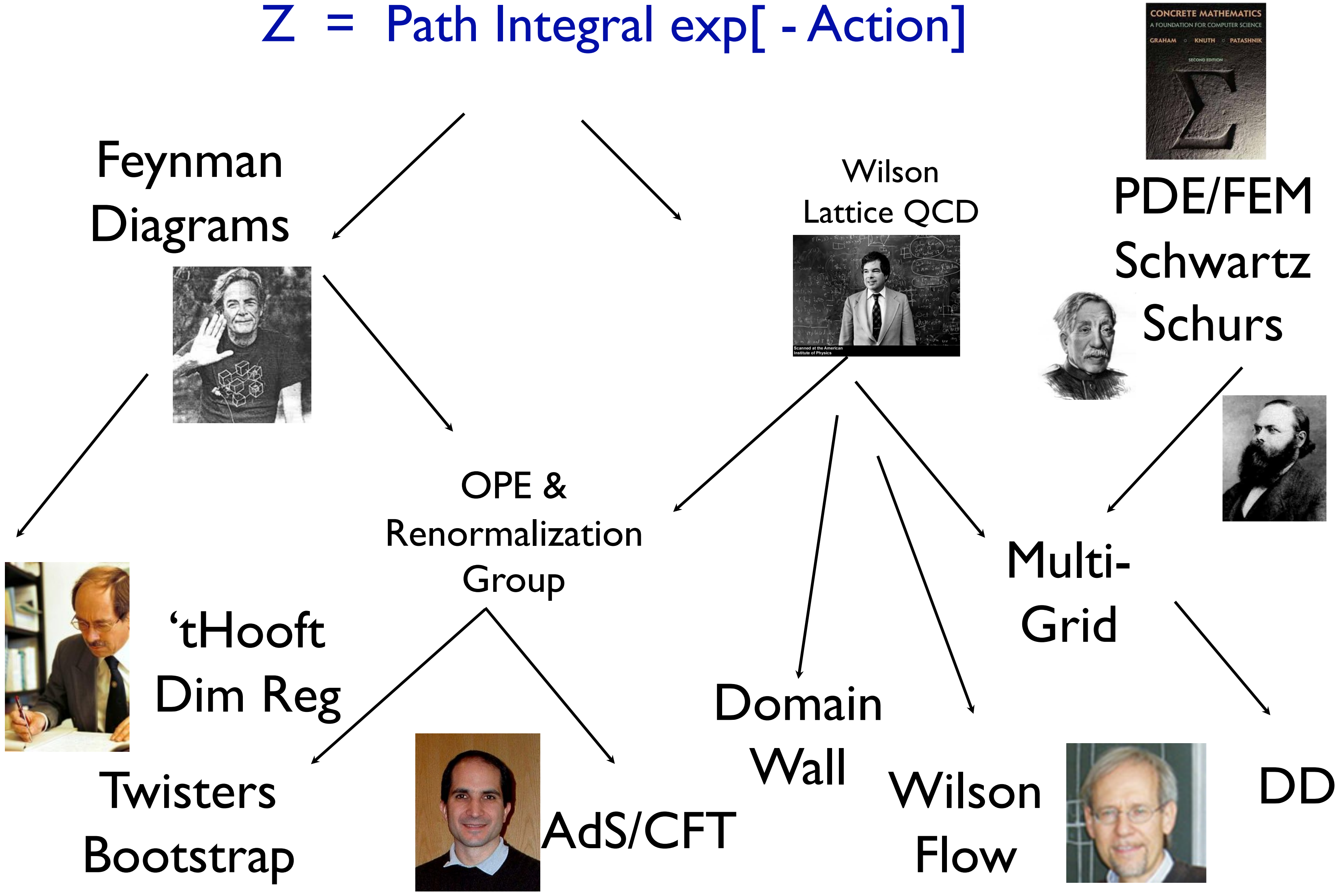
Belle II

LHCb

- Strongly coupled quantum systems for Novel Materials

All prediction from Quantum Field Theory require "Algorithms"

Z = Path Integral exp[- Action]



Feynman
Diagrams

'tHooft
Dim Reg

OPE &
Renormalization
Group

Twisters
Bootstrap

AdS/CFT

Wilson
Lattice QCD

Domain
Wall

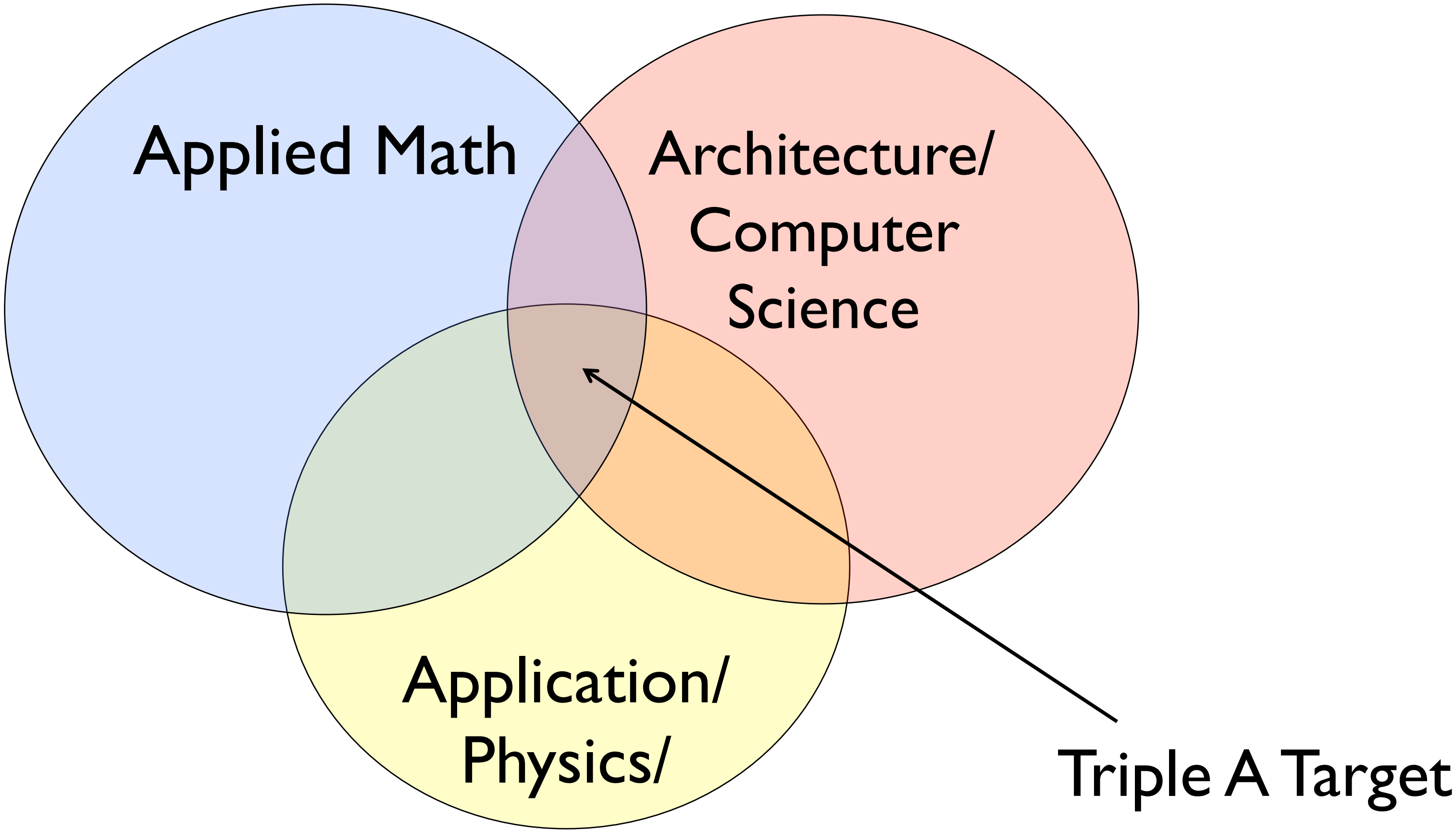
Wilson
Flow

Multi-
Grid

DD

PDE/FEM
Schwartz
Schurs

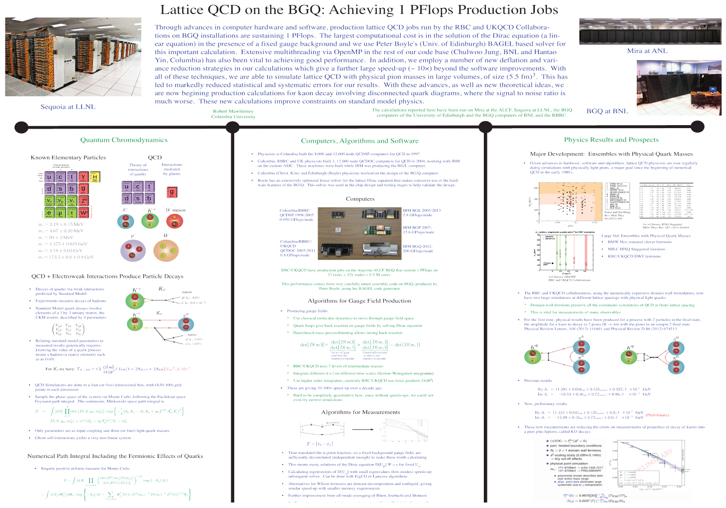
Intersection of Algorithm Design



CHALLENGE: Need to collaborate between 3 disciplines BUT also need suitable abstraction/interfaces so each discipline can proceed semi-autonomously!

See posters for Details

BG-Q/IBM



Lattice QCD on the BGQ: Achieving 1 PFlops Production Jobs

Through advances in computer hardware and software, production lattice QCD jobs run by the RBC and UKQCD Collaborations on BGQ installations are sustaining 1 PFlops. The largest computational cost is in the solution of the Dirac equation (a linear equation) in the presence of a fixed gauge background and we use Peter Boyle's (Univ. of Edinburgh) BAGEL based solver for this important calculation. Extensive multithreading via OpenMP in the rest of our code base (Chulwoo Jung, BNL and Hantao Yin, Columbia) has also been vital to achieving good performance. In addition, we employ a number of new deflation and variance reduction strategies in our calculations which give a further large speed-up (~ 10×) beyond the software improvements. With all of these techniques, we are able to simulate lattice QCD with physical pion masses in large volumes, of size $(5.5\ \text{fm})^3$. This has led to markedly reduced statistical and systematic errors for our results. With these advances, as well as new theoretical ideas, we are now begining production calculations for kaon decay involving disconnected quark diagrams, where the signal to noise ratio is much worse. These new calculations improve constraints on standard model physics.

Sequoia at LLNL

Mira at ANL

BGQ at BNL

Robert Mawhinney
Columbia University

The calculations reported here have been run on Mira at the ALCF, Sequoia at LLNL, the BGQ computers of the University of Edinburgh and the BGQ computers of BNL and the RBRC.

Quantum Chromodynamics

Known Elementary Particles

QCD

Computers, Algorithms and Software

- Physicists at Columbia built the 8,000 and 12,000 node QCDSP computers for QCD in 1997
- Columbia, RBRC and UK physicists built 3, 12,000 node QCDOC computers for QCD in 2004, working with IBM on the custom ASIC. These machines were built while IBM was producing the BG/L computer.
- Columbia (Christ, Kim) and Edinburgh (Boyle) physicists worked on the design of the BG/Q computer.
- Boyle has an extensively optimized linear solver for the lattice Dirac equation that makes extensive use of the hardware features of the BG/Q. This solver was used in the chip design and testing stages to help validate the design.

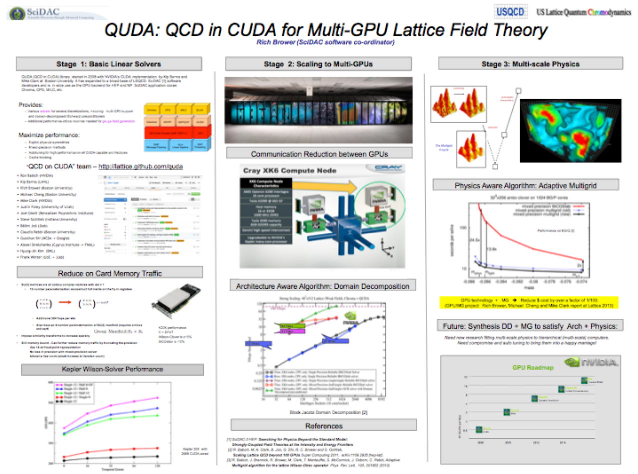
Computers

Physics Results and Prospects

Major Development: Ensembles with Physical Quark Masses

- Given advances in hardware, software and algorithms, lattice QCD physicists are now regularly doing simulations with physically light pions, a major goal since the beginning of numerical QCD in the early 1980's.

GPU/NVIDIA



QUDA: QCD in CUDA for Multi-GPU Lattice Field Theory

Rich Brower (SciDAC software co-ordinator)

SciDAC

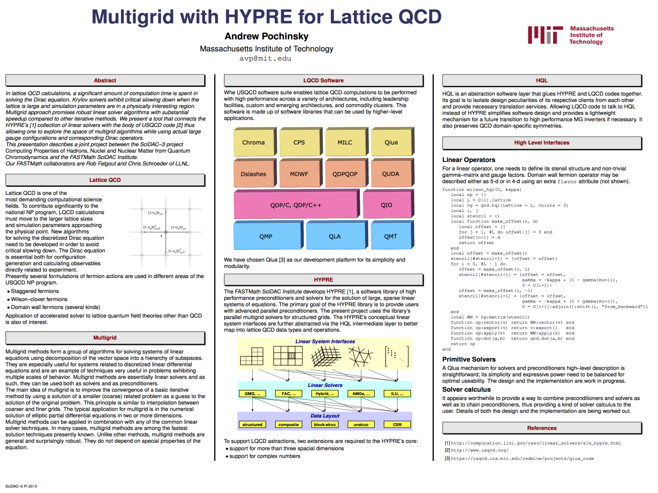
USQCD US Lattice Quantum Chromodynamics

Stage 1: Basic Linear Solvers

Stage 2: Scaling to Multi-GPUs

Stage 3: Multi-scale Physics

MG/fastMATH



Multigrid with HYPRE for Lattice QCD

Andrew Pochinsky
Massachusetts Institute of Technology
avp@mit.edu

Massachusetts Institute of Technology

Abstract

In lattice QCD calculations, a significant amount of computation time is spent in solving the Dirac equation. Krylov solvers exhibit critical slowing down when the lattice is large and simulation parameters are in a physically interesting region. Multigrid approach promises robust linear solver algorithms with substantial speedup compared to other iterative methods. We present a tool that connects the HYPRE's [1] collection of linear solvers with the body of USQCD code [2] thus allowing one to explore the space of multigrid algorithms while using actual large gauge configurations and corresponding Dirac operators.
This presentation describes a joint project between the SciDAC–3 project Computing Properties of Hadrons, Nuclei and Nuclear Matter from Quantum Chromodynamics *and the FASTMath SciDAC Institute.*
Our FASTMath collaborators are Rob Falgout and Chris Schroeder of LLNL.

Lattice QCD

Lattice QCD is one of the most demanding computational science fields. To contribute significantly to the national NP program, LQCD calculations must move to the larger lattice sizes and simulation parameters approaching the physical point. New algorithms for solving the discretized Dirac equation need to be developed in order to avoid critical slowing down. The Dirac equation is essential both for configuration generation and calculating observables directly related to experiment.
Presently several formulations of fermion actions are used in different areas of the USQCD NP program.

- Staggered fermions
- Wilson–clover fermions
- Domain wall fermions (several kinds)

Application of accelerated solver to lattice quantum field theories other than QCD is also of interest.

Multigrid

Multigrid methods form a group of algorithms for solving systems of linear equations using decomposition of the vector space into a hierarchy of subspaces. They are especially useful for systems related to discretized linear differential equations and are an example of techniques very useful in problems exhibiting multiple scales of behavior. Multigrid methods are essentially linear solvers and as such, they can be used both as solvers and as preconditioners.
The main idea of multigrid is to improve the convergence of a basic iterative method by using a solution of a smaller (coarse) related problem as a guess to the solution of the original problem. This principle is similar to interpolation between coarser and finer grids. The typical application for multigrid is in the numerical solution of elliptic partial differential equations in two or more dimensions.
Multigrid methods can be applied in combination with any of the common linear solver techniques. In many cases, multigrid methods are among the fastest solution techniques presently known. Unlike other methods, multigrid methods are general and surprisingly robust. They do not depend on special properties of the equation.

LQCD Software

Whe USQCD software suite enables lattice QCD computations to be performed with high performance across a variety of architectures, including leadership facilities, custom and emerging architectures, and commodity clusters. This software is made up of software libraries that can be used by higher–level applications.

We have chosen Qlua [3] as our development platform for its simplicity and modularity.

HYPRE

The FASTMath SciDAC Institute develops HYPRE [1], a software library of high performance preconditioners and solvers for the solution of large, sparse linear systems of equations. The primary goal of the HYPRE library is to provide users with advanced parallel preconditioners. The present project uses the library's parallel multigrid solvers for structured grids. The HYPRE's conceptual linear system interfaces are further abstracted via the HQL intermediate layer to better map into lattice QCD data types and operations.

To support LQCD astractions, two extensions are required to the HYPRE's core:

- support for more than three spacial dimensions
- support for complex numbers

HQL

HQL is an abstraction software layer that glues HYPRE and LQCD codes together. Its goal is to isolate design peculiarities of its respective clients from each other and provide necessary translation services. Allowing LQCD code to talk to HQL instead of HYPRE simplifies software design and provides a lightweight mechanism for a future transition to high performance MG inverters if necessary. It also preserves QCD domain-specific symmetries.

High Level Interfaces

Linear Operators

For a linear operator, one needs to define its stensil structure and non-trivial gamma–matrix and gauge factors. Domain wall fermion operator may be described either as 5-d or in 4-d using an extra `flavor` attribute (not shown).

```
function wilson_hql(U, kappa)
   local op = {}
   local L = U[1].lattice
   local hg = qcd.hql{Lattice = L, Colors = 3}
   local i, j
   local stencil = {}
   local function make_offset(n, d)
      local offset = {}
      for j = 1, #L do offset[j] = 0 end
      offset[n+1] = d
      return offset
   end
   local offset = make_offset()
   stencil[#stencil+1] = {offset = offset}
   for i = 0, #L - 1 do
      offset = make_offset(i, 1)
      stencil[#stencil+1] = {offset = offset,
                             gamma = -kappa * (1 - gamma{mu=i}),
                             U = U[i+1]}
      offset = make_offset(i, -1)
      stencil[#stencil+1] = {offset = offset,
                             gamma = -kappa * (1 + gamma{mu=i}),
                             U = U[i+1]:adjoin():shift(i, "from_backward")}
   end
   local WM = hg:matrix(stencil)
   function op:vector(v)  return WM:vector(v)  end
   function op:export(v)  return v:export()   end
   function op:apply(v)   return WM:apply(v)   end
   function op:dot(a,b)   return qcd.dot(a,b) end
   return op
end
```

Primitive Solvers

A Qlua mechanism for solvers and preconditioners high–level description is straightforward; its simplicity and expressive power need to be balanced for optimal useability. The design and the implementation are work in progress.

Solver calculus

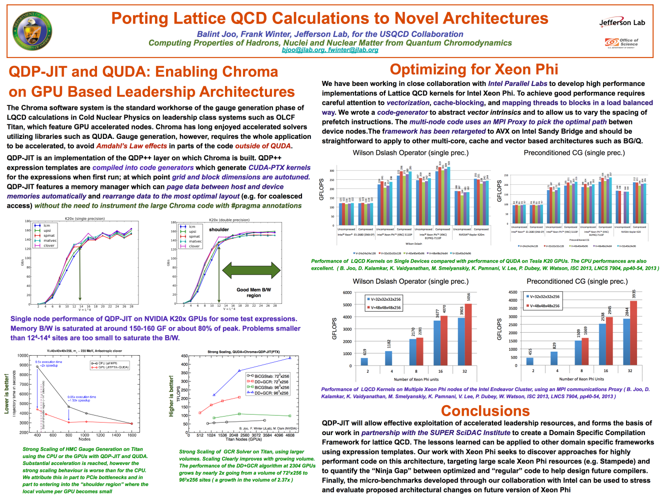
It appears worthwhile to provide a way to combine preconditioners and solvers as well as to chain preconditioners, thus providing a kind of solver calculus to the user. Details of both the design and the implementation are being worked out.

References

[1] http://computation.llnl.gov/casc/linear_solvers/sls_hypre.html
[2] http://www.usqcd.org/
[3] https://usqcd.lns.mit.edu/redmine/projects/qlua_code

SciDAC–3 PI 2013

Titan/Intel-Phi/SUPER



Porting Lattice QCD Calculations to Novel Architectures

Balint Joo, Frank Winter, Jefferson Lab, for the USQCD Collaboration
Computing Properties of Hadrons, Nuclei and Nuclear Matter from Quantum Chromodynamics
bjoo@jlab.org, fwinter@jlab.org

Jefferson Lab

Office of Science

QDP-JIT and QUDA: Enabling Chroma on GPU Based Leadership Architectures

The Chroma software system is the standard workhorse of the gauge generation phase of LQCD calculations in Cold Nuclear Physics on leadership class systems such as OLCF Titan, which feature GPU accelerated nodes. Chroma has long enjoyed accelerated solvers utilizing libraries such as QUDA. Gauge generation, however, requires the whole application to be accelerated, to avoid *Amdahl's Law effects* in parts of the code *outside of QUDA*.

QDP-JIT is an implementation of the QDP++ layer on which Chroma is built. QDP++ expression templates are *compiled into code generators* which generate *CUDA-PTX kernels* for the expressions when first run; at which point *grid and block dimensions are autotuned.* QDP-JIT features a memory manager which can *page data between host and device memories automatically* and *rearrange data to the most optimal layout* (e.g. for coalesced access) *without the need to instrument the large Chroma code with #pragma annotations*

Single node performance of QDP-JIT on NVIDIA K20x GPUs for some test expressions. Memory B/W is saturated at around 150-160 GF or about 80% of peak. Problems smaller than 12^4-14^4 sites are too small to saturate the B/W.

Strong Scaling of HMC Gauge Generation on Titan using the CPU or the GPUs with QDP-JIT and QUDA. Substantial acceleration is reached, however the strong scaling behaviour is worse than for the CPU. We attribute this in part to PCIe bottlenecks and in part to entering into the "shoulder region" where the local volume per GPU becomes small

Strong Scaling of GCR Solver on Titan, using larger volumes. Scaling Clearly improves with growing volume. The performance of the DD+GCR algorithm at 2304 GPUs grows by nearly 2x going from a volume of 72^3x256 to 96^3x256 sites (a growth in the volume of 2.37x)

Optimizing for Xeon Phi

We have been working in close collaboration with *Intel Parallel Labs* to develop high performance implementations of Lattice QCD kernels for Intel Xeon Phi. To achieve good performance requires careful attention to *vectorization*, cache-blocking, and mapping threads to blocks in a load balanced way. We wrote a *code-generator* to abstract *vector intrinsics* and to allow us to vary the spacing of prefetch instructions. The *multi-node code uses an MPI Proxy to pick the optimal path* betwen device nodes.The f*ramework has been retargeted* to AVX on Intel Sandy Bridge and should be straightforward to apply to other multi-core, cache and vector based architectures such as BG/Q.

Performance of LQCD Kernels on Single Devices compared with performance of QUDA on Tesla K20 GPUs. The CPU performances are also excellent. (B. Joo, D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnani, V. Lee, P. Dubey, W. Watson, ISC 2013, LNCS 7904, pp40-54, 2013)

Performance of LQCD Kernels on Multiple Xeon Phi nodes of the Intel Endeavor Cluster, using an MPI communications Proxy (B. Joo, D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnani, V. Lee, P. Dubey, W. Watson, ISC 2013, LNCS 7904, pp40-54, 2013)

Conclusions

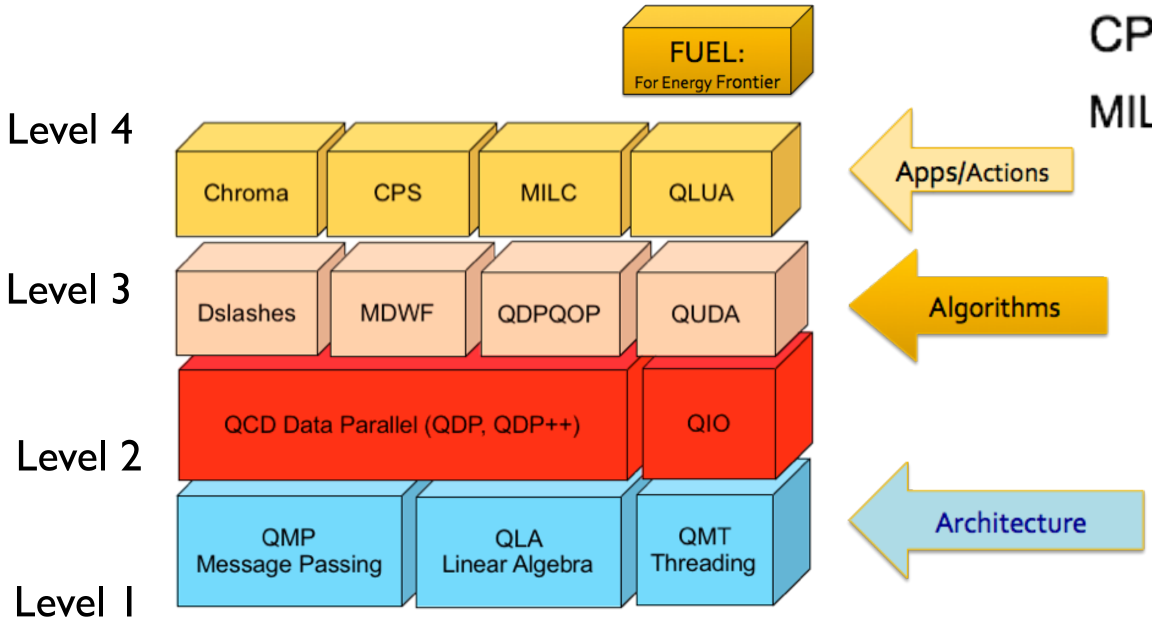
QDP-JIT will allow effective exploitation of accelerated leadership resources, and forms the basis of our work in *partnership with the SUPER SciDAC Institute* to create a Domain Specific Compilation Framework for lattice QCD. The lessons learned can be applied to other domain specific frameworks using expression templates. Our work with Xeon Phi seeks to discover approaches for highly performant code on this architecture, targeting large scale Xeon Phi resources (e.g. Stampede) and to quantify the "Ninja Gap" between optimized and "regular" code to help design future compilers. Finally, the micro-benchmarks developed through our collaboration with Intel can be used to stress and evaluate proposed architectural changes on future version of Xeon Phi

Lattice Gauge Theory HEP and NP posters

1. Lattice QCD on the BGQ:Achieving 1 PFlops Production Jobs
    - IBM/Columbia (Bob Mawhinney)
2. Multigrid with Hyper for Lattice QCD
    - FastMath/MIT (Rob Faulgot,Andrew Pochinsky)
3. Porting Lattice QCD Calculation to Novel Architectures
    - SUPER/Jlab (Rob Fower/Balint Joo)
4. QUDA: QCD in CUDA fore Multi-GPU Lattice Field Theory
    - NVIDIA/BU (Mike Clark/Rich Brower)

USQCD Software Stack Stack

On line distribution: http://usqcd.jlab.org/usqcd-software/



Chroma = 4856 files
CPS = 1749 files
MILC = 2300 files

} Apps Software
not
SciDAC funded

QUDA/python = 221 files

QLA/perl = 23000 files

The application codes
Chroma/CPS/MILC and a new QDP LUA
code base provide a rich set of tools.

SciDAC LGT contributors

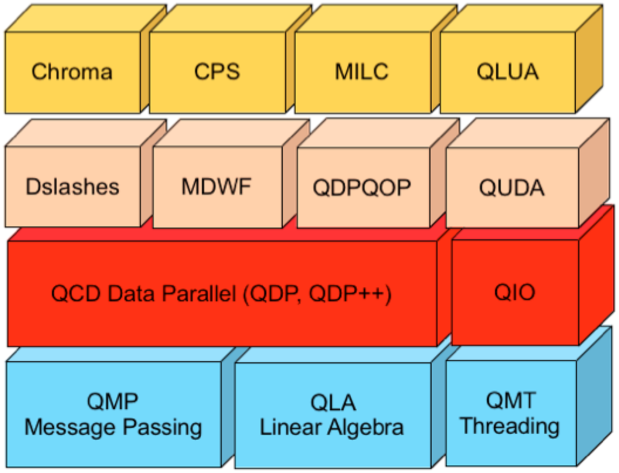
- ANL: James Osborn, Meifeng Lin, Heechang Na, (George T. Fleming)
- BNL: Frithjof Karsch, Chulwoo Jung, Hyung-Jin Kim,Yu Maezawa
- Columbia: Robert Mawhinney, Hantao Yin
- FNAL: James Simone, Alexei Strelchenko, Don Holmgren, Paul Mackenzie
- JLab: Robert Edwards, Balint Joo, Jie Chen, Frank Winter, Chip Watson
- W&M/UNC: Kostas Orginos, Andreas Stathopoulos, Rob Fowler (SUPER)
- LLNL: Pavlos Vranas, Chris Schroeder, Rob Faulgot (FASTmath)
- NVIDIA: Ron Babich, Mike Clark
- Arizona: Doug Toussaint, Alexei Bazavov
- Indiana/NCSA: Steve Gottlieb, Ran Zhou
- Utah: Carleton DeTar, Justin Foley
- BU: Richard Brower, Michael Cheng, Oliver Witzel
- MIT: Pochinsky Andrew, John Negele,
- Syracuse: Simon Catterall, (David Schaich in fall)
- Washington: Martin Savage, Saul Cohen
- Others: Peter Boyle, Jim Hetrick, Massimo Di Pierro, Patrick Dreher, et al
- “Team of Rivals” (apologies to contributors and projects *NOT* mentioned in 6 slides!)

Highest Priority is moving to 3 new architecture!

Argonne NATIONAL LABORATORY IBM

TACC

intel



Chroma | CPS | MILC | QLUA

Dslashes | MDWF | QDPQOP | QUDA

QCD Data Parallel (QDP, QDP++) | QIO

QMP Message Passing | QLA Linear Algebra | QMT Threading

OAK RIDGE National Laboratory

NVIDIA

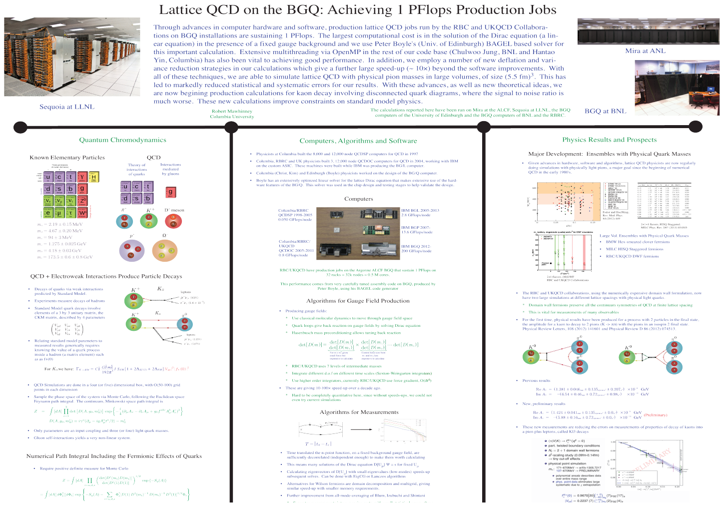
NCSA

CRAY X1

(May you live in Interesting Times!)

See posters for more details

BG-Q/IBM



Lattice QCD on the BGQ: Achieving 1 PFlops Production Jobs

Through advances in computer hardware and software, production lattice QCD jobs run by the RBC and UKQCD Collaborations on BGQ installations are sustaining 1 PFlops. The largest computational cost is in the solution of the Dirac equation (a linear equation) in the presence of a fixed gauge background and we use Peter Boyle's (Univ. of Edinburgh) BAGEL based solver for this important calculation. Extensive multithreading via OpenMP in the rest of our code base (Chulwoo Jung, BNL and Hantao Yin, Columbia) has also been vital to achieving good performance. In addition, we employ a number of new deflation and variance reduction strategies in our calculations which give a further large speed-up (~ 10×) beyond the software improvements. With all of these techniques, we are able to simulate lattice QCD with physical pion masses in large volumes, of size $(5.5\ \text{fm})^3$. This has led to markedly reduced statistical and systematic errors for our results. With these advances, as well as new theoretical ideas, we are now begining production calculations for kaon decay involving disconnected quark diagrams, where the signal to noise ratio is much worse. These new calculations improve constraints on standard model physics.

Sequoia at LLNL

Mira at ANL

BGQ at BNL

Robert Mawhinney
Columbia University

The calculations reported here have been run on Mira at the ALCF, Sequoia at LLNL, the BGQ computers of the University of Edinburgh and the BGQ computers of BNL and the RBRC.

Quantum Chromodynamics

Known Elementary Particles

QCD

Theory of interactions of quarks

Interactions mediated by gluons

$m_u = 2.19 \pm 0.15$ MeV
$m_d = 4.67 \pm 0.20$ MeV
$m_s = 94 \pm 3$ MeV
$m_c = 1.275 \pm 0.025$ GeV
$m_b = 4.18 \pm 0.03$ GeV
$m_t = 173.5 \pm 0.6 \pm 0.8$ GeV

QCD + Electroweak Interactions Produce Particle Decays

- Decays of quarks via weak interactions predicted by Standard Model.
- Experiments measure decays of hadrons
- Standard Model quark decays involve elements of a 3 by 3 unitary matrix, the CKM matrix, described by 4 parameters

$$\begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}$$

- Relating standard model parameters to measured results generically requires knowing the value of a quark process inside a hadron (a matrix element) such as as f+(0)

- QCD Simulations are done in a four (or five) dimensional box, with O(50-100) grid points in each dimension
- Sample the phase space of the system via Monte Carlo, following the Euclidean space Feynamn path integral. The continuum, Minkowski space path integral is
- Only parameters are an input coupling and three (or four) light quark masses.
- Gluon self-interactions yields a very non-linear system.

Numerical Path Integral Including the Fermionic Effects of Quarks

- Require positive definite measure for Monte Carlo

Computers, Algorithms and Software

- Physicists at Columbia built the 8,000 and 12,000 node QCDSP computers for QCD in 1997
- Columbia, RBRC and UK physicists built 3, 12,000 node QCDOC computers for QCD in 2004, working with IBM on the custom ASIC. These machines were built while IBM was producing the BG/L computer.
- Columbia (Christ, Kim) and Edinburgh (Boyle) physicists worked on the design of the BG/Q computer.
- Boyle has an extensively optimzed linear solver for the lattice Dirac equation that makes extensive use of the hardware features of the BG/Q. This solver was used in the chip design and testing stages to help validate the design.

Computers

Columbia/RBRC
QCDSP 1998-2005
0.050 GFlops/node

Columbia/RBRC/
UKQCD
QCDOC 2005-2011
0.8 GFlops/node

IBM BGL 2005-2013
2.8 GFlops/node

IBM BGP 2007-
13.6 GFlops/node

IBM BGQ 2012-
200 GFlops/node

RBC/UKQCD have production jobs on the Argonne ALCF BGQ that sustain 1 PFlops on 32 racks = 32k nodes = 0.5 M cores.

This performance comes from very carefully tuned assembly code on BGQ, produced by Peter Boyle, using his BAGEL code generator

Algorithms for Gauge Field Production

- Producing gauge fields:
  - Use classical molecular dynamics to move through gauge field space
  - Quark loops give back reaction on gauge fields by solving Dirac equation
  - Hasenbusch mass preconditioning allows tuning back reaction

$$\det[D(m)] = \frac{\det[D(m)]}{\det[D(m_1)]} \times \frac{\det[D(m_1)]}{\det[D(m_2)]} \cdots \det[D(m_n)]$$

  - RBC/UKQCD uses 7 levels of intermediate masses
  - Integrate different d.o.f on different time scales (Sexton-Weingarten integrators)
  - Use higher order integrators, currently RBC/UKQCD use force gradient, $O(dt^4)$
- These are giving 10-100× speed-up over a decade ago.
  - Hard to be completely quantitative here, since without speeds-ups, we could not even try current simulations

Algorithms for Measurements

$T = |t_K - t_\pi|$

- Time translated the n-point function, on a fixed background gauge field, are sufficiently decorrelated (independent enough) to make them worth calculating
- This means many solutions of the Dirac equation $D[U_\mu]\,\Psi = s$ for fixed U_μ
- Calculating eigenvectors of $D[U_\mu]$ with small eigenvalues (low-modes) speeds up subsequent solves. Can be done with EigCG or Lanczos algorithms
- Alternatives for Wilson fermions are domain decomposition and multigrid, giving similar speed-up with smaller memory requirements.
- Further improvement from all-mode-averaging of Blum, Izubuchi and Shintani

Physics Results and Prospects

Major Development: Ensembles with Physical Quark Masses

- Given advances in hardware, software and algorithms, lattice QCD physicists are now regularly doing simulations with physically light pions, a major goal since the beginning of numerical QCD in the early 1980's.

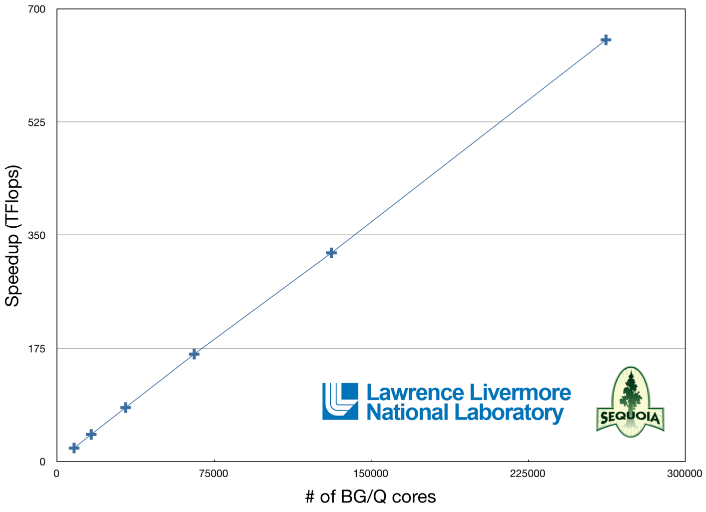
Fodor and Hoelbling, Rev. Mod. Phys. 84 (2012) 449

2+1+1 flavors, HISQ Staggered. MILC Phys. Rev. D87 (2013) 054505

2+1 flavors, (M)DWF
RBC and UKQCD Collaborations

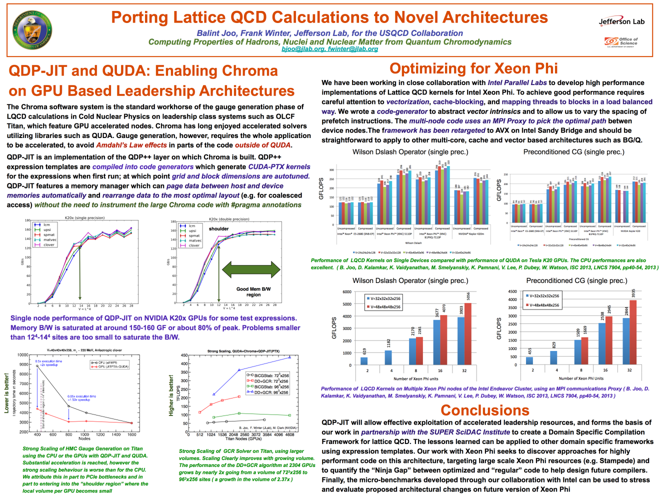
Large Vol. Ensembles with Physical Quark Masses
- BMW Hex-smeared clover fermions
- MILC HISQ Staggered fermions
- RBC/UKQCD DWF fermions

- The RBC and UKQCD collaborations, using the numerically expensive domain wall formulation, now have two large simulations at different lattice spacings with physical light quarks.
  - Domain wall fermions preserve all the continuum symmetries of QCD at finite lattice spacing
  - This is vital for measurements of many observables
- For the first time, physical results have been produced for a process with 2 particles in the final state, the amplitude for a kaon to decay to 2 pions (K -> ππ) with the pions in an isospin 2 final state. Physical Review Letters, 108 (2012) 141601 and Physical Review D 86 (2012) 074513.
- Previous results
- New, preliminary results (Preliminary)
- These new measurements are reducing the errors on measurements of properties of decay of kaons into a pion plus leptons, called Kl3 decays



Weak Scaling for BAGEL DWF CG Inverter

Titan/Super/Intel-Phi



Porting Lattice QCD Calculations to Novel Architectures

Balint Joo, Frank Winter, Jefferson Lab, for the USQCD Collaboration
Computing Properties of Hadrons, Nuclei and Nuclear Matter from Quantum Chromodynamics
bjoo@jlab.org, fwinter@jlab.org

Jefferson Lab

Office of Science

QDP-JIT and QUDA: Enabling Chroma on GPU Based Leadership Architectures

The Chroma software system is the standard workhorse of the gauge generation phase of LQCD calculations in Cold Nuclear Physics on leadership class systems such as OLCF Titan, which feature GPU accelerated nodes. Chroma has long enjoyed accelerated solvers utilizing libraries such as QUDA. Gauge generation, however, requires the whole application to be accelerated, to avoid *Amdahl's Law effects* in parts of the code *outside of QUDA*.

QDP-JIT is an implementation of the QDP++ layer on which Chroma is built. QDP++ expression templates are *compiled into code generators* which generate *CUDA-PTX kernels* for the expressions when first run; at which point *grid and block dimensions are autotuned.* QDP-JIT features a memory manager which can *page data between host and device memories automatically* and *rearrange data to the most optimal layout* (e.g. for coalesced access) *without the need to instrument the large Chroma code with #pragma annotations*

Single node performance of QDP-JIT on NVIDIA K20x GPUs for some test expressions. Memory B/W is saturated at around 150-160 GF or about 80% of peak. Problems smaller than 12^4-14^4 sites are too small to saturate the B/W.

Strong Scaling of HMC Gauge Generation on Titan using the CPU or the GPUs with QDP-JIT and QUDA. Substantial acceleration is reached, however the strong scaling behaviour is worse than for the CPU. We attribute this in part to PCIe bottlenecks and in part to entering into the "shoulder region" where the local volume per GPU becomes small

Strong Scaling of GCR Solver on Titan, using larger volumes. Scaling Clearly improves with growing volume. The performance of the DD+GCR algorithm at 2304 GPUs grows by nearly 2x going from a volume of 72^3x256 to 96^3x256 sites (a growth in the volume of 2.37x)

Optimizing for Xeon Phi

We have been working in close collaboration with *Intel Parallel Labs* to develop high performance implementations of Lattice QCD kernels for Intel Xeon Phi. To achieve good performance requires careful attention to *vectorization*, *cache-blocking*, and *mapping threads to blocks in a load balanced way*. We wrote a *code-generator* to abstract *vector intrinsics* and to allow us to vary the spacing of prefetch instructions. The *multi-node code uses an MPI Proxy to pick the optimal path* betwen device nodes.The f*ramework has been retargeted* to AVX on Intel Sandy Bridge and should be straightforward to apply to other multi-core, cache and vector based architectures such as BG/Q.

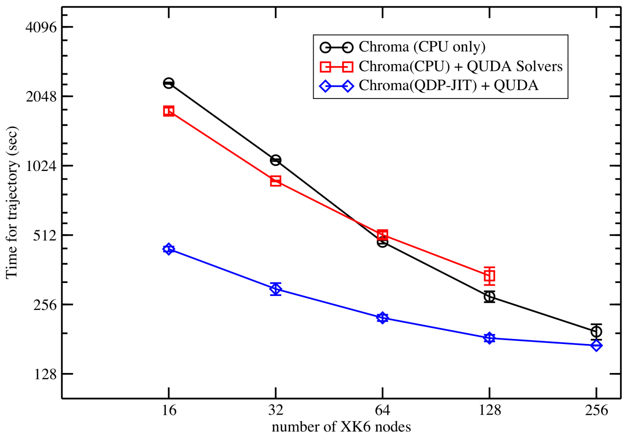
Performance of LQCD Kernels on Single Devices compared with performance of QUDA on Tesla K20 GPUs. The CPU performances are also excellent. (B. Joo, D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnani, V. Lee, P. Dubey, W. Watson, ISC 2013, LNCS 7904, pp40-54, 2013)

Performance of LQCD Kernels on Multiple Xeon Phi nodes of the Intel Endeavor Cluster, using an MPI communications Proxy (B. Joo, D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnani, V. Lee, P. Dubey, W. Watson, ISC 2013, LNCS 7904, pp40-54, 2013)

Conclusions

QDP-JIT will allow effective exploitation of accelerated leadership resources, and forms the basis of our work in *partnership with the SUPER SciDAC Institute* to create a Domain Specific Compilation Framework for lattice QCD. The lessons learned can be applied to other domain specific frameworks using expression templates. Our work with Xeon Phi seeks to discover approaches for highly performant code on this architecture, targeting large scale Xeon Phi resources (e.g. Stampede) and to quantify the "Ninja Gap" between optimized and "regular" code to help design future compilers. Finally, the micro-benchmarks developed through our collaboration with Intel can be used to stress and evaluate proposed architectural changes on future version of Xeon Phi

2 Flavor Wilson HMC (Gauge + 2 Flavor + Hasenbusch monomials), 32^3x96 lattice



QUDA: (QCD in CUDA) Library

QUDA (QCD in CUDA) library started in 2008 with NVIDIA's CUDA implementation by Kip Barros and Mike Clark at Boston University. It has expanded to a broad base of USQCD SciDAC [1] software developers and is in wide use as the GPU backend for HEP and NP SciDAC application codes: Chroma, CPS, MILC, etc.

Provides:

Various solvers for several discretizations,

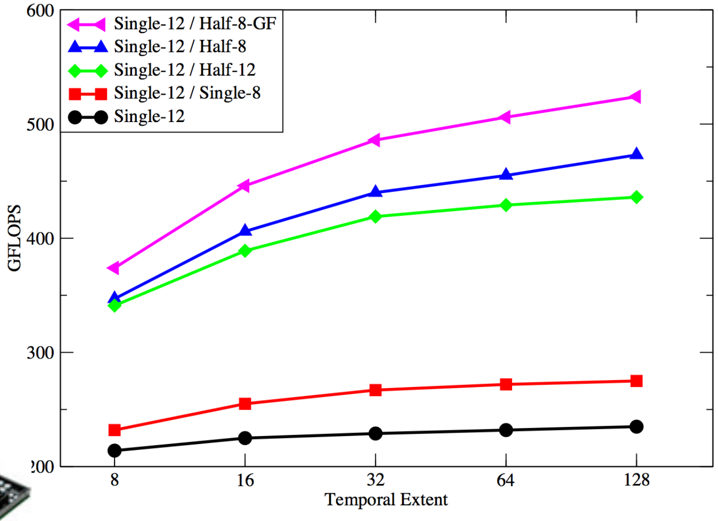
-- including multi-GPU support and
domain-decomposed (Schwarz) preconditioners

-- Additional performance-critical routines
needed for gauge-field generation

Maximize performance:

- Exploit physical symmetries
- Mixed-precision methods
- Autotuning for high performance
  on all CUDA-capable architectures
- Cache blocking

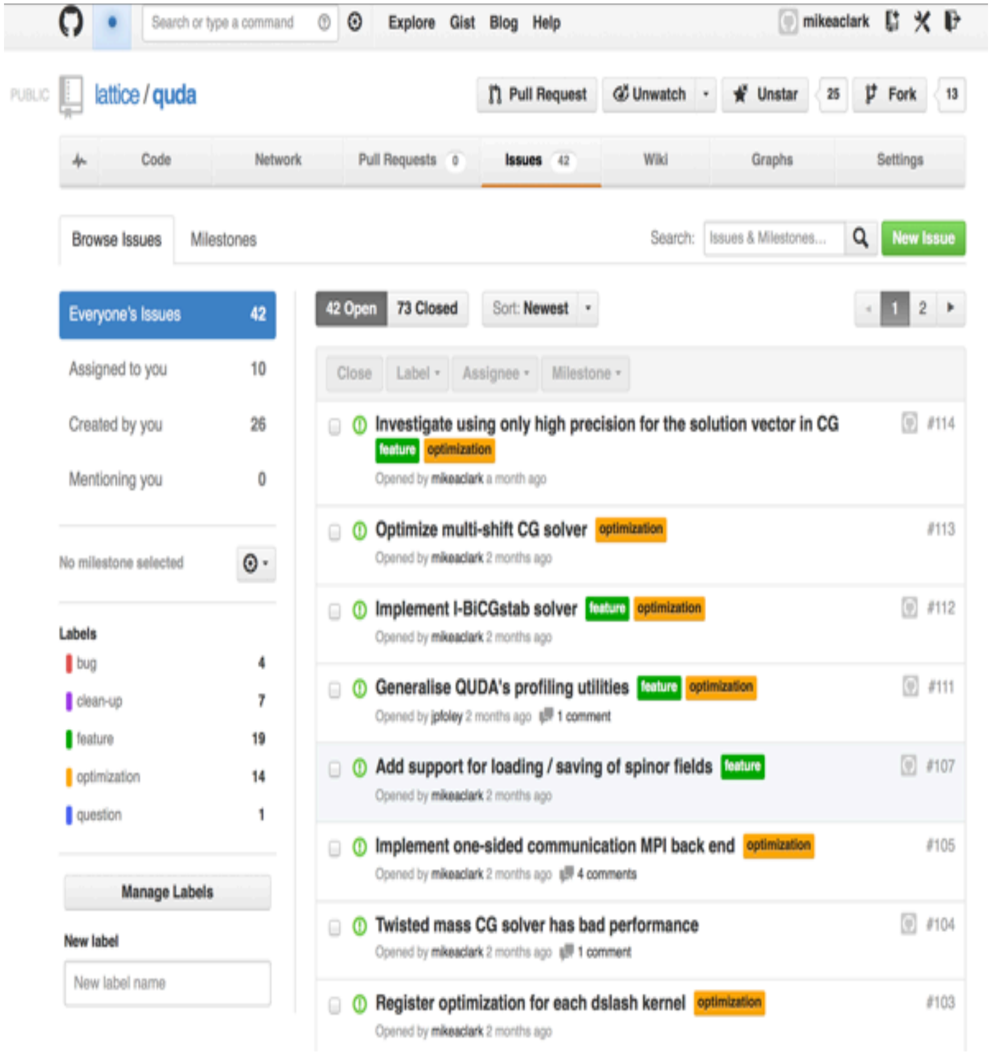
Kepler Wilson-Solver Performance



Kepler 20X with
2688 CUDA cores!

“QCD on CUDA” team – http://lattice.github.com/quda

- Ron Babich (NVIDIA)
- Kip Barros (LANL)
- Rich Brower (Boston University)
- Michael Cheng (Boston University)
- Mike Clark (NVIDIA)
- Justin Foley (University of Utah)
- Joel Giedt (Rensselaer Polytechnic)
- Steve Gottlieb (Indiana University)
- Bálint Joó (Jlab)
- Claudio Rebbi (Boston University)
- Guochun Shi (NCSA -> Google)
- Alexei Strelchenko (FNAL)
- Hyung-Jin Kim (BNL)
- Frank Winter (UoE -> Jlab)



Everyone's Issues	42
Assigned to you	10
Created by you	26
Mentioning you	0

Labels

bug	4
clean-up	7
feature	19
optimization	14
question	1

☐ **Investigate using only high precision for the solution vector in CG** feature optimization #114
Opened by **mikeaclark** a month ago

☐ **Optimize multi-shift CG solver** optimization #113
Opened by **mikeaclark** 2 months ago

☐ **Implement l-BiCGstab solver** feature optimization #112
Opened by **mikeaclark** 2 months ago

☐ **Generalise QUDA's profiling utilities** feature optimization #111
Opened by **jfoley** 2 months ago 1 comment

☐ **Add support for loading / saving of spinor fields** feature #107
Opened by **mikeaclark** 2 months ago

☐ **Implement one-sided communication MPI back end** optimization #105
Opened by **mikeaclark** 2 months ago 4 comments

☐ **Twisted mass CG solver has bad performance** #104
Opened by **mikeaclark** 2 months ago 1 comment

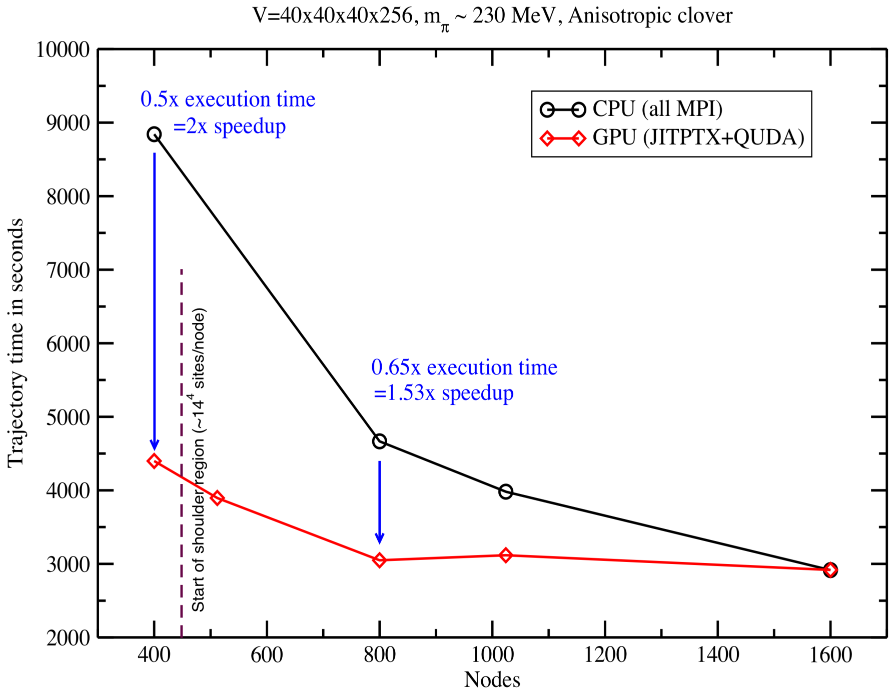
☐ **Register optimization for each dslash kernel** optimization #103
Opened by **mikeaclark** 2 months ago

*Gauge Generation using QDP-JIT/C on Titan**

"QDP-JIT .. implementation of the QDP++ layer on Chroma ...

QDP++ expression templates are compiled into code generators which generate CUDA-PTX ....

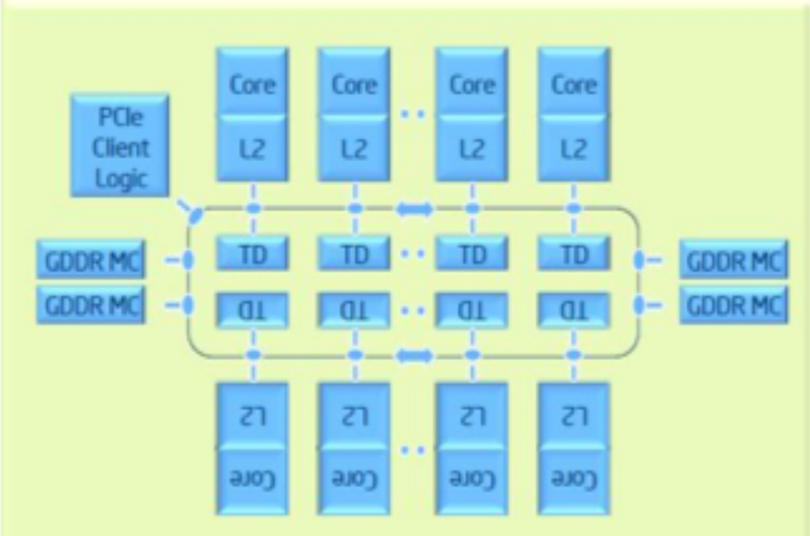
*Grid and block dimensions are autotuned. QDP-JIT features a memory manager which can page data between host and device memories automatically and rearrange data to the most optimal layout (e.g. for coalesced access) without the need to instrument the large Chroma code with #pragma annotations" **



Gauge generation strong scaling on Titan: still worse then for CPU probably due to PCIe

Quoted from Poster on "Porting Lattice QCD Calculations to Novel Architecture by Balint Joo and Frank Winter***

Intel Xeon-Phi (stampede)



Xeon Phi 5110P

- 60 cores @ 1.053 GHz
  - connected by ring
  - 512Kb L2$ / core
  - 32KB L1I$ and 32KB L1D$
  - in-order cores, 4 way SIMT
  - 512 bit wide Vector Engine
    - 16 way SP/8 way DP
  - can do multiply-add
- Peak DP Flops: 1.0108 TF
- Peak SP Flops: 2.0216 TF
- 8 GB GDDR (ECC)
  - 'top' shows ~6GB free when idle

Source: http://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-codename-knights-corner

http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-detail.html

Lagrangian for QCD

What so difficult about this!

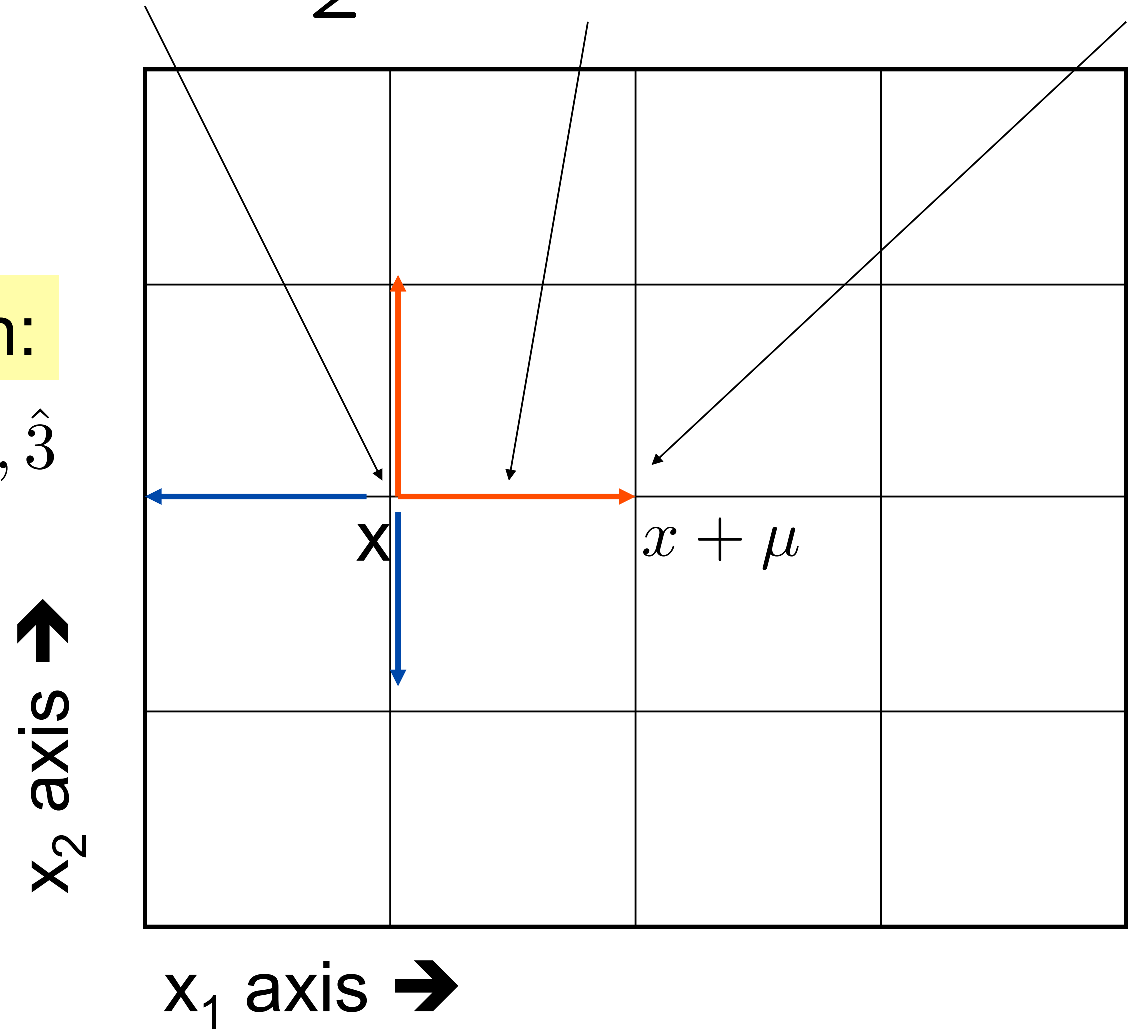
$$S = \int d^4x \mathcal{L}$$

$$\mathcal{L}(x) = \boxed{\frac{1}{4g^2} F^{ab}_{\mu\nu} F^{ab}_{\mu\nu}} + \boxed{\bar{\psi}_a \delta^{ab} \gamma_\mu (\partial_\nu + A^{ab}_\mu) \psi_b} + \boxed{m \bar{\psi} \psi}$$

- 3x3 “Maxwell” matrix field & 2+ Dirac quarks
- 1 “color” charge g & “small” quark masses m.
- Sample quantum “probability”: $\mathrm{Prob} \sim \exp[-S]$
- projectiles for HEP (Energy/Intensity) & Nuclear mater.

Wilson Dirac PDE on hypercubic Lattice

$$\bar{\Psi}_{ia}(x)\ \overbrace{\frac{1-\gamma_\mu}{2}}^{ij}\ U^{ab}_\mu(x)\ \Psi_{jb}(x+\mu)$$



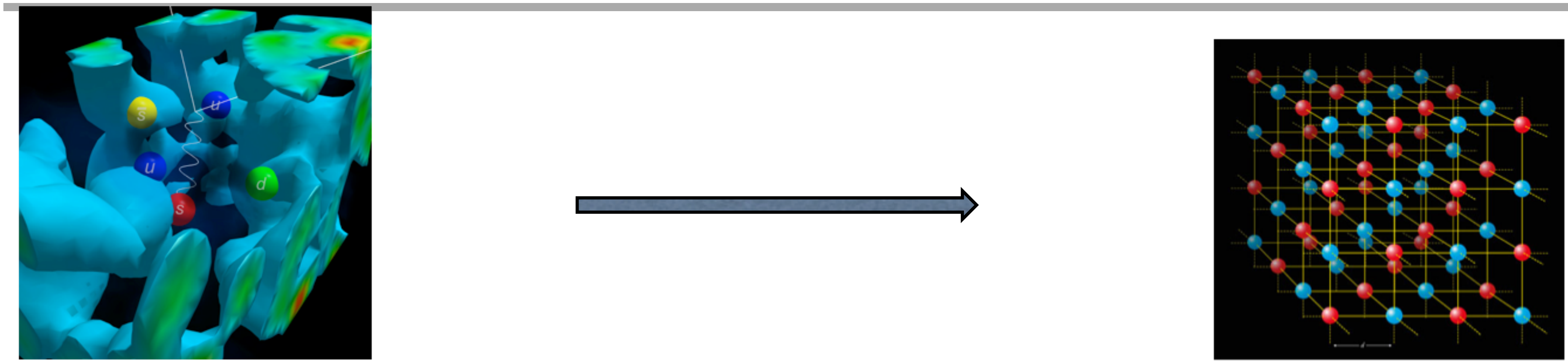
Dimension:

$\mu = \hat{0}, \hat{1}, \hat{2}, \hat{3}$

Color
a = 1,2,3

Spin
i = 1,2,3,4

Put it on a Lattice



- Quarks and Gluon on Lattice :
  - L^4 = 100x100x100x100 points
  - Discretize Dirac PDE as sparse 24xL^4 by 24xL^4 Matrix
  - Gauge Field 4xL^4 3x3 complex matrices
- Algorithms:
  - Krylov Solver for Quark PDE on 12x3x2 L^4 unknows for each
  - Semi-implicit Symplectic Hamiltonian Integrator:
  - Monte Carlo sample by Markov Process

Exact symmetries are powerful quantifier of errors that must vanish at zero lattice spacing

	Classical Langragian/ PDE's	Lattice (i.e.Computer)	Quantum (i.e.Nature)
Rotational(Lorentz) Invariance	✓	✗	✓
Gauge Invariance	✓	✓	✓
Scale Invariance	✓	✗	✗ *
Chiral Invariance	✓	✓	✗ *

Result: QM spontaneously causes large (and unexpected) large scales.

* *Plus a small extra breaking due to mass of quarks.*

New Frontier: Higher resolution QCD

- Lattice scales:

  a(lattice) << $1/M_{proton}$ << $1/m_\pi$ << L (box)

  0.06 fermi << 0.2 fermi << 1 .4 fermi << 6.0 fermi

  3.3 x 7 x 4.25 $\simeq$ 100

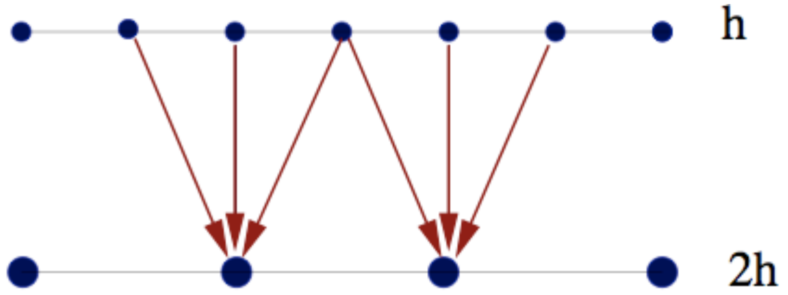
- Consequences:
  - Increasing ill-conditioned Dirac operator
  - Suffer from worse critical slowing down (CSD)
  - O(100^4) lattice volumes or more
  - Single grid Krylov methods and homogenous Arch are optimal

Many more LGT mass scales to come

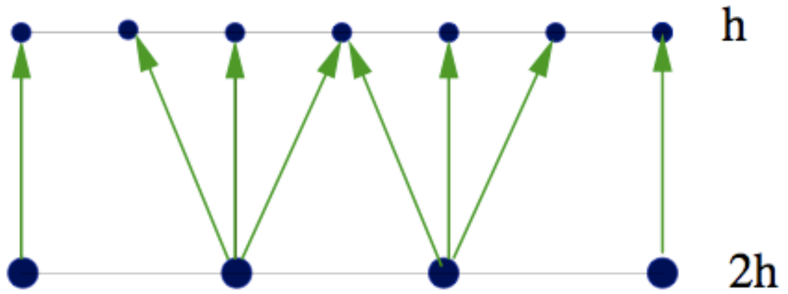
- quarks masses:
  (udscbt = 2, 5, 100, 1300, 4190, 200000 MeV)
- Electromagnetism (proton-nucleon splitting, g-2)
- Binding energy of nuclei (2.2 Mev for deuteron)
- TeV Strong Gauge BSM (near conformal) dynamics for composite Higgs
- ETC.

MG Scaling for Conformal Laplace

$$A\phi = b \implies \phi(x+h) - 2\phi(x) + \phi(x-h) + h^2 m^2 \phi(x) = b(x)$$



1 h ➔ 2h Restriction R = P†



2 h ➔ h Prologation P

(1) Blocking preserves the scale invariant const solutions (null state)

(2) Coarse operator is renormalized: m ➔ 2 m (in units h = 1)

QCD MG attempts in 1990’s

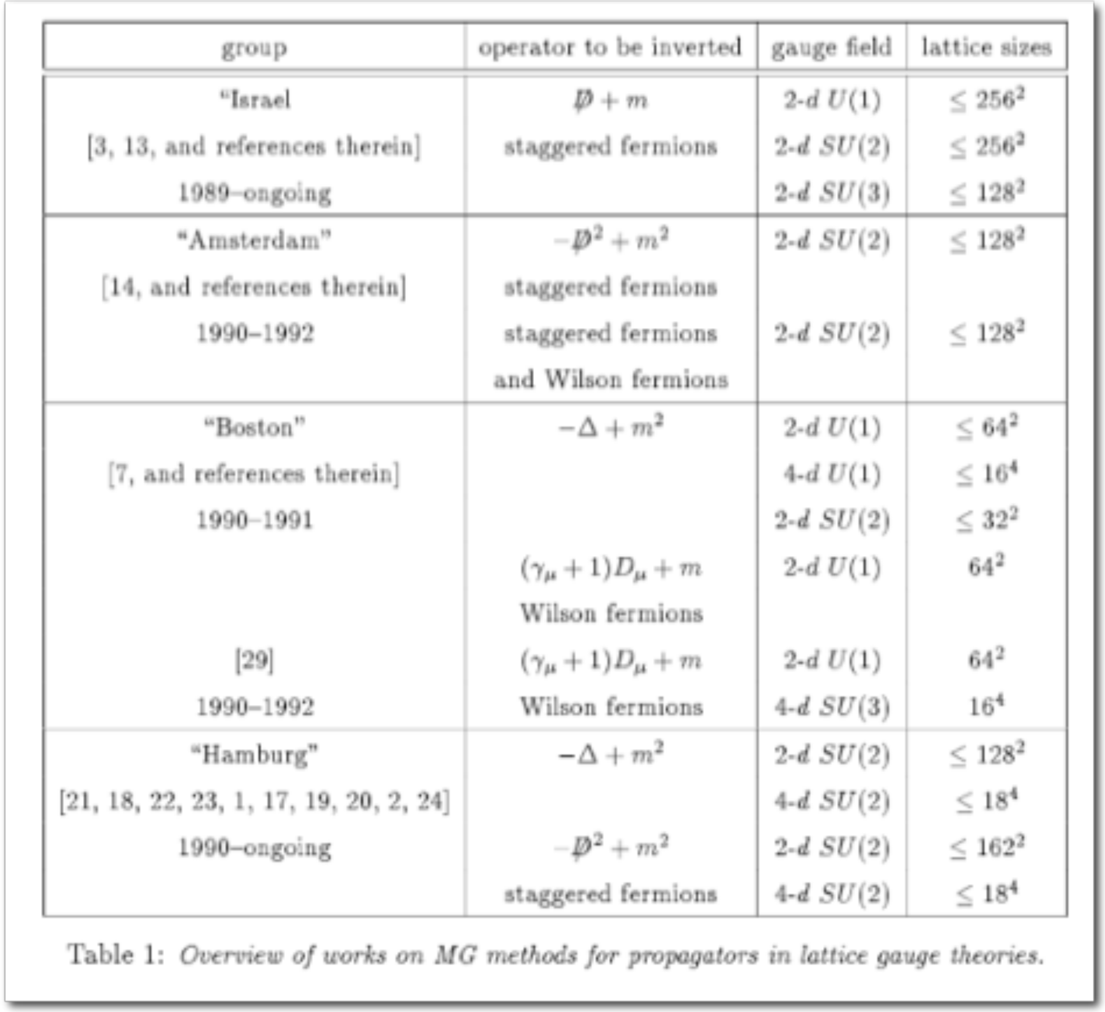
See Thomas Kalkretuer hep-lat/9409008 review on “MG Methods for Propagators in LGT”.

Israel: Ben-Av, M. Harmatz, P.G. Lauwers & S.Solomon

Boston: Brower, Edwards, Rebbi & Vicari

Amsterdam: A. Hulsebos, J Smit J. C. Vick

Amsterdam: A. Hulsebos, J Smit J. C. Vick



group	operator to be inverted	gauge field	lattice sizes
“Israel	$\not{D}+m$	2-d $U(1)$	$\leq 256^2$
[3, 13, and references therein]	staggered fermions	2-d $SU(2)$	$\leq 256^2$
1989–ongoing		2-d $SU(3)$	$\leq 128^2$
“Amsterdam”	$-\not{D}^2+m^2$	2-d $SU(2)$	$\leq 128^2$
[14, and references therein]	staggered fermions		
1990–1992	staggered fermions	2-d $SU(2)$	$\leq 128^2$
	and Wilson fermions		
“Boston”	$-\Delta+m^2$	2-d $U(1)$	$\leq 64^2$
[7, and references therein]		4-d $U(1)$	$\leq 16^4$
1990–1991		2-d $SU(2)$	$\leq 32^2$
	$(\gamma_\mu+1)D_\mu+m$	2-d $U(1)$	64^2
	Wilson fermions		
[29]	$(\gamma_\mu+1)D_\mu+m$	2-d $U(1)$	64^2
1990–1992	Wilson fermions	4-d $SU(3)$	16^4
“Hamburg”	$-\Delta+m^2$	2-d $SU(2)$	$\leq 128^2$
[21, 18, 22, 23, 1, 17, 19, 20, 2, 24]		4-d $SU(2)$	$\leq 18^4$
1990–ongoing	$-\not{D}^2+m^2$	2-d $SU(2)$	$\leq 162^2$
	staggered fermions	4-d $SU(2)$	$\leq 18^4$

Table 1: *Overview of works on MG methods for propagators in lattice gauge theories.*

QCD MG “failure” in 1990’s:

SUPERCOMPUTER
COMPUTATIONS
RESEARCH INSTITUTE

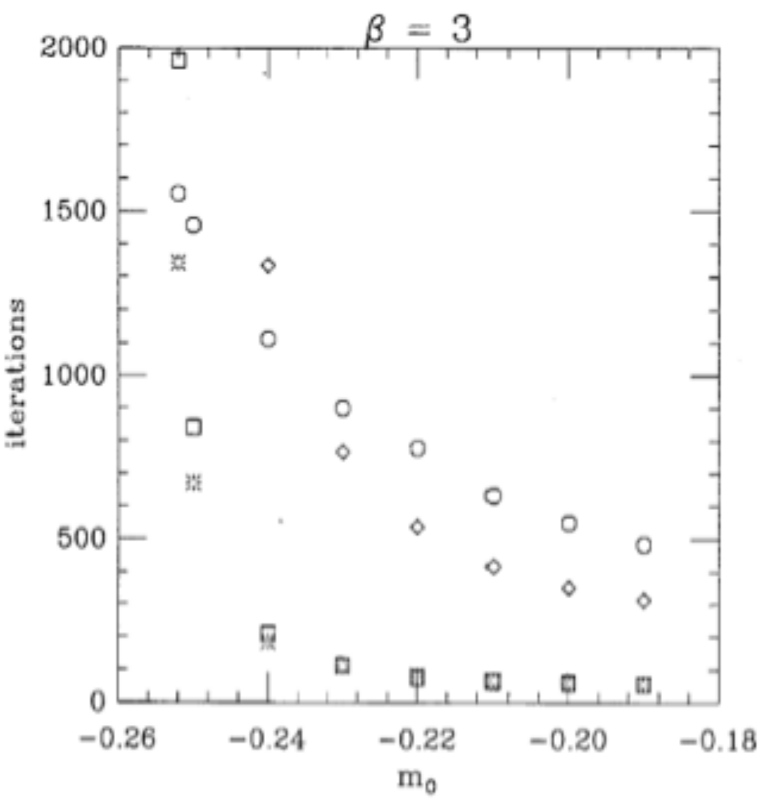
**PROJECTIVE MULTIGRID FOR
WILSON FERMIONS**

by

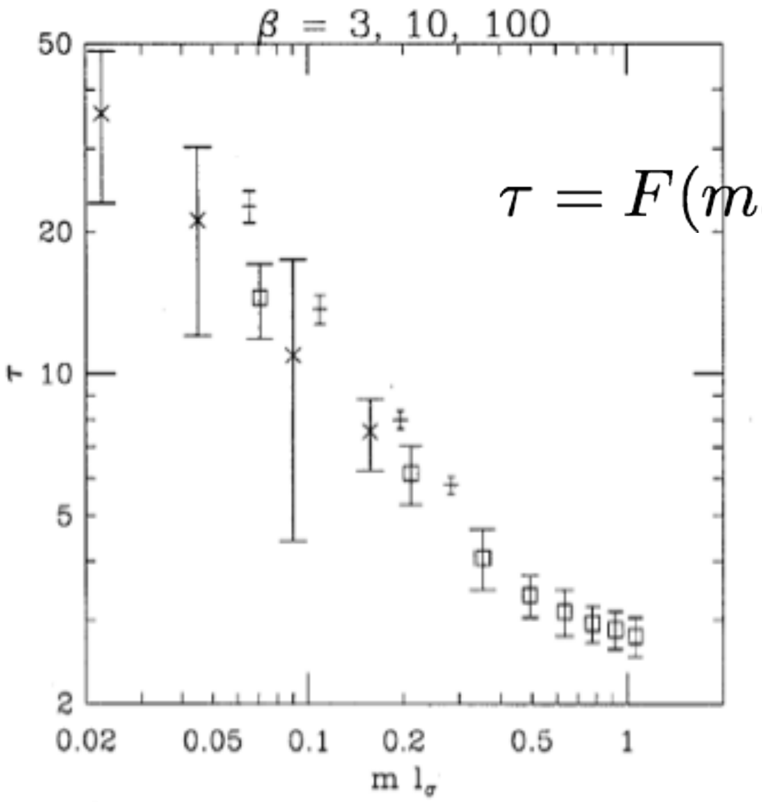
Richard C. Brower, Robert G. Edwards,
Claudio Rebbi, and Ettore Vicari

FSU-SCRI-91-54

Universal of Critical Slowing down:



Gauss-Jacobi (Diamond), CG(circle), 3 level (square & star)

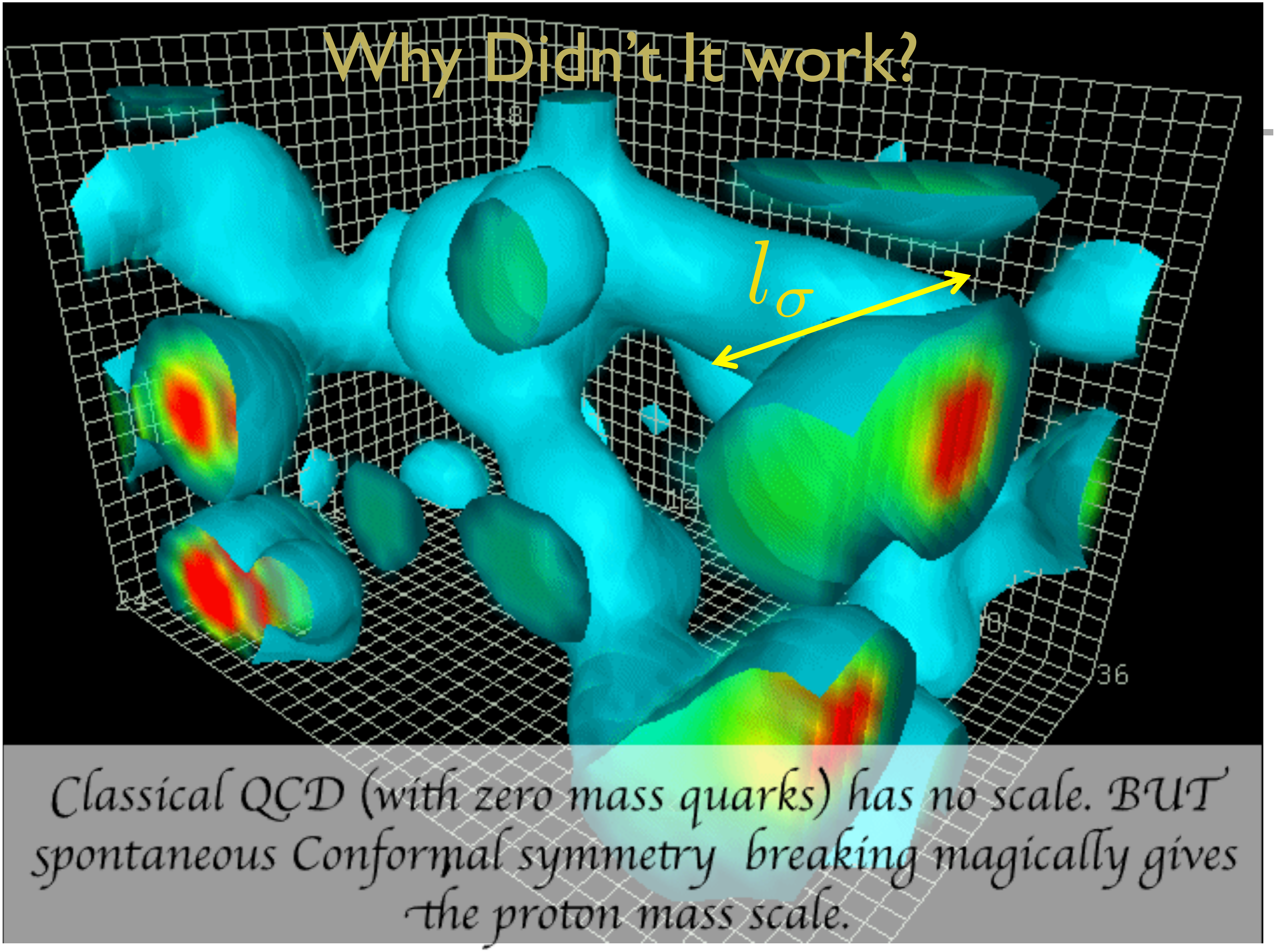


$$\tau = F(ml_\sigma)$$

☒ = 3 (cross) 10(plus) 100(square)

Time = F(spontaneous length/quark compton lenght)

Why Didn't It work?



Classical QCD (with zero mass quarks) has no scale. BUT spontaneous Conformal symmetry breaking magically gives the proton mass scale.

Success & Failures of MG attempts in 1990's : Why?

- Partial success: weak coupling "renormalization"
- Maintain Exact Gauge invariance
- Maintain exact γ_5 Hermiticity
- Local adaptive blocking: Projective MG
- Chiral Symmetry (density of small e.v.)
- Null vector: Atya- Singer Index Theorem

Boris Grigoryevich Galerkin (Russian: Бори́с Григо́рьевич Галёркин, surname more accurately romanized as **Galyorkin**; March 4 [O.S. February 20, 1871] 1871 – July 12, 1945),

$$H = \gamma_5 D = D^\dagger \gamma_5$$

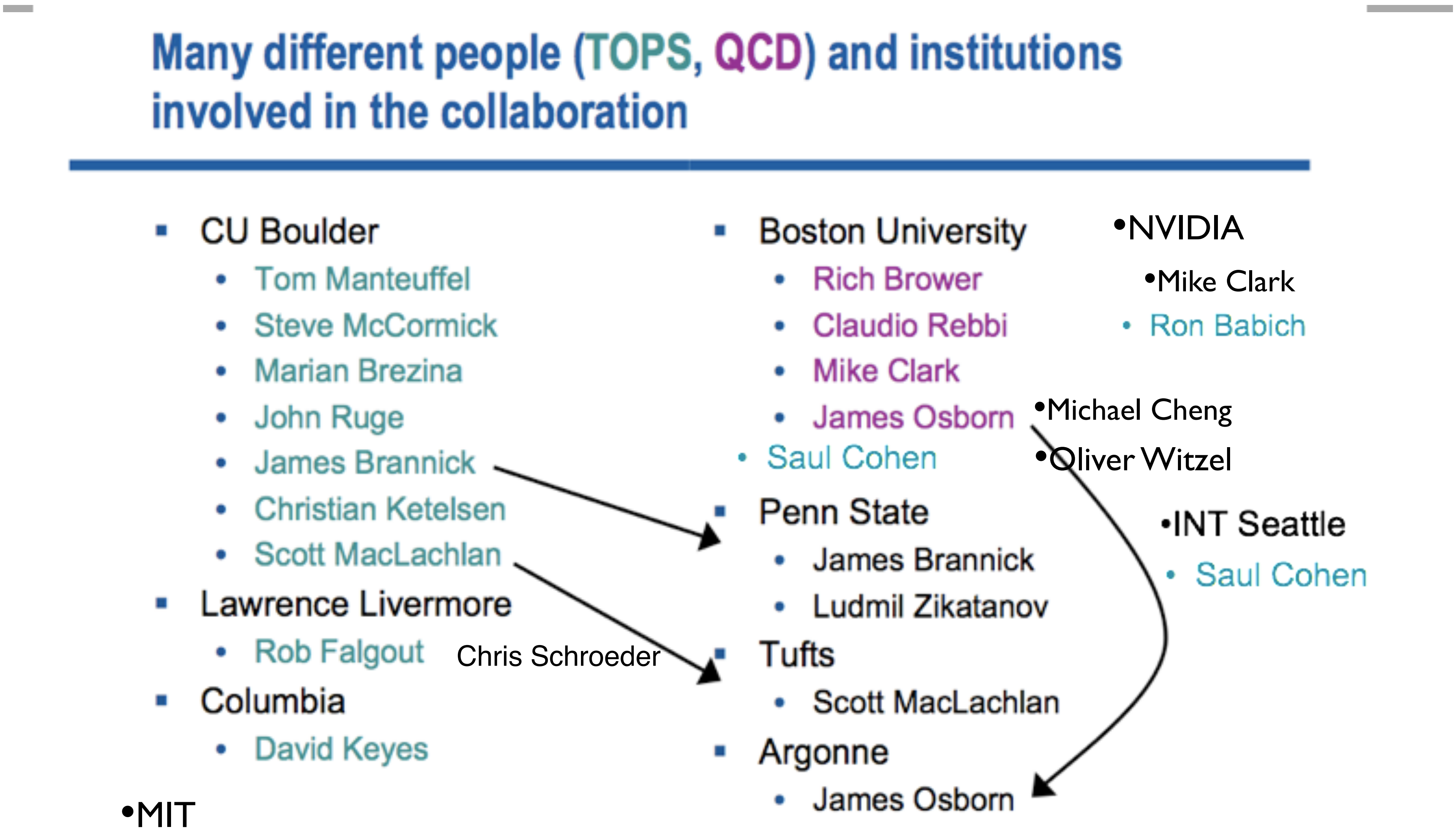
$$\text{Prolongator} \quad P \implies \text{Restrictor} \quad R = P^\dagger \gamma_5$$

How do we get beyond the rough confinement barrier?

- 1990 Projective MG: First “partitioned” in Jacobi grid blocks. Second “project” near null block vectors
- Ok for weak fields (weak coupling Renormalization

  Group, ignores instantons for example.)
- 2005 David Keyes to BU with new “adaptive MG idea”. Brannick et al tried it for 2-d Dirac Eq. – slow algorithm but no critical slowing down!
- 2010 Practical QCD AMG: First “project” onto near null vectors (bad guys). Second “partition” into coarse grid.

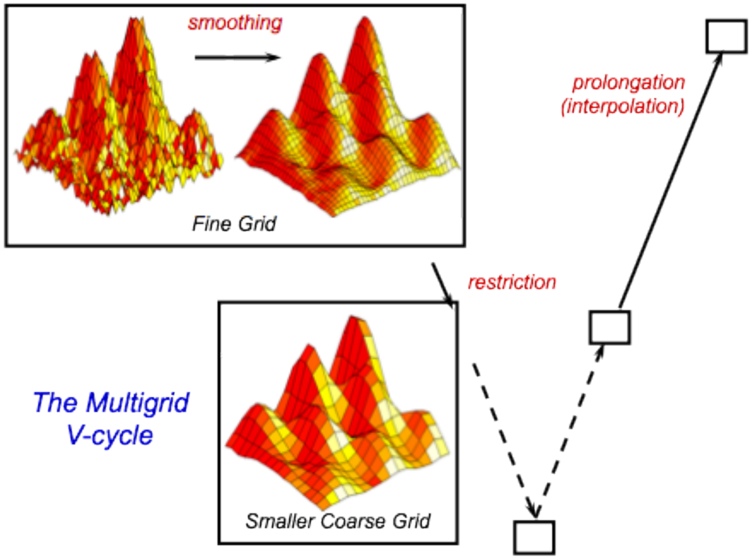
First Success: Applied Math/Physics Collaboration Collaboration



Many different people (TOPS, QCD) and institutions involved in the collaboration

- CU Boulder
  - Tom Manteuffel
  - Steve McCormick
  - Marian Brezina
  - John Ruge
  - James Brannick
  - Christian Ketelsen
  - Scott MacLachlan
- Lawrence Livermore
  - Rob Falgout
  - Chris Schroeder
- Columbia
  - David Keyes
- MIT
  - Andrew Pochinsky
- Boston University
  - Rich Brower
  - Claudio Rebbi
  - Mike Clark
  - James Osborn
  - Saul Cohen
  - Michael Cheng
  - Oliver Witzel
- Penn State
  - James Brannick
  - Ludmil Zikatanov
- Tufts
  - Scott MacLachlan
- Argonne
  - James Osborn
- Meifeng Lin
- NVIDIA
  - Mike Clark
  - Ron Babich
- INT Seattle
  - Saul Cohen

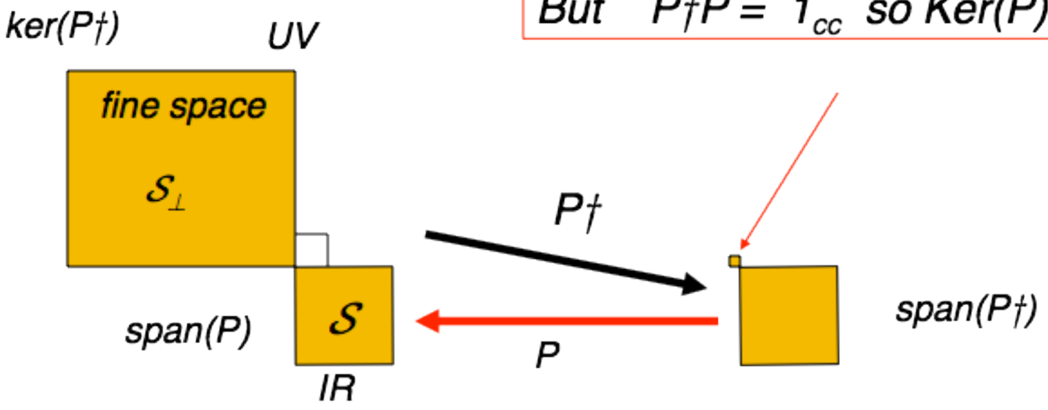
Adaptive Smooth Aggregation Algebraic Multigrid



Spilt the vector space into near null space $\mathcal{S}$ and the complement $\mathcal{S}_\perp$

$$D: \mathcal{S} \simeq 0$$

But $P\dagger P = 1_{cc}$ so $Ker(P) = 0$



(see Front cover of Strang's Undergraduate MIT math text!)

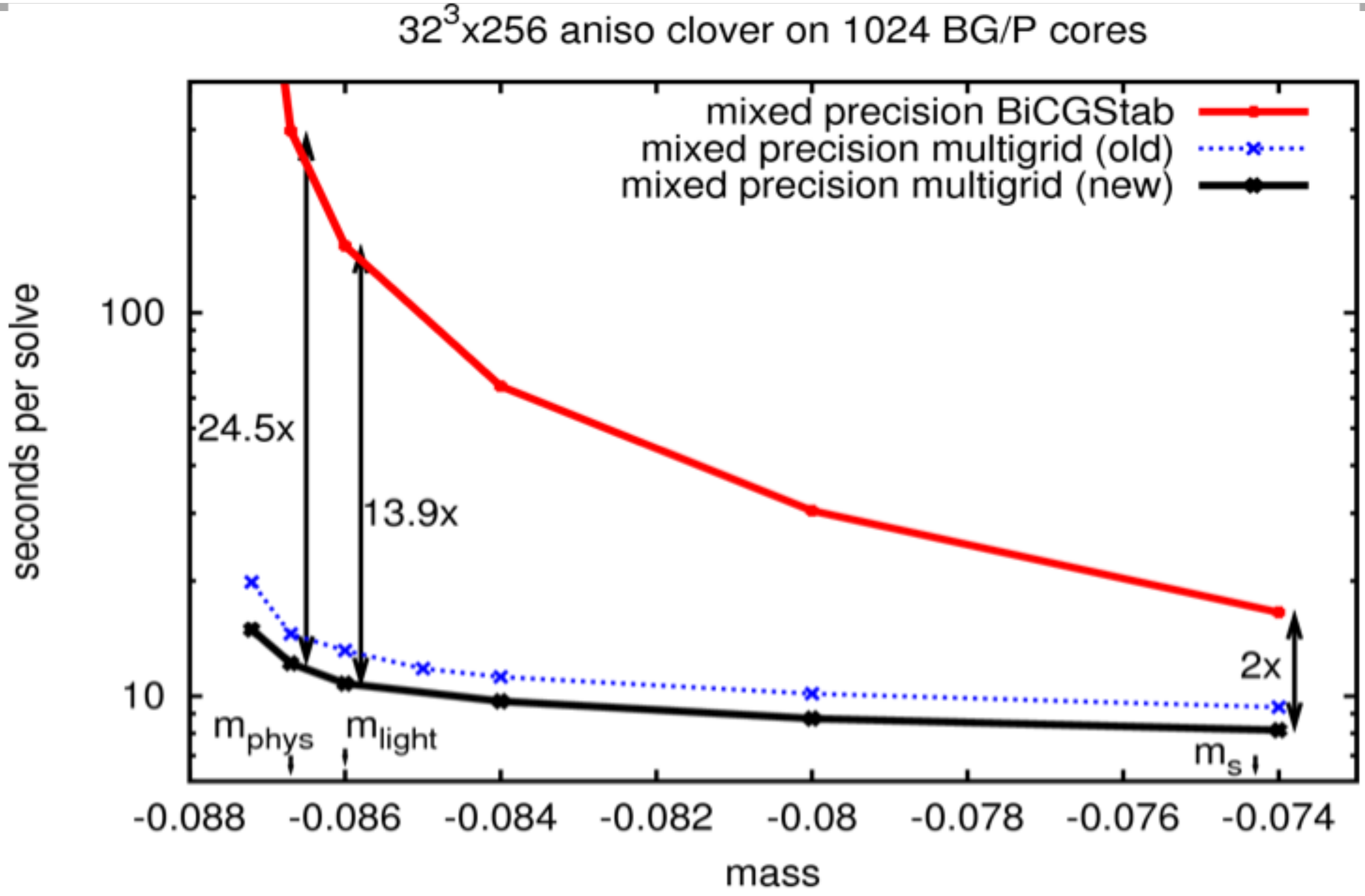
AMG on Wilson-clover Dirac Operator

- Devil is in the details!
  - Rigorous MG proofs for normal equation ($D^\dagger D \psi = b$)
  - But would like to project D to avoid higher complexity.
  - Multigrid is recursive to multi-levels.
  - Must preserves Gauge invariance and γ_5 ($[\gamma_5, P] = 0$)
- First benchmarks for Wilson-Dirac Operator:
  - Asym $V = 16^3 \times 64$, $24^3 \times 64$, $32^3 \times 96$ ($N_f = 2$, 400MeV pion)
  - $N_\nu = 20$ null vectors $\psi^s{}_x$ with 4th order MR with subset refinement.
  - MG Blocks = $4^4 \times N_c \times 2$ and 3 level V MG cycle
  - pre and post-smoothing is done by 4 iteration GCR (later GMRES)
  - Extend to Red/Black preconditioning

James Osborn implement on BG/P in SciDAC-2 API

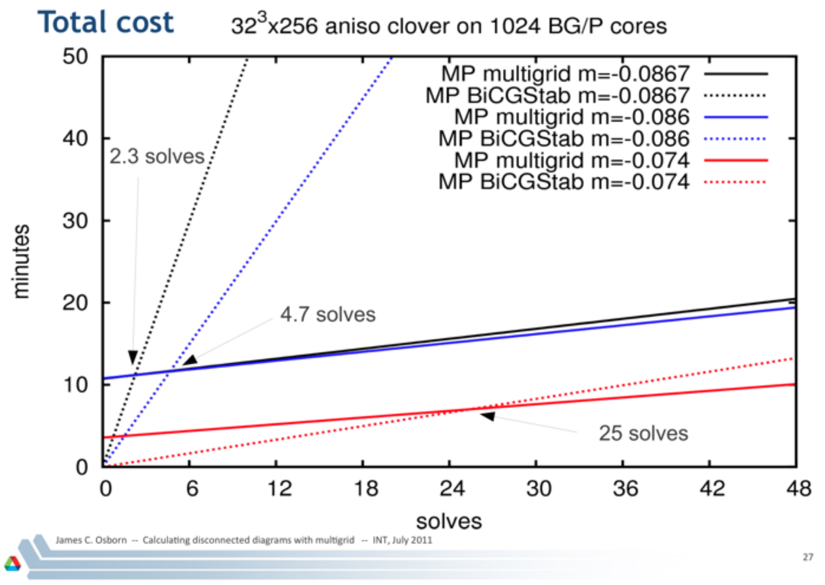
Future SciDAC-3 develop in HYPER framework/GPUs etc).

Adaptive Smooth Aggregation Algebraic Multigrid



"Adaptive multigrid algorithm for the lattice Wilson-Dirac operator" R. Babich, J. Brannick, R. C. Brower, M. A. Clark, T. Manteuffel, S. McCormick, J. C. Osborn, and C. Rebbi, PRL. (2010).

Good News/Bad News



Total cost

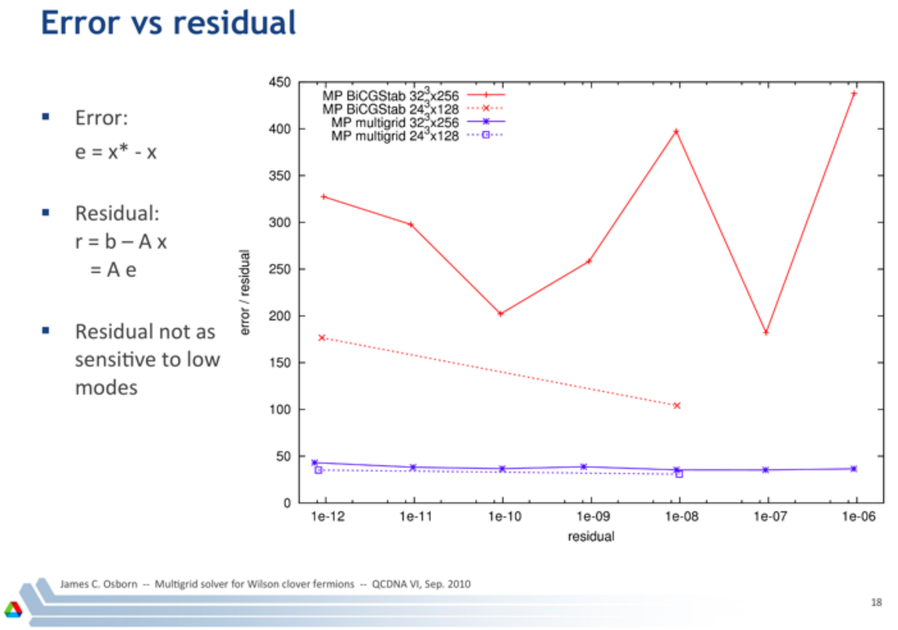
32^3x256 aniso clover on 1024 BG/P cores

James C. Osborn -- Calculating disconnected diagrams with multigrid -- INT, July 2011

27

More Data: Should Archive MG projectors with lattice

Actually MG error is smaller at fixed Residual



Error vs residual

- Error:
  e = x* - x
- Residual:
  r = b – A x
  = A e
- Residual not as sensitive to low modes

James C. Osborn -- Multigrid solver for Wilson clover fermions -- QCDNA VI, Sep. 2010

18

Must put Best Algorithm on Best Hardware

- Problem: Wilson Clover for Light Quark is FASTER on the CPU than using the QUDA solver on GPUs!
- Solution put MG on GPU of course

-
  - Cost in $s reduced by a factor of at least

100+

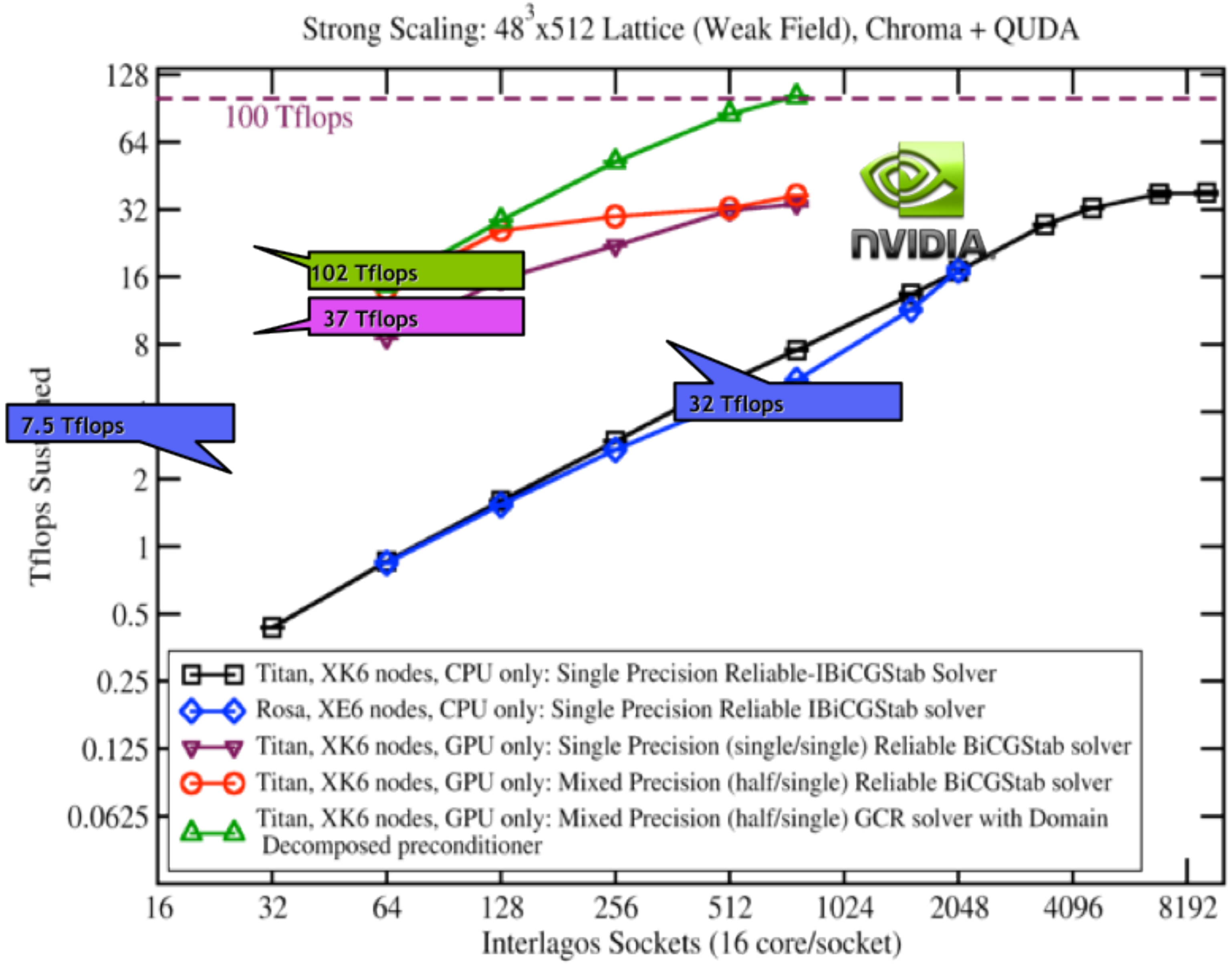
GPU O(10+)

MG O(10+)

(now with Mike Clark and Michael Cheng on NSF grant)

Communication Redcution:

DD (Block Jacobi) on Titan



102 Tflops

37 Tflops

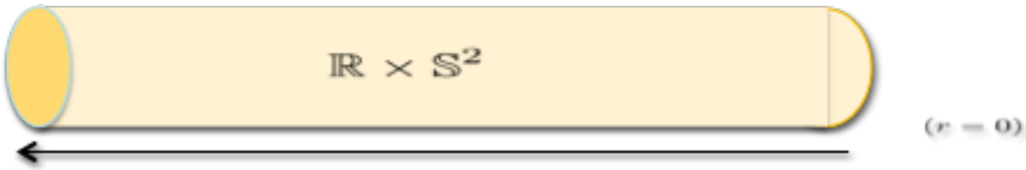
32 Tflops

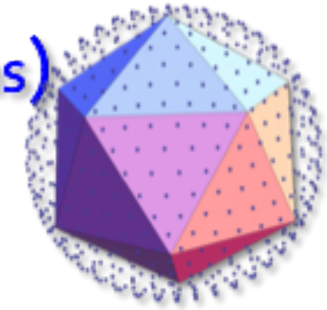
7.5 Tflops

New Lattice Geometries on Curved Manifolds

- Conformal Radial Quantization

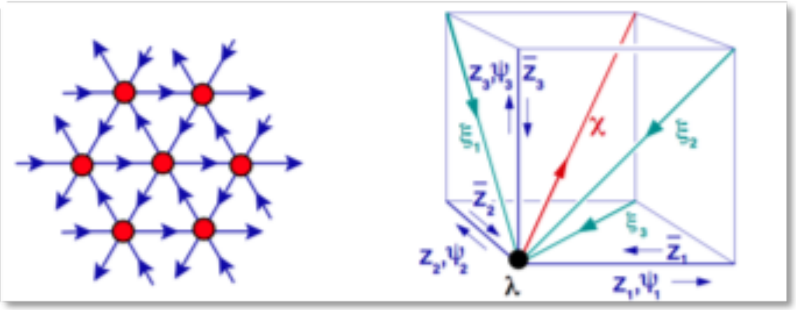
(finite element & graph theoretic algorithms)

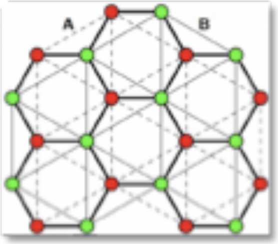




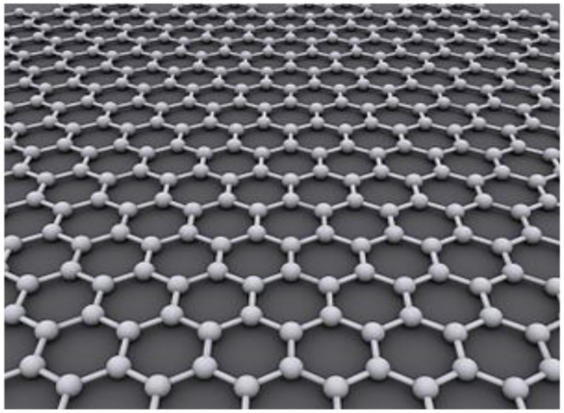
$$\tau = \log r$$

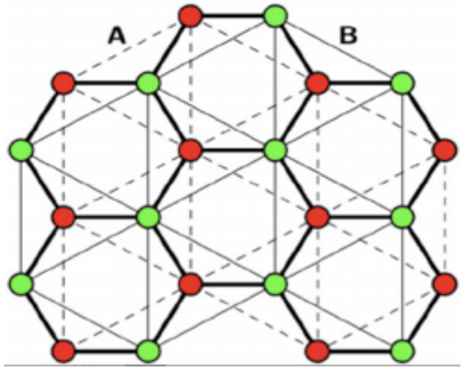
- SUSY (Super Symmetry) & Graphene!



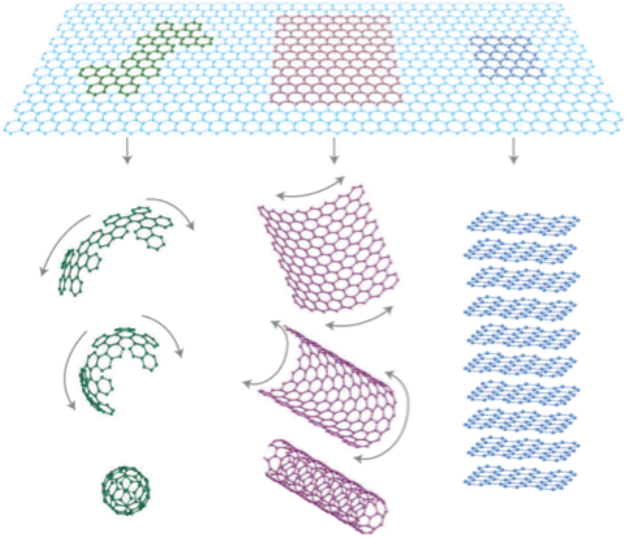


Graphene





A/B Bravais sub lattices
Effective field 2+1 relativistic theory:
4 copies of 2 component Dirac fields
Phonons act like gauge fields.

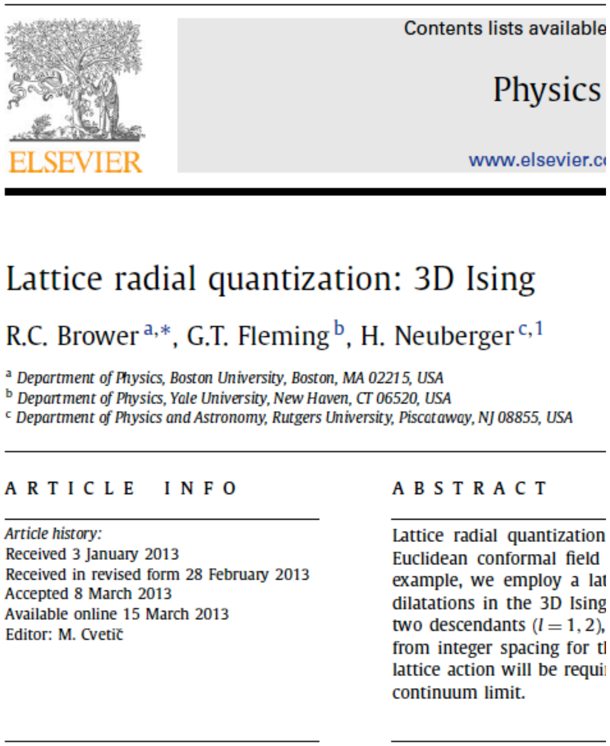


$$e_{eff}^2 = \frac{e^2}{\hbar v} \simeq 300 \times \frac{e^2}{\hbar c}$$

Graphene is 2+1 dimension Carbon sheet with Dirac fields: But lattice is real Hexagonal structure. Couple to coulomb potential and phones act like gauge fields! Ideal for Lattice field theory, MG and GPU! (Brower, Rebbi and Schaich)

$\mathbb{R}^4 \to \mathbb{R} \times \mathbb{S}^3$ *Conformal Lattice*

- Radial Quantization requires “spatial” spheres!
  - Goal is Conformal BSM Fix Points(or Scattering Length?)
  - Need Finite Elements Method to do 3d Ising on curves space!



Contents lists available

Physics

www.elsevier.c

ELSEVIER

Lattice radial quantization: 3D Ising

R.C. Brower [a,*], G.T. Fleming [b], H. Neuberger [c,1]

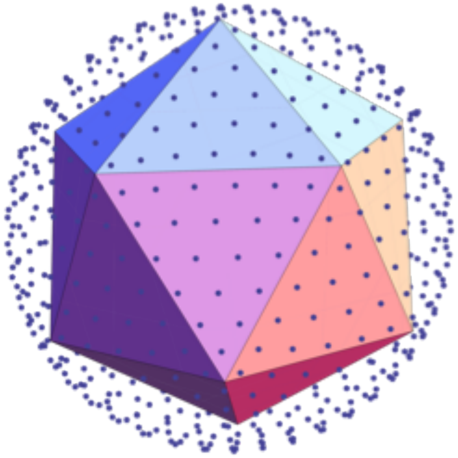
[a] Department of Physics, Boston University, Boston, MA 02215, USA
[b] Department of Physics, Yale University, New Haven, CT 06520, USA
[c] Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08855, USA

ARTICLE INFO

Article history:
Received 3 January 2013
Received in revised form 28 February 2013
Accepted 8 March 2013
Available online 15 March 2013
Editor: M. Cvetič

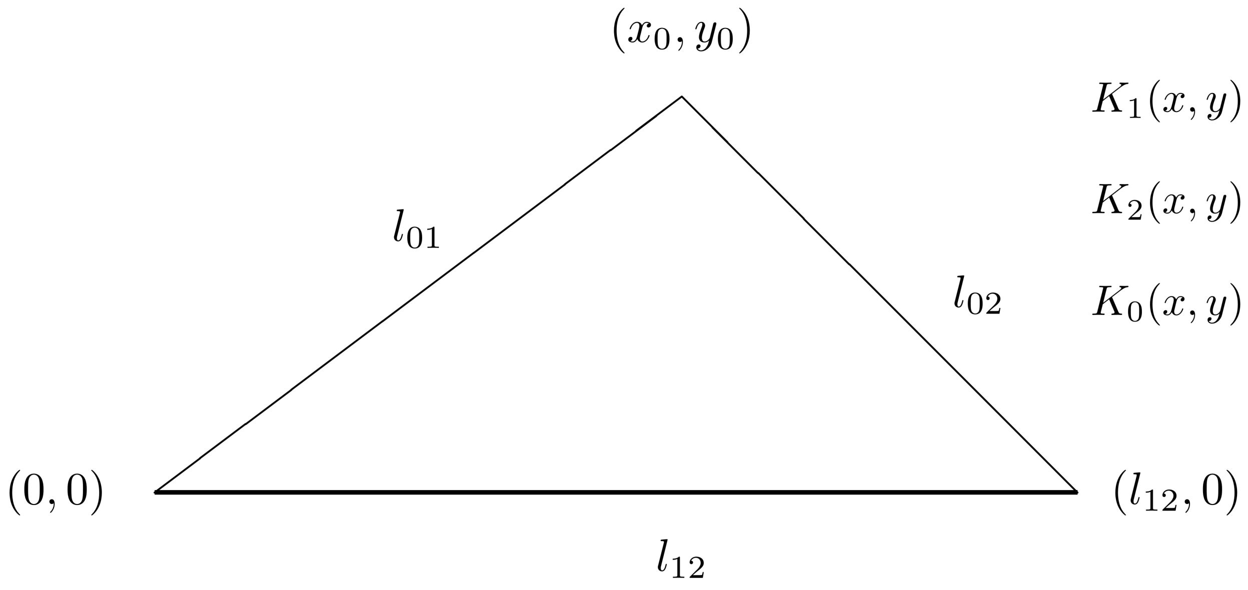
ABSTRACT

Lattice radial quantization
Euclidean conformal field
example, we employ a lat
dilatations in the 3D Ising
two descendants ($l = 1, 2$),
from integer spacing for t
lattice action will be requi
continuum limit.



$$a < \Delta r < L \quad \text{vs} \quad a < \Delta log(r) < L$$

Finite Element Method Laplacian on a sphere



$$\begin{aligned} K_1(x,y) &= [l_{12} - x - \frac{(l_{12} - x_0)y}{y_0}]/l_{12} \\ K_2(x,y) &= [x - \frac{x_0 y}{y_0}]/l_{12} \\ K_0(x,y) &= \frac{y}{y_0} \end{aligned}$$

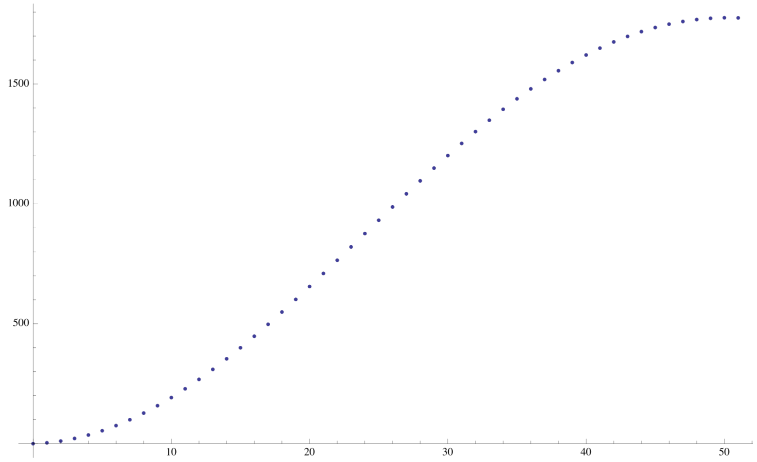
triangle on the tangent plane

$$\text{with} \quad \phi(x,y) = K_i(x,y)\phi_i \quad \Longrightarrow \quad \int_{A_{012}} dx dy \partial_\mu \phi \partial_\mu \phi(x) =$$

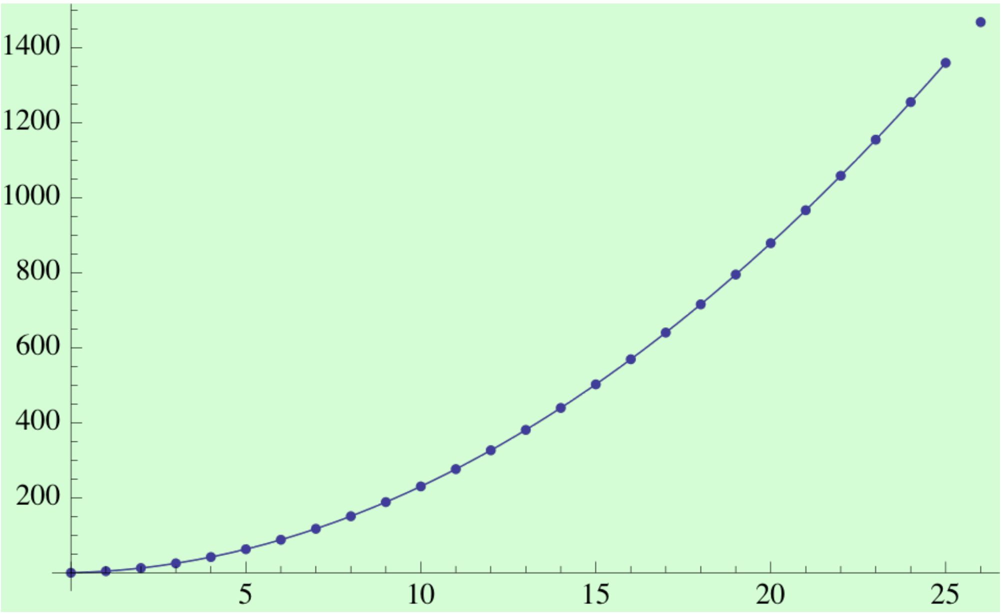
$$\frac{1}{4A_{012}}[l_{02}^2\phi_1^2 + l_{01}^2\phi_2^2 + l_{12}^2\phi_0^2] \quad - \quad \frac{1}{4A_{012}}[(l_{01}^2 + l_{02}^2 - l_{12}^2)\phi_1\phi_2 + \text{cyclic}]$$

See *WEIGHTS OF LINKS AND PLAQUETI'ES IN A RANDOM LATTICE** N.H. CHRIST, R FRIEDBERG and T D. LEE Nuclear Physics (1982) quarks and gluons are more difficult!

Spectrum of Laplacian on a sphere



s =8

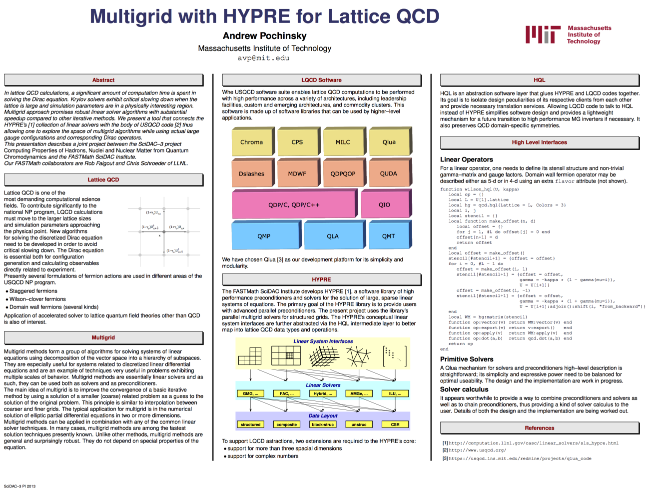


s = 512

$$2.09439l + 2.09439l^2 - 5.75 * 10^{-6} l^3 - 2.95833 * 10^{-6} l^4$$

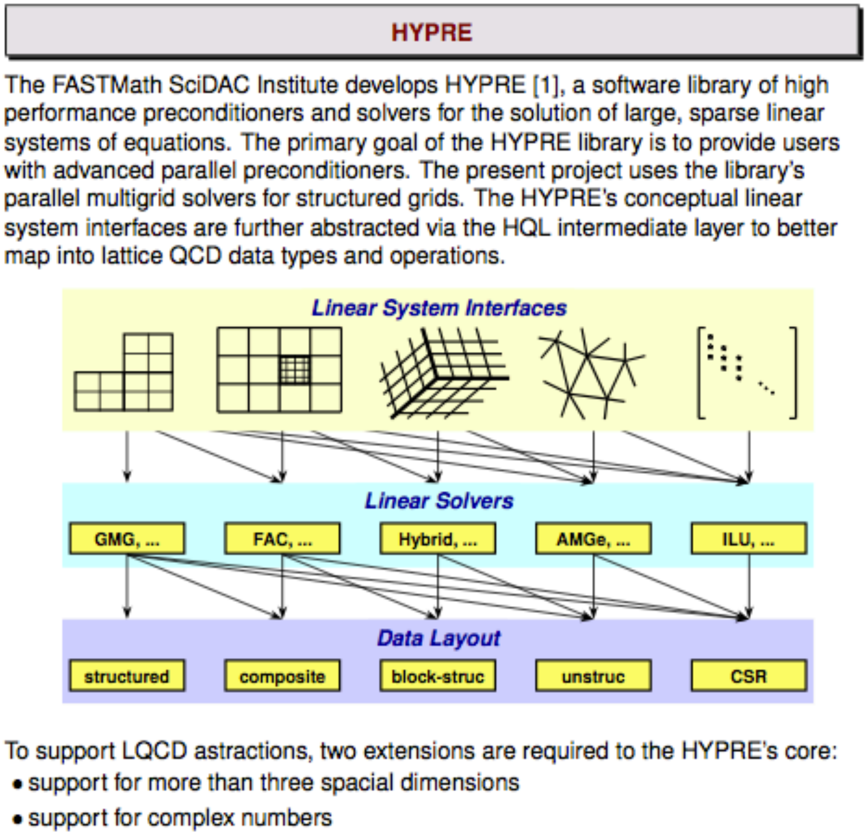
See posters for more Details

MG/fastMATH


Multigrid with HYPRE for Lattice QCD
Andrew Pochinsky
Massachusetts Institute of Technology
avp@mit.edu

New ideas for Rapid Prototyping using Lua Scripting language: Qlua (MIT) & FUEL (ANL)

Auto-tuning in use in QUDA and FUEL


HYPRE

The FASTMath SciDAC Institute develops HYPRE [1], a software library of high performance preconditioners and solvers for the solution of large, sparse linear systems of equations. The primary goal of the HYPRE library is to provide users with advanced parallel preconditioners. The present project uses the library's parallel multigrid solvers for structured grids. The HYPRE's conceptual linear system interfaces are further abstracted via the HQL intermediate layer to better map into lattice QCD data types and operations.

Linear System Interfaces

Linear Solvers

GMG, ... FAC, ... Hybrid, ... AMGe, ... ILU, ...

Data Layout

structured composite block-struc unstruc CSR

To support LQCD astractions, two extensions are required to the HYPRE's core:
- support for more than three spacial dimensions
- support for complex numbers

Summary

- Success in Higher Resolution Physics and Heterogeneous computers are great for Lattice Gauge Theory Physics.

- Both require a increasingly sophisticated suit of multi-scale algorithms Combine MG a nd DD to get both Fast convergence & Communication reduction: BLTN algorithms?

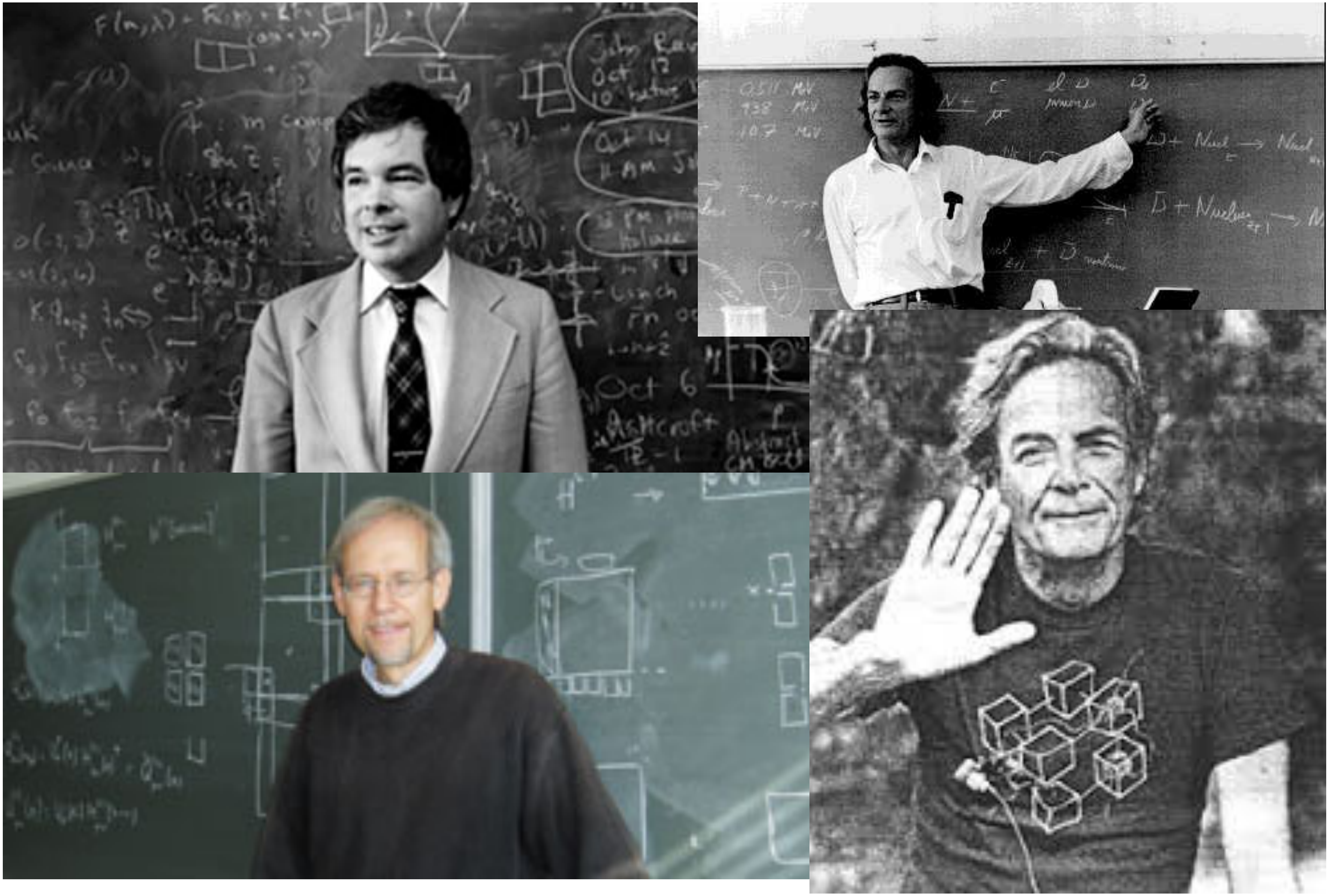
- Lattice Field Software will be come increasingly intricate and expensive. Need new emphasis on tools adopted from or developed in collaboration with Applied Math and Computer Science.

- Rapid prototyping frameworks, auto-tuning, better compilers and restructuring of the API/Domain Specific Language.
- There is probably magic bullet but current develops are beginning to suggest solutions.

Questions & Extra Slides

- BLTN (Better Late Than Never) Solver?

Physics+Math+Computing <=> Algorithm



New FUEL HMC framework
(Framework for Unified Evolution of Lattices)

- High level layer focused on gauge configuration generation
  - motivation is to have flexible HMC framework to support wide range of beyond standard model theories
  - algorithmic abstraction: generation algorithm independent of gauge group, action, etc.
  - easy to write new high-level algorithms, tune parameters
  - serves as wrapper for efficient “level 3” routines
  - easy to plug in new routines
  - new routines can be written in any other language/framework

- Uses scripting language Lua
  - Small
  - Easy to port (ANSI C89)
  - Easy to use, yet powerful
  - Easy to embed and interface with libraries

Rich: Documentation is being addressed this spring/summer. Michael Chang and Mike Clark are nearly finished implementing the fine level of MG on the GPU. James, Meifeng and I

are working on integrating Multigrid into Wilson colver evolution code Kostas and Will 's Wilson isotropic HMC/Chroma trajectory. Looks promising.

Andrew & Chris: Integration of HYPRE and Qlua is well underway. Rob Falgout, Christopher Schroeder and Andrew Pochinsky have completed an overall design of a HYPRE/USQCD interface (HQL) and begun its implementation. RF is largely finished extending HYPRE to handle more than 3 dimensions and fully expects to finish implementing complex numbers on schedule. CS and RF are making progress on the implementation of the HQL interface, and RF and AP are proceeding with the HQL-Qlua interface. AP is finishing extending Qlua to handle data types and procedures required to support HQL.

Some future directions

- Fermions PDEs are ubiquitous in Quantum Field theories an Nano materials. Lattice geometries and boundary condition present new fun challenges.
  - Finite T lattice (e.g. 32 x speed up on $128^3 \times 96$ lattice*)
  - Old/New RG Geometric /Adaptive Hybrid MG
  - Monte Carlo Evolution of QCD: Sime implicit integrator
  - Graphene (again classical conformal LGT!)
  - Conformal Theories for LHC Higgsless models
  - Radial lattices for conform/string duals
  - Domain Wall/Overlap 5-d fore EXACT chirality

Other Dirac Operators (For quarks & Electrons)

- Multigrid & DD for Staggered and DW
- Eigenvector for Deflation and Disconnected Diagrams.
- Multi-scale Extension of Symplectic Integrators
- The Hexagonal Lattice for Dirac Electrons of Graphene
- Conformal Latices on Spheres.
- Topological Defects and the Spin Connection of GR
- Anti-deSitter Space and Conformal Invariance for BSM
- and condensed matter
- etc.

Outline:

One: Keeping pace with current platforms!

BG/Q(IBM) ,Titan (NVIDIA), Stampede (Intel),...

Two: Exposing and Exploiting Multiscale Physics

Protons, Nuclei and beyond to Higgs

Three: Conforming Physics to Hardware.

“physics” and “architecture” is multi-scaled but not necessarily compatible.

Future: New lattices LHC Physics & Condensed Matter