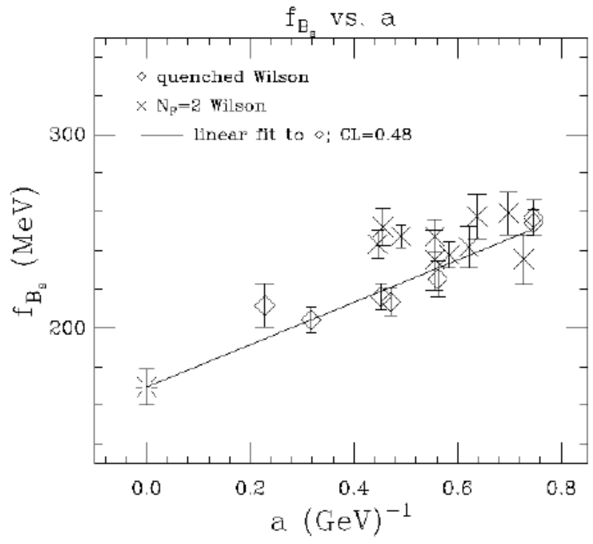
Computation in Lattice QCD

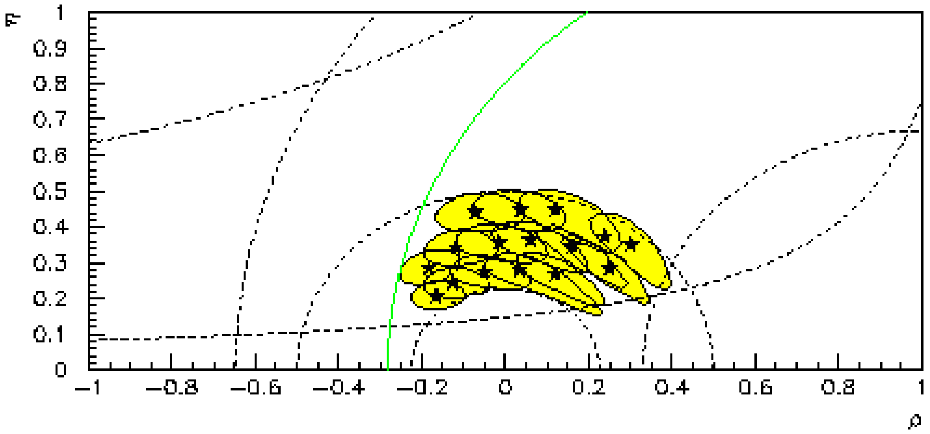
Paul Mackenzie
Fermilab
ACAT 2000
October, 2000

Physics Challenge.

Current errors in hadronic amplitudes,



dominate uncertainties in extracting CKM matrix elements:



Lattice QCD will reduce these to a few % over the next 3 – 10 years. Only $B \to \psi K => \beta$ achieves this accuracy without lattice QCD.

Reducing Calculational Errors.

Total errors ϵ fall slowly with \$, CPU time t:

$$\epsilon \sim 1/t^{1/p}$$

p=2,	statistical errors only
p$\sim$ 6,	statistical, lattice spacing, step size, m_q
p$\sim$ 10,	QCD thermodynamics

Suppose $p = 6$: CPU $\times 10 \Rightarrow 0.68\epsilon$.

$p = 4$: CPU $\times 10 \Rightarrow 0.56\epsilon$.

Accuracy goal for f_B:

BaBar: $\delta X_d \sim 1\%$

CDF: $\delta X_s < 0.5\%$.

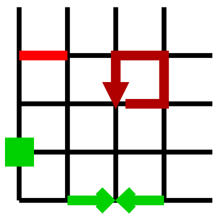
$\Rightarrow$ For V_{td}, V_{ts}, would like $\delta f_B < 1\%$ eventually.
(Now $\sim$25%.)

By brute force with existing methods,
CPU needs: $\times 25^4 \sim 400,000$.

A 10 TF or 100 TF computer provides a step forward, but for serious error reduction, need improved methods.

Lattice Monte Carlo Basics

I. Path integral: Monte Carlo



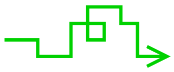
Gluons on links: $U_{x\mu}$

$S = \Sigma \mathrm{Tr} UUUU + \ldots$

Quarks on sites: ψ_x

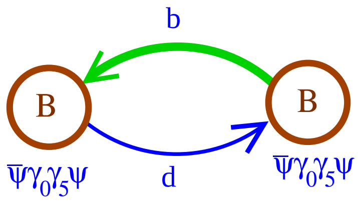
$S = \Sigma \overline{\psi}_x \gamma_\mu \frac{1}{2} (U_{x,\mu} \psi_{x+\hat{\mu}} - U^\dagger_{x-\hat{\mu};\mu} \psi_{x-\hat{\mu}}) + \ldots$

II. Quark propagators: Sparse matrix methods

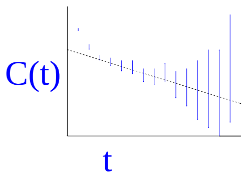


Invert $(i\not{\partial} - m)$ with conjugate gradient, minimum residual, ...

III. Hadron correlation functions



IV. Fit to exponentails



$$C(t) = A \exp(-Et) + A' \exp(-E't) + \ldots$$

Discretization errors can be reduced by

a) reducing the lattice spacing (takes money), or

b) adding next nearest neigbor interactions (takes thought). (Like trapezoidal rule $\Rightarrow$ Simpson's rule.)

Spectacular speed-ups have been achieved in pure QCD without light quarks.

Are such gains achievable in full QCD?

Potential.

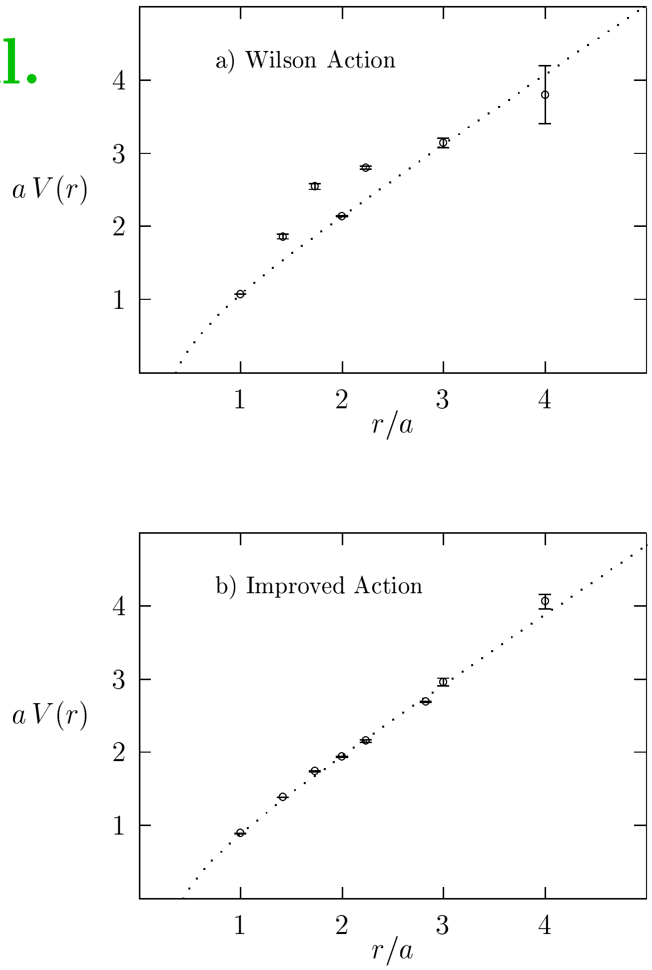


Figure 1: Static-quark potential computed on 6^4 lattices with $a \approx 0.4$ fm using the $\beta = 4.5$ Wilson action and the improved action with $\beta_{\rm pl} = 6.8$.

Errors reduced almost a factor of ten.

Computation speeded up by a factor of $10^4 - 10^5$.

Dimm et al.

Automated Lattice Perturbation Theory.

Same issues as continuum QCD, with additional challenges:

Nonlinearity of action $\Rightarrow$ Infinite series of vertices, of increasing complexity.

E. g.,

Further, no Lorentz symmetry to reduce number of possible terms.

$\Rightarrow$

- Automatic generation of vertices. (Lüscher and Weisz,...).
- Monte Carlo momentum integration (Vegas, ...).

Highly improved lattice actions require more computationally interesting departures from standard methods of evaluation. Propagators and vertex functions are pages long, before performing Lorentz algebra and Dirac traces.

$(x_1 + x_2 + \ldots + x_n) * (y_1 + y_2 + \ldots + y_m) * \ldots$

$\Rightarrow$ Do momentum Monte Carlo before trace algebra by creating C++ objects that return a numeric value for a given set of external momenta. (Morningstar; Trottier.)

Derivatives can be gotten via "automatic differentiation".

Objects represent functions and first n derivatives.

$sinp1 * sinp2$, $\exp(sinp1)$.

Overloaded operators understand the chain rule, etc.

Hardware.

Largest current machines:

- CP-PACS. Tsukuba. 300 GF installed, going to 900 GF. Built by Hitachi. 2048 nodes. Crossbar switch.
- QCDSP. Columbia. 300 GF installed. 320 boards, 64 dsp's per board. Hardwired 2D mesh for node-node comunications.
- APEmille. Rome. Several 250 GF and 125 GF machines this year. Custom processors, 8 to a board. Hardwired 3D mesh.

Older, smaller hardware:

- Fujitsu VPP500, KEK. (Big upgrade scheduled.)
- T3E, Edinburgh.
- ACPMAPS, Fermilab.

Future.

Columbia's QCDOC and the apeNEXT aim at $< \$1/\mathrm{MF}$ by 2003. Hardwired communications. Both custom chips. QCDOC incorporates IBM Power PC module on chip. Can use Power PC software.

Closest competing technologies: commodity clusters. \$20/MF today, naive guess \$5/MF in 2003, \$1.25/MF a few years later.

Discuss trade offs further after software section.

Software.

The concepts and algebra of grids, sites, and fields is conceptually simple and easily expressed in a high level language such as C++.

OBJECTS	OPERATIONS
sites s	grid: s' = s + d
directions d	
paths p	
umatrix u	SU(3) multiply: u3 = u2 * u1
quark q	q'=u*q
fields U, Q	Dirac equation: D Q = delta

Three related lattice codes based on these concepts.

1) NRQCD (Lepage)

Actual C++:

```
double WLoop(LinkField U, Path p){
  Grid g = U.grid();
  site x(g);
  double w = 0.;
  for_all(x)
    w+=ReTr(U(x,p));
  return(w/g.nsites);
  }
```

Not parallelized, but could be, as in the next example.

2) Canopy (Fermilab)

- Implemented in C rather than C++, but would be straightforward to translate.
- Parallelized. System, not user code, knows which sites are on which nodes.

```
s2 = move_site(s1, dir);
u = field_pointer(f, s2);
```

field_pointer routine looks up in system table whether s2 is on-node or not, executes a communications call if not.

Requires low latency off-node communications. (A few microseconds on a Pentium cluster.)

3) MILC code (MILC collaboration)

Based on the same concepts, but with user controlled scatters and gathers.

```
start_gather(field,size,direction,parity,dest);
```

Tolerates poorer latency.

Runs well on PC clusters.

Machines with hardwired communications.

QCDSP. C++ for high level code.

APE. Tao language ($\rightarrow$ C++).

- No switch. Cheaper, faster.
- Communications routines typically hand coded. Next-nearest neighbor look ups may be intricate.
- Some items usually done in software (changing the volume) may require reconfiguring hardware. Not all volumes possible.

Some convergence.

- Everyone uses higher level languages for much of their code.
- Everyone agrees large computers are neccesary.

Trade offs: Custom built vs. commodity clusters.

Clusters:

- Come with Linux, open source software.
- Same programs work on desktop and supercomputer, easy debugging.
- Easy to clone, upgrade, but
- How large a cluster can be effectively managed?
- How quickly will components really speed up?

Custom Built Machines:

- Best bang for the buck, but
- How big a diversion of resources away from physics and into computing is required to use such machines?
- How much do you lose due to inflexibility (e. g., by difficult tuning volume)?
- Late or less efficient than planned?

Plans.

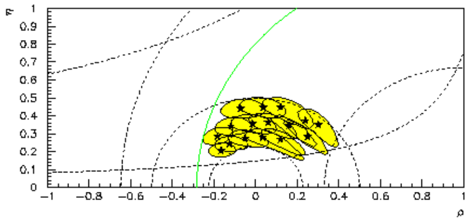
ECFA report and DOE whitepaper on lattice QCD call for teraflops of computing power within the next few years.

Europe. Endorsement of apeNEXT project. Aim: 10 TF @ 0.4 euro/MF.

US. Endorsement of Columbia/BNL QCDOC project and of large cluster projects at Fermilab and JLab. QCDOC aims at 10 TF @ $1/MF. US plan envisions further large hardware deployments, in a mix of QCDOC and clusters as dictated by experience with the initial projects.

Summary.

- Lattice QCD has achieved the ability to calculate nonperturbative quantities untouchable by any previous method.
- Some of these are essential to the analysis of very expensive experiments.



- Experiment will eventually require over an order of magnitude improvement in accuracy over what is now achieved.

  ⇒ Orders of magnitude improvement in combination of CPU power and power of methods.
- Large, well organized efforts are underway around the world to meet these challenges.