QCD on the lattice

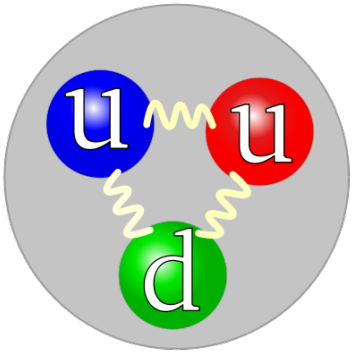
S. Katz and F. Pittler et al

Eötvös Loránd University
Budapest, Hungary

May 29, 2014

Quantum Chromodynamics (*QCD*)

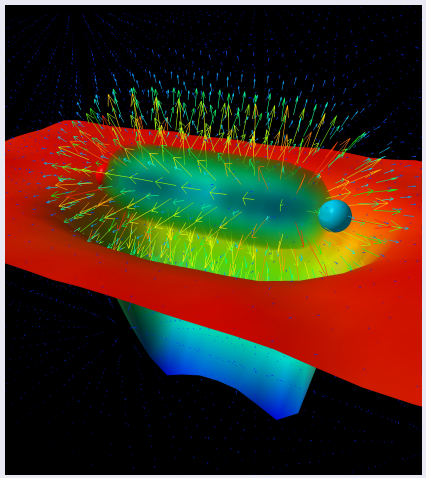
- Field theoretic description of the strong interaction
- Theory of quarks and gluons
- They are the building blocks of hadronic matter, like proton
- They come in three different colors
- Three quarks are bound together to form a proton



Basic properties of QCD

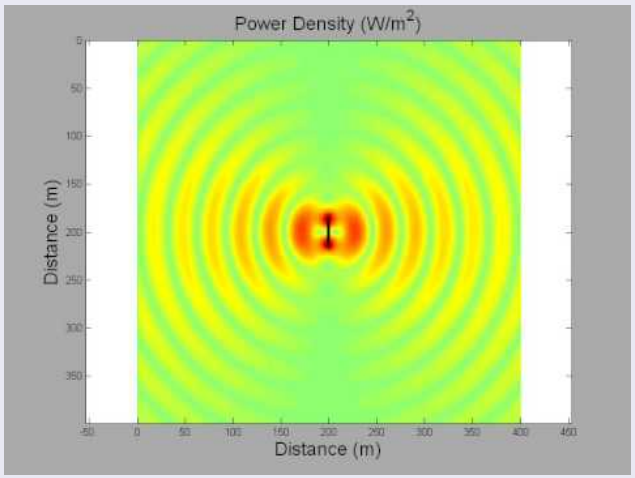
Confinement

- Free quark cannot be observed
- The interaction at large distances is very strong



Meson in QCD

http://www.physics.adelaide.edu.au



Dipole field in electrodynamics

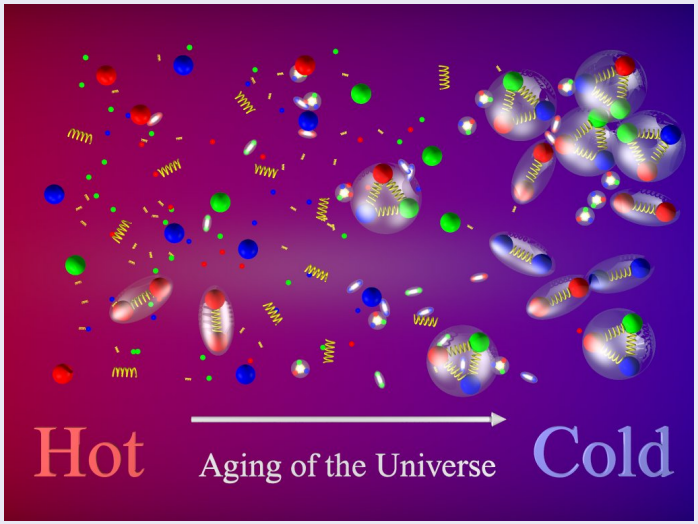
- Flux tube between the two quarks, the energy density is constant in the tube!

Basic properties of QCD

Asymptotic freedom

- In high energy hadronic collisions the interaction between the quarks is small
- At high energy the quarks and gluons form a so-called quark gluon plasma

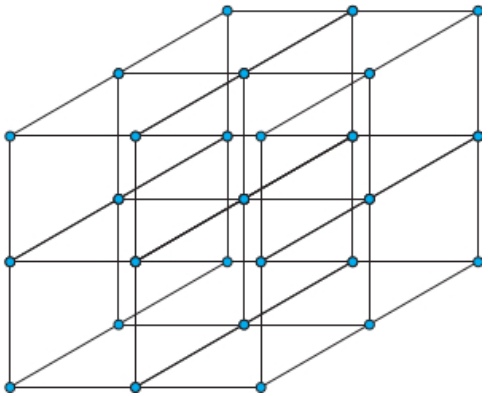
Transition between the two forms of strongly interacting matter



- To study the transition we need a non-perturbativ definition of QCD.

Lattice QCD

- Discretize the space-time on a hypercubic Lattice
- Quarks: complex 3D vectors on the sites: $\psi(x) = \begin{pmatrix} \psi_1(x) \\ \psi_2(x) \\ \psi_3(x) \end{pmatrix}$
- Gluons: SU(3) matrices on the links $U_\mu(x)$
- Finite number of degrees of freedom : stat. mech. system.
- Computation of observables $O(U, \psi, \bar{\psi})$ by taking into account all possible configurations:



$$\frac{1}{Z} \int \prod_{x,\mu} \mathrm{d}U_\mu(x)\, \mathrm{d}\psi(x)\, \mathrm{d}\bar{\psi}(x)$$
$$O(U, \psi, \bar{\psi}) \exp(-S(U, \psi, \bar{\psi}))$$

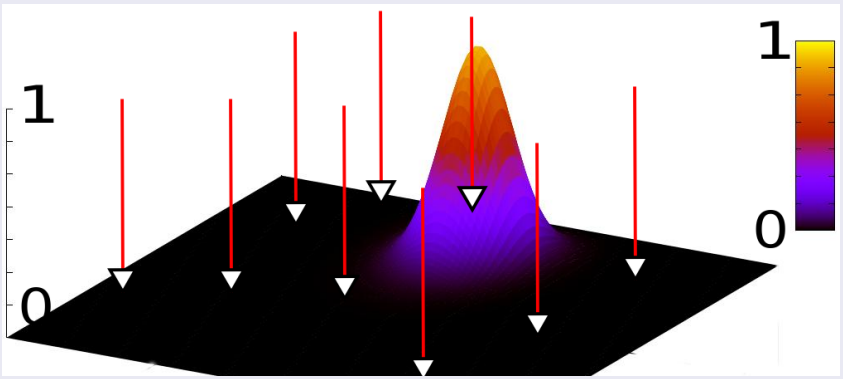
S contains the form of the interaction

Monte-Carlo integration and Importance sampling

- In a typical simulation number of integrations scales with the volume: $N_s \sim O(50), N_t \sim O(100) \rightarrow V \sim O(10^7)$
- Direct evaluation is unfeasible

Monte Carlo methods and importance sampling

- Selecting points randomly in the configuration space
- Average O over these configurations with weight $P(U)$
- Problem: Most configurations will have small weight



- Solution: Sampling the configurations with $P(U)$.
- $\langle O \rangle = \sum_{i \in \text{ all config}} O(i)$

Parallel improvement

- Even in this case the problem is computationally demanding
- Today's trend: Computing using many cores

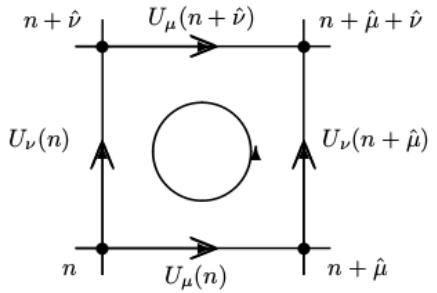
Locality

- All field theoretic models have this property
- Common task: Computing plaquettes

$$P(x) = U_\mu(x)\, U_\nu(x+\mu)\, U_\mu^\dagger(x+\nu)\, U_\nu^\dagger(x)$$

Translational invariance

- We have to do the same operation on all sites



- Communication only between neighbors

Lattice QCD on the GPU

- We have a lattice QCD code in CUDA
- Each site is processed by one cuda thread
- Global sum is needed in

$$\sum_{x \in \text{ all sites}} P(x)$$

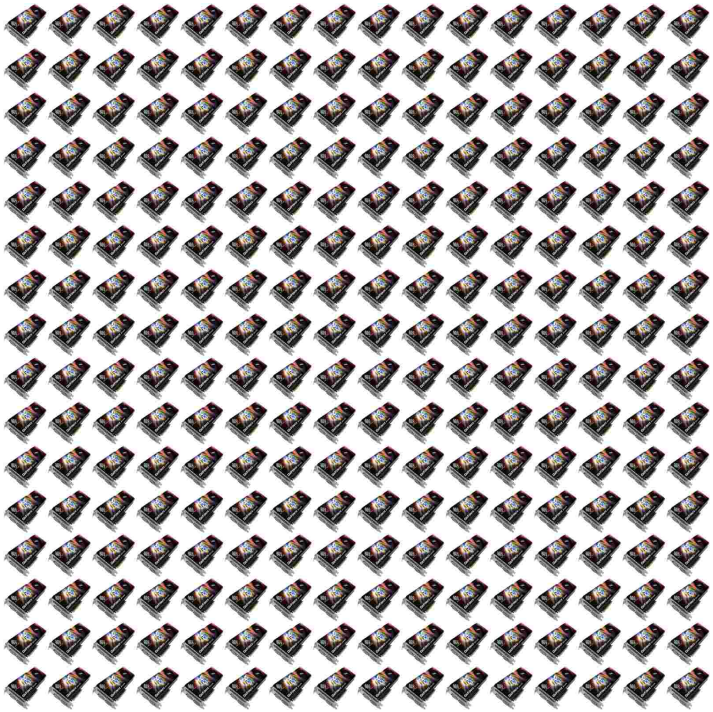
$$\langle \psi | \chi \rangle = \sum_{x \in \text{ all sites}} \psi^{\dagger}(x) \chi(x)$$

Graphical cards at the Eötvös Loránd University

Nvidia 670 Kepler architecture

- 1344 cores
- 980 MHz clock speed
- 2048 MB memory
- 192 $\frac{GB}{s}$ mem. bandwidth
- 3.9 $\frac{Tflop}{s}$ peak performance
- 250 $\frac{Gflop}{s}$ max. performance with our code

Graphical cards at the Eötvös Loránd University



GPU cluster

- 176 nodes
- 352 GPUs: GTX 670 and GTX 470
- 387072 cores
- 1.1 $\frac{Pflop}{s}$ peak performance
- 78 $\frac{Tflop}{s}$ max. performance with our code

GPU cluster at the Eötvös Loránd University

Computations in Lattice QCD

Dirac operator : $\mathrm{D}(U)+m$

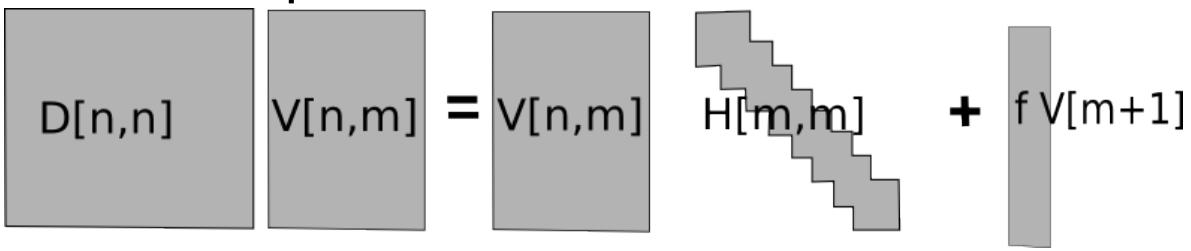
- Fermionic action is bilinear: $S_f = \bar{\psi}(D(U)+m)\psi$
- $D(U)$ is a large and sparse matrix
- We need to compute $D^{-1}\chi$ for many χ
- Conjugate gradient algorithm (CG)
- The large number of small eigenvalues slows the convergence of the CG
- Solution: Explicitly determine and deflate the low modes
- Krylov-Schur algorithm

Krylov-Schur algorithm

- Sparse eigenvalue solver, only needs a multiplication routine
- Determines only a part of the full spectrum
- From a random vector v we generate a Krylov subspace ($\mathscr{V}$):

$$v, Dv, D^2v \cdots D^m v \quad m \ll n$$

- Using Gram Schmidt orthogonalization (GSO) we obtain the decomposition:



- H is the projection of D on $\mathscr{V}$.
- H contains the best approximation of the eigenvalues of D in $\mathscr{V}$: $Hy_i = \theta_i y_i \rightarrow$ Restore approx. eigenv. of D from y_i

Krylov-Schur continued

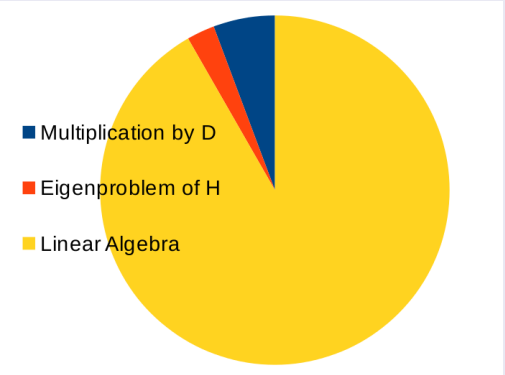
- Problem: m vectors (U) have to be stored
- m has to be as small as possible $\rightarrow$ slow convergence

Solution

- Restart: Truncate to order p and extend to order m

Parallelization

- Most of the time is spent on Gram Schmidt Orthogonalization
- Parallelize the linear algebra:
  1. Multiplication of a vector with scalar
  2. Addition of two vectors

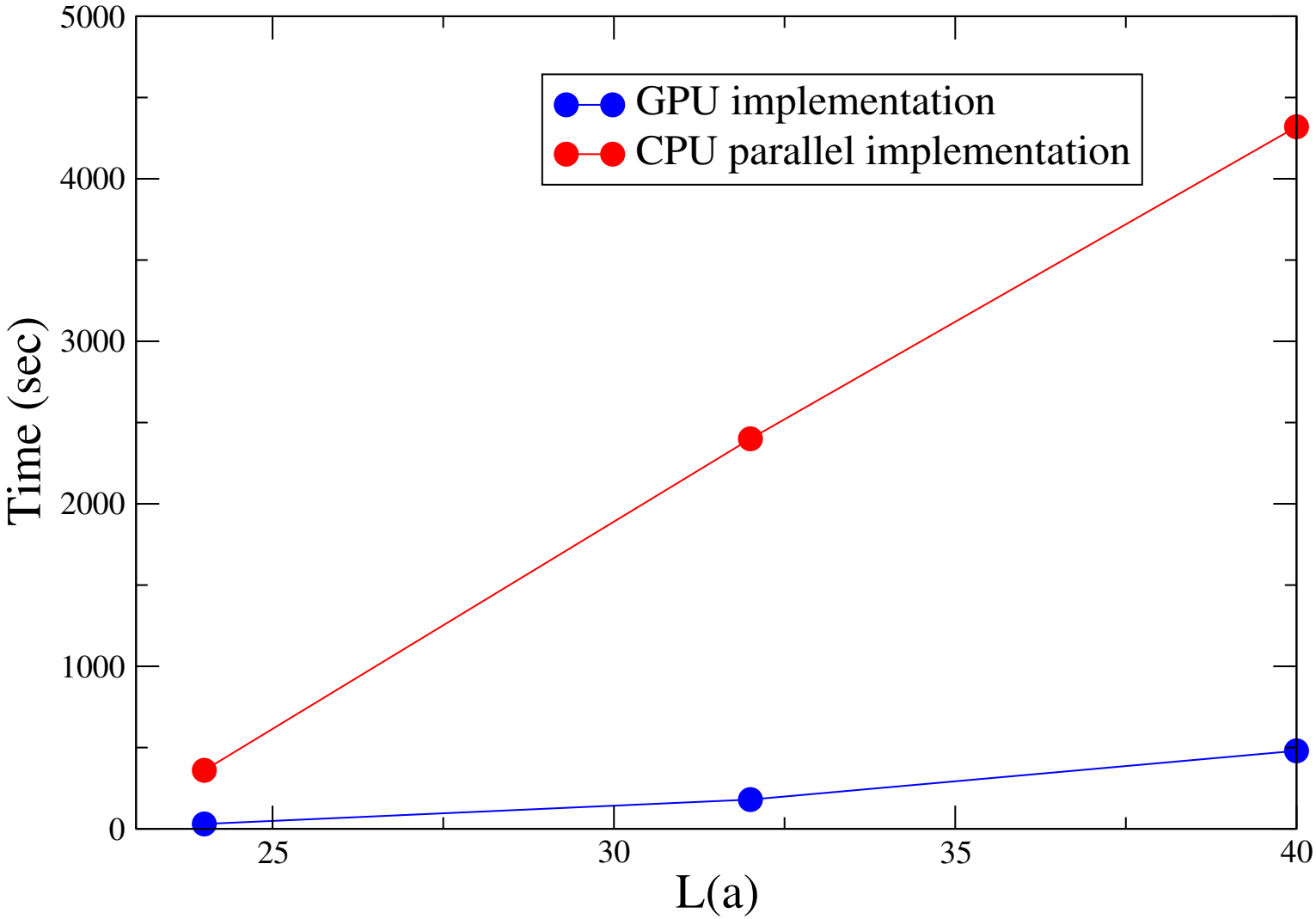


CUDA implementation

Vector plus scalar times vector kernel routine

```
__global__ void kernel_latvec_Vp_StV ( int par,
    handler v0, double s, handler v1 )
{
    //Get the index of the vector component
    int tid= blockIdx.x*blockDim.x+threadIdx.x;
    if ( tid>=CONST(nsites)[par] ) return;
    int i= tid + CONST(oddoffs)[par];
    //Move the components to the registers
    double2 rvec0[3], rvec1[3];
    d_read_fvec(rvec0, v0.pt+i, v0.stride);
    d_read_fvec(rvec1, v1.pt+i, v1.stride);
    //Do the arithmetic
    FVEC_V_VpStV(rvec0, rvec0, s, rvec1);
    //Write back to the global memory
    d_write_fvec(v0.pt+i, rvec0, v0.stride);
}
```

Performance of the parallel CPU and GPU implementation



Summary

- We presented the GPU cluster at the Eötvös Loránd University
- We showed one piece of our code: The Krylov-Schur algorithm

Plans:

- Further code optimization
- Implementing the domain decomposition based multigrid method for the inverter

- Thank you for your attention!