QCD on the lattice

Dr. Ferenc Pittler

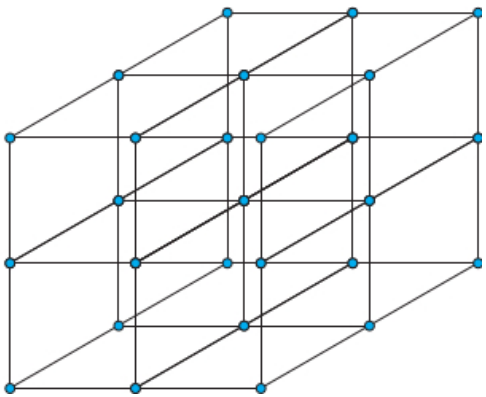
MTA-ELTE Lattice Gauge Theory Group
Budapest, Hungary

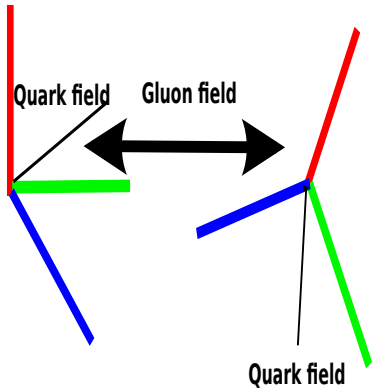
in collaboration with:
Sz. Borsányi,Y. Delgado,S. Dürr,Z. Fodor,S.D. Katz,T.G. Kovács,S.Krieg,T. Lippert,D. Nógrádi,A. Pásztor, K.K. Szabó,B.C. Tóth

May 21, 2015

Lattice Quantum Chromodynamics (*QCD*)

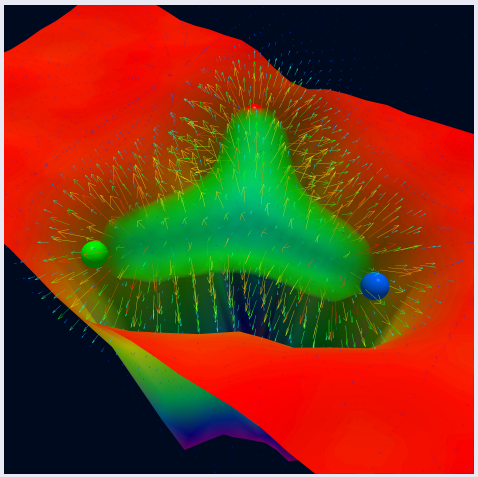
- First principles calculations
- Field theory on a discrete space-time lattice
- Building blocks:
  - Quarks: Complex 3d vectors on the sites $\psi(x) = \begin{pmatrix} \psi_1(x) \\ \psi_2(x) \\ \psi_3(x) \end{pmatrix}$
  - Gluons: SU(3) matrices on the links $U_\mu(x)$





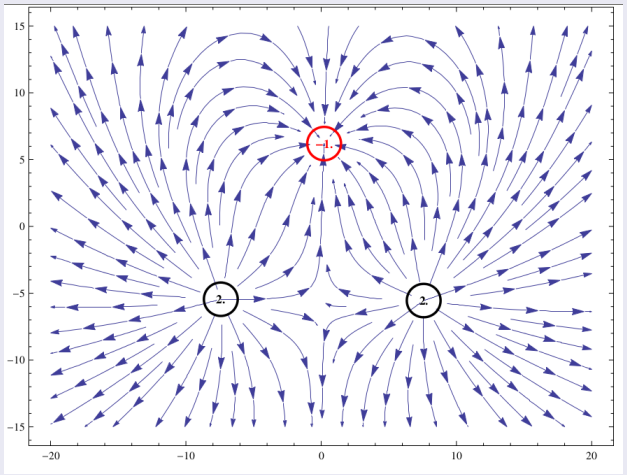
Basic properties of QCD

Color and electric field of three charges



color field of three quarks

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



electric field of three quarks

Mathematica

Confinement

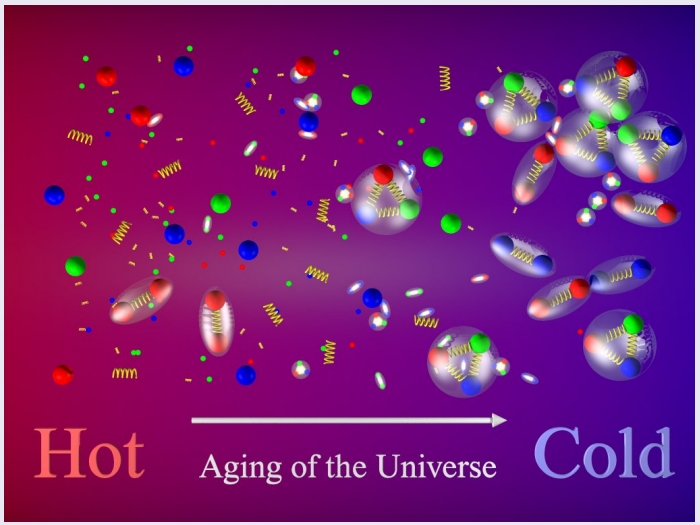
- Free quark cannot be observed
- The interaction at large distances is very strong

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

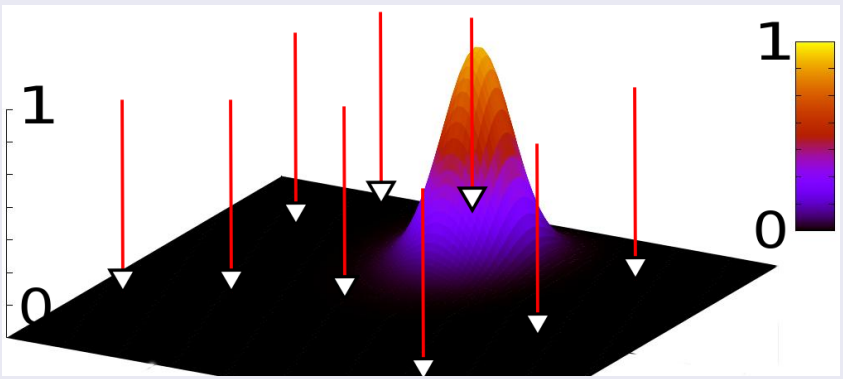


Monte-Carlo integration and Importance sampling

- Computations of obversables (O) by taking into account all possible configurations with weight $P(U)$
- In a typical simulation: $O(10^7)$ dimensional integrals
- 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

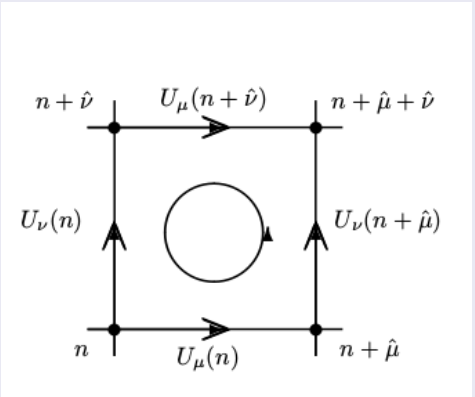
Locality

- 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

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

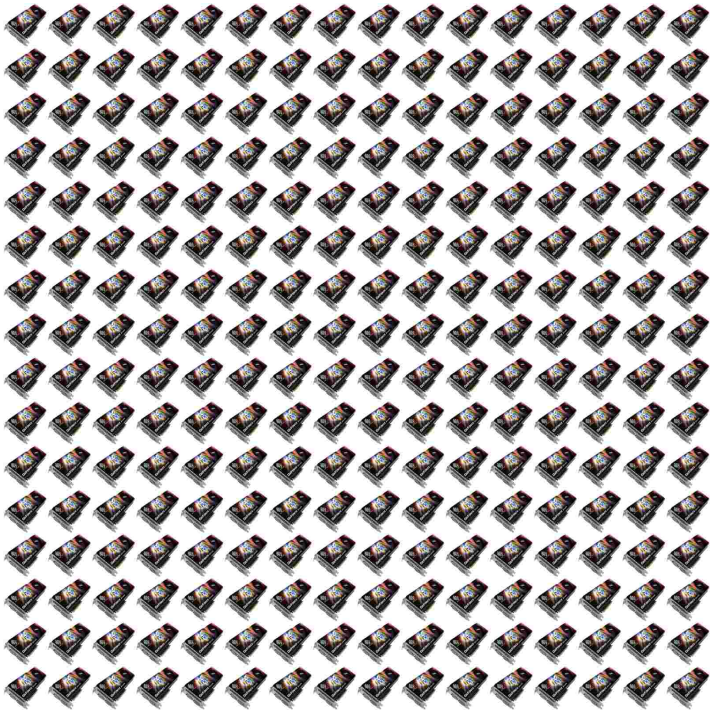
$$\langle \psi | \chi \rangle = \sum_{x \in \text{ all sites}} \psi^{\dagger}(x) \chi(x)$$

Graphical cards at the Eötvös Loránd University

Nvidia 770 Kepler architecture

- 1536 cores
- 1046 MHz clock speed
- 2048 MB memory
- 224 $\frac{GB}{s}$ 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

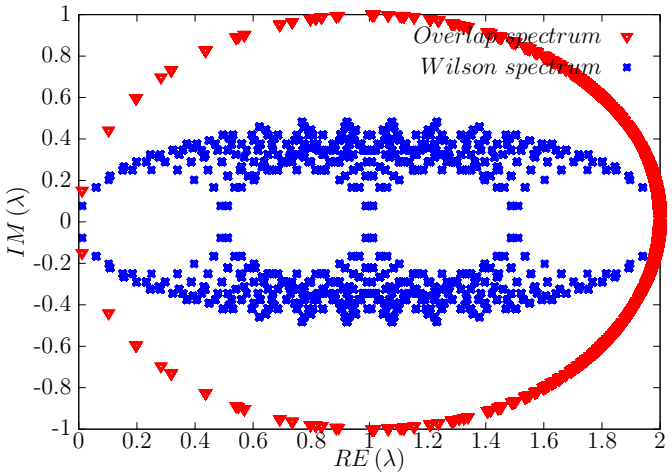
- 176 nodes
- 352 GPUs: GTX 470/670/770
- 387072 cores
- 1.1 $\frac{Pflop}{s}$ peak performance
- 78 $\frac{Tflop}{s}$ max. performance with our code

GPU cluster at the Eötvös Loránd University

Computations in Lattice QCD

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

- Fermionic action is bilinear: $S_f = \bar{\psi}\left(D(U)+m\right)\psi$
- Many possibilities for D.



$$D_w(x,y) = \sum_{\mu=\pm1}^{\pm4} \left(1+\gamma_\mu\right)\delta_{x,y+\mu}$$

$$D_{overlap}(x,y) = 1+\gamma_5 \mathrm{sign}\left(\gamma_5 D_w(x,y)\right)$$

- We choose the best $D(U)$ available:*Overlap*
- Drawback: expensive to compute and extremely expensive to invert.

Inversion, solving $D_{overlap}x = b$ for x

Iterative methods

- All methods essentially work in a Krylov subspace:
- We generate the sequence

$$\mathcal{V} = \{b, Db, D^2 b \cdots D^m b \quad m \ll n\}$$

- For example in GMRES the new approximation to the solution will be $x_1 \in \mathcal{V}$ for which

$$||r_1||_2 = ||Dx_1 - b||_2 \text{ is minimal}$$

- The error after this step is $e = x - x_1$
- To obtain a correction to x_1 we have to solve a similar equation:

$$D \cdot e = D \cdot x - D \cdot x_1 = b - D \cdot x_1 = r_1$$

Domain Decomposition Multigrid

Iterative Methods: Smoother

- In the Krylov subspace the high modes of D dominate
- The components of the error in the direction of low modes decreases much more slowly as the iteration proceeds
- Smoother is very efficient if the error contains high frequency components

What to do with the low frequency components of the error?

- After smoothing restricting the residual to a coarser grid
- Low components of the error appear more oscillatory on the coarse grid
- Smoothing on the coarse grid
- Correct the error on the fine grid with the interpolation of the coarse grid solution

Example:Restriction

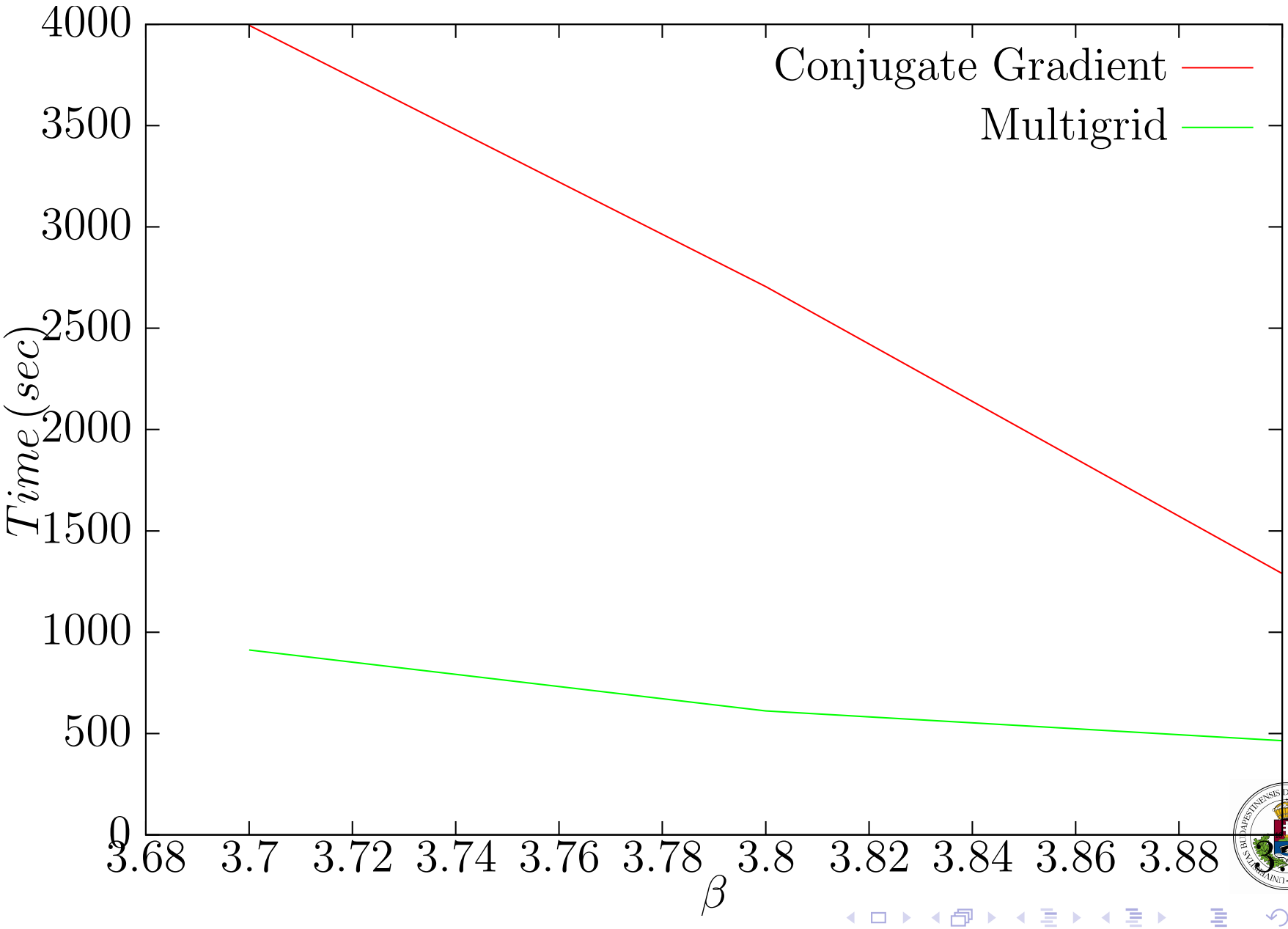
- Goal: Move to a basis where the low modes dominate

Restriction Frommer et al.[2013], Luscher[2007]

- Pick n crude approximation of the low modes (ϕ)
- Project them to the blocks:$\phi_n^b(x) = \begin{cases} \phi(x) & x \in b \\ 0 & \text{otherwise} \end{cases}$
- Orthogonalize them in each block to get a ”much higher dimensional” space
- Project the original residual to this new basis

$$\psi_B(b)[n] = \sum_{x \in b} \phi_n^b(x)\,\psi(x)$$

CPU results


Conjugate Gradient
Multigrid
Time (sec)
β
4000
3500
3000
2500
2000
1500
1000
500
0
3.68
3.7
3.72
3.74
3.76
3.78
3.8
3.82
3.84
3.86
3.88
3.9

CUDA implementation details

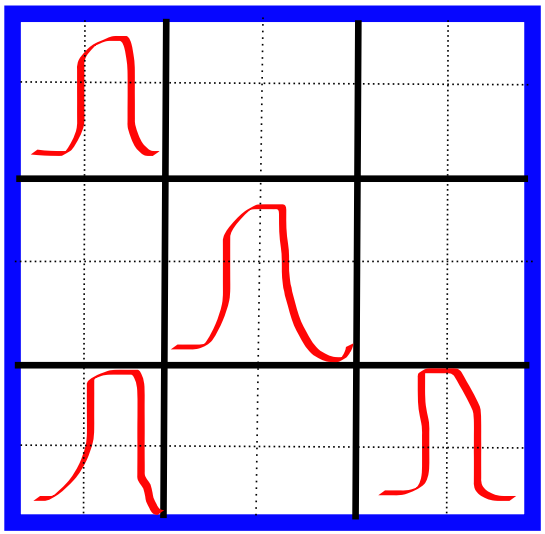
- Assign to each lattice site a cuda thread
- Organize them in such a way that each physical lattice block corresponds to a cuda thread block
- Use available reduction routines to compute scalar products within a block

Summary

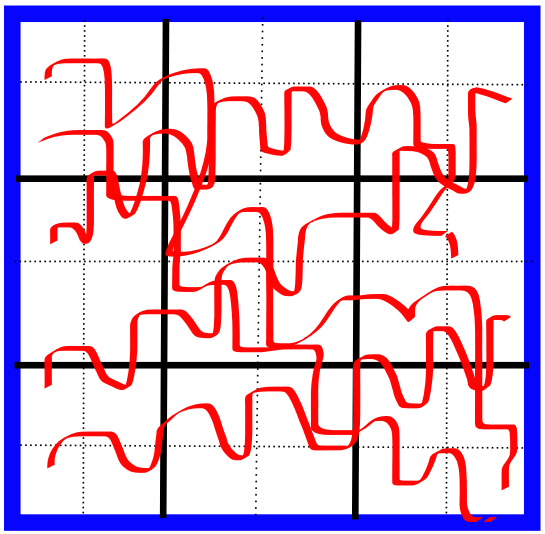
- This work is in progress
- Different parts of the code are ready
- The whole program for the overlap inversion is not done yet
- From the CPU experience we expect a factor 4 gain

- Thank you for your attention!

Backup slide:Blocking efficiency for localized and delocalized eigenmodes



Localized



Delocalized