Twisted mass fermions on GPUs
An application to the LOEWE cluster at CSC

Lars Zeidlewicz

–

In collaboration with Matthias Bach, Owe Philipsen and Christopher Pinke

Institut für Theoretische Physik
Goethe-Universität Frankfurt

Lattice QCD and high performance computing on GPUs
March 11, 2011

The plan of this talk

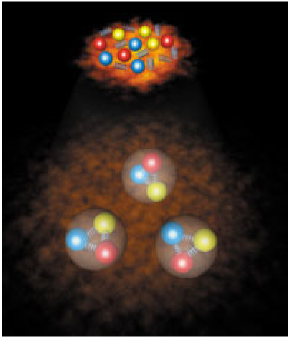
Introduction

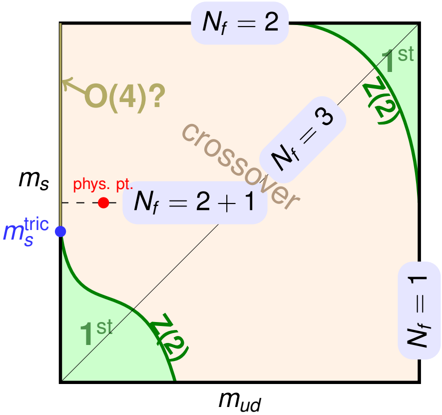
- Finite temperature QCD
- Twisted mass lattice fermions

Status Report

- Solver
- Heatbath
- Using the LOEWE-CSC cluster

Physical objects of interest


GSI

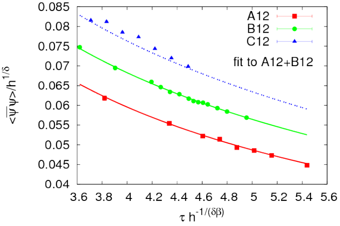


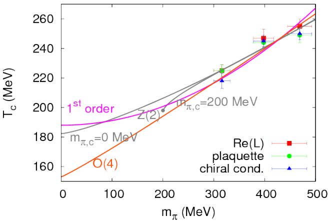
properties of the
Quark-Gluon-Plasma
$T > T_c$

properties of the thermal
transition
$T \approx T_c$, possibly $\mu \neq 0$

Chiral limit

What is the nature of the $N_f = 2$ chiral transition?





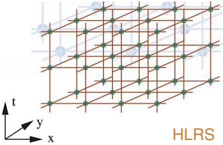
- Computational cost grows with inverse power of pion mass
- Infer chiral limit properties from universal scaling behaviour
- In any case: Small pion masses necessary

tmfT 2011

Lattice QCD at finite T

- Discretise spacetime by a hypercube of size $N_\sigma^3 \times N_\tau$ and lattice spacing a
- Temperature:

$$T = \frac{1}{aN_\tau}$$



- Carefully control cutoff effects:

$$F\left(\frac{1}{N_\tau}\right) = F_{\text{continuum}} + F_{(2)}\frac{1}{T^2 N_\tau^2} + \dots$$

- Need aspect ratio N_σ / N_τ as large as possible $\Rightarrow$ large spatial volumes

Which lattice fermions?

Staggered fermions

+ Computationally cheap
+ Results at physical masses and below

- Theoretical doubts (“rooting”)

Wilson fermions

+ Theoretically sound
+ Improvement possible: clover & maximally twisted mass

- Explicitly broken chiral symmetry
  - $\Rightarrow$ linear cutoff effects
  - $\Rightarrow$ additive mass renormalisation

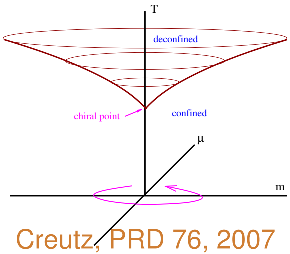
Chiral fermions

+ Theoretically sound
+ Lattice chiral symmetry

- Very expensive (non-local or 5 dimensional)

Twisted mass fermions: continuum

Frezzotti, Grassi, Sint, Weisz



Creutz, PRD 76, 2007

Chiral symmetry transformation in $N_f = 2$ flavour space:

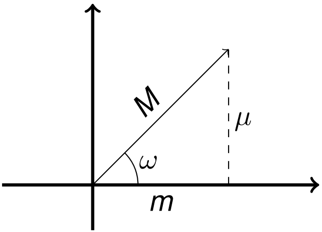
$$\psi = \begin{pmatrix} u \\ d \end{pmatrix} \longrightarrow \chi = e^{-i\gamma_5\tau^3\omega/2}\psi$$

$$M\overline{\psi}\psi \longrightarrow \overline{\chi}\left(\underbrace{M\cos\omega}_{m_0} + i\gamma_5\tau^3\underbrace{M\sin\omega}_{\mu_0}\right)\chi$$

- Quark mass:

$$M_R = \sqrt{Z_m^2 m_0^2 + Z_\mu^2 \mu_0^2}$$

- Maximal twist:
$m_0 = 0 \Rightarrow \omega = \pi/2$



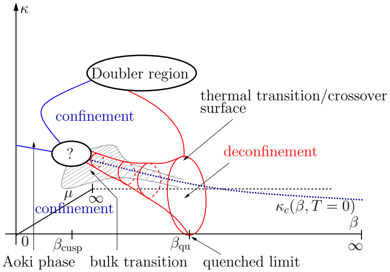
Twisted mass fermions: lattice

Frezzotti, Grassi, Sint, Weisz

fermion matrix with twisted mass term

$$M_{\rm tm} = 1 + 2\mathrm{i}\kappa a\mu_0\gamma_5\tau^3 - \kappa D_W[U]$$

- $N_f = 2$ mass-degenerate Wilson type fermions (inclusion of strange & charm possible)
- Hopping parameter: $\kappa = 1/(2am_0 + 8r)$
- Non-trivial bare parameter phase space (Aoki phase & Sharpe-Singleton plane)



tmfT, PRD 80, 2010

Fermion determinant

- Twisted mass acts as infrared regulator for the fermion matrix
- Flavour determinant can be evaluated explicitly:

$$\begin{aligned}\mathrm{Det}_f\, M_{\mathrm{tm}} &= \mathrm{Det}_f\left(M_W[U]\,\mathbf{1}_f + 2\mathrm{i}\kappa a\mu_0\gamma_5\tau^3\right)\\ &= M_W[U]\,M_W^\dagger[U] + 4\kappa^2(a\mu_0)^2\end{aligned}$$

- Original motivation for twisted mass term

Improvement

- Maximal twist is achieved for vanishing untwisted mass component
- Hopping parameter must be tuned to its critical value $\kappa_c(\beta)$
- Maximal twist ensures automatic $\mathcal{O}(a)$ improvement, i. e. absence of linear cutoff effects in physical observables
  Frezzotti, Rossi, JHEP 0408

Example: Free dispersion relation

$$E(\mathbf{p}) = \sqrt{\mathbf{p}^2 + M^2} - a\frac{1}{2}\frac{M^3 \cos\omega}{\sqrt{\mathbf{p}^2 + M^2}} + \mathcal{O}(a^2)$$

tmLQCD package

- MPI-parallel CPU code for twisted mass fermions exists (freely available)
- `C` code developed within the European Twisted Mass Collaboration
  Jansen, Urbach, Comp. Phys. Commun., 180, 2009
- Algorithm: Mass preconditioning, multiple time-scale integrators, even-odd preconditioning...
  Jansen et al. Comp. Phys. Commun., 174, 2006
- Both $N_f = 2$ and $N_f = 2 + 1 + 1$ are implemented
- ETMC have used the two-flavour code down till 280 MeV pion masses
  ETMC, JHEP 1008
- We test the code on the CPU part of LOEWE-CSC already

Motivation: LOEWE-CSC

- Top500 rank 22 in November 2010
  (285 Tflops, rank 2 in Germany)
- Rank 8 of green IT machines
  (741 Mflops/W, rank 4 in Germany)
- Approximately 800 nodes constitute a CPU/GPU hybrid structure
  - 2 AMD Opteron 12 core ($\rightarrow$ 24 cores per node)
  - 1 AMD ATI Radeon 5870 (Cypress)
- Infiniband connected

CSC

Twisted mass simulations using OpenCL

Wishlist

- Have an OpenCL heatbath working
- Have a solver in OpenCL that can be applied to both CPU and GPU
- Have an OpenCL HMC
- Expand fermionic part to $N_f = 2 + 1 + 1$ PHMC
- Have a hybrid Hybrid Monte-Carlo, especially for the 24 core / 1 GPU architecture of a LOEWE-CSC node
- Ultimately: Have a program that can take advantage of the complete LOEWE-CSC cluster

Our Program

- New architecture & programming language
  ⇒ write code from scratch
- OpenCL is rapidly evolving
- Embed OpenCL code in C/C++ host environment

```
class opencl {
public:
opencl(cl_device_type wanted){init(wanted);};
~opencl(){finalize();};
...
}
```

Our Program

- New architecture & programming language
  ⇒ write code from scratch
- OpenCL is rapidly evolving
- Embed OpenCL code in C/C++ host environment

```
..
#ifdef _RECONSTRUCT_TWELVE_
collect_options<<'' -D_RECONSTRUCT_TWELVE_'';
#endif
#ifdef _USEDOUBLEPREC_
collect_options<<'' -D_USEDOUBLEPREC_'';
#endif
...
clerr = clBuildProgram(clprogram,1,
            &device,buildoptions.c_str(),0,0);
```

Our Program

- New architecture & programming language
  $\Rightarrow$ write code from scratch
- OpenCL is rapidly evolving
- Embed OpenCL code in C/C++ host environment

- Compiler switches for single/double, 12-double reconstruction of $SU(3)$ matrices...
- Development started in December 2010
- OpenCL frontend class: $\sim$ 2200 lines of C++ code
- Kernel code: $\mathcal{O}(3k)$ lines
  (no BLAS libraries)

Solver in OpenCL

Status

- Target: Even-odd preconditioned solvers
  BiCGStab, CG,...
- Achieved: Solver in OpenCL, no even-odd yet
- We have some experience from an existing CUDA solver
  (F. Burger, HU Berlin)
- We have used the CUDA solver on the *gpu-scout* of CSC
  (NVIDIA Teslas)

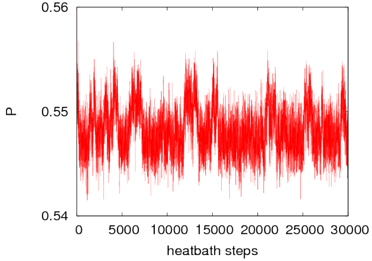
Current Limits

- For Cypress: RAM = 1 GB $\Rightarrow N_\sigma^3 \times N_\tau \leq 10^6$
  e. g. $42^3 \times 12$, $38^3 \times 16$
- double precision, 12-double reconstruction, even-odd,
  BiCGStab

Heatbath

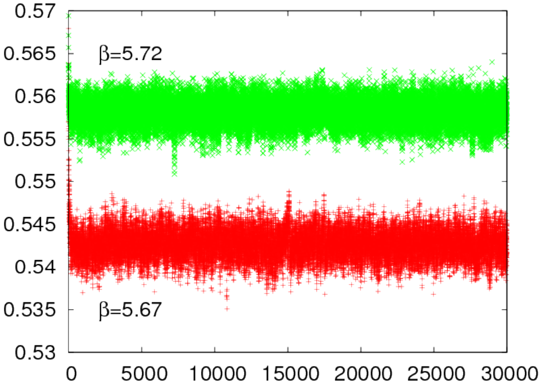
Heatbath algorithm for $SU(3)$ pure gauge theory

- 12-double reconstruction for gauge fields (Clark et al.)
- For Cypress GPUs: 1 GB $\Rightarrow N_\sigma^3 \times N_\tau \leq 5 \cdot 10^6$
  e. g. $76^3 \times 12$, $66^3 \times 16$
- Non-fermionic applications in our group:
  QGP properties, hydrodynamics
- Use it independently and as first step towards an HMC



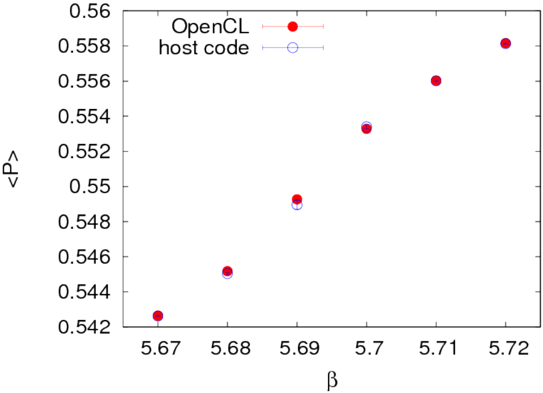
Checking the Heatbath

- Comparison to finite temperature results
  Boyd et al. Nucl. Phys. B469, 1996
- $N_\tau = 8$, $N_\sigma = 16$, thermal transition at $\beta \approx 5.69$



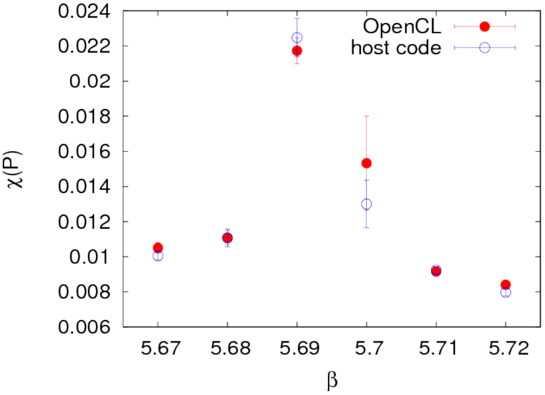
Checking the Heatbath

- Comparison to finite temperature results
  Boyd et al. Nucl. Phys. B469, 1996
- $N_\tau = 8$, $N_\sigma = 16$, thermal transition at $\beta \approx 5.69$



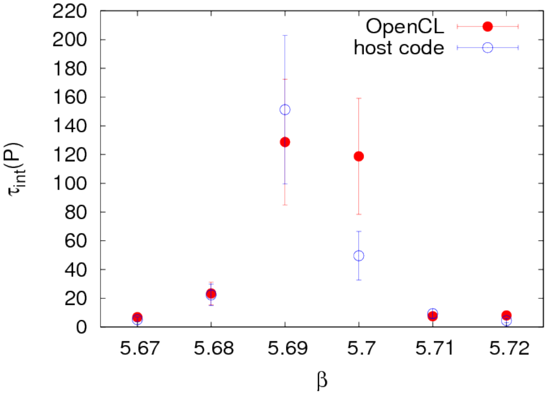
Checking the Heatbath

- Comparison to finite temperature results
  Boyd et al. Nucl. Phys. B469, 1996
- $N_\tau = 8$, $N_\sigma = 16$, thermal transition at $\beta \approx 5.69$



Checking the Heatbath

- Comparison to finite temperature results
  Boyd et al. Nucl. Phys. B469, 1996
- $N_\tau = 8$, $N_\sigma = 16$, thermal transition at $\beta \approx 5.69$



Hybrid OpenCL HMC

- One GPU is memory limited
- Try to distribute calculation on CPUs and attached GPU efficiently
  LOEWE-CSC 24 cores & 1 GPU
- Distributing the calculation over several nodes leads to extra communication
- How can we make optimal use of the LOEWE-CSC cluster beyond one node?
- Possibly different algorithms are more suitable for that structure (e. g. DDHMC)

Conclusions and future plans

lots to do…

- Heatbath is ready, complete benchmarking
- Solver code needs further checks
- Probably, heavy performance optimisation possible
- Hybrid code must be implemented. We have not addressed questions concerning the global cluster structure yet

perspectives

- Fermion matrix can be exchanged
- OpenCL is programmed device independently

Thank you