Towards Exascale Computing for Lattice QCD

Masterarbeit in Physik

Martin Ueding
mu@martin-ueding.de

Wednesday, 2017-08-09 15:15
Seminar Room 1, HISKP

Motivation

Demands

- Physical masses
- Larger physical systems
- Faster solutions

Computers

- More complex machines
- Hierarchy (Nodes, Sockets, Cores, Threads, SIMD)
- Heterogeneity (CPU, GPUs)

Outline

Today

- Lattice Theory
- Hybrid Monte Carlo
- QPhiX
- Performance (not Exascale, yet)

Not in this Talk

- Meson Masses
- Chroma

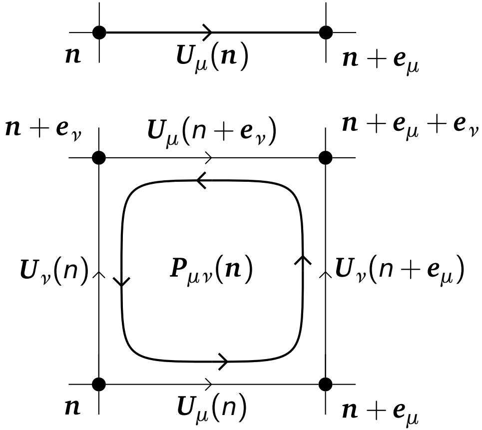
QCD Lagrangian density with gluon tensor $\boldsymbol{G} = \mathrm{d}\boldsymbol{A} + \boldsymbol{A} \wedge \boldsymbol{A}$

$$L = \frac{1}{2} \operatorname*{Tr}_{\text{color}} \left(\boldsymbol{G}_{\mu\nu} \boldsymbol{G}^{\mu\nu} \right) + \bar{\psi} \underbrace{\left(\mathrm{i}\not{D} - \boldsymbol{m} \right)}_{\boldsymbol{M}} \psi \,,$$

Path Integral with action $S = \int \mathrm{d}^4 x \, L$

$$\langle 0 | T \hat{O} | 0 \rangle = \frac{\int \mathscr{D}\boldsymbol{A} \, \mathscr{D}\bar{\psi} \, \mathscr{D}\psi \, O(\boldsymbol{A}, \psi, \bar{\psi}) \, \exp(\mathrm{i} S(\boldsymbol{A}, \psi, \bar{\psi}))}{\int \mathscr{D}\boldsymbol{A} \, \mathscr{D}\bar{\psi} \, \mathscr{D}\psi \, \exp(\mathrm{i} S(\boldsymbol{A}, \psi, \bar{\psi}))} \,,$$

- Fermions on the sites
- Wilson term to remove doublers
- Gauge field on the links
- Gauge invariant trace of plaquette



Hybrid Monte Carlo

Euclidian Time $t = \mathrm{i}\tau$

$$\langle O \rangle = \frac{1}{Z} \int \mathscr{D}\boldsymbol{A}\, \mathscr{D}\bar{\psi}\, \mathscr{D}\psi\, O(\boldsymbol{A}, \psi, \bar{\psi}) \exp(-S(\boldsymbol{A}, \psi, \bar{\psi}))$$

Importance sampling of probability density

$$P(S) = \frac{\exp(-S)}{Z}$$

Metropolis Algorithm (Metropolis et al. 1953)

$$P(y_i \to y_j) = \min(1, \mathrm{e}^{-\Delta S})$$

Hybrid Monte Carlo

Pseudo Fermions

Grassmann fields ψ not feasible on computer, only c-number fields ϕ possible.

Hybrid Monte Carlo

Pseudo Fermions

Grassmann fields ψ not feasible on computer, only c-number fields ϕ possible.

Single quark flavor, $M = -\not{D}_{\mathrm{W}}/2 + 4 + m$. Integrate fields out

$$\int \mathscr{D}\bar{\psi}\, \mathscr{D}\psi\, \exp(-\bar{\psi} M \psi) = \det(M)$$

and back in (Fucito et al. 1981; Weingarten and Petcher 1981)

$$\det(M) = \int \mathscr{D}\bar{\phi}\, \mathscr{D}\phi\, \exp(-\bar{\phi} M^{-1} \phi)$$

Hybrid Monte Carlo

Pseudo Fermions

Grassmann fields ψ not feasible on computer, only c-number fields ϕ possible.

Single quark flavor, $M = -\not{D}_\mathrm{W}/2 + 4 + m$. Integrate fields out

$$\int \mathscr{D}\bar{\psi}\,\mathscr{D}\psi\ \exp(-\bar{\psi}M\psi) = \det(M)$$

and back in (Fucito et al. 1981; Weingarten and Petcher 1981)

$$\det(M) = \int \mathscr{D}\bar{\phi}\,\mathscr{D}\phi\ \exp(-\bar{\phi}M^{-1}\phi)$$

- Inversions of M needed
- Physical M has positive eigenvalues
- Discrete M might have negative eigenvalues
- No probability density

Simple Solution

Need a degenerate doublet!

$$\int \mathscr{D}\bar{\psi}_{\mathrm{u}}\,\mathscr{D}\psi_{\mathrm{u}}\,\mathscr{D}\bar{\psi}_{\mathrm{d}}\,\mathscr{D}\psi_{\mathrm{d}}\ \exp(-\bar{\psi}_{\mathrm{u}}\boldsymbol{M}\psi_{\mathrm{u}}-\bar{\psi}_{\mathrm{d}}\boldsymbol{M}\psi_{\mathrm{d}}) = \det(\boldsymbol{M})^2$$

$$\det(\boldsymbol{M})^2 = \det(\boldsymbol{\gamma}_5\boldsymbol{M}\boldsymbol{\gamma}_5\,\boldsymbol{M}) = \det(\boldsymbol{M}^\dagger\boldsymbol{M}) = \det(\boldsymbol{Q}^2)$$

$$\det(\boldsymbol{M}^\dagger\boldsymbol{M}) = \int \mathscr{D}\bar{\phi}\,\mathscr{D}\phi\ \exp(-\bar{\phi}[\boldsymbol{M}^\dagger\boldsymbol{M}]^{-1}\phi)$$

One pseudo fermion field, two quark flavors

Hybrid Monte Carlo

Rational Approximation

What about single flavors?

$$\det(M) = \int \mathscr{D}\bar{\phi}\, \mathscr{D}\phi\, \exp(-\bar{\phi}\sqrt{[Q^2]^{-1}}\phi)$$

Rational approximation (Kennedy, Horvath, and Sint 1999)

$$\sqrt{[Q^2]^{-1}} \approx \sum_i \frac{\alpha_i}{Q^2 + \beta_i}$$

Multi-Shift solvers (Jegerlehner 1996)

- Ritz eigenvalues
- Remez algorithm

Hybrid Monte Carlo

Full HMC algorithm

Full HMC algorithm (Duane et al. 1987)

1. Create pseudo fermion fields, standard normal, one operator application
2. Generate random momenta (gauge algebra)
3. Molecular dynamics
4. Acceptance step (Metropolis)

- Computational complexity $O(V)$
- Exact in stoachastic mean
- Ergodic

Hybrid Monte Carlo

Inversions Everywhere

Computations of $M^{-1}\phi$ per configuration:

- **HMC** 10 to 100 inversions per monomial
- **Point Source** 12 inversions per propagator
- **Perambulator** Hundreds of inversions per perambulator

Hybrid Monte Carlo

Inversions Everywhere

Computations of $M^{-1}\phi$ per configuration:

HMC 10 to 100 inversions per monomial

Point Source 12 inversions per propagator

Perambulator Hundreds of inversions per perambulator

Most expensive part of Lattice QCD

Hybrid Monte Carlo

Inversions Everywhere

Computations of $M^{-1}\phi$ per configuration:

HMC 10 to 100 inversions per monomial

Point Source 12 inversions per propagator

Perambulator Hundreds of inversions per perambulator

Most expensive part of Lattice QCD

M^{-1} cannot be stored in memory (PB to EB)
Solve $Mx = \phi$ with Krylov subspace solvers:

- Conjugate Gradient (CG) (Hestenes and Stiefel 1952)
- Biconjugate Gradient Stabilized (BiCGStab) (Van der Vorst 1992)

Hybrid Monte Carlo

Inversions Everywhere

Computations of $M^{-1}\phi$ per configuration:

HMC 10 to 100 inversions per monomial

Point Source 12 inversions per propagator

Perambulator Hundreds of inversions per perambulator

Most expensive part of Lattice QCD

M^{-1} cannot be stored in memory (PB to EB)
Solve $Mx = \phi$ with Krylov subspace solvers:

- Conjugate Gradient (CG) (Hestenes and Stiefel 1952)
- Biconjugate Gradient Stabilized (BiCGStab) (Van der Vorst 1992)

Multigrid solvers do not exhibit critical slowing down

Wilson Dirac operator:

$$M = -\frac{1}{2}\not{D}_{\mathrm{W}} + \underbrace{4 + m}_{\alpha},$$

Checkerboarding (Degrand and Rossi 1990):

$$M = \begin{pmatrix} M_{\mathrm{ee}} & M_{\mathrm{eo}} \\ M_{\mathrm{oe}} & M_{\mathrm{oo}} \end{pmatrix}_{\mathrm{EO}} .$$

Even-even and odd-odd part

$$M_{\mathrm{ee/oo}} = \alpha$$

Hybrid Monte Carlo

Even-Odd Preconditioning

LU decomposition:

$$M = \begin{pmatrix} M_{ee} & \mathbf{0} \\ M_{oe} & \mathbf{1} \end{pmatrix}_{EO} \begin{pmatrix} \mathbf{1} & M_{ee}^{-1} M_{eo} \\ \mathbf{0} & M_{oo} - M_{oe} M_{ee}^{-1} M_{eo} \end{pmatrix}_{EO} .$$

Define

$$\tilde{M}_{oo} = M_{oo} - M_{oe} M_{ee}^{-1} M_{eo} .$$

Easily inverted:

$$M^{-1} = \begin{pmatrix} \mathbf{1} & -M_{ee}^{-1} M_{eo} \tilde{M}_{oo}^{-1} \\ \mathbf{1} & \tilde{M}_{oo}^{-1} \end{pmatrix}_{EO} \begin{pmatrix} M_{ee}^{-1} & \mathbf{0} \\ -M_{oe} M_{ee}^{-1} & \mathbf{1} \end{pmatrix}_{EO} .$$

$$M_{ee}^{-1} = \frac{1}{\alpha}$$

Hybrid Monte Carlo

Mass Preconditioning

Rewrite determinant (Hasenbusch 2006):

$$\det(Q^2) = \frac{\det(Q^2)}{\det(Q^2 + \rho_1^2)} \cdot \det(Q^2 + \rho_1^2)\,.$$

With pseudo fermions,

$$S_\mathrm{f} = \phi_0^\dagger (M_0^\dagger M_0)^{-1} \phi_0$$

becomes with $M_1 = M_0 + \mathrm{i}\rho$

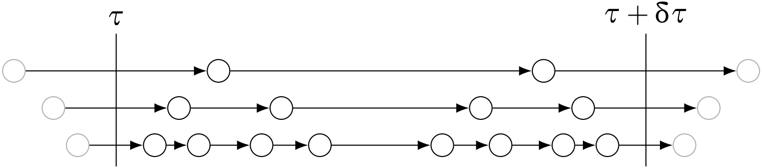
$$S_\mathrm{f} = \phi_0^\dagger M_1^\dagger (M_0^\dagger M_0)^{-1} M_1 \phi_0 + \phi_1^\dagger (M_1^\dagger M_1)^{-1} \phi_1\,.$$

More inversions, each much cheaper

Hybrid Monte Carlo

Multiple Time Scales

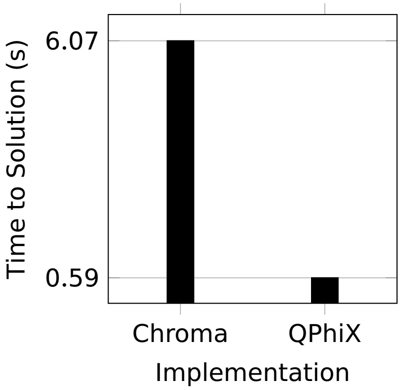
- Forces of summands differ
- Want $\Delta\pi = F\,\delta t$ similar in summands
- Use different δt for summands



Second order symplectic integrator (Omelyan, Mryglod, and Folk 2002)

Performance on JURECA

Chroma/QDP++ against QPhiX



- Markov Chain not parallelizable
- Simulations take months to years
- Fast solver is crucial

Intel Xeon Phi Knights Landing (KNL)

- CPU (host) with 64, 68 or 72 cores
- Two VPUs per core
- Four threads per core
- Two cores in a tile (share L2 cache)
- 16 GB MCDRAM

- New architectures like KNL are hard to program *effectively* (like GPUs)
- Also benefits AVX, AVX2

Let’s see how it works

QPhiX

Code Generator

Operation: $r = ax + y$

Need tailor-made code for each ISA:

No ISA	`ret = (a * x) + y;`
AVX	`ret = _mm256_add_pd(_mm256_mul_pd(a, x), y);`
AVX2	`ret = _mm256_fmadd_pd(a, x, y);`
AVX512	`ret = _mm512_fmadd_pd(a, x, y);`

Operation: $r = ax + y$

Need tailor-made code for each ISA:

No ISA	`ret = (a * x) + y;`
AVX	`ret = _mm256_add_pd(_mm256_mul_pd(a, x), y);`
AVX2	`ret = _mm256_fmadd_pd(a, x, y);`
AVX512	`ret = _mm512_fmadd_pd(a, x, y);`

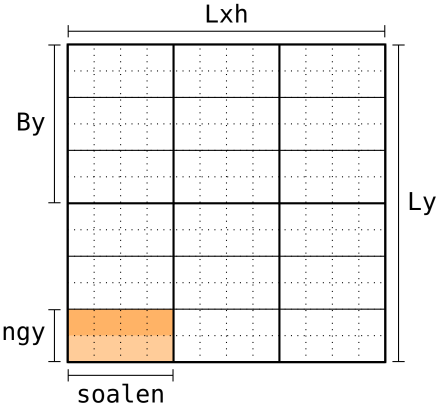
Code Generator

```
fmaddFVec(ivector, Ret, A, X, Y, mask);
```

Tons of kernel code (20 MSLOC on KNL), ridiculous compilation time

QPhiX

Data Layout

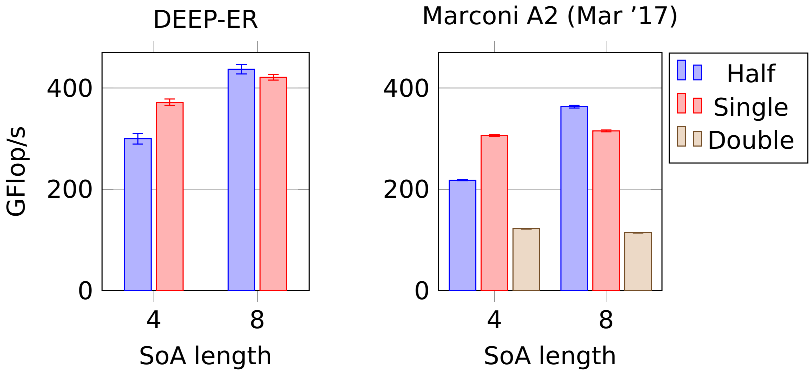


- SIMD oriented
- Vector folding
- Cache blocking

```
typedef FT FourSpinorBlock[3][4][2][soalen];
typedef FT TwoSpinorBlock[3][2][2][veclen];
typedef FT SU3MatrixBlock[8][(compress12 ?  2 :  3)][3][2][veclen];
```

Performance on KNL

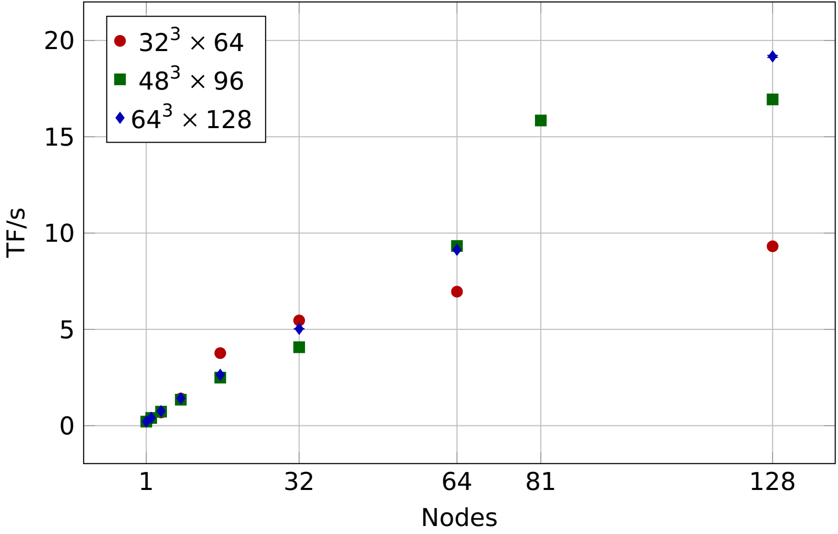
Single Node



- One KNL, one MPI rank, 64×4 threads, $L = 48$, $T = 96$
- KNL has $450\,\mathrm{GB\,s^{-1}}$ of MCDRAM bandwith
- Arithmetic intensity is 1.17 in float, roofline $526\,\mathrm{GFlop\,s^{-1}}$
- Plots from (Labus and Ueding 2017), action is W-Clover

Performance on JURECA

QPhiX Strong Scaling



Extending QPhiX

QPhiX already has *degenerate doublet* even-odd preconditioned

- Wilson and Clover (Joó et al. 2013)
- Twisted Mass (Schröck, Simula, and Strelchenko 2016)
- Twisted Mass Clover (Labus 2016)

Extending QPhiX

QPhiX already has *degenerate doublet* even-odd preconditioned

- Wilson and Clover (Joó et al. 2013)
- Twisted Mass (Schröck, Simula, and Strelchenko 2016)
- Twisted Mass Clover (Labus 2016)

We want

TM and TM clover with $N_{\mathrm{f}} = 1 + 1$ (non-degenerate doublet)

Extending QPhiX

Even-Odd Preconditioning

Twisted mass kinetic operator

$$\boldsymbol{M} = -\frac{1}{2}\not{D}_\mathrm{W} + \alpha + \mathrm{i}\mu\boldsymbol{\gamma}_5\boldsymbol{\tau}^3 - \epsilon\tau^1$$

Odd-odd preconditioned operator

$$\tilde{M}_\mathrm{oo} = M_\mathrm{oo} - M_\mathrm{oe} M_\mathrm{ee}^{-1} M_\mathrm{eo}$$

$$\tilde{M}_\mathrm{oo} = \alpha + \mathrm{i}\mu\boldsymbol{\gamma}_5\boldsymbol{\tau}^3 - \epsilon\boldsymbol{\tau}^1 - \frac{1}{4}\not{D}_\mathrm{oe}\frac{\alpha - \mathrm{i}\mu\boldsymbol{\gamma}_5\boldsymbol{\tau}^3 + \epsilon\boldsymbol{\tau}^1}{\alpha^2 + \mu^2 - \epsilon^2}\not{D}_\mathrm{eo}$$

Previously implemented in QPhiX:

$$\tilde{M}_\mathrm{oo}^\mathrm{uu} = \alpha + \mathrm{i}\mu\boldsymbol{\gamma}_5 - \frac{1}{4d}\not{D}_\mathrm{oe}(\alpha - \mathrm{i}\mu\boldsymbol{\gamma}_5)\not{D}_\mathrm{eo}\,.$$

Extending QPhiX

Reusing the existing implementation

Flavor structure of two-flavor operator

$$\tilde{M}_{\mathrm{oo}} = \begin{pmatrix} \alpha + \mathrm{i}\mu\gamma_5 - \frac{1}{4d}\not{D}_{\mathrm{oe}}(\alpha - \mathrm{i}\mu\gamma_5)\not{D}_{\mathrm{eo}} & -\epsilon - \frac{\epsilon}{4d}\not{D}_{\mathrm{oe}}\not{D}_{\mathrm{eo}} \\ -\epsilon - \frac{\epsilon}{4d}\not{D}_{\mathrm{oe}}\not{D}_{\mathrm{eo}} & \alpha - \mathrm{i}\mu\gamma_5 - \frac{1}{4d}\not{D}_{\mathrm{oe}}(\alpha + \mathrm{i}\mu\gamma_5)\not{D}_{\mathrm{eo}} \end{pmatrix}_{\mathrm{f}}$$

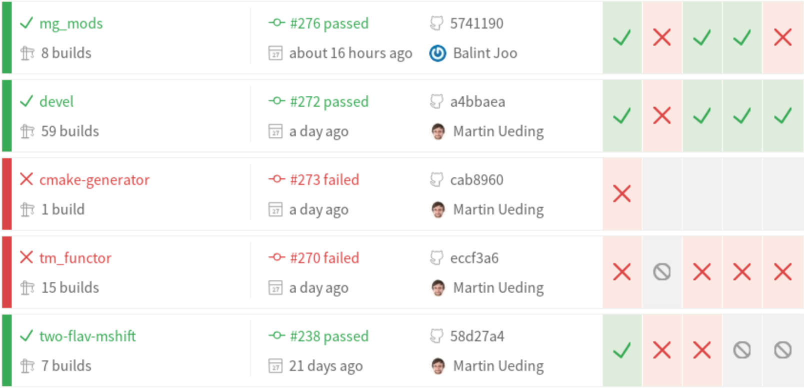
Implementation in QPhiX now

$$\left(\tilde{M}_{\mathrm{oo}}\chi_{\mathrm{o}}\right)^{\mathrm{u}} = \left[\alpha + \mathrm{i}\mu\gamma_5 - \frac{1}{4d}\not{D}_{\mathrm{oe}}(\alpha - \mathrm{i}\mu\gamma_5)\not{D}_{\mathrm{eo}}\right]\chi_{\mathrm{o}}^{\mathrm{u}} - \left[\epsilon + \frac{\epsilon}{4d}\not{D}_{\mathrm{oe}}\not{D}_{\mathrm{eo}}\right]\chi_{\mathrm{o}}^{\mathrm{d}}$$

$$\left(\tilde{M}_{\mathrm{oo}}\chi_{\mathrm{o}}\right)^{\mathrm{d}} = \left[\alpha - \mathrm{i}\mu\gamma_5 - \frac{1}{4d}\not{D}_{\mathrm{oe}}(\alpha + \mathrm{i}\mu\gamma_5)\not{D}_{\mathrm{eo}}\right]\chi_{\mathrm{o}}^{\mathrm{d}} - \left[\epsilon + \frac{\epsilon}{4d}\not{D}_{\mathrm{oe}}\not{D}_{\mathrm{eo}}\right]\chi_{\mathrm{o}}^{\mathrm{u}}$$

Software Engineering

Travis CI



Conclusion

- Lattice QCD needs a fast inverter
- QPhiX is fast for heavy masses
- And it now has new features
- Interface to tmLQCD

Outlook

- Make two-flavor operator faster
- Optimize QPhiX communication strategy
- Interface new features to Chroma

Towards Exascale Computing for Lattice QCD

Masterarbeit in Physik

Martin Ueding
mu@martin-ueding.de

Wednesday, 2017-08-09 15:15
Seminar Room 1, HISKP

Bibliography I

- Degrand, Thomas A. and Pietro Rossi (1990). "Conditioning techniques for dynamical fermions". In: *Computer Physics Communications* 60.2, pp. 211–214. ISSN: 0010-4655. DOI: 10.1016/0010-4655(90)90006-M.
- Duane, S. et al. (1987). "Hybrid Monte Carlo". In: *Phys. Lett.* B195, pp. 216–222. DOI: 10.1016/0370-2693(87)91197-X.
- Fucito, F. et al. (1981). "A Proposal for Monte Carlo Simulations of Fermionic Systems". In: *Nucl. Phys.* B180. [,586(1980)], p. 369. DOI: 10.1016/0550-3213(81)90055-9.
- Hasenbusch, Martin (2006). "Speeding up the HMC algorithm: Some new results". In: *PoS* LAT2005, p. 116. arXiv: hep-lat/0509080 [hep-lat].

Bibliography II

Hestenes, Magnus Rudolph and Eduard Stiefel (1952). "Methods of conjugate gradients for solving linear systems". In: 49.1. DOI: 10.6028/jres.049.044.

Jegerlehner, Beat (1996). "Krylov space solvers for shifted linear systems". In: arXiv: hep-lat/9612014 [hep-lat].

Joó, Bálint et al. (2013). "Lattice QCD on Intelő Xeon PhiTM Coprocessors". In: *Supercomputing: 28th International Supercomputing Conference, ISC 2013, Leipzig, Germany, June 16-20, 2013. Proceedings*. Berlin, Heidelberg: Springer, pp. 40–54. ISBN: 978-3-642-38750-0. DOI: 10.1007/978-3-642-38750-0_4. URL: http://www.opencirrus.intel-research.net/publications/QCD_ISC13_Full_Final.pdf.

Bibliography III

Kennedy, A. D., Ivan Horvath, and Stefan Sint (1999). "A New exact method for dynamical fermion computations with nonlocal actions". In: *Nucl. Phys. Proc. Suppl.* 73, pp. 834–836. DOI: 10.1016/S0920-5632(99)85217-7. arXiv: hep-lat/9809092 [hep-lat].

Labus, Peter (2016). "Lattice Quantum Chromodynamics on Intel Xeon Phi based supercomputers". MA thesis.

Labus, Peter and Martin Ueding (2017). *Lattice QCD Stencils on Intel Xeon Phis*. Talk given at Cineca, Bologna, Italy on 2017-03-23.

Metropolis, Nicholas et al. (1953). "Equation of state calculations by fast computing machines". In: *The journal of chemical physics* 21.6, pp. 1087–1092.

Bibliography IV

- Omelyan, I. P., I. M. Mryglod, and R. Folk (2002). "New optimized algorithms for molecular dynamics simulations". In: *Condensed Matter Physics* 5.3(31), pp. 369–390. URL: http://www.icmp.lviv.ua/journal/zbirnyk.31/001/art01.pdf.
- Schröck, Mario, Silvano Simula, and Alexei Strelchenko (2016). "Accelerating Twisted Mass LQCD with QPhiX". In: *PoS* LATTICE2015, p. 030. arXiv: 1510.08879 [hep-lat].
- Van der Vorst, Henk A (1992). "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems". In: *SIAM Journal on scientific and Statistical Computing* 13.2, pp. 631–644.
- Weingarten, D. H. and D. N. Petcher (1981). "Monte Carlo Integration for Lattice Gauge Theories with Fermions". In: *Phys. Lett.* 99B, pp. 333–338. DOI: 10.1016/0370-2693(81)90112-X.