Implementation of the Neuberger-Dirac operator on GPUs

Bjoern Walk*[*] and Hartmut Wittig
Institut für Kernphysik, Johannes Gutenberg-Universität Mainz,
Johann Joachim Becher-Weg 45, 55099 Mainz, Germany
E-mail: bwalk@kph.uni-mainz.de, wittig@kph.uni-mainz.de

Egor Dranischnikow and Elmar Schömer
Institut für Informatik, Johannes Gutenberg-Universität Mainz,
Staudingerweg 9, 55099 Mainz, Germany
E-mail: {dranisch,schoemer}@informatik.uni-mainz.de

Recent developments have shown that a lot can be gained for QCD simulations from GPU hardware. This can be exploited especially in the case of Ginsparg-Wilson fermions when the computational costs are particularly high. In this work, we use the Neuberger-Dirac operator as our realisation of Ginsparg-Wilson fermions, which greatly facilitate lattice investigations of decays like $K \to \pi\pi$. We report on the ongoing study of our GPU implementation of the Neuberger-Dirac operator including the exact treatment of the low lying eigenmodes of the Wilson-Dirac operator. Our benchmarks show that we achieve speed-up factors of around 23 and 16 in single and double precision, respectively.

The XXVIII International Symposium on Lattice Field Theory, Lattice2010
June 14-19, 2010
Villasimius, Italy

[*]Speaker.

1. Introduction

In the past years the GPU, the core of hardware responsible to display images on computer screens, became more and more flexible and powerful. The GPU soon surpassed the CPU in terms of raw computational power, and various fields in science found a way to exploit this (see also the review in [1]). For the lattice community, ref. [2] set out one ground-breaking paper in which it was demonstrated how to map the Wilson-Dirac operator on the GPU. With the introduction of NVIDIA CUDA, a flexible programming model was given and made it even easier to access the ever-increasing computational power. This was the basis for the reference paper [3].

Using GPUs could be particularly important in order to simulate Ginsparg-Wilson fermions [4] which possess an order-of-magnitude higher computational costs than Wilson fermions.

One particular solution to the Ginsparg-Wilson relation is the Neuberger-Dirac operator which we are studying in this paper. A lattice Dirac operator which satisfies chiral symmetry at nonzero lattice spacing greatly facilitates the investigation of certain processes like non-leptonic kaon decays. These decays raise the long-standing problem for QCD phenomenology why the $\Delta I = 1/2$ amplitudes are so much larger than the $\Delta I = 3/2$ amplitudes. The bulk of the enhancement must be due to strong interactions [5] at low energies and therefore, a reliable explanation must be based on systematic non-perturbative methods, in particular on lattice QCD [6].

So far, a lot of effort has been invested to understand and quantify the effects responsible for the enhancement of the $\Delta I = 1/2$ amplitudes. A series of simulations were performed using quenched calculations in the SU(4)-symmetric case in order to understand the effect of the charm quark [7]. One particular approach is to do simulations outside the GIM-limit with an active charm quark and reasonable large volumes to study finite-volume effect.

In this paper we report on the ongoing development of our tools for numerical simulations in the ε-regime utilizing GPU hardware for acceleration. We show details of the theoretical background and implementation for our Wilson-Dirac kernel and for the Neuberger-Dirac operator. We give performance results for both operators and the calculation of low lying eigenmodes of the Wilson-Dirac operator. At the end we give an outlook about planned work.

2. Implementation details

2.1 The Wilson-Dirac operator

Following Wilsons formulation for lattice QCD, the γ_5-Hermitian, massive Wilson-Dirac operator Q used in our implementation is defined by

$$D_{\mathrm{W}}\phi(x) = (4+m)\phi(x) - \frac{1}{2}\sum_{\mu=\pm 0}^{\pm 3} U_\mu(x)(1-\gamma_\mu)\phi(x+a\hat{\mu}), \quad Q = \gamma_5 D_{\mathrm{W}}, \tag{2.1}$$

where m is the bare mass of the fermion. Gauge links $U_\mu(x)$ and the Dirac matrices γ_μ follow

$$U_{-\mu}(x) = U_\mu^{-1}(x-a\hat{\mu}) \quad \text{and} \quad \gamma_{-\mu} = -\gamma_\mu. \tag{2.2}$$

The most important part in the implementation of the operator is the data layout. Data read and written to global device memory on the GPU has to fulfill several constraints, known as coalescing rules [8], to ensure maximum performance. One of them requires that consecutive threads read data words in sequential order. A data word can have 32-, 64- or 128-Bytes. In [2] a data layout for the fermions fields and the gauge links was introduced which is nowadays considered as the default and our implementation follows that, too.

Since the Wilson-Dirac kernel is memory-bound, we intend to minimize memory access. It is possible to save memory load instructions by realizing that one does not need the whole 18 entries of the gauge link matrices as they fulfill the orthogonalization relation $U^\dagger U = 1$ of the SU(3) gauge group. In total, we need a minimum of 8 parameters to uniquely define an element of the gauge links. Each element of SU(3) can be expressed as a linear combination of Gell-Mann-matrices, and in principle one could choose the coefficients in this linear combination as parameters. However, although the reconstruction comes practically for free, we should not introduce any overhead in the reconstruction. A reconstruction based on Gell-Mann matrices is not beneficial in terms of numerical effort. A much better approach is presented in [3, 9] which also has the advantage, that the inverse operation, i.e. finding the parameters for a given SU(3)-matrix, is simple. Another simple reconstruction scheme is based on the property

$$U = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \qquad \vec{c} = (\vec{a} \times \vec{b})^*, \tag{2.3}$$

that one column of the matrix can be reconstructed by the other two.

The rest of the implementation is fairly straightforward since there is no communication necessary if we assign each lattice site of the resulting fermion field a single thread. We explicitly write down the multiplication in Dirac-space of the eight directions $\mu = \pm 0 \ldots \pm 3$, unroll the matrix-vector multiplication in color space after reconstruction and accumulate the resulting fermion in local registers. After the multiplication with γ_5, we write the result to global device memory.

2.2 The Neuberger-Dirac operator

Following the conventions and notations from [10], the Neuberger-Dirac operator D_N [11] can be defined in terms of the Wilson-Dirac operator D_W

$$D_\mathrm{N} = \frac{1 + \gamma_5\,\mathrm{sign}(Q)}{\bar{a}}, \qquad Q = \gamma_5(aD_\mathrm{W} - 1 - s). \tag{2.4}$$

In this definition, $|s| < 1$ is a tunable parameter while the sign-function has to be defined by its series expansion. For numerical stability we choose Chebychev polynomials, so the sign-function of the operator Q is given by

$$\mathrm{sign}(Q) \simeq XP_n(X^2), \qquad X \equiv \frac{Q}{||Q||} \quad \text{and} \quad P_n(x) = \sum_{k=0}^{n} c_k T_k(x). \tag{2.5}$$

The definition of the Chebychev polynomials $T_k(y)$ can be found in [12]. Because Chebychev polynomials fulfill a recursion relation they permit an evaluation via the Clenshaw recurrence formula.

In order to find the coefficients c_k of the expansion, we aim to minimize the error

$$\delta = \max_{\varepsilon \le y \le 1} |h(y)|, \qquad h(y) \equiv 1 - \sqrt{y}P(y) \tag{2.6}$$

of the polynomial expansion for specified ε. In the range $\sqrt{\varepsilon} \leq |x| \leq 1$ the function $xP(x^2)$ then approximates $\mathrm{sign}(x)$ uniformly with a maximal deviation δ. Polynomials constructed in such a way are often referred to as minmax polynomials. If ε is chosen such that $Q^2 \geq \varepsilon \|Q\|^2$, the error in (2.5) is an operator with norm less than or equal to δ. The approximation error is always bounded by $\delta \|\eta\|$, uniformly in the field η to which the operator is applied.

This method is straightforward, but in the case of the ε-regime actually not recommended. In this case, the operator Q^2 may have some exceptionally low-lying eigenvalues, and it is far more efficient first to separate the few lowest modes and to treat them exactly.

2.3 Low-mode projection

The spectrum of Q in the vicinity of the origin can be reliably determined by minimizing the Ritz functional of Q^2 [13]. Thereby, we also find an approximation of the associated eigenvectors. In order to control the error of the total approximation, we need to examine by how much these vectors deviate from the true eigenvector [10].

Assuming that a specified number l of approximate eigenvectors has been computed, we denote the linear space spanned by these vectors by V and by $\mathbb{P}$ the corresponding orthonormal projector. We shall take it for granted that the eigenvalues ν_k of Q are separated from zero and from the rest of the spectrum by a distance greater than ρ. This number measures the deviation of V from being an exact eigenspace of Q. The presence of a spectral gap around zero implies that the subset of positive and negative eigenvalues can be identified without any numerical ambiguity. With the introduction of the eigenvectors $u_k \in V$, $\mathbb{P}Qu_k = \nu_k u_k$, the associated orthonormal projectors are given by

$$\mathbb{P}_+ = \sum_{\nu_k>0} u_k \otimes (u_k)^\dagger \quad \text{and} \quad \mathbb{P}_- = \sum_{\nu_k<0} u_k \otimes (u_k)^\dagger \tag{2.7}$$

respectively. In the same way we may define the projectors $(\mathbb{P}_\pm)_{\text{exact}}$ to the subspaces spanned by the corresponding exact eigenvectors of Q with positive and negative eigenvalues λ_k. We can give a quantitative estimate on the deviation of the computed projectors $\mathbb{P}_\pm$ from the exact projectors $(\mathbb{P}_\pm)_{\text{exact}}$. The total error depends on the size of the residues

$$\rho_k = \|(Q-\nu_k)u_k\| \tag{2.8}$$

and also on the distance between the eigenvalues, as a small value of ρ_k does not exclude sizeable mixing of u_k with several eigenvectors of Q if these have eigenvalues that are within a distance ρ_k of ν_k. When estimating the deviation of the projectors rather than that of the individual eigenvectors, the interesting quantity is the distance d_k of ν_k from the exact spectrum of Q in the subspace that is orthogonal to the range of $(\mathbb{P}_+)_{\text{exact}}$ if $\nu_k > 0$ if $(\mathbb{P}_-)_{\text{exact}}$ if $\nu_k < 0$. The quality of the approximation is then controlled by the parameters

$$\kappa_\pm^2 = \sum_{\pm\nu_k>0} \rho_k^2/d_k^2, \quad \kappa_\pm > 0 \tag{2.9}$$

and an upper bound for the deviation of the projectors can by given by

$$\|\mathbb{P}_\pm - (\mathbb{P}_\pm)_{\text{exact}}\| \leq \frac{\kappa_\pm(1+2\kappa_\pm)}{1-2\kappa_\pm(1+2\kappa_\pm)}. \tag{2.10}$$

	GTX285	GTX480	C2050
No. of Cores	240	480	448
Memory amount [MB]	2048	1536	3072
Shader Clock [MHz]	1476	1401	1150
Memory Clock [MHz]	1242	1848	1500
Memory Bandwidth [GB/s]	159.0	177.4	144.0

Table 1: Key specification of the three NVIDIA GPUs available for benchmarking. For the C2050, roughly 10% of total memory amount has to be subtracted if the error correction ECC is enabled by the driver.

In our implementation, the whole computation of the projectors is performed on the GPU. The number of low modes included in the projector will be determined dynamically in such a way that the spectral distance from the other modes is not accidentally small. The parameters $\kappa_\pm$ can be estimated without difficulty and the Ritz functional is stopped when the desired level of precision is reached. Afterwards, we replace Equation 2.5 by

$$\mathrm{sign}(Q) \simeq \mathbb{P}_+ - \mathbb{P}_- + (1 - \mathbb{P}_+ - \mathbb{P}_-) X P_n(X^2), \tag{2.11}$$

and it can be shown that the approximation $\tilde{D}_{\mathrm{N}}^m$ to the massive Neuberger-Dirac operator satisfies

$$\|\tilde{D}_{\mathrm{N}}^m - D_{\mathrm{N}}^m\| \leq \frac{2}{a}\big(1 + s - \frac{am}{2}\big)\big(\kappa_+ + \kappa_-\big). \tag{2.12}$$

3. Results

For performance results we have tested the GeForce GTX285 GPU, a current generation card, and the GeForce GTX480, as well as a Tesla C2050 GPU, the latter of which are both based on the recently released Fermi chipset. The code, however, was not optimized for the new architecture and we expect further enhancements in the future. An overview of the key features of each GPU is given in Table 1.

We have already stated that the performance of the Wilson-Dirac kernel is memory bound. In our optimization we aimed for maximizing the total memory bandwidth achieved by our implementation. This also means that we should not expect strong scaling as the number of cores is increased. The major contribution to the performance of the calculation comes from the memory bandwidth.

In Figure 1 the performance of the Wilson-Dirac kernel in single- and double precision is given as a function of the lattice volume. The double precision data curve on the C2050 is missing because we had only limited access to the GPU and it was not available anymore at the time of our testings.

We can see a nearly constant behaviour for sufficient lattice volumes. On the GTX285 we observe a small peak at $T = 64$ which we believe is largely accidental. For double precision we see a decrease of the performance by a factor of around 8. On the GTX285, this comes from the fact that there is only 1 double precision unit while 8 single precision units are available. In principle, the architectural design of the GTX480 should give up to half of the single precision speed in

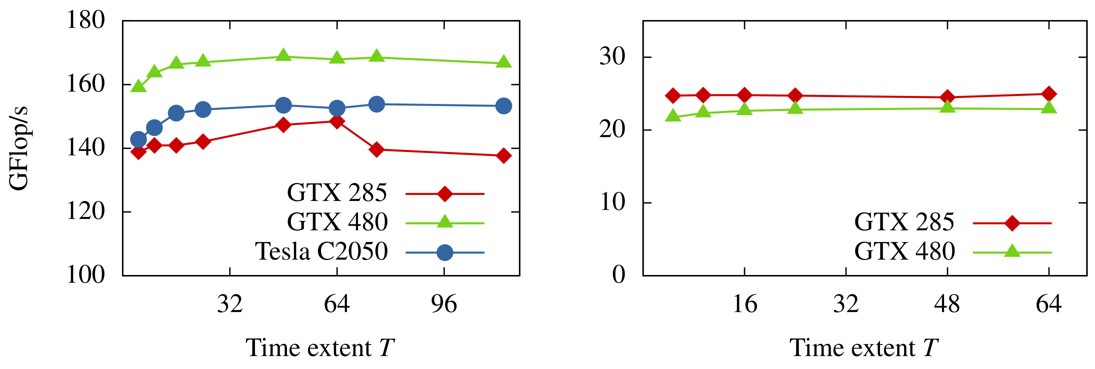

Figure 1: Performance result for the application of the massive Wilson-Dirac operator Q. **LEFT:** Single precision performance. **RIGHT:** Double precision performance. The spatial extent L is fixed at 24^3 throughout.

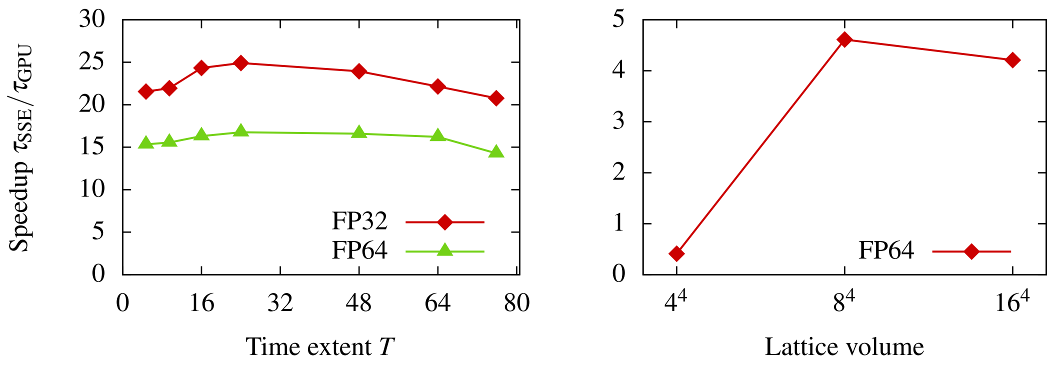

Figure 2: **LEFT:** Performance result of the application of the Neuberger-Dirac operator D as compared to the SSE2-optimized CPU version. The spatial extend L is fixed at 24^3. **RIGHT:** Performance result for the calculation of low-lying eigenmodes of the Wilson-Dirac operator Q. The plots show the speed-up factor $\tau_{\text{SSE}}/\tau_{\text{GPU}}$ in single and double precision, respectively.

double precision. NVIDIA, however, restricts this to a quarter of the single precision speed inside the driver. This drop in performance can be explained by the fact that we did not optimize for the new chipset. The final result for the Wilson-Dirac kernel gives around 140 GFlop/s on the GTX285, 150 GFlop/s on the C2050 and 170 GFlop/s on the GTX480 in single precision. For double precision we achieve 22 GFlop/s and 25 GFlop/s on the GTX285 and GTX480 respectively.

In Figure 2 the performance of the Neuberger-Dirac operator in single and double precision is illustrated as a comparison to the SSE2-optimized CPU version. The execution time of the operator is normalized on the number of Clenshaw iterations and the lattice volume. The CPU version was run on an Intel E6400 Core2 with a single core clock rate of 2.13 GHz. The GPU version was run on the GTX285. We can observe a nearly constant behaviour in the execution time in dependence to the lattice volume. For single precision, the SSE2 version has an average of $\tau_{\text{SSE}} = 0.877\,\mu$s and the GPU version of $\tau_{\text{GPU}} = 0.038\,\mu$s, giving a speedup factor of around 23. For double precision, the numbers read $\tau_{\text{SSE}} = 1.459\,\mu$s and $\tau_{\text{GPU}} = 0.090\,\mu$s, hence, a speedup factor of around 16.

4. Conclusions and outlook

In this proceedings article, we gave a first introduction of our study of a GPU-based simulation program for lattice QCD based on the Neuberger-Dirac operator. We introduced the basic theoretical background and gave first performance results of our implementation of the Wilson-Dirac operator and the Neuberger-Dirac operator. We have shown that the performance for the Wilson-Dirac operator is of the same order of magnitude compared to other implementations previously published. For the Neuberger-Dirac operator, we have shown that we can reach a speedup factor to the Wilson-Dirac operator of around 23 on single precision and 16 on double precision. We are going to develop code for the calculation of the index of our Neuberger-Dirac operator via zero-mode counting [7]. In the long run, we aim to integrate our code into a larger set of simulation programs to calculate observables in the process $K \to \pi\pi$ in order to increase the lattice volume on those calculations.

5. Acknowledgements

Part of the performance results were obtained on GPU of the KOMET collaboration of the University of Mainz. We are indebted to the institute for these opportunity. B. W. is funded by the DFG via GK 1581.

References

[1] M. A. Clark, PoS **LATTICE2009** (2009) 003 [arXiv:0912.2268 [hep-lat]].

[2] G. I. Egri, Z. Fodor, C. Hoelbling, S. D. Katz, D. Nogradi and K. K. Szabo, Comput. Phys. Commun. **177** (2007) 631.

[3] K. Barros, R. Babich, R. Brower, M. A. Clark and C. Rebbi, PoS **LATTICE2008** (2008) 045.

[4] P. H. Ginsparg and K. G. Wilson, Phys. Rev. D **25** (1982) 2649; D. B. Kaplan, Nucl. Phys. Proc. Suppl. **30** (1993) 597; R. Narayanan and H. Neuberger, Nucl. Phys. B **443**, 305 (1995).

[5] M. K. Gaillard and B. W. Lee, Phys. Rev. Lett. **33**, 108 (1974); G. Altarelli and L. Maiani, Phys. Lett. B **52**, 351 (1974).

[6] N. Cabibbo, G. Martinelli and R. Petronzio, Nucl. Phys. B **244**, 381 (1984); R. C. Brower, G. Maturana, M. Belen Gavela and R. Gupta, Phys. Rev. Lett. **53** (1984) 1318.

[7] P. Hernandez, M. Laine, C. Pena, E. Torro, J. Wennekers and H. Wittig, JHEP **0805**, 043 (2008); L. Giusti, P. Hernandez, M. Laine, C. Pena, J. Wennekers and H. Wittig, Phys. Rev. Lett. **98**, 082003 (2007); L. Giusti, P. Hernandez, M. Laine, P. Weisz and H. Wittig, JHEP **0411** (2004) 016; P. Hernandez and M. Laine, JHEP **0409** (2004) 018.

[8] NVIDIA, NVIDIA CUDA C Programming Guide.

[9] B. Bunk and R. Sommer, Comput. Phys. Commun. **40** (1986) 229.

[10] L. Giusti, C. Hoelbling, M. Lüscher and H. Wittig, Commput. Phys. Commun. **153** (1998) 31.

[11] H. Neuberger, Phys. Lett. B **417** (1998) 141; H. Neuberger, Phys. Lett. B **427** (1998) 353.

[12] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Numerical Recipes, Third Edition.

[13] B. Bunk, K. Jansen, M. Lüscher, H. Simma, ALPHA collaboration internal report (1994); T. Kalkreuther and H. Simma, Comput. Phys. Comm. **93** (1996) 33.