Chapter 3

Accelerating the Hybrid Monte Carlo algorithm

In chapter 2 we have presented a scaling test of mtmQCD in the quenched approximation with values of the pseudo scalar mass lower than 300 MeV. Of course, it would be desirable to repeat such a study with two dynamical flavors of quarks and light values of their masses, at least for a first study. But, unfortunately, simulations with light dynamical quark flavors are by orders of magnitude more expensive in terms of computer time than the corresponding simulations in the quenched approximation. The reason for this is basically that the generation of one gauge field configuration is much more ($\mathcal{O}(100)$) expensive and the gauge configurations depend in full QCD on the values of the quark masses.

The Hybrid Monte Carlo (HMC) algorithm [75] is one widely used algorithm to perform dynamical simulations. It is an exact algorithm, which combines molecular dynamics evolution of the gauge fields with a Metropolis accept/reject step to correct for discretization errors in the numerical integration of the corresponding equations of motion. However, in its original form the HMC algorithm is even on computers available today not able to tackle simulations with light quarks on fine lattices. Due to increasing iteration numbers in the solvers and autocorrelation times the costs C are expected to increase as [105]

$$C = K \left(\frac{m_{\mathrm{PS}}}{m_{\mathrm{V}}}\right)^{-6} L^5\ a^{-7}\,,$$

where m_{PS} and m_{V} are the pseudo scalar and the vector mass, L is the spatial lattice extent and a the lattice spacing. The proportionality factor K was found to be too large to allow for simulations with realistic mass values on fine lattices [105]. Hence, during the last years a lot of effort has been invested to decrease K and to improve the cost scaling behavior of the HMC. The list of improvements that were found reaches, for instance, from even/odd preconditioning [106] over multiple time scale integration [107] to mass preconditioning (Hasenbusch acceleration) [108, 109], to mention only those that are immediately relevant for the present work. It is worth noting that many of the known improvement tricks can be combined. In addition, alternative multi-boson methods [74] have been suggested, which, however, appear not to be superior to the HMC algorithm, although they have conceptual advantages compared to the HMC algorithm.

Recently in Ref. [110] a HMC variant as a combination of multiple time scale integration with domain decomposition as preconditioner with excellent scaling properties with the quark mass was presented. In addition, the value of K seems to be significantly lower than for other HMC variants. Thus this algorithm can be expected to be most promising when one wants to simulate small quark masses on fine lattices.

In this chapter we are going to present yet another variant of the HMC algorithm similar to the one of Refs. [111, 112] comprising multiple time scale integration with mass preconditioning. We test this algorithm for standard Wilson fermions at $\beta = 5.6$ and at pseudo scalar masses ranging from $m_{\mathrm{PS}} = 380$ MeV to $m_{\mathrm{PS}} = 670$ MeV, which are the simulation points of Ref. [110]. We show that in this situation the algorithm has similar scaling properties and performance as the method presented in Ref. [110]. From the performance data obtained with our HMC variant we find that K is reduced and the scaling of the cost with the quark mass is improved when compared to performance data available in the literature [105, 113, 114].

3.1 HMC algorithm

The variant of the HMC algorithm we will present here is applicable to a wide class of lattice Dirac operators, including twisted mass fermions, various improved versions, staggered fermions, and even the overlap operator. Nevertheless, in order to discuss a concrete example, we restrict ourselves in this chapter to the Wilson-Dirac operator (1-26) for $n_f = 2$ flavors of mass degenerate quarks with Wilson parameter r set to one. We do not expect that the algorithm properties depend significantly on the particular choice of the Dirac operator.

Since D_{W} (cf. Eq. (1-26)) fulfills the property $\gamma_5 D_{\mathrm{W}} \gamma_5 = D_{\mathrm{W}}^\dagger$ it is convenient to define the hermitian Wilson-Dirac operator

$$Q = \gamma_5 D_{\mathrm{W}} \,. \tag{3-1}$$

After integrating out the fermion fields (cf. Eq. (1-15)) we have to deal with $\det(D_{\mathrm{W}})^2 = \det(Q^2)$ in the simulations. This is usually done by re-expressing the determinant with a Gaussian integral over bosonic fields $\phi, \phi^\dagger$:

$$\det(D_{\mathrm{W}})^2 = \det(Q)^2 \propto \int \mathcal{D}\phi^\dagger \, \mathcal{D}\phi \;\exp\Big(-S_{\mathrm{PF}}[U, \phi^\dagger, \phi]\Big) \,, \tag{3-2}$$

where $S_{\rm PF}[U,\phi^\dagger,\phi] = |Q^{-1}\phi|^2$ is the so called pseudo fermion action. Formally the ϕ-fields are identical to the matter fields ψ since they have the same degrees of freedom, but follow the statistic of bosonic fields and are therefore called *pseudo fermion* fields. The partition function (1-21) is then given by

$$\mathcal{Z} = \int \mathcal{D}U\,\mathcal{D}\phi\,\mathcal{D}\phi^\dagger\; e^{-S_{\rm G}[U]\,-S_{\rm PF}[U,\phi,\phi^\dagger]}\,. \tag{3-3}$$

In order to perform Monte Carlo simulations for this partition function the integral over the pseudo fermion fields could be included in the important sampling process. This is, however, not needed, since for a given gauge field

$$e^{-S_{\rm PF}[U,\phi,\phi^\dagger]} = e^{-R^\dagger R} \tag{3-4}$$

is a Gaussian distribution, which can be generated exactly. Therefore only the gauge fields have to be generated in a Markov chain, which is implemented by the Hybrid Monte Carlo algorithm. To set up the HMC algorithm [75] we introduce traceless Hermitian momenta $P_{x,\mu}$ as conjugate fields to the gauge fields $U_{x,\mu}$ and a Hamiltonian

$$H(P,U,\phi,\phi^\dagger) = \frac{1}{2}\sum_{x,\mu} P^2_{x,\mu} + S_{\rm G}[U] + S_{\rm PF}[U,\phi,\phi^\dagger]\,. \tag{3-5}$$

The algorithm is then composed out of the following steps:

1. Global heat-bath for momenta and pseudo fermion fields.

   The initial momenta are randomly produced according to a Gaussian distribution $\exp(-P^2/2)$. Moreover, random fields R are produced from a distribution like $\exp(-R^\dagger R)$ and the initial pseudo fermion fields are computed with $\phi = QR$.

2. Molecular dynamics evolution.

   Production of a proposal gauge configuration U' and proposal momenta P' by *molecular dynamics* evolution (integrating Hamilton's equations of motion) of the gauge fields U and the momenta P at fixed pseudo fermion fields ϕ. If the integration of the corresponding equations of motion can be performed exactly, the Hamiltonian is conserved under this evolution.

3. Metropolis accept/reject step.

   The proposals U' and P' are accepted with the probability $\min\{1, \exp(-\Delta H)\}$, where $\Delta H = H(P',U',\phi,\phi^\dagger) - H(P,U,\phi,\phi^\dagger)$.

   This step is needed because the integration of the equations of motion can in practice be done only numerically and hence an acceptance step is needed to correct for discretization errors.

If the integration scheme is reversible and area preserving it is possible to prove that the HMC algorithm satisfies the *detailed balance* condition [75] and hence the HMC algorithm is exact.

Since the Hamiltonian is conserved up to discretization errors, the integration can be set up such that the acceptance rate is high while the gauge configurations are globally updated.

3.1.1 Molecular dynamics evolution

In the molecular dynamics part of the HMC algorithm the gauge fields U and the momenta P need to be evolved in a fictitious computer time t. With respect to t, Hamilton's equations of motion read

$$\frac{dU}{dt} = \frac{dH}{dP} = P\,, \qquad \frac{dP}{dt} = -\frac{dH}{dU} = -\frac{dS}{dU}\,, \tag{3-6}$$

where we set $S = S_{\mathrm{G}} + S_{\mathrm{PF}}$ and d/dU, d/dP formally denote the derivative with respect to group elements. Since analytical integration of the former equations of motion is normally not possible, these equations must in general be integrated with a discretized integration scheme that is area preserving and reversible, such as the leap frog algorithm. The discrete update with integration step size $\Delta\tau$ of the gauge field and the momenta can be defined as

$$\begin{aligned} T_{\mathrm{U}}(\Delta\tau): \quad & U \quad \rightarrow \quad U' = \exp(i\Delta\tau P)\,U\,, \\ T_{\mathrm{S}}(\Delta\tau): \quad & P \quad \rightarrow \quad P' = P - i\Delta\tau\delta S\,, \end{aligned} \tag{3-7}$$

where δS is an element of the Lie algebra of SU(3) and denotes the variation of S with respect to the gauge fields. The computation of δS is the most expensive part in the HMC algorithm since the variation of S_{PF} reads

$$\delta S_{\mathrm{PF}} = -\phi^{\dagger}\frac{1}{Q^2}\delta(Q^2)\frac{1}{Q^2}\phi \tag{3-8}$$

and thus involves the inversion of the Wilson-Dirac operator. With (3-7) one basic time evolution step of the so called leap frog algorithm reads

$$T = T_{\mathrm{S}}(\Delta\tau/2)\; T_{\mathrm{U}}(\Delta\tau)\; T_{\mathrm{S}}(\Delta\tau/2)\,, \tag{3-9}$$

and a whole trajectory of length τ is achieved by $N_{\mathrm{MD}} = \tau/\Delta\tau$ successive applications of the transformation T.

3.1.2 Integration with multiple time scales

Consider a Hamiltonian of the form

$$H = \frac{1}{2}\sum_{x,\mu} P_{x,\mu}^2 + \sum_{i=0}^{k} S_i[U]\,, \tag{3-10}$$

with $k \geq 1$. For instance, with $k = 1$ S_0 might be identified with the gauge action and S_1 with the pseudo fermion action of Eq. (3-5).

Clearly, in order to keep the discretization errors in a leap frog like algorithm small, the time steps have to be small if the driving forces are large. Then, if the forces originating from the single parts in the Hamiltonian (3-10) differ significantly in their absolute values, it might be valuable to integrate the different S_i on time scales inverse proportionally deduced from the corresponding forces. This will maximally improve the algorithm performance if the most expensive part can be integrated with the largest molecular dynamics steps size.

The leap frog integration scheme can be generalized to multiple time scales as has been proposed in Ref. [107] without loss of reversibility and the area preserving property. The scheme with only one time scale can be recursively extended by starting with the definition

$$T_0 = T_{\mathrm{S}_0}(\Delta\tau_0/2)\ T_{\mathrm{U}}(\Delta\tau_0)\ T_{\mathrm{S}_0}(\Delta\tau_0/2)\,, \tag{3-11}$$

with T_{U} defined as in Eq. (3-7) and where $T_{\mathrm{S}_i}(\Delta\tau)$ is given by

$$T_{\mathrm{S}_i}(\Delta\tau) \quad : \quad P \quad \rightarrow \quad P - i\Delta\tau\delta S_i[U]\,. \tag{3-12}$$

As $\Delta\tau_0$ will be the smallest time scale, we can recursively define the basic update steps T_i, with time scales $\Delta\tau_i$ as

$$T_i = T_{\mathrm{S}_i}(\Delta\tau_i/2)\ [T_{i-1}]^{N_{i-1}}\ T_{\mathrm{S}_i}(\Delta\tau_i/2)\,, \tag{3-13}$$

with integers N_i and $0 < i \leq k$. One full trajectory τ is then composed by $[T_k]^{N_k}$. The different time scales $\Delta\tau_i$ in Eq. (3-13) must be chosen such that the total number of steps on the i-th time scale N_{MD_i} times $\Delta\tau_i$ is equal to the trajectory length τ for all $0 \leq i \leq k$: $N_{\mathrm{MD}_i}\Delta\tau_i = \tau$. This is obviously achieved by setting

$$\Delta\tau_i = \frac{\tau}{N_k \cdot N_{k-1} \cdot \ldots \cdot N_i} = \frac{\tau}{N_{\mathrm{MD}_i}}\,, \qquad 0 \leq i \leq k\,, \tag{3-14}$$

where $N_{\mathrm{MD}_i} = N_k \cdot N_{k-1} \cdot \ldots \cdot N_i$.

In Ref. [107] also a partially improved integration scheme with multiple time scales was introduced, which reduces the size of the discretization errors. Again, we assume a Hamiltonian of the form (3-10) with now $k = 1$. By defining similar to T_0

$$T_{\mathrm{SW}_0} \;=\; T_{\mathrm{S}_0}(\Delta\tau_0/6)\ T_{\mathrm{U}}(\Delta\tau_0/2)\ T_{\mathrm{S}_0}(2\Delta\tau_0/3)\ T_{\mathrm{U}}(\Delta\tau_0/2)\ T_{\mathrm{S}_0}(\Delta\tau_0/6)\,, \tag{3-15}$$

the basic update step of the improved scheme – usually referred to as the Sexton-Weingarten (SW) integration scheme – reads

$$\begin{aligned} T_{\mathrm{SW}_1} =&T_{\mathrm{S}_1}(\Delta\tau_1/6) \\ &\qquad [T_{\mathrm{SW}_0}]^{N_0}\ T_{\mathrm{S}_1}(2\Delta\tau_1/3) \\ &\qquad [T_{\mathrm{SW}_0}]^{N_0}\ T_{\mathrm{S}_1}(\Delta\tau_1/6)\,, \end{aligned} \tag{3-16}$$

where $\Delta\tau_0 = \Delta\tau_1/(2N_0)$. This integration scheme not only reduces the size of the discretization errors, but also sets for S_0 a different time scale than for S_1. Hence, it is one special example for an integration scheme with multiple time scales and can easily be extended to more than two time scales by recursively defining $(0 < i \leq k)$:

$$\begin{aligned} T_{\mathrm{SW}_i} =& T_{S_i}(\Delta\tau_i/6) \\ & [T_{\mathrm{SW}_{i-1}}]^{N_{i-1}}\ T_{S_i}(2\Delta\tau_i/3) \\ & [T_{\mathrm{SW}_{i-1}}]^{N_{i-1}}\ T_{S_i}(\Delta\tau_i/6)\,. \end{aligned} \tag{3-17}$$

The different time scales for the SW integration scheme are defined by

$$\Delta\tau_i = \frac{\tau}{(2N_k)\cdot(2N_{k-1})\cdot\ldots\cdot(2N_i)} = \frac{\tau}{N_{\mathrm{MD}_i}}\,, \qquad i \leq k\,. \tag{3-18}$$

Note that the SW partially improved integration scheme was originally invented to make use of the fact that the computation of the variation of the gauge action is cheap as compared to the variation of the pseudo fermion action and in addition the time scales are chosen in order to cancel certain terms in the discretization error exactly [107]. It can be generalized to integrators of the form

$$T_{2\mathrm{MN}_0} \;=\; T_{\mathrm{S}_0}(\lambda\Delta\tau_0)\ T_{\mathrm{U}}(\Delta\tau_0/2)\ T_{\mathrm{S}_0}((1-2\lambda)\Delta\tau_0)\ T_{\mathrm{U}}(\Delta\tau_0/2)\ T_{\mathrm{S}_0}(\lambda\Delta\tau_0)\,, \tag{3-19}$$

with a real parameter λ that needs to be tuned [115], which are called second order Minimal Norm (2MN) integration schemes. These schemes can be generalized to multiple time scales in exactly the same way as the SW integration scheme. However, we restrict ourselves in this work to the LF and the SW integration scheme only, also because from the results of Ref. [115] we do not expect a large improvement of 2MN schemes when compared to the SW scheme.

3.2 Mass Preconditioning

The arguments presented in this section are made for simplicity only for the not even/odd preconditioned Wilson-Dirac operator. The generalization to the even/odd preconditioned case is simple and can be found in Ref. [108] and the appendix B.1.

Mass preconditioning [108] – also known as Hasenbusch acceleration – relies on the observation that one can rewrite the fermion determinant as follows

$$\begin{aligned} \det(Q^2) &= \det(W^+W^-)\frac{\det(Q^2)}{\det(W^+W^-)} \\ &= \int \mathcal{D}\phi_1^\dagger\mathcal{D}\phi_1\ \mathcal{D}\phi_2^\dagger\mathcal{D}\phi_2\ e^{-\phi_1^\dagger\frac{1}{W^+W^-}\phi_1-\phi_2^\dagger W^+\frac{1}{Q^2}W^-\phi_2} \\ &= \int \mathcal{D}\phi_1^\dagger\mathcal{D}\phi_1\ \mathcal{D}\phi_2^\dagger\mathcal{D}\phi_2\ e^{-S_{\mathrm{PF}_1}-S_{\mathrm{PF}_2}}\,. \end{aligned} \tag{3-20}$$

The preconditioning operators $W^{\pm}$ can in principle be freely chosen, but in order to let the preconditioning work W^+W^- should be a reasonable approximation of Q^2, which is, however, cheaper to simulate. Moreover, to allow for Monte Carlo simulations, $\det(W^+W^-)$ must be positive. The generalized Hamiltonian (3-5) corresponding to Eq. (3-20) reads

$$H = \frac{1}{2}\sum_{x,\mu} P_{x,\mu}^2 + S_{\mathrm{G}}[U] + S_{\mathrm{PF}_1}[U, \phi_1, \phi_1^\dagger] + S_{\mathrm{PF}_2}[U, \phi_2, \phi_2^\dagger]\,, \tag{3-21}$$

and it can of course be extended to more than one additional field.

Note that a similar approach was presented in Ref. [116], in which the introduction of n pseudo fermion fields was coupled with the n-th root of the fermionic kernel.

One particular choice for $W^{\pm}$ is to take for W^+ and W^- the one flavor components of a two flavor twisted mass operator

$$W^{\pm} = Q \pm i\mu\,, \tag{3-22}$$

which can be written as a two flavor operator as known from Eq. (1-47)

$$\begin{pmatrix} W^+ & \\ & W^- \end{pmatrix} = \gamma_5 \left[\begin{pmatrix} D_{\mathrm{W}} & \\ & D_{\mathrm{W}} \end{pmatrix} + i\mu\gamma_5\tau_3 \right] \tag{3-23}$$

One important property of this choice is that $W^+W^- = Q^2 + \mu^2$ and that $(W^+)^\dagger = W^-$. For small values of μ the product W^+W^- is certainly a reasonable approximation for Q^2, but due to the mass shift μ^2 it is cheaper to invert. We remark that in general also Q itself can be a twisted mass operator.

In Ref. [109, 117] it was argued that the optimal choice for μ is given by $\mu^2 = \sqrt{\lambda_{\max}\lambda_{\min}}$. Here $\lambda_{\max}$ ($\lambda_{\min}$) is the maximal (minimal) eigenvalue of Q^2. The reason for the above quoted choice is as follows: the condition number of $Q^2 + \mu^2$ is approximately $\lambda_{\max}/\mu^2$ and the one of $Q^2/(Q^2 + \mu^2)$ approximately $\mu^2/\lambda_{\min}$. With $\mu^2 = \sqrt{\lambda_{\max}\lambda_{\min}}$ these two condition numbers are equal to $\sqrt{\lambda_{\max}/\lambda_{\min}}$, both of them being much smaller than the condition number of Q^2 which is $\lambda_{\max}/\lambda_{\min}$.

Since the force contribution in the molecular dynamics evolution is supposed to be proportional to some power of the condition number, the force contribution from the pseudo fermion part in the action is reduced and therefore the step size $\Delta\tau$ can be increased, in practice by about a factor of 2 [108, 109]. Therefore Q^2 must be inverted only about half as often as before and if the inversion of W^+W^-, which is needed to compute δS_{PF_1}, is cheap compared to the one of Q^2 the simulation speeds up by about a factor of two [108, 109].

One might wonder why the reduction of the condition number from K to $\sqrt{K}$ gives rise to only a speedup factor of about 2. One reason for this is that one cannot make use of the reduced condition number of $Q^2/(Q^2+\mu^2)$ in the inversion of this operator, because in the actual simulation still the badly conditioned operator Q^2 must be inverted to compute the variation of $S_{\mathrm{PF}_2} = \phi_2^\dagger \frac{W^+W^-}{Q^2}\phi_2$.

3.2.1 Mass preconditioning and multiple time scale integration

In the last subsection we have seen that mass preconditioning is indeed an effective tool to change the condition numbers of the single operators appearing in the factorization (3-20) compared to the original operator. But, this reduction of the condition numbers only influences the forces and *not* the number of iterations to invert the physical operator Q^2.

Therefore, it might be advantageous to change the point of view: instead of tuning the condition numbers in a way à la Refs. [108, 109] we will exploit the possibility of arranging the forces by the help of mass preconditioning with the aim to arrange for a situation in which a multiple time scale integration scheme is favorable, as explained at the beginning of section 3.1.2.

The procedure can be summarized as follows: use mass preconditioning to split the Hamiltonian in different parts. The forces of the single parts should be adjusted by tuning the preconditioning mass parameter μ such that the more expensive the computation of δS_{PF_i} is, the less it contributes to the total force. This is possible because the variation of $(Q^2+\mu^2)/Q^2$ is, for $|\mu|<1$, (formally) reduced by a factor μ^2 compared to the variation of $1/Q^2$. In addition, $W^+W^- = Q^2+\mu^2$ is significantly cheaper to invert than Q^2. Then integrate the different parts on time scales chosen according to the magnitude of their force contribution.

The idea presented in this chapter is very similar to the idea of separating infrared and ultraviolet modes as proposed in Ref. [118]. This idea was applied to mass preconditioning by using only two time scales in Refs. [111, 112] in the context of clover improved Wilson fermions. However, a comparison of our results presented in the next section to the ones of Refs. [111, 112] is not possible, because volume, lattice spacing and masses are different.

3.3 Numerical results

3.3.1 Simulation points

In order to test the HMC variant introduced in the last sections, we decided to compare it with the algorithm proposed and tested in Ref. [110]. To this end we performed simulations with the same parameters as have been used in Ref. [110]:

	κ	m_q [MeV]	m_{PS} [MeV]	m_{V} [MeV]	r_0/a
A	0.1575	66(3)	665(17)	947(20)	6.04(10)
B	0.1580	34(1)	485(13)	836(24)	6.18(07)
C	0.15825	22(1)	380(17)	839(33)	6.40(15)

Table 3.1: The (un-renormalized) quark mass m_q, the pseudo scalar mass m_{PS} and the vector mass m_{V} are given in in physical units at the three simulation points A, B and C. We use Wilson fermions at $\beta = 5.6$ on $24^3 \times 32$ lattices. The scale was set by the use of $r_0 = 0.5\,\mathrm{fm}$ and we give the value of r_0/a at each simulation point. The values of all the quantities agree within the errors with the numbers quoted in Refs. [110, 68, 114], apart from the value for r_0/a at simulation point B, which disagrees by two sigmas to the value quoted in Ref. [68]. This is presumably due to the different methods to measure this quantity. For the measurements we used at each simulation point 100 thermalized configurations separated by 5 trajectories.

Wilson-Dirac operator with plaquette gauge action at $\beta = 5.6$ on $24^3 \times 32$ lattices. We have three simulation points A, B and C with values of the hopping parameter $\kappa = 0.1575$, $\kappa = 0.1580$ and $\kappa = 0.15825$, respectively. The trajectory length was set to $\tau = 0.5$. The details of the physical parameters corresponding to the different simulation points can be found in table 3.1. Additionally, this choice of simulation points allows at the two parameter sets A and B a comparison to results published in Ref. [114], where a HMC algorithm with a plain leap frog integration scheme was used.

In addition to the three simulation points A, B and C we have one additional point D with $\kappa = 0.15835$. According to Ref. [119] this value of κ corresponds to a pseudo scalar mass of about 294 MeV. Unfortunately, the history of our run is too short to be really conclusive, nevertheless we will use run D to get a preliminary idea of the performance of our algorithm towards even smaller quark and pseudo scalar masses.

3.3.2 Details of the implementation

We have implemented a HMC algorithm for two flavors of mass degenerate quarks with even/odd preconditioning and mass preconditioning with up to three pseudo fermion fields (cf. appendix B.1 on page 109). The boundary conditions are periodic in all directions apart from anti-periodic ones for the fermion fields in time direction. For the gauge action the usual Wilson plaquette gauge action (1-25) is used. The implementation is written in C and uses double precision throughout.

For the mass preconditioning we use

$$W_j^{\pm} = \gamma_5(D_{\mathrm{W}}[U, m_0] \pm i\mu_j\gamma_5)\,, \tag{3-24}$$

with $j = 1, 2$ for the factorization in Eq. (3-20), where the μ_j are the additional (unphysical) twisted mass parameters. Therefore, the pseudo fermion actions S_{PF_j} are given by

$$S_{\mathrm{PF}_j}[U] = \begin{cases} \phi_1^\dagger \left(\frac{1}{W_1^+ W_1^-} \right) \phi_1 & j = 1\,, \\ \phi_j^\dagger \left(\frac{W_{j-1}^+ W_{j-1}^-}{Q^2} \right) \phi_j & j = N_{\mathrm{PF}}\,, \\ \phi_j^\dagger \left(\frac{W_{j-1}^+ W_{j-1}^-}{W_j^+ W_j^-} \right) \phi_j & \text{otherwise}\,, \end{cases} \tag{3-25}$$

where we always chose $\mu_j > 0$ and $\mu_{j+1} > \mu_j$ for all values of j. N_{PF} denotes the actually used number of pseudo fermion fields.

We have implemented the leap frog (LF) and the Sexton-Weingarten (SW) integration schemes with multiple time scales each as described by Eq. (3-13) and Eq. (3-17), respectively, where k in both equations has to be identified with N_{PF}.

The time scales are defined as in Eq. (3-14) for the LF integration scheme and as in Eq. (3-18) for the SW scheme, with N_0 corresponding to the gauge action and N_j to S_{PF_j} for $N_{\mathrm{PF}} \geq j > 0$. Note that for the LF integration scheme for one trajectory there are $N_{N_{\mathrm{PF}}} \cdot \ldots \cdot N_j + 1$ inversions of the corresponding operator needed, while for the SW integration scheme there are $2N_{N_{\mathrm{PF}}} \cdot \ldots \cdot 2N_j + 1$ inversions needed.

For the inversions we used the CG and the BiCGstab iterative solvers. As reported in section 2.4.1 the CG iterative solver is best suited for the even/odd preconditioned twisted mass operator. Thus we used for all inversions of mass preconditioning operators exclusively the CG iterative solver.

For the pure Wilson-Dirac operator D_{W} the BiCGstab iterative solver is known to perform best [120]. In case of dynamical simulations, however, usually the squared hermitian operator needs to be inverted and in this case the CG is comparable to the BiCGstab. Only in the acceptance step, where $\gamma_5 D_{\mathrm{W}}$ (or rather the even/odd preconditioned version of it) needs to be inverted to a high precision, the usage of the CG would be wasteful. For this work we used the BiCGstab iterative solver for all inversions of either the pure Wilson-Dirac operator itself or $(\gamma_5 D_{\mathrm{W}})^2$.

The accuracy in the inversions was set during the computation of δS_{PF_j} to ϵ_j, which means that the inversions were stopped when the approximate solution ψ_j of $A_j \psi_j = \phi_j$ fulfills

$$\frac{\|\phi_j - A_j \psi_j\|}{\|\phi_j\|} \leq \epsilon_j\,,$$

where A_j denotes the operator corresponding to S_{PF_j}. During the inversions needed for the acceptance step the accuracy was set to $\tilde{\epsilon} = 10^{-10}$ for all pseudo fermion actions. The inversions in the acceptance step must be rather precise in order not to introduce systematic errors in the simulation, while for the force computation the precision can be relaxed as long as the reversibility violations are not too large. The

	Int.	N_{PF}	N_{therm}	$N_{\{0,1,2,3\}}$	$\epsilon_1, \epsilon_2, \epsilon_3$	μ_1, μ_2	P_{acc}
A	SW	3	600	$3, 2, 1, 3$	$10^{-7}, 10^{-8}, 10^{-8}$	$0.29, 0.057$	0.86
B	SW	3	1000	$3, 2, 1, 3$	$10^{-8}, 10^{-8}, 10^{-8}$	$0.25, 0.057$	0.81
C	LF	2	1500	$5, 6, 10,$ -	$10^{-8}, 10^{-8}$, -	0.054, -	0.80

Table 3.2: HMC algorithm parameters for the three simulation points. We give the integration scheme, the number of pseudo fermion fields N_{PF}, the number N_{therm} of trajectories of length 0.5 used to thermalize the systems, the number N_i of molecular dynamics steps for the multiple time scale integration scheme, the residues ϵ_i used in the solver for the force computation, the preconditioning mass parameter μ_i and the acceptance rate. We remind that N_0 corresponds to the gauge action.

values of ϵ_j and $\tilde{\epsilon}$ have been set such that the reversibility violations, which should be under control [121, 122, 123, 124], are on the same level as reported in Ref. [110], which means that the differences in the Hamiltonian are of the order[*] of 10^{-5}. The values for ϵ_j can be found in table 3.2.

The errors and autocorrelation times were computed with the so called Γ-method as explained in section 1.4.3, Eq. (1-103) on page 33 and in Ref. [78] (see also Ref. [77]).

3.3.3 Force contributions

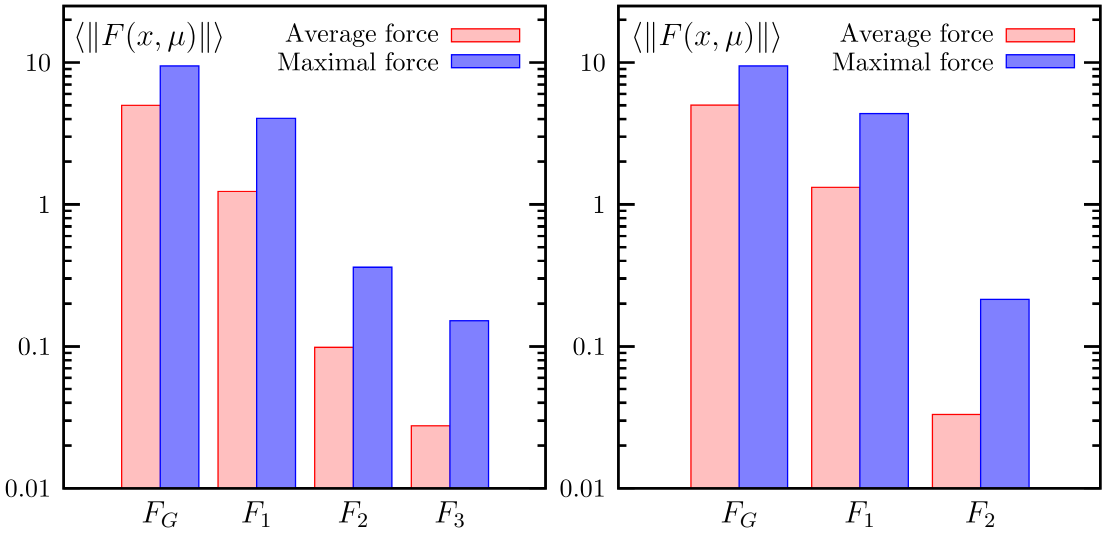
The force contributions to the total force from the separate parts in the action we label by F_G for the gauge action and by F_j for the pseudo fermion action S_{PF_j}. Since the variation of the action with respect to the gauge fields is an element of the Lie algebra of SU(3), we used $\|X\|^2 = -2\,\mathrm{Tr}\,X^2$ as the definition of the norm of such an element.

In order to better understand the influence of mass preconditioning on the HMC algorithm we computed the average and the maximal norm of the forces F_G, F_1, F_2 and F_3 on a given gauge field after all corresponding gauge field updates:

$$\begin{aligned} \|F\|_{\mathrm{aver}} &= \frac{1}{4L^3T}\sum_{x,\mu}\|F(x,\mu)\|\,, \\ \|F\|_{\max} &= \max_{x,\mu}\{\|F(x,\mu)\|\}\,, \end{aligned} \tag{3-26}$$

and averaged them over all measurements, which we indicate with $\langle . \rangle$. Examples of force distributions for different runs can be found in figure 3.1. These investigations lead to the following observations generic to our simulation points:

[*] In case of 80% acceptance rate the average value of $\sqrt{\Delta H^2}$ is about 0.1. Therefore, a reversibility violation of the order 10^{-5} is supposed to be safe.



(a) Forces for run B.

(b) Forces for run C.

Figure 3.1: Average and maximal forces for simulation points B and C. The statistical errors are too small to be visible due to the large number of measurements.

- With the choice of parameters as given in table 3.2 the single force contributions are strictly hierarchically ordered with $\|F_G\|_{\text{aver,max}} > \|F_1\|_{\text{aver,max}} > \|F_2\|_{\text{aver,max}} > \|F_3\|_{\text{aver,max}}$.
- The maximal force is up to one order of magnitude larger than the average force. This can only be explained by large local fluctuations in this quantity. These fluctuations become larger the smaller the mass is.

Moreover, the force ordering and sizes look very similar to the one reported in Ref. [110].

In a next step we performed some test trajectories without mass preconditioning in order to compare the fermionic forces with and without mass preconditioning. For the value of $\kappa = 0.15825$ (run C) the result can be found in figure 3.2. The bars labeled with F correspond to the fermion force without mass preconditioning. The labels F_1 and F_2 refer to the two fermionic forces for the run C with mass preconditioning. The following ratios are of interest:

$$\frac{\|F\|_{\text{aver}}}{\|F_1\|_{\text{aver}}} \approx 1\,, \qquad \frac{\|F\|_{\text{aver}}}{\|F_2\|_{\text{aver}}} \approx 42\,,$$
$$\frac{\|F\|_{\text{max}}}{\|F_1\|_{\text{max}}} \approx 1.3\,, \qquad \frac{\|F\|_{\text{max}}}{\|F_2\|_{\text{max}}} \approx 29\,.$$

These ratios show that the average and maximal norm of F_2 is strongly reduced compared to the average and maximal norm of F. We observe that the maximal

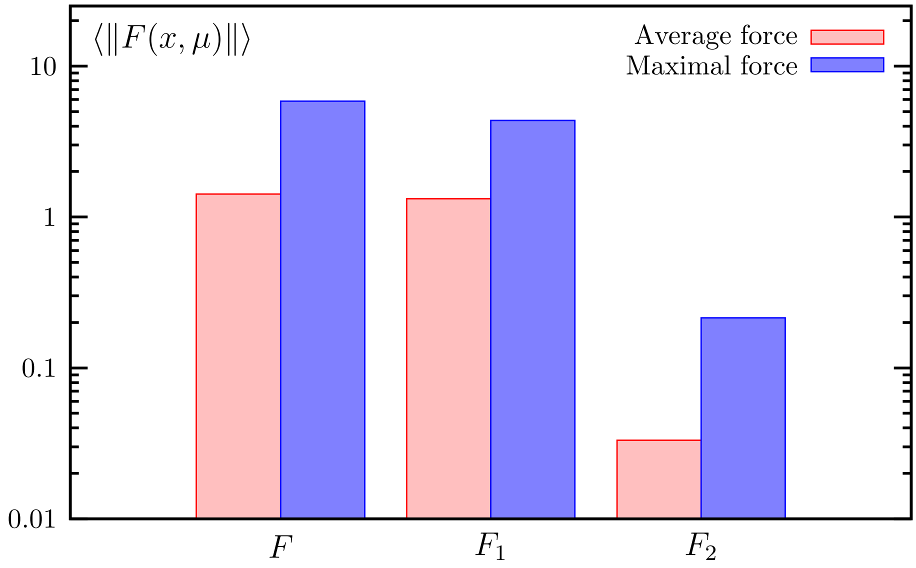


Figure 3.2: Comparison between the fermionic forces of run C (F_1 and F_2) and a run with $\kappa = 0.15825$ without mass preconditioning and multiple time scales (F). The statistical errors are too small to be visible.

norm is slightly less reduced than the average norm and, by varying μ_1, we could confirm that the norm (average and maximal) of F_2 is roughly proportional to μ_1^2.

As a further observation, one sees from figure 3.2 or from the ratios quoted above that the norm of F_1 is almost identical to the norm of F, which is the case for both the average and the maximal values.

From these investigations we think one can conclude the following: in the first place it is possible to tune the value of μ_1 (and possibly μ_2) such that the most expensive force contribution of F_2 (or F_3) to the total force becomes small. Secondly, since in the example above the force contributions for F and F_1 are almost identical – even though the masses are very different – we conclude that the norm of the forces does not explain the whole dynamics of the HMC algorithm. For this point see also the discussion in the next subsection.

3.3.4 Tuning the algorithm

As mentioned already in section 3.2.1 the tuning of the different mass parameters and time scales could become a delicate task. Therefore we decided to tune the parameters μ_1 and possibly μ_2 such that the molecular dynamics steps number $N_{N_{\mathrm{PF}}}$ for the LF or $2N_{N_{\mathrm{PF}}}$ for the SW integration scheme – the number of inversions of the original Wilson-Dirac operator in the course of one trajectory – is about the same as the corresponding values in Ref. [110]. The values we have chosen for the mass parameters μ_i and the step numbers N_i can be found in table 3.2 and one can

	κ	N_{meas}	$\langle P \rangle$	$\tau_{\mathrm{int}}(P)$
A	0.1575	740	0.57250(3)	6(2)
B	0.1580	1020	0.57339(3)	7(2)
C	0.15825	905	0.57384(4)	10(4)

Table 3.3: For the three runs this table contains the number of measurements for the plaquette N_{meas}, the mean plaquette expectation values and the corresponding autocorrelation times.

see by comparing to Ref. [110] that the step numbers N_i (or $2N_i$) are indeed quite similar.

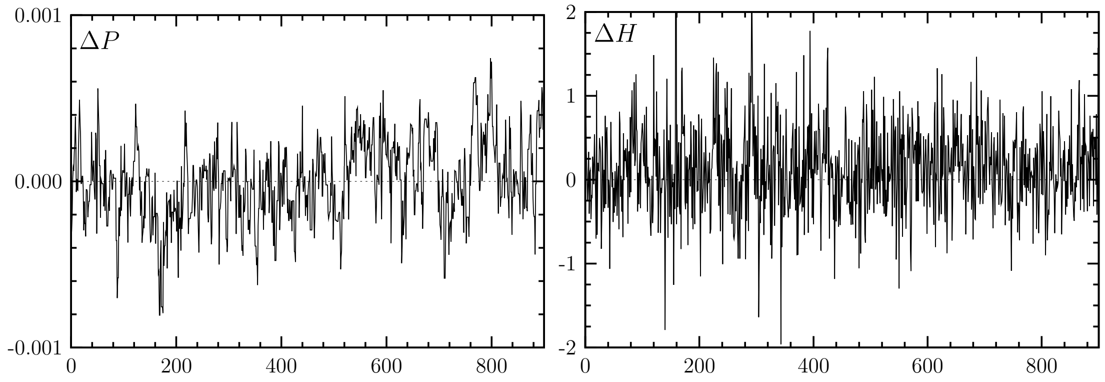
The computation of the variation of S_{G} is, compared to the variations of the other action parts, almost negligible in terms of computer time. Therefore we set N_0 always large enough to ensure that the gauge part does not influence the acceptance rate negatively and we leave the gauge part out in the following discussion.

If one compares e.g. for simulation point C the average norm of the fermionic forces, then one finds that it increases like $1 : 40$ ($\|F_2\| : \|F_1\|$). The maximal norm of the forces is accordingly strongly ordered, approximately like $1 : 20$. The corresponding relations in the step numbers we had to choose (see the values in table 3.2) increase only like $1 : 6$.

This indicates that the norm of the forces can indeed serve as a first criterion to tune the time scales and the preconditioning masses, by looking for a situation in which $\Delta\tau_i \|F_i\|_{\max}$ is a constant independent of i. But, it cannot be the only criterion. Finally, the acceptance rate is determined by $\langle \exp(-\Delta H) \rangle$, which depends in a more complicated way on the forces, see e.g. Ref. [125].

It is well known that simulations with the HMC algorithm in particular for small quark masses become often unstable if the step sizes are too large. It is an important result that with the choice of parameters as can be found in table 3.2 our simulations appear to be very stable down to quark masses of the order of 20 MeV. We did encounter only few large, but not exceptional, fluctuations in ΔH during the runs. A typical history of ΔH and the average plaquette value can be found in figure 3.3 for run C. Note that even a pion mass of about 380 MeV might be still to large to observe the asymptotic behavior of the algorithm.

All our runs reproduce the average plaquette expectation values quoted in Ref. [110] and, where available, in Ref. [114] within the statistical errors. Our results together with the number of measurements N_{meas} and the integrated autocorrelation time can be found in table 3.3. We also measured the values of the pseudo scalar, the vector and the current quark mass and our numbers agree within errors with the



(a) Monte Carlo history of ΔP.

(b) Monte Carlo history of ΔH.

Figure 3.3: Monte Carlo histories of the deviation ΔP of the average plaquette from its mean value and of ΔH, both for simulation point C.

values quoted in Refs. [110, 114]. These measurements were done on 100 configurations separated by 5 trajectories at each simulation point and we computed the aforementioned quantities with the methods explained in section 1.3. In order to improve the signal we used Jacobi smearing and random sources. Our results in physical units can be found in table 3.1. Note that the value for $m_{\rm V}$ at simulation point C has to be taken with some caution, because the lattice time extend was a bit too small to be totally sure about the plateau.

In order to set the scale we determined the Sommer parameter r_0/a [59] as defined and explained in section 1.3.4 on page 28. For our calculation of r_0/a in this chapter we used the HYP static action[†] [64, 66], the tree-level improved force and potential [59] and we enhanced the overlap with the ground state of the potential using APE smeared [60] spatial gauge links. The results can be found in table 3.1. For run A and B our values for r_0/a agree very well within the errors with the value quoted in Ref. [68, 69]. One should keep in mind, however, that the values for r_0/a are computed on rather low statistics[‡].

3.3.5 Algorithm performance

Any statement about the algorithm performance has to include autocorrelation times. Since different observables can have in general rather different autocorrelation times, also the algorithm performance is observable dependent. However, in the following we will use the plaquette integrated autocorrelation time $\tau_{\rm int}(P)$ to determine the performance.

[†] First results applying an improved static action in the computation of the static potential already appeared in [63, 69].

[‡] The computation of the values for r_0/a was performed by A. Shindler and U. Wenger.

	κ	ν	ν from [110]	ν from [114]
A	0.15750	0.09(3)	0.69(29)	1.8(8)
B	0.15800	0.11(3)	0.50(17)	5.1(5)
C	0.15825	0.23(9)	0.28(9)	-

Table 3.4: Values of the cost figure ν compared to the corresponding values of Refs. [110] and [114], where available.

The values we measured for $\tau_{\text{int}}(P)$ can be found in table 3.3. It is interesting to observe that for runs A and B the values for $\tau_{\text{int}}(P)$ are smaller than the one found for the domain decomposition method. An explanation for this may be that in the algorithm of Ref. [110] a subset of all link variables is kept fixed during the molecular dynamics evolution, while in our HMC variant all link variables are updated.

Our value for $\tau_{\text{int}}(P)$ for run A is almost identical to the corresponding one found in Ref. [114]. In contrast, for simulation point B our value is a factor of three smaller, which is – we think – partly due to the significantly smaller acceptance rate of about 60% quoted in Ref. [114] for this point and partly due to the algorithmic improvements presented in this chapter.

A measure for the performance of the pure algorithm, which is implementation and machine independent, but incorporating the autocorrelation times is provided by the cost figure

$$\nu = 10^{-3}(2n+3)\tau_{\text{int}}(P) \tag{3-27}$$

that has been introduced in Ref. [110]. n in Eq. (3-27) stands for either $N_{N_{\text{PF}}}$ in case a LF integration scheme is used or $2N_{N_{\text{PF}}}$ in case a SW integration scheme is used. ν represents the average number of inversions of the Wilson-Dirac operator with the physical mass in units of thousands as needed to generate a statistically independent value of the average plaquette. Hence, in giving values for ν, we neglect the overhead coming from the remaining parts of the Hamiltonian.

Our values for ν together with the corresponding numbers from Ref. [110] and Ref. [114] are given in table 3.4. Compared to Ref. [110] our values for ν are smaller for simulation points A and B and comparable for run C. In contrast, the cost figure for the HMC algorithm with plain leap frog integration scheme is at least a factor 10 larger than the values found for our HMC algorithm variant. This gain is, of course, what we aimed for by combining multiple time scale integration with mass preconditioning and hence confirms our expectation. Unfortunately, due to the large statistical uncertainties of the ν values it is not possible to give a scaling of the cost figure with the mass. This holds for our values of ν as well as the ones of Ref. [110].

	N_{MV}			$\tau_{\mathrm{int}}(P) \cdot \sum N_{\mathrm{MV}}$	
	S_{PF_1}	S_{PF_2}	S_{PF_3}	this work	Ref. [114]
A	3800	4600	6600	90000	190750
B	6000	6900	11900	173600	1280000
C	31000	25500	-	565000	–

Table 3.5: Rounded number of matrix vector multiplications needed during one trajectory of length 0.5 for the different pseudo fermion actions without the usage of a chronological solver guess. We give also the sum of our numbers multiplied by the plaquette autocorrelation time and as a comparison the corresponding number from Ref. [114], where available.

3.3.6 Simulation cost

Although the value of ν is a sensible performance measure for the algorithm itself, since it is independent of the machine, the actual implementation and the solver, it cannot serve to estimate the actual computer resources (costs) needed to generate one independent configuration. Assuming that the dominant contribution to the total cost stems from the matrix vector (MV) multiplications, we give in table 3.5 the average number of MV multiplications N_{MV} needed for the different pseudo fermion actions to evolve the system for one trajectory of length $\tau = 0.5$. In addition we give the sum of these MV multiplications multiplied with the plaquette autocorrelation time together with the corresponding number from Ref. [114].

In order to compare to the numbers of Ref. [114] we remark that the lattice time extent is $T = 40$ in Ref. [114] compared to $T = 32$ in our case, but we do not expect a large influence on the MV multiplications coming from this small difference. Large influence on the MV multiplications, however, we expect from ll-SSOR preconditioning [126] that was used in Ref. [114] in combination with a chronological solver guess (CSG) [127].

Initially, when one compares the values of the cost figure for our HMC algorithm with the one of the plain leap frog algorithm as used in Ref. [114], one might expect that the number of MV multiplications shows a similar behavior as a function of the quark mass. However, inspecting table 3.5, we see that in terms of MV multiplications at simulation point A the HMC algorithm of Ref. [114] is only a factor of 2 slower than the variant presented in this chapter, while the values of ν are by a factor of about 20 different. The reason for this is two-fold: On the one hand ll-SSOR preconditioning together with a CSG method is expected to perform better than only even/odd preconditioning. On the other hand we think that the quark mass at this simulation point is still not small enough to gain significantly from multiple time scale integration. This illustrates that indeed the value of ν is not immediately conclusive for the actual cost of the algorithm.

At simulation point B the relative factor between the MV multiplications needed by the two algorithms is already about 7. And finally, it is remarkable that for simulation point C the costs with our HMC variant are still a factor of 2 smaller than the costs for simulation point B with the algorithm used in Ref. [114], even though the masses are very different.

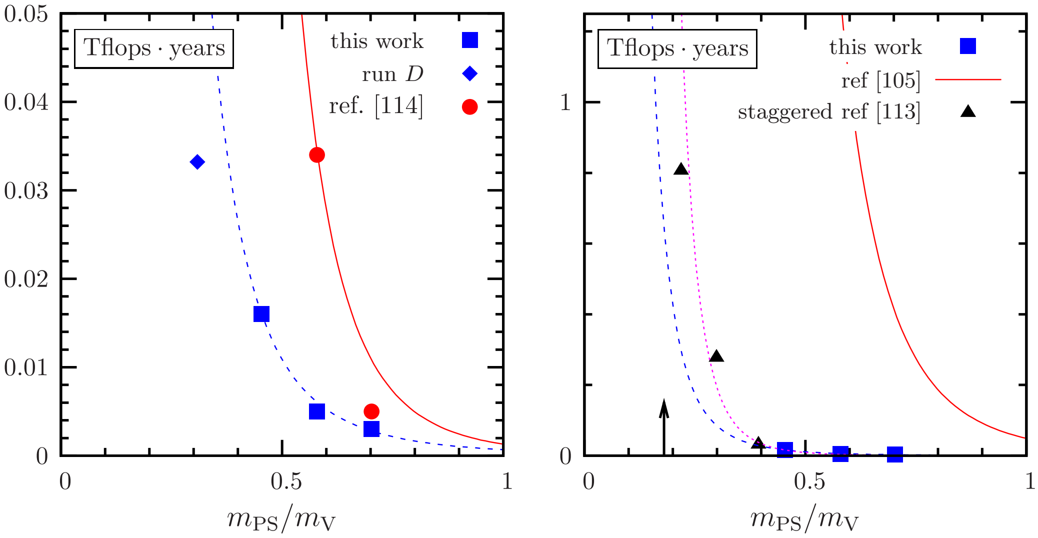
From this comparison we conclude that especially in the regime of small quark masses the HMC algorithm presented in this work is significantly faster than a HMC algorithm with single time scale leap frog integration scheme.

By looking at table 3.5 one notices that especially for simulation point C the number of MV multiplications needed for preconditioning is larger than the one needed for the physical operator. This comes from the fact that with the choice of algorithm parameters we have used the number of molecular dynamics steps for the mass preconditioned operator is large. This possibly indicates potential to further improve the performance by tuning the preconditioning masses and time scales.

We stress here again that the number of matrix vector operations is highly solver dependent, and therefore, every improvement to reduce the solver iterations will decrease the cost for one trajectory. Promising improvements are for instance the use of a chronological inversion method [127] (or similar methods [128]) or the use of a solver based on domain decomposition as adopted for QCD in Ref. [129]. We tested the chronological inversion method and found in total not more than 20% gain in matrix vector operations.

Finally, it is interesting to compare the number of matrix vector multiplications reported in table 3.5 with a HMC algorithm where mass preconditioning and multiple time scale improvements are switched off and CSG is not used. For instance for a simulation with a Sexton-Weingarten improved integration scheme at $\kappa = 0.15825$ there are 120 molecular dynamics steps needed to get acceptance. This corresponds to 240 inversions of Q^2, which amounts to about 720000 matrix vector multiplications. Compared to run C this is at least a factor 10 more. We did only a few trajectories to get an estimate for this number, so we cannot say anything about autocorrelation time.

Of course it would be interesting to compare also to a HMC algorithm with mass preconditioning but without multiple time scale integration. This, however, needs again a tuning of the mass parameters and would therefore be quite costly and we did not attempt to test this situation here.



(a) Comparison to Ref. [114].

(b) Update of the "Berlin Wall" plots of Refs. [105, 113].

Figure 3.4: Computer resources needed to generate 1000 independent configurations of size $24^3 \times 40$ at a lattice spacing of about 0.08 fm in units of Tflops · years as a function of $m_{\mathrm{PS}}/m_{\mathrm{V}}$. In (a) we compare our results represented by squares to the results of Ref. [114] represented by circles. The lines are functions proportional to $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-4}$ (dashed) and $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-6}$ (solid) with a coefficient such that they cross the data points corresponding to the lightest pseudo scalar mass. The diamond represents the preliminary result of run D (see text). In (b) we compare to the formula of Eq. 3-28 [105] (solid line) by extrapolating our data with $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-4}$ (dashed) and with $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-6}$ (dotted), respectively. The arrow indicates the physical pion to rho meson mass ratio. Additionally, we add points from staggered simulations as were used for the corresponding plot in Ref. [113]. Note that all the cost data were scaled to match a lattice time extend of $T = 40$.

3.3.7 Scaling with the mass

An important property of an algorithm for lattice QCD is the scaling of the costs with the simulated quark mass. The naive expectation is that the number of solver iterations grows like m_q^{-1} and also the number of molecular dynamics steps is proportional to m_q^{-1}, see for instance Ref. [130] or Ref. [105]. Since also the integrated autocorrelation time is assumed to grow like m_q^{-1}, it is expected that the HMC algorithm costs scale with the quark mass as m_q^{-3} or equivalently as m_{PS}^{-6}. In contrast, for our HMC algorithm variant we expect a much weaker scaling of $\Delta\tau$ and also of the number of solver iterations. Indeed, we see that the costs for our HMC algorithm variant is consistent with a m_q^{-2} or m_{PS}^{-4} behavior when the autocorrelation time is taken into account.

We have translated the number of matrix vector multiplications from table 3.5 into costs in units Tflops · years and plotted the computer resources needed to generate 1000 independent configurations of size $24^3 \times 40$ at a lattice spacing of ~ 0.08 fm as a function of $m_{\mathrm{PS}}/m_{\mathrm{V}}$ in figure 3.4(a) together with the results of Ref. [114]. Note that we have scaled our costs like $(40/32)^{1.25}$ corresponding to the expected volume dependence (cf. [105]) to match the different time extents and, moreover, we used the plaquette autocorrelation time as an estimate for the autocorrelation time.

The solid (dashed) line is not a fit to the data, but a function proportional to $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-4}$ ($(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-6}$) with a coefficient that is fixed by the data point corresponding to the lightest pseudo scalar mass. These functional dependencies on $(m_{\mathrm{PS}}/m_{\mathrm{V}})$ describe the data reasonably well. However, from our few data points it is not possible to decide on the value of the exponent in the quark mass dependence of the costs. But, it is clear from the figure that with multiple time scale integration and mass preconditioning the "wall" – which renders simulations at some point infeasible – is moved towards smaller values of the quark mass.

An additional indication for the scaling properties of the algorithm towards smaller masses is given by the preliminary result of run D. It is represented in figure 3.4(a) by the single diamond. For this point we used our current number of MV multiplications as measured for run D and the value for m_{PS} as given in Ref. [119]. Moreover, we extrapolated the value am_{V} in κ and the value of τ_{int} in $1/m_{\mathrm{PS}}^2$. The result as we plot it in the figure thus has certainly a significant error. Nevertheless, even if the "true" result will be a factor of two larger, the point is still in excellent agreement with the anticipated scaling proportional to $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-4}$.

On a larger scale we can compare the extrapolations of our cost data to the formula given in Ref. [105]

$$C = K \left(\frac{m_{\mathrm{PS}}}{m_{\mathrm{V}}}\right)^{-z_\pi} L^{z_L}\ a^{-z_a}\,, \tag{3-28}$$

where the constant K can be found in Ref. [105] and $z_\pi = 6$, $z_L = 5$ and $z_a = 7$. The result of this comparison is plotted in figure 3.4(b), which is an update of the "Berlin Wall" figure that can be found in Ref. [113]. We plot the simulation costs in units of Tflops · years versus $m_{\mathrm{PS}}/m_{\mathrm{V}}$, where we again scaled the numbers in order to match a lattice time extend of $T = 40$. The dashed and the dotted lines are extrapolations from our data proportional to $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-4}$ and $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-6}$, respectively, again matching the data point corresponding to the lightest pseudo scalar mass. The solid line corresponds to Eq. (3-28) with K taken from Ref. [105]. In addition we plot data from staggered simulations as were used for the plot in Ref. [113]. That the corresponding points lie nearly on top of the dotted line is accidental.

Conservatively one can conclude from figure 3.4(b) that with the HMC algorithm described in this chapter at least simulations with $m_{\mathrm{PS}}/m_{\mathrm{V}} \approx 0.3$ are feasible, even though $L = 1.93\,\mathrm{fm}$ is too small for such values of the masses. Taking the more optimistic point of view by assuming that the costs scale with $z_\pi = 4$, even simulation with the physical $m_{\mathrm{PS}}/m_{\mathrm{V}}$ ratio and a lattice spacing of 0.08 fm become accessible, with again the caveat that L/a needs to be increased.

Independent of the value for z_π, figure 3.4(b) reveals that the costs for simulations with staggered fermions and with Wilson fermions in a comparable physical situation are of the same order of magnitude, if for the simulations with Wilson fermions an algorithm like the one presented in this work is used. It would be interesting to see whether the techniques applied in this work perform similarly well for staggered fermions.

We would like to point out that we did not try to tune the parameters to their optimal values. The aim of this work was to give a first comparison of mass preconditioned HMC algorithm with multiple time scale integration to existing performance data, i.e. data for a HMC algorithm preconditioned by domain decomposition [110] and data for the HMC algorithm variant of Ref. [114]. We are confident that there are still improvements possible by further tuning of the parameters in our variant of the HMC algorithm.

3.4 Conclusion

In this chapter we have presented and tested a variant of the HMC algorithm combining multiple time scale integration with mass preconditioning (Hasenbusch acceleration). It is based on the idea to arrange mass preconditioning such that the force contributions from the different parts in the Hamiltonian are strictly ordered with respect to the absolute value of the force and that the most expensive part has the smallest contribution to the total force. Then the most expensive part can be integrated on the largest time scale.

Our aim was to perform a first investigation of the performance properties of this HMC algorithm by comparing it to other state of the art HMC algorithm variants in the same physical situation, i.e. for pseudo scalar masses in the range of 380 to 670 MeV, a lattice spacing of about 0.08 fm and a lattice size of $L \approx 2$ fm with two flavors of mass degenerate Wilson fermions. We verified our implementation by comparing results for the plaquette and for the pseudo scalar, the vector and the current quark mass to results available in the literature finding full agreement.

We have shown that indeed the aforementioned idea can be realized by tuning the additional (unphysical) mass parameters introduced for mass preconditioning. In this set-up the performance of our variant in terms of the cost figure in Eq. (3-27) is compatible to the one observed for the HMC algorithm with multiple time scales and domain decomposition as preconditioner introduced in Ref. [110] and clearly superior to the one for the HMC algorithm with a simple leap frog integration scheme as used in Ref. [114].

While the cost figure provides a clean algorithm performance measure we also compare the simulation costs in units of Tflops · years to existing data. This comparison is summarized in an update of the "Berlin wall" plot of Ref. [113], which can be found in figure 3.4. We could show that with the HMC algorithm presented in this chapter the wall is moved towards smaller values of the quark mass and that simulations with a ratio of $m_{\mathrm{PS}}/m_{\mathrm{V}} \approx 0.3$ become feasible at a lattice spacing of around 0.08 fm and $L \approx 2$ fm. We have preliminary results for a simulation point with a pseudo scalar mass of around 300 MeV and $m_{\mathrm{PS}}/m_{\mathrm{V}} \approx 0.3$, which is in excellent agreement with all the results mentioned above. In particular this simulation point seems to confirm that the algorithm costs scale proportional to $(m_{\mathrm{PS}}/m_{\mathrm{V}})^{-4}$.

The HMC variant presented here has the advantage of being applicable to a wide variety of Dirac operators, including in principle also the overlap operator. In addition its implementation is straightforward, in particular in an already existing HMC code. We remark that the parallelization properties of our HMC variant and the one of the algorithm presented in [110] can be very different depending on whether a fine- or a coarse-grained massively parallel computer architecture is used.

From a stability point of view our results reveal that even for Wilson fermions it is very well possible to simulate quark masses of the order of 20 MeV when using the algorithmic ideas presented in this work. The presently ongoing simulation with even smaller quark mass is also running without any practical problems, but the statistics is not yet adequate to say something definite. However, it is a remarkable result by itself that there are now at least two algorithms available allowing for *stable* simulations with Wilson fermions at low values of the quark masses. Remarkable, because only short time ago this was thought to be hardly possible and it immediately raises the question for an explanation: one can speculate that the observed stability is mainly due to noise reduction provided by the additional pseudo fermion fields and former simulations yielded problems, because the stochastic approximation for the determinant was not sufficient.

The results presented in this chapter are mostly based on empirical observations and on simulations for only one value of the coupling constant $\beta = 5.6$. It remains to be seen how our HMC variant behaves for larger values of β, which, as well as smaller quark masses and theoretical considerations about the scaling properties with the quark mass needs further investigations. Moreover, a more systematic study of the interplay between integration schemes, step sizes, (preconditioning and physical) masses and the simulation costs is needed. Those investigations will hopefully also provide a better understanding of the algorithm itself and its dynamics. Of course, the algorithm should also be tested for tmQCD, even though we do not expect a large difference to the pure Wilson case.

Finally, we think that there are further improvements possible by the usage of a Polynomial HMC (PHMC) algorithm [131, 132, 133, 134]. With such an algorithm one could treat the lowest eigenvalues of the Dirac operator exactly and/or by reweighting. In this set-up the large fluctuations in the force might be significantly reduced, if the lowest eigenvalues are responsible for those. Then it might be possible to further reduce the number of inversions of the badly conditioned physical operator needed to evolve the system. In addition, a PHMC algorithm would immediately allow for simulations with three or more flavors of quarks.