**Innovations in Lattice QCD Algorithms**

**Konstantinos Orginos**

Department of Physics, The College of William and Mary, Williamsburg, VA 23187-8795.
Jefferson Laboratory, 12000 Jefferson Avenue, Newport News, VA 23606.

E-mail: kostas@jlab.org

**Abstract.** Lattice QCD calculations demand a substantial amount of computing power in order to achieve the high precision results needed to better understand the nature of strong interactions, assist experiment to discover new physics, and predict the behavior of a diverse set of physical systems ranging from the proton itself to astrophysical objects such as neutron stars. However, computer power alone is clearly not enough to tackle the calculations we need to do today. A steady stream of recent algorithmic developments has made an important impact on the kinds of calculations we can currently perform. In this talk I am reviewing these algorithms and their impact on the nature of lattice QCD calculations performed today.

1. **Introduction**

Quantum Chromo Dynamics (QCD), the dynamics of chroma ($\chi$), is the theory describing the interactions of quarks and gluons, the building blocks of protons and nucleons. Color is the charge both quarks and gluons carry. Unlike electrodynamics, which provides a weak force at low energies, QCD is weak at high energies. This phenomenon, called asymptotic freedom and it is direct consequence of the fact that the gluons, the carriers of the interaction, are themselves charged with color [1, 2]. However, the weak force at high energies becomes strong at low energies making QCD intractable with analytic methods. The lattice formulation of QCD provides a powerful tool to calculate the consequences of the theory at low energies. A plethora of phenomena can be understood using results from lattice QCD. The calculation of the mass spectrum of hadrons (protons, neutrons, pions etc.), the physics of matter produced at heavy ion collisions (SPC, RHIC, LHC) [3, 4], and to the discovery of new physics [5, 6, 7] are a few of the areas lattice QCD can have a significant impact.

In this paper I describe some of the algorithms used in lattice QCD calculations. In particular, I am focusing on some recent developments that allow us to make computations with small and controlled systematic errors with current computing resources. In the subsequent sections, I first describe the basic setup of the calculation and point out the numerical algorithms used. Then I present some recent modifications to these basic algorithms that greatly improve the efficiency of calculation.

2. **Lattice QCD**

Quantum Chromo Dynamics, being a quantum field theory, can be defined in the continuum through the path integral formalism

$$Z = \int DA_\mu D\bar{\psi} D\psi e^{\int d^4x \left( -\frac{i}{4} F_{\mu\nu} F^{\mu\nu} + \bar{\psi} i D_\mu \gamma_\mu - m \bar{\psi} \psi \right)}$$ (1)
where $A_\mu$ is the gauge field representing the gluons, $F_{\mu\nu}$ is the gauge field strength and $\bar{\psi}$, $\psi$ are the fermion fields representing the quarks. $D_\mu$ is the covariant derivative which ensures gauge invariance and $\gamma_\mu$ are matrices satisfying the Clifford algebra. The details of this equation are not important for what follows and the interested reader should consult standard quantum field theory text books for details. The physical quantities in this theory can be calculated as correlation functions of operators $O$ that are functions of the quantum fields (quarks and gluons).

\[ \langle O \rangle = \frac{1}{Z} \int D\psi D\bar{\psi} D\psi e^{\int d^4x \left( -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \bar{\psi} [i D_\mu \gamma_\mu - m] \psi \right)} \]  

(2)

The above seemingly simple definitions hide a lot of the complexities of quantum field theory. In particular in order to rigorously define the path integrals above we need to discretize the continuum space time converting the path integral into a large but finite dimension regular integral. The limit of the discretization scale $a$ (lattice spacing) going to zero is the continuum path integral. This is how the lattice formulation [8, 9] of QCD first comes into play. Fortunately, this formulation is also a great tool to perform numerical calculations in this theory if we rotate to real time to imaginary time (i.e. go to Euclidean space).

In order to preserve gauge invariance the gauge fields are discretized as special unitary matrices, $SU(3)$, living on the links of the lattice (see Figure 1). The discrete gauge action is given by the sum over all plaquettes $P_{\mu\nu}(x)$ which are the product of the links $U$ going around the elementary plaquettes of the lattice.

\[ S_g(U) = \beta \sum_{x_{\mu\nu}} \left( 1 - \frac{1}{3} \text{Re} \text{Tr} P_{\mu\nu}(x) \right) \]  

(3)

with

\[ P_{\mu\nu} = U_\mu(x) U_\nu(x + \hat{\mu}) U_\mu^\dagger(x + \hat{\nu}) U_\nu^\dagger(x), \]  

(4)

and $\beta$ the lattice gauge coupling. Taking the lattice spacing to zero the above action becomes the continuum gauge action $\int d^4x \frac{1}{2} F_{\mu\nu}^2(x)$. This is the well know Wilson gauge action. This discretization is not unique but it’s the simplest. One can modify this discrete action in order to achieve better convergence to the continuum limit, which is the ultimate goal of the calculation.
The fermions, which live on the vertices of the lattice, present a more challenging problem. Naive discretization results in the so-called fermion doubling problem, i.e., lattice fermions come naturally in sixteen copies, much too many for describing real QCD which has three light quarks (up, down, and strange) and three heavy quarks (top, bottom, and charm). The doublers can be avoided by several ingenious formulations of lattice fermions. For the purpose of this paper, describing the variants of these fermionic discretizations is not essential. All we need to keep in mind is that the lattice fermion action can be written as following

\[ S_f = \bar{\psi} D(U) \psi \]  

where \( \psi \) is the fermion "vector" and \( D(U) \) is a sparse matrix acting on this vector, that depends on the gauge field \( U \).

Now we can write down the lattice theory. First the partition function in the case of two quark flavors

\[ Z = \int \prod_{\mu,x} dU_\mu(x) \prod_x d\bar{\psi} d\psi \ e^{-S_g(U) - S_f(\bar{\psi}, \psi, U)} = \int \prod_{\mu,x} dU_\mu(x) \ \det (D(U)^\dagger D(U)) \ e^{-S_g(U)}. \]  

Note that we need to integrate out the quarks since they are represented by Grassman numbers. These are anti-commuting numbers with which we know how to do integrals but can’t represent in the computer. In case of correlation functions we also integrate out the quarks resulting the following expression

\[ \langle O \rangle = \frac{1}{Z} \int \prod_{\mu,x} dU_\mu(x) \ O \left( \frac{1}{D(U)}, U \right) \ \det (D(U)^\dagger D(U)) \ e^{-S_g(U)}. \]  

The integration over the Grassman fermion fields results operators \( \mathcal{O} \) that depend on the inverse of the fermion matrix. Here I have to point out that the above manipulation is only valid for the case of two flavors of quarks (the up and the down) which both have the same mass. This is a good approximation to the low energy physics of QCD. The formulation the includes the strange quark exists but for simplicity I will avoid presenting it here.

3. Monte Carlo sampling
The computation of Eq. 7 is the main numerical task we phase in lattice QCD calculations. The integral in Eq. 7 over the gauge fields is of extremely large dimensionality. Considering that we have discretized QCD which has a fundamental scale of 1fm (\( \sim 1 e^{-13} cm \)), we need to work with a lattice that has a much larger than 1fm physical size in order to control finite volume effects, with a much smaller than 1fm lattice spacing in order to control the continuum limit. With moderate choices for the volume and the lattice spacing we come up with typical lattice sizes of \( 32^4 \). Counting the color, flavor and spin degrees of freedom we can see that we are dialing with \( \approx 10^8 \) degrees of freedom. The only way this computation can be done is by using Monte Carlo integration. Fortunately, the combination of the fermionic determinant and the gauge action,

\[ \mathcal{P}(U) = \frac{1}{Z} \det (D(U)^\dagger D(U)) \ e^{-S_g(U)}, \]  

is a positive definite quantity which can be interpreted as a probability, hence we can apply importance sampling. So in our basic algorithm we produce \( N \) gauge field configurations \( \{U\} \) with probability distribution \( \mathcal{P}(U) \) and then we evaluate

\[ \langle \mathcal{O} \rangle = \frac{1}{N} \sum_{i=1}^N \mathcal{O}(U_i, \frac{1}{D(U_i)}) \]  

1 In certain cases such as the overlap fermions the matrix is not sparse but has sparse like properties i.e. the matrix vector multiplication is a cheap operation.
As we will see in the following, both for the gauge field configuration generation and the evaluation of Eq. 9 we need to solve the linear system of equations

\[ D^\dagger(U)[m]D(U)[m]\chi = \phi, \] (10)

where here I explicitly denote the dependence of the fermionic matrix on the quark mass \( m \). Since the fermionic matrix is sparse, iterative solvers such as conjugate gradient can be used. The condition number of the fermionic matrix is inversely proportional to the quark mass. Since the physical quark masses for the up and down quarks are quite small the fermionic matrix has a bad condition number. With current computer resources this linear system cannot be solved exactly at the physical quark mass. For that reason we perform the calculation at heavier quark masses and then we extrapolate the results to the physical point. The vast majority of the computer time used is devoted to the solution of this linear system both in the context of gauge field generation and in the later stage of the calculation of physical observables through Eq. 9.

3.1. Hybrid Monte Carlo

Hybrid Monte Carlo (HMC) [10, 11] is the algorithm used to generate gauge field configurations need for performing the importance sampling outlined in the previous paragraph. The basic feature of our problem that needs special treatment is the fermionic determinant that makes the couplings between gauge links non-local. Standard algorithms such as Metropolis or Heat bath, rely on locality for achieving their efficiency hence they are not suited to deal with this problem.

In order to proceed we introduce a bosonic complex valued field to exponentiate the determinant,

\[ \det(D(U)^\dagger D(U)) = \int \phi^\dagger \phi e^{-\phi^\dagger D^\dagger(U)D(U)\phi}. \] (11)

Then we introduce ”momenta” \( P_\mu(x) \) conjugate to the gauge links \( U_\mu(x) \). These are traceless hermitian \( 3 \times 3 \) matrices. These variables are random gaussian variables. In other words their action is quadratic.

\[ S_P = \frac{1}{2} \sum_{\mu,x} P_\mu(x)^2 \] (12)

Now the partition function is

\[ Z = \int \prod_{\mu,x} dU_\mu(x) \prod_x d\phi^\dagger d\phi e^{-S_P(P) - S_g(U) - \phi^\dagger D^\dagger(U)D(U)\phi}. \] (13)

It only differs from the original partition function by a multiplicative constant. This has no effect on correlation functions so we have not changed the physical content of our theory.

In order to generate gauge fields with probability distribution given by Eq. 4, we need to make a sequence of updates to the gauge fields that satisfy detailed balance and ergodicity. This is achieved by updating the extended configuration \( \{P,U\} \) as following:

- Update \( P_\mu(x) \) using gaussian random noise
- Update \( \phi \) using gaussian random noise \( \eta \) by

\[ \phi = D^\dagger(U)\eta \] (14)

- Evolve \( \{P,U\} \) in a fictitious time using the hamiltonian equations of motion resulting from the hamiltonian

\[ H = \frac{1}{2} \sum_{x,\mu} P_\mu(x)^2 + S_g(U) + \phi^\dagger \frac{1}{D^\dagger(U)D(U)} \phi \] (15)
- Accept or reject the final configuration \( \{ P', U' \} \) with probability

\[
P_{\text{accept}} = \min \left\{ 1, e^{-[\mathcal{H}(P', U') - \mathcal{H}(P, U)]} \right\}
\]

(16)

The final step is not needed if one solves exactly the Hamiltonian equations of motion, which are deterministic and preserve the energy (\( \mathcal{H} \)) hence the update satisfies detailed balance. Since we need to solve numerically the equations of motion the energy is not preserved and the last step is needed to correct the violation of the detailed balance condition. One additional requirement of detailed balance is the reversibility of the integration algorithm. For that reason the leapfrog algorithm is used. Recently [12] the so called Omelian integrators which are also reversible have been shown to be more efficient. For simplicity in the remaining I use leapfrog as the integrator of choice.

The discrete Hamiltonian equations of motion dictate the following update for the gauge fields and the momenta

\[
T_U(\epsilon) : \quad U \rightarrow e^{i\epsilon P} U
\]

\[
T_P(\epsilon) : \quad P \rightarrow P + \epsilon F,
\]

(17)

where \( \epsilon \) is the time step and \( F \) is the force given by the variation of the Hamiltonian with respect to the gauge field,

\[
F = -\frac{\delta \mathcal{H}}{\delta U}.
\]

(18)

This variation is along the SU(3) manifold so that \( F \) is a traceless hermitian 3 \( \times \) 3 matrix\(^2\). The evolution operators in Eq. 17 can be used to describe the leapfrog evolution as following

\[
T(\epsilon) = T_P(\epsilon/2) T_U(\epsilon) T_P(\epsilon/2)
\]

\[
T_{\text{lf}}(\epsilon, N) = T(\epsilon)^N,
\]

(19)

where \( N \) is the total number of time steps. Typically \( \epsilon N \approx 1 \) is used, and \( \epsilon \) is tuned so that the acceptance rate is roughly 65\%. The largest \( \epsilon \) can be made preserving acceptance the smaller the cost of the algorithm is. If acceptance drops too much then the autocorrelation time grows, while if acceptance is too high excessive work in the evolution is done with out significant reduction in autocorrelation time.

In order to conclude the description of the HMC algorithm let me point out that the force calculation is the most costly part of the calculation. Because of the inverse of the fermion matrix involved in the fermionic part of the action one needs to solve a linear system of the form of Eq. 10. This solution is done using iterative solvers such as conjugate gradient with the aid of suitable preconditioners.

Finally, let me point out that there exist other algorithms such as the R and \( \Phi \) algorithms [10], the Polynomial HMC [13, 14, 15, 16] and the Rational HMC [17] which I cannot describe here. A lot of the basic features of the HMC algorithm are common with the rest of the algorithms. The Rational HMC seems to be a very good choice for working in the case of odd number of flavors i.e. including the strange quarks. In addition, there exist other algorithms such as the multi-boson algorithm [18] that are quite different from HMC but they have not being proven to be more efficient than HMC.

\(^2\) The momenta \( P \) are really conjugate to the fields \( A \) that are defined by \( U = e^{iA} \). So the equation 18 implies a variation with respect to \( A \).
3.2. The Berlin Wall

The cost of HMC is rather high. The algorithm scales poorly as one approaches the continuum limit and takes the quark mass closer to the physical point. The cost as a function of the parameters of the calculation scales as [19, 20, 21]

\[ C = K \left( \frac{m_{ps}}{m_v} \right)^{-z} V^{5/4} \frac{1}{a^2}, \]  

(20)

where \( a \) is the lattice spacing, \( \frac{m_{ps}}{m_v} \) is proportional to the square root of the quark mass, \( V \) is the volume of the system, \( K \) is a constant of appropriate dimensions, and \( z \) is an exponent that ranges between 4 and 6. This formula tells us that as one moves close to the physical point where \( \frac{m_{ps}}{m_v} \approx 0.18 \), the cost explodes. Though this scaling was known the lattice QCD community discussed its consequences on realistic lattice calculations during the 2001 Lattice conference in Berlin. Hence this bad scaling is now referred as the "Berlin Wall". Back then it was only possible to compute using the Wilson fermion action at \( \frac{m_{ps}}{m_v} > 0.5 \). Since then intense research in algorithm modifications has resulted major improvements that shifted the "Wall" much closer to the physical point.

Before turning to the description of these algorithm improvements, let me point out that one can use the freedom of choosing the fermion discretization method to reduce the cost of the calculation. Such cheap fermionic action variants are the Kugot-Susskind fermions which have been successfully used for important phenomenological calculations [5, 6, 7]. Unfortunately, there are problems that these cheap fermions cannot be used for, hence there is need to develop better algorithms for other fermionic actions. The twisted mass QCD is an other cheap lattice fermion variant that is currently under investigation. Both these variants benefit immensely from innovative new algorithms [22, 21, 23, 24].

4. Algorithm improvements

Although the "Berlin Wall" was faced in 2001, some algorithmic improvements of the HMC algorithm had started before that. In this section I will describe some of these developments in chronological order.

4.1. Chronological inverter

As pointed out before, the most costly part of HMC is the calculation of the force. A key part of this calculation is the solution of a linear system shown in Eq. 10 at every time step of the leapfrog algorithm. An important feature of this system is that the right hand side is fixed during the hamiltonian evolution of the configuration. The fermion matrix though is evolving smoothly in time, hence the solutions form a smooth time sequence. As a result, knowing the solutions of previous time steps we can make a reasonable guess of the solution at any given time step. This is achieved by minimizing the norm of the error, defined using the fermionic matrix, in the subspace spanned by the past solutions. If this method is used, special care is needed to ensure the reversibility of the integration algorithm by requiring a stringent stopping condition on the solver. The chronological inverter [25] achieves a factor of 2 to 3 speed up over standard HMC implementation. Unfortunately, the new algorithms described in the following achieve large enough time steps to make the benefit of this algorithm less important.

4.2. Multiple time steps

The fact that the fermionic part of the force requires the linear system solve makes this part of the force much more expensive than the force coming from the gauge piece of the action. Large force results larger energy violation at the end of the trajectory, enforcing smaller time step. It was realized in [26] that one can obtain a leapfrog algorithm with multiple time steps
breaking up the update of the momenta in two pieces. One coming from the large and cheap gauge force and one coming from the small and expensive fermion force. The resulting algorithm has two time steps. One large associated with the fermion force and one small for the gauge force. Hence the expensive fermion force is only evaluated infrequently during the hamiltonian evolution. This algorithm can be written down in a concise form using the evolution operators of Eq. 17 and Eq. 19 as following

$$T^g_p(\epsilon) : P \rightarrow P + \epsilon F^g$$
$$T^f_p(\epsilon) : P \rightarrow P + \epsilon F^f,$$  
(21)

with the two forces been defined by

$$F^g = -\frac{\delta S_g}{\delta U}$$
$$F^f = -\frac{\delta S_f}{\delta U}. $$  
(22)

The algorithm consists of a small time step $\epsilon_1$ evolution with basic update step defined by

$$T_1(\epsilon_1) = T^g_p(\epsilon_1/2)T_U(\epsilon_1)T^g_p(\epsilon_1/2)$$  
(23)

and a large time step $\epsilon_2$ evolution whose basic update step is

$$T_2(\epsilon_1, \epsilon_2, N_1) = T^f_p(\epsilon_2/2)T^{N_1}_p(\epsilon_1)T^f_p(\epsilon_2/2).$$  
(24)

The total evolution is described in terms of these basic update steps as following

$$T_{mts}(\epsilon_1, \epsilon_2, N_1, N_2) = T^{N_2}_2(\epsilon_1, \epsilon_2, N_1).$$  
(25)

In the above evolutions one needs to satisfy $N_1 = \epsilon_2/\epsilon_1$. $N_1$ and $N_2$ are integers denoting the total number of time steps in each trajectory. This algorithm, though not widely used when first proposed, is very effective in reducing the cost of the calculation and pushing the "Berlin Wall" closer to the physical point.

In addition this algorithm is the central idea in developing the new algorithms described in the following that achieve even better performance. It has been realized that the fermion force itself can be split up in two pieces. One coming from the UV (the large eigenvalue part of the spectrum) and one coming from the IR (low end of the spectrum). The idea that the low eigenmodes of fermion matrix are moving slowly in HMC is not new. The reason for that slow movement is the fact the fermion force from the IR is much smaller than the gauge force and the UV fermion force. As a result if a single small time step is used in order to achieve small energy violations coming from the UV part of the spectrum, slow evolution of the IR modes is observed. One other way to understand why the UV part of the spectrum produces such a large force is the simple observation that the UV modes, though each one contributes very little to the force, are the overwhelming majority of modes, resulting a total force larger than the total force coming from the IR. Eventually, the IR force will dominate since it diverges inversely proportionally to the quark mass but this happens at significantly lower quark masses than those currently used. With this in mind, the clear objective for developing a more efficient HMC algorithm is to split up the fermion force in a cheap UV part and an expensive IR part and use the multiple time step update to reduce the total number of the expensive IR force evaluation.
4.3. The mass preconditioning
The first implementation of the above idea is the so-called mass preconditioning [27], which uses a fermion matrix with a large quark mass to approximate the UV part of the spectrum. The UV part of the spectrum is very little dependent on the quark mass hence such approach can be taken. The split up of the fermion force is achieved through the following identity

\[
\det \left( D(U)^{\dagger}D(U) \right) = \det(M(U)^{\dagger}M(U))\det \left( \frac{1}{M(U)^{\dagger}}D(U)^{\dagger}D(U)\frac{1}{M(U)} \right)
\]  

(26)

where \(M(U)\) is a fermion matrix just like \(D(U)\) but with heavier quark mass. This equation is the basic starting point for all the algorithms described in the following. One can now use two bosonic fields to exponentiate the two determinants. The force resulting from \(M(U)\) is cheap and large while the second determinant results a small and expensive force. The choice of heavy mass can be optimized to achieve the best performance. The resulting speed up is close to a factor of 10 in certain situations [21].

4.4. The polynomial filtering
In this algorithm the split up of the force in UV and IR is taken more literally. The matrix \([M(U)^{\dagger}M(U)]^{-1}\) of Eq. 26 is taken to be a polynomial approximation to the inverse of \(D(U)^{\dagger}D(U)\) in the UV part of its spectrum i.e. in the range \([\mu, 1]\) if \(D(U)^{\dagger}D(U)\) is normalized such that the maximum eigenvalue is one. This can be done since the maximum eigenvalue of \(D(U)^{\dagger}D(U)\) is analytically known. In other words we rewrite Eq. 26 as

\[
\det \left( D(U)^{\dagger}D(U) \right) = \det(\Pi(D(U)^{\dagger}D(U)))\det \left( D(U)^{\dagger}D(U)\frac{1}{\Pi(D(U)^{\dagger}D(U))} \right)
\]  

(27)

where \(\Pi(x) \approx 1/x\) in the interval \([\mu, 1]\). This method introduced and tested by [28, 29] using Chebyshev polynomials. Since \(\mu\) turns out not to be very small, a very low degree polynomial is needed. All methods introduced in the Polynomial HMC algorithm are also applicable here. The first evidence they have indicates that this algorithm works very well.

4.5. Domain decomposition
One other choice for the preconditioner matrix \(M(U)\) is the so-called Schwarz preconditioner introduced in [30]. In this approach the lattice is decomposed in domains (see Fig. 2) and thus
the original fermion matrix $D(U)$ can be written as

$$D(U) = \begin{pmatrix} D_{\Omega} & D_{\Omega'\Omega} \\ D_{\Omega'\Omega} & D_{\Omega'} \end{pmatrix}$$  

(28)

where the dependence on the gauge field is suppressed and $\Omega$ and $\Omega'$ denote the two types of domains the lattice is split up to. For simple ultra-local fermion actions (containing couplings to nearest neighbors only), $\partial\Omega$ and $\partial\Omega'$ are just the boundaries of the domains. In this notation the preconditioning matrix $M(U)$ is

$$M(U) = \begin{pmatrix} D_{\Omega} & 0 \\ 0 & D_{\Omega'} \end{pmatrix}$$  

(29)

As it was shown in [30] the second determinant of Eq. 26 can be significantly simplified requiring introduction of bosonic fields that leave only on the interface of the domains. Again the preconditioner matrix describes the UV physics while the correction term describes the IR physics which is expensive but can be put in a large time scale significantly reducing the cost of the computation. The overall efficiency of this approach is demonstrated [30] to be about a factor of 10 better than the original HMC formulation at the light quark masses and lattice spacings used. This formulation is now used by at least a couple of groups to compute in lattice QCD with Wilson fermions.

5. Conclusions

In this paper I have outlined the basic algorithms used in lattice QCD calculations and reviewed some new developments that allow us today to perform computations that are of great phenomenological interest. The future of lattice QCD in helping understand the physics of basic building blocks of matter is bright. The development of better computer architectures is playing a significant role. However, it is clear that with out the significant developments in both the algorithms which I outlined, and the formulations of lattice QCD, which I did not cover here, there results we are obtaining today would not have being possible.

In order to move forward in improving precision, better controlling our systematic errors and become able to ask new questions, we need a steady stream of progress in all these three directions: algorithms, lattice formulation improvements and computer hardware.

References

[1] H. David Politzer. Reliable perturbative results for strong interactions? Phys. Rev. Lett., 30:1346–1349, 1973.
[2] D. J. Gross and Frank Wilczek. Ultraviolet behavior of non-abelian gauge theories. Phys. Rev. Lett., 30:1343–1346, 1973.
[3] Frithjof Karsch. Lattice qcd at high temperature and the qgp. 2006.
[4] C. Bernard et al. Qcd thermodynamics with three flavors of improved staggered quarks. Phys. Rev., D71:034504, 2005.
[5] C. Aubin et al. Properties of light quarks from lattice qcd simulations. J. Phys. Conf. Ser., 16:160–164, 2005.
[6] A. S. Kronfeld et al. Predictions from lattice qcd. PoS, LAT2005:206, 2006.
[7] C. T. H. Davies et al. High-precision lattice qcd confronts experiment. Phys. Rev. Lett., 92:022001, 2004.
[8] Kenneth G. Wilson. Confinement of quarks. Phys. Rev., D10:2445–2459, 1974.
[9] John B. Kogut and Leonard Susskind. Hamiltonian formulation of wilson’s lattice gauge theories. Phys. Rev., D11:395, 1975.
[10] Steven A. Gottlieb, W. Liu, D. Toussaint, R. L. Renken, and R. L. Sugar. Hybrid molecular dynamics algorithms for the numerical simulation of quantum chromodynamics. Phys. Rev., D35:2531–2542, 1987.
[11] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. Phys. Lett., B195:216–222, 1987.
[12] Tetsuya Takaishi and Philippe de Forcrand. Testing and tuning new symplectic integrators for hybrid monte carlo algorithm in lattice qcd. *Phys. Rev.*, E73:036706, 2006.
[13] P. de Forcrand and A. Galli. Exact local bosonic algorithm for full qcd. 1996.
[14] Roberto Frezzotti and Karl Jansen. A polynomial hybrid monte carlo algorithm. *Phys. Lett.*, B402:328–334, 1997.
[15] Roberto Frezzotti and Karl Jansen. The phmc algorithm for simulations of dynamical fermions. i: Description and properties. *Nucl. Phys.*, B555:395–431, 1999.
[16] Roberto Frezzotti and Karl Jansen. The phmc algorithm for simulations of dynamical fermions. ii: Performance analysis. *Nucl. Phys.*, B555:432–453, 1999.
[17] M. A. Clark, A. D. Kennedy, and Z. Sroczynski. Exact 2+1 flavour rhmc simulations. *Nucl. Phys. Proc. Suppl.*, 140:835–837, 2005.
[18] Martin Luscher. A new approach to the problem of dynamical quarks in numerical simulations of lattice qcd. *Nucl. Phys.*, B418:637–648, 1994.
[19] C. Bernard et al. Panel discussion on the cost of dynamical quark simulations. *Nucl. Phys. Proc. Suppl.*, 106:199–205, 2002.
[20] A. Ukawa. Computational cost of full qcd simulations experienced by cp-pacs and jlqcd collaborations. *Nucl. Phys. Proc. Suppl.*, 106:195–196, 2002.
[21] C. Urbach, K. Jansen, A. Shindler, and U. Wenger. Hmc algorithm with multiple time scale integration and mass preconditioning. *Comput. Phys. Commun.*, 174:87–98, 2006.
[22] M. A. Clark and A. D. Kennedy. The rhmc algorithm for 2 flavors of dynamical staggered fermions. *Nucl. Phys. Proc. Suppl.*, 120:850–852, 2004.
[23] Karl Jansen, Andrea Shindler, Carsten Urbach, and Urs Wenger. Hmc algorithm with multiple time scale integration and mass preconditioning. PoS, LAT2005:118, 2006.
[24] Thomas Chiarappa, Roberto Frezzotti, and Carsten Urbach. A (p)hmc algorithm for n(f) = 2+1+1 flavours of twisted mass fermions. PoS, LAT2005:103, 2006.
[25] R. C. Brower, T. Ivanenko, A. R. Levi, and K. N. Orginos. Chronological inversion method for the dirac matrix in hybrid monte carlo. *Nucl. Phys.*, B484:353–374, 1997.
[26] D. H. Weingarten and J. C. Sexton. Hamiltonian evolution for the hybrid monte carlo algorithm. *Nucl. Phys. Proc. Suppl.*, 26:613–616, 1992.
[27] Martin Hasenbusch. Speeding up the hybrid-monte-carlo algorithm for dynamical fermions. *Phys. Lett.*, B519:177–182, 2001.
[28] Mike J. Peardon and James Sexton. Multiple molecular dynamics time-scales in hybrid monte carlo fermion simulations. *Nucl. Phys. Proc. Suppl.*, 119:985–987, 2003.
[29] W. Kamleh and M. J. Peardon. Polynomial filtering for hmc in lattice qcd. PoS, LAT2005:106, 2006.
[30] Martin Luscher. Schwarz-preconditioned hmc algorithm for two-flavour lattice qcd. *Comput. Phys. Commun.*, 165:199–220, 2005.