# Comment on the PHMC algorithm

Martin Lüscher December 2002

## 1. Introduction

Let us consider a formulation of lattice QCD with a doublet of quarks of equal mass $m$ and let us denote by $D_m$ the massive lattice Dirac operator. We may then define the hermitian operator

$$Q = c\gamma_5 D_m \tag{1.1}$$

and fix the constant $c$ such that $\|Q\| \leq 1$ for all gauge fields $U(x,\mu)$.

The standard version of the PHMC algorithm [1–3] is based on the representation of the fermion determinant through a functional integral

$$\det\left\{Q^2 P_{n,\epsilon}(Q^2)\right\} \int \mathrm{D}[\phi^\dagger]\mathrm{D}[\phi]\, \mathrm{e}^{-S_B[U,\phi]}, \qquad S_B[U,\phi] = \big(\phi, P_{n,\epsilon}(Q^2)\phi\big), \tag{1.2}$$

over all complex-valued Dirac fields $\phi$ on the lattice, where $(\phi,\psi)$ denotes the natural scalar product in the space of these fields and $P_{n,\epsilon}(y)$ some polynomial of degree $n$ that approximates the function $1/y$ in the range $\epsilon \leq y \leq 1$. The idea is then to apply the HMC algorithm to the integral over the pseudo-fermion field $\phi$ and the gauge field $U$, while the first factor in eq. (1.2) is treated as a correction.

When this algorithm is used, most of the computer time is spent for the calculation of the force field

$$F_\mu^a(x) = \frac{\mathrm{d}}{\mathrm{d}s} S_B[U_s,\phi]_{\,s=0}, \tag{1.3}$$

$$U_s(y,\nu) \equiv \left\{1 + s\,\delta_{xy}\delta_{\mu\nu}T^a + \mathrm{O}(s^2)\right\} U(y,\nu), \tag{1.4}$$

where $T^a$ is any basis of (anti-hermitian) generators of the gauge group. The field is usually evaluated by decomposing the polynomial $P_{n,\epsilon}(y)$ into its root factors and differentiating one factor after the other. This method is efficient in terms of both memory and speed, but it is also well-known to be numerically delicate (there can be catastrophic significance losses) [3,4].

In the present note an alternative method of evaluation is derived that is equally efficient and that can be shown to be numerically safe. It makes use of some particular properties of the Chebyshev polynomials and does not rely on the root decomposition of the polynomial $P_{n,\epsilon}(y)$.

## 2. Chebyshev polynomials

### *2.1 Definition*

Usually only the Chebyshev polynomials $T_n(z)$ of the first kind are considered, but in the following the second series $U_n(z)$ of polynomials will play an equally important rôle. The polynomials are defined by

$$T_n(\cos\theta) = \cos(n\theta), \qquad U_n(\cos\theta) = \frac{\sin((n+1)\theta)}{\sin\theta}, \tag{2.1}$$

and are related to each other through

$$T_n(z) = U_n(z) - zU_{n-1}(z). \tag{2.2}$$

Both series of polynomials satisfy the same recursion relation,

$$T_{n+1}(z) = 2zT_n(z) - T_{n-1}(z), \tag{2.3}$$

$$U_{n+1}(z) = 2zU_n(z) - U_{n-1}(z), \tag{2.4}$$

for all $n \geq 1$ but with different initial values at $n = 1$.

### *2.2 Generating functions*

In many cases algebraic identities involving the Chebyshev polynomials can be deduced using the generating functions

$$(1-t^2)(1-2tz+t^2)^{-1} = T_0(z) + 2\sum_{k=1}^{\infty} T_k(z)t^k, \tag{2.5}$$

$$(1 - 2tz + t^2)^{-1} = \sum_{k=0}^{\infty} U_k(z) t^k. \tag{2.6}$$

We may, for example, establish the relation

$$T_{n+1}(w) - T_{n+1}(z) = (w - z)\Big\{U_n(w)T_0(z) + 2\sum_{k=1}^{n} U_{n-k}(w)T_k(z)\Big\} \tag{2.7}$$

for any $n \geq 0$ by expanding the product of the generating functions (2.5) and (2.6). Similarly the identity

$$U_n(z) = \sum_{k=0}^{n} P_{n-k}(z)P_k(z) \tag{2.8}$$

is obtained by noting that $(1-2tz+t^2)^{-1/2}$ is a generating function for the Legendre polynomials $P_n(z)$.

*2.3 Clenshaw recursion — first & second kind*

Sums of Chebyshev polynomials such as

$$s = \tfrac{1}{2}a_0 + \sum_{k=1}^{n} a_k T_k(z) \tag{2.9}$$

may be evaluated for any specified value of $z$ using the Clenshaw recursion [7]. In the form in which it is normally quoted, the recursion reads

$$s_0 = a_n, \qquad s_1 = 2zs_0 + a_{n-1}, \tag{2.10}$$

$$s_k = 2zs_{k-1} - s_{k-2} + a_{n-k}, \qquad k = 2, \ldots, n-1, \tag{2.11}$$

and the sum $s$ is then obtained through

$$s = zs_{n-1} - s_{n-2} + \tfrac{1}{2}a_0. \tag{2.12}$$

To establish the correctness of this procedure, it suffices to note that

$$s_k = \sum_{j=0}^{k} a_{n-k+j} U_j(z), \qquad k = 0, \ldots, n-1, \tag{2.13}$$

satisfies the recursion. This is a direct consequence of eq. (2.4). In the last step, eq. (2.12), the identity (2.2) is then applied to convert the result of the recursion to the sum $s$.

There is an alternative form of the Clenshaw recursion that generates the sums

$$\tilde{s}_k = \tfrac{1}{2}a_{n-k} + \sum_{j=1}^{k} a_{n-k+j}T_j(z) \tag{2.14}$$

instead of the sums $s_k$. It is given by

$$\tilde{s}_0 = \tfrac{1}{2}a_n, \qquad \tilde{s}_1 = 2z\tilde{s}_0 + \tfrac{1}{2}a_{n-1}, \tag{2.15}$$

$$\tilde{s}_k = 2z\tilde{s}_{k-1} - \tilde{s}_{k-2} + \tfrac{1}{2}(a_{n-k} - a_{n-k+2}), \qquad k = 2, \ldots, n, \tag{2.16}$$

and the last term that is obtained, $\tilde{s}_n$, is then equal to the desired sum $s$. Apart from the fact that the coefficients on the right-hand sides of the equations are different, this recursion is identical to the standard Clenshaw recursion (2.11). In particular, both are numerically stable in the sense that there is no exponential amplification of rounding errors.

A subtle point to be noted is, however, that the polynomials $U_k(z)$ can be considerably larger than $T_k(z)$. In the vicinity of the points $z = \pm 1$ the ratio of the polynomials is in fact of order $k$. The significance of the result will thus often be reduced by this much in the last step (2.12) of the standard recursion. On the other hand, if the alternative form of the recursion is used, a similar significance loss may occur when the differences $a_{n-k} - a_{n-k+2}$ are calculated. In general there is hence no particular advantage in using one or the other recursion.

An important case where this argumentation does not apply is when the sum $s$ is to be evaluated in single-precision arithmetic while the coefficients $a_k$ are known in double precision. If the differences $a_{n-k} - a_{n-k+2}$ are computed in double precision and rounded to single precision, the second form of the recursion will then usually yield more accurate results than the standard recursion. At least this is so for the polynomials $P_{n,\epsilon}$, $L_k$, $R_k$ and $A_{n,\epsilon}$ discussed later in this note.

## 3. Definition and properties of $P_{n,\epsilon}(y)$

The techniques that will be discussed later for the calculation of the force $F^a_\mu(x)$ are expected to apply for any decent choice of the polynomial $P_{n,\epsilon}(y)$. To be completely explicit we shall, however, consider a particular polynomial in this note that has previously been employed in the context of the multi-boson [5,6] and the PHMC [2,3] algorithms.

*3.1 Definition*

We first map the range $\epsilon \le y \le 1$ to the interval $[-1, 1]$ by introducing the variable

$$z = (2y - 1 - \epsilon)/(1 - \epsilon). \tag{3.1}$$

It is then also convenient to set

$$w = -(1 + \epsilon)/(1 - \epsilon) \equiv -\cosh\omega, \quad \omega > 0, \tag{3.2}$$

which is the value of $z$ at $y = 0$. The division by $y$ on the right-hand side of

$$P_{n,\epsilon}(y) = \frac{T_{n+1}(w) - T_{n+1}(z)}{yT_{n+1}(w)} \tag{3.3}$$

thus leaves no remainder and $P_{n,\epsilon}(y)$ is hence a well-defined polynomial in $y$ of degree $n$.

*3.2 Approximation property*

In the range $\epsilon \le y \le 1$ this polynomial approximates the function $1/y$ with a uniform relative error equal to

$$\delta = \frac{1}{|T_{n+1}(w)|}. \tag{3.4}$$

For $y \in [0, \epsilon)$ and large $n$, the polynomial converges monotonically to $1/y$ from below with a gradually reduced rate of convergence. In particular, its value at $y = 0$ is

$$P_{n,\epsilon}(0) = -\frac{2(n+1)}{1-\epsilon} \cdot \frac{U_n(w)}{T_{n+1}(w)}. \tag{3.5}$$

In the cases of interest where $\epsilon \ll 1$ and $n\omega \gg 1$ we have

$$T_{n+1}(w) \simeq \tfrac{1}{2}(-1)^{n+1}\mathrm{e}^{(n+1)\omega}, \qquad U_n(w) \simeq \tfrac{1}{2}(-1)^n \frac{\mathrm{e}^{(n+1)\omega}}{\sinh\omega}, \tag{3.6}$$

up to exponentially small corrections. The approximation error is thus exponentially decreasing while $P_{n,\epsilon}(0)$ is roughly equal to $n\omega/2\epsilon$.

*3.3 Chebyshev series*

From the definition (3.3) and the identity (2.7) it is now immediate that

$$P_{n,\epsilon}(y) = \tfrac{1}{2}a_0 + \sum_{k=1}^{n} a_k T_k(z), \tag{3.7}$$

$$a_0 = -\frac{4}{1-\epsilon}\cdot\frac{U_n(w)}{T_{n+1}(w)}, \qquad a_k = a_0\frac{U_{n-k}(w)}{U_n(w)} \quad (k = 1,\ldots,n). \tag{3.8}$$

Note that the coefficients $a_k$ have alternating sign and are approximately exponentially decreasing in magnitude with exponent $\omega$. The series can thus be safely evaluated using the Clenshaw recursion at any value of $y$ in the interval $[0,1]$ (and not only for $\epsilon \le y \le 1$).

## 4. Explicit formulae for the force $F^a_\mu(x)$

When the right-hand side of eq. (1.3) is evaluated, a series of terms is generated that are linear expressions in the operator

$$\dot{Q}^a_\mu(x) = \left\{\frac{\mathrm{d}}{\mathrm{d}s}Q_{U\to U_s}\right\}_{s=0}. \tag{4.1}$$

The series can be rearranged in many different ways and we now derive some particular representations in terms of the Chebyshev polynomials.

*4.1 Basic representation*

If we substitute $y \to Q^2$ in the definition (3.1) of the variable $z$, the polynomials $T_k(z)$ and $U_k(z)$ become hermitian operators that depend on the gauge field through $Q$. In particular, they can be differentiated which yields a sum of expressions of the form "polynomial in $Q$ times $\dot{Q}$ times another polynomial" (for simplicity the indices $a$, $\mu$ and the argument $x$ are now often suppressed).

Using the generating functions (2.5) and (2.6), it is not difficult to work this out. In the case of the polynomials $T_k(z)$ we have

$$\sum_{k=1}^{\infty} \dot{T}_k(z) t^k = (1 - 2tz + t^2)^{-1} t\dot{z}(1 - t^2)(1 - 2tz + t^2)^{-1}, \tag{4.2}$$

and after expanding the right-hand side of this equation we conclude that

$$\dot{T}_k(z) = U_{k-1}(z)\dot{z}T_0(z) + 2\sum_{j=1}^{k-1} U_{k-j-1}(z)\dot{z}T_j(z) \tag{4.3}$$

for all $k \geq 1$. An explicit formula for the force is thus given by

$$F_\mu^a(x) = \sum_{k=1}^{n} a_k \left\{ (U_{k-1}(z)\phi, \dot{z}T_0(z)\phi) + 2\sum_{j=1}^{k-1} (U_{k-1-j}(z)\phi, \dot{z}T_j(z)\phi) \right\}, \tag{4.4}$$

where use has been made of the fact that $Q$ is hermitian and that the Chebyshev polynomials are real.

*4.2 Exploiting hermiticity*

If we now insert

$$\dot{z} = \frac{2}{1 - \epsilon}\big(Q\dot{Q} + \dot{Q}Q\big) \tag{4.5}$$

in this expression, the number of terms appears to be doubled since the polynomials on the left and the right in the scalar products are not the same. As it turns out, half of the terms are precisely the complex-conjugate of the other half, and the number of terms that need to be computed is hence only $n$.

To show this we first note that

$$\big(U_{k-1}(z)\phi, \dot{Q}QT_0(z)\phi\big) + 2\sum_{j=1}^{k-1} \big(U_{k-1-j}(z)\phi, \dot{Q}QT_j(z)\phi\big) =$$

$$\big(T_0(z)\phi, \dot{Q}QU_{k-1}(z)\phi\big) + 2\sum_{j=1}^{k-1} \big(T_j(z)\phi, \dot{Q}QU_{k-1-j}(z)\phi\big). \tag{4.6}$$

This identity is easily established by multiplication with $t^k$ and summing from $k = 1$ to infinity. The equation is then seen to be equivalent to

$$(1 - 2tz + t^2)^{-1} t\dot{Q}Q(1 - t^2)(1 - 2tz + t^2)^{-1} =$$

$$(1 - t^2)(1 - 2tz + t^2)^{-1} t\dot{Q}Q(1 - 2tz + t^2)^{-1} \tag{4.7}$$

which is obviously satisfied.

Since all operators are hermitian, it now follows that

$$F^a_\mu(x) = \frac{4}{1-\epsilon} \sum_{k=1}^{n} a_k \Big\{ \mathrm{Re}\left(QU_{k-1}(z)\phi, \dot{Q}T_0(z)\phi\right) + 2\sum_{j=1}^{k-1} \mathrm{Re}\left(QU_{k-1-j}(z)\phi, \dot{Q}T_j(z)\phi\right) \Big\}. \tag{4.8}$$

This formula would already be suitable for numerical evaluation, but there is an even better representation where the number of applications of $Q$ that need to be performed is reduced to its absolute minimum.

*4.3 Improved formula*

To derive this representation, we introduce another set $b_1, \ldots, b_n$ of coefficients in such a way that

$$a_{n-k} = \tfrac{1}{4}(1-\epsilon)(-1)^n \sum_{l=0}^{k} b_{n-l} b_{n-k+l} \quad \text{for all} \quad k = 0, \ldots, n-1. \tag{4.9}$$

In the present case these equations assume the form

$$U_k(w) = \tfrac{1}{16}(1-\epsilon)^2(-1)^{n+1} T_{n+1}(w) \sum_{l=0}^{k} b_{n-l} b_{n-k+l}, \tag{4.10}$$

and from eq. (2.8) we then infer that

$$b_n = \frac{4}{1-\epsilon} \left|T_{n+1}(w)\right|^{-1/2}, \qquad b_l = b_n P_{n-l}(w). \tag{4.11}$$

It is easy to show that this is the unique solution of eq. (4.9) with $b_n > 0$.

Next we insert (4.9) in (4.8) and rearrange the summations over the indices $k, j$ and $l$ in a particular way. As a result the representation

$$F_\mu^a(x) = (-1)^n \sum_{k=1}^{n} \mathrm{Re}\left(QL_{k-1}(z)\phi, \dot{Q}_\mu^a(x) R_{n-k}(z)\phi\right) \tag{4.12}$$

is obtained, where the "left" and "right" polynomials are given by

$$L_k(z) = \sum_{j=0}^{k} b_{n-k+j} U_j(z), \tag{4.13}$$

$$R_k(z) = b_{n-k} T_0(z) + 2 \sum_{j=1}^{k} b_{n-k+j} T_j(z) \tag{4.14}$$

$(k = 0, \ldots, n-1)$. As explained below, these polynomials can be calculated simultaneously using the Clenshaw recursion, thus saving half of the work that would be required for the evaluation of the force via the representation (4.8).

## 5. Asymptotic analysis

Using a well-known integral representation of the Legendre polynomials we have

$$b_l = (-1)^{n-l} b_n \int_{-\pi}^{\pi} \frac{\mathrm{d}\alpha}{2\pi} \left(\cosh\omega + \sinh\omega \cos\alpha\right)^{n-l}. \tag{5.1}$$

It follows from this that the coefficients have alternating sign and decreasing magnitude. If $(n-l)\omega$ is large, the integral can be expanded about the saddle point at $\alpha = 0$ which yields the asymptotic expression

$$b_l \sim (-1)^{n-l} b_n \frac{\mathrm{e}^{(n-l+\frac{1}{2})\omega}}{\sqrt{2\pi(n-l)\sinh\omega}}. \tag{5.2}$$

The ratio $|b_l/b_1|$ thus falls roughly exponentially with exponent $\omega$ from 1 to a value at $l = n$ of about $(2\pi n\omega)^{1/2}\,\mathrm{e}^{-n\omega}$.

When eq. (4.11) is inserted in the definition (4.13), the series

$$L_k(z) = b_n \sum_{j=0}^{k} P_{k-j}(w) U_j(z) \tag{5.3}$$

is obtained, which shows that these polynomials depend on $n$ through the normalization factor $b_n$ only. Their behaviour at large $k\omega$ can be worked out by summing the geometric series in

$$L_k(z) = b_n \int_{-\pi}^{\pi} \frac{\mathrm{d}\alpha}{2\pi} \sum_{j=0}^{k} (-1)^{k-j} \left(\cosh\omega + \sinh\omega\cos\alpha\right)^{k-j} \frac{\sin((j+1)\theta)}{\sin\theta} \tag{5.4}$$

and evaluating the integral at the saddle point $\alpha = 0$. All these remarks also apply (with the obvious modifications) to the second series of polynomials. For the leading terms the calculation yields

$$L_k(z) \sim -\tfrac{1}{4}(1-\epsilon)\, b_{n-k-1}/y, \qquad R_k(z) \sim \tfrac{1}{2}(1-\epsilon)\sinh\omega\, b_{n-k}/y, \tag{5.5}$$

and both polynomials are thus proportional to $1/y$ when $k\omega \gg 1$.

We may conclude from this that the sum (4.12) is approximately of the form

$$(-1)^{n+1}\tfrac{1}{8}(1-\epsilon)^2 \sinh\omega \sum_k b_{n-k} b_k \,\mathrm{Re}\left((Q/y)\phi, \dot{Q}^b_\mu(x)(1/y)\phi\right) \tag{5.6}$$

except at the boundaries of the summation range where either $k\omega$ or $(n-k)\omega$ is not large. In particular, each term makes a contribution of the same sign and there are, therefore, no important cancellations. Note that

$$(-1)^{n+1}\tfrac{1}{8}(1-\epsilon)^2 \sinh\omega\, b_{n-k} b_k \sim -\frac{2}{\pi}\left[k(n-k)\right]^{-1/2}, \tag{5.7}$$

$$\sum_{k=1}^{n-1} \left[k(n-k)\right]^{-1/2} \underset{n\to\infty}{=} \int_0^1 \mathrm{d}x\, \left[x(1-x)\right]^{-1/2} = \pi. \tag{5.8}$$

These results are in nice agreement with the expectation that the sum (4.12) should be an approximation of $-2\,\mathrm{Re}\left(Q^{-2}\phi, Q\dot{Q}^b_\mu(x)Q^{-2}\phi\right)$ for sufficiently large $n$.

## 6. Numerical procedure

We now describe in some detail how to evaluate the representation (4.12) of the force $F^a_\mu(x)$ on a computer.

*6.1 Computation of the coefficients $b_l$*

To calculate the coefficients we may use the backward recursion

$$b_{n-k} = \frac{1-2k}{k}\cosh\omega\, b_{n-k+1} + \frac{1-k}{k} b_{n-k+2}, \qquad k = 2, \ldots, n-1, \tag{6.1}$$

starting from the initial values

$$b_n = \frac{4}{1-\epsilon}\left[\cosh((n+1)\omega)\right]^{-1/2}, \qquad b_{n-1} = -\cosh\omega\, b_n. \tag{6.2}$$

This method is numerically safe since the coefficients $b_{n-k}$ are increasing with $k$ while the other independent solution of the recursion, the Legendre functions of the second kind, decreases exponentially.

*6.2 Application of the Clenshaw recursion*

Let us now consider the fields

$$\psi_k = L_k(z)\phi, \qquad \chi_k = R_k(z)\phi. \tag{6.3}$$

Using the second form of the Clenshaw recursion we have

$$\chi_0 = b_n\phi, \qquad \chi_1 = 2z\chi_0 + b_{n-1}\phi, \tag{6.4}$$

$$\chi_k = 2z\chi_{k-1} - \chi_{k-2} + (b_{n-k} - b_{n-k+2})\phi \quad (k = 2, \ldots, n-1), \tag{6.5}$$

and the other fields are then given by

$$\psi_0 = \chi_0, \qquad \psi_1 = \chi_1, \tag{6.6}$$

$$\psi_k = \psi_{k-2} + \chi_k \quad (k = 2, \ldots, n-1). \tag{6.7}$$

The fields that are saved in the course of the recursion are $\chi_k$ and $Q\psi_k$, $0 \le k < n$, since these are the ones that are needed for the calculation of the matrix elements

in eq. (4.12). Note that the latter can be directly obtained through

$$Q\psi_0 = Q\chi_0, \qquad Q\psi_1 = Q\chi_1, \tag{6.8}$$

$$Q\psi_k = Q\psi_{k-2} + Q\chi_k. \tag{6.9}$$

The additional work is miminal here, since $Q\chi_k$ has to be computed anyway when $\chi_{k+1}$ is calculated.

*6.3 Operation count and memory usage*

It is obvious that in this way the total number of applications of $Q$ that is required is equal to $2n-1$. In addition about $4n$ real linear combinations of the type $\phi_1 \to r_1\phi_1 + r_2\phi_2$ must be formed.

When implemented straigthforwardly, this computation needs storage space for $2n$ fields. We may, however, start the computation of the force as soon as the recursion reaches $k = n/2$. It is then possible to progressively overwrite the fields that have already been used so that storage space for only $n$ fields needs to be allocated.

Some additional space is required for the force $F^a_\mu(x)$ and perhaps a set of auxiliary arrays (such as those related to the Pauli term if O($a$) improved Wilson fermions are used). As discussed in ref. [3], it is also possible to trade memory space for more computational work.

*6.4 Numerical stability*

Since the coefficients $b_l$ are alternating in sign and decreasing in magnitude, the Clenshaw recursion is numerically safe in the whole range $0 \le y \le 1$. Note that all intermediate results are within reasonable limits, for any realistic choice of $n$ and $\epsilon$, so that over- or underflow conditions will never be raised.

If standard 32 bit arithmetic is used for the computation of the fields $\chi_k$ and $Q\psi_k$, it is advisable to accumulate the sum (4.12) in double precision since the terms are of the same sign and have similar size. In this way any numerical errors should be pushed to the lower digits because the sum is significantly larger than the individual terms, i.e. the relative accuracy of the final result is improved. Evidently the differences $b_{n-k} - b_{n-k+2}$ of the coefficients that appear in eq. (6.5) should be computed in double precision (cf. subsect. 2.3).

## 7. Square root of $P_{n,\epsilon}(y)$

In the global heatbath step, before the molecular dynamics equations are solved, the PHMC algorithm makes use of the factorization

$$P_{n,\epsilon}(y) = |A_{n,\epsilon}(z)|^2 \,, \tag{7.1}$$

where $A_{n,\epsilon}(z)$ is another (possibly complex) polynomial in $z$ [2,3]. Evidently such a factorization can only exist if $n$ is even, and we thus set

$$n = 2r, \qquad r \in \mathbb{Z}, \qquad r \geq 1, \tag{7.2}$$

in the following.

To define the new polynomial explicitly, we start from the decomposition

$$P_{n,\epsilon}(y) = 2^{n-1} a_n \prod_{k=1}^{n} (z - z_k), \tag{7.3}$$

$$z_k = -\cos(i\omega + 2k\nu), \qquad \nu \equiv \frac{\pi}{n+1}. \tag{7.4}$$

The roots in this formula occur in complex conjugate pairs, $(z_k)^* = z_{n+1-k}$, and any polynomial $A_{n,\epsilon}(z)$ that satisfies eq. (7.1) is thus obtained by selecting one of the roots in each pair. For numerical purposes it is, however, not advisable to use this representation. Instead the coefficients $c_k$ in the expansion

$$A_{n,\epsilon}(z) = \tfrac{1}{2} c_0 + \sum_{k=1}^{r} c_k T_k(z) \tag{7.5}$$

should first be determined and after that one may apply the Clenshaw recursion to evaluate the polynomial.

When selecting the roots, one should be careful to avoid that $A_{n,\epsilon}(z)$ will have a rapidly oscillating phase. A good choice in this respect is the symmetric product

$$A_{n,\epsilon}(z) = i\,(-1)^{r+1} 2^{r-\frac{1}{2}} \sqrt{a_n} \prod_{k=1}^{r} (z - z_{2k-1}) \tag{7.6}$$

whose asymptotic behaviour

$$A_{n,\epsilon}(z) \underset{n\to\infty}{=} \frac{1}{\sqrt{y}} \left\{ 1 - \frac{i\pi}{2(n+1)} \cdot \frac{\sinh\omega}{z + \cosh\omega} + \mathrm{O}(n^{-2}) \right\} \tag{7.7}$$

can be worked out using the Euler summation formula.

The computation of the coefficients $c_k$ for this particular polynomial is discussed in appendix A. It is shown there that they may be obtained recursively through

$$c_{r+1} = 0, \qquad c_r = i\,(-1)^{r+1}\sqrt{2a_n}, \tag{7.8}$$

$$\sin((2k+1)\nu)\,c_{k+1} + 2\sin(2k\nu)\cosh(\omega + i\nu)\,c_k + \sin((2k-1)\nu)\,c_{k-1} = 0, \tag{7.9}$$

where $k$ runs from $r$ to 1. An important point to note here is that the sine factors in the three terms of eq. (7.9) are practically the same when $n$ is large. The recursion is thus rather similar to the one satisfied by the Chebyshev polynomials, and it may thus be expected to be stable.

Numerical studies confirm this and moreover demonstrate that the coefficients $c_k$ are approximately exponentially decreasing in magnitude. Actually this is the case for both the real and the imaginary parts separately, and they also have oscillating sign. The series (7.5) can thus be accurately evaluated for all $0 \le y \le 1$ using the Clenshaw recursion.

## 8. Other polynomials

Most formulae in this note refer to a particular choice of the polynomial $P_{n,\epsilon}(y)$. In practice other polynomials may be considered for special purposes. The procedures that have been described are, however, of a general nature and are expected to be applicable in all these cases too.

First note that the chosen polynomial can always be expanded in a series of the form (3.7). While the coefficients $a_k$ in this expansion may not be analytically calculable, it is certainly possible to compute them numerically with sufficient precision, perhaps using an algebraic manipulation program. Once they are known, the coefficients $b_n, b_{n-1}, \ldots, b_1$ may be obtained (in this order) by solving eq. (4.9) recursively. The representation (4.12) can then be used to evaluate the force $F_\mu^a(x)$ as before.

If a factorization of the form (7.1) is needed, there is probably no other way than to select a suitable subset of the roots of $P_{n,\epsilon}(y)$. After that the coefficients $c_k$ in the expansion (7.5) may be determined via the linear equations

$$\tfrac{1}{2}c_0 + \sum_{k=1}^{r-1} c_k T_k(w_l) = -c_r T_r(w_l), \qquad l = 1, \ldots, r, \tag{8.1}$$

for example, where $w_l$ are the chosen roots and the coefficient $c_r$ of the highest-degree polynomial is assumed to be known at this point.

## 9. Related earlier work

The variation of a polynomial of the Wilson–Dirac operator has previously been discussed by Liu [8] in his paper on the simulation of lattice QCD with Ginsparg–Wilson fermions. In this case the polynomial approximates the sign function of the hermitian Dirac operator and is expressed as a series of Legendre polynomials. Liu then derives a fairly complicated formula for its variation that can also be written in terms Legendre polynomials.

In the approach of de Forcrand and Takaishi [1], degenerate complex Chebyshev polynomials are used to approximate the inverse of the Wilson–Dirac operator. The polynomial that appears in the bosonic action is then automatically of the factorized form and its variation can be obtained essentially by applying eq. (4.3) to one of the factors (the other factor does not need to be considered in view of the reality of the expression). All this is explained in a study of the JLQCD collaboration of QCD with $2 + 1$ flavours of dynamical quarks [9]. The recursion used in this paper to evaluate the polynomials is the Horner scheme, which is the appropriate choice for degenerate complex Chebyshev polynomials.

Recently similar methods have also been proposed for the simulation of staggered fermions, where fractional powers of the square of the Dirac operator are approximated by real polynomials that are expanded in Chebyshev polynomials [10]. The force is then again obtained by writing the polynomial as the square of another polynomial and applying eq. (4.3) to one of the factors. All polynomials are evaluated using the standard Clenshaw recursion in this case.

There appears to be no reference to the factorization (4.9) and the associated representation (4.12) in any of these papers. An advantage of this representation is that its numerical stability can be established analytically. On the other hand, for the computation of the force one might also use the factorization discussed in sect. 7 and proceed as in ref. [10].

## Appendix A

To derive eq. (7.9) we introduce the phase factor $g = \mathrm{e}^{2i\nu}$ and set

$$z = \tfrac{1}{2}(u + (1/u)), \qquad u_k = -\mathrm{e}^{-\omega} g^k. \tag{A.1}$$

The root factors then become

$$z - z_k = \tfrac{1}{2}(1/u)(u - u_k)(u - (1/u_k)) \tag{A.2}$$

so that

$$A_{n,\epsilon}(z) \propto u^{-r} \prod_{k=1}^{r} (u - u_{2k-1})(u - (1/u_{2k-1})). \tag{A.3}$$

Since the roots $u_k$ only differ from each other by factors of $g$, it follows from this that the polynomial satisfies

$$g\big\{A_{n,\epsilon}(z)\big\}_{u \to ug^2} = \frac{u - u_{-1}}{u - u_{-2}} \cdot \frac{u - (1/u_0)}{u - (1/u_1)} A_{n,\epsilon}(z) \tag{A.4}$$

which may also be written in the form

$$\begin{aligned} \left[u^2 g - (u_{-1} + (1/u_0))u + g^{-2}\right] \left\{A_{n,\epsilon}(z)\right\}_{u \to ug^2} = \\ \left[u^2 - (u_{-1} + (1/u_0))u + g^{-1}\right] A_{n,\epsilon}(z). \end{aligned} \tag{A.5}$$

Now if we substitute the series

$$A_{n,\epsilon}(z) = \tfrac{1}{2} \sum_{k=-r}^{r} c_k u^k \tag{A.6}$$

in this identity and collect the coefficients of the powers of $u$, a three-term recursion for the coefficients $c_k$ is obtained that reduces to eq. (7.9) after some algebra.

## References


[1] P. de Forcrand, T. Takaishi, Nucl. Phys. B (Proc. Suppl.) 53 (1997) 968
[2] R. Frezzotti, K. Jansen, Phys. Lett. B 402 (1997) 328
[3] R. Frezzotti, K. Jansen, Nucl. Phys. B 555 (1999) 395 and 432
[4] B. Bunk, S. Elser, R. Frezzotti, K. Jansen, Comput. Phys. Commun. 118 (1999) 95
[5] M. Lüscher, Nucl. Phys. B418 (1994) 637
[6] B. Bunk, K. Jansen, B. Jegerlehner, M. Lüscher, H. Simma, R. Sommer, Nucl. Phys. B (Proc. Suppl.) 42 (1995) 49
[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical recipes in FORTRAN, 2nd ed. (Cambridge University Press, Cambridge, 1992)
[8] C. Liu, Nucl. Phys. B 554 (1999) 313
[9] S. Aoki at al. (JLQCD collab.), Phys. Rev. D65 (2002) 094507
[10] S. Aoki at al. (JLQCD collab.), An exact algorithm for any-flavour lattice QCD with Kogut-Susskind fermion, hep-lat/0208058