Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 189 (2003) 454–462
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Computational methods for UV-suppressed fermions

Artan Boriçi *

Department of Physics, University of Edinburgh, James Clerk Maxwell Building, Office rm. 4408, Mayfield Road, Edinburgh EH9 3JZ, UK

Received 4 September 2002; received in revised form 10 April 2003; accepted 13 April 2003

Abstract

Lattice fermions with suppressed high momentum modes solve the ultraviolet slowing down problem in lattice QCD. This paper describes a stochastic evaluation of the effective action of such fermions. The method is a based on the Lanczos algorithm and it is shown to have the same complexity as in the case of standard fermions.

1. Introduction

There has been recent interest in the so-called 'Ultraviolet slowing down' of fermionic simulations in lattice QCD [7,8,14,15,18]. These studies try to address large fluctuations of the high end modes of the fermion determinant using various algorithms. The goal is to increase the signal-to-noise ratio of the infrared modes and to accelerate fermion simulations as well.

In fact, all the computational effort needed to treat UV-modes by above algorithms can be reduced to zero by suppressing them in the first place [2]. The lattice Dirac operator of this fermion theory is given by

$$D = \frac{\mu}{a}\Gamma_5 \tanh\frac{a\Gamma_5 D_{\mathrm{W/S/o}}}{\mu}, \tag{1.1}$$

where $D_{\mathrm{W/S/o}}$ is the input lattice Dirac operator, a the lattice spacing and $\mu > 0$ is a dimensionless parameter. For Wilson (W) and overlap (o) fermions as the input theory one has $\Gamma_5 = \gamma_5$. For staggered fermions Γ_5 is a diagonal matrix with entries $+1/-1$ on even/odd lattice sites. The theory converges to the input theory in the continuum limit and is local and unitary as shown in detail in [2]. The input theory is also recovered in the limit $\mu \to \infty$. For $\mu \to 0$ one has $D \to \mu$, i.e., a quenched theory.

* Tel.: +44-131-650-5236; fax: +44-131-650-5902.
E-mail address: borici@ph.ed.ac.uk.

0021-9991/$ - see front matter
doi:10.1016/S0021-9991(03)00227-4

Perturbative calculations with this theory are straightforward. To fix the idea I assume in the following Wilson fermions to be input fermions. The inverse fermion propagator is given by

$$\tilde{D}(p) = \frac{\mu}{a}\gamma_5 \tanh\frac{a\tilde{H}_{\mathrm{W}}(p)}{\mu} \tag{1.2}$$

with $p = \{p_v,\ v = 1,\ldots,4\}$ being the four-momentum vector. As usual, gauge fields are parametrized by $su(3)$ elements

$$U(x)_v = \mathrm{e}^{\mathrm{i}agA(x)_v}, \quad A(x)_v \in su(3) \tag{1.3}$$

and the Wilson operator is written as a sum of the free and interaction terms

$$D_{\mathrm{W}} = D^0_{\mathrm{W}} + D^{\mathrm{I}}_{\mathrm{W}}.$$

The splitting of the lattice Dirac operator is written in the same form

$$D = D^0 + D^{\mathrm{I}}, \quad D^0 = \frac{\mu}{a}\gamma_5 \tanh\frac{aH^0_{\mathrm{W}}}{\mu},$$

where the interaction term has to be determined. This can be done by expanding D in terms of a/μ:

$$D = D_{\mathrm{W}}\left[\mathbb{1} + c_1\left(\frac{aH_{\mathrm{W}}}{\mu}\right)^2 + c_2\left(\frac{aH_{\mathrm{W}}}{\mu}\right)^4 + \cdots\right], \tag{1.4}$$

where $c_1, c_2, \ldots$ are real expansion coefficients. Calculation of D^{I} is an easy task if one stays with a finite number of terms in the right hand side of (1.4). Also, the number of terms can be minimized using a Chebyshev approximation for the hyperbolic tangent. [1]

In this paper I describe computational methods needed to evaluate the effective action of the theory defined above. In particular, the complexity of the proposed Lanczos method does not depend on the input sparse matrix that describes a fermion theory on the lattice.

In the following Section 1 derive a class of Lanczos based methods for computations with the proposed theory and then in Section 3 conclusions follow.

2. Lanczos based methods for computations with fermions

The effective action of the theory defined above can be written as

$$S_{\mathrm{eff}} = \mathrm{tr} f(A), \tag{2.1}$$

where $A \in \mathbb{C}^{N\times N}$ and $f(s)$ is a real and smooth function of $s \in \mathbb{R}^+$. The matrix A is assumed to be Hermitian and positive definite. Since the trace is difficult to obtain one can use the stochastic method of Bai et al. [1]. The method bounds bilinear forms of the type

[1] I would like to thank Joachim Hein for discussions related to lattice perturbation theory.

$$\mathscr{F}(b,A) = b^{\mathrm{T}} f(A) b, \tag{2.2}$$

where $b \in \mathbb{R}^N$ is a random vector. By evaluating lower and upper bounds of $\mathscr{F}(b,A)$ for many such vectors b, authors of Ref. [1] compute a confidence interval for the $\operatorname{tr} f(A)$.

The method described here is similar to the method of Bai et al. [1]. Its viability for lattice QCD computations has been demonstrated in the recent work of Cahill et al. [6]. Bai et al. [1] derive their method using quadrature rules and Lanczos polynomials. Here, I give an alternative derivation which uses familiar tools (in lattice simulations) such as sparse matrix inversions and Padé approximations. The Lanczos method enters the derivation as an algorithm for solving linear systems of the form

$$Ax = b, \quad x \in \mathbb{C}^N. \tag{2.3}$$

2.1. Lanczos algorithm

I follow standard texts as Golub and Van Loan [11] and notations and arguments of Boriçi [3–5]. n steps of the Lanczos algorithm [17] on the pair (A,b) are given by Algorithm 1.

Algorithm 1 The Lanczos algorithm

Set $\beta_0 = 0,\ q_0 = o,\ q_1 = b/\|b\|^2$
for $i = 1,\ldots,n$ **do**
  $v = Aq_i$
  $\alpha_i = q_i^{\dagger} v$
  $v := v - q_i\alpha_i - q_{i-1}\beta_{i-1}$
  $\beta_i = \|v\|_2$
  $q_{i+1} = v/\beta_i$
end for

The Lanczos vectors $q_1,\ldots,q_n \in \mathbb{C}^N$ can be compactly denoted by the matrix $Q_n = [q_1,\ldots,q_n]$. They are a basis of the Krylov subspace $\mathscr{K}_n = \operatorname{span}\{b, Ab, \ldots, A^{n-1}b\}$. It can be shown that the following identity holds:

$$AQ_n = Q_n T_n + \beta_n q_{n+1} e_n^{\mathrm{T}}, q_1 = b/\|b\|_2. \tag{2.4}$$

e_n is the last column of the identity matrix $\mathbb{1}_n \in \mathbb{R}^{n\times n}$ and T_n is the tridiagonal and symmetric Lanczos matrix (2.5) given by

$$T_n = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{pmatrix}. \tag{2.5}$$

The matrix (2.5) is usually referred to as the Lanczos matrix. Its eigenvalues, the so called Ritz values, tend to approximate the extreme eigenvalues of the original matrix A as n increases.

To solve the linear system (2.3) I seek an approximate solution $x_n \in \mathscr{K}_n$ as a linear combination of the Lanczos vectors

$$x_n = Q_n y_n, y_n \in \mathbb{C}^n, \tag{2.6}$$

and project the linear system (2.3) on to the Krylov subspace $\mathscr{K}_n$:

$$Q_n^\dagger A Q_n y_n = Q_n^\dagger b = Q_n^\dagger q_1 \|b\|_2.$$

Using (2.4) and the orthonormality of Lanczos vectors, I obtain

$$T_n y_n = e_1 \|b\|_2,$$

where e_1 is the first column of the identity matrix $\mathbb{1}_n$. By substituting y_n into (2.6) one obtains the approximate solution

$$x_n = Q_n T_n^{-1} e_1 \|b\|_2. \tag{2.7}$$

2.2. Algorithms for the bilinear form (2.2)

The theoretical framework of the algorithm of Bai et al. [1] can be based on the Padé approximation of the smooth and bounded function $f(\cdot)$ in an interval [13]. Without loss of generality one can assume a diagonal Padé approximation in the interval $s \in (0, 1)$. It can be expressed as a partial fraction expansion. Therefore, one can write

$$f(s) \approx \sum_{k=1}^{m} \frac{c_k}{s + d_k} \tag{2.8}$$

with $c_k \in \mathbb{R}, d_k \geqslant 0,\ k = 1, \ldots, m$. Since the approximation error $\mathrm{O}(s^{2m+1})$ can be made small enough as m increases, it can be assumed that the right-hand side converges to the left-hand side as the number of partial fractions becomes large enough. For the bilinear form I obtain

$$\mathscr{F}(b, A) \approx \sum_{k=1}^{m} b^{\mathrm{T}} \frac{c_k}{A + d_k \mathbb{1}} b. \tag{2.9}$$

A first algorithm can already be written down at this point. Having computed the partial fraction coefficients one can use a multi-shift iterative solver of Freund [9] to evaluate the right hand side (2.9). To see how this works, I solve the shifted linear system

$$(A + d_k \mathbb{1}) x^k = b$$

using the same Krylov subspace $\mathscr{K}_n$. A closer inspection of the Lanczos algorithm, Algorithm 1 suggests that in the presence of the shift d_k I get

$$\alpha_i^k = \alpha_i + d_k,$$

while the rest of the algorithm remains the same. This is the so-called shift-invariance of the Lanczos algorithm. From this property and by repeating the same arguments which led to (2.7), I get

$$x_n^k = Q_n \frac{1}{T_n + d_k \mathbb{1}_n} e_1 \|b\|_2. \tag{2.10}$$

Using the shift-invariance of the Lanczos algorithm I obtain Algorithm 2.

Algorithm 2 The Lanczos algorithm for solving $(A + d_k\mathbb{1})x^k = b$

Set $\beta_0 = 0,\ \rho_1^1 = 1/\|b\|_2,\ q_0 = o,\ q_1 = \rho_1^1 b,\ x_0^k = o,\ \tilde{x}_0^k = o,\ \rho_0^k = 0$
for $i = 1, \ldots$ **do**
  $v = Aq_i$
  $\alpha_i = q_i^{\mathrm{T}} v$
  $v := v - q_i\alpha_i - q_{i-1}\beta_{i-1}$
  $\beta_i = \|v\|_2$
  $q_{i+1} = v/\beta_i$
  for $k = 1, \ldots, m$ **do**
    $\tilde{x}_{i+1}^k = -(\tilde{x}_i^k\alpha_i + \tilde{x}_{i-1}^k\beta_{i-1})/\beta_i$
    $\rho_{i+1}^k = -(\rho_i^k\alpha_i + \rho_{i-1}^k\beta_{i-1})/\beta_i$
    $r_{i+1}^k = q_{i+1}/\rho_{i+1}^k$
    $x_{i+1}^k = \tilde{x}_{i+1}^k/\rho_{i+1}^k$
  end for
  if $1/|\rho_{i+1}^1| < \epsilon$ **then**
    $n = i$
    stop
  end if
end for

Note that the residual errors $r_i^k, i = 1, \ldots, n, k = 1, \ldots, m$ are given by

$$r_i^k = b - Ax_i^k - d_k x_i^k.$$

In exact arithmetic their norm is given by

$$1/\rho_i^k = \left\|b - Ax_i^k - d_k x_i^k\right\|_2. \tag{2.11}$$

By applying Algorithm 2 one can solve the shifted linear systems on the right hand side of (2.9). The algorithm stops if the linear system with the smallest shift is solved to the desired accuracy ϵ. This is a well-known technique [9] which is used also in lattice QCD [10]. However, the problem with this method is that one needs to store a large number of vectors that is proportional to m. This could be prohibitive if m is say larger than 10.

In fact, the right-hand side of (2.9) can be written in terms of solutions $x_n^k, k = 1, \ldots, m$ as a sum of scalars

$$\mathscr{F}(b, A) \approx \sum_{k=1}^{m} c_k w^k, \quad w^k = b^{\mathrm{T}} x^k. \tag{2.12}$$

Therefore, it is easy to replace the vector recurrences by scalar recurrences of the form

$$\tilde{w}_{i+1}^k = -(\tilde{w}_i^k\alpha_i + \tilde{w}_{i-1}^k\beta_{i-1})/\beta_i. \tag{2.13}$$

In this way one obtains the Algorithm 3. It is clear that by applying Algorithm 3 one gains substantial storage savings compared to Algorithm 2. If one has a good Padé approximant for the function $f(\cdot)$ one can apply Algorithm 3.

Algorithm 3 The Lanczos algorithm for computing $w^k, k = 1, \ldots, m$

Set $\beta_0 = 0, \rho_1^1 = 1/\|b\|_2,\ q_0 = o,\ q_1 = \rho_1^1 b,\ w_0^k = o,\ \tilde{w}_0^k = o,\ \rho_0^k = 0$
for $i = 1, \ldots$ **do**
  $v = Aq_i$
  $\alpha_i = q_i^{\mathrm{T}} v$
  $v := v - q_i\alpha_i - q_{i-1}\beta_{i-1}$
  $\beta_i = \|v\|_2$
  $q_{i+1} = v/\beta_i$
  for $k = 1, \ldots, m$ **do**
    $\tilde{w}_{i+1}^k = -(\tilde{w}_i^k\alpha_i + \tilde{w}_{i-1}^k\beta_{i-1})/\beta_i$
    $\rho_{i+1}^k = -(\rho_i^k\alpha_i + \rho_{i-1}^k\beta_{i-1})/\beta_i$
    $w_{i+1}^k = \tilde{w}_{i+1}^k/\rho_{i+1}^k$
  end for
  if $1/|\rho_{i+1}^1| < \epsilon$ **then**
    $n = i$
    stop
  end if
end for

Note that another way to save storage is using the multi-shift Conjugate Gradient variant of [16].

If a Padé approximation is not sufficient or difficult to obtain, the Lanczos method combined with exact evaluation of the function at Ritz values remains the only viable alternative to evaluate the bilinear forms of type (2.2).

To see how this is realized I assume that the linear system (2.3) is solved to the desired accuracy using the Lanczos algorithm, Algorithm 1 and (2.7). In the application considered here one can show that

$$\sum_{k=1}^{m} b^{\mathrm{T}} \frac{c_k}{A + d_k \mathbb{1}} b = \|b\|^2 \sum_{k=1}^{m} e_1^{\mathrm{T}} \frac{c_k}{T_n + d_k \mathbb{1}_n} e_1. \tag{2.14}$$

For the result (2.14) to hold, it is sufficient to show that

$$b^{\mathrm{T}} \frac{c_k}{A + d_k \mathbb{1}} b = \|b\|^2 e_1^{\mathrm{T}} \frac{c_k}{T_n + d_k \mathbb{1}_n} e_1,$$

which can be shown using the orthonormality property of the Lanczos vectors and (2.10). Note however that in presence of roundoff errors the orthogonality of the Lanczos vectors is lost but the result (2.14) is still valid. This issue is beyond the scope of this paper. However, for a detailed survey the reader may consult the work of Golub and Strakos, Cahill et al. [6,12].

From this result and the convergence of the partial fractions to the matrix function $f(\cdot)$, it is clear that

$$\mathscr{F}(b, A) \approx \hat{\mathscr{F}}_n(b, A) = \|b\|^2 e_1^{\mathrm{T}} f(T_n) e_1. \tag{2.15}$$

Note that the evaluation of the right hand side is a much easier task than the evaluation of the right-hand side of (2.2). A straightforward method is the spectral decomposition of the symmetric and tridiagonal matrix T_n:

$$T_n = Z_n \Omega_n Z_n^{\mathrm{T}}, \tag{2.16}$$

where $\Omega_n \in \mathbb{R}^{n\times n}$ is a diagonal matrix of eigenvalues $\omega_1, \ldots, \omega_n$ of T_n and $Z_n \in \mathbb{R}^{n\times n}$ is the corresponding matrix of eigenvectors, i.e., $Z_n = [z_1, \ldots, z_n]$. From (2.15) and (2.16) it is easy to show that (see for example [11])

$$\hat{\mathscr{F}}_n(b,A) = \|b\|^2 e_1^{\mathrm{T}} Z_n f(\Omega_n) Z_n^{\mathrm{T}} e_1, \tag{2.17}$$

where the function $f(.)$ is now evaluated at individual eigenvalues of the tridiagonal matrix T_n.

The eigenvalues and eigenvectors of a symmetric and tridiagonal matrix can be computed by the QL method with implicit shifts [19]. The method has an $O(n^3)$ complexity. Fortunately, one can compute (2.17) with only an $O(n^2)$ complexity. Closer inspection of Eq. (2.17) shows that besides the eigenvalues, only the first elements of the eigenvectors are needed

$$\hat{\mathscr{F}}_n(b,A) = \|b\|^2 \sum_{i=1}^{n} z_{1i}^2 f(\omega_i). \tag{2.18}$$

It is easy to see that the QL method delivers the eigenvalues and first elements of the eigenvectors with $O(n^2)$ complexity. [2]

A similar formula (2.18) is suggested by Bai et al. [1]) based on quadrature rules and Lanczos polynomials. The Algorithm 4 is thus another way to compute the bilinear forms of the type (2.2).

Algorithm 4 The Lanczos algorithm for computing (2.2)

Set $\beta_0 = 0,\ \rho_1 = 1/\|b\|_2,\ q_0 = o,\ q_1 = \rho_1 b$
for $i = 1, \ldots$ **do**
  $v = Aq_i$
  $\alpha_i = q_i^{\dagger} v$
  $v := v - q_i\alpha_i - q_{i-1}\beta_{i-1}$
  $\beta_i = \|v\|_2$
  $q_{i+1} = v/\beta_i$
  $\rho_{i+1} = -(\rho_i\alpha_i + \rho_{i-1}\beta_{i-1})/\beta_i$
  if $1/|\rho_{i+1}| < \epsilon$ **then**
    $n = i$
    stop
  endif
end for
Set $(T_n)_{i,i} = \alpha_i, (T_n)_{i+1,i} = (T_n)_{i,i+1} = \beta_i$, otherwise $(T_n)_{i,j} = 0$
Compute ω_i and z_{1i} by the QL method
Evaluate (2.2) using (2.18)

Clearly, the Lanczos algorithm and Algorithm 3 has an $O(nN)$ complexity, whereas Algorithm 4 has a greater complexity: $O(nN) + O(n^2)$. However, Algorithm 4 delivers an exact evaluation of (2.2). For typical applications in lattice QCD the $O(n/N)$ additional relative overhead is small and therefore Algorithm 4 is the recommended algorithm among all three algorithms presented in this section.

A remark on stopping criteria is also desirable. The method of Bai et al. [1] computes the relative differences of (2.18) between two successive Lanczos steps and stops if they do not decrease below a given accuracy. In order to perform the test their algorithm needs to compute the eigenvalues of T_i at each Lanczos step i. This may be a large computational overhead. The test proposed here is safe since the computation ends when the the underlying liner system is solved to the desired accuracy. However, this may be too demanding since the prime interest here is the computation of the bilinear form (2.2).

To illustrate this situation I give an example from a $12^3 \times 24$ lattice with $\mu = 0.2$, Wilson input operator and a $SU(3)$ gauge field background at bare gauge coupling $\beta = 5.9$. I compute the bilinear form (2.2) for

[2] I thank Alan Irving for the related comment on the QL implementation in [19].

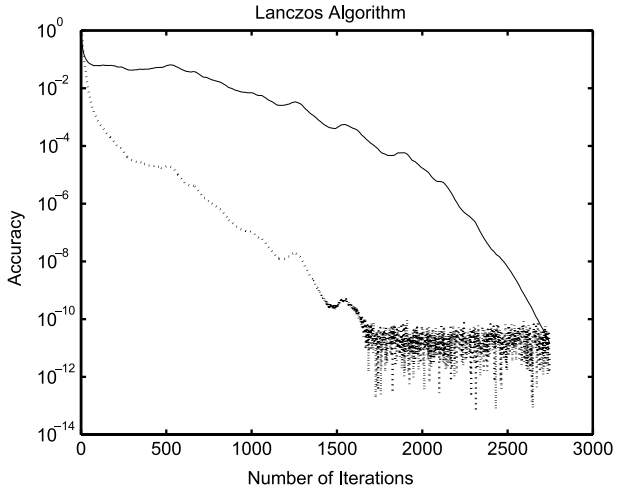

Fig. 1. Normalized recursive residual (solid line) and relative differences of (2.18) (dotted line) produced by Algorithm 4.

$$f(s) = \log \tanh \sqrt{s}, \quad s \in \mathbb{R}^{+} \tag{2.19}$$

and $A = H_{\mathrm{W}}^2$, $b \in \mathbb{C}^N$. The real and imaginary parts of the $b-$elements are chosen randomly from the set $\{+1, -1\}$.

In Fig. 1 are shown the normalized recursive residuals $\rho_0/\rho_i = \|b - Ax_i\|_2/\|b\|_2$, $i = 1, \ldots, n$ and relative differences of (2.18) between two successive Lanczos steps. The figure illustrates clearly the different regimes of convergence for the linear system and the bilinear form. The relative differences of the bilinear form converge faster than the computed recursive residual. This example indicates that a stopping criterion based on the solution of the linear system may indeed be strong and demanding. Therefore, the recommended stopping criteria would be to monitor the relative differences of the bilinear form but less frequently than proposed by Bai et al. [1]. More investigations are needed to settle this issue. Note also the roundoff effects (see Fig. 1) in the convergence of the bilinear form which are a manifestation of the finite precision of the machine arithmetic.

3. Conclusion

In this paper I have described computational methods needed to evaluate the effective action of the theory with suppressed cutoff modes.

All methods described in this paper show that the effective action of UV-suppressed fermions can be computed with at most $\mathrm{O}(Nn) + \mathrm{O}(n^2)$ complexity. This makes a $\mathrm{O}(n^2)$ overhead compared to standard fermions. The additional relative overhead $\mathrm{O}(n/N)$ tends to be small on large lattices. Therefore, it may be concluded that UV-suppressed fermions have similar computational complexity as standard fermions.

Acknowledgements

I would like to thank Philippe de Forcrand and Alan Irving for discussions and useful suggestions at different stages of this work.

References

[1] Z. Bai, M. Fahey, G.H. Golub, Some large-scale matrix computation problems, J. Comput. Appl. Math. 74 (1996) 71–89.
[2] A. Boriçi, Lattice QCD with suppressed high momentum modes of the Dirac operator, hep-lat/0205011.
[3] A. Boriçi, On the Neuberger overlap operator, Phys. Lett. B 453 (1999) 46–53.
[4] A. Boriçi, Fast methods for computing the neuberger operator, in: A. Frommer, et al. (Eds.), Numerical Challenges in Lattice Quantum Chromodynamics, Springer, Heidelberg, 2000.
[5] A. Boriçi, A lanczos approach to the inverse square root of a large and sparse matrix, J. Comput. Phys. 162 (2000) 123–131 (and hep-lat/9910045).
[6] E. Cahill, A. Irving, C. Johnson, J. Sexton, Numerical stability of lanczos methods, Nucl. Phys. Proc. Suppl. 83 (2000) 825–827.
[7] A. Duncan, E. Eichten, H. Thacker, An efficient algorithm for QCD with light dynamical quarks, Phys. Rev. D 59 (1999) 014505.
[8] Ph. de Forcrand, UV-filtered fermionic monte carlo, Nucl. Phys. Proc. Suppl. 73 (1999) 822–824.
[9] R.W. Freund, Solution of shifted linear systems by quasi-minimal residual iterations, in: L. Reichel, A. Ruttan, R.S. Varga (Eds.), Numerical Linear Algebra, W. de Gruyter, 1993.
[10] A. Frommer, S. Güsken, T. Lippert, B. Nöckel, K. Schilling, Many masses on one stroke: economic computation of quark propagators, Int. J. Mod. Phys. C 6 (1995) 627–638, and hep-lat/9504020.
[11] G.H. Golub, C.F. Van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, 1989.
[12] G.H. Golub, Z. Strakos, Estimates in quadratic formulas, Numerical Algorithms 8 (1994) 241–268.
[13] P.R. Graves-Morris, in: L. Wuytack (Ed.), Padé Approximation and Its Applications, Lecture Notes in Mathematics, vol. 765, Springer, Berlin, 1979.
[14] A. Hasenfratz, F. Knechtli, Simulating dynamical fermions with smeared links, Nucl. Phys. Proc. Suppl. 106 (2002) 1058–1060, See also A. Hasenfratz, Dynamical Simulation of Smeared Link Actions, Talk given at Lattice 2002, Boston, MA, USA.
[15] A.C. Irving, J.C. Sexton, E. Cahill, J. Garden, B. Joo, S.M. Pickles, Z. Sroczynsk, Tuning actions and observables in lattice QCD, Phys. Rev. D 58 (1998) 114504.
[16] B. Jegerlehner, Multiple mass solvers, Nucl. Phys. Proc. Suppl. 63 (1998) 958–960.
[17] C. Lanczos, Solution of systems of linear equations by minimized iterations, J. Res. Nat. Bur. Stand. 49 (1952) 33–53.
[18] M. Peardon, Progress in lattice algorithms, Nucl. Phys. B (Proc. Suppl.) 106–107 (2002) 3–11, See also M. Peardon, Multiple molecular dynamics time-scales in Hybrid Monte Carlo fermion simulations, Talk given at Lattice 2002, Boston, MA, USA.
[19] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in FORTRAN, Cambridge University Press, Cambridge, MA, 1993.