William & Mary
CHARTERED 1693

W&M ScholarWorks

Arts & Sciences Articles

Arts and Sciences

2009

Deflation for inversion with multiple right-hand sides in QCD

A. Stathopoulos
William & Mary, andreas@cs.wm.edu

A. M. Abdel-Rehim
William & Mary

K. Orginos
William & Mary

Follow this and additional works at: https://scholarworks.wm.edu/aspubs

Recommended Citation

Stathopoulos, A.; Abdel-Rehim, A. M.; and Orginos, K., Deflation for inversion with multiple right-hand sides in QCD (2009). *Scidac 2009: Scientific Discovery Through Advanced Computing*, 180. 10.1088/1742-6596/180/1/012073

This Article is brought to you for free and open access by the Arts and Sciences at W&M ScholarWorks. It has been accepted for inclusion in Arts & Sciences Articles by an authorized administrator of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.

Journal of Physics: Conference Series **180** (2009) 012073 doi:10.1088/1742-6596/180/1/012073

Deflation for inversion with multiple right-hand sides in QCD

A Stathopoulos[1], A M Abdel-Rehim[1] and K Orginos[2]

[1] Department of Computer Science, College of William and Mary, Williamsburg, VA 23187
[2] Department of Physics, College of William and Mary, Williamsburg, VA 23187

E-mail: andreas@cs.wm.edu,amrehim@cs.wm.edu,kostas@wm.edu

Abstract. Most calculations in lattice Quantum Chromodynamics (QCD) involve the solution of a series of linear systems of equations with exceedingly large matrices and a large number of right hand sides. Iterative methods for these problems can be sped up significantly if we deflate approximations of appropriate invariant spaces from the initial guesses. Recently we have developed *eigCG*, a modification of the Conjugate Gradient (CG) method, which while solving a linear system can reuse a window of the CG vectors to compute eigenvectors almost as accurately as the Lanczos method. The number of approximate eigenvectors can increase as more systems are solved. In this paper we review some of the characteristics of eigCG and show how it helps remove the critical slowdown in QCD calculations. Moreover, we study scaling with lattice volume and an extension of the technique to nonsymmetric problems.

1. Introduction

Lattice QCD is one of the most computationally demanding applications. The heart of the computation is the solution of the lattice-Dirac equation in a Euclidean space-time lattice [1], which translates to a linear system of equations $Mx = b$, often for a large number of right hand sides [2]. Because of large matrix size, iterative methods are the only way to solve the problem. The Dirac operator M is γ_5-Hermitian, or $\gamma_5 M = M^H \gamma_5$, for some γ_5 matrix, but it is often preferable to solve the normal equations $A = M^H M$ problem. Also, $M = m_q I - D$, where m_q is a parameter related to the quark mass and D is an operator. Beyond the very large dimension and number of right hand sides, M becomes increasingly ill-conditioned as $m_q \rightarrow m_{critical}$. In lattice QCD this is known as critical slowdown and is a limiting computational factor.

The solution of linear systems with many right hand sides is a challenging problem. Iterative methods that solve each system one by one tend to regenerate search directions within previously explored subspaces, thus wasting iterations. Seed and block methods are often used to share information between systems [3, 4, 5]. For Hermitian matrices, the most useful information lies in invariant subspaces near zero. Deflating these eigenpairs significantly improves conditioning of iterative methods. Such ideas have been tried in QCD [6], but effective deflation methods have only appeared recently. Moreover, one does not want to precompute the required eigenpairs through an eigensolver but to reuse the information built by the linear solvers.

For non-Hermitian matrices, the GMRESDR method [7, 8] computes approximate eigenspace information during GMRES(m). In GMRESDR, when the GMRES basis reaches m vectors, it is restarted not only with the residual of the approximate solution of the linear system but also with the $nev < m$ smallest magnitude harmonic Ritz pairs. Therefore, being equivalent to

© 2009 IOP Publishing Ltd

a version of the Implicitly Restarted Arnoldi, GMRESDR solves the linear system and at the same time converges to the *nev* smallest magnitude eigenpairs.

The GMRESDR approach transfers also to the Hermitian case. For Hermitian systems, however, GMRESDR is much more expensive per step than CG and, more importantly, because it restarts every m steps, it impairs both the optimal convergence of CG and the eigenvalue convergence of unrestarted Lanczos. Recently, a Recycled MINRES method was developed in [9] which reuses a window of m iteration vectors of the MINRES to approximate *nev* eigenvectors. When m vectors have been reached the window is restarted similarly to GMRESDR, but the MINRES itself is not restarted. Our independently developed eigCG method similarly keeps a window of m vectors without restarting CG but restarts the window in a locally optimal way, thus achieving nearly optimal convergence for the eigenvalues [10].

2. EigCG and incremental EigCG

It is well known that the residuals produced by the CG iteration are co-linear with the Lanczos iteration vectors, and that the Lanczos tridiagonal projection matrix can be obtained through the CG coefficients [11]. Therefore, if we kept all the CG vectors, V, we could solve the Lanczos projected eigenvalue problem $V^T AV$ and obtain approximate eigenpairs that satisfy a certain optimality. This however would imply unlimited storage for V or it would require a second CG pass to recompute V on the fly. Moreover, spurious eigenvalues would have to be dealt with.

The idea in our eigCG method is to use a limited memory window V with CG residuals as an eigenvector search space to keep track of the lowest *nev* Ritz vectors of the matrix A. When V grows to $m > 2nev$ vectors, we restart it as follows: We combine $u_i^{(m)},\ i = 1, nev$, lowest Ritz vectors at the m-th step (thick restarting) with their *nev* locally optimal CG recurrence vectors ($u_i^{(m-1)}$ from $m-1$ step) and restart the basis V with an orthonormal basis for this set [12]:

$$\left[u_1^{(m)}, u_2^{(m)}, \ldots, u_{nev}^{(m)}, u_1^{(m-1)}, \ldots, u_{nev}^{(m-1)}\right]. \tag{1}$$

In contrast to a restarted eigensolver, we do not use V to determine the next search direction. Instead, we leverage a window of the following $m - 2nev$ residuals computed by the unrestarted CG to extend the search space. The linear system convergence is unaffected. The proposed algorithm, eigCG, can be implemented on top of CG in an efficient way that does not require extra matrix-vector multiplications or long vector orthogonalizations, and that is fully parallelizable. For details the reader is referred to [10].

Once a set of approximate eigenvectors U have been computed, we can "deflate" subsequent CG runs by using the init-CG approach [13]. Given an initial guess $\tilde{x}_0$, we consider another guess whose U components are removed through the following Galerkin oblique projection:

$$x_0 = \tilde{x}_0 + U(U^H AU)^{-1} U^H (b - A\tilde{x}_0). \tag{2}$$

Then x_0 is passed as initial guess to CG. The init-CG converges similarly to a deflated system until the linear system error approaches the eigenvalue error in U. At that point, we restart init-CG, re-applying the initial deflation. One or two restarts are sufficient to obtain linear convergence, equaling the convergence of a linear system deflated with the accurate eigenvalues.

Solving one linear system would rarely yield enough eigenvector approximations to effectively deflate subsequent systems. Our Incremental eigCG employs a simple outer scheme that calls eigCG for $n_1 < s$ of the s right hand sides and accumulates the resulting *nev* approximate Ritz vectors from each run into a larger set, U. While solving each new system, this incremental scheme computes additional Ritz vectors and also improves the ones that have not sufficiently converged in previous systems.

In all our experiments and with modest values of *nev*, eigCG has demonstrated the surprising property of converging to the *nev* targeted eigenvectors with a convergence rate identical to

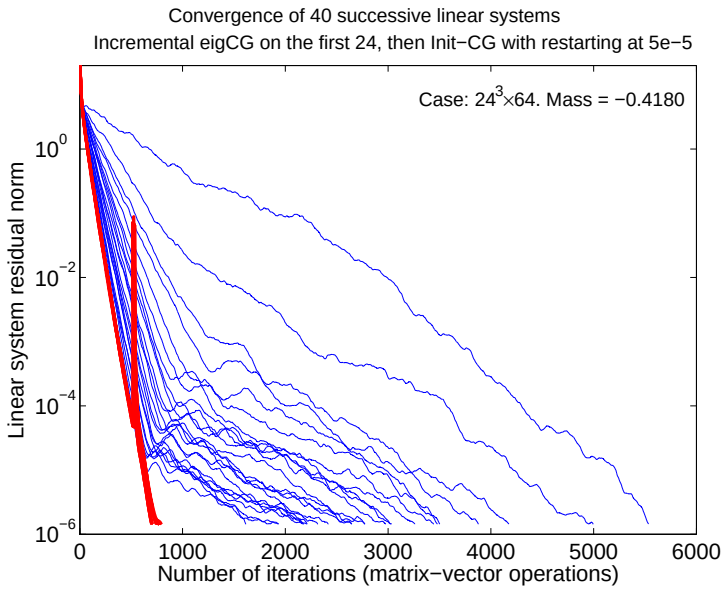


Figure 1. Improving convergence for the first 24 linear systems with Incremental eigCG(10,100) (blue). After the first 24 systems, we use the 240 vectors in U to deflate init-CG for another 16 vectors (red). Restarting is not used during the Incremental part, hence the knee in convergence curves. Restarting once at 5E-5 restores linear convergence. This is the 10 million matrix with a quark mass very close to the critical mass which causes the ill conditioning. The majority of the systems are solved with a speedup of 8.

that of unrestarted Lanczos. For example, a parameter $nev = 8$ has been sufficient for this behavior, while increasing m (the window size) offers only computational, not convergence, benefits. Several experiments in [10] study various aspects of this behavior.

3. Speeding up linear systems in QCD

We have implemented our algorithms in C and interfaced with Chroma, a lattice QCD C++ software base developed at Jefferson National Lab [14]. We have run both on an 8 node dual socket, dual core cluster with 4GB of memory per node, and on 256 processors of the Cray XT4 at NERSC, Lawerence Berkeley National Lab. Our computations involve all-to-all propagators which require the solution of several hundreds of right hand sides. Thus, we can afford to run Incremental eigCG on the first 24-32 systems accumulating about 250 vectors in U. For the rest of the systems, we use this U to deflate the init-CG as we described earlier.

The Dirac matrices used in this paper come from two ensembles of an anisotropic, two flavor dynamical Wilson fermion calculation. In both cases the sea pion mass is about 400(36) MeV and the lattice spacing is 0.108(7)fm. The anisotropy factor of the time direction is about 3 and the critical mass was determined to be -0.4188 in all cases. We consider three typical lattice sizes: one $16^3 \times 64$ and with the 12 variables per point the matrix size is $N = 3,145,728$; one $24^3 \times 64$ for a matrix size of $N = 10,616,832$; one $32^3 \times 96$ for a matrix size of $N = 37,748,736$.

Figure 1 shows a typical convergence behavior for Incremental eigCG for the 10 million matrix and a quark mass close to critical. The more ill-conditioned a matrix is, the more iterations it requires initially, so eigCG has enough time to identify more eigenvalues and more accurately.

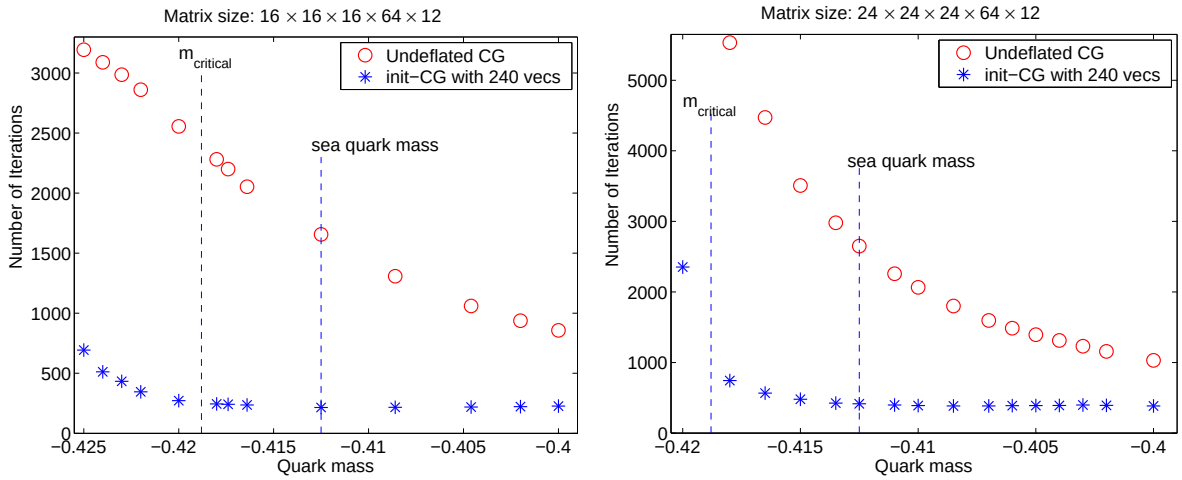


Figure 2. For each quark mass and its corresponding matrix we plot the average number of iterations required by init-CG to solve the rest 24 systems. For comparison the number of iterations of the non-deflated CG is reported. Critical slowdown is removed.

3.1. Removing critical slowdown in QCD

As the quark mass approaches $m_{critical}$, conditioning of the system rapidly deteriorates. However, shifting cannot change the density of the eigenvalues. Ill conditioning is simply due to a small number of eigenvalues approaching zero. Therefore if this small number of eigenvalues is captured and deflated, the conditioning of the matrix should be independent of the shift (quark mass). It is for this reason that our Incremental eigCG successfully removes the critical slowdown that has long plagued calculations in this field. Figure 2 shows the effects of the critical slowdown in the original, undeflated CG and how our method takes about the same number of iterations for any shift up to very close to the critical mass.

3.2. Scaling with volume

For the same configuration, increasing the lattice volume increases the matrix condition number. This volume scaling problem is much harder to address with deflation methods than critical slowdown because the eigenvalue density increases. To achieve a constant condition number, we need to capture the same portion of the spectrum which means computing more eigenvalues. Figure 3 shows the effect of 250 vector deflation on CG convergence for the three different size lattices. Note that deflated CG takes about 2.9 times more iterations on the largest lattice than on the smallest one, even though the volume increased by a factor of 12. By running Incremental eigCG with more systems we can remove this volume dependency at the expense of storing more vectors. Figure 4 shows the effect of deflating with 256, 512 and 1024 vectors on the largest lattice. We can achieve the same number of CG iterations on the large lattice as by deflating with about 5 times more vectors. However, there are diminishing returns in time.

4. EigBICG: extending to the nonsymmetric case

Recently, we extended the eigCG idea to eigBICG which is based on the nonsymmetric Lanczos. The main difference is that we need to keep two windows, W and V, corresponding to the left and right residuals built by BICG. We restart these windows similarly to (1) using left and right Ritz vectors. An Incremental eigBICG outer method accumulates the $2nev$ vectors to improve accuracy of the eigenspace, which is then used to deflated the BICGSTAB method. Figure 5 shows the deflation improvements on two smaller lattices in Matlab. Although eigBICG

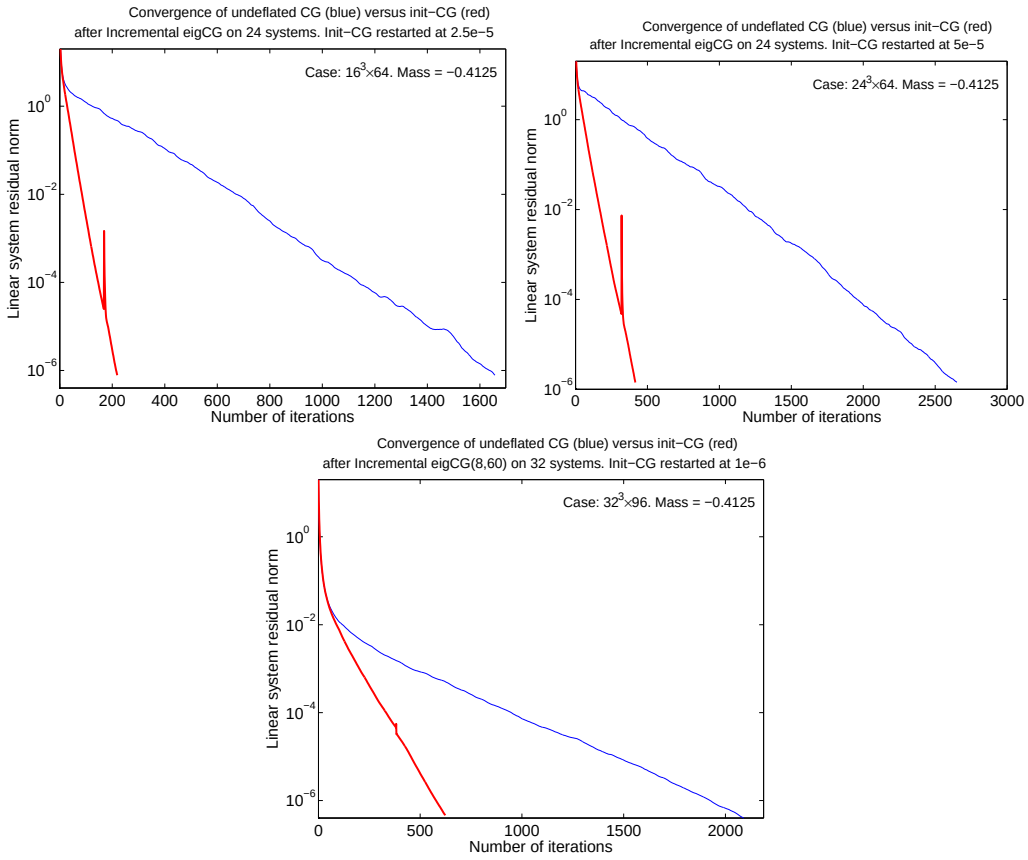
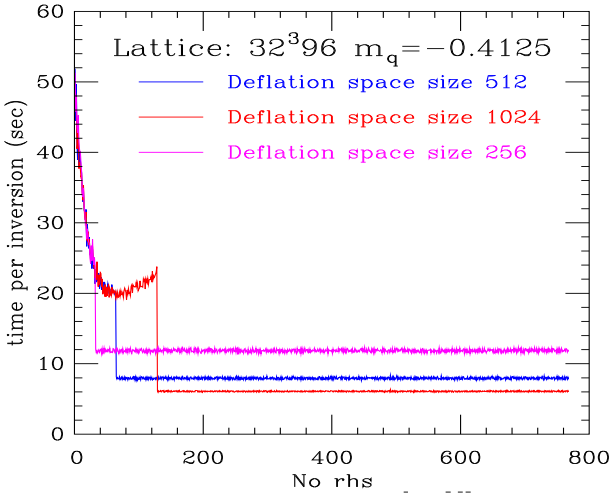


Figure 3. CG convergence improvement by deflating 240 vectors obtained through Incremental eigCG, for three lattices with increasing volumes. Volume slowdown is reduced but not removed.



Case	CG iterations undeflated	vectors deflated 240	256	512	1024
3M	1628	215			
10M	1907	337			
37M	2135	-	624	415	278

Figure 4. Left graph shows the time to solve the 780 linear systems after solving the first 32, 64, 128 systems with Incremental eigCG. Right table shows the CG iterations each case took.

converges similarly to unrestarted Lanczos, the Incremental part is less effective than in the symmetric case. Still, the nonsymmetric version is useful if the experiments involve much heavier quark masses as shown in the figure.

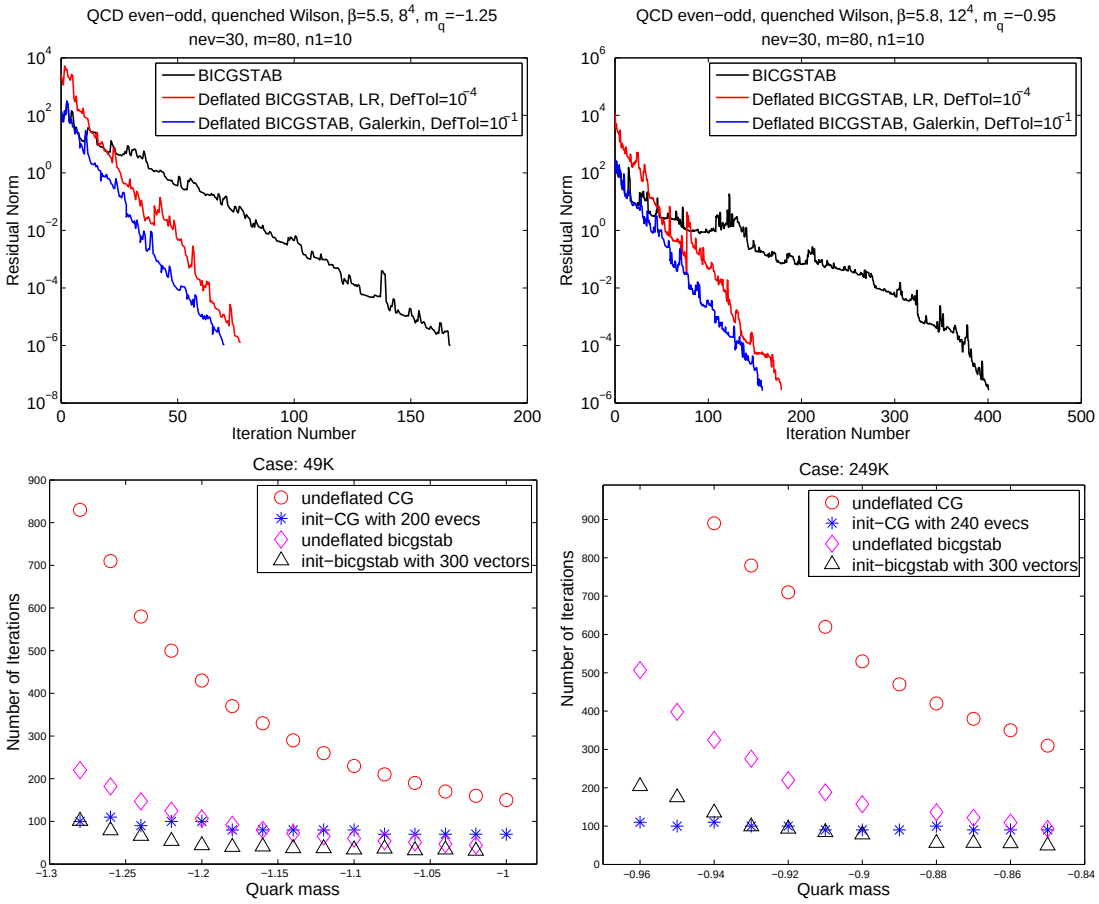


Figure 5. Deflated BICGSTAB using eigenspace from eigBICG and comparison with eigCG.

5. Conclusions

We have outlined a successful method, eigCG, that has helped us speedup dramatically the linear system solves in QCD and remove the critical slowdown. Volume slowdown is reduced but we have not yet reached lattice sizes where it will become the bottleneck. We have also presented a promising extension to the nonsymmetric case.

Acknowledgments

This work was supported by the NSF grant CCF-0728915, the DOE Jefferson Lab and the Jeffress Memorial Trust grant J-813.

References

[1] Wilson K G 1974 *Phys. Rev.* **D10** 2445–2459
[2] Foley J, Juge K J, O'Cais A, Peardon M, Ryan S and Skullerud J I 2005 *Comput. Phys. Commun.* **172** 145–162 (*Preprint* `hep-lat/0505023`)
[3] Simoncini V and Gallopoulos E 1995 *SIAM J. Sci. Comput.* **16** 917–933
[4] Chan T F and Wan W 1997 *SIAM J. Sci. Comput.* **18** 1698–1721
[5] O'Leary D P 1980 *Lin. Alg. Appl.* **29** 293–322

[6] de Forcrand P 1996 *Nucl. Phys. B (Proc. Suppl.)* **47** 228–235
[7] Morgan R B 2002 *SIAM J. Sci. Comput.* **24** 20–37
[8] Morgan R and Wilcox W 2004 Deflated iterative methods for linear equations with multiple right-hand sides Tech. Rep. BU-HEPP-04-01 Baylor University
[9] Wang S, de Sturler E and Paulino G H 2007 *International Journal for Numerical Methods in Engineering* **69** 2441–2468
[10] Stathopoulos A and Orginos K June 12, 2008 (original version July 1, 2007) Computing and deflating eigenvalues while solving multiple right hand side linear systems with an application to quantum chromodynamics Tech. Rep. arXiv.org 0707.0131v2 http://arxiv.org/pdf/0707.0131v2, Revised version under review in SIAM J. Sci. Comput.
[11] Saad Y 2003 *Iterative methods for sparse linear systems* (Philadelphia, PA, USA: SIAM)
[12] Stathopoulos A 2007 *SIAM Journal on Scientific Computing* **29** 481–514
[13] Saad Y, Yeung M, Erhel J and Guyomarc'h F 2000 *SIAM J. Sci. Comput.* **21** 1909–1926
[14] Edwards R G and Joo B (SciDAC) 2005 *Nucl. Phys. Proc. Suppl.* **140** 832 (*Preprint* `hep-lat/0409003`)