PROCEEDINGS OF SCIENCE

Double-pass variants for multi-shift BiCGstab(ℓ)

Simon Heybrock[*†]
Institute for Theoretical Physics, University of Regensburg, 93040 Regensburg, Germany
E-mail: simon.heybrock@physik.uni-regensburg.de

In analogy to Neuberger's double-pass algorithm for the Conjugate Gradient inversion with multi-shifts we introduce a double-pass variant for BiCGstab(ℓ). One possible application is the overlap operator of QCD at non-zero chemical potential, where the kernel of the sign function is non-Hermitian. The sign function can be replaced by a partial fraction expansion, requiring multi-shift inversions. We compare the performance of the new method with other available algorithms, namely partial fraction expansions with restarted FOM inversions and the Krylov-Ritz method using nested Krylov subspaces.

The XXVIII International Symposium on Lattice Field Theory, Lattice2010
June 14-19, 2010
Villasimius, Italy

[*]Speaker.
[†]Supported by the DFG collaborative research center SFB/TR-55 "Hadron Physics from Lattice QCD".

PoS(Lattice 2010)322

1. Introduction and Motivation

In this contribution we present double-pass variants for the multi-shift inverter BiCGstab(ℓ)[1] which, in some cases, can perform better than the conventional single-pass. The method is an analogue to Neuberger's double-pass Conjugate Gradient (CG) method [1, 2]. The use of BiCGstab(ℓ) instead of CG can be a speed advantage (for Hermitian matrices) or necessary (for non-Hermitian matrices). One possible application is the computation of quark propagators for a set of distinct masses. Here, however, we focus on computing the overlap operator of QCD. At non-zero quark chemical potential, $\mu \neq 0$, it is defined as

$$D_{ov}(\mu) = 1 + \gamma_5 \operatorname{sign}\left(\gamma_5 D_w(\mu)\right), \tag{1.1}$$

where $D_w(\mu)$ is the (Wilson) Dirac operator with chemical potential. For $\mu \neq 0$ the matrix $\gamma_5 D_w(\mu)$ is non-Hermitian. One way to compute the sign function of such a matrix, acting on a given vector b, is via a partial fraction expansion (PFE),

$$f(A)b \approx \sum_{s=1}^{N_s} \frac{\omega_s}{A+\sigma_s} b, \tag{1.2}$$

where we are especially interested in the case of $A = (\gamma_5 D_w)^2$ with $f(A) = 1/\sqrt{A}$, since $\operatorname{sign} z = z/\sqrt{z^2}$. The vectors $(A+\sigma_s)^{-1}b$ for a set of shifts $\{\sigma_s\}$ can be approximated by iterative inverters which find solutions in a Krylov subspace, defined as

$$\mathscr{K}_k(A,b) = \operatorname{span}(b, Ab, \ldots, A^{k-1}b). \tag{1.3}$$

A crucial feature of Krylov subspaces is their shift invariance, $\mathscr{K}_k(A+\sigma_s, b) = \mathscr{K}_k(A,b)$, which allows for so called *multi-shift inversions*, where one Krylov subspace suffices to compute $(A+\sigma_s)^{-1}b$ for a set $\{\sigma_s\}$ with little overhead per additional shift. We will refer to methods employing Eq. (1.2) as PFE methods.

2. Double-pass algorithm

As a starting point, we consider established algorithms to compute the sign function of a non-Hermitian matrix, (i) the Krylov-Ritz method with nested Krylov subspaces, introduced in [3], (ii) PFEs with FOM inversions, introduced in [4] and (iii) PFEs with BiCGstab(ℓ) as inverter. The latter has so far not been considered in the context of the sign function. For details on the BiCGstab(ℓ) method see [5], a version with shifts was introduced in [6]. Benchmark results are given in Fig. 1. The nested Krylov-Ritz algorithm outperforms both PFE methods, which is somewhat surprising since all rely on a similar Krylov subspace.[2] We can gain more insight by analysing the bad performance of BiCGstab(ℓ) in this case:

- Denote by N_s the number of shifts and by M_s the number of outer iterations of the BiCGstab(ℓ) algorithm until the system with shift σ_s is converged. Then, the multi-shift version has $\sum_{s=1}^{N_s} M_s l(0.5l+4.5)$ "*axpy*" operations ($y \leftarrow \alpha x + y$, for scalar α and vectors x and y) more than the BiCGstab(ℓ) algorithm without shifts.

[1] ℓ is the degree of the Minimal Residual polynomial in the algorithm

[2] Note however that the employed single-pass (nested) Krylov-Ritz method requires a huge amount of memory.

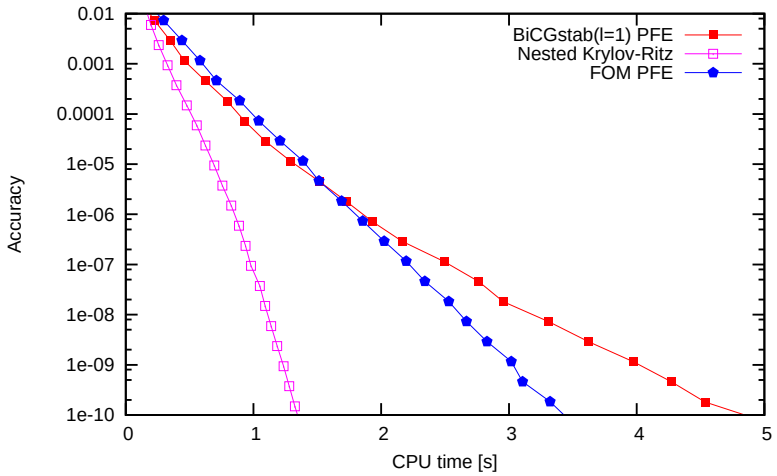

Figure 1: Accuracy vs. computation time for the overlap operator on a $4^3 \times 8$ lattice, $\beta = 5.32$, $\mu = 0.05$ with the Neuberger PFE. The number of poles N_s is chosen minimal for the desired accuracy as in [4]. In this plot it ranges from 10 (accuracy 0.01) to 58 (accuracy 10^{-10}). The 4 eigenvalues smallest in magnitude are deflated in advance.

- BiCGstab(ℓ) requires $2l+5$ vectors and the multi-shift version has $N_s(l+1)$ additional shift vectors, where typically $N_s = \mathcal{O}(10)$. These figures should be seen in relation to the typical cache size of current processors ($\mathcal{O}(1$ MByte)) and the size of a vector, e.g., 50 kByte (local volume 4^4) or 800 kByte (local volume 8^4). In a typical case not all shift vectors fit into cache and the access to main memory can become the bottleneck of the algorithm.

To tackle these performance restraints one can try a *double-pass* approach in analogy to Neuberger's double-pass algorithm for a multi-shift CG inversion. Schematically the idea is as follows: The quantity computed in Eq. (1.2) and approximated in a Krylov subspace is

$$\sum_{s=1}^{N_s} \omega_s (A+\sigma_s)^{-1} b \approx \sum_{s=1}^{N_s} \omega_s \sum_{n=1}^{N} w_s^{(n)}, \tag{2.1}$$

where N is the number of iterations in the inverter and $w_s^{(n)}$ is a vector for shift s in iteration n. To remove s dependent vectors one could try to swap the sums over s and n, however $w_s^{(n)}$ is given by a recursion relation,

$$w_s^{(n)} = \alpha_s^{(n)} w_s^{(n-1)} + \beta_s^{(n)} v^{(n)} = \sum_{i=1}^{n} \gamma_{s,i}^{(n)} v^{(i)}, \tag{2.2}$$

where $v^{(n)}$ is an unshifted iteration vector. In the last step the recursion of the vectors $w_s^{(n)}$ was resolved. By combining Eqs. (2.1) and (2.2) and summing over s (and n), all vectors depending on s are removed from the algorithm. However, the coefficients $\gamma_i = \sum_{s,n} \gamma_{s,i}^{(n)}$ are not known until the end of the iteration. There are two options

1. (*double-pass*): Follow Neuberger's approach by running the algorithm once to obtain γ_i. In a second pass generate the vectors $v^{(i)}$ again and compute $\sum_i \gamma_i v^{(i)}$.

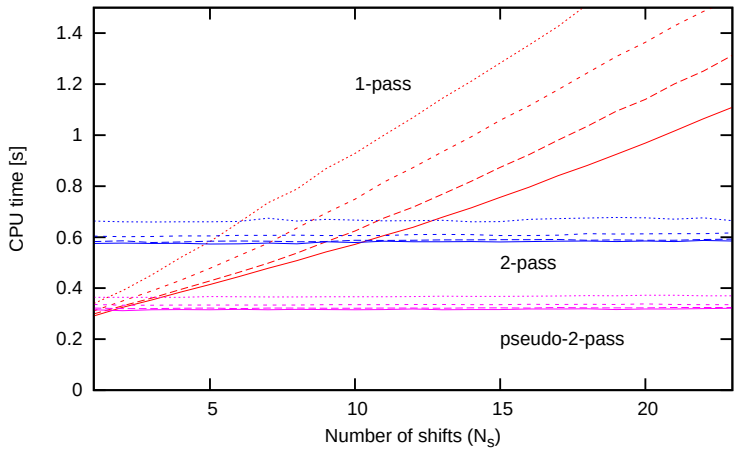

Figure 2: Computation time vs. N_s for fixed $Ml = 256$ for a $4^3 \times 8$ lattice (Wilson Dirac operator) for all three BiCGstab(ℓ) variants. Results are given for $l = 1, 2, 4, 8$ with lines solid to dotted.

2. (*pseudo-double-pass*): Compute the coefficients γ_i as in double-pass, but store all $v^{(i)}$ during the first pass instead of recomputing them in a second pass.

Both methods remove all s-dependent vectors from the algorithm, hence reducing the number of operations and number of vectors to be held in cache. In our case, to obtain the coefficients corresponding to the (schematic) coefficients γ_i, the recursion has to be solved for the BiCGstab(ℓ) algorithm. The result is given in Sec. 4.

3. Cost analysis and benchmarks

The number of operations (scalar ones are omitted) and vectors of the multi-shift BiCGstab(ℓ) algorithms are given in the following table (M denotes the number of outer iterations of the algorithm and the dimension of the Krylov space is $2Ml$):

Method	#Mv	#axpy	#dot-products	#vectors
1-pass	$2Ml$	$Ml(1.5l+5.5)+\sum_{s=1}^{N_s} M_s l(0.5l+4.5)$	$Ml(0.5l+3.5)$	$2l+5+N_s(l+1)$
2-pass	$4Ml$	$Ml(1.5l+5.5)+Ml(1.5l+4.5)+2Ml$	$Ml(0.5l+3.5)$	$2l+5$
pseudo-2-pass	$2Ml$	$Ml(1.5l+5.5)+2Ml$	$Ml(0.5l+3.5)$	$2l+5+2Ml$

The number of vectors alone is not always meaningful: In single-pass a considerable subset[3] of the $N_s(l+1)$ vectors is accessed in each iteration of the algorithm. In pseudo-double-pass each of the $2Ml$ vectors is written and read exactly once, in total. That is, the access pattern of pseudo-double-pass requires less memory access than single-pass, even though many more vectors are involved.

As a naive test of the figures given in the table we consider the algorithm runtime for fixed Ml with a varying number of shifts, given in Fig. 2. The two-pass and pseudo-two-pass runtime is largely independent of N_s. The pure operation count of two-pass would yield an almost doubled computation time compared to pseudo-two-pass. In practice, however, it is less since no (or less)

[3] depending on ℓ, and on the removal of converged systems from the iteration

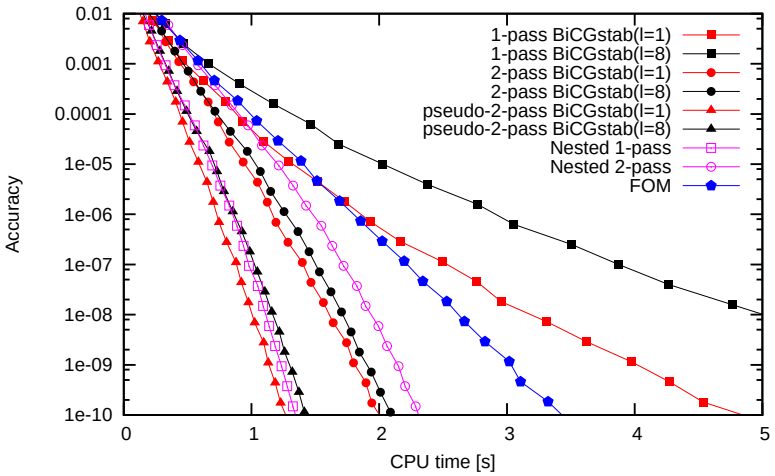

Figure 3: Accuracy vs. computation time for the overlap operator ($4^3 \times 8$ lattice with $\beta = 5.32$, $\mu = 0.05$). The timings are averaged over 200 independent gauge configurations. Note that an extreme case with little deflation (4 eigenvalues smallest in magnitude) was chosen for this plot where many poles are required (as before N_s is scaled from 10 to 58), yielding a large speed advantage of the double-pass algorithms. When less poles are needed (e.g., for small μ when the Zolotarev PFE can be used instead of the Neuberger expansion) the performance difference is often smaller.

main memory access is required. An effect of the cache size can be seen from the single-pass $l = 1$ curve. The slope changes in the vicinity of $N_s = 10$, which is consistent with the cache size of 4 MByte and the size of a vector of 100 kByte. As a further observation, the relative performance loss for large ℓ is much smaller in the double-pass methods compared to single-pass, which is not surprising since the number of required vectors increases with ℓ.[4]

As a more realistic benchmark we compute the overlap operator for given configurations in Fig. 3. The double-pass and pseudo-double-pass BiCGstab(ℓ) algorithms perform as well or even better than the nested double-pass and single-pass algorithms, respectively. Note that also the respective memory requirements are similar. In double-pass the performance does not degrade for $l > 1$ as it does for single pass. To explore differences between the nested Krylov-Ritz method and the BiCGstab(ℓ) methods a series of benchmarks was performed, where both the number of deflated eigenvectors and the chemical potential μ were varied. The tests indicate that BiCGstab(ℓ) profits more from deflation than the Krylov-Ritz method does. On the other hand, for large μ, BiCGstab(ℓ) tends to stagnate earlier than Krylov-Ritz.

Finally we give results of a larger-scale simulation in Fig. 4. As before, pseudo-double-pass BiCGstab(ℓ) is the algorithm performing best. The optimal ℓ depends on the number of cores N_c. Due to memory limitations for small N_c and network limitations for large N_c, there is an optimal N_c minimizing the total CPU time.

4. Algorithm details

We follow the notation in Ref. [6] where also a listing of BiCGstab(ℓ) is given. Upper indices

[4] Since we work at fixed Ml this plot does not tell which ℓ is optimal, since the convergence rate depends on ℓ.

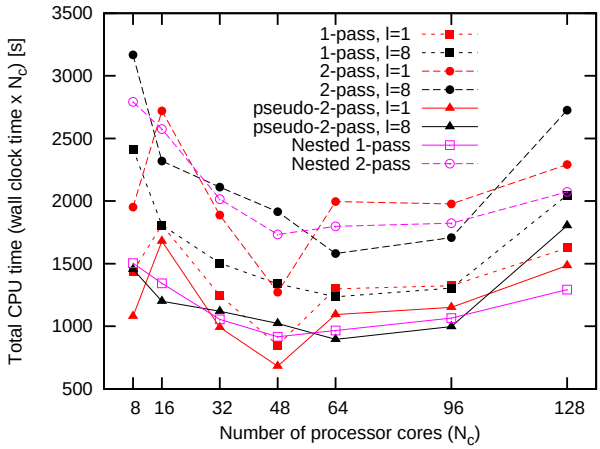

Figure 4: Total computation time vs. number of cores for the nested Krylov-Ritz method and pseudo-double-pass BiCGstab(ℓ) for $l = 1$ and 8. The simulation uses a $12^3 \times 24$ lattice, $\beta = 5.71$, $\mu = 0.016667$ with 44 deflated eigenvalues. The accuracy is 10^{-10}, obtained with $N_s = 16$ shifts. The dimension of the Krylov subspace is about 2000 for BiCGstab($\ell = 1, 8$) as well as for nested Krylov-Ritz. The benchmarks where done on an Opteron 2354 Cluster (2.2 GHz, 2 quad-core processors per node, 16 GByte RAM per node, Infiniband network). Lines are drawn to guide the eye.

mj denote iteration j of BiCG part and outer iteration m. Define the coefficients

$$A_{mj} = \sum_s \omega_s \frac{1}{(\vartheta^s \varphi^s)^{mj}} \sum_{j'=j}^{l-1} (\alpha^s)^{mj'} \left(\prod_{k=j+1}^{j'} (-\beta^s)^{mk} \right), \tag{4.1}$$

$$B^s_m = \sum_{n=m+1}^{M-1} \left\{ \sum_{j=0}^{l-1} (\alpha^s)^{nj} \left(\prod_{k=0}^{j} (-\beta^s)^{nk} \right) \right\} \left\{ \prod_{k=m+1}^{n-1} \left(\sum_{j=0}^{l} \frac{-\gamma_j^k}{(\psi^s)^k} \sigma_s^j (-1)^{l-j} \right) \prod_{k'=0}^{l-1} (\beta^s)^{kk'} \right\}, \tag{4.2}$$

$$D^s_{mij} = \frac{-\gamma_i^m}{(\psi^s)^m} \frac{1}{(\vartheta^s \varphi^s)^{mj}} \prod_{k=j+1}^{l-1} (-\beta^s)^{mk}, \qquad E^s_{mj} = \frac{-1}{(\psi^s)^m} \left(\sum_{i=j+1}^{l} \gamma_i^m \sigma_s^{i-j-1} (-1)^{l-i} \right) \prod_{k=j+1}^{l-1} (\beta^s)^{mk}, \tag{4.3}$$

$$F^s_{mj} = \frac{1 - (\alpha^s)^{mj} \sigma_s}{(\alpha^s)^{mj} (\vartheta^s \varphi^s)^{mj}}, \qquad G^s_{mj} = -\frac{1}{(\alpha^s)^{mj} (\vartheta^s \varphi^s_{\text{new}})^{mj}}, \tag{4.4}$$

where $\gamma_0 = -1$, and a matrix

$$(M^m)_{jk} = -\sum_{q=k}^{j-1} \alpha^{mq} \left(\prod_{p=k+1}^{q} (-\beta^{mp}) \right), \qquad j,k = 0, \ldots, l-1. \tag{4.5}$$

Then the contribution to $\sum_s \omega_s x^s$ from the BiCG part is given by

$$\begin{aligned} \mathbf{x}_{\text{BiCG}} = \sum_{m=0}^{M-1} \sum_{j=0}^{l-1} \Bigg\{ & \sum_{p=j}^{l-1} \sum_{k=j}^{p} \left[A_{mp} ((M^m)^j)_{pk} + \sum_{i=0}^{j} \left(\sum_s \omega_s B^s_m D^s_{mip} \right) ((M^m)^{j-i})_{pk} \right] \\ & \times \left[(\mathbf{r}_j)^{mj} - \sum_{q=j}^{k-1} \alpha^{mq} \left(\prod_{p'=j+1}^{q} (-\beta^{mp'}) \right) (\mathbf{u}_{j+1})^{mj} \right] \\ & + \left(\sum_s \omega_s B^s_m E^s_{mj} F^s_{mj} \right) (\mathbf{r}_j)^{mj} + \left(\sum_s \omega_s B^s_m E^s_{mj} G^s_{mj} \right) \left((\mathbf{r}_j)^{mj} - \alpha^{mj} (\mathbf{u}_{j+1})^{mj} \right) \Bigg\}. \end{aligned} \tag{4.6}$$

All operations involving the vectors $\mathbf{u}_i^s$ can be removed from the original algorithm. The final result is given by adding $\mathbf{x}_{\text{BiCG}}$ to the contributions of the seed system and the MR-part of the algorithm, $\sum_s \omega_s \mathbf{x}_{MR}^s$, which is computed trivially. For reference an implementation of the algorithms is provided online at http://sourceforge.net/projects/bicgstabell2p/.

5. Conclusions

We have presented an extension of the double-pass trick from Conjugate Gradient to the more general BiCGstab(ℓ). While initially PFE methods looked inferior to the nested Krylov-Ritz method in the non-Hermitian case, our new (pseudo-)double-pass BiCGstab(ℓ) is a method with similar performance. Our benchmarks concentrated on the overlap operator where pseudo-double-pass performs as well or even better than the nested Krylov-Ritz method on the tested architectures. Current supercomputers might have enough main memory such that pseudo-double-pass is feasible, but this will depend on details of the simulation. Large values of ℓ yield less overhead in the double-pass methods compared to single-pass. This could boost the application of the algorithm in problems where $l > 1$ is crucial for convergence. We plan to investigate the efficiency of the double-pass BiCGstab(ℓ) algorithms for other functions aside from the inverse square root.

As a closing remark let us mention that a pseudo-double-pass method can also be used instead of the usual (double-pass) Conjugate Gradient method. This extension seems trivial, though we are not aware of any mention in the literature.

Acknowledgements

I want to thank Jacques C.R. Bloch and Tilo Wettig for support, advice and discussions.

References

[1] H. Neuberger, *Minimizing storage in implementations of the overlap lattice-Dirac operator*, *Int. J. Mod. Phys. C* **10** (1999) 1051–1058, [hep-lat/9811019].

[2] T.-W. Chiu and T.-H. Hsieh, *A note on Neuberger's double pass algorithm*, *Phys. Rev. E* **68** (2003) 066704, [hep-lat/0306025].

[3] J. C. R. Bloch and S. Heybrock, *A nested Krylov subspace method to compute the sign function of large complex matrices*, *Comput. Phys. Commun.* (to be published) [arXiv:0912.4457].

[4] J. C. R. Bloch *et. al.*, *Short-recurrence Krylov subspace methods for the overlap Dirac operator at nonzero chemical potential*, *Comput. Phys. Commun.* **181** (Oct., 2010) 1378–1387, [arXiv:0910.1048].

[5] G. L. G. Sleijpen and D. R. Fokkema, *BiCGstab(ℓ) For Linear Equations Involving Unsymmetric Matrices With Complex Spectrum*, *Electronic Transactions on Numerical Analysis* **1** (1993) 11–32.

[6] A. Frommer, *BiCGStab(ℓ) for families of shifted linear systems*, *Computing* **70** (2003), no. 2 87–109.