Andreas Stathopoulos

A new method for computing eigenvectors during unrestarted CG while solving multiple right hand sides

College of William and Mary
Department of Computer Science
P O Box 8795
Williamsburg
VA 23187-8795
andreas@cs.wm.edu
Konstantinos Orginos

Solving large, Hermitian linear systems with multiple right hand sides (MRHS) presents a computational bottleneck for many applications, including our target application, Quantum Chromodynamics (QCD). We focus on Krylov iterative methods and in particular the Conjugate Gradient (CG). Although our results apply also to Preconditioned CG, for presentation simplicity we discuss only the unpreconditioned case.

For MRHS, the common approach is to use either block or seed Krylov methods. We focus on seed methods as the right hand sides may not be all available at the same time. Seed methods use information computed during the solution of one linear system to project subsequent systems. However, for QCD problems the right hand sides are statistically unrelated which implies that the best information to reuse across systems is the approximated extreme eigenspaces. Instead of running a separate eigensolver, some methods have arisen that attempt to compute these eigenspaces from within the Krylov linear solver. A notable example is the GMRESDR method which exploits the relationship between implicitly restarted Arnoldi and GMRES for non-Hermitian matrices. For Hermitian matrices, GMRESDR is much more expensive than CG and, most importantly, its restarting impairs the optimal convergence achieved by CG/Lanczos for both the linear system and the eigenvalue problem.

One approach that is often tried is to store all CG residual iterates (which are the same as the Lanczos vectors) and the tridiagonal Lanczos matrix, and once the linear system converges to compute the required Ritz vectors and values. In theory, this is equivalent to the optimal Lanczos method but has several computational shortcomings. The Lanczos vectors must be stored or recomputed during a second pass and the tridiagonal matrix may become too large. More importantly, loss of orthogonality gives rise to ghost eigenvalues that must be managed at the end.

In this talk, we present an algorithm, eigCG, which computes eigenvalue and eigenvector approximations of a Hermitian operator by reusing information from

the *unrestarted* CG method. This is achieved by keeping a search space that includes current eigenvector approximations and only the last few CG iteration vectors. The crucial step is how we restart this search space to keep computations tractable. The CG iteration is completely unaffected.

The idea is to use the locally optimal restarting strategy from our Generalized Davidson+k (GD+k) eigenvalue method to update a window of Ritz vectors but without restarting the Lanczos process. Consider the CG(Lanczos) method on matrix A, and a set of $m > nev$ vectors, V, that keep track of the required nev Ritz vectors and the most recent Lanczos vectors. Initially, V is composed of the first m Lanczos vectors with the usual tridiagonal projection matrix $T_m = V^H AV$ (this also can be computed from the scalar iterates of CG). At this point, we apply the Rayleigh-Ritz on T_m and restart V with the current nev Ritz vectors $u_i^{(m)}$. The critical feature of our algorithm is that we also include in V the nev Ritz vectors from step $m-1$. Thus, the restarted V is computed as an orthonormal set of vectors spanning:

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

From this point forth, our algorithm deviates from thick restarted Lanczos or GD+k because it does not use the residual of $u_1^{(m)}$ to build a new Lanczos iteration. It keeps using the next $(m-2nev)$ CG vectors, v_i, from the original, unrestarted process to append to V:

$$[u_1^{(m)}, u_2^{(m)}, \ldots, u_{nev}^{(m)}, u_1^{(m-1)}, \ldots, u_k^{(m-1)}, v_{m+1}, \ldots, v_{m+m-2nev}].$$

Note that orthogonality need not be maintained as the Lanczos vectors v_i are orthogonal (in exact arithmetic) to all previous Lanczos vectors and therefore to any Ritz vectors in V. The projection matrix for the resulting V is $H = V^H AV = \begin{bmatrix} \Theta & C^H \\ C & T \end{bmatrix}$, where Θ is a diagonal matrix of size $2nev$ containing the nev Ritz values, T is the section from $m+1$ to $2m-2nev$ of the tridiagonal matrix of the unrestarted Lanczos, and only the first row of C is non-zero. Thus, H can be computed from the Lanczos coefficients with negligible additional cost. At this point, the Rayleigh-Ritz is applied again and V is restarted with the nev Ritz vectors from H and the previous nev Ritz vectors from $H(1:m-1, 1:m-1)$. Then the update algorithm repeats, while the original CG continues.

The use of Rayleigh Ritz and thick restart guarantee monotonic convergence of the nev Ritz values in V, but not necessarily to any eigenvalues. To our surprise, in all of our experiments with $nev > 4$, the nev updated vectors converge to the required eigenpairs at a rate identical to that of unrestarted Lanczos! Note that any information that is discarded from V when we restart cannot be reintroduced in V because future Lanczos vectors v_i are orthogonal to it. By restarting through eq. (1) almost all Lanczos information regarding the nev eigenpairs is kept in condensed form in V. In fact, with thick restarting alone (i.e., without the $u_i^{(m-1)}$ directions as performed in RMINRES) the algorithm stagnates. We present experiments that demonstrate this behavior.

We have used eigCG to solve large QCD problems where the number of right hand sides is in the order of hundreds. We apply eigCG with $nev = 10, m = 100$ for the first 12 systems, incrementally building more than 40 eigenvectors of A computed to machine precision. When these eigenvectors are deflated from subsequent systems through simple init-CG, factors of 8 speedup are obtained in the number of iterations. More importantly for QCD, these speedups increase as the quark mass of the system approaches a critical mass which gives rise to the so-called critical slow-down in these problems. Our method completely removes this type of slowdown for any type of matrix size.