Seed methods for linear equations in lattice QCD problems with multiple right-hand sides

Abdou M. Abdel-Rehim*, Ronald B. Morgan# and Walter Wilcox*

*Dept. of Physics, Baylor University, Waco, TX 76798-7316
#Dept. of Mathematics, Baylor University, Waco, TX 76798-7328

I. Introduction:

Lattice QCD problems often involve solving complex non-Hermitian linear systems of equations $A\,x_i = b_i$ with $i=1,2,\ldots,nrhs$. These systems can be solved by finding the solution of the positive-definite Hermitian systems $A^* A\,x_i = A^*\,b_i$. The seed conjugate gradient method (seed-CG) [1-4] solves one right-hand side with the conjugate gradient method and simultaneously projects for the other right-hand sides over the Krylov subspace thus developed. Then the next system is solved and used to seed the remaining ones. Rounding error in the conjugate gradient method limits how much the seeding can improve convergence. We propose three changes to seed-CG: only the first right-hand side is used for seeding, this system is solved past convergence, and the roundoff error is controlled with some reorthogonalization.

II. Method:

Before giving the seed-CG algorithm for multiple right-hand sides, it will be helpful to highlight its main features:

- Even though the algorithm is applied to the Hermitian problem, we always determine convergence of the residual of the non-Hermitian system.
- We build the Krylov subspace for the first r.h.s. using the Lanczos algorithm with occasional reorthogonalization to control roundoff errors.
- The first right-hand side is solved past the convergence limit in order to make sure that the Krylov subspace is large enough to approximate small eigenvalues even though no explicit eigenvalue problem is solved.
- Reorthogonalization is repeated after every new *nf* Krylov vectors are generated. The current vector as well as the one before are reorthogonalized over all previously generated vectors.
- The solution of the first r.h.s. and the projections for subsequent right-hand sides are done simultaneously.
- The solution of the first r.h.s. and the projections for subsequent right-hand sides are built in a cumulative manner with the generation of every new Krylov subspace vector.
- After solving the first right-hand side, the seed solutions for subsequent right-hand sides are used as initial guesses that are passed to a standard conjugate gradient algorithm.

III. Algorithm:

- Consider the transformed systems $M x_i = \varphi_i$ for i=1,2,…nrhs where $M = A^\dagger A$ and $\varphi_i = A^\dagger b_i$. Set $x_i = 0$ as initial guess for all right-hand sides and choose the convergence residual tolerance *resmax* and the frequency of re-orthogonalization *nf* for the first right-hand side. Choose *resmax'* for subsequent right-hand sides such that *resmax* > *resmax'*.
- ***Start***: $r_0 = \varphi_1$, $\rho = ||r_0||$, $v_1 = r_0 / \rho$, $\beta_0 = 0$
- ***Solve first right-hand side and build seed for other right-hand sides:***

```
For j=1,2,....until convergence Do:
   wf = M v_j
   wf = wf - β_{j-1} v_{j-1}
   α_j = Re{ (v_j , wf) }
   wf = wf - α_j v_j
   if ( j =1 ) then
     w = v_1 / α_1 ,  y = ρ , x_1 = x_1 + y w , δ_1 = α_1
     For k=2...,nrhs Do
        η_k = (v_1 , φ_k)
        x_k = x_k + η_k ω
     EndDO
   else
     γ_{j-1} = β_{j-1} / δ_{j-1} , δ_j = α_j - γ_{j-1} β_{j-1}
     w = v_j / δ_j - β_{j-1} w / δ_j
     y = - γ_{j-1} y
     x_1 = x_1 + y w
     For k=2...,nrhs Do
        η' = (v_j , φ_k) - γ_{j-1} η_k
        x_k = x_k + η' w
        η_k = η'
     EndDO
   endif
   Re-Orthogonalization:
   if( (j mod nf =0) OR (j+1 mod nf =0) ) then
     For k=1,2,...j Do
       wf = wf – (v_k , wf ) v_k
     EndDo
   endif
   β_j = ||wf||   ,   v_{j+1} = wf / β_j
   h = b_1 – A x_1
   if ( ||h|| < resmax ) STOP
endDO jdo
```

- ***Solve subsequent right-hand sides:***

```
For k=2,..,nrhs Do
  solve M x_k = φ_k using CG and the initial guesses x_k .
EndDo
```

IV. Results:

The method is tested for Wilson fermions on a 20^3x32 quenched Wilson configuration at β=6.0 . We choose κ to be essentially $\kappa_{critical}$ to make the problem difficult. We considered two point sources with Dirac and color indices 1,1 and 1,2 respectively. The relative residual norm at convergence is 1.0E-08. Standard CG for this problem requires a Krylov subspace of dimension about 2,500 vectors (5000 matrix-vector products). In order to reduce the matrix-vector products for the second right-hand side it will be necessary to solve the first right-hand side past convergence. In Figure [1], results for the second right-hand side is shown when the first right-hand side is solved using Krylov subspaces of dimension 2500,3000,3500, and 4000. For this test, we used *nf*=100 for the reorthogonalization frequency. Finding the largest value of *nf* that could be used for this problem was done by experimenting with different choices of *nf* values. In Figure [2], we show results for the second right-hand side using a subspace dimension 3500 and *nf* values 10,100,200, and 250. As can be seen from Figure [2], a value less than 250 is suitable for this problem. Having a large *nf* value reduces the cost associated with reorthogonalization.

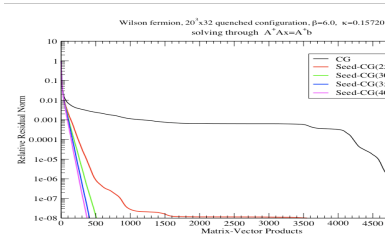

Figure 1: Results for the second right-hand side with seed-CG(*nk*,*nf*) where *nk* and *nf* are the Krylov subspace dimension and the frequency of reorthogonalization used while solving the first right-hand side. Standard CG results are shown for comparison.

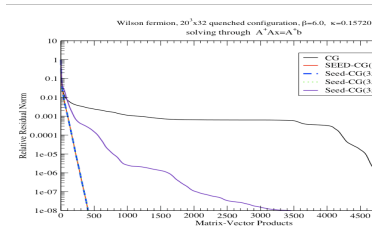

Figure 2: Effect of changing the frequency of reorthogonalization on the number of matrix-vector products for the second right-hand side.

V. Storage and Reorthogonalization Cost:

Because of the reorthogonalization step, the algorithm requires the storage of all the Krylov subspace vectors while solving the first right-hand side. The method requires more memory than standard CG which has to be taken into consideration. On the other hand, this storage is required only while solving the first right-hand side and should be done in a dynamical manner by allocating and deallocating memory during the run of the program. The parameter *nf* that determine the frequency of re-orthognalization has to be found by experimenting with different values to find a balance between the cost of reorthogonalization and the loss of improvement because of roundoff errors. In Table [1], we compare the time used by CG and seed-CG for the first right-hand side. Even though they have the same number of matrix-vector products, there is the cost of re-orthogonalization and the seeding for the second right-hand side. The results were obtained using the high performance cluster at Baylor university which has 8 processors per node at 2.66GHZ and 16GB RAM per node. The run was done on 50 processors using 5 processors per node allowing for 3.2 GB RAM per process. It was possible to allocate up to 4000 Krylov vectors using this configuration without a need for memory swapping that leads to a slower run. The timing results shows that as the frequency of reorthogonalization increases, the cost for solving the first right-hand side increases. However, it is possible to control roundoff errors with less frequent reorthogonalization as in the case with nf=200 where the extra overhead is very close to the standard CG time. In general, an optimization step will be necessary to find the best value of *nf* and *nk* depending on how much memory is available and the number of right-hand sides.

nk , *nf*	Approximate time (sec)
3500, 10	1253
3500, 100	598
3500, 200	320
3500, 250	305

Table 1: Time used to solve the first right-hand with seed-CG. For standard CG the time is 227 seconds for the same number of matrix-vector products.

VI. Conclusions:

- For linear systems with multiple right-hand sides, seed conjugate gradient methods significantly reduce the number of matrix-vector products.
- Storage cost for the Krylov subspace vectors is needed only while solving the seed system and should be allocated/deallocated at run time to reduce memory requirements.
- Re-orthogonalization is essential in order for the method to be effective, specially when the dimension of the needed Krylov subspace is large. However, the extra re-orthogonalization cost is very modest as compared to standard CG.

Acknowledgements:

Calculations done with HPC systems at Baylor University.

REFERENCES

[1] C. Smith, A. Peterson, and R. Mittra, IEEE Trans. Anten. Propag., 37:1490-1493, 1989.

[2] . T. F. Chan and W. Wan, SIAM J. Sci. Comput., 18:1698-1721, 1997.

[3] B. N. Parlett, Linear Algebra Appl., 29 (1980), pp.323-346.

[4] Y. Saad, Math.Comp., 48 (1987), pp. 651-662.