Fast Iterative Solvers: Project 3

Jaeyong Jung (359804)

August 31, 2016

1. **Step 1.**

The biggiest eigenvalue of nos6 matrix is $\mathbf{7.650603 \times 10^6}$ when the tolerance is $\mathbf{10^{-8}}$. The convergence aginst iteration k is described on Figure 1. It converges at $\boldsymbol{777 - th}$ step.

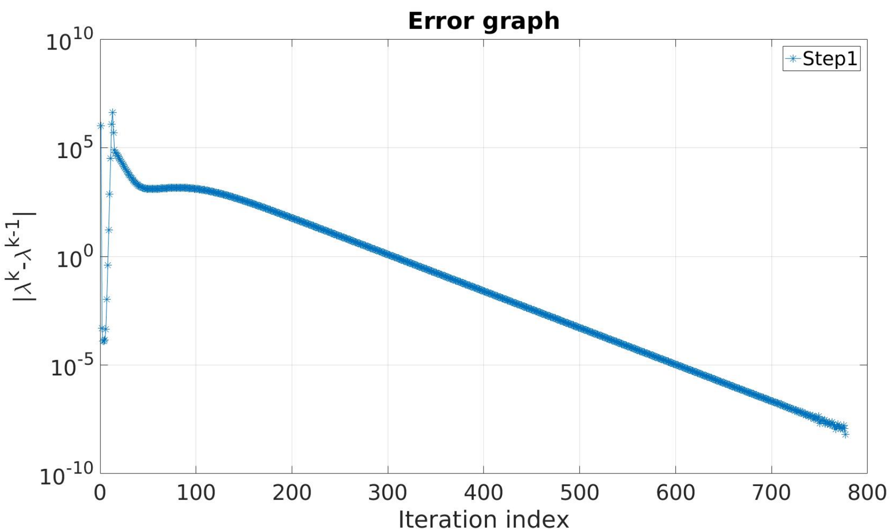

Figure 1: The solution converges at 777-th iteration

2. **Step 2.**

The smallest eigenvalue of nos6 matrix is approximated as **1.000059** when the tolerance is $\mathbf{10^{-8}}$. Conjugate Gradient method is used to solve linear systems in this method. It is observed that more iteration is necessary to solve minimum eigenvalue problem. It converges at 4416-th step, which is much grater than iteration in step 1.

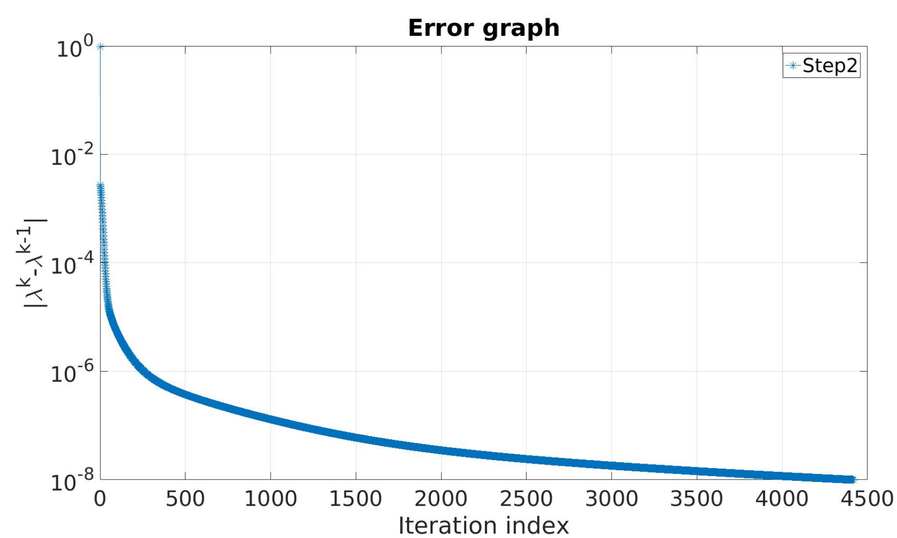

Figure 2: The solution converges at 4416-th iteration

3. Implement a Lanczos method to find the largest eigenvalue of the matrix s3rmt3m3.mtx.

- Run the power method for 100, 500, 1000, and 5000, iterations and record the error, and the runtime for each of those runs.

Number of iterations	Error	Runtime (s)
100	**64.9891**	**0.0458**
500	**0.2948**	**0.232**
1000	$\mathbf{2.797 \times 10^{-3}}$	**0.4761**
5000	$\mathbf{4.547 \times 10^{-11}}$	**2.3596**

Table 1: Power iteration results according to number of iterations, s3rmt3m3.mtx

- Run the Lanczos method for m = 30, 50, 75, 100, where m is the dimension of the Krylov space, in order to compute the maximum eigenvalue of the triangular Lanczos matrix.

Size of krylov space	Error	Runtime (s)	tolerance
30	**15.688**	$\mathbf{7.366 \times 10^{-3}}$	$\mathbf{10^{-2}}$
50	$\mathbf{7.686 \times 10^{-2}}$	$\mathbf{1.249 \times 10^{-2}}$	$\mathbf{10^{-4}}$
75	$\mathbf{1.5203 \times 10^{-4}}$	$\mathbf{1.998 \times 10^{-2}}$	$\mathbf{10^{-6}}$
100	$\mathbf{1.5304 \times 10^{-8}}$	$\mathbf{2.998 \times 10^{-2}}$	$\mathbf{10^{-10}}$

Table 2: The Lanczos method, with adjustable tolerance, for m = 30, 50, 75, 100, s3rmt3m3.mtx

- You may optionally try to optimize the tolerances for the Lanczos method yourself. (In that case, you need to understand first, why it makes sense to use different tolerances for the different Krylov spaces!)).

A simple linear interpolation is implemented to calculate tolerance with respect to size of Krylov space.

C++ code:

```
// tolerance calculation according to Krylov space size.
// next time, I am going to change it. Apply linear
   interpolation in each region.
// y = y1 + (x-x1)/(x2-x1)*(y2-y1)
if ((30<=m) && (m<50))
{
    tol = 1e-2 + (m-30)/(50-30)*(1e-4 - 1e-2);
}
else if ((50<=m) && (m<75))
```

```
{
    tol = 1e-4 + (m-50)/(75-50)*(1e-6 - 1e-4);
}
else if ((75<=m) && (m<100))
{
    tol = 1e-6 + (m-75)/(100-75)*(1e-10 - 1e-6);
}
else if (m>=100)
{
    tol = 1e-10;
}
```

- Plot the error for both schemes against execution time..

For comparision, both schemes are set to have the same converging tolerance $\mathbf{10^{-6}}$. The size of Krylov space in Lanczos method is 75. After the numerical iterations, power method scheme's error is $\mathbf{1.521878 \times 10^{-4}}$ and the other one's is $\mathbf{1.520357 \times 10^{-4}}$. Since those values are close to each other, it is valid that both schemes can be compared to each other.

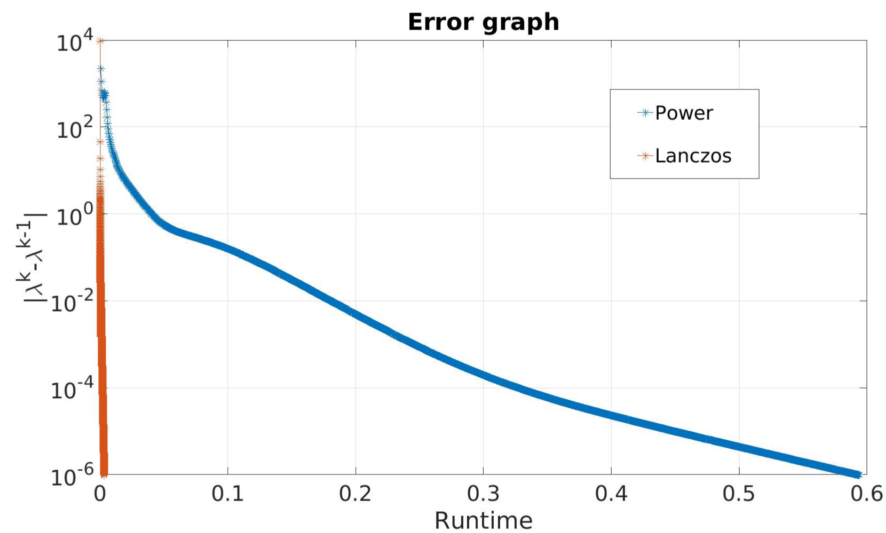

Figure 3: Lanczos method converges much faster than Power method does.

The error graph with respect to runtime on Figure 3 shows that Lanczos method scheme converges incomparably faster than power does. It only takes $\mathbf{2.022 \times 10^{-2}}$ seconds for Lanczos scheme to converge. On the other hand, for Power scheme, **0.5941** seconds are necessary for convergence. The ratio of Krylov space's size with respect to that of real matrix is only $\mathbf{1.4\%, 75/5357}$. This runtime comparision verifies that using small size of Krylov space is precise enough to estimate eigenvalue of a matrix. In conclusion, Lanczos much faster with the same precision for estimating the largest eigenvalue.