MAT 226B Large Scale Matrix Computation
Final Project

Ahmed Mahmoud

March, 22nd 2020

Problem 1:

(a)

(b)

(c)

Problem 2:

Here we are required to find an efficient way to compute $q = Mv$ and $q = M^T v$ for $v \in \mathbb{C}^n$ where $M = (A - s_0E)^{-1}E$. We can compute the matrix-vector multiplication efficiently using LU factorization. We first can write the multiplication as

$$
\begin{aligned}
&q = (A - s_0E)^{-1}Ev = (\underbrace{A - s_0E}_{W})^{-1}\underbrace{Ev}_{f} \\
&q = W^{-1}f \quad \Rightarrow \quad Wq = f \quad \Rightarrow \quad \underbrace{PD^{-1}WQ}_{LU}\underbrace{Q^Tq}_{d} = PD^{-1}f
\end{aligned}
$$

Thus, we can fist solve $Lc = PD^{-1}f$ for $c \in \mathbb{C}^n$ via forward substitution, then solve $Ud = c$ for $d \in \mathbb{C}^n$ via backward substitution, and finally set $q = Qd$.

We can use the same LU factorization to compute $q = M^T v$ efficiently. We first not that transposing the LU factorization for a given matrix W is $U^T L^T = Q^T W^T D^{-T} P^T$ We can write this multiplication as

$$q = ((A - s_0E)^{-1}E)^T v = E^T \underbrace{(A - s_0E)^{-T} v}_{g}$$

$$g = W^{-T} v \quad \Rightarrow \quad W^T g = v \quad \Rightarrow \quad \underbrace{Q^T W^T D^{-T} P^T}_{U^T L^T} \underbrace{(D^{-T} P^T)^{-1} g}_{d} = Q^T v$$

Thus, we can first solve $U^T c = Q^T v$ for c via forward substitution, then solve $L^T d = c$ for d via backward substitution, and then set $g = D^{-T} P^T d$. Finally, we multiply g from the left by E^T to get q. The functions `Mv` and `transposeMv` implements these operations as discussed. d^o

Problem 3:

The leading $2k$ moments $\mu_j = c^T M^j r$ for $j = 0, 1, \ldots, 2k - 1$ can be computing as follows. Let $f_j = M^j r$. It is easy to see that $f_j = M f_{j-1}$ from which we can compute the moment at j as $\mu_j = c^T f_j$ and compute f_j recursively. We can use the same LU factorization to compute r and used the function `Mv` to compute f_j. The function `computeMoments` compute the moments as discussed here.