MATH 8610 (SPRING 2018) HOMEWORK 3

Assigned 02/22/19, due 03/14/19 in class.

Instructor: Dr. Fei Xue, Martin O-203, fxue@clemson.edu.

1. **[Q1] (10 pts)** Trefethen's book, Problem 5.3 (page 37).

2. **[Q2] (10 pts)** (a) If $A \in \mathbb{R}^{m\times n}$ and $E \in \mathbb{R}^{m\times n}$, show that

   $$\sigma_{\max}(A+E) \le \sigma_{\max}(A) + \|E\|_2 \quad \text{and} \quad \sigma_{\max}(A+E) \ge \sigma_{\max}(A) - \|E\|_2.$$

   Comment on the (absolute) condition number of $\|A\|_2$ as a function of A.

   (b) If $A \in \mathbb{R}^{m\times n}$, $m > n$ and $z \in \mathbb{R}^m$, show that

   $$\sigma_{\max}([A\ \ z]) \ge \sigma_{\max}(A) \quad \text{and} \quad \sigma_{\min}([A\ \ z]) \le \sigma_{\min}(A).$$

3. **[Q3] (10 pts)** (a) Show that if $A \in \mathbb{R}^{m\times n}$, then $\|A\|_F \le \sqrt{\text{rank}(A)}\|A\|_2$.

   (b) Show that if $A \in \mathbb{R}^{m\times n}$ has rank n, then $\|A(A^TA)^{-1}A^T\|_2 = 1$.

4. **[Q4]** (a*) (10 points) Given $A \in \mathbb{R}^{n\times n}$, let $A = U\Sigma V^T$ be an SVD of A, where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_n > 0$, and assume that $\det(A) < 0$. Let $B = [U\text{diag}(1, \ldots, 1, -1)]\,\Sigma V^T$ such that $\det(B) = |\det(A)|$ and $\|A - B\|_F = 2\sigma_n$. For *any singular values* $\sigma_1, \sigma_2, \ldots \sigma_{n-1}$ ($\ge \sigma_n$), show that there exists $C \in \mathbb{R}^{n\times n}$ such that $\det(C) = \det(B) = |\det(A)|$, and $\|A - C\|_F < \|A - B\|_F = 2\sigma_n$.

   (Hint: to construct C, modify σ_n and σ_{n-1} of A only (change the sign of one and keep the sign of the other, but make sure that their product does not change))

   (b) (5 points) Trefethen's book, Problem 6.1.

   (c*) (10 points) Trefethen's book, Problem 6.5.

5. **[Q5] (10 pts)** Read the introduction to the Golub-Kahan-Lanczos (GKL) method, `http://www.netlib.org/utk/people/JackDongarra/etemplates/node198.html` and the uploaded code `HW3_GKLsvds.m`.

   (a) Give a general description of the functionality of GKL; describe the main difference between the original GKL and the code.

   (b) Download the zipped file `HW3_pics.zip`, unzip it, load `picA.mat` file, and run

```
rk = 160;
tic; [Us1,Ss1,Vs1] = HW3_GKLsvds(pic_A(:,:,1),rk); toc;
tic; [Us2,Ss2,Vs2] = HW3_GKLsvds(pic_A(:,:,2),rk); toc;
tic; [Us3,Ss3,Vs3] = HW3_GKLsvds(pic_A(:,:,3),rk); toc;
tic; [U1,S1,V1] = svd(pic_A(:,:,1),0); toc;
tic; [U2,S2,V2] = svd(pic_A(:,:,2),0); toc;
tic; [U3,S3,V3] = svd(pic_A(:,:,3),0); toc;
```

   Then, run MATLAB's command `whos` to see the memory used by `pic_A`, and by `Us1,Vs1,Us2,Vs2,Us3` and `Vs3` all together. Compare the timing used for computing and the memory used for storing the full and partial SVD of this picture.

(*Note that we are competing MATLAB code with the built-in C/FORTRAN* code *in timing, and our timing may improve significantly if our GKL is in C/FORTRAN*)

(c) Finally, run MATLAB's command

```
Ahat = zeros(size(pic_A));
Ahat(:,:,1) = Us1*Ss1*Vs1';
Ahat(:,:,2) = Us2*Ss2*Vs2';
Ahat(:,:,3) = Us3*Ss3*Vs3';
disp([norm(Ahat(:,:,1)-pic_A(:,:,1),'fro')/norm(pic_A(:,:,1),'fro') ...
norm(Ahat(:,:,2)-pic_A(:,:,2),'fro')/norm(pic_A(:,:,2),'fro') ...
norm(Ahat(:,:,3)-pic_A(:,:,3),'fro')/norm(pic_A(:,:,3),'fro')]);

figure(1);  image(pic_A);  axis equal;
figure(2);  image(Ahat);  axis equal;
```

Are you satisfied with the quality of the image generated by `Ahat`? If not, let `rk = 320`, rerun `HW3_GKLsvds`, compare the timing and memory cost for computing the partial SVD. Then show the images again. Are you satisfied now?

Repeat the above procedure for the other three pictures. Make some general comments on the computation and use of partial SVD for compressing images.

(d) (**5 extra credit for fun, only for those who finished (a)-(c)**). State the title of each artwork, the name of the artist, the approximate year of creation, and the current location of the artwork. Info of 3 paintings qualifies full extra credit.

6. **[Q6]** (10 pts) (a) Implement the Golub-Kahan (GK) bidiagonalization of a matrix. Test it on $F \in \mathbb{R}^{10\times 10}$ obtained as follows

```
rgn('default');
F = randn(10,10);
```

Make sure that your bidiagonal matrix has the same singular values as F.

(b) Generate a matrix $A \in \mathbb{R}^{(1024^2+1)\times 32}$ as follows

```
col = linspace(-1,1,1024*1024+1)';
A = col.^(0:31);
```

Apply Householder QR to A and get $R \in \mathbb{R}^{32\times 32}$, then apply GK to R and get bidiagonal $B \in \mathbb{R}^{32\times 32}$ (no need to retrieve Q for this problem). Compute the 5 largest and 5 smallest singular values of A from the eigenvalues of $\begin{bmatrix} 0 & B^T \\ B & 0 \end{bmatrix}$. Compare these singular values with those computed by taking the square root of the 5 largest and 5 smallest eigenvalues of A^TA. What conclusion do you draw? Is it a good idea to compute the eigenvalues of $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$ directly, and why?

7. **[Q7*]** (10 extra pts) Review our analysis of the bound on the relative forward error of singular value computation by using a backward stable eigenvalue algorithm for A^TA.

That is, $\frac{|\widetilde{\sigma}_k - \sigma_k|}{\sigma_k} \leq \mathcal{O}\left(\frac{\sigma_1^2}{\sigma_k^2}\epsilon_{mach}\right)$, where $\widetilde{\sigma}_k = \sqrt{\widetilde{\lambda}_k}$ and $\widetilde{\lambda}_k$ denotes the computed k-th largest eigenvalue of A^TA. Show that the relative forward error of singular value computation would be bounded by $\mathcal{O}\left(\frac{\sigma_1}{\sigma_k}\epsilon_{mach}\right)$, if we use a backward stable eigenvalue algorithm for $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$. Explain the virtue of the new error bound.