Homework set 6 — CSE 383C / CS 383C / M 383E / ME 397, Fall 2024

Hand in solutions to: 23.1, 24.1, 25.3. from the book.

Optional (do not hand in): 24.2 (useful result!), 25.2, 27.2, 27.4.

Problem 1: Let $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and set $r(\mathbf{x}) = \frac{\mathbf{x}^*\mathbf{A}\mathbf{x}}{\|\mathbf{x}\|^2}$.

(a) Prove that $\frac{\partial r}{\partial x_1} = \|\mathbf{x}\|^{-2}\big(2a_{11}x_1 + a_{12}x_2 + a_{21}x_2 - 2x_1 r(\mathbf{x})\big)$.

(b) Prove that $\nabla r(\mathbf{x}) = \|\mathbf{x}\|^{-2}\big(\mathbf{A}\mathbf{x} + \mathbf{A}^*\mathbf{x} - 2r(\mathbf{x})\,\mathbf{x}\big)$.

(c) Prove that if $\mathbf{A}$ is symmetric, and $\mathbf{v}$ is an eigenvector of $\mathbf{A}$, then $\nabla r(\mathbf{x}) = \mathbf{0}$.

(d) *[Optional!]* Prove (b) and (c) in the case of an $m \times m$ matrix.

Problem 2 (optional – do not hand in): Let $\mathbf{A}$ be an $m \times m$ real matrix that is symmetric. Fix a positive integer p such that $p < m$, and let $\mathbf{B}_0$ be an $m \times p$ matrix that forms the starting point for a block power iteration. In other words, define matrices $\mathbf{B}_k$ via

$$\mathbf{B}_k = \mathbf{A}\mathbf{B}_{k-1}, \qquad k = 1, 2, 3, \dots$$

Suppose the eigenvalues $\{\lambda_j\}_{j=1}^m$ of $\mathbf{A}$ are ordered so that $|\lambda_1| \geqslant |\lambda_2| \geqslant |\lambda_3| \geqslant \cdots$, and that $|\lambda_p| > |\lambda_{p+1}|$. Then the column space of $\mathbf{B}_k$ will "converge" to the space spanned by the first p eigenvectors of $\mathbf{A}$. (We use the term "converge" loosely, since we have not defined any notion of a distance between subspaces.) However, the columns of $\mathbf{B}_k$ will typically form a very ill-conditioned basis for this space. A standard fix to this problem is to orthonormalize the columns between each iteration:

$\mathbf{Q}_0 = \mathbf{B}_0$
for $k = 1, 2, 3, \dots$
  $\mathbf{C}_k = \mathbf{A}\mathbf{Q}_{k-1}$
  $[\mathbf{Q}_k, \sim] = \mathtt{qr}(\mathbf{C}_k, 0)$
end for

(a) (Optional) Run the function `hw6p1.m` to see how well block power iteration approximates the eigenvalues of $\mathbf{A}$. Observe that the larger eigenvalues seem to converge faster. Can you explain why?

(b) Prove that the column space of the matrix $\mathbf{Q}_k$ resulting from the numerically stable method equals the column space of $\mathbf{B}_k$ computed by the "pure" power iteration when exact arithmetic is used. You may assume that each matrix $\mathbf{B}_k$ and $\mathbf{C}_k$ has full rank. *Hint:* If $\mathbf{X} \in \mathbb{R}^{m \times p}$ is a matrix of rank p, and if $\mathbf{Y} \in \mathbb{R}^{p \times p}$ is an invertible matrix, then $\mathbf{X}$ and $\mathbf{XY}$ have the same column space.

(c) (Optional) The function `hw6p1_extra.m` compares the eigenvalues that result when block power iteration is used. Run and see if the results match what you would expect.

Problem 3 (optional – do not hand in): The code snippet below is from the file `hw6p2.m`:

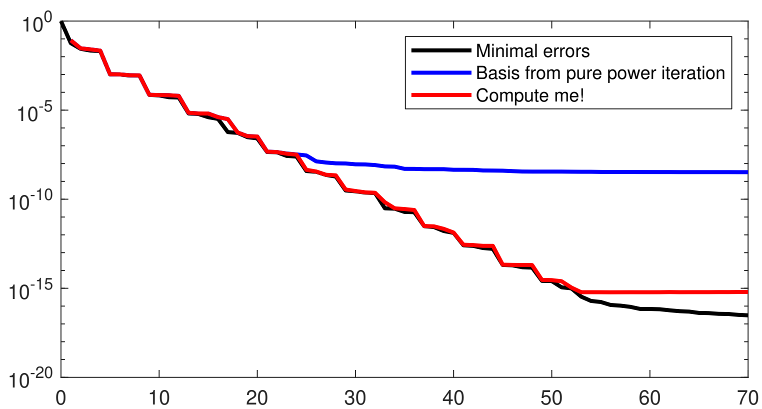
```
%%% Run power iteration to build an ON basis Q1 for the range of A.
Y = randn(m,b);
for k = 1:q
  Y = A*Y;
end
[Q1,~] = qr(Y,0);

%%% Compute the norm of the difference between A and the projection of A
%%% onto the first k columns of Q1.
E = zeros(1,b);
for k = 1:b
  E(k) = norm(A - Q1(:,1:k)*Q1(:,1:k)'*A);
end
```

The matrix $\mathbf{A}$ is symmetric, and the code computes an ON matrix $\mathbf{Q}$ whose first k columns approximately span the space spanned by the k dominant eigenvectors. The code also computes a vector E that measures how well the first k columns of $\mathbf{Q}$ span the column space of $\mathbf{A}$. To be precise,

$$E_k = \|\mathbf{A} - \mathbf{Q}(:,1:k)\mathbf{Q}(:,1:k)^*\mathbf{A}\|.$$

When the code is run for a matrix $\mathbf{A}$ whose eigenvalues decay rapidly, the error E_k is the blue line in the graph shown below:



The black line shows the minimal error that results when $\mathbf{A}$ is approximated by a matrix of rank k (this equals σ_{k+1}). We do great until accuracy 10^{-8} is reached, and then we do terribly after that.

(a) Fix the code so that you get the red line instead of the blue line. Hint: Use what you learned in Problem 2!

(b) The parameter q in the code specifies how many steps of power iteration is taken. If you play around with different values of q, you will see that the basic version of the code loses accuracy at around $(1/\epsilon_{\mathrm{mach}})^{q-1}$. Try to explain this.

Problem 4 (optional – do not hand in): The file hw6p3.m implements the so called *Jacobi method*, which is an iterative technique for driving a symmetric matrix $\mathbf{A}$ to a diagonal matrix $\mathbf{T}$ through a sequence of similarity transforms. The essential piece of code is:

```
T = A;
for k = 1:niter
  for i = 1:(m-1)
    for j = (i+1):m
      th = 0.5*atan(2*T(i,j)/(T(j,j)-T(i,i)));
      J  = [cos(th),sin(th);-sin(th),cos(th)];
      T([i,j],:) = J'*T([i,j],:);
      T(:,[i,j]) = T(:,[i,j])*J;
    end
  end
end
```

(a) Read Lecture 30 in the book to learn about the Jacobi method.

(b) To monitor the convergence of the Jacobi method, let us define an "error"

$$E = \texttt{sqrt(sum(sum(triu(T,1).\^{}2)))}$$

that measures how much mass is left off the diagonal. (To be precise, E is the Frobenius norm of the strictly upper triangular part of $\mathbf{T}$.) Edit the code that you are given so that it computes E_k as a function of k. Hand in a plot of E_k versus k for some matrix of your choice (specify the matrix). You will probably want to use semilogy for the plot.

(c) Repeat problem (b), but now for the basic QR iteration. In other words:

$$\begin{array}{l} \mathbf{T}_0 = \mathbf{A} \\ \textbf{for } k = 1,2,3,\dots \\ \quad [\mathbf{Q},\mathbf{R}] = \texttt{qr}\,\mathbf{T}_{k-1} \\ \quad \mathbf{T}_k = \mathbf{RQ} \\ \textbf{end for} \end{array}$$

Note: There is a more numerically stable formula for computing $\mathbf{J}$:

$$\begin{aligned} \beta &= (T(j,j) - T(i,i))/(2T(i,j)), \\ t &= \text{sign}(\beta)/(|\beta| + \sqrt{\beta^2+1}), \\ c &= 1/\sqrt{t^2+1}; \\ J &= \begin{bmatrix} c & ct \\ -ct & c \end{bmatrix}. \end{aligned}$$

P.G. Martinsson, UT-Austin, Nov. 2024