Homework III

Due on March 13, 2023

1. Let R be an upper triangular matrix $n \times n$, and u, v be two column vectors of length n. Show how to compute the QR factorisation of the matrix $R + uv^T$ in $O(n^2)$ time (**Hint**: first work on the second matrix in the sum and reduce the overall matrix to a Hessenberg matrix (as in Minor 1 exam)).

   Let T denote the matrix uv^T. Note that each column of T is a multiple of u. Notice that if H is a Householder transformation which maps a vector v to a multiple of e_1 (recall that e_1 is $(1, 0, \ldots, 0)$), then H will map every multiple of v also to a multiple of e_1 (because it is a reflection transformation).

   Let T^1 be the first column of T. Let H_n be the Householder transformation which does not affect the first $n-2$ coordinate of T^1 and maps the last two coordinates (a, b) of T^1 to a vector of the form $(\star, 0)$. By the above observation it follows that $T' := H_n \cdot T$ will have the property that the last row will be 0. Notice that $T' = H_n T = (H_n u)v^T$, i.e., again all columns of T' will be multiples of each other. Next time we apply H_{n-1} which affects rows $n-2, n-1$ of the matrix T' such that the last two rows become 0, and it remains a matrix for the form $u'v^T$ for some vector u'. Applying this repeatedly, we see that we can find a sequence of Householder transformations $H_n, H_{n-1}, \ldots, H_2$ such that if H denotes $H_2 H_3 \ldots H_n$, then HT has all entries 0 except for those in the first row. Now let us see what H does to the matrix R. When we perform $H_n R$, only the last 2 rows are affected. Since all these entries are 0 except for the last two columns, it follows that $H_n R$ will still be upper triangular except that the entry at coordinate $(n, n-1)$ will become non-zero. Repeating this argument, it follows that the only non-zero entries in $H \cdot R$ will be in the upper triangular region and in the sub-diagonal below the diagonal of this matrix, i.e., $(HR)_{ij}$ will be 0 if $i \leq j - 2$. Such a matrix is called a Hessenberg matrix. Thus we see that $H(R + uv^T)$ is Hessenberg. Notice that the time taken for performing all this operations is $O(n^2)$ (check).

   Now it was an exam question (Minor 1) that QR factorization of a Hessenberg matrix can be computed in $O(n^2)$ time. Thus, we get $H(R + uv^T) = QR'$. So, $R + uv^T = H^{-1}QR' = H_n \ldots H_2 QR'$ (recall that the inverse of a Householder matrix is the matrix itself). Note that H_i affects only two rows of a matrix, and so, we can compute $H_n \ldots H_2 Q$ iteratively in $O(n^2)$ time.

2. In this exercise, you will prove stability of QR factorisation. You can hide factors depending on m and n in the stability calculations. Let ε denote the machine precision.

   - Let $x = (x_1, \ldots, x_m)$ be a vector and consider the Householder transformation corresponding to it, i.e., let v be the unit vector along $x + \texttt{sign}(x_1)||x||e_1$. Suppose we compute a unit vector $\tilde{v}$ instead. Show that $||\tilde{v} - v|| = O(\varepsilon)$.

Assume without loss of generality that $x_1 > 0$. So we first compute $(x_1 + ||x||, x_2, \ldots, x_n)$, and then normalize it to get v. This computation takes several steps (all the ε_i terms here are assumed to be $O(\varepsilon)$, where ε is the machine precision, and we hide factors depending on n in the $O()$ notation):

- When we compute $x_1^2 + x_2^2 + \ldots + x_n^2$, the result is a number $x_1^2(1+\varepsilon_1) + \ldots + x_n^2(1+\varepsilon_n)$. This number lies in the range $[||x||^2(1-n\varepsilon), ||x||^2(1+n\varepsilon)]$. When we compute its square root, we will incur another multiplicative error of $(1+\varepsilon')$. Thus the compute value is $||x||(1+\varepsilon'')$.
- Let x_1' denote $x_1 + ||x||$. When we compute x_1', the computed value would be $(1+\varepsilon''')(x_1+||x||(1+\varepsilon''))$, which can be written as $x_1'(1+\tilde{\varepsilon})$, for some $\tilde{\varepsilon} = O(\varepsilon)$ (note that we are critically using the fact that $x_1 > 0$). If w denotes the vector $(x_1', x_2, \ldots, x_n)$, we have formed the vector $\tilde{w} = (x_1'(1+\tilde{\varepsilon}), x_2, \ldots, x_n)$.
- Now we will compute the norm of $\tilde{w}$. As in the first part above, the compute value would be $||\tilde{w}||(1+\varepsilon_1)$. It is also easy to check that $||\tilde{w}|| = ||w||(1+\varepsilon_2)$. Thus, we see that the computed value of $||\tilde{w}||$ is $||w||(1+\varepsilon_3)$.
- Finally, the computed $\tilde{v}$ is $\frac{\tilde{w}}{||w||(1+\varepsilon_3)} = \frac{(1-\varepsilon_4)(x_1'(1+\tilde{\varepsilon}), x_2, \ldots, x_n)}{||w||}$, where v is $(x_1', x_2, \ldots, x_n)/||w||$. So we see that $||v - \tilde{v}|| = O(\varepsilon)||v|| = O(\varepsilon)$.

- Let $x, v, \tilde{v}$ be as above, and let y be a vector of length m. We would like to compute $b = y - 2 < v, y > v$, but we use $\tilde{v}$ instead of v here, and end up computing a vector $\tilde{b}$ instead. Prove that $\tilde{b} = \tilde{y} - 2 < v, \tilde{y} >$, for some $\tilde{y}$ such that $||\tilde{y} - y|| = O(\varepsilon) \cdot ||y||$.

  Let A denote the (unitary) matrix $I - 2v^T v$. Note that $b = Ay$, whereas our algorithm tries to compute $\tilde{A}y$, where $\tilde{A} = I - 2\tilde{v}\tilde{v}^T$. Using Assignment 2, we know that the computed value of $\tilde{A}y$ would be $(\tilde{A} + \Delta\tilde{A})(y + \delta y)$, where $||\Delta\tilde{A}||/||\tilde{A}||, ||\delta y||/||y||$ are $O(\varepsilon)$. Note that $||\tilde{A}||$ is 1 because $\tilde{A}$ is also unitary. Now, we want to express $\tilde{b} = (\tilde{A} + \Delta\tilde{A})(y + \delta y)$ as $A\tilde{y}$ for some $\tilde{y}$. This is possible if $\tilde{y} = A^{-1}(\tilde{A} + \Delta\tilde{A})(y + \delta y)$. Since A is Householder matrix, $A^{-1} = A$. Also, note that if δv denotes $\tilde{v} - v$, then $\tilde{A} = A + 2(\delta v)(\delta v)^T - 2v(\delta v)^T - 2(\delta v)v^T$. Substituting this in the expression for $\tilde{y}$ above we see that $\tilde{y} = (I + \Delta)(y + \delta y)$, where Δ is a matrix of norm $O(\varepsilon)$. It follows that $||\tilde{y} - y||/||y||$ is $O(\varepsilon)$.

- Suppose our algorithm computes QR factorization of an $m \times n$ matrix A as $\tilde{Q}, \tilde{R}$, where $\tilde{Q}$ is implicitly represented by the vectors corresponding to the Householder transformation. Then $\tilde{Q} \cdot \tilde{R} = A + \Delta A$, where $||\Delta A||/||A||$ is $O(\varepsilon)$.

  We have shown above that if H is householder matrix corresponding to QR factorization and y is a row of the matrix A (whose QR factorization is being computed), then the computed value of Hy can be written as $H\tilde{y}$, where $||\tilde{y} - y||/||y||$ is $O(\varepsilon)$. So, if H_1 is the Householder matrix corresponding to the first column of A, then the computed value of $H_1 A$ is a matrix which is same as $H_1(A + \Delta A)$, where $||\Delta A||/||A||$ is $O(\varepsilon)$. Continuing this argument, we see that if $H_1, H_2, \ldots, H_n$ are the sequence of Householder matrices, then the computed value of HA, where $H = H_n \ldots H_1$, can be written as $H(A + \Delta A)$, where $||\Delta A||/||A||$ is $O(\varepsilon)$. Notice that HA is supposed to be R and so, $\tilde{R} = H(A + \delta A)$, and so, $A + \Delta A = H\tilde{R}$, where H is same as Q and note that inverse of H is same as H.

3. (**Heath**) To demonstrate how results from the normal equations method and QR factorization can differ numerically, we need a least squares problem that is ill-conditioned and also has a small residual. We can generate such a problem as follows. We will fit a polynomial of degree $n-1$,

$$p_{n-1}(t) = x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1},$$

to m data points (t_i, y_i), $m > n$. We choose $t_i = \frac{i-1}{m-1}$, $i = 1, \ldots, m$, so that the data points are evenly spaced on the interval $[0, 1]$. We will generate the corresponding values y_i by first choosing values for the x_j, say $x_j = 1$, $j = 1, \ldots, n$, and evaluating the polynomial to obtain $y_i = p_{n-1}(t), i = 1, \ldots, m$. We could now see whether we can recover the x_j that we used to generate the y_i, but to make it more interesting, we first randomly perturb the y_i values to simulate the data error. Specifically, we take $y_i = y_i + (2u_i - 1) * \epsilon$, $i = 1, \ldots, m$, where each u_i is a random number uniformly generated from $[0, 1]$, and ϵ is a small positive number. For double precision, use $m = 21, n = 12, \epsilon = 10^{-10}$.

Having generated the data set, we will now compare the two methods for least squares. First, form the system of normal equations, and solve it using Cholesky factorization (you can use MATLAB built-in function). Next solve the least squares using QR factorization implemented in Question 2. Compare the two resulting solutions. For which method is the solution more sensitive to the perturbation we introduced in the data ? Which method comes closer to recovering the x that we used to generate the data ?

4. For an arbitrary $m \times m$ matrix A with complex entries, let $\rho(A)$ denote the largest absolute value of an eigenvalue of A, and $\alpha(A)$ denote the maximum over all eigenvalues λ of A of the real part of λ. Prove the following :

   (a) $\lim_{n\to\infty} ||A^n|| = 0$ if and only if $\rho(A) < 1$.

   One direction of the proof is easy. Suppose $\lim_{n\to\infty} ||A^n|| = 0$. Then we claim that $|\lambda| < 1$ for each eigenvalue of A. Indeed, suppose not, i.e., λ is an eigenvalue with $|\lambda| \geq 1$ and let v be a unit eigenvector corresponding to λ. Then $A^n v = \lambda^n v$ and so $||A^n v||$ does not approach 0 as n goes to infinity.

   The converse is trickier. We cannot assume that A is Hermitian.

   (b) $\lim_{t\to\infty} ||e^{tA}|| = 0$ if and only if $\alpha(A) < 0$. So assume that $\rho(A) < 1$ and we need to show that for any small $\delta > 0$, there exists a large enough N such that $||A^n|| \leq \delta$ for all $n \geq N$.

   By Schur's theorem, we know that $A = XTX^*$ where T is upper triangular and X is unitary. Let T' be an upper triangular matrix obtained from T by slightly perturbing the diagonal entries of T such that all of them are distinct (and assume that these perturbations are at most ε for a small enough ε, we will choose ε to be much smaller than δ). Let A' denote $XT'X^*$. Note that A' is a diagonalizable (since the eigenvalues of A' are same as the diagonal entries of T' and T' has n distinct diagonal entries). So we can write A' as VDV^{-1}, where the D has diagonal entries as eigenvalues of A' and V is the matrix formed by the corresponding eigenvectors of A'. Let $k(V)$ denote $||V|| \cdot ||V^{-1}||$. Since all the eigenvalues of A' have size at most 1,

$$||A'^n|| = ||VD^nV^{-1}|| \leq k(V)||D^n||,$$

which goes to 0 as n goes to infinity. So we can make this quantity as small as we wish. Also,

$$||A - A'|| = ||V(T - T')V^{-1}|| \leq \varepsilon k(V).$$

Now

$$||A^n|| = ||(A - A' + A')^n||.$$

When we expand the RHS, we get 2^n terms, where each term is a product of n matrices $M_1 \dots M_n$, where each M_i is either $A - A'$ or A'. Consider such a product where $A - A'$ appears i times and A' appears $n - i$ times. Then

$$||M_1 \dots M_n|| \leq ||(A - A')||^i ||A'||^{n-i}|| \leq k(V)||D^{n-i}||(\varepsilon k(V))^i.$$

If $i > 1$, then the fact that $||D|| < 1$ implies that the above is at most $(\varepsilon k(V))^i k(V)$. Choose ε such that $\varepsilon k(V) \lll \delta/2^n$ and so the above is at most $\delta/2^n$. Similarly, if $i = 0$ (there is only one such term), we can choose n high enough so that $||D^n|| \leq \delta/2$. Thus the overall sum is at most δ.

5. Suppose A is a real symmetric matrix with one eigenvalue much smaller than the rest in absolute value. Suppose we run the inverse power iteration algorithm with $\mu = 0$. Let $q_1, \dots, q_n$ be n orthonormal eigenvectors of A. Suppose v is a vector which has non-zero components along each of these eigenvectors, and suppose we solve for $Aw = v$ (as in the inverse power iteration algorithm). Suppose we compute a vector $\tilde{w}$ here (using a backward stable algorithm). Show that the vectors $w/||w||$ and $\tilde{w}/||\tilde{w}||$ are close to each other.

Suppose $Aw = v$ and we use a stable algorithm to solve this. Without loss of generality assume that v is a unit vector. This yields a solution w' to a system of equations $(A + \Delta A)w' = v + \Delta v$, where $\frac{||\Delta A||}{||A||}, \frac{||\Delta v||}{||v||}$ are $O(\varepsilon)$. Let the eigenvalues of A be $|\lambda_1| < |\lambda_2|, \dots, |\lambda_n|$, with λ_1 being much closer to 0 than others. Let v_i be the unit eigenvector corresponding to λ_i; note that the v_i's are orthonormal.

Express v as $\sum_i \alpha_i v_i$. Now,

$$w = A^{-1}v = \sum_i \frac{\alpha_i v_i}{\lambda_i},$$

since the eignevalues of A^{-1} are $1/\lambda_1, \dots, 1/\lambda_n$ with corresponding eigenvectors being $\lambda_1, \dots, \lambda_n$. When normalized, we get

$$\frac{w}{||w||} = \sum_i \gamma_i v_i, \qquad \frac{\gamma_j}{\gamma_1} = \frac{\lambda_1}{\lambda_j}\frac{\alpha_j}{\alpha_1}. \tag{1}$$

Now,

$$Aw' = v + \Delta v - \Delta A \cdot w',$$

and so

$$w' = A^{-1}v - A^{-1}(\Delta v - \Delta A \cdot w') = \sum_i \frac{\alpha_i}{\lambda_i} v_i - A^{-1}(\Delta v - \Delta A \cdot w').$$

We now bound the length of $\Delta v - \Delta A \cdot w'$. Let λ_n be the largest (in magnitude) eigenvalue of A. Then,

$$||\Delta v - \Delta A \cdot w'|| \leq ||\Delta v|| + ||\Delta A|| \cdot ||w'|| \leq \varepsilon(||v|| + ||A|| ||w'||) \leq \varepsilon(1 + \lambda_n ||w'||) = O(\varepsilon ||w'||),$$

where we have included λ_n inside the order notation. So if $\Delta v - \Delta A \cdot w'$ is expressed as $\sum_i \beta_i v_i$, then each β_i is $O(\varepsilon ||w'||)$. Now

$$A^{-1}(\Delta v - \Delta A \cdot w') = \sum_i \frac{\beta_i}{\lambda_i} v_i.$$

Using this expression above, we get

$$w' = \sum_i \frac{\alpha_i - \beta_i}{\lambda_i} v_i,$$

and so the normalized vector w' can be written as

$$\frac{w'}{||w'||} = \sum_i \gamma'_i v_i, \qquad \frac{\gamma_j}{\gamma_1} = \frac{\lambda_1}{\lambda_j} \frac{\alpha_j - \beta_j}{\alpha_1} = \frac{\lambda_1}{\lambda_j} \frac{\alpha_j + O(\varepsilon ||w'||)}{\alpha_1}.$$

Now observe that $||w'|| = O(1/\lambda_1)$, and so comparing the above with (1), we see that for $j \neq 1$,

$$\gamma'_j - \gamma_j = O(\varepsilon/\lambda_j),$$

which implies that $\frac{w}{||w||} - \frac{w'}{||w'||}$ is $O(\varepsilon)$ (since if two unit vectors have all coordinates except one with ε, then the distance between them is also $O(\varepsilon)$).

6. Let A be a tridiagonal Hermitian matrix with all its sub- and super-diagonal entries being non-zero. Prove that the eigenvalues of A are distinct.

7. Let A be a square matrix, which is not necessarily Hermitian. Prove that a complex number z is a Rayleigh quotient of A if and only if it is a diagonal entry of $Q^\star AQ$ for some unitary matrix Q.

8. Experiment with solving 60×60 systems of equations $Ax = b$ by Gaussian elimination with partial pivoting, with A having the form

$$\begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

Do you observe that the results have high error because of the growth factor of the order of 2^{60} ? At first attempt, you may not observe this, because the integer entries of A may prevent any rounding errors from occuring. If so, find a way to modify the problem slightly so that the growth factor is the same or nearly so and catastrophic rounding errors really do take place.

9. **(from HW2)** Consider the 4×3 matrix A given below, with $\mu = O(\sqrt{\epsilon_m})$ (where ϵ_m denotes the machine precision):

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \mu & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \mu \end{bmatrix}$$

Suppose its QR factorization is computed in floating-point arithmetic with (a)the classical Gram-Schmidt method, and (b)the modified Gram-Schmidt method. For both cases, work out on paper the columns $\tilde{q}_1, \tilde{q}_2, \tilde{q}_3$ and $\tilde{r}_1, \tilde{r}_2, \tilde{r}_3$ of the computed factorization, along with bounds on the errors of all the entries. For example,

$$\tilde{q}_1 = \begin{bmatrix} 1 + O(\epsilon_m) \\ \mu + O(\epsilon_m) \\ 0 \\ 0 \end{bmatrix}.$$

Assume that addition of 0 and multiplication by 0 or 1 do not incur any rounding error. What can you conclude about the orthogonality of the computed Q in both cases?

We use ε to denote ϵ_m. Let A_i denote column i of A. We considered the Gram Schmidt process first. The computed length of the first column of A is $(1+\varepsilon)(1+\mu^2)^{1/2} = 1 + O(\varepsilon)$. Therefore, compute q_1 is $\begin{bmatrix} 1 + O(\varepsilon) \\ \mu(1 + O(\varepsilon)) \\ 0 \\ 0 \end{bmatrix}$. Now q_2 before normalization is $A_2 - \langle A_2, q_1 \rangle q_1$, which can be written as

$$\begin{bmatrix} 1 \\ 0 \\ \mu \\ 0 \end{bmatrix} - q_1 = \begin{bmatrix} O(\varepsilon) \\ -\mu(1 + O(\varepsilon)) \\ \mu \\ 0 \end{bmatrix}.$$

Note that there is no error in third coordinate because it is just negation. Now the length of this vector is

$$||q_2||^2 = 2\mu^2(1 + O(\varepsilon)).$$

Therefore $||q_2|| = \sqrt{2}\mu(1 + O(\varepsilon))$. So if we divide the above vector by this length, we see that q_2 will turn out to be

$$q_2 = \begin{bmatrix} O(\mu) \\ -\frac{1}{\sqrt{2}}(1 + O(\varepsilon)) \\ \frac{1}{\sqrt{2}}(1 + O(\varepsilon)) \\ 0 \end{bmatrix}.$$

Now q_3 is computed as $A_3 - \langle A_3, q_2\rangle - \langle A_3, q_1\rangle q_1$. After simplification and accounting for $(1+\varepsilon)$-multiplicative error this will turn out to be

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ \mu \end{bmatrix} - (1+O(\varepsilon))\begin{bmatrix} 1+O(\varepsilon) \\ \mu(1+O(\varepsilon)) \\ 0 \\ 0 \end{bmatrix} - O(\mu)\begin{bmatrix} O(\mu) \\ -\frac{1}{\sqrt{2}}(1+O(\varepsilon)) \\ \frac{1}{\sqrt{2}}(1+O(\varepsilon)) \\ 0 \end{bmatrix} = \begin{bmatrix} O(\varepsilon) \\ O(\mu) \\ O(\mu) \\ \mu \end{bmatrix}.$$

The (computed) length of this vector is $O(\mu)$, and so the normalized q_3 vector is $\begin{bmatrix} O(\mu) \\ O(1) \\ O(1) \\ O(1) \end{bmatrix}$. It can be easily seen that q_2 and q_3 are far from being orthogonal. We now come to modified Gram Schmidt. The computation of q_2 happens as before. But before computing q_3, we now modify A_3 to $A_3' = A_3 - \langle A_3, q_1\rangle$, which turns out to be

$$\begin{bmatrix} O(\varepsilon) \\ -\mu(1+O(\varepsilon)) \\ 0 \\ \mu \end{bmatrix}.$$

We now compute $A_3' - \langle A_3', q_2\rangle q_2 = A_3' - \frac{\mu(1+O(\varepsilon))}{\sqrt{2}} q_2$, which turns out to be

$$\begin{bmatrix} O(\varepsilon) \\ \frac{\mu(1+\varepsilon)}{\sqrt{2}} \\ -\frac{\mu(1+\varepsilon)}{\sqrt{2}} \\ \mu \end{bmatrix}.$$

The computed length of this vector turns out to be $\sqrt{2}\mu(1+\varepsilon)$, and so the normalized vector q_3' is

$$\begin{bmatrix} O(\mu) \\ \frac{1+O(\varepsilon)}{\sqrt{2}} \\ -\frac{1+O(\varepsilon)}{\sqrt{2}} \\ \frac{1+O(\varepsilon)}{\sqrt{2}} \end{bmatrix}.$$

It can be seen that q_2' and q_3' are nearly orthogonal.