Prof. Ming Gu, 861 Evans, tel: 2-3145
Office Hours: TuWTh 12:00-1:30PM
Email: mgu@math.berkeley.edu
http://www.math.berkeley.edu/~mgu/MA221

Math221: Matrix Computations
Solutions to Homework #6

- **Problem 3.8:** P and Q can never be equal. The determinent of a Householder reflection matrix is always -1. On the other hand, since the determinent of a Givens rotation matrix is always 1, the determinent of Q, the product of $n-1$ Givens rotations is, still 1.

- **Problems 3.12:** Let

$$A = U_A \begin{pmatrix} \Sigma_A & 0 \\ 0 & 0 \end{pmatrix} V_A^T \quad \text{and} \quad B = U_B \begin{pmatrix} \Sigma_B & 0 \\ 0 & 0 \end{pmatrix} V_B^T$$

be the full SVDs of A and B, respectively, meaning U_A, V_A, U_B, V_B are all square orthogonal matrices of the right dimensions. Here we have also partitioned the singular values so that $\Sigma_A > 0$ and $\Sigma_B > 0$. Define and partition

$$\begin{aligned} \widehat{X} &= V_A^T X U_B = \begin{pmatrix} \widehat{X}_{1,1} & \widehat{X}_{1,2} \\ \widehat{X}_{2,1} & \widehat{X}_{2,2} \end{pmatrix}, \\ \widehat{C} &= U_A^T C V_B = \begin{pmatrix} \widehat{C}_{1,1} & \widehat{C}_{1,2} \\ \widehat{C}_{2,1} & \widehat{C}_{2,2} \end{pmatrix}, \end{aligned}$$

where the partitions are done according to the dimensions of Σ_A and Σ_B. Then

$$\begin{aligned} \|AXB - C\|_F &= \| \begin{pmatrix} \Sigma_A & 0 \\ 0 & 0 \end{pmatrix} \widehat{X} \begin{pmatrix} \Sigma_B & 0 \\ 0 & 0 \end{pmatrix} - \widehat{C} \|_F \\ &= \left\| \begin{pmatrix} \Sigma_A \widehat{X}_{1,1} \Sigma_B - \widehat{C}_{1,1} & -\widehat{C}_{1,2} \\ -\widehat{C}_{2,1} & -\widehat{C}_{2,2} \end{pmatrix} \right\|_2. \end{aligned}$$

So to minimize $\|AXB - C\|_F$ is to solve $\Sigma_A \widehat{X}_{1,1} \Sigma_B - \widehat{C}_{1,1} = 0$. This implies that every minimizer satisfies

$$\widehat{X}_{1,1} = \Sigma_A^{-1} \widehat{C}_{1,1} \Sigma_B^{-1}.$$

The minimum norm one is the one with all other parts to be zero. Hence

$$X_0 = V_A \begin{pmatrix} \Sigma_A^{-1} \widehat{C}_{1,1} \Sigma_B^{-1} & 0 \\ 0 & 0 \end{pmatrix} U_B^T = A^\dagger C B^\dagger.$$

- **Problem 3.15:** We will only do QR factorization in this solution. Let

$$A^T = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix}$$

be the full QR factorization of A^T, and let

$$\widehat{x} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}^T x = \begin{pmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{pmatrix}.$$

With this notation, we have

$$Ax - b = R^T \widehat{x}_1 - b.$$

To solve the LS problem is to solve $R^T \widehat{x}_1 - b = 0$. The minimum norm solution is

$$x = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} \widehat{x}_1 \\ 0 \end{pmatrix}.$$

- **Problem 3.18:** Let

$$C^T = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix}$$

be the QR factorization of C^T, and let

$$\widehat{x} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}^T x = \begin{pmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{pmatrix}.$$

The constraint now becomes $R^T \widehat{x}_1 = d$ and hence $\widehat{x}_1 = R^{-T} d$.

Define

$$A \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} = \begin{pmatrix} A_1 & A_2 \end{pmatrix}.$$

We rewrite

$$Ax - b = \begin{pmatrix} A_1 & A_2 \end{pmatrix} \begin{pmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{pmatrix} - b = A_2 \widehat{x}_2 - (b - A_1 \widehat{x}_1).$$

This reduces the constrained LS problem into an unconstrained probelm with coefficient matrix A_2. Now let $\widehat{Q}\widehat{R} = A_2$ be the QR factorization of A_2 and solve $\widehat{x}_2 = \widehat{R}^{-1} \widehat{Q}^T (b - A_1 \widehat{x}_1)$. To recover x we compute

$$x = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{pmatrix}.$$

- For any non-zero vector $x = (x_1, \cdots, x_n)^T$, the standard way to compute the Householder transformation is to compute $\widetilde{u} = (x_1 - c, x_2, \cdots, x_n)^T$ with $c = -\mathbf{sign}(x_1)\|x\|_2$ and $u = \widetilde{u}/\|\widetilde{u}\|_2$ so that

$$\left(I - 2uu^T\right) x = (c, 0, \cdots, 0)^T.$$

The special sign of c ensures that $\widetilde{u}$ and u are computed to full relative accuracy.

However, the sign choice in c is actually not necessary. Let $c = \|x\|_2$. Show that $\widetilde{u}$, and hence u, can still be computed to full relative accuracy with a computationally different but mathematically equivalent formula. Perform an error analysis to support your claim. You can assume the square root function is always accurate to full relative accuracy. Write a matlab code to demonstrate that the straightforward formula for computing $\widetilde{u}$ can be unstable and yours is always stable. The matlab code housetest.m on the class website generates vectors that fail the straightforward formula.

Solution: If x_1 is non-positive, then the formula for computing $\widetilde{u}$ is completely stable. In fact, both c and $x_1 - c$ can be computed to high relative accuracy, and so can u.

When x_1 is positive, the picture changes. The rounding errors in computing c could cause very large relative errors in the computed $x_1 - c$. The trick to avoid this problem is to note that $c = \sqrt{x_1^2 + \cdots + x_n^2}$ so that

$$x_1 - c = \frac{x_1^2 - c^2}{x_1 + c} = -\frac{\sum_{k=2}^n x_k^2}{x_1 + c}.$$

Since x_1 is positive, every operation in computing $x_1 - c$ is now accurate to small relative error.

Since we can compute $\widetilde{u}$ to component-wise full relative accuracy in both cases, we can do the same on u. By Lemma 3.1 and Theorem 3.5, the procedure here is backward stable.

- – Let $c^2 + s^2 = 1$ and let $q \in \mathbf{R}^{n-1}$ be a unit vector. Find vectors $r, u, v \in \mathbf{R}^{n-1}$ so that the matrix

  $$Q = \begin{pmatrix} c & r^T \\ sq & I - uv^T \end{pmatrix} \in \mathbf{R}^{n \times n}$$

  is an orthogonal matrix.

  Solution: We want

  $$\begin{pmatrix} c & r^T \\ sq & I - uv^T \end{pmatrix} \begin{pmatrix} c & r^T \\ sq & I - uv^T \end{pmatrix}^T = I. \tag{1}$$

  In particular, we have

  $$s^2 qq^T + \left(I - uv^T\right)\left(I - uv^T\right)^T = I,$$

  which leads to

  $$uv^T + vu^T - v^T v uu^T = s^2 qq^T. \tag{2}$$

  Hence the expression on the left hand side must have rank 1. We claim that u and v must be linearly dependent of each other. Otherwise, let $[uv] = QR$ be the QR factorization of $[u, v]$. Then R is non-singular, and

  $$uv^T + vu^T - v^T v uu^T = QR \begin{pmatrix} 0 & 1 \\ 1 & -v^T v \end{pmatrix} R^T Q^T$$

  is a rank-2 matrix, a contradiction.

Hence u and v are linearly dependent, and thus must be multiples of q as well. Since we are only concerned with the product uv^T, we choose $uv^T = \alpha qq^T$ for some constant α. Equation (2) leads to the equation

$$2\alpha - \alpha^2 = s^2 = 1 - c^2,$$

and $\alpha = 1 \pm c$.

From equation (1), we also have

$$csq^T + r^T \left(I - \alpha qq^T \right) = 0,$$

which means r must be a multiple of q. Let $r = \beta q$, we obtain

$$csq^T + \beta q^T \left(I - \alpha qq^T \right) = 0,$$

which simplifies to

$$csq^T + \beta q^T (1 - \alpha) = 0,$$

Since $\alpha = 1 \pm c$, this relation leads to $\beta = \pm s$, where α and β should choose the same $+$ or $-$ sign. For simplicify, we choose $\alpha = 1 + c$ and $\beta = s$.

- For any non-zero vector x, find a Q matrix of the form above such that $Qx = (\|x\|_2, 0, \cdots, 0)^T$.
- Develop a QR factorization algorithm that is based on the Q matrices, and show that it is stable. Compare the cost of your algorithm with that based on Householder transformations.

  Solution: Let $Qx = (\|x\|_2, 0, \cdots, 0)^T$. This implies $x = Q^T(\|x\|_2, 0, \cdots, 0)^T$. Hence all we have to do is to choose c, s, and q so that the first column of Q^T is in the same direction as x. This is similar to the Householder reflections in elimination. The cost of multiplying Q or Q^T to a matrix is also similar to that of Householder reflections. With these remarks, we do QR factorization using the existing QR procedure, replacing only Householder reflections with our new orthogonal matrices.
- Correctly implement your algorithm in matlab.