Utah State University
ECE 6030
Homework # 5
Due Friday Feb. 24, 2023

- Homework:

  1. The handout on "Knapsacks, Lattices, Gram-Schmidt, and Short Vectors: the LLL Algorithm" contains several problems and a programming exercise. These exercises are due on March 13, 2023.
  2. Problems 2.11-54 (10 pts), 2.12-57 (5 pts) , 2.12-60 (10 pts), 2.13-64 ($\mathbf{v}$ is a vector in $\mathbb{C}^n$) (5 pts), 2.13-65 (5 pts), 2.13-66 (10 pts), 2.13-67 (Determine the vector $\hat{\mathbf{x}}$ in span$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$ nearest to $\mathbf{x}$.) (5 pts), 2.13-69 (10 pts), 2.13-71 (5 pts), 2.13-73 (10 pts) .
  3. 2.15-80. This is re-written to try to clarify things as follows. Let $A = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_n \end{bmatrix} \in \mathbb{R}^{m \times n}$. At the kth step of the Gram-Schmidt algorithm (see eq. (2.25))

     $$\mathbf{e}_k = \mathbf{p}_k - \sum_{i=1}^{k-1} \langle \mathbf{p}_k, \mathbf{q}_i \rangle \mathbf{q}_i,$$

     with $\mathbf{q}_k = \mathbf{e}_k / \|\mathbf{e}_k\|$. We can stack the results of these computations to form

     $$\begin{aligned} A &= \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix} \begin{bmatrix} \|\mathbf{p}_1\| & \langle \mathbf{p}_2, \mathbf{q}_1 \rangle & \langle \mathbf{p}_3, \mathbf{q}_1 \rangle & \langle \mathbf{p}_4, \mathbf{q}_1 \rangle & \cdots & \langle \mathbf{p}_n, \mathbf{q}_1 \rangle \\ 0 & \|\mathbf{e}_2\| & \langle \mathbf{p}_3, \mathbf{q}_2 \rangle & \langle \mathbf{p}_4, \mathbf{q}_2 \rangle & \cdots & \langle \mathbf{p}_n, \mathbf{q}_2 \rangle \\ 0 & 0 & \|\mathbf{e}_3\| & \langle \mathbf{p}_4, \mathbf{q}_3 \rangle & \cdots & \langle \mathbf{p}_n, \mathbf{q}_3 \rangle \\ \vdots & & & & & \\ & & & & \cdots & \|\mathbf{e}_n\| \end{bmatrix} \\ &= QR \end{aligned}$$

     where Q is orthogonal ($Q^T Q = I$) and R is the upper triangular matrix shown.

     The method of doing Gram-Schmidt described in class works theoretically, but has "very poor numerical properties in that there is typically a severe loss of orthogonality among the compute $\mathbf{q}_i$." In this problem you reformulate the computations into an algorithm that has better numerical properties.

     Recall that in the G-S process as we described it, at every step we find a new column $\mathbf{q}_k$ of Q and also a new *column* of R. In this new computation, at each step a new column $\mathbf{q}_k$ and a new *row* of R is obtained.

     Let the ith row of the matrix R be denoted by $\mathbf{r}_i^T$.

(a) (5 pts) Show that

$$A = \sum_{i=1}^{n} \mathbf{q}_i \mathbf{r}_i^T.$$

(b) (5 pts) Let $A^{(k)}$ be an $m \times (n-k+1)$ matrix defined by

$$A - \sum_{i=1}^{k-1} \mathbf{q}_i \mathbf{r}_i^T = \begin{bmatrix} \mathbf{0}_{m\times k-1} & A^{(k)} \end{bmatrix}$$

Show that

$$\begin{bmatrix} \mathbf{0}_{m\times k-1} & A^{(k)} \end{bmatrix} = \sum_{i=k}^{n} \mathbf{q}_i \mathbf{r}_i^T.$$

(c) (10 pts) Write

$$A^{(k)} = \begin{bmatrix} \mathbf{e}_k & B^{(k)} \end{bmatrix}$$

where $B^{(k)}$ is $m \times n - k$. Then $\mathbf{q}_k = \mathbf{e}_k / \|\mathbf{e}_k\|$. (Recall that when $k = 1$, $\mathbf{e}_k = \mathbf{p}_1$, and observe that $A^{(1)} = A$.) Show that

$$\mathbf{q}_k^T B^{(k)} = \begin{bmatrix} r_{k,k+1} & r_{k,k+2} & \cdots & r_{k,n} \end{bmatrix}$$

(d) (10 pts) Explain why we can move to the next step by computing

$$A^{(k+1)} = B^{(k)} - \mathbf{q}_k \begin{bmatrix} r_{k,k+1} & r_{k,k+2} & \dots & r_{k,n} \end{bmatrix}.$$

(Then we set $k \leftarrow k+1$ and return to step (c) above.)

(e) (10 pts) Based on these ideas, code up the modified Gram-Schmidt algorithm in Matlab (or python).