Mario L. Gutierrez Abed
364009832
mlg3843@rit.edu

# Problem Set 4

Numerical Analysis I

01-19-2021

**Problem 1.** *Show that* $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1$ *for all* $\mathbf{x} \in \mathbb{R}^m$*. Moreover, show that for any* $\mathbf{x} \in \mathbb{R}^m$ *and any* $A \in \mathbb{R}^{m\times n}$*, the following inequalities hold:*

$$\|\mathbf{x}\|_1 \leq m\|\mathbf{x}\|_\infty, \tag{1a}$$

$$\|\mathbf{x}\|_2 \leq \sqrt{m}\|\mathbf{x}\|_\infty, \tag{1b}$$

$$\|A\|_\infty \leq \sqrt{n}\|A\|_2, \tag{1c}$$

$$\|A\|_2 \leq \sqrt{m}\|A\|_\infty. \tag{1d}$$

*Solution.* Let $\hat{i} \in [1, m]$ be the index that maximizes $|x_i|$; that is,

$$\|\mathbf{x}\|_\infty = \max_{1\leq i\leq m} |x_i| = |x_{\hat{i}}|.$$

Then,

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{m}|x_i|^2\right)^{1/2} = \left(|x_{\hat{i}}|^2 + \sum_{i\neq\hat{i}}|x_i|^2\right)^{1/2} \geq |x_{\hat{i}}| = \|\mathbf{x}\|_\infty. \qquad \checkmark$$

As for the second inequality, note that

$$\|\mathbf{x}\|_2^2 = \sum_{i=1}^{m}|x_i|^2 \leq \sum_{i=1}^{m}|x_i|^2 + 2\sum_{i\neq j}|x_i||x_j| = \|\mathbf{x}\|_1^2 \quad\Longrightarrow\quad \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1. \qquad \checkmark$$

For Eq. (1a),it is clear that

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{m}|x_i| \leq \sum_{i=1}^{m}|x_{\hat{i}}| = m|x_{\hat{i}}| = m\|\mathbf{x}\|_\infty. \qquad \checkmark$$

Similarly, for Eq. (1b) we have

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{m}|x_i|^2\right)^{1/2} \leq \left(\sum_{i=1}^{m}|x_{\hat{i}}|^2\right)^{1/2} = \left(m|x_{\hat{i}}|^2\right)^{1/2} = \sqrt{m}|x_{\hat{i}}| = \sqrt{m}\|\mathbf{x}\|_\infty. \qquad \checkmark$$

As for the last two inequalities, I'm fairly convinced that there was a typo on the problem and $m$ and $n$ should be switched... Here's my argument: For Eq. (1c),

$$\begin{aligned}
\|A\|_\infty &= \max_{\mathbf{x}\neq 0}\frac{\|A\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} \\
&\leq \max_{\mathbf{x}\neq 0}\frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_\infty} && \text{(Since } \|\cdot\|_\infty \leq \|\cdot\|_2\text{)} \\
&\leq \max_{\mathbf{x}\neq 0}\frac{\|A\mathbf{x}\|_2}{\frac{\|\mathbf{x}\|_2}{\sqrt{m}}} && \text{(By Inequality (1b))} \\
&= \sqrt{m}\max_{\mathbf{x}\neq 0}\frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sqrt{m}\|A\|_2. \qquad \checkmark
\end{aligned}$$

Lastly, for Eq. (1d),

$$\begin{aligned}
\|A\|_2 &= \max_{\mathbf{x}\neq 0}\frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \\
&\leq \sqrt{n}\max_{\mathbf{x}\neq 0}\frac{\|A\mathbf{x}\|_\infty}{\|\mathbf{x}\|_2} && \text{(By Inequality (1b))} \\
&\leq \sqrt{n}\max_{\mathbf{x}\neq 0}\frac{\|A\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} && \text{(Since } \|\cdot\|_\infty \leq \|\cdot\|_2\text{)} \\
&= \sqrt{n}\|A\|_\infty.
\end{aligned}$$

□

**Problem 2.** *For a nonsingular $A \in \mathbb{R}^{m\times m}$ show that the weighted norm $\|\mathbf{x}\|_A = \|A\mathbf{x}\|_p$ is a vector norm.*

*Solution.* In order for $\|\mathbf{x}\|_A$ to be a vector norm, it would have to satisfy the following properties:

**(i)** $\|\mathbf{x}\|_A \geq 0$ for all $\mathbf{x} \in \mathbb{C}^n$, and $\|\mathbf{x}\|_A = 0$ iff $\mathbf{x} = 0$.

*Proof.* We have, for any $\mathbf{x} \in \mathbb{C}^n$,

$$\|\mathbf{x}\|_A = \|A\mathbf{x}\|_p \geq 0$$

since $\|\cdot\|_p \geq 0$. In fact, since $A$ is nonsingular $A\mathbf{x} = 0 \implies \mathbf{x} = 0$; thus $\|\mathbf{x}\|_A = \|A\mathbf{x}\|_p = 0 \iff \mathbf{x} = 0$. $\surd$

**(ii)** $\|\mathbf{x}+\mathbf{y}\|_A \leq \|\mathbf{x}\|_A + \|\mathbf{y}\|_A$, for all $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$.

*Proof.* For any $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$,

$$\begin{aligned} \|\mathbf{x}+\mathbf{y}\|_A &= \|A(\mathbf{x}+\mathbf{y})\|_p \\ &= \|A\mathbf{x} + A\mathbf{y}\|_p && \text{(By linearity of } A\text{)} \\ &\leq \|A\mathbf{x}\|_p + \|A\mathbf{y}\|_p && \text{(By Triangle Inequality of } \|\cdot\|_p\text{)} \\ &= \|\mathbf{x}\|_A + \|\mathbf{y}\|_A. \quad \surd \end{aligned}$$

**(iii)** $\|\alpha\mathbf{x}\|_A = |\alpha|\ \|\mathbf{x}\|_A$ for all $\alpha \in \mathbb{C}$ and $\mathbf{x} \in \mathbb{C}^n$.

*Proof.* For any $\mathbf{x} \in \mathbb{C}^n$ and $\alpha \in \mathbb{C}$, we have

$$\|\alpha\mathbf{x}\|_A = \|A(\alpha\mathbf{x})\|_p = \|\alpha A\mathbf{x}\|_p = |\alpha|\|A\mathbf{x}\|_p = |\alpha|\|\mathbf{x}\|_A,$$

where on the second equality we used linearity of $A$, and on the second-to-last equality we used the property $\|\alpha\mathbf{y}\|_p = |\alpha|\|\mathbf{y}\|_p$. $\surd$

Thus we have shown that $\|\cdot\|_A$ satisfies all properties of a vector norm. □

**Problem 3.** *Determine whether the following expressions define norms on $\mathbb{R}^n$:*

*a)* $\max\{|x_2|, |x_3|, \ldots, |x_n|\}$.

*b)* $\sum_{i=1}^{n} |x_i|^3$.

*c)* $\sum_{i=1}^{n} 2^{-i}|x_i|$.

*Solution to a).* This is not a norm. Firstly, consider the case where $n = 1$; in this case the norm is not even defined. Now, let $n \geq 2$, and assume that $\|\mathbf{x}\| = \max\{|x_2|, |x_3|, \ldots, |x_n|\} = |x_{\hat{i}}| = 0$, where $x_{\hat{i}}$ is the coordinate of maximum absolute value in $\{x_2, \ldots, x_n\}$. This guarantees that every coordinate in $\{x_2, \ldots x_n\} \setminus \{x_{\hat{i}}\}$ is also zero, but it doesn't say anything of $x_1$, which for all we know may very well be nonzero. This violates the norm property that requires $\|\mathbf{x}\| = 0 \iff \mathbf{x} = 0$. □

*Solution to b).* This is not a norm either, as it violates the triangle inequality. That is, it is not true, in general, that

$$\|x+y\| = \sum_{i=1}^{n} |x_i + y_i|^3 \leq \sum_{i=1}^{n} |x_i|^3 + \sum_{i=1}^{n} |y_i|^3.$$

Counterexample: consider $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$, with $\mathbf{x} = \{2, 5\}$ and $\mathbf{y} = \{3, 4\}$. Then

$$|2+3|^3 + |5+4|^3 = 754,$$

while

$$|2|^3 + |5|^3 + |3|^3 + |4|^3 = 224.$$

□

*Solution to c).* This one does check out all the conditions of a norm:

**(i)** It is clear that $\|\mathbf{x}\| \geq 0$, since the norm is a sum of nonnegative terms: we have exponents of a positive integer (2) multiplying the absolute value of the coordinates $x_i$. Now, if $\mathbf{x} = \{x_1, \ldots, x_n\} = \{0, \ldots, 0\}$, it is clear that the sum in *c)* vanishes. On the other hand, if this sum vanishes, it would mean that every single coordinate $x_i$ is zero, because every term in the sum is nonnegative, so there wouldn't be any cancellation of nonnegative coordinates that could yield a zero sum. Hence this proves that $\|\mathbf{x}\| = 0 \iff \mathbf{x} = \{0, \ldots, 0\}$. $\quad\surd$

**(ii)** For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, note that

$$\begin{aligned} \|\mathbf{x}+\mathbf{y}\| &= \sum_{i=1}^{n} \frac{1}{2^i} |x_i + y_i| \\ &\leq \sum_{i=1}^{n} \frac{1}{2^i} \left(|x_i| + |y_i|\right) \\ &= \sum_{i=1}^{n} \frac{1}{2^i} |x_i| + \sum_{i=1}^{n} \frac{1}{2^i} |y_i| \\ &= \|\mathbf{x}\| + \|\mathbf{y}\|. \qquad \surd \end{aligned}$$

**(iii)** For any $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$,

$$\|\alpha\mathbf{x}\| = \sum_{i=1}^{n} \frac{1}{2^i} |\alpha x_i| = \sum_{i=1}^{n} \frac{1}{2^i} \alpha |x_i| = \alpha \sum_{i=1}^{n} \frac{1}{2^i} |x_i| = \alpha \|\mathbf{x}\|. \qquad \surd$$

Thus, since all properties are satisfied, we conclude that this is indeed a norm. □

---

**Problem 4.** *Evaluate the Frobenius matrix norm and the induced* $1$-, $2$-, *and* $\infty$*—norms for the following matrix:*

$$A = \begin{bmatrix} 4 & -2 & 4 \\ -2 & 1 & -2 \\ 4 & -2 & 4 \end{bmatrix}$$

*Solution.* The ***Frobenius matrix norm*** of a matrix $A$ is given by

$$\boxed{\|A\|_F = \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{m} |a_{ij}|^2} = \sqrt{\mathrm{Tr}(A^*A)} = \sqrt{\mathrm{Tr}(AA^*)}} \tag{2}$$

So we can either just sum (the square of the absolute value of) all the entries, or take the trace of $AA^*$ (or, equivalently, of $A^*A$). Let's show both approaches, since this is a relatively small matrix. . . Starting with the first approach,

$$\|A\|_F = \sqrt{4^2 + (-2)^2 + 4^2 + (-2)^2 + 1^2 + (-2)^2 + 4^2 + (-2)^2 + 4^2} = \sqrt{81} = 9. \qquad \surd$$

Now, for the second approach, since we are in the reals we swap $A^* \leftrightarrow A^\top$; moreover, note that $A$ is symmetric, so $A = A^\top \implies AA^\top = A^2$. So we have

$$AA^\top = A^2 = \begin{bmatrix} 4 & -2 & 4 \\ -2 & 1 & -2 \\ 4 & -2 & 4 \end{bmatrix} \begin{bmatrix} 4 & -2 & 4 \\ -2 & 1 & -2 \\ 4 & -2 & 4 \end{bmatrix} = \begin{bmatrix} 36 & -18 & 36 \\ -18 & 9 & -18 \\ 36 & -18 & 36 \end{bmatrix}.$$

Then,

$$\|A\|_F = \sqrt{\mathrm{Tr}(AA^\top)} = \sqrt{36+9+36} = \sqrt{81} = 9. \qquad \surd$$

Now, as for the ***induced $p$-norms***,

$$\boxed{\|A\|_p = \max_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p} = \max_{\|\mathbf{x}\|_p=1} \|A\mathbf{x}\|_p} \tag{3}$$

and we have theorems that demonstrate the following:

For $A \in \mathbb{C}^{m\times n}$,

$$\boxed{\begin{aligned} \|A\|_1 = \max_{\|\mathbf{x}\|_1=1} \|A\mathbf{x}\|_1 &= \max_{1\le j\le n} \sum_{i=1}^{m} |a_{ij}| \\ &= \max_{1\le j\le n} \|\mathbf{c}_j\|_1 \\ &= \text{the largest absolute column sum} \end{aligned}} \tag{4}$$

where $\mathbf{c}_j$ is the $j^{\text{th}}$ column of $A$. Similarly,

$$\boxed{\begin{aligned} \|A\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty &= \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}| \\ &= \max_{1\le i\le m} \|\mathbf{r}_i\|_1 \\ &= \text{the largest absolute row sum} \end{aligned}} \tag{5}$$

where $\mathbf{r}_i$ is the $i^{\text{th}}$ row of $A$.

Moreover, for $A \in \mathbb{R}^{m\times n}$,

$$\boxed{\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2 = \sqrt{|\lambda_{\max}|}} \tag{6}$$

where $\lambda_{\max}$ is the largest eigenvalue of $A^\top A$.

Now, since $A$ is symmetric, Eqs. (4) and (5) demonstrate that $\|A\|_\infty = \|A\|_1$. Moreover, the first and third columns of $A$ are identical, i.e., $\mathbf{c}_1 = \mathbf{c}_3$. Thus,

$$\|A\|_\infty = \|A\|_1 = \max_{1\le j\le n} \|\mathbf{c}_j\|_1 = \|\mathbf{c}_1\|_1 = \|\mathbf{c}_3\|_1 = \sum_{i=1}^{3} |a_{i3}| = |4| + |-2| + |4| = 10. \qquad \surd$$

Lastly, we calculate $\|A\|_2$. Note that

$$AA^\top - \lambda I = \begin{bmatrix} 36 & -18 & 36 \\ -18 & 9 & -18 \\ 36 & -18 & 36 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} 36-\lambda & -18 & 36 \\ -18 & 9-\lambda & -18 \\ 36 & -18 & 36-\lambda \end{bmatrix}.$$

Hence, solving the characteristic polynomial,

$$0 = \det(AA^\top - \lambda I) = -\lambda^3 + 81\lambda^2 \implies \lambda = \{0, 81\}.$$

Thus $\lambda_{\max} = 81$, and therefore

$$\|A\|_2 = \sqrt{|81|} = 9. \qquad \surd$$

□

**Problem 5.** *If $A\mathbf{x} = \mathbf{b}$ and $A\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ then show that*

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \kappa(A)\frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|}.$$

*Show that for every nonsingular $A$, this inequality will become an equality for some vectors $\mathbf{b}$ and $\tilde{\mathbf{b}}$. (Of course, we want $\mathbf{b} \neq \mathbf{0}$ and $\mathbf{b} \neq \tilde{\mathbf{b}}$).*

*Proof.* Expanding the LHS,

$$\begin{aligned}
\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} &= \frac{\|A^{-1}\mathbf{b} - A^{-1}\tilde{\mathbf{b}}\|}{\|\mathbf{x}\|} \\
&= \frac{\|A^{-1}(\mathbf{b} - \tilde{\mathbf{b}})\|}{\|\mathbf{x}\|} && \text{(By linearity of } A^{-1}\text{)} \\
&\leq \frac{\|A^{-1}\| \, \|(\mathbf{b} - \tilde{\mathbf{b}})\|}{\|\mathbf{x}\|} \times \frac{\|A\|}{\|A\|} && \text{(By Cauchy-Schwarz)} \\
&= \overbrace{\|A\|\|A^{-1}\|}^{\equiv \kappa(A)} \frac{\|(\mathbf{b} - \tilde{\mathbf{b}})\|}{\|A\|\|\mathbf{x}\|} \\
&\leq \kappa(A) \frac{\|(\mathbf{b} - \tilde{\mathbf{b}})\|}{\|A\mathbf{x}\|} && \text{(By Cauchy-Schwarz)} \\
&= \kappa(A)\frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|}.
\end{aligned}$$

It is clear that for vectors $\mathbf{b}, \tilde{\mathbf{b}}$ that satisfy $\|A^{-1}(\mathbf{b} - \tilde{\mathbf{b}})\| = \|A^{-1}\| \, \|(\mathbf{b} - \tilde{\mathbf{b}})\|$ the above inequality becomes equality. □

**Problem 6.** *Let $A$ be an $n \times n$ matrix with $\|A\| < 1$. Then show that $(I + A)$ is invertible and*

$$\|(I + A)^{-1}\| \leq \frac{1}{1 - \|A\|}. \tag{7}$$

*Proof.* We first show invertibility. Assume, to the contrary, that $I + A$ is singular; then

$$\det(I + A) = 0 \implies \exists \mathbf{x} \neq 0 \,|\, (I + A)\mathbf{x} = 0 \implies \|\mathbf{x}\| = \|A\mathbf{x}\|.$$

But then, from Eq. (3) we have

$$\|\mathbf{x}\| = \|A\mathbf{x}\| \implies \|A\| = \frac{\|\mathbf{x}\|}{\|\mathbf{x}\|} \implies \|A\| = 1,$$

which contradicts the assumption that $\|A\| < 1$. $(\Rightarrow\Leftarrow)$

Now, to show (7), we note that the RHS is a geometric sum; thus we consider expanding the following expression:

$$\begin{aligned}
(I + A)\sum_{i=0}^{N} A^i &= \sum_{i=0}^{N}\left(A^i + A^{i+1}\right) \\
&= \underbrace{A^0}_{=I} + A^1 + A^1 + A^2 + A^2 + A^3 + \ldots \\
&= I + 2\sum_{i=1}^{N} A^i.
\end{aligned} \tag{8}$$

Now, from applying Cauchy-Schwarz we get that

$$\|A^i\| \leq \|A\|^i. \tag{9}$$

But then, since we are assuming that $\|A\| < 1$,

$$\lim_{i\to\infty}\|A^i\| \leq \lim_{i\to\infty}\|A\|^i = \lim_{i\to\infty}\overbrace{\underbrace{\|A\|}_{<1}\cdots\underbrace{\|A\|}_{<1}}^{i\text{ times}} \to 0. \tag{10}$$

Thus we have

$$\lim_{i\to\infty}\|A^i\| = 0 \implies \lim_{i\to\infty} A^i = 0.$$

Hence, from these results and Eq. (8) we get

$$\begin{aligned}(I+A)\left(\lim_{N\to\infty}\sum_{i=0}^{N} A^i\right) &= \lim_{N\to\infty}\left(I + 2\sum_{i=1}^{N} A^i\right) = I.\\ \implies \lim_{N\to\infty}\sum_{i=0}^{N} A^i &= (I+A)^{-1}.\end{aligned}$$

Then, taking norms in this last equality,

$$\begin{aligned}\|(I+A)^{-1}\| &= \lim_{N\to\infty}\left\|\sum_{i=0}^{N} A^i\right\| \\ &\leq \lim_{N\to\infty}\sum_{i=0}^{N}\|A^i\| && \text{(By Triangle Inequality)}\\ &\leq \lim_{N\to\infty}\sum_{i=0}^{N}\|A\|^i && \text{(By inequality (9))}\\ &= \sum_{i=0}^{\infty}\|A\|^i \\ &= \frac{1}{1-\|A\|}. && \text{(Geometric sum)}\end{aligned}$$

Thus we have shown the validity of Inequality (7), as desired. □

**Problem 7.** *Apply Householder reflectors and Gram-Schmidt orthogonalization to find $QR$-factorization of the following matrices:*

$$A = \begin{bmatrix} 4 & 8 & 1 \\ 0 & 2 & -2 \\ 3 & 6 & 7 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix}.$$

*Solution.* We first tackle both matrices using Householder reflectors, and then Gram-Schmidt:

- *Householder:* A ***Householder reflector*** is a symmetric, orthogonal matrix $H$ of the form

$$\boxed{H = I - 2\frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}}} \tag{11}$$

  Applying such a matrix to an $m$-vector $\mathbf{x}$ reflects this vector accross an $(m-1)$-dimensional plane, while preserving its norm; thus $H\mathbf{x} = \mathbf{y}$, with $\|\mathbf{y}\| = \|\mathbf{x}\|$. Our goal is to find a decomposition of a matrix $A$ such that

$$A = \underbrace{H_1\cdots H_k}_{Q} R,$$

  where $Q$ is an orthogonal matrix formed by a product of Householder matrices $H_1,\ldots,H_k$, and $R$ is a matrix whose square upper section is in upper-triangular form (if $R$ is square, then of course it is an upper-triangular matrix). As for the ***Householder vector*** $\mathbf{u}$ we set [1]

$$\mathbf{u} = \mathbf{x} - \operatorname{sgn}(x_1)\|\mathbf{x}\|_2\mathbf{e}_{(1)}.$$

[1] In the case where $x_1 = 0$, we simply choose $\operatorname{sgn}(x_1) = 1$. Also note that this is a matter of convention; occasionally we may find the opposite sign in the literature, i.e., $\mathbf{u} = \mathbf{x} + \operatorname{sgn}(x_1)\|\mathbf{x}\|_2\mathbf{e}_{(1)}$.

Applying this procedure to the matrix $A$, the vector columns $\mathbf{a}_{(i)}$ play the role of $\mathbf{x}$ from the discussion above; thus we start with

$$\mathbf{a}_{(1)} = \begin{bmatrix} 4 \\ 0 \\ 3 \end{bmatrix}.$$

Then,

$$\begin{aligned}
\mathbf{u} &= \mathbf{a}_{(1)} - \operatorname{sgn}((a_{(1)})_1)\|\mathbf{a}_{(1)}\|_2\mathbf{e}_{(1)} \\
&= \begin{bmatrix} 4 \\ 0 \\ 3 \end{bmatrix} - \operatorname{sgn}(4) \cdot \left(\sqrt{|4|^2+|3|^2}\right) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 4 \\ 0 \\ 3 \end{bmatrix} - \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} -1 \\ 0 \\ 3 \end{bmatrix}.
\end{aligned}$$

This Householder vector $\mathbf{u}$ yields the Householder matrix

$$\begin{aligned}
H_1 &= I - 2\frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - 2\left(\begin{bmatrix} -1 & 0 & 3 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 3 \end{bmatrix}\right)^{-1} \begin{bmatrix} -1 \\ 0 \\ 3 \end{bmatrix} \begin{bmatrix} -1 & 0 & 3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{5}\begin{bmatrix} 1 & 0 & -3 \\ 0 & 0 & 0 \\ -3 & 0 & 9 \end{bmatrix} \\
&= \begin{bmatrix} 4/5 & 0 & 3/5 \\ 0 & 1 & 0 \\ 3/5 & 0 & -4/5 \end{bmatrix}.
\end{aligned}$$

This yields

$$H_1 A = \begin{bmatrix} 4/5 & 0 & 3/5 \\ 0 & 1 & 0 \\ 3/5 & 0 & -4/5 \end{bmatrix} \begin{bmatrix} 4 & 8 & 1 \\ 0 & 2 & -2 \\ 3 & 6 & 7 \end{bmatrix} = \begin{bmatrix} 5 & 10 & 5 \\ 0 & 2 & -2 \\ 0 & 0 & -5 \end{bmatrix} = R.$$

Note how the resulting matrix $R$ is already in upper-triangular form, so we only had to construct one Householder matrix. Since Householder matrices are both symmetric and orthogonal, we have $H = H^{-1}$; thus

$$H_1 A = R \implies A = H_1^{-1} R = H_1 R.$$

Thus, setting $Q \equiv H_1$, we have our QR decomposition of $A$:

$$\underbrace{\begin{bmatrix} 4 & 8 & 1 \\ 0 & 2 & -2 \\ 3 & 6 & 7 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 4/5 & 0 & 3/5 \\ 0 & 1 & 0 \\ 3/5 & 0 & -4/5 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 5 & 10 & 5 \\ 0 & 2 & -2 \\ 0 & 0 & -5 \end{bmatrix}}_{R}.$$

We now follow the same steps for the matrix $B$, where

$$\mathbf{b}_{(1)} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}.$$

Now,

$$\begin{aligned}\mathbf{u} &= \mathbf{b}_{(1)} - \operatorname{sgn}((b_{(1)})_1)\|\mathbf{b}_{(1)}\|_2\mathbf{e}_{(1)} \\ &= \begin{bmatrix}2\\1\\2\end{bmatrix} - \begin{bmatrix}3\\0\\0\end{bmatrix} \\ &= \begin{bmatrix}-1\\1\\2\end{bmatrix}.\end{aligned}$$

This yields the Householder matrix

$$\begin{aligned}H_1 &= I - 2\frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}} \\ &= \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} - 2\left(\begin{bmatrix}-1&1&2\end{bmatrix}\begin{bmatrix}-1\\1\\2\end{bmatrix}\right)^{-1}\begin{bmatrix}-1\\1\\2\end{bmatrix}\begin{bmatrix}-1&1&2\end{bmatrix} \\ &= \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} - \frac{1}{3}\begin{bmatrix}1&-1&-2\\-1&1&2\\-2&2&4\end{bmatrix} \\ &= \begin{bmatrix}2/3&1/3&2/3\\1/3&2/3&-2/3\\2/3&-2/3&-1/3\end{bmatrix}.\end{aligned}$$

Thus, we have

$$H_1B = \begin{bmatrix}2/3&1/3&2/3\\1/3&2/3&-2/3\\2/3&-2/3&-1/3\end{bmatrix}\begin{bmatrix}2&1\\1&-1\\2&1\end{bmatrix} = \begin{bmatrix}3&1\\0&-1\\0&1\end{bmatrix}.$$

That takes care of the first column; we must now work on the (truncated) second column

$$\widehat{\mathbf{b}}_{(2)} = \begin{bmatrix}-1\\1\end{bmatrix}.$$

The corresponding Householder vector is given by

$$\begin{aligned}\mathbf{u} &= \widehat{\mathbf{b}}_{(2)} - \operatorname{sgn}((\hat{b}_{(2)})_1)\|\widehat{\mathbf{b}}_{(2)}\|_2\widehat{\mathbf{e}}_{(1)} \\ &= \begin{bmatrix}-1\\1\end{bmatrix} + \begin{bmatrix}\sqrt{2}\\0\end{bmatrix} \\ &= \begin{bmatrix}-1+\sqrt{2}\\1\end{bmatrix}.\end{aligned}$$

This yields the Householder matrix

$$\begin{aligned}\widehat{H}_2 &= \widehat{I} - 2\frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}} \\ &= \begin{bmatrix}1&0\\0&1\end{bmatrix} - 2\left(\begin{bmatrix}-1+\sqrt{2}&1\end{bmatrix}\begin{bmatrix}-1+\sqrt{2}\\1\end{bmatrix}\right)^{-1}\begin{bmatrix}-1+\sqrt{2}\\1\end{bmatrix}\begin{bmatrix}-1+\sqrt{2}&1\end{bmatrix} \\ &= \begin{bmatrix}1&0\\0&1\end{bmatrix} - \frac{2}{4-2\sqrt{2}}\begin{bmatrix}3-2\sqrt{2}&-1+\sqrt{2}\\-1+\sqrt{2}&1\end{bmatrix} \\ &= \begin{bmatrix}1/\sqrt{2}&-1/\sqrt{2}\\-1/\sqrt{2}&-1/\sqrt{2}\end{bmatrix}.\end{aligned}$$

Then, setting

$$H_2 = \begin{bmatrix}1&0&0\\0&\widehat{H}_2&\\0&&\end{bmatrix},$$

we get

$$H_2H_1B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & -1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 0 & -\sqrt{2} \\ 0 & 0 \end{bmatrix} = R.$$

Thus, since $H = H^{-1}$ for Householder matrices, we get

$$H_2H_1B = R \implies B = (H_2H_1)^{-1}R = H_1^{-1}H_2^{-1}R = H_1H_2R.$$

Thus, setting

$$Q \equiv H_1H_2 = \begin{bmatrix} 2/3 & 1/3 & 2/3 \\ 1/3 & 2/3 & -2/3 \\ 2/3 & -2/3 & -1/3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & -1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 2/3 & -1/(3\sqrt{2}) & -1/\sqrt{2} \\ 1/3 & 2\sqrt{2}/3 & 0 \\ 2/3 & -1/(3\sqrt{2}) & 1/\sqrt{2} \end{bmatrix},$$

which is indeed orthogonal (i.e., $Q^{-1} = Q^\top$), we have our QR decomposition of $B$:

$$\underbrace{\begin{bmatrix} 2 & 1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 2/3 & -1/(3\sqrt{2}) & -1/\sqrt{2} \\ 1/3 & 2\sqrt{2}/3 & 0 \\ 2/3 & -1/(3\sqrt{2}) & 1/\sqrt{2} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 3 & 1 \\ 0 & -\sqrt{2} \\ 0 & 0 \end{bmatrix}}_{R}.$$

- *Gram-Schmidt:* In typical Gram-Schmidt fashion, if we start with an arbitrary basis $\{\mathbf{a}_{(1)}, \ldots, \mathbf{a}_{(n)}\}$ we may form an orthonormal basis $\{\mathbf{q}_{(1)}, \ldots, \mathbf{q}_{(n)}\}$ such that at the $j^{\text{th}}$-step we have

$$\begin{aligned} \mathbf{q}_{(j)} &= \frac{\mathbf{a}_{(j)} - \left(\mathbf{q}_{(1)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(1)} - \left(\mathbf{q}_{(2)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(2)} - \cdots - \left(\mathbf{q}_{(j-1)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(j-1)}}{\|\mathbf{a}_{(j)} - \left(\mathbf{q}_{(1)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(1)} - \left(\mathbf{q}_{(2)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(2)} - \cdots - \left(\mathbf{q}_{(j-1)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(j-1)}\|_2} \\ &= \frac{\mathbf{a}_{(j)} - \sum_{i=1}^{j-1} \left(\mathbf{q}_{(i)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(i)}}{\|\mathbf{a}_{(j)} - \sum_{i=1}^{j-1} \left(\mathbf{q}_{(i)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(i)}\|_2}. \end{aligned} \tag{12}$$

This way, by construction, $\mathbf{q}_{(j)}$ is orthogonal to $\mathbf{q}_{(i)}$ for $i < j$ (i.e., $\mathbf{q}_{(i)}\mathbf{q}_{(j)}^\top = 0$ for all $i < j$), and also the vectors $\mathbf{q}_{(j)}$ have norm 1 by the denominator of Eq. (12). Thus, $\{\mathbf{q}_{(1)}, \ldots, \mathbf{q}_{(n)}\}$ is indeed an orthonormal basis.

Now, if we use the notation

$$r_{ij} \equiv \mathbf{q}_{(i)}^\top \mathbf{a}_{(j)} \quad \forall i < j \qquad \text{and} \qquad r_{jj} \equiv \|\mathbf{a}_{(j)} - \sum_{i=1}^{j-1} \left(\mathbf{q}_{(i)}^\top \mathbf{a}_{(j)}\right) \mathbf{q}_{(i)}\|_2,$$

then Eq. (12) can be rewritten as

$$\mathbf{a}_{(j)} = \sum_{i=1}^{j} \mathbf{q}_{(i)} r_{ij}$$

or, putting together all the column vectors in matrix form,

$$\underbrace{\begin{bmatrix} \mathbf{a}_{(1)} & \cdots & \mathbf{a}_{(n)} \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \mathbf{q}_{(1)} & \cdots & \mathbf{q}_{(n)} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}}_{R}. \tag{13}$$

This is how we get QR factorization of a matrix via ***Classical Gram-Schmidt*** (CGS). We will now apply this procedure to the two matrices given in this problem. Starting with $A$, we have

$$\mathbf{a}_{(1)} = \begin{bmatrix} 4 \\ 0 \\ 3 \end{bmatrix}; \qquad \mathbf{a}_{(2)} = \begin{bmatrix} 8 \\ 2 \\ 6 \end{bmatrix}; \qquad \mathbf{a}_{(3)} = \begin{bmatrix} 1 \\ -2 \\ 7 \end{bmatrix}.$$

Successive applications of Eq. (12) yield

$$\mathbf{q}_{(1)} = \frac{\mathbf{a}_{(1)}}{r_{11}} = \frac{\mathbf{a}_{(1)}}{\|\mathbf{a}_{(1)}\|_2} = \frac{1}{5}\begin{bmatrix}4\\0\\3\end{bmatrix} = \begin{bmatrix}4/5\\0\\3/5\end{bmatrix};$$

$$\mathbf{q}_{(2)} = \frac{\mathbf{a}_{(2)} - r_{12}\mathbf{q}_{(1)}}{r_{22}} = \frac{\mathbf{a}_{(2)} - r_{12}\mathbf{q}_{(1)}}{\|\mathbf{a}_{(2)} - r_{12}\mathbf{q}_{(1)}\|_2} = \frac{1}{2}\left(\begin{bmatrix}8\\2\\6\end{bmatrix} - 10\begin{bmatrix}4/5\\0\\3/5\end{bmatrix}\right) = \begin{bmatrix}0\\1\\0\end{bmatrix};$$

$$\mathbf{q}_{(3)} = \frac{\mathbf{a}_{(3)} - r_{13}\mathbf{q}_{(1)} - r_{23}\mathbf{q}_{(2)}}{r_{33}} = \frac{\mathbf{a}_{(3)} - r_{13}\mathbf{q}_{(1)} - r_{23}\mathbf{q}_{(2)}}{\|\mathbf{a}_{(3)} - r_{13}\mathbf{q}_{(1)} - r_{23}\mathbf{q}_{(2)}\|_2} = \frac{1}{5}\left(\begin{bmatrix}1\\-2\\7\end{bmatrix} - 5\begin{bmatrix}4/5\\0\\3/5\end{bmatrix} + 2\begin{bmatrix}0\\1\\0\end{bmatrix}\right) = \begin{bmatrix}-3/5\\0\\4/5\end{bmatrix}.$$

On these calculations we used

$$r_{12} = \mathbf{q}_{(1)}^\top \mathbf{a}_{(2)} = \begin{bmatrix}4/5 & 0 & 3/5\end{bmatrix}\begin{bmatrix}8\\2\\6\end{bmatrix} = 10,$$

$$\implies r_{22} = \|\mathbf{a}_{(2)} - r_{12}\mathbf{q}_{(1)}\|_2 = \left\|\begin{bmatrix}8\\2\\6\end{bmatrix} - 10\begin{bmatrix}4/5\\0\\3/5\end{bmatrix}\right\|_2 = \left\|\begin{bmatrix}0\\2\\0\end{bmatrix}\right\|_2 = 2;$$

$$r_{13} = \mathbf{q}_{(1)}^\top \mathbf{a}_{(3)} = \begin{bmatrix}4/5 & 0 & 3/5\end{bmatrix}\begin{bmatrix}1\\-2\\7\end{bmatrix} = 5,$$

$$r_{23} = \mathbf{q}_{(2)}^\top \mathbf{a}_{(3)} = \begin{bmatrix}0 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\-2\\7\end{bmatrix} = -2,$$

$$\implies r_{33} = \|\mathbf{a}_{(3)} - r_{13}\mathbf{q}_{(1)} - r_{23}\mathbf{q}_{(2)}\|_2 = \left\|\begin{bmatrix}1\\-2\\7\end{bmatrix} - 5\begin{bmatrix}4/5\\0\\3/5\end{bmatrix} + 2\begin{bmatrix}0\\1\\0\end{bmatrix}\right\|_2 = \left\|\begin{bmatrix}-3\\0\\4\end{bmatrix}\right\|_2 = 5.$$

Hence we end up with the QR decomposition of $A$:

$$\underbrace{\begin{bmatrix}4 & 8 & 1\\0 & 2 & -2\\3 & 6 & 7\end{bmatrix}}_{A} = \underbrace{\begin{bmatrix}4/5 & 0 & -3/5\\0 & 1 & 0\\3/5 & 0 & 4/5\end{bmatrix}}_{Q}\underbrace{\begin{bmatrix}5 & 10 & 5\\0 & 2 & -2\\0 & 0 & 5\end{bmatrix}}_{R}.$$

We now follow the same steps for $B$, where

$$\mathbf{b}_{(1)} = \begin{bmatrix}2\\1\\2\end{bmatrix}; \qquad \mathbf{b}_{(2)} = \begin{bmatrix}1\\-1\\1\end{bmatrix}.$$

Successive applications of Eq. (12) yield

$$\mathbf{q}_{(1)} = \frac{\mathbf{b}_{(1)}}{r_{11}} = \frac{\mathbf{b}_{(1)}}{\|\mathbf{b}_{(1)}\|_2} = \frac{1}{3}\begin{bmatrix}2\\1\\2\end{bmatrix} = \begin{bmatrix}2/3\\1/3\\2/3\end{bmatrix};$$

$$\mathbf{q}_{(2)} = \frac{\mathbf{b}_{(2)} - r_{12}\mathbf{q}_{(1)}}{r_{22}} = \frac{\mathbf{b}_{(2)} - r_{12}\mathbf{q}_{(1)}}{\|\mathbf{b}_{(2)} - r_{12}\mathbf{q}_{(1)}\|_2} = \frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\-1\\1\end{bmatrix} - \begin{bmatrix}2/3\\1/3\\2/3\end{bmatrix}\right) = \begin{bmatrix}1/(3\sqrt{2})\\-2\sqrt{2}/3\\1/(3\sqrt{2})\end{bmatrix}.$$

On these calculations we used

$$\begin{aligned} r_{12} = \mathbf{q}_{(1)}^{\top}\mathbf{b}_{(2)} &= \begin{bmatrix} 2/3 & 1/3 & 2/3 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = 1, \\ \implies r_{22} = \|\mathbf{b}_{(2)} - r_{12}\mathbf{q}_{(1)}\|_2 &= \left\| \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2/3 \\ 1/3 \\ 2/3 \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} 1/3 \\ -4/3 \\ 1/3 \end{bmatrix} \right\|_2 = \sqrt{2}. \end{aligned}$$

Hence we end up with the (reduced) QR decomposition of $B$:

$$\underbrace{\begin{bmatrix} 2 & 1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 2/3 & 1/(3\sqrt{2}) \\ 1/3 & -2\sqrt{2}/3 \\ 2/3 & 1/(3\sqrt{2}) \end{bmatrix}}_{\widehat{Q}} \underbrace{\begin{bmatrix} 3 & 1 \\ 0 & \sqrt{2} \end{bmatrix}}_{\widehat{R}}.$$

To get the full QR decomposition of $B$, we introduce an extra column vector

$$\widehat{\mathbf{b}}_{(3)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Then,

$$\mathbf{q}_{(3)} = \frac{\widehat{\mathbf{b}}_{(3)} - \left(\mathbf{q}_{(1)}^{\top}\widehat{\mathbf{b}}_{(3)}\right)\mathbf{q}_{(1)} - \left(\mathbf{q}_{(2)}^{\top}\widehat{\mathbf{b}}_{(3)}\right)\mathbf{q}_{(2)}}{\left\|\widehat{\mathbf{b}}_{(3)} - \left(\mathbf{q}_{(1)}^{\top}\widehat{\mathbf{b}}_{(3)}\right)\mathbf{q}_{(1)} - \left(\mathbf{q}_{(2)}^{\top}\widehat{\mathbf{b}}_{(3)}\right)\mathbf{q}_{(2)}\right\|_2} = \frac{2}{\sqrt{2}}\left( \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \frac{2}{3}\begin{bmatrix} 2/3 \\ 1/3 \\ 2/3 \end{bmatrix} - \frac{1}{3\sqrt{2}}\begin{bmatrix} 1/(3\sqrt{2}) \\ -2\sqrt{2}/3 \\ 1/(3\sqrt{2}) \end{bmatrix} \right) = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{bmatrix},$$

where we used

$$\begin{aligned} \mathbf{q}_{(1)}^{\top}\widehat{\mathbf{b}}_{(3)} &= \begin{bmatrix} 2/3 & 1/3 & 2/3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \frac{2}{3}, \\ \mathbf{q}_{(2)}^{\top}\widehat{\mathbf{b}}_{(3)} &= \begin{bmatrix} 1/(3\sqrt{2}) & -2\sqrt{2}/3 & 1/(3\sqrt{2}) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{3\sqrt{2}}, \\ \implies \left\|\widehat{\mathbf{b}}_{(3)} - \left(\mathbf{q}_{(1)}^{\top}\widehat{\mathbf{b}}_{(3)}\right)\mathbf{q}_{(1)} - \left(\mathbf{q}_{(2)}^{\top}\widehat{\mathbf{b}}_{(3)}\right)\mathbf{q}_{(2)}\right\|_2 &= \left\| \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \frac{2}{3}\begin{bmatrix} 2/3 \\ 1/3 \\ 2/3 \end{bmatrix} - \frac{1}{3\sqrt{2}}\begin{bmatrix} 1/(3\sqrt{2}) \\ -2\sqrt{2}/3 \\ 1/(3\sqrt{2}) \end{bmatrix} \right\|_2 \\ &= \left\| \begin{bmatrix} 1/2 \\ 0 \\ -1/2 \end{bmatrix} \right\|_2 = \frac{\sqrt{2}}{2}. \end{aligned}$$

Thus we have the full QR decomposition of $B$,

$$\underbrace{\begin{bmatrix} 2 & 1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 2/3 & 1/(3\sqrt{2}) & 1/\sqrt{2} \\ 1/3 & -2\sqrt{2}/3 & 0 \\ 2/3 & 1/(3\sqrt{2}) & -1/\sqrt{2} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 3 & 1 \\ 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix}}_{R}.$$

□

**Problem 8.** $\boxed{\text{C}}$ **Classical vs. Modified Gram-Schmidt**: *Construct a real square matrix $A = U\Sigma V^\top$ of size 80, where*

$$\Sigma_{i,j} = \begin{cases} 2^{-i} & \text{if} \quad i = j, i = 1, \ldots, 80 \\ 0 & \text{if} \quad i \neq j, \end{cases}$$

*and $U$ and $V$ are random orthogonal matrices. Write programs for the classical and the modified Gram-Schmidt algorithms and perform $QR$ factorization for $A$. Plot the diagonal elements of $R$ versus $j$ for each algorithm.*

*Solution.* The following code generates the matrix $A$:

```
using namespace std;
using namespace Eigen;

int main(int argc, const char * argv[]) {

        const int n {80};

        MatrixXd A(n,n);
        MatrixXd Sigma(n,n);
        MatrixXd Vrand = MatrixXd::Random(n,n), V;
        MatrixXd Urand = MatrixXd::Random(n,n), U;

        //Generate matrix Sigma
        for (int i {0}; i < n; i++) {
            for (int j {0}; j < n; j++){
                //construction of matrix Sigma
                if (i == j)
                     Sigma(i,j) = pow(2.0,(-(i+1)));
                 else
                     Sigma(i,j) = 0.0;
            }
        }

        //Orthogonalize random matrices Urand and Vrand
        HouseholderQR<MatrixXd> qr1(Vrand);
        HouseholderQR<MatrixXd> qr2(Urand);
        V = qr1.householderQ();
        U = qr2.householderQ();

        A = U * Sigma * V.transpose();
}
```

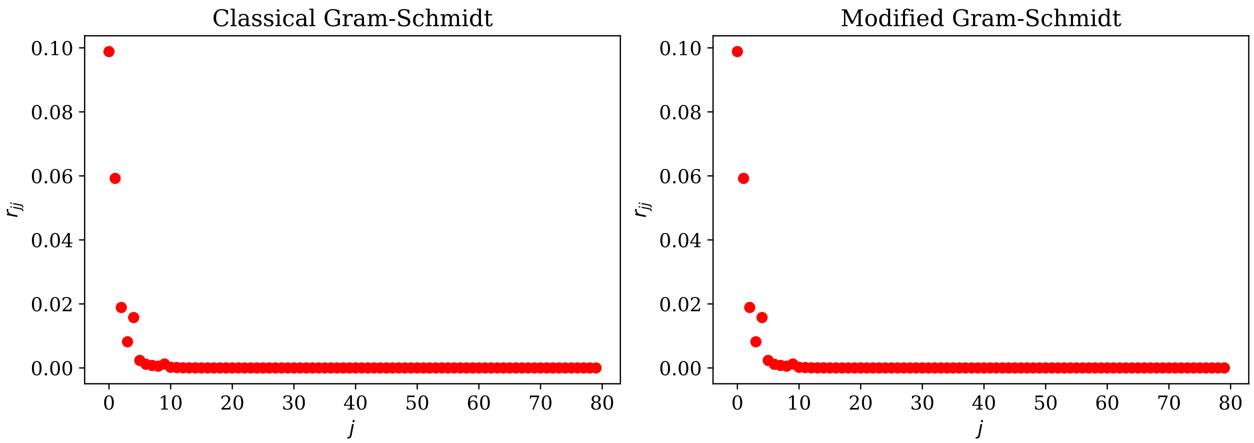
Note that we have used Eigen's HouseholderQR solver to orthogonalize the generated random matrices Urand and Vrand. We will not, howevever, make use of this or any other solver to orthogonalize $A$, since that is the whole point of the exercise after all! The function householderQ() picks the orthogonal $Q$ matrix from the QR factorizations of Urand and Vrand.

The code for the QR decomposition of $A$ using both Classical Gram-Schmidt (CGS) and the Modified Gram-Schmidt (MGS) algorithms is presented in the following snippet:

```
    MatrixXd Q(n,n);
    MatrixXd R = MatrixXd::Zero(n,n);
    VectorXd v(n);

//    Classical Gram-Schmidt
    for (int j {0}; j < n; j++){
        v = A.col(j);
        for (int i {0}; i < j; i++) {
            R(i,j) = Q.col(i).transpose() * A.col(j);
            v = v - (R(i,j) * Q.col(i));
        }
        R(j,j) = v.norm();
        Q.col(j) = v/R(j,j);
    }
//    Modified Gram-Schmidt
    for (int i {0}; i < n; i++)
        Q.col(i) = A.col(i);
    for (int i {0}; i < n; i++){
        R(i,i) = Q.col(i).norm();
        Q.col(i) = Q.col(i)/R(i,i);
        for (int j {i+1}; j < n; j++) {
            R(i,j) = Q.col(i).transpose() * Q.col(j);
            Q.col(j) = Q.col(j) - (R(i,j) * Q.col(i));
        }
    }
```

**Remark:** Note that, unlike $Q$, the matrix $R$ had to initialized to a zero matrix; this is a quirk of the C++ language in that if an object is not initialized it is assigned random values. Since the Gram-Schmidt algorithms only assign values to the upper-triangular part of $R$, if we do not initialize the matrix then the lower-triangular entries are assigned random values.

We conclude the exercise by showing plots of the diagonal elements of $R$ versus $j$ for both algorithms:



□

**Problem 9.** *Consider the matrix*

$$A = \begin{bmatrix} -2 & 11 \\ -10 & 5 \end{bmatrix}.$$

a) *Determine, on paper, the SVD of $A$ in the form $A = U\Sigma V^\top$. List singular values, left singular vectors, and right singular vectors of $A$.*

b) *What are the $1$-, $2$-, $\infty$- and Frobenius norms of $A$?*

c) *Find $A^{-1}$ not directly, but via the SVD.*

d) *Find the eigenvalues $\lambda_1, \lambda_2$ of $A$.*

e) *Verify that $\det(A) = \lambda_1\lambda_2$ and $|\det(A)| = \sigma_1\sigma_2$.*

*Solution to a).* For the calculation of the nonzero singular values $\sigma_i \in \Sigma$, we start by focusing on the direction of the largest action of $A$; thus setting $\sigma_1 = \|A\|_2 = \sqrt{|\lambda_{\max}|}$, where $\lambda_{\max}$ is the largest eigenvalue of $AA^\top$ (or, equivalently, of $A^\top A$; c.f., Eq (6)). Then $\sigma_2$ would be the second largest such value and so on ... Hence, we start with the calculation

$$\begin{aligned} AA^\top - \lambda I &= \begin{bmatrix} -2 & 11 \\ -10 & 5 \end{bmatrix} \begin{bmatrix} -2 & -10 \\ 11 & 5 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \\ &= \begin{bmatrix} 125-\lambda & 75 \\ 75 & 125-\lambda \end{bmatrix}, \end{aligned}$$

which leads to

$$0 = \det\left(AA^\top - \lambda I\right) = \lambda^2 - 250\lambda + 10000 \implies \lambda_1 = 200,\ \lambda_2 = 50.$$

Thus we have the *singular values* $\sigma_1$ and $\sigma_2$ given by

$$\sigma_1 = \sqrt{|\lambda_{\max}|} = \sqrt{|\lambda_1|} = \sqrt{200} = 10\sqrt{2} \qquad \text{and} \qquad \sigma_2 = \sqrt{|\lambda_2|} = \sqrt{50} = 5\sqrt{2},$$

so that

$$\Sigma = \begin{bmatrix} 10\sqrt{2} & 0 \\ 0 & 5\sqrt{2} \end{bmatrix}.$$

Now, from the calculations

$$AA^\top = \left(U\Sigma V^\top\right)\left(U\Sigma V^\top\right)^\top = U\Sigma V^\top V\Sigma^\top U^\top = U\left(\Sigma\Sigma^\top\right)U^\top \tag{14a}$$

$$A^\top A = \left(U\Sigma V^\top\right)^\top\left(U\Sigma V^\top\right) = V\Sigma^\top U^\top U\Sigma V^\top = V\left(\Sigma^\top\Sigma\right)V^\top, \tag{14b}$$

and the fact that both $U$ and $V$ are (by construction) orthogonal, we see that $AA^\top$ and $A^\top A$ are *similar* to $\Sigma\Sigma^\top$ and $\Sigma^\top\Sigma$, respectively, so they share the same eigenvalues. Moreover, the column vectors $\mathbf{u}_{(i)} \in U$ and $\mathbf{v}_{(i)} \in V$ are the eigenvectors of $AA^\top$ and $A^\top A$, respectively. We proceed first with the calculation of the $\mathbf{v}_{(i)}$:

- For $\lambda_1 = 200$,

$$\begin{aligned}\left(A^\top A - \lambda_1 I\right)\mathbf{v}_{(1)} &= 0\\ \left(\begin{bmatrix}-2 & -10\\ 11 & 5\end{bmatrix}\begin{bmatrix}-2 & 11\\ -10 & 5\end{bmatrix} - \begin{bmatrix}200 & 0\\ 0 & 200\end{bmatrix}\right)\begin{bmatrix}v_1\\ v_2\end{bmatrix} &= \begin{bmatrix}0\\ 0\end{bmatrix}\\ \left(\begin{bmatrix}104 & -72\\ -72 & 146\end{bmatrix} - \begin{bmatrix}200 & 0\\ 0 & 200\end{bmatrix}\right)\begin{bmatrix}v_1\\ v_2\end{bmatrix} &= \begin{bmatrix}0\\ 0\end{bmatrix}\\ \begin{bmatrix}-96 & -72\\ -72 & -54\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix} &= \begin{bmatrix}0\\ 0\end{bmatrix}.\end{aligned}$$

  A straightforward Gaussian elimination yields

$$\left[\begin{array}{cc|c}1 & \frac{3}{4} & 0\\ 0 & 0 & 0\end{array}\right] \implies v_2 = \alpha \in \mathbb{R},\ v_1 = -\frac{3}{4}\alpha.$$

  Letting $\alpha = 4$ and normalizing, we have our first *right singular vector*

$$\mathbf{v}_{(1)} = \frac{1}{\sqrt{(-3)^2 + 4^2}}\begin{bmatrix}-3\\ 4\end{bmatrix} = \begin{bmatrix}-3/5\\ 4/5\end{bmatrix}.$$

- For $\lambda_2 = 50$,

$$\begin{aligned}\left(A^\top A - \lambda_2 I\right)\mathbf{v}_{(2)} &= 0\\ \left(\begin{bmatrix}104 & -72\\ -72 & 146\end{bmatrix} - \begin{bmatrix}50 & 0\\ 0 & 50\end{bmatrix}\right)\begin{bmatrix}v_1\\ v_2\end{bmatrix} &= \begin{bmatrix}0\\ 0\end{bmatrix}\\ \begin{bmatrix}54 & -72\\ -72 & 96\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix} &= \begin{bmatrix}0\\ 0\end{bmatrix}.\end{aligned}$$

  A straightforward Gaussian elimination yields

$$\left[\begin{array}{cc|c}1 & -\frac{4}{3} & 0\\ 0 & 0 & 0\end{array}\right] \implies v_2 = \alpha \in \mathbb{R},\ v_1 = \frac{4}{3}\alpha.$$

  Letting $\alpha = 3$ and normalizing, we have our second right singular vector

$$\mathbf{v}_{(2)} = \begin{bmatrix}4/5\\ 3/5\end{bmatrix}.$$

Thus we end up with

$$V = \begin{bmatrix}-3/5 & 4/5\\ 4/5 & 3/5\end{bmatrix}.$$

Now, for the computation of the *left singular vectors* $\mathbf{u}_{(i)} \in U$, we can proceed by calculating the eigenvectors of $AA^\top$, as we stated above; however, note that from the SVD decomposition of $A$ (i.e., $A = U\Sigma V^\top$), we have $AV\Sigma^{-1} = U$, and thus

$$\mathbf{u}_{(i)} = \frac{1}{\sigma_i}A\mathbf{v}_{(i)}. \tag{15}$$

This yields

$$\begin{aligned}
\mathbf{u}_{(1)} &= \frac{1}{\sigma_1} A\mathbf{v}_{(1)} \\
&= \frac{1}{10\sqrt{2}} \begin{bmatrix} -2 & 11 \\ -10 & 5 \end{bmatrix} \begin{bmatrix} -3/5 \\ 4/5 \end{bmatrix} \\
&= \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix};
\end{aligned}$$

$$\begin{aligned}
\mathbf{u}_{(2)} &= \frac{1}{\sigma_2} A\mathbf{v}_{(2)} \\
&= \frac{1}{5\sqrt{2}} \begin{bmatrix} -2 & 11 \\ -10 & 5 \end{bmatrix} \begin{bmatrix} 4/5 \\ 3/5 \end{bmatrix} \\
&= \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}.
\end{aligned}$$

Hence, in conclusion, we have

$$\underbrace{\begin{bmatrix} -2 & 11 \\ -10 & 5 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 10\sqrt{2} & 0 \\ 0 & 5\sqrt{2} \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} -3/5 & 4/5 \\ 4/5 & 3/5 \end{bmatrix}}_{V^\top}. \qquad \square \tag{16}$$

*Solution to b).* From Eq. (4), we have

$$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{m} |a_{ij}| = |11| + |5| = 16. \qquad \surd$$

From Eq. (6) and our calculations in part a),

$$\|A\|_2 = \sqrt{|\lambda_{\max}|} = \sigma_1 = 10\sqrt{2}. \qquad \surd$$

From Eq. (5),

$$\|A\|_\infty = \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}| = |-10| + |5| = 15. \qquad \surd$$

Lastly, from Eq. (2),

$$\|A\|_F = \sqrt{\operatorname{Tr}(A^\top A)} = \sqrt{104 + 146} = 5\sqrt{10}. \qquad \surd \qquad \square$$

*Solution to c).* Using the fact that both $U$ and $V$ are orthogonal matrices, we have

$$\begin{aligned}
A^{-1} &= \left(U\Sigma V^\top\right)^{-1} \\
&= V\Sigma^{-1}U^\top \\
&= \begin{bmatrix} -3/5 & 4/5 \\ 4/5 & 3/5 \end{bmatrix} \begin{bmatrix} \sqrt{2}/20 & 0 \\ 0 & \sqrt{2}/10 \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix} \\
&= \begin{bmatrix} 1/20 & -11/100 \\ 1/10 & -1/50 \end{bmatrix}.
\end{aligned} \qquad \square$$

*Solution to d).* The characteristic polynomial yields

$$A - \lambda I = \begin{bmatrix} -2-\lambda & 11 \\ -10 & 5-\lambda \end{bmatrix}; \quad 0 = \det(A - \lambda I) = \lambda^2 - 3\lambda + 100 \implies \lambda_{\substack{1\\2}} = \frac{3 \pm i\sqrt{391}}{2}. \qquad \square$$

*Solution to e).* Note that on the one hand,

$$\det A = -2 \cdot 5 - (-10) \cdot 11 = 100,$$

while also

$$\lambda_1 \lambda_2 = \frac{3 + i\sqrt{391}}{2} \frac{3 - i\sqrt{391}}{2} = 100. \qquad \surd$$

Thus the equality $\det A = \lambda_1 \lambda_2$ does hold. Moreover,

$$\sigma_1 \sigma_2 = 10\sqrt{2} \cdot 5\sqrt{2} = 100 = |\det A|. \qquad \surd$$

□

**Problem 10.** *Let $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ be an orthonormal base for a subspace $U$ in an inner-product space $X$. Define $P : X \to U$ by the equation*

$$P(\mathbf{x}) = \sum_{i=1}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i.$$

*Show that*

a) *$P$ is linear.*

b) *$P$ is idempotent, that is, $P^2 = P$.*

c) *$P(\mathbf{x}) = \mathbf{x}$, if $\mathbf{x} \in U$.*

d) *$\|P(\mathbf{x})\|_2 \leq \|\mathbf{x}\|_2$, for every $\mathbf{x} \in X$.*

*Solution to a).* In what follows we use the fact that the bilinear form $\langle \_, \_ \rangle$ is linear (in fact, in both slots). Thus, we have, for any $\alpha \in \mathbb{R}$,

$$P(\alpha \mathbf{x}) = \sum_{i=1}^{n} \langle \alpha \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i = \alpha \sum_{i=1}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i = \alpha P(\mathbf{x}). \qquad \surd$$

Moreover, for any $\mathbf{y} \in X$,

$$P(\mathbf{x} + \mathbf{y}) = \sum_{i=1}^{n} \langle \mathbf{x} + \mathbf{y}, \mathbf{u}_i \rangle \mathbf{u}_i = \sum_{i=1}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i + \sum_{i=1}^{n} \langle \mathbf{y}, \mathbf{u}_i \rangle \mathbf{u}_i = P(\mathbf{x}) + P(\mathbf{x}). \qquad \surd$$

□

*Solution to b).*

$$\begin{aligned}
P^2(\mathbf{x}) &= P\left( \sum_{i=1}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i \right) \\
&= \sum_{j=1}^{n} \left\langle \sum_{i=1}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i, \mathbf{u}_j \right\rangle \mathbf{u}_j \\
&= \sum_{j=1}^{n} \sum_{i=1}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle \left\langle \mathbf{u}_i, \mathbf{u}_j \right\rangle \mathbf{u}_j && \text{(By linearity of } \langle \_, \_ \rangle\text{)} \\
&= \sum_{j=1}^{n} \sum_{i=1}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle \delta_{ij} \mathbf{u}_j && \text{(By orthonormality of the } \mathbf{u}_i\text{)} \\
&= \sum_{i=1}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i \\
&= P(x).
\end{aligned}$$

□

*Solution to c).* If $\mathbf{x} \in U$, it can be written in terms of the basis $\{\mathbf{u}_j\}$ as

$$\mathbf{x} = \sum_{j=1}^{n} x_j \mathbf{u}_j,$$

where the $x_j$ are the components of $\mathbf{x}$ in this basis. But then

$$\begin{aligned}
P(\mathbf{x}) &= \sum_{i=1}^{n} \left\langle \sum_{j=1}^{n} x_j \mathbf{u}_j, \mathbf{u}_i \right\rangle \mathbf{u}_i \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} x_j \left\langle \mathbf{u}_j, \mathbf{u}_i \right\rangle \mathbf{u}_i && \text{(By linearity of } \langle \_ , \_ \rangle) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} x_j \delta_{ji} \mathbf{u}_i && \text{(By orthonormality of the } \mathbf{u}_i) \\
&= \sum_{i=1}^{n} x_i \mathbf{u}_i \\
&= \mathbf{x}.
\end{aligned}$$

□

*Solution to d).*

$$\begin{aligned}
\|P(\mathbf{x})\|_2 &= \left( \sum_{i=1}^{n} |\langle \mathbf{x}, \mathbf{u}_i \rangle|^2 \right)^{1/2} \\
&= \left( \sum_{i=i}^{m} \left| \left\langle \sum_{j=1}^{n} x_j \mathbf{u}_j, \mathbf{u}_i \right\rangle \right|^2 \right)^{1/2} \\
&= \left( \sum_{i=1}^{n} \left| \sum_{j=1}^{n} x_j \left\langle \mathbf{u}_j, \mathbf{u}_i \right\rangle \right|^2 \right)^{1/2} && \text{(By linearity of } \langle \_ , \_ \rangle) \\
&\leq \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \left| x_j \left\langle \mathbf{u}_j, \mathbf{u}_i \right\rangle \right|^2 \right)^{1/2} && \text{(By Triangle Inequality)} \\
&= \left( \sum_{i=1}^{n} \sum_{j=1}^{n} |x_j \delta_{ji}|^2 \right)^{1/2} && \text{(By orthonormality of the } \mathbf{u}_i) \\
&= \left( \sum_{i=1}^{n} |x_i|^2 \right)^{1/2} \\
&= \|\mathbf{x}\|_2.
\end{aligned}$$

□

**Problem 11.** $\boxed{\textbf{C}}$ *Let $n$ be an even integer, and consider the $n \times n$ matrix $A$ with 3 on the main diagonal, $-1$ on the super- and sub-diagonal, and $1/2$ in the $(i, n+1-i)$ position for all $i = 1, \dots, n$ except for $i = n/2$ and $n/2 + 1$. Define a vector $\mathbf{b} = [2.5, 1.5, \dots, 1.5, 1.0, 1.0, 1.5, \dots, 1.5, 2.5]^\top$ where there are $n - 4$ repetitions of $1.5$ and two repetitions of $1$. Write ~~MATLAB~~ (C++!) codes and solve the system $A\mathbf{x} = \mathbf{b}$ for $n = 12$ (a system of 12 equations with 12 unknowns) using the following methods:*

i) *Jacobi*

ii) *Gauss-Seidel*

iii) *SOR (with $\omega = 1.1$)*

*The correct solution of the system is $[1, \dots, 1]^\top$. Make a table of errors in infinity norm after 10 iterations for each method.*

*Solution.* The idea behind these iterative methods is that, if we have a ***large and sparse*** linear system $A\mathbf{x} = \mathbf{b}$, we rewrite it in an equivalent form

$$\mathbf{x} = B\mathbf{x} + \mathbf{d}. \tag{17}$$

(How the matrix $B$ and the vector $\mathbf{d}$ are defined depends on which iterative method we use; see respective implementations below.) Then, starting with an initial approximation $\mathbf{x}^{(0)}$ of the solution vector $\mathbf{x}$, we generate a sequence $\{\mathbf{x}^{(k)}\}$ by the iterative scheme

$$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{d} \qquad k = 0, 1, \ldots \tag{18}$$

We stop this algorithm either when the ***relative residual norm*** satisfies

$$\frac{\|\mathbf{b} - A\mathbf{x}^{(k)}\|}{\|\mathbf{b}\|} \leq \epsilon \tag{19}$$

for some user-defined tolerance $\epsilon > 0$, or when the algorithm reaches a maximum number of iterations that the user is willing to allow. In our work here we will not use a tolerance parameter because we are explicitly told to run the code for a total of ten times.

The first order of business for all such iterative methods is to rewrite the matrix $A$ in the form $A = L + D + U$, where

$$\begin{aligned} L &= \text{lower triangular with zeroes on the diagonal;} \\ D &= \text{diagonal;} \\ U &= \text{upper triangular with zeroes on the diagonal.} \end{aligned}$$

That is,

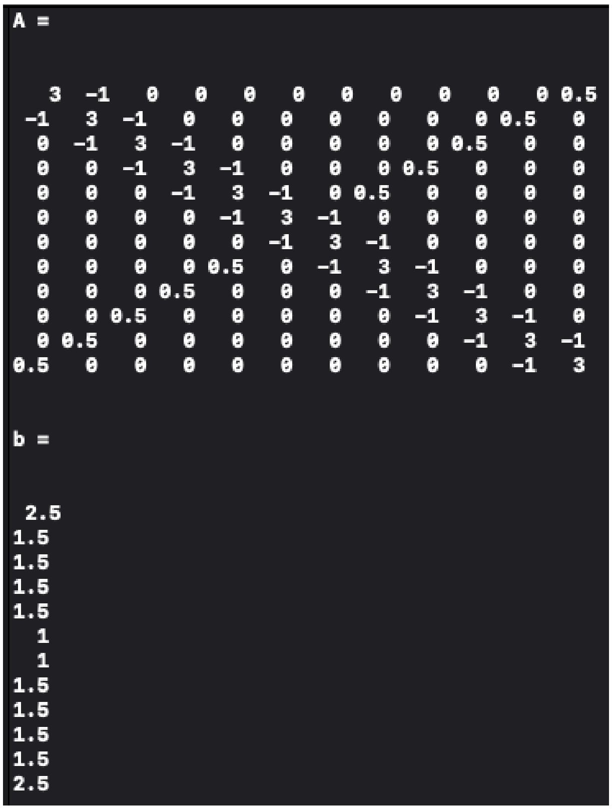
$$\underbrace{\begin{pmatrix} \ddots & & & & U \\ & \ddots & & & \\ & & D & & \\ & & & \ddots & \\ L & & & & \ddots \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 0 & & 0 \\ & \ddots & & \\ & & 0 & \\ & & & \ddots \\ * & & & 0 \end{pmatrix}}_{L} + \underbrace{\begin{pmatrix} a_{11} & & 0 \\ & \ddots & & \\ & & a_{ii} & \\ & & & \ddots \\ 0 & & & a_{nn} \end{pmatrix}}_{D} + \underbrace{\begin{pmatrix} 0 & & * \\ & \ddots & & \\ & & 0 & \\ & & & \ddots \\ 0 & & & 0 \end{pmatrix}}_{U}.$$

Before we apply any of the iteration methods, let us construct the matrix $A$ and vector $\mathbf{b}$; the following snippet shows how: [2]

```
using namespace std;
using namespace Eigen;

const int n {12};

MatrixXd A(n,n);
VectorXd b(n);

//Generate matrix A
for (int i {0}; i < n; i++) {
    for (int j {0}; j < n; j++){
        if (i == j)
            A(i,j) = 3.0;
        else if ( (i == j+1) || (j == i+1))
           A(i,j) = -1.0;
        else if ( (j == n-i-1) && (  (i != n/2 - 1) || (i != n/2)    )   )
            A(i,j) = 1.0/2.0;
        else
            A(i,j) = 0.0;
    }
}

//Generate vector b
for (int j {0}; j < n; j++) {
    if ( (j == 0) || (j == n-1) )
        b(j) = 2.5;
    else if ( (j == n/2 - 1) || (j == n/2)  )
        b(j) = 1.0;
    else
        b(j) = 1.5;
}

cout << "A = \n " << "\n \n " << A << "\n \n "  << endl;
cout << "b = \n " << "\n \n " << b << "\n \n " << endl;
```

[2] Do keep in mind that the indexing in the code looks slightly different from the one in the exercise's statement because I'm accommodating for the C++ convention of starting at index 0 rather than 1.

The output for the above code is the following matrix and vector:



```
A =


   3  -1   0   0   0   0   0   0   0   0   0 0.5
 -1   3  -1   0   0   0   0   0   0   0 0.5   0
  0  -1   3  -1   0   0   0   0   0 0.5   0   0
  0   0  -1   3  -1   0   0   0 0.5   0   0   0
  0   0   0  -1   3  -1   0 0.5   0   0   0   0
  0   0   0   0  -1   3  -1   0   0   0   0   0
  0   0   0   0   0  -1   3  -1   0   0   0   0
  0   0   0   0 0.5   0  -1   3  -1   0   0   0
  0   0   0 0.5   0   0   0  -1   3  -1   0   0
  0   0 0.5   0   0   0   0   0  -1   3  -1   0
  0 0.5   0   0   0   0   0   0   0  -1   3  -1
0.5   0   0   0   0   0   0   0   0   0  -1   3


b =


 2.5
1.5
1.5
1.5
1.5
  1
  1
1.5
1.5
1.5
1.5
2.5
```

We can now implement the three algorithms to find the solution $\mathbf{x}$ to the system $A\mathbf{x} = \mathbf{b}$. We are already told that the correct solution of the system is $\widehat{\mathbf{x}} = [1, \ldots, 1]^\top$; thus let us pick a different vector for our starting guess, say, $\mathbf{x}^{(0)} = [2, \ldots, 2]^\top$.

For i), the ***Jacobi*** method, we rewrite $(L + D + U)\mathbf{x} = \mathbf{b}$ as

$$D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x},$$

which implies

$$\begin{aligned} \mathbf{x} &= D^{-1}\left(\mathbf{b} - (L + U)\mathbf{x}\right) \\ &= \underbrace{-D^{-1}(L + U)}_{B}\,\mathbf{x} + \underbrace{D^{-1}\mathbf{b}}_{\mathbf{d}}. \end{aligned}$$

Thus, from $\mathbf{x} = D^{-1}\left(\mathbf{b} - (L + U)\mathbf{x}\right)$, we see that the Jacobi algorithm is given by

$$\boxed{x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=0 \\ i \neq j}}^{n-1} a_{ij} x_j^{(k)}\right)} \tag{20}$$

For ii), the ***Gauss-Seidel*** method, we rewrite $(L + D + U)\mathbf{x} = \mathbf{b}$ as

$$(L + D)\mathbf{x} = \mathbf{b} - U\mathbf{x},$$

which implies

$$\begin{aligned} \mathbf{x} &= (L + D)^{-1}\left[\mathbf{b} - U\mathbf{x}\right] \\ &= \underbrace{-(L + D)^{-1}U}_{B}\,\mathbf{x} + \underbrace{(L + D)^{-1}\mathbf{b}}_{\mathbf{d}}. \end{aligned}$$

If we write $\mathbf{x} = (L + D)^{-1}\left[\mathbf{b} - U\mathbf{x}\right]$ in the iterative form

$$\mathbf{x}^{(k+1)} = (L + D)^{-1}\left[\mathbf{b} - U\mathbf{x}^{(k)}\right],$$

we see that this expression may also be written as

$$\mathbf{x}^{(k+1)} = D^{-1}\left[\mathbf{b} - L\mathbf{x}^{(k+1)} - U\mathbf{x}^{(k)}\right]. \tag{21}$$

This shows the main difference between the Jacobi and Gauss-Seidel algorithms: Gauss-Seidel updates the new values as soon as they become available, whereas Jacobi always updates the new values in terms of the values from the previous iteration.

Hence, from Eq. (21) we see that the Gauss-Seidel algorithm is given by

$$\boxed{x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=0}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n-1} a_{ij}x_j^{(k)}\right)} \tag{22}$$

The Gauss-Seidel method can be slow to converge in some applications. ***SOR*** is a fairly minimal adjustment that can help performance tremendously. For iii), we introduce a ***relaxation factor*** $\omega$ which is typically in the range $1 < \omega < 2$,[3] so that $\omega A\mathbf{x} = \omega\mathbf{b}$ takes the form

$$(\omega L + \omega D + \omega U)\mathbf{x} = \omega\mathbf{b},$$

which in turn implies

$$\begin{aligned}(D + \omega L)\mathbf{x} &= \omega\mathbf{b} - \omega U\mathbf{x} + (1-\omega)D\mathbf{x}\\ \implies \mathbf{x} &= \underbrace{(D+\omega L)^{-1}\left[(1-\omega)D - \omega U\right]}_{B}\mathbf{x} + \underbrace{\omega(D+\omega L)^{-1}\mathbf{b}}_{\mathbf{d}}.\end{aligned}$$

This yields the SOR algorithm

$$\boxed{x_i^{(k+1)} = \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) + (1-\omega)x_i^{(k)}} \tag{23}$$

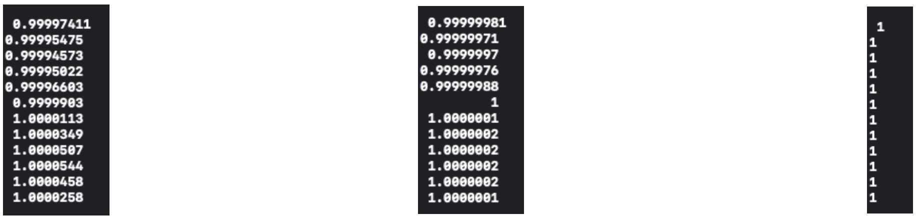
We now show our code for all three algorithms:

```
using namespace std;
using namespace Eigen;

VectorXd Jacobi(const MatrixXd &A, const VectorXd &b, VectorXd &x0, const int dim = 12,
     const int max_it = 10){

    for (int k {0}; k < max_it; k++){

        VectorXd vect = VectorXd::Zero(dim);
        VectorXd sumvect = VectorXd::Zero(dim);

        for (int i {0}; i < dim; i++){
            for (int j {0}; j < dim; j++){
                if (i !=j){
                vect(i) = A(i,j) * x0(j);
                sumvect(i) += vect(i);
                }
            }
        }

        for (int i {0}; i < dim; i++)
            x0(i) = 1.0/A(i,i) * ( b(i) - sumvect(i) );
    }
    return x0;
}


VectorXd GaussSeidel(const MatrixXd &A, const VectorXd &b, VectorXd &x0, const int dim = 12,
     const int max_it = 10){

    for (int k {0}; k < max_it; k++){

        VectorXd vect = VectorXd::Zero(dim);
        VectorXd sumvect = VectorXd::Zero(dim);
```

[3] If $\omega > 1$, then when calculating the $(k+1)^{\text{st}}$ iteration there is more weight put on the current values than when $\omega < 1$. The case $\omega = 1$ reduces to the Gauss-Seidel method.

```cpp
        VectorXd currentsumvect = VectorXd::Zero(dim);

        for (int i {0}; i < dim; i++){
            for (int j {i+1}; j < dim; j++){
                vect(i) = A(i,j) * x0(j);
                sumvect(i) += vect(i);
            }
        }

        for (int i {0}; i < dim; i++){
            for (int j {0}; j < i; j++){
                vect(i) = A(i,j) * x0(j);
                currentsumvect(i) += vect(i);
            }
                x0(i) = 1.0/A(i,i) * ( b(i) - currentsumvect(i) - sumvect(i) );
        }
    }
    return x0;
}


VectorXd SOR(const MatrixXd &A, const VectorXd &b, VectorXd &x0, const double omega = 1.1, const int dim = 12, const int max_it = 10){

    for (int k {0}; k < max_it; k++){

        VectorXd vect = VectorXd::Zero(dim);
        VectorXd sumvect = VectorXd::Zero(dim);
        VectorXd currentsumvect = VectorXd::Zero(dim);

        for (int i {0}; i < dim; i++){
            for (int j {i+1}; j < dim; j++){
                vect(i) = A(i,j) * x0(j);
                sumvect(i) += vect(i);
            }
        }

        for (int i {0}; i < dim; i++){
            for (int j {0}; j < i; j++){
                vect(i) = A(i,j) * x0(j);
                currentsumvect(i) += vect(i);
            }
                x0(i) = omega/A(i,i) * ( b(i) - currentsumvect(i) - sumvect(i) ) + (1.0 - omega) * x0(i) ;
        }
    }
    return x0;
}
```

The results from applying all three functions to an initial guess $\mathbf{x}^{(0)} = [2, \ldots, 2]^\top$ are as follows:



```
 0.99997411
0.99995475
0.99994573
0.99995022
0.99996603
 0.9999903
 1.0000113
 1.0000349
 1.0000507
 1.0000544
 1.0000458
 1.0000258
```

```
 0.99999981
0.99999971
 0.9999997
0.99999976
0.99999988
         1
 1.0000001
 1.0000002
 1.0000002
 1.0000002
 1.0000002
 1.0000001
```

```
 1
1
1
1
1
1
1
1
1
1
1
1
```

Figure 2: Resulting solution vector $\mathbf{x}$ from Jacobi (left), Gauss-Seidel (middle), and SOR (right) algorithms.

Lastly, we show the inifnity norm of the errors after ten iterations (using 8-digits precision):

| | $\|\|\widehat{\mathbf{x}} - \mathbf{x}\|\|_\infty$ |
|---:|---|
| Jacobi | 0.00005427 |
| Gauss-Seidel | 0.00000029 |
| SOR | 0 |

□

**Problem 12. Legendre Polynomials**: *Find the (reduced) QR-factorization of the Vandermonde matrix $A = [1, x, x^2, \ldots, x^{n-1}]$ by using 256 equally spaced points in $[-1, 1]$ and plot the first five Legendre polynomials (in the same graph). You may use the built-in $QR$ factorization of your computing environment.*

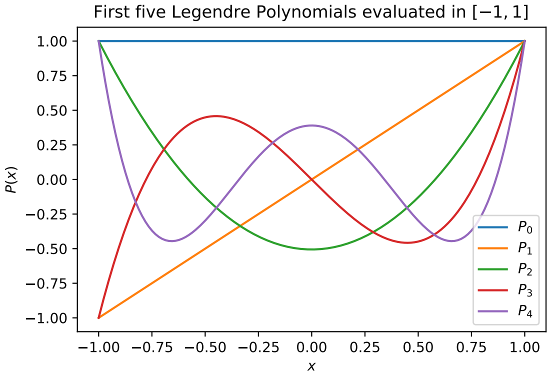
*Solution.* We recall that the set of *Legendre polynomials* is given by

$$P_k(x) = \frac{1}{2^k\, k!}\,\frac{\mathrm{d}^k}{\mathrm{d}x^k}\left[(x^2-1)^k\right] \qquad \text{for } 0 \le k \le n, \tag{24}$$

which is an orthogonal set on the interval $[-1, 1]$. It turns out that, after performing a reduced QR factorization of the Vandermonde matrix, the $n$ columns of the resulting matrix $Q$ are the first $n$ Legendre polynomials. These polynomials, however, are not yet normalized (e.g., $P_0(x) \neq 1$). The following code shows how to construct the Vandermonde matrix $A$, compute its reduced QR factorization, and get the proper normalization of $Q$ to yield the first five Legendre polynomials:

```
using namespace std;
using namespace Eigen;

int main(int argc, const char * argv[]) {
    const int n {5};
    const int m {256};
    const int a {-1};
    const int b {1};
    const double h { double(b-a)/double(m)  };

    cout << setprecision(8);

    //Generate matrix A
    MatrixXd A(m,n);
    for (int i {0}; i < m; i++) {
        for (int j {0}; j < n; j++)
            A(i,j) = pow( (i-128)*h, j );
    }

    //Reduced QR Factorization of A
    MatrixXd thinQ(MatrixXd::Identity(m,n)), Q;
    HouseholderQR<MatrixXd> qr(A);
    Q = qr.householderQ();
    thinQ = qr.householderQ() * thinQ;

    //Normalize thinQ
    MatrixXd scaleQ(m,n);
    for (int i {0}; i < m; i++) {
        for (int j {0}; j < n; j++){
            if (i == j)
                scaleQ(i,j) = 1.0/thinQ.row(m-1)(i);
            else
                scaleQ(i,j) = 0.0;
        }
    }
    Q = Q * scaleQ;
}
```

The five columns of the resulting matrix $Q$ are the first five Legendre polynomials, shown in the figure. □


First five Legendre Polynomials evaluated in [−1, 1]
P(x)
x
P0
P1
P2
P3
P4