MATH 235: Inner Product Spaces, SOLUTIONS to Assign. 7

Questions handed in: 3,4,5,6,9,10.

Contents

1 Orthogonal Basis for Inner Product Space 2
2 Inner-Product Function Space 2
3 Weighted Inner Product in $\mathbb{R}^2$ *** 2
   3.1 SOLUTIONS ............................................................... 2
4 Inner Product in $S^3$ *** 3
   4.1 SOLUTIONS ............................................................... 3
5 Inner Products in $M_{22}$ and $P_n$ *** 4
   5.1 SOLUTIONS ............................................................... 5
6 Cauchy-Schwarz and Triangle Inequalities *** 5
   6.1 SOLUTIONS ............................................................... 5
7 Length in Function Space 6
8 More on Function Space 7
9 Linear Transformations and Inner Products *** 8
   9.1 SOLUTIONS ............................................................... 8
10 MATLAB *** 9
   10.1 SOLUTIONS .............................................................. 12
      10.1.1 MATLAB Program for Solution .................................. 12
      10.1.2 Output of MATLAB Program for Solution                 13
1 Orthogonal Basis for Inner Product Space

If \( V = \mathbb{P}_3 \) with the inner product \( \langle f, g \rangle = \int_{-1}^{1} f(x)g(x)dx \), apply the Gram-Schmidt algorithm to obtain an orthogonal basis from \( B = \{1, x, x^2, x^3\} \).

2 Inner-Product Function Space

Consider the vector space \( C[0,1] \) of all continuously differentiable functions defined on the closed interval \([0,1]\). The inner product in \( C[0,1] \) is defined by \( \langle f, g \rangle = \int_{0}^{1} f(x)g(x)dx \). Find the inner products of the following pairs of functions and state whether they are orthogonal.

1. \( f(x) = \cos(2\pi x) \) and \( g(x) = \sin(2\pi x) \)
2. \( f(x) = x \) and \( g(x) = e^x \)
3. \( f(x) = x \) and \( g(x) = 3x \)

3 Weighted Inner Product in \( \mathbb{R}^2 \)

Show whether or not the following are valid inner products in \( \mathbb{R}^2 \). If one is a valid inner product, then find a nonzero vector that is orthogonal to the vector \( y = (2 \ 1)^T \).

\begin{enumerate}
    \item \( \langle u, v \rangle := 7u_1v_1 + 1.2u_2v_2 \).
    \item \( \langle u, v \rangle := -7u_1v_1 - 1.2u_2v_2 \).
    \item \( \langle u, v \rangle := 7u_1v_1 - 1.2u_2v_2 \).
\end{enumerate}

3.1 SOLUTIONS

BEGIN SOLUTION: Note that in each case, the inner product can be written as \( \langle u, v \rangle = u^T D v \), for an appropriate diagonal matrix \( D \). We see that \( \langle u, v \rangle = u^T D v = (u^T D)v^T = v^T Du = \langle v, u \rangle \). And, \( \langle \alpha u, v + w \rangle = \alpha (u^T D(v + w)) = \alpha (\langle u, v \rangle + \langle u, w \rangle) \). Therefore, the first three properties for an inner product all hold true.

1. For \( \langle u, v \rangle := 7u_1v_1 + 1.2u_2v_2 \), the diagonal matrix \( D = \begin{pmatrix} 7 & 0 \\ 0 & 1.2 \end{pmatrix} \). Therefore, \( \langle u, u \rangle := 7u_1^2 + 1.2u_2^2 \geq 0 \), with equality if and only if the vector \( u = 0 \), i.e. this is a valid innerproduct. A vector orthogonal to the given \( y \) satisfies \( \langle u, y \rangle := 7u_1(2) + 1.2u_2 = 0 \), e.g. we could set \( u_1 = -1 \) and get that \( u = \begin{pmatrix} -1 \\ 14 \end{pmatrix} \). (To understand the geometry, we could draw the ellipse \( \|u\| = 1 \) and see which vectors are orthogonal to each other.)

2. For \( \langle u, v \rangle := -7u_1v_1 - 1.2u_2v_2 \), the matrix \( D = \begin{pmatrix} -7 & 0 \\ 0 & -1.2 \end{pmatrix} \). Therefore, \( \langle u, u \rangle := -7u_1^2 - 1.2u_2^2 < 0 \), if the vector \( u \neq 0 \), i.e. this is not a valid innerproduct.
3. For \( \langle u, v \rangle := 7u_1v_1 - 1.2u_2v_2 \), the matrix \( D = \begin{pmatrix} 7 & 0 \\ 0 & -1.2 \end{pmatrix} \). Therefore, \( \langle u, u \rangle := 7u_1^2 - 1.2u_2^2 \leq 0 \), if the vector \( u \neq 0 \), and \( u_1^2 < \frac{1.2u_2^2}{7} \), i.e. this is not a valid inner product.

END SOLUTION.

4 Inner Product in \( S^3 \) ***

Let \( S^3 \) denote the vector space of \( 3 \times 3 \) symmetric matrices.

1. Prove that \( \langle S, T \rangle = \text{tr} ST \) is an inner product on \( S^3 \).

2. Let \( S_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \) and \( S_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 0 \end{pmatrix} \). Apply the Gram-Schmidt process in \( S^3 \) to find a matrix orthogonal to both \( S_1 \) and \( S_2 \).

3. Find an orthonormal basis for \( S^3 \) using the above three matrices.

4. Let \( T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \). Find the projection of \( T \) on the span of \( \{ S_1, S_2 \} \).

4.1 SOLUTIONS

BEGIN SOLUTION:

1. (a) That \( \text{tr} ST = \text{tr} TS \) was proved in class already.
   
   (b) \( \text{tr} S(T + V) = \text{tr} (ST + SV) = \text{tr} ST + \text{tr} SV \), where the last equality follows straight from the definition of taking the sum of the diagonal elements.

   (c) \( \text{tr} \alpha S = \alpha \text{tr} S \), again follows from the definition, i.e. both are equal to \( \sum_i (\alpha S_{ii}) \).
   
   (d) \( \langle S, S \rangle = \text{tr} SS = \text{tr} S^2 = \sum_{ij} S_{ij}^2 \geq 0 \) and it is = 0 if an only if each \( S_{ij}^2 = 0 \), i.e. if and only if \( S = 0 \).

2. We first note that \( \text{tr} S_1 S_2 = 1 + 1 + 1 - 3 = 0 \), i.e. \( S_1 \perp S_2 \). Therefore, we can apply one step of the process. But we need a matrix that is linearly independent to both \( S_1, S_2 \). By observation the matrix 

\[
S_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}
\]

works and in fact is orthogonal to both. Note that any matrix of the form 

\[
\begin{pmatrix} -2d & a & d \\ a & 0 & b \\ d & b & c \end{pmatrix}
\]

does the job. Also, note we could also choose a random matrix (linearly independent with
S_1, S_2) and then apply one step of the process, e.g. choose \( R = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \). Then we can find the third matrix by subtracting the projection onto the span of the first two, i.e.

\[
\hat{S}_3 = R - \frac{\langle R, S_1 \rangle}{\langle S_1, S_1 \rangle} S_1 - \frac{\langle R, S_2 \rangle}{\langle S_2, S_2 \rangle} S_2 = R - \frac{1}{2} S_1 + \frac{1}{6} S_2
\]

\[
= \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} + \frac{1}{6} \begin{pmatrix} 1 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 0 \end{pmatrix}
\]

\[
= \begin{pmatrix} 2/3 & 1 & -1/3 \\ 1 & 0 & 0 \\ -1/3 & 0 & 0 \end{pmatrix}
\]

3. An orthonormal basis using \( S_3 \) is

\[
\left\{ \frac{1}{2} S_1, \frac{1}{2\sqrt{3}} S_2, S_3 \right\}
\]

Alternatively, using \( \hat{S}_3 \), we get

\[
\left\{ \frac{1}{2} S_1, \frac{1}{2\sqrt{3}} S_2, \frac{2\sqrt{5}}{3} \hat{S}_3 \right\}
\]

4. The projection of \( T \) is

\[
(\frac{1}{2} \text{tr} T S_1) S_1 + (\frac{1}{2\sqrt{3}} \text{tr} T S_2) S_2 = (\frac{1}{2} 4) S_1 + (\frac{1}{2\sqrt{3}} 0) S_2 = 2 S_1.
\]

END SOLUTION.

5 Inner Products in \( \mathcal{M}_{22} \) and \( \mathbb{P}_2 \) ***

Show which of the following are valid inner products.

1. Let \( A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \) and \( B = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \) be real matrices.

   (a) Define \( \langle A, B \rangle := a_1 b_1 + 2a_2 b_3 + 3a_3 b_2 + a_4 b_4 \).

   (b) Define \( \langle A, B \rangle := \text{tr} B^T A \).

2. Let \( p(x) \) and \( q(x) \) be polynomials in \( \mathbb{P}_2 \).

   (a) Define \( \langle p, q \rangle := p(-1) q(-1) + p(\frac{1}{2}) q(\frac{1}{2}) + p(-3) q(-3) \).

   (b) Define \( \langle p, q \rangle := p(-3) q(-3) + p(\frac{1}{2}) q(\frac{1}{2}) + p(-1) q(-1) \).
5.1 SOLUTIONS

BEGIN SOLUTION:

1. (a) Note that \( \langle A, B \rangle := a_1 b_1 + 2a_2 b_3 + 3a_3 b_2 + a_4 b_4 = \text{tr} AB. \) Let \( A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \) Then
\[
\langle A, A \rangle = \text{tr} AA = \text{tr} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = 0, \text{ i.e. } \|A\| = 0, \text{ but } A \neq 0. \text{ Therefore, this is not a valid inner product.}
\]

(b) For any \( m \times n \) matrix \( A \), define \( \text{vec}(A) = \left( \begin{array}{c} A(:,1) \\ \vdots \\ A(:,n) \end{array} \right) \), i.e. \( \text{vec}(A) \) is the vector formed from \( A \) using the columns of \( A \). Note that \( \text{tr} B^T A = \sum_{ij} B_{ij} A_{ij} = \text{vec}(B)^T \text{vec}(A) \). We can now show that this is a valid inner product in the usual way that the standard dot product is an inner product.

2. (a) This follows as does the Example 2 in the text on page 429.

(b) Changing the order of the points does not change the verification of any of the rules for verifying this is an inner product.

END SOLUTION.

6 Cauchy-Schwarz and Triangle Inequalities ***

1. Let \( D = \begin{pmatrix} 3 & 0 & 0 \\ 0 & \pi & 0 \\ 0 & 0 & 2 \end{pmatrix} \). Consider the inner product defined on \( \mathbb{R}^3 \) by: \( \langle u, v \rangle = u^T D v. \) Let \( u = \begin{pmatrix} -2 \\ 3 \\ -7 \end{pmatrix} \) and \( v = \begin{pmatrix} -4 \\ 8 \\ 9 \end{pmatrix} \). Verify that both inequalities hold.

2. Find the projection of \( v \) on the span of \( u \) and call it \( w \). Verify that equality holds for the Cauchy-Schwarz inequality applied to \( u, w. \)

6.1 SOLUTIONS

BEGIN SOLUTION: Following are the MATLAB lines and the output for the solution.

```matlab
D=sym(diag([3 pi 2]));
u=[-2 3 -7]';
v=[-4 8 9]';
lhs=abs(double(u'*D*v));
rhs=sqrt(double(u'*D*u))*sqrt(double(v'*D*v))
rhs =
238.4100
disp('verify C.S. inequality, i.e. that rhs - lhs is nonnegative')
verify C.S. inequality, i.e. that rhs - lhs is nonnegative
```


rhs-lhs
ans =
  211.8082

lhs= (double((u+v)'*D*(u+v)));
rhs= (double((u)'*D*(u))) + (double((v)'*D*(v)));
disp('verify triangle inequality, i.e. that rhs - lhs is nonnegative')

verify triangle inequality, i.e. that rhs - lhs is nonnegative
rhs-lhs
ans =
  53.2036

disp('find w, the projection on the span of u')
find w, the projection on the span of u
w= (double(v'*D*u))/(double(u'*D*u))*u

w =
  0.3848
  -0.5772
  1.3467

lhs=abs(double(u'*D*w));
rhs= sqrt(double(u'*D*u))*sqrt(double(w'*D*w))
rhs =
  26.6018
disp('verify that rhs - lhs is now zero, i.e. C.S. inequality holds with equality')

verify that rhs - lhs is now zero, i.e. C.S. inequality holds with equality
rhs-lhs
ans =
  0

echo off

END SOLUTION.

7 Length in Function Space

Let \( W = \text{Span}\{1, \cos(x), \cos(2x)\} \in C[-\pi, \pi] \) with inner product
\[ <f, g> = \int_{-\pi}^{\pi} f(x)g(x)dx. \] It is known that \( \left\{ \frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}} \cos(x), \frac{1}{\sqrt{\pi}} \cos(2x) \right\} \) is an orthonormal basis for \( W \). A function \( f \in W \) satisfies \( \int_{-\pi}^{\pi} f(x)dx = a, \)
\( \int_{-\pi}^{\pi} f(x) \cos(x)dx = b \) and \( \int_{-\pi}^{\pi} f(x) \cos(2x)dx = c. \) Find an expression for \( ||f|| \) in terms of \( a, b \) and \( c. \)
More on Function Space

Consider the vector space $P_2$ consisting of polynomials of degree at most 2 together with the inner product

$$< f, g > = \int_0^1 f(x)g(x)dx , f, g \in P_2 .$$

Let $W$ be the subspace of $P_2$ having a basis $\{1, x\}$.

1. Find an orthonormal basis for $W$.
2. Let $h(x) = x^2 + 1$. Find the vector in $W$ that is closest to $h(x)$.
9 Linear Transformations and Inner Products ***

Let $V$ be a real inner product space (with the usual notation) and let \( \{v_1, v_2, \ldots, v_n\} \) be an orthonormal basis for $V$ (so $V$ is finite dimensional). Fix a linear transformation $T : V \to \mathbb{R}$. Define $v_0 = T(v_1)v_1 + T(v_2)v_2 + \cdots + T(v_n)v_n$.

1. Show that for all $v \in V$, $T(v) = \langle v_0, v \rangle$.

2. Prove that, for vectors $u_0, w_0 \in V$, if $\langle u_0, v \rangle = \langle w_0, v \rangle$ for all $v \in V$, then $u_0 = w_0$.

3. Deduce from Item 2 that $v_0$ is the only vector in $V$ with the displayed property in Item 1.

**NOTE:** This question shows how a suitable inner product produces a 1–1 correspondence between vectors in the vector space and linear functionals $T$ from it to the reals. This is used to identify tangent vectors with what are called cotangent vectors in differential geometry, general relativity and in other symbol manipulations of mathematical physics, where it is sometimes referred to as “moving indices up-and-down”. A slightly surprising consequence of Item 3 is that the vector $v_0$, despite appearances, depends only on the inner product and $T$, but not on the choice of orthonormal basis.

9.1 SOLUTIONS

**BEGIN SOLUTION:**

1. Since $S := \{v_1, v_2, \ldots, v_n\}$ is an orthonormal basis, we get that

\[
v = \langle v, v_1 \rangle v_1 + \langle v, v_2 \rangle v_2 + \ldots + \langle v, v_n \rangle v_n, \quad \forall v \in V.
\]

Therefore, for all $v \in V$,

\[
\langle v_0, v \rangle = \langle T(v_1)v_1 + T(v_2)v_2 + \cdots + T(v_n)v_n, v \rangle = T(v_1)\langle v_1, v \rangle + T(v_2)\langle v_2, v \rangle + \cdots + T(v_n)\langle v_n, v \rangle
\]

and also

\[
T(v) = T(\langle v, v_1 \rangle v_1 + \langle v, v_2 \rangle v_2 + \ldots + \langle v, v_n \rangle v_n) = \langle v, v_1 \rangle T(v_1) + \langle v, v_2 \rangle T(v_2) + \ldots + \langle v, v_n \rangle T(v_n).
\]

The result now follows since $\langle u, w \rangle = \langle w, u \rangle, \forall u, w \in V$, a real inner product space, and $T(u) \in \mathbb{R}, \forall u \in V$.

2. Suppose that $\langle u_0, v \rangle = \langle w_0, v \rangle$ for all $v \in V$. Then, in particular, this is true with $v$ replaced by elements of the orthonormal basis $v_i, i = 1, 2, \ldots, n$. Therefore, the expansion in (1) (with $v$ replace by $u_0$ or by $w_0$) is the same. This implies that $u_0 = w_0$.

3. Suppose that $u_0 \in V$ and for all $v \in V$, $T(v) = \langle f u_0, v \rangle$. From Item 2 this means that for all $v \in V$, $T(v) = \langle u_0, v \rangle = \langle v_0, v \rangle$, i.e. $u_0 = v_0$ is the only such vector.

**END SOLUTION.**
10 MATLAB ***

Gram-Schmidt & QR Factorization

Overview of the Gram-Schmidt Process

Given a basis \( \{x_1, \ldots, x_p\} \) for a subspace \( W \) of \( \mathbb{R}^n \), define

\[
\begin{align*}
v_1 &= x_1 \\
v_2 &= x_2 - \frac{x_2 \cdot v_1}{v_1 \cdot v_1} v_1 \\
v_3 &= x_3 - \frac{x_3 \cdot v_1}{v_1 \cdot v_1} v_1 - \frac{x_3 \cdot v_2}{v_2 \cdot v_2} v_2 \\
&\quad \vdots \\
v_p &= x_p - \frac{x_p \cdot v_1}{v_1 \cdot v_1} v_1 - \frac{x_p \cdot v_2}{v_2 \cdot v_2} v_2 - \cdots - \frac{x_p \cdot v_{p-1}}{v_{p-1} \cdot v_{p-1}} v_{p-1}
\end{align*}
\]

Then \( \{v_1, \ldots, v_p\} \) is an orthogonal basis for \( W \).

When working on a computer, a basis is often presented as a matrix \( A \) with linearly independent columns. The columns of matrix \( A \) are a basis for its column space (or range). The Gram-Schmidt process can be applied to the column vectors of matrix \( A \) so that the output of the process is an orthogonal set of vectors.

For example, let

\[
\begin{align*}
x_1 &= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \\
x_2 &= \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \\
x_3 &= \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}
\end{align*}
\]

Then, by inspection, \( \{x_1, x_2, x_3\} \) are linearly independent and thus form a basis for the subspace \( W \) of \( \mathbb{R}^3 \). Consider the following matrix \( A \) formed by using the basis \( \{x_1, x_2, x_3\} \) as column vectors:

\[
A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}
\]

Matrix \( A \) will be used to illustrate the Gram-Schmidt process in MATLAB.

First, enter matrix \( A \).

\[
\begin{align*}
\text{>> } A &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} & \%\text{Create matrix } A
\end{align*}
\]

Next, calculate the orthogonal basis \( \{v_1, v_2, v_3\} \) by using the Gram-Schmidt algorithm stated above. Recall that in MATLAB, the notation \( A(:,1) \) selects the first column of \( A \), \( A(:,2) \) selects the second column, and so on... Also recall that \( v1' \) is the short form for \( \text{transpose}(A) \) (changes a row vector to a column vector, or vice versa).

\[
\begin{align*}
\text{>> } v1 &= A(:,1) & \%\text{Calculate } v1 \\
\text{>> } v2 &= A(:,2) - (A(:,2)'*v1)/(v1'*v1)*v1 & \%\text{Calculate } v2 \\
\text{>> } v3 &= A(:,3) - (A(:,3)'*v1)/(v1'*v1)*v1 - (A(:,3)'*v2)/(v2'*v2)*v2 & \%\text{Calculate } v3
\end{align*}
\]
MATLAB tells us that by using the Gram-Schmidt process on matrix $A$, the corresponding orthogonal basis is

\[
\begin{bmatrix}
1 \\
1 \\
1 \\
1 \\
\end{bmatrix}, \quad \begin{bmatrix}
-0.75 \\
0.25 \\
0.25 \\
0.25 \\
\end{bmatrix}, \quad \begin{bmatrix}
0 \\
-0.6667 \\
0.3333 \\
0.3333 \\
\end{bmatrix}
\]

To convert from an Orthogonal basis to an Orthonormal Basis:
We must simply divide each orthogonal vector by its norm:

\[
\begin{align*}
&>> \text{vn1 = v1/norm(v1)} \quad \%\text{Calculate normalized v1} \\
&>> \text{vn2 = v2/norm(v2)} \quad \%\text{Calculate normalized v2} \\
&>> \text{vn3 = v3/norm(v3)} \quad \%\text{Calculate normalized v3}
\end{align*}
\]

The following theorem provides a method for checking whether the columns of a matrix are orthonormal:

An $m \times n$ matrix $U$ has orthonormal columns if and only if

\[
U^TU = I
\]

Using this theorem and MATLAB, we can check whether $\{vn1, vn2, vn3\}$ forms an orthonormal basis. First define a new matrix $U$ that uses $\{vn1, vn2, vn3\}$ as its column vectors.

\[
\begin{align*}
&>> U = [vn1 \ vn2 \ vn3] \quad \%\text{Create matrix U} \\
&>> \text{transpose(U)} * U \quad \%\text{Apply theorem}
\end{align*}
\]

Indeed, MATLAB returns the identity matrix $I$, so by the theorem, $\{vn1, vn2, vn3\}$ form an orthonormal basis.
**QR Factorization**

QR factorization is the *matrix version* of the Gram-Schmidt method. QR factorization is particularly useful because computer algorithms exist that implement this factorization method with good control over computer round-off error. Additionally, there is no simple method for computing eigenvalues of matrices that are larger than $3 \times 3$ since numerically it is difficult to calculate the roots of the characteristic polynomial. Using QR factorization iteratively is the most successful algorithm that exists today for computing eigenvalues. For these reasons, QR is an important factorization technique in Linear Algebra.

**Overview of QR Factorization**

If $A$ is an $m \times n$ matrix with linearly independent columns, then $A$ can be factored as

$$A = QR$$

where $Q$ is an $m \times n$ matrix with orthonormal columns consisting of an orthonormal basis of the column space, and $R$ is an invertible, upper triangular matrix.

To illustrate this method in MATLAB, consider the following matrix $A$:

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

This is the same matrix that was used in the section on the Gram-Schmidt process. We have already used MATLAB to calculate the orthonormal basis for matrix $A$ as:

$$vn_1 = \begin{bmatrix} 0.5000 \\ 0.5000 \\ 0.5000 \\ 0.5000 \end{bmatrix}, \quad vn_2 = \begin{bmatrix} -0.8660 \\ 0.2887 \\ 0.2887 \end{bmatrix}, \quad vn_3 = \begin{bmatrix} 0 \\ -0.8165 \\ 0.4082 \end{bmatrix}$$

As such, according to the description of the QR factorization method, it must be that we find:

$$Q = \begin{bmatrix} 0.5000 & -0.8660 & 0 \\ 0.5000 & 0.2887 & -0.8165 \\ 0.5000 & 0.2887 & 0.4082 \\ 0.5000 & 0.2887 & 0.4082 \end{bmatrix}$$

Let’s use MATLAB to verify this result. MATLAB’s command for QR Factorization is

$$[Q \ R] = \text{qr}(A)$$

where $A$ is an $m \times n$ matrix, $Q$ is returned as an $m \times m$ orthogonal matrix and $R$ is returned as an $m \times n$ upper triangular matrix.

This MATLAB command for QR factorization is very *general* since it applies to *any* matrix and it returns more information than required in some cases.

If $A$ has rank $n$, then only the first $n$ columns of $Q$ will be an orthonormal basis for the column space of $A$.

Enter the following in MATLAB:

```matlab
>> A = [1 0 0; 1 1 0; 1 1 1; 1 1 1] %Create matrix A
>> rank(A) %Compute rank of matrix A
```

Since $A$ is a $4 \times 3$ matrix and its rank is 3, we should expect that the orthonormal basis is the first 3 columns of $Q$ in result of the following command:
>> [Q R] = qr(A)   %Calculate the QR factorization of A

Indeed, the first 3 columns of Q match those of the orthonormal basis that we calculated using the Gram-Schmidt process.

Notice that the MATLAB qr command is much easier to use than the Gram-Schmidt process. The qr command in MATLAB is also less computationally demanding on your computer (more efficient and not prone to rounding errors). The Gram-Schmidt method is numerically unstable when implemented on a computer—the vectors that are returned are often not quite orthogonal because of rounding errors in the calculation. Therefore, the QR factorization method is usually preferred when using a computer.

Consider the following matrix A:

\[
A = \begin{bmatrix}
4 & -5 & 1 \\
1 & 2 & 1 \\
-2 & 5 & 2 \\
4 & -8 & 8
\end{bmatrix}
\]

Use MATLAB to determine if the columns of A are linearly independent? State which MATLAB commands you used and explain your answer.

Using MATLAB, find the QR factorization of matrix A (i.e. find Q and R such that \( A = QR \)).

What is an orthonormal basis for the column space of matrix A?

Given any \( m \times n \) matrix B that has already been entered in MATLAB, what single command could you use to determine if matrix B has orthonormal columns?

### 10.1 SOLUTIONS

BEGIN SOLUTION: Following is a MATLAB program and the output:

#### 10.1.1 MATLAB Program for Solution

diary off
echo off
!rm temp.txt
diary temp.txt
format compact
echo on

A=[ 4 -5 1; 1 2 1; -2 5 2; 4 -8 8]
disp('first check if the columns are linearly independent - using RREF')
rrefA=rref(A)
disp('add the number of nonzero columns and display the rank')
rankA=sum(sum(rrefA));
disp(['The rank is ',num2str(rankA)]);
if rankA==3,
echo off
disp('The rank is 3 and the columns are linearly independent')
echo on
end
disp('now find the qr factorization')
\[ \begin{bmatrix} Q & R \end{bmatrix} = \text{qr}(A) \]

disp('since the rank is 3, the first 3 columns of Q form an orthonormal basis for the column space

\[ m=5; n=3; \]
\[ B = \text{randn}(m, n) \]

disp('to determine if B has orthonormal columns we can check: \( B'B = I? \));
\[ B' * B \]

disp('so B does not have orthonormal columns')
\[ [Q, R] = \text{qr}(B, 0); \quad \% \text{use the economy version of qr} \]
\[ Q \]

disp('check if Q has orthonormal columns')
\[ Q' * Q \]

disp('so Q does have orthonormal columns')

echo off
diary off

10.1.2 Output of MATLAB Program for Solution

The output now follows.
\[ A = \begin{bmatrix} 4 & -5 & 1; & 1 & 2 & 1; & -2 & 5 & 2; & 4 & -8 & 8 \end{bmatrix} \]
\[ A = \]
\[ \begin{array}{ccc}
4 & -5 & 1 \\
1 & 2 & 1 \\
-2 & 5 & 2 \\
4 & -8 & 8 \\
\end{array} \]

disp('first check if the columns are linearly independent - using RREF')
first check if the columns are linearly independent - using RREF
\[ \text{rref}A = \text{rref}(A) \]
\[ \text{rref}A = \]
\[ \begin{array}{ccc}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
\end{array} \]

disp('add the number of nonzero columns and display the rank')
add the number of nonzero columns and display the rank
\[ \text{rank}A = \text{sum}(\text{sum}(\text{rref}A)); \]

disp(['The rank is ', num2str(rankA)]);
The rank is 3
if rankA == 3,
echo off
The rank is 3 and the columns are linearly independent
end
disp('now find the qr factorization')
now find the qr factorization
\[ [Q, R] = \text{qr}(A) \]
Q =
-0.6576  -0.3267   0.3665  -0.5714
-0.1644  -0.7960  -0.1136  0.5714
 0.3288  -0.3861  -0.6452  -0.5714
-0.6576   0.3326  -0.6607   0.1429
R =
-6.0828   9.8639  -5.4252
 0  -4.5500   0.7663
 0      0    -6.3230
 0      0      0

disp('since the rank is 3, the first 3 columns of Q form an orthonormal basis for the column space of
m=5; n=3;
B = randn(m, n)
B =
-0.3999   0.6686  -1.6041
 0.6900   1.1908   0.2573
 0.8156  -1.2025  -1.0565
 0.7119  -0.0198   1.4151
 1.2902  -0.1567  -0.8051
disp('to determine if B has orthonormal columns we can check: B''B=I?');
to determine if B has orthonormal columns we can check: B'B=I?
B'*B
ans =
 3.4728  -0.6427  -0.0740
-0.6427   3.3360   0.6024
-0.0740   0.6024   6.4062
disp('so B does not have orthonormal columns')
so B does not have orthonormal columns

[Q, R] = qr(B, 0);  % use the economy version of qr
Q
Q =
-0.2146  -0.3315  -0.6860
 0.3703  -0.7351   0.0122
 0.4377   0.5863  -0.3374
 0.3820  -0.0624   0.5618
 0.6924  -0.0458  -0.3159
disp('check if Q has orthonormal columns')
check if Q has orthonormal columns
Q'*Q
ans =
 1.0000   0.0000   0.0000
-0.0000   1.0000   0.0000
-0.0000   0.0000   1.0000
disp('so Q does have orthonormal columns')
so Q does have orthonormal columns
echo off

END SOLUTION.