Fundamental concepts in vector analysis: scalar product, magnitude, angle, orthogonality, projection.
Instructor: Nam Sun Wang

In a linear vector space (LVS), there are rules on 1) addition of two vectors and 2) multiplication by a scalar. The resulting vector must also lie in the same LVS. We now impose another rule: multiplication of two vectors to yield a scalar.

Real Euclidean Space. Real Euclidean Space is a subspace of LVS where there is a real-valued function (called scalar product) defined on pairs of vectors \( x \) and \( y \) with the following four properties.

1. Commutative  \( (x, y) = (y, x) \)
2. Associative \( (\alpha x, y) = \alpha(x, y) \)
3. Distributive \( (x + y, z) = (x, z) + (y, z) \)
4. Positive magnitude \( (x, x) > 0 \) unless \( x = 0 \)

Note that it is up to us to come up with a definition of the scalar product. Any definition that satisfies that above four properties is a valid one. Not only do we define the scalar product accordingly for different types of vectors, we conveniently adopt different definitions of scalar product for different applications even for one type of vectors. A scalar product is also known as an inner product or a dot product. We have vectors in linear vector space if we come up with rules of addition and multiplication by a scalar; furthermore, these vectors are in real Euclidean space if we come up with a rule on multiplication of two vectors.

There are two metrics of a vector: magnitude and direction. They are defined in terms of the scalar product.

Magnitude, or length or Euclidean norm, of \( x \): \[ |x| = \sqrt{(x, x)} \]

Direction or angle between two vectors \( x \) and \( y \): \[ \cos(\theta) = \frac{(x, y)}{|x| \cdot |y|} \]

Orthogonal means (is defined as) \( \theta = \pi/2 = 90^\circ \) or \( (x, y) = 0 \).

Schwarz Inequality. \[ |(x, y)| \leq |x| \cdot |y| \]

Schwarz inequality results from the four properties of scalar product. It does not depend on what types of vectors or how the scalar product is defined.

Special case. For linearly dependent vectors, \( |(x, y)| = |x| \cdot |y| \)

Triangular Inequality. \[ |x + y| \leq |x| + |y| \]

Proof. \[ (|x + y|)^2 = (x + y, x + y) = (x, x) + 2 \cdot (x, y) + (y, y) \]

\[ = |x|^2 + 2(x, y) + |y|^2 \leq |x|^2 + 2 |x| \cdot |y| + |y|^2 = (|x| + |y|)^2 \]

Thus, \( |x + y| \leq |x| + |y| \)

Special case. For orthogonal vectors \( x \) and \( y \), the scalar product is 0, i.e., \( (x, y) = 0 \). Thus, \( (|x + y|)^2 = (|x|)^2 + (|y|)^2 \) ... Pythagorean theorem (True not just for the conventional distances, but for all vectors in general).
Example -- A column of real numbers.

Define \( (x, y) = \sum_{i=1}^{n} x_i y_i \)

Show that this definition is valid by demonstrating that it satisfies the four properties.

Property #1. \( (x, y) = (y, x) \)

\[
(x, y) = \sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{n} y_i x_i = (y, x) \quad \text{... ok}
\]

Property #2. \( (\alpha \cdot x, y) = \alpha \cdot (x, y) \)

\[
(\alpha \cdot x, y) = \sum_{i=1}^{n} (\alpha x_i) y_i = \alpha \sum_{i=1}^{n} x_i y_i = \alpha \cdot (x, y) \quad \text{... ok}
\]

Property #3. \( (x + y, z) = (x, z) + (y, z) \)

\[
(x + y, z) = \sum_{i=1}^{n} (x_i + y_i) z_i = \sum_{i=1}^{n} x_i z_i + \sum_{i=1}^{n} y_i z_i = (x, z) + (y, z) \quad \text{... ok}
\]

Property #4. \( (x, x) > 0 \) unless \( x = 0 \)

\[
(x, x) = \sum_{i=1}^{n} x_i x_i > 0 \text{ unless all elements are 0.}
\]

Thus, the definition is a valid one. Being vectors in real Euclidean space, they automatically inherit all sorts of properties, including Schwarz inequality and triangular inequality, Pythagorean rule, etc.

Schwarz inequality:
\[
\left( \sum_{i=1}^{n} x_i y_i \right)^2 \leq \left( \sum_{i=1}^{n} x_i x_i \right) \left( \sum_{i=1}^{n} y_i y_i \right)
\]

Is the following definition of a scalar product a valid one?

\[
(x, y) = \sum_{i=1}^{n} w_i x_i y_i
\]

What about the following?

\[
(x, y) = x_1 y_1 + x_2 y_2
\]

What about the following?

\[
(x, y) = x_1 y_2 + x_2 y_1
\]
Example -- continuous functions $f(t)$ in $0 \leq t \leq 1$.

Define

$$\langle f, g \rangle = \int_0^1 f(t) \cdot g(t) \, dt$$

Show that this definition is valid by demonstrating that it satisfies the four properties.

**Property #1.**

$$(f, g) = (g, f) \quad \text{and} \quad \int_0^1 f(t) \cdot g(t) \, dt = \int_0^1 g(t) \cdot f(t) \, dt = (f, g) \quad \text{... ok}$$

**Property #2.**

$$(\alpha \cdot x, y) = \alpha (x, y) \quad \text{and} \quad \int_0^1 (\alpha \cdot f(t)) \cdot g(t) \, dt = \alpha \int_0^1 f(t) \cdot g(t) \, dt = \alpha (f, g) \quad \text{... ok}$$

**Property #3.**

$$(f + g, h) = (f, h) + (g, h) \quad \text{and} \quad \int_0^1 (f(t) + g(t)) \cdot h(t) \, dt = \int_0^1 f(t) \cdot h(t) \, dt + \int_0^1 g(t) \cdot h(t) \, dt = (f, h) + (g, h) \quad \text{... ok}$$

**Property #4.**

$$(f, f) \geq 0 \quad \text{unless} \quad f = 0 \quad \text{and} \quad \int_0^1 f(t) \cdot f(t) \, dt > 0 \quad \text{... ok}$$

Thus, the definition is a valid one.

Schwarz inequality:

$$\left| \int_0^1 f(t) \cdot g(t) \, dt \right| \leq \sqrt{\int_0^1 f(t) \cdot f(t) \, dt} \cdot \sqrt{\int_0^1 g(t) \cdot g(t) \, dt}$$

Is the following definition of a scalar product valid one?

$$(f, g) = \int_0^1 w(t) \cdot f(t) \cdot g(t) \, dt$$

What about the following, where $\alpha$ and $\beta$ are within $[0, 1]$, say, $[0.5, 0.6]$?

$$(f, g) = \int_0^\beta \int_\alpha \int f(t) \cdot g(t) \, dt$$
Gram-Schmidt Process. Given a set of \( n \) linearly independent vectors in real Euclidean space, \( f_1, f_2, \ldots, f_n \), we can construct a set of \( n \) vectors \( g_1, g_2, \ldots, g_n \) that are pair-wise orthogonal.

\[
\langle g_i, g_j \rangle = 0 \quad \text{for} \quad i \neq j
\]

First, we choose any one of the given vectors \( f_i \) to be the first vector. Here we simply choose \( f_1 \).

\[
g_1 = f_1
\]

To construct a second vector that is orthogonal to the first vector, we take any one of the remaining given vectors (typically \( f_2 \)) and subtract from it the component aligned with the first vector \( g_1 \). Whatever is left behind has nothing in common with the first vector, and, therefore, is orthogonal to the first vector.

\[
g_2 = f_2 - \frac{\langle f_2, g_1 \rangle}{\langle g_1, g_1 \rangle} g_1
\]

To construct a third vector that is orthogonal to the first two vectors that have already been constructed, we take any one of the remaining given vectors (typically \( f_3 \)) and subtract from it both the component aligned with the first vector \( g_1 \) and that aligned with the second vector \( g_2 \). Whatever is left behind has nothing in common with the first two vectors, and, therefore, is orthogonal to the first two vectors.

\[
g_3 = f_3 - \frac{\langle f_3, g_1 \rangle}{\langle g_1, g_1 \rangle} g_1 - \frac{\langle f_3, g_2 \rangle}{\langle g_2, g_2 \rangle} g_2
\]

\[
g_4 = f_4 - \frac{\langle f_4, g_1 \rangle}{\langle g_1, g_1 \rangle} g_1 - \frac{\langle f_4, g_2 \rangle}{\langle g_2, g_2 \rangle} g_2 - \frac{\langle f_4, g_3 \rangle}{\langle g_3, g_3 \rangle} g_3
\]

We repeat the process.

\[
g_i = f_i - \frac{\langle f_i, g_1 \rangle}{\langle g_1, g_1 \rangle} g_1 - \frac{\langle f_i, g_2 \rangle}{\langle g_2, g_2 \rangle} g_2 - \cdots - \frac{\langle f_i, g_{i-1} \rangle}{\langle g_{i-1}, g_{i-1} \rangle} g_{i-1} = f_i - \sum_{j=1}^{i-1} \frac{\langle f_i, g_j \rangle}{\langle g_j, g_j \rangle} g_j
\]

Finally, the last orthogonal vector is,

\[
g_n = f_n - \frac{\langle f_n, g_1 \rangle}{\langle g_1, g_1 \rangle} g_1 - \frac{\langle f_n, g_2 \rangle}{\langle g_2, g_2 \rangle} g_2 - \cdots - \frac{\langle f_n, g_{n-1} \rangle}{\langle g_{n-1}, g_{n-1} \rangle} g_{n-1} = f_n - \sum_{j=1}^{n-1} \frac{\langle f_n, g_j \rangle}{\langle g_j, g_j \rangle} g_j
\]

In general, we generate orthogonal basis function by the Gram-Schmidt orthogonalization process,

\[
g_i = f_i - \sum_{j=1}^{i-1} \frac{\langle g_j, f_i \rangle}{\langle g_j, g_j \rangle} g_j \quad \text{for} \quad i = 1, 2, \ldots, n
\]

We have an orthonormal basis, if the basis vectors are normalized to have length of unity.

\[
\langle g_i, g_j \rangle = 0 \quad \text{for} \quad i \neq j
\]

\[
\langle g_i, g_i \rangle = 1 \quad \text{for} \quad i = j
\]
Example. All polynomials of degree \( n \) or less in \(-1 \leq t \leq 1\).

Define scalar project.

\[
(p, q) = \int_{-1}^{1} p(t)q(t) \, dt
\]

Given \( f_1(t) = 1 \) \( f_2(t) = t \) \( f_3(t) = t^2 \) \( ... \) \( f_{n+1} = t^n \)

The results of Gram-Schmidt orthogonalization process is a set of polynomials known as Legendre polynomials, \( L_1, L_2, ..., L_{n+1} \), which form an orthogonal basis for the set of all polynomials of degree \( n \) or less. Thus, we have the choice of expressing a continuous function in \(-1 \leq t \leq 1\) as a linear combination of the power series (which are not orthogonal) or as a linear combination of the Legendre polynomials (which are orthogonal). Because the basis is defined only for finite dimensions, we can only approximate, not represent exactly, a general continuous function in \(-1 \leq t \leq 1\).

\[
\begin{align*}
g_1 &= 1 \\
g_2 &= t \\
g_3 &= t^2 - \frac{1}{3} \\
g_4 &= t^3 - \frac{3}{5} t \\
&\vdots \\
g_i &= \frac{1}{2^i \cdot i!} \frac{d^i}{dt^i} (t^2 - 1)^i
\end{align*}
\]

Example. All functions \( f(t) \) of the form in \( 0 \leq t \leq \pi \).

\[
f(t) = \sum_{j=0}^{n} \alpha_j \cos(j \cdot t) + \sum_{j=1}^{n} \beta_j \sin(j \cdot t)
\]

The following set of \( 2n+1 \) vectors provide a basis (not orthogonal in \([0 \ \pi]\)).

\[
\begin{align*}
f_1(t) &= 1 \\
f_2(t) &= \sin(t) \\
f_3(t) &= \cos(t) \\
f_4(t) &= \sin(2 \cdot t) \\
f_5(t) &= \cos(2 \cdot t) \\
&\vdots \\
f_{2n}(t) &= \sin(n \cdot t) \\
f_{2n+1}(t) &= \cos(n \cdot t)
\end{align*}
\]
Orthogonalization process

\[ g_1 = f_1 = 1 \]

\[ g_2 = f_2 - \frac{(f_2 \cdot g_1)}{(g_1 \cdot g_1)} \cdot g_1 = \sin(t) - \int_{0}^{\pi} 1 \cdot \sin(t) \, dt \]

\[ g_3 = f_3 - \frac{(f_3 \cdot g_1)}{(g_1 \cdot g_1)} \cdot g_1 - \frac{(f_3 \cdot g_2)}{(g_2 \cdot g_2)} \cdot g_2 = \cos(t) - \int_{0}^{\pi} 1 \cdot \cos(t) \, dt \]

\[ g_4 = f_3 - \frac{(f_4 \cdot g_1)}{(g_1 \cdot g_1)} \cdot g_1 - \frac{(f_4 \cdot g_2)}{(g_2 \cdot g_2)} \cdot g_2 - \frac{(f_4 \cdot g_3)}{(g_3 \cdot g_3)} \cdot g_3 = \sin(2t) - \int_{0}^{\pi} 1 \cdot \sin(2t) \, dt \]

The fact that we can apply the Gram-Schmidt orthogonalization process means that we have proven the **Existence** (but not uniqueness) of orthogonal or orthonormal basis. In fact, the set of orthogonal basis is not unique. Had we started the orthogonalization process with a different function (say, \( \sin(t) \)), we would have resulted in a different set of orthogonal basis.
Non-Example. vector = a column of three numbers.

Given three basis vectors $x^0$, $x^1$, $x^2$

\[
x = \begin{pmatrix} x^0 \\ x^1 \\ x^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}
\]

Define scalar product \(\prod(x, y) = x_0y_0 + x_1y_1\)

Gram-Schmidt orthogonalization in the order $x^0$, $x^1$, $x^2$

\[
g_0 = x^0
\]
\[
g_1 = x^1 - \frac{\langle x^1, g_0 \rangle}{\langle g_0, g_0 \rangle} g_0
\]
\[
g_2 = x^2 - \frac{\langle x^2, g_0 \rangle}{\langle g_0, g_0 \rangle} g_0 - \frac{\langle x^2, g_1 \rangle}{\langle g_1, g_1 \rangle} g_1
\]

Check orthogonality

\[
\prod(g_0, g_1) = 0 \quad \prod(g_0, g_2) = 0 \quad \prod(g_1, g_2) = 0
\]

Check magnitude

\[
\prod(g_0, g_0) = 1 \quad \prod(g_1, g_1) = 1 \quad \prod(g_2, g_2) = 1
\]

something is wrong!
\[ g_2 = \langle x_2 \rangle - \frac{\langle x_1 \rangle \cdot g_0}{\langle g_0, g_0 \rangle} g_0 - \frac{\langle x_1 \rangle \cdot g_1}{\langle g_1, g_1 \rangle} g_1 \]

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

Check orthogonality
Check magnitude $> 0$
\[
\prod \langle g_0, g_1 \rangle = 0 \quad \prod \langle g_0, g_0 \rangle = 1 \\
\prod \langle g_0, g_2 \rangle = 0 \quad \prod \langle g_1, g_1 \rangle = 1 \\
\prod \langle g_1, g_2 \rangle = 0 \quad \prod \langle g_2, g_2 \rangle = 0
\]

Non-Example of projection (come back here after projection is introduced later in this worksheet)

Task: express $y$ as a linear combination of the basis vectors $x \langle 0 \rangle$, $x \langle 1 \rangle$, $x \langle 2 \rangle$.

\[
y := \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\]

A quick common-sense inspection of the above equation indicates $y = x \langle 2 \rangle$.  
\[ a_0 \neq 0 \quad a_1 = 0 \quad a_2 = 1 \]

\[ n := 2 \quad i := 0 \ldots n \quad j := 0 \ldots n \]

\[
XX_{i,j} := \prod \langle x \langle i \rangle, x \langle j \rangle \rangle \\
XX = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \end{pmatrix} \\
Xy := \prod \langle y, x \langle i \rangle \rangle \\
Xy = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
XX \cdot a = Xy
\]

\[ \text{cond2}(XX) = 2.368 \times 10^{34} \quad \text{... very large; theoretically infinite } \infty. \]

An inspection of the matrix $XX$ and vector $Xy$ above yields (in a naive way) the following two cases

Case 1  
\[ a_0 = 0 \quad a_1 = 1 \quad a_2 = 0 \quad \text{but this makes no sense!} \]

or Case 2  
\[ a_0 = 0 \quad a_1 = 0 \quad a_2 = 1 \]

However, mathematically, we have a singular $XXa := XX^{-1} \cdot Xy$ and coefficient $a$ cannot be determined

(Mathcad often fails to raise this red flag.)

(Singularity)

Cause of the above nonsense: invalid definition of scalar product (which violates Rule #4 by having $w_2 = 0$). As shown below, the contradiction disappears when a weight (albeit very small) is included.

\[ w_2 := 10^{-10} \quad \prod (x, y) := x_0 \cdot y_0 + x_1 \cdot y_1 + w_2 \cdot x_2 \cdot y_2 \quad \ldots \text{include a small weight} \]

\[
XX_{i,j} := \prod \langle x \langle i \rangle, x \langle j \rangle \rangle \\
Xy := \prod \langle y, x \langle i \rangle \rangle \\
a := XX^{-1} \cdot Xy
\]

\[ \text{cond2}(XX) = 9.123 \times 10^{10} \quad \text{... very large; but theoretically finite.} \]
Non-Example. vector = a continuous function in x=[0 3]. Given three basis functions \( f_0, f_1, f_2 \):

\[
\begin{align*}
  f_0(x) &= 1 \\
  f_1(x) &= 1 + x \\
  f_2(x) &= 1 + x + x^2
\end{align*}
\]

Define scalar product \( \prod(f,g) := \int_0^1 x \cdot f(x) \cdot g(x) \, dx + \int_2^3 x \cdot f(x) \cdot g(x) \, dx \) i.e., we skip \( x=[1 2] \)

Non-Example of projection (come back here after projection is introduced later in this worksheet)

Task: express \( y(x) \) as a linear combination of the basis vectors \( f_0(x), f_1(x), f_2(x) \).

\[
y(x) = a_0 f_0(x) + a_1 f_1(x) + a_2 f_2(x) = a_0 (1) + a_1 (1 + x) + a_2 (1 + x + x^2)
\]

A quick common-sense inspection of the above equation indicates \( y(x)=f_0(x) \) \( \Rightarrow a_0=1 \), \( a_1=0 \), \( a_2=0 \)

With a casual inspection, the numbers look reasonable.

\[
P = \begin{bmatrix}
  3 & 9.667 & 26.167 \\
  26.167 & 91.733 & 261.633
\end{bmatrix} \quad F = \begin{bmatrix}
  3 \\
  26.167
\end{bmatrix} \quad \text{cond}_2(P) = 2.18 \cdot 10^4
\]

The definition is equivalent to considering interval \( x=[0 3] \) with \( w(x)=0 \) for \( x=[1 2] \) and \( w(x)=x \) elsewhere.

Define scalar product \( w := 0 \) \( \prod(f,g) := \int_0^1 x \cdot f(x) \cdot g(x) \, dx + \int_1^2 w \cdot x \cdot f(x) \cdot g(x) \, dx + \int_2^3 x \cdot f(x) \cdot g(x) \, dx \)

\[
P_{i,j} := \int_0^1 x \cdot f_i(x) \cdot f_j(x) \, dx + \int_1^2 w \cdot x \cdot f_i(x) \cdot f_j(x) \, dx + \int_2^3 x \cdot f_i(x) \cdot f_j(x) \, dx
\]

\[
F_i := \int_0^1 x \cdot y(x) \cdot f_i(x) \, dx + \int_1^2 w \cdot x \cdot y(x) \cdot f_i(x) \, dx + \int_2^3 x \cdot y(x) \cdot f_i(x) \, dx
\]
The above example of \( y(x) = 1 \) seems to be working, but that is misleading. However, strictly speaking, the weight \( w(x) \) needs to be positive (not zero). Otherwise, we violate Rule #4 that \((y,y) = 0 \) if and only if \( y = 0 \).

\[
y(x) := (1 \leq x) \cdot (x \leq 1.5) \cdot (x - 1) + (1.5 < x) \cdot (x \leq 2) \cdot (2 - x) \text{ or } y(x) : = (1 < x) \cdot (x < 2) \cdot (\cos(\pi \cdot (x - 1.5)))
\]

\[
x := 0, 0.01, \ldots, 3
\]

Find coefficient \( a \) based on projection

\[
P_{i,j} := \int_0^1 x \cdot f(x)_i \cdot f(x)_j \, dx + \int_2^3 x \cdot f(x)_i \cdot f(x)_j \, dx \quad F_i := \int_0^1 x \cdot y(x) \cdot f(x)_i \, dx + \int_2^3 x \cdot y(x) \cdot f(x)_i \, dx
\]

With a casual inspection, the numbers look reasonable.

\[
P = \begin{bmatrix} 3 & 9.667 & 26.167 \\ 9.667 & 32.833 & 91.733 \\ 26.167 & 91.733 & 261.633 \end{bmatrix} \quad F = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{cond2}(P) = 2.18 \times 10^4
\]

But the answer based on the projection idea defies common-sense. \( a := P^{-1} \cdot F \quad a = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \)

**Non-Example.** Here is another example where an invalid definition makes a big difference.

\[
y(x) := \exp \left[ \frac{(x - 1.5)^2}{0.1} \right] \quad f(x) := \left[ 1 \quad x \quad x^2 \quad x^3 \quad x^4 \quad x^5 \quad x^6 \quad x^7 \quad x^8 \quad x^9 \quad x^{10} \right]^T
\]

\[
n := 10 \quad i := 0 \ldots n \quad j := 0 \ldots n
\]

\[
P_{i,j} := \int_0^1 x \cdot f(x)_i \cdot f(x)_j \, dx + \int_2^3 x \cdot f(x)_i \cdot f(x)_j \, dx \quad F_i := \int_0^1 x \cdot y(x) \cdot f(x)_i \, dx + \int_2^3 x \cdot y(x) \cdot f(x)_i \, dx \quad a_{\text{invalid}} := P^{-1} \cdot F
\]

\[
a_{\text{invalid}}^T = \begin{bmatrix} 2.084 & \ldots & 318.255 & -1.07 \times 10^3 & 2.006 \times 10^3 & -2.262 \times 10^3 & 1.591 \times 10^3 & -702.199 & 188.876 & -28.293 & 1.809 \end{bmatrix}
\]

\[
P_{i,j} := \int_0^3 x \cdot f(x)_i \cdot f(x)_j \, dx \quad F_i := \int_0^3 x \cdot y(x) \cdot f(x)_i \, dx \quad a_{\text{valid}} := P^{-1} \cdot F
\]

\[
a_{\text{valid}}^T = \begin{bmatrix} 0.215 & \ldots & 36.822 & -101.793 & 138.004 & -98.519 & 37.299 & -6.745 & 0.316 & 0.035 & 0 \end{bmatrix}
\]
Non-example (because \( w(x) = x < 0 \) when \( x \) is negative)

\[
\text{prod}(f, g) := \int_{-1}^{1} x \cdot f(x) \cdot f(x) \, dx
\]

The following orthogonality seems to imply that \( J_0(1x), J_0(2x), \) & \( J_0(3x) \) are mutually orthogonal.

\[
\int_{-1}^{1} x \cdot J_0(1x) \cdot J_0(2x) \, dx = 0 \quad \int_{-1}^{1} x \cdot J_0(1x) \cdot J_0(3x) \, dx = 0 \quad \int_{-1}^{1} x \cdot J_0(2x) \cdot J_0(3x) \, dx = 0
\]

However, magnitude=0, when the functions are not 0!

\[
\int_{-1}^{1} x \cdot J_0(1x) \cdot J_0(1x) \, dx = 0 \quad \int_{-1}^{1} x \cdot J_0(2x) \cdot J_0(2x) \, dx = 0 \quad \int_{-1}^{1} x \cdot J_0(3x) \cdot J_0(3x) \, dx = 0
\]

Thus, the results from vector projection and orthogonality are wrong! The vector projection idea is not wrong, nor is the orthogonality idea wrong, but the definition of a scalar product is not valid. Basically, if the scalar definition is invalid, everything we do that depends on scalar product (such as projection, orthogonality, Gram-Schmidt, magnitude, angle, ...) becomes garbage-in, garbage-out. That is why we have to make sure that we start off correctly.
Gram-Schmidt Process in matrix-vector notation & cholesky decomposition. Given a set of \( n \) linearly independent vectors in real Euclidean space, \( f_1, f_2, \ldots, f_n \), we construct a set of \( n \) vectors \( g_1, g_2, \ldots, g_n \) that are pair-wise orthonormal.

Start with \( f_1 \), then normalize \( g_1 \).

\[
g_1 = a_{11} f_1
\]

where \( a_{11} \) is the inverse of magnitude of \( f_1(x) \) that makes \( g_1(x) \) normal.

Add \( f_2 \) to the bag to construct \( g_2 \).

\[
g_2 = f_2 - \frac{\langle f_2, g_1 \rangle}{\langle g_1, g_1 \rangle} g_1
\]

\[
\rightarrow g_2 = a_{12} f_1 + a_{22} f_2
\]

Add \( f_3 \) to the bag to construct \( g_3 \).

\[
g_3 = f_3 - \frac{\langle f_3, g_1 \rangle}{\langle g_1, g_1 \rangle} g_1 - \frac{\langle f_3, g_2 \rangle}{\langle g_2, g_2 \rangle} g_2
\]

\[
\rightarrow g_3 = a_{13} f_1 + a_{23} f_2 + a_{33} f_3
\]

We repeat the process.

\[
g_n = f_n - \sum_{j=1}^{n-1} \frac{\langle f_n, g_j \rangle}{\langle g_j, g_j \rangle} g_j
\]

\[
\rightarrow g_n = \sum_{i=1}^{n} a_{in} f_i
\]

In compact "matrix-vector" notation

\[
\begin{bmatrix}
\langle g_1, f_1 \rangle & \langle g_1, f_2 \rangle & \langle g_1, f_3 \rangle & \cdots & \langle g_1, f_n \rangle \\
\langle g_2, f_1 \rangle & \langle g_2, f_2 \rangle & \langle g_2, f_3 \rangle & \cdots & \langle g_2, f_n \rangle \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\langle g_n, f_1 \rangle & \langle g_n, f_2 \rangle & \langle g_n, f_3 \rangle & \cdots & \langle g_n, f_n \rangle
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\]

\[
g = f A
\]

where \( A \) is an upper triangular matrix

\( g \) are all mutually orthonormal; thus, the matrix \( G \) that has as elements all scalar products of \( g \) is a symmetrical positive definite matrix. Likewise, the matrix \( F \) with all scalar products of \( f \) is also a symmetrical positive definite matrix.

\[
\begin{bmatrix}
\langle g_1, g_1 \rangle & \langle g_1, g_2 \rangle & \langle g_1, g_3 \rangle & \cdots & \langle g_1, g_n \rangle \\
\langle g_2, g_1 \rangle & \langle g_2, g_2 \rangle & \langle g_2, g_3 \rangle & \cdots & \langle g_2, g_n \rangle \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\langle g_n, g_1 \rangle & \langle g_n, g_2 \rangle & \langle g_n, g_3 \rangle & \cdots & \langle g_n, g_n \rangle
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\]

\[
G = I
\]
Pre-multiply the above expression by \(A\)^{-1} and post-multiply by \(A\)^{-1}. Note that \(U=A\)^{-1} is upper triangular, because \(A\) is upper triangular. Likewise, \(U^T\) is lower triangular.

The above is a special case of LU decomposition called \textit{cholesky decomposition} for a symmetrical positive definite matrix \(F\).

Summary:

Step 0. Given \(n\) linearly independent basis \(f\)

Step 1. Construct matrix \(F\), where \(F_{ij}=(f_i, f_j)\)

Step 2. \(L=\text{cholesky}(F)\) (\(L \cdot L^T=F\) in Mathcad); or \(U=\text{chol}(F)\) (in Matlab)

Step 3. \(A=(L^{-1})^T\) (in Mathcad); or \(A=U^{-1}\) in Matlab

Step 4. Construct \(g=fA\), where \(g\) is a row of vectors \(g_1, g_2, \ldots, g_n\); likewise for \(f\).

Caution: Negative diagonals in \(A\) get converted to positive diagonals upon cholesky decomposition.

\[
A := \begin{pmatrix} 0 & -1 & -5 \\ 0 & 2 & \end{pmatrix}, \quad F := (A^T)^{-1} \cdot A^{-1} \quad L := \text{cholesky}(F) \quad \text{Check:} \quad F = \begin{pmatrix} 2 & 5 & 9.5 \\ 3.5 & 9.5 & 18.75 \end{pmatrix}, \quad L \cdot L^T = \begin{pmatrix} 1 & 2 & 3.5 \\ 2 & 5 & 9.5 \end{pmatrix}
\]

\(\text{not } A!\). However, this is not a problem, because the whole column \(A^T\) is sign-flipped, meaning cholesky decomposition gives a sign-flipped basis vector \(g_1\).
Express \( x \) in terms of other vectors. We can express any vector \( x \) in real Euclidean space as a linear combination of linearly independent basis.

\[
x = \sum_{i=1}^{n} \xi_i \mathbf{g}_i = \xi_1 \mathbf{g}_1 + \xi_2 \mathbf{g}_2 + \cdots + \xi_n \mathbf{g}_n
\]

With the following equation, we find the coefficients \( \xi_i \). The above one (vector) equation contains \( n \) unknown parameters \( \xi_i, i=1..n \). To create \( n \) equations needed to solve for \( n \) unknown parameters, we scalar-product (multiply) the above vector equation with each of the \( n \) basis vectors \( \mathbf{g}_i, i=1..n \).

\[
\langle x, \mathbf{g}_i \rangle = \left( \sum_{j=1}^{n} \xi_j \mathbf{g}_j \right) \cdot \mathbf{g}_i = \sum_{j=1}^{n} \xi_j \langle \mathbf{g}_j, \mathbf{g}_i \rangle = \xi_i \Rightarrow \xi_i = \langle x, \mathbf{g}_i \rangle
\]

There are 4 equal signs in the above equation. We re-examine each step, because the intermediate results are applicable/useful for different cases.

1st = sign ... substitute \( x \), which is expressed as a linear combination of basis; apply scalar product to both side of the equation.

2nd = sign ... apply distributive property of scalar product (i.e., scalar product of sum is sum of scalar product. Also apply associative property of scalar product to take \( \xi_j \) out of scalar product.

\[
\langle \mathbf{g}_i, x \rangle = \sum_{j=1}^{n} \xi_j \langle \mathbf{g}_j, \mathbf{g}_i \rangle \quad \text{... for any linearly independent basis vectors } \mathbf{g}
\]

(non-orthogonal as well as orthogonal)

The above equation expanded for each \( i=1,2..,n \).

\[
\langle \mathbf{g}_i, x \rangle = \sum_{j=1}^{n} \langle \mathbf{g}_i, \mathbf{g}_j \rangle \xi_j \quad \text{for} \quad i=1,2..,n
\]

\[
\begin{align*}
\langle \mathbf{g}_1, x \rangle &= \langle \mathbf{x}, \mathbf{g}_1 \rangle = \langle \mathbf{g}_1, \mathbf{g}_1 \rangle \xi_1 + \langle \mathbf{g}_1, \mathbf{g}_2 \rangle \xi_2 + \cdots + \langle \mathbf{g}_1, \mathbf{g}_n \rangle \xi_n \\
\langle \mathbf{g}_2, x \rangle &= \langle \mathbf{x}, \mathbf{g}_2 \rangle = \langle \mathbf{g}_2, \mathbf{g}_1 \rangle \xi_1 + \langle \mathbf{g}_2, \mathbf{g}_2 \rangle \xi_2 + \cdots + \langle \mathbf{g}_2, \mathbf{g}_n \rangle \xi_n \\
&\vdots \\
\langle \mathbf{g}_n, x \rangle &= \langle \mathbf{x}, \mathbf{g}_n \rangle = \langle \mathbf{g}_n, \mathbf{g}_1 \rangle \xi_1 + \langle \mathbf{g}_n, \mathbf{g}_2 \rangle \xi_2 + \cdots + \langle \mathbf{g}_n, \mathbf{g}_n \rangle \xi_n
\end{align*}
\]

The same equation in a compact notation. (Each element in the following matrix-vector is a scalar).

\[
\begin{bmatrix}
\langle \mathbf{g}_1, x \rangle \\
\langle \mathbf{g}_2, x \rangle \\
\vdots \\
\langle \mathbf{g}_n, x \rangle
\end{bmatrix}
= 
\begin{bmatrix}
\langle \mathbf{g}_1, \mathbf{g}_1 \rangle & \langle \mathbf{g}_1, \mathbf{g}_2 \rangle & \cdots & \langle \mathbf{g}_1, \mathbf{g}_n \rangle \\
\langle \mathbf{g}_2, \mathbf{g}_1 \rangle & \langle \mathbf{g}_2, \mathbf{g}_2 \rangle & \cdots & \langle \mathbf{g}_2, \mathbf{g}_n \rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle \mathbf{g}_n, \mathbf{g}_1 \rangle & \langle \mathbf{g}_n, \mathbf{g}_2 \rangle & \cdots & \langle \mathbf{g}_n, \mathbf{g}_n \rangle
\end{bmatrix}
\begin{bmatrix}
\xi_1 \\
\xi_2 \\
\vdots \\
\xi_n
\end{bmatrix}
\Rightarrow 
\mathbf{x} = \mathbf{G}^{-1} \cdot \xi
\]
Note that the above "normal" equation appears again and again in applied mathematics. It applies to many fields, including function approximation (Fourier series, Bessel series, ...), data compression, modeling, and of course linear regression where we express a dependent vector \( x \) as a linear combination of \( n \) independent vectors \( g_i, i=1,2,...,n \).

**Linear regression problem:**

\[
x = \sum_{i=1}^{n} \xi_i g_i = \xi_1 g_1 + \xi_2 g_2 + \ldots + \xi_n g_n = \begin{bmatrix} g_1 & g_2 & \ldots & g_n \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix} = g \cdot \xi
\]

where

\[
g = \begin{bmatrix} g_1 \\ g_2 \\ \ldots \\ g_n \end{bmatrix}, \quad \xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix}
\]

The above compact notation shows

\[
\begin{bmatrix} g_1 \\ g_2 \\ \ldots \\ g_n \end{bmatrix} \cdot x = x, \quad \begin{bmatrix} g_1 \\ g_2 \\ \ldots \\ g_n \end{bmatrix} = x, \quad g \cdot \xi = \begin{bmatrix} g_1 & g_2 & \ldots & g_n \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix}
\]

If we supply a scalar definition: \((f, g) = f^T \cdot g\) then the two scalar products in the above equation become

\[
(f^T \cdot x) = (f^T \cdot g) \cdot \xi \quad \rightarrow \quad \xi = (f^T \cdot g)^{-1} \cdot (f^T \cdot x)
\]

If we supply a scalar definition: \((f, g) = f^T \cdot W \cdot g\) then the two scalar products in the above equation become

\[
(g^T \cdot W \cdot x) = (g^T \cdot W \cdot g) \cdot \xi \quad \rightarrow \quad \xi = (g^T \cdot W \cdot g)^{-1} \cdot (g^T \cdot W \cdot x)
\]

The above solution is the formula in function interpolation at finite number of nodes. It is also the normal equation in linear regression. Note that we arrived at it (not even worthy of the word "derived it") simply by applying a scalar product. Note the simplicity. And the equation is independent on the type of vectors or how the scalar product is defined. The numerical value of \( \xi \) though does depend on the definition of the scalar product. Note that we did not explicitly minimize the sum of squares (we did not do least-squares). This is equivalent to the "hand-waving derivation" that does not go through minimization of sum of squared errors.

start with: \( x = g \cdot \xi \) where \( g \) is a collection of basis vectors \( g = \begin{bmatrix} g_1 & g_2 & \ldots & g_n \end{bmatrix} \)

We want to modify "g" into something that we can invert. This is done by taking a scalar product (or, equivalently, multiplying by \( g^T \) in the linear regression matrix-vector notation)

\[
(f^T \cdot x) = (f^T \cdot g) \cdot \xi
\]

With a flip of the matrix inverse, we are done! \( \xi = (f^T \cdot g)^{-1} \cdot (f^T \cdot x) \)
For orthogonal basis vectors $g_i$, $(g_i, g_j) = 0$ for $i \neq j$; $(g_i, g_i) \neq 0$. Thus, out of the summation of $n$ terms, only one scalar product term remains. This simplifies the evaluation of $\xi$, because each $\xi_i$ is decoupled from other $\xi_i$s and can be evaluated independently.

$$\left( x, g_i \right) = \xi_i \left( g_i, g_i \right) \quad \rightarrow \quad \xi_i = \frac{\left( x, g_i \right)}{\left( g_i, g_i \right)} \quad \text{... for orthogonal basis vectors } g$$

In the compact matrix-vector notation (actually not so compact anymore because the above equation is even more compact), orthogonal basis vectors make the off-diagonal elements in the $G$ matrix to be all 0 and simplifies the inverse of $G$.

$$\begin{bmatrix}
(x, g_1) \\
(x, g_2) \\
\vdots \\
(x, g_n)
\end{bmatrix} =
\begin{bmatrix}
(g_1, g_1) & 0 & \cdots & 0 \\
0 & (g_2, g_2) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & (g_n, g_n)
\end{bmatrix}
\begin{bmatrix}
\xi_1 \\
\xi_2 \\
\vdots \\
\xi_n
\end{bmatrix}
\quad \rightarrow \quad x = G\xi \quad \rightarrow \quad \xi = G^{-1}x$$

The above bold-faced words "evaluated independently" mean each $\xi_i$ can be evaluated one at a time in a sequential manner (no matrix inverse, only scalar inverse). It also means that it matters not which $\xi_i$ is evaluated first, which is evaluated next; we can randomly pick out a basis $g_i$ and evaluate the coefficient $\xi_i$ corresponding to that basis $g_i$. It also means that we can work successively on the residual quantity rather than the original $x$. It also means that, if we so choose, we can start off the problem by extracting out first the most important $g_i$ (most relevant to the problem on hand, the largest in magnitude as measured by $\xi_i$, ...).

Start with an initial residual $x_{\text{residual.0}} = x$

$$x_{\text{residual.0}} = x = \xi_1 g_1 + \xi_2 g_2 + \cdots + \xi_n g_n$$

We consider one of the basis vectors, say $g_1$. Any random one basis vector! After we take out this term, the residual becomes,

$$x_{\text{residual.1}} = x_{\text{residual.0}} - \xi_1 g_1 = x - \xi_1 g_1 = \xi_2 g_2 + \cdots + \xi_n g_n$$

We consider one of the remaining basis vectors, say $g_2$. Any random one remaining basis vector! After we take out this term, the residual becomes,

$$x_{\text{residual.2}} = x_{\text{residual.1}} - \xi_2 g_2 = x - \xi_1 g_1 - \xi_2 g_2 = \xi_3 g_3 + \cdots + \xi_n g_n$$

Repeat until we have extracted all (or enough number of) vectors.
4th = sign ... For basis vectors of unit length, \((g_i, g_i) = 1\).

\[
\langle x, g_i \rangle \equiv \xi_i, \quad \langle g_i, g_i \rangle \equiv \xi_i^2 = 1 \quad \longrightarrow \quad \xi_i = \langle x, g_i \rangle
\]

... for orthonormal basis vectors \(g\)

In the compact matrix-vector notation, orthonormal basis vectors make the diagonal elements in the \(G\) matrix to be all 1, and the \(G\) matrix is now an identity matrix, \(G = I\).

\[
\begin{bmatrix}
\langle x, g_1 \rangle \\
\langle x, g_2 \rangle \\
\vdots \\
\langle x, g_n \rangle \\
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 \\
\end{bmatrix}
\begin{bmatrix}
\xi_1 \\
\xi_2 \\
\vdots \\
\xi_n \\
\end{bmatrix}

\rightarrow 
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n \\
\end{bmatrix}
= G \xi

\rightarrow 
\xi = G^{-1} x \xi = G^\top x \xi

\rightarrow 
\xi_i = \langle x, g_i \rangle

Thus, for an orthonormal set of basis vectors, the coefficient corresponding to the \(i\)th basis vector is simply the scalar product of the given vector \(x\) with that \(i\)th basis vector. With these coefficients, the given vector \(x\) is expressed as,

\[
x = \sum_{i=1}^{n} \xi_i g_i
\]

\[
= \sum_{i=1}^{n} \langle x, g_i \rangle \cdot g_i = \langle x, g_1 \rangle \cdot g_1 + \langle x, g_2 \rangle \cdot g_2 + \ldots + \langle x, g_n \rangle \cdot g_n
\]
**Projection.** A vector $x$ in real Euclidean space (not necessarily of finite dimension) is represented as a linear combination of $m$ orthonormal basis vectors $g_1, g_2, \ldots, g_m$ in a real Euclidean subspace.

$$x = \sum_{i=1}^{n} \xi_i g_i = \sum_{i=1}^{m} \xi_i g_i + \sum_{i=m+1}^{n} \xi_i g_i = x_{\text{proj}} + \text{residual or error to be ignored}$$

$\text{Black plane} = n$-dimensional vector space

$x=$solid red arrow

$x_{\text{proj}}=$dotted red arrow

residual error=dotted blue arrow

Note: dotted blue arrow $\perp$ dotted red arrow

For a general set of $m$ basis vectors $g_1, g_2, \ldots, g_m$ that are not necessarily orthogonal, the projection formula is (with coupled coefficients $\xi$),

$$\langle x, g_i \rangle = \sum_{j=1}^{m} \xi_j \langle g_j, g_i \rangle \quad \Rightarrow \quad \text{Solve for } \xi \quad \forall i$$

where $G$ is a matrix of scalar products. $G_{ij} = \langle g_i, g_j \rangle$

$xg$ is a column of scalar products. $xg_i = \langle x, g_i \rangle$

If the basis vectors $g_1, g_2, \ldots, g_m$ are merely orthogonal, but not orthonormal, the projection formula is,

$$x_{\text{proj}} = \sum_{i=1}^{m} \xi_i g_i = \sum_{i=1}^{m} \left( \frac{\langle x, g_i \rangle}{\langle g_i, g_i \rangle} \right) g_i = \xi^T G^T x \quad \Rightarrow \quad \xi = \left( G^T x \right)^{-1}$$

$G=\text{diagonal}$

For orthonormal basis vectors, projection of $x$ into $m$-dimensional subspace is,

$$x_{\text{proj}} = \sum_{i=1}^{m} \xi_i g_i = \sum_{i=1}^{m} \left( \frac{\langle x, g_i \rangle}{\langle g_i, g_i \rangle} \right) g_i = \xi^T G^T x \quad \Rightarrow \quad \xi = \left( G^T x \right)$$

$G=I$
Error $e = x - x_{\text{proj}}$ is \perp to any vector $y$ in the m-dimensional subspace spanned by $g_1, g_2, ..., g_m$.

Depending on the applications, error $e$ is referred to as the error vector or the residual vector.

**Proof.**

$$x = \sum_{i=1}^{m} \xi_i g_i + \sum_{i=m+1}^{n} \xi_i g_i = x_{\text{proj}} + e$$

$$x_{\text{proj}} = \sum_{i=1}^{m} \xi_i g_i$$

$$y = \sum_{i=1}^{m} \eta_i g_i$$

$$x_{\text{proj}} = \sum_{i=1}^{m} \frac{(x, g_i)}{(g_i, g_i)} g_i$$

$$(e, y) = (x_{\text{proj}}, y) = (x, y) - (x_{\text{proj}}, y) = \left[ x, \sum_{i=1}^{m} \eta_i g_i \right] - \left[ x_{\text{proj}}, \sum_{i=1}^{m} \eta_i g_i \right] = 0 : e \perp y \text{ for every } y \text{ in the subspace.}$$

$y = x_{\text{proj}}$ minimizes $|e| = |x - y|$ among all vectors $y$ in the subspace.

**Proof.**

$$(|x - y|^2) = (|x_{\text{proj}} + x_{\text{proj}} - y|^2)$$

$$= (x_{\text{proj}} + x_{\text{proj}} - y, x_{\text{proj}} + x_{\text{proj}} - y)$$

$$= (x_{\text{proj}} + x_{\text{proj}} - y, x_{\text{proj}} + x_{\text{proj}} - y) + 2(x_{\text{proj}} + x_{\text{proj}} - y, x_{\text{proj}} - y) + (x_{\text{proj}} - y, x_{\text{proj}} - y)$$

$$= (x_{\text{proj}} - y, x_{\text{proj}} - y)$$

because $(x_{\text{proj}} - y, x_{\text{proj}} - y) = 0$

$$= (|x_{\text{proj}}|^2 + |x_{\text{proj}} - y|^2)$$

$$: y = x_{\text{proj}}, \text{ where } x_{\text{proj}} \text{ minimizes } |e|$$

In other words,projection yields the best approximation of $x$ in a reduced m-dimensional subspace. This idea is useful in many fields of applied mathematics: linear regression, function approximation, solution of ODE, orthogonal collocation, methods of weighted residuals, calculus of variation, noise filtering, digital signal processing, Laplace transform, Fourier transform, etc.

**Example. Linear regression**

General linear formula:

$$y = Xa + e$$

$$y = \begin{bmatrix} y^{<1>} \\ X^{<2>} \\ \vdots \\ X^{<m>} \end{bmatrix} + e$$

Error $e = y - x_{\text{proj}} = y - Xa$ is \perp to any vector $y$ in the m-dimensional subspace spanned by $X^{<1>}, X^{<2>}, ..., X^{<m>}$

$$0 = (e, X) = (y - Xa, X)^T X = y^T X - a^T X^T X$$

$X^T X a = X^T y$  \quad $a = (X^T X)^{-1} X^T y$

Note: no calculus $d(\text{sse})/da = 0$