ECE551, Fall 2016
Homework Problem Set #2
Rev. 0, Due Sep. 20th 2016 in class

1. **Frames and Bases**
Given are the following sets of vectors in $\mathbb{R}^2$:

$$\Phi_1 = \{\varphi_{1,0},\ \varphi_{1,1}\} = \left\{ \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \tag{1}$$

$$\Phi_2 = \{\varphi_{2,0},\ \varphi_{2,1},\ \varphi_{2,3},\ \varphi_{2,4}\} = \left\{ \begin{bmatrix} \frac{1}{2\sqrt{2}} \\ \frac{\sqrt{3}}{2\sqrt{2}} \end{bmatrix}, \begin{bmatrix} -\frac{\sqrt{3}}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \right\} \tag{2}$$

$$\Phi_3 = \{\varphi_{3,0},\ \varphi_{3,1}\} = \left\{ \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}, \begin{bmatrix} \frac{-\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} \right\} \tag{3}$$

$$\Phi_4 = \{\varphi_{4,0},\ \varphi_{4,1},\ \varphi_{4,2}\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \tag{4}$$

For each of the vector sets $\{\Phi_k\}$ above, do the following:

(a) Write the matrix representation (synthesis operator) for the set.
(b) Find the dual basis or canonical dual frame $\tilde{\Phi}$. **Sketch** the original sets and their duals on $\mathbb{R}^2$.
(c) If the set is a basis, specify if it is orthonormal; otherwise, it is a frame - specify if it is tight.
(d) For $x = \begin{bmatrix} 2 & 0 \end{bmatrix}^T$, write down the projection coefficients, $\alpha_{i,k} = \langle x, \tilde{\varphi}_{i,k} \rangle$.
(e) For the x above, verify the expansion $x = \sum_k \alpha_{i,k} \varphi_{i,k}$.
(f) Verify that $\Phi \tilde{\Phi}^T = I$.
(g) Specify whether the expansion preserves the norm, that is, if $\|x\|^2 = \sum_k |\alpha_{i,k}|^2$.
(h) Specify whether the expansion is redundant. Justify your answer.

2. **Linear Least-Squares approximation**
Consider the general least squares (with the standard norm) solution of a linear problem:

$$\hat{x} = \arg\min \|y - Ax\|^2$$

whose formula was given in (2.225) in the textbook, and let $\hat{y} := A\hat{x}$,

(a) Show that if $y \in \mathrm{colsp}(A)$, then $\hat{y} = y$.
(b) Show that if $y \perp \mathrm{colsp}(A)$, then $\hat{y} = 0$.
(c) Show that for the least-squares solution, the partial derivatives vanish:

$$\frac{\partial \|y - A\hat{x}\|^2}{\partial \hat{x}_i} = 0 \quad \text{for all } i$$

Hint: if $B(t)$ and $C(t)$ are two compatible matrices depending on a parameter t, then $\frac{d}{dt}\big(B(t)C(t)\big) = \frac{dB(t)}{dt}C(t) + B(t)\frac{dC(t)}{dt}$. Apply this formula with $C(t) = B(t)^T$.

3. **Orthogonalization of a projection**
Consider a linear mapping on $\mathbb{C}^N$ defined by a matrix $P \in \mathbb{C}^{N\times N}$. The mapping is said to be an *Oblique Projection* if $P^2 = P$, and an *Orthogonal Projection* if P is additionally self-adjoint matrix, namely $P = P^*$. The orthogonality in this definition applies to matrices and the standard inner product $\langle u, v\rangle_0 := v^*u$ on $\mathbb{C}^N$. Our goal is to extend the notion of orthogonal projections to other inner products on $\mathbb{C}^N$.

Definition 1 (General Projection) *Let $P : V \to V$ be a linear mapping on V, equipped with an inner product $\langle \cdot, \cdot\rangle$. We say that P is* Self-adjoint with respect to $\langle \cdot, \cdot\rangle$ *if $\langle Px, y\rangle = \langle x, Py\rangle$ for all pairs $x, y \in V$. Also, P is an* Orthogonal Projection with respect to $\langle \cdot, \cdot\rangle$ *if $P^2 = P$ and self-adjoint with $\langle \cdot, \cdot\rangle$.*

(a) Show that if $P = P^*$ then P is self adjoint with respect to the standard inner product on $\mathbb{C}^N$.
(b) Find the eigenvalues of an oblique projection P.
(c) For simplicity assume that $V = \mathbb{C}^N$. Let P be oblique, with the diagonal form $P = T^{-1}DT$, where D is a diagonal matrix with the eigenvalues found in part (b). Show that P is self-adjoint with respect to the inner product defined below

$$\langle x, y\rangle_T := y^*T^*Tx$$

Bonus: prove that P is diagonalizable.

(d) Show that $x - Px \perp Px$ with respect to the inner product $\langle \cdot, \cdot\rangle_T$.
(e) Show that $I - P$ is an oblique projection, and orthogonal with respect to $\langle \cdot, \cdot\rangle_T$. How is the range space of $I - P$ related to the null-space of P ?
(f) Show that P minimizes the norm $\|x - Px\|_T$ where $\|z\|_T^2 := \langle z, z\rangle_T$.

4. **Approximation by Orthogonal Indicator Tiles**
Let A and I be some finite sets such that $|A| < |I|$, and let $\{E_a\}_{a\in A}$ be a collection of subsets of I (that is, $E_a \subset I$ for all $a \in A$). Define the tile indicator vector $\phi_a \in \mathbb{R}^I$ as

$$\phi_a[i] = \mathbb{1}_{E_a}[i] := \begin{cases} 1 & i \in E_a \\ 0 & \text{else} \end{cases},$$

taking the value 1 on the set E_a and zero otherwise. We call E_a a **tile** or a **patch**.

(a) For this part, assume that I is a 6×6 grid, and $A = \{a, b, c\}$. Sketch a sample of tiles on I.
(b) Are $\{\phi_a\}_{a\in A}$ a basis or a frame of $\mathbb{R}^I$? explain.
(c) Find a necessary and sufficient condition for $\{\phi_a\}_{a\in A}$ to be an orthogonal set on $\mathbb{R}^I$ with the standard inner product $\langle u, v\rangle := \sum_{i\in I} u[i]v[i]$. Prove your statement, and illustrate using the sketch of the first part (**Hint:** recall that $\mathbb{1}_A[i] \cdot \mathbb{1}_B[i] = \mathbb{1}_{A\cap B}[i]$ for any two sets A, B)
(d) Assume that $\{\phi_a\}_{a\in A}$ are orthogonal. For $x \in \mathbb{R}^I$, find the best approximation (with respect to standard norm) for x within $\mathrm{span}\{\phi_a\}_{a\in A}$.
(e) How the answer of the last part would change if $\{\phi_a\}_{a\in A}$ were not orthogonal ? (**Hint:** you may use the canonical dual for $\{\phi_a\}_{a\in A}$).

5. **Python Problem**
Note: for this problem, you'll need some color image of resolution 1024×768 (pick a picture of your favorite thing). Use the `matplotlib.pyplot` package to plot your output. **Note:** if you work with Spyder and want your plots to appear in independent windows (rather than in-line), open a Python console ("Consoles→Open a Python console") and run your program there. You can then save your figures.

(a) Write a Python function that applies the Gram-Schmidt orthogonalization on the rows of a given matrix. That is

```python
from numpy import zeros
from numpy.linalg import norm
# and later ...
def gram_schmidt(V):
    Vo = zeros(V.shape) # Create a matrix of similar dimension
    Vo[0] = V[0]/norm(V[0]) # First vector is same vector
# [ Fill in your Gram-Schmidt algorithm on the rows of V here ]
    return Vo
```

(b) Define set of vectors $p_0, p_1, \ldots, p_{N-1} \in \mathbb{R}^N$ by

$$p_k := \begin{bmatrix} 1^k & 2^k & \ldots & n^k \end{bmatrix}^T, \quad k = 0, \ldots, N-1.$$

Let S_d be the span of the first d of those vectors, here $1 \le d \le N$:

$$S_d := \operatorname{span}\{p_0, p_1, \ldots, p_{d-1}\}.$$

With the orthonormal bases computed above, given an input signal $x \in \mathbb{C}^N$, compute the successive orthogonal projections of x onto the subspaces $S_1, S_2, \ldots, S_5$ and S_N. Plot the orthogonal projections and the error signals. Pick x to be a scan line of your favorite image; you can use the code below.

```python
import numpy as np
import scipy
# and later ...
img_rgb = scipy.misc.imread('your_UIN.jpeg')
img = np.mean(img_rgb,axis=2) # Average RGB colors to a single channel
x = img[17] # Choose some random line number as your wish
```

(c) In this part we implement Problem 4, using the Gram-Schmidt process on non-orthogonal tiles. Let $I = \{0, \ldots, H-1\} \times \{0, \ldots, W-1\}$ and consider the space $\mathbb{R}^I$ embedding digital images of size $H \times W$. Let $R > 0$ be some radius, and define the circular tile centered at (y_0, x_0)

$$E_{y_0,x_0} := \left\{ (y, x) \in I \;\middle|\; (x - x_0)^2 + (y - y_0)^2 \le R^2 \right\}$$

Assume that $A = \{(y_k, x_k) \mid k = 0, \ldots, N-1\}$ is a list of points scattered on I. Write a Python program that projects an image (a $H \times W$ NumPy array) to the space spanned by circular patches defined by A. As output, print the original photo, the projected photo, and 2-3 of the orthogonal basis elements.

For your convenience, the code below does most of the job.

```
def gen_circ_tiles(shape,R):
    V = list()
    H,W = shape
    rs, cs = np.mgrid[0:H, 0:W]

    num_circ = round(W*H/(R**2))
    A = np.array([rand(num_circ)*H, rand(num_circ)*W]).T

    for (r,c) in A:
        mask = (cs-c)**2 + (rs-r)**2 <=R**2
        V.append(1*mask) # The 1* will convert boolean to integer

    return np.array(V)  # Convert list of 2D arrays to a 3D array

def project_on_tiles(img, tiles):
    pimg = np.zeros(img.shape)

    for q in tiles: # That iterates over the fist dimension
        pimg = pimg + np.dot(q.ravel(), img.ravel())*q

    return np.clip(pimg,0,255) # Hard threshold to 8uint values

# Generate circular tiles. Smaller R -> better (and slower) approximation
T = gen_circ_tiles(img.shape,10)
T = np.reshape(T,(T.shape[0], W*H)) # Flatten tiles from I to 0 .. W*H-1
To = gram_schmidt(T) # This is where your code should run
To = np.reshape(To,(To.shape[0], H, W)) # Re-rectangle tiles back to I
pimg = project_on_tiles(img,To) # Project image

# Example of plotting images
plt.subplot(1,2,1)
plt.imshow(img, interpolation="nearest", cmap="gray")
plt.subplot(1,2,2)
plt.imshow(pimg, interpolation="nearest", cmap="gray")
plt.show()
```

Bonus: modify the code to work on color images.