Problem 2

Part a) A square $N \times N$ matrix $\boldsymbol{G}$ is invertible if for every $\boldsymbol{y} \in \mathbb{R}^N$ there is *exactly one* $\boldsymbol{x} \in \mathbb{R}^N$ such that $\boldsymbol{G}\boldsymbol{x} = \boldsymbol{y}$. Show that if $\boldsymbol{G}$ is invertible if and only if its columns are linearly independent and $\boldsymbol{G}\boldsymbol{x} \neq 0$ for all $\boldsymbol{x} \neq 0$

Let $\boldsymbol{G}$ be invertible. If the columns of $\boldsymbol{G}$ are not linearly independent, then $\exists \boldsymbol{v}$ such that $\boldsymbol{G}\boldsymbol{v} = 0$, $\boldsymbol{v} \neq 0$. (This is because a linear combination of columns of $\boldsymbol{G}$, $\sum_i g_i \boldsymbol{v}_i$ can be expressed as $\boldsymbol{G}\boldsymbol{v}$). Since $\boldsymbol{G}\boldsymbol{x} = 0$ is also true for $\boldsymbol{x} = \boldsymbol{0}$, there are two solutions to the equation $\boldsymbol{G}\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{v}$ and $\boldsymbol{0}$. This contradicts the assumption that $\boldsymbol{G}$ is invertible. Hence, columns of $\boldsymbol{G}$ are linearly independent and $\boldsymbol{G}\boldsymbol{x} \neq \boldsymbol{0}$ for all $\boldsymbol{x} \neq \boldsymbol{0}$.

Part b) Let $\psi_1(t), \ldots, \psi_N(t)$ be signals in $L_2([0,1])$. Show that if the the $N \times N$ Gramian

$$\boldsymbol{G} = \begin{bmatrix} \langle \boldsymbol{\psi}_1, \boldsymbol{\psi}_1 \rangle & \langle \boldsymbol{\psi}_2, \boldsymbol{\psi}_1 \rangle & \cdots & \langle \boldsymbol{\psi}_N, \boldsymbol{\psi}_1 \rangle \\ \langle \boldsymbol{\psi}_1, \boldsymbol{\psi}_2 \rangle & \langle \boldsymbol{\psi}_2, \boldsymbol{\psi}_2 \rangle & & \langle \boldsymbol{\psi}_N, \boldsymbol{\psi}_2 \rangle \\ \vdots & & \ddots & \vdots \\ \langle \boldsymbol{\psi}_1, \boldsymbol{\psi}_N \rangle & \cdots & & \langle \boldsymbol{\psi}_N, \boldsymbol{\psi}_N \rangle \end{bmatrix}$$

is invertible if and only if the $\{\boldsymbol{\psi}_n\}$ are linearly independent. (It is helpful to realize that since $\boldsymbol{G}$ is square, it is invertible if and only if $\boldsymbol{G}\boldsymbol{x} \neq \boldsymbol{0}$ for all $\boldsymbol{x} \neq \boldsymbol{0}$.)

We first show that **$\boldsymbol{G}$ is invertible $\Rightarrow$ the $\{\boldsymbol{\psi}_n\}$ are linearly independent**. We do this using the contrapositive: we will show that if the $\{\boldsymbol{\psi}_n\}$ are *not* linearly independent, then $\boldsymbol{G}$ is *not* invertible. Suppose that there exists $\alpha_1, \ldots, \alpha_N$, such that

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} \neq \boldsymbol{0}, \quad \text{and} \quad \sum_{n=1}^{N} \alpha_n \psi_n(t) = 0 \quad \text{for all } t \in [0,1].$$

Then

$$\boldsymbol{G}\boldsymbol{\alpha} = \begin{bmatrix} \sum_{k=1}^N \langle \boldsymbol{\psi}_k, \boldsymbol{\psi}_1 \rangle \alpha_k \\ \sum_{k=1}^N \langle \boldsymbol{\psi}_k, \boldsymbol{\psi}_2 \rangle \alpha_k \\ \vdots \\ \sum_{k=1}^N \langle \boldsymbol{\psi}_k, \boldsymbol{\psi}_N \rangle \alpha_k \end{bmatrix} = \begin{bmatrix} \langle \sum_{k=1}^N \alpha_k \boldsymbol{\psi}_k, \boldsymbol{\psi}_1 \rangle \\ \langle \sum_{k=1}^N \alpha_k \boldsymbol{\psi}_k, \boldsymbol{\psi}_2 \rangle \\ \vdots \\ \langle \sum_{k=1}^N \alpha_k \boldsymbol{\psi}_k, \boldsymbol{\psi}_N \rangle \end{bmatrix} = \begin{bmatrix} \langle \boldsymbol{0}, \boldsymbol{\psi}_1 \rangle \\ \langle \boldsymbol{0}, \boldsymbol{\psi}_2 \rangle \\ \vdots \\ \langle \boldsymbol{0}, \boldsymbol{\psi}_N \rangle \end{bmatrix} = \boldsymbol{0},$$

and so $\boldsymbol{G}$ is not invertible.

Now we show that **the $\{\boldsymbol{\psi}_n\}$ are linearly independent $\Rightarrow$ $\boldsymbol{G}$ is invertible**. We again do this with the contrapositive, showing that if $\boldsymbol{G}$ is not invertible, then the $\{\boldsymbol{\psi}_n\}$ cannot be linearly independent. Suppose there is an $\boldsymbol{x} \neq \boldsymbol{0}$ but $\boldsymbol{G}\boldsymbol{x} = \boldsymbol{0}$. Consider the signal

$$z(t) = \sum_{k=1}^{N} x_k \psi_k(t),$$

where the coefficients x_k are the entries of $\boldsymbol{x}$. Since $\boldsymbol{G}\boldsymbol{x} = \boldsymbol{0}$,

$$\langle \boldsymbol{z}, \boldsymbol{\psi}_n \rangle = \langle \sum_{k=1}^{N} x_k \boldsymbol{\psi}_k, \boldsymbol{\psi}_n \rangle = \sum_{k=1}^{N} \langle \boldsymbol{\psi}_k, \boldsymbol{\psi}_n \rangle x_k = (\boldsymbol{G}\boldsymbol{x})_n = 0 \quad \text{for all } n = 1, \ldots, N.$$

Thus $\boldsymbol{z}$ is orthogonal to all of the $\boldsymbol{\psi}_n$. But then

$$\langle \boldsymbol{z}, \boldsymbol{z} \rangle = \langle \sum_{n=1}^{N} x_n \boldsymbol{\psi}_n, \boldsymbol{z} \rangle = \sum_{n=1}^{N} x_n \langle \boldsymbol{\psi}_n, \boldsymbol{z} \rangle = 0,$$

and so, by the definition of inner product, it must be the case that $\boldsymbol{z} = \boldsymbol{0}$, i.e.

$$z(t) = 0 \quad \text{for all} \quad t \in [0, 1].$$

Thus the $\{\psi_n\}$ are not linearly independent.

Problem 3

In this problem, we will develop the computational framework for approximating a continuous-time signal on $[0, 1]$ uing scaled and shifted versions of the classic bell-curve bump:

$$\phi(t) = e^{-t^2}.$$

Fix an integer $N > 0$ and define $\phi_k(t)$ as

$$\phi_k(t) = \phi(\frac{t - (k - 1/2)/N}{1/N}) = \phi(Nt - k + 1/2)$$

for $k = 1, 2, \cdots, N$. The $\phi_k(t)$ are a basis for the subspace

$$T_N = \text{Span}\,\{\phi_k(t)\}_{k=1}^{N}.$$

a. For a fixed value of N, we can plot all the $\phi_k(t)$ on the same set of axes in MATLAB using:

```
phi = @(z) exp(-z.^2);
t = linspace(0,1, 1000);
figure(1); clf
hold on
for kk = 1:N
plot(t, phi(N*t - kk + 1/2));
end
```

   Do this for $N = 10$ and $N = 25$ and turn in your plots.

b. Since $\phi_k(t)$ is a basis for T_N, we can write any $y(t) \in T_N$ as

$$y(t) = \sum_{k=1}^{N} a_k \phi_k(t)$$

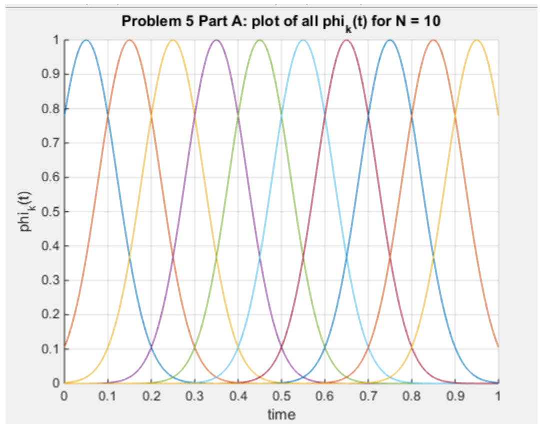

Figure 1: Problem 3, Part A: $\phi_k(t)$ for $N = 10$

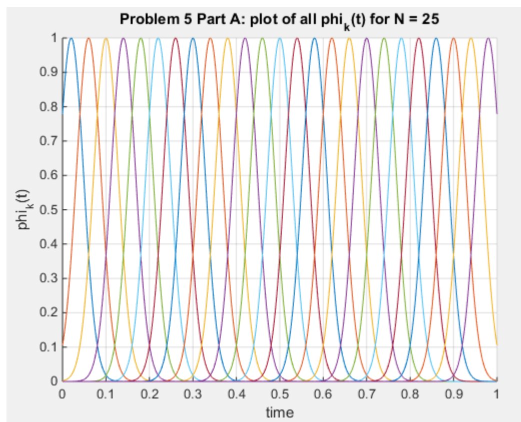

Figure 2: Problem 3, Part A: $\phi_k(t)$ for $N = 25$

for some set of coefficients $a_1, \ldots, a_N \in \mathbb{R}^N$. If these coefficients are stacked in an N-vector $\boldsymbol{a}$ in MATLAB, we can plot $y(t)$ using

```
t = linspace(0,1,1000);
y = zeros(size(t));
for jj = 1:N
y = y + a(jj)*phi(N*t - jj + 1/2);
end
plot(t, y);
```

Do this for $N = 4$, and $a_1 = 1$, $a_2 = -1$, $a_3 = 1$, $a_4 = -1$ and turn in your plot.

c. Define the continuous-time signal $x(t)$ on $[0, 1]$ as

$$x(t) = \begin{cases} 4t & 0 \le t < 1/4 \\ -4t + 2 & 1/4 \le t < 1/2 \\ -\sin(20\pi t) & 1/2 \le t \le 1 \end{cases}.$$

Write MATLAB code that finds the closest point $\hat{x}(t)$ in T_N to $x(t)$ for any fixed N. By

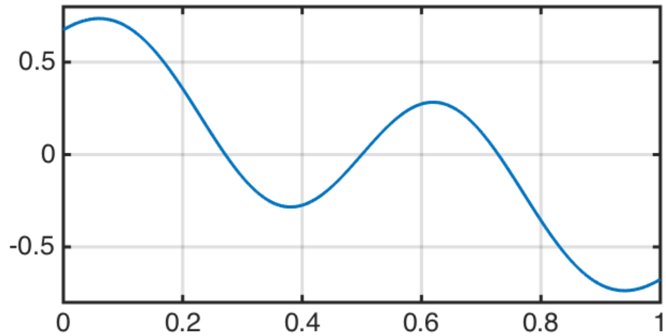

Figure 3: Problem 3, Part B: $y(t)$ for $N = 4$, $\boldsymbol{a} = [1, -1, 1, 1]^T$

"closest point", we mean that $\hat{x}(t)$ is the solution to

$$\min_{y \in T_N} \|x(t) - y(t)\|_{L_2([0,1])}.$$

Turn in your code and four plots; one of which has $x(t)$ and $\hat{x}(t)$ plotted on the same set of axes for $N = 5$, and then repeat for $N = 10, 20,$ and 50.
Hint: You can create a function pointer for $x(t)$ in MATLAB using

```
x = @(z) (z < 1/4).*(4*z) + (z>=1/4).*(z<1/2).*(-4*z+2) - (z>=1/2).*sin(20*pi*z);
```

OR in python using

```
x = lambda z: (z < .25)*(4*z) + (z >= 0.25)*(z < 0.5)*(-4*z+2) - \
 (z>= 0.5)*np.sin(20*np.pi*z)
```

and then calculate the continuous-time inner product $\langle x, \phi_k \rangle$ in MATLAB with

```
x_phik = @(z) x(z).*phi(N*z - jj + 1/2);
integral(x_phik, 0, 1)
```

OR in Python with

```
import scipy.integrate as integrate
x_phik = lambda z: x(z)*phi(N*z - jj + 0.5)
integrate.quad(x_phik, 0, 1)
```

You can use similar code to calculate the entries of the Gram matrix $\langle \phi_j, \phi_k \rangle$. (There is actually a not-that-hard way to calculate the $\langle \phi_j, \phi_k \rangle$ analytically that you can derive if you are feeling industrious — just think about what happens when you convolve a bump with itself.)

```
phi = @(z) exp(-z.^2);
t = linspace(0, 1, 1000);

x = @(z) (z < 1/4).*(4*z) + (z>=1/4).*(z<1/2).*(-4*z+2) ...
```

```
- (z>=1/2).*sin(20*pi*z);


N = 50; % N = 25;
G = zeros(N);
b = zeros(N,1);
for jj = 1:N
for kk = 1:N
G(jj,kk) = integral(@(z) phi(N*z - jj + 1/2).*phi(N*z - kk + 1/2),0,1);
end
b(jj) = integral(@(z) phi(N*z - jj + 1/2).*x(z), 0, 1);
end
a = G\b;

xhat = zeros(size(t));
for jj = 1:N
xhat = xhat + a(jj)*phi(N*t - jj + 1/2);
end
figure
hold on
plot(t, xhat);
plot(t, x(t), 'r')
title(sprintf('Problem 3 Part C: plot of x hat(t) for N = %d',N));
legend('x hat(t)','x(t)');
```

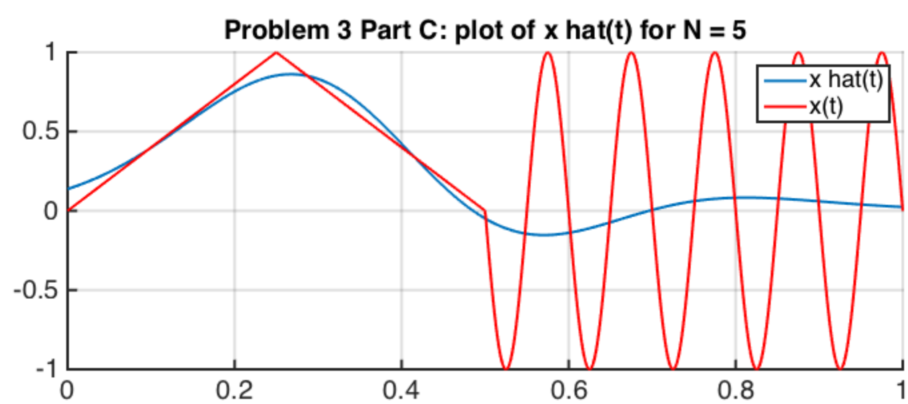

Figure 4: Problem 3, Part C: $\hat{x}(t)$ and $x(t)$ for N = 5

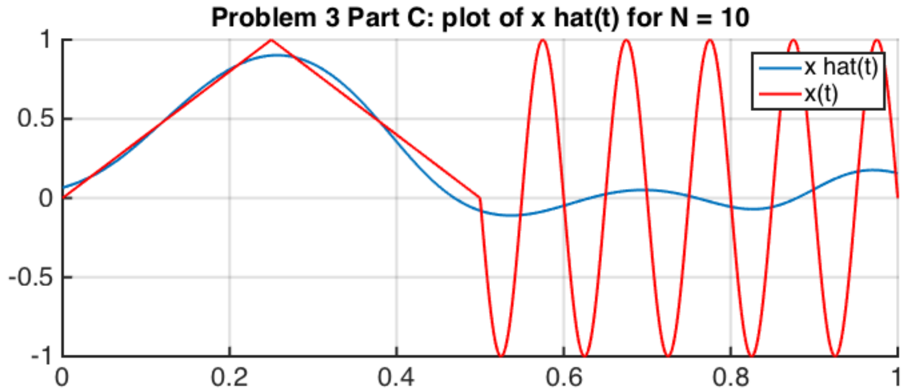

Figure 5: Problem 3, Part C: $\hat{x}(t)$ and $x(t)$ for N = 10

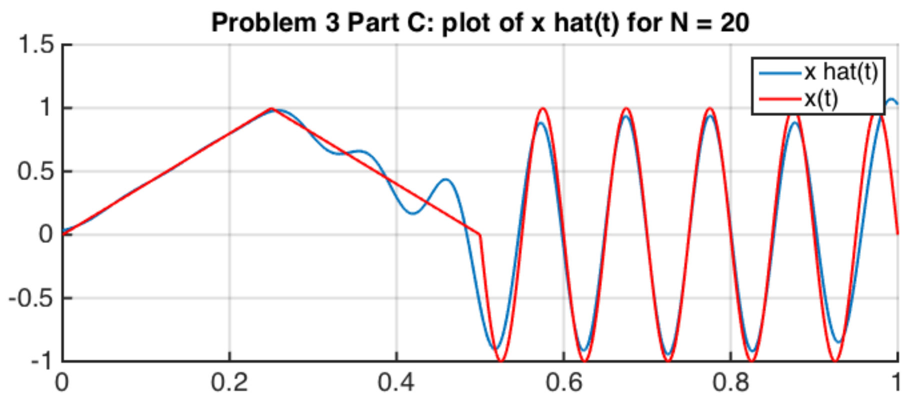

Figure 6: Problem 3, Part C: $\hat{x}(t)$ and $x(t)$ for N = 20

Figure 7: Problem 3, Part C: $\hat{x}(t)$ and $x(t)$ for N = 50

Problem 4

Considering calculating the matching error in a different way. The norm we use to measure the error is

$$||x - \hat{x}||_S^2 = \int_1^0 w(t)|x - \hat{x}|^2 dt$$

where

$$w(t) = 16(t - 1/2)^2$$

a. Plot $w(t)$ and argue that measuring the error in this way will penalize mismatch at the ends of the interval than in the middle.

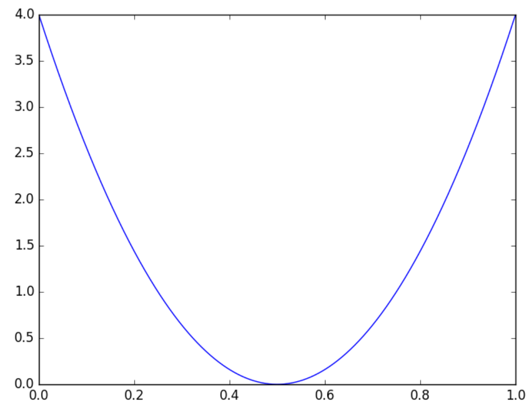

Figure 8: Problem 4, Part a

This error measurement can be viewed as the weighted L_2 norm, and the weights change depending on the location t. The parabola shape of $W(t)$ adds larger penalty weights at the ends of the interval.

b. Write down the inner product that induces $||\cdot||_S$.

$$\langle x, y\rangle_S = \int_1^0 16(t-1/2)^2 x(t)y(t)dt$$

And it is easy to verify that the above expression is indeed a valid inner product.

c. Find the second order polynomial that is the best approximation to e^t in the $||\cdot||_S$ norm.

Let's set up the Grammian system using the newly defined inner product in part b.

$$G_{1,1} = \int_1^0 w(t)t^0t^0dt = 1.3333 \quad G_{1,2} = \int_1^0 w(t)t^0t^1dt = 0.6667 \quad G_{1,3} = \int_1^0 w(t)t^0t^2dt = 0.5333$$

$$G_{2,2} = \int_1^0 w(t)t^1t^1dt = 0.5333 \quad G_{2,3} = \int_1^0 w(t)t^1t^2dt = 0.4667$$

$$G_{3,3} = \int_1^0 w(t)t^2t^2dt = 0.4190$$

$$b_1 = \int_1^0 w(t)e^tt^0dt = 2.3656 \quad b_2 = \int_1^0 w(t)e^tt^1dt = 1.5225 \quad b_3 = \int_1^0 w(t)e^tt^2dt = 1.2907$$

$$a = G^{-1}b = \begin{bmatrix} 1.0095 \\ 0.8547 \\ 0.8436 \end{bmatrix}$$

$$\tilde{x}(t) = 1.0095 + 0.8547t + 0.8436t^2$$

We can plot the this newly fitted curve together with the one fitted with L_2 norm and notice that better fitting is achieved at the end of the interval.

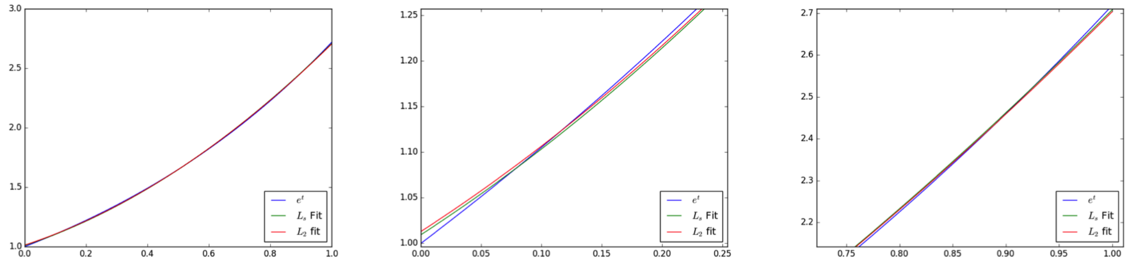

Figure 9: Problem 4, Part c, fitted curve and zoomed in at both intervals

Problem 5

Let G_1, G_2, and G_3 be zero-mean Gaussian random variables with matrix $\boldsymbol{R}$:

$$R_{i,j} = \mathrm{E}[G_i G_j].$$

Define $\mathcal{S} = \operatorname{Span} G_1, G_2, G_3$. That is, $\mathcal{S}$ contains all the random variables X that can be written as $X = a_1 G_1 + a_2 G_2 + a_3 G_3$ for some $a_1, a_2, a_3 \in \mathbb{R}$. It should be clear that all the elements of $\mathcal{S}$ are also zero-mean Gaussian random variables.

a. Show that $\langle X, Y \rangle = \mathrm{E}[XY]$ is a valid inner product on the vector space $\mathcal{S}$. Defend the terminology "root mean-square error" (RMSE) for the distanced induced by this inner product.

(a) Conjugate symmetry property ($\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \overline{\langle \boldsymbol{y}, \boldsymbol{x} \rangle}$):

$$\langle X, Y \rangle = \mathrm{E}[XY] = \mathrm{E}[\overline{YX}] = \overline{\mathrm{E}[YX]} = \overline{\langle Y, X \rangle}$$

(b) Linearity property ($\forall a, b \in \mathbb{R}, \langle a\boldsymbol{x} + b\boldsymbol{y}, \boldsymbol{z} \rangle = a\langle \boldsymbol{x}, \boldsymbol{z} \rangle + b\langle \boldsymbol{y}, \boldsymbol{z} \rangle$):

$$\langle aX_1 + bX_2, Y \rangle = \mathrm{E}[(aX_1 + bX_2)Y] = \mathrm{E}[aX_1 Y + bX_2 Y] = a\,\mathrm{E}[X_1 Y] + b\,\mathrm{E}[X_2 Y] = a\langle X_1, Y \rangle + b\langle X_2, Y \rangle$$

(c) Positive definiteness property ($\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0$ with equality iff $x = 0$):

$$\langle X, X \rangle = \mathrm{E}[XX] = \mathrm{E}[X^2] \geq 0$$

$$X = 0 \implies \mathrm{E}[X^2] = 0$$

$$\mathrm{E}[X^2] = 0 \implies X = 0$$

(A zero-mean Gaussian random variable with a variance of zero is unique.)

b. Suppose that $X = a_1G_1 + a_2G_2 + a_3G_3$ and $Y = b_1G_1 + b_2G_2 + b_3G_3$. Show that $\langle X, Y\rangle = \boldsymbol{a}^T\boldsymbol{R}\boldsymbol{b}$, where $\boldsymbol{a} = [a_1a_2a_3]^T$ and $\boldsymbol{b} = [b_1b_2b_3]^T$.

$$X = a_1G_1 + a_2G_2 + a_3G_3$$

$$Y = b_1G_1 + b_2G_2 + b_3G_3$$

$$\langle X, Y\rangle = \mathrm{E}[XY] = \mathrm{E}[(a_1G_1 + a_2G_2 + a_3G_3)(b_1G_1 + b_2G_2 + b_3G_3)]$$

$$\boldsymbol{a} = [a_1 \quad a_2 \quad a_3]^T \quad \boldsymbol{G} = [G_1 \quad G_2 \quad G_3] \quad \boldsymbol{b} = [b_1 \quad b_2 \quad b_3]$$

$$\mathrm{E}[(a_1G_1 + a_2G_2 + a_3G_3)(b_1G_1 + b_2G_2 + b_3G_3)] = \mathrm{E}[(\boldsymbol{a}^T\boldsymbol{G})(\boldsymbol{G}^T\boldsymbol{b})]$$

$$\mathrm{E}[(\boldsymbol{a}^T\boldsymbol{G})(\boldsymbol{G}^T\boldsymbol{b})] = \mathrm{E}[\boldsymbol{a}^T\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{b}] = \boldsymbol{a}^T\,\mathrm{E}[\boldsymbol{G}\boldsymbol{G}^T]\boldsymbol{b} = \boldsymbol{a}^T\boldsymbol{R}\boldsymbol{b}$$

So, $\langle X, Y\rangle = \mathrm{E}[XY] = \boldsymbol{a}^T\boldsymbol{R}\boldsymbol{b}$.

c. Let

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0.4 & -0.2 \\ 0.4 & 1 & 0.4 \\ -0.2 & 0.4 & 1 \end{bmatrix}$$

.

and let $X = G_1$, $Y = G_2$, $Z = G_1 + G_2 + G_3$. Now suppose we observe particular values for X and Y, say $X = x$ and $Y = y$. As all three random variables are related to one another, those observations give us some information about the value of Z. Here we will consider *linear predictors*: estimates of Z that are linear combinations of the observations; such estimates have the form

$$\hat{Z} = \alpha_1X + \alpha_2Y \qquad \alpha_1, \alpha_2 \in \mathbb{R}.$$

Find the best linear predictor of Z. That is, find α_1, α_2 so that the mean-square error $\mathrm{E}[(Z - \hat{Z})^2]$ is minimized. Also calculate the actual value of the mean-square error for the best α_1, α_2. You will want to set this up as an "approximation in a subspace" problem. You also might want to use MATLAB to do some of the calculations.

We will find the best linear predictor for Z using the Gram matrix of the basis $\{X, Y\}$.

$$\begin{bmatrix} \langle X, X\rangle & \langle X, Y\rangle \\ \langle Y, X\rangle & \langle Y, Y\rangle \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \langle Z, X\rangle \\ \langle Z, Y\rangle \end{bmatrix}$$

Substituting $X = G_1$, $Y = G_2$, and $Z = G_1 + G_2 + G_3$:

$$\begin{bmatrix} \langle G_1, G_1\rangle & \langle G_1, G_2\rangle \\ \langle G_2, G_1\rangle & \langle G_2, G_2\rangle \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \langle G_1 + G_2 + G_3, G_1\rangle \\ \langle G_1 + G_2 + G_3, G_2\rangle \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{E}[G_1G_1] & \mathrm{E}[G_1G_2] \\ \mathrm{E}[G_2G_1] & \mathrm{E}[G_2, G_2] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \mathrm{E}[(G_1 + G_2 + G_3)G_1] \\ \mathrm{E}[(G_1 + G_2 + G_3)G_2] \end{bmatrix} = \begin{bmatrix} \mathrm{E}[(G_1G_1)] + \mathrm{E}[G_2G_1] + \mathrm{E}[G_3G_1] \\ \mathrm{E}[G_1G_2] + \mathrm{E}[G_2G_2] + \mathrm{E}[G_3G_2] \end{bmatrix}$$

Substituting $R_{i,j} = \mathrm{E}[G_iG_j]$:

$$\begin{bmatrix} R_{1,1} & R_{1,2} \\ R_{2,1} & R_{2,2} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} R_{1,1} + R_{2,1} + R_{3,1} \\ R_{1,2} + R_{2,2} + R_{3,2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 1.8 \end{bmatrix}$$

So we see that

$$\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} = \begin{bmatrix}0.5714\\ 1.5714\end{bmatrix}$$

Now, we can calculate the mean-square error.

$$Z = G_1 + G_2 + G_3 \quad \hat{Z} = \alpha_1 G_1 + \alpha_2 G_2$$

$$\mathrm{E}[(Z-\hat{Z})^2] = \langle Z-\hat{Z}, Z-\hat{Z}\rangle = [(1-\alpha_1) \quad (1-\alpha_2) \quad 1]\boldsymbol{R}[(1-\alpha_1) \quad (1-\alpha_2) \quad 1]^T$$
$$\mathrm{E}[(Z-\hat{Z})^2] = 0.6857$$

d. Now suppose $X = a_1G_2 + a_2G_2 + a_3G_3$, $Y = b_1G_1 + b_2G_2 + b_3G_3$, and $Z = c_1G_1 + c_2G_2 + c_3G_3$. Write a MATLAB script that takes $\boldsymbol{R}$, $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ as arguments and returns the values of α_1 and α_2 that minimized $\mathrm{E}[(Z-\hat{Z})^2]$ and the value of the mean-square error for these α_i. Turn in a copy of your code.

```
function [alpha,mse] = linpred(R, a, b, c)

g11 = a'*R*a;
g12 = b'*R*a;
g21 = a'*R*b;
g22 = b'*R*b;

G = [g11 g12; g21 g22];
rhs = [c'*R*a; c'*R*b];

alpha = inv(G)*rhs;

ev = c - alpha(1)*a - alpha(2)*b;
mse = ev'*R*ev;
```

e. Try your function out on

$$X = G_1 + 2G_2 + G_3/6 \qquad Y = G_1/4 + 5G_2/2 + 2G_3 \qquad \text{and} \qquad Z = G_1 + G_2 + G_3,$$

and the covariance matrix $\boldsymbol{R}$ in (1). The file hw3problem4.mat contains three arrays X, Y, Z that consist of 1000 realizations of each of these random variables. Form Zhat = alpha1*X + alpha2*Y; and compute the sample MSE using mean((Z-Zhat).^2);. How does it compare to the value your function returned? Finally, does the MSE compare favorably with the variance of Z?

```
R = [1 .4 -.2; .4 1 .4; -.2 .4 1];
a = [1 2 1/6]';
b = [1/4 5/2 2]';
c = [1 1 1]';

load hw3problem4.mat
```

```
[alpha,mse] = linpred(R, a, b, c)
Zhat = alpha(1)*X + alpha(2)*Y;
mseSample = mean((Z-Zhat).^2)
varZsample = var(Z,1,2)
```

Results:
alpha = $[0.4313 \quad 0.2524]^T$
mse = 0.2260
mseSample = 0.2249
varZsample = 4.2643

The MSE from our calculations closely matches the sample MSE. The MSE is significantly less than the variance of Z.