1. **Analysis of a power control algorithm.** In this problem we consider again the power control method described in homework 1 problem 1. Please refer to this problem for the setup and background. In that problem, you expressed the power control method as a discrete-time linear dynamical system, and simulated it for a specific set of parameters, with several values of initial power levels, and two target SINRs. You found that for the target SINR value $\gamma = 3$, the powers converged to values for which each SINR exceeded $\gamma$, no matter what the initial power was, whereas for the larger target SINR value $\gamma = 5$, the powers appeared to diverge, and the SINRs did not appear to converge. You are going to analyze this, now that you know alot more about linear systems.

a) *Explain the simulations.* Explain your simulation results from the problem 1(b) for the given values of $G$, $\alpha$, $\sigma$, and the two SINR threshold levels $\gamma = 3$ and $\gamma = 5$.

b) *Critical SINR threshold level.* Let us consider fixed values of $G$, $\alpha$, and $\sigma$. It turns out that the power control algorithm works provided the SINR threshold $\gamma$ is less than some critical value $\gamma_{\text{crit}}$ (which might depend on $G$, $\alpha$, $\sigma$), and doesn’t work for $\gamma > \gamma_{\text{crit}}$. (‘Works’ means that no matter what the initial powers are, they converge to values for which each SINR exceeds $\gamma$.) Find an expression for $\gamma_{\text{crit}}$ in terms of $G \in \mathbb{R}^{n \times n}$, $\alpha$, and $\sigma$. Give the simplest expression you can. Of course you must explain how you came up with your expression.

**Solution.**

a) In the homework we found that the powers propagate according to a linear system. The power update rule for a single transmitter can be found by manipulating the definitions given in the problem.

$$p_i(t + 1) = \frac{\alpha \gamma p_i(t)}{S_i(t)} = \frac{\alpha \gamma p_i(t)q_i(t)}{s_i(t)} = \frac{\alpha \gamma p_i(t) \left[ \sigma + \sum_{j \neq i} G_{ij} p_j(t) \right]}{G_{ii} p_i(t)}$$

$$= \frac{\alpha \gamma \left[ \sigma + \sum_{j \neq i} G_{ij} p_j(t) \right]}{G_{ii}}$$

In matrix form the equations represent a linear dynamical system with constant input,
\[ p(t + 1) = Ap(t) + b. \]

\[
\begin{bmatrix}
  p_1(t + 1) \\
  p_2(t + 1) \\
  \vdots \\
  p_n(t + 1)
\end{bmatrix}
= \alpha \gamma
\begin{bmatrix}
  0 & G_{12} & G_{13} & \cdots & G_{1n} \\
  G_{21} & 0 & G_{23} & \cdots & G_{2n} \\
  G_{31} & G_{32} & 0 & \cdots & G_{3n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  G_{n1} & G_{n2} & G_{n3} & \cdots & 0
\end{bmatrix}
\begin{bmatrix}
  p_1(t) \\
  p_2(t) \\
  \vdots \\
  p_n(t)
\end{bmatrix}
+ \begin{bmatrix}
  0 \\
  0 \\
  \vdots \\
  0
\end{bmatrix}
\]

where \( A = \alpha \gamma P \). This is a discrete LDS, and is stable if and only if \(|\lambda_i| < 1\) for all \( i = 1, \ldots, n \), where \( \lambda_i \) are the eigenvalues of \( A \). When \( \gamma = 3 \) the eigenvalues of \( A \) are 0.6085, -0.3600, and -0.2485, so the system is stable; for all initial conditions, the powers converge to their equilibrium values.

Also, the SINR at each receiver \( i \), given by \( S_i \), converges to the same constant value \( \alpha \gamma \), which is enough for a successful signal reception. This can be shown by observing that at equilibrium \( p_i(t + 1) = p_i(t) = \bar{p}_i \), and the power update equation gives

\[ \bar{p}_i = \bar{p}_i(\alpha \gamma / S_i(t)). \]

After cancellation, we obtain the constant value for each SINR, \( S_i = \alpha \gamma \).

When \( \gamma = 5 \), the eigenvalues of \( A \) are 1.0141, -0.6000, and -0.4141. This system is unstable because of the first eigenvalue, so this means there are initial conditions from which the powers diverge.

\[
\text{>> inv(v)*b}
\]
-0.0670
-0.0000
-0.0182

b) The critical SINR threshold level is a function of dominant system eigenvalue. We will assume that matrix \( P \) is diagonizable and that its eigenvalues are ordered by their magnitude when forming \( \Lambda \) matrix. Using the property that scaling of any matrix scales its eigenvalues by the same constant, we can derive:

\[
A = \alpha \gamma P = \alpha \gamma T \Lambda T^{-1}
= T \text{diag}(\alpha \gamma \lambda_1, \ldots, \alpha \gamma \lambda_n) T^{-1}
\]

For a marginally stable system we need to have \(|\alpha \gamma \lambda_1| \leq 1\). Manipulating equation \( \alpha \gamma_{\text{crit}} |\lambda_1| = 1 \), we obtain the critical SINR threshold level,

\[
\gamma_{\text{crit}} = \frac{1}{\alpha |\lambda_1|}.
\]

2. **Real modal form.** We learned about the modal form of a system in class. Show that when some of eigenvalues of the dynamics matrix \( A \) are complex, the system can be put in **real modal form** (Assuming the eigenvectors of \( A \) are independent):
\[ S^{-1}AS = \text{diag} (\Lambda_r, M_{r+1}, M_{r+3}, \ldots, M_{n-1}) \]

where \( \Lambda_r = \text{diag}(\lambda_1, \ldots, \lambda_r) \) are the real eigenvalues, and

\[ M_j = \begin{bmatrix} \sigma_j & \omega_j \\ -\omega_j & \sigma_j \end{bmatrix}, \quad \lambda_j = \sigma_j + i\omega_j, \quad j = r + 1, r + 3, \ldots, n - 1 \]

where \( \lambda_j \) are the complex eigenvalues (one from each conjugate pair). Clearly explain what the matrix \( S \) is.

Generate a matrix \( A \) in \( \mathbb{R}^{10 \times 10} \) using \( A = \text{randn}(10) \). (The entries of \( A \) will be drawn from a unit normal distribution.) Find the eigenvalues of \( A \). If by chance they are all real, generate a new instance of \( A \). Find the real modal form of \( A \), i.e., a matrix \( S \) such that \( S^{-1}AS \) has the real modal form. Your solution should include the source code that you use to find \( S \), and some code that checks the results (i.e., computes \( S^{-1}AS \) to verify it has the required form).

Solution. Assuming \( A \) is diagonalizable, it can be written as \( A = T \Lambda T^{-1} \). Here

\[ \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_r, \lambda_{r+1}, \ldots, \lambda_n), \]

where \( \lambda_1, \ldots, \lambda_r \) are the real eigenvalues of \( A \) and \( \lambda_{r+1}, \ldots, \lambda_n \) are the complex eigenvalues of \( A \) and come in complex conjugate pairs. Let \( t_i \) be the \( i \)th column of \( T \). Take \( S \) to be

\[ S = [t_1 \cdots t_r \ \Re(t_{r+1}) \ \Im(t_{r+1}) \cdots \Re(t_{n-1}) \ \Im(t_{n-1})]. \]

Let us now prove why constructing \( S \) in this way will give us the desired result. Let \( v = \Re(v) + i\Im(v) \) be a complex eigenvector of \( A \) associated with the eigenvalue \( \lambda = \sigma + i\omega \). We must have \( Av = \lambda v \), i.e.,

\[ A(\Re(v) + i\Im(v)) = (\sigma + i\omega)(\Re(v) + i\Im(v)). \]

This implies that

\[ A\Re(v) = \sigma \Re(v) - \omega \Im(v), \quad A\Im(v) = \omega \Re(v) + \sigma \Im(v), \]

or equivalently

\[ A [\Re(v) \ \Im(v)] = [\Re(v) \ \Im(v)] \begin{bmatrix} \sigma & \omega \\ -\omega & \sigma \end{bmatrix}. \]

If we do the same derivation for the other complex conjugate eigenvalue pairs, we get the construction of \( S \) that was presented above.

Here is a short MATLAB script that checks that our solution is correct.

```matlab
randn('seed', 21395);
A = randn(10);
[V, D] = eig(A);
% Take eigenvectors of complex eigenvalues and arrange them in pairs.
S = zeros(10);
S(:,1) = V(:,5);
```
\( S(:,2) = V(:,6); \)
\( S(:,3) = V(:,7); \)
\( S(:,4) = V(:,10); \)
\( S(:,5) = \text{real}(V(:,1)); \)
\( S(:,6) = \text{imag}(V(:,1)); \)
\( S(:,7) = \text{real}(V(:,3)); \)
\( S(:,8) = \text{imag}(V(:,3)); \)
\( S(:,9) = \text{real}(V(:,8)); \)
\( S(:,10) = \text{imag}(V(:,8)); \)

% Inspect \( S^{-1}AS \)
\( \text{inv}(S)*A*S \)

3. **Output response envelope for linear system with uncertain initial condition.** We consider the autonomous linear dynamical system \( \dot{x} = Ax, \ y(t) = Cx(t), \) where \( x(t) \in \mathbb{R}^n \) and \( y(t) \in \mathbb{R}. \) We do not know the initial condition exactly; we only know that it lies in a ball of radius \( r \) centered at the point \( x_0: \)

\[ \| x(0) - x_0 \| \leq r. \]

We call \( x_0 \) the *nominal* initial condition, and the resulting output, \( y_{\text{nom}}(t) = Ce^{tA}x_0, \) the *nominal output*. We define the maximum output or upper output envelope as

\[ \bar{y}(t) = \max \{ y(t) \mid \| x(0) - x_0 \| \leq r \}, \]

i.e., the maximum possible value of the output at time \( t \), over all possible initial conditions. (Here you can choose a different initial condition for each \( t \); you are not required to find a single initial condition.) In a similar way, we define the minimum output or lower output envelope as

\[ \underline{y}(t) = \min \{ y(t) \mid \| x(0) - x_0 \| \leq r \}, \]

i.e., the minimum possible value of the output at time \( t \), over all possible initial conditions.

a) Explain how to find \( \bar{y}(t) \) and \( \underline{y}(t) \), given the problem data \( A, C, x_0, \) and \( r. \)

b) Carry out your method on the problem data in *uie_data.m*. On the same axes, plot \( y_{\text{nom}}, \bar{y}, \) and \( \underline{y} \) versus \( t \), over the range \( 0 \leq t \leq 10. \)

**Solution.** We have

\[ y(t) = Cx(t) = Ce^{tA}x(0) = Ce^{tA}(x(0) - x_0 + x_0) = y_{\text{nom}}(t) + Ce^{tA}(x(0) - x_0). \]

Using the Cauchy-Schwarz inequality, we get

\[ -\|(e^{tA})^T C^T\| \| x(0) - x_0 \| \leq Ce^{tA}(x(0) - x_0) \leq \|(e^{tA})^T C^T\| \| x(0) - x_0 \|. \]

However, since \( \| x(0) - x_0 \| \leq r \), we can deduce that

\[ y_{\text{nom}}(t) - \|(e^{tA})^T C^T\| r \leq y(t) \leq y_{\text{nom}}(t) + \|(e^{tA})^T C^T\| r. \]
In fact, these inequalities are tight. This is so, since if

\[ x(0) = x_0 + r \frac{e^{tA^T}C^T}{\|e^{tA^T}C^T\|} (e^{tA^T}C^T) \]

then

\[ y(t) = y_{nom}(t) + \|e^{tA^T}C^T\| \|

Similarly, if

\[ x(0) = x_0 - r \frac{e^{tA^T}C^T}{\|e^{tA^T}C^T\|} (e^{tA^T}C^T) \]

then

\[ y(t) = y_{nom}(t) - \|e^{tA^T}C^T\| \|

Therefore we finally have

\[ \underline{y}(t) = y_{nom}(t) - \|e^{tA^T}C^T\| \|
\]

\[ \bar{y}(t) = y_{nom}(t) + \|e^{tA^T}C^T\| \|
\]

The following matlab code performs this for the given data:

```matlab
clear all
uie_data
T = linspace(0,10,1000);
y_nom = [];
y_over = [];
y_under = [];
for k = T
    v = C*expm(k*A);
y_c = v*x_0;
y_nom = [y_nom y_c];
y_under = [y_under y_c-norm(v)*r];
y_over = [y_over y_c+norm(v)*r];
end
figure
plot(T,y_nom,'b-',T,y_under,'r--',T,y_over,'r--')
```

5
This code generates the following figure:

Note that the bounds actually move towards each other near $t \approx 5$, but then diverge again.

4. Spectral mapping theorem. Suppose $f : \mathbb{R} \rightarrow \mathbb{R}$ is analytic, i.e., given by a power series expansion

\[ f(u) = a_0 + a_1 u + a_2 u^2 + \cdots \]

(where $a_i = f^{(i)}(0)/(i!)$). (You can assume that we only consider values of $u$ for which this series converges.) For $A \in \mathbb{R}^{n \times n}$, we define $f(A)$ as

\[ f(A) = a_0 I + a_1 A + a_2 A^2 + \cdots \]

(again, we’ll just assume that this converges).

Suppose that $A v = \lambda v$, where $v \neq 0$, and $\lambda \in \mathbb{C}$. Show that $f(A) v = f(\lambda) v$ (ignoring the issue of convergence of series). We conclude that if $\lambda$ is an eigenvalue of $A$, then $f(\lambda)$ is an eigenvalue of $f(A)$. This is called the spectral mapping theorem.

To illustrate this with an example, generate a random $3 \times 3$ matrix, for example using $A=\text{randn}(3)$. Find the eigenvalues of $(I + A) (I - A)^{-1}$ by first computing this matrix, then finding its eigenvalues, and also by using the spectral mapping theorem. (You should get very close agreement; any difference is due to numerical round-off errors in the various computations.)

Solution. Assuming $A v = \lambda v$, $v \neq 0$, we have

\[ f(A) v = a_0 I v + a_1 A v + a_2 A^2 v + \cdots \]
\[ = a_0 v + a_1 \lambda v + a_2 \lambda^2 v + \cdots \]
\[ = f(\lambda) v, \]
using $A^k v = \lambda^k v$.

The matlab code that illustrates this is shown below.

```matlab
randn('state',0); % makes it repeatable
A=randn(3);
% eigenvalues of A
lambdas=eig(A);
% create matrix f(A)=(I+A)(I-A)^(-1)
fA=(eye(3)+A)*inv(eye(3)-A);
% eigenvalues of f(A)
lambdas_fA=eig(fA);
% eigenvalues of B via spectral mapping theorem
lambdas_fA_smt = (1+lambdas)./(1-lambdas);
% compare (need not be in same order!)
[lambdas_fA lambdas_fA_smt]
```

5. **Interconnection of linear systems.** Often a linear system is described in terms of a block diagram showing the interconnections between components or subsystems, which are themselves linear systems. In this problem you consider the specific interconnection shown below:

Here, there are two subsystems $S$ and $T$. Subsystem $S$ is characterized by

$$\dot{x} = Ax + B_1 u + B_2 w_1,$$

and subsystem $T$ is characterized by

$$\dot{z} = Fz + G_1 v + G_2 w_2,$$

$$w_1 = H_1 z, \quad y = H_2 z + J w_2.$$

We don’t specify the dimensions of the signals (which can be vectors) or matrices here. You can assume all the matrices are the correct (i.e., compatible) dimensions. Note that the subscripts in the matrices above, as in $B_1$ and $B_2$, refer to different matrices. Now the problem. Express the overall system as a single linear dynamical system with input, state, and output given by

$$\begin{bmatrix} u \\ v \end{bmatrix}, \quad \begin{bmatrix} x \\ z \end{bmatrix}, \quad y,$$

respectively. Be sure to explicitly give the input, dynamics, output, and feedthrough matrices of the overall system. If you need to make any assumptions about the rank or invertibility of any matrix you encounter in your derivations, go ahead. But be sure to let us know what assumptions you are making.
Solution. This one is easier than it might appear. All we need to do is write down all the equations for this system, and massage them to be in the form of a linear dynamical system with the given input, state, and output. Substituting the expression for \( w_1 \) into the first set of equations gives

\[
\dot{x} = Ax + B_1 u + B_2 H_1 z, \quad w_2 = Cx + D_1 u + D_2 H_1 z.
\]

Similarly, substituting the expression for \( w_2 \) into the second set of equations yields

\[
\dot{z} = Fz + G_1 v + G_2 (Cx + D_1 u + D_2 H_1 z), \quad y = H_2 z + J(Cx + D_1 u + D_2 H_1 z).
\]

Now we just put this together into the required form:

\[
\begin{bmatrix}
\dot{x} \\
\dot{z}
\end{bmatrix} =
\begin{bmatrix}
A & B_2 H_1 \\
G_2 C & F + G_2 D_2 H_1
\end{bmatrix}
\begin{bmatrix}
x \\
z
\end{bmatrix} +
\begin{bmatrix}
B_1 & 0 \\
G_2 D_1 & G_1
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
\]

and

\[
y = 
\begin{bmatrix}
J C & H_2 + J D_2 H_1
\end{bmatrix}
\begin{bmatrix}
x \\
z
\end{bmatrix} + 
\begin{bmatrix}
J D_1 & 0
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
\]

6. Analysis of investment allocation strategies. Each year or period (denoted \( t = 0, 1, \ldots \)) an investor buys certain amounts of one-, two-, and three-year certificates of deposit (CDs) with interest rates 5\%, 6\%, and 7\%, respectively. (We ignore minimum purchase requirements, and assume they can be bought in any amount.)

- \( B_1(t) \) denotes the amount of one-year CDs bought at period \( t \).
- \( B_2(t) \) denotes the amount of two-year CDs bought at period \( t \).
- \( B_3(t) \) denotes the amount of three-year CDs bought at period \( t \).

We assume that \( B_1(0) + B_2(0) + B_3(0) = 1 \), i.e., a total of 1 is to be invested at \( t = 0 \). (You can take \( B_j(t) \) to be zero for \( t < 0 \).) The total payout to the investor, \( p(t) \), at period \( t \) is a sum of six terms:

- 1.05\( B_1(t-1) \), i.e., principle plus 5\% interest on the amount of one-year CDs bought one year ago.
- 1.06\( B_2(t-2) \), i.e., principle plus 6\% interest on the amount of two-year CDs bought two years ago.
- 1.07\( B_3(t-3) \), i.e., principle plus 7\% interest on the amount of three-year CDs bought three years ago.
- 0.06\( B_2(t-1) \), i.e., 6\% interest on the amount of two-year CDs bought one year ago.
- 0.07\( B_3(t-1) \), i.e., 7\% interest on the amount of three-year CDs bought one year ago.
- 0.07\( B_3(t-2) \), i.e., 7\% interest on the amount of three-year CDs bought two years ago.
The total wealth held by the investor at period $t$ is given by
\[ w(t) = B_1(t) + B_2(t) + B_2(t-1) + B_3(t) + B_3(t-1) + B_3(t-2). \]

Two re-investment allocation strategies are suggested.

- **The 35-35-30 strategy.** The total payout is re-invested 35% in one-year CDs, 35% in two-year CDs, and 30% in three-year CDs. The initial investment allocation is the same: $B_1(0) = 0.35$, $B_2(0) = 0.35$, and $B_3(0) = 0.30$.

- **The 60-20-20 strategy.** The total payout is re-invested 60% in one-year CDs, 20% in two-year CDs, and 20% in three-year CDs. The initial investment allocation is $B_1(0) = 0.60$, $B_2(0) = 0.20$, and $B_3(0) = 0.20$.

a) Describe the investments over time as a linear dynamical system $x(t+1) = Ax(t)$, $y(t) = Cx(t)$ with $y(t)$ equal to the total wealth at time $t$. Be very clear about what the state $x(t)$ is, and what the matrices $A$ and $C$ are. You will have two such linear systems: one for the 35-35-30 strategy and one for the 60-20-20 strategy.

b) **Asymptotic wealth growth rate.** For each of the two strategies described above, determine the asymptotic growth rate, defined as $\lim_{t \to \infty} w(t+1)/w(t)$. (If this limit doesn’t exist, say so.) *Note:* simple numerical simulation of the strategies (e.g., plotting $w(t+1)/w(t)$ versus $t$ to guess its limit) is not acceptable. (You can, of course, simulate the strategies to check your answer.)

c) **Asymptotic liquidity.** The total wealth at time $t$ can be divided into three parts:
- $B_1(t) + B_2(t-1) + B_3(t-2)$ is the amount that matures in one year (i.e., the amount of principle we could get back next year)
- $B_2(t) + B_3(t-1)$ is the amount that matures in two years
- $B_3(t)$ is the amount that matures in three years (i.e., is least liquid)

We define liquidity ratios as the ratio of these amounts to the total wealth:
\[
L_1(t) = (B_1(t) + B_2(t-1) + B_3(t-2))/w(t),
L_2(t) = (B_2(t) + B_3(t-1))/w(t),
L_3(t) = B_3(t)/w(t).
\]

For the two strategies above, do the liquidity ratios converge as $t \to \infty$? If so, to what values? *Note:* as above, simple numerical simulation alone is not acceptable.

d) Suppose you could change the initial investment allocation for the 35-35-30 strategy, i.e., choose some other nonnegative values for $B_1(0)$, $B_2(0)$, and $B_3(0)$ that satisfy $B_1(0) + B_2(0) + B_3(0) = 1$. What allocation would you pick, and how would it be better than the (0.35, 0.35, 0.30) initial allocation? (For example, would the asymptotic growth rate be larger?) How much better is your choice of initial investment allocations? *Hint for part d:* think very carefully about this one. *Hint for whole problem:* watch out for nondiagonalizable, or nearly nondiagonalizable, matrices. Don’t just blindly type in matlab commands; check to make sure you’re computing what you think you’re computing.
Solution.

a) We take as state vector

\[ x(t) = \begin{bmatrix} B_1(t) \\ B_2(t) \\ B_2(t-1) \\ B_3(t) \\ B_3(t-1) \\ B_3(t-2) \end{bmatrix}. \]

The components consist of the six critical quantities: the amount of one-, two-, and three-year CDs held of each possible maturity date (i.e., one, two, and three years). We can express the system as

\[ x(t+1) = Fx(t) + Gu(t) \]

where \( u(t) \in \mathbb{R}^3 \) gives the amount of one, two, and three year CDs purchased at \( t+1 \), and

\[ F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \]

The payout is given by \( p(t) = hx(t) \), where

\[ h = [1.05 \ 0.06 \ 1.06 \ 0.07 \ 0.07 \ 1.07]. \]

The allocation of the payout is given by \( u(t) = [0.35 \ 0.35 \ 0.30]^T p(t) \) for the 35-35-30 strategy and by \( u(t) = [0.60 \ 0.20 \ 0.20]^T p(t) \) for the 60-20-20 strategy. Finally, putting it all together, we end up with \( x(t+1) = Ax(t) \) where

\[ A_{35} = F + G[0.35 \ 0.35 \ 0.30]^T h \]

for the 35-35-30 case, and similarly for the 60-20-20 case. To be fully explicit, we have

\[ A_{35} = \begin{bmatrix} 0.3675 & 0.0210 & 0.3710 & 0.0245 & 0.0245 & 0.3745 \\ 0.3675 & 0.0210 & 0.3710 & 0.0245 & 0.0245 & 0.3745 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 0.3150 & 0.0180 & 0.3180 & 0.0210 & 0.0210 & 0.3210 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \end{bmatrix}, \]

\[ A_{60} = \begin{bmatrix} 0.6300 & 0.0360 & 0.6360 & 0.0420 & 0.0420 & 0.6420 \\ 0.2100 & 0.0120 & 0.2120 & 0.0140 & 0.0140 & 0.2140 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 0.2100 & 0.0120 & 0.2120 & 0.0140 & 0.0140 & 0.2140 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \end{bmatrix}. \]

The following matlab code calculates the matrix \( A \) for each allocation strategy.

\[ AA = [0 \ 0 \ 0 \ 0 \ 0 \ 0; \]
BB = [1 0 0;
0 1 0;
0 0 0;
0 0 1;
0 0 0;
0 0 0];

PP = [1.05 0.06 1.06 0.07 0.07 1.07]; % PP*x gives payout
alloc35 = [.35 .35 .3]'; % for 35-35-30 allocation
A35 = AA+BB*alloc35*PP;
alloc60 = [.60 .20 .20]'; % for 60-20-20 allocation
A60 = AA+BB*alloc60*PP;
WW=ones(1,6); %WW*x gives total wealth

This yields:

A35 =
0.3675 0.0210 0.3710 0.0245 0.0245 0.3745
0.3675 0.0210 0.3710 0.0245 0.0245 0.3745
0 1.0000 0 0 0 0
0.3150 0.0180 0.3180 0.0210 0.0210 0.3210
0 0 0 1.0000 0 0
0 0 0 0 1.0000 0

A60 =
0.6300 0.0360 0.6360 0.0420 0.0420 0.6420
0.2100 0.0120 0.2120 0.0140 0.0140 0.2140
0 1.0000 0 0 0 0
0.2100 0.0120 0.2120 0.0140 0.0140 0.2140
0 0 0 1.0000 0 0
0 0 0 0 1.0000 0

There are several other correct answers. For example, some people decided to use a state vector that included the past three samples of each of the $B_i$'s, i.e., a state of dimension 9. Provided no errors were made, this works fine. At the other end, some people found a state description that has a state dimension of 4.

b) Asymptotic wealth growth rate. This is going to depend on the eigenvalues of $A$, so first we check the eigenvalues, which turn out to be

$1.0627, -0.3266 \pm 0.4421i, 0, 0, 0$

for $A_{35}$, and

$1.0598, -0.2019 \pm 0.4015i, 0, 0, 0$
for $A_{60}$. Each matrix has one real, positive eigenvalue that has larger magnitude than one, and than the other eigenvalues, which in fact all have magnitude less than one. Therefore, as $t \to \infty$,

$$A^t \to \lambda^t v w^T,$$

where $\lambda$ is the dominant eigenvalue, and $v$ and $w$ are the right and left eigenvectors associated with $\lambda$, normalized so that $w^T v = 1$. There is a subtlety in computing $v$ and $w$, since the matrices have nontrivial Jordan form. Forming a matrix of right eigenvectors and inverting won’t work because there is no set of independent eigenvectors! But finding the eigenvector of $A^T$ associated with the dominant eigenvector, and then normalizing properly, does work. It follows that as $t \to \infty$ we have

$$x(t) \to \lambda^t v w^T x(0).$$

A quick check shows that $w^T x(0) \neq 0$ (which means the initial condition does excite the dominant mode), so as $t \to \infty$, $x(t)$ grows exponentially with rate $\lambda$. The wealth satisfies

$$w(t) \to \lambda^t (1^T v)(w^T x(0))$$

as $t \to \infty$. Hence $w(t + 1)/w(t)$ converges to $\lambda$. For the 35-35-30 scheme, the growth rate is 1.0627, i.e., 6.27% per year. For the 60-20-20 scheme, the growth rate is 1.0598, i.e., 5.98% per year. It makes sense that the first scheme has a higher growth rate since a higher fraction is invested in higher-yield CDs. A few people made numerical errors and ended up with growth rates of, say, 45%. How can investments in a portfolio of 5%, 6%, and 7% CDs end up yielding 45%? We can only assume this problem was solved very late at night . . .

c) Asymptotic liquidity. The analysis above shows that as $t \to \infty$, the state vector grows exponentially with rate given by $\lambda$, and asymptotic distribution shape given by $v$, the right eigenvector associated with the dominant eigenvalue $\lambda$. Define

$$l_1 = [1 0 1 0 0 1]^T, \quad l_2 = [0 1 0 0 1 0]^T, \quad l_3 = [0 0 0 1 0 0]^T,$$
so that \( L_i(t) = l_i x(t)/1^T x(t) \). Since \( x(t) \to \lambda^T w^T x(0) \) we have \( L_i(t) \to \frac{\lambda^T w}{1^T} \). Therefore the asymptotic values of \( L_1, L_2, \) and \( L_3 \) for the 35–35–30 strategy are 0.5034, 0.3368, and 0.1598, respectively. For the 60–20–20 strategy, they are 0.6215, 0.2499, and 0.1286, respectively. It makes sense that the second scheme is asymptotically more liquid, since a higher fraction is invested in fast-maturing CDs.

![Liquidity ratios for 35-35-30 strategy](image1)

![Liquidity ratios for 60-20-20 strategy](image2)

d) When we change \( x(0) \) we affect the term \( w^T x(0) \) in the asymptotic state. Hence by changing the initial condition we do not change the asymptotic growth rate. (Technically
there is one stupid case where we can change it: if we could arrange for $w^T x(0) = 0$, then asymptotic growth rate would be determined by the second largest eigenvalue. In fact for this problem you can prove that $w^T x(0) = 0$ is impossible since the vector $w$ has all positive components.) The asymptotic total wealth, which is $\lambda' (w^T x(0)) (1^T v)$, depends linearly on $w^T x(0)$; evidently we want to choose $x(0)$ to maximize $w^T x(0)$. Thus our problem is to maximize

$$w^T [B_1(0) B_2(0) 0 B_3(0) 0 0] = w_1 B_1(0) + w_2 B_2(0) + w_4 B_3(0),$$

subject to

$$B_1(0) + B_2(0) + B_3(0) = 1, \quad B_1(0) \geq 0, \quad B_2(0) \geq 0, \quad B_3(0) \geq 0.$$ 

The solution is clear if you think about it: find the largest of $w_1$, $w_2$, and $w_4$ and allocate all the initial investment in the corresponding $B_i$. We can obtain the dominant left eigenvector $w$ in matlab, by finding the dominant right eigenvector of $A^T$:

```matlab
[W35, Lambda35] = eig(A35'); % to find the left eigenvectors
w1 = W35(:, 1) % the dominant left eigenvector
w1 = w1 / (w1' * v1_35) % normalize w1 so that w1' * v = 1
ans =
0.4045
0.4074
0.4084
0.4174
0.4149
0.4122
```

The optimal initial allocation will be $B_1(0) = 0$, $B_2(0) = 0$, and $B_3(0) = 1$. It’s just a bit better than the original scheme, since with original scheme $w^T x(0) = 0.4094$, whereas with our new optimal initial investment allocation we have $w^T x(0) = 0.4174$ (only 1.95% better). Plots below show the total wealth for the original allocation (solid line) and the optimal allocation (dashed line).
Several people guessed this solution, but didn’t justify it. Others attempted SVD methods, or attempted to set up the initial condition along $v$, maybe by projecting $v$ onto the first, second, and fourth components. None of these methods is correct.