1. Optimal operation of a two-state chemical reactor. Consider a chemical reactor containing \( n \) compounds, labeled \( 1, \ldots, n \). Let \( x_i(t) \) be the amount of compound \( i \) in the reactor at time \( t \). The chemical reactor has two modes of operation, labeled 1 and 2. (For example, the first mode may be operating the reactor at a low temperature, and the second mode may be operating the reactor at a high temperature.) For simplicity we assume that the mode of operation can be changed instantaneously. When we operate the reactor in mode \( j \), the vector of compound amounts evolves according to the equation

\[
\dot{x}(t) = A_j x(t).
\]

We are given the vector \( x(0) \) of initial compound amounts, and the dynamics matrices \( A_1 \) and \( A_2 \). Our objective is to maximize the amount of compound \( k \) at time \( T \), where \( k \in \{1, \ldots, n\} \) and \( T > 0 \) are given.

a) Suppose the reactor operates in mode 1 for \( 0 \leq t \leq T_0 \), and mode 2 for \( T_0 < t \leq T \). Explain how to choose the time \( T_0 \) in order to maximize the amount of compound \( k \) at time \( T \). Your answer only needs to be accurate to two decimal digits.

b) Apply your method to the data given in `chemical_reactor_data.m`. Report the optimal value of \( T_0 \) and the corresponding amount of compound \( k \) at time \( T \); submit a plot showing all of the components of \( x(t) \) as functions of time on a single set of axes.

c) Suppose the reactor operates in mode 1 for \( 0 \leq t \leq T_1 \) and \( T_2 < t \leq T \), and mode 2 for \( T_1 < t \leq T_2 \). Explain how to choose the times \( T_1 \) and \( T_2 \) in order to maximize the amount of compound \( k \) at time \( T \). Your answers for \( T_1 \) and \( T_2 \) only need to be accurate to two decimal digits.

d) Apply your method to the data given in `chemical_reactor_data.m`. Report the optimal values of \( T_1 \) and \( T_2 \) and the corresponding amount of compound \( k \) at time \( T \); submit a plot showing all of the components of \( x(t) \) as functions of time on a single set of axes.

Solution.

a) Since the reactor operates in mode 1 for \( 0 \leq t \leq T_0 \), we have that

\[
x(T_0) = \exp(T_0 A_1) x(0).
\]

Similarly, because the reactor operates in mode 2 for \( T_0 < t \leq T \), we have that

\[
x(T) = \exp((T - T_0) A_2) x(T_0) = \exp((T - T_0) A_2) \exp(T_0 A_1) x(0).
\]
We want to maximize the amount of compound \( k \) at time \( T \):

\[
x_k(T) = e_k^T x(T) = e_k^T \exp((T - T_0)A_2) \exp(T_0 A_1) x(0).
\]

Since our answer for \( T_0 \) only needs to be accurate to two decimal digits, we simply use the formula above to compute \( x_k(T) \) for all \( T_0 \in \{0.00, 0.01, 0.02, \ldots, T\} \), and then choose the value that maximizes \( x_k(T) \).

b) We find that the optimal time to change the mode of operation is \( T_0 = 0.61 \), and the corresponding amount of compound \( k \) at time \( T \) is 0.3655. A plot showing all of the components of \( x(t) \) as functions of time is given in ??.

c) Since the reactor operates in mode 1 for \( 0 \leq t \leq T_1 \), we have that

\[
x(T_1) = \exp(T_1 A_1) x(0).
\]

Similarly, because the reactor operates in mode 2 for \( T_1 < t \leq T_2 \), we have that

\[
x(T_2) = \exp((T_2 - T_1) A_2) x(T_1) = \exp((T_2 - T_1) A_2) \exp(T_1 A_1) x(0).
\]

Finally, since the reactor operates in mode 1 for \( T_2 < t \leq T \), we have that

\[
x(T) = \exp((T - T_2) A_1) x(T_2) = \exp((T - T_2) A_1) \exp((T_2 - T_1) A_2) \exp(T_1 A_1) x(0).
\]

We want to maximize

\[
x_k(T) = e_k^T x(T) = e_k^T \exp((T - T_2) A_1) \exp((T_2 - T_1) A_2) \exp(T_1 A_1) x(0).
\]

Since our answers for \( T_1 \) and \( T_2 \) only need to be accurate to two decimal digits, we simply use the formula above to compute \( x_k(T) \) for all \( T_1, T_2 \in \{0.00, 0.01, 0.02, \ldots, T\} \) such that \( T_1 < T_2 \), and then choose the values that maximize \( x_k(T) \).

d) We find that the optimal times to change the mode of operation are \( T_1 = 0.49 \) and \( T_2 = 0.87 \), and the corresponding amount of compound \( k \) at time \( T \) is 0.3763. A plot showing all of the components of \( x(t) \) as functions of time is given in ??.
xk(i) = sparse(1,k,1,1,n) ...
    * expm((T - t(i)) * A2) ...
    * expm(t(i) * A1) * x0;
end
[~, idx] = max(xk);
T0_opt = t(idx)
xk_opt = xk(idx)

% simulate the system using the optimal value of T0
 tplot = 0.00:0.01:T;
dt = tplot(2) - tplot(1);
xplot = [x0 nan(n,length(tplot)-1)];
for i = 1:(length(tplot)-1)
    if tplot(i) <= T0_opt
        xplot(:,i+1) = expm(dt * A1) * xplot(:,i);
    else
        xplot(:,i+1) = expm(dt * A2) * xplot(:,i);
    end
end

figure();
plot(t , xplot );
xlabel('t');
ylabel('x(t)');
legend('x1(t)' , 'x2(t)' , 'x3(t)');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% find the optimal times to change the mode of the system
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xk = -Inf(length(t) , length(t));
for i = 1:(length(t)-1)
    for j = (i+1):length(t)
        xk(i,j) = sparse(1,k,1,1,n) ...
            * expm((T - t(j)) * A1) ...
            * expm((t(j) - t(i)) * A2) ...
            * expm(t(i) * A1) * x0;
    end
end
[~, idx] = max(xk(:));
[iopt , jopt] = ind2sub(size(xk) , idx);
T1_opt = t(iopt)
T2_opt = t(jopt)
xk_opt = xk(idx)
% simulate the system using the optimal values of T1 and T2
xplot = [x0 nan(n,length(tplot)-1)];
for i = 1:(length(tplot)-1)
    if tplot(i) <= T1_opt || tplot(i) > T2_opt
        xplot(:,i+1) = expm(dt * A1) * xplot(:,i);
    else
        xplot(:,i+1) = expm(dt * A2) * xplot(:,i);
    end
end
figure();
plot(t , xplot);
xlabel('t');
ylabel('x(t)');
legend('x1(t)' , 'x2(t)' , 'x3(t)');
grid on;
Figure 2: the trajectory of $x(t)$ with two optimal mode changes
2. 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, $p(t+1) = Ap(t) + b$.

$$\begin{pmatrix} p_1(t+1) \\ p_2(t+1) \\ p_3(t+1) \\ \vdots \\ p_n(t+1) \\ p(t+1) \end{pmatrix} = \alpha \gamma \begin{pmatrix} 0 & G_{12} & G_{13} & \cdots & G_{1n} \\ G_{22} & 0 & G_{23} & \cdots & G_{2n} \\ G_{32} & G_{33} & 0 & \cdots & G_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{n2} & G_{n3} & G_{n4} & \cdots & 0 \end{pmatrix} \begin{pmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ \vdots \\ p_n(t) \end{pmatrix} + \begin{pmatrix} \alpha \gamma \sigma \\ G_{11} \\ G_{22} \\ G_{33} \\ \vdots \end{pmatrix}$$

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.

$$\gg \text{inv}(v) * b$$
$$-0.0670$$
$$-0.0000$$
$$-0.0182$$

b) The critical SINR threshold level is a function of dominant system eigenvalue. We will assume that matrix $P$ is diagonalizable 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, 

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

3. Some basic properties of eigenvalues. Show the following:

   a) The eigenvalues of $A$ and $A^T$ are the same.

   b) $A$ is invertible if and only if $A$ does not have a zero eigenvalue.

   c) If the eigenvalues of $A$ are $\lambda_1, \ldots, \lambda_n$ and $A$ is invertible, then the eigenvalues of $A^{-1}$ are $1/\lambda_1, \ldots, 1/\lambda_n$.

   d) The eigenvalues of $A$ and $T^{-1}AT$ are the same.

*Hint:* you’ll need to use the facts that $\det A = \det(A^T)$, $\det(AB) = \det A \det B$, and, if $A$ is invertible, $\det A^{-1} = 1/\det A$. 

7
Solution.

a) The eigenvalues of a matrix $A$ are given by the roots of the polynomial $\det(sI - A)$. From determinant properties we know that $\det(sI - A) = \det(sI - A)^T = \det(sI - A^T)$. We conclude that the eigenvalues of $A$ and $A^T$ are the same.

b) First we recall that $A$ is invertible if and only if $\det(A) \neq 0$. But $\det(A) \neq 0 \iff \det(-A) \neq 0$.

i. If $0$ is an eigenvalue of $A$, then $\det(sI - A) = 0$ when $s = 0$. It follows that $\det(-A) = 0$ and thus $\det(A) = 0$, and $A$ is not invertible. From this fact we conclude that if $A$ is invertible, then $0$ is not an eigenvalue of $A$.

ii. If $A$ is not invertible, then $\det(A) = \det(-A) = 0$. This means that, for $s = 0$, $\det(sI - A) = 0$, and we conclude that in this case $0$ must be an eigenvalue of $A$. From this fact it follows that if $0$ is not an eigenvalue of $A$, then $A$ is invertible.

c) From the results of the last item we see that $0$ is not an eigenvalue of $A$. Now consider the eigenvalue/eigenvector pair $(\lambda_i, x_i)$ of $A$. This pair satisfies $Ax_i = \lambda_i x_i$. Now, since $A$ is invertible, $\lambda_i$ is invertible. Multiplying both sides by $A^{-1}$ and $\lambda_i^{-1}$ we have $\lambda_i^{-1} x_i = A^{-1} x_i$, and from this we conclude that the eigenvalues of the inverse are the inverse of the eigenvalues.

d) First we note that $\det(sI - A) = \det(I(sI - A)) = \det(T^{-1}T(sI - A))$. Now, from determinant properties, we have $\det(T^{-1}T(sI - A)) = \det(T^{-1}(sI - A)T)$. But this is also equal to $\det(sI - T^{-1}AT)$, and the conclusion is that the eigenvalues of $A$ and $T^{-1}AT$ are the same.

4. Controlling a system using the initial conditions. Consider the mechanical system shown below:

Here $q_i$ give the displacements of the masses, $m_i$ are the values of the masses, and $k_i$ are the spring stiffnesses, respectively. The dynamics of this system are

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_1 + k_2}{m_1} & \frac{k_2}{m_1} & 0 & 0 \\ \frac{k_2}{m_2} & -\frac{k_2}{m_2} & 0 & 0 \end{bmatrix} x$$

where the state is given by

$$x = \begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}.$$
Immediately before $t = 0$, you are able to apply a strong impulsive force $\alpha_i$ to mass $i$, which results in initial condition

$$x(0) = \begin{bmatrix} 0 \\ 0 \\ \alpha_1/m_1 \\ \alpha_2/m_2 \end{bmatrix}.$$  

(i.e., each mass starts with zero position and a velocity determined by the impulsive forces.)

This problem concerns selection of the impulsive forces $\alpha_1$ and $\alpha_2$. For parts a–c below, the parameter values are

$$m_1 = m_2 = 1, \ k_1 = k_2 = 1.$$  

Consider the following specifications:

a) $q_2(10) = 2$

b) $q_1(10) = 1$, $q_2(10) = 2$

c) $q_1(10) = 1$, $q_2(10) = 2$, $\dot{q}_1(10) = 0$, $\dot{q}_2(10) = 0$

d) $q_2(10) = 2$ when the parameters have the values used above (i.e., $m_1 = m_2 = 1$, $k_1 = k_2 = 1$), and also, $q_2(10) = 2$ when the parameters have the values $m_1 = 1$, $m_2 = 1.3$, $k_1 = k_2 = 1$.

Determine whether each of these specifications is feasible or not (i.e., whether there exist $\alpha_1$, $\alpha_2 \in \mathbb{R}$ that make the specification hold). If the specification is feasible, find the particular $\alpha_1$, $\alpha_2$ that satisfy the specification and minimize $\alpha_1^2 + \alpha_2^2$. If the specification is infeasible, find the particular $\alpha_1$, $\alpha_2$ that come closest, in a least-squares sense, to satisfying the specification. (For example, if you cannot find $\alpha_1$, $\alpha_2$ that satisfy $q_1(10) = 1$, $q_2(10) = 2$, then find $\alpha_i$ that minimize $(q_1(10) - 1)^2 + (q_2(10) - 2)^2$.) Be sure to be very clear about which alternative holds for each specification.

**Solution.** The dynamics of the system is given by $\dot{x} = Ax$ where (for $m_1 = m_2 = 1$ and $k_1 = k_2 = 1$)

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}.$$  

$x(10)$ is related to $x(0)$ through $x(10) = \Phi x(0)$ where $\Phi = e^{10A}$ is

$$\Phi = \begin{bmatrix} -0.3694 & 0.8431 & -0.2494 & 0.0515 \\ 0.8431 & 0.4737 & 0.0515 & -0.1979 \\ 0.5503 & -0.3009 & -0.3694 & 0.8431 \\ -0.3009 & 0.2494 & 0.8431 & 0.4737 \end{bmatrix}.$$
The values of $q_1(0)$ and $q_2(0)$ are taken as zero and therefore

$$x(10) = \begin{bmatrix} -0.2494 & 0.0515 \\ 0.0515 & -0.1979 \\ -0.3694 & 0.8431 \\ 0.8431 & 0.4737 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix},$$

in other words, the first two columns of $\Phi$ are irrelevant. Take

$$\Psi = \begin{bmatrix} -0.2494 & 0.0515 \\ 0.0515 & -0.1979 \\ -0.3694 & 0.8431 \\ 0.8431 & 0.4737 \end{bmatrix},$$

so that $x(10) = \Psi \alpha$ where $\alpha = [\alpha_1 \alpha_2]^T$. We define the matrix $\psi$ in the matlab environment:

```matlab
gamma = phi(:,3:4)
gamma =
-0.2494 0.0515
0.0515 -0.1979
-0.3694 0.8431
0.8431 0.4737
```

a) We have $q_2(10) = e_2^T x(10)$ where $e_2$ is the 2nd unit vector in $\mathbb{R}^4$. Therefore, $\alpha_1$ and $\alpha_2$ should satisfy the linear equation $2 = e_2^T \Psi \alpha$. There are two variables $\alpha_1$ and $\alpha_2$ but only one equation. Therefore the choice of $\alpha_1$ and $\alpha_2$ is not unique and we pick the minimum norm solution. In matlab:

```matlab
alpha = pinv(gamma(2,:))*2
alpha =
 2.4614
-9.4647
```

So here we can meet the spec, and even have one extra degree of freedom, which we use to minimize the norm of $\alpha$.

b) The requirements in this part are more stringent than in the previous one. Here we have the additional requirement $q_1(10) = 1$ and we get two linear equations in two unknowns, i.e., $[e_1^T e_2^T] x(10) = \Psi \alpha$ which has a unique solution since the resulting matrix is invertible. In matlab:

```matlab
alpha = psi(1:2,:)
alpha =
-6.4408
-11.7798
```

So here we can meet the spec, and even have one extra degree of freedom, which we use to minimize the norm of $\alpha$. 

10
In this case there is only one $\alpha$ that meets the specs; there is no extra freedom.

c) Here we have two more requirements $\dot{q}_1(10) = 0$ and $\dot{q}_2(10) = 0$ and therefore we get an overdetermined system of linear equations (four equations in two unknowns) $x(10) = \Psi \alpha$. We solve for $\alpha$ in a least-squares sense, which will show us if we are lucky and can find an exact solution. Using matlab:

```matlab
>> alpha=psi\[1;2;0;0]
alpha =
-0.1361
-0.3435
```

It can be checked that this $\alpha$ does not meet the spec, but it comes closest in the sense that $\|x(10) - [1 2 0 0]^T\|$ is minimized.


d) With the new set of parameters $m_1 = 1$, $m_2 = 1.3$, $k_1 = k_2 = 1$ we get a new system $\dot{x} = A x$ with

```matlab
>> tilde_A=[0 0 1 0;0 0 0 1;-2 1 0 0;1/1.3 -1/1.3 0 0]
tilde_A =
0 0 1.0000 0
0 0 0 1.0000
-2.0000 1.0000 0 0
0.7692 -0.7692 0 0
```

Similarly we define $\tilde{\Phi}$ and $\tilde{\Psi}$:

```matlab
>> tilde_phi=expm(10*tilde_A)
tilde_phi =
-0.6221 0.8272 -0.2229 -0.5395
0.6363 0.3961 -0.4150 -0.8869
0.0308 0.1921 -0.6221 0.8272
0.1478 0.2672 0.6363 0.3961
>> tilde_psi=tilde_phi(:,3:4)
tilde_psi =
-0.2229 -0.5395
-0.4150 -0.8869
-0.6221 0.8272
0.6363 0.3961
```

The requirements $q_2(10) = 2$ for $m_1 = m_2 = 1$, $k_1 = k_2 = 1$ and $q_2(10) = 2$ for $m_1 = 1$, $m_2 = 1.3$, $k_1 = k_2 = 1$ can be written as

$$ 2 = \epsilon_2^T \Psi \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}, \quad 2 = \epsilon_2^T \tilde{\Psi} \begin{bmatrix} \alpha_1 \\ \alpha_2/1.3 \end{bmatrix} = \epsilon_2^T \tilde{\Psi} \begin{bmatrix} 1 & 0 \\ 0 & 1/1.3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}, $$
or in matrix form

\[
\begin{bmatrix}
2 \\
2
\end{bmatrix} =
\begin{bmatrix}
e_2^T \tilde\Psi \\
e_2^T \Psi
\end{bmatrix}
\begin{bmatrix}
\frac{1}{2} \\
0 \\
0 \\
\frac{1}{1.3}
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\
\alpha_2
\end{bmatrix}.
\]

Using matlab:

```matlab
>> alpha=[psi(2,:);tilde_psi(2,:)*diag([1;1/1.3])];
alpha =
8.2608
-7.9566
>>
```

Here we can meet the spec, but there is only one solution.

### 5. Ranking teams in a round-robin tournament.

Suppose \( n \) teams play a round-robin tournament (that is, each pair of teams plays exactly one game). For concreteness consider a round-robin tournament with four teams and the following outcomes:

- team 1 beats teams 2 and 3 by 10 points each,
- team 1 loses to team 4 by 1 point,
- team 2 beats teams 3 and 4 by 1 point each, and
- team 3 beats team 4 by 1 point.

After the tournament we assign each team an overall score that we hope reflects how good the team is. A simple method for assigning these scores is to let the score for each team be the fraction of games that the team won. We let the vector of these scores be \( z(0) \in \mathbb{R}^n \). For the example given above, we have that:

\[
z_1(0) = \frac{2}{3}, \quad z_2(0) = \frac{2}{3}, \quad z_3(0) = \frac{1}{3}, \quad \text{and} \quad z_4(0) = \frac{1}{3}.
\]

These scores do not seem entirely satisfactory for our example because:

- teams 1 and 2 have the same ranking, even though team 1 seemed to dominate the tournament except for a close loss against team 4;
- teams 3 and 4 have the same ranking, even though team 4 managed to beat the dominant team 1, and only suffered slight defeats to teams 2 and 3.

We can try to improve our scores by accounting for the quality of the opponents that a team defeated, and the margin by which a team beat its opponents. Define the matrix \( A \in \mathbb{R}^{n \times n} \) such that

\[
A_{ij} = \begin{cases} 
\text{the margin by which team } i \text{ defeated team } j & \text{if team } i \text{ defeated team } j, \\
0 & \text{otherwise.}
\end{cases}
\]
For the simple example given above, we have that

\[ A = \begin{bmatrix}
0 & 10 & 10 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0
\end{bmatrix} \]

We refine our scores by letting \( z(1) = Az(0) \); the refined ranking of team \( i \) is

\[ z_i(1) = \sum_{j=1}^{n} A_{ij} z_j(0). \]

This says that the refined ranking of team \( i \) is a weighted combination of the margins of victory of team \( i \) with weights given by our simple estimates of the strengths of the teams that team \( i \) defeated. (Note that this is a very simple refinement; in particular, we have not accounted for team \( i \)’s losses.) Our refined estimates for the example above are

\[ z_1(1) = 10, \quad z_2(1) = \frac{2}{3}, \quad z_3(1) = \frac{1}{3}, \quad \text{and} \quad z_4(1) = \frac{2}{3}. \]

Note that these refined scores reflect the dominance of team 1, and the fact that team 4 has a reasonable case for being ranked above team 3. Now that we have more refined estimates of the strength of each team, we repeat the refinement process, and let \( z(2) = Az(1) \). We continue this process using the recursion \( z(t+1) = Az(t) \). In practice we only care about the relative scores of the teams, and the computation of \( z(t) \) may be numerically unstable for large values of \( t \). Therefore, we prefer to use the normalized scores

\[ x(t) = \frac{1}{\|z(t)\|} z(t), \quad t = 0, 1, 2, \ldots. \]

Note that \( x(t) \) satisfies the recursion

\[ x(t+1) = \frac{1}{\|Az(t)\|} Ax(t), \]

which is numerically stable, even for large values of \( t \). Now return to the general case of a round-robin tournament. Assume that \( A \) is diagonalizable, and that its eigenvalue decomposition is

\[ A = \sum_{i=1}^{n} \lambda_i v_i w_i^T, \]

where the eigenvalues are ordered such that \( |\lambda_1| \geq \cdots \geq |\lambda_n| \), and \( v_1, \ldots, v_n \) are normalized to be unit vectors.

a) Let \( \bar{x} = \lim_{t \to \infty} x(t) \). According to our usual analysis, what is the value of \( \bar{x} \)? State the usual assumptions that we use to justify our analysis.

b) Let \( e(t) = x(t) - \bar{x} \) be the difference between \( x(t) \) and its limiting value. The asymptotic convergence rate is defined to be

\[ \rho = \lim_{t \to \infty} \frac{\|e(t+1)\|}{\|e(t)\|}. \]

According to our usual analysis, what is the value of \( \rho \)? State the usual assumptions that we use to justify our analysis.
c) Now consider the specific data given in `nba_ranking_data.m`. This file defines the following variables.

- \( n \), the number of teams in the National Basketball Association (NBA)
- \( T \), the number of iterations of refinement to perform
- `teams`, a cell array of length \( n \) whose \( i \)th entry is the name of the \( i \)th NBA team
- `records`, an \( n \times 2 \) matrix whose \( i \)th row is the number of wins (first column) and losses (second column) of the \( i \)th NBA team in the 2014 – 2015 regular season
- `scores`, an \( n \times n \) matrix whose \((i,j)\) entry is the average margin of victory of team \( i \) over team \( j \) in the 2014 – 2015 NBA regular season (this is negative if team \( j \) outscored team \( i \))

Form the matrix \( A \) such that

\[
A_{ij} = \begin{cases} 
  \text{scores}(i,j) & \text{scores}(i,j) > 0, \\
  0 & \text{otherwise},
\end{cases}
\]

the vector \( z(0) \) such that

\[
z_j(0) = \text{fraction of its games that team } j \text{ won},
\]

and the vector \( x(0) = z(0)/\|z(0)\| \).

i. Compute \( x(t) \) for \( t = 1, \ldots, T \). Submit a plot showing \( x_j(t) \) for \( j = 10, 20, 30 \) as functions of \( t \) on a single set of axes. Compute \( \bar{x} \), and indicate \( \bar{x}_j \) for \( j = 10, 20, 30 \) on your plot.

ii. Compute \( e(t) \) for \( t = 0, \ldots, T \). Submit a plot showing \( \|e(t)\| \) as a function of \( t \); use a logarithmic scale for the vertical axis, and a linear scale for the horizontal scale. Report the value of \( \rho \). How is the slope of your plot related to \( \rho \)?

iii. Are the assumptions that we usually use to justify our analysis of the asymptotic behavior of \( x(t) \) satisfied? Does our usual conclusion hold?

iv. Are the assumptions that we usually use to justify our analysis of the asymptotic behavior of \( \|e(t+1)\|/\|e(t)\| \) satisfied? Does our usual conclusion hold?

v. Which team is most overrated if we use win/loss record instead of our refined scoring system? (In other words, which value of \( i \) maximizes \( x_i(0) - \bar{x}_i \)?)

vi. Which team is most underrated if use win/loss record instead of our refined scoring system? (In other words, which value of \( i \) maximizes \( \bar{x}_i - x_i(0) \)?)

**Solution.**

a) According to our usual analysis, we have that \( \bar{x} = \text{sign}(w_1^T x(0))v_1 \). Our usual analysis relies on the assumptions that \( \lambda_1 > |\lambda_i| \) for \( i = 2, \ldots, n \) (which implies that \( \lambda_1 \) is positive), and \( w_1^T x(0) \) is nonzero.

b) According to our usual analysis, we have that \( \rho = |\lambda_2/\lambda_1| \). Our analysis relies on the assumptions that \( \lambda_1 > |\lambda_2| > |\lambda_i| \) for \( i = 3, \ldots, n \), and \( w_1^T x(0), w_2^T x(0) \neq 0 \).
c) i. A plot showing $x_{10}(t)$, $x_{20}(t)$, and $x_{30}(t)$ as functions of $t$ is given in ??; the corresponding components of $\bar{x}$ are shown as dashed lines.

ii. A plot of $\|e(t)\|$ as a function of $t$ is given in ?? . We find that $\rho = 0.4177$. Note that $\log_{10}(\rho)$ is the asymptotic slope of the plot in ?? , which is shown as the red dashed line.

iii. We find that $A$ has a unique dominant eigenvalue, and the weight of $x(0)$ in the dominant mode is nonzero. Thus, our usual assumptions for the asymptotic analysis of $x(t)$ are satisfied. We see from ?? that $x(t) \to \bar{x}$.

iv. We find that $A$ does not have a unique eigenvalue with the second largest magnitude (a pair of complex-conjugate eigenvalues have the second largest magnitude). Thus, our usual assumptions for the asymptotic analysis of $\|e(t + 1)\|/\|e(t)\|$ are not satisfied. ?? shows that our usual analysis is still a good approximation because the asymptotic slope of $\|e(t)\|$ is very close to $\rho$. However, if we look closely at ??, we see that the slope actually oscillates around $\rho$; this is clearer in ?? , where the dashed red line shows $\rho$. Thus, our usual conclusion does not quite hold, although it is a good approximation.

v. The most overrated team using win/loss record instead of our refined analysis is the Milwaukee Bucks.

vi. The most underrated team using win/loss record instead of our refined analysis is the Golden State Warriors.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% clean up the workspace, and load the problem data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all ; close all ; clc
nba_ranking_data;
idx_plot = [10 20 30];
normalize = @(x) x / norm(x);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% construct A and x0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
z0 = records(:,1)./(records(:,1) + records(:,2));
x0 = normalize(z0);
A = max(scores,0);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% compute the eigenvalue decomposition of A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[eigvec , eigval] = eig(A);
[~, idx] = sort(abs(diag(eigval)) , 'descend');
eigvec = eigvec(:,idx);
eigval = diag(eigval(idx,idx));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% simulate x(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = [x0 nan(n,T)];
for t = 1:T
    x(:,t+1) = normalize(A * x(:,t));
end
xbar = eigvec(:,1);

figure();
plot(0:T , x(idx_plot,:));
hold on;
    plot(0:T , xbar(idx_plot) * ones(1,T+1) , '--');
hold off;
xlabel('t');
ylabel('x(t)');
legend_str = sprintfc('x%d(t)' , idx_plot);
legend(legend_str{:} , 'Location' , 'East');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% analyze the convergence of x(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rho = abs(eigval(2) / eigval(1))
e = x - xbar * ones(1,T+1);
norm_e = sqrt(sum(abs(e).^2));

figure();
semilogy(0:T , norm_e);
hold on;
    plot([0 T] , norm_e(end) * rho.^[0 -T] , 'r--');
hold off;
xlabel('t');
ylabel('||e(t)||');
grid on;

r = norm_e(2:(T+1)) ./(norm_e(1:T));
figure();
plot(1:T , r , [1 T] , rho*[1 1] , 'r--');
xlabel('t');
ylabel('||e(t+1)|| / ||e(t)||');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% check assumptions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
largest_eigenvalues = eigval(1:4)
largest_eigenvalues_magnitude = abs(eigval(1:4))
W = inv(eigvec)';
first_three_wTx0 = W(:,1:3)' * x0

% find overrated and underrated teams

[~, idx] = max(x0 - xbar);
fprintf('overrated: %s (%d)
', teams{idx}, idx);
[~, idx] = max(xbar - x0);
fprintf('underrated: %s (%d)
', teams{idx}, idx);

6. Convergence rate of power iteration. Consider the power iteration

\[ x(t + 1) = \frac{1}{\|Ax(t)\|} Ax(t), \]

where \( A \in \mathbb{R}^{n \times n} \) is given. We are interested in the limit

\[ \bar{x} = \lim_{t \to \infty} x(t), \]
Figure 4: the convergence error, $\|e(t)\|$
Figure 5: the convergence rate, $\|e(t+1)\|/\|e(t)\|$. 
and how quickly $x(t)$ converges to this limiting value. Suppose $A$ has an eigenvalue decomposition:

$$
\sum_{i=1}^{n} \lambda_i v_i w_i^T,
$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $A$, $v_1, \ldots, v_n$ are unit right eigenvectors of $A$, and $w_1, \ldots, w_n$ are left eigenvectors of $A$. We assume that the eigenvalues are ordered such that $|\lambda_1| \geq \cdots \geq |\lambda_n|$.

a) Under certain assumptions, we have that

$$
\bar{x} = \text{sign}(w_1^T x(0))^1 v_1.
$$

Prove this result, stating any assumptions that are needed for your analysis to hold.

b) Define the convergence error

$$e(t) = x(t) - \bar{x}.$$

Under certain assumptions, we have that

$$\lim_{t \to \infty} \frac{\|e(t+1)\|}{\|e(t)\|} = \left| \frac{\lambda_2}{\lambda_1} \right|.$$

This gives the asymptotic rate of convergence of the iteration. Prove this result, stating any assumptions that are needed for your analysis to hold.

c) The file `power_iteration_convergence_rate_data.m` defines the matrix $A$, the initial vector $x(0)$, and the time horizon $T$. Compute $x(t)$ for $t = 1, \ldots, T$. Make a plot showing all of the components of $x(t)$ as functions of time on a single set of axes. Also compute $\bar{x}$, and indicate its components on your plot.

d) Make a plot $\|e(t)\|$ as a function of time; use a logarithmic scale for the vertical axis, and a linear scale for the horizontal axis. What is the asymptotic slope of this plot?

**Solution.**

**First approach: simple solution**

a) The solution of the recursion is

$$x(t) = \frac{1}{\|A^t x(0)\|} A^t x(0).$$

Assuming $|\lambda_1| > |\lambda_i|$ for $i = 2, \ldots, n$, and $w_1^T x(0) \neq 0$, we have that

$$A^t x(0) \sim \lambda_1^t v_1 w_1^T x(0)$$
for large values of $t$. This implies that
\[
\lim_{t \to \infty} x(t) = \lim_{t \to \infty} \left\{ \frac{1}{\|A^T x(0)\|} A^T x(0) \right\} \\
= \lim_{t \to \infty} \left\{ \frac{1}{\|\lambda_1 v_1 w_1^T x(0)\|} \lambda_1 v_1 w_1^T x(0) \right\} \\
= \lim_{t \to \infty} \left\{ \text{sign}(\lambda_1)^t \text{sign}(w_1^T x(0)) v_1 \right\} \\
= \lim_{t \to \infty} \left\{ \text{sign}(\lambda_1)^t \right\} \text{sign}(w_1^T x(0)) v_1.
\]

In order for the limit to exist, we require that $\lambda_1 > 0$. Then, we have that
\[
\bar{x} = \lim_{t \to \infty} x(t) = \text{sign}(w_1^T x(0)) v_1.
\]

This analysis holds under the assumptions that

i. $\lambda_1 > |\lambda_i|$ for $i = 2, \ldots, n$ (note that this implies that $\lambda_1 > 0$), and

ii. $w_1^T x(0) \neq 0$.

b) Assuming $|\lambda_1| > |\lambda_2| > |\lambda_i|$ for $i = 3, \ldots, n$, $w_1^T x(0) \neq 0$, and $w_2^T x(0) \neq 0$, we have that
\[
A^T x(0) \sim \lambda_1^t v_1 w_1^T x(0) + \lambda_2^t v_2 w_2^T x(0) \quad \text{and} \quad \|A^T x(0)\| \sim \|\lambda_1^t v_1 w_1^T x(0)\|
\]
for large values of $t$. This implies that
\[
e(t) = x(t) - \bar{x} \\
= \frac{1}{\|A^T x(0)\|} A^T x(0) - \bar{x} \\
\sim \frac{1}{\|\lambda_1^t v_1 w_1^T x(0)\|} \left( \lambda_1^t v_1 w_1^T x(0) + \lambda_2^t v_2 w_2^T x(0) \right) - (\text{sign}(w_1^T x(0)) v_1) \\
= (\text{sign}(\lambda_1)^t - 1) \text{sign}(w_1^T x(0)) v_1 + \left( \frac{\lambda_2}{|\lambda_1|} \right)^t \frac{w_2^T x(0)}{|w_1^T x(0)|} v_2.
\]

If we assume (as above) that $\lambda_1 > 0$, then this expression simplifies to
\[
e(t) \sim \left( \frac{\lambda_2}{|\lambda_1|} \right)^t \frac{w_2^T x(0)}{|w_1^T x(0)|} v_2.
\]

Then, we have that
\[
\lim_{t \to \infty} \frac{\|e(t + 1)\|}{\|e(t)\|} = \lim_{t \to \infty} \left\{ \frac{\|\lambda_2/|\lambda_1|\|^{t+1}(w_2^T x(0)/w_1^T x(0)) v_2\|}{\|\lambda_2/|\lambda_1|\|^t(w_2^T x(0)/w_1^T x(0)) v_2\|} \right\} \\
= \lim_{t \to \infty} \left\{ \frac{\lambda_2}{|\lambda_1|} \right\} \\
= \left| \frac{\lambda_2}{\lambda_1} \right|.
\]

The assumptions needed for this analysis are

i. $\lambda_1 > |\lambda_2| > |\lambda_i|$ for $i = 3, \ldots, n$ (note that this implies that $\lambda_1 > 0$),

ii. $\lambda_1 > |\lambda_2| > |\lambda_i|$ for $i = 2, \ldots, n$.
ii. \( w_1^T x(0) \neq 0 \), and 

iii. \( w_2^T x(0) \neq 0 \).

c) A plot of the components of \( x(t) \) as functions of time is given in ??.

The components of \( \bar{x} = (0.88, 0.47, 0.06) \) are indicated by the dashed lines; note that each component of \( x(t) \) appears to be converging to the corresponding component of \( \bar{x} \).

d) A plot of \( \|e(t)\| \) as a function of time is given in ??.

The asymptotic slope of this plot is \( \log_{10}(\lambda_2/\lambda_1) = -0.2223 \).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% clean up the workplace, and load the problem data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all ; close all ; clc
power_iteration_convergence_rate_data;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% compute the eigenvalue decomposition of A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[eigvec , eigval] = eig(A);
[\~, idx] = sort(abs(diag(eigval)) , 'descend');
eigval = eigval(idx,idx);
eigvec = eigvec(:,idx);
xbar = eigvec(:,1)
rho = abs(eigval(2,2) / eigval(1,1))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% plot the trajectory of the power iteration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = [x0 nan(length(x0),T)];
for t = 1:T
    x(:,t+1) = (A * x(:,t)) / norm(A * x(:,t));
end

figure();
plot(0:T , x);
hold on;
    plot(0:T , xbar * ones(1,T+1) , '--');
hold off;
xlabel('t');
ylabel('x(t)');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% plot the error in the power iteration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

22
Alternative approach: alternate solution

a) The solution of the recursion is

\[
x(t) = \frac{1}{\|A^T x(0)\|} A^T x(0)
\]

\[
= \frac{1}{\|\sum_{i=1}^{n} \lambda_i v_i w_i^T x(0)\|} \sum_{i=1}^{n} \lambda_i v_i w_i^T x(0)
\]
Figure 7: the norm of the convergence error $e(t) = x(t) - \bar{x}$
\[
\begin{align*}
&= \frac{\lambda_1^2 w_1^T x(0)}{\| (\lambda_1^2 w_1^T x(0)) (v_1 + \sum_{i=1}^n (\lambda_i/\lambda_1)^t (w_i^T x(0)/w_1^T x(0)) v_i) \|
\begin{pmatrix}
1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_1} \right)^t \left( \frac{w_i^T x(0)}{w_1^T x(0)} \right) v_i
\end{pmatrix}
\end{align*}
\]

where we implicitly assume that \( \lambda_1 \) and \( w_1^T x(0) \) are nonzero when we divide by these quantities. Suppose \( A \) has a dominant positive eigenvalue — that is, \( \lambda_1 > |\lambda_i| \) for \( i = 1, \ldots, n \). Then, we have that \( (\lambda_i/\lambda_1)^t \to 0 \) as \( t \to \infty \) for \( i = 2, \ldots, n \). Moreover, we have that \( \text{sign}(\lambda_1) = 1 \) because \( \lambda_1 > 0 \). Therefore, we have that

\[
\bar{x} = \lim_{t \to \infty} x(t) = \text{sign}(w_1^T x(0)) v_1.
\]

In summary the assumptions used to justify this analysis are

\begin{itemize}
  \item[i.] \( \lambda_1 > |\lambda_i| \) for \( i = 2, \ldots, n \), and
  \item[ii.] \( w_1^T x(0) \neq 0 \).
\end{itemize}

b) Note that if \( \lambda_2 = 0 \), then \( e(t) = 0 \) for \( t > 0 \). Thus, the desired result holds when \( \lambda_2 = 0 \). Now consider the case when \( \lambda_2 \) is nonzero. The convergence error is

\[
e(t) = x(t) - \bar{x}
\]

\[
= \frac{1}{\| A^T x(0) \|} A^T x(0) - \text{sign}(w_1^T x(0)) v_1
\]

\[
= \frac{1}{\| \sum_{i=1}^n \lambda_i^2 v_i w_i^T x(0) \|} \left( \sum_{i=1}^n \lambda_i^2 v_i w_i^T x(0) \right) - \text{sign}(w_1^T x(0)) v_1
\]

\[
= \frac{\lambda_2^2 w_2^T x(0)}{\| (\lambda_2^2 w_2^T x(0)) (v_1 + \sum_{i=2}^n (\lambda_i/\lambda_2)^t (w_i^T x(0)/w_2^T x(0)) v_i) \|
\begin{pmatrix}
1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_2} \right)^t \left( \frac{w_i^T x(0)}{w_2^T x(0)} \right) v_i
\end{pmatrix}
\end{align*}
\]

\[
= \left( \frac{\lambda_2}{|\lambda_1|} \right)^t \frac{w_2^T x(0)}{w_1^T x(0)} \left( \frac{w_1^T x(0)}{w_2^T x(0)} v_1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_2} \right)^t \left( \frac{w_i^T x(0)}{w_2^T x(0)} \right) v_i \right)
\]

\[
= \left( \frac{\lambda_2}{|\lambda_1|} \right)^t \frac{w_2^T x(0)}{w_1^T x(0)} \| v_1 + \sum_{i=2}^n (\lambda_i/\lambda_2)^t (w_i^T x(0)/w_1^T x(0)) v_i \|
\begin{pmatrix}
1 - \text{sign}(\lambda_1)^t \left( v_1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_1} \right)^t \left( \frac{w_i^T x(0)}{w_1^T x(0)} \right) v_i \right) v_1
\end{pmatrix}
\]

\[
+ v_2 + \sum_{i=3}^n \left( \frac{\lambda_i}{\lambda_2} \right)^t \left( \frac{w_i^T x(0)}{w_2^T x(0)} \right) v_i
\end{pmatrix}
\]
where we implicitly assume that \( w_2^T x(0) \) is nonzero when we divide by this quantity. Using the assumption that \( \lambda_1 > 0 \) that we made above, we have that \( |\lambda_1| = \lambda_1 \) and \( \text{sign}(\lambda_1) = 1 \), and hence that

\[
e(t) = \left( \frac{\lambda_2}{\lambda_1} \right)^t \frac{w_2^T x(0)}{|w_1^T x(0)|} \left[ \frac{1}{\|v_1 + \sum_{i=2}^n (\lambda_i / \lambda_1)^t (w_i^T x(0) / w_1^T x(0)) v_i\|} \right] \left( \begin{array}{c} \left( \frac{\lambda_1}{\lambda_2} \right)^t \frac{w_1^T x(0)}{w_2^T x(0)} \left( 1 - \left\| v_1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_1} \right)^t \frac{w_i^T x(0)}{w_1^T x(0)} v_i \right\| v_1 \right) \right) \\
+ v_2 + \sum_{i=3}^n \left( \frac{\lambda_i}{\lambda_2} \right)^t \frac{w_i^T x(0)}{w_2^T x(0)} v_i \end{array} \right).\]

Applying this result, we find that

\[
\frac{||e(t+1)||}{||e(t)||} = \left( \frac{\lambda_2}{\lambda_1} \right)^{t+1} \left( \begin{array}{c} \left( \frac{\lambda_1}{\lambda_2} \right)^t \frac{w_1^T x(0)}{w_2^T x(0)} \left( 1 - \left\| v_1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_1} \right)^t \frac{w_i^T x(0)}{w_1^T x(0)} v_i \right\| v_1 \right) \right) \\
+ v_2 + \sum_{i=3}^n \left( \frac{\lambda_i}{\lambda_2} \right)^t \frac{w_i^T x(0)}{w_2^T x(0)} v_i \end{array} \right).\]
Consider the limit
\[
\lim_{t \to \infty} \left( \frac{\lambda_1}{\lambda_2} \right)^t \left( 1 - \left\| v_1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_1} \right)^t \left( \frac{w_i^T x(0)}{w_1^T x(0)} \right) v_i \right\| \right)
\]
\[
= \lim_{t \to \infty} \frac{1 - \left\| v_1 + \sum_{i=2}^n (\lambda_i/\lambda_1)^t (w_i^T x(0)/w_1^T x(0)) v_i \right\|}{(\lambda_2/\lambda_1)^t}
\]
Since \( \lambda_1 > |\lambda_i| \) for \( i = 2, \ldots, n \) (as we assumed above), and we assume that \( \|v_1\| = 1 \), both the numerator and denominator in the limit approach zero as \( t \to \infty \). Using l'Hôpital's rule, we find that
\[
\lim_{t \to \infty} \left( \frac{\lambda_1}{\lambda_2} \right)^t \left( 1 - \left\| v_1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_1} \right)^t \left( \frac{w_i^T x(0)}{w_1^T x(0)} \right) v_i \right\| \right)
= \lim_{t \to \infty} \left( \frac{\lambda_1}{\lambda_2} \right)^t \left( 1 - \left\| v_1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_1} \right)^t \left( \frac{w_i^T x(0)}{w_1^T x(0)} \right) v_i \right\| \right)
\]
\[
= \lim_{t \to \infty} \frac{t \left( v_1 + \sum_{i=2}^n (\lambda_i/\lambda_1)^t (w_i^T x(0)/w_1^T x(0)) v_i \right)}{(\lambda_2/\lambda_1)^t - \sum_{i=2}^n (\lambda_i/\lambda_1)^t (w_i^T x(0)/w_1^T x(0)) v_i) v_i}
\]
\[
= \lim_{t \to \infty} \frac{(v_1 + \sum_{i=2}^n (\lambda_i/\lambda_1)^t (w_i^T x(0)/w_1^T x(0)) v_i) v_i}{(\lambda_2/\lambda_1)^t - \sum_{i=2}^n (\lambda_i/\lambda_1)^t (w_i^T x(0)/w_1^T x(0)) v_i) v_i}
\]
If we assume that \( |\lambda_2| > |\lambda_i| \), then we have that \( (\lambda_i/\lambda_2)^t \to 0 \) as \( t \to \infty \) for \( i = 3, \ldots, n \).
Thus, we have that
\[
\lim_{t \to \infty} \left( \frac{\lambda_1}{\lambda_2} \right)^t \left( 1 - \left\| v_1 + \sum_{i=2}^n \left( \frac{\lambda_i}{\lambda_1} \right)^t \left( \frac{w_i^T x(0)}{w_1^T x(0)} \right) v_i \right\| \right) = \frac{(v_1^T v_2)(w_2^T x(0))}{w_1^T x(0)}.
\]
Applying this result, we find that
\[
\lim_{t \to \infty} \frac{\|e(t+1)\|}{\|e(t)\|} = \left| \frac{\lambda_2}{\lambda_1} \right| \frac{\|v_1\|}{\|v_1\|} \left| \frac{\left( \frac{v_1^T v_2}{w_1^T x(0)} \right) v_1 + v_2 \right|}{\left| \frac{v_1^T v_2}{w_1^T x(0)} \right| v_1 + v_2} = \frac{\lambda_2}{\lambda_1}.
\]
In summary the assumptions needed to justify this analysis are

i. \( \lambda_1 > |\lambda_2| > |\lambda_i| \) for \( i = 3, \ldots, n \), and

ii. \( w_1^T x(0), w_2^T x(0) \neq 0 \).

c) A plot of the components of \( x(t) \) as functions of time is given in ??.

The components of \( \bar{x} = (0.88, 0.47, 0.06) \) are indicated by the dashed lines; note that each component of \( x(t) \) appears to be converging to the corresponding component of \( \bar{x} \).

d) A plot of \( \|e(t)\| \) as a function of time is given in ??.

The dashed red line shows that the asymptotic slope of this plot is \( \log_{10}(\left| \lambda_2/\lambda_1 \right|) = -0.2223 \).
7. Optimal control for maximum asymptotic growth. We consider the controllable linear system
\[ x(t + 1) = Ax(t) + Bu(t), \quad x(0) = 0, \]
where \( x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m \). You can assume that \( A \) is diagonalizable, and that it has a single dominant eigenvalue (which here, means that there is one eigenvalue with largest magnitude). An input \( u(0), \ldots, u(T - 1) \) is applied over time period \( 0, 1, \ldots, T - 1; \) for \( t \geq T \), we have \( u(t) = 0 \). The input is subject to a total energy constraint:
\[ \|u(0)\|^2 + \cdots + \|u(T - 1)\|^2 \leq 1. \]
The goal is to choose the inputs \( u(0), \ldots, u(T - 1) \) that maximize the norm of the state for large \( t \). To be more precise, we’re searching for \( u(0), \ldots, u(T - 1) \) that satisfies the total energy constraint, and, for any other input sequence \( \tilde{u}(0), \ldots, \tilde{u}(T - 1) \) that satisfies the total energy constraint, satisfies \( \|x(t)\| \geq \|\tilde{x}(t)\| \) for \( t \) large enough. Explain how to do this. You can use any of the ideas from the class, e.g., eigenvector decomposition, SVD, pseudo-inverse, etc. Be sure to summarize your final description of how to solve the problem. Unless you have to, you should not use limits in your solution. For example you cannot explain how to make \( \|x(t)\| \) as large as possible (for a specific value of \( t \)), and then say, “Take the limit as \( t \to \infty \)” or “Now take \( t \) to be really large”.

Solution. We have
\[ x(T) = \sum_{\tau=0}^{T-1} A^{T-1-\tau} Bu(\tau) = C_T U, \]
where
\[ C_T = \begin{bmatrix} B & AB & \cdots & A^{T-1}B \end{bmatrix}, \quad U = \begin{bmatrix} u(T-1) \\ \vdots \\ u(0) \end{bmatrix}. \]
The vector \( U \) is our design variable; the energy constraint on the input is just \( \|U\| \leq 1 \). For \( t > T \), there is no input, so we have
\[ x(t) = A^{t-T} x(T) = A^{t-T} C_T U. \]
Now, what does the requirement that \( x(t) \) be as large as possible for large \( t \) mean? Let’s look at what \( A^k z \) looks like, for large \( k \). Using the hint given, we write the eigenvalue decomposition of \( A \) as
\[ A = T \Lambda T^{-1}, \]
where the columns of \( T \) are the eigenvectors of \( A \), and \( \Lambda \) is a diagonal matrix with entries \( \Lambda_{ii} = \lambda_i \). Therefore we have
\[ A^k = T \Lambda^k T^{-1}. \]
Let’s say that \( \lambda_1 \) is the dominant eigenvalue, i.e., \( |\lambda_1| > |\lambda_i| \) for \( i = 2, \ldots, n \). (This means that \( \lambda_1 \) is real; if it were complex, then its conjugate would be another eigenvalue with equal magnitude.) Now, for large \( k \) we have
\[ A^k z = T \Lambda^k T^{-1} z \approx \lambda_1^k v_1(w_1^T z), \]
where \( v_1 \) is the first column of \( T \) (i.e., the eigenvector associated with \( \lambda_1 \)) and \( w_1 \) is the first row of \( T^{-1} \) (i.e., a left eigenvector associated with \( \lambda_1 \)). Therefore

\[
\|A^k z\| \approx |\lambda_1^k| \|v_1\| |w_1^T z|,
\]

which shows that our goal is to choose \( z \) so that \( |w_1^T z| \) is as large as possible. Finally we can put it together. Our goal is to choose \( \|U\| \leq 1 \) that maximizes

\[
|w_1^T C T U| = \left| (C_T w_1)^T U \right|.
\]

This looks like a problem involving the SVD, but it's not: It's a problem involving the Cauchy-Schwartz inequality. The solution is

\[
U = \frac{1}{\|C_T w_1\|} C_T^T w_1
\]

(or the negative of this vector, which is also a solution). We should mention a common error. Many of you mentioned that \( U \) should be chosen such that \( z \) lies in the direction of the first eigenvector and then derived the least norm solution

\[
U_{ln} = C_T^T (C_T C_T^T)^{-1} v_1.
\]

For this to work, \( C_T \) needs to be fat and full rank. Some of you mentioned that if \( C_T \) were skinny one would use least squares to get “as close as you can”. Although this gives the maximum gain, \( C_T \) needs to be full rank.


a) Show that \( e^{A+B} = e^A e^B \) if \( A \) and \( B \) commute, i.e., \( AB = BA \).

b) Carefully show that \( \frac{d}{dt} e^{At} = A e^{At} = e^{At} A \).

Solution.

a) We will show that if \( A \) and \( B \) commute then \( e^{A} e^{B} = e^{A+B} \). We begin by writing the expressions for \( e^{A} \) and \( e^{B} \)

\[
e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots \]

\[
e^B = I + B + \frac{B^2}{2!} + \frac{B^3}{3!} + \cdots
\]

Now we multiply both expressions and get

\[
e^A e^B = I + A + B + AB + \frac{A^2}{2!} + \frac{B^2}{2!} + \frac{A^3}{3!} + \frac{A^2 B}{2!} + \frac{AB^2}{2!} + \frac{B^3}{3!} + \cdots
\]

\[
= I + A + B + \frac{A^2 + 2AB + B^2}{2!} + \frac{A^3 + 3A^2 B + 3AB^2 + B^3}{3!} + \cdots
\]

Now we note that, if \( A \) and \( B \) commute, we are able to write things such as \( (A + B)^2 = A^2 + 2AB + B^2 \). So, if \( A \) and \( B \) commute we can finally write

\[
e^A e^B = I + (A + B) + \frac{(A + B)^2}{2!} + \frac{(A + B)^3}{3!} + \cdots = e^{A+B}
\]
b) It suffices to note that $A$ commute with itself. Then one can write

\[
\frac{de^{At}}{dt} = A + A^2 t + \frac{A^3 t^2}{2!} + \cdots
\]

\[
= A(I + At + \frac{(At)^2}{2!} + \cdots)
\]

\[
= (I + At + \frac{(At)^2}{2!} + \cdots)A
\]

\[
= Ae^{At} = e^{At}A
\]