1. (25 points)

Apply the method explained in the Example of an Optimization-Oriented Analysis for Truss Design and Safety Analysis to create ... an Optimization-Oriented analysis of the 2D truss shown below.

Joints 1 and 7 are located at coordinates (0,0) and (20,0) meters. Node 1 is pinned in both the X and Y directions. Node 7 is on a “roller support” ... it can move freely horizontally and is bounded from moving in the vertical direction by a reaction force. Node 3 is is loaded in the downward vertical direction with a force $P_3$ and node 3 is is loaded in the downward vertical direction with a force $P_5$. The ten design variables in this problem are the $x$ and $y$ locations of nodes 2, 3, 4, 5, and 6: $(x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6)$.

(a) Complete steps 6 and 7 of section 1 of the example problem with pencil and paper.
(b) Derive the equilibrium matrix for this truss.
2. (15 points)

Apply the method explained in class to assess the balance of electrical power distribution in the power grid shown below, where the power generating stations are $A$ and $B$ and nodes 1 through 10 are power stations with specified demands, $D_1, \ldots, D_{10}$. Assume that the power generating stations have no demand of their own.

![Power Grid Diagram]

The power generating stations can produce only up to a maximum of $G_A$ and $G_B$ MW of power. ($G_A$ and $G_B$ are positive values.)

The power arriving at a station must be no less than the demand of the station.

Adopt the convenient convention of nomenclature that power that flows from station $i$ to station $j$ is called $P_{i,j}$, and assume that $P_{i,j}$ is positive if power flows from $i$ to $j$.

The set of 22 power flows within the branches of the network are the design variables. In your solution, put the variables in this order ...

$$
\mathbf{x} = \begin{bmatrix}
P_{A,1} & P_{A,2} & P_{A,9} & P_{A,10} & P_{B,4} & P_{B,5} & P_{B,6} & P_{B,7} & \ldots \\
P_{1,2} & P_{1,3} & P_{1,10} & P_{2,3} & P_{3,4} & P_{3,5} & P_{4,5} & P_{5,6} & P_{6,7} & P_{7,8} & P_{8,9} & P_{8,10} & P_{9,10}
\end{bmatrix}^T
$$

Power generating station $A$ produces power at $c_A$ $$/MW and that power generating station $B$ produces power at $c_B$ $$/MW. The cost of power generation is

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$$

where $\mathbf{c}$ and $\mathbf{x}$ are both column vectors.
(a) Write the 22-element vector \( \mathbf{c} \) in terms of \( c_A \) and \( c_B \).

(b) Write two inequalities that are satisfied if the power generating stations stay within their power generating capacities.

(c) Write ten inequalities that are satisfied if the power arriving at the power consuming stations generate no less than the power demanded by the stations.

(d) Express these twelve inequalities in a matrix form ...

\[ A \mathbf{x} \leq \mathbf{b} \]

by providing the 12-by-22 matrix \( A \) and the 12-element vector \( \mathbf{b} \). (The matrix \( A \) is populated with values of 0, 1, or -1. and the vector \( \mathbf{b} \) is populated with generation capacities and the demand quantities. Please use a ruler or graph paper to keep the rows and columns all lined up.)
3. (20 points)

Consider the water supply system described in the figure below.

Water enters the system through precipitation $Q_P$ over a watershed and leaves the system through a river $Q_R$, transpiration $Q_T$, evaporation $Q_E$, and the consumption in two communities $Q_{C1}$ and $Q_{C2}$. The units of these flows is million gallons per day (Mg/d).

As water flows through the system it is stored in surface water $V_s$, ground water $V_g$, reservoir water $V_r$, and treated water for consumption for the two communities, $V_1$ and $V_2$. Water stored is in units of millions of gallons (Mg).

Water flows from storage volume $i$ to storage volume $j$ with flow $Q_{i,j}$. For example, the flow of surface water into the reservoir is $Q_{s,r}$. 
Using a simplifying assumption that the flows are proportional to the connected volumes,

\[ Q_T = \alpha_{s,T} V_s \]
\[ Q_E = \alpha_{r,E} V_r \]
\[ Q_{s,g} = \alpha_{s,g} (V_s - V_g) \]
\[ Q_{s,r} = \alpha_{s,r} V_s \]
\[ Q_{g,r} = \alpha_{g,r} V_g \]
\[ Q_{r,1} = \alpha_{r,1} (V_r - V_1) \]
\[ Q_{r,2} = \alpha_{r,2} (V_r - V_2) \]

The coefficients \( \alpha_{i,j} \) are positive constants. The external (environmental) factors that drive the system are precipitation, river flow, and consumption.

The key modeling assumption here is that the rate of change of stored water is the difference between the inflow and the outflow. For example,

\[ \frac{dV_s}{dt} = Q_P - Q_T - Q_{s,r} - Q_{s,g} = -\alpha_{s,T} V_s - \alpha_{s,r} V_s - \alpha_{s,g} (V_s - V_g) + Q_P \]

Carrying out this analysis for the other four tanks, this system of five equations may be written in matrix form

\[
\frac{d}{dt} \begin{bmatrix} V_s \\ V_g \\ V_r \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix} \begin{bmatrix} V_s \\ V_g \\ V_r \\ V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} Q_P \\ Q_R \\ Q_{c1} \\ Q_{c2} \end{bmatrix}
\]

where \( A \) is a 5-by-5 matrix consisting of the \( \alpha_{i,j} \) coefficients and \( B \) is a 5-by-4 matrix consisting of values of 0, 1, or -1.

Systems of ordinary differential equations that can be represented by

\[ \dot{x}(t) = Ax(t) + Bu(t) \]

with constant matrices \( A \) and \( B \) are called linear time invariant (LTI) systems. The vector \( x(t) \) represents state of the system. In this system the state represents stored volumes of water. All the flows can be determined from the state \( x(t) \) and the inputs \( u(t) \).

(a) derive equations for \( \frac{d}{dt} V_g, \frac{d}{dt} V_r, \frac{d}{dt} V_1, \) and \( \frac{d}{dt} V_2 \), in terms of the \( \alpha \) coefficients, the stored volumes \( x \) and the environmental flows \( u \).

(b) from these four equations and the equation for \( \frac{d}{dt} V_s \), above, determine the matrices \( A \) and \( B \) in terms of the \( \alpha_{i,j} \) coefficients (in \( A \)) and values of 0,1, and -1 (in \( B \)).
4. (25 points)
Consider the exchange of insecticide ($X$) and insecticide metabolite ($M$) among pond water ($W$), pond plants ($P$), and fish ($F$ and $S$) shown below. Insecticide in the pond water $X_W$ is absorbed and released by plants $X_P$ and into fish vesicles $X_F$. Over time, insecticide in fish is stored in tissue $X_S$ and can return into fish vesicles $X_F$. Fish metabolize insecticide and eject metabolite into pond water $M_W$ which are then absorbed by and released by plants $M_P$.

The states in the system are:

- $X_W$: insecticide in pond water (g)
- $X_P$: insecticide in pond plants (g)
- $X_F$: insecticide in fish with fast uptake (g)
- $X_S$: insecticide in fish with slow uptake (g)
- $M_W$: metabolites in pond water (g)
- $M_P$: metabolites in pond plants (g)

How does this system behave?

- Will insecticide concentrate in fish tissue?
- Will it be fully metabolized?
- Will it just stay in the water, or be absorbed by plants?

Let's find out!
The diagram above indicates how insecticide and insecticide metabolite flow between the pond, fish, and plants. For example, referring to the diagram above, the evolution of insecticide in pond water \( X_W(t) \) is

\[
\frac{d}{dt}X_W(t) = -\alpha_{WF}X_W(t) - \alpha_{WP}X_W(t) + \alpha_{PW}X_P(t)
\]

and the evolution of insecticide in pond plants \( X_P(t) \) is

\[
\frac{d}{dt}X_P(t) = \alpha_{WP}X_W(t) - \alpha_{PW}X_P(t).
\]

(a) Derive four more equations like the two above for:

\[
\frac{d}{dt}X_F, \quad \frac{d}{dt}X_S, \quad \frac{d}{dt}M_W, \quad \frac{d}{dt}M_P.
\]

(b) Express the state derivative as a linear matrix equation \( \dot{x}(t) = Ax(t) \)

\[
\begin{bmatrix} X_W \\ X_P \\ X_F \\ X_S \\ M_W \\ M_P \end{bmatrix} = \begin{bmatrix} -(\alpha_{WF} + \alpha_{WP}) & \alpha_{PW} & 0 & 0 & 0 & 0 \\ \alpha_{WP} & -\alpha_{PW} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_W \\ X_P \\ X_F \\ X_S \\ M_W \\ M_P \end{bmatrix}
\]

- Are the diagonal elements of \( A \) all positive, all negative, or a mix of positive and negative numbers?
- Do the columns of \( A \) add up to zero?
- What does that mean?

(c) write a .m-script to:

i. Assign these values to the \( \alpha \) and \( \beta \) rate coefficients. These values indicate how quickly insecticide and metabolite move between “compartments.” Notice how some values are larger and some are smaller and that all values are positive numbers.

\begin{itemize}
  \item \( \alpha_{WP} = 0.388 \) (1/h) uptake of insecticide from water into plants
  \item \( \alpha_{PW} = 0.0515 \) (1/h) release of insecticide from plants into water
  \item \( \alpha_{WF} = 0.136 \) (1/h) uptake of insecticide from water into fish vesicles
  \item \( \alpha_{FS} = 0.067 \) (1/h) update of insecticide from vesicles into tissue
  \item \( \alpha_{SF} = 0.0254 \) (1/h) release of insecticide from tissue into vesicles
  \item \( \beta_{FW} = 0.0788 \) (1/h) excretion of metabolite from vesicles into water
  \item \( \beta_{WP} = 0.0068 \) (1/h) uptake of metabolite from water into plants
  \item \( \beta_{PW} = 0.001 \) (1/h) release of metabolite from plants into water
\end{itemize}

ii. Create the dynamics matrix \( A \) in terms of the rate coefficients,

iii. Create an initial state (column) vector with \( X_W = 100 \) and the other five states equal to zero. This indicates that at \( t = 0 \), there will be 100 grams of insecticide in the water and no insecticide or insecticide metabolite anywhere else.
iv. Look at the numbers in the dynamics matrix $A$. Can you tell which states change fastest, which states change slowest, which states end up at zero, and which states end up at some other value? No? That’s alright.

v. Solve the system of linear time invariant (LTI) ordinary differential equations

$$\dot{x}(t) = Ax(t), \quad x(0) = x_0$$

for the time response of each of the six states using the matrix exponential ...

$$x(t) = e^{At} x_0$$

... using the `matlab` language commands below ...

```matlab
x0 = ... % initial state vector
t = [ 0 : 1000 ]; % time values
n = length(t); % number of points in time
X = NaN(6,n); % initialize the matrix to save the state response
for k = 1:n
    X(:,k) = expm(A*t(k))*x0; % expm: matrix exponential
end
plot(t,X, 'LineWidth', 4) % plot it!
xlabel('time t hours')
ylabel('insecticide X and metabolite M, g')
legend('X_W', 'X_P', 'X_F', 'X_S', 'M_W', 'M_P', 'location','east')
```

vi. Show a plot for the full 1000 hours and a zoom in on the first 150 hours. Write a sentence or two about what each curve represents.

- Which states change quickly, which states change slowly?
- Do some states change quickly at first and then change more slowly later on?
- Which states end up close to zero, and which states do not?

Would you be interested in a way to figure all this out without first solving the LTI system for $x(t)$?

vii. Compute the eigenvectors $\vec{x}_i$ and the eigenvalues $\lambda_i$ of the dynamics matrix $A$, like this ...

```matlab
[eVec , eVal ] = eig(A)
```

The eigenvectors are the columns of `eVec`. Each eigenvector $\vec{x}_i$ is paired with an eigenvalue $\lambda_i$, so that $A \vec{x}_i = \vec{x}_i \lambda_i$.

Are the eigenvalues of $A$ all real and positive, all real and negative, a mix of positive and negative, or complex? (In this problem, if any are positive, you’ve made a mistake somewhere.)

- If all $\Re(\lambda_i) < 0$ all the states stabilize to zero.
- If any $\Re(\lambda_i) > 0$ some or all states are unstable.
- If any $\lambda_i = 0$ some or all states will stabilize to a non-zero value.

The eigenvalues indicate how particular combinations of states change exponentially in time. The $i$th eigenvector $\vec{x}_i$ indicates the particular combination of states associated with the $i$th eigenvalue. In this problem if $\vec{x}_{ij} > 0$ then state $x_i$ will decrease exponentially with $e^{\lambda_j t}$, and if $\vec{x}_{ij} < 0$ then state $x_i$ will
increase exponentially with $1 - e^{\lambda t}$. Note that, by convention, eigenvectors are unit-length vectors, $\bar{x}_i^T \bar{x}_i = 1$.

Considering each eigenvalue and its eigenvector,

- Which states participate most strongly with the fastest (largest) eigenvalue?
- Which states participate most strongly with the slowest (smallest) eigenvalue?
- Which states participate most strongly with the zero eigenvalue?
- Will insecticide in the Water, Plants, and Fish disappear?
- Will insecticide metabolite in the Water and Plants disappear?
- Do your plots from step (vi.) (for time going out to 1000 hours) confirm these answers?
- Do you feel like you’ve just seen a magic trick?

Welcome to the world of dynamic system analysis!
5. (0 points ... air-ball)

Watch this experiment!

A basketball of mass $m$ and radius $r$ moving through air of density $\rho$ is acted upon by drag force $f_D(t)$, lift force $f_L(t)$ and gravitational force $f_G$. The ball moves with horizontal and vertical velocity components $v_x(t)$ and $v_y(t)$ and has a backspin $\omega = \omega \hat{k}$.

In three-dimensional physical space, the three forces are

\[
\begin{align*}
  f_G &= -mg\hat{j} \\
  f_D &= -C_D \left(\pi r^2\right) \left(\frac{1}{2}\rho ||v|| v\right) \\
  f_L &= C_L \frac{16}{3} \pi^2 r^3 \rho (\omega \times v)
\end{align*}
\]

where $C_D$ and $C_L$ are empirical drag and lift coefficients and depend on details like the bumps of the surface of the basketball.

The acceleration of the ball is simply

\[
a = \left(\frac{1}{m}\right) (f_G + f_D + f_L)
\]

The state of the motion includes positions ($p_x(t)$ and $p_y(t)$) and velocities ($v_x(t)$ and $v_y(t)$). The forces (and accelerations) are not linear in positions and velocities. So this is a nonlinear dynamical system.

\[
\frac{d}{dt} \begin{bmatrix} p_x(t) \\ p_y(t) \\ v_x(t) \\ v_y(t) \end{bmatrix} = \mathcal{F} \begin{bmatrix} p_x(t) \\ p_y(t) \\ v_x(t) \\ v_y(t) \end{bmatrix}, \quad \boldsymbol{u}(t), \quad t, \quad \boldsymbol{c}
\]

Where $\boldsymbol{u}(t)$ is a set of external forces and $\boldsymbol{c}$ represents system constants, like gravitational acceleration. The vector $[p_x(t), p_y(t), v_x(t), v_y(t)]^T$ is the state vector of this system.

Through numerical simulation, one may investigate, and possibly even optimize, the trajectory of a basketball by adjusting values in the initial state vector ($\boldsymbol{x}_0$) and system constants, like backspin, the drag coefficient and the lift coefficient.
To compute the velocities and accelerations (state derivatives) from the positions and velocities (states), fill in the missing parts of the following system dynamics function `ball_flight_dyn.m`

```matlab
function [dxdt, y] = ball_flight_dyn(t, x, u, c)

vx = x(3);  % velocity in x-direction, m/s
vy = x(4);  % velocity in y-direction, m/s

g = c(1);    % gravitational acceleration, m/s^2
r = c(2);    % radius of size 7 basketball, m
m = c(3);    % mass of size 7 basketball, kg
rho = c(4);  % mass density of air, kg/m^3
omega = c(5); % backspin angular velocity, rad/s
CD = c(6);   % drag coefficient, unitless
CL = c(7);   % lift coefficient, unitless

j = [0; ?; ?];  % unit vector in y direction
v = [?; ?; 0];   % velocity 3-vector
w = [0; 0; ?];   % backspin vector

fG = ???;        % gravity force vector
fD = ???;        % drag force vector
fL = ???;        % lift force vector
f = fG + fD + fL; % net force vector

dxdt = [v; v; f(1)/m; f(2)/r];  % state derivative

y = [norm(fG); norm(fD); norm(fL)]; % individual force magnitudes
```

Simulate the flight of the ball through the air.

```matlab
g = 9.806;  % gravitational acceleration, m/s^2
r = 0.075;  % radius of size 7 basketball, m
m = 0.60;   % mass of size 7 basketball, kg
rho = 1.204; % mass density of air, kg/m^3
omega = pi*1.0*15; % backspin angular velocity, rad/s
CD = 0.5;   % drag coefficient of a spinning sphere, unitless
CL = 0.5;   % lift coefficient of a spinning sphere, unitless

% set of constants
c = {g, r, m, rho, omega, CD, CL};

T = 8;  % total time duration, s
dt = 0.01;  % time step increment, s
nT = floor(T/dt);  % number of time steps
t = [0:nT-1]*dt;  % time values, s
u = zeros(1, nT);  % external forcing – zero in this problem

% initial state vector
px = x(1);  % x-position, m
py = x(2);  % y-position, m

[t, x_sol, x_drv, y_sol] = ode45('ball_flight_dyn', t, x0, u, c);  % hpg m-files

figure(1)  % plot the trajectory
cif
hold on
plot(px, py)
```