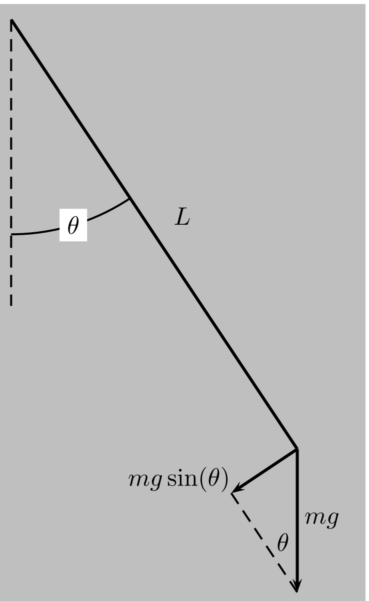
MNTC P01 - Week #9 - Differential Equations and Engineering

Pendulum



Consider the motion of a frictionless pendulum.

$$\begin{aligned} \text{Newton's Second Law: } mL^2\theta'' &= T_g \\ &= -mLg\sin(\theta) \\ \text{Solving for } \theta''\text{: } \theta'' &= -\frac{g}{L}\sin(\theta) \end{aligned}$$

1. **Without** simulating the actual motion of the pendulum, we can compute the period, T, using the formula below:

$$T = 4\sqrt{L/g}\int_0^{\pi/2} \frac{dx}{\sqrt{1 - k^2\sin^2 x}}$$

where $k = \sin\left(\frac{1}{2}\theta_0\right)$ and g is the acceleration due to gravity, 9.8 m/s.

For each set of values for L and θ_0 given below,

(a) Use the MATLAB `integral` function to find the period of the pendulum oscillations by evaluating the integral given above, and

(b) confirm the period by using `ode45` to simulate the motion of the pendulum for exactly that length of time, and plot a graph of the angular **velocity** against time. The velocity should just reach zero at the end of one cycle.

Do this with the following sets of L and θ_0 values.

(i) $L = 2$ m, $\theta_0 = 40^o$,

(ii) $L = 2.5$ m, $\theta_0 = 20^o$.

(iii) $L = 5.0$ m, $\theta_0 = 90^o$.

Link to the MATLAB code:
W09Pendulum1.m
pendulumDE.m

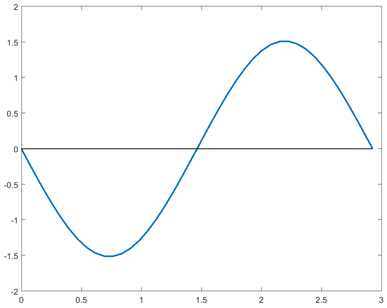
Note that we simply re-used the `pendulumDE.m` from the lectures, and set the friction coefficient $\mu = 0$.

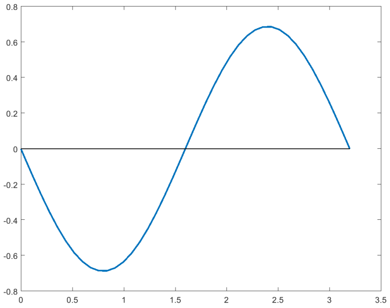
(i) $L = 2$ m, $\theta_0 = 40^o$: **T = 2.9274** seconds.

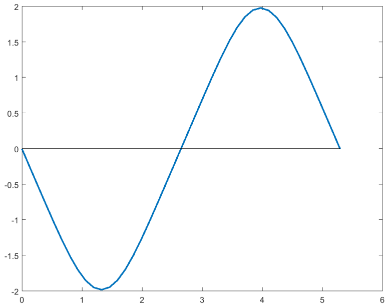
(ii) $L = 2.5$ m, $\theta_0 = 20^o$: **T = 3.1978** seconds.

(iii) $L = 5.0$ m, $\theta_0 = 90^o$: **T = 5.2974** seconds.

Plots:

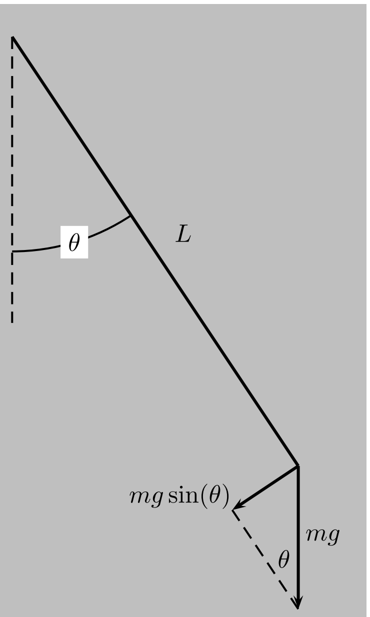






In all the graphs, we see that that the velocity returns to 0 at the end of the cycle, indicating that our integral calculation of the period matches the period from the `ode45` simulation.

2. Consider the motion of a pendulum, this time **with** friction.



$$\begin{aligned} &\text{Newton's Second Law:} \\ mL^2\theta'' &= T_g + T_f \\ &= -mLg\sin(\theta) - (\mu L^2 m)\theta' \end{aligned}$$

$$\text{Solving for } \theta''\text{: } \theta'' = -\frac{g}{L}\sin(\theta) - \mu\theta'$$

(a) Write a MATLAB function for the differential equation, and a script that will simulate the scenario for $L = 1.5$ m, $g = 9.8$ m/s^2, and $\mu = 0.2$. Use an initial condition of $\theta_0 = \frac{7\pi}{8}$, which is close to vertical.

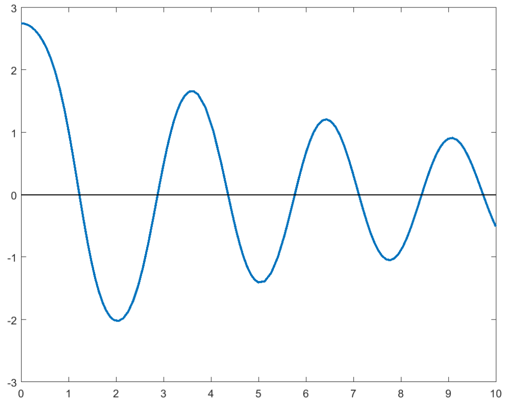
(b) Experiment with the initial **angular velocity** of the pendulum and find the smallest **positive** initial velocity that will result in the pendulum passing over the top of the axis of rotation. Find the value to the nearest 0.1 rad/s.

Have MATLAB generate a plot of the angle vs time graph for both the initial velocity that achieves this result, and for the initial velocity 0.1 rad/s smaller, which does *not* go 'over the top'.

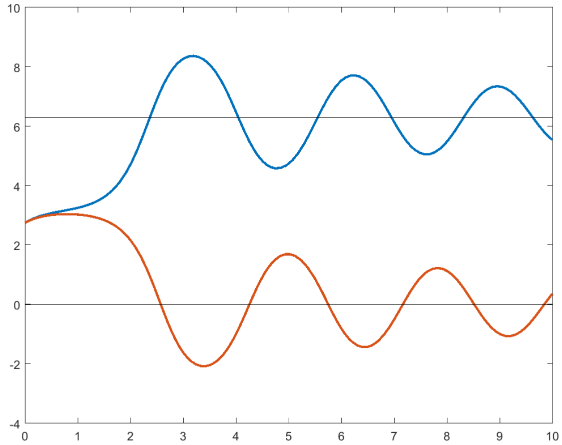
(c) Repeat the analysis in part (b), but this time using a **negative** initial velocity.

(a) Link to the MATLAB code:
W09Pendulum2.m
pendulumDE.m

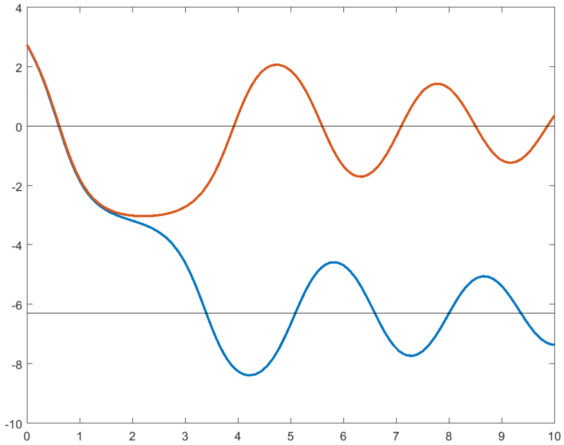
Here is the graph of the angle over time for the pendulum, when it has no initial velocity.



(b) With some experimentation, we find that an initial angular velocity of $\theta'(0) = 1.1$ rad/s will be enough to push the pendulum over the top of the axis of rotation. Comparing $\theta'(0) = 1.1$ and 1.0, we obtain the following graph of angle against time.



(c) With further experimentation, we find that using negative initial angular velocities requires a higher initial velocity compared to positive initial velocities, because friction eats away at the effect of that first push when we are going down first and then over the top. Still, a value of $\theta'(0) = -3.3$ rad/s will be enough to push the pendulum over the top of the axis. Comparing $\theta'(0) = -3.3$ and -3.2, we obtain the following graph of angle against time.



Single Tank Problems

3. An aquarium pool has volume 2×10^6 liters. The pool initially contains pure fresh water. At $t = 0$ minutes, water containing 10 grams/liter of salt is poured into the pool at a rate of 60 liters/minute. The salt water instantly mixes with the fresh water, and the excess mixture is drained out of the pool at the same rate (60 liters/minute).

   (a) Write a differential equation for $S(t)$, the mass of salt in the pool at time t.

   (b) Use MATLAB solve the differential equation to predict $S(t)$ over time.

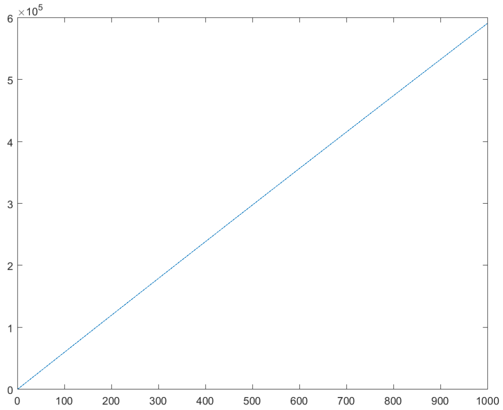
   (c) Based on the graph of the solution, what happens to $S(t)$ as $t \to \infty$?

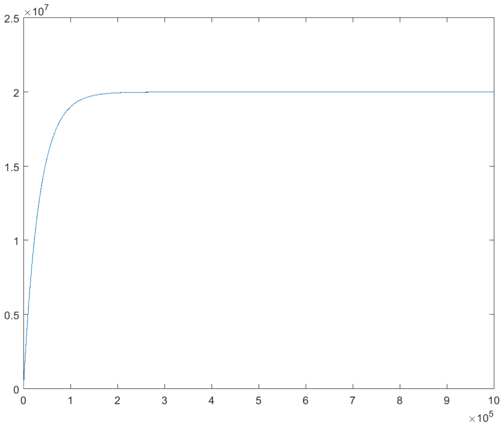
   (d) Find this same value using only the information about the volume and the concentration of the incoming salt solution.

(a)

$$\begin{aligned}
\text{Rate of change of salt amount (g/min)} &= \text{ Rate in } - \text{ Rate out}\\
\text{Rate in (g/min)} &= \text{Flow rate} \times \text{Concentration}\\
&= (60 \text{ liters/min}) \times (10 \text{ g/liter}) = 600 \text{ g/min}\\
\text{Rate out (g/min)} &= \text{Flow rate} \times \text{Concentration}\\
&= \text{Flow rate} \times \text{amount (g) / Pool volume (liters)}\\
&= (60 \text{ liters/min})(S(t) \text{ grams})/(2 \times 10^6 \text{ liters})\\
&= (3 \times 10^{-5})S(t)\\
\text{Finally, we get our DE:} \qquad \frac{dS}{dt} &= 600 - (3 \times 10^{-5})S
\end{aligned}$$

(b) Note: to see anything interesting in this simulation, you have to simulate for a **long** simulation time, i.e. a long `tspan`. Here are two graphs of the simulation results, one with 1000 minutes, and then with 1 million minutes.





The reason a long simulation time is needed is because adding 60 L/min to a 2 million liter pool takes a long time to have an effect.

(c) As $t \to \infty$, we see the graph of $S(t)$ plateau at $S \to 2 \times 10^7$ grams.

(d) We expect that the salt in the aquarium will tend to the same **concentration** as the incoming water, as all of the original water is replaced with the new inflow solution. At a concentration of 10 g/liter, in a volume of 2×10^6 liters, we expect to see eventually $S = C \times V = (10)(2 \times 10^6) = 2 \times 10^7$ grams of salt in the aquarium, which matches our graphical results.

4. A 150 liter tank initially contains 60 liters of water with 0.5 kgs of salt dissolved in it. Water enters the tank at a rate of 0.9 liters/hr and the water entering the tank has a salt concentration of $\frac{1}{5}(1 + \cos(t))$ kgs/liter.

   (a) Draw a diagram of the inflow and outflow for this scenario.

   (b) Build a formula for the volume of water in the tank over time.

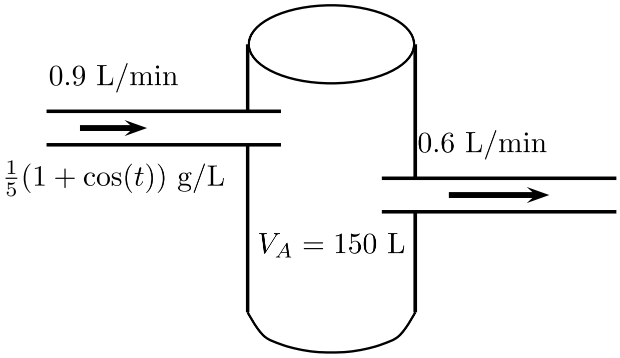
   (c) Find out how long it will be until the tank overflows.

   (d) Write a differential equation that describes the rate of change of the **amount of salt** in the tank.

   (e) Use MATLAB to generate a graph of the amount of salt in the tank over time, up until the tank overflows.

   (f) How much salt is in the tank when it overflows?

(a) Here is a diagram of the system.



(b) Since the tank has a volume of 150 L, is gaining 0.9 - 0.6 = 0.3 L/hour, and starts at 60 L, we obtain the volume expression $V(t) = 60 + 0.3t$.

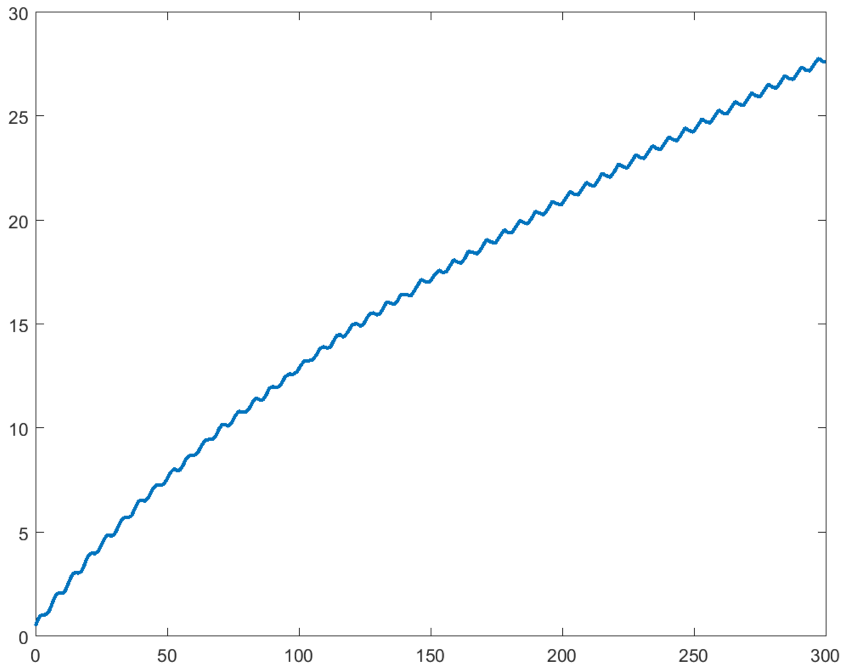
(c) Solving $V(t) = 150$ for t, gives us $60 + 0.3t = 150$, or $t = 300$ hours until the tank overflows.

(d) The differential equation will be the same "rate in - rate out" form.

$$\text{Rate of change of salt amount (kg/hr)} = \text{ Rate in } - \text{ Rate out}$$

$$\begin{aligned}
\text{Rate in (kg/hr)} &= \text{Flow rate} \times \text{Concentration} \\
&= (0.9 \text{ liters/hr}) \times (\frac{1}{5}(1+\cos(t)) \text{ kg/liter}) \\
&= 0.18(1+\cos(t))\text{kg/hr} \\
\text{Rate out (kg/hr)} &= \text{Flow rate} \times \text{Concentration} \\
&= \text{Flow rate} \times \text{amount (g) / Pool volume (liters)} \\
&= (0.6 \text{ liters/hr})(S(t) \text{ kg})/(60+0.3t) \text{ liters)} \\
\text{Finally, we get our DE:} \qquad \frac{dS}{dt} &= 0.18(1+\cos(t)) - \frac{0.6}{60+0.3t}S
\end{aligned}$$

(e) Here is the graph of the predicted amount of salt in the tank over time.



Note that the $\cos(t)$ effect has a short period ($2\pi \approx 6$ hours) relative to the 300 hours of the simulation time, which is why the graph looks like it has the high-frequency oscillations in it.

(f) By either zooming in, or typing `S` at the MATLAB command line to show all the `S` values coming out of the simulation and grabbing the last one, we find at the end of 300 hours that there is $S(300) \approx 27.6212$ kg of salt in the tank.

Other First Order Models

5. Differential equations are not only well-suited for physics applications: they are are also widely used in biology, particularly in population models.

Consider the fish population model below, based on a standard limited-resource population growth, minus a function of harvesting.

$$\frac{dP}{dt} = \underbrace{[(10 - P) \cdot P]}_{\text{natural population growth rate}} - \underbrace{h(t)}_{\text{harvesting rate}}$$

where

- P = population of fish (in thousands), and
- $\frac{dP}{dt}$ = rate of population change, in thousands per year
- $h(t)$ is the harvesting rate (in thousands of fish per year)

We want to study the impact of two harvesting models:

- $h_1 = k_1$; constant harvesting
- $h_2(t) = k_2(\sin(\pi t) + 1)$; seasonal model where the harvesting has a yearly cycle.

(a) Generate a prediction of the population over time, starting at initial populations of $P(0) = 15$ for each model. Use $k_1 = k_2 = 5$. Produce a graph showing the predicted population over time on the same graph, over a long enough time interval to show the long-term behaviour of both solutions.

One question that arises in such harvesting models is which fishing strategy permits a higher average harvesting rate can be maintained: seasonal harvesting, or constant harvesting? To decide this, we note that the average harvest rate for h_1 is k_1, and for h_2 is k_2, so whichever value of k_1 and k_2 is larger indicates the strategy with the greater average harvesting rate.

We will define the *maximum sustainable harvest rate* for both models as the *highest harvest rate for which the population is not driven to zero.*

(b) Find and report the maximum sustainable harvest level k_1 for the constant harvesting model (to the nearest integer). (Use trial and error if necessary, though more insightful DE-related ways are possible.) Indicate how you found the cut-off level.

NOTE: during this process, your model will predict a population of zero, which will then lead to large negative populations. This clearly makes no sense, so limit your plots with the command `ylim([0, P0])`. *This same problem will also trigger warnings in ode45 about error tolerances; you can safely ignore those warnings.*

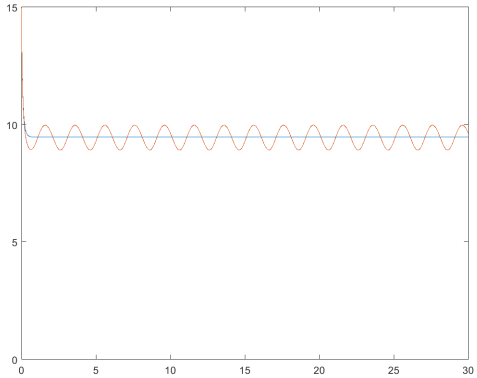
(c) Generate a plot showing the population over time, using the same initial value used earlier, but using both the k_1 value just above, and just below the extinction level. (One line should remain positive, while the other should crash to zero at some point on the graph.)

(d) Use trial and error (theory isn't much help here) to find the maximum sustainable harvest level k_2 for the cyclic harvesting model (to the nearest integer). Include a plot showing the population over time with this harvesting level.

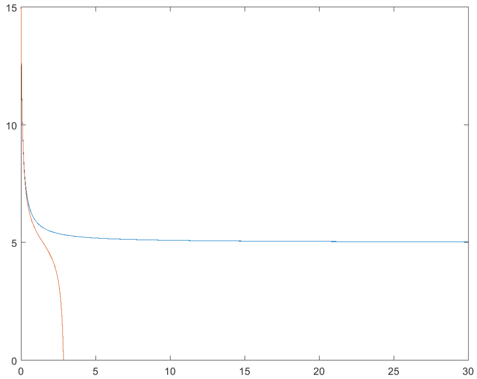
(e) Based on your experiments, can constant harvesting or cyclic harvesting sustain a greater average harvest in the long run? Explain your reasoning.

(a) The code in the link below generates the basic graph of the populations over time. It can be adapted to help answer the later sections.

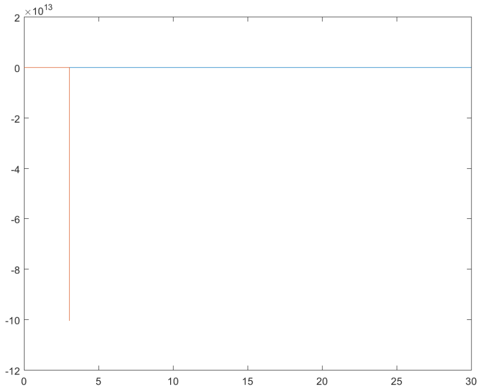
Link to the MATLAB code:
W09PopulationModel1.m



(b) Through trial and error, we find that there is a transition in behaviour between $k_1 = 25$ and $k_1 = 26$.

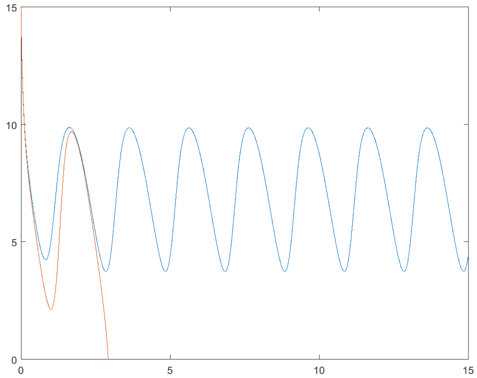


Notice that the `ylim([0, P0])` line in the script is very important, because once the population P drops below zero, the differential equation solution will keep going down forever. Here's the graph *without* the `ylim` command:



Notice the vertical scales shows 10^{13}, indicating that the negative values predicted by the model are huge, and much larger than the initial population of $P_0 = 15$. However, all those negative values are essentially meaningless for populations, so using the `ylim` command to focus the attention on the relevant population range is a reasonable tactic to make the graphs useful.

(c) In the seasonal harvest scenario, we experiment with values of k_2, and find that between $k_2 = 16$ and $k_2 = 17$, we see our seasonal pattern stop repeating and start reaching extinction:



(d) Based on these experiments, it seems that seasonal harvesting leads to extinction at lower average harvesting levels, because a lower average rate of harvest ($k_2 = 17$ thousand fish per year) leads to extinction, compared to the constant harvest case (where $k_1 = 26$ thousand fish per year can be harvested).

Multi-Tank Systems

6. Consider two interconnected tanks. Tank A initially contains 100 L of water and 200 g of salt, and tank B initially contains 200 L of water and 900 g of salt. The liquid inside each tank is kept well stirred.

   - Liquid flows from tank A to tank B at a rate of $3 \text{ L} \cdot \text{min}^{-1}$;
   - liquid flows from tank B to tank A at rate of $2 \text{ L} \cdot \text{min}^{-1}$.
   - A salt brine with concentration $7 \text{ g} \cdot \text{L}^{-1}$ of salt flows into tank A at a rate of $5 \text{ L} \cdot \text{min}^{-1}$; an outflow pipe drains out of the Tank A at $4 \text{ L} \cdot \text{min}^{-1}$.
   - Moreover, a salt brine with concentration $3 \text{ g} \cdot \text{L}^{-1}$ of salt flows into tank B at a rate of $7 \text{ L} \cdot \text{min}^{-1}$; the solution is drained out at $8 \text{ L} \cdot \text{min}^{-1}$.

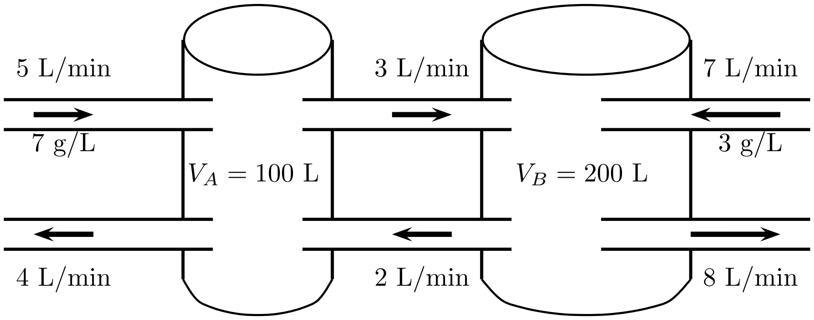
   (a) Draw a diagram for the flows and concentrations in this scenario.

   (b) Write a set of differential equations for the **amount** of salt in each tank.

   (c) Use MATLAB to simulate the concentration of salt in each tank over time. Generate two separate graphs, one for Tank A and one for Tank B, and use the `title` command to label each one.

   (d) Use the solution generated by MATLAB to estimate when Tank B reaches its lowest salt amount, and what that lowest amount is.

(a) Here is a diagram with all the information added.



(b) Let $S_A(t)$ and $S_B(t)$ denote the amount (in grams) of salt in tanks A and B respectively at time t (in minutes). It follows that

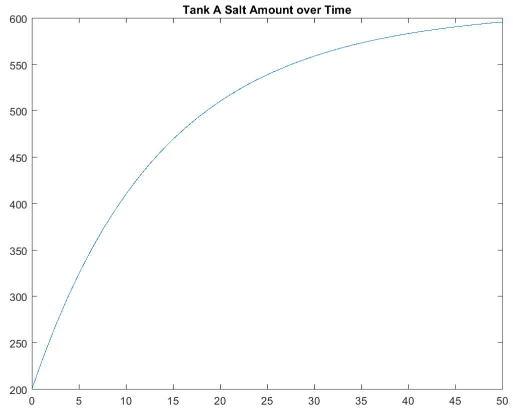
$$\begin{aligned}
\frac{dS_A}{dt} &= \text{input rate} - \text{output rate} = \left(7\frac{\mathrm{g}}{\mathrm{L}}\right)\left(5\frac{\mathrm{L}}{\mathrm{min}}\right) + \left(\frac{S_B\ \mathrm{g}}{200\ \mathrm{L}}\right)\left(2\tfrac{\mathrm{L}}{\mathrm{min}}\right) - \left(\frac{S_A\ \mathrm{g}}{100\ \mathrm{L}}\right)\left(4+3\frac{\mathrm{L}}{\mathrm{min}}\right)\\
&= -\frac{7}{100}S_A + \frac{2}{200}S_B + 35\\
\frac{dS_B}{dt} &= \left(3\frac{\mathrm{g}}{\mathrm{L}}\right)\left(7\frac{\mathrm{L}}{\mathrm{min}}\right) + \left(\frac{S_A\ \mathrm{g}}{100\ \mathrm{L}}\right)\left(3\frac{\mathrm{L}}{\mathrm{min}}\right) - \left(\frac{S_B\ \mathrm{g}}{200\ \mathrm{L}}\right)\left(2+8\tfrac{\mathrm{L}}{\mathrm{min}}\right)\\
&= \frac{3}{100}S_A - \frac{10}{200}S_B + 21\,,
\end{aligned}$$

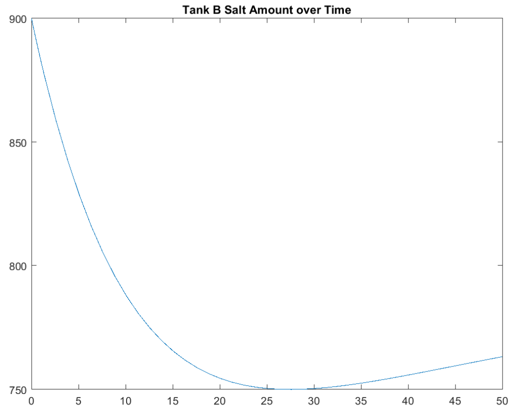
The coefficients in these expressions could be simplified (e.g. $\frac{10}{200} = \frac{1}{20} = 0.05$) but keeping the origins of each element of the coefficient is sometimes helpful for identifying any errors. E.g. $\frac{2}{200}S_B$ comes from a flow rate of 2 L/min, a volume of 200 L in Tank B, and the amount of salt S_B.

(c) To implement this system of equations in MATLAB, we define a vector $\vec{w} = [S_A, S_B]$, and then write the differential equation system in a separate MATLAB function file, here called
tankSystem1.m

The main script that sets up the initial conditions and then uses `ode45` to solve the equations is in W09TankSystem1.m

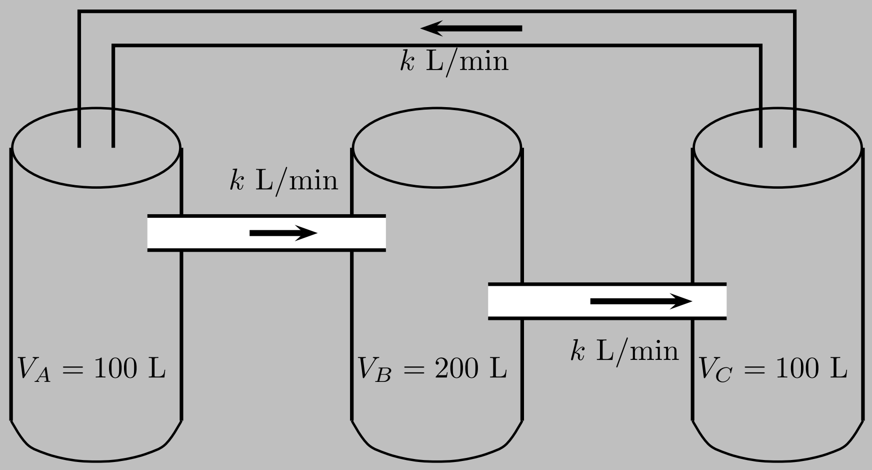
The solution produces the two simulated salt concentrations shown below.





(d) Tank B reaches its minimum salt amount at approximately $t = 27 - 28$ minutes, and has a level of 750 g at that moment.

7. Consider the 3-tank system shown below.



(a) Write the set of differential equations that governs the **amount** of salt in each tank, S_A, S_B and S_C.

(b) Convert your answer from part (a) into a set of differential equations for the **concentrations** in each tank, C_A, C_B and C_C.

(c) Using the first line `function dw_dt = tankSystem2(t, w, k, VA, VB, VC)` and the definition that $\vec{w} = [C_A, C_B, C_C]$ to group the three dependent variables, write a MATLAB function file that implements the differential equation system from part (b).

(d) Write a script that simulates the changes in concentration over time, using the volumes shown in the diagram, a flow rate of $k = 2$ L/min for each connection, and a time span of 250 minutes.

(e) Use your knowledge of chemistry to explain the fact that all the tanks converge to a common concentration of 2.5 g/L.

(a) The differential equations are all based on the "net rate = rate in - rate out". The new feature is that **every** concentration will of the form $\dfrac{S}{V}$.

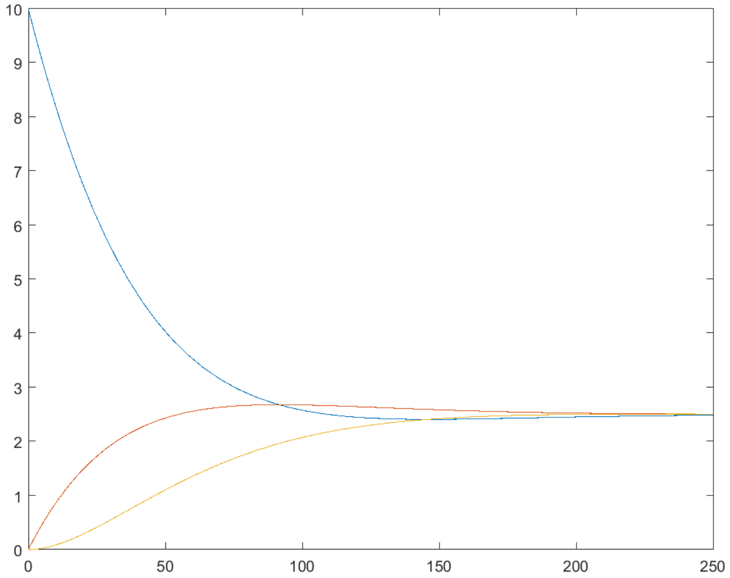
$$\begin{aligned}
\frac{dS_A}{dt}\left(\frac{\text{g}}{\text{min}}\right) &= \text{input rate} - \text{output rate} = \left(k\frac{\text{L}}{\text{min}}\right)\left(\frac{S_C}{V_C}\frac{\text{g}}{\text{L}}\right) - \left(k\frac{\text{L}}{\text{min}}\right)\left(\frac{S_A}{V_A}\frac{\text{g}}{\text{L}}\right)\\
&= k\frac{S_C}{V_C} - k\frac{S_A}{V_A}\\
\frac{dS_B}{dt}\left(\frac{\text{g}}{\text{min}}\right) &= \text{input rate} - \text{output rate} = \left(k\frac{\text{L}}{\text{min}}\right)\left(\frac{S_A}{V_A}\frac{\text{g}}{\text{L}}\right) - \left(k\frac{\text{L}}{\text{min}}\right)\left(\frac{S_B}{V_B}\frac{\text{g}}{\text{L}}\right)\\
&= k\frac{S_A}{V_A} - k\frac{S_B}{V_B}\\
\frac{dS_C}{dt}\left(\frac{\text{g}}{\text{min}}\right) &= \text{input rate} - \text{output rate} = \left(k\frac{\text{L}}{\text{min}}\right)\left(\frac{S_B}{V_B}\frac{\text{g}}{\text{L}}\right) - \left(k\frac{\text{L}}{\text{min}}\right)\left(\frac{S_C}{V_C}\frac{\text{g}}{\text{L}}\right)\\
&= k\frac{S_B}{V_B} - k\frac{S_C}{V_C}
\end{aligned}$$

(b) Converting each of these differential equations to concentrations requires building off the two relationships $C = \dfrac{S}{V}$ and $\dfrac{dC}{dt} = \dfrac{1}{V}\dfrac{dS}{dt}$.

$$\begin{aligned}
\frac{dC_A}{dt} &= \frac{1}{V_A}\frac{dS_A}{dt} = \frac{k}{V_A}C_C - \frac{k}{V_A}C_A\\
\frac{dC_B}{dt} &= \frac{1}{V_B}\frac{dS_B}{dt} = \frac{k}{V_B}C_A - \frac{k}{V_B}C_B\\
\frac{dC_C}{dt} &= \frac{1}{V_C}\frac{dS_C}{dt} = \frac{k}{V_C}C_B - \frac{k}{V_C}C_C
\end{aligned}$$

(c) You can find the MATLAB function for this simulation at tankSystem2.m.

(d) You can find the MATLAB script that runs the simulation at W09TankSystem2.m.
Here is the graph of the simulated concentration of salt in the 3 tanks.



(e) With the continuous mixing, eventually all three tanks will reach the same concentration. We can predict that concentration by noting that the only salt is the salt in Tank A at the start. That amount of salt is $S_A(0) = C_A(0)V_A = (10 \text{ g/L})(100 \text{ L}) = 1000$ g

After all the mixing the 1000 g of salt will be evenly distributed over the whole volume which is $V_A + V_B + V_C = 100 + 200 + 100 = 400$ L, so a final concentration of $\dfrac{1000 \text{ g}}{400 \text{ L}} = 2.5$ g/L in every tank.