Application of Systems of Ordinary Differential Equations to Clean Contaminated Containers

Raphael Alexander Lesmana (2540128114)
Yennifer Wilanata (2540123725)
Mario Iskandar (2502014841)

January 5, 2024

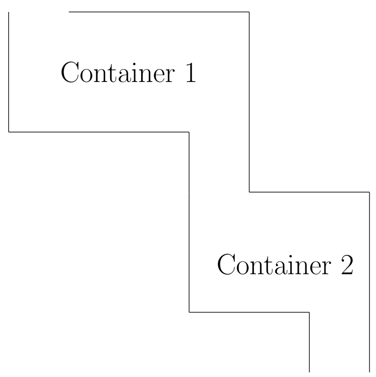
1 Introduction

A cascade system is a dynamic system where each component is connected serially, such that the output of one subsystem becomes the input of the next subsystem.

An example of this cascade system is the movement of water from one container to another. Not only water, cascade system can also be used to control air. It aims for efficient filling of container.

Ordinary Differential Equations (ODE) can be used to model and calculate the movement or flow of water. Cascading system can be modelled using ODE where the rate of entry of a component is dependent to the rate of drain of another component.

2 Mathematical modelling



Let $s_1(t), s_2(t)$ be the pollutant content in containers 1 and 2 respectively. The parameters of this system are V_1, V_2: the total capacity of containers 1 and 2 repsectively, and r, the rate of fluid flow. All parameters are real, positive, nonzero numbers.

To clean the two interconnected containers, water is fed in through container 1's opening at a rate of r. We assumed uniform mixing in the containers and that the volume of the connectors are negligible.

The rate of pollutant change in a container is equal to the volume of pollutant entering the container and the volume of pollutant leaving the container at any given time. Because only clean water is being pumped into container 1, no new pollutant enters container 1, while pollutants that were already in container 1 leave the container.

The water from container 1 enters container 2 at a rate of r, carrying with it some of container 1's pollutants. At the same time, container 2 is drained at a rate of r as well.

$$s_1'(t) = -\frac{r}{V_1} s_1(t) \tag{1}$$
$$s_2'(t) = \frac{r}{V_1} s_1(t) - \frac{r}{V_2} s_2(t) \tag{2}$$

This is the system of first-order ODEs that describes the behaviour of pollutant in our containers.

3 Analytic solution

We opted to use the eigenvalues method so that we can characterize some of the possible behaviors of the system under different parameters. To do that, we translate the system into a matrix equation:

$$\begin{bmatrix} s_1'(t) \\ s_2'(t) \end{bmatrix} = \begin{bmatrix} -\frac{r}{V_1} & 0 \\ \frac{r}{V_1} & -\frac{r}{V_2} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \tag{3}$$

Which can be succintly written as

$$\mathbf{s}' = A\mathbf{s}$$

We solve for the eigenvalues of A:

$$\begin{aligned} &|A - \lambda I| = 0 \\ \Leftrightarrow &\begin{vmatrix} -\frac{r}{V_1} - \lambda & 0 \\ \frac{r}{V_1} & -\frac{r}{V_2} - \lambda \end{vmatrix} = 0 \\ \Leftrightarrow &\lambda^2 + r\left(\frac{1}{V_1} + \frac{1}{V_2}\right)\lambda + \frac{r^2}{V_1 V_2} = 0 \end{aligned}$$

This quadratic equation has three possible outcomes, which can be determined by looking at its discriminant:

$$\begin{aligned} &D = b^2 - 4ac \\ \Leftrightarrow &D = \frac{r^2(V_1 - V_2)^2}{V_1^2 V_2^2} \end{aligned}$$

In the case that $D < 0$, we will have two conjugate eigenvalues. However, we can see that this is an impossible scenario, as both V_1 and V_2 are positive real numbers, and the squared difference of two positive real numbers is always positive. Neither is it the case that r is negative.

$$\begin{aligned}&\frac{r^2(V_1 - V_2)^2}{V_1^2 V_2^2} < 0\\ \Leftrightarrow &(V_1 - V_2)^2 < 0\end{aligned}$$

In the case that $D = 0$, we will have a repeating eigenvalue. This happens when the two containers have the same volume.

$$\begin{aligned}&\frac{r^2(V_1 - V_2)^2}{V_1^2 V_2^2} = 0\\ \Leftrightarrow &(V_1 - V_2)^2 = 0\end{aligned}$$

In the case that $D > 0$, we will have two distinct and real eigenvalues. This happens when the two containers are of unequal volumes.

$$\begin{aligned}&\frac{r^2(V_1 - V_2)^2}{V_1^2 V_2^2} = 0\\ \Leftrightarrow &(V_1 - V_2)^2 > 0\end{aligned}$$

We will only be solving the for third case. We use the quadratic formula to obtain the two distinct and real eigenvalues:

$$\begin{aligned}\lambda_{1,2} &= \frac{-b \pm \sqrt{D}}{2a}\\ \Leftrightarrow \lambda_{1,2} &= \frac{1}{2}\left[-r\left(\frac{1}{V_1} + \frac{1}{V_2}\right) \pm \frac{r(V_1 - V_2)}{V_1 V_2}\right] \end{aligned} \tag{4}$$

We will solve this system for the parameters $V_1 = 10, V_2 = 20, r = 10$, and the initial conditions $s_1(0) = 4, s_2(0) = 8$. Using the formula above, we obtain the eigenvalues:

$$\begin{aligned}\lambda_1 &= -1\\ \lambda_2 &= -0.5\end{aligned}$$

Our system's coeffient matrix becomes:

$$A = \begin{bmatrix} -1 & 0 \\ 1 & -0.5 \end{bmatrix} \tag{5}$$

By plugging the values of λ_1 and λ_2 to the eigenvector equation $(A - \lambda I)\mathbf{v} = 0$ and solving for $\mathbf{v}$, we obtain the two distinct eigenvectors of the coefficient matrix:

$$\begin{aligned}\mathbf{v}_1 &= \begin{bmatrix} -0.5 \\ 1 \end{bmatrix}\\ \mathbf{v}_2 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}\end{aligned}$$

Plugging the eigenvalues and their corresponding eigenvector to the *ansatz*, we obtain the following general solution:

$$\begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = C_1 \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} e^{-t} + C_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} e^{-0.5t}$$

To find the constants C_1 and C_2, we provide the initial condition to the equation:

$$\begin{bmatrix} 40 \\ 80 \end{bmatrix} = C_1 \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} + C_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$C_1 = -8, C_2 = 16$$

Which gives us the following specifc solution:

$$\begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = -8 \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} e^{-t} + 160 \begin{bmatrix} 0 \\ 1 \end{bmatrix} e^{-0.5t}$$

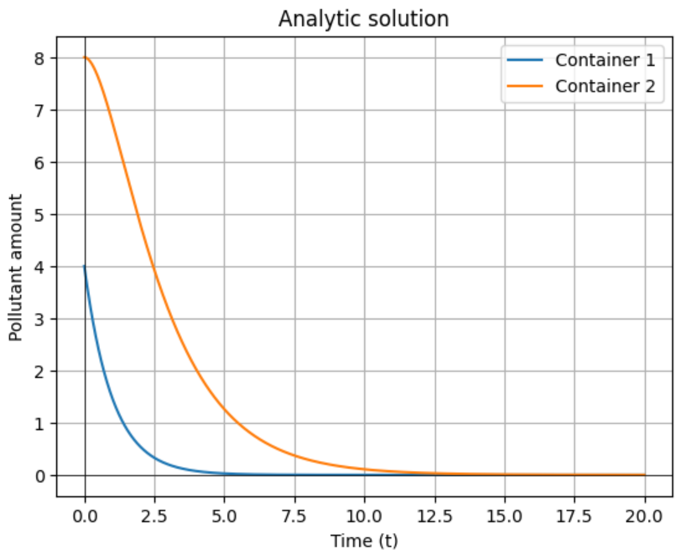


Figure 1: Plot of the analytic solution

4 Numeric solution

We use the parameters and initial conditions specified in the previous section to find the numeric solution of the system. Our script is written in Python and uses SciPy to compute the Runge-Kutta method. The matrix A in equation (5) is used as the computation matrix. The following result is obtained:

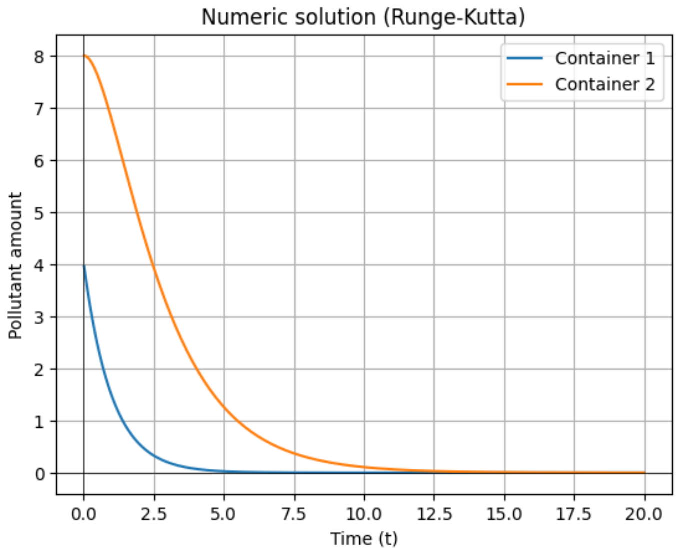


Figure 2: Plot of the numeric solution

The following is a comparison of the numeric and analytic methods' results:

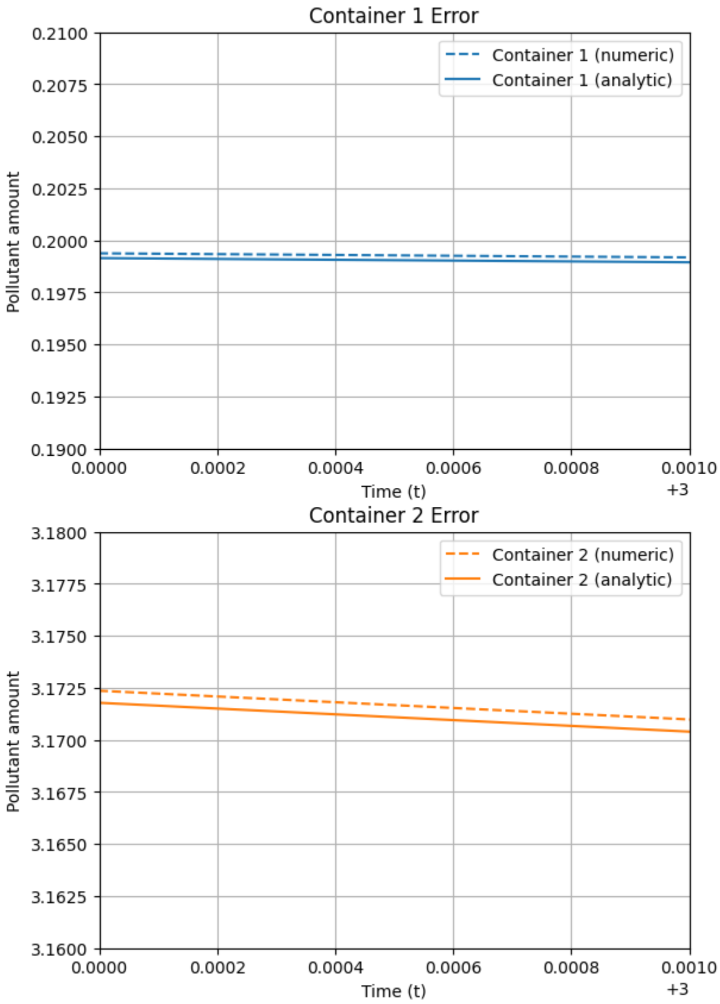


Figure 3: Comparison of the numeric and analytic methods

4.1 Parameter variation

In this section, we demonstrate the different numerical solutions for different parameter values. Here, we choose to vary one parameter while fixing the rest to the values we have set before. It is also worth noting that because we only solved for cases where $V_1 \neq V_2$, we have adjusted the volume variations to match this restriction.

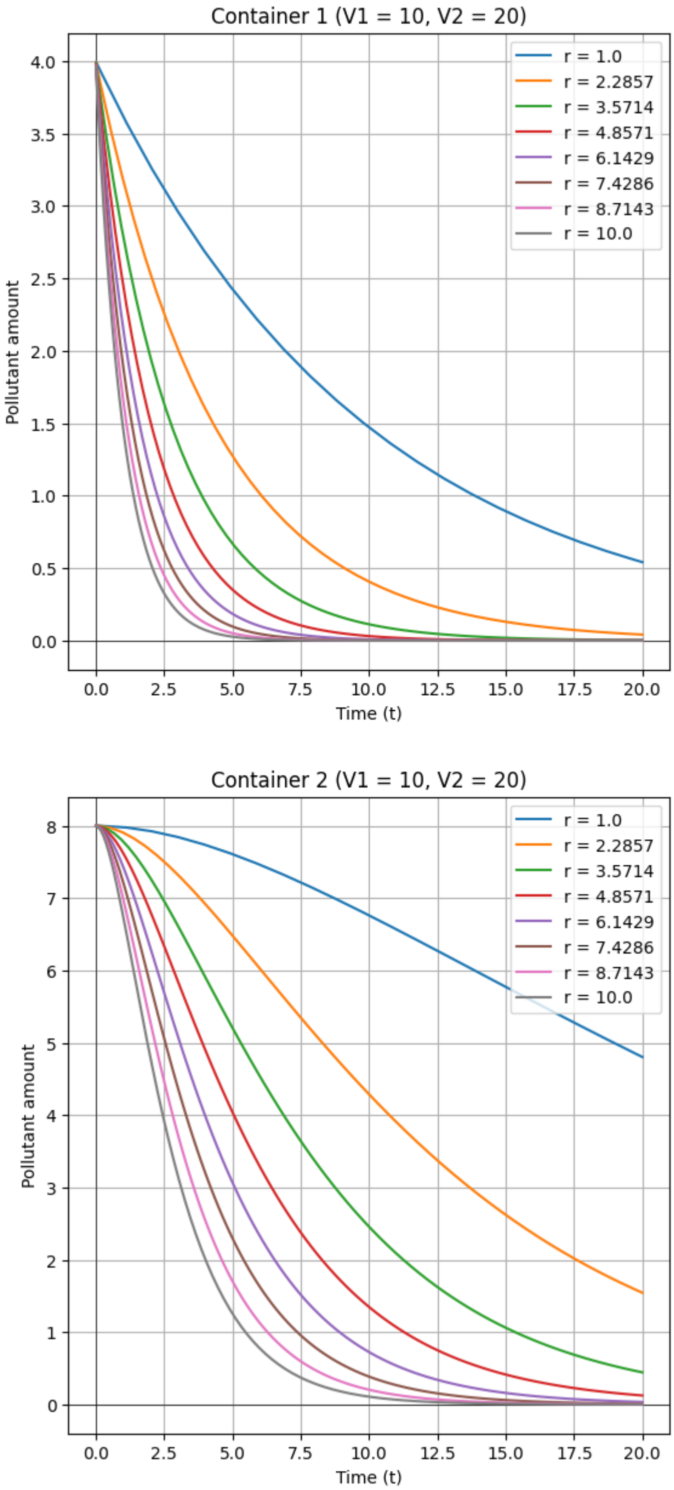


Figure 4: Variable r values

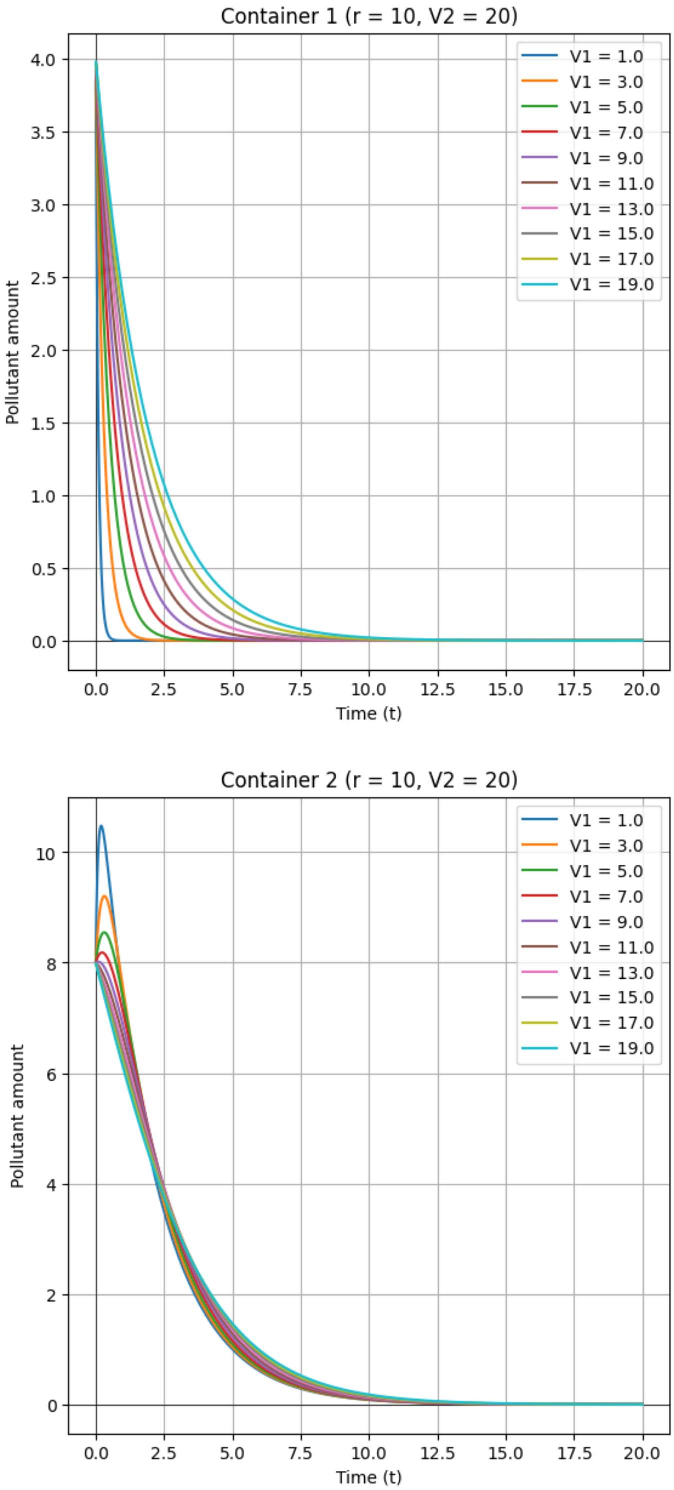


Figure 5: Variable V_1 values

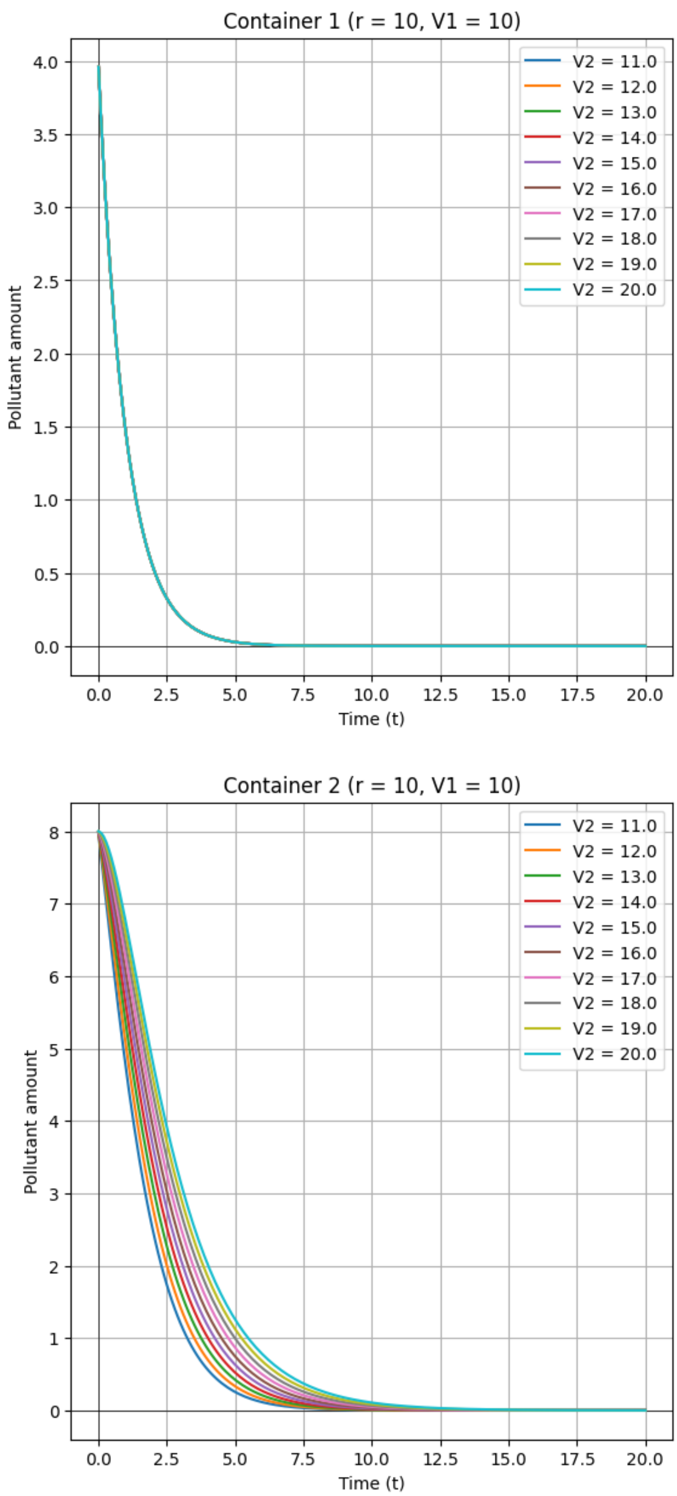


Figure 6: Variable V_2 values

5 Analysis

5.1 Varying r

As r is the rate at which clean water is pumped into the containers, it only affects how fast a container is drained. The greater the value of r is, the faster both containers are drained.

5.2 Varying V_1

Plotting for $s_2(t)$ under varying V_1 values shows peaks near $t = 0$ that disappear after making V_1 large enough. This is not a numerical error, and we demonstrate this by showing that a stationary point can exist after $t = 0$, where the peak would be visible. We begin by deriving the general form of the eigenvector of A:

$$\begin{aligned}
(A - I\lambda_1)\mathbf{v}_1 &= \mathbf{0} \\
\Leftrightarrow \begin{bmatrix} 0 & 0 \\ \frac{r}{V_1} & \frac{r(V_2-V_1)}{V_1V_2} \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\Leftrightarrow \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} &= \alpha \begin{bmatrix} \frac{V_1-V_2}{V_2} \\ 1 \end{bmatrix}, \alpha \in \mathbb{R}
\end{aligned} \tag{6}$$

$$\begin{aligned}
(A - I\lambda_2)\mathbf{v}_2 &= \mathbf{0} \\
\Leftrightarrow \begin{bmatrix} \frac{r(V_1-V_2)}{V_1V_2} & 0 \\ \frac{r}{V_1} & 0 \end{bmatrix} \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\Leftrightarrow \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} &= \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \beta \in \mathbb{R}
\end{aligned} \tag{7}$$

We can take $\alpha = 1, \beta = 1$ to obtain the solution:

$$\begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = \frac{s_1(0)V_2}{V_1 - V_2} \begin{bmatrix} \frac{V_1-V_2}{V_2} \\ 1 \end{bmatrix} e^{\lambda_1 t} + \left(s_2(0) - \frac{s_1(0)V_2}{V_1 - V_2} \right) \begin{bmatrix} 0 \\ 1 \end{bmatrix} e^{\lambda_2 t}$$

We take the derivative of $s_2(t)$ to find where the peak is:

$$s_2'(t) = \frac{s_1(0)V_2}{V_1 - V_2} \lambda_1 e^{\lambda_1 t} + \left(s_2(0) - \frac{s_1(0)V_2}{V_1 - V_2} \right) \lambda_2 e^{\lambda_2 t}$$

$$\begin{aligned}
0 &= \frac{s_1(0)V_2}{V_1 - V_2} \lambda_1 e^{\lambda_1 t} + \left(s_2(0) - \frac{s_1(0)V_2}{V_1 - V_2} \right) \lambda_2 e^{\lambda_2 t} \\
\Leftrightarrow \ln \left(\frac{s_1(0)V_2}{V_1 - V_2} \lambda_1 e^{\lambda_1 t} \right) &= \ln \left[\left(\frac{s_1(0)V_2}{V_1 - V_2} - s_2(0) \right) \lambda_2 e^{\lambda_2 t} \right] \\
\Leftrightarrow t &= \frac{V_1V_2}{r(V_1 - V_2)} \ln \left(\frac{V_1}{V_2} - \frac{s_2(0)V_1(V_1 - V_2)}{s_1(0)V_2^2} \right)
\end{aligned} \tag{8}$$

To determine under which conditions the peaks are visible, we use the inequality $t > 0$:

$$\frac{V_1 V_2}{r(V_1 - V_2)} \ln\left(\frac{V_1}{V_2} - \frac{s_2(0)V_1(V_1 - V_2)}{s_1(0)V_2^2}\right) > 0$$

$$\begin{cases} \frac{V_1}{V_2} - \frac{s_2(0)V_1(V_1-V_2)}{s_1(0)V_2^2} & > 1 \\ \frac{V_1}{V_2} - \frac{s_2(0)V_1(V_1-V_2)}{s_1(0)V_2^2} & > 0 \end{cases}$$

With further modifications we obtain

$$\begin{cases} \frac{V_1}{V_2} - \frac{s_2(0)V_1^2}{s_1(0)V_2^2} + \frac{s_2(0)V_1}{s_1(0)V_2} & > 1 \\ \frac{V_1}{V_2} - \frac{s_2(0)V_1^2}{s_1(0)V_2^2} + \frac{s_2(0)V_1}{s_1(0)V_2} & > 0 \end{cases}$$

Let $x = \frac{V_1}{V_2}$, the ratio of the first container's volume and the second container's volume.

$$\begin{cases} -\frac{s_2(0)}{s_1(0)}x^2 + \left(\frac{s_2(0)}{s_1(0)} + 1\right)x - 1 & > 0 \\ -\frac{s_2(0)}{s_1(0)}x^2 + \left(\frac{s_2(0)}{s_1(0)} + 1\right)x & > 0 \end{cases} \tag{9}$$

The roots of the first polynomial are:

$$x_1 = 1$$
$$x_2 = \frac{s_1(0)}{s_2(0)}$$

Keeping in mind that we explicitly solved for the case where $V_1 \neq V_2$, x, and all parameters are positive, non-zero real numbers, this gives us the regions $1 < x < \frac{s_1(0)}{s_2(0)}$ or $\frac{s_1(0)}{s_2(0)} < x < 1$. These are the regions where the peaks are visible after $t = 0$.

The second inequality tells us when the peak ceases to exist. The roots of the polynomial are:

$$x_1 = 0$$
$$x_2 = \frac{s_1(0)}{s_2(0)} + 1$$

As neither containers can have a negative volume, it must be the case that $x < 0$. Therefore $x > \frac{s_1(0)}{s_2(0)} + 1$ is the region where the peak ceases to exist.

5.3 Varying V_2

V_2 does not appear at all in $s_1(t)$, thus it has no effect on the first container. It only affects how $s_2(t)$ behaves. The smaller the volume, the faster the pollutants are evacuated from the second container.

6 Conclusion

An effective way to characterize a cascade system is to formulate ordinary differential equations, and then solve them using both analytical and numerical techniques. After the formulas for the cascade system are established, the analytical solution is obtained by determining the eigenvalues of the equations and then processing the obtained information to formulate the final solution equation.

The volume of the first container effects the peak visibility in the second container, the pumping rate influences the total drainage speed, and the second container's volume controls the pace at which pollutants are evacuated from it. The study includes conditions for certain system behaviors as well as mathematical derivations.

References

[1] Varberg, D. E., Purcell, E. J., and Rigdon, S. E. *Calculus early transcendentals.* Pearson custom library. Pearson, first edition, pearson new international edition edition, (2014).

[2] Boyce, W. E., DiPrima, R. C., and Meade, D. B. *Elementary differential equations and boundary value problems.* Wiley, Hoboken, NJ, eleventh edition edition, (2017).

[3] Gustafson, G. B. *Differential Equations and Linear Algebra*, volume 2. (2022).

Appendix: Numerical method code

```python
import numpy as np
import matplotlib.pyplot as plt

from scipy.integrate import RK45

"""## Analytic plot"""

t = np.arange(0, 20, 0.001, dtype=np.float64)

s1 = 4*np.exp(-t)
s2 = -8*np.exp(-t) + 16*np.exp(-0.5*t)

fig, ax = plt.subplots()
ax.plot(t, s1, label='Container 1')
ax.plot(t, s2, label='Container 2')
ax.axhline(0, color='black', linewidth=.5)
ax.axvline(0, color='black', linewidth=.5)
ax.grid(True, which='both')
ax.set_title('Analytic solution')
ax.set_xlabel('Time (t)')
ax.set_ylabel('Pollutant amount')
ax.legend()

"""## Numerical method and plot"""

def fun(t, x):
    A = np.array([[-1, 0], [1, -0.5]])
    return A @ x

    t0 = 0
    y0 = [4, 8]
    rk = RK45(fun=fun, t0=t0, y0=y0, t_bound=20, vectorized=True,
        max_step=1, rtol=1e-8, atol=1e-10)
    n_t = []
    n_y = []
    for i in range(200):
        rk.step()
        n_t.append(rk.t)
        n_y.append(rk.y)
        if rk.status == 'finished':
            break
            n_t = np.array(n_t)
            n_y = np.array(n_y)

fig, ax = plt.subplots()
ax.plot(n_t, n_y, label=["Container 1", "Container 2"])
ax.axhline(0, color='black', linewidth=.5)
ax.axvline(0, color='black', linewidth=.5)
```

```python
ax.grid(True, which='both')
ax.set_title('Numeric solution (Runge-Kutta)')
ax.set_xlabel('Time (t)')
ax.set_ylabel('Pollutant amount')
ax.legend()

"""### Comparison with analytic result"""

fig, ax = plt.subplots(2,1)
fig.set_figheight(10)
s = [s1, s2]
xl = [(3, 3.001), (3, 3.001)]
yl = [(0.19, 0.21), (3.16, 3.18)]
color = ['tab:blue', 'tab:orange']
for i in range(2):
    AX = ax[i]
    AX.plot(n_t, n_y[:,i], label=f"Container {i+1} (numeric)", linestyle='--', color=color[i])
    AX.plot(t, s[i], label=f"Container {i+1} (analytic)", color=color[i])
    AX.set_xlim(xl[i])
    AX.set_ylim(yl[i])
    AX.axhline(0, color='black', linewidth=.5)
    AX.axvline(0, color='black', linewidth=.5)
    AX.grid(True, which='both')
    AX.set_title(f'Container {i+1} Error')
    AX.set_xlabel('Time (t)')
    AX.set_ylabel('Pollutant amount')
    AX.legend()

def factory(V1, V2, r):
    def f(t, x):
        A = np.array([[-r/V1, 0], [r/V1, -r/V2]])
        return A @ x
    return f

"""### Varying £r£"""

r = np.linspace(1, 10, num=8)
fig, ax = plt.subplots(2, 1)
fig.set_figheight(15)
t0 = 0
y0 = [4, 8]
for i in range(2):
    AX = ax[i]
    AX.axhline(0, color='black', linewidth=.5)
    AX.axvline(0, color='black', linewidth=.5)
    AX.grid(True, which='both')
    AX.set_title(f'Container {i + 1} (V1 = 10, V2 = 20)')
    AX.set_xlabel('Time (t)')
```

```
    AX.set_ylabel('Pollutant amount')

for i in r:
    r_f = factory(10, 20, i)
    rk = RK45(fun=r_f, t0=t0, y0=y0, t_bound=20, vectorized=True,
     ↪  max_step=1, rtol=1e-8, atol=1e-10)
    r_t = []
    r_y = []
    for j in range(200):
        rk.step()
        r_t.append(rk.t)
        r_y.append(rk.y)
        if rk.status == 'finished':
        break
        r_t = np.array(r_t)
        r_y = np.array(r_y)
    for j in range(2):
        AX = ax[j]
        AX.plot(r_t, r_y[:, j], label=f"r = {i:.05}")

for i in range(2):
AX = ax[i]
AX.legend()

"""### Varying £V_1£"""

V1 = np.linspace(1, 19, num=10)
fig, ax = plt.subplots(2, 1)
fig.set_figheight(15)
t0 = 0
y0 = [4, 8]
for i in range(2):
    AX = ax[i]
    AX.axhline(0, color='black', linewidth=.5)
    AX.axvline(0, color='black', linewidth=.5)
    AX.grid(True, which='both')
    AX.set_title(f'Container {i + 1} (r = 10, V2 = 20)')
    AX.set_xlabel('Time (t)')
    AX.set_ylabel('Pollutant amount')

for i in V1:
    r_f = factory(i, 20, 10)
    rk = RK45(fun=r_f, t0=t0, y0=y0, t_bound=20, vectorized=True,
     ↪  max_step=1, rtol=1e-8, atol=1e-10)
    r_t = []
    r_y = []
    for j in range(200):
    rk.step()
    r_t.append(rk.t)
    r_y.append(rk.y)
```

```
    if rk.status == 'finished':
    break
    r_t = np.array(r_t)
    r_y = np.array(r_y)
    for j in range(2):
        AX = ax[j]
        AX.plot(r_t, r_y[:, j], label=f"V1 = {i:.05}")

for i in range(2):
    AX = ax[i]
    AX.legend()

"""### Varying £V_2£"""

V2 = np.linspace(11, 20, num=10)
fig, ax = plt.subplots(2, 1)
fig.set_figheight(15)
t0 = 0
y0 = [4, 8]
for i in range(2):
    AX = ax[i]
    AX.axhline(0, color='black', linewidth=.5)
    AX.axvline(0, color='black', linewidth=.5)
    AX.grid(True, which='both')
    AX.set_title(f'Container {i + 1} (r = 10, V1 = 10)')
    AX.set_xlabel('Time (t)')
    AX.set_ylabel('Pollutant amount')

for i in V2:
    r_f = factory(10, i, 10)
    rk = RK45(fun=r_f, t0=t0, y0=y0, t_bound=20, vectorized=True,
    ↪  max_step=1, rtol=1e-8, atol=1e-10)
    r_t = []
    r_y = []
    for j in range(200):
    rk.step()
    r_t.append(rk.t)
    r_y.append(rk.y)
    if rk.status == 'finished':
    break
    r_t = np.array(r_t)
    r_y = np.array(r_y)
    for j in range(2):
        AX = ax[j]
        AX.plot(r_t, r_y[:, j], label=f"V2 = {i:.05}")

for i in range(2):
    AX = ax[i]
    AX.legend()
```