Homework 2 Solutions

Overview

Load Environment

The following code loads the environment and makes sure all needed packages are installed. This should be at the start of most Julia scripts.

```julia
import Pkg
Pkg.activate(@__DIR__)
Pkg.instantiate()
```

```julia
using Plots
using LaTeXStrings
using CSV
using DataFrames
using Roots
```

Problems (Total: 50/60 Points)

Problem 1 (25 points)

CRUD decays exponentially at a rate of 0.36 d^{-1}, so the differential equation for its mass M is

$$\frac{dM}{dt} = -0.36M.$$

By separating variables, the solution to this equation is

$$M(t) = M_0 \exp(-0.36t).$$

However, we need to convert time t (d) to distance x (km) as the river is in steady-state. As the velocity of the river is 10 km/d,

$$t = \frac{x}{10},$$

so

$$M(x) = M_0 \exp(-0.36x/10).$$

To find the initial condition(s), we have to split the river into two different segments. The first segment goes from $x = 0$ (the first wastewater discharge) to $x = 15$ (the second discharge), and the initial condition satisfies $M_0 = M(0)$. To calculate this mass, we have to compute the total mass of CRUD in the combined inflow and wastewater discharge, which is

$$M_0 = M(0) = (250,000\text{m}^3/\text{d})\,(0.5\text{kg}/(1000\text{m}^3)) + (40,000\text{m}^3/\text{d})\,(9\text{kg}/(1000\text{m}^3))$$
$$= 485\text{kg}/\text{d}.$$

The second segment starts at $x = 15$ and follows the equation

$$M(x) = M_1 \exp(-0.36(x - 15)/10), \qquad x \leq 15,$$

where the initial condition satisfies

$$M_1 = M(15) = (M_0 \exp\left((-0.36)(15)/10\right)\text{kg}/\text{d}) + (60,000\text{m}^3/\text{d})\,(7\text{kg}/(1000\text{m}^3))$$
$$= 485\exp(-0.54)\text{kg}/\text{d} + 420\text{kg}/\text{d}$$
$$\approx 703\text{kg}/\text{d}.$$

Converting to concentration for the regulatory standard, we combine the inflow and discharge volumes to get a volume in the first segment of $290,000\text{m}^3$ and $350,000\text{m}^3$ in the second segment. This means that the concentration initial conditions[1] are

$$C_0 \approx 1.7\text{kg}/(1000\text{m}^3)$$

and

$$C_1 \approx 2.0\text{kg}/(1000\text{m}^3).$$

---

[1]You might get some variations in the specifics due to the number of significant figures you select.

Denoting concentration (in kg/(1000m³)) as $C(t)$, this means the model for concentration is

$$C(t) = \begin{cases} 1.7\exp(-0.36x/10), & \text{if } 0 \le x < 15 \\ 2.0\exp(-0.36(x-15)/10), & \text{if } x > 15. \end{cases}$$

To determine if the system is in compliance with the regulatory standard of 2.5kg/(1000m³), notice that, as there are no sources of CRUD inbetween discharges, the maximum value in each segment occurs at the discharge locations $x = 0$ and $x = 15$[2]. The maximum CRUD concentration occurs at $x = 15$, where $C(15) = 2.0$. As a result, the river is in compliance with the standard.

Problem 2 (25 points)

First, let's write a function to simulate the model from $t = 0, \dots, 30$.

```
# define functions for lake recycling and outflows
lake_P_cycling(x, q) = x.^q ./ (1 .+ x.^q);
lake_P_out(x, b) = b .* x;


T = 30
X_vals = collect(0.0:0.1:2.5)
function simulate_lake_P(X_ic, T, b, q, a, y)
    X = zeros(T)
    X[1] = X_ic
    for t = 2:T
        X[t] = X[t-1] .+ a[t] .+ y[t].+ lake_P_cycling(X[t-1], q) .-
 ↪  lake_P_out(X[t-1], b)
    end
    return X
end
```

```
simulate_lake_P (generic function with 1 method)
```

We can then make the initial conditions plot by evaluating this function across a range of values of X_ic (Figure 1)

---

[2]We could also have written some code to simulate this model, but using this observation makes the solution simpler.

```
X_vals = 0.0:0.05:2.5
# set parameters
T = 30
b = 0.5
q = 1.5
y = zeros(T)
a = zeros(T)

# evaluate model for initial values
X = map(x -> simulate_lake_P(x, T, b, q, a, y), X_vals)          ①
plot(X, label=false, ylabel=L"$X_t$", xlabel="Time")
```

① The `map()` function evaluates the function in the first argument (which is here an anony-
mous function) over all values in an array and returns an array of the same size; in this
case, we get back an array of the same size as `X_vals`, where each element in that array
is a vector of length `T` with the model outputs. We could also have used broadcasting,
but `map()` and its variants (such as `mapslices()`) can be more versatile when applying
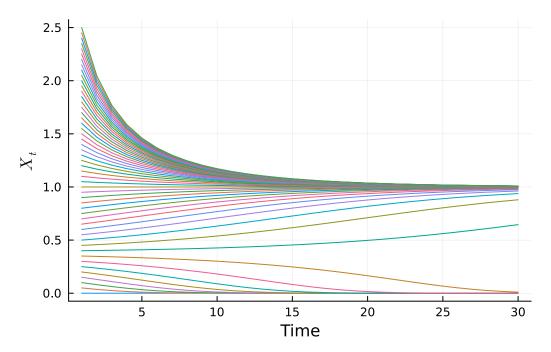a function over more complex structures, such as rows and columns of matrices.

Figure 1: Initial conditions plot for the lake problem for values of X_0 between 0 and 2 with no
inflows.

4

To find the equilibria, we use `Roots.find_zero()` for a function giving us the difference $X_{t+1} - X_t$, namely `lake_P_cycling(x, q) - lake_P_out(x, b)` (since we have no inflows). This function is versatile, and lets us either choose an initial point to start the search for a zero or a bounding interval. We can use Figure 1 to make guesses at initial values for the search.

```
eq_ic = [0.0, 0.5, 2.0]                                                    ①
eq_noinflows = [find_zero(x -> lake_P_cycling(x, q) - lake_P_out(x, b), eq)
↪    for eq in eq_ic]                                                       ②
```

① These are rough guesses from a visual scan of Figure 1; they just need to be closer to the "true" zeroes of interest than any others.
② We could also have used `map()` (as above) or broadcasting, but this illustrates the use of a comprehension to achieve the same goal, evaluating the function over the values in the array `eq_ic`. You can decide what you prefer to use in the future!

```
3-element Vector{Float64}:
 0.0
 0.3819660112501051
 1.0
```

So we end up with equilibria [0.0, 0.38, 1.0]. Of these, 0.0 and 1.0 are stable and 0.38 are unstable, based on Figure 1 (we could make this more rigorous by calculating the derivatives of `lake_P_cycling(x, q)` and `lake_P_out(x, b)` at the equilibrium values to see how the difference is changing, but that is not required here).

Now, let's repeat that analysis when $a_t = 0.05$ for all t.

```
a = 0.02 .+ zeros(T)                                                        ①

# evaluate model for initial values
X = map(x -> simulate_lake_P(x, T, b, q, a, y), X_vals)
plot(X, label=false, ylabel=L"$X_t$", xlabel="Time")
```

① This creates a vector of constant values by adding them to a zero vector; we could also do this by multiplying against a vector of ones, such as `a = 0.02 * ones(T)`.
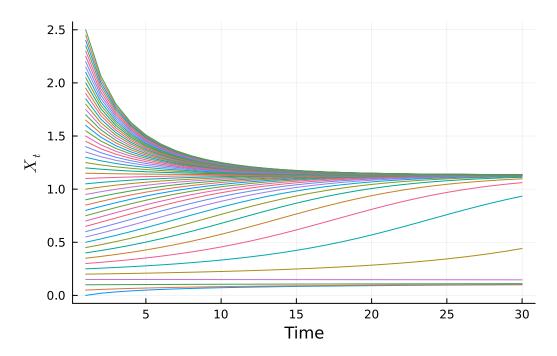
Figure 2: Initial conditions plot for the lake problem for values of X_0 between 0 and 2 with constant inflow $a_t = 0.05$.

```
eq_ic = [0.0, 0.5, 1.0]
eq_inflows = [find_zero(x -> lake_P_cycling(x, q) - lake_P_out(x, b) .+ a,
↪   eq) for eq in eq_ic]
```

```
3-element Vector{Float64}:
 0.11624441966074756
 0.15442860418310414
 1.1341008852434813
```

With $a_t = 0.02$, the equilibria are $[0.12, 0.15, 1.13]$, where 0.12 and 1.13 are stable and 0.38 are unstable, based on Figure 2.

How has the stability of the system changed? The unstable equilibrium has dropped from 0.38 to 0.15, which means a smaller disturbance to the system (such as a non-point source inflow or a change in the outflow rate) could trigger eutrophication. This is particularly true as the distance between the stable (oligotrophic) equilibrium and the unstable equilibrium has been reduced, which means moving past the unstable equilibrium is easier. Moreover, the eutrophic equilibrium is increased (by more than the 0.02 units of P that we are adding to the lake each time period), meaning that, upon eutrophication, the lake is likely to suffer from more intense consequences of higher nutrient levels, such as greater hypoxia.

Problem 3 (10 points)

Denote by β the rate of atmospheric deposition in $\text{kg}/(\text{yr} \cdot \text{m}^2)$. The atmospheric deposition input is then βA, where A is the area of the lake. The steady-state mass-balance equation becomes:

$$\frac{\partial \text{Mass}}{\partial t} = Inputs - Outputs - Decay = 0$$

$$\frac{\partial (CV)}{\partial t} = \sum_j PS_j + \sum_i NPS_i + \beta A - CQ_{out} - \alpha CV = 0. \tag{1}$$

Solving (1) for C:

$$C = \frac{\sum_j PS_j + \sum_i NPS_i + \beta A}{Q_{out} + \alpha V}.$$

With a constraint of $C \le 0.02\text{mg/L} = 2 \times 10^{-5}\text{kg/m}^3$, this means

$$\sum_j PS_j + \sum_i NPS_i = (2 \times 10^{-5})(Q_{out} + \alpha V) - \beta A,$$

and using the Vollenweider model for sedimentation ($\alpha \approx 10A/V$),

$$\sum_j PS_j + \sum_i NPS_i = (2 \times 10^{-5})(Q_{out} + 10A) - \beta A$$

$$= (2 \times 10^{-5})Q_{out} + ((2 \times 10^{-4}) - \beta)A.$$

Subbing in the parameters from class and the problem statement ($Q_{out} = 400 \times 10^6 \text{m}^3/\text{yr}, A = 30 \times 10^6 \text{m}^2, \beta = 1.6 \times 10^{-4}\text{kg}/(\text{yr} \cdot \text{m}^2), \sum_i NPS_i = 9000\text{kg/yr}$) yields

$$\sum_j PS_j \le 200\text{kg/yr},$$

which is the maximum allowable annual rate.

References

List any external references consulted, including classmates.