Andrew Branum

HW3

Problem 1:

Given the constraints, set of differential equations, and the equations in the appendix a simulation can be created where h1 and h2 are the heights of the fluid in the tanks, a1 and a2 are the areas of the tanks, f1 is flow into tank 1, f2 is flow from tank 1 to 2, and f0 is the flow out of tank 2. To run the simulation, we must set the initial conditions: a1 =1, a2 = 2, f0 = 0.02, f1 = 0.01, f2 = 0.01, and (h1,h2) = (0,0),(2,0),(0,1.5),(1,0),(0,1). After these initial conditions are set, we can either select set of heights for the tanks or iterate over all of them. I chose to use one set at a time to clearly see the results. Then the number of iterations must be chosen: 5. While iterating through the 5 steps, h1 and h2 must be updated using the given equations and their results must be stored to display on a graph. To calculate the change in height we can use Change = (Flow in – flow out)/Area and add it to the current height.

Some things I noticed while playing around with the simulation are that if f1 = f2, tank 1 will keep a constant height of liquid. If f1 > f2 it will overflow and if f1<f2 tank 1 will slowly empty. The same goes for tank 2 but with f2 and f0.

Here is an example simulation of the initial conditions given with h1 = 0 and h2 = 0.

A graph of fluid levels

Description automatically generated

Here is an example simulation of the values given in the appendix:

A graph of a fluid level

Description automatically generated

Problem 2:

Given the probability function f(s), the simulation needs to generate random numbers that are uniformly distributed between 0 and 1. To use the inverse distribution function method, we must first derive the Cumulative Distribution Function by using each formula with their constraints. For the first function x we get which equates to . The limit is from 0 to x since the constraint of the equation shows that x will be less than 1.

For the second function we must account for the probability from x to 1 since we only accounted for 0 to x in the first equation. We get where the first integral is the added probability, and the second one is the second equation from 1 to x. The outcome is . Now the updated function is:

To generate random numbers that follow the uniform distribution between 0 and 1, the simulation must solve for x in terms of U.

The simulation can now use these equations to generate random numbers following the triangular distribution. The pseudo-code for the simulation is: define a function that generates a random uniform variable between 0 and 1, if this variable is less than 0.5 then plug it into the function and return the variable. Else, plug it into the other function and return. This function is then run in a loop using X number of iterations (I will show multiple variations below). The data is appended to an array and is displayed as a histogram graph.

100 iterations: starts to show form, not enough values

A graph of random values

AI-generated content may be incorrect.

1000 iterations: much better, triangular shape clearly visible

A graph of a random variable

AI-generated content may be incorrect.

10000 iterations: even better, sides of triangle are becoming sharper

A graph of a random value

AI-generated content may be incorrect.

100000 iterations: can’t get much better than this

A graph of a random value

AI-generated content may be incorrect.

Problem 3:

This problem aims to evaluate the chances of passing an interview by randomly guessing on the answers. The probability of randomly guessing the correct answer is 1/5 = 20%. In the first assumption, HR’s last answer is always correct (1 free correct answer), and the candidate selects random answers. In the second assumption, the candidate simply guesses randomly for each question. Since no success criteria was given, I will assume that all questions must be answered correctly to pass. Since the candidate is randomly guessing on each question, the probability of passing is slim. It slightly increases with the first assumption since the candidate only must guess for 2 questions instead of 3. Without running a simulation, assumption 1’s probability can be calculated with since 1 question is given and 2 must be guessed. Assumption 2 can be calculated with . After running a simulation with 10,000 candidates, I found that these numbers are accurate. Assumption 1 yielded 3.66% and assumption 2 yielded 0.89%. Even randomly guessing on 2 questions offers a very small chance to pass the interview.

A white box with black text

AI-generated content may be incorrect.

If we assume that 2/3 need to be correct to pass, then the outcome is much different. The probability of assumption 1 grows to about 32% and assumption 2 is about 9%. I have appended additional results of this below.

Here are some more simulation runs:

A white box with black text

AI-generated content may be incorrect.

A white box with black text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

2/3 correct:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A white box with black text

AI-generated content may be incorrect.