Your group (you and up to two partner classmates) will be writing four different Python programs in four different files named hw5a.py, hw5b.py, hw5c.py and hw5d.py. You should make a private repository and invite Ashley to be a collaborator. Submit your repository address on Canvas. I have created ‘Stem’ files that have much of the necessary code written, BUT, key pieces of code have been deleted and marked with a comment #JES MISSING CODE. Often this tag has a comment associated to give you a hint at to what needs to be done. You are not required to use these program stems, but most students find them a nice starting point.

Note: most functions already have docstrings that don’t need modification, but for any function missing a docstring, write one.

1. BYOMD (Build Your Own Moody Diagram). The Moody diagram is used in fluid mechanics to relate the friction factor (*f*) to Reynolds number (*Re*) for pipes with various Relative roughness (*ϵ/d*) in the laminar and turbulent flow ranges with some ambiguity in transitional flow (2×103<Re<4×103). In the laminar range, the relative roughness seems to be irrelevant and whereas in the turbulent range, *f* is calculated with the Colebrook equation:

We note that the Colebrook equation has friction factor on both sides and that it is very difficult to isolate f. So, use fsolve to find the root of the Colebrook equation at each Re and relative roughness.

Write a program that produces a Moody diagram that has all the features like the one below.

Diagram

Description automatically generated

1. Beyond BYOMD: Create a function that takes as input the Reynolds number and relative roughness and outputs a red icon on the Moody diagram for the *Re*, *f* location. The icon should be an upward triangle if the flow is transition or a circle if otherwise. You should solicit input from the user through the command line (see <https://www.w3schools.com/python/python_user_input.asp> ). You may import functions from part a) as needed. **Note:** since there are three domains on your Moody diagram (laminar, transition and turbulent) your function should use either the laminar equation, the Colebrook equation or a random number from a normal distribution with:

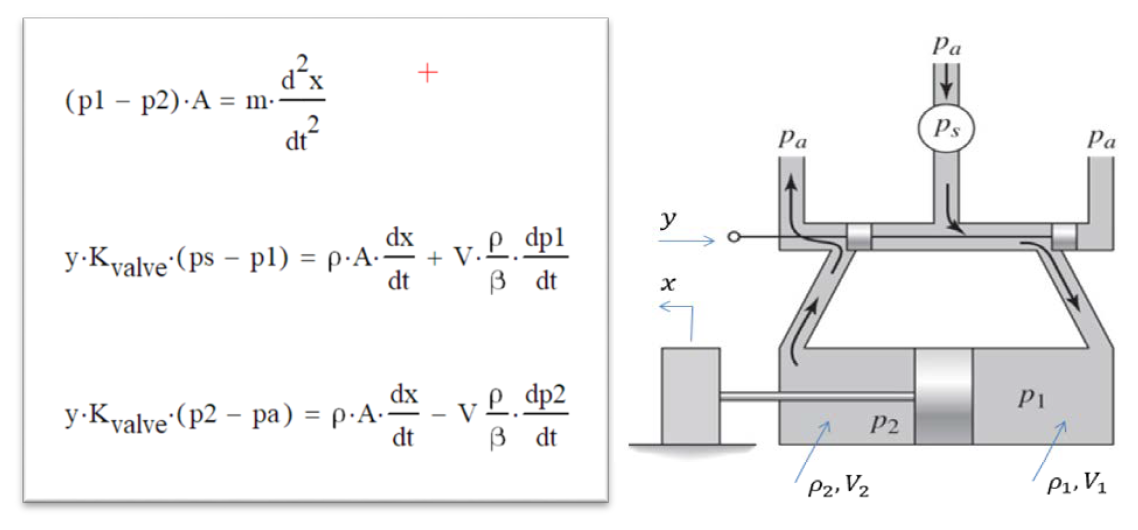
Where: *flam* and *fCB* are the friction factors at the given Re predicted by the laminar and Colebrook equations, respectively. See following figures for example output:

Diagram

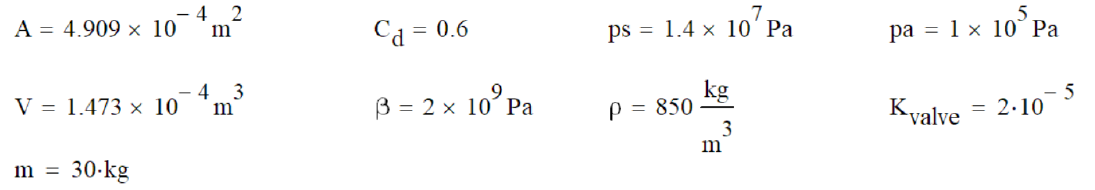
Description automatically generatedDiagram

Description automatically generated

1. The following differential equations describe the behavior of a hydraulic valve system



The values for the constant parameters are:



*NOTE:* **All units** *for the variables and the constants are* ***consistent*** *as given, and no unit conversions of any kind are necessary.*

*Required:*

Use solve\_ivp() from scipy.optimize to solve the differential equations for the response to a constant input of y=0.002

The initial conditions are: x=0, ẋ=0, p1=pa, p2=pa

1. From that solution, plot *ẋ* as a function of time, with nice title and labels.

2. From that solution, plot p1 and p2 together as functions of time, on a new graph, with nice title and labels and legend.

|  |  |  |
| --- | --- | --- |
| State Variable | Old Name | Derivative |
| X[0] | x | xdot |
| X[1] | xdot | xddot=(p1-p2)\*A/m |
| X[2] | p1 |  |
| X[3] | p2 | p2dot = -[y⋅Kvalve⋅(p2-pa)-ρ⋅A⋅xdot]β/(ρV) |

1. Use scipy.optimize.curve\_fit to fit some x-y data (see program stem) with a linear and a cubic model and plot the results as shown in the following graphs:

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated