In this homework, you (and you alone) will be writing programs that satisfy the following descriptions. In general, you should write your program in a modular fashion (meaning, where possible, create functions that can be called to do a particular task like produce a Moody diagram), use #region markers, write descriptive docstrings with step-by-step plans, and use the modules discussed in class effectively.

1. **BYOMD (Build Your Own Moody Diagram).** The Darcy-Weisbach friction factor (*f*) is used to compute head loss in pipe flow through:

where *L*=pipe length, *D* = pipe diameter, *V* = average velocity of the fluid, and *g* is the acceleration of gravity. *f* is known to vary with both Reynolds number (*Re*) and pipe wall roughness (Relative roughness (*ϵ/d*)). In the laminar range (*Re*<2000), the relative roughness seems to be irrelevant where whereas in the turbulent range (*Re*>4000), *f* is described by the empirical and implicit Colebrook equation:

We note that in the Colebrook equation, *f* cannot be found analytically so, we must use an iterative method (i.e., fsolve) to find *f* at each *Re* and *ϵ*/*d* coordinate. At intermediate *Re*, the flow is called *transitional*, and *f* is not easily predicted due to instability.

The Moody diagram graphically displays *f* as a function of *Re* and *ϵ*/*d* for a finite set of 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 program that solicits input from the user for: *pipe diameter* in inches, *pipe roughness* (*ϵ*) in micro-inches or mics (10-6 inches), *flow rate* in gallons/min and then returns the head loss per foot (*hf*/*L*) in appropriate English units.

Furthermore, you should display the Moody diagram with an icon of an upward triangle if the flow is transition or a circle if otherwise.

Your program should allow the user to re-specify the parameters and keep track of each *f* by just adding a new icon to the Moody diagram with each new set of parameters.

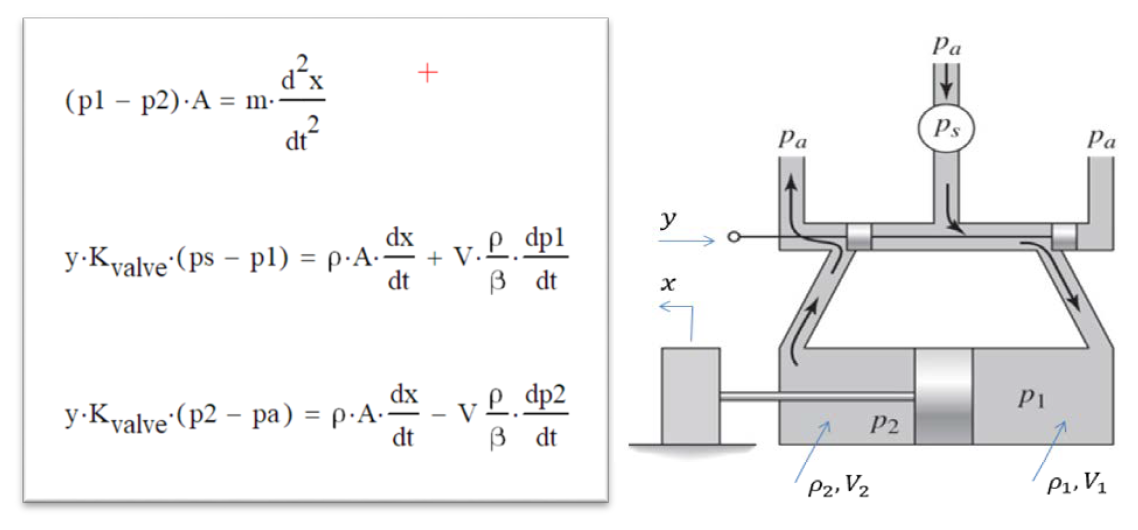
**Note:** It is likely that some user input may land us in the transition flow range where we will interpolate *f* between the prediction of the laminar and turbulent predictions at that Re and ϵ/dsuch that we exactly match these predictions at Re=2000 and Re=4000. At intermediate Re, add some randomness to *f* by assuming *f* follows a normal distribution with:

Diagram

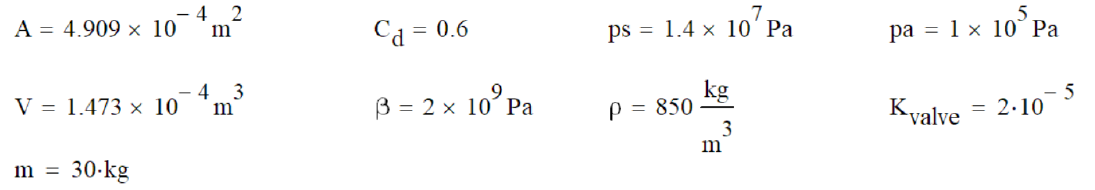
Description automatically generatedDiagram

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) |