ChE 390 Experiment 3: Process Model Discrimination and Simulation of Pipeline Network

John Zhang
Department of Chemical Engineering, University of Waterloo, Waterloo, Ontario

3.1. Lab Introduction

The design, simulation, and optimization of many chemical engineering processes often come down to a series of constrained or non-constrained nonlinear process equations. Solving these nonlinear equations requires a computing tool that is powerful, robust, and versatile in engineering computing and programming. Python has become such a computing tool that readily meets all aforementioned requirements in solving large-scale non-constrained and constrained nonlinear equations. Together with Python libraries such as Numpy, Scipy, Pandas, Python programming provides many built-in regression and optimization functions that enable the construction of reliable process models out of experimental data to be a relatively easy routine task. In terms of engineering learning and research, Python packages offer a variety of capability and flexibility for dealing with all types of complex and real chemical engineering problems with minimal programming effort required, and can therefore greatly enhance and extend the understanding of basic chemical engineering theories and their applications to real processes. In this computing lab, you will use Python functions of nonlinear regression and nonlinear equation solver to discriminate process models based on experimental data and to simulate pipeline network. The main objectives of this lab are:

- To reinforce the understanding of chemical engineering concepts by applying them to formulating process equations for complex chemical engineering processes.
- To enhance engineering problem-solving skills by using Python functions to discriminate process models and find the solutions to nonlinear process equations.
- To practise Python programming skills in functions and sub-functions.

3.2. Lab Problem Statement

Problem 1. Process Model Discrimination

The prediction of vapor-liquid equilibrium (VLE) of a non-ideal solution depends on the accurate prediction of activity coefficients of components in the liquid mixture. Table 1 lists some of the most commonly used model equations for the prediction of activity coefficients of a non-ideal binary system. In these equations, the parameters are adjustable parameters which must be determined from an experimentally derived set of vapor-liquid equilibrium data (i.e., values of activity coefficients γ_1 and γ_2 at different liquid compositions x_1 and x_2). For any specified system, one model equation is usually more accurate than others for predicting the activity coefficients. Hence, there is a need to discern the best model for a given system based on the experimental data.

Table 1. Some common model equations for binary liquid activity coefficients.

Model	Model Equations	Parameters
Margules Equation	$\ln\gamma_1 = x_2^2\left[A_{12} + 2(A_{21} - A_{12})x_1\right]$ $\ln\gamma_2 = x_1^2\left[A_{21} + 2(A_{12} - A_{21})x_2\right]$	A_{12} , A_{21}
Van Laar Equation	$\ln\gamma_1 = \dfrac{B_{12}}{\left[1 + (B_{12}x_1 / B_{21}x_2)\right]^2}$ $\ln\gamma_2 = \dfrac{B_{21}}{\left[1 + (B_{21}x_2 / B_{12}x_1)\right]^2}$	B_{12} , B_{21}
Wilson Equation	$\ln\gamma_1 = -\ln(x_1 + x_2G_{12}) + x_2\left(\dfrac{G_{12}}{x_1 + x_2G_{12}} - \dfrac{G_{21}}{x_2 + x_1G_{21}}\right)$ $\ln\gamma_2 = -\ln(x_2 + x_1G_{21}) - x_1\left(\dfrac{G_{12}}{x_1 + x_2G_{12}} - \dfrac{G_{21}}{x_2 + x_1G_{21}}\right)$	G_{12} , G_{21}
NonRandom Two Liquid (NRTL)	$\ln\gamma_1 = x_2^2\left[\tau_{21}\left(\dfrac{G_{21}}{x_1 + x_2G_{21}}\right)^2 + \dfrac{\tau_{12}G_{12}}{(x_2 + x_1G_{12})^2}\right]$ $\ln\gamma_2 = x_1^2\left[\tau_{12}\left(\dfrac{G_{12}}{x_2 + x_1G_{12}}\right)^2 + \dfrac{\tau_{21}G_{21}}{(x_1 + x_2G_{21})^2}\right]$ Where $G_{12} = \exp(-\alpha_{12}\tau_{12})$; $G_{21} = \exp(-\alpha_{12}\tau_{21})$ $\tau_{12} = \dfrac{\Delta g_{12}}{RT}$; $\tau_{21} = \dfrac{\Delta g_{21}}{RT}$	α_{12} , Δg_{12} , Δg_{21}

The problem in this lab requires you to use the most reliable regression method, nonlinear regression with statistical inferences, to discriminate the thermodynamic model best suitable for a binary system of ethanol and water. The experimental VLE data is normally collected using the same method as the VLE experiment in ChE 291, that is, a series of liquid and vapor compositions are measured at various boiling temperatures (a set of experimental data for ethanol-water system is available in the course page on LEARN). For the ethanol-water system at low pressures, the Raoult's law is usually used to determine experimentally measured activity coefficient of each component, i.e.

$$Py_i = \gamma_i x_i P_i^o$$

Where P is total pressure; y_i is the vapor composition, and P_i^o is the vapor pressure of pure component which can be calculated using the Antoine equation,

$$\log(P_i^o)(kPa) = A_i - \frac{B_i}{T(K) + C_i}$$

With the Antoine parameters given as follows:

Compound	A	B	C
Ethanol	7.28781	1623.22	-44.170
Water	7.19621	1730.63	-39.724

Problem 2. Pipeline Network Simulation

Pipeline networks such as water distribution systems, natural gas distribution systems, and chemical distribution systems in various chemical plants are some of the most encountered engineering systems. Figure 1 shows a typical pipeline network for the distribution of a coolant fluid (60 wt.% ethylene glycol in water) at 20°. The coolant needs to be withdrawn from nodes 3, 5, 6, and 7 at a constant flow rate. All the pipe sections are schedule 40 steel pipe with the roughness of $\varepsilon = 5.01 \times 10^{-5}\ m$. The equivalent lengths of the pipes connecting different nodes are as follows: $L_{01} = 260\ m$, $L_{12} = L_{23} = L_{45} = L_{67} = 450\ m$, $L_{24} = L_{35} = L_{46} = L_{57} = 400\ m$, and $L_{14} = 600\ m$. The density and viscosity of the coolant fluid component at 20° are as follows.

	Density (kg/m^3)	Viscosity (Pa·s)
Ethylene glycol	1113	0.0161
Water	998	0.0010

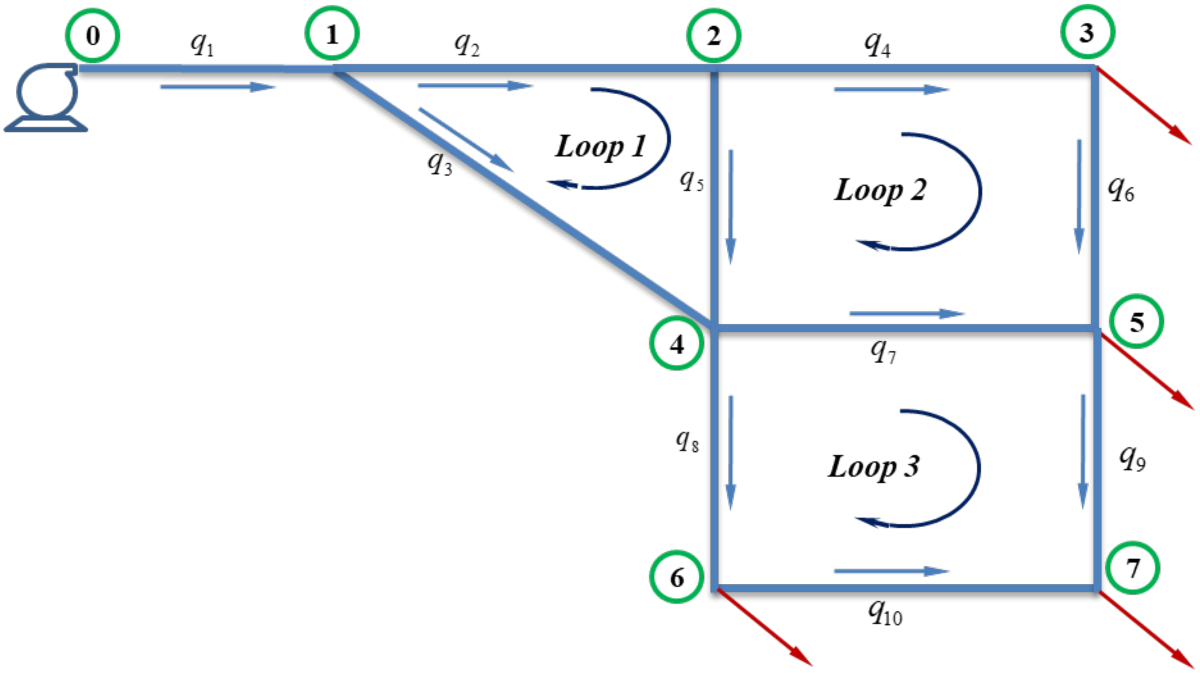

Figure 1. Schematic of water pipeline network

The design and operation of such a pipeline network usually requires the calculations of flow rates and pressure losses in each pipe section of the network as well as optimum pipe size.

3.3. Theories and Formulation of Process Equations for Python

3.3.1. Process Model Discrimination Using Nonlinear Regression

Nonlinear regression is based on the same principle, as linear regression, of minimizing the sum of squared errors (SSE) between experimentally measured values and model predicted values, i.e.,

$$SSE = \sum_{i=1}^{N} (y_{\exp t,i} - y_{pred,i})^2$$

Where $y_{\exp t,i}$ denotes the experimental value at independent variable x_i ; and $y_{pred,i}$ is the corresponding value predicted by a model $y_{pred,i} = f(x_i,\beta)$ with β being the model parameters.

Since we have two sets of measured activity coefficients in this binary system, we can combine them to create a single measured quantity or an objective function. Considering the physical significance of activity coefficient, a simple combination of the two activity coefficients can be used for nonlinear regression analysis, i.e.,

$$\gamma = \gamma_1 + \gamma_2 \qquad (1)$$

More often, a rather more meaningful thermodynamic quantity defined by the activity coefficients and referred to as the excess Gibbs free energy, g^E , can be used for the nonlinear regression,

$$g^E = x_1 \ln \gamma_1 + x_2 \ln \gamma_2 \qquad (2)$$

With all the experimentally measured values for the problem, Python Scipy function *curve_fit* can be used to compute the values of the model parameters in Table 1 and their confidence intervals, which can in return be used, along with regression R^2 and residual plot, to determine the best model for the system.

3.3.2. Pipeline Network Simulation and Optimization

One of the governing equations for fluid system design and simulation in chemical engineering processes is the general mechanical energy balance equation, which is, for an incompressible fluid between any point 1 and pint 2, written as,

$$Z_1 + \frac{P_1}{\rho g} + \frac{V_1^2}{2g} + W_s = Z_2 + \frac{P_2}{\rho g} + \frac{V_2^2}{2g} + \sum h_i \qquad (3)$$

Where

v: Fluid velocity in m/s.

z: Fluid level in m.

$\Delta P_{12} = P_1 - P_2$: Pressure loss in Pa.

ρ: Fluid density in kg/m^3.

g: gravitational constant in m/s^2.

Σh_i: Sum of head or energy loss due to friction in m.

W_s: Mechanical energy to the fluid in m.

Equation 3 indicates that the pressure loss in the absence of mechanical energy ($W_s = 0$) is a result of friction loss of fluid flow, and is therefore often predicted through friction factor. For isothermal fluid flow in pipes, the friction factor, f , is defined as the ratio of inertia energy to kinetic energy of flowing fluid,

$$f = \frac{\tau_w}{\rho v^2/2} = \frac{\Delta P \pi D^2/4}{\pi D L}/(\rho v^2/2) = \frac{\Delta P D}{2\rho v^2 L} \tag{4}$$

Where

ΔP : Pressure loss in Pa.
f : The Fanning friction factor.
L : Pipe length in m.
D : Pipe diameter in m.

Thus, the pressure drop can be calculated through friction factor for a flow system defined in terms of flow rate, pipe geometry, and fluid properties, i.e.,

$$\Delta P = 2 f \rho v^2 \frac{L}{D} \tag{5}$$

The friction factor for different flow regimes is generally correlated in terms of pipe surface roughness factor, ε in m and the Reynolds number which is defined as,

$$\mathrm{Re} = \frac{\rho v D}{\mu} \tag{6}$$

With μ being the fluid viscosity in $Pa \cdot s$.

Table 2. Common correlations of friction factor in pipes

Equation	formula	flow range	accuracy rating*
Hagen-Poiseuille	$f = \frac{16}{\mathrm{Re}}$	$\mathrm{Re} \le 2100$ Laminar flow	A
Nikuradse equation (1932)	$\frac{1}{\sqrt{f}} = 4.0\log(\mathrm{Re}\sqrt{f}) - 0.4$	$3000 < \mathrm{Re} < 3.4 \times 10^6$ Smooth pipe, $\varepsilon / D = 0$	B
Blasius	$f = \frac{0.079}{\mathrm{Re}^{0.25}}$	$4000 < \mathrm{Re} < 10^5$ Smooth pipe and turbulent flow	B
Colebrook and White equation (1939)	$\frac{1}{\sqrt{f}} = -4.0\log(\frac{\varepsilon}{D} + \frac{4.647}{\mathrm{Re}\sqrt{f}}) + 2.28$	$4000 < \mathrm{Re} < 1.0 \times 10^7$ Turbulent flow (also used for transition flow with no better correlations	A
Churchill Equation (1977)	$f = 2\left[(8/\mathrm{Re})^{12} + 1/(A+B)^{3/2}\right]^{1/12}$ $A = \left[2.457\ln\frac{1}{(7/\mathrm{Re})^{0.9} + 0.27(\varepsilon/D)}\right]^{16}$ $B = (37530/\mathrm{Re})^{16}$	$0 < \mathrm{Re} < 1.0 \times 10^7$ Smooth and rough pipes, and all flow regimes.	C

** Accuracy rating from the highest to lowest in alphabetical order.*

Table 2 summarizes some explicit and implicit correlations commonly used for predicting the friction factor in pipes for different flow regimes. Once the friction factor for a flow system is known with a guessed flow rate, the pressure drop can be determined using Equation 5, and then Equation 3 can be solved iteratively for the flow rate typical of pipeline design and simulation.

Just like any other chemical engineering problems, the underlying principles we can rely on to formulate the process equations for a very complex pipeline network are the mass balance equation and the energy balance equation. For the pipeline network in Figure 1, the steady-state mass balance entails the sum of all the flow rates into and out of each node be zero, while the steady-state energy balance (Equation 3) requires that the sum of all pressure drops surrounding a closed loop must be balanced and therefore be zero. This simple rule, best known as **the Hardy-Cross method**, allows us to easily formulate all the necessary questions for solving all the flow rates in any complicated pipeline network. For the pipeline network in Figure 1, we can use 7 nodes and 3 loops in the network system to come up with 10 equations necessary for 10 unknown flow rates.

The mass balance equations at the 7 nodes:

Node 1: $q_1 - q_2 - q_3 = 0$

Node 2: $q_2 - q_4 - q_5 = 0$

Node 3: $q_4 - q_6 - qs_3 = 0$

Node 4: $q_3 + q_5 - q_7 - q_8 = 0$

Node 5: $q_6 + q_7 - q_9 - qs_5 = 0$

Node 6: $q_8 - q_{10} - qs_6 = 0$

Node 7: $q_9 + q_{10} - qs_7 = 0$

Where qs_i is the side flow withdrawn from node i.

The energy balance equations for all 3 loops:

Loop 1: $\Delta P_{12} + \Delta P_{24} - \Delta P_{14} = 0$

Loop 2: $\Delta P_{23} + \Delta P_{35} - \Delta P_{24} - \Delta P_{45} = 0$

Loop 3: $\Delta P_{45} + \Delta P_{57} - \Delta P_{46} - \Delta P_{67} = 0$

As shown above, the pressure drop in each pipe section is related to its flow rate and friction factor (Equation 5), and the friction factor can be evaluated based on Reynolds number or flow rate and appropriate correlation in Table 1, so essentially, we have 10 equations for 10 unknown flow rates, which suffice to be solved for the flow rates first, then pressure losses at each pipe section can be calculated using Equation 5.

3.3.3. Costs and Economical Pipe Size of Pipeline Network

Once the pressure losses in pipeline network are determined as above, the total power requirement for the pipeline network can be determined by,

$$W = \frac{\sum_{i=1}^{n} q_i \Delta P_i}{\eta_p \eta_m} \tag{7}$$

Where

q_i: Flow rate of ith pipe section in m^3/s.

ΔP_i: Pressure loss ith pipe section in Pa.

η_p: Pump efficiency, assuming $\eta_p = 0.60$ as the average efficiency.

η_m: Pump motor efficiency, $\eta_m = 0.80$.

n: Total number of pipe sections.

The annualized operation cost (AOC) is then,

$$AOC = WN_{op}C_E \tag{8}$$

Where N_{op} is total number of operation hours (assume $N_{op} = 8420$ hours/year) and so WN_{op} is total annual energy consumption in kWh. C_E is the price of electricity (assume $C_E = 0.105\ \$/kWh$ in Ontario).

The annualized capital cost (ACC) for the pipeline network can be estimated by,

$$ACC = (1+F)C_o\left(\frac{D}{D_o}\right)^m (a+b)\sum_{i=1}^{n} L_i \tag{9}$$

Where

D: Pipe diameter

D_o: Diameter of reference pipe.

m: Pipe size cost exponent, $m = 1.25$

C_o: Purchase price of the reference pipe, use $C_o = 5.92\ \$/ft$ for $D_o = 1.0$ *inch*.

F: Factor that accounts for fittings, installation, finance, etc., assume $F = 1.0$.

$a,\ b$: Factors for maintenance, repair, etc., assume $a = b = 0.12$.

The total annualized cost (TAC) is the sum of the annualized operation cost and capital cost,

$$TAC = AOC + ACC \tag{10}$$

The values of TAC can be calculated for different pipe diameter. In general, AOC decreases with increasing pipe diameter while ACC is the opposite. The optimum or most economical pipe diameter can then be determined at the minimum TAC.

3.4. Computing Tasks, Discussion Questions, and Recommended Python Functions

3.4.1. Lab Computation Tasks

The lab includes the following computation tasks.

Task 1. Use the experimental VLE data of ethanol and water binary system at 101.325 *kPa* (adopted from *Journal of Chemical & Engineering Data, 38, 446,1993*) in Excel file on the course page of LEARN to do the following calculations:

a. Use Equation 2 and ***optimize.curve_fit*** to determine the values of the two parameters in the Wilson model and their 95% confidence intervals. Calculate the goodness of the fit by computing R^2 value.

b. Repeat (a) with the NRTL model, respectively. Which model best represents the system and why?

c. Use the best model from Parts a and b to compute the VLE data of ethanol and water binary system at 32.86 *kPa*, and compare with the experimental VLE data published in *Fluid Phase Equilibria, 308, 135-141, 2011* (use *plot* function to plot y-x and T-xy diagram, respectively, and use symbols for the literature values and lines for computed values).

Task 2. Use the pipeline network theory to do the following calculations for an operation case where the end flows from nodes 3, 5, 6, and 7 are all 78.0 L/min:

a. With the pipe diameter of 1.0 inch, compute the steady-state flow rates and corresponding pressure losses for all pipe sections using the explicit Churchill equation, and use Pandas data frame to present your results.

b. Repeat part (a) using the implicit Colebrook and White equation (Note: Hagen-Poiseuille equation should be used for $\mathrm{Re} < 3{,}000$).

c. Using your codes from part (a) to calculate AOC, ACC, and TAC for pipe diameter ranging from 1 inch to 4 inches with an increment of 0.25 inch. Plot AOC, ACC, and TAC versus pipe diameter in inches and determine the optimum pipe diameter for the system.

d. Repeat part (c) using your codes from part (b).

Of all the computation tasks, tasks 1a, 2a, and 2c are in-class exercise which is due at 11:59 pm three days after your lab session, the remainder of the computation tasks is the lab assignment which is due two weeks after the lab session.

For the in-class exercise, you only need to submit your Python script file and a summary of your results. For the assignment, you need to submit your code and a short report that must include a summary and discussion of your results as well as answers to all the questions in Section 3.2. It is important that you organize your computation results properly into a word file, and answer all the questions in your report. You need to submit both word file and your Python script files.

3.4.2. Discussion Questions

Based on the results from the above computation tasks, you need to answer the following questions in your lab report:

1. In the report introduction, give at least two examples from your core courses where process models or correlations need to be determined by nonlinear regression.

2. Tabulate the parameter values and their statistical inferences (R^2 values, SSE , and confidence intervals) for each model, and determine the best model for the ethanol-water system. Comment on your selection based on above results and residual plots.
3. How well do the water-ethanol VLE data at 32.86 kPa computed from your selected model agree with the experimental values from another independent source? Comment on your findings in terms of the rigor of the VLE theory and computation algorithm involved in your computation.
4. Compare and discuss the simulation results obtained by using the implicit correlation (the Colebrook and White equation) and using the explicit correlation (Churchill equation)? Why do we want to use the implicit correlation (the Colebrook and White equation) when the explicit correlation (Churchill equation) is available?
5. Discuss the effects of the pipe diameter on the economics (AOC, ACC, and TAC) of the pipeline network based on different correlations. Further discuss about the optimum pipe diameter determined based on different correlations.
6. Based on your experience, what recommendation do you have to improve the solution process of this computing lab?

3.4.3. Recommended Python Functions

The following Numpy and Scipy functions should work for the computation tasks in this lab, although there are other Python functions that may also be suitable.

1. Scipy Function for Bounded Nonlinear Regression (*optimize.curve_fit)*

   curve_fit function syntax:

   from scripy.optimize import curve_fit
   m, mcov=curve_fit(func, x_data, y_data, mo, bounds)

   Where

   func: User defined function that is declared to evaluate predicted y before cuve_fit is called.
   x_data: A vector of the independent variable data points.
   y-data: A vector of the dependent variable data points.
   mo: Vector of the initial guessed values for unknown parameter m.
   bounds: A tuple vector that defines lower and upper bounds for parameters m.
   m: Vector of unknown parameters that best fits nonlinear equation $y = f(m, x)$.
   mcov: Covariance of the parameters.

2. Scipy Function for Solving a Series of Nonlinear Algebraic Equation: *optimize.fsolve*

   fsolve function syntax:
   from scripy.optimize import fsolve
   x=fsolve(nlinFunc, xo, args, xtol)

   Where

nlinFunc: function that specifies the nonlinear equations in the form of $f(x)=0$.
xo: A vector of initial guessed values for unknown *x.*
args: Arguments that need to be passed for the nonlinear function evaluation.
xtol: Tolerance for unkown x.
x: A vector of solved values that best satisfy the nonlinear equations $f(x)=0$.

3.5. Some Useful Python Programming Tips and Solution Techniques

1. The goodness or adequacy of the fit for nonlinear regression, R^2, can be calculated from the Python output as follows:
First calculate the Sum of Squared Errors (SSE),

$$SSE = sum((y_{pred} - y).\hat{}2)$$

Where y_{pred} is the vector of model predicted values at parameter solution, and y is the corresponding vector of experimental values.
Next, calculate the Total Sum of Squares (SST),

$$SST = sum((y - \overline{y}).\hat{}2)$$

Where $\overline{y}$ is the average value of the experimentally measured y.
Last, R^2 is defined and calculated by,

$$R^2 = 1 - \frac{SSE}{SST}$$

2. To take advantage of the array and matrix operations in Numpy, we can assign the lengths of all the 10 pipe sections in **Figure 1** into a vector/array which matches flow rate vector, i.e.,

$$L = \begin{bmatrix} L(1) \\ L(2) \\ L(3) \\ L(4) \\ L(5) \\ L(6) \\ L(7) \\ L(8) \\ L(9) \\ L(10) \end{bmatrix} = \begin{bmatrix} L_{01} \\ L_{12} \\ L_{14} \\ L_{23} \\ L_{24} \\ L_{35} \\ L_{45} \\ L_{46} \\ L_{57} \\ L_{67} \end{bmatrix}$$

The unknown flow rate vector/array for each pipe section above becomes,

$$q = \begin{bmatrix} q(1) \\ q(2) \\ q(3) \\ q(4) \\ q(5) \\ q(6) \\ q(7) \\ q(8) \\ q(9) \\ q(10) \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \\ q_{10} \end{bmatrix} = \begin{bmatrix} q_{01} \\ q_{12} \\ q_{14} \\ q_{23} \\ q_{24} \\ q_{35} \\ q_{45} \\ q_{46} \\ q_{57} \\ q_{67} \end{bmatrix}$$

According to Table 1 and Equations 5 and 6, we may use the same set of equations to evaluate friction factor and then pressure drops in pipes, depending on chosen correlation and flow regime. It is therefore much more convenient and efficient to write a Python function for evaluating friction factor and pressure drop. The sub-function should include the following equations:

1. $\mathrm{Re} = \dfrac{4\rho q}{\pi\mu D}$
2. Proper correlation in Table 1 to evaluate $f[i]$
3. $\Delta P = 2\,f\rho v^2 \dfrac{L}{D} = 2\,f\rho\left(\dfrac{4q}{\pi D^2}\right)^2 \dfrac{L}{D}$

Once all the pressure drops are evaluated in terms of corresponding flow rates, the above mass and energy balance equations can be solved simultaneously using the ***scipy.fsolve*** function for the all the unknown flow rates.