

## 5.2 CALCULATION OF THE FLOW RATE IN A PIPELINE

### 5.2.1 Concepts Demonstrated

Application of the general mechanical energy balance for incompressible fluids, and calculation of flow rate in a pipeline for various pipe diameters and lengths.

### 5.2.2 Numerical Methods Utilized

Solution of a single nonlinear algebraic equation.

### 5.2.3 MATLAB Options and Functions Demonstrated

Use of the MATLAB *fzero* function for solving a nonlinear equation, logarithmic functions.

### 5.2.4 Problem Definition

Rework Problem 4.2 using MATLAB

- (a) Prepare a MATLAB program using the MATLAB output generated by POLYMATH for this problem creation. Incorporate the MATLAB *fzero* function to calculate the flow velocity. Determine the velocity for a pipeline with effective length of  $L = 1000$  ft and made of nominal 8-inch diameter schedule 40 commercial steel pipe. (Solution:  $v = 11.61$  ft/s)
- (b) Modify the MATLAB program from part (a) to become a function that will take the flow velocity ( $v$ ), pipe diameter ( $D$ ), the pipe length ( $L$ ) and the ambient temperature ( $T$ ) as input parameters and will return to the calling point the value of  $f(v)$  (as defined in Section 4.2.6a).
- (c) Calculate the flow velocity and flow rate for pipelines with effective length of  $L = 500, 1000, \dots, 10000$  ft and made of nominal 4, 5, 6 and 8-inch schedule 40 commercial steel pipe. Use the *fzero* function for solving the equations for the various cases and present the results in tabular form. Prepare plots of flow velocity versus  $D$  and  $L$  and flow rate versus  $D$  and  $L$ .

### 5.2.5 Solution

(a) The set of POLYMATH equations for this problem are shown in Table 4-3 of Problem 4.2. The option within POLYMATH to “Show MATLAB formatted problem in report” must be enabled in the POLYMATH Settings menu under Nonlinear Equations in order to obtain the MATLAB equation set. This option is accessible by opening the icon marked as: “Setup preferences and numerical parameters” as discussed in Problem 5.1. The MATLAB formatted equation set is then displayed in the POLYMATH report page at the end of the solution

results.

The resultant MATLAB code generated by POLYMATH should be copied into the MATLAB editor, and this is shown in Table 5-9 after the comments have been placed at the end of each line of code. This code includes a specified pipe length and a pipe diameter. The **OneNLEtemplate.m** file shown in Table 5-10 (and available also from POLYMATH's Help section) can be used to determine the flow velocity for the specified parameter values with the MATLAB *fzero* function for solving simultaneous nonlinear equations.

**Table 5-9** MATLAB Code Generated by POLYMATH for Problem 5.2 - File **P5-2a1.m**

Line	Command, %Comment
1	xguess = 10.5 ;
2	function fv = NLEfun(v);
3	T = 60; %Temperature (deg F)
4	epsilon = 0.00015; %Surface roughness of the pipe (ft)
5	rho = 62.122 + T * (0.0122 + T * (-1.54e-4 + T * (2.65e-7 - T * 2.24e-10)));%Fluid density (lb/cu. ft.)
6	deltaz = 300; %Elevation difference (ft)
7	deltaP = -150; %Pressure difference (psi)
8	vis = exp(-11.0318 + 1057.51 / (T + 214.624)); %Fluid viscosity (lbm/ft-s)
9	L = 1000; %Effective length of pipe (ft)
10	D = 7.981 / 12; %Inside diameter of pipe (ft)
11	pi = 3.1416; %The constant pi
12	eoD = epsilon / D; %Pipe roughness to diameter ratio (dimensionless)
13	Re = D * v * rho / vis; %Reynolds number (dimensionless)
14	if (Re < 2100) %Fanning friction factor (dimensionless)
15	fF = (16 / Re) ;
16	else
17	fF = (1 / (16 * (log10(eoD / 3.7 - 5.02 * log10(eoD / 3.7 + 14.5 / Re) / Re) ) ^ 2));
18	end
19	q = v * pi * D ^ 2 / 4; %Flow rate (sq. ft./s)
20	gpm = q * 7.481 * 60; %Flow rate (gpm)
21	fv = v - sqrt((32.174 * deltaz + deltaP * 144 * 32.174 / rho) / (0.5 - 2 * fF * L / D));%Flow velocity (ft/s)

**Table 5-10** Template for Solving One Nonlinear Algebraic Equation - File **OneNLEtemplate.m**

Line	Command, %Comment
1	function % Insert here your file name after function (Use Alphanumeric name only)
2	clear, clc, format short g, format compact
3	xguess= % Replace this line with the xguess line(s) from Polymath report.
4	disp('Variable values at the initial estimate');
5	disp([' Unknown value ' num2str(xguess) ' Function Value ' num2str(NLEfun(xguess))]);
6	xsolv=fzero(@NLEfun,xguess);
7	disp(' Variable values at the solution');
8	disp([' Unknown value ' num2str(xsolv) ' Function Value ' num2str(NLEfun(xsolv))]);
9	%-----
10	% Replace this and the following line with the function copied from the Polymath report
11	% Do not include the xguess line(s)

The creation of the MATLAB program requires that lines 1-11 of the template shown in Table 5-10 be saved as a new MATLAB m-file. The complete function NLEfun should be copied from the POLYMATH report (shown in Lines 2-21 of Table 5-9) and pasted into the same new m-file (replacing the comments in lines 10-11). The initial guess for the unknown should be copied from line 1 of the POLYMATH report and pasted into the m-file where indicated (line 3 in Table 5-10). An arbitrary function name, **Prob\_5\_2a2.m** for this example, should be assigned for the m-file, and it should be inserted in line 1. The same name, **Prob\_5\_2a2.m**, should be used to store the file. The resulting completed m-file is given in Table 5-11.

**Table 5-11** MATLAB Code for Problem 5.2(a) - File **Prob\_5\_2a2.m**

Line	Command, %Comment
1	function Prob_5_2a2
2	clear, clc, format short g, format compact
3	xguess = 10.5 ;
4	disp('Variable values at the initial estimate');
5	disp([' Unknown value ' num2str(xguess) ' Function Value ' num2str(NLEfun(xguess))]);
6	xsolv=fzero(@NLEfun,xguess);
7	disp(' Variable values at the solution');
8	disp([' Unknown value ' num2str(xsolv) ' Function Value ' num2str(NLEfun(xsolv))]);
9	%-----
10	function fv = NLEfun(v);
11	T = 60; %Temperature (deg F)
12	epsilon = 0.00015; %Surface roughness of the pipe (ft)
13	rho = 62.122 + T * (0.0122 + T * (-1.54e-4 + T * (2.65e-7 - T * 2.24e-10)));%Fluid density (lb/cu. ft.)
14	deltaz = 300; %Elevation difference (ft)
15	deltaP = -150; %Pressure difference (psi)
16	vis = exp(-11.0318 + 1057.51 / (T + 214.624)); %Fluid viscosity (lbm/ft-s)
17	L = 1000; %Effective length of pipe (ft)
18	D = 7.981 / 12; %Inside diameter of pipe (ft)
19	pi = 3.1416; %The constant pi
20	eoD = epsilon / D; %Pipe roughness to diameter ratio (dimensionless)
21	Re = D * v * rho / vis; %Reynolds number (dimensionless)
22	if (Re < 2100) %Fanning friction factor (dimensionless)
23	fF = (16 / Re) ;
24	else
25	fF = (1 / (16 * (log10(eoD / 3.7 - 5.02 * log10(eoD / 3.7 + 14.5 / Re)) / Re)) ^ 2));
26	end
27	q = v * pi * D ^ 2 / 4 * 7.481 * 60; %Flow rate (gpm)
28	fv = v - sqrt((32.174 * deltaz + deltaP * 144 * 32.174 / rho) / (0.5 - 2 * fF * L / D));%Flow velocity (ft/s)

The MATLAB provided function *fzero* is used to solve the nonlinear equation. This function can be called with several different sets of input and output parameters. The particular form used in Table 5-10 is

$$xsolv=fzero(@NLEfun,xguess);$$

The first input parameter is the name of the function (*NLEfun*) while the second input parameter (*xguess*) is an initial estimate for the value of the unknown at the solution. The parameter returned by the *fzero* function is the value of *v* at the solution (*xsolv*). Figure 5-7 shows the result of running this MATLAB code.

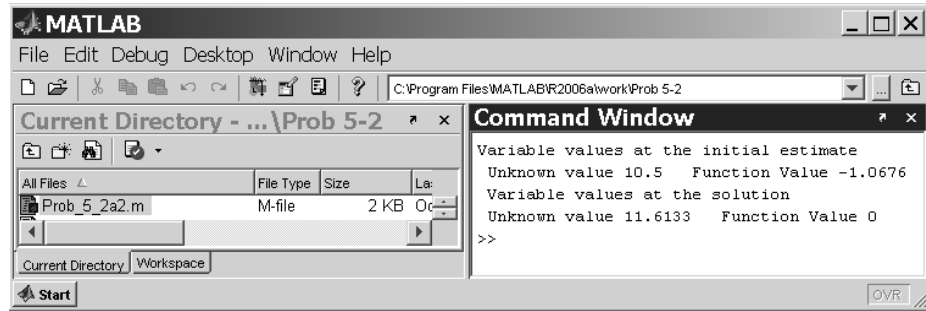


Figure 5-7 MATLAB Solution to Problem 5-2(a)

The calculated value of the flow velocity *v* is the same as the value provided in the problem statement.

(b) The MATLAB code can be modified so that it obtains the flow velocity *v* pipe diameter *D*, the pipe length *L* and the ambient temperature *T* as input parameters. Thus the following changes can be made in the MATLAB file **Prob\_5\_2a2m**:

1. The calculation of the flow rate *q* (in line 27) can be removed as this variable will have to be calculated outside of the function.
2. The definition of *pi* (shown in line 19 of Table 5-11) can be removed as MATLAB recognizes *pi* as a special value ( $pi = \pi$ ).
3. The definition of *D*, *L* and *T* (in lines 18, 17 and 11 of Table 5-11) must be removed as those variables serve as function parameters.
4. Lines 1 through 9 should be removed.
5. The variables *D*, *L* and *T* must be added as input parameters to the function statement in line 1: *function fv = NLEfun(v,D,L,T)*

This MATLAB code should be saved as **NLEfun.m**. The final form of the function is shown in Table 5-12.

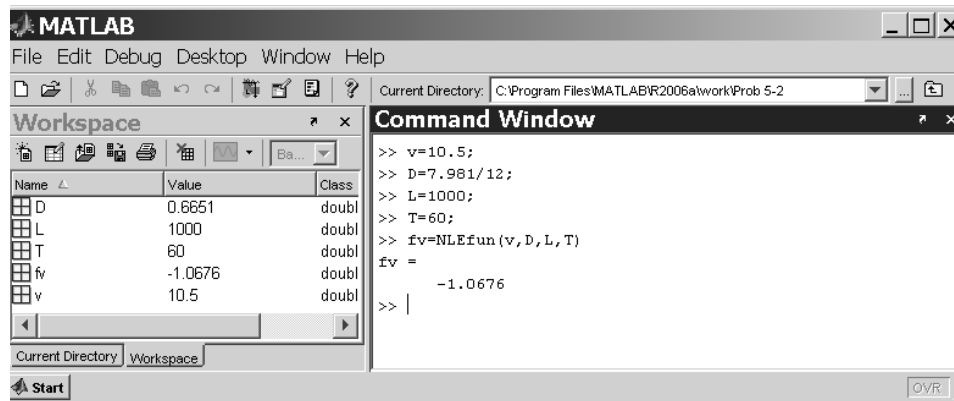
Table 5-12 MATLAB Function for Introducing Input Parameters - File **NLEfun.m**

Line	Command, %Comment
1	function fv = NLEfun(v,D,L,T)
2	epsilon = 0.00015;%Surface roughness of the pipe (ft)
3	rho = 62.122 + T * (0.0122 + T * (-0.000154 + T * (0.000000265 - (T * 0.000000000224))))); %Fluid density (lb/cu. ft.)
4	deltaz = 300; %Elevation difference (ft)
5	deltaP = -150; %Pressure difference (psi)
6	vis = exp(-11.0318 + 1057.51 / (T + 214.624)); %Fluid viscosity (lbm/ft-s)
7	pi = 3.1416; %The constant pi

**Table 5–12** MATLAB Function for Introducing Input Parameters - File **NLEfun.m**

Line	Command, %Comment
8	eoD = epsilon / D; %Pipe roughness to diameter ratio (dimensionless)
9	Re = D * v * rho / vis; %Reynolds number (dimensionless)
10	if (Re < 2100) %Fanning friction factor (dimensionless)
11	fF = 16 / Re;
12	else
13	fF = 1 / (16 * log10(eoD / 3.7 - (5.02 * log10(eoD / 3.7 + 14.5 / Re) / Re)) ^ 2);
14	end
15	fv = v - sqrt((32.174 * deltaZ + deltaP * 144 * 32.174 / rho) / (0.5 - (2 * fF * L / D))); %Flow velocity (ft/s)

This function can be used in MATLAB to make the calculation of  $f_v$  after  $v$ ,  $D$ ,  $L$ , and  $T$  have been specified. This is shown in Figure 5–8 when  $v = 10.5$ ,  $D = 7.981 / 12$ ,  $L = 1000$ , and  $T = 60$  are entered in the MATLAB Command Window. The result should and do agree with that in Figure 5–7 where  $f_v = -1.0676$ .

**Figure 5–8** MATLAB Solution to Problem 5-2(b)

(c) Repeated calculations with different  $L$  and  $D$  values and presentation of the results in a tabular form with plotting require the preparation of a separate MATLAB script file. A representative MATLAB file is shown in Table 5–13.

**Table 5–13** MATLAB *script* file for Calculating  $v$  and  $q$  Values and Presenting the Results in Tabular and Graphical Forms - File **Prob\_5\_2c.m**

Line	Command, %Comment
1	function Prob_5_2c
2	clear, clc, format short g, format compact
3	D_list=[4.026/12 5.047/12 6.065/12 7.981/12]; % Inside diameter of pipe (ft)
4	T = 60; %Temperature (deg. F)
5	for i = 1:4
6	D = D_list(i);
7	j=0;

**Table 5-13** MATLAB *script* file for Calculating  $v$  and  $q$  Values and Presenting the Results in Tabular and Graphical Forms - File **Prob\_5\_2c.m**

8	for L=500:500:10000
9	j = j+1;
10	L_list(j)=L; % Effective length of pipe (ft)
11	[v(j,i),fval]=fzero(@NLEfun,[1 20],[],D,L,T);
12	if abs(fval)>1e-10
13	disp([' No Convergence for L = ' num2str(L) ' and D = ' num2str(D)]);
14	end
15	q(j,i) = v(j,i) * pi * D ^ 2 / 4 * 7.481 * 60; %Flow rate (gpm)
16	end
17	end
18	disp('          Flow Velocity (ft/s) versus Pipe Length and Diameter');
19	disp('                          Tabular Results');
20	disp('');
21	disp('    L\D    D=4"    D=5"    D=6"    D=8"');
22	Res=[L_list' v];
23	disp(Res);
24	plot(L_list,v(:,1),'-',L_list,v(:,2),'+',L_list,v(:,3),'*',L_list,v(:,4),'x');
25	legend(' D=4"', ' D=5"', ' D=6"', ' D=8"');
26	title(' Flow Velocity')
27	xlabel('Pipe Length (ft)');
28	ylabel('Velocity (ft/s)');
29	pause
30	disp('          Flow Rate (gpm) versus Pipe Length and Diameter');
31	disp('                          Tabular Results');
32	disp('');
33	disp('    L\D    D=4"    D=5"    D=6"    D=8"');
34	Res=[L_list' q(:,1) q(:,2) q(:,3) q(:,4)];
35	disp(Res);
36	plot(L_list,q(:,1),'-',L_list,q(:,2),'+',L_list,q(:,3),'*',L_list,q(:,4),'x');
37	legend(' D=4"', ' D=5"', ' D=6"', ' D=8"');
38	title(' Flow rate')
39	xlabel('Pipe Length (ft)');
40	ylabel('Flow rate (gpm)');

Line 3 of the script file contains a row vector ( $D\_list$ ) which contains the inside diameters of the pipes (in ft) to be calculated. The ambient temperature is defined in line 4.

The flow velocity and flow rate values are calculated inside two nested *for* loops in lines 5 through 17. The variable  $i$  obtains the values 1, 2, 3 and 4; the current value of  $D$  is set as the  $i$ -th element of the vector  $D\_list$ . For every value of  $i$  the variable  $L$  obtains the values 500, 1000 ... 10000, the counter  $j$  is increased by one and  $L$  is set as the  $j$ -th element of the vector  $L\_list$ . The function *fzero* is called for all the  $D$  and  $L$  values. The resulting  $v$  values are stored in two matrices where every row is associated with one  $L$  value and every column is associated with a  $D$  value (see line 11 in Table 5-13).

Some of the parameters that are used in calling the *fzero* function are dif-

ferent here than in Table 5-10. The second input parameter is a vector which contains the bounds of an interval in which  $f(v)$  changes sign (an error message displayed if this is not true). Instead of the third (optional) parameter the space saver null vector: `[]` is entered. This parameter is a *structure* that contains information that controls the solution algorithm (error tolerances, max. number of iterations allowed etc.). If the default values are used this parameter can be replaced by a null matrix). The additional input parameters  $D$ ,  $L$  and  $T$  contain values that have to be transferred to the *NLEfun* function.

The parameters returned by the *fzero* function include the value of  $v$  at the solution and the value of  $f(v)$  (*fval*, an optional parameter) at the solution. In lines 11 through 13, a warning is issued in case where the  $f(v)$  value is not close enough to zero at the solution ( $f(v) \geq 10^{-10}$ ). The flow rates (*gpm*) are calculated in line 15 and stored in the matrix *q* which has the same structure as the *v* matrix.

In lines 18 through 21 the title and the column headings of the table are displayed. In line 22 the transposed *L\_list* vector is attached to the *v* matrix as an additional column and the combined matrix (*Res*) is displayed in line 23.

Parts of the results obtained and displayed by this program are shown in Table 5-14. The results with MATLAB are exactly the same as obtained by Excel (see Problem 4.2).

**Table 5-14** Partial Results for Flow Velocity Calculations for Various  $D$  and  $L$  Values - File **Prob\_5\_2c.m**

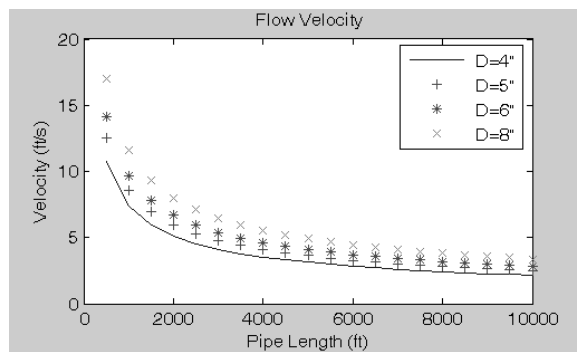
Flow velocity (ft/s) versus Pipe Length and Diameter				
Tabular Results				
L\D	D=4"	D=5"	D=6"	D=8"
500	10.773	12.516	14.15	17.035
1000	7.4207	8.6048	9.7032	11.613
1500	5.9721	6.9243	7.8051	9.3295
2000	5.1188	5.9361	6.6912	7.9953
2500	4.5409	5.2674	5.9382	7.0953

The commands used for plotting the flow velocity versus the pipe length and diameter are shown in lines 24 through 28 of Table 5-13. The pipe length is put on the horizontal axis ( $x$  - axis), the flow velocity on the vertical axis ( $y$  - axis) and the pipe diameter is a parameter.

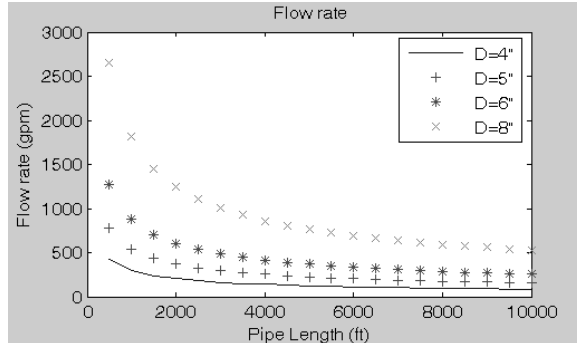
In line 24 of the MATLAB code, separate curves of  $v$  versus  $L$  are being plotted for the different  $D$  values. In the first triplet, the *L\_list* vector is put on the  $x$  - axis, the first column of the *v* - vector (corresponding to the first  $D$  value) is put on the  $y$  axis and the string '-' indicates that solid line should be used to plot this curve. There are four such triplets for the four  $V$  values. The commands *legend*, *title*, *xlabel* and *ylabel* are self explanatory. The plot obtained using this set of commands is shown in Figure 5-9.

The pause command in line 29 causes the program to wait for a key-press

before continuing to display another table and plot. The table and the plot for flow rates are obtained similarly. The plot of the flow rate versus pipe length and diameter is shown in Figure 5-10. Note that both the program file **Prob\_5\_2c.m** and the function file **NLEfun.m** must be in the MATLAB Workspace when this problem is solved.



**Figure 5-9** Flow Velocity versus Pipe Length and Diameter



**Figure 5-10** Flow Rate versus Pipe Length and Diameter



The problem solution files are found in directory CHAPTER 5 and designated **NLEfun.m**, **OneNLEtemplate.m**, **P5-2A.POL**, **Prob\_5\_2a1.m**, **Prob\_5\_2a2.m**, **Prob\_5\_2b.m**, and **Prob\_5\_2c.m**.