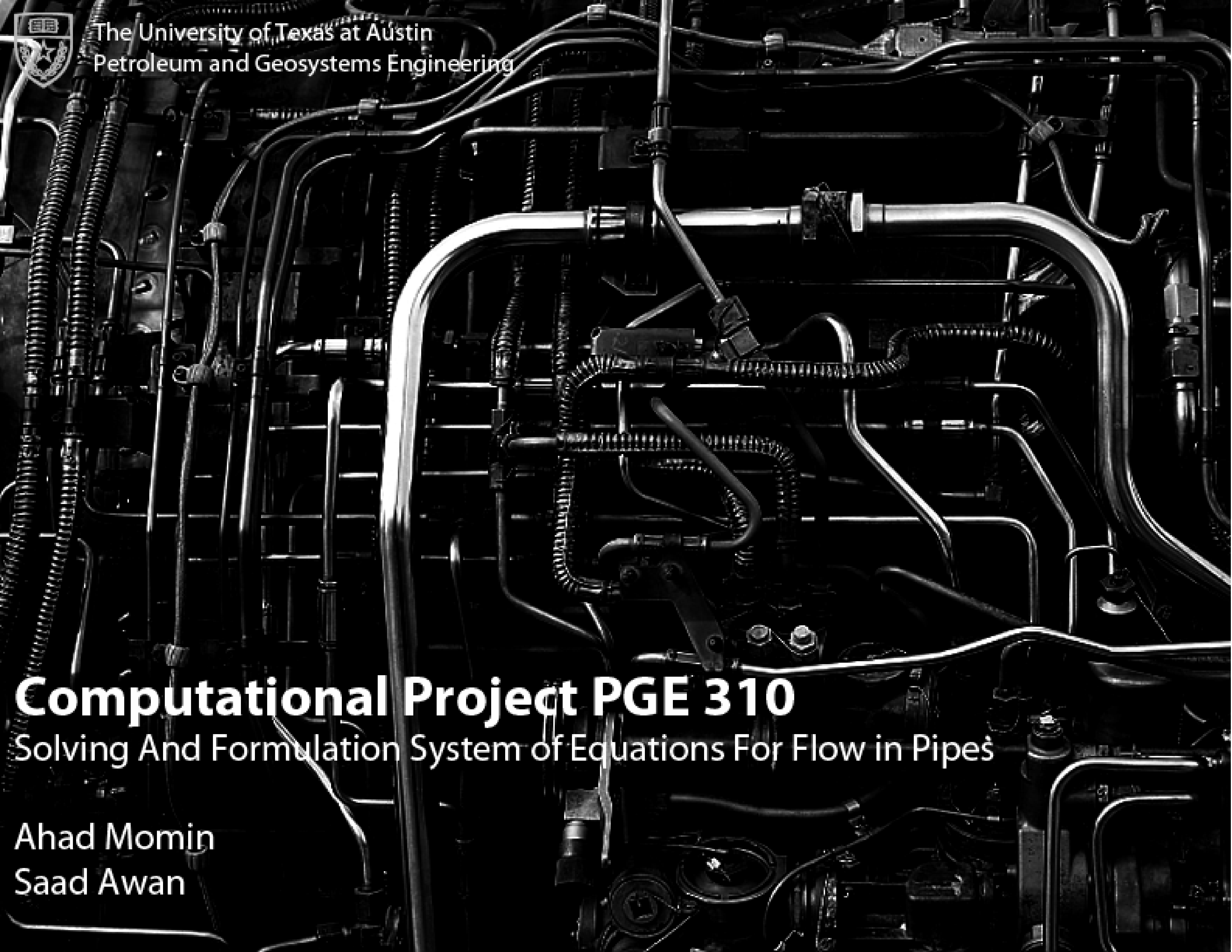


The University of Texas at Austin
Petroleum and Geosystems Engineering

Computational Project PGE 310

Solving And Formulation System of Equations For Flow in Pipes

Ahad Momin
Saad Awan

Prepared by:
+Infinite Loop+
(Saad Awan, Ahad Momin)
PGE 310 Spring 2017
Professor: Masha Prodanovic
Teaching Assistant: Ningyu Wang

Section 1

$$\Delta P = \frac{8 \rho L Q^2 \lambda}{\pi^2 D^5}$$

$$\boxed{\lambda = \frac{\Delta P \pi^2 D^5}{8 \rho L Q^2}} \quad (1)$$

$$Re = \frac{\rho V D}{\mu} = \frac{V.D}{\nu} = \frac{4.|Q|}{\pi.\nu.D} \quad (2)$$

$$\frac{1}{\sqrt{\lambda}} = -2\log_{10}\left(\frac{2.51}{Re.\sqrt{\lambda}} + \frac{\varepsilon}{3.71.D}\right) \quad (3)$$

$$1 \div \sqrt{\frac{\Delta P \pi^2 D^5}{8 \rho L Q^2}} = -2\log_{10}\left(2.51 \div \frac{4Q\sqrt{\Delta P \pi^2 D^5}}{\pi \nu D \sqrt{8 \rho L Q^2}} + \frac{\varepsilon}{3.71D}\right)$$

$$\frac{2Q\sqrt{\rho L.2}}{\pi D^2 \sqrt{D \Delta P}} = 2\log_{10}\left(\frac{2.51}{2D}\frac{\nu\sqrt{2\rho L}}{\sqrt{D\Delta P}} + \frac{\varepsilon}{3.71D}\right)$$

$$-\frac{Q\sqrt{\rho L.2}}{\pi D^2 \sqrt{D \Delta P}} = \log_{10}\left(\frac{2.51}{2D}\frac{\nu\sqrt{2\rho L}}{\sqrt{D\Delta P}} + \frac{\varepsilon}{3.71D}\right)$$

$$\boxed{Q = -\frac{D^2.\pi.\sqrt{D.\Delta P}}{\sqrt{2.\rho.L}} . \log_{10}\left(\frac{2.51.\nu\sqrt{2\rho L}}{2D.\sqrt{D.\Delta P}} + \frac{\varepsilon}{3.71.D}\right)}$$

Section 3

Node 1

$$-Q_1 - Q_9 + Q_{1\text{-input}} = 0$$

Node 2

$$Q_1 - Q_2 - Q_{15} = 0$$

Node 3

$$Q_2 - Q_3 - Q_{17} = 0$$

Node 4

$$Q_3 - Q_{19} - Q_4 = 0.0315$$

Node 5

$$Q_4 + 0.0351 - Q_{13} - Q_5 = 0$$

Node 6

$$Q_5 - Q_6 - Q_{18} - 0.0252 = 0$$

Node 7

$$Q_6 + Q_7 - Q_{16} = 0$$

Node 8

$$Q_{14} + Q_8 - Q_7 = 0$$

Node 9

$$Q_9 - Q_{10} - Q_8 - 0.0379 = 0$$

Node 10

$$Q_{15} + Q_{10} - Q_{11} - Q_{14} = 0$$

Node 11

$$Q_{12} + Q_{17} + Q_{11} + Q_{16} - 0.0442 = 0$$

loop 1

$$-\Delta P_1 + \Delta P_9 + \Delta P_{10} - \Delta P_{15} = 0$$

loop 2

$$\Delta P_8 - \Delta P_{10} - \Delta P_{14} = 0$$

loop 3

$$-\Delta P_2 + \Delta P_{11} + \Delta P_{15} - \Delta P_{17} = 0$$

loop 4

$$\Delta P_7 - \Delta P_{10} + \Delta P_{14} + \Delta P_{16} = 0$$

loop 5

$$-\Delta P_3 - \Delta P_{12} + \Delta P_{17} - \Delta P_9 = 0$$

loop 6

$$-\Delta P_6 + \Delta P_{12} - \Delta P_{16} + \Delta P_{18} = 0$$

loop 7

$$-\Delta P_4 - \Delta P_{13} + \Delta P_{19} = 0$$

8 loop 8

$$-\Delta P_5 + \Delta P_{13} - \Delta P_{18} = 0$$

Code 1: comp_1

```
% we will create matrix A, that will represent df/dx

% but first we will need to initialize the matrix A

A = zeros(38);

% the first 11 equations are for Q at the 11 nodes

A(1,:)=[-1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(2,:)=[1 -1 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(3,:)=[0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(4,:)=[0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(5,:)=[0 0 0 1 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(6,:)=[0 0 0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(7,:)=[0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(8,:)=[0 0 0 0 0 0 -1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(9,:)=[0 0 0 0 0 0 0 -1 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(10,:)=[0 0 0 0 0 0 0 0 0 1 -1 0 0 -1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(11,:)=[0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

% the next 8 equations represent delta p at 8 loops

A(12,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 -1 0 0 0 0];
A(13,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 -1 0 0 0 -1 0 1 0 0 0];
A(14,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 1 0 0 1 0 -1 0 0 0];
A(15,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 -1 0 0 1 0 1 0 0 0];
A(16,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 -1];
A(17,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 1 0 0 0 -1 0 1 0];
A(18,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 1];
A(19,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 1 0 0 0 0 -1 0];

% now we will put random values for the non linear equations

% the Q values are correct, but the delta p is just an initialization for further operations

A(20,:)=[-1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(21,:)=[0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(22,:)=[0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(23,:)=[0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
```

```
A(24,:)=[0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(25,:)=[0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0];
A(26,:)=[0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0];
A(27,:)=[0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0];
A(28,:)=[0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0];
A(29,:)=[0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0];
A(30,:)=[0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0];
A(31,:)=[0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0];
A(32,:)=[0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0];
A(33,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0];
A(34,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0];
A(35,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0];
A(36,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0];
A(37,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0];
A(38,:)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];

% now we will create a f matrix such that f = df/dx * dx we will call this
% matrix b

b = [.1037;0;0;-.0315;.0351;-.0252;0;0;-.0379;0;.0442;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];

% the aim of this project is to find dx i.e values for Q and delta P
% however first we will need to operate on our A matrix as our df/dp values
% are incorrect

% the following data has been provided in the project

% vector d represents the diameter of every pipe
d =
[0.305;0.203;0.203;0.203;0.203;0.203;0.203;0.203;0.203;0.152;0.152;0.152;0.152;0.152;0.152;0.152;0.254;0.152;0.152
];

% vector l represents the length
l =
[457.2;304.8;365.8;609.6;853.4;335.3;304.8;762.0;243.8;396.2;304.8;335.3;304.8;548.6;335.3;548.6;365.9;548.6;396.2
];

% vector Qh represents the guess values for the flow rate
Qh = [173.32;150;130;6.6;100;.28;16.88;13.56;200;50;70;51.96;32.96;3.32;23.32;17.16;20;9;10];

% vector Q is the flow rate in cubic meters per second
Q = Qh/3600;

% vector p is the initial guess for delta p given in the prompt
p = [1342;5324;4855;35;6867;.1;92;156;7420;3693;5387;260;103;38;754;702;50;219;191];x1 = zeros(19,1);
```

```
% vector f is the initialization of equation 4 in the prompt
f = zeros(19,1);

% vector df represents the partial derivative of f wrt to delta p
df = zeros(19,1);

% now we will find accurate values of f and df by iteration processes
for i = 1:19
    [p(i),f(i),df(i),iter(i)]=newtonraphson(p(i),Qh(i),d(i),l(i));
end

% now we will plug in the values of df into our A matrix
for i = 20:38
    A(i,i) = df(i-19);
end

% now our A matrix is complete and ready for further calculation

% we will preform A\b to get values of dx

dx = A\b;

% the first 19 values of dx represents the of flow rate
% the next 19 values of dx represent values of delta p
for i = 1:19
    Q(i) = dx(i)*3600; % we convert the flow rate into cubic feet per hour
    p(i) = dx(i+19);
end
```

Function 1: G

```
function [ f,df ] = g( p,Qh,d,l )

% the inputs are as follows
% p:  guess value of pressure
% Qh:   guess value of flow rate,
% d:    diameter of pipe
% l:    length of pipe
Q = Qh/3600; % converting flow rate into cubic meters per second
% f:    rearranging equation 4 so that Q is on the rhs
f = (((-d^2)*pi*sqrt(abs(d*p)))/sqrt(2000*l))*log10(((2.5193e-
6)*sqrt(2000*l))/(2*d*sqrt(abs(d*p)))+.00026/(3.71*d))-Q;
% df:   this represents the partial derivative df/d(delta_p)
df = -.07025*(d^2.5)*((-1e-5)*sqrt(l)*d + 0.5*log10(((6e-5)*sqrt(l))/((d^1.5)*sqrt(abs(p))) + (7e-5)/d)*((6e-
5)*sqrt(l)*d + (7e-5)*d^(1.5)*sqrt(abs(p)))/(sqrt(abs(l*p))*((6e-5)*sqrt(l)*d +(7e-5)*sqrt(abs(p*d^3)))));
end
```

Function 2: newtonraphson

```
function [ p,f,df ] = newtonraphson(p_estimate,Qh,d,l)

% the function serves the function of using estimates provided by the prompt to provide accurate values of
pressure, f and df/dv corresponding to the given Q values

% p_estimate:  guess value of pressure
% Qh:          guess value of flow rate,
% d:           diameter of pipe
% l:           length of pipe
[f,df] = g(p_estimate,Qh,d,l);% the parameters are sent to function g
% Newton Raphson method applied
p = p_estimate-(f/df);
error = 0;
while abs(p_estimate - p) > error && iter < 100
    p_estimate = p;
    [f,df] = g(p,Qh,d,l);
    p = p_estimate - (f/df);
    error = (norm(p)-norm(p_estimate))/norm(p);
end
end
```

Results

```
comp_1

p % pressure in every pipe in pascals

p =

   1.0e+04 *

   -1.6150
   -2.8830
   -3.4897
    0.4277
    0.7535
    0.0621
   -0.7869
   -0.5371
   -5.4070
    2.7893
    0.4434
    4.8642
   -1.7702
   -1.0481
   -4.2327
    2.2784
    0.0319
   -2.5238
   -1.3426


Q % Flow rate in every pipe in cubic meters per hour


Q =

 -223.7527
  -84.4328
  -98.4608
   92.3783
   13.0314
  219.2445
 -158.4042
```

```
  -52.8895
 -149.5673
   39.7622
    5.9570
   78.2947
  -47.0131
 -105.5147
 -139.3199
   60.8403
   14.0281
 -115.4931
  -77.4391

iter % iterations required for every estimate

iter =

    13
   100
    12
   100
   100
     1
    15
    14
   100
   100
    13
    10
    11
    18
   100
   100
    15
    15
   100
```

The End

INDIVIDUAL CONTRIBUTION FORM

Team: **Infinite Loop**

Member: **Ahad Momin**

Signature:

Briefly Describe in your own words what is the main objective of the computational assignment, and what did you learn in PGE 310 that enabled you to solve the problem (200 words or less).

It was a great deal of work and one of the hardest project that I have ever faced. It involved deep thinking, a lot of work and a lot of time. However, I learned how to work as a team and now that I have completed it, I feel as high as a satellite.

Briefly describe in your own words how you individually contributed to the program (200 words or less)

I derived equations and executed iteration method necessary for the project. Also, I found derivative and values for non linear equation which my partner transformed into a matrix.
I also executed the debugging process.

Below list yourself and group members' names along with the percentage you feel each member contributed effort to the project. For example, if there are 2 members including you and everyone contributed equally, then write 50%.

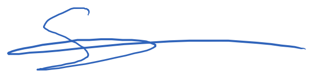
	Name	Percentage
1.	Saad Awan	54%
2.	Ahad Momin	50%

Total = 100%

INDIVIDUAL CONTRIBUTION FORM

Team: **Infinite Loop**

Member: **Saad Awan**

Signature: 

Briefly Describe in your own words what is the main objective of the computational assignment, and what did you learn in PGE 310 that enabled you to solve the problem (200 words or less).

It gave me a sense of how to tackle real life problems and provided me with an outlook of how things are beyond the classroom. Needless to say, this was the most difficult assignment I have had since coming to college, but now that I have completed it, I feel a joy that is difficult to express.

Briefly describe in your own words how you individually contributed to the program (200 words or less)

I coded out everything that my partner had handwritten down, for example the equations and derivations etc.

I also executed the tedious debugging process

Below list yourself and group members' names along with the percentage you feel each member contributed effort to the project. For example, if there are 2 members including you and everyone contributed equally, then write 50%.

	Name	Percentage
1.	Saad Awan	50 %
2.	Ahad Momin	50 %

		Total = 100%