Scilab Textbook Companion for Fluid Mechanics by J. F. Douglas[1]

Created by
Jay Chakra
B.Tech
Others
IIT Bombay
College Teacher
Madhu Belur
Cross-Checked by
Mukul R. Kulkarni

August 9, 2013

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Fluid Mechanics

Author: J. F. Douglas

Publisher: Pearson Prentice Hall

Edition: 5

Year: 2005

ISBN: 9780131292932

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes 5

1 Fluids and their properties 11

2 Pressure and Head 12

3 Static Forces on Surfaces Buoyancy 19

4 Motion of Fluid Particles and Streams 26

5 The Momentum Equation and its Applications 30

6 The Energy Equation and its Applications 38

7 Two dimensional Ideal Flow 46

8 Dimensional Analysis 50

9 Similarity 51

10 Laminar and Turbulent Flows in Bounded Systems 56

11 Boundary Layer 61

12 Incompressible Flow around a Body 63

13 Compressible Flow around a Body 68

14 Steady Incompressible Flow in Pipe and Duct Systems 71

15 Uniform Flow in Open Channels **81**

16 Non uniform Flow in Open Channels **83**

17 Compressible Flow in Pipes **85**

20 Pressure Transient Theory and Surge Control **89**

22 Theory of Rotodynamic Machines **91**

23 Performance of Rotodynamic Machines **94**

24 Positive Displacement Machines **100**

25 Machine Network Interactions **104**

List of Scilab Codes

Exa 1.1	Excess Pressure	11
Exa 2.1	VARIATION OF PRESSURE VERTICALLY	12
Exa 2.2	VARIATION OF PRESSURE WITH ALTITUDE . .	12
Exa 2.3	VARIATION OF PRESSURE WITH ALTITUDE IN A GAS UNDER ADIABATIC CONDITIONS	13
Exa 2.4	VARIATION OF PRESSURE AND DENSITY WITH ALTITUDE FOR A CONSTANT TEMPERATURE GRADIENT	14
Exa 2.5	PRESSURE AND HEAD	14
Exa 2.6	PRESSURE MEASUREMENT BY MANOMETER .	15
Exa 2.7	PRESSURE MEASUREMENT BY MANOMETER .	15
Exa 2.8	PRESSURE MEASUREMENT BY MANOMETER .	16
Exa 2.9	PRESSURE MEASUREMENT BY MANOMETER .	17
Exa 2.10	RELATIVE EQUILIBRIUM	17
Exa 3.1	RESULTANT FORCE AND CENTRE OF PRESSURE ON A PLANE SURFACE UNDER UNIFORM PRESSURE	19
Exa 3.2	RESULTANT FORCE AND CENTRE OF PRESSURE ON A PLANE SURFACE UNDER UNIFORM PRESSURE	20
Exa 3.3	PRESSURE DIAGRAMS	21
Exa 3.4	FORCE ON A CURVED SURFACE DUE TO HYDROSTATIC PRESSURE	21
Exa 3.5	BUOYANCY	22
Exa 3.6	DETERMINATION OF THE POSITION OF THE METACENTRE RELATIVE TO THE CENTRE OF BUOYANCY	23

Exa 3.7	STABILITY OF A VESSEL CARRYING LIQUID IN TANKS WITH A FREE SURFACE	24
Exa 4.1	DISCHARGE AND MEAN VELOCITY	26
Exa 4.2	CONTINUITY OF FLOW	28
Exa 4.3	CONTINUITY EQUATIONS FOR THREE DIMENSIONAL FLOW USING CARTESIAN COORDINATES	28
Exa 5.1	GRADUAL ACCELERATION OF A FLUID IN A PIPELINE NEGLECTING ELASTICITY	30
Exa 5.2	FORCE EXERTED BY A JET STRIKING A FLAT PLATE	30
Exa 5.3	FORCE DUE TO THE DEFLECTION OF A JET BY A CURVED VANE	32
Exa 5.4	FORCE EXERTED WHEN A JET IS DEFLECTED BY A MOVING CURVED VANE	32
Exa 5.5	FORCE EXERTED ON PIPE BENDS AND CLOSED CONDUITS	33
Exa 5.6	REACTION OF A JET	34
Exa 5.7	REACTION OF A JET	35
Exa 5.8	REACTION OF A JET	36
Exa 5.10	ANGULAR MOTION	36
Exa 6.1	MECHANICAL ENERGY OF A FLOWING FLUID	38
Exa 6.2	CHANGES OF PRESSURE IN A TAPERING PIPE	39
Exa 6.3	PRINCIPLE OF THE VENTURI METER	40
Exa 6.4	THEORY OF SMALL ORIFICES DISCHARGING TO ATMOSPHERE	40
Exa 6.5	THEORY OF LARGE ORIFICES	41
Exa 6.6	ELEMENTARY THEORY OF NOTCHES AND WEIRS	41
Exa 6.7	ELEMENTARY THEORY OF NOTCHES AND WEIRS	42
Exa 6.8	THE POWER OF A STREAM OF FLUID	43
Exa 6.9	VORTEX MOTION	43
Exa 6.10	Free vortex or potential vortex	44
Exa 7.2	STRAIGHT LINE FLOWS AND THEIR COMBINATIONS	46
Exa 7.3	COMBINED SOURCE AND SINK FLOWS DOUBLET	47
Exa 7.4	FLOW PAST A CYLINDER	48
Exa 7.5	CURVED FLOWS AND THEIR COMBINATIONS	48

Exa 8.1	CONVERSION BETWEEN SYSTEMS OF UNITS INCLUDING THE TREATMENT OF DIMENSIONAL CONSTANTS	50
Exa 9.1	MODEL STUDIES FOR FLOWS WITHOUT A FREE SURFACE	51
Exa 9.2	MODEL STUDIES FOR FLOWS WITHOUT A FREE SURFACE	51
Exa 9.3	ZONE OF DEPENDENCE OF MACH NUMBER	52
Exa 9.4	SIGNIFICANCE OF THE PRESSURE COEFFICIENT	52
Exa 9.5	MODEL STUDIES IN CASES INVOLVING FREE SURFACE FLOW	53
Exa 9.6	SIMILARITY APPLIED TO ROTODYNAMIC MACHINES	54
Exa 9.8	RIVER AND HARBOUR MODELS	54
Exa 10.1	INCOMPRESSIBLE STEADY AND UNIFORM LAMINAR FLOW BETWEEN PARALLEL PLATES	56
Exa 10.2	INCOMPRESSIBLE STEADY AND UNIFORM LAMINAR FLOW IN CIRCULAR CROSS SECTION PIPES	57
Exa 10.3	INCOMPRESSIBLE STEADY AND UNIFORM TURBULENT FLOW IN CIRCULAR CROSS SECTION PIPES	58
Exa 10.4	STEADY AND UNIFORM TURBULENT FLOW IN OPEN CHANNELS	59
Exa 10.5	VELOCITY DISTRIBUTION IN TURBULENT FULLY DEVELOPED PIPE FLOW	59
Exa 10.7	SEPARATION LOSSES IN PIPE FLOW	60
Exa 11.1	PROPERTIES OF THE LAMINAR BOUNDARY LAYER FORMED OVER A FLAT PLATE IN THE ABSENCE OF A PRESSURE GRADIENT IN THE FLOW DIRECTION	61
Exa 11.2	PROPERTIES OF THE TURBULENT BOUNDARY LAYER OVER A FLAT PLATE IN THE ABSENCE OF A PRESSURE GRADIENT IN THE FLOW DIRECTION	62
Exa 12.1	DRAG	63
Exa 12.2	RESISTANCE OF SHIPS	64
Exa 12.3	FLOW PAST A CYLINDER	64
Exa 12.4	FLOW PAST A SPHERE	65

Exa 12.5	FLOW PAST A SPHERE	66
Exa 12.6	FLOW PAST AN AEROFOIL OF FINITE LENGTH	66
Exa 13.1	EFFECTS OF COMPRESSIBILITY	68
Exa 13.2	SHOCK WAVES	69
Exa 14.1	INCOMPRESSIBLE FLOW THROUGH DUCTS AND PIPES	71
Exa 14.3	INCOMPRESSIBLE FLOW THROUGH PIPES IN PARALLEL	72
Exa 14.4	INCOMPRESSIBLE FLOW THROUGH BRANCHING PIPES THE THREE RESERVOIR PROBLEM	73
Exa 14.5	INCOMPRESSIBLE STEADY FLOW IN DUCT NETWORKS	74
Exa 14.6	INCOMPRESSIBLE STEADY FLOW IN DUCT NETWORKS	75
Exa 14.7	RESISTANCE COEFFICIENTS FOR PIPELINES IN SERIES AND IN PARALLEL	76
Exa 14.8	THE QUANTITY BALANCE METHOD FOR PIPE NETWORKS	77
Exa 14.9	QUASI STEADY FLOW	78
Exa 14.12	QUASI STEADY FLOW	79
Exa 15.1	RESISTANCE FORMULAE FOR STEADY UNIFORM FLOW IN OPEN CHANNELS	81
Exa 15.2	OPTIMUM SHAPE OF CROSS SECTION FOR UNIFORM FLOW IN OPEN CHANNELS	82
Exa 16.2	EFFECT OF LATERAL CONTRACTION OF A CHANNEL	83
Exa 16.3	CLASSIFICATION OF WATER SURFACE PROFILES	84
Exa 17.1	MASS FLOW THROUGH A VENTURI METER . .	85
Exa 17.2	THE LAVAL NOZZLE	86
Exa 17.3	NORMAL SHOCK WAVE IN A DIFFUSER	86
Exa 17.4	COMPRESSIBLE FLOW IN A DUCT WITH FRICTION UNDER ADIABATIC CONDITIONS FANNO FLOW	87
Exa 17.5	ISOTHERMAL FLOW OF A COMPRESSIBLE FLUID IN A PIPELINE	88
Exa 20.3	APPLICATION OF THE SIMPLIFIED EQUATIONS TO EXPLAIN PRESSURE TRANSIENT OSCILLATIONS	89

Exa 20.5 CONTROL OF SURGE FOLLOWING VALVE CLOSURE WITH PUMP RUNNING AND SURGE TANK APPLICATIONS 90
Exa 22.1 ONE DIMENSIONAL THEORY 91
Exa 22.2 DEPARTURES FROM EULERS THEORY AND LOSSES 92
Exa 22.3 COMPRESSIBLE FLOW THROUGH ROTODYNAMIC MACHINES 93
Exa 23.1 DIMENSIONLESS COEFFICIENTS AND SIMILARITY LAWS 94
Exa 23.2 THE PELTON WHEEL 96
Exa 23.3 FRANCIS TURBINES 96
Exa 23.4 AXIAL FLOW TURBINES 97
Exa 23.5 HYDRAULIC TRANSMISSIONS 98
Exa 24.1 RECIPROCATING PUMPS 100
Exa 24.2 RECIPROCATING PUMPS 102
Exa 25.1 FANS PUMPS AND FLUID NETWORKS 104
Exa 25.4 AN APPLICATION OF THE STEADY FLOW ENERGY EQUATION 106
Exa 25.7 JET FANS 107
Exa 25.8 JET FANS 108
Exa 25.9 CAVITATION IN PUMPS AND TURBINES 109
Exa 25.11 VENTILATION AND AIRBORNE CONTAMINATION AS A CRITERION FOR FAN SELECTION 110
AP 1 UDF Intersect Function 112

List of Figures

4.1 DISCHARGE AND MEAN VELOCITY 27

23.1 DIMENSIONLESS COEFFICIENTS AND SIMILARITY LAWS 95

25.1 FANS PUMPS AND FLUID NETWORKS 105

Chapter 1

Fluids and their properties

Scilab code Exa 1.1 Excess Pressure

```
funcprot(0); clc;
//Example 1.1
//Initialization of Variable
sigma = 72.7*10^-3;          //Surface Tension
r = 1 *10^-3;                //Radius of Bubble

//Calculations
P = 2*sigma/r;
disp(P,"Excess Pressure(N/m2) :")
```

Chapter 2

Pressure and Head

Scilab code Exa 2.1 VARIATION OF PRESSURE VERTICALLY

```
clc ;funcprot(0);
//Example 2.1

//Initializing the variables
z1 = 0; //Taking Ground as reference
z2 = -30;//Depth
rho = 1025;//Density
g = 9.81;//Acceleration due to gravity

//Calculation
pressureIncrease = -rho*g*(z2-z1);
disp(pressureIncrease/1000,"Increase in Pressure (KN
    /m2):");
```

Scilab code Exa 2.2 VARIATION OF PRESSURE WITH ALTITUDE

```
clc ;funcprot(0);
//Example 2.2
```

```

//Initializing the variables
p1 = 22.4*10^3;//Initial Pressure
z1 = 11000;//Initial Height
z2 = 15000;//final Height
g = 9.81;//Acceleration due to gravity
R = 287; //Gas Constant
T = 273-56.6;//Temperature

//Calculations
p2 = p1*exp(-g*(z2-z1)/(R*T));
rho2=p2/(R*T);
disp(rho2,"Final Density (kg/m3) :",p2/1000,"Final
    Pressure (kN/m2):");
```

Scilab code Exa 2.3 VARIATION OF PRESSURE WITH ALTITUDE IN A GAS UNDER ADIABATIC CONDITIONS

```
clc ;funcprot(0);
//Example 2.3

//Initializing the variables
p1 = 101*10^3;//Initial Pressure
z1 = 0;//Initial Height
z2 = 1200;//Final Height
T1 = 15+273;//Initial Temperature
g = 9.81;//Acceleration due to gravity
gamma = 1.4;//Heat capacity ratio
R = 287;//Gas Constant

//Calculations
p2 = p1*(1-g*(z2-z1)*(gamma-1)/(gamma*R*T1))^(gamma
    /(gamma-1));
dT_dZ = -(g/R)*((gamma-1)/gamma);
T2 = T1 + dT_dZ*(z2-z1);
```

```
disp(T2-273,"Density at 1200 m (in degree celcius) : ",p2/1000,"Final Pressure(kN/m2) :");
```

Scilab code Exa 2.4 VARIATION OF PRESSURE AND DENSITY WITH ALTITUDE FOR A CONSTANT TEMPERATURE GRADIENT

```
clc ;funcprot(0);
//Example 2.4

//Initializing the variables
p1 = 101*10^3;//Initial Pressure
z1 = 0;//Initial Height
z2 = 7000;//Final Height
T1 = 15+273;//Initial Temperature
g = 9.81;//Acceleration due to gravity
R = 287;//Gas Constant
dT = 6.5/1000;//Rate of Variation of Temperature

//Calculations
p2 = p1*(1-dT*(z2-z1)/T1)^(g/(R*dT));
T2 = T1 - dT*(z2-z1);
rho2 = p2/(R*T2);
disp(rho2,"Final Density (kg/m3 ):",p2/1000,"Final Pressure (kN/m2) :");
```

Scilab code Exa 2.5 PRESSURE AND HEAD

```
clc ;funcprot(0);
//Example 2.5

//Initializing the variables
p = 350*10^3;       //Gauge Pressure
pAtm = 101.3*10^3;   //Atmospheric Pressure
```

```
rhoW = 1000;          //Density of Water
sigma = 13.6;        //Relative Density of Mercury
g = 9.81;//Acceleration due to gravity

//Calculations
function[head]=Head(rho)
    head = p/(rho*g);
endfunction
rhoM = sigma*rhoW;
pAbs = p + pAtm;

disp(pAbs/1000,"Absolute pressure(kN/m2)",Head(rhoM)
   ,"Equivalent head of water (m):","Part (b)",Head(
   rhoW),"Equivalent head of water (m) :","Part (a)"
    );
```

Scilab code Exa 2.6 PRESSURE MEASUREMENT BY MANOMETER

```
clc ;funcprot(0);
//Example 2.6

//Initializing the variables
rho = 10^3;          //Density of water
h = 2;              //Height
g = 9.81;              //Acceleration due to gravity

//Calculations
p=rho*h*g;
disp(p/1000,"Gauge pressure (k/m2) :");
```

Scilab code Exa 2.7 PRESSURE MEASUREMENT BY MANOMETER

```
clc ;funcprot(0);
```

```
//Example 2.7

//Initializing the variables
rho = 0.8*10^3;    //Density of fluid
rhoM = 13.6*10^3;    //Density of manometer liquid
g = 9.81;//Acceleration due to gravity

//Calculations
function[P]=fluidPressure(h1,h2)
    P = rhoM*g*h2-rho*g*h1;
endfunction

disp(fluidPressure(0.1,-0.2)/1000,"Gauge pressure (
   kN/m2):","!-----Part (b)-----!",fluidPressure
   (0.5,0.9)/1000,"Gauge pressure (kN/m2):","!-----
   Part (a)-----!");
```

Scilab code Exa 2.8 PRESSURE MEASUREMENT BY MANOMETER

```
clc ;funcprot(0);
//Example 2.8

//Initializing the variables
rho = 10^3;    //Density of fluid
rhoM = 13.6*10^3;    //Density of manometer liquid
g = 9.81;    //Acceleration due to gravity
H = 0.3;    // Differnce in height = b-a as in text
h = 0.7;

//Calculations
result = rho*g*H + h*g*(rhoM-rho);
disp( result/1000,"Pressure difference (kN/m2):");
```

Scilab code Exa 2.9 PRESSURE MEASUREMENT BY MANOMETER

```
clc ;funcprot(0);
//Example 2.9

//Initializing the variables
rho = 10^3;     //Density of fluid
rhoM = 0.8*10^3;     //Density of manometer liquid
g = 9.81;     //Acceleration due to gravity
a = 0.25;
b = 0.15;
h = 0.3;
//Calculations
function[P]=PressureDiff(a,b,h,rho,rhoM)
    P = rho*g*(b-a) + h*g*(rho-rhoM);
endfunction

disp(PressureDiff(a,b,h,rho,rhoM),"Pressure Differnece(N/m2):","!-----Part (b)-----!",PressureDiff(a,b,h,rho,0)/1000,"Pressure Differnece(kN/m2):","!-----Part (a): rhoM is negligible assuming zero-----!");
```

Scilab code Exa 2.10 RELATIVE EQUILIBRIUM

```
clc ;funcprot(0);
//Example 2.10

//Initializing the variables
phi = 30; //30 degree
h = 1.2 ; // Height of tank
l = 2;     // Length of tank

//Calculations
function[Theta]=SurfaceAngle(a,phi)
```

```
    Theta = atand(-a*cosd(phi)/(g+a*sind(phi)));
endfunction

//case (a) a = 4
disp(tand(SurfaceAngle(4,phi)),"Tan of Angle between
    surface of fluid and horizontal :");
disp(180 + SurfaceAngle(4,phi),"ThetaA (degree):");

//Case (b) a = - 4.5
tanThetaR = tand(SurfaceAngle(-4.5,phi));
disp(tanThetaR,"Tan of Angle between surface of
   fluid and horizontal :");
disp(SurfaceAngle(-4.5,phi),"ThetaR (degree):");

Depth = h - l*tanThetaR/2;
disp(Depth,"Maximum Depth of tank (m):");
```

Chapter 3

Static Forces on Surfaces Buoyancy

Scilab code Exa 3.1 RESULTANT FORCE AND CENTRE OF PRESSURE ON A PLANE SURFACE UNDER UNIFORM PRESSURE

```
clc ;funcprot(0);
//Example 3.1

//Initializing the variables
a = 2.7;     //Upper edge
b = 1.2 ;    //Lower edge
width = 1.5;      //Width of trapezoidal plate
h = 1.1;          //Height of water column above
    surface
rho = 1000;
g = 9.81;//Acceleration due to gravity
phi = 90 ;//Angle between wall and surface

//Calculations
A = 0.5*(a+b)*width;         //Area of Trapezoidal
    Plate
y = (2*(0.5*width*0.75)*0.5 + (1.2*width)*0.75)/A;
z = y+h; //Depth of center of pressure
```

```
R = rho*g*A*z ;//Resultant force

I0 = 1.2*1.5^3/12 +1.2*1.5*1.85^2 + 1.5*1.5^3/36 + 1.5*0.75*1.6^2 ;//Second moment of area
D = (sind(phi))^2*I0/(A*z); //depth of center of pressure
M = R*(1.8533-1.1); //Moment about hinge
disp(M/1000,"Moment about the hinge line (kN/m):")
```

Scilab code Exa 3.2 RESULTANT FORCE AND CENTRE OF PRESSURE ON A PLANE SURFACE UNDER UNIFORM PRESSURE

```
clc;funcprot(0);
//Example 3.2

//Initializing the variables
w = 1.8;      //Width of plate
h1 = 5;         //Height of plate and water in upstream
h2 = 1.5;       //Height of water in downstream
rho = 1000;
g = 9.81 ;      //Acceleration due to gravity

//Calculations
function[F]=waterForce(area,meanHeight)
    F = rho * g * area * meanHeight;
endfunction

P = waterForce(w*h1,h1/2)-waterForce(w*h2,h2/2); // Resultant force on gate
x = (waterForce(w*h1,h1/2)*(h1/3) - waterForce(w*h2,h2/2)*(h2/3))/P; // point of action of p from bottom
R = P/(2*sind(20));          // Total Reaction force
Rt = 1.18*R/4.8;        //Reaction on Top
```

```
Rb = R - Rt ;          //Reaction at bottom

disp(Rb/1000, "Reaction at bottom (kN):",Rt/1000, "
    Reaction at top(kN) :");
```

Scilab code Exa 3.3 PRESSURE DIAGRAMS

```
clc;funcprot(0);
//Example 3.3

//Initializing the variables
D = 1.8;      //Depth of tank
h = 1.2;         //Depth of water
l = 3;       //Length of wall of tank
p = 35000;    //Air pressure
rho = 10^3;      //Density of water
g = 9.81;      //Acceleration due to gravity


//Calculations
Ra = p*D*l; //Force due to air
Rw = .5*(rho*g*h)*h*l; //Force due to water
R = Ra + Rw; // Resultant force
x = (Ra*0.9+Rw*0.4)/R; // Height of center of
    pressure from base
disp(x,"Height of the centre of pressure above the
    base(m) :",R/1000,"Resultant force on the wall(kN
    )");
```

Scilab code Exa 3.4 FORCE ON A CURVED SURFACE DUE TO HYDROSTATIC PRESSURE

```
clc;funcprot(0);
```

```
//Example 3.4

//Initializing the variables
R = 6; // Radius of arc
h = 2*R*sind(30);      //Depth of water
rho = 10^3;      //Density of water
g = 9.81;       //Acceleration due to gravity

//Calculations
Rh = (rho*g*h^2)/2; // Resultant horizontal force
   per unit length
Rv = rho*g*((60/360)*%pi*R^2 -R*sind(30)*R*cosd(30))
   ; // Resultant vertical force per unit length
R = sqrt(Rh^2+Rv^2); // Resultant force on gate
theta = atand(Rv/Rh); //Angle between resultant
   force and horizontal

disp(theta,"Direction of resultant force to the
   horizontal(Degree):",R/1000,"Magnitute of
   resultant force(kN/m ) :");
```

Scilab code Exa 3.5 BUOYANCY

```
clc;funcprot(0);
//Example 3.5

//Initializing the variables
B = 6; // Width of pontoon
L = 12;     //Length of pontoon
D = 1.5;               //Draught of pontoon
Dmax = 2;                   //Maximum permissible
   draught
rhoW = 1000;     //Density of fresh water
rhoS = 1025;     //Density of sea water
g = 9.81;       //Acceleration due to gravity
```

```
//Calculations
function[W]=Weight(D)
    W = rhoW*g*B*L*D;
endfunction

W = Weight(D); // Weight of pontoon in fresh water = weight of water displaced
Ds = W/(rhoS*g*B*L); //Draught in sea water
L = Weight(Dmax) - Weight(D); // maximum load that can be supported

disp(L/1000,"Load (kN) :",Ds,"Draught in sea (m):",W/1000,"Weight of pontoon (kN): ");
```

Scilab code Exa 3.6 DETERMINATION OF THE POSITION OF THE METACENTRE RELATIVE TO THE CENTRE OF BUOYANCY

```
clc;funcprot(0);
//Example 3.6

//Initializing the variables
D = 1.8;     // Diameter of buoy
H = 1.2;      //Height of buoy
W = 10*10^3;  //Weight of buoy
L = 2*10^3;   //Load
G = 0.45;     // Center of gravity
rho = 1025;    //Density of sea water
g = 9.81;     //Acceleration due to gravity

//Calculations
Z = 4*(W+L)/(rho*g*%pi*D^2);          // Depth of Immersion
BG# = (%pi*D^4/64)/(%pi*D^2*Z/4);
Z# = 0.5*Z +BG#;                      // Position of
```

```
    combined center of gravity
Z1 = ((W+L)*Z#-0.45*W)/L;          //Maximum height
    of load above bottom

disp(Z1,"Maximum height of center of gravity above
    bottom(m) :");
```

Scilab code Exa 3.7 STABILITY OF A VESSEL CARRYING LIQUID IN TANKS WITH A FREE SURFACE

```
clc;funcprot(0);
//Example 3.7

//Initializing the variables
l = 20;      // Length of barage
b = 6;        //Width of barage
r = 3;    //Radius of circular top of barage
W = 200*10^3;     //Weight of empty barage
d1 = 0.8;       // Depth of water in 1st half
d2 = 1;           // Depth of water in 2nd half
rho = 1000;     //Density of water
R = 0.8;      //Relative density of liquid
g = 9.81;       //Acceleration due to gravity
ZG = 0.45;          // Center of gravity of barage

//Calculations
I00 = l*b^3/12 +%pi*b^4/128;
ICC = l*(.5*b)^3/12;
L = d1*rho*g*l*b/2*(d1+d2);      // Weight of liquid
    load
W# = L + W;                //Total weight
A = l*b +%pi*r^2/2;        // Area of plane of
    waterline
V = W#/(rho*g);         // Volume of vessel submerged
D = V/A ;             //Depth submerged
```

```
ZB = .5*D;           //Height of center of buoyancy
NM = ZB-ZG +(1/V)*(I00-R*2*ICC);              //
    Effective metacentric height
P = R*rho*g*l*b/2*(d2-d1);      //overturning moment
theta = atand(P*1.5/(W#*NM));      //Angle of roll

disp(theta,"Angle of roll (Degree) :");
```

Chapter 4

Motion of Fluid Particles and Streams

Scilab code Exa 4.1 DISCHARGE AND MEAN VELOCITY

```
clc;funcprot(0);
//Example 4.1

//Initializing the variables
y = linspace(0,80,9);
x = [0 23 28 31 32 29 22 14 0];
xlabel('Velocity (m/s)');
ylabel('Distance from one side(mm)');
xgrid(1);

//Calculations
plot(x,y,'-*');
mu=[17.5 26.0 29.6 31.9 30.7 25.4 18.1 7.7];
         // mean velocity
disp(mean(mu),"Mean velocity (m/s):");

// the plot is attached as 4.1.png
```

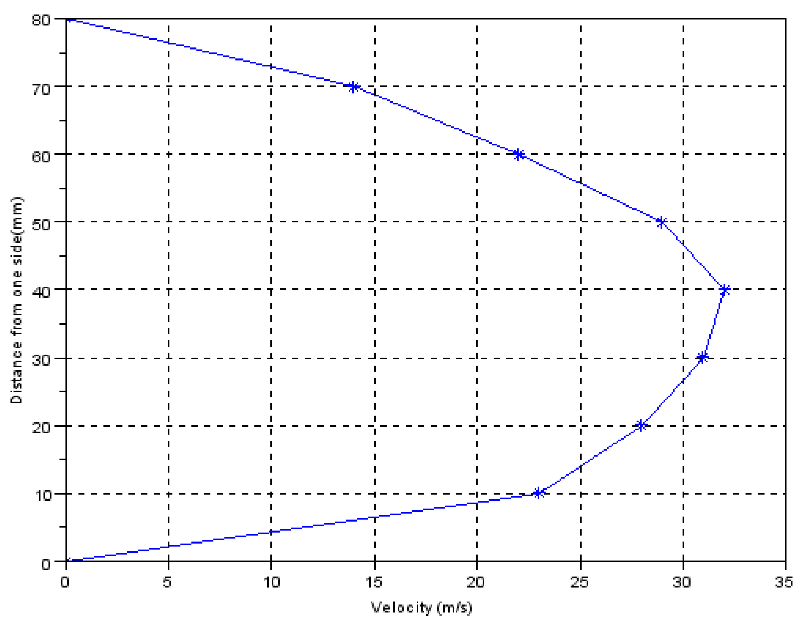

Figure 4.1: DISCHARGE AND MEAN VELOCITY

Scilab code Exa 4.2 CONTINUITY OF FLOW

```
clc;funcprot(0);//Example 4.2

//Initializing the variables all unknowns are
    assigned 0

d = [0.05 0.075 0 0.030];
Q = [0 0 0 0];
V = [0 2 1.5 0];
//Calculations
A2 = %pi*d(2)^2/4;
Q(2) =A2*V(2);
Q = [Q(2) Q(2) Q(2)/1.5 0.5*Q(2)/1.5];
d(3) = (Q(3)*4/(V(3)*%pi))^0.5;
A = %pi*d^2/4;
V(1) = V(2)*(A(2)/A(1));
V(4)=Q(4)/A(4);

disp([d*1000;A;Q;V]' , "! Diameter(mm)   Area(m2)
    Flow Rate(m3/s) Velocity(m/s) !");
```

Scilab code Exa 4.3 CONTINUITY EQUATIONS FOR THREE DIMENSIONAL FLOW USING CARTESIAN COORDINATES

```
clc;funcprot(0);
//Example 4.3

//Initializing the variables
vx= -poly(3,'x');
vy = 2*poly(-2,'x');
```

```
vz = -poly(2,'x');

//Calculations
delVx = derivat(vx);
delVy = derivat(vy);
delVz = derivat(vz);

result = derivat(vx)+derivat(vy)+derivat(vz); //
    requirement of continuity equation (result = 0)
disp("Satisfy requirement of continuity ");
```

Chapter 5

The Momentum Equation and its Applications

Scilab code Exa 5.1 GRADUAL ACCELERATION OF A FLUID IN A PIPELINE NEGLECTING ELASTICITY

```
clc;funcprot(0);
//Example 5.1

//Initializing the variables

l = 60 ;        //Length of pipeline
rho = 1000;       // Density of liquid
a = 0.02;          //Acceleration of fluid

//Calculations
delP = rho*l*a;       //Change in pressure
disp(delP/1000,"Increase of pressure difference
   required (kN/m2):");
```

Scilab code Exa 5.2 FORCE EXERTED BY A JET STRIKING A FLAT PLATE

```
clc;funcprot(0);
//Example 5.2

//Initializing the variables
v = 5;      //Velocity of jet
rho = 1000;          //density of water
d = 0.025;          //Diameter of fixed nozzle

//Calculations
//--Part(a) Variation of force exerted normal to the
    plate with plate angle--//
header = ["Theta" "       vcos(x)" "            Av" "
        Force"];
unit =   ["  deg" "            m/s" "          kg/s" "
          N"];

A = %pi*d^2/4;
x = linspace(0,90,7);
vcomp = v*cosd(x);
m = rho*A*v;
ma = linspace(m,m,7);
force = rho*A*v^2*cosd(x);
value = [x;vcomp;ma;force]' ;
disp(value,unit, header );

//--Part(b) Variation of force exerted normal to the
    plate with plate velocity--//
header = ["Theta" "   v" "   u" "    v-u" "         A(v-u
  )" "    Force"] ;
unit =   [" deg " "m/s" "m/s " "   m/s" "          kg/s
  " "        N"] ;

x = linspace(0,0,5);
v = linspace(5,5,5);
u = linspace(2,-2,5);
D = v-u;
Prod = rho*A*D;
Force = rho*A*D^2;
```

```
value = [x;v;u;D;Prod;Force]';
disp(value,unit,header)    ;
```

Scilab code Exa 5.3 FORCE DUE TO THE DEFLECTION OF A JET BY A CURVED VANE

```
clc;funcprot(0);
//Example 5.3

//Initializing the variables
x = 60;          //Angle of deflection
rho = 1000;      // Density of liquid
V1 = 30;         //Acceleration of fluid
V2 = 25;
m = .8;          //Discharge through A

//Calculations
function[R] = Reaction(Vin , Vout)
    R = m*(Vin -Vout) ;
endfunction

Rx = Reaction(V1,V2*cosd(x));
Ry = -Reaction(0,V2*sind(x));
disp(Rx,"Reaction in X-direction (N):");
disp(Ry, "Reaction in Y-direction (N) :");
disp(sqrt(Rx^2 +Ry^2), "Net Reaction (N):");
disp(atand(Ry/Rx),"Inclination of Resultant Force
   with x-direction (Degree):");
```

Scilab code Exa 5.4 FORCE EXERTED WHEN A JET IS DEFLECTED BY A MOVING CURVED VANE

```
clc;funcprot(0);
```

```
//Example 5.4

//Initializing the variables
v1 = 36 ;     //Exit velocity
u = 15;         //Velocity of vane\
x = 30;         // Angle between vanes and flow
rho = 1000;      // Density of water
d = .1;         // Diameter of jet

//Calculations
alp = atand(v1*sind(x)/(v1*cosd(x)-u));
v2 = 0.85*v1*sind(x);
bta =  acosd(u*sind(alp)/v2);
m = (rho*%pi*v1*d^2)/4;
Vin = v1*cosd(x);
Vout = v2*cosd(90);
Rx = m*(Vin-Vout);

disp( Rx ,"Force exerted by vanes(N ) :",bta,"Outlet
    Angle(Degree):",alp,"Inlet Angle(Degree): ") ;
```

Scilab code Exa 5.5 FORCE EXERTED ON PIPE BENDS AND CLOSED CONDUITS

```
clc;funcprot(0);
//Example 5.5

//Initializing the variables
rho = 850 ;     // Density of liquid
a = 0.02    ;      //Acceleration of fluid
x = 45 ;
d1 = .5 ;
d2 = .25;
p1 = 40*10^3;
p2 = 23*10^3;
```

```
Q = .45;

//Calculations
A1 = (%pi*d1^2)/4;
A2 = (%pi*d2^2)/4;
v1 = Q/A1;
v2 = Q/A2;

Rx = p1*A1 - p2*A2*cosd(x) - rho*Q*(v2*cosd(x)-v1);
Ry = p2*A2*sind(x) + rho*Q*v2*sind(x);

disp(sqrt(Rx^2 +Ry^2)/1000, "Resultant force on the
   bend(kN):");
disp(atand(Ry/Rx),"Inclination of Resultant Force
   with x-direction(Degree) :");
```

Scilab code Exa 5.6 REACTION OF A JET

```
clc;funcprot(0);
//Example 5.6

//Initializing the variables
v = 4.9;        //Velocity of Jet
rho = 1000;      // Density of water
d = 0.05;
u = 1.2   ;        // Velocity of tank
//Calculations
Vout = v;
Vin = 0;
m = rho*%pi*d^2*v/4;
R = m*(Vout-Vin);
disp(R,"Reaction (N):");
disp(R*u,"Work done per second (W):");
```

Scilab code Exa 5.7 REACTION OF A JET

```
clc;funcprot(0);
//Example 5.7

//Initializing the variables
Vj = 5*10^6;         //Velocity of Jet
Mr = 150000;       // Mass of Rocket
Mf0 = 300000;      // Mass of initial fuel
Vr = 3000;          //Velocity of jet relative to
   rocket
g = 9.81;           // Acceleration due to gravity

//Calculations
m = Vj/Vr;     //Rate of fuel consumption
T = Mf0/m;     // Burning time
function[DVt]=f(t)
    DVt = m*Vr /(Mr + Mf0 - m*t) - g;
endfunction

function[V]=h(t)
    V = -g*t - Vr*log(1 - t/269.95);
endfunction

Vt = intg(0, 180 ,f);
Z1=intg(0,180,h);

Z2 = Vt^2/(2*g);
disp(T, "(a)Burning time (s): ");
disp(Vt,"(b)Speed of rocket when all fuel is burned
   (m/s):");
disp((Z2+Z1)/1000,"(c)Maximum height reached (km):")
   ;
```

Scilab code Exa 5.8 REACTION OF A JET

```
clc;funcprot(0);//Example 5.8

//Initializing the variables
V = 200;      //Velocity in still air
Vr = 700;     //velocity of gas relative to engine
mf = 1.1;     // Fuel Consumption
r = 1/40 ;
P1 =0;
P2 = 0;

//Calculations
m1 =  mf/r;
T = m1*((1+r)*Vr -V);
disp(T/1000, "(a)Thrust (kN) :");

W = T*V;
disp(W/1000, "(b)Work done per second (kW) :");

Loss = 0.5*m1*(1+r)*(Vr-V)^2;
disp(W/(W+Loss)*100 ,"(c)Efficiency (%):");
```

Scilab code Exa 5.10 ANGULAR MOTION

```
clc;funcprot(0);//Example 5.10

//Initializing the variables
rho = 1000;      // Density of water
Q = 10;          //Acceleration of fluid
r2 = 1.6;
r1 = 1.2;
```

```
V1 = 2.3;
V2 = 0.2;
rot = 240;

//Calculations
Tf = rho*Q*(V2*r2 - V1*r1);
T = -Tf;
n = rot / 60;
P = 2*%pi*n*T;

disp(T, "Torque exerted (N- m):");
disp(P/1000, "Theoretical power output (kW) :");
```

Chapter 6

The Energy Equation and its Applications

Scilab code Exa 6.1 MECHANICAL ENERGY OF A FLOWING FLUID

```
clc;funcprot(0); //Example 6.1

//Initializing the variables
Pc = 0;           // Atmospheric Pressure
Z3 = 30+2;          //height of nozzle
Ep = 50 ;          //Energy per unit weight supplied
    by pump
d1 = 0.150;          //Diameter of sump
d2 = 0.100;          //Diameter of delivery pipe
d3 = 0.075 ;          //Diameter of nozzle
g = 9.81;          // Acceleration due to gravity
Z2 = 2;          //Height of pump
rho = 1000;          // Density of water

//Calculations
U3 = sqrt(2*g*(Ep-Z3)/(1+5*(d3/d1)^4 + 12*(d3/d2)^4)
    );
U1 = U3/4;
Pb = rho*g*Z2 + 3*rho*U1^2;
```

```
disp(U3, "Velocity of Jet through nozzle (m/s) :");
disp(Pb/1000 , "Pressure in the suction pipe at the
    inlet to the pump at B(kN/m2) :");
```

Scilab code Exa 6.2 CHANGES OF PRESSURE IN A TAPERING PIPE

```
clc;funcprot(0); //Example 6.2

//Initializing the variables
x = 45;          // Inclination of pipe
l = 2;         //Length of pipe under consideration
Ep = 50 ;         //Energy per unit weight supplied
    by pump
d1 = 0.2;         //Diameter of sump
d2 = 0.1;         //Diameter of delivery pipe
g = 9.81;         // Acceleration due to gravity
rho = 1000;         // Density of water
V1 = 2;
RD_oil = 0.9;         // relative density of oil
RD_Merc = 13.6;         // Relative density of
    Mercury

//Calculations
V2 = V1*(d1/d2)^2;
dZ = l*sind(x);
rho_Oil = RD_oil*rho;
rho_Man = RD_Merc*rho;
dP = 0.5*rho_Oil*(V2^2-V1^2) + rho_Oil*g*dZ;
h = rho_Oil *( dP/(rho_Oil*g)- dZ)/(rho_Man -
    rho_Oil);

disp(h, "Difference in the level of mercury (m):",dP
    , "Pressure Difference(N/m2) : ");
```

Scilab code Exa 6.3 PRINCIPLE OF THE VENTURI METER

```
clc;funcprot(0); //Example 6.3

//Initializing the variables
d1 = 0.25;                      //Pipeline diameter
d2 = 0.10;                      //Throat diameter
h =0.63;                      //Difference in height
rho = 1000;                   //Density of water
g = 9.81                    //Acceleration due to gravity

//Calculations
rho_Hg = 13.6*rho;
rho_Oil = 0.9*rho;
A1 = (%pi*d1^2)/4;                //  Area at entry
m = (d1/d2)^2;                    //Area ratio
Q = (A1/sqrt(m^2-1))*sqrt(2*g*h*(rho_Hg/rho_Oil -1))
   ;

disp(Q,"Thepretical Volume flow rate (m3/s ):");
```

Scilab code Exa 6.4 THEORY OF SMALL ORIFICES DISCHARGING TO ATMOSPHERE

```
clc;funcprot(0); //Example 6.4

//Initializing the variables

x = 1.5;
y =0.5;
H = 1.2;
A = 650*10^-6;
```

```
Q =0.117;
g = 9.81;

//Calculations
Cv = sqrt(x^2/(4*y*H));
Cd = Q / (60*A*sqrt(2*g*H));
Cc = Cd/Cv;

disp(Cc, "Coefficient of contraction :",Cd, "Coefficient of Discharge :", Cv, "Coefficient of velocity :");
```

Scilab code Exa 6.5 THEORY OF LARGE ORIFICES

```
clc;funcprot(0);//Example 6.5

//Initializing the variables
B = 0.7;
H1 = 0.4;
H2 = 1.9;
g =9.81;
z = 1.5 ;              // height of opening

//Calculations
Q_Th = 2/3 *B*sqrt(2*g)*(H2^1.5 - H1^1.5);
A = z*B;
h = 0.5*(H1+H2);
Q = A*sqrt(2*g*h);

disp((Q-Q_Th)*100/Q_Th, "Percentage error in discharge (%):");
```

Scilab code Exa 6.6 ELEMENTARY THEORY OF NOTCHES AND WEIRS

```
clc;funcprot(0); //Example 6.6

//Initializing the variables
Cd = 0.6;         //Coefficient of discharge
Q = 0.28;
x = 90;              //Theta
g = 9.81;
dH = 0.0015;

//Calculations
H = (Q*(15/8)/(Cd*sqrt(2*g)*tand(x/2)))^(2/5)
Frac_Q = 5/2 *( dH/H);

disp(Frac_Q*100, "Percentage error in discharge(%)")
   ;
```

Scilab code Exa 6.7 ELEMENTARY THEORY OF NOTCHES AND WEIRS

```
clc;funcprot(0); //Example 6.7

//Initializing the variables
B = 0.9;
H = 0.25;
alpha = 1.1;
g = 9.81;

//Calculations
Q = 1.84 * B * H^(3/2);
disp(Q, "Q :");

i = 1;
while(i < = 3)
    v = Q /(1.2* (H+0.2));
    disp(v, "V(m/s) :");
    k =  ((1 + alpha*v^2/(2*g*H))^1.5 -(alpha*v
```

```
        ^2/(2*g*H))^1.5 );
    Q = k* 1.84 * B * H^(3/2);
    disp(Q, "Q(m3/s) :");
    i = i+1;
end
```

Scilab code Exa 6.8 THE POWER OF A STREAM OF FLUID

```
clc;funcprot(0); //Example  6.8

//Initializing the variables
rho = 1000;
v = 66 ;
Q = 0.13;
g = 9.81;
z =240;

//Calculations
P_Jet =  0.5*rho*v^2*Q;
P_Supp = rho*g*Q*z;
P_Lost =  P_Supp -P_Jet;
h = P_Lost/(rho*g*Q);
eff = P_Jet/P_Supp;

disp(eff*100,"Part(d) Efficiency(%) :", h, "Part(C)
   head used to overcome losses (m): ", P_Supp/1000
   , "Part(b) power supplied from the reservoir (kW)
   :", P_Jet/1000 ,"PartI(a) power of the jet(kW)");
```

Scilab code Exa 6.9 VORTEX MOTION

```
clc;funcprot(0); //Example 6.9

```

```
//Initializing the variables
r1 = 0.2;
Z1 = 0.500;
Z2 = 0.340;
g = 9.81;
rho = 0.9*1000 ;

//Calculations
r0 = r1*(sqrt(2-2*Z2/Z1));
omega =  sqrt(2*g*Z1/r0^2);

function[out]=G(r)
    out =r^3 - r*r0^2;
endfunction

F = rho*omega^2*%pi*intg(r0,r1,G);

disp(F,"Part(b) Upward force on the cover (N): ",
    omega ,"Part(a) Speed of rotation (rad/s ):");
```

Scilab code Exa 6.10 Free vortex or potential vortex

```
clc;funcprot(0); //Example 6.10

//Initializing the variables
Ra = 0.2;
Rb = 0.1;
H = 0.18;
Za = 0.125;

//Calculations
Y =  Ra^2*(H-Za);
Zb = H - Y/Rb^2;

disp(Zb*1000,"Height above datum of a point B on the
```

```
free surface at a radius of 100 mm (mm):");
```

Chapter 7

Two dimensional Ideal Flow

Scilab code Exa 7.2 STRAIGHT LINE FLOWS AND THEIR COMBINATIONS

```
clc;funcprot(0); //Example 7.2

//Initializing the variables
x = 120*(2*%pi)/180; //Theta
r = 1;
v0 = 0.5;
q = 2;

//Calculations
function[y] =shi(r,theta)
    y = v0*r*sin(theta) +q*theta/(2*%pi);
endfunction


//--Approx differentiation at a point using central
   difference formula--//
h=0.0000001;
at_theta=x;
at_r = r;
Vr = (shi(r,at_theta+h)-shi(r,at_theta-h))/(r*2*h);
```

```
Vth = (shi(r+h,at_theta)-shi(r-h,at_theta))/(2*h);
V = sqrt(Vr^2+Vth^2);
alpha =  atand(abs(Vth/Vr));
bet = x*180/(2*%pi)-alpha;
disp(bet, "Beta (Degree):",alpha,"Alpha (Degree) :",
    V, "Fluid Velocity(m/s) :");
```

Scilab code Exa 7.3 COMBINED SOURCE AND SINK FLOWS DOUBLET

```
clc;funcprot(0); //Example 7.3

//Initializing the variables
q = 10;
function[Z] = shi(x,y)
    Z = (q/2/%pi)*(atan(y/(x-1))-atan(y/(x+1))) -
       25*y;
endfunction
h = 0.0000001;
Vinf = 25;

//Calculations
x = poly(0,'x');
f = x^2 - 2/(5*%pi) -1;
root = roots(f);
l = abs(root(1))+abs(root(2));
Ymax = 0.047;
width = 2*Ymax;
Vx = (shi(1-h,1)-shi(1-h,1-h))/h; // At x=1 the
   function atan is not defined hence taking x a
   little smaller.
Vy = -1*(shi(1-2*h,1)-shi(1-h,1))/h;  // At x=1 the
   function atan is not defined hence taking x a
   little smaller.

```

```
V = sqrt(Vx^2+Vy^2);
rho = poly(0,'rho');
dP = rho/2 *(V^2 - Vinf^2); //difference in pressure

disp(dP, 'Pressure Difference(N/m2) :',V, 'Velocity
    (m/s):', l, 'Length of Rankine Body(m ) :', width
    ,'Width of Rankine Body (m):' );
```

Scilab code Exa 7.4 FLOW PAST A CYLINDER

```
funcprot(0);clc; //Example 7.4

//Initializing the variables
a = 0.02;
r = 0.05;
V0 = 1;
x = 135; // Theta
function[Z] = shi(r,x)
    Z = V0*sind(x)*(r - ((a^2)/r));
endfunction
h = 0.0001;

//Calculations
Vr = 57*(shi(r,x+h)-shi(r,x))/(r*h);
Vx = -1*(shi(r+h,x)-shi(r,x))/h;

disp(Vr, 'Radial Velocity (m/s):',Vx, 'Normal
    component of velocity (m/s):');
```

Scilab code Exa 7.5 CURVED FLOWS AND THEIR COMBINATIONS

```
clc;funcprot(0); //Example 7.5

```

```
//Initializing the variables
rho = 1000;
r = 2;
psi = 2*log(r);

//Calculations
y = psi/log(r); // y = GammaC / 2*pi
v = y/r;
dPbydr = rho*v^2/r;
disp(dPbydr, 'Pressuer Gradient (N/m3 ) :');
```

Chapter 8

Dimensional Analysis

Scilab code Exa 8.1 CONVERSION BETWEEN SYSTEMS OF UNITS INCLUDING THE TREATMENT OF DIMENSIONAL CONSTANTS

```
clc;funcprot(0)//Example 8.1

//Initializing the variables
P1 = 57;                    //Power in SI
M = 1/14.6;                 //Ratio of mass in SI/
    British
L = 1/0.3048;             //Ratio of length in SI/
    British
T = 1;                    //Ratio of time in SI/British

//Calculations

P2 = M*(L^2)*(T^-3)*P1 ;    //Power in kW
P2/P1;

disp(P2/P1,"Conversion Factor :",P2, 'Power(Kw)  :'
   );
```

Chapter 9

Similarity

Scilab code Exa 9.1 MODEL STUDIES FOR FLOWS WITHOUT A FREE SURFACE

```
clc;funcprot(0); //Example 9.1

//Initializing the variables
Vp = 10;
LpByLm = 20;
rhoPbyRhoM = 1;
muPbymuM = 1;
//Calculations
Vm = Vp*LpByLm*rhoPbyRhoM*muPbymuM;

disp(Vm, 'Mean water tunnel flow velocity (m/s):');
```

Scilab code Exa 9.2 MODEL STUDIES FOR FLOWS WITHOUT A FREE SURFACE

```
funcprot(0);clc; //Example 9.2

```

```
//Initializing the variables
Vp = 3;
LpByLm = 30;
rhoPbyRhoM = 1;
muPbymuM = 1;

//Calculations
Vm = Vp*LpByLm*rhoPbyRhoM*muPbymuM;

disp(Vm, 'Mean water tunnel flow velocity (m/s):');
```

Scilab code Exa 9.3 ZONE OF DEPENDENCE OF MACH NUMBER

```
funcprot(0);clc; //Example 9.3

//Initializing the variables
Vp = 100;
cP = 340;
cM = 295;
rhoM = 7.7;
rhoP = 1.2;
muM = 1.8*10^-5;
muP = 1.2*10^-5;

//Calculations
Vm = Vp*(cM/cP);
LmByLp = 1/((Vm/Vp)*(muM/muP)*(rhoM/rhoP));
FmByFp = (rhoM/rhoP)*(Vm/Vp)^2*(LmByLp)^2;

disp(FmByFp*100, "Percentage ratio of forces (%):",
   Vm, 'Mean wind tunnel flow velocity(m/s) :');
```

Scilab code Exa 9.4 SIGNIFICANCE OF THE PRESSURE COEFFICIENT

```
funcprot(0);clc; //Example 9.4

//Initializing the variables
function[Z] =pLossRatio(RatRho,RatMu,RatL)
    Z = RatRho*RatMu^2*RatL^2;
endfunction

//Calculations
//Case (a) : water is used
RatRho = 1;
RatMu = 1;
RatL = 10;
disp(pLossRatio(RatRho,RatMu,RatL), "(a)Ratio of
    pressure losses between the model and the
    prototype if water is used ");

RatRho = 1000/1.23;
RatMu = 1.8*10^-5/10^-3;

disp(pLossRatio(RatRho,RatMu,RatL), "(b)Ratio of
    pressure losses between the model and the
    prototype if air is used ");
```

Scilab code Exa 9.5 MODEL STUDIES IN CASES INVOLVING FREE SURFACE FLOW

```
funcprot(0);clc; //Example 9.5

//Initializing the variables
scale = 1/50;
ratArea = scale^2;
Qp = 1200;

//Calculations
LmByLp = sqrt(ratArea);
```

```
VmByVp = sqrt(LmByLp);
Qm = Qp*ratArea*VmByVp;

disp(Qm, "Water flow rate (m3/s ):");
```

Scilab code Exa 9.6 SIMILARITY APPLIED TO ROTODYNAMIC MACHINES

```
funcprot(0);clc; //Example 9.6

//Initializing the variables
Qa = 2;
Na = 1400;
rhoA = 0.92;
rhoS = 1.3;
DaByDs = 1;
dPa = 200;

//Calculations
Ns = Na*(rhoA/rhoS)*(DaByDs);
Qs = Qa*(Ns/Na);
dPs = dPa *(rhoS/rhoA)*(Ns/Na)^2*(1/DaByDs)^2;

 disp(dPs,"Pressure rise(N/m2 ) :",Qs, "Flow rate (
    m3/s):",Ns,"Fan test speed (rev/s):");
```

Scilab code Exa 9.8 RIVER AND HARBOUR MODELS

```
funcprot(0);clc; //Example 9.8

//Initializing the variables
V = 300 ;// Volume rate
w = 3;
```

```
d = 65;
l = 30;
scaleH = 30/1000/18;
scaleV = 1/60;
ZmByZr = 1/60;
LmByLr = 1/600;
rho = 1000;
mu = 1.14*10^-3;

//Calculations
Vr = V/(w*d);
Vm =Vr*sqrt(ZmByZr);
m = (w*d*scaleH*scaleV)/(d*scaleH + 2*w*scaleV);
Rem = rho*Vm*m/mu;
TmByTr = LmByLr*sqrt(1/ZmByZr);
Tm = 12.4*60*TmByTr;

disp(Tm, "Tidal Period (minutes):");
```

Chapter 10

Laminar and Turbulent Flows in Bounded Systems

Scilab code Exa 10.1 INCOMPRESSIBLE STEADY AND UNIFORM LAMINAR FLOW BETWEEN PARALLEL PLATES

```
clc; funcprot(0); //Example 10.1

//Initializing the variables
mu = 0.9;
rho = 1260;
g = 9.81;
x = 45; //theta in degrees
P1 = 250 * 10^3;
P2 = 80* 10^3;
Z1 = 1;
Z2 = 0; // datum
U = -1.5;
Y = 0.01;

//Calculations
gradP1 = P1+ rho*g*Z1;
gradP2 = P2+ rho*g*Z2;
DPstar = (gradP1-gradP2 )*sind(x)/(Z1-Z2);
```

```
A = U/Y; // Coefficient U/Y for equation 10.6
B = DPstar/(2*mu);// Coefficient dp*/dx X(1/2mu) for
    equation 10.6
y = poly(0,'y');
v = (A + B*Y)*y -B*y^2;
duBYdy = derivat(v);
tau = 0.9*duBYdy;
ymax = roots(duBYdy);        //value of y where
   derivative vanishes.;
umax = (A + B*Y)*ymax + B*ymax^2; // Check the value
    there is slight mistake in books answer
function[z] = u(y)
    z = (A + B*Y)*y -B*y^2;
endfunction
tauMax =abs( mu*derivative(u,Y));
ymax
disp(tauMax/1000,"Maximum Shear Stress (kN/m2):",
   umax, "Maximum Flow Velocity (m/s)",tau, "Shear
   Distribution :", v,"Velocity Distribution :"  )
```

Scilab code Exa 10.2 INCOMPRESSIBLE STEADY AND UNIFORM LAMINAR FLOW IN CIRCULAR CROSS SECTION PIPES

```
clc; funcprot(0); //Example 10.2

//Initializing the variables
mu = 0.9;
rho = 1260;
d = 0.01;
Q = 1.8/60*10^-3; //Flow in m^3 per second
l = 6.5;
ReCrit = 2000;
//Calculations
A = (%pi*d^2)/4;
MeanVel = Q/A;
```

```
Re = rho*MeanVel*d/mu; // Check properly the answer
    in book there is something wrong
Dp = 128*mu*l*Q/(%pi*d^4)
Qcrit = Q*ReCrit/Re*10^3;
disp(Qcrit, "Maximum Flow rate(litres/s) :", Dp/1000
    , "Pressure Loss (N/m2) :");
```

Scilab code Exa 10.3 INCOMPRESSIBLE STEADY AND UNIFORM TURBULENT FLOW IN CIRCULAR CROSS SECTION PIPES

```
clc; funcprot(0);//Example 10.3

//Initializing the variables
mu = 1.14*10^-3;
rho = 1000;
d = 0.04;
Q = 4*10^-3/60; //Flow in m^3 per second
l = 750;
ReCrit = 2000;
g = 9.81;
k = 0.00008; // Absolute Roughness

//Calculations
A = (%pi*d^2)/4;
MeanVel = Q/A;
Re = rho*MeanVel*d/mu;
Dp = 128*mu*l*Q/(%pi*d^4);
hL = Dp/(rho*g);
f = 16/Re;
hlDa = 4*f*l*MeanVel^2/(2*d*g); // By Darcy Equation
Pa = rho*g*hlDa*Q;

//Part(b)
Q = 30*10^-3/60; //Flow in m^3 per second
MeanVel = Q/A;
```

```
Re = rho*MeanVel*d/mu;
RR = k/d; // relative roughness
f = 0.008 ;//by Moody diagram for Re =  1.4 x 10^4
    and relative roughness = 0.002
hlDb = 4*f*l*MeanVel^2/(2*d*g); // By Darcy Equation
Pb = rho*g*hlDb*Q;
disp(Pb, "Power Required (W) :",hlDb , "Head Loss(m)
    :","!---- Case(b) ----!",Pa, "Power Required(W)
    :",hlDa*1000 , "Head Loss(mm) :", "!---- Case (a)
    ----!");
```

Scilab code Exa 10.4 STEADY AND UNIFORM TURBULENT FLOW IN OPEN CHANNELS

```
clc; funcprot(0); //Example 10.4

//Initializing the variables
w = 4.5;
d = 1.2 ;
C = 49;
i = 1/800;

//Calculations
A = d*w;
P = 2*d + w;
m = A/P;
v = C*sqrt(m*i);
Q = v*A;

disp(Q,"Discharge (m3/s):",v, "Mean Velocity (m/s):"
    );
```

Scilab code Exa 10.5 VELOCITY DISTRIBUTION IN TURBULENT FULLY DEVELOPED PIPE FLOW

```
clc; funcprot(0); //Example 10.5

//Initializing the variables
R = poly(0,'R');

//Calculations
r = R*(1 - (49/60)^7);

disp(r,"Radius at which the actual velocity is equal
    to the mean velocity :");
```

Scilab code Exa 10.7 SEPARATION LOSSES IN PIPE FLOW

```
clc; funcprot(0); //Example 10.7

//Initializing the variables
d1 = 0.140;
d2 = 0.250;
DpF_DpR = 0.6; //Difference in head loss when in
    forward and in reverse direction
K = 0.33 ;//From table
g = 9.81;
//Calculations
ratA = (d1/d2)^2;

v = sqrt(DpF_DpR*2*g/((1 - ratA)^2 - K));

disp(v,"Velocity (m/s):");
```

Chapter 11

Boundary Layer

Scilab code Exa 11.1 PROPERTIES OF THE LAMINAR BOUNDARY LAYER FORMED OVER A FLAT PLATE IN THE ABSENCE OF A PRESSURE GRADIENT IN THE FLOW DIRECTION

```
clc; funcprot(0); //Example 11.1

//Initializing the variables
rho = 860;
v = 10^-5;
Us = 3;
b = 1.25;
l = 2;

//Calculations
x = 1; // At x =1
Rex = Us*x/v;
ReL = Us*l/v ;
mu =  rho*v;
T0 = 0.332*mu*Us/x*Rex^0.5;
Cf = 1.33*ReL^-0.5;
F = rho*Us^2*l*b*Cf ;

disp(F,"Total, double-sided resistance of the plate
```

```
(N) :",T0,"Shear stress at mid-length(N/m2)");
```

Scilab code Exa 11.2 PROPERTIES OF THE TURBULENT BOUNDARY LAYER OVER A FLAT PLATE IN THE ABSENCE OF A PRESSURE GRADIENT IN THE FLOW DIRECTION

```
clc; funcprot(0); //Example 11.2

//Initializing the variables
Us = 6;
b = 3;
l = 30;
rho = 1000;
mu = 10^-3;
T = 20+273; // Temperature in Kelvin

//Calculations
1/mu;
ReL = rho*Us*l/mu;
Cf = 0.455*log10(ReL)^-2.58 ;

F = rho*Us^2*l*b*Cf ;
Lt = 10^5*mu/(rho*Us); // Assuming transition at Rel = 10^5
disp(Lt,"Transition length (m) :",F/1000,"Total drag on the plate (kN):");
```

Chapter 12

Incompressible Flow around a Body

Scilab code Exa 12.1 DRAG

```
clc; funcprot(0);//Example 12.1

//Initializing the variables
x =35;
T = 50;
m = 1;
g =9.81;
rho = 1.2;
A = 1.2;
U0 = 40*1000/3600; // Velocity in m/s

//Calculations
L = T*sind(x)+m*g;
D =T*cosd(x);
Cl = 2*L/(rho*U0^2*A);
Cd = 2*D/(rho*U0^2*A);
disp(Cd,"Drag Coefficient :",Cl,"Lift Coefficient :"
    );
```

Scilab code Exa 12.2 RESISTANCE OF SHIPS

```
clc; funcprot(0); //Example 12.2

//Initializing the variables
Vp =12;
lp = 40;
lm = 1;
As = 2500;
Dm = 32;
rhoP = 1025;
rhoM = 1000;
Ap = As;

//Calculations
Am = As/40^2;
Vm = Vp*sqrt(lm/lp);
Dfm = 3.7*Vm^1.95*Am;
Rm = Dm - Dfm;
Rp = Rm *(rhoP/rhoM)*(lp/lm)^2*(Vp/Vm)^2;
Dfp = 2.9*Vp^1.8*Ap;
Dp = Rp + Dfp;

disp(Dp/1000,"Expected total resistance (kN) :");
```

Scilab code Exa 12.3 FLOW PAST A CYLINDER

```
clc; funcprot(0); //Example 12.3

//Initializing the variables
U0 = 80*1000/3600;
d = 0.02;
```

```
rho =1.2;
mu = 1.7*10^-5;
A = 0.02*500; // Projected area of wire
N = 20; // No of cables

//Calculations
Re = rho*U0*d/mu;
Cd = 1.2;// From figure 12.10 for given Re;
D = 0.5*rho*Cd*A*U0^2
F = N*D;
f =  0.198*(U0/d)*(1-19.7/Re);

disp(f,"Frequency (Hz):",F/1000,"Total force on
    tower (kN):");
```

Scilab code Exa 12.4 FLOW PAST A SPHERE

```
clc; funcprot(0); //Example 12.4

//Initializing the variables
mu = 0.03;
d = 10^-3;
rhoP = 1.1*10^3;
g = 9.81;
rho0 = 0.9*10^3;
//Calculations
B = 18*mu/(d^2*rhoP);
t = 4.60/B;
Vt = d^2*(rhoP - rho0)*g/(18*mu);
Re = rho0*Vt*d/mu;


disp(Re,"Reynolds No corrosponding to the velocity :
    ",t,"Time taken by the particle take to reach 99
    per cent of its terminal velocity (s):");
```

Scilab code Exa 12.5 FLOW PAST A SPHERE

```
clc; funcprot(0); //Example 12.5

//Initializing the variables
mu0 = 0.0027;
Vt = 3*10^-3;
rhoW = 1000;
rhoP = 2.4*rhoW;
rho0 = 0.9*rhoW;
g = 9.81;
muA = 1.7*10^-5;
rhoA = 1.3;

//Calculations
d = sqrt(18*mu0*Vt/(rhoP-rho0)/g);
Re = Vt*d*rho0/mu0;

//Movement of particle in upward direction
if(Re < 1 ) then
    v = 0.5;
    Re = 5 ; // from fig 12.15
    vt = muA*Re/(rhoA*d);
    u = vt+v;
    disp(u ,"Velocity of air stream blowing
       vertically up (m/s) :");
else
    disp("strokes law is not valid");
end
```

Scilab code Exa 12.6 FLOW PAST AN AEROFOIL OF FINITE LENGTH

```
clc; funcprot(0); //Example 12.6

//Initializing the variables
c = 2;
s = 10;
rho = 5.33;
rho_ellip = 1.2;
D = 400;
L = 45000;
scale = 20;
U_windTunnel = 500;
U_proto = 400*1000/3600;

//Calculations
A = c*s;
U_model = U_windTunnel/scale;
Cd = D/(0.5*rho*U_model^2*A);
Cl = L/(0.5*rho_ellip*U_proto^2*A); // Considering elliptical Lift model
Cdi = Cl^2/(%pi*s/c); // Aspect Ratio = s/c
Cdt = Cd + Cdi;
Dw = 0.5*Cdt*rho_ellip*U_proto^2*A;
disp(Dw/1000, "Total drag on full sized wing (kN) :");
```

Chapter 13

Compressible Flow around a Body

Scilab code Exa 13.1 EFFECTS OF COMPRESSIBILITY

```
clc; funcprot(0); //Example 13.1

//Initializing the variables
rho0 = 1.8;
R = 287;
T = 75+273; // Temperature in kelvin
gma = 1.4;
Ma = 0.7;

//Calculations
P0 = rho0*R*T;
c = sqrt(gma*R*T);
V0 = Ma*c;
Pt = (P0^((gma-1)/gma) + rho0*((gma-1)/gma)*(V0
    ^2/(2*P0^(1/gma))))^(gma/(gma-1));
rhoT =  rho0*(Pt/P0)^(1/gma);
Tt =  Pt/(R*rhoT)-273;

disp(rhoT,"Density of airstream (kg/M3):",Tt,"
```

```
Temperature (Degree) :", Pt/1000,"Staganation
Pressure (kN/m2 ) :");
```

Scilab code Exa 13.2 SHOCK WAVES

```
clc; funcprot(0); //Example 13.2

//Initializing the variables
R = 287;
T = 28+273;
gma = 1.4;
P = 1.02*10^5;
rhoHg = 13.6*10^3;
g = 9.81;

//Calculations
//Case(a)
U0 = 50;
c = sqrt(gma*R*T);
Ma = U0/c;
rho = P/(R*T);
DelP = 0.5*rho*U0^2; //Pt-P
ha = DelP/(rhoHg*g);

//Case(b)
U0 = 250;
Ma = U0/c;
Pt = P*(1+(gma-1)*Ma^2/2)^(gma/(gma-1));
DelP = Pt-P
hb = DelP/(rhoHg*g);

//Case (c)
U0 = 420;
Ma1 =U0/c;
P2 = P*((2*gma/(gma+1))*Ma1^2 - ((gma-1)/(gma+1)));
```

```
N = Ma1^2 +2/(gma-1); // Numerator
D = 2*gma*Ma1^2/(gma-1)-1;
Ma2 = sqrt(N/D);
Pt2 = P2*(1+(gma-1)*Ma2^2/2)^(gma/(gma-1));
hc = (Pt2-P2)/(rhoHg*g) ;

disp(hc*1000,"Difference in height of mercury column
    in case (c) in mm :",hb*1000,"Difference in
   height of mercury column in case (b) in mm :",ha
   *1000,"Difference in height of mercury column in
   case (a) in mm :");
```

Chapter 14

Steady Incompressible Flow in Pipe and Duct Systems

Scilab code Exa 14.1 INCOMPRESSIBLE FLOW THROUGH DUCTS AND PIPES

```
clc; funcprot(0);//Example 14.1

//Initializing the variables
L1 = 5;
L2 = 10;
d = 0.1;
f = 0.08;
Za_Zc = 4;          //difference in height between A
   and C
g = 9.81 ;
Pa = 0;
Va = 0;
Za_Zb = -1.5;
V = 1.26;
rho = 1000;

//Calculations
D = 1.5 + 4*f*(L1+L2)/d ; // Denominator in case of
```

```
    v^2
v = sqrt(2*g*Za_Zc/D);
Pb = rho*g*Za_Zb - rho*V^2/2*(1.5+4*f*L1/d);
disp(Pb/1000,"Pressure in the part at B (kN/m2):",v,
    "Mean Velocity at C (m/s):");
```

Scilab code Exa 14.3 INCOMPRESSIBLE FLOW THROUGH PIPES IN PARALLEL

```
clc; funcprot(0);//Example 14.3

//Initializing the variables
Za_Zb = 10;
f = 0.008;
L = 100;
d1 = 0.05;
g = 9.81;
d2 = 0.1;
//Calculations
function[z] = flowRate(d)
     D = 1.5 + 4*f*L/d ; // Denominator in case of
        v1^2
     A = %pi*d^2/4;
     v = sqrt(2*g*Za_Zb/D);
     z = A*v;
endfunction

Q1 = flowRate(d1);
Q2 = flowRate(d2);

Q = Q1+Q2;
D = poly(0, 'D');
v = 4*Q/(%pi*D^2);
X = 1.5 + 4*f*L/D;
f = 10*2*g/(X*v^2)  - 1;
```

```
f = numer(f) ;
diameter = roots(f); // Taking roots of numerator
    denominator will be multiplied by zero
i = 1;
while (i<=length(diameter))
    x = diameter(i);
    if(imag(x) == 0) then
        dia = diameter(i);
        i= i+1;
    else
        i = i+1;
    end
end

disp(dia*1000,"Diameter of single equivalent pipe(mm
    ) :", Q2 ,"Flow throught pipe 2 (m3/s):", Q1 ,"
    Flow throught pipe 1 (m3/s):");
```

Scilab code Exa 14.4 INCOMPRESSIBLE FLOW THROUGH BRANCHING PIPES THE THREE RESERVOIR PROBLEM

```
clc; funcprot(0); //Example 14.4

//Initializing the variables
Za_Zb = 16;
Za_Zc = 24;
f = 0.01;
l1 = 120;
l2 = 60;
l3 = 40;
d1 = 0.12;
d2 = 0.075;
d3 = 0.060;
g = 9.81;
//Calculations
```

```

A = [%pi*d1^2/4 %pi*d2^2/4 %pi*d3^2/4]
function[z] = Coeff(l,d)
    z = 4*f*l/(d*2*g);
endfunction

function[f] = F(x)
    f(1) = Coeff(l1,d1)*x(1)^2 + Coeff(l2,d2)*x(2)^2
         - Za_Zb;
    f(2) = Coeff(l1,d1)*x(1)^2 + Coeff(l3,d3)*x(3)^2
         - Za_Zc;;
    f(3) = x(1)*d1^2 - x(2)*d2^2 - x(3)*d3^2; // Q1=
       Q2
endfunction

function[j] = jacob(x)
    j(1,1) = 2*Coeff(l1,d1)*x(1); j(1,2) = 2*Coeff(
       l2,d2)*x(2);j(1,3) = 0;
    j(2,1) = 2*Coeff(l1,d1)*x(1); j(2,2) = 0;j(2,3)
       = 2*Coeff(l3,d3)*x(3);
    j(3,1) = d1^2; j(3,2) = -d2^2;j(3,3) = -d3^2;
endfunction

x = [1.8 0 0];
v = fsolve(x,F,jacob, 10^-20);
disp(v(3)*A(3),"Flow rate in pipe 3 (m3/s):",v(2)*A
   (2),"Flow rate in pipe 2 (m3/s):",v(1)*A(1),"Flow
    rate in pipe 1 (m3/s) :");
```

Scilab code Exa 14.5 INCOMPRESSIBLE STEADY FLOW IN DUCT NETWORKS

```
clc; funcprot(0); //Example 14.5

//Initializing the variables
```

```
D = 0.3;
Q = 0.8;
rho = 1.2;
f = 0.008;
L_entry = 10;
L_exit = 30;
Lt = 20*D;//Transition may be represented by a
    separation loss equivalent length of 20    the
    approach duct diameter
K_entry = 4;
K_exit = 10
l = 0.4;         // length of cross section
b = 0.2;         // width of cross section

//Calculations
A = %pi*D^2/4;
Dp1 = 0.5*rho*Q^2/A^2*(K_entry + 4*f*(L_entry+Lt)/D)
    ;
area = l*b;
perimeter =2*(l+b);
m = area/perimeter;
Dp2 = 0.5*rho*Q^2/area^2*(K_exit + f*L_exit/m);
Dfan =  Dp1+Dp2;

disp(Dfan,"fan Pressure input (N/m2) :");
```

Scilab code Exa 14.6 INCOMPRESSIBLE STEADY FLOW IN DUCT NETWORKS

```
clc; funcprot(0); //Example 14.6

//Initializing the variables
D = [0.15 0.3];
rho = 1.2;
f = 0.008;
```

```
L_entry = 10;
L_exit = 20;
Lt = 20*D(2);//Transition may be represented by a
    separation loss equivalent length of 20     the
    approach duct diameter
K = 4;
Q1 = 0.2;

//Calculations
Q2 = 4*Q1;
A = %pi*D^2/4;
Dp1 = 0.5*rho*Q1^2/A(1)^2*(K + 4*f*L_entry/D(1));
Dp2 = 0.5*rho*Q2^2/A(2)^2*(4*f*(L_exit + Lt)/D(2));
Dfan =  Dp1+Dp2;

disp(Dfan,"fan Pressure input (N/m2) :");
```

Scilab code Exa 14.7 RESISTANCE COEFFICIENTS FOR PIPELINES IN SERIES AND IN PARALLEL

```
clc; funcprot(0);//Example 14.7

//Initializing the variables
d = [0.1 0.125 0.15 0.1 0.1 ]; // Corrosponding to
    AA1B AA2B BC CD CF
l = [30 30 60 15 30];                         //
    Corrosponding to AA1B AA2B BC CD CF
rho = 1.2;
f = 0.006;
Ha = 100;
Hf = 60;
He = 40;

//Calculations
for(i=1:length(d))
```

```
    K(i) = f*l(i)/(3*d(i)^5);
end

K_ab = K(1)*K(2)/(sqrt(K(1))+sqrt(K(2)))^2;
K_ac = K_ab + K(3);
Hc = (K_ac*Hf +K(5)*Ha/4)/(K_ac+K(5)/4);
Q = sqrt((Ha - Hc)/K_ac);

function[z] = f(n)
    z = He - Hc + (0.5*Q)^2 *(K(4)+(4000/n)^2);
endfunction

n = fsolve(1,f);

disp(n,"Percentage valve opening (%) :", Hc,"Head at
    C (m):", Q, "total Volume flow rate (m3/s):");
```

Scilab code Exa 14.8 THE QUANTITY BALANCE METHOD FOR PIPE NETWORKS

```
clc; funcprot(0);//Example 14.8

//Initializing the variables
d = [0.3 0.25 0.2];
l = [1500 800 400];
f = 0.01;
Ha = 60;
Hb = 30;
Hc = 15;
Hd = 35; // Assumption
H(1) = Ha - Hd;
H(2) = Hb - Hd;
H(3) = Hc - Hd;

//Calculations
```

```
K = 0;
for(i=1:length(d))
    K(i) = f*l(i)/(3*d(i)^5);
end
Qsum = 0.001;
for(i=1:2)
    Q = 0;Qby2h = 0;Qs = 0;Qby2hsum = 0;
    for(i=1:3)
        if(imag(sqrt(H(i)/K(i)))~=0) then
            Q(i) = -1*abs(imag(sqrt(H(i)/K(i))));
        else
            Q(i) = sqrt(H(i)/K(i));
        end,

        Qby2h = Q(i)/(2*H(i));
        Qs = Qs+Q(i);
       Qby2hsum = Qby2hsum +Qby2h

    end
    dH = Qs/Qby2hsum;
    for(i=1:3)
        H(i)=H(i)+dH;
    end
    Qsum = Qs;
end

disp(Q(3),"Q_dc (m3/s) :",Q(2),"Q_db (m3/s) :",Q(1),
    "Q_ad (m3/s) :");
```

Scilab code Exa 14.9 QUASI STEADY FLOW

```
clc; funcprot(0);//Example 14.9

//Initializing the variables
As = 6;
```

```
d = 0.02;
f =0.01;
L = 1.5;
K = 0.9;
g = 9.81;

//Calculations
Ap = %pi*d^2/4;
function[y] = Qinv(h)
    y = sqrt((4*f*L/d +K+1)/(2*g*h))/Ap;
endfunction
//By direct integration
t = -As*intg(3.5,2.25,Qinv); // Discharge is 2 m
    below
disp(t, "Time of discharge by direct integration (s)
    : ");

// By numerical integrations
interval = [0.250 0.125 0.0083 0.0063 0.005 0.0042];
for(i=1:length(interval))

    start=3.5;piece=3.5:-interval(i):2.25;
    X=-As*integrate('Qinv(h)','h',start,piece);

    disp(X(length(X)),"Value of t (s) : ",interval(i
        ),"______For Interval(Dh in m)_____");
end
```

Scilab code Exa 14.12 QUASI STEADY FLOW

```
clc; funcprot(0);//Example 14.12

//Initializing the variables
As = 6;
A2 = 4.5;
```

```
d = 0.02;
f =0.01;
L = 1.5;
K = 0.9;
g = 9.81;

//Calculations
Ap = %pi*d^2/4;
C = Ap*sqrt(2*g/(4*f*L/d+K+1));

function[y] = Qinv(h)
    y = sqrt(1/h)/(C*(1+As/A2));
endfunction

//By direct integration
t = -As*intg(3.0,2.0,Qinv); // Discharge is 2 m
   below
disp(t, "Time of discharge by direct integration (s)
   : ");

//By Numerical Integration
interval = [0.250 0.125 0.0083 0.0063 0.005 0.0042];
for(i=1:length(interval))

    start=3.0;piece=3.5:-interval(i):2.0;
    X=-As*integrate('Qinv(h)','h',start,piece);

    disp(X(length(X)),"Value of t (s): ",interval(i)
       ,"______For Interval(Dh in m)_____");
end
```

Chapter 15

Uniform Flow in Open Channels

Scilab code Exa 15.1 RESISTANCE FORMULAE FOR STEADY UNIFORM FLOW IN OPEN CHANNELS

```
clc; funcprot(0);//Example 15.1

//Initializing the variables
B =4;
D = 1.2;
C = 7.6;
n = 0.025;
s = 1/1800;

//Calculations
W = B + 2*1.5*D;
A = D*(B+C)/2; // Area of parallelogram formed
P = B +2*1.2*sqrt(D^2 +(1.5D)^2);
m =A/P;
i=s;
C = (23+0.00155/i+1/n)/(1+(23+0.00155/i)*n/sqrt(m));
    // By Kutter formula
Q1 = C*A*sqrt(m*i);
```

```
Q2 = A*(1/n)*m^(2/3)*sqrt(i);
disp(Q2,"Q using the Manning formula(m3/s) :",Q2,"Q
   using Chezy formula with C determined from the
   Kutter formula (m3/s) :");
```

Scilab code Exa 15.2 OPTIMUM SHAPE OF CROSS SECTION FOR UNIFORM FLOW IN OPEN CHANNELS

```
clc; funcprot(0);//Example 15.2

//Initializing the variables
Q = 0.5;
C = 80;
i = 1/2000;

//Calculations
function[y] = f(D)
    y = (7/4)*C*D^(5/2)*sqrt(i/2) -Q;
endfunction
disp(fsolve(2,f), "Optimum depth = Optimum Width (in
    metres):");
```

Chapter 16

Non uniform Flow in Open Channels

Scilab code Exa 16.2 EFFECT OF LATERAL CONTRACTION OF A CHANNEL

```
clc; funcprot(0);//Example 16.2

//Initializing the variables
B = [1.4 0.9];
D = [0.6 0.32];
g = 9.81;
h = 0.03;
Z = 0.25;

//Calculations
Q1 = B(2)*D(2)*sqrt(2*g*h/(1-(B(2)*D(2)/B(1)*D(1))
    ^2))
E = D(1)-Z;
Q2 = 1.705*B(2)*E^1.5;
disp(Q2 , "Volume flow rate (m3/s) :");
```

Scilab code Exa 16.3 CLASSIFICATION OF WATER SURFACE PROFILES

```
clc; funcprot(0); //Example 16.3

//Initializing the variables
a =0.5;
b = 0.5;
Dn = 1.2;
s = 1/1000;
C = 55;
g = 9.81;

//Calculations
c = (1 + a)/b;
QbyB = Dn*C*sqrt(Dn*s);
q = QbyB;
Dc = (q^2/g)^(1/3);

m = 2.4:-0.15:1.35;
total = 0;Dm = 0; N = 0;  D = 0; Lm = 0;
for(i=1:length(m)-1)

    Dm(i)= (m(i)+m(i+1))/2;
    N(i) = 1 - (Dc/Dm(i))^3 ;    // Numerator
    D(i) = 1 - (Dn/Dm(i))^3; // Denominator
    Lm(i) = 150*(N(i)/D(i));
    total = total +Lm(i);
end
result  = [Dm N D Lm];
disp(total,"distance upstream covered (approx in m):
  ",result,"Mean Depth(Dm) Numerator Denominaotor
       L(m)");
```

Chapter 17

Compressible Flow in Pipes

Scilab code Exa 17.1 MASS FLOW THROUGH A VENTURI METER

```
clc; funcprot(0);//Example 17.1

//Initializing the variables
g = 9.81;
rho = 1000;
rhoHg = 13.6*rho;
d1 = 0.075;
d2 = 0.025;
Pi = 0.250;
Pt = 0.150;
P_Hg = 0.760;
rho1 = 1.6;
gma = 1.4;

//Calculations
P1 = (Pi+P_Hg)*rhoHg*g;
P2 = (Pt+P_Hg)*rhoHg*g;
rho2 = rho1*(P2/P1)^(1/gma);

function[f] = velocity(V)
    f(1) = d2^2*V(2)*rho2-d1^2*V(1)*rho1;
```

```
    f(2) = 0.5*(V(2)^2 - V(1)^2)*((gma-1)/gma)*(rho2
       *rho1/(rho2*P1-rho1*P2))-1;
endfunction
V = [0 0];
Velo = fsolve(V,velocity);
Flow = %pi*d1^2/4*Velo(1);
disp(Flow, "Volume of flow (m3/s):");
```

Scilab code Exa 17.2 THE LAVAL NOZZLE

```
clc; funcprot(0); //Example 17.2

//Initializing the variables
Ma = 4;
gma = 1.4;
At = 500; // in mm

//Calculations
N = 1 + (gma-1)*Ma^2/2;
D = (gma+1)/2 ;
A = At*(N/D)^((gma+1)/(2*(gma-1)))/Ma;

disp(A, "Area of test section (mm2):");
```

Scilab code Exa 17.3 NORMAL SHOCK WAVE IN A DIFFUSER

```
clc; funcprot(0);//Example 17.3

//Initializing the variables
Ma1 = 2;
gma = 1.4;
T1 = 15+273; // In kelvin
P1 = 105;
```

```

//Calculations
Ma2 = sqrt(((gma-1)*Ma1^2 +2)/(2*gma*Ma1^2-gma+1));
P2 = P1*(1+gma*Ma1^2)/(1+gma*Ma2^2);
T2 = T1*(1 +(gma-1)/2*Ma1^2)/(1 +(gma-1)/2*Ma2^2);
disp(T2 - 273, "Temperature (Degree C) of downstream
    shock wave :",P2, "Pressure (bar) of downstream
   shock wave :",Ma2, "Mach No downstream shock wave
    :");
```

Scilab code Exa 17.4 COMPRESSIBLE FLOW IN A DUCT WITH FRICTION UNDER ADIABATIC CONDITIONS FANNO FLOW

```
clc; funcprot(0); //Example 17.4

//Initializing the variables
gma = 1.4;
f = 0.00375;
d = 0.05;

//Calculations
m = d/4;
function[y] = x(Ma)
    A =(1 -Ma^2 )/(gma*Ma^2);
    B = (gma+1)*Ma^2/(2+(gma-1)*Ma^2);
    y = m/f*(A+ (gma+1)*log(B)/(2*gma));
endfunction

X1 = x(0.2); // At entrance Ma = 0.2;
X06_X1 =x(0.6); // Section(b) Ma = 0.6;

X06 =  X1-X06_X1;
disp(X06, "Distance from the entrance (m):",X1,"The
   Distance X1 at which the Mach number is unity (m)
    :");
```

Scilab code Exa 17.5 ISOTHERMAL FLOW OF A COMPRESSIBLE FLUID IN A PIPELINE

```
clc; funcprot(0); //Example 17.4

//Initializing the variables
gma = 1.4;
Q = 28/60; // m3/s
d = 0.1;
p1 = 200*10^3;
f = 0.004;
x_x1 = 60;
R = 287;
T = 15+273;

//Calculations
A = %pi*d^2/4;
m = d/4;
v1 = Q/A;
pa = p1*sqrt(1-f*(x_x1)*v1^2/(m*R*T));

function[y] =g(p)
    A = (v1*p1)^2/(R*T)
    B = f*A*x_x1/(2*m);
    y = (p^2 - p1^2)/2 -A*log(p/p1) +B;
endfunction
pb=fsolve(pa,g);// Guessing solution around pa
disp(pb/1000,"Pressure at the outlet, allowing for
   velocity changes (kN) :",pa/1000,"Pressure at the
    outlet, neglecting velocity changes (kN)");
```

Chapter 20

Pressure Transient Theory and Surge Control

Scilab code Exa 20.3 APPLICATION OF THE SIMPLIFIED EQUATIONS TO EXPLAIN PRESSURE TRANSIENT OSCILLATIONS

```
clc; funcprot(0); //Example 20.3

//Initializing the variables
c = 1250;
Dt = 0.02;
Dv = 0.5;
rho = 1000;
v =0.5;

//Calculations
cDt = c*Dt;
Dp = rho*c*Dv;
DOv_DOt = Dv/Dt;
vDOv_DOt = v*Dv/cDt;
DOp_DOt = Dp/Dt;
vDOp_DOx = v*Dp/cDt;
Error = [vDOv_DOt*100/DOv_DOt vDOp_DOx*100/DOp_DOt];
disp(Error, "The percentage errors are given below
```

```
    are very small hence can be neglected :");
```

Scilab code Exa 20.5 CONTROL OF SURGE FOLLOWING VALVE CLOSURE WITH PUMP RUNNING AND SURGE TANK APPLICATIONS

```
clc; funcprot(0); //Example 20.5

//Initializing the variables
f = 0;
Atunnel = 1.227;
Ashaft = 12.57;
Q =2;
L = 200;
g = 9.81;

//Calculations
Zmax = (Q/Ashaft)*sqrt(Ashaft*L/(Atunnel*g));
T = 2*%pi*sqrt(Ashaft*L/(Atunnel*g));
disp(T,"Mass Oscillation Period (s) : ",Zmax,"Peak
    water level (m):");
```

Chapter 22

Theory of Rotodynamic Machines

Scilab code Exa 22.1 ONE DIMENSIONAL THEORY

```
 clc; funcprot(0); //Example 22.1

//Initializing the variables
Q = 5;
R1 = 1.5/2;
R2 = 2/2;
w = 18;
rho = 1000;
rhoA = 1.2;
Hth = 0.017;
g=9.81;

//Calculations
A = %pi*(R2^2-R1^2);
Vf = Q/A;
Ut = w*R2;
Uh = w*R1;
B1t = acotd(Ut/Vf);
B1h = acotd(Uh/Vf);
```

```
E = Hth*rho/rhoA;
function[y] = Beta(u)
    y = acotd((u-E*g/u)/Vf);
endfunction
B2t = Beta(Ut);
B2h = Beta(Uh);

disp(B2h,"At Hub :",B2t,"At tip :","!----Blade
   Outlet Angles (Degrees)----!",B1h,"At Hub :",B1t,
   "At tip :","!----Blade Inlet Angles----!");
```

Scilab code Exa 22.2 DEPARTURES FROM EULERS THEORY AND LOSSES

```
clc; funcprot(0); //Example 22.2

//Initializing the variables
D = 0.1;
t = 15*10^-3;
Q = 8.5/3600;
N = 750/60;
B2 = 25; // Beta 2 ind degrees
g = 9.81;
z = 16;

//Calculations
A = %pi*D*t;
V_f2 = Q/A;
U2 = %pi*N*D;
V_w2 = U2 - V_f2*cotd(B2);
Hth = U2*V_w2/g;
Sf = 1 - %pi*sind(B2)/(z*(1-(V_f2/U2)*cotd(B2)));
H = Sf*Hth;

disp(H, "Part (b) - Head developed (m): ",Hth, "Part
```

```
(a) - Head developed (m): ");
```

Scilab code Exa 22.3 COMPRESSIBLE FLOW THROUGH ROTODYNAMIC MACHINES

```
clc; funcprot(0); //Example 22.3.

//Initializing the variables
Ma = 0.6;
Cl = 0.6;
tByC = 0.035; // Thickness to chord ratio
cByC = 0.015; // Camber to chord ratio
x = 3; // Angle of incidence

//Calculations
lamda = 1/sqrt(1-Ma^2);
Cl# = lamda*Cl;
tByC1 = tByC*lamda;
cByC1 = cByC*lamda;
Cl1 = Cl*lamda^2;
Ae = x*lamda;

disp(Ae,"angle of incidence (Degree) :", Cl1, "Lift
   Coefficient :",cByC1, "Camber to chord ratio :",
   tByC1,"Thickness to chord ratio :","____Geometric
    Characterstics____" );
```

Chapter 23

Performance of Rotodynamic Machines

Scilab code Exa 23.1 DIMENSIONLESS COEFFICIENTS AND SIMILARITY LAWS

```
clc; funcprot(0); //Example 23.1

//Initializing the variables
Q = [0:7:56];
H = [40 40.6 40.4 39.3 38 33.6 25.6 14.5 0];
n = [0 41 60 74 83 83 74 51 0];
N1 = 750;
N2 = 1450;
D1 = 0.5;
D2 = 0.35;

//Calculations
Q2 = Q*(N2/N1)*(D2/D1)^3;
H2 = H*(N2/N1)^2*(D2/D1)^2;
xlabel("Q (m3/s)");
ylabel("H (m of water )and n(percent)");
```

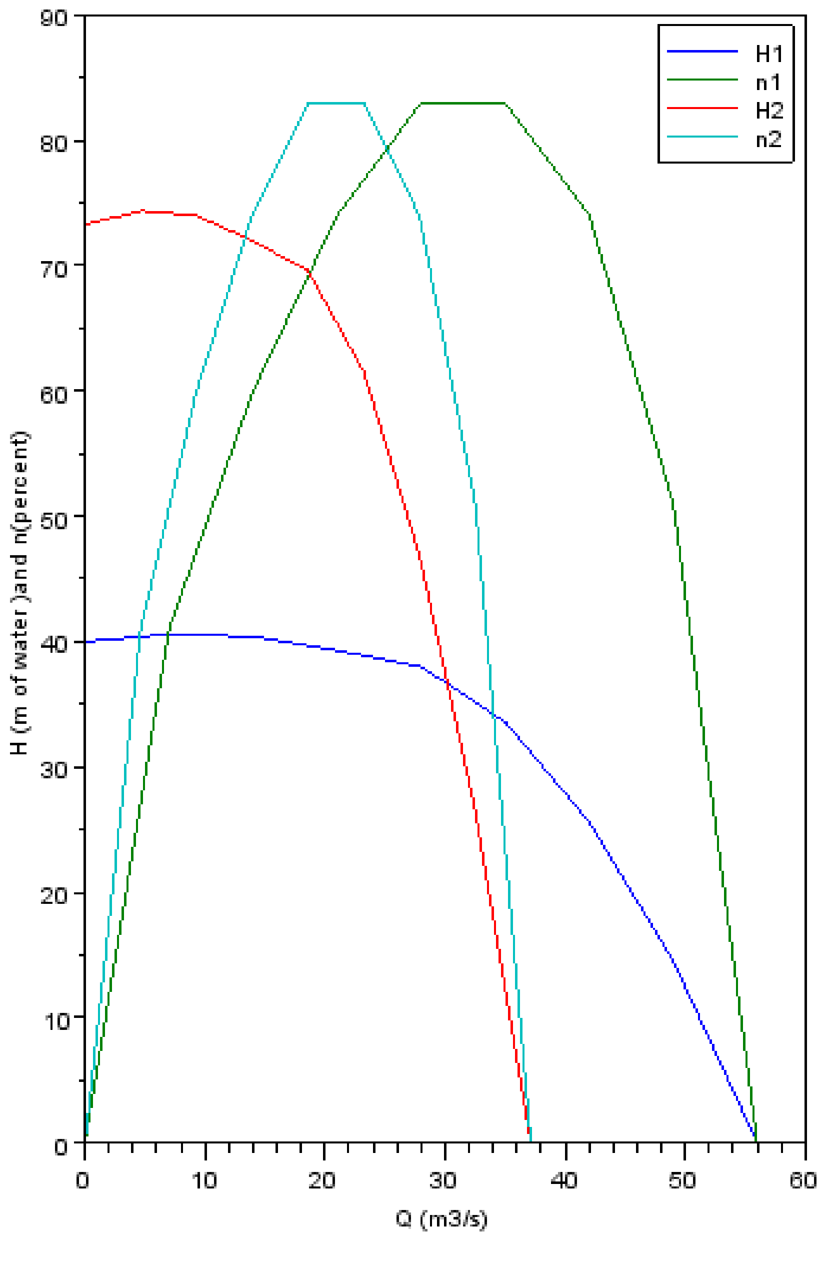

Figure 23.1: DIMENSIONLESS COEFFICIENTS AND SIMILARITY LAWS

```
plot(Q,H,Q,n,Q2,H2,Q2,n);

legend("H1","n1","H2","n2");
```

Scilab code Exa 23.2 THE PELTON WHEEL

```
clc; funcprot(0); //Example 23.2

//Initializing the variables
n = 0.9;
g = 9.81;
D = 1.45;
N = 375/60;
H = 200; // Real height
x = 165; // Theta
P = 3750*10^3;
rho = 1000;

//Calculations
h = n*H;            //Effective Head
v1 = sqrt(2*g*h);
u = %pi*D*N;

n_a = (2*u/v1^2)*(v1-u)*(1-n*cosd(x));

P_b = P/n_a;
ppj = P_b/2; // Power per jet
d = sqrt(8*ppj/(rho*%pi*v1^3)) ;

disp(d,"Diameter of Jet (m) :",n_a*100, "E fficiency
    (%) :")
```

Scilab code Exa 23.3 FRANCIS TURBINES

```
clc; funcprot(0); //Example 23.3

//Initializing the variables
g = 9.81;
H = 12;
n = 0.8;
w = 300*2*%pi/60;
Q = 0.28;

//Calculations
V_f1 = 0.15*sqrt(2*g*H);
V_f2 =V_f1;
V_w1 = sqrt(n*g*H);
u1 = V_w1;
theta = atand(V_f1/u1);
u2 =0.5*u1;
B2 = atand(V_f2/u2);
r1 = u1/w;
b1 = Q/(V_f2*0.9*2*%pi*r1); // vanes occupy 10 per cent of the circumference hence 0.9
b2 = 2*b1;

disp(b2*1000,"Width of runner at exit(mm) :", b1*1000,"Width of runner at inlet (mm) :", B2, "Vane angle at exit (degree) :",theta, "Guide vane angle (degree) :");
```

Scilab code Exa 23.4 AXIAL FLOW TURBINES

```
clc; funcprot(0);//Example 23.4

//Initializing the variables
H = 35;
g = 9.81;
D = 2;
```

```
N = 145/60;
z = 30; // angle between vanes and direction of
    runner rotation
y = 28; // angle between runner blades at the outlet
    .

//Calculations
H_net = 0.93*H ; // since 7% head is lost
v1 = sqrt(2*g*H_net);
u = %pi*N*D;
V_r2 = u*cosd(y);
V2= u*sind(y);
V_w2 = V2*sind(y);

// Function to solve the vector for Vr1 and B1 by
    just re writing the parallelogram law in arranged
     form
function[f] = F(x)
    f(1) = u^2 + x(1)^2 + 2*u*x(1)*cosd(x(2))-v1^2;
    f(2) = x(1)*sind(x(2)) - tand(z)*(u + x(1)*cosd(
        x(2)));
endfunction
X = [10  50]; // An innitial guess of vector length
    and angle by figure
result=fsolve(X,F);
V_r1 =result(1);
B1 = result(2);
V_w1 = u + V_r1*cosd(B1)
E = (u/g)*(V_w1 - V_w2);
n = E/H;
disp(n*100,"Efficiency (%):",B1, "Blade angle at
    inlet (Degree) :" );
```

Scilab code Exa 23.5 HYDRAULIC TRANSMISSIONS

```
clc; funcprot(0);
// Example 23.5

//Initializing the variables
s = 0.03;
P = 185*10^3;
rho = 0.86*10^3;
A = 2.8*10^-2;
N = 2250/60;
D = 0.46;

//Calculations
R0 = 0.46/2;
Ws_Wp = 1-s;
n = Ws_Wp;
Pf = s*P;
Q = (2*Pf*A^2/(3.5*rho))^(1/3);
Wp = 2*%pi*N;
Ri = sqrt((1/Ws_Wp)*(R0^2 -P/(rho*Q*Wp^2))); //
   Modified equation for power transmission.
Di = 2*Ri;
T = P/(rho*Wp^3 *D^5);

disp(T,"Torque Coefficient :", Di*1000,"Mean
   diameter (mm) : ");
```

Chapter 24

Positive Displacement Machines

Scilab code Exa 24.1 RECIPROCATING PUMPS

```
clc; funcprot(0);
//Example 24.1

//Initializing the variables
H_at = 10.3;
Hs = 1.5;
Hd = 4.5;
Ls = 2;
Ld = 15;
g = 9.81;
Ds = 0.4; // Diameter of stroke
Db = 0.15;     // Diameter of bore
Dd = 0.05;      // Diameter of discharge and suction
   pipe
nu = 0.2;
f = 0.01;
abs_pump_pressure = 2.4;

//Calculations
```

```
A = %pi*(Db)^2/4;
a = %pi*(Dd)^2/4;
r = Ds/2;
W = 2*%pi*nu;
Hsf = 0;
function[y] = H_suck(n) // n for checking the sign of Hsi = 4fl/2dg *(vA/a)^2
y = H_at - Hs +(-1)^n*(L/g)*(A/a)*W^2*r;
endfunction

function[y] = H(n,DischargeOrSuction)// n for checking the sign of Hsi = 4fl/2dg *(vA/a)^2, for suction 1 and for discharge2
    if(DischargeOrSuction == 1) then
        y = H_at - Hs +(-1)^n*(Ls/g)*(A/a)*W^2*r;
    elseif(DischargeOrSuction == 2) then
        y = H_at + Hd +(-1)^n*(Ld/g)*(A/a)*W^2*r;
    else disp("There is something wrong :")
    end
endfunction

function[y] = H_mid(DischargeOrSuction,uA)// n for checking the sign of Hsi = 4fl/2dg *(vA/a)^2, for discharge 1 and for suction 2

    if(DischargeOrSuction == 1) then
        Hsf = 4*f*Ls/(2*Dd*g)*(uA/a)^2;
        y = H_at - Hs - Hsf;
    elseif(DischargeOrSuction == 2) then
        Hsf = 4*f*Ld/(2*Dd*g)*(uA/a)^2;
        y = H_at + Hd + Hsf;
    else disp("There is something wrong :")
    end
endfunction

Hs_start = H(1,1);      // Inertia head negative hence n = 1
Hs_end = H(2,1);      // Inertia head positive hence
```

```
   n = 2
Hd_start = H(1,2);
Hd_end = H(2,2);
u = W*r;
Hs_mid = H_mid(1,u*A);
slip = 0.04;
Hd_mid = H_mid(2,u*A);
suction = [Hs_start Hs_end Hs_mid];
discharge = [Hd_start Hd_end Hd_mid];
header = ["    Start(m)","    End(m)","      Mid(m)"];
W_max = sqrt((abs_pump_pressure - H_at + Hs)*(g/Ls)
   *(a/A)*(1/r));
W_max_rev = W_max/(2*%pi)*60; // maximum rotation
   speed in rev/min
disp(W_max_rev,"Drive speed for s eperation (rev/min
   ) :","!----Part(c)----1",discharge,header,"!----
   Part(b)----! Head at",suction,header,"!----Part(a
   )----! Head at");
```

Scilab code Exa 24.2 RECIPROCATING PUMPS

```
clc; funcprot(0);
//Example 24.2

//Initializing the variables
H_friction = 2.4;
H_at = 10.3;
Hs = 1.5;
L =2;
f = 0.01;
d = 0.05;
g = 9.81;
Ds = 0.4; // Diameter of stroke
Db = 0.15;    // Diameter of bore
r = 0.2;
```

```
//Calculations
A = %pi*(Db)^2/4;
a = %pi*(Dd)^2/4;
W= sqrt((H_at - Hs -  H_friction )*(2*d*g/(4*f*L)))
    *(a/A)*(%pi/r);
W_rev = W/(2*%pi)*60;  // maximum rotation speed in
    rev/min

disp(W_rev-40, "Increase in speed (rev/min):");
```

Chapter 25

Machine Network Interactions

check Appendix AP 1 for dependency:

`intersectFunc.sci`

Scilab code Exa 25.1 FANS PUMPS AND FLUID NETWORKS

```
clc; funcprot(0);

//————————————————————————Important Note
    ————————————————————————————————//
// Please keep intersectFunc.sci in the same folder
    in which this file//
// is kept and change the current working directory
    to the directory  //
// in which both the files are kept  using chdir "
    absolute path"      //
//
    ————————————————————————————————————————————————————————————————————————


```

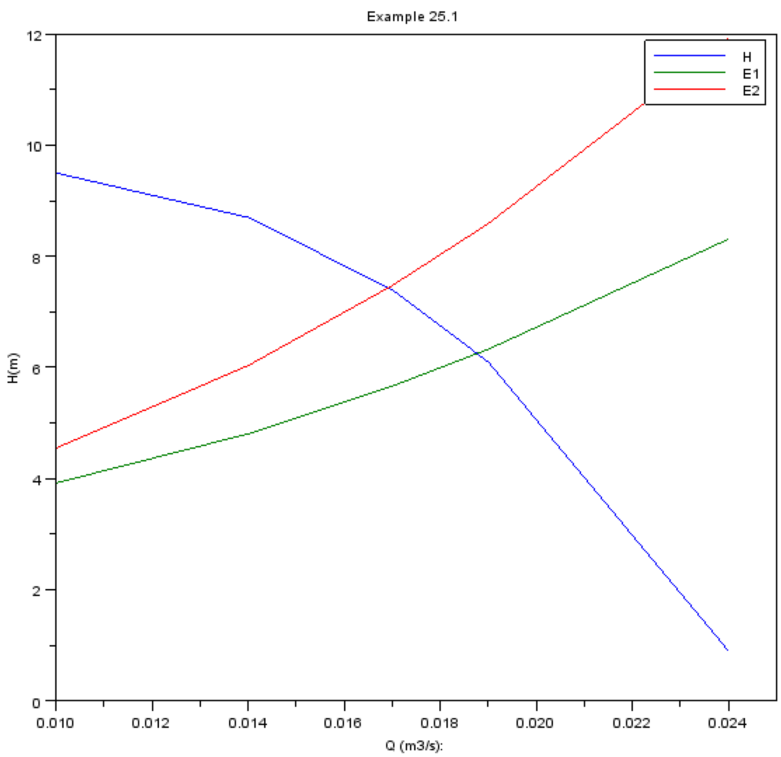

Figure 25.1: FANS PUMPS AND FLUID NETWORKS

```
//Example 25.1

//Initializing the variables
exec("intersectFunc.sci");
Q =[0.010 0.014 0.017 0.019 0.024]';
H = [9.5 8.7 7.4 6.1 0.9]';
n = [65 81 78 68 12]';
d = 0.15;
mu = 1.14*10^-3;
rho = 1000;
g = 9.81;

//Calculations
E1 = 3+9218*Q^2; // f = 0.0025 from moody chart
Q1 = intersectFunc(H,E1,Q);
v1 = 4*Q1/(%pi*d^2);
Re1 = rho*v1*d/mu;
E2 = 3+15486*Q^2;  // since f = 0.0042
Q2 = intersectFunc(H,E2,Q);
n = 0.78; // efficiency at Q2 from graph
H1 = 7.45; // From Graph
P = rho*g*H1*Q2/n;

title("Example 25.1");
xlabel("Q (m3/s):");
ylabel("H(m)");
plot(Q,H, Q,E1,Q,E2);
legend("H","E1","E2");

disp(P/100, "Power consumed (kW) :", Q2,"Flow betwen
    the reservoirs (m3/s) :");
```

Scilab code Exa 25.4 AN APPLICATION OF THE STEADY FLOW ENERGY EQUATION

```
clc; funcprot(0);
// Example 25.4

//Initializing the variables
Pa_P1 = -200; // From previous Question
Q = 1.4311 ; // From previous questions.

//Calculations
DpSys = Pa_P1 + 98.9*Q^2;
disp(DpSys,"System Operating point (m3/s):");
```

Scilab code Exa 25.7 JET FANS

```
clc; funcprot(0);
//Example 25.7

//Initializing the variables
Vo = 25.3;           //Outlet velocity
D = 10 ;               // Mean hydraulic diameter
f = 0.008;              // friction factor
X = 1000;               // Length of road
P = 12600;              // Absorbing power
Va = 300;              // Tunnel air flow
K1 = 0.96;
K2 = 0.9;
T = 590;               //Thrust
rho = 1.2;              // Air density

//Calculations
alpha = (1/D)^2;
A = %pi*D^2/4;      // Area of tunnel
Vt = Va/A;
W = Vo/Vt;           //Omega
E = (1-alpha*W);
C = (1-alpha*W)*(1-E)^2 + E^2 - 1;
```

```
// Manipulating equation 25.20;
LHS = f*X*(E+1)^2/D + C + 1 ;
n = poly(0,"n");
RHS = K1*(2*((alpha*W^2 + (1-alpha)*E^2-1)+(n-1)*(
    alpha*W*(W-1)-C/2)));
Equation = RHS -LHS;
roots(Equation);

// Alternative approach using equation 25.22
n = (rho*((4*f*X*Vt^2)/(2*D)  + 1.5*Vt^2/2))*A/(K1*
    K2*T);
Pt = round(n)*P;
disp(Pt/1000,"Total  power consumed (KW) :", round(n
    ), "Number of fans required :");
```

Scilab code Exa 25.8 JET FANS

```
clc; funcprot(0);
//Example 25.8

//Initializing the variables
f = 0.008;
T = 290;
L = 750;
Dt = 9;          // Diameter Tunnel
Df = 0.63;          // Diameter fan
K1 = 0.98;
K2 = 0.92;
Vo = 27.9;
n = 10;

//Calculations
alpha = (Df/Dt)^2;
// equation 25.20 becomes when E = 1 nad C = 0
W = poly(0,'W' );
```

```
Equation = 2*K1* (alpha*W^2 +(n-1)*alpha*W*(W-1)) -
    4*f*L/Dt -1;
omega  = roots(Equation);
     for(i = 1:length(omega))
         if(real(omega(i))>0) then // since omega is
             always positive and real
             w = omega(i);
         end,
     end
Vt = Vo/w;
disp(Vt, "Tunnel Velocity(m/s) :");
```

Scilab code Exa 25.9 CAVITATION IN PUMPS AND TURBINES

```
clc; funcprot(0);
//Example 25.9

//Initializing the variables
Ws = 0.45;
Ks = 3.2;
H = 152;
h = 0;
Hatm = 10.3;
Pv = 350;      //vapour pressure
g = 9.81;
rho = 1000;

//Calculations
Ht1 = H*(Ws/Ks)^(4/3)
Hvap = Pv/(rho*g);
Z = Hatm -h -Hvap -Ht1;
disp(Z,"Elevation of pump (m):");
```

Scilab code Exa 25.11 VENTILATION AND AIRBORNE CONTAMINATION AS A CRITERION FOR FAN SELECTION

```
clc; funcprot(0);
//Example 25.11

//Initializing the variables
Co = 0;
Qc = 0.0024;
V = 5400;
c = 10;
//Calculations
function[y] =  partA(n)
    Ci = 10;
    t = 10^1000; // infinity (a very large number)
    Q = V*n/3600;
    y =  (Co + 10000*Qc/Q)*(1-%e^(-n*t)) + Ci*%e^(-n
       *t) - c;
endfunction

Sol_A = fsolve(1,partA);

function[y] =  partB(n)
    Ci = 0;
    t = 1; // time in hours
    Q = V*n/3600;
    A = Co + 10000*Qc/Q;
    B = Ci*%e^(-n*t) - c;
    y =  A*(1-%e^(-n*t)) + B;
endfunction

Sol_B = fsolve(1,partB);

function[y] =  partC(c)
    Ci = 0;
    n = 1;
    t = 0.333333; // 20 minutes in hours
    Q = V*n/3600;
```

```
    y =  (Co + 10000*Qc/Q)*(1-%e^(-n*t)) + Ci*%e^(-n
       *t) - c;
endfunction

Sol_C = fsolve(1,partC);

function[y] =  partD(t)
    Ci = 10;
    n = 1;
    c = 0.1;
    y = Ci*%e^(-n*t) - c;
endfunction

Sol_D = fsolve(0.001 , partD);


disp(Sol_D,"Part(D) : time necessary to run the
   ventilation system at the rate calculated in (b)
   to reduce the concentration to 0.001 per cent (in
    hours) :", Sol_C,"Part(C) :the concentration
   after 20 minutes (Parts per 10000) :",Sol_B,"Part
   (B) : number of air changes per hour if this
   maximum level is reached after 1 hour and the
   garage is out of use :" , Sol_A,"Part(A) : number
    of air changes per hour if the garage is in
   continuous use and the maximum permissible
   concentration of carbon monoxide is 0.1 per cent.
    :");
```

Appendix

Scilab code AP 1 UDF Intersect Function

```
//**********************intersectFunc scilab function***********************//
//Takes argument as three arrays namely f1,f2(functions) and their domain    //
//It finds intersecting points in all the subdomains                        //
//Gives output as point(s) of intersection of the two functions             //
// Domain should be INCREASING                                              //
//Created by : Jay Chakra (www.jaychakra.co.cc)                             //
//              Undergraduate                                               //
//              Aerospace Engineering                                       //
//              IIT Bombay                                                  //
//Comments, suggestions, bugs welcomed at jaychakra.jc@gmail.com            //
//****************************************************************************

```

```
function[y] = intersectFunc(f1,f2,domain)
    L1 = length(f1);
    L2 = length(f2);
    L3 = length(domain);
    if((L1~=L2)|(L1~=L3)|(L2~=L3))then
        error("Check Dimensions of input parameters !!  ");
    else
        R = 1;
        y = [];
            for(i=1:L1-1)
                N1 = f1(i+1)-f1(i);
                N2 = f2(i+1)-f2(i);
                D = domain(i+1)-domain(i);
                x = poly(0,'x');
                //Writing equation of straight lines joining the terminal points
                f(1) = f1(i) +(N1/D)*(x-domain(i));
                f(2) = f2(i) +(N2/D)*(x-domain(i));
                difference = f(2) - f(1);
                root = roots(difference);        // Solution will be the roots of difference
                if(difference == 0) then         // if both functions are same
                    y = domain;
                    break;
                elseif(root~=[]&root<=domain(i+1)&root>=domain(i)) then //if roots lie in the subdomain
                    y(R) = root;
                    R=R+1;
                 end,
             end,
    end
endfunction
```