

Scilab Textbook Companion for
Fluid Mechanics
by J. F. Douglas¹

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

May 24, 2016

¹Funded 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	4
1 Fluids and their properties	6
2 Pressure and Head	7
3 Static Forces on Surfaces Buoyancy	14
4 Motion of Fluid Particles and Streams	21
5 The Momentum Equation and its Applications	25
6 The Energy Equation and its Applications	33
7 Two dimensional Ideal Flow	41
8 Dimensional Analysis	45
9 Similarity	46
10 Laminar and Turbulent Flows in Bounded Systems	51
11 Boundary Layer	56
12 Incompressible Flow around a Body	58
13 Compressible Flow around a Body	63
14 Steady Incompressible Flow in Pipe and Duct Systems	66

15 Uniform Flow in Open Channels	76
16 Non uniform Flow in Open Channels	78
17 Compressible Flow in Pipes	80
20 Pressure Transient Theory and Surge Control	84
22 Theory of Rotodynamic Machines	86
23 Performance of Rotodynamic Machines	89
24 Positive Displacement Machines	95
25 Machine Network Interactions	99

List of Scilab Codes

Exa 1.1	Excess Pressure	6
Exa 2.1	VARIATION OF PRESSURE VERTICALLY	7
Exa 2.2	VARIATION OF PRESSURE WITH ALTITUDE	7
Exa 2.3	VARIATION OF PRESSURE WITH ALTITUDE IN A GAS UNDER ADIABATIC CONDITIONS	8
Exa 2.4	VARIATION OF PRESSURE AND DENSITY WITH ALTITUDE FOR A CONSTANT TEMPERATURE GRA- DIENT	9
Exa 2.5	PRESSURE AND HEAD	9
Exa 2.6	PRESSURE MEASUREMENT BY MANOMETER	10
Exa 2.7	PRESSURE MEASUREMENT BY MANOMETER	10
Exa 2.8	PRESSURE MEASUREMENT BY MANOMETER	11
Exa 2.9	PRESSURE MEASUREMENT BY MANOMETER	12
Exa 2.10	RELATIVE EQUILIBRIUM	12
Exa 3.1	RESULTANT FORCE AND CENTRE OF PRESSURE ON A PLANE SURFACE UNDER UNIFORM PRES- SURE	14
Exa 3.2	RESULTANT FORCE AND CENTRE OF PRESSURE ON A PLANE SURFACE UNDER UNIFORM PRES- SURE	15
Exa 3.3	PRESSURE DIAGRAMS	16
Exa 3.4	FORCE ON A CURVED SURFACE DUE TO HYDRO- STATIC PRESSURE	16
Exa 3.5	BUOYANCY	17
Exa 3.6	DETERMINATION OF THE POSITION OF THE META- CENTRE RELATIVE TO THE CENTRE OF BUOY- ANCY	18

Exa 3.7	STABILITY OF A VESSEL CARRYING LIQUID IN TANKS WITH A FREE SURFACE	19
Exa 4.1	DISCHARGE AND MEAN VELOCITY	21
Exa 4.2	CONTINUITY OF FLOW	23
Exa 4.3	CONTINUITY EQUATIONS FOR THREE DIMENSIONAL FLOW USING CARTESIAN COORDINATES	23
Exa 5.1	GRADUAL ACCELERATION OF A FLUID IN A PIPELINE NEGLECTING ELASTICITY	25
Exa 5.2	FORCE EXERTED BY A JET STRIKING A FLAT PLATE	25
Exa 5.3	FORCE DUE TO THE DEFLECTION OF A JET BY A CURVED VANE	27
Exa 5.4	FORCE EXERTED WHEN A JET IS DEFLECTED BY A MOVING CURVED VANE	27
Exa 5.5	FORCE EXERTED ON PIPE BENDS AND CLOSED CONDUITS	28
Exa 5.6	REACTION OF A JET	29
Exa 5.7	REACTION OF A JET	30
Exa 5.8	REACTION OF A JET	31
Exa 5.10	ANGULAR MOTION	31
Exa 6.1	MECHANICAL ENERGY OF A FLOWING FLUID	33
Exa 6.2	CHANGES OF PRESSURE IN A TAPERING PIPE	34
Exa 6.3	PRINCIPLE OF THE VENTURI METER	35
Exa 6.4	THEORY OF SMALL ORIFICES DISCHARGING TO ATMOSPHERE	35
Exa 6.5	THEORY OF LARGE ORIFICES	36
Exa 6.6	ELEMENTARY THEORY OF NOTCHES AND WEIRS	36
Exa 6.7	ELEMENTARY THEORY OF NOTCHES AND WEIRS	37
Exa 6.8	THE POWER OF A STREAM OF FLUID	38
Exa 6.9	VORTEX MOTION	38
Exa 6.10	Free vortex or potential vortex	39
Exa 7.2	STRAIGHT LINE FLOWS AND THEIR COMBINATIONS	41
Exa 7.3	COMBINED SOURCE AND SINK FLOWS DOUBLET	42
Exa 7.4	FLOW PAST A CYLINDER	43
Exa 7.5	CURVED FLOWS AND THEIR COMBINATIONS	43

Exa 8.1	CONVERSION BETWEEN SYSTEMS OF UNITS INCLUDING THE TREATMENT OF DIMENSIONAL CONSTANTS	45
Exa 9.1	MODEL STUDIES FOR FLOWS WITHOUT A FREE SURFACE	46
Exa 9.2	MODEL STUDIES FOR FLOWS WITHOUT A FREE SURFACE	46
Exa 9.3	ZONE OF DEPENDENCE OF MACH NUMBER	47
Exa 9.4	SIGNIFICANCE OF THE PRESSURE COEFFICIENT	47
Exa 9.5	MODEL STUDIES IN CASES INVOLVING FREE SURFACE FLOW	48
Exa 9.6	SIMILARITY APPLIED TO ROTODYNAMIC MACHINES	49
Exa 9.8	RIVER AND HARBOUR MODELS	49
Exa 10.1	INCOMPRESSIBLE STEADY AND UNIFORM LAMINAR FLOW BETWEEN PARALLEL PLATES	51
Exa 10.2	INCOMPRESSIBLE STEADY AND UNIFORM LAMINAR FLOW IN CIRCULAR CROSS SECTION PIPES	52
Exa 10.3	INCOMPRESSIBLE STEADY AND UNIFORM TURBULENT FLOW IN CIRCULAR CROSS SECTION PIPES	53
Exa 10.4	STEADY AND UNIFORM TURBULENT FLOW IN OPEN CHANNELS	54
Exa 10.5	VELOCITY DISTRIBUTION IN TURBULENT FULLY DEVELOPED PIPE FLOW	54
Exa 10.7	SEPARATION LOSSES IN PIPE FLOW	55
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	56
Exa 11.2	PROPERTIES OF THE TURBULENT BOUNDARY LAYER OVER A FLAT PLATE IN THE ABSENCE OF A PRESSURE GRADIENT IN THE FLOW DIRECTION	57
Exa 12.1	DRAG	58
Exa 12.2	RESISTANCE OF SHIPS	59
Exa 12.3	FLOW PAST A CYLINDER	59
Exa 12.4	FLOW PAST A SPHERE	60

Exa 12.5	FLOW PAST A SPHERE	61
Exa 12.6	FLOW PAST AN AEROFOIL OF FINITE LENGTH	61
Exa 13.1	EFFECTS OF COMPRESSIBILITY	63
Exa 13.2	SHOCK WAVES	64
Exa 14.1	INCOMPRESSIBLE FLOW THROUGH DUCTS AND PIPES	66
Exa 14.3	INCOMPRESSIBLE FLOW THROUGH PIPES IN PAR- ALLEL	67
Exa 14.4	INCOMPRESSIBLE FLOW THROUGH BRANCHING PIPES THE THREE RESERVOIR PROBLEM	68
Exa 14.5	INCOMPRESSIBLE STEADY FLOW IN DUCT NET- WORKS	69
Exa 14.6	INCOMPRESSIBLE STEADY FLOW IN DUCT NET- WORKS	70
Exa 14.7	RESISTANCE COEFFICIENTS FOR PIPELINES IN SERIES AND IN PARALLEL	71
Exa 14.8	THE QUANTITY BALANCE METHOD FOR PIPE NETWORKS	72
Exa 14.9	QUASI STEADY FLOW	73
Exa 14.12	QUASI STEADY FLOW	74
Exa 15.1	RESISTANCE FORMULAE FOR STEADY UNIFORM FLOW IN OPEN CHANNELS	76
Exa 15.2	OPTIMUM SHAPE OF CROSS SECTION FOR UNI- FORM FLOW IN OPEN CHANNELS	77
Exa 16.2	EFFECT OF LATERAL CONTRACTION OF A CHAN- NEL	78
Exa 16.3	CLASSIFICATION OF WATER SURFACE PROFILES	79
Exa 17.1	MASS FLOW THROUGH A VENTURI METER . . .	80
Exa 17.2	THE LAVAL NOZZLE	81
Exa 17.3	NORMAL SHOCK WAVE IN A DIFFUSER	81
Exa 17.4	COMPRESSIBLE FLOW IN A DUCT WITH FRIC- TION UNDER ADIABATIC CONDITIONS FANNO FLOW	82
Exa 17.5	ISOTHERMAL FLOW OF A COMPRESSIBLE FLUID IN A PIPELINE	83
Exa 20.3	APPLICATION OF THE SIMPLIFIED EQUATIONS TO EXPLAIN PRESSURE TRANSIENT OSCILLA- TIONS	84

Exa 20.5	CONTROL OF SURGE FOLLOWING VALVE CLOSURE WITH PUMP RUNNING AND SURGE TANK APPLICATIONS	85
Exa 22.1	ONE DIMENSIONAL THEORY	86
Exa 22.2	DEPARTURES FROM EULERS THEORY AND LOSSES	87
Exa 22.3	COMPRESSIBLE FLOW THROUGH ROTODYNAMIC MACHINES	88
Exa 23.1	DIMENSIONLESS COEFFICIENTS AND SIMILARITY LAWS	89
Exa 23.2	THE PELTON WHEEL	91
Exa 23.3	FRANCIS TURBINES	91
Exa 23.4	AXIAL FLOW TURBINES	92
Exa 23.5	HYDRAULIC TRANSMISSIONS	93
Exa 24.1	RECIPROCATING PUMPS	95
Exa 24.2	RECIPROCATING PUMPS	97
Exa 25.1	FANS PUMPS AND FLUID NETWORKS	99
Exa 25.4	AN APPLICATION OF THE STEADY FLOW ENERGY EQUATION	101
Exa 25.7	JET FANS	102
Exa 25.8	JET FANS	103
Exa 25.9	CAVITATION IN PUMPS AND TURBINES	104
Exa 25.11	VENTILATION AND AIRBORNE CONTAMINATION AS A CRITERION FOR FAN SELECTION	105
AP 1	UDF Intersect Function	107

List of Figures

4.1 DISCHARGE AND MEAN VELOCITY	22
23.1 DIMENSIONLESS COEFFICIENTS AND SIMILARITY LAWS	90
25.1 FANS PUMPS AND FLUID NETWORKS	100

Chapter 1

Fluids and their properties

Scilab code Exa 1.1 Excess Pressure

```
1 funcprot(0); clc;
2 //Example 1.1
3 //Initialization of Variable
4 sigma = 72.7*10^-3;           //Surface Tension
5 r = 1 *10^-3;                //Radius of Bubble
6
7 //Calculations
8 P = 2*sigma/r;
9 disp(P,"Excess Pressure(N/m2) :")
```

Chapter 2

Pressure and Head

Scilab code Exa 2.1 VARIATION OF PRESSURE VERTICALLY

```
1  clc ;funcprot(0);
2  //Example 2.1
3
4  //Initializing the variables
5  z1 = 0; //Taking Ground as reference
6  z2 = -30; //Depth
7  rho = 1025; //Density
8  g = 9.81; //Acceleration due to gravity
9
10 //Calculation
11 pressureIncrease = -rho*g*(z2-z1);
12 disp(pressureIncrease/1000,"Increase in Pressure (KN
    /m2):");
```

Scilab code Exa 2.2 VARIATION OF PRESSURE WITH ALTITUDE

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

```

3
4 //Initializing the variables
5 p1 = 22.4*10^3; //Initial Pressure
6 z1 = 11000; //Initial Height
7 z2 = 15000; //final Height
8 g = 9.81; //Acceleration due to gravity
9 R = 287; //Gas Constant
10 T = 273-56.6; //Temperature
11
12 //Calculations
13 p2 = p1*exp(-g*(z2-z1)/(R*T));
14 rho2=p2/(R*T);
15 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

```

1 clc ;funcprot(0);
2 //Example 2.3
3
4 //Initializing the variables
5 p1 = 101*10^3; //Initial Pressure
6 z1 = 0; //Initial Height
7 z2 = 1200; //Final Height
8 T1 = 15+273; //Initial Temperature
9 g = 9.81; //Acceleration due to gravity
10 gamma = 1.4; //Heat capacity ratio
11 R = 287; //Gas Constant
12
13 //Calculations
14 p2 = p1*(1-g*(z2-z1)*(gamma-1)/(gamma*R*T1))^(gamma
    /(gamma-1));
15 dT_dZ = -(g/R)*((gamma-1)/gamma);
16 T2 = T1 + dT_dZ*(z2-z1);

```

```
17 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

```
1 clc ;funcprot(0);  
2 //Example 2.4  
3  
4 //Initializing the variables  
5 p1 = 101*10^3; //Initial Pressure  
6 z1 = 0; //Initial Height  
7 z2 = 7000; //Final Height  
8 T1 = 15+273; //Initial Temperature  
9 g = 9.81; //Acceleration due to gravity  
10 R = 287; //Gas Constant  
11 dT = 6.5/1000; //Rate of Variation of Temperature  
12  
13 //Calculations  
14 p2 = p1*(1-dT*(z2-z1)/T1)^(g/(R*dT));  
15 T2 = T1 - dT*(z2-z1);  
16 rho2 = p2/(R*T2);  
17 disp(rho2,"Final Density (kg/m3) :",p2/1000,"Final  
    Pressure (kN/m2) :");
```

Scilab code Exa 2.5 PRESSURE AND HEAD

```
1 clc ;funcprot(0);  
2 //Example 2.5  
3  
4 //Initializing the variables  
5 p = 350*10^3; //Gauge Pressure  
6 pAtm = 101.3*10^3; //Atmospheric Pressure
```

```

7 rhoW = 1000;           //Density of Water
8 sigma = 13.6;         //Relative Density of Mercury
9 g = 9.81; //Acceleration due to gravity
10
11 //Calculations
12 function [head]=Head(rho)
13     head = p/(rho*g);
14 endfunction
15 rhoM = sigma*rhoW;
16 pAbs = p + pAtm;
17
18 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

```

1 clc ;funcprot(0);
2 //Example 2.6
3
4 //Initializing the variables
5 rho = 10^3;           //Density of water
6 h = 2;                //Height
7 g = 9.81;             //Acceleration due to gravity
8
9 //Calculations
10 p=rho*h*g;
11 disp(p/1000,"Gauge pressure (k/m2) :");

```

Scilab code Exa 2.7 PRESSURE MEASUREMENT BY MANOMETER

```

1 clc ;funcprot(0);

```



```

2 //Example 2.7
3
4 //Initializing the variables
5 rho = 0.8*10^3; //Density of fluid
6 rhoM = 13.6*10^3; //Density of manometer liquid
7 g = 9.81; //Acceleration due to gravity
8
9 //Calculations
10 function [P]=fluidPressure(h1,h2)
11     P = rhoM*g*h2-rho*g*h1;
12 endfunction
13
14 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

```

1 clc ; funcprot(0);
2 //Example 2.8
3
4 //Initializing the variables
5 rho = 10^3; //Density of fluid
6 rhoM = 13.6*10^3; //Density of manometer liquid
7 g = 9.81; //Acceleration due to gravity
8 H = 0.3; // Differnce in height = b-a as in text
9 h = 0.7;
10
11 //Calculations
12 result = rho*g*H + h*g*(rhoM-rho);
13 disp( result/1000,"Pressure difference (kN/m2):");

```

Scilab code Exa 2.9 PRESSURE MEASUREMENT BY MANOMETER

```
1 clc ;funcprot(0);
2 //Example 2.9
3
4 //Initializing the variables
5 rho = 10^3; //Density of fluid
6 rhoM = 0.8*10^3; //Density of manometer liquid
7 g = 9.81; //Acceleration due to gravity
8 a = 0.25;
9 b = 0.15;
10 h = 0.3;
11 //Calculations
12 function [P]=PressureDiff(a,b,h,rho,rhoM)
13     P = rho*g*(b-a) + h*g*(rho-rhoM);
14 endfunction
15
16 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

```
1 clc ;funcprot(0);
2 //Example 2.10
3
4 //Initializing the variables
5 phi = 30; //30 degree
6 h = 1.2 ; // Height of tank
7 l = 2; // Length of tank
8
9 //Calculations
10 function [Theta]=SurfaceAngle(a,phi)
```

```
11     Theta = atand(-a*cosd(phi)/(g+a*sind(phi)));
12 endfunction
13
14 //case (a) a = 4
15 disp(tand(SurfaceAngle(4,phi)),"Tan of Angle between
    surface of fluid and horizontal :");
16 disp(180 + SurfaceAngle(4,phi),"ThetaA (degree):");
17
18 //Case (b) a = - 4.5
19 tanThetaR = tand(SurfaceAngle(-4.5,phi));
20 disp(tanThetaR,"Tan of Angle between surface of
    fluid and horizontal :");
21 disp(SurfaceAngle(-4.5,phi),"ThetaR (degree):");
22
23 Depth = h - l*tanThetaR/2;
24 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

```
1  clc ;funcprot(0);
2  //Example 3.1
3
4  //Initializing the variables
5  a = 2.7;    //Upper edge
6  b = 1.2 ;   //Lower edge
7  width = 1.5;    //Width of trapezoidal plate
8  h = 1.1;    //Height of water column above
   surface
9  rho = 1000;
10 g = 9.81; //Acceleration due to gravity
11 phi = 90 ; //Angle between wall and surface
12
13 //Calculations
14 A = 0.5*(a+b)*width;    //Area of Trapezoidal
   Plate
15 y = (2*(0.5*width*0.75)*0.5 + (1.2*width)*0.75)/A;
16 z = y+h; //Depth of center of pressure
```

```

17 R = rho*g*A*z ; //Resultant force
18
19 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
20 D = (sind(phi))^2*I0/(A*z); //depth of center of
      pressure
21 M = R*(1.8533-1.1); //Moment about hinge
22 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

```

1  clc;funcprot(0);
2  //Example 3.2
3
4  //Initializing the variables
5  w = 1.8; //Width of plate
6  h1 = 5; //Height of plate and water in
      upstream
7  h2 = 1.5; //Height of water in downstream
8  rho = 1000;
9  g = 9.81 ; //Acceleration due to gravity
10
11 //Calculations
12 function [F]=waterForce(area,meanHeight)
13     F = rho * g * area * meanHeight;
14 endfunction
15
16 P = waterForce(w*h1,h1/2)-waterForce(w*h2,h2/2); //
      Resultant force on gate
17 x = (waterForce(w*h1,h1/2)*(h1/3) - waterForce(w*h2,
      h2/2)*(h2/3))/P; // point of action of p from
      bottom
18 R = P/(2*sind(20)); // Total Reaction force
19 Rt = 1.18*R/4.8; //Reaction on Top

```

```

20 Rb = R - Rt ;           //Reaction at bottom
21
22 disp(Rb/1000, "Reaction at bottom (kN):", Rt/1000, "
    Reaction at top(kN) :");

```

Scilab code Exa 3.3 PRESSURE DIAGRAMS

```

1  clc;funcprot(0);
2  //Example 3.3
3
4  //Initializing the variables
5  D = 1.8;           //Depth of tank
6  h = 1.2;           //Depth of water
7  l = 3;             //Length of wall of tank
8  p = 35000;        //Air pressure
9  rho = 10^3;        //Density of water
10 g = 9.81;         //Acceleration due to gravity
11
12
13 //Calculations
14 Ra = p*D*l; //Force due to air
15 Rw = .5*(rho*g*h)*h*l; //Force due to water
16 R = Ra + Rw; // Resultant force
17 x = (Ra*0.9+Rw*0.4)/R; // Height of center of
    pressure from base
18 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

```

1  clc;funcprot(0);

```

```

2 //Example 3.4
3
4 //Initializing the variables
5 R = 6; // Radius of arc
6 h = 2*R*sind(30); //Depth of water
7 rho = 10^3; //Density of water
8 g = 9.81; //Acceleration due to gravity
9
10 //Calculations
11 Rh = (rho*g*h^2)/2; // Resultant horizontal force
    per unit length
12 Rv = rho*g*((60/360)*%pi*R^2 -R*sind(30)*R*cosd(30))
    ; // Resultant vertical force per unit length
13 R = sqrt(Rh^2+Rv^2); // Resultant force on gate
14 theta = atand(Rv/Rh); //Angle between resultant
    force and horizontal
15
16 disp(theta,"Direction of resultant force to the
    horizontal(Degree):",R/1000,"Magnitute of
    resultant force(kN/m ) :");

```

Scilab code Exa 3.5 BUOYANCY

```

1 clc;funcprot(0);
2 //Example 3.5
3
4 //Initializing the variables
5 B = 6; // Width of pontoon
6 L = 12; //Length of pontoon
7 D = 1.5; //Draught of pontoon
8 Dmax = 2; //Maximum permissible
    draught
9 rhoW = 1000; //Density of fresh water
10 rhoS = 1025; //Density of sea water
11 g = 9.81; //Acceleration due to gravity

```

```

12
13 // Calculations
14 function [W]=Weight(D)
15     W = rhoW*g*B*L*D;
16 endfunction
17
18 W = Weight(D); // Weight of pontoon in fresh water =
    weight of water displaced
19 Ds = W/(rhoS*g*B*L); // Draught in sea water
20 L = Weight(Dmax) - Weight(D); // maximum load that
    can be supported
21
22 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

```

1  clc;funcprot(0);
2  //Example 3.6
3
4  //Initializing the variables
5  D = 1.8; // Diameter of buoy
6  H = 1.2; // Height of buoy
7  W = 10*10^3; //Weight of buoy
8  L = 2*10^3; //Load
9  G = 0.45; // Center of gravity
10 rho = 1025; //Density of sea water
11 g = 9.81; //Acceleration due to gravity
12
13 // Calculations
14 Z = 4*(W+L)/(rho*g*%pi*D^2); // Depth of
    Immersion
15 BG# = (%pi*D^4/64)/(%pi*D^2*Z/4);
16 Z# = 0.5*Z +BG#; // Position of

```



```

    combined center of gravity
17 Z1 = ((W+L)*Z#-0.45*W)/L;           //Maximum height
    of load above bottom
18
19 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

```

1  clc;funcprot(0);
2  //Example 3.7
3
4  //Initializing the variables
5  l = 20;           // Length of barage
6  b = 6;           //Width of barage
7  r = 3;           //Radius of circular top of barage
8  W = 200*10^3;    //Weight of empty barage
9  d1 = 0.8;        // Depth of water in 1st half
10 d2 = 1;          // Depth of water in 2nd half
11 rho = 1000;      //Density of water
12 R = 0.8;         //Relative density of liquid
13 g = 9.81;        //Acceleration due to gravity
14 ZG = 0.45;       // Center of gravity of barage
15
16 //Calculations
17 I00 = l*b^3/12 +%pi*b^4/128;
18 ICC = l*(.5*b)^3/12;
19 L = d1*rho*g*l*b/2*(d1+d2);       // Weight of liquid
    load
20 W# = L + W;           //Total weight
21 A = l*b +%pi*r^2/2;   // Area of plane of
    waterline
22 V = W#/(rho*g);      // Volume of vessel submerged
23 D = V/A ;           //Depth submerged

```

```
24 ZB = .5*D;           //Height of center of buoyancy
25 NM = ZB-ZG +(1/V)*(I00-R*2*ICC);           //
    Effective metacentric height
26 P = R*rho*g*l*b/2*(d2-d1);           //overturning moment
27 theta = atand(P*1.5/(W#*NM));           //Angle of roll
28
29 disp(theta,"Angle of roll (Degree) :");
```

Chapter 4

Motion of Fluid Particles and Streams

Scilab code Exa 4.1 DISCHARGE AND MEAN VELOCITY

```
1  clc; funcprot(0);
2  //Example 4.1
3
4  //Initializing the variables
5  y = linspace(0,80,9);
6  x = [0 23 28 31 32 29 22 14 0];
7  xlabel('Velocity (m/s)');
8  ylabel('Distance from one side (mm)');
9  xgrid(1);
10
11 //Calculations
12 plot(x,y, '-*');
13 mu=[17.5 26.0 29.6 31.9 30.7 25.4 18.1 7.7];
14           // mean velocity
15 disp(mean(mu),"Mean velocity (m/s):");
16
17 // the plot is attached as 4.1.png
```

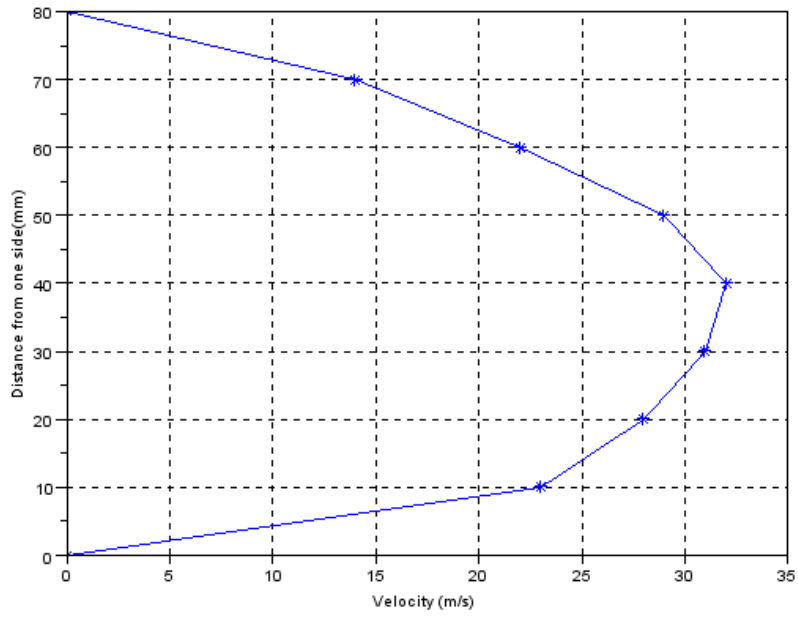


Figure 4.1: DISCHARGE AND MEAN VELOCITY

Scilab code Exa 4.2 CONTINUITY OF FLOW

```
1 clc; funcprot(0); //Example 4.2
2
3 //Initializing the variables all unknowns are
   assigned 0
4
5 d = [0.05 0.075 0 0.030];
6 Q = [0 0 0 0];
7 V = [0 2 1.5 0];
8 //Calculations
9 A2 = %pi*d(2)^2/4;
10 Q(2) =A2*V(2);
11 Q = [Q(2) Q(2) Q(2)/1.5 0.5*Q(2)/1.5];
12 d(3) = (Q(3)*4/(V(3)*%pi))^0.5;
13 A = %pi*d^2/4;
14 V(1) = V(2)*(A(2)/A(1));
15 V(4)=Q(4)/A(4);
16
17 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 DIMEN-
SIONAL FLOW USING CARTESIAN COORDINATES

```
1 clc; funcprot(0);
2 //Example 4.3
3
4 //Initializing the variables
5 vx= -poly(3, 'x');
6 vy = 2*poly(-2, 'x');
```

```
7 vz = -poly(2, 'x');
8
9 // Calculations
10 delVx = derivat(vx);
11 delVy = derivat(vy);
12 delVz = derivat(vz);
13
14 result = derivat(vx)+derivat(vy)+derivat(vz); //
    requirement of continuity equation (result = 0)
15 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

```
1  clc;funcprot(0);
2  //Example 5.1
3
4  //Initializing the variables
5
6  l = 60 ;           //Length of pipeline
7  rho = 1000;       // Density of liquid
8  a = 0.02;         //Acceleration of fluid
9
10 //Calculations
11 delP = rho*l*a;    //Change in pressure
12 disp(delP/1000,"Increase of pressure difference
    required (kN/m2):");
```

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

```

1  clc; funcprot(0);
2  //Example 5.2
3
4  //Initializing the variables
5  v = 5;           //Velocity of jet
6  rho = 1000;     //density of water
7  d = 0.025;     //Diameter of fixed nozzle
8
9  //Calculations
10 //--Part(a) Variation of force exerted normal to the
    plate with plate angle--//
11 header = ["Theta" "          vcos(x)" "          Av" "
    Force"];
12 unit =   [" deg" "          m/s" "          kg/s" "
    N"];
13
14 A = %pi*d^2/4;
15 x = linspace(0,90,7);
16 vcomp = v*cosd(x);
17 m = rho*A*v;
18 ma = linspace(m,m,7);
19 force = rho*A*v^2*cosd(x);
20 value = [x;vcomp;ma;force]' ;
21 disp(value,unit, header );
22
23 //--Part(b) Variation of force exerted normal to the
    plate with plate velocity--//
24 header = ["Theta" " v" " u" " v-u" "          A (v-u
    )" " Force"];
25 unit =   [" deg" " m/s" "m/s" " m/s" "          kg/s
    " "          N"];
26
27 x = linspace(0,0,5);
28 v = linspace(5,5,5);
29 u = linspace(2,-2,5);
30 D = v-u;
31 Prod = rho*A*D;
32 Force = rho*A*D^2;

```



```

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

```

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

```

1  clc;funcprot(0);
2  //Example 5.3
3
4  //Initializing the variables
5  x = 60;           //Angle of deflection
6  rho = 1000;      // Density of liquid
7  V1 = 30;         //Acceleration of fluid
8  V2 = 25;
9  m = .8;          //Discharge through A
10
11 //Calculations
12 function[R] = Reaction(Vin , Vout)
13     R = m*(Vin -Vout) ;
14 endfunction
15
16 Rx = Reaction(V1,V2*cosd(x));
17 Ry = -Reaction(0,V2*sind(x));
18 disp(Rx,"Reaction in X-direction (N):");
19 disp(Ry, "Reaction in Y-direction (N) :");
20 disp(sqrt(Rx^2 +Ry^2), "Net Reaction (N):");
21 disp(atan(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

```

1  clc;funcprot(0);

```

```

2 //Example 5.4
3
4 //Initializing the variables
5 v1 = 36 ; //Exit velocity
6 u = 15; //Velocity of vane\
7 x = 30; // Angle between vanes and flow
8 rho = 1000; // Density of water
9 d = .1; // Diameter of jet
10
11 //Calculations
12 alp = atand(v1*sind(x)/(v1*cosd(x)-u));
13 v2 = 0.85*v1*sind(x);
14 bta = acosd(u*sind(alp)/v2);
15 m = (rho*pi*v1*d^2)/4;
16 Vin = v1*cosd(x);
17 Vout = v2*cosd(90);
18 Rx = m*(Vin-Vout);
19
20 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

```

1 clc;funcprot(0);
2 //Example 5.5
3
4 //Initializing the variables
5 rho = 850 ; // Density of liquid
6 a = 0.02 ; //Acceleration of fluid
7 x = 45 ;
8 d1 = .5 ;
9 d2 = .25;
10 p1 = 40*10^3;
11 p2 = 23*10^3;

```

```

12 Q = .45;
13
14 // Calculations
15 A1 = (%pi*d1^2)/4;
16 A2 = (%pi*d2^2)/4;
17 v1 = Q/A1;
18 v2 = Q/A2;
19
20 Rx = p1*A1 - p2*A2*cosd(x) - rho*Q*(v2*cosd(x)-v1);
21 Ry = p2*A2*sind(x) + rho*Q*v2*sind(x);
22
23 disp(sqrt(Rx^2 +Ry^2)/1000, " Resultant force on the
    bend(kN):");
24 disp(atan(Ry/Rx)," Inclusion of Resultant Force
    with x-direction (Degree) :");

```

Scilab code Exa 5.6 REACTION OF A JET

```

1  clc;funcprot(0);
2  //Example 5.6
3
4  //Initializing the variables
5  v = 4.9;          //Velocity of Jet
6  rho = 1000;      // Density of water
7  d = 0.05;
8  u = 1.2 ;        // Velocity of tank
9  //Calculations
10 Vout = v;
11 Vin = 0;
12 m = rho*pi*d^2*v/4;
13 R = m*(Vout-Vin);
14 disp(R," Reaction (N):");
15 disp(R*u," Work done per second (W):");

```

Scilab code Exa 5.7 REACTION OF A JET

```
1  clc; funcprot(0);
2  //Example 5.7
3
4  //Initializing the variables
5  Vj = 5*10^6;      //Velocity of Jet
6  Mr = 150000;     // Mass of Rocket
7  Mf0 = 300000;    // Mass of initial fuel
8  Vr = 3000;       //Velocity of jet relative to
   rocket
9  g = 9.81;        // Acceleration due to gravity
10
11 //Calculations
12 m = Vj/Vr;       //Rate of fuel consumption
13 T = Mf0/m;       // Burning time
14 function [DVt]=f(t)
15     DVt = m*Vr / (Mr + Mf0 - m*t) - g;
16 endfunction
17
18 function [V]=h(t)
19     V = -g*t - Vr*log(1 - t/269.95);
20 endfunction
21
22 Vt = intg(0, 180 ,f);
23 Z1=intg(0,180,h);
24
25 Z2 = Vt^2/(2*g);
26 disp(T, "(a) Burning time (s): ");
27 disp(Vt, "(b) Speed of rocket when all fuel is burned
   (m/s):");
28 disp((Z2+Z1)/1000, "(c) Maximum height reached (km):")
   ;
```

Scilab code Exa 5.8 REACTION OF A JET

```
1  clc; funcprot(0); // Example 5.8
2
3  // Initializing the variables
4  V = 200;           // Velocity in still air
5  Vr = 700;         // velocity of gas relative to engine
6  mf = 1.1;         // Fuel Consumption
7  r = 1/40 ;
8  P1 = 0;
9  P2 = 0;
10
11 // Calculations
12 m1 = mf/r;
13 T = m1*((1+r)*Vr - V);
14 disp(T/1000, "(a) Thrust (kN) :");
15
16 W = T*V;
17 disp(W/1000, "(b) Work done per second (kW) :");
18
19 Loss = 0.5*m1*(1+r)*(Vr-V)^2;
20 disp(W/(W+Loss)*100, "(c) Efficiency (%) :");
```

Scilab code Exa 5.10 ANGULAR MOTION

```
1  clc; funcprot(0); // Example 5.10
2
3  // Initializing the variables
4  rho = 1000;        // Density of water
5  Q = 10;           // Acceleration of fluid
6  r2 = 1.6;
7  r1 = 1.2;
```

```
8 V1 = 2.3;
9 V2 = 0.2;
10 rot = 240;
11
12 // Calculations
13 Tf = rho*Q*(V2*r2 - V1*r1);
14 T = -Tf;
15 n = rot / 60;
16 P = 2*pi*n*T;
17
18 disp(T, "Torque exerted (N- m):");
19 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

```
1  clc; funcprot(0); //Example 6.1
2
3  //Initializing the variables
4  Pc = 0;           // Atmospheric Pressure
5  Z3 = 30+2;       //height of nozzle
6  Ep = 50 ;        //Energy per unit weight supplied
   by pump
7  d1 = 0.150;      //Diameter of sump
8  d2 = 0.100;      //Diameter of delivery pipe
9  d3 = 0.075 ;    //Diameter of nozzle
10 g = 9.81;       // Acceleration due to gravity
11 Z2 = 2;         //Height of pump
12 rho = 1000;     // Density of water
13
14 // Calculations
15 U3 = sqrt(2*g*(Ep-Z3)/(1+5*(d3/d1)^4 + 12*(d3/d2)^4)
   );
16 U1 = U3/4;
17 Pb = rho*g*Z2 + 3*rho*U1^2;
```

```

18 disp(U3, "Velocity of Jet through nozzle (m/s) :");
19 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

```

1  clc;funcprot(0); //Example 6.2
2
3  //Initializing the variables
4  x = 45;           // Inclination of pipe
5  l = 2;           //Length of pipe under consideration
6  Ep = 50 ;        //Energy per unit weight supplied
    by pump
7  d1 = 0.2;        //Diameter of sump
8  d2 = 0.1;        //Diameter of delivery pipe
9  g = 9.81;        // Acceleration due to gravity
10 rho = 1000;      // Density of water
11 V1 = 2;
12 RD_oil = 0.9;    // relative density of oil
13 RD_Merc = 13.6;  // Relative density of
    Mercury
14
15 //Calculations
16 V2 = V1*(d1/d2)^2;
17 dZ = l*sind(x);
18 rho_Oil = RD_oil*rho;
19 rho_Man = RD_Merc*rho;
20 dP = 0.5*rho_Oil*(V2^2-V1^2) + rho_Oil*g*dZ;
21 h = rho_Oil *( dP/(rho_Oil*g)- dZ)/(rho_Man -
    rho_Oil);
22
23 disp(h, "Difference in the level of mercury (m):",dP
    , "Pressure Difference(N/m2) : ");

```

Scilab code Exa 6.3 PRINCIPLE OF THE VENTURI METER

```
1 clc; funcprot(0); //Example 6.3
2
3 //Initializing the variables
4 d1 = 0.25; //Pipeline diameter
5 d2 = 0.10; //Throat diameter
6 h =0.63; //Difference in height
7 rho = 1000; //Density of water
8 g = 9.81 //Acceleration due to gravity
9
10 //Calculations
11 rho_Hg = 13.6*rho;
12 rho_Oil = 0.9*rho;
13 A1 = (%pi*d1^2)/4; // Area at entry
14 m = (d1/d2)^2; //Area ratio
15 Q = (A1/sqrt(m^2-1))*sqrt(2*g*h*(rho_Hg/rho_Oil -1))
    ;
16
17 disp(Q,"Thepretical Volume flow rate (m3/s ):");
```

Scilab code Exa 6.4 THEORY OF SMALL ORIFICES DISCHARGING TO ATMOSPHERE

```
1 clc; funcprot(0); //Example 6.4
2
3 //Initializing the variables
4
5 x = 1.5;
6 y =0.5;
7 H = 1.2;
8 A = 650*10^-6;
```

```

9 Q =0.117;
10 g = 9.81;
11
12 // Calculations
13 Cv = sqrt(x^2/(4*y*H));
14 Cd = Q / (60*A*sqrt(2*g*H));
15 Cc = Cd/Cv;
16
17 disp(Cc, "Coefficient of contraction :",Cd, "
    Coefficient of Discharge :", Cv, "Coefficient of
    velocity :");

```

Scilab code Exa 6.5 THEORY OF LARGE ORIFICES

```

1 clc;funcprot(0);//Example 6.5
2
3 //Initializing the variables
4 B = 0.7;
5 H1 = 0.4;
6 H2 = 1.9;
7 g =9.81;
8 z = 1.5 ;           // height of opening
9
10 // Calculations
11 Q_Th = 2/3 *B*sqrt(2*g)*(H2^1.5 - H1^1.5);
12 A = z*B;
13 h = 0.5*(H1+H2);
14 Q = A*sqrt(2*g*h);
15
16 disp((Q-Q_Th)*100/Q_Th, "Percentage error in
    discharge (%)");

```

Scilab code Exa 6.6 ELEMENTARY THEORY OF NOTCHES AND WEIRS

```

1  clc;funcprot(0); //Example 6.6
2
3  //Initializing the variables
4  Cd = 0.6;          //Coefficient of discharge
5  Q = 0.28;
6  x = 90;           //Theta
7  g = 9.81;
8  dH = 0.0015;
9
10 //Calculations
11 H = (Q*(15/8)/(Cd*sqrt(2*g)*tand(x/2)))^(2/5)
12 Frac_Q = 5/2 *( dH/H);
13
14 disp(Frac_Q*100, "Percentage error in discharge (%)")
    ;

```

Scilab code Exa 6.7 ELEMENTARY THEORY OF NOTCHES AND WEIRS

```

1  clc;funcprot(0); //Example 6.7
2
3  //Initializing the variables
4  B = 0.9;
5  H = 0.25;
6  alpha = 1.1;
7  g = 9.81;
8
9  //Calculations
10 Q = 1.84 * B * H^(3/2);
11 disp(Q, "Q :");
12
13 i = 1;
14 while(i <= 3)
15     v = Q / (1.2* (H+0.2));
16     disp(v, "V(m/s) :");
17     k = ((1 + alpha*v^2/(2*g*H))^1.5 -(alpha*v

```

```

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

```

Scilab code Exa 6.8 THE POWER OF A STREAM OF FLUID

```

1  clc;funcprot(0); //Example 6.8
2
3  //Initializing the variables
4  rho = 1000;
5  v = 66 ;
6  Q = 0.13;
7  g = 9.81;
8  z =240;
9
10 //Calculations
11 P_Jet = 0.5*rho*v^2*Q;
12 P_Supp = rho*g*Q*z;
13 P_Lost = P_Supp -P_Jet;
14 h = P_Lost/(rho*g*Q);
15 eff = P_Jet/P_Supp;
16
17 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

```

1  clc;funcprot(0); //Example 6.9
2

```

```

3 //Initializing the variables
4 r1 = 0.2;
5 Z1 = 0.500;
6 Z2 = 0.340;
7 g = 9.81;
8 rho = 0.9*1000 ;
9
10 //Calculations
11 r0 = r1*(sqrt(2-2*Z2/Z1));
12 omega = sqrt(2*g*Z1/r0^2);
13
14 function[out]=G(r)
15     out =r^3 - r*r0^2;
16 endfunction
17
18 F = rho*omega^2*pi*intg(r0,r1,G);
19
20 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

```

1 clc;funcprot(0); //Example 6.10
2
3 //Initializing the variables
4 Ra = 0.2;
5 Rb = 0.1;
6 H = 0.18;
7 Za = 0.125;
8
9 //Calculations
10 Y = Ra^2*(H-Za);
11 Zb = H - Y/Rb^2;
12
13 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

```
1  clc;funcprot(0); //Example 7.2
2
3  //Initializing the variables
4  x = 120*(2*%pi)/180; //Theta
5  r = 1;
6  v0 = 0.5;
7  q = 2;
8
9  //Calculations
10 function [y] =shi(r,theta)
11     y = v0*r*sin(theta) +q*theta/(2*%pi);
12 endfunction
13
14
15 //---Approx differentiation at a point using central
    difference formula---//
16 h=0.0000001;
17 at_theta=x;
18 at_r = r;
19 Vr = (shi(r,at_theta+h)-shi(r,at_theta-h))/(r*2*h);
```

```

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

```

Scilab code Exa 7.3 COMBINED SOURCE AND SINK FLOWS DOU- BLET

```

1  clc;funcprot(0); //Example 7.3
2
3  //Initializing the variables
4  q = 10;
5  function[Z] = shi(x,y)
6      Z = (q/2/%pi)*(atan(y/(x-1))-atan(y/(x+1))) -
          25*y;
7  endfunction
8  h = 0.0000001;
9  Vinf = 25;
10
11 //Calculations
12 x = poly(0,'x');
13 f = x^2 - 2/(5*%pi) -1;
14 root = roots(f);
15 l = abs(root(1))+abs(root(2));
16 Ymax = 0.047;
17 width = 2*Ymax;
18 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.
19 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.
20

```



```

21 V = sqrt(Vx^2+Vy^2);
22 rho = poly(0, 'rho');
23 dP = rho/2 *(V^2 - Vinf^2); //difference in pressure
24
25 disp(dP, 'Pressure Difference(N/m2) :',V, 'Velocity
      (m/s):', 1, 'Length of Rankine Body(m ) :', width
      , 'Width of Rankine Body (m):' );

```

Scilab code Exa 7.4 FLOW PAST A CYLINDER

```

1 funcprot(0);clc; //Example 7.4
2
3 //Initializing the variables
4 a = 0.02;
5 r = 0.05;
6 V0 = 1;
7 x = 135; // Theta
8 function[Z] = shi(r,x)
9     Z = V0*sind(x)*(r - ((a^2)/r));
10 endfunction
11 h = 0.0001;
12
13 //Calculations
14 Vr = 57*(shi(r,x+h)-shi(r,x))/(r*h);
15 Vx = -1*(shi(r+h,x)-shi(r,x))/h;
16
17 disp(Vr, 'Radial Velocity (m/s):',Vx, 'Normal
      component of velocity (m/s):');

```

Scilab code Exa 7.5 CURVED FLOWS AND THEIR COMBINATIONS

```

1 clc;funcprot(0); //Example 7.5
2

```

```
3 //Initializing the variables
4 rho = 1000;
5 r = 2;
6 psi = 2*log(r);
7
8 //Calculations
9 y = psi/log(r); // y = GammaC / 2*pi
10 v = y/r;
11 dPbydr = rho*v^2/r;
12 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

```
1  clc; funcprot(0) //Example 8.1
2
3  //Initializing the variables
4  P1 = 57; //Power in SI
5  M = 1/14.6; //Ratio of mass in SI/
   British
6  L = 1/0.3048; //Ratio of length in SI/
   British
7  T = 1; //Ratio of time in SI/British
8
9  //Calculations
10
11 P2 = M*(L^2)*(T^-3)*P1 ; //Power in kW
12 P2/P1;
13
14 disp(P2/P1," Conversion Factor :",P2, 'Power(Kw) :'
   );
```

Chapter 9

Similarity

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

```
1 clc; funcprot(0); //Example 9.1
2
3 //Initializing the variables
4 Vp = 10;
5 LpByLm = 20;
6 rhoPbyRhoM = 1;
7 muPbymuM = 1;
8 //Calculations
9 Vm = Vp*LpByLm*rhoPbyRhoM*muPbymuM;
10
11 disp(Vm, 'Mean water tunnel flow velocity (m/s):');
```

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

```
1 funcprot(0); clc; //Example 9.2
2
```

```

3 //Initializing the variables
4 Vp = 3;
5 LpByLm = 30;
6 rhoPbyRhoM = 1;
7 muPbymuM = 1;
8
9 //Calculations
10 Vm = Vp*LpByLm*rhoPbyRhoM*muPbymuM;
11
12 disp(Vm, 'Mean water tunnel flow velocity (m/s):');

```

Scilab code Exa 9.3 ZONE OF DEPENDENCE OF MACH NUMBER

```

1 funcprot(0);clc; //Example 9.3
2
3 //Initializing the variables
4 Vp = 100;
5 cP = 340;
6 cM = 295;
7 rhoM = 7.7;
8 rhoP = 1.2;
9 muM = 1.8*10^-5;
10 muP = 1.2*10^-5;
11
12 //Calculations
13 Vm = Vp*(cM/cP);
14 LmByLp = 1/((Vm/Vp)*(muM/muP)*(rhoM/rhoP));
15 FmByFp = (rhoM/rhoP)*(Vm/Vp)^2*(LmByLp)^2;
16
17 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

```

1 funcprot(0);clc; //Example 9.4
2
3 //Initializing the variables
4 function [Z] =pLossRatio(RatRho,RatMu,RatL)
5     Z = RatRho*RatMu^2*RatL^2;
6 endfunction
7
8 //Calculations
9 //Case (a) : water is used
10 RatRho = 1;
11 RatMu = 1;
12 RatL = 10;
13 disp(pLossRatio(RatRho,RatMu,RatL), "(a) Ratio of
    pressure losses between the model and the
    prototype if water is used ");
14
15 RatRho = 1000/1.23;
16 RatMu = 1.8*10^-5/10^-3;
17
18 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

```

1 funcprot(0);clc; //Example 9.5
2
3 //Initializing the variables
4 scale = 1/50;
5 ratArea = scale^2;
6 Qp = 1200;
7
8 //Calculations
9 LmByLp = sqrt(ratArea);

```

```

10 VmByVp = sqrt(LmByLp);
11 Qm = Qp*ratArea*VmByVp;
12
13 disp(Qm, "Water flow rate (m3/s ):");

```

Scilab code Exa 9.6 SIMILARITY APPLIED TO ROTODYNAMIC MACHINES

```

1 funcprot(0);clc; //Example 9.6
2
3 //Initializing the variables
4 Qa = 2;
5 Na = 1400;
6 rhoA = 0.92;
7 rhoS = 1.3;
8 DaByDs = 1;
9 dPa = 200;
10
11 //Calculations
12 Ns = Na*(rhoA/rhoS)*(DaByDs);
13 Qs = Qa*(Ns/Na);
14 dPs = dPa *(rhoS/rhoA)*(Ns/Na)^2*(1/DaByDs)^2;
15
16 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

```

1 funcprot(0);clc; //Example 9.8
2
3 //Initializing the variables
4 V = 300 ;// Volume rate
5 w = 3;

```

```
6 d = 65;
7 l = 30;
8 scaleH = 30/1000/18;
9 scaleV = 1/60;
10 ZmByZr = 1/60;
11 LmByLr = 1/600;
12 rho = 1000;
13 mu = 1.14*10^-3;
14
15 // Calculations
16 Vr = V/(w*d);
17 Vm = Vr*sqrt(ZmByZr);
18 m = (w*d*scaleH*scaleV)/(d*scaleH + 2*w*scaleV);
19 Rem = rho*Vm*m/mu;
20 TmByTr = LmByLr*sqrt(1/ZmByZr);
21 Tm = 12.4*60*TmByTr;
22
23 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

```
1  clc; funcprot(0); //Example 10.1
2
3  //Initializing the variables
4  mu = 0.9;
5  rho = 1260;
6  g = 9.81;
7  x = 45; //theta in degrees
8  P1 = 250 * 10^3;
9  P2 = 80* 10^3;
10 Z1 = 1;
11 Z2 = 0; // datum
12 U = -1.5;
13 Y = 0.01;
14
15 //Calculations
16 gradP1 = P1+ rho*g*Z1;
17 gradP2 = P2+ rho*g*Z2;
18 DPstar = (gradP1-gradP2 )*sind(x)/(Z1-Z2);
```

```

19 A = U/Y; // Coefficient U/Y for equation 10.6
20 B = DPstar/(2*mu); // Coefficient dp*/dx X(1/2mu) for
    equation 10.6
21 y = poly(0, 'y');
22 v = (A + B*Y)*y -B*y^2;
23 duBYdy = derivat(v);
24 tau = 0.9*duBYdy;
25 ymax = roots(duBYdy); //value of y where
    derivative vanishes.;
26 umax = (A + B*Y)*ymax + B*ymax^2; // Check the value
    there is slight mistake in books answer
27 function [z] = u(y)
28     z = (A + B*Y)*y -B*y^2;
29 endfunction
30 tauMax =abs( mu*derivative(u,Y));
31 ymax
32 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 LAM-
 INAR FLOW IN CIRCULAR CROSS SECTION PIPES

```

1 clc; funcprot(0); //Example 10.2
2
3 //Initializing the variables
4 mu = 0.9;
5 rho = 1260;
6 d = 0.01;
7 Q = 1.8/60*10^-3; //Flow in m^3 per second
8 l = 6.5;
9 ReCrit = 2000;
10 //Calculations
11 A = (%pi*d^2)/4;
12 MeanVel = Q/A;

```

```

13 Re = rho*MeanVel*d/mu; // Check properly the answer
    in book there is something wrong
14 Dp = 128*mu*l*Q/(%pi*d^4)
15 Qcrit = Q*ReCrit/Re*10^3;
16 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

```

1  clc; funcprot(0); //Example 10.3
2
3  //Initializing the variables
4  mu = 1.14*10^-3;
5  rho = 1000;
6  d = 0.04;
7  Q = 4*10^-3/60; //Flow in m^3 per second
8  l = 750;
9  ReCrit = 2000;
10 g = 9.81;
11 k = 0.00008; // Absolute Roughness
12
13 //Calculations
14 A = (%pi*d^2)/4;
15 MeanVel = Q/A;
16 Re = rho*MeanVel*d/mu;
17 Dp = 128*mu*l*Q/(%pi*d^4);
18 hL = Dp/(rho*g);
19 f = 16/Re;
20 h1Da = 4*f*l*MeanVel^2/(2*d*g); // By Darcy Equation
21 Pa = rho*g*h1Da*Q;
22
23 //Part (b)
24 Q = 30*10^-3/60; //Flow in m^3 per second
25 MeanVel = Q/A;

```

```

26 Re = rho*MeanVel*d/mu;
27 RR = k/d; // relative roughness
28 f = 0.008 ;//by Moody diagram for Re = 1.4 x 10^4
    and relative roughness = 0.002
29 h1Db = 4*f*l*MeanVel^2/(2*d*g); // By Darcy Equation
30 Pb = rho*g*h1Db*Q;
31 disp(Pb, "Power Required (W) :",h1Db , "Head Loss(m)
    :", "!----- Case(b) -----!", Pa, "Power Required (W)
    :", h1Da*1000 , "Head Loss (mm) :", " !----- Case (a)
    -----!");

```

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

```

1 clc; funcprot(0); //Example 10.4
2
3 //Initializing the variables
4 w = 4.5;
5 d = 1.2 ;
6 C = 49;
7 i = 1/800;
8
9 //Calculations
10 A = d*w;
11 P = 2*d + w;
12 m = A/P;
13 v = C*sqrt(m*i);
14 Q = v*A;
15
16 disp(Q," Discharge (m3/s):",v, "Mean Velocity (m/s):"
    );

```

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

```
1 clc; funcprot(0); //Example 10.5
2
3 //Initializing the variables
4 R = poly(0, 'R');
5
6 //Calculations
7 r = R*(1 - (49/60)^7);
8
9 disp(r,"Radius at which the actual velocity is equal
    to the mean velocity :");
```

Scilab code Exa 10.7 SEPARATION LOSSES IN PIPE FLOW

```
1 clc; funcprot(0); //Example 10.7
2
3 //Initializing the variables
4 d1 = 0.140;
5 d2 = 0.250;
6 DpF_DpR = 0.6; //Difference in head loss when in
    forward and in reverse direction
7 K = 0.33 ;//From table
8 g = 9.81;
9 //Calculations
10 ratA = (d1/d2)^2;
11
12 v = sqrt(DpF_DpR*2*g/((1 - ratA)^2 - K));
13
14 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

```
1  clc; funcprot(0); //Example 11.1
2
3  //Initializing the variables
4  rho = 860;
5  v = 10^-5;
6  Us = 3;
7  b = 1.25;
8  l = 2;
9
10 //Calculations
11 x = 1; // At x =1
12 Rex = Us*x/v;
13 ReL = Us*l/v ;
14 mu = rho*v;
15 T0 = 0.332*mu*Us/x*Rex^0.5;
16 Cf = 1.33*ReL^-0.5;
17 F = rho*Us^2*l*b*Cf ;
18
19 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

```
1  clc; funcprot(0); //Example 11.2
2
3  //Initializing the variables
4  Us = 6;
5  b = 3;
6  l = 30;
7  rho = 1000;
8  mu = 10^-3;
9  T = 20+273; // Temperature in Kelvin
10
11 //Calculations
12 1/mu;
13 ReL = rho*Us*l/mu;
14 Cf = 0.455*log10(ReL)^-2.58 ;
15
16 F = rho*Us^2*l*b*Cf ;
17 Lt = 10^5*mu/(rho*Us); // Assuming transition at Rel
    = 10^5
18 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

```
1  clc; funcprot(0); // Example 12.1
2
3  // Initializing the variables
4  x = 35;
5  T = 50;
6  m = 1;
7  g = 9.81;
8  rho = 1.2;
9  A = 1.2;
10 U0 = 40*1000/3600; // Velocity in m/s
11
12 // Calculations
13 L = T*sind(x)+m*g;
14 D = T*cosd(x);
15 Cl = 2*L/(rho*U0^2*A);
16 Cd = 2*D/(rho*U0^2*A);
17 disp(Cd," Drag Coefficient :",Cl," Lift Coefficient :");
    );
```

Scilab code Exa 12.2 RESISTANCE OF SHIPS

```
1  clc; funcprot(0); //Example 12.2
2
3  //Initializing the variables
4  Vp =12;
5  lp = 40;
6  lm = 1;
7  As = 2500;
8  Dm = 32;
9  rhoP = 1025;
10 rhoM = 1000;
11 Ap = As;
12
13 //Calculations
14 Am = As/40^2;
15 Vm = Vp*sqrt(lm/lp);
16 Dfm = 3.7*Vm^1.95*Am;
17 Rm = Dm - Dfm;
18 Rp = Rm *(rhoP/rhoM)*(lp/lm)^2*(Vp/Vm)^2;
19 Dfp = 2.9*Vp^1.8*Ap;
20 Dp = Rp + Dfp;
21
22 disp(Dp/1000,"Expected total resistance (kN) :");
```

Scilab code Exa 12.3 FLOW PAST A CYLINDER

```
1  clc; funcprot(0); //Example 12.3
2
3  //Initializing the variables
4  U0 = 80*1000/3600;
5  d = 0.02;
```

```

6 rho =1.2;
7 mu = 1.7*10^-5;
8 A = 0.02*500; // Projected area of wire
9 N = 20; // No of cables
10
11 //Calculations
12 Re = rho*U0*d/mu;
13 Cd = 1.2; // From figure 12.10 for given Re;
14 D = 0.5*rho*Cd*A*U0^2
15 F = N*D;
16 f = 0.198*(U0/d)*(1-19.7/Re);
17
18 disp(f,"Frequency (Hz):",F/1000,"Total force on
tower (kN):");

```

Scilab code Exa 12.4 FLOW PAST A SPHERE

```

1 clc; funcprot(0); //Example 12.4
2
3 //Initializing the variables
4 mu = 0.03;
5 d = 10^-3;
6 rhoP = 1.1*10^3;
7 g = 9.81;
8 rho0 = 0.9*10^3;
9 //Calculations
10 B = 18*mu/(d^2*rhoP);
11 t = 4.60/B;
12 Vt = d^2*(rhoP - rho0)*g/(18*mu);
13 Re = rho0*Vt*d/mu;
14
15
16 disp(Re,"Reynolds No corresponding 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

```
1  clc; funcprot(0); //Example 12.5
2
3  //Initializing the variables
4  mu0 = 0.0027;
5  Vt = 3*10^-3;
6  rhoW = 1000;
7  rhoP = 2.4*rhoW;
8  rho0 = 0.9*rhoW;
9  g = 9.81;
10 muA = 1.7*10^-5;
11 rhoA = 1.3;
12
13 //Calculations
14 d = sqrt(18*mu0*Vt/(rhoP-rho0)/g);
15 Re = Vt*d*rho0/mu0;
16
17 //Movement of particle in upward direction
18 if(Re < 1 ) then
19     v = 0.5;
20     Re = 5 ; // from fig 12.15
21     vt = muA*Re/(rhoA*d);
22     u = vt+v;
23     disp(u ,” Velocity of air stream blowing
           vertically up (m/s) :”);
24 else
25     disp(” strokes law is not valid”);
26 end
```

Scilab code Exa 12.6 FLOW PAST AN AEROFOIL OF FINITE LENGTH

```

1  clc; funcprot(0); //Example 12.6
2
3  //Initializing the variables
4  c = 2;
5  s = 10;
6  rho = 5.33;
7  rho_ellip = 1.2;
8  D = 400;
9  L = 45000;
10 scale = 20;
11 U_windTunnel = 500;
12 U_proto = 400*1000/3600;
13
14 //Calculations
15 A = c*s;
16 U_model = U_windTunnel/scale;
17 Cd = D/(0.5*rho*U_model^2*A);
18 Cl = L/(0.5*rho_ellip*U_proto^2*A); // Considering
    elliptical Lift model
19 Cdi = Cl^2/(%pi*s/c); // Aspect Ratio = s/c
20 Cdt = Cd + Cdi;
21 Dw = 0.5*Cdt*rho_ellip*U_proto^2*A;
22 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

```
1  clc; funcprot(0); //Example 13.1
2
3  //Initializing the variables
4  rho0 = 1.8;
5  R = 287;
6  T = 75+273; // Temperature in kelvin
7  gma = 1.4;
8  Ma = 0.7;
9
10 //Calculations
11 P0 = rho0*R*T;
12 c = sqrt(gma*R*T);
13 V0 = Ma*c;
14 Pt = (P0^(((gma-1)/gma) + rho0*((gma-1)/gma)*(V0
           ^2/(2*P0^(1/gma))))^(gma/(gma-1)));
15 rhoT = rho0*(Pt/P0)^(1/gma);
16 Tt = Pt/(R*rhoT)-273;
17
18 disp(rhoT,"Density of airstream (kg/M3):",Tt,"
```

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

Scilab code Exa 13.2 SHOCK WAVES

```
1  clc; funcprot(0); //Example 13.2
2
3  //Initializing the variables
4  R = 287;
5  T = 28+273;
6  gma = 1.4;
7  P = 1.02*10^5;
8  rhoHg = 13.6*10^3;
9  g = 9.81;
10
11 //Calculations
12 //Case(a)
13 U0 = 50;
14 c = sqrt(gma*R*T);
15 Ma = U0/c;
16 rho = P/(R*T);
17 DelP = 0.5*rho*U0^2; //Pt-P
18 ha = DelP/(rhoHg*g);
19
20 //Case(b)
21 U0 = 250;
22 Ma = U0/c;
23 Pt = P*(1+(gma-1)*Ma^2/2)^(gma/(gma-1));
24 DelP = Pt-P
25 hb = DelP/(rhoHg*g);
26
27 //Case (c)
28 U0 = 420;
29 Ma1 =U0/c;
30 P2 = P*((2*gma/(gma+1))*Ma1^2 - ((gma-1)/(gma+1)));
```

```

31 N = Ma1^2 + 2/(gma-1); // Numerator
32 D = 2*gma*Ma1^2/(gma-1) - 1;
33 Ma2 = sqrt(N/D);
34 Pt2 = P2*(1+(gma-1)*Ma2^2/2)^(gma/(gma-1));
35 hc = (Pt2-P2)/(rhoHg*g) ;
36
37 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

```
1  clc; funcprot(0); //Example 14.1
2
3  //Initializing the variables
4  L1 = 5;
5  L2 = 10;
6  d = 0.1;
7  f = 0.08;
8  Za_Zc = 4;           //difference in height between A
                        and C
9  g = 9.81 ;
10 Pa = 0;
11 Va = 0;
12 Za_Zb = -1.5;
13 V = 1.26;
14 rho = 1000;
15
16 //Calculations
17 D = 1.5 + 4*f*(L1+L2)/d ; // Denominator in case of
```



```

    v^2
18 v = sqrt(2*g*Za_Zc/D);
19 Pb = rho*g*Za_Zb - rho*V^2/2*(1.5+4*f*L1/d);
20 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

```

1  clc; funcprot(0); //Example 14.3
2
3  //Initializing the variables
4  Za_Zb = 10;
5  f = 0.008;
6  L = 100;
7  d1 = 0.05;
8  g = 9.81;
9  d2 = 0.1;
10 //Calculations
11 function[z] = flowRate(d)
12     D = 1.5 + 4*f*L/d ; // Denominator in case of
        v1^2
13     A = %pi*d^2/4;
14     v = sqrt(2*g*Za_Zb/D);
15     z = A*v;
16 endfunction
17
18 Q1 = flowRate(d1);
19 Q2 = flowRate(d2);
20
21 Q = Q1+Q2;
22 D = poly(0, 'D');
23 v = 4*Q/(%pi*D^2);
24 X = 1.5 + 4*f*L/D;
25 f = 10*2*g/(X*v^2) - 1;

```

```

26 f = numer(f) ;
27 diameter = roots(f); // Taking roots of numerator
    denominator will be multiplied by zero
28 i = 1;
29 while (i<=length(diameter))
30     x = diameter(i);
31     if(imag(x) == 0) then
32         dia = diameter(i);
33         i= i+1;
34     else
35         i = i+1;
36     end
37 end
38
39 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 BRANCH-
ING PIPES THE THREE RESERVOIR PROBLEM**

```

1  clc; funcprot(0); //Example 14.4
2
3  //Initializing the variables
4  Za_Zb = 16;
5  Za_Zc = 24;
6  f = 0.01;
7  l1 = 120;
8  l2 = 60;
9  l3 = 40;
10 d1 = 0.12;
11 d2 = 0.075;
12 d3 = 0.060;
13 g = 9.81;
14 //Calculations

```

```

15
16 A = [%pi*d1^2/4 %pi*d2^2/4 %pi*d3^2/4]
17 function[z] = Coeff(1,d)
18     z = 4*f*l/(d*2*g);
19 endfunction
20
21 function[f] = F(x)
22     f(1) = Coeff(11,d1)*x(1)^2 + Coeff(12,d2)*x(2)^2
           - Za_Zb;
23     f(2) = Coeff(11,d1)*x(1)^2 + Coeff(13,d3)*x(3)^2
           - Za_Zc;;
24     f(3) = x(1)*d1^2 - x(2)*d2^2 - x(3)*d3^2; // Q1=
           Q2
25 endfunction
26
27 function[j] = jacob(x)
28     j(1,1) = 2*Coeff(11,d1)*x(1); j(1,2) = 2*Coeff(
           12,d2)*x(2);j(1,3) = 0;
29     j(2,1) = 2*Coeff(11,d1)*x(1); j(2,2) = 0;j(2,3)
           = 2*Coeff(13,d3)*x(3);
30     j(3,1) = d1^2; j(3,2) = -d2^2;j(3,3) = -d3^2;
31 endfunction
32
33 x = [1.8 0 0];
34 v = fsolve(x,F,jacob, 10^-20);
35 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

```

1 clc; funcprot(0); //Example 14.5
2
3 //Initializing the variables

```

```

4 D = 0.3;
5 Q = 0.8;
6 rho = 1.2;
7 f = 0.008;
8 L_entry = 10;
9 L_exit = 30;
10 Lt = 20*D; // Transition may be represented by a
      separation loss equivalent length of 20 the
      approach duct diameter
11 K_entry = 4;
12 K_exit = 10
13 l = 0.4; // length of cross section
14 b = 0.2; // width of cross section
15
16 // Calculations
17 A = %pi*D^2/4;
18 Dp1 = 0.5*rho*Q^2/A^2*(K_entry + 4*f*(L_entry+Lt)/D)
      ;
19 area = l*b;
20 perimeter = 2*(l+b);
21 m = area/perimeter;
22 Dp2 = 0.5*rho*Q^2/area^2*(K_exit + f*L_exit/m);
23 Dfan = Dp1+Dp2;
24
25 disp(Dfan,"fan Pressure input (N/m2) :");

```

Scilab code Exa 14.6 INCOMPRESSIBLE STEADY FLOW IN DUCT NETWORKS

```

1 clc; funcprot(0); //Example 14.6
2
3 //Initializing the variables
4 D = [0.15 0.3];
5 rho = 1.2;
6 f = 0.008;

```

```

7 L_entry = 10;
8 L_exit = 20;
9 Lt = 20*D(2); // Transition may be represented by a
    separation loss equivalent length of 20 the
    approach duct diameter
10 K = 4;
11 Q1 = 0.2;
12
13 // Calculations
14 Q2 = 4*Q1;
15 A = %pi*D^2/4;
16 Dp1 = 0.5*rho*Q1^2/A(1)^2*(K + 4*f*L_entry/D(1));
17 Dp2 = 0.5*rho*Q2^2/A(2)^2*(4*f*(L_exit + Lt)/D(2));
18 Dfan = Dp1+Dp2;
19
20 disp(Dfan,"fan Pressure input (N/m2) :");

```

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

```

1 clc; funcprot(0); // Example 14.7
2
3 // Initializing the variables
4 d = [0.1 0.125 0.15 0.1 0.1 ]; // Corrospounding to
    AA1B AA2B BC CD CF
5 l = [30 30 60 15 30]; //
    Corrospounding to AA1B AA2B BC CD CF
6 rho = 1.2;
7 f = 0.006;
8 Ha = 100;
9 Hf = 60;
10 He = 40;
11
12 // Calculations
13 for(i=1:length(d))

```

```

14     K(i) = f*l(i)/(3*d(i)^5);
15 end
16
17 K_ab = K(1)*K(2)/(sqrt(K(1))+sqrt(K(2)))^2;
18 K_ac = K_ab + K(3);
19 Hc = (K_ac*Hf +K(5)*Ha/4)/(K_ac+K(5)/4);
20 Q = sqrt((Ha - Hc)/K_ac);
21
22 function [z] = f(n)
23     z = He - Hc + (0.5*Q)^2 *(K(4)+(4000/n)^2);
24 endfunction
25
26 n = fsolve(1,f);
27
28 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

```

1  clc; funcprot(0); //Example 14.8
2
3  //Initializing the variables
4  d = [0.3 0.25 0.2];
5  l = [1500 800 400];
6  f = 0.01;
7  Ha = 60;
8  Hb = 30;
9  Hc = 15;
10 Hd = 35; // Assumption
11 H(1) = Ha - Hd;
12 H(2) = Hb - Hd;
13 H(3) = Hc - Hd;
14
15 //Calculations

```

```

16 K = 0;
17 for(i=1:length(d))
18     K(i) = f*l(i)/(3*d(i)^5);
19 end
20 Qsum = 0.001;
21 for(i=1:2)
22     Q = 0; Qby2h = 0; Qs = 0; Qby2hsum = 0;
23     for(i=1:3)
24         if(imag(sqrt(H(i)/K(i)))~=0) then
25             Q(i) = -1*abs(imag(sqrt(H(i)/K(i))));
26         else
27             Q(i) = sqrt(H(i)/K(i));
28         end,
29
30         Qby2h = Q(i)/(2*H(i));
31         Qs = Qs+Q(i);
32         Qby2hsum = Qby2hsum +Qby2h
33
34     end
35     dH = Qs/Qby2hsum;
36     for(i=1:3)
37         H(i)=H(i)+dH;
38     end
39     Qsum = Qs;
40 end
41
42 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

```

1 clc; funcprot(0); //Example 14.9
2
3 //Initializing the variables
4 As = 6;

```

```

5 d = 0.02;
6 f =0.01;
7 L = 1.5;
8 K = 0.9;
9 g = 9.81;
10
11 // Calculations
12 Ap = %pi*d^2/4;
13 function[y] = Qinvt(h)
14     y = sqrt((4*f*L/d +K+1)/(2*g*h))/Ap;
15 endfunction
16 //By direct integration
17 t = -As*intg(3.5,2.25,Qinv); // Discharge is 2 m
    below
18 disp(t, "Time of discharge by direct integration (s)
    : ");
19
20 // By numerical integrations
21 interval = [0.250 0.125 0.0083 0.0063 0.005 0.0042];
22 for(i=1:length(interval))
23
24     start=3.5;piece=3.5:-interval(i):2.25;
25     X=-As*integrate('Qinv(h)', 'h', start, piece);
26
27     disp(X(length(X)), "Value of t (s) : ", interval(i)
        ), " -----For Interval(Dh in m) -----");
28 end

```

Scilab code Exa 14.12 QUASI STEADY FLOW

```

1 clc; funcprot(0); //Example 14.12
2
3 //Initializing the variables
4 As = 6;
5 A2 = 4.5;

```



```

6 d = 0.02;
7 f =0.01;
8 L = 1.5;
9 K = 0.9;
10 g = 9.81;
11
12 // Calculations
13 Ap = %pi*d^2/4;
14 C = Ap*sqrt(2*g/(4*f*L/d+K+1));
15
16 function[y] = Qinvs(h)
17     y = sqrt(1/h)/(C*(1+As/A2));
18 endfunction
19
20 //By direct integration
21 t = -As*intg(3.0,2.0,Qinvs); // Discharge is 2 m
    below
22 disp(t, "Time of discharge by direct integration (s)
    : ");
23
24 //By Numerical Integration
25 interval = [0.250 0.125 0.0083 0.0063 0.005 0.0042];
26 for(i=1:length(interval))
27
28     start=3.0;piece=3.5:-interval(i):2.0;
29     X=-As*integrate('Qinvs(h)', 'h', start, piece);
30
31     disp(X(length(X)), "Value of t (s): ", interval(i)
        , " -----For Interval(Dh in m) -----");
32 end

```

Chapter 15

Uniform Flow in Open Channels

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

```
1  clc; funcprot(0); //Example 15.1
2
3  //Initializing the variables
4  B =4;
5  D = 1.2;
6  C = 7.6;
7  n = 0.025;
8  s = 1/1800;
9
10 //Calculations
11 W = B + 2*1.5*D;
12 A = D*(B+C)/2; // Area of parallelogram formed
13 P = B +2*1.2*sqrt(D^2 +(1.5D)^2);
14 m =A/P;
15 i=s;
16 C = (23+0.00155/i+1/n)/(1+(23+0.00155/i)*n/sqrt(m));
    // By Kutter formula
17 Q1 = C*A*sqrt(m*i);
```

```

18 Q2 = A*(1/n)*m^(2/3)*sqrt(i);
19 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

```

1  clc; funcprot(0); //Example 15.2
2
3  //Initializing the variables
4  Q = 0.5;
5  C = 80;
6  i = 1/2000;
7
8  //Calculations
9  function [y] = f(D)
10     y = (7/4)*C*D^(5/2)*sqrt(i/2) -Q;
11 endfunction
12 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

```
1  clc; funcprot(0); //Example 16.2
2
3  //Initializing the variables
4  B = [1.4 0.9];
5  D = [0.6 0.32];
6  g = 9.81;
7  h = 0.03;
8  Z = 0.25;
9
10 //Calculations
11 Q1 = B(2)*D(2)*sqrt(2*g*h/(1-(B(2)*D(2)/B(1)*D(1))
    ^2))
12 E = D(1)-Z;
13 Q2 = 1.705*B(2)*E^1.5;
14 disp(Q2 , "Volume flow rate (m3/s) :");
```

Scilab code Exa 16.3 CLASSIFICATION OF WATER SURFACE PROFILES

```
1  clc; funcprot(0); //Example 16.3
2
3  //Initializing the variables
4  a =0.5;
5  b = 0.5;
6  Dn = 1.2;
7  s = 1/1000;
8  C = 55;
9  g = 9.81;
10
11 //Calculations
12 c = (1 + a)/b;
13 QbyB = Dn*C*sqrt(Dn*s);
14 q = QbyB;
15 Dc = (q^2/g)^(1/3);
16
17 m = 2.4:-0.15:1.35;
18 total = 0;Dm = 0; N = 0; D = 0; Lm = 0;
19 for(i=1:length(m)-1)
20
21     Dm(i)= (m(i)+m(i+1))/2;
22     N(i) = 1 - (Dc/Dm(i))^3 ; // Numerator
23     D(i) = 1 - (Dn/Dm(i))^3; // Denominator
24     Lm(i) = 150*(N(i)/D(i));
25     total = total +Lm(i);
26 end
27 result = [Dm N D Lm];
28 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

```
1  clc; funcprot(0); //Example 17.1
2
3  //Initializing the variables
4  g = 9.81;
5  rho = 1000;
6  rhoHg = 13.6*rho;
7  d1 = 0.075;
8  d2 = 0.025;
9  Pi = 0.250;
10 Pt = 0.150;
11 P_Hg = 0.760;
12 rho1 = 1.6;
13 gma = 1.4;
14
15 //Calculations
16 P1 = (Pi+P_Hg)*rhoHg*g;
17 P2 = (Pt+P_Hg)*rhoHg*g;
18 rho2 = rho1*(P2/P1)^(1/gma);
19
20 function [f] = velocity(V)
21     f(1) = d2^2*V(2)*rho2-d1^2*V(1)*rho1;
```

```

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

```

Scilab code Exa 17.2 THE LAVAL NOZZLE

```

1  clc; funcprot(0); //Example 17.2
2
3  //Initializing the variables
4  Ma = 4;
5  gma = 1.4;
6  At = 500; // in mm
7
8  //Calculations
9  N = 1 + (gma-1)*Ma^2/2;
10 D = (gma+1)/2 ;
11 A = At*(N/D)^((gma+1)/(2*(gma-1)))/Ma;
12
13 disp(A, "Area of test section (mm2):");

```

Scilab code Exa 17.3 NORMAL SHOCK WAVE IN A DIFFUSER

```

1  clc; funcprot(0); //Example 17.3
2
3  //Initializing the variables
4  Ma1 = 2;
5  gma = 1.4;
6  T1 = 15+273; // In kelvin
7  P1 = 105;

```

```

8
9 // Calculations
10 Ma2 = sqrt(((gma-1)*Ma1^2 +2)/(2*gma*Ma1^2-gma+1));
11 P2 = P1*(1+gma*Ma1^2)/(1+gma*Ma2^2);
12 T2 = T1*(1 +(gma-1)/2*Ma1^2)/(1 +(gma-1)/2*Ma2^2);
13 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 FRIC-
TION UNDER ADIABATIC CONDITIONS FANNO FLOW**

```

1 clc; funcprot(0); //Example 17.4
2
3 //Initializing the variables
4 gma = 1.4;
5 f = 0.00375;
6 d = 0.05;
7
8 //Calculations
9 m = d/4;
10 function [y] = x(Ma)
11     A =(1 -Ma^2 )/(gma*Ma^2);
12     B = (gma+1)*Ma^2/(2+(gma-1)*Ma^2);
13     y = m/f*(A+ (gma+1)*log(B)/(2*gma));
14 endfunction
15
16 X1 = x(0.2); // At entrance Ma = 0.2;
17 X06_X1 =x(0.6); // Section(b) Ma = 0.6;
18
19 X06 = X1-X06_X1;
20 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

```
1  clc; funcprot(0); //Example 17.4
2
3  //Initializing the variables
4  gma = 1.4;
5  Q = 28/60; // m3/s
6  d = 0.1;
7  p1 = 200*103;
8  f = 0.004;
9  x_x1 = 60;
10 R = 287;
11 T = 15+273;
12
13 //Calculations
14 A = %pi*d2/4;
15 m = d/4;
16 v1 = Q/A;
17 pa = p1*sqrt(1-f*(x_x1)*v12/(m*R*T));
18
19 function [y] =g(p)
20     A = (v1*p1)2/(R*T)
21     B = f*A*x_x1/(2*m);
22     y = (p2 - p12)/2 -A*log(p/p1) +B;
23 endfunction
24 pb=fsolve(pa,g);// Guessing solution around pa
25 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

```
1  clc; funcprot(0); //Example 20.3
2
3  //Initializing the variables
4  c = 1250;
5  Dt = 0.02;
6  Dv = 0.5;
7  rho = 1000;
8  v =0.5;
9
10 //Calculations
11 cDt = c*Dt;
12 Dp = rho*c*Dv;
13 D0v_D0t = Dv/Dt;
14 vD0v_D0t = v*Dv/cDt;
15 D0p_D0t = Dp/Dt;
16 vD0p_D0x = v*Dp/cDt;
17 Error = [vD0v_D0t*100/D0v_D0t vD0p_D0x*100/D0p_D0t];
18 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

```
1  clc; funcprot(0); //Example 20.5
2
3  //Initializing the variables
4  f = 0;
5  Atunnel = 1.227;
6  Ashaft = 12.57;
7  Q =2;
8  L = 200;
9  g = 9.81;
10
11 //Calculations
12 Zmax = (Q/Ashaft)*sqrt(Ashaft*L/(Atunnel*g));
13 T = 2*%pi*sqrt(Ashaft*L/(Atunnel*g));
14 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

```
1  clc; funcprot(0); //Example 22.1
2
3  //Initializing the variables
4  Q = 5;
5  R1 = 1.5/2;
6  R2 = 2/2;
7  w = 18;
8  rho = 1000;
9  rhoA = 1.2;
10 Hth = 0.017;
11 g=9.81;
12
13 //Calculations
14 A = %pi*(R2^2-R1^2);
15 Vf = Q/A;
16 Ut = w*R2;
17 Uh = w*R1;
18 B1t = acotd(Ut/Vf);
19 B1h = acotd(Uh/Vf);
```

```

20 E = Hth*rho/rhoA;
21 function[y] = Beta(u)
22     y = acotd((u-E*g/u)/Vf);
23 endfunction
24 B2t = Beta(Ut);
25 B2h = Beta(Uh);
26
27 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

```

1  clc; funcprot(0); //Example 22.2
2
3  //Initializing the variables
4  D = 0.1;
5  t = 15*10^-3;
6  Q = 8.5/3600;
7  N = 750/60;
8  B2 = 25; // Beta 2 ind degrees
9  g = 9.81;
10 z = 16;
11
12 //Calculations
13 A = %pi*D*t;
14 V_f2 = Q/A;
15 U2 = %pi*N*D;
16 V_w2 = U2 - V_f2*cotd(B2);
17 Hth = U2*V_w2/g;
18 Sf = 1 - %pi*sind(B2)/(z*(1-(V_f2/U2)*cotd(B2)));
19 H = Sf*Hth;
20
21 disp(H, "Part (b) - Head developed (m): ",Hth, "Part

```

(a) – Head developed (m): ");

Scilab code Exa 22.3 COMPRESSIBLE FLOW THROUGH ROTODYNAMIC MACHINES

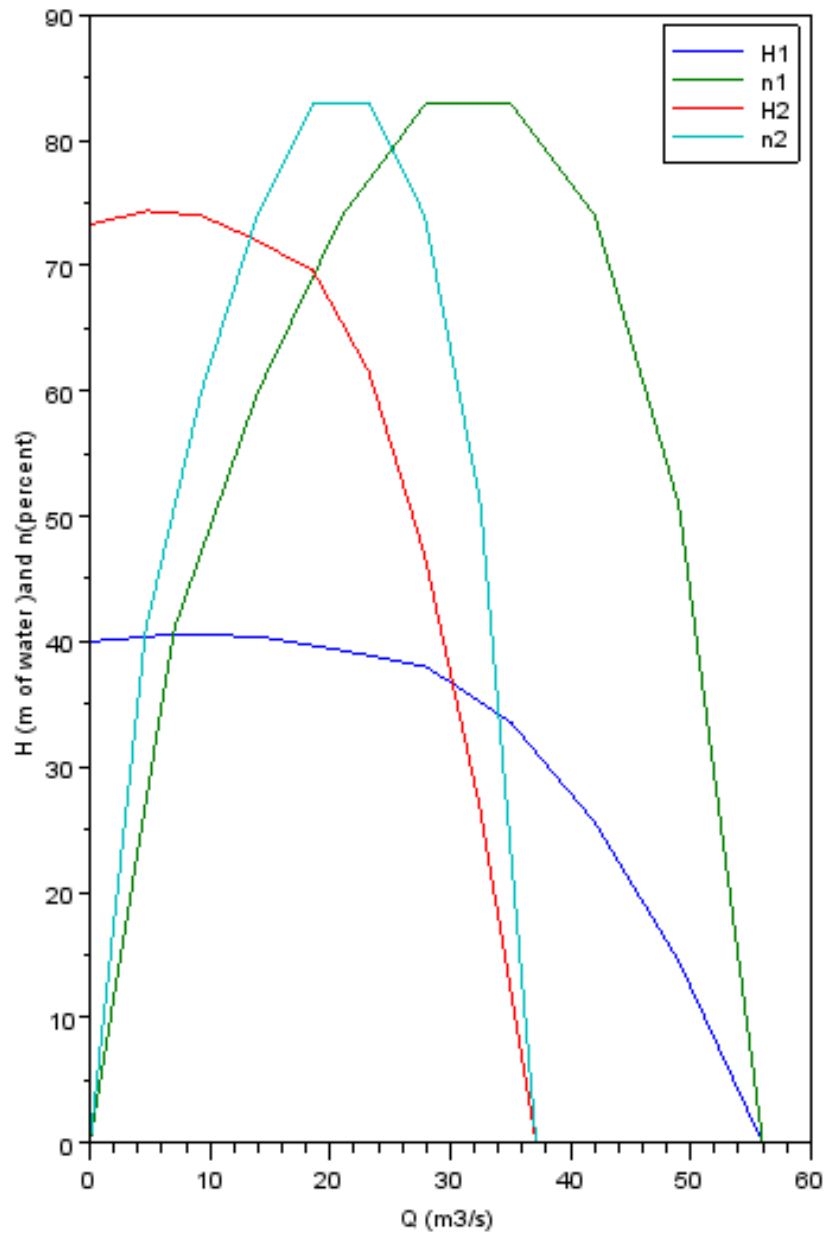
```
1  clc; funcprot(0); //Example 22.3.
2
3  //Initializing the variables
4  Ma = 0.6;
5  Cl = 0.6;
6  tByC = 0.035; // Thickness to chord ratio
7  cByC = 0.015; // Camber to chord ratio
8  x = 3; // Angle of incidence
9
10 //Calculations
11 lamda = 1/sqrt(1-Ma^2);
12 Cl# = lamda*Cl;
13 tByC1 = tByC*lamda;
14 cByC1 = cByC*lamda;
15 Cl1 = Cl*lamda^2;
16 Ae = x*lamda;
17
18 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

```
1  clc; funcprot(0); //Example 23.1
2
3  //Initializing the variables
4  Q = [0:7:56];
5  H = [40 40.6 40.4 39.3 38 33.6 25.6 14.5 0];
6  n = [0 41 60 74 83 83 74 51 0];
7  N1 = 750;
8  N2 = 1450;
9  D1 = 0.5;
10 D2 = 0.35;
11
12 //Calculations
13 Q2 = Q*(N2/N1)*(D2/D1)^3;
14 H2 = H*(N2/N1)^2*(D2/D1)^2;
15 xlabel("Q (m3/s)");
16 ylabel("H (m of water )and n(percent)");
```



95

Figure 23.1: DIMENSIONLESS COEFFICIENTS AND SIMILARITY LAWS


```

17 plot(Q,H,Q,n,Q2,H2,Q2,n);
18
19 legend("H1","n1","H2","n2");

```

Scilab code Exa 23.2 THE PELTON WHEEL

```

1  clc; funcprot(0); //Example 23.2
2
3  //Initializing the variables
4  n = 0.9;
5  g = 9.81;
6  D = 1.45;
7  N = 375/60;
8  H = 200; // Real height
9  x = 165; // Theta
10 P = 3750*10^3;
11 rho = 1000;
12
13 //Calculations
14 h = n*H; // Effective Head
15 v1 = sqrt(2*g*h);
16 u = %pi*D*N;
17
18 n_a = (2*u/v1^2)*(v1-u)*(1-n*cosd(x));
19
20 P_b = P/n_a;
21 ppj = P_b/2; // Power per jet
22 d = sqrt(8*ppj/(rho*%pi*v1^3)) ;
23
24 disp(d,"Diameter of Jet (m) :",n_a*100, "E fficiency
    (%):")

```

Scilab code Exa 23.3 FRANCIS TURBINES

```

1  clc; funcprot(0); //Example 23.3
2
3  //Initializing the variables
4  g = 9.81;
5  H = 12;
6  n = 0.8;
7  w = 300*2*%pi/60;
8  Q = 0.28;
9
10 //Calculations
11 V_f1 = 0.15*sqrt(2*g*H);
12 V_f2 =V_f1;
13 V_w1 = sqrt(n*g*H);
14 u1 = V_w1;
15 theta = atand(V_f1/u1);
16 u2 =0.5*u1;
17 B2 = atand(V_f2/u2);
18 r1 = u1/w;
19 b1 = Q/(V_f2*0.9*2*%pi*r1); // vanes occupy 10 per
   cent of the circumference hence 0.9
20 b2 = 2*b1;
21
22 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

```

1  clc; funcprot(0);//Example 23.4
2
3  //Initializing the variables
4  H = 35;
5  g = 9.81;
6  D = 2;

```

```

7 N = 145/60;
8 z = 30; // angle between vanes and direction of
runner rotation
9 y = 28; // angle between runner blades at the outlet
.
10
11 // Calculations
12 H_net = 0.93*H ; // since 7% head is lost
13 v1 = sqrt(2*g*H_net);
14 u = %pi*N*D;
15 V_r2 = u*cosd(y);
16 V2= u*sind(y);
17 V_w2 = V2*sind(y);
18
19 // Function to solve the vector for Vr1 and B1 by
just re writing the parallelogram law in arranged
form
20 function [f] = F(x)
21     f(1) = u^2 + x(1)^2 + 2*u*x(1)*cosd(x(2))-v1^2;
22     f(2) = x(1)*sind(x(2)) - tand(z)*(u + x(1)*cosd(
x(2)));
23 endfunction
24 X = [10 50]; // An innitial guess of vector length
and angle by figure
25 result=fsolve(X,F);
26 V_r1 =result(1);
27 B1 = result(2);
28 V_w1 = u + V_r1*cosd(B1)
29 E = (u/g)*(V_w1 - V_w2);
30 n = E/H;
31 disp(n*100,"Efficiency (%) :",B1, "Blade angle at
inlet (Degree) :");

```

Scilab code Exa 23.5 HYDRAULIC TRANSMISSIONS

```

1  clc; funcprot(0);
2  // Example 23.5
3
4  //Initializing the variables
5  s = 0.03;
6  P = 185*103;
7  rho = 0.86*103;
8  A = 2.8*10-2;
9  N = 2250/60;
10 D = 0.46;
11
12 //Calculations
13 R0 = 0.46/2;
14 Ws_Wp = 1-s;
15 n = Ws_Wp;
16 Pf = s*P;
17 Q = (2*Pf*A2/(3.5*rho))(1/3);
18 Wp = 2*%pi*N;
19 Ri = sqrt((1/Ws_Wp)*(R02 -P/(rho*Q*Wp2))); //
    Modified equation for power transmission.
20 Di = 2*Ri;
21 T = P/(rho*Wp3 *D5);
22
23 disp(T,"Torque Coefficient :", Di*1000,"Mean
    diameter (mm) : ");

```

Chapter 24

Positive Displacement Machines

Scilab code Exa 24.1 RECIPROCATING PUMPS

```
1  clc; funcprot(0);
2  //Example 24.1
3
4  //Initializing the variables
5  H_at = 10.3;
6  Hs = 1.5;
7  Hd = 4.5;
8  Ls = 2;
9  Ld = 15;
10 g = 9.81;
11 Ds = 0.4; // Diameter of stroke
12 Db = 0.15; // Diameter of bore
13 Dd = 0.05; // Diameter of discharge and suction
    pipe
14 nu = 0.2;
15 f = 0.01;
16 abs_pump_pressure = 2.4;
17
18 //Calculations
```

```

19 A = %pi*(Db)^2/4;
20 a = %pi*(Dd)^2/4;
21 r = Ds/2;
22 W = 2*%pi*nu;
23 Hsf = 0;
24 function[y] = H_suck(n) // n for checking the sign
    of Hsi = 4fl/2dg *(vA/a)^2
25 y = H_at - Hs +(-1)^n*(L/g)*(A/a)*W^2*r;
26 endfunction
27
28 function[y] = H(n,DischargeOrSuction)// n for
    checking the sign of Hsi = 4fl/2dg *(vA/a)^2, for
    suction 1 and for discharge2
29     if(DischargeOrSuction == 1) then
30         y = H_at - Hs +(-1)^n*(Ls/g)*(A/a)*W^2*r;
31     elseif(DischargeOrSuction == 2) then
32         y = H_at + Hd +(-1)^n*(Ld/g)*(A/a)*W^2*r;
33     else disp("There is something wrong :")
34     end
35 endfunction
36
37 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
38
39     if(DischargeOrSuction == 1) then
40         Hsf = 4*f*Ls/(2*Dd*g)*(uA/a)^2;
41         y = H_at - Hs - Hsf;
42     elseif(DischargeOrSuction == 2) then
43         Hsf = 4*f*Ld/(2*Dd*g)*(uA/a)^2;
44         y = H_at + Hd + Hsf;
45     else disp("There is something wrong :")
46     end
47 endfunction
48
49 Hs_start = H(1,1); // Inertia head negative
    hence n = 1
50 Hs_end = H(2,1); // Inertia head positive hence

```

```

    n = 2
51 Hd_start = H(1,2);
52 Hd_end = H(2,2);
53 u = W*r;
54 Hs_mid = H_mid(1,u*A);
55 slip = 0.04;
56 Hd_mid = H_mid(2,u*A);
57 suction = [Hs_start Hs_end Hs_mid];
58 discharge = [Hd_start Hd_end Hd_mid];
59 header = ["    Start(m)", "    End(m)", "    Mid(m)"];
60 W_max = sqrt((abs_pump_pressure - H_at + Hs)*(g/Ls)
    *(a/A)*(1/r));
61 W_max_rev = W_max/(2*pi)*60; // maximum rotation
    speed in rev/min
62 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

```

1  clc; funcprot(0);
2  //Example 24.2
3
4  //Initializing the variables
5  H_friction = 2.4;
6  H_at = 10.3;
7  Hs = 1.5;
8  L = 2;
9  f = 0.01;
10 d = 0.05;
11 g = 9.81;
12 Ds = 0.4; // Diameter of stroke
13 Db = 0.15; // Diameter of bore
14 r = 0.2;

```

```
15
16 // Calculations
17 A = %pi*(Db)^2/4;
18 a = %pi*(Dd)^2/4;
19 W= sqrt((H_at - Hs - H_friction )*(2*d*g/(4*f*L)))
    *(a/A)*(%pi/r);
20 W_rev = W/(2*%pi)*60; // maximum rotation speed in
    rev/min
21
22 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

```
1 clc; funcprot(0);
2
3 //-----Important Note
  //-----//
4 // Please keep intersectFunc.sci in the same folder
  in which this file//
5 // is kept and change the current working directory
  to the directory //
6 // in which both the files are kept using chdir "
  absolute path" //
7 //
  -----
8
9
```

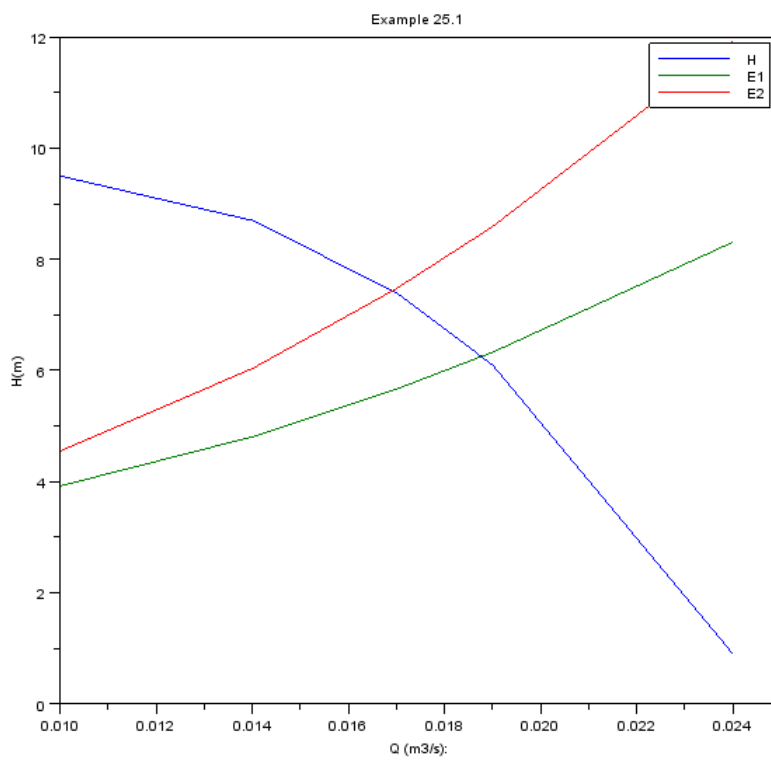


Figure 25.1: FANS PUMPS AND FLUID NETWORKS

```

10 //Example 25.1
11
12 //Initializing the variables
13 exec("intersectFunc.sci");
14 Q =[0.010 0.014 0.017 0.019 0.024]';
15 H = [9.5 8.7 7.4 6.1 0.9]';
16 n = [65 81 78 68 12]';
17 d = 0.15;
18 mu = 1.14*10^-3;
19 rho = 1000;
20 g = 9.81;
21
22 //Calculations
23 E1 = 3+9218*Q^2; // f = 0.0025 from moody chart
24 Q1 = intersectFunc(H,E1,Q);
25 v1 = 4*Q1/(%pi*d^2);
26 Re1 = rho*v1*d/mu;
27 E2 = 3+15486*Q^2; // s ince f = 0.0042
28 Q2 = intersectFunc(H,E2,Q);
29 n = 0.78; // efficiency at Q2 from graph
30 H1 = 7.45; // From Graph
31 P = rho*g*H1*Q2/n;
32
33 title("Example 25.1");
34 xlabel("Q (m3/s):");
35 ylabel("H(m)");
36 plot(Q,H, Q,E1,Q,E2);
37 legend("H", "E1", "E2");
38
39 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

```

1  clc; funcprot(0);
2  // Example 25.4
3
4  //Initializing the variables
5  Pa_P1 = -200; // From previous Question
6  Q = 1.4311 ; // From previous questions.
7
8  //Calculations
9  DpSys = Pa_P1 + 98.9*Q^2;
10 disp(DpSys,"System Operating point (m3/s):");

```

Scilab code Exa 25.7 JET FANS

```

1  clc; funcprot(0);
2  //Example 25.7
3
4  //Initializing the variables
5  Vo = 25.3;           //Outlet velocity
6  D = 10 ;            // Mean hydraulic diameter
7  f = 0.008;          // friction factor
8  X = 1000;           // Length of road
9  P = 12600;          // Absorbing power
10 Va = 300;           // Tunnel air flow
11 K1 = 0.96;
12 K2 = 0.9;
13 T = 590;            //Thrust
14 rho = 1.2;          // Air density
15
16 //Calculations
17 alpha = (1/D)^2;
18 A = %pi*D^2/4;      // Area of tunnel
19 Vt = Va/A;
20 W = Vo/Vt;          //Omega
21 E = (1-alpha*W);
22 C = (1-alpha*W)*(1-E)^2 + E^2 - 1;

```

```

23 // Manipulating equation 25.20;
24 LHS = f*X*(E+1)^2/D + C + 1 ;
25 n = poly(0,"n");
26 RHS = K1*(2*((alpha*W^2 + (1-alpha)*E^2-1)+(n-1)*(
      alpha*W*(W-1)-C/2)));
27 Equation = RHS -LHS;
28 roots(Equation);
29
30 // Alternative approach using equation 25.22
31 n = (rho*((4*f*X*Vt^2)/(2*D) + 1.5*Vt^2/2))*A/(K1*
      K2*T);
32 Pt = round(n)*P;
33 disp(Pt/1000,"Total power consumed (KW) :", round(n
      ), "Number of fans required :");

```

Scilab code Exa 25.8 JET FANS

```

1  clc; funcprot(0);
2  //Example 25.8
3
4  //Initializing the variables
5  f = 0.008;
6  T = 290;
7  L = 750;
8  Dt = 9;           // Diameter Tunnel
9  Df = 0.63;       // Diameter fan
10 K1 = 0.98;
11 K2 = 0.92;
12 Vo = 27.9;
13 n = 10;
14
15 //Calculations
16 alpha = (Df/Dt)^2;
17 // equation 25.20 becomes when E = 1 nad C = 0
18 W = poly(0, 'W' );

```

```

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

```

Scilab code Exa 25.9 CAVITATION IN PUMPS AND TURBINES

```

1  clc; funcprot(0);
2  //Example 25.9
3
4  //Initializing the variables
5  Ws = 0.45;
6  Ks = 3.2;
7  H = 152;
8  h = 0;
9  Hatm = 10.3;
10 Pv = 350; //vapour pressure
11 g = 9.81;
12 rho = 1000;
13
14 //Calculations
15 Ht1 = H*(Ws/Ks)^(4/3)
16 Hvap = Pv/(rho*g);
17 Z = Hatm -h -Hvap -Ht1;
18 disp(Z,"Elevation of pump (m):");

```

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

```
1  clc; funcprot(0);
2  //Example 25.11
3
4  //Initializing the variables
5  Co = 0;
6  Qc = 0.0024;
7  V = 5400;
8  c = 10;
9  //Calculations
10 function [y] = partA(n)
11     Ci = 10;
12     t = 10^1000; // infinity (a very large number)
13     Q = V*n/3600;
14     y = (Co + 10000*Qc/Q)*(1-%e^(-n*t)) + Ci*%e^(-n
        *t) - c;
15 endfunction
16
17 Sol_A = fsolve(1,partA);
18
19 function [y] = partB(n)
20     Ci = 0;
21     t = 1; // time in hours
22     Q = V*n/3600;
23     A = Co + 10000*Qc/Q;
24     B = Ci*%e^(-n*t) - c;
25     y = A*(1-%e^(-n*t)) + B;
26 endfunction
27
28 Sol_B = fsolve(1,partB);
29
30 function [y] = partC(c)
31     Ci = 0;
32     n = 1;
33     t = 0.333333; // 20 minutes in hours
34     Q = V*n/3600;
```

```

35     y = (Co + 10000*Qc/Q)*(1-%e^(-n*t)) + Ci*%e^(-n
        *t) - c;
36 endfunction
37
38 Sol_C = fsolve(1,partC);
39
40 function[y] = partD(t)
41     Ci = 10;
42     n = 1;
43     c = 0.1;
44     y = Ci*%e^(-n*t) - c;
45 endfunction
46
47 Sol_D = fsolve(0.001 , partD);
48
49
50 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

```
1 //*****intersectFunc scilab
   function*****//
2 //Takes argument as three arrays namely f1,f2(
   functions) and their domain //
3 //It finds intersecting points in all the subdomains
   //
4 //Gives output as point(s) of intersection of the
   two functions //
5 // Domain should be INCREASING
6 //Created by : Jay Chakra (www.jaychakra.co.cc) //
   //
7 //           Undergraduate
8 //           //
   //           Aerospace Engineering
9 //           //
   //           IIT Bombay
10 //           //
   //Comments, suggestions, bugs welcomed at jaychakra.
   jc@gmail.com //
11 //
   *****
12
```

```

13 function [y] = intersectFunc(f1,f2,domain)
14     L1 = length(f1);
15     L2 = length(f2);
16     L3 = length(domain);
17     if((L1~=L2)|(L1~=L3)|(L2~=L3)) then
18         error("Check Dimensions of input parameters
19             !! ");
19     else
20         R = 1;
21         y = [];
22         for(i=1:L1-1)
23             N1 = f1(i+1)-f1(i);
24             N2 = f2(i+1)-f2(i);
25             D = domain(i+1)-domain(i);
26             x = poly(0, 'x');
27             //Writing equation of straight lines
28             //joining the terminal points
29             f(1) = f1(i) +(N1/D)*(x-domain(i));
30             f(2) = f2(i) +(N2/D)*(x-domain(i));
31             difference = f(2) - f(1);
32             root = roots(difference); //
33             //Solution will be the roots of
34             //difference
35             if(difference == 0) then //
36                 if both functions are same
37                     y = domain;
38                     break;
39             elseif (root ~= [] & root <= domain(i+1) &
40                 root >= domain(i)) then //if roots
41                 lie in the subdomain
42                     y(R) = root;
43                     R=R+1;
44             end,
45         end,
46     end
47 endfunction

```
