Scilab Textbook Companion for
Fluid Mechanics
by Y. A. Cengel and J. M. Cimbala[1]

Created by
Karthikeyan S
B.E
Computer Engineering
Nandha Engineering college
College Teacher
None
Cross-Checked by
None

July 31, 2019

---

[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: Y. A. Cengel and J. M. Cimbala

Publisher: McGraw-Hill, New Delhi

Edition: 1

Year: 2006

ISBN: 0072472367

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

Chapter 1

INTRODUCTION AND BASIC CONCEPTS

Scilab code Exa 1.3 `Obtaining Formulas from Unit Considerations`

```
// Example 1_3
clc;clear;funcprot(0);
// Given values
rho=850; // Density of oil in kg/m^3
V=2; // Volume of the tank in m^3

// Calculation
m=rho*V;// kg
printf('The amount of mass in the tank,m=%0.0fkg\n',
   m);
```

Scilab code Exa 1.4 `The Weight of One Pound to Mass`

```
// Example 1_4
clc;clear;funcprot(0);
//Properties
```

```
g=32.174; // The gravitational constant in ft/s^2

//Given values
m=1; //  Mass in lbm

// Calculation
W=m*g/32.174;// Weight is mass times the local value of gravitational acceleration
printf('The weight of the object in earth ,W =%0.2f lbf\n',W);
```

Scilab code Exa 1.5 Solving a System of Equations with EES

```
//Example 1_5
clc;clear;funcprot(0);
//Given relations
//  x-y=4;
//x^2+y^2=x+y+20;

//Solution
// Assume x=y(1);y=y(2);
function[X]=unknowns(y);
    X(1)=y(1)-y(2)-4;
    X(2)=y(1)^2+y(2)^2-y(1)-y(2)-20;
endfunction
y=[1 1];
z=fsolve(y,unknowns);
printf('x=%0.0f \n',z(1));
printf('y=%0.0f \n',z(2));
```

Scilab code Exa 1.6 Significant Digits and Volume Flow Rate

```
// Example 1_6
```

```
clc;clear;funcprot(0);
//Given values
dv=1.1//The volume of water collected in gal
dt=45.62;// Time period in s

//Calculation
V=dv/dt;// gal/s
V=V*(3.785*10^-3*60);// m^3/min
printf('The volume flow rate of water through the hose,V=%0.1e m^3/min\n',V);
```

Chapter 2

PROPERTIES OF FLUIDS

Scilab code Exa 2.1 `Density Specific Gravity and Mass of Air in a Room`

```
// Example 2_1
clc;clear;funcprot(0);
// Given values
p=100; // The pressure of air in kPa
T=25+273; // The temperature of air in K
R=0.287; // The gas constant of air in (kPa.m^3)/(kg
    .K)

//Calculation

rho=p/(R*T); // From ideal gas relation
printf('The density of air,rho =%0.2f kg/m^3\n',rho)
    ;

rho_1=1000; // Density of water in kg/m^3
SG=rho /rho_1; // The specific gravity of air
printf('The specific gravity of air,SG =%0.5f \n',SG
    );

V=4*5*6; // The volume of air in m^3
m=rho*V; // The mass of air in the room in kg
```

```
printf('The mass of air ,m =%0.0f kg\n',m);
```

Scilab code Exa 2.2 `Minimum Pressure to Avoid Cavitation`

```
// Example 2_2
clc;clear;funcprot(0);
//Properties
P_sat=4.25;//The vapor pressure of water at 30 C  in KPa
//Analysis
P_min=P_sat;
printf('The minimum pressure allowed in the system to avoid cavitation P_min=%0.2f kPa\n',P_min);
```

Scilab code Exa 2.3 `Variation of Density with Temperature and Pressure`

```
// Example 2_3
clc;clear;funcprot(0);
//Given values
T_1=20;// degree celsius
T_2=50;//degree celsius
P_1=1;// atm
P_2=100;//atm
rho_1=998.0;// The density of water in kg/m^3

//Properties
//The coefficient of volume expansion at the average temperature T_avg=35 C
beta=0.337*10^-3;// k^-1
alpha=4.80*10^-5;//The isothermal compressibility of water in atm^-1

// Calculation
```

```
//(a)
gradT=(T_2-T_1);// K
gradrho=-(beta*rho_1*gradT);// The change in density in kg/m^3
rho_2=rho_1+gradrho;// The density of water at 50 C and 1 atm in kg/m^3
printf('The density of water at 50 C  and 1 atm is rho_2 =%0.0f kg/m^3\n',rho_2);
//(b)
gradP=(P_2-P_1);
gradrho=alpha*rho_1*gradP;// kg/m^3
rho_2=rho_1+gradrho;//The density of water at 100 atm and 20 C  in kg/m^3
printf('The density of water at 100 atm and 20 C  is rho_2 =%0.1f kg/m^3\n',rho_2);
```

Scilab code Exa 2.4 `Determining the Viscosity of a Fluid`

```
// Example 2_4
clc;clear;funcprot(0);
// Given values
l=0.0015; // Gap between two cylinders in m
T=1.8;  // Torque in N.m
L=.4;  //  Length  in m
R=.06;  // Outer radius of inner cylinder in m
n=300/60;  // Number of revolutions per unit time (seconds)

//Calculation
mu=(T*l/(4*%pi^2*R^3*n*L)); // Viscosity of the fluid in N.s/m^2
printf('The viscosity of the fluid ,mu =%0.3f N.s/m^2\n',mu);
```

Scilab code Exa 2.5 The Capillary Rise of Water In a Tube

```
// Example 2_5
clc;clear;funcprot(0);
// Given values
T=20;// degree celsius
sigma_s=0.073; // the surface tension of water in N/m
phi=0; // the contact angle of water with glass in degree
rho=1000;// kg/m^3
g=9.81;// m/s^2
R=0.3*10^-3; // Radius of glass tube in m

//Calculation
h=((2*sigma_s)/(rho*g*R))*cos(phi);// the capillary rise of water in m
h=h*100;// m to cm
printf('The capillary rise of water in a tube h=%0.1f cm\n',h);
```

Chapter 3

PRESSURE AND FLUID STATICS

Scilab code Exa 3.1 `Absolute Pressure of a Vacuum Chamber`

```
// Example 3_1
clc;clear;funcprot(0);
// Given values
P_atm=14.5; // The atmospheric pressure in psi
P_vac=5.8; // The vacuum pressure in psi

//Calculation
P_abs=P_atm-P_vac;
printf('The absolute pressure in the chamber,P_abs=%0.1f psi\n',P_abs);
```

Scilab code Exa 3.2 `Measuring Pressure with a Manometer`

```
//Example 3_2
clc;clear;funcprot(0);
//Given values
```

```
SG=0.85;// Specific gravity of manometer fluid
h=0.55;// The manometer column height in m
P_atm=96;// Local atmospheric pressure in kPa
rho_w=1000;// The density of water in kg/m^3
g=9.81;// The acceleration due to gravity in m/s^2

//Calculation
rho=SG*rho_w;// kg/m^3
P=P_atm+(rho*g*h)/1000; // The pressure of the fluid in kPa
printf('The absolute pressure with in the tank,P=%0.1f kPa\n',P);
```

Scilab code Exa 3.3 `Measuring Pressure with a Multifluid Manometer`

```
// Example 3_3
clc;clear;funcprot(0);
// Constants used
g=9.81;//The acceleration due to gravity in m/s^2

// Given values
h=1400;//m
h_1=0.1;//m
h_2=0.2;//m
h_3=0.35;// respective heights in m
P_atm=85.6; // The atmosphereic pressure in kPa;
rho_w=1000;// kg/m^3
rho_o=850;// kg/m^3
rho_m=13600; // The density of water, mercury and oil in kg/m^3

// Calculation
P_1=P_atm+((rho_m*g*h_3)-(rho_w*g*h_1)-(rho_o*g*h_2))/1000;
printf('The air pressure in the tank P_1=%0.0fkPa\n'
```

```
    ,P_1);
```

Scilab code Exa 3.4 Analyzing a Multifluid Manometer with EES

```
// Example 3_4
clc;clear;funcprot(0);
// Constants used
g=9.81;//The acceleration due to gravity in m/s^2

// Given values
h=1400;//m
h_1=0.1;// m
h_2=0.2;// m
h_3=0.35;// respective heights in m
P_atm=85.6*1000; // The atmospheric pressure in Pa;
rho_w=1000;// kg/m^3
rho_o=850;// kg/m^3
rho_hg=13600; // The density of water,oil and
    mercury in kg/m^3

//Calculation
P_1r=(P_atm+((rho_hg*g*h_3)-(rho_w*g*h_1)-(rho_o*g*
    h_2)));// Modified equation
P_1=P_1r/1000;// Pa to kPa
printf('The air pressure in the tank P_1~=%0.0f kPa\
    n',P_1);
rho_hg~=1030;// kg/m^3
H_3=(P_atm-P_1r-(rho_w*g*h_1)-(rho_o*g*h_2))/(-
    rho_hg*g);
printf('The height of the fluid column, h_3=%0.2f m\
    n',H_3);
```

Scilab code Exa 3.5 Measuring Atmospheric Pressure with a Barometer

```
// Example 3_5
clc;clear;funcprot(0);
// Given values
h=0.740;// m
g=9.81;// The gravitational acceleration in m/s^2
T=10;// degree celsius
rho=13570;// The density of mercury in kg/m^3

// Calculation
P_atm=(rho*g*h)/1000;// 1kPa=1000 N/m^2
printf('The atmospheric pressure,P_atm=%0.1f kPa\n',
   P_atm);
```

Scilab code Exa 3.6 `Effect of Piston Weight on Pressure in a Cylinder`

```
// Example 3_6
clc;clear;funcprot(0);
// Given values
m=60;// Mass in kg
A=0.04;// The Cross sectional area in m^2
P_atm=.97;// Local atmospheric pressure in bar
g=9.81;// The gravitational acceleration in m/s^2

// Calculation
//(a)
P=P_atm+((m*g)/A)/10^5;// 1 bar=10^5 N/m^2
printf('The gas pressure in the piston cylinder P=%0
   .2f bars\n',P);
```

Scilab code Exa 3.7 `Hydrostatic Pressure in a Solar Pond with Variable Density`

```
//Example 3_7
clc;clear;funcprot(0);
```

```
//Constants used
g=9.81;// The acceleration due to gravity in m/s^2

//Given values
rho_0=1040;// The density of brine in kg/m^3
h_1=0.8;// m
H=4;// m
z_0=0;
z_1=4;// z_0 & z_1 are limits of integration

//Calculation
P_1=rho_0*g*h_1/1000;// Standard  pressure
   determination formula in kPa
P_2=integrate('rho_0*g*(sqrt(1+(tan(3.14*z/4/H)^2)))
   ','z',z_0,z_1);//integrant
P_2=P_2/1000;// kPa
P=P_1+P_2;
printf('The gage pressure at the bottom of gradient
   zone P=%0.1fkPa\n',P);
```

Scilab code Exa 3.8 Hydrostatic Force Acting on the Door of a Submerged Car

```
// Example 3_8
clc;clear;funcprot(0);
// Properties
rho=1000;// The density of lake water through out
g=9.81;//The acceleration due to gravity in m/s^2

// Given values
s=8;// m
b=1.2;//m
h_c=s+b/2; // m

// Calculation
P_ave=(rho*g*h_c)/1000;// kN/m^2
```

```
printf('The average pressure on the door,P_ave=%0.1f kN/m^2\n',P_ave);
A=1*1.2;// m^2
F_r=P_ave*A;// kN
printf('The resultant hydrostatic force on the door,F_r=%0.1f kN\n',F_r);
y_p=s+b/2+((b^2)/(12*(s+b/2)));// m
printf('The pressure center,y_p=%0.2f m\n',y_p);
// The answer vary due to round off error
```

Scilab code Exa 3.9 `A Gravity Controlled Cylindrical Gate`

```
//Example 3_9
clc;clear;funcprot(0);
// Properties
rho=1000; // The density of water in kg/m^3
g=9.81; // The acceleration due to gravity in m/s^2

// Given values
R=0.8;// Radius of solid cylinder in m
h_bottom=5;// m
A=0.8*1;// m^2
s=4.2;
h_c=s+R/2;// m

// Calculation
// (a)
F_x=(rho*g*h_c*A)/1000;//kN
printf('(a)Horizontal force on vertical surface F_x=%0.1f kN\n',F_x);
F_y=(rho*g*h_bottom*A)/1000;// kN
V=(R^2-(%pi*(R^2)/4))*1;// m^3
W=(rho*g*V)/1000;// kN
F_v=F_y-W;// kN
F_r=sqrt(F_x^2+F_v^2);// kN
```

```
theta=atand(F_v/F_x);// degree
printf('The hydrostatic force acting on the cylinder,F_r=%0.1f kN\n',F_r);
printf('The direction of the hydrostatic force acting on the cylindrical surface,theta=%0.1f degree\n',theta);
//(b)
W_cyl=F_r*sind(theta);// kN
printf('(b)The weight of the cylinder per m length,W_cyl=%0.1f kN\n',W_cyl);
```

Scilab code Exa 3.10 `Measuring Specific Gravity by a Hydrometer`

```
//Example 3_10
clc;clear;funcprot(0);
//Given values
h_sub=0.1;// m
rho_w=1000;// Density of water in kg/m^3
R=0.005;// m

// Calculation
V_sub=%pi*R^2*h_sub;// m^3
m=rho_w*V_sub;// kg
printf('The mass of lead,m=%0.5f kg\n',m);
```

Scilab code Exa 3.11 `Weight Loss of an Object in Seawater`

```
// Example 3_11
clc;clear;funcprot(0);
// Given values
rho_sw=1025;// The density of sea water in kg/m^3
rho_con=2300;// The density of concrete in kg/m^3
g=9.81;// The acceleration due to gravity in m/s^2
```

```

// Calculation (a)
V=0.4*0.4*3; // Volume of the block in m^3;
F_air=(rho_con*g*V)/1000;// kN
printf('Tension in rope must be equal to the weight of the block,F_air=%0.1f kN\n',F_air);
// Calculation (b)
F_b=(rho_sw*g*V)/1000;// kN
printf('Balance force F_b=%0.1f kN\n',F_b);
F_water=F_air-F_b;// kN
printf('Tension in rope when it is completely immersed in water,F_water=%0.1f kN\n',F_water);
```

Scilab code Exa 3.12 `Overflow from a Water Tank During Acceleration`

```
//Example 3_12
clc;clear;funcprot(0);
// Given values
dV=90-0;//Change in velocity in km/h
dt=10;// s
b_1=2;// m
b_2=0.6// m;
g=9.81;// m/s^2
a_z=0;// m/s^2

// Calculation
a_x=(dV/dt)/3.6;// The acceleration of the truck in m/s^2
theta=atand(a_x/(g+a_z));// degree
// Case 1:
gradZ_s1=(b_1/2)*tand(theta)*100;// cm
printf('Case 1 :The long side is parallel to the direction of motion:Vertical rise,grad_Zs1=%0.1f cm\n' ,gradZ_s1);
// Case 2:
```

```
gradZ_s2=(b_2/2)*tand(theta)*100;// cm
printf('Case 2 :The short side is parallel to the direction of motion:Vertical rise,grad_Zs2=%0.1f cm\n',gradZ_s2);
```

Scilab code Exa 3.13 `Rising of a Liquid During Rotation`

```
// Example 3_13
clc;clear;funcprot(0);
// Given values
h_0=0.5;// m
Z_R=0.6;// m
g=9.81;// m/s^2
R=0.1;// m
//Calculation
omega=sqrt((4*g*(Z_R-h_0))/R^2);// rad/s
printf('The maximum rotational speed of the container,omega=%0.1f rad/s \n',omega);
n=(omega/(2*%pi))*60;// rpm
printf('The rotational speed of the container expressed in terms of rpm,n=%0.0f rpm\n',n);
```

Chapter 4

FLUID KINEMATICS

Scilab code Exa 4.1 `A Steady Two Dimensional Velocity Field`

```
// Example 4_1
clc;clear;funcprot(0);

//Given values
// u=0.5+0.8x
// v=1.5-0.8y

//Calculation
//Since V is a vector, all its components must equal
    zero in order for V itself to be zero.
x=-0.5/0.8;
y=-1.5/-0.8;
disp(y,x,"Stagnation point x&y in m");
```

Scilab code Exa 4.2 `Acceleration of a Fluid Particle through a Nozzle`

```
//Example 4_2
clc;clear;funcprot(0);
```

```
// Given values
v=0.00187;// The volume flow rate in ft^3/s
D_inlet=0.0350;// The diameter at inlet in ft
D_outer=0.182/12;// The diameter at outlet in ft
dx=0.325;// Length of the nozzle in ft

// Calculation
A_inlet=(%pi*D_inlet^2)/4;// Area at inlet in ft^2
A_outer=(%pi*D_outer^2)/4;// Area at outer in ft^2
u_in=v/A_inlet;// ft/s
u_out=v/A_outer;;// ft/s
a_x=(u_out^2-u_in^2)/(2*dx);// Axial acceleration in ft/s^2
printf('Axial acceleration,a_x=%0.0f ft/s^2 \n',a_x);
// The answer is bit different due to round off error in book
```

Scilab code Exa 4.3 `Material Acceleration of a Steady Velocity Field`

```
//Example 4_3
clc;clear;funcprot(0);
//Given values
x=2;
y=3;
// Analysis
function[ax,ay]=vecac(x,y)
    [X,Y] = meshgrid(x,y);
    ax = 0.4+0.64*X;
    ay=-1.2+0.64*Y;
endfunction
x = linspace(-2,2,5);
y = linspace(0,5,6);
[ax,ay]=vecac(x,y);
[ax1,ay1]=vecac(2,3);
```

```
printf('\nThe material acceleration at point (x=2m,y=3m),a_x=%0.2f m/s^2 and a_y=%0.2f m/s^2',ax1,ay1);
champ(x',y',ax',ay');
xgrid(1);
xtitle('Scale:10 m/s^2');
xlabel('x');
ylabel('y');
```

Chapter 5

MASS BERNOULLI AND ENERGY EQUATIONS

Scilab code Exa 5.1 `Water Flow through a Garden Hose Nozzle`

```
//Example 5_1
clc;clear;funcprot(0);
// Given values
V=10; // Volume of water in gallon
dt=50;// Time in seconds
rho=1;//The density of water in kg/L
r_e=0.4;// Radius of  nozzle at exit in cm

// Calculation (a)
v=V/dt; // Volume flow rate in L/s
v=v*3.7854;// Convert gal into L
printf('(a)The volume flow rate of water,v=%0.3f L/s\n',v);
m=rho*v;// Mass flow rate of water in kg/s
printf('The mass flow rate of water,m=%0.3f kg/s\n',m);

// Calculation (b)
A_e=%pi*r_e^2;// The cross sectional area of nozzle
```

```
at exit in cm^2
V_e=v/A_e;
V_e=(V_e*10000/1000);// Convert to m/s
printf('(b)The average velocity of water at the nozzle exit,V_e=%0.1f m/s\n',V_e);
```

Scilab code Exa 5.2 Discharge of Water from a Tank

```
//Example 5_2
clc;clear;funcprot(0);
// Given values
h_0=4;
h_2=2;// Corresponding heights in ft
D_tank=3*12;
D_jet=0.5;// Corresponding diameters in inch
g=32.2;// The acceleration due to gravity in ft/s^2

// Calculation
t=((sqrt(h_0)-sqrt(h_2))/sqrt(g/2))*((D_tank/D_jet)^2);
t=t/60;// Convert seconds to minutes
printf('The time of discharge,t=%0.1f min\n',t);
```

Scilab code Exa 5.3 Performance of a Hydraulic Turbine Generator

```
//Example 5_3
clc;clear;funcprot(0);
// Given values
m=5000;// Mass flow rate of water in kg/s
W_eout=1862;//The electric power generated is measured in kW
rho=1000;// The density of water in kg/m^3
h=50;// The depth of the water in m
```

```
g=9.81;// m/s^2
e_min=g*h;// kJ/kg
e_mout=0;// kJ/kg
n_gen=0.95;// The generator efficiency

// Calculation
//(a)
dE_mech=(m*(e_min-e_mout))/1000;//kW
n_o=(W_eout/dE_mech);// The over all efficiency
printf('(a)The over all efficiency,n_o=%0.2f\n',n_o);
//(b)
n_t=n_o/n_gen;// )The mechanical efficiency of the turbine
printf('(b)The mechanical efficiency of the turbine,n_t=%0.2f\n',n_t);
//(c)
W_sout=n_t*dE_mech;// kW
printf('(c)The shaft power output,W_shaft,out=%0.0f kW\n',W_sout);
//The answer is a bit different due to rounding off error in textbook
```

Scilab code Exa 5.5 Spraying Water into the Air

```
// Example 5_5
clc;clear;
// given values
P_gage=400;// kPa
rho=1000;// the density of water in kg/m^3
g=9.81;// the accleration due to gravity in m/s^2

// Calculation
z_2=P_gage*1000/(rho*g);// m
printf('The water jet can rise as high,z_2=%0.1f m\n
```

```
   ',z_2);
```

Scilab code Exa 5.6 `Water Discharge from a Large Tank`

```
// Example 5_6
clc;clear;funcprot(0);
// Given values
z_1=5;// m
g=9.81;// The acceleration due to gravity in m/s^2

// Calculation
V_2=sqrt(2*g*z_1);// Toricelli equation
printf('The water leaves the tank with an initial
   velocity,V_2=%0.1f m/s\n',V_2);
```

Scilab code Exa 5.7 `Siphoning Out Gasoline from a Fuel Tank`

```
// Example 5_7
clc;clear;funcprot(0);
// Given values
P_atm=101.3;// The atmospheric pressure in kPa;
rho=750;//The density of gasoline in kg/m^3
g=9.81;//m/s^2
z_1=0.75;//  m
z_3=2.75;// m
D=(5/1000);// m

// Calculation
//(a)
V_2=sqrt(2*g*z_1);
A=(%pi*D^2)/4;//The cross-sectional area of the tube
    in m^2
v=V_2*A*1000;//The flow rate of gasoline in  L/s
```

```
V=4;// Volume of gasoline in litre
gradt=V/v;
printf('(a)The time needed to siphon 4 L of gasoline from the tank,gradt=%0.1f s\n',gradt);
//(b)
P_3=P_atm-((rho*g*z_3)/1000);// kPa
printf('(b)The pressure at point 3,P_3=%0.1f kPa\n',P_3);
```

Scilab code Exa 5.8 `Velocity Measurement by a Pitot Tube`

```
//Example 5_8
clc;clear;funcprot(0);
// Given values
h_1=0.03;// m
h_2=0.07;// m
h_3=0.12;// m
g=9.81;//m/s^2

//Calculation
V_1=sqrt(2*g*h_3);// m/s
printf('The velocity at the center of the pipe,V_1=%0.2f m/s\n',V_1);
```

Scilab code Exa 5.9 `The Rise of the Ocean Due to a Hurricane`

```
//Example 5_9
clc;clear;funcprot(0);
// Given values
rho_hg=848;//The density of mercury in lbm/ft^3
rho_sw=64;//The density of seawater in lbm/ft^3
rho_atm=0.076;//The density of atmosphereic air in lbf/ft^3
```

```
H_hg=(30-22);// inch
V_a=155;//mph
V_a=155*1.4667;// convert mph into ft/s
P_air=22;// The hurricane atmospheric pressure at the eye of the storm is in Hg
P_atm=30;// in hg
g=32.2;// ft/s^2

// Calculation
//(a)
h_1=((rho_hg/rho_sw)*H_hg)/12;
printf('(a)The pressure difference between points 1 and 3 in terms of the seawater column height,h_1=%0.2f ft\n',h_1);
//(b)
H_air=((V_a^2)/(2*g));//ft
rho_air=(P_air/P_atm)*rho_atm;//the density of air in the hurricane in lbm/ft^3
h_dynamic=(rho_air/rho_sw)*H_air;//ft
h_2=h_1+h_dynamic;//ft
printf('(b)The total storm surge at point 2,h_2=%0.2f ft\n',h_2);
```

Scilab code Exa 5.12 Pumping Power and Frictional Heating in a Pump

```
//Example 5_12
clc;clear;funcprot(0);
// Given values
rho=1;//The density of water to be 1 kg/L = 1000 kg/m^3
v=50;// The water flow rate through the pump in L/s
n_m=.90;//The efficiency of electric motor
W_e=15;//Power in kW;
P_1=100;// The pressure at the inlet of the pump in kPa
```

```
P_2=300;// The pressure at the outlet of the pump in kPa
rho_1=1000;//The density of water in kg/m^3
c=4.18;// The specific heat in kJ/kg  C .

// Calculation
// (a)
m=rho*v;//The mass flow rate of water through the pump in kg/s
W_p=n_m*W_e;//The mechanical (shaft) power delivers to the pump kW
dE_m=m*((P_2-P_1)/rho_1);//The increase in the mechanical energy of the fluid in kW
n_p=dE_m/W_p;// The mechanical efficiency of the pump
printf('(a)The mechanical efficiency of the pump, n_pump=%0.3f (or)%0.1f percentage \n',n_p,n_p*100);
//(b)
E_mloss=W_p-dE_m;//     lost     mechanical energy in kW
dT=E_mloss/(m*c);//  C
printf('(b)The temperature rise of water due to the mechanical inefficiency ,dT=%0.3f degree Celsius\n',dT);
```

Scilab code Exa 5.13 `Hydroelectric Power Generation from a Dam`

```
//Example 5_13
clc;clear;
// Given values
rho=1000;// the density of water in kg/m^3
v=100;//Flow rate of water in kg/m^3
z_1=120;// m
h_l=35;// m
```

```
n_t=0.8;
g=9.81;// The acceleration due to gravity in m/s^2

// Calculation
m=rho*v;//The mass flow rate of water in kg/s
h_t=z_1-h_l;// m
W_t=(m*g*h_t)/1000;// kW
W_e=(n_t*(W_t/1000));// MW
printf('The electric power generated by the actual unit=%0.1f MW\n',W_e);
```

Scilab code Exa 5.14 `Fan Selection for Air Cooling of a Computer`

```
//Example 5_14
clc;clear;funcprot(0);
// Given values
rho=1.20;//The density of air in kg/m^3
alpha_2=1.10;
dt=1;// s
D=0.05;// Diameter in m
n_f=0.3;// Efficiency of fan motor

// Calculation
//(a)
V=0.5*(12*40*40);//The air volume in the computer case in cm^3
V=V/10^6;// cm^3 to m^3
v=V/dt;//The volume  flow rate of air through the case in m^3/s
m=rho*V;//The mass flow rate of air through the case in kg/s
A=(%pi*D^2)/4;// m^2
V_1=v/A;//m/s
W_fan=m*alpha_2*(V_1^2/2);
W_e=W_fan/n_f;//Electric power input to the fan in W
```

```
printf('(a)Electric power input to the fan,W_elect=%0.3f W\n',W_e);
//(b)
dP=(rho*W_fan)/m;//dp=P_4-P_3
printf('(b)The pressure rise across the fan is %0.1f Pa.\n',dP);
//The answer is bit different due to round off error in the book
```

Scilab code Exa 5.15 `Head and Power Loss During Water Pumping`

```
//Example 5_15
clc;clear;funcprot(0);
// Properties
rho=1000;//The density of water in kg/m^3
// Given values
v=0.03;//The flow rate of water in m^3/s
W_p=20;// kW
g=9.81;//The acceleration due to gravity in m/s^2
z_2=45;// m

// Calculation
m=rho*v;//The mass flow rate of water through the system in kg/s
E_ml=(W_p-(m*g*z_2)/1000);
printf('The lost mechanical power,E_mechloss=%0.2f kW\n',E_ml);
h_l=E_ml*1000/(m*g);
printf('The irreversible head loss,h_L=%0.1f m\n',h_l);
```

Chapter 6

MOMENTUM ANALYSIS OF FLOW SYSTEMS

Scilab code Exa 6.1 `Momentum Flux Correction Factor for Laminar Pipe Flow`

```
//Example 6_1
clc;clear;
// Given values
y_0=0;// Lower limit of the integral
y_1=1;// Upper limit of the integral

// Analysis
b=-4*integrate('(y^2)','y',y_1,y_0);
printf("The momentum-flux correction factor for
   fully developed laminar flow becomes %0.2f \n",b)
   ;
```

Scilab code Exa 6.2 `The Force to Hold a Deflector Elbow in Place`

```
//Example 6_2
clc;clear;
```

```
// Given values
m=14;//Water flow rate in kg/s
rho=1000;//The density of water in kg/m^3
A_1=0.0113;// The cross sectional area of the elbow at inlet in m^2
A_2=7*10^-4;// The cross sectional area of the elbow at outlet in m^2
z_2=0.3;// m
z_1=0;// m
g=9.81;// The acceleration due to gravity in m/s^2
theta=30;// degree
b=1.03;// The momentum-flux correction factor

// Calculation
//(a)
v_1=m/(rho*A_1);
v_2=m/(rho*A_2);//The inlet and the outlet velocities in m/s
P_1g=(rho*g*(((v_2^2-v_1^2)/(2*g))+(z_2-z_1)))/1000;// kPa
printf("The gage pressure at the center of the inlet of the elbow=%0.1f kPa\n",P_1g);
//(b)   z
F_Rx=b*m*(((v_2*cosd(theta))-v_1))-(P_1g*1000*A_1);// N
F_Rz=b*m*v_2*sind(theta);// N
printf("The anchoring force of the elbow be F_Rx=%0.0f N,F_Rz=%0.0f N\n",F_Rx,F_Rz);
// The answer vary due to round off error
```

Scilab code Exa 6.3 The Force to Hold a Reversing Elbow in Place

```
//Example 6_3
clc;clear;
// Given values
```

```
//From example 6_2
b=1.03;// The momentum-flux correction factor
m=14;// Water flow rate in kg/s
v_1=1.24;// The inlet velocity in m/s
v_2=20;// The outlet velocity in m/s
P_1g=202200;// Gage pressure in N/m^2
A_1=0.0113;// m^2

//Calculation
F_Rx=(-b*m*(v_2+v_1))-(P_1g*A_1);// N
printf("The anchoring force needed to hold the elbow
    in place=%0.0f N\n",F_Rx);
// The answer vary due to round off error
```

Scilab code Exa 6.4 `Water Jet Striking a Stationary Plate`

```
//Example 6_4
clc;clear;
// Given values
b=1;// The momentum-flux correction factor
m=10;//Mass flow rate at kg/s
V_1=20;// m/s

// Calculation
F_r=b*m*V_1;
printf("The force needed to prevent the plate from
   moving horizontally due to the water stream=%0.0f
    N\n",F_r);
```

Scilab code Exa 6.5 `Power Generation and Wind Loading of a Wind Turbine`

```
//Example 6_5
clc;clear;
```

```
// given values
rho=0.076;//The density of air in lbm/ft^3
V_1=7*1.4667;// Wind speed in ft/s
D=30;//Diameter in ft
W_act=0.4;//kW

//Calculation
//(a)
A_1=(%pi*D^2)/4;
m=rho*V_1*A_1;
Ke_1=((V_1^2)/(2*32.2*737.56));
W_max=m*Ke_1;
n_wt=W_act/W_max;
printf("The efficiency of the wind turbine generator unit=%0.3f or(%0.1f percentage)\n",n_wt,n_wt*100);
//(b)
V_2=V_1*sqrt(1-n_wt);//The exit velocity in m/s
F_r=(m*(V_2-V_1))/32.2;
printf("The horizontal force exerted by the wind on the supporting mast of the wind turbine=%0.1f lbf\n",F_r);
// The answer vary due to round off error
```

Scilab code Exa 6.6 Repositioning of a Satellite

```
//Example 6_6
clc;clear;
// Given values
m_f=100;// kg
V_f=3000;//Velocity of solid fuel in m/s
dt=2;// seconds
m_sat=5000;// kg

// Calculation
```

```
//(a)
a_sat=((m_f/dt)*V_f)/m_sat;
printf(”The acceleration of the satellite during the first 2 s=%0.0f m/s^2\n”,a_sat);
//(b)
dV_sat=a_sat*dt;
printf(”The change of velocity of the satellite=%0.0f m/s\n”,dV_sat);
//(c)
F_sat=(0-(m_f/dt)*(-V_f))/1000;
printf(”The thrust exerted on the satellite=%0.0f kN\n”,F_sat);
```

Scilab code Exa 6.7 `Net Force on a Flange`

```
//Example 6_7
clc;clear;
// Given values
v=18.5;// Flow rate of water in gal/min
D=0.0650;// The inner diameter of the pipe in ft
rho=62.3;// The density of water at room temperature in lbm/ft^3
P_1g=13// lbf/in^2
W_f=12.8;// The total weight of the faucet assembly plus the water in lbf

// Calculation
A_c=(%pi*D^2/4);// ft^2
V=(v*0.1337)/(A_c*60);// ft/s
//V=V_1=V_2
m=(rho*v)*(0.1337/60);// lbm/s
A_1=((%pi*(0.780)^2)/4);// ft^2
F_rx=((-m*V)/32.2)-(P_1g*A_1);// lbf
F_rz=((-m*V)/32.2)+W_f;// lbf
F_r=[F_rx F_rz]// lbf
```

```
F_f=-[F_r];// lbf
printf('The net force on the flange,F_faucet on
   flange=%0.2fi+(%0.1f)k \n',F_f(1),F_f(2));
```

Scilab code Exa 6.8 `Bending Moment Acting at the Base of a Water Pipe`

```
//Example 6_8
clc;clear;
// Given values
rho=1000;//The density of water in kg/m^3
D=0.10;// Diameter in m
V=3;// Average velocity in m/s
g=9.81;// The acceleration due to gravity m/s^2
m=12;//Mass per meter length in kg/m
r_1=0.5;
r_2=2;// The average moment arm at inlet & outlet in
    m

// Calculation
A_c=((%pi*D^2)/4);// m^2
m_1=rho*A_c*V;// The mass flow rate in kg/s
W=m*g;//The weight of the horizontal section of the
   pipe in N
M_a=(r_1*W)-(r_2*m_1*V);// N.m
printf("The bending moment acting at the base of the
    pipe (point A)=%0.1f N.m\n",M_a);
```

Scilab code Exa 6.9 `Power Generation from a Sprinkler System`

```
//Example 6_9
clc;clear;
// Given values
rho=1;//The density of water in kg/L
```

```
n=300;//rpm
D=0.01;// Diameter of each jet in m
V_t=20;// L/s
V_n=V_t/4;// L/s
r=0.6;//m

// Calculation
A_j=(%pi*D^2)/4;//Area of jet in m^2
V_j=(V_n)/(A_j*1000);//Average jet exit velocity in m/s
w=(2*(%pi)*n)/60;// The angular momentum of the nozzle in rad/s
v_n=r*w;//The tangential velocities of the nozzle in m/s
v_r=V_j-v_n;//The average velocity of the water jet relative to the control volume in m/s
m_t=rho*V_t;// Mass flow rate in kg/s
T_shaft=r*m_t*v_r;// The torque transmitted through the shaft in Nm
W=(w*T_shaft)/1000;
printf(”The electric power generated=%0.1f kW\n”,W);
```

Chapter 7

DIMENSIONAL ANALYSIS AND MODELING

Scilab code Exa 7.4 `Extrapolation of Nondimensionalized Data`

```
//Example 7_4
clc;clear;
// Given values
g_moon=9.81/6;//The gravitational constant on the
   moon in m/s^2
t_1=2.75;
V=21;// Initial speed of the ball in m/s
theta=5;// degree
z_0=2.0;// m

//Calculation
//(a)
w_0=V*sind(theta);
Fr=w_0^2/(g_moon*z_0);
Fr=(Fr)^2;
t=(t_1*z_0)/w_0
printf("Estimated time to strike the ground=%0.2f s\
   n",t);
//(b)
```

```
t_2=(w_0+sqrt(w_0^2+(2*z_0*g_moon)))/g_moon;
printf("Exact time to strike the ground=%0.2f s\n",
   t_2);
// The answers vary due to round off error
```

Scilab code Exa 7.5 `Similarity between Model and Prototype Cars`

```
//Example 7_5
clc;clear;
// Given values
// Properties
//For air at atmospheric pressure and at T = 25 C
T=25;//  C
rho_p=1.184;//kg/m^3
mu_p=1.849*10^-5;//kg/m.s
//Similarly ,at T=5 C
T=5;//  C
rho_m=1.269;//kg/m^3
mu_m=1.754*10^-5;// kg/m.s
V_p=50;//Speed in mi/h
// (L_p/L_m)=5 The ratio of Lp to Lm is known
   because the prototype is five times larger than
   the scale model

// Calculation
V_m=(V_p*(mu_m/mu_p)*(rho_p/rho_m)*(5));
printf("The unknown wind tunnel speed for the model
   tests=%0.0f mi/h\n",V_m);
```

Scilab code Exa 7.6 `Prediction of Aerodynamic Drag Force on the Prototype Car`

```
//Example 7_6
clc;clear;
```

```
// Given values
// Properties
//For air at atmospheric pressure and at T = 25 C
T=25;//degree celsius
rho_p=1.184;//kg/m^3
mu_p=1.849*10^-5;//kg/m.s
V_p=50;//Speed in mi/h
//Similarly,at T=5 C
T=5;//degree celsius
rho_m=1.269;//kg/m^3
mu_m=1.754*10^-5;// kg/m.s
V_m=221;//mi/h
// (L_p/L_m)=5 The ratio of Lp to Lm is known
   because the prototype is five times larger than
   the scale model
F_dm=21.2;//The average drag force on the model in
   lbf

// Calculation
F_dp=F_dm*(rho_p/rho_m)*(V_p/V_m)^2*(5)^2;
printf("The aerodynamic drag force on the prototype=
   %0.1f lbf",F_dp);
```

Scilab code Exa 7.10 `Model Truck Wind Tunnel Measurements`

```
//Example 7_10
clc;clear;
// Given values
L_m=0.991;// Length of the model truck in m
h_m=0.257;// Height of the model truck in m
w_m=0.159;// Width of the model truck in m
V_p=26.8;// Velocity of the prototype in m/s
T=25;//  C
C=16;// Geometric ratio

```

```
// Properties
//For air at atmospheric pressure and at T=25 C ,
rho_m=1.184;// Density of air in kg/m^3
mu_m=1.849*10^-5;// Viscosity of air in kg/m.s

// Calculation
// From table 7.7,
V_m=[20 25 30 35 40 45 50 55 60 65 70];// Velocity of the model truck in m/s
F_D=[12.4 19.0 22.1 29.0 34.3 39.9 47.2 55.5 66.0 77.6 89.9];// Drag force of the model truck in N
for(i=1:11)
    A_m=w_m*h_m;// Area of the model truck in m^2
    C_Dm(i)=(F_D(i))/((1/2)*rho_m*(V_m(i))^2*A_m);// Drag coefficient
    Re_m(i)=(rho_m*V_m(i)*w_m)/(mu_m);// Reynolds number of the model truck
end
xlabel('Re*10^-5');
ylabel('C_D');
xtitle('FIGURE 7-41');
plot((Re_m/10^5),C_Dm,'o');
rho_p=rho_m;// Density of air in kg/m^3
w_p=w_m;// Width of the prototype in m
mu_p=mu_m;// Viscosity of air in kg/m.s
Re_p=(rho_p*V_p*w_p)/(mu_p);// Reynolds number of the prototype
A_p=A_m;// // Area of the prototype in m^2
C_Dp=C_Dm(10);// Drag coefficient
F_Dp=(1/2)*rho_p*V_p^2*C^2*A_p*C_Dp;// Aerodynamic drag on the prototype in N
printf("The aerodynamic drag on the vehicle=%0.0f N\n",F_Dp);
// The answer provided in the textbook is wrong
```

Scilab code Exa 7.11 `Model Lock and River`

```
//Example 7_11
clc;clear;
// Given values
L_r=1/100;// (L_r=L_m/L_p) Length scale factor

//Properties
// For water at atmospheric pressure and at T = 20 C
nu_p=1.002*10^-6;// The prototype kinematic viscosity in m^2/s

// Calculation
nu_m=nu_p*(L_r)^(3/2);
printf("Required kinematic viscosity of model liquid :nu_m=%0.2e m^2/s\n",nu_m);
```

Chapter 8

FLOW IN PIPES

Scilab code Exa 8.1 `Flow Rates in Horizontal and Inclined Pipes`

```
//Example 8_1
clc;clear;funcprot(0);
// Given values
P_i=745;
P_o=97;//The pressure at the pipe inlet and outlet
   in kPa
D=0.05;// m
L=40;// m
//Properties
rho=888;//kg/m^3
mu=0.800;// kg/m.s
g=9.81;// m/s^2

//Calculation
gradP=P_i-P_o;//kPa
A_c=(%pi*D^2)/4;// m^2
//(a)
//For the horizontal case, theta=0
theta=0;// degree
V=((((gradP*1000)-(rho*g*L*sind(theta)))*(%pi*D^4))
   /(128*mu*L));// m^3/s
```

```
V_horiz=V;// m^3/s
printf('(a)The flow rate for the horizontal case, theta=0,V_horiz=%0.5f m^3/s\n',V_horiz);

//(b)
// For uphill flow with an inclination of 15 , we have theta=+15 ,
theta_1=+15;// degree
V=((((gradP*1000)-(rho*g*L*sind(theta_1)))*(%pi*D^4))/(128*mu*L));
V_uphill=V;//m^3/s
printf('(b)The flow rate for uphill flow with an inclination of 15 ,V_uphill=%0.5f m^3/s\n',V_uphill);

//(c)
//For downhill flow with an inclination of 15 ,we have theta=-15 ,
theta_2=-15;//degree
V=((((gradP*1000)-(rho*g*L*sind(theta_2)))*(%pi*D^4))/(128*mu*L));
V_downhill=V;//m^3/s
printf('(c)The flow rate for downhill flow with an inclination of 15 ,V_downhill=%0.5f m^3/s\n',V_downhill);
V_avg=(V_downhill/A_c);
Re=(rho*V_avg*D)/mu;
disp("Re=100.Re<2300.Therefore, the flow is laminar for all three cases and the analysis is valid.");
```

Scilab code Exa 8.2 Pressure Drop and Head Loss in a Pipe

```
//Example 8_2
clc;clear;funcprot(0);
// Given values
```

```
rho=62.42;//lbm/ft^3
mu=1.038*10^-3;// lbm/ft.s
D=0.01;// ft
L=30;// ft
V_avg=3;// ft/s
g=32.2;// Ft/s^2

//Calculation
//(a)
Re=(rho*V_avg*D)/mu;// Reynolds number
f=64/Re;// Friction factor
h_l=f*(L/D)*((V_avg^2)/(2*g));// ft
printf('(a)The head loss, h_l =%0.1f ft\n',h_l);
//(b)
gradP_l=(f*(L/D)*rho*(V_avg^2/2))/32.2;// lbf/ft^2
gradP_1=(gradP_l/144);// psi
printf('(b)The pressure drop,gradP_l=%0.0f lbf/ft^2=%0.2f psi\n',gradP_1*144,gradP_1);
//(c)
A_c=(%pi*D^2)/4;// ft^2
v=V_avg*A_c;// ft^3/s
W_pump=v*gradP_l/0.737;// W
printf('(c)The pumping power requirements,W_pump=%0.2f W\n',W_pump);
// The answer vary due to round off error
```

Scilab code Exa 8.3 Determining the Head Loss in a Water Pipe

```
//Example 8_3
clc;clear;funcprot(0);
// Given values
rho=62.36;// lbm/ft^3
mu=7.536*10^-4;// lbm/ft.s
D=2/12;// ft
v=0.2;// ft^3/s
```

```
L=200;// ft
g=32.2;// ft/s^2

//Calculation
A_c=(%pi*D^2)/4;//  ft^2
V=v/A_c;// Average velocity in ft/s
Re=(rho*V*D)/(mu);// Reynolds number
// Re is greater than 4000. Therefore, the flow is turbulent. The relative roughness of the pipe is calculated using Table 8 2 , (epsilon/D)=e
E=0.000007;
e=E/(D);
//To avoid any reading error, we determine f from the Colebrook equation:(1/sqrt)=-2.0*log10*((e/3.7)+(2.51/(Re*sqrt(f)))
// f=y(1)
function[X]=frictionfactor (y)
    X(1)=(-2.0*log10((0.000042/3.7)+(2.51/(126400*sqrt(y(1))))))-(1/sqrt(y(1)));
endfunction
y=[0.001];
z=fsolve(y,frictionfactor);// Friction factor
gradP_L1=(z*(L/D)*(rho*(V^2)/2))*(1/32.2);// lbm/ft^2
gradP_L=gradP_L1/144;// psi
printf('The pressure drop,gradP_L=%0.0f lbf/ft^2=%0.1f psi \n',gradP_L*144,gradP_L);
h_L=(z*(L/D)*(V^2/(2*g)));// ft
printf('The head loss,h_L=%0.1f ft\n',h_L);
W_p=(v*gradP_L1)/0.737;// W
printf('The required power input,W_pump=%0.0f W \n',W_p);
// The answer vary due to round off error
```

Scilab code Exa 8.4 Determining the Diameter of an Air Duct

```
//Example 8_4
clc;clear;funcprot(0);
// Given values
P=1;// atm
T=35;// degree celsius
L=150;// m
h_L=20;// m
v=0.35;// m^3/s
g=9.81;// m/s^2
//Properties
rho=1.145;// kg/m^3
mu=1.895*10^-5;// kg/m.s
nu=1.655*10^-5;// m^2/s

//Calculation
// V=y(1); Re=y(2); f=y(3);D=y(4)
function[X] = Diameter(y)
    X(1)=(v/(%pi*(y(4)^2)/4))-y(1);
    X(2)=((y(1)*y(4))/(nu))-y(2);
    X(3)=(-2.0*log10(2.51/(y(2)*sqrt(y(3)))))-(1/sqrt(y(3)));
    X(4)=(y(3)*(L/(y(4))*((y(1)^2)/(2*g))))-h_L;
endfunction
y=[1 100000 0.01 0.1];
z=fsolve(y,Diameter);
V=z(1);// m/s
Re=z(2);// Reynolds number
f=z(3);
D=z(4);// m
printf('The minimum diameter of the duct,D=%0.3f m\n',D);
//The diameter can also be determined directly from the third SwameeJain formula to be
y=0;
D=0.66*(((y^1.25*((L*v^2)/(g*h_L))^4.75))+(nu*v^9.4*(L/(g*h_L))^5.2))^0.04;
printf('The diameter can also be determined directly from the third SwameeJain formula to be D=%0
```

```
.3f m\n',D);
```

Scilab code Exa 8.5 `Determining the Flow Rate of Air in a Duct`

```
//Example 8_5
clc;clear;funcprot(0);
//From Example 8_4
// Given values
P=1;// atm
T=35;// degree celsius
L=300;// m
D=0.267;// m
h_L=20;// m
v_old=0.35;// m^3/s
g=9.81;// m/s^2
//Properties
rho=1.145;// kg/m^3
mu=1.895*10^-5;// kg/m.s
nu=1.655*10^-5;// m^2/s

//Calculation
//V=y(1); Re=y(2); f=y(3);v=y(4)
function[X]=flowrate(y);
    X(1)=real((y(4)/(%pi*D^2/4))-y(1));
    X(2)=real(((y(1)*D)/(nu))-y(2));
    X(3)=real((-2.0*log10(2.51/(y(2)*sqrt(y(3)))))-(1/sqrt(y(3))));
    X(4)=real(((y(3)*L*y(1)^2)/(D*2*9.81))-20);
endfunction
y=[1 10000 0.01 0.1];
z= fsolve(y,flowrate);
v_new=z(4);// m^3/s
v_drop=v_old-v_new;//The drop in the flow rate
printf('The drop in the flow rate through the duct.v_drop=%0.2f m^3/s\n',v_drop);
```

Scilab code Exa 8.6 Head Loss and Pressure Rise during Gradual Expansion

```
//Example 8_6
clc;clear;funcprot(0);
//Given values
V_1=7;// m/s
P_1=150// kPa
D_1=0.06;// m
D_2=0.09;// m
// Assumptions
//alpha_1=alpha_2=1.06
alpha_1=1.06;
alpha_2=1.06;
g=9.81;// m/s^2
//Properties
rho=1000;//The density of water in kg/m^3
K_L=0.07;// The loss coefficient for gradual
    expansion
theta=60;// Total included angle in degree

//Calculation
V_2=(D_1^2/D_2^2)*V_1;// The downstream velocity of
    water in m/s
h_L=K_L*(V_1^2/(2*g));// m
printf('The irreversible head loss in the expansion
    section,h_L=%0.3f m\n',h_L);
P_2=P_1+(rho*(((alpha_1*V_1^2)-(alpha_2*V_2^2))/2-(g
    *h_L)))/1000;// kPa
printf('The pressure in the larger-diameter pipe,P_2
    =%0.0f kPa\n',P_2);
```

Scilab code Exa 8.7 Pumping Water through Two Parallel Pipes

```
//Example 8_7
clc;clear;funcprot(0);
//Given values
Z_a=5;// m
Z_b=13;// m
D_1=0.04;
D_2=0.08;// The diameters of the two pipes m
L_1=36;// m
L_2=36;// m
W_elect=8000;// W
n_pump=0.70;
g=9.81;// m/s^2
//Properties
rho=998;// kg/m^3
mu=1.002*10^-3;// kg/m.s
eps=0.000045;// m

//Calculation
// V1=y(1); V2=y(2); Re1=y(3); Re2=y(4); f1=y(5);f2=y(6);h_L1=y(7);h_L2=y(8); h_pump=y(9);v1=y(10);v2=y(11);v=y(12);h_L=y(13)
function[X]=flowrate(y);
    X(1)=real(((rho*y(12)*g*y(9))/n_pump)-W_elect);
    X(2)=real(((y(10)*4)/(%pi*D_1^2))-y(1));
    X(3)=real(((y(11)*4)/(%pi*D_2^2))-y(2));
    X(4)=real(((rho*y(1)*D_1)/(mu))-y(3));
    X(5)=real(((rho*y(2)*D_2)/(mu))-y(4));
    X(6)=real((-2.0*log10(((eps)/(3.7*D_1))+(2.51/(y(3)*sqrt(y(5))))))-(1/sqrt(y(5))));
    X(7)=real((-2.0*log10(((eps)/(3.7*D_2))+(2.51/(y(4)*sqrt(y(6))))))-(1/sqrt(y(6))));
    X(8)=real(((y(5)*L_1*(y(1)^2))/(D_1*g*2))-(y(7)));
    X(9)=real(((y(6)*L_2*(y(2)^2))/(D_2*g*2))-(y(8)));
    X(10)=real((y(10)+y(11))-y(12));
    X(11)=real(((Z_b-Z_a)+y(13))-y(9));
    X(12)=real(y(7)-y(13));
```

```
    X(13)=real(y(8)-y(13));
endfunction
y=[1 1 100000 100000 0.01 0.01 10 10 10 0.01 0.001 0.01 10];
fr=fsolve(y,flowrate);
printf('The total flow rate between the reservoirs , v=%0.4f m^3/s\n',fr(12));
printf('The flow rate through pipe 1,v_1=%0.5f m^3/s\n',fr(10));
printf('The flow rate through pipe 2,v_2=%0.4f m^3/s\n',fr(11));
// The answer vary due to round off error
```

Scilab code Exa 8.8 `Gravity Driven Water Flow in a Pipe`

```
//Example 8_8
clc;clear;funcprot(0);
D=0.05;//m
v=0.006;// m^3/s
K_Lentrance=0.5;
K_Lelbow=0.3;
K_Lvalve=0.2;
K_Lexit=1.06;// The loss coefficients
L=89;// m
z_2=4;// m
//Properties
rho=999.7;// kg/m^3
mu=1.307*10^-3;// kg/m.s
epsilon=0.00026; // m
g=9.81;// m/s^2

//Calculation
A_c=(%pi*D^2)/4;//m^2
V=v/A_c;// m/s
Re=(rho*V*D)/mu;
```

```
e=epsilon/D;
// f=y(1)
function[X]=frictionfactor(y);
    X(1)=(-2.0*log10((e/3.7)+(2.51/(Re*sqrt(y(1))))))-(1/sqrt(y(1)));
endfunction
y=[0.01];
z=fsolve(y,frictionfactor);
f=z;//friction factor
SigmaK_L=K_Lentrance+(2*K_Lelbow)+K_Lvalve+K_Lexit;
h_l=((f*(L/D))+(SigmaK_L))*(V^2/(2*g));// The total head loss in m
z_1=z_2+h_l;// m
printf('The elevation of the source,z_1=%0.1f m\n',z_1);
// The answer vary due to round off error
```

Scilab code Exa 8.9 Effect of Flushing on Flow Rate from a Shower

```
//Example 8_9
clc;clear;funcprot(0);
//Given values
P_g=2*10^5;// N/m^2
D=0.015;// m

//Properties
rho=998;// kg/m^3
mu=1.002*10^-3;// kg/m.s
nu=1.004*10^-6;// m^2/s
epsilon=1.5*10^-6;//The roughness of copper pipes in m
g=9.81;// m/s^2

//Calculation
//(a)
```

```
SigmaK_l=0.9+(2*0.9)+10+12;
h_l=(P_g/(rho*g))-2;// m
// h_l=((f*L/D)+SigmaK_l)*(V^2/(2*g))
// V=(v/A_c)
// Re=(V*D)/nu
// (1/sqrt(f))=-2.0*log*(((e/D)/3.7)+(2.51/(Re*sqrt(f))))
// f = y(1) ; V = y(2); Vdot = y(3); Re= y(4);
function[X] = flowrate(y)

    X(1)=(((y(1)*(11/D))+SigmaK_l)*y(2)^2/(2*g))-h_l;
    X(2)=(y(3)*4)/(%pi*D^2)-y(2);
    X(3)=((y(2)*D)/(nu))-y(4);
    X(4)=-2.0*log10(((epsilon)/(3.7*D))+((2.51)/(y(4)*sqrt(y(1)))))-1/sqrt(y(1));
endfunction
y=[0.001 1 0.0001 10000]; //Initial conditions for all four variables
z = fsolve(y,flowrate); // Solver Initilisation
y(3)=z(3)*1000// The flow rate of water through the shower head in L/s
printf('(a)The flow rate of water through the shower head,v=%0.02f L/s\n',y(3));

//(b)
h_l3=P_g/(rho*g);
SigmaK_l3=2+10+0.9+14;
// f1=y(1) ; V1=y(2); V2=y(3); V3=y(4) ;Vdot1=y(5);Vdot2=y(6);Vdot3=y(7);Re1=y(8);Re2=y(9);Re3=y(10); f2=y(11); f3=y(12)
function[X]=flowrate(y)
    X(1)=real(((y(1)*(5*y(2)^2)/(D*2*g))+(((y(11)*6/D)+24.7)*(y(3)^2)/(2*g))-h_l);
    X(2)=real(((y(1)*(5*y(2)^2)/(D*2*g))+((((y(12)*1/D))+SigmaK_l)*(y(4)^2/(2*g)))-h_l3);
    X(3)=real(((y(5)*4)/(%pi*D^2))-y(2));
    X(4)=real(((y(6)*4)/(%pi*D^2))-y(3));
```

```
    X(5)=real(((y(7)*4)/(%pi*D^2))-y(4));
    X(6)=real((y(2)*(D)/(nu))-y(8));
    X(7)=real((y(3)*(D)/(nu))-y(9));
    X(8)=real((y(4)*(D)/(nu))-y(10));
    X(9)=real((-2.0*log10(((epsilon)/(3.7*D))+((2.51)/(y(8)*sqrt(y(1))))))-1/sqrt(y(1)));
    X(10)=real((-2.0*log10(((epsilon)/(3.7*D))+((2.51)/(y(9)*sqrt(y(11))))))-1/sqrt(y(11)));
    X(11)=real((-2.0*log10(((epsilon)/(3.7*D))+((2.51)/(y(10)*sqrt(y(12))))))-1/sqrt(y(12)));
    X(12)=real(y(6)+y(7)-y(5));
endfunction
y=[0.001 1 1 1 0.0001 0.0001 0.0001 10000 10000 10000 0.001 0.001];
z=fsolve(y,flowrate);
printf('(b)reduces the flow rate of cold water through the shower by 21 percent\n')
```

Scilab code Exa 8.10 Measuring Flow Rate with an Orifice Meter

```
//Example 8_10
clc;clear;funcprot(0);
// Given values
rho_m=788.4;// kg/m^3
mu=5.857*10^-4;// The dynamic viscosity of methanol in kg/m.s
d=0.03;// Diameter of orifice in m
D=0.04;// Diameter of pipe in m
rho_Hg=13600;// kg/m^3
g=9.81;// m/s^2
h=0.11;// m
//Assumptions
C_d=0.61;
```

```

//Calculation
beta=(d/D);// The diameter ratio
A_0=(%pi*d^2)/4;// The throat area of the orifice in m^2
gradP=(rho_Hg-rho_m)*g*h;
v=A_0*C_d*sqrt((2*((rho_Hg/rho_m)-1)*g*h)/(1-beta^4));// m^3/s
printf('The flow rate of methanol through the pipe,v=%0.2e  m^3/s\n',v);
A_1=(%pi*D^2)/4;// m^2
V=v/A_1;// m/s
printf('The average flow velocity,V_1=%0.2f m/s\n',V)
```

Chapter 9

DIFFERENTIAL ANALYSIS OF FLUID FLOW

Scilab code Exa 9.11 `Volume Flow Rate Deduced from Streamlines`

```
//Example 9_11
clc;clear;funcprot(0);
// Given values
V=1.0;//  Uniform velocity in m/s
w=2.0;// Width in m
psi_w=0; // m^2/s
psi_d=1.0;// m^2/s

// Calculation
V_1=psi_d-psi_w; // The volume flow rate per unit
    width (V/w)in m^3/s
v=V_1*w;// m^3/s
printf('The total volume flow rate through the slot ,
    v =%0.1f m^3/s\n',v);
delta=0.21;// m
psi_a=1.6;// m^2/s
psi_b=1.8;// m^2/s
V_2=psi_b-psi_a;
V_a=(1/delta)*V_2;
```

```
printf('The speed at point A,V_a =%0.2f m/s\n',V_a);
```

Chapter 10

APPROXIMATE SOLUTIONS OF THE NAVIER STOKES EQUATION

Scilab code Exa 10.2 `Terminal Velocity of a Particle from a Volcano`

```
// Example 10_2
clc;clear;funcprot(0);
//Given data
D=50*10^-6;// Diameter of spherical ash particle in m
T=-50;//  C
P=55;// kPa
rho_p=1240;// The density of the particle in kg/m^3
//Properties
mu=1.474*10^-5;// kg/m.s
rho_air=0.8588;// kg/m^3
g=9.81;// The acceleration due to gravity  in m/s^2

//Calculation
V=(D^2/(18*mu))*(rho_p-rho_air)*g;//The terminal velocity of this particle in m/s
printf('\nThe terminal velocity of this particle,V=
```

```
%0.3f m/s',V);
Re=(rho_air*V*D)/mu;
```

Scilab code Exa 10.6 Velocity in a Flow Composed of Three Components

```
// Example 10_6
clc;clear;
//Given data
// Assume (vdot/L)_1=V1,(vdot/L)_2=V2;
V1=2.00;// m^2/s
V2=-1.00;// m^2/s
gamma1=1.50;// m^2/s
x_1=0;
y_1=1;
x_2=1;
y_2=-1;
x=1.0;
y=0;// where all spatial coordinates are in meters.

//Calculation
//From fig.10-53,The vortex is located 1 m above the point (1, 0) and vortex velocity has positive i direction
r_vortex=1.00;// m
V_vortex=[gamma1/(2*%pi*r_vortex) 0];// m/s
//Similarly, the first source induces a velocity at point (1, 0) at a 45  angle from the x-axis as shown in Fig. 10 53 .
r_source1=sqrt(2);// m
V_source1=(V1)/(2*%pi*r_source1);// Resultant vector in m/s
theta=45;// angle between two vectors
// Function to find the velocity vector in i and j direction from resultant vector
 function [X]=fric(f)
```

```
    X(1)=f(1)^2 + f(2)^2-V_source1^2; // modulus(r)=sqrt(x^2+y^2)
    X(2)=tand(theta)*f(1)-f(2);// theta=tan^-1(y/x)
 endfunction

    f=[0.01 0.01]; // Initial guess to solve X
    V_source1_vec=fsolve(f,fric);// m/s (Calculating friction factor)

//Finally, the second source (the sink) induces a velocity straight down i.e in the negative j direction
r_source2=1.00;/// m
V_source2=[0 (V2)/(2*%pi*r_source2)];// m/s
V=V_vortex+V_source1_vec+V_source2;//The resultant velocity in m/s
printf('\nThe resultant velocity, V = %0.3fi %1.0fj\n',V);
```

Scilab code Exa 10.8 `Flow into a Vacuum Cleaner Attachment`

```
//Example 10_8
clc;clear;funcprot(0)
// Given values
w=2.0;// Width in mm
L=35.0;// Length in cm
b=2.0;// Distance in cm
v_dot=0.110;// The total volume flow rate in m^3/s
u_starmax=0.159;// m/s
// Calculation
v_dotbyL=-(v_dot/(L/100));// Strength of line source in m^2/s
u_max=-(u_starmax*(v_dotbyL/(b/100)));// Maximum speed along the floor
printf('\nStrength of line source=%0.3f m^2/s \
```

```
   nMaximum speed along the floor ,u_max=%0.2 f m/s ',
   v_dotbyL ,u_max);
```

Scilab code Exa 10.9 `Laminar or Turbulent Boundary Layer`

```
// Example 10_9
clc;clear;funcprot(0);
//Given data
V=5.0;// Uniform speed in mi/h
x=16;// Length in ft
T=50;//  F
nu=1.407*10^-5;// The kinematic viscosity of water
   in ft^2/s

// Calculation
Re_x=(V*x)/nu;// The Reynolds number at the stern of
    the canoe
Re_cr=1*10^5;// Critical Reynolds number
if(Re_x>Re_cr)
    printf('\nThe boundary layer is definitely
       turbulent by the back of the canoe.');
else
    printf('\nThe boundary layer is definitely
       laminar');
end
```

Scilab code Exa 10.11 `Displacement Thickness in the Design of a Wind Tunnel`

```
// Example 10_11
clc;clear;funcprot(0);
//Given data
T=19;//  C
D=30/100;// Diameter in m
```

```
x=30/100;// Length of the tunnel in m
V_b=4.0;// Velocity at beginning in m/s
nu=1.507*10^-5;// m^2/s

// Calculation
Re_x=(V_b*x)/nu;// Reynolds number
delta=((1.72*x)/(sqrt(Re_x))*10^3;// The
   displacement thickness at the end of the test
   section in mm
R=D/2;// Radius of the tunnel in m
V_end=(V_b*(%pi*R^2))/(%pi*(R-(delta/1000))^2);//
   The average air speed at the end of the test
   section in m/s
printf('\nThe average air speed at the end of the
   test section=%0.2f m/s',V_end);
```

Scilab code Exa 10.12 Comparison of Laminar and Turbulent Boundary Layers

```
// Example 10_12
clc;clear;funcprot(0);
//Given data
V=10.0;// m/s
L=1.52;// m

//Properties
nu=1.516*10^-5;// m^2/s

//Calculation
//(a)
x=L;// m
Re_x=(V*x)/nu;// Reynolds number
L=L*1000;// mm
x=[0,L];// mm

//For laminar case
```

```
for(i=1:2)
del_laminar(i)=(4.91*x(i))/sqrt(Re_x);// mm
del_turbulenta(i)=(0.16*x(i))/(Re_x)^(1/7);// mm
del_turbulentb(i)=(0.38*x(i))/(Re_x)^(1/5);// mm
end
xlabel('x,m');
ylabel('delta,mm');
x=x/1000;
plot(x,del_laminar,'b',x,del_turbulenta,'r',x,del_turbulentb,'g');
legend(['Laminar','Turbulent(a)','Turbulent(b)'],"in_upper_left");
//(b)
// For laminar boundary layer,
C_fxl=0.664/sqrt(Re_x);
// For turbulent boundary layer,
C_fxt=0.027/(Re_x)^(1/7);
printf('\nThe laminar boundary layer thickness at this same x-location=%0.2f mm \nThe turbulent boundary layer thickness at this same x-location=%0.1f mm \nThe local skin friction coefficient for the laminar boundary layer=%0.2e \nThe local skin friction coefficient for the turbulent boundary layer=%0.1e',del_laminar(2),del_turbulenta(2),C_fxl,C_fxt);
// The answer vary due to round off error
```

Scilab code Exa 10.15 Drag on the Wall of a Wind Tunnel Test Section

```
// Example 10_15
clc;clear;funcprot(0);
//Given data
T=20;//  C
L=1.8;// Length in m
w=0.50;// Width in m
```

```
U=10;// Velocity of the flow in m/s
delta_1=4.2/100;// Boundary layer thickness 1 in m
delta_2=7.7/100;// Boundary layer thickness 2 in m
nu=1.516*10^-5;// m^2/s
rho=1.204;// kg/m3

// Calculation
F_d=(w*rho*U^2)*(4/45)*(delta_2-delta_1);// Drag force in N
printf('\nThe total skin friction drag force=%0.2f N',F_d);
```

Chapter 11

FLOW OVER BODIES DRAG AND LIFT

Scilab code Exa 11.1 `Measuring the Drag Coefficient of a Car`

```
//Example 11_1
clc;clear;funcprot(0);
//Properties
rho=0.07489;//The density of air in lbm/ft^3
//Given values
P_atm=1;// atm
T=70;// F
F_d=68;// Force in lbf
V=60*1.467;// ft/s^2
A=22.26;// ft^2

//Calculation
C_d=(2*F_d*(32.2))/(rho*A*V^2);//The drag
    coefficient of the car
printf('The drag coefficient of the car ,C_d=%0.2f \
   n',C_d);
```

Scilab code Exa 11.2 `Effect of Mirror Design on the Fuel Consumption of a Car`

```
//Example 11_2
clc;clear;funcprot(0);
//Properties
rho_a=1.20;// The density of air in kg/m^3
rho_g=.8;//The density of gasoline in kg /L
n_o=0.3;// The over all efficiency of the engine
C_dc=1.1;// The drag coefficient for a circular disk
C_dh=0.4;//The drag coefficient for a hemispherical
   body
HV=44000;// The heating value of gasoline in kJ/kg

// Given values
V=95;// km/h
Pr=0.60;//Price of gasoline in $/L
D=0.13;// m
L=24000;// km/year

//Calculation
A=(%pi*0.13^2)/4;//m^2
F_d=(C_dc*A*rho_a*V^2)/(2*3.6^2);//The drag force
   acting on the flat mirror in N
W_drag=F_d*L;// kJ/year
E_in=W_drag/n_o;// kJ/year
m_f=E_in/HV; // kg/year
Amount=m_f/rho_g;// L/year
Cost=(Amount*Pr);// $/year
Rr=(C_dc-C_dh)/C_dc;// Reduction ratio
Fr=Rr*Amount;// Fuel reduction in L/year
printf('Fuel reduction =%0.2f L/year\n',Fr);
Cr=Rr*Cost;// Cost reduction in $/year
printf('Cost reduction =%0.2f $/year\n',Cr);
// The answer vary due to round off error
```

Scilab code Exa 11.3 `Flow of Hot Oil over a Flat Plate`

```
//Example 11_3
clc;clear;funcprot(0);
//Assumptions
Re_cr=5*10^5;
//Properties
rho=876;//The density of engine oil at 40 C  kg/m^3
nu=2.485*10^-4;//m^2/s
//Given values
V=2;// Free stream velocity in m/s
L=5;// m
b=1;//m

//Calculation
Re_L=(V*L)/nu;// The Reynolds number at the end of the plate
C_f=1.328*Re_L^(-0.5);// The average friction coefficient
A=L*b;// m^2
F_d=C_f*A*rho*(V^2/2);// N
printf('The drag force,F_d =%0.0f N\n',F_d);
```

Scilab code Exa 11.4 `Drag Force Acting on a Pipe in a River`

```
//Example 11_4
clc;clear;funcprot(0);
//Properties
rho=999.1;//kg/m^3
mu=1.138*10^-3;// kg/m.s
//Given values
D=0.022;// m
V=4;// m/s
L=30;// m
A=L*D;// m^2
```

```

//Calculation
Re=(rho*V*D)/mu;
//The drag coefficient corresponding to the value Re from Fig. 11 34
C_d=1;
F_d=C_d*A*rho*(V^2/2);
printf('The drag force acting on the pipe,F_d =%0.0f N\n',F_d);
disp('The drag force acting on the pipe,F_d ~=5300 N');
```

Scilab code Exa 11.5 Lift and Drag of a Commercial Airplane

```
//Example 11_5
clc;clear;
//Properties
rho_ag=1.20;// kg/m^3
rho_ac=0.312;// kg/m^3
C_Lmax1=1.52;// The maximum lift coefficient of the wing with flaps
C_Lmax2=3.48;// The maximum lift coefficient of the wing without flaps
//Given values
m=70000;// kg
A=150;// m^2
V=558;/// km/h
g=9.81;// m/s^2

// Calculation
//(a)
W=m*g;// N
V=V/3.6;// m/s
V_min1=sqrt((2*W)/(rho_ag*C_Lmax1*A));// m/s
V_min2=sqrt((2*W)/(rho_ag*C_Lmax2*A));// m/s
```

```
V_1s=1.2*V_min1*3.6;// 1 m/s=3.6 km/h
printf('(a)Without flaps:V_min1,safe =%0.0f km/h\n',V_1s);
V_2s=1.2*V_min2*3.6;// 1 m/s=3.6 km/h
printf('     With flaps:V_min2,safe =%0.0f km/h\n',V_2s);
//(b)
F_l=W;// N
C_l=F_l/(1/2*rho_ac*V^2*A);// The lift coefficient
//For the case with no flaps, the angle of attack corresponding to thrist value of C_L is determined from Fig. 11 45  to be
alpha=10;// The angle of attack in degree
printf('(b)The angle of attack,alpha~=%0.0f degree\n',alpha);
//(c)
// From Fig.11-45,C_d~=0.03
C_d=0.03;// The drag coefficient
F_d=(C_d*A*rho_ac*(V^2/2))/1000;//kN
P=F_d*V;// kW
printf('(c)The power that needs to be supplied to provide enough thrust to overcome wing drag,P=%0.0f kW\n',P);
// The answer vary due to round off error
```

Scilab code Exa 11.6 Effect of Spin on a Tennis Ball

```
//Example 11_6
clc;clear;funcprot(0);
//Properties
rho=0.07350;// lbm/ft^3
nu=1.697*10^-4;// ft^2/s
//Given values
m=0.125;//lbm
D=2.52;// in
```

```
V=45;// mi/h
n=4800;// rpm
P=1;// atm
T=80;// degree F
g=9.81;// m/s^2

//Calculation
V=(45*5280)/3600;// ft/s
omega=(2*%pi*n)/60;// rad/s
C=(omega*D)/(2*V);//rad
//From Fig. 11 53 , the lift coefficient corresponding to C
C_l=0.21;
A=(%pi*D^2)/4;// ft^2
F_l=(C_l*A*rho*V^2)/(2*32.2);// lbf
W=(m*g)/32.2;// lbf
if(W<=0.125)
    printf('drop')
else
    printf('Wrong')
end
```

Chapter 12

COMPRESSIBLE FLOW

Scilab code Exa 12.1 `Compression of High Speed Air in an Aircraft`

```
//Example 12_1
clc;clear;funcprot(0);
// Given values
T_1=255.7;// The ambient air temperature in K
P_1=54.05;//The atmospheric pressure in kPa
V_1=250;// m/s
h=5000;// m
P_r=8;// Pressure ratio of the compressor
// Properties
C_p=1.005;//The constant-pressure specific heat C_p
   in kJ/kg.k
k=1.4;// The specific heat ratio

//Calculation
//(a)
T_01=T_1+(V_1^2/(2*C_p*1000));//The stagnation
   temperature at the compressor inlet in K
P_01=P_1*(T_01/T_1)^(k/(k-1));//kPa
printf('The stagnation pressure at the compressor
   inlet,P_01=%0.2f kPa\n',P_01);
//(b)
```

```
// P_r=(P_02/P_01)
T_02=T_01*(P_r)^((k-1)/k);//The stagnation temperature of air at the compressor exit in K
W_in=C_p*(T_02-T_01);//the compressor work per unit mass of air in kJ/kg
printf('The compressor work per unit mass of air, W_in=%0.1f kJ/kg\n',W_in);
```

Scilab code Exa 12.2 `Mach Number of Air Entering a Diffuser`

```
//Example 12_2
clc;clear;funcprot(0);
// Given values
V=200;// Velocity in m/s
T=303;// Temperature in K
//Properties
k=1.4;// The specific heat ratio
R=0.287;//The gas constant of air in kJ/(kg.K)

//Calculation
//(a)
c=sqrt(k*R*T*1000);//The speed of sound in air at 30 C in m/s
printf('(a)The speed of sound in air at 30 C ,c=%0.0f m/s\n',c);
//(b)
Ma=V/c;
printf('(b)The Mach number ,Ma=%0.3f \n',Ma);
```

Scilab code Exa 12.3 `Gas Flow through a Converging Diverging Duct`

```
//Example 12_3
clc;clear;funcprot(0);
```

```
// Given values
m=3;//Mass flow rate in kg/s
T_0=473;// T_0=T_1 in K
P_0=1400;// P_0=P_1 in kPa
P=1200;// kPa
// Properties
C_p=0.846;// kJ/(kg.K)
k=1.289;
R=0.1889;// kJ/(kg.K)

//Calculation
T=T_0*(P/P_0)^((k-1)/k);// k
V=sqrt(2*C_p*(T_0-T)*1000);// m/s
printf('Velocity ,V=%0.1f m/s\n',V);
rho=P/(R*T);// kg/m^3
printf('Density ,rho=%0.1f kg/m^3\n',rho);
A=(m/(rho*V))*10000;//cm^2
printf('Area ,A=%0.1f cm^2\n',A);
c=sqrt(k*R*T*1000);// m/s
Ma=V/c;
printf('Mach number ,Ma=%0.3f \n',Ma);
// The answer vary due to round off error
```

Scilab code Exa 12.4 Critical Temperature and Pressure in Gas Flow

```
//Example 12_4
clc;clear;funcprot(0);
// Given values
T_0=473;// T_0=T_1 in K
P_0=1400;// P_0=P_1 in kPa
// Properties
k=1.289;//The specific heat ratio of carbon dioxide

//Calculation
//T_1=T_c/T_0
```

```
T_1=2/(k+1);
T_c=T_1*T_0;//The critical temperature in K
printf('The critical temperature T*=%0.0f  K\n',T_c)
   ;
//P_1=P_c/P_0
P_1=(2/(k+1))^(k/(k-1));
P_c=P_1*P_0;//The critical pressure in KPa
printf('The critical pressure P*=%0.0f KPa\n',P_c);
```

Scilab code Exa 12.5 `Effect of Back Pressure on Mass Flow Rate`

```
//Example 12_5
clc;clear;funcprot(0);
//Properties
C_p=1.005;// kJ/kg.K
k=1.4;//The specific heat ratio
R=0.287;//kJ/kg.K
//Given values
P_i=1;// MPa
T_i=873;// K
V_i=150;// m/s
A_t=.0050;// m^2
P_b1=0.7;// MPa
P_b2=0.4;//MPa

//Calculation
T_0i=T_i+((V_i^2/(2*C_p)))/1000;// K
P_0i=P_i*(T_0i/T_i)^(k/(k-1)); // MPa
T_0=T_0i;// K
P_0=P_0i;// K
//P_cr=P*/P_0
P_cr=(2/(k+1))^(k/(k-1));

//(a)
P_br=P_b1/P_0;
```

```
P_t=P_b1;
//From table A-13
Ma_1=0.778;
T_cr=0.892;// T_cr=T_t/T_0
T_t=0.892*T_0;
rho_t=P_t*1000/(R*T_t);// kg/m^3
V_t=Ma_1*sqrt(k*R*T_t*1000);// m/s
m=rho_t*A_t*V_t;//kg/s
printf(' (a) The mass flow rate through the nozzle,m=%0.2f kg/s\n',m);

//(b)
P_br=P_b2/P_0;
//P_br is less than the critical-pressure ratio, 0.5283.Therefore, sonic conditions exist at the exit plane (throat) of the nozzle, and Ma =1.
m_1=(A_t*P_0*1000*sqrt(k/(R*T_0))*(2/(k+1))^((k+1)/(2*(k-1))))*sqrt(1000);// kg/s
printf(' (b) The mass flow rate through the nozzle,m=%0.2f kg/s\n',m_1);
// The answer vary due to round off error
```

Scilab code Exa 12.6 Gas Flow through a Converging Nozzle

```
// Example 12_6
clc;clear;funcprot(0);
//Given values
T_1=400; // K
P_1=100; // kPa
Ma_1=0.3;// Mach number

// Calculation
//From table A-13.At the initial Mach number of Ma=0.3, we read
// a_1=A1/A*; t_1=T1/T0; p_1=P1/P0;t_2=T1/T0;p_2=P2/
```

```
    P0;
a_1=2.031;
t_1=0.9823;
p_1=0.9395;
// A2=0.8*A1;
//a_2=(A2/A*)=(A2/A1)*(A1/A*);
a_2=0.8*a_1;
//From table A-13,for the value of a_2
t_2=0.9703;
p_2=0.9000;
Ma_2=0.391;
T_2=T_1*(t_2/t_1);// K
P_2=P_1*(p_2/p_1);// kPa
printf('Mach number,Ma_2=%0.3f\n',Ma_2);
printf('Temperature,T_2=%0.0f K\n',T_2);
printf('Pressure,P_2=%0.1f kPa\n',P_2);
```

Scilab code Exa 12.7 Airflow through a Converging Diverging Nozzle

```
// Example 12_7
clc;clear;funcprot(0);
//Given values
P_0=1000;// kPa;
T_0=800;// K
k=1.4;//The specific heat ratio of air
Ma_2=2;// Exit Mach number
a=20;// Throat area in cm^2
//Properties
R=0.287;// kJ/kg.k

// Calculation
rho_0=P_0/(R*T_0);// kg/m^3
P_0=1;// MPa
//(a)At the throat of the nozzle Ma=1, and from
    Table A 13
```

```
//P*=P_c;T*=T_c;rho*=rho_c;V*=V_c;c*=c_c;
P_c=0.5283*P_0;// MPa
printf('(a)The throat conditions,P*=%0.4f MPa\n',P_c);
T_c=0.8333*T_0;// K
printf('                         T*=%0.1f K\n',T_c);
rho_c=0.6339*rho_0;// kg/m^3
printf('                         rho*=%0.3f kg/m^3\n',rho_c);
V_c=sqrt(k*R*T_c*1000);// m/s
printf('                         V*=c*=%0.1f m/s\n',V_c);

//(b)For Ma_2=2,by using data from Table A 13
P_e=0.1278*P_0;// MPa
printf('(b)The exit plane conditions,P_e=%0.4f MPa\n',P_e);
T_e=0.5556*T_0;// K
printf('                             T_e=%0.1f K\n',T_e);
rho_e=0.23000*rho_0;// kg/m^3
printf('                             rho_e=%0.3f kg/m^3\n',rho_e);
A_e=1.6875*a;// cm^2
printf('                             A_e=%0.2f cm^2\n',A_e);
Ma_e=1.6330;// Critical Mach number
V_e=Ma_e*V_c;// m/s
printf('                             V_e=%0.1f m/s\n',V_e);
c_e=sqrt(k*R*T_e*1000);// The speed of sound at the exit condition in m/s
V_e=Ma_2*c_e;// m/s

//(c)
m=rho_c*(a*10^-4)*V_c;
printf('(c)The mass flow rate,m=%0.2f kg/s\n',m);
```

Scilab code Exa 12.9 `Shock Wave in a Converging Diverging Nozzle`

```
// Example 12_9
clc;clear;funcprot(0);
//From example 12_7
//Given values
P_0=1000;// kPa;
T_0=800;// K
Ma_1=2;// Exit Mach number
a=20;// Throat area in cm^2
//Properties
R=0.287;// kJ/kg.k
C_p=1.005;// kJ/kg.k
k=1.4;//The specific heat ratio of air

// Calculation
//(a)
//From example 12_7
P_01=1.0;// MPa
P_1=0.1278; // MPa
T_1=444.5;// K
rho_1=1.002;// kg/m^3
// From table A-14,For Ma_1=2,we read
Ma_2=0.5774
P_02=0.7209*P_01;// MPa
printf('(a)The stagnation pressure,P_02=%0.3f MPa\n',P_02);
P_2=4.5000*P_1;// MPa
printf('The static pressure,P_2=%0.3f MPa\n',P_2);
T_2=1.6875*T_1;// K
printf('The static temperature,T_2=%0.0f K\n',T_2);
rho_2=2.6667*rho_1;// kg/m^3
printf('The static density,rho_2=%0.2f kg/m^3\n',rho_2);
```

```

//(b)
//gradS=s2-s1
gradS=(C_p*(log(T_2/T_1)))-(R*log((P_2/P_1)));
printf('(b)The entropy change across the shock,s2-s1
    =%0.4f kJ/kg.K\n',gradS);

//(c)
c_2=sqrt(k*R*T_2*1000);// The speed of sound at the
    exit conditions in m/s
V_2=Ma_2*c_2;
printf('(c)The exit velocity,V_2=%0.0f m/s\n',V_2);

//(d)
//The mass flow rate in this case is the same as
    that determined in Example 12_7:
V_1=517.5;// m/s
rho_c=2.761;// kg/m^3
m=rho_c*(a*10^-4)*V_1;// kg/s
printf('(d)The mass flow rate,m=%0.2f kg/s\n',m);
```

Scilab code Exa 12.10 `Estimation of the Mach Number from Mach Lines`

```
//Example 12_10
clc;clear;
// Given values
mu=19;// Angle of Mach lines in degrees

// Calculation
// mu=asind(1/Ma_1)
Ma_1=1/sind (19);// Mach number
printf('Mach number,Ma=%0.2f \n',Ma_1);
```

Scilab code Exa 12.11 `Oblique Shock Calculations`

```
// Example 12_11
clc;clear;funcprot(0);
//Given values
Ma_1=2;// Mach number
delta=10;// degree
P_1=75.0;// kPa
//Properties
k=1.4;// Specific heat ratio

// Calculation
theta=delta;// Deflection in degrees
beta_w=39.3;// Oblique shock angle in degrees
beta_s=83.7;// Oblique shock angle in degrees
Ma_1nw=Ma_1*sind(beta_w);// Mach Number on upstream side
Ma_1ns=Ma_1*sind(beta_s);// Mach Number on upstream side
Ma_2nw=0.8032;// Mach number
Ma_2ns=0.5794;// Mach number
P_2w=P_1*((2*k*(Ma_1nw)^2)-k+1)/(k+1);// Pressure in kPa
P_2s=P_1*((2*k*(Ma_1ns)^2)-k+1)/(k+1);// Pressure in kPa
Ma_2w=(Ma_2nw)/(sind(beta_w-theta));// Mach Number on the downstream side
Ma_2s=(Ma_2ns)/(sind(beta_s-theta));// Mach Number on the downstream side
printf('\nThe pressure on the downstream side,P_2=%0.0f kPa(weak shock) & P_2=%0.0f kPa(strong shock)\nThe Mach number on the downstream side of the oblique shock,Ma_2=%0.2f (weak shock) & Ma_2=%0.3f (strong shock)',P_2w,P_2s,Ma_2w,Ma_2s);
disp(Ma_1nw)
```

Scilab code Exa 12.12 `Prandtl Meyer Expansion Wave Calculations`

```
// Example 12_12
clc;clear;funcprot(0);
//Given values
Ma_1=2.0;// Mach number
P_1=230;// kPa
delta=10;// degree
//Properties
k=1.4//The specific heat ratio

//Calculation
theta=delta;
v_1=(sqrt((k+1)/(k-1))*atand(sqrt(((k-1)*(Ma_1^2-1))/(k+1))))-atand(sqrt(Ma_1^2-1));// degree
v_2=theta+v_1;// degree
// Ma_2=y(1);
function[X]=Machnumber(y);
    X(1)=((sqrt((k+1)/(k-1))*atand(sqrt(((k-1)*(y(1)^2-1))/(k+1))))-atand(sqrt(y(1)^2-1))-v_2);
endfunction
y=[1];
z=fsolve(y,Machnumber);
printf('The downstream Mach number Ma_2=%0.3f\n',z(1));
Ma_2=z(1);
P_2=((((1+(((k-1)/2)*Ma_2^2)))^(-k/(k-1)))/(((1+(((k-1)/2)*Ma_1^2)))^(-k/(k-1))))*(P_1);
printf('The downstream pressure,P_2=%0.0f kPa\n',P_2);
```

Scilab code Exa 12.15 `Rayleigh Flow in a Tubular Combustor`

```
//Example 12_15
clc;clear;
//Properties
k=1.4;
C_p=1.005;// kJ/kg*K
R=0.287;// kJ/kg*K
// given values
D=0.15;// m
V_1=80;// m/s
T_1=550;// K
P_1=480;// kPa
HV=42000;// kJ/kg
AF=40;

//Calculation
rho_1=P_1/(R*T_1);// kg/m^3
A=%pi*D^2*V_1;// m^2
m_air=rho_1*A*V_1; // kg/s
m_f=m_air/AF;// kg/s
Q=m_f*HV;// kW
q=Q/m_air;// kJ/kg
T_01=T_1+(V_1^2/(2*C_p*1000));// K
c_1=sqrt(k*R*T_1); // m/s
Ma_1=V_1/c_1;
T_02=+(q+C_p);// K
// From Table A-15
T_c=T_01/0.1291;// K
T_c1=T_02/T_c;
//Using T_c1 value & From Table A-15
Ma_2=0.3142;
printf('The exit Mach number ,Ma_2=%0.4f \n',Ma_2);
T_2=2.848*T_1;// K
printf('The exit temperature,T_2=%0.0f K\n',T_2);
P_2=0.9142*P_1;// kPa
printf('The exit pressure ,P_2=%0.0f kPa\n',P_2);
V_2=3.117*V_1;// m/s
printf('The exit velocity ,V_2=%0.0f m/s\n',V_2);
```

Scilab code Exa 12.16 `Choked Fanno Flow in a Duct`

```
//Example 12_16
clc;clear;
// Given values
D=3/100;// Diameter in m
P_1=150;// kPa
T_1=300;// K
Ma_1=0.4;// Mach number

// Properties
k=1.4;// Specific heat ratio
C_p=1.005;// kJ/kg.K
R=0.287;// kJ/kg.K
nu=1.58*10^-5;//Kinematic viscosity in m^2/s

// Calculation
c_1=sqrt(k*R*T_1*1000);// m/s
V_1=Ma_1*c_1;// Mach number
Re_1=(V_1*D)/nu;// The inlet Reynolds number
// The friction factor is determined from the Colebrook equation,
function[X]=frictionfactor(y)
    X(1)=real(-(2.0*log10((0/3.7)+(2.51/((Re_1)*sqrt(y(1)))))))-(1/sqrt(y(1)));
endfunction
y=[0.01];
z=fsolve(y,frictionfactor);
f=z(1);
// The Fanno flow functions corresponding to the inlet Mach number of 0.4,From Table A-16
P_0r=1.5901;// (P_0r=P_01/P_0*)
T_r=1.1628;// (T_1r=T_1/T*)
P_r=2.6958;// (P_1r=P_1/P*)
```

```
V_r=0.4313;// (V_1r=V_1/V*)
fL_D=2.3085;
L_1=((fL_D*D)/f);// m
T_c=T_1/T_r;// K
P_c=P_1/P_r;// kPa
V_c=V_1/V_r;// m/s
P_01L=(1-(1/P_0r))*100;
printf('\nThe duct length=%0.2f m \nThe temperature
   at exit=%0.0f K \nThe pressure at exit=%0.1f kPa
   \nThe velocity at exit=%0.0f m/s \nThe percentage
    of stagnation pressure lost in the duct=%0.1f
   percentage',L_1,T_c,P_c,V_c,P_01L);
```

Scilab code Exa 12.17 `Exit Conditions of Fanno Flow in a Duct`

```
//Example 12_17
clc;clear;
// Given values
V_1=85;// m/s
P_1=220;// kPa
T_1=450;// K
f=0.023;// The average friction factor for the duct
L=27;// m

// Properties
k=1.4;// Specific Heat ratio
C_p=1.005;// kJ/kg.K
R=0.287;// kJ/kg.K

// Calculation
c_1=sqrt(k*R*T_1*1000);// m/s
Ma_1=(V_1/c_1);
// From Table A-16,
fLbyDh1=14.5333;
D_h=0.05;// m
```

```
fLbyDh=(f*L)/D_h;
fLbyDh2=fLbyDh1-fLbyDh;
// The Mach number corresponding to this value of fL*/D is 0.42, obtained from Table  A 16 ,
Ma_2=0.42;// The Mach number at the duct exit
rho_1=(P_1)/(R*T_1);// kg/m^3
A=(%pi/4)*(D_h)^2;// m^2
m_air=rho_1*A*V_1;// kg/s
printf('\nThe Mach number at the duct exit=%0.2f \nThe mass flow rate of air=%0.3f kg/s',Ma_2,m_air);
L_max1=fLbyDh1*(D_h/f);// m
L_max2=fLbyDh2*(D_h/f);// m
printf('\nThe maximum length at inlet=%0.1f m \nThe maximum length at exit=%0.1f m',L_max1,L_max2);
```

Chapter 13

OPEN CHANNEL FLOW

Scilab code Exa 13.1 `Character of Flow and Alternate Depth`

```
//Example 13_1
clc;clear;
// given values
b=0.4;// Width in m
v=0.2;// Flow rate in m^3/s
y_1=0.15;// Flow depth in m
g=9.81;// m/s^2

// Calculation
A_c=y_1*b;// m^2
V=(v/A_c);//The average flow velocity in m/s
printf('The average flow velocity,V=%0.2f m/s\n',V);
y_c=(v^2/(g*b^2))^(1/3);// The critical depth  in m
printf('The critical depth for this flow,y_c=%0.3f m\n',y_c);
printf('Therefore, the flow is SUPER CRITICAL since the actual flow depth is y=0.15 m, and y<yc.\n');
Fr=(V*sqrt(g*y_1));// The Froude number
E_s1=y_1+((v^2/(2*g*b^2*y_1^2)));//The specific energy for the given condition in m
//Then the alternate depth is determined E_s1=E_s2;
```

```
   y_2=y(1)
function[X]=depth(y);
    X(1)=(y(1)+((0.2^2)/(2*9.81*0.4^2*y(1)^2)))
       -0.7163;
endfunction
y=[0.5];
z=fsolve(y,depth);
printf('The alternate depth y_2=%0.2f m\n',z);
```

Scilab code Exa 13.2 `Flow Rate in an Open Channel in Uniform Flow`

```
//Example 13_2
clc;clear;
// Given values
b=0.8;// Width in m
y=0.52;// Flow depth in m
g=9.81;// m/s^2
theta=60;// Trapezoid angle  in degree
alpha=0.3;// Bottom slope angle
//Properties
n=0.030;// The Manning coefficient for an open
   channel with weedy surfaces

//Calculation
A_c=(y*(b+(y/tand(theta))));//The cross-sectional
   area in m^2
p=b+((2*y)/sind(theta));// Perimeter in m
R_h=A_c/p;// Hydraulic radius of the channel
S_0=tand(alpha);//The bottom slope of the channel
a=1;// m^(1/3)/s
v=(a/n)*(A_c*R_h^(2/3)*S_0^(1/2));// The flow rate
   through the channel in m^3/s
printf('The flow rate through the channel is
   determined from the Manning equation to be,v=%0.2
   f m^3/s\n',v);
```

```
//The flow rate for a bottom angle of 1  can be determined by using S_0= tan alpha=tan 1
alpha_1=1;// degree
S_01=tand(alpha_1);// The bottom slope of the channel
v=(a/n)*(A_c*R_h^(2/3)*S_01^(1/2));// The flow rate through the channel in m^3/s
printf('The flow rate for a bottom angle of 1 ,v=%0.1f m^3/s\n',v);
```

Scilab code Exa 13.3 `The Height of a Rectangular Channel`

```
//Example 13_3
clc;clear;funcprot(0);
// Given values
b=4;// Bottom width in m
V=51;// Flow rate in ft^3/s
// Properties
n=0.014;//The Manning coefficient
// Calculation
//The cross-sectional area, perimeter, and hydraulic radius of the channel are A_c=4y;p=4+2y;R_h=A_c/p=(4y)/(4+y);
S_0=2/1000;

//Using the Manning equation, the flow rate through the channel can be expressed as Vdot=(a/n)*A_c*R_h^(2/3)*S_0^(1/2)
// y=y(1)
function[X]=flowdepth(y);
    X(1)=real(((1.486/n)*(4*y(1))*((4*y(1))/(4+(2*y(1))))^(2/3)*(S_0)^(1/2))-V);
endfunction
y=[1];
z=fsolve(y,flowdepth);
```

```
printf('If S_0=2/1000=0.002.The flow depth is determined to be y=%0.1f ft\n',z(1));

// If the bottom drop were just 1 ft per 1000 ft length, the bottom slope would be
S_0=0.001;
// y=y(2)
function[X]=flowdepth(z);
    X(1)=real(((1.486/0.014)*(4*z(1))*((4*z(1))/(4+(2*z(1))))^(2/3)*(0.001)^(1/2))-51);
endfunction
y=[1];
y=fsolve(z,flowdepth);
printf('If the bottom slope would be S_0=.001, and the flow depth would be y=%0.1f ft\n',y(1));
```

Scilab code Exa 13.4 `Channels with Nonuniform Roughness`

```
//Example 13_4
clc;clear;
// Given values
S_0=0.003;// Bottom slope
n_1=0.030;
n_2=0.050;

// Calculation
s=sqrt(3^2+3^2);
//Then the flow area, perimeter, and hydraulic radius for each subsection and the entire channel become
// Subsection 1:
A_c1=21;// m^2
p_1=10.486; // m
R_h1=A_c1/p_1;// m
// Subsection 2:
```

```
A_c2=16;// m^2
p_2=10;// m
R_h2=A_c2/p_2;// m
// Entire channel
A_c=A_c1+A_c2;// m^2
p=p_1+p_2;// m
R_h=A_c/p;// m
//Using the Manning equation for each subsection,
a=1;//m^(1/3)/s
v_1=(a/n_1)*(A_c1*R_h1^(2/3))*(S_0)^(1/2);// m^3/s
v_2=(a/n_2)*(A_c2*R_h2^(2/3))*(S_0)^(1/2);// m^3/s
v=v_1+v_2;// m^3/s
printf('The total flow rate through the channel,V=%0.0f m^3/s\n',v);
n_eff=(a*A_c*R_h^(2/3)*S_0^(1/2))/v;
printf('The effective Manning coefficient for the entire channel ,n_eff=%0.3f \n',n_eff);
```

Scilab code Exa 13.5 `Best Cross Section of an Open Channel`

```
//Example 13_5
clc;clear;funcprot(0);
// Given values
v=2;// m^3/s
S_0=0.001;
a=1;// m^1/3
//Properties
n=0.016;

//Calculation
//(a)
b=((2*n*v*4^(2/3))/(a*sqrt(S_0)))^(3/8); //The channel width in  m
y=b/2;// The flow height in m
printf('(a)The channel width,b=%0.2f m\n',b);
```

```
printf('The flow height,y=%0.2f m\n',y);
//(b)
b_1=((n*v)/((0.75*sqrt(3))*(sqrt(3)/4)^(2/3)*(1*sqrt(0.001))))^(3/8);
p=3*b;// m
y_1=(sqrt(3)/2)*b_1;// m
theta=60;// degree
printf('(b)The channel width,b=%0.2f m\n',b_1);
printf('The flow height,y=%0.3f m\n',y_1);
printf('The trapezoidal angle,theta=%0.0f degree\n',theta);
// The answer vary due to round off error
```

Scilab code Exa 13.6 `Classification of Channel Slope`

```
//Example 13_6
clc;clear;
// Given values
b=6;//Width in m
S_0=0.004;// The bottom slope
y=2;// m
g=9.81;// m/s^2
//Properties
n=0.014;// The Manning coefficient
a=1;//The factor a is a dimensional constant in m^(1/3)/s

//Calculation
A_c=y*b;//The cross sectional area in m^2
p=b+(2*y);// Perimeter in m
R_h=A_c/p;// Hydraulic radius in m
V=(a/n)*A_c*R_h^(2/3)*S_0^(1/2);
printf('The flow rate,V=%0.1f m^3/s\n',V);
// y=y_n=2 m
y_c=V^2/(g*A_c^2);
```

```
disp("This channel at these flow conditions is
    classified as STEEP since y_n <y_c ,and the flow
    is supercritical.")
```

Scilab code Exa 13.7 Hydraulic Jump

```
//Example 13_7
clc;clear;
// Given values
b=10;// Width in m
y_1=0.8;// The flow depth in m
V_1=7;// Velocity before the jump in m/s
g=9.81;// m/s^2
rho=1000;// kg/m^3

// Calculation
//(a)
Fr_1=V_1/(sqrt(g*y_1));
y_2=0.5*y_1*(-1+sqrt(1+(8*Fr_1^2)));// The flow
    depth after the jump in m
printf('(a)The flow depth after the jump,y_2=%0.2f m
    \n',y_2);
V_2=(y_1/y_2)*V_1;//The flow depth after the jump in
     m/s
y_2=2.46;// m
Fr_2=V_2/(sqrt(g*2.46));
printf('    The Froude number after the jump,Fr_2=%0
    .3f \n',Fr_2);
//(b)
H_l=(y_1-2.46)+((V_1^2-V_2^2)/(2*g));// m
printf('(b)The head loss ,H_l=%0.3f m\n',H_l);
E_s1=y_1+(V_1^2/(2*g));//The specific energy of
    water before the jump in m
Dr=H_l/E_s1;
printf('   The dissipation ratio,Dr=%0.3f \n',Dr);
```

```
//(c)
V=b*y_1*V_1;// m/s
m=rho*V;// The mass flow rate of water in kg/s
E_d=(m*g*H_l)/1000;//kW
printf('(c)The wasted power production potential due to the hydraulic jump,E_d=%0.0f kW\n',E_d);
// The answers vary due to round off error
```

Scilab code Exa 13.8 `Sluice Gate with Drowned Outflow`

```
//Example 13_8
clc;clear;
// Given values
y_1=3;// m
y_2=1.5// m
a=0.25// m
b=6;// m
g=9.81// m/s^2

// Calculation
x_1=y_1/a;//The depth ratio
x_2=y_2/a;// The contraction coefficient
//The corresponding discharge coefficient is determined from Fig. 13 38
C_d=0.47;
v=C_d*b*a*sqrt(2*g*y_1);
printf('The rate of discharge,V=%0.2f m^3/s\n',v);
```

Scilab code Exa 13.9 `Subcritical Flow over a Bump`

```
//Example 13_9
clc;clear;funcprot(0)
// Given values
```

```
V_1=1.2;// The velocity in m/s
y_1=0.80;// The flow depth in m
gradz_b=0.15;// m
g=9.81;// m/s^2

// Calculation
Fr_1=(V_1/sqrt(g*y_1));// The upstream Froude number
y_c=(((y_1)^2*(V_1)^2)/(g))^(1/3);// The critical
    depth in m
E_s1=y_1+(((V_1)^2)/(2*g));// The upstream specific
    energy in m
// Solving equation y_2^3-(E_s1-gradz_b)y^2+(V_1^2)
    /(2*g)*y_1^2
coeff=[1,-(E_s1-gradz_b),0,((V_1^2)/(2*g)*y_1^2)];
y=roots(coeff);
d=y_1-(y(1)+gradz_b);// Depression in m
printf("The water surface is depressed over the bump
     in the amount of %0.2f m \n",d);
```

Scilab code Exa 13.10 Measuring Flow Rate by a Weir

```
//Example 13_10
clc;clear;
// Given values
b=5;// Width in m
y_1=1.5;// m
P_w=0.6;// m
g=9.81;// m^2/s

// Calculation
H=y_1-P_w;//The weir head in m
C_wd=0.598+(0.0897*(H/P_w));// The discharge
    coefficient of the weir
V=C_wd*(2/3)*b*sqrt(2*g)*H^(3/2);// TShe water flow
    rate through the channel
```

```
printf('The  water flow rate through the channel,V=%0.2f m^3/s\n',V);
// The answer vary due to round off error
```

Chapter 14

TURBOMACHINERY

Scilab code Exa 14.1 `Operating Point of a Fan in a Ventilation System`

```
//Example 14-1
clc;clear;funcprot(0);
//Given data
//Properties
//For air at 25 C
v=1.562*10^-5;// m^2/s
rho_a=1.184;// kg/m^3
rho_w=998.0;// kg/m^3
P_atm=101.3;// kPa
eps=0.15*10^-3;//Pipe roughness in m
D=0.230; //Inner diameter (ID) of the duct in  m
L=13.4;// m
V_cfm=50:50:700;// Volume flow rate in cfm (ft^3/min)
V=V_cfm*0.3048^3/60;// Volume flow rate in m^3/s
alpha=1.05;
g=9.81;// m/s^2

//Calculation
for i=1:1:length(V_cfm);
    Re=(4*V(i))/(v*%pi*D);//Reynolds number
```

```
    V_1=(4*V(i))/(%pi*D^2);//Velocity as a function of volume flow rate in m/s
    function [X]=fric(f)
    X=-2.0*log10(((eps)/(3.7*D))+((2.51)/(Re*sqrt(f))))-1/sqrt(f); //Friction factor as a implicit function of Re using Colebrook equation
    endfunction
    f=0.0001; //Initial guess to solve X
    fr=fsolve(f,fric);//Calculating friction factor
    sigmaK_l=1.3+5*(0.21)+1.8;// Minor losses
    H_ra=(alpha+(fr*L)/D+sigmaK_l)*(V_1^2/(2*g));//The required net head of the fan at the minimum flow rate
    H_req(i)=H_ra*(rho_a/(rho_w*0.0254));
end

printf('The operating point is at a volume flow rate of about 650 cfm, at which both the required and available net head equal about 0.83 inches of water. We conclude that the chosen fan is more than adequate for the job.\n');;
//From table 14-1
x = [0 250 500 750 1000 1200];
y = [0.90 0.95 0.90 0.75 0.40 0.0];
yi=smooth([x;y],0.1); //This script used to smooth the linear curve of x,y defined above
xgrid(3);
xlabel('Volume flow rate in cfm',"fontsize", 2);
ylabel('H, inches of water',"fontsize", 2);
plot(V_cfm',H_req,'r',yi(1,:),yi(2,:));
legend('H_required','H_available');
```

Scilab code Exa 14.2 `Selection of Pump Impeller Size`

```
//Example 14-2
clc;clear;
// Properties
rho=62.30;//The density of water at 70 F  in lbm/ft
    ^3
v=370;// gal/min
g=32.2;// ft/s^2
H_1=24;// ft
H_2=72.0;// ft
n_p2=0.765;
n_p1=0.70;//Efficiency of the pump

// Calculation
bhp_1=((rho*g*v*H_1)/n_p1)*((0.1337)/(32.2*60*550));
printf('Required bhp for the 8.25-in impeller option
    :bhp=%0.2f hp\n',bhp_1);
bhp_2=((rho*g*v*H_2)/n_p2)*((0.1337)/(32.2*60*550));
printf('Required bhp for the 12.75-in impeller
    option:bhp=%0.2f hp\n',bhp_2);
printf('Clearly, the smaller-diameter impeller
    option is the better choice in spite of its lower
     efficiency, because it uses less than half the
    power.');
```

Scilab code Exa 14.3 `Maximum Flow Rate to Avoid Pump Cavitation`

```
//Example 14-3
clc;clear;
// Given values
P_atm=101.3*1000; // Pa
g=9.81;// m/s^2
alpha=1.05;
eps=0.02*0.0254;//Roughness in m
D=4*0.0254;// in 'm' converted from 'in'
L=10.5*0.3048;//in 'm' converted from 'ft'
```

```
gradz=1.219;// grad z=(z_1-z_2) in m

// Calculation
A=((%pi*D^2)/4);//Area in m^2
v=300:10:700;//Volume flow rate in gpm
T=[25 60];//Temperature matrix
for j=1:1:length(T)
    //Water properties at T = 25 C  and 60 C respectively
    if T(j)==25 then
        rho=997.0;// kg/m^3
        nu=8.91*10^-4;// Kinematic viscosity in kg/m.s
        mu=nu/rho;
        P_v=3.169*1000;// Pa
    else
        rho=983.3;// kg/m^3
        nu=4.67*10^-4;// Kinematic viscosity in kg/m.s
        mu=nu/rho;
        P_v=19.94*1000;// Pa

    end

for i=1:1:length(v);

    v_(i)=(6.309*10^-5)*v(i); //Volume flow rate in m3^s converted from gpm
    V(i)=v_(i)/A;//Velocity in m/s
    Re=(4*v_(i))/(mu*%pi*D);//Reynolds number

    function [X]=fric(f)
    X=-2.0*log10(((eps)/(3.7*D))+((2.51)/(Re*sqrt(f))))-1/sqrt(f); //Friction factor as a implicit function of Re using Colebrook equation
    endfunction

```

```
    f=0.00001; //Initial guess to solve X
    fr=fsolve(f,fric);//Calculating friction factor

    sigmaK_l=0.5+(3*0.3)+6.0;// Minor losses
    H_l=((fr*L)/D+sigmaK_l)*(V(i)^2/(2*g));//The required net head of the fan at the minimum flow rate

    NPSH(j,i)=((P_atm-P_v)/(rho*g))+(gradz)-(H_l)-((alpha-1)*(V(i)^2)/(2*g));
end
end
F=[300 400 500 600 680];//Flow rate in gpm
N=[3.8 4.44 5.06 6.13 7.0];//minimum NPSH required approximately taken from Fig.14-21
plot(v',NPSH'*3.28,'r',F,N,'-o');
xlabel('v,gpm');
ylabel('NPSH,ft');
legend('Available NPSH, 25 C ','Available NPSH, 60 C ','Required NPSH');
printf('\nCavitation occurs at flow rates above approximately 600 gpm. \nThe maximum volume flow rate without cavitation decreases with temperature.')
```

Scilab code Exa 14.4 Volume Flow Rate through a Positive Displacement Pump

```
//Example 14-4
clc;clear;
//given values
V_lobe=0.45// cm^3
n=900;//rot/min
V_closed=2*V_lobe;
n_1=0.5;//rot(rotations)

```

```
// Calculation
v=(n*V_closed)/n_1;
printf('The volume flow rate of oil,v=%0.0f cm^3/min\n',v);
```

Scilab code Exa 14.5 `Idealized Blower Performance`

```
//Example 14-5
clc;clear;
// Properties
rho_a=1.20;// kg/m^3
rho_w=998;// kg/m^3
n=1750;
alpha_1=0;
alpha_2=40;
r_1=0.04;// m
r_2=0.08;// m
b_1=0.052;// m
b_2=0.023;// m
v=0.13;// m^3/s
g=9.81// m/s^2

// Calculation
V_1n=(v/(2*%pi*r_1*b_1));
V_1t=0;//since alpha_1=0
V_2n=(v/(2*%pi*r_2*b_2));
V_2t=V_2n*tand(40);
omega=(2*%pi*n)/60;
H=((omega/g)*((r_2*V_2t)-(r_1*V_1t)));
H_wc=H*(rho_a/rho_w)*1000;// mm
bhp=(rho_a*g*v*H);
printf('The required brake horsepower,bhp=%0.1f W\n',bhp);
```

Scilab code Exa 14.6 Preliminary Design of a Centrifugal Pump

```
//Example 14_6
clc;clear;
// Properties
//For refrigerant R-134a at T=20 C
v_f=0.0008157; // m^3/kg
rho=1/v_f; // kg/m^3
// Given values
r_1=0.100;
r_2=0.180;// The impeller inlet and outlet radii in m
b_1=0.050;
b_2=0.030;//The impeller inlet and outlet widths in m
v=0.25;// m^3/s
H=14.5; // Net head in m
n=1720;// rpm
g=9.81; // m/s^2

// Calculation
W_whp=rho*g*v*H;// The required water horsepower in W
// We assume 100 percent efficiency such that bhp is approximately equal to W_whp
bhp=W_whp/745.7;//The required brake horsepower in hp
printf('The required brake horsepower,bhp=%0.0f hp\n',bhp);
omega=(2*%pi*n)/60;
beta_1=atand(v/(2*%pi*b_1*omega*r_1^2));
printf('The blade inlet angle ,beta_1=%0.0f degree\n',beta_1);
V_2t=(g*H)/(omega*r_2);
```

```
V_2n=(v)/(2*%pi*r_2*b_2);
beta_2=atand(V_2n/((omega*r_2)-V_2t));
printf('The trailing edge blade angle ,beta_2=%0.0f degree\n',beta_2);
```

Scilab code Exa 14.7 `Calculation of Twist in an Airplane Propeller`

```
//Example 14_7
clc;clear;
// Given values
D_p=34.0;// The overall diameter of the propeller in cm
D_h=5.5;// The hub assembly diameter in cm
n=1700; // rpm
alpha=14; // The angle of attack in degree
V_wind=13.4;// m/s

// Calculation
r=D_h/(2*100);// Radius in m
omega=(2*%pi*n)/60;
phi=atand((V_wind/(omega*r)));
theta_1=alpha+phi;// Pitch angle at arbitrary radius r in degree
printf('The Pitch angle at arbitrary radius ,theta=%0.1f degree\n',theta_1);
r_1=D_p/(2*100);
phi_1=atand((V_wind/(omega*r_1)));
theta_2=alpha+phi_1;
printf('The Pitch angle at tip ,theta=%0.1f degree\n',theta_2);
```

Scilab code Exa 14.8 `Design of a Vane Axial Flow Fan for a Wind Tunnel`

```
// Example 14_8
clc;clear;funcprot(0);
//Given data
V_in=47.1; //Velocity at the inlet in m/s
beta_sl=0.0;// The leading edge angle of each stator blade in degree
beta_st=60;// The trailing edge angle in degrees
n=1750;//Speed of the impeller in rpm
r=0.40;//Radius in

// Calculation
V_st=(V_in/(cosd((beta_st))));// The velocity leaving the trailing edge of the stator in m/s
u_theta=(n*2*%pi*r)/(60);// The tangential velocity of the rotor blades in m/s
beta_rl=atand(((u_theta)+(V_in*tand(beta_st)))/(V_in));// The angle of the leading edge of the rotor in degree
beta_rt=atand((u_theta)/(V_in));// The angle of the trailing edge of the rotor in degree
printf('\nThe rotor blade at this radius has a leading edge anle of about %0.2f degree and a trailing edge angle of about %0.2f degree \nWe pick a number like 13, 15, or 17 rotor blades.',beta_rl,beta_rt);
```

Scilab code Exa 14.9 Using Pump Specific Speed for Preliminary Pump Design

```
//Example 14_9
clc;clear;funcprot(0);
// Given values
n=1170;// Rotation rate in rpm
H=23.5;// Net head in ft
v=320;// gpm
g=9.81;// m/s^2
```

```

// Calculation
N_spUS=(n*v^(1/2)/(H)^(3/4));
printf('The pump specific speed in customary U.S. units,N_sp,US=%0.1f \n',N_spUS);
omega=(2*%pi*n)/60;
// N_sp1=N_spUS*(N_sp/N_spUS) using conversion factor
N_sp1=N_spUS*(3.658*(10^-4));
printf('Normalized pump specific speed using the conversion factor =%0.3f \n',N_sp1);
// The answer vary due to round off error
```

Scilab code Exa 14.10 `The Effects of Doubling Pump Speed`

```
// Example 14_10
clc;clear;funcprot(0);
//Given data
omega_a=1;// Unit Speed

// Calculation
omega_b=2*omega_a;// Speed
bhp_ratio=(omega_b/omega_a)^3;
printf('\nThe power to the pump motor must be increased by a factor of %0.0f.',bhp_ratio);
```

Scilab code Exa 14.11 `Design of a New Geometrically Similar Pump`

```
//Example 14-11
clc;clear;
// Properties
rho_w=998;// kg/m^3
rho_R134=1226;// kg/m^3
```

```
// Given values
D_a=6;// Impeller diameter in cm
n=1725;// rpm
omega=180.6;// m^3/s
g=9.81// m/s^2
v_b=2400/10^6;// cm^3/s
H_b=450/100;// cm

// Calculation
v=[100 200 300 400 500 600 700];// cm^3/s
H=[180 185 175 170 150 95 54];// cm
n_pump=[32 54 70 79 81  66 38];// %
for(i=1:7)
    bhp(i)=((rho_w*g*v(i)*H(i))/(n_pump(i)/100))*(1/100)^4;// W
    C_Q(i)=((v(i))/(omega*D_a^3));// The capacity coefficient
    C_H(i)=((g*(H(i)/100))/(omega^2*(D_a/100)^2));// The head coefficient
    C_P(i)=((bhp(i))/(rho_w*omega^3*(D_a/100)^5));// The power coefficient
end
subplot(2,1,1);
plot(v,H,'r',v,n_pump,'b');
xlabel('Vdot,m^3/s');
ylabel('H,cm(or n,%)');
legend('H','n_pump')
a = gca();
a.y_location = "left";
a.filled = "on";
a.axes_visible = ["on","on","on"];
a.font_size = 1;
b = newaxes();
b.y_location = "right";
b.filled = "off";
b.axes_visible = ["off","on","on"];
b.axes_bounds = a.axes_bounds;
b.y_label.text = "bhp";
```

```
b.font_size = a.font_size;
plot(v,bhp,'g');
legend(['bhp'],"in_lower_right");
subplot(2,1,2);
xlabel('C_Q*100');
plot(C_Q*100,C_H*10,'b',C_Q*100,C_P*100,'g',C_Q*100,n_pump/100,'r');
legend('C_H*10','C_p*100','n_pump');
C_q=0.0112;
C_h=0.133;
C_p=0.00184;
n_pump=0.812;
D_b=(((v_b^2)*C_h)/(((C_q)^2)*g*H_b))^(1/4);// m
omega_b=(v_b)/((C_q*(D_b)^3));// rad/s
n=(omega_b*60)/(2*%pi);// rpm
bhp_b=C_p*rho_R134*omega_b^3*D_b^5;// W
printf('\nThe design diameter for pump B=%0.3f m \nThe design rotational speed for pump B=%0.0f rpm \nThe required brake horsepower for pump B=%0.0f W',D_b,n,bhp_b);
// The answer vary due to round off error
```

Scilab code Exa 14.12 Hydroturbine Design

```
//Example 14-12
clc;clear;
// Properties
rho_w=998;// kg/m^3
//Given values
r_2=2.50;// m
r_1=1.77;// m
b_2=0.914; // m
b_1=2.62; // m
n=120; // rpm
omega=12.57;// rad/s
```

```
alpha_2=33;// degree
v=599;// m^3/s
g=9.81;// m/s^2

//Calculation
//(a)
V_2n=(v/(2*%pi*r_2*b_2));//The normal component of velocity at the inlet in m/s
V_2t=V_2n*tand(alpha_2);//The tangential velocity component at the inlet in m/s
beta_2=atand(V_2n/((omega*r_2)-(V_2t)));
disp('(a) alpha=10 degree')
printf('The runner leading edge angle at runner inlet , beta_2=%0.1f degree\n',beta_2);
//Equations 1 through 3 are repeated for the runner outlet , with the following results:
V_1n=(v/(2*%pi*r_1*b_1));//
alpha_1=10;// degree
V_1t=V_1n*tand(alpha_1);
beta_1=atand(V_1n/((omega*r_1)-(V_1t)));
printf(' The runner blade trailing edge angle , beta_1=%0.1f degree\n',beta_1);
W_shaft=(rho_w*omega*v*((r_2*V_2t)-((r_1*V_1t))))/10^6;
W_shaft_hp=(W_shaft)*1341.02209;
printf(' The shaft output power,W_shaft =%0.2e hp\n',W_shaft_hp);
// Assume Efficiency of turbine=100%
// bhp=W_shaft
H_1=(W_shaft)*10^6/(rho_w*g*v);// m
printf(' The required net head,H =%0.1f m\n',H_1);

//
disp('(b) alpha=0 degree')
alpha_11=0;// degree
V_11t=V_1n*tand(alpha_11);
beta_11=atand(V_1n/((omega*r_1)-(V_11t)));// degree
printf(' The runner blade trailing edge angle ,
```

```
beta_1=%0.1f degree\n',beta_11);
W_shaft1=(rho_w*omega*v*((r_2*V_2t)-((r_1*V_11t))))/10^6;// MW
W_shaft1_hp=(W_shaft1)*1341.02209;// hp
printf(' The shaft output power,W_shaft =%0.2e hp\n',W_shaft1_hp);
H_2=(W_shaft1)*10^6/(rho_w*g*v);// m
printf(' The required net head,H =%0.1f m\n',H_2);

//
disp('(c) alpha=-10 degree')
alpha_12=-10;// degree
V_12t=V_1n*tand(alpha_12);
beta_12=atand(V_1n/((omega*r_1)-(V_12t)));
printf(' The runner blade trailing edge angle , beta_1=%0.1f degree\n',beta_12);
W_shaft12=(rho_w*omega*v*((r_2*V_2t)-((r_1*V_12t))))/10^6;// MW
W_shaft12_hp=(W_shaft12)*1341.02209;// hp
printf(' The shaft output power,W_shaft =%0.2e hp\n',W_shaft12_hp);
H_3=(W_shaft12)*10^6/(rho_w*g*v);// m
printf(' The required net head,H =%0.1f m\n',H_3);
alpha=[33 0 -10];
bhp=[W_shaft W_shaft1 W_shaft12];
H=[H_1 H_2 H_3];
plot(alpha,H,'r');
legend('H');
xlabel('alpha,degrees');
ylabel('H,m');
set(gca(),"data_bounds",matrix([-30,30,0,100],2,-1));
a = gca();
a.y_location = "left";
a.filled = "on";
a.axes_visible = ["on","on","on"];
a.font_size = 1;
b = newaxes();
```

```
b.y_location = "right";
b.filled = "off";
b.axes_visible = ["off","on","on"];
b.axes_bounds = a.axes_bounds;
b.y_label.text = "bhp,MW";
b.font_size = a.font_size
plot(alpha,bhp,'g');
legend(['bhp'],"in_upper_left");
set(gca(),"data_bounds",matrix([-30,30,0,700],2,-1));
```

Scilab code Exa 14.13 `Application of Turbine Affinity Laws`

```
//Example 14-13
clc;clear;
// Properties
rho=998;//The density of water at 20 C  in kg/m^3
// Given values
D_a=2.05;//Diameter in m
n_a=120;//rpm
n_b=120;//rpm
omega_a=12.57;//rad/s;
V_a=350;//m^3/s
H_a=75.0;//m
H_b=104;//m
bhp_a=242;//MW
rho_a=998;// kg/m^3
rho_b=998;// kg/m^3
g=9.81//m/s^2
n_ta=bhp_a*10^6/(rho_a*g*H_a*V_a);//Efficiency of turbine A

// Calculation
D_b=D_a*(sqrt(H_b/H_a))*(n_a/n_b);
printf('The diameter of the new turbine,D_b=%0.2f m\
```

```
    n',D_b);
V_b=V_a*(n_b/n_a)*(D_b/D_a)^3;
printf('Volume flow rate,V_b=%0.0f m^3/s\n',V_b);
bhp_b=bhp_a*(rho_b/rho_a)*(n_b/n_a)^3*(D_b/D_a)^5;
printf('The brake horsepower of new turbine, bhp_b=
    %0.0f MW\n',bhp_b);
n_tb=1-((1-n_ta)*(D_a/D_b)^(1/5));
printf('Efficiency of the turbine B=%0.3f \n',n_tb);
```

Scilab code Exa 14.14 Turbine Specific Speed

```
//Example 14-14
clc;clear;funcprot(0);
// Properties
rho=998;//The density of water at 20 C in kg/m^3
//Given values
D_a=2.05;//Diameter in m
n_a=120;//rpm
n_b=120;//rpm
omega_a=12.57;//rad/s
omega_b=12.57;//rad/s
V_a=350;//m^3/s
H_a=75.0;//m
H_b=104;//m
bhp_a=242*10^6;//MW
bhp_b=548*10^6
g=9.81;//The acceleration due to gravity in m/s^2

// Calculation
N_StA=(((omega_a*bhp_a^(1/2)))/((rho)^(1/2)*(g*H_a)
    ^(5/4)));
printf('The dimensionless turbine specific speed for
     turbine A,N_StA=%0.2f\n',N_StA);
N_StB=(((omega_b*bhp_b^(1/2)))/((rho)^(1/2)*(g*H_b)
    ^(5/4)));
```

```
printf('The dimensionless turbine specific speed for turbine B,N_StB=%0.2f\n',N_StB);
//N_(St,US,A)=N_(St,US,B)=N_stus
N_St=1.615;
N_stus=43.46*N_St;
printf('The turbine specific speed in customary U.S. units,N_(st,us)=%0.1f\n',N_stus);
```