Scilab Textbook Companion for
Textbook Of Heat Transfer
by S. P. Sukhatme[1]

Created by
Tushar Goel
B.Tech
Chemical Engineering
Indian Institute of Technology (BHU) , Varanasi
College Teacher
Dr. Prakash Kotecha
Cross-Checked by

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: Textbook Of Heat Transfer

Author: S. P. Sukhatme

Publisher: Universities Press

Edition: 4

Year: 2005

ISBN: 9788173715440

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

Scilab code Exa 1.1 `Viscosity in SI system`

```
clear;
clc;
// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 1
// Introduction


// Example 1.1
// Page 5
// Given that the viscosity of water at 100 degree
   Celsius is 28.8 * 10^-6 kgf s/m^2 in MKS system ,
    express this value in SI system.
printf("Example 1.1, Page 5 \n \n")

// Solution:

//at 100 degree Celsius
v1=28.8 * 10^-6; // [kgf s/m^2]
v2=28.8 * 10^-6 * 9.8; // [N s/m^2]
printf("Viscosity of water at 100 degree celsius in
   the SI system is %e N.s/m^-2 (or kg/m s)",v2)
```

Scilab code Exa 1.2 `Useful heat gain and thermal efficiency`

```
clear;
clc;
// Textbook of Heat Transfer(4th Edition)) , S P Sukhatme
// Chapter 1 - Introduction

//Example 1.2
// Page 14
printf("Example 1.2, Page 14 \n \n")
//Solution:
i=950; // radiation flux [W/m^2]
A=1.5; // area [m^2]
T_i=61; // inlet temperature
T_o=69; // outlet temperature
mdot=1.5; // [kg/min] , mass flow rate
Mdot=1.5/60; // [kg/sec]
Q_conductn=50; //[W]
t=0.95; // transmissivity
a=0.97;// absoptivity
// from appendix table A.1 at 65 degree C
C_p= 4183 ; // [J/kg K]
// Using Equation 1.4.15 , assuming that the flow through the tubes is steady and one dimensional.
// in this case (dW/dt)_shaft = 0
// assuming (dW/dt)_shear is negligible
// eqn(1.4.15) reduces to
q=Mdot*C_p*(T_o-T_i);

// let 'n' be thermal efficiency
n=q/(i*A);
n_percent=n*100;

```

```

// equation 1.4.13 yields dQ/dt = 0
Q_re_radiated=(i*A*t*a)-Q_conductn-q; // [W]


printf("Useful heat gain rate is %f W \n",q);
printf("Thermal efficiency is %e i.e. %f per cent \n",n,n_percent);
printf("The rate at which energy is lost by re-radiation and convection is %f W",Q_re_radiated)
```

Scilab code Exa 1.3 Exit velocity and Temperature

```
clear;
clc;
// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 1
// Introduction


//Example 1.3
// Page 16
printf("Example 1.3, Page 16\n\n");

//Solution:
// Given
v_i=10; // [m/s]
q=1000; // [W]
d_i=0.04; // [m]
d_o=0.06; // [m]

// From appendix table A.2
rho1=0.946; // [kg/m^3] at 100 degree C
C_p=1009; // [J/kg K]

```

```
mdot=rho1*(%pi/4)*(d_i^2)*v_i; // [kg/s]


// In this case (dW/dt)_shaft=0 and (z_o - z_i)=0
// From eqn 1.4.15 , q=mdot*(h_o-h_i)
// Let dh = (h_o-h_i)
dh=q/mdot; // [J/kg]
// Let T_o be the outlet temperature
T_o=dh/C_p+100;

rho2=0.773; // [kg/m^3] at T_o = 183.4 degree C
// From eqn 1.4.6
v_o=mdot/(rho2*(%pi/4)*(d_o)^2); // [m/s]

dKE_kg=(v_o^2-v_i^2)/2; // [J/kg]


printf("Exit Temperature is %f degree C \n",T_o);
printf("Exit velocity is %f m/s \n",v_o);
printf("Change in Kinetic Energy per kg = %f J/kg",
    dKE_kg);
```

Chapter 2

Heat Conduction in Solids

Scilab code Exa 2.1 `Heat flow rate`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.1
// Page 27
printf("Example 2.1, Page 27 \n\n")

d_i=0.02; // [m] inner radius
d_o=0.04; // [m] outer radius
r_i=d_i/2; // [m] inner radius
r_o=d_o/2; // [m] outer radius
k=0.58; // [w/m K] thermal conductivity of tube
   material
t_i=70; //[degree C]
t_o=100; // [degree C]
l=1; // [m] per unit length
// using equation 2.1.5
```

```
q=l*2*(%pi)*k*(t_i-t_o)/log(r_o/r_i);
printf("Heat flow per unit length is %f W/m",q);
```

Scilab code Exa 2.2 `Heat flow rate`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.2
// Page 31
printf("Example 2.2, Page 31 \n\n")

d_i=0.02; // [m] inner radius
d_o=0.04; // [m] outer radius
r_i=d_i/2; // [m] inner radius
r_o=d_o/2; // [m] outer radius
k=0.58; // [w/m K] thermal conductivity of tube material
t_i=70; //[degree C]
t_o=100; // [degree C]
l=1; // [m] per unit length

// thermal resistance of tube per unit length
R_th_tube=(log(r_o/r_i))/(2*%pi*k*l); // [K/W]

//from table 1.3 , heat transfer co-efficient for condensing steam may be taken as
h=5000; // [W/m^2 K]
// thermal resistance of condensing steam per unit length
R_th_cond=1/(%pi*d_o*l*h);
```

```

// since R_th_tube is much less than R_th_cond , we can assume outer surface to be at 100 degree C
//hence heat flow rate per unit meter is
q=l*2*(%pi)*k*(t_i-100)/log(r_o/r_i);

printf("Thermal resistance of tube per unit length is %f K/W\n",R_th_tube);
printf("Thermal resistance of condensing steam per unit length is %f K/W\n",R_th_cond);
printf("Heat flow per unit length is %f W/m",q);
```

Scilab code Exa 2.3 `Engineers decision`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.3
// Page 31
printf("Example 2.3, Page 31 \n\n")

h_w=140; // heat transfer coefficient on water side , [W/m^2 K]
h_o=150; // heat transfer coefficient on oil side , [W/m^2 K]
k=30; // thermal conductivity [W/m K]
r_o=0.01; // inner diameter of GI pipe on inside
r_i=0.008; // outer diameter GI pipe on inside
l=1; // [m] , per unit length

// Thermal resistance of inner GI pipe
```

```
R_inner_GI=log((r_o/r_i))/(2*%pi*k*l);


// Thermal resistance on the oil side per unit
   length
R_oilside=1/(h_o*%pi*2*r_i*l);


// Thermal resistance on cold water side per unit
   length
R_waterside=1/(h_w*%pi*2*r_o*l);


// we see thermal resistance of inner GI pipe
   contributes less than 0.5 percent to the total
   resistance


printf("Thermal resistance of inner GI pipe = %f K/W
    \n",R_inner_GI);
printf("Thermal resistance on the oil side per unit
   length = %f K/W \n",R_oilside);
printf("Thermal resistance on cold water side per
   unit length = %f K/W \n",R_waterside);
printf("So, Engineer in-charge has made a bad
   decision");
```

Scilab code Exa 2.4 `Thickness of insulation`

```
clear all;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids
```

```

// Example 2.4
// Page 32
printf(”Example 2.4, Page 32 \n\n”)

Ti = 300;           //Internal temp of hot gas in degree Celsius
OD = 0.1;           //Outer diameter of long metal pipe in meters
ID = 0.04;          //Internal diamtere of long metal pipe in meters
ki = 0.052;         //thermal conductivity of mineral wood in W/mK
To = 50;            //Outer surface temperature in degree celsius
hi = 29;            //heat transfer coefficient in the inner side in W/m^2 K
ho = 12;            //heat transfer coefficient in the outer pipe W/m^2 K

//Determination of thickness of insulation
function[f] = thickness(r)
    f = r*(10.344 + 271.15*log(r*(0.05)^-1))-11.75
    funcprot(0);
endfunction
r = 0.082;
while 1
    rnew = r - thickness(r)/diff(thickness(r));
    if rnew == r then
        r3 = rnew;
        break;
    end
    r = rnew;
end
t = r3 - OD/2;
printf(”\n Thickness of insulation = %f cm”,t*100);
//Heat loss per unit length
q = 600*(22/7)*r3;
```

```
printf("\n Heat loss per unit length = %.1f W/m",q);
```

Scilab code Exa 2.5 `Heat loss rate`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.5
// Page 34
printf("Example 2.5, Page 34 \n\n")

Ti = 90;                                //Temp on inner side in degree celsius
To = 30;                                //Temp on outer side in degree celsius
hi = 500;                               //heat transfer coeffcient in W/m^2 K
ho = 10;                                //heat transfer coeffcient in W/m^2 K
ID = 0.016;                             //Internal diameter in meters
t = [0 0.5 1 2 3 4 5];                 //Insulation thickness in cm
OD = 0.02;                             //Outer diameter in meters
r3 = OD/2 + t/100;                     //radius after insulation in meters

i=1;
printf("\n Insulation thickness(cm)        r3(m)    heat loss rate per meter(W/m)");
```

```
while i<=7
    ql(i) = [2*(%pi)*(ID/2)*(Ti-To)]/[(1/hi)+(0.008/0.2)*log(r3(i)/0.01) + (0.008/r3(i))*(1/ho)];
printf("\n     %.1f                         %.3f               %.1f",t(i),r3(i),ql(i));
    i = i+1;
end
plot(t,ql);
xtitle("","Insulation thickness(cm)","Heat loss rate per unit length,W/m");
printf("\n The maxima in the curve is at r_3 = 0.02 m");
```

Scilab code Exa 2.6 Critical radius

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.6
// Page 36
printf("Example 2.6, Page 34 \n\n")

h_natural = 10;//heat transfer coefficient for natural convection in W/m^2 K
h_forced = 50; //heat transfer coefficient for forced convection in W/m^2 K
//for asbestos
k1 = 0.2;      //thermal conductivity in W/m K
//for mineral wool
k2 = 0.05;     //thermal conductivity in W/m K
```

```
printf(”\n critical radius of insulation in cm”);
printf(”\n                              h = 10                    h = 50”);
printf(”\n Asbestos                     %.1f                 %.1f”,k1*100/h_natural,k1*100/h_forced);
printf(”\n Mineral wool                 %.1f                 %.1f”,k2*100/h_natural,k2*100/h_forced);
```

Scilab code Exa 2.7 `Maximum temperature`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.7
// Page 43
printf(”Example 2.7, Page 43 \n\n”)

H = 5 ; // Height, [m]
L = 10 ; // Length, [m]
t = 1 ; // thickness, [m]
b = t/2;
k = 1.05 ; // [W/m K]
q = 58 ; // [W/m^3]
T = 35 ; // [C]
h = 11.6 ; // Heat transfer coefficient, [W/m^2 K]
// Substituting the values in equation 2.5.6
T_max = T + q*b*(b/(2*k)+1/h);
printf(”Maximum Temperature = %f degree C”,T_max);
```

Scilab code Exa 2.8 `Steady state temperature`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.8
// Page 47
printf("Example 2.8, Page 47 \n\n")

// The bar will have two dimensional variation in temperature
// the differential equation is subject to boundary conditions
x1 = 0; // [cm]
Tx1 = 30; // [C]
x2 = 5; // [cm]
Tx2 = 30; // [C]
y1 = 0; // [cm]
Ty1 = 30; // [C]
y2 = 10; // [cm]
Ty2 = 130; // [C]
// substituting theta = T-30 and using eqn 2.6.11
// putting x = 2.5cm and y = 5cm in infinite summation series


n = 1;
x1 = (1-cos(%pi*n))/(sinh(2*%pi*n))*sin(n^%pi/2)*sinh(n*%pi);

```

```
n = 3;
x3 = (1-cos(%pi*n))/(sinh(2*%pi*n))*sin(n^%pi/2)*sinh(n*%pi);

n = 5;
x5 = (1-cos(%pi*n))/(sinh(2*%pi*n))*sin(n^%pi/2)*sinh(n*%pi);

x = x1+x3+x5;

T = x*100+30;
printf("Steady state temperature = %f C",T);
```

Scilab code Exa 2.9 `Time taken by the rod to heat up`

```
clear all;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.9
// Page 51
printf("Example 2.9, Page 51 \n\n")

k = 330;             //thermal conductivity in W/m K
a = 95*10^(-6);      //thermal expansion coefficient
R = 0.01;            //radius in meters
To = 77;            //temperature in kelvins
Tf = 273+50;        //temperature in kelvins
theta1 = To - Tf;
T = 273+10;         //temperature in kelvins
theta = T - Tf;
h = 20;             //heat transfer coefficient in W
```

```
/m^2 K
printf("\n Theta1 = %d K",theta1);
printf("\n Theta = %d K ",theta);
printf("\n v/A = %.3f m",R/2);
printf("\n k/a = %.4f*10^(6) J/m^3 K",(k/a)*10^(-6));

time = (k/a)*(R/2)/h*log(theta1/theta);

printf("\n Time taken by the rod to heat up = %.1f secs",time);
Bi = h*R/k;
printf("\n Biot number Bi = %.2f*10^(-4) ",Bi*10^4);
printf("\n Since Biot number is much less than 0.1, therefore assumption  that internal temperature gradients are negligible is a good one");
```

Scilab code Exa 2.10.i Heat transfer coefficient at the centre

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.10(i)
// Page 58
printf("Example 2.10(i), Page 58 \n\n")

// Centre of the slab
// Given data
b = 0.005 ; // [m]
t = 5*60; // time, [sec]
Th = 200 ; // [C]
```

```
Tw = 20 ; // [C]
h = 150 ; // [W/m^2 K]
rho = 2200 ; //[kg/m^3]
Cp = 1050 ; // [J/kg K]
k = 0.4 ; // [W/m K]
// Using charts in fig 2.18 and 2.19 and eqn 2.7.19
   and 2.7.20

theta = Th - Tw;
Biot_no = h*b/k;
a = k/(rho*Cp); // alpha
Fourier_no = a*t/b^2;

// From fig 2.18, ratio = theta_x_b0/theta_o
ratio_b0 = 0.12;
// From fig 2.18, ratio = theta_x_b1/theta_o
ratio_b1 = 0.48;

// Therefore
theta_x_b0 = theta*ratio_b0; // [C]
T_x_b0 = theta_x_b0 + Tw ; // [C]
theta_x_b1 = theta*ratio_b1; // [C]
T_x_b1 = theta_x_b1 + Tw ; // [C]

// From Table 2.2 for Bi = 1.875
lambda_1_b = 1.0498;
x = 2*sin(lambda_1_b)/[lambda_1_b+(sin(lambda_1_b))
   *(cos(lambda_1_b))];

// From eqn 2.7.20
theta_x_b0 = theta*x*(exp((-lambda_1_b^2)*Fourier_no
   ));
T_x_b0 = theta_x_b0 + Tw;
printf("Temperature at b=0 is %f degree C\n",T_x_b0)
   ;
```

Scilab code Exa 2.10.ii `heat transfer coefficient at the surface`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.10(ii)
// Page 58
printf("Example 2.10(ii), Page 58 \n\n")

// (ii) Surface of the slab

b = 0.005 ; // [m]
t = 5*60; // time, [sec]
Th = 200 ; // [C]
Tw = 20 ; // [C]
h = 150 ; // [W/m^2 K]
rho = 2200 ; //[kg/m^3]
Cp = 1050 ; // [J/kg K]
k = 0.4 ; // [W/m K]
// Using charts in fig 2.18 and 2.19 and eqn 2.7.19
   and 2.7.20
theta = Th - Tw;
Biot_no = h*b/k;
a = k/(rho*Cp); // alpha
Fourier_no = a*t/b^2;

// From fig 2.18, ratio = theta_x_b0/theta_o
ratio_b0 = 0.12;
// From fig 2.18, ratio = theta_x_b1/theta_o
ratio_b1 = 0.48;
```

```

// Therefore
theta_x_b0 = theta*ratio_b0; // [C]
T_x_b0 = theta_x_b0 + Tw ; // [C]
theta_x_b1 = theta*ratio_b1; // [C]
T_x_b1 = theta_x_b1 + Tw ; // [C]

// From Table 2.2 for Bi = 1.875
lambda_1_b = 1.0498;
x = 2*sin(lambda_1_b)/[lambda_1_b+(sin(lambda_1_b))*(cos(lambda_1_b))];

// From 2.7.19
theta_x_b1 = theta_x_b0*(cos(lambda_1_b*1));
T_x_b1 = theta_x_b1 + Tw;
printf("Temperature at b=1 is %f degree C\n",T_x_b1);
```

Scilab code Exa 2.11.a `Time taken by the centre of ball`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.11(a)
// Page 65
printf("Example 2.11(a), Page 65 \n\n")

D = 0.05 ; // [m]
To = 450 ; // [degree C]
Tf = 90 ; // [degree C]
T = 150 ; // [degree c]
```

```
h = 115 ; // [W/m^2 K]
rho = 8000 ; // [kg/m^3]
Cp = 0.42*1000 ; // [J/kg K]
k = 46 ; // [W/m K]
R = D/2;

// (a)
// From eqn 2.7.3 for a sphere
t1 = rho*Cp*R/(3*h)*log((To-Tf)/(T-Tf)); // [sec]
t1_min = t1/60 ; // [min]
printf("Time taken by the centre of the ball to
   reach 150 degree C if internal gradients are
   neglected is %f seconds i.e. %f minutes \n",t1,
   t1_min);
```

Scilab code Exa 2.11.b `time taken by the centre of ball to reach temperature`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.11(b)
// Page 65
printf("Example 2.11(b), Page 65 \n\n")

D = 0.05 ; // [m]
To = 450 ; // [degree C]
Tf = 90 ; // [degree C]
T = 150 ; // [degree c]
h = 115 ; // [W/m^2 K]
rho = 8000 ; // [kg/m^3]
Cp = 0.42*1000 ; // [J/kg K]
```

```
k = 46 ; // [W/m K]
R = D/2;

// (b)
// let ratio = theta_R_0/theta_o
ratio = (T-Tf)/(To - Tf);
Bi = h*R/k;
// From Table 2.5
lambda_1_R = 0.430;
x = 2*[sin(lambda_1_R) - lambda_1_R*cos(lambda_1_R)]/[lambda_1_R - sin(lambda_1_R)*cos(lambda_1_R)];

// Substituting in equattion 2.7.29, we have an equation in variable y(=at/R^2)
// Solving
function[eqn] = parameter(y)
eqn = ratio - x*exp(-(lambda_1_R^2)*(y));
funcprot(0);
endfunction

y = 5; // (initial guess, assumed value for fsolve function)
Y = fsolve(y,parameter);

a = k/(Cp*rho); // alpha
t2 = Y*(R^2)/(a); // [sec]
t2_min = t2/60; // [min]
printf("Time taken by the centre of the ball to reach 150 degree C if internal temperature gradients are not neglected is %f seconds i.e. %f minutes",t2,t2_min);
```

Scilab code Exa 2.12 Temperature at the centre of the brick

```
clear ;
```

```
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.12
// Page 67
printf(”Example 2.12, Page 67 \n\n”)

a = 0.12 ; // [m]

T = 400 ; // [C]
To = 25 ; //[C]
t = 100/60 ; // [hour]
h = 10 ; // [W/m^2 K]
k = 1.0 ; // [W/m K]
alpha = 3.33*10^-3 ; // [m^2/h]
// using fig 2.18 and eqn 2.7.20

x1 = h*a/k ;
x2 = k/(h*a);
x3 = alpha*t/a^2;

// Let ratio_x = theta/theta_o for x direction, from
    fig 2.18
ratio_x = 0.82 ;

// Similarly, for y direction
ratio_y = 0.41;

// Similarly, for z direction
ratio_z = 0.30;

// Therefore
total_ratio = ratio_x*ratio_y*ratio_z ;

T_centre = To + total_ratio*(T-To) ; // [degree C]
```

```
printf("Temperature at the centre of the brick = %f degree C \n\n",T_centre);

// Alternatively
printf("Alternatively, obtaining Biot number and values of lambda_1_b and using eqn 2.7.20, we get \n")

ratio_x = 1.1310*exp(-(0.9036^2)*0.385);
ratio_y = 1.0701*exp(-(0.6533^2)*2.220);
ratio_z = 1.0580*exp(-(0.5932^2)*3.469);
ratio = ratio_x*ratio_y*ratio_z;

T_centre = To + total_ratio*(T-To) ; // [degree C]
printf("Temperature at the centre of the brick = %f degree C \n",T_centre);
```

Scilab code Exa 2.13.a `Temperature at the copper fin tip`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.13(a)
// Page 73
printf("Example 2.13(a), Page 73 \n\n")

D = 0.003 ; // [m]
L = 0.03 ; // [m]
h = 10 ; // [W/m^2]
Tf = 20 ; // [C]
T1 = 120 ; // [C]
```

```

// (a) Copper fin
k = 350 ; // [W/m K]

// For a circular cross section
m = [4*h/(k*D)]^(1/2);
mL = m*0.03 ;
// T at x = L
T = Tf + (T1-Tf)/cosh(m*L);
printf("mL = %f \n",mL);
printf("Temperature at the tip of fin made of copper
    is %f degree C \n",T);
```

Scilab code Exa 2.13.b `Temperature at the steel fin tip`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.13(b)
// Page 73
printf("Example 2.13(b), Page 73 \n\n")

D = 0.003 ; // [m]
L = 0.03 ; // [m]
h = 10 ; // [W/m^2]
Tf = 20 ; // [C]
T1 = 120 ; // [C]


// (b) Stainless steel fin
k = 15 ; // [W/m K]
```

```

// For a circular cross section
m = [4*h/(k*D)]^(1/2);
mL = m*0.03 ;
// T at x = L
T = Tf + (T1-Tf)/cosh(m*L);
printf("mL = %f \n",mL);
printf("Temperature at the tip of fin made of steel
    is %f degree C \n",T);
```

Scilab code Exa 2.13.c `Temperature at the teflon fin tip`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.13(c)
// Page 73
printf("Example 2.13(c), Page 73 \n\n")

D = 0.003 ; // [m]
L = 0.03 ; // [m]
h = 10 ; // [W/m^2]
Tf = 20 ; // [C]
T1 = 120 ; // [C]

// (c) Teflon fin
k = 0.35 ; // [W/m K]

// For a circular cross section
m = [4*h/(k*D)]^(1/2);
mL = m*0.03 ;
```

```
// T at x = L
T = Tf + (T1-Tf)/cosh(m*L);
printf(”mL = %f \n”,mL);
printf(”Temperature at the tip of fin made of teflon is %f degree C \n”,T);
```

Scilab code Exa 2.14 `Heat loss rate`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.14
// Page 74
printf(”Example 2.14, Page 74 \n\n”)

L = 0.02 ; // [m]
t = 0.002 ; // [m]
b = 0.2 ; // [m]
theta1 = 200 ; // [C]
h = 15 ; // [W/m^2 K]
k = 45 ; // [W/m K]

Bi = h*(t/2)/k ;

// We have
P = 2*(b+t); // [m]
A = b*t ; // [m^2]
// Therefore
mL = ([(h*P)/(A*k)]^(1/2))*L;

// From equation 2.8.6, fin effectiveness n
```

```
n = tanh(mL)/mL;
printf("Fin Effectiveness = %f \n",n);

q_loss = n*h*40.4*2*10^-4*200; // [W]
printf("Heat loss rate from fin surface = %f W", q_loss);
```

Scilab code Exa 2.15 Decrease in thermal resistance

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.1
// Page 74
printf("Example 2.15, Page 74 \n\n")


// Find Decrease in thermal Resistance
// Find Increase in heat transfer rate

h = 15 ;         //[W/m^2.K]
k = 300;         //[W/m.K]
T = 200;          //[C]
Tsurr = 30;       //[C]
d = .01;         //[m]
L = .1;           //[m]
A = .5*.5         //[m^2]
n = 100           //Number of Pins

Bi = h*d/2/k;          //Biot Number
//Value of Biot Number is much less than .1
```

```
//Thus using equation 2.8.6
mL = (h*4/k/d)^.5*L;
zi = tanh(mL)/mL;
Res1 = 1/h/A;         // Thermal resistance without fins, [K/W]
Res2 = 1/(h*(A - n*%pi/4*d^2 + zi*(n*%pi*d*L)));// Thermal resistance with fins,[K/W]

delRes = Res1-Res2;                          //[K/W]
// Increase in heat transfer rate
q = (T-Tsurr)/Res2 - (T-Tsurr)/Res1;         //[W]

printf("\n\n Decrease in thermal resistane at surface %.4f K/W.\n Increase in heat transfer rate %.1f W",delRes,q)
//END
```

Scilab code Exa 2.16 `Overall heat transfer coefficient`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 2
// Heat Conduction in Solids

// Example 2.16
// Page 75
printf("Example 2.16, Page 75 \n\n")

//Theoretical Problem

printf('\n\n This is a Theoretical Problem, does not involve any mathematical computation.');
//END
```

Chapter 3

Thermal Radiation

Scilab code Exa 3.1 `Monochromatic emissive power`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.1
// Page 114
printf("Example 3.1, Page 114 \n\n");

T = 5779 ; // [Temperature,in Kelvin]
// From Wein's law, eqn 3.2.8
lambda_m = 0.00290/T ; // [m]
// Substituting this value in plank's law, we get
e = 2*(%pi)*0.596*(10^-16)/(((0.5018*10^-6)^5)*(exp
    (0.014387/0.00290)-1)) ; //[W/m^2 m]

e_bl_max= e / 10^6 ;

printf("Value of emissivity on sun surface is %f W/m
```

```
^2 um \n",e_bl_max); //[W/m^2 um]

e_earth = e_bl_max*((0.695*10^6)/(1.496*10^8))^2 ;

printf("The value of emmissivity on earths surface is %f W/m^2 um", e_earth)
```

Scilab code Exa 3.2 `Heat flux`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.2
// Page 115
printf("Example 3.2, Page 115 \n\n")

//Heat emission
Stefan_constt = 5.67*10^(-8);     //(W/m^2.K^4)
T = 1500;                          //temperature is in kelvins
eb = (Stefan_constt)*(T^(4));            //energy radiated by blackbody
//emission in 0.3um to 1um
e = 0.9;                          //emissivity
lamda1 = 1;                       //wavelength is in um
lamda2 = 0.3;                     //wavelength is in um
D0_1=0.5*(0.01972+0.00779);        //From table 3.1 page- 114
D0_2=0;                           //From table 3.1 page - 114
q = e*(D0_1-D0_2)*Stefan_constt*T^(4);//in W/m^2
```

```
printf("\n wavelength*temp = %d um K",1*1500);
printf("\n wavelength*temp at 0.3um = %d um K",0.3*1500);
printf("\n\n Required heat flux, q = %d W/m^2",q);
```

Scilab code Exa 3.3 Absorbed radiant flux and absorptivity and reflectivity

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.3
// Page 119
printf("Example 3.3, Page 119 \n\n")


a0_2=1; //absorptivity
a2_4=1; //absorptivity
a4_6=0.5; //absorptivity
a6_8=0.5; //absorptivity
a8_=0; //absorptivity
H0_2=0; //Irradiation in W/m^2 um
H2_4=750; //Irradiation in W/m^2 um
H4_6=750; //Irradiation in W/m^2 um
H6_8=750; //Irradiation in W/m^2 um
H8_=750; //Irradiation in W/m^2 um
Absorbed_radiant_flux=1*0*(2-0)+1*750*(4-2)+0.5*750*(8-4)+0;
H = 750*(8-2);      //Incident flux
a = Absorbed_radiant_flux/H;
p = 1-a;            //Since the surface is opaque
printf("\n Absorbed radiant flux = %d W/m^2",
```

```
Absorbed_radiant_flux);
printf("\n Incident flux = %d W/m^2",H);
printf("\n Absorptivity = %.3f",a);
printf("\n Since the surface is opaque reflectivity = %.3f",p);
```

Scilab code Exa 3.4.a `Total intensity in normal direction`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.4(a)
// Page 123
printf("Example 3.4(a), Page 123 \n\n")


e = 0.08; //emissivity
T = 800; //temperature, [K]

Stefan_constt = 5.67*10^(-8);     //[W/m^2.K^4]
// From Stefan Boltzmann law, equation 3.2.10
q = e*Stefan_constt*T^4;          //[W/m^2]
printf("\n Energy emitted = %.1f W/m^2",q);

// (a)
// Therefore
in = (q/(%pi));
printf("\n Energy emitted normal to the surface = %.1f W/m^2 sr",in);
```

Scilab code Exa 3.4.b `Ratio of radiant flux to the emissive power`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.4(b)
// Page 123
printf("Example 3.4(b), Page 123 \n\n")


e = 0.08; //emissivity
T = 800; //temperature, [K]

Stefan_constt = 5.67*10^(-8);     //[W/m^2.K^4]
// From Stefan Boltzmann law, equation 3.2.10
q = e*Stefan_constt*T^4;          //[W/m^2]
in = (q/(%pi));

// (b)
// Radiant flux emitted in the cone 0 <= pzi <= 50
   degree, 0 <= theta <= 2*pi
q_cone=2*(%pi)*in*(-cos(100*(%pi/180))+cos(0))/4;

printf("\n Radiant flux emitted in the cone =%.1f W/
  m^2",q_cone);

Ratio = q_cone/q;
printf("\n Ratio = %.3f",Ratio);
```

Scilab code Exa 3.5 `Rate of incident radiation`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.5
// Page 124
printf("Example 3.5, Page 124 \n\n")

l1 = 0.5 ; // wavelength, [um]
l2 = 1.5 ; // wavelength, [um]
l3 = 2.5 ; // wavelength, [um]
l4 = 3.5 ; // wavelength, [um]
H1 = 2500 ; // [W/m^2 um]
H2 = 4000 ; // [W/m^2 um]
H3 = 2500 ; // [W/m^2 um]

// Since the irridiation is diffuse, the spectral
   intensity is given by eqn 3.4.14 and 3.4.8
// Integrating i_lambda over the directions of the
   specified solid angle and using fig 3.12


flux = 3/4*[H1*(l2-l1)+H2*(l3-l2)+H3*(l4-l3)];
printf("Rate at which radiation is incident on the
   surface = %f W/m^2",flux);
```

Scilab code Exa 3.6 `Shape factor F12`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.6
// Page 132
printf("Example 3.6, Page 132 \n\n")

// This is a theoretical problem with no numerical data
printf("This is a theoretical problem with no numerical data \n");

// Considering an elementary ring dA2 of width dr at an arbitary radius r, we have
// r = h*tanB1
// dA2 = 2*%pi*r*dr
// dA2 = 2*%pi*(h^2)*tan(B1)*sec^2(B1)*dB1
// B2 = B1, since surfaces ate parallel, and
// L = h/cos(B1)
// Substituting in eqn 3.6.7
// F12 = sin^2(a)


printf("Considering an elementary ring dA2 of width dr at an arbitary radius r, we have \n");
printf("r = h*tanB1 \n");
printf("dA2 = 2*pi*r*dr \n");
printf("dA2 = 2*pi*(h^2)*tan(B1)*sec^2(B1)*dB1 \n");
printf("B2 = B1, since surfaces ate parallel, and \n");
printf("L = h/cos(B1) \n");
printf("Substituting in eqn 3.6.7 \n");
printf("F12 = sin^2(a) \n");
```

Scilab code Exa 3.7 `Shape factor`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.7
// Page 134
printf("Example 3.7, Page 134 \n\n")

// This is a theoretical problem with no numerical data
printf("This is a theoretical problem with no numerical data \n");


// Considering an elementary circular ring on the surface of the sphere's surface at any arbitary anglr B,
// we have B1 = B, B2 = 0, L = R and dA_2 = 2*%pi*(R^2)*(sin(B))dB
// Therefore, from equation 3.6.7
// F12 = sin^2(a)

printf("Considering an elementary circular ring on the surface of the sphere surface at any arbitary anglr B \n");
printf("we have B1 = B, B2 = 0, L = R and dA_2 = 2*pi*(R^2)*(sin(B))dB \n");
printf("Therefore, from equation 3.6.7 \n");
printf("F12 = sin^2(a)");
```

Scilab code Exa 3.8 Shape factor F12

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.8
// Page 135
printf("Example 3.8, Page 135 \n\n")

// From eqn 3.7.5 or fig 3.19
F65 = 0.22;
F64 = 0.16;
F35 = 0.32;
F34 = 0.27;
A1 = 3; // [m^2]
A3 = 3; // [m^2]
A6 = 6; // [m^2]

// Using additive and reciprocal relations
// We have F12 = F16 - F13

F61 = F65 - F64 ;
F31 = F35 - F34 ;

F16 = A6/A1*F61 ;
F13 = A3/A1*F31 ;

F12 = F16 - F13;

printf("F_1-2 = %f",F12);
```

Scilab code Exa 3.9 `Shape factor`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.9
// Page 136
printf(”Example 3.9, Page 136 \n\n”)

// This is a theoretical problem, does not involve any numerical computation
printf(”This is a theoretical problem, does not involve any numerical computation \n”);
// Denoting area of conical surface by A1
// Considering an imaginary flat surface A2 closing the conical cavity

F22 = 0 ; // Flat surface

// from eqn 3.7.2 , we have F11 + F12 = 1 and F22 + F21 = 1
F21 = 1 - F22 ;

// F12 = A2/A1∗F21 ;
// F11 = 1 − F12 ;
// F11 = 1 − sin(a)
```

Scilab code Exa 3.10 `Net radiative heat transfer`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.10
// Page 138
printf("Example 3.10, Page 138 \n\n")

sigma = 5.670*10^-8 ;
T1 = 473 ; // [K]
T2 = 373 ; // [K]
A1 = 1*2 ; // area, [m^2]
X = 0.25;
Y = 0.5 ;
// From eqn 3.7.4
F12 = (2/(%pi*X*Y))*[log((((1+X^2)*(1+Y^2))/(1+X^2+Y
    ^2))^(1/2)) + Y*((1+X^2)^(1/2))*atan(Y/((1+X^2)
    ^(1/2))) + X*((1+Y^2)^(1/2))*atan(X/((1+Y^2)
    ^(1/2))) - Y*atan(Y) - X*atan(X) ];


q1 = sigma*A1*(T1^4-T2^4)*[(1-F12^2)/(2-2*F12)];

printf("Net radiative heat transfer from the surface
     = %f W \n",q1);
```

Scilab code Exa 3.11 `steady state heat flux`

```
clear all;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
```

```
// Chapter 3
// Thermal Radiation

// Example 3.11
// Page 141
printf("Example 3.11, Page 141 \n\n")

// All modes of heat transfer are involved
// let steady state heat flux flowing through the composite slab be (q/a)
h1 = 20;      //[W/m^2 K]
w1 = 0.2;     //[m]
k1 = 1;       //[W/m K]
e1 = 0.5;     //emmisivity at surfce 1
e2 = 0.4;     //emmisivity at surfce 2
w2 = 0.3;     //[m]
k2 = 0.5;     //[W/m K]
h2 = 10;      //[W/m^2 K]
T1 = 473;     //[Kelvin]
T2 = 273+40;  //[Kelvin]
stefan_cnst = 5.67e-08;    //[W/m^2 K^4]

// For resistances 1 and 2
function[f]=temperature(T)
    f(1) = (T1-T(1))/(1/h1 + w1/k1) - (T(2) - T2)/(w2/k2 + 1/h2);
    f(2) = stefan_cnst*(T(1)^4 - T(2)^4)/(1/e1 + 1/e2 -1) - (T(2) - T2)/(w2/k2 + 1/h2);
    funcprot(0);
endfunction

T = [10 10]; // assumed initial values for fsolve function
y = fsolve(T,temperature);

printf("\n Steady state heat flux q/A = %.1f W/m^2",(T1-y(1))/(1/h1 + w1/k1));
```

Scilab code Exa 3.12 `Rate of heat loss`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.12
// Page 145
printf("Example 3.12, Page 145 \n\n")

D = 0.02 ; // [m]
T1 = 1000+273 ; // [K]
T2 = 27+273 ; // [K]
s = 5.670*10^-8 ; // stefans constant
// Assuming the opening is closed by an imaginary surface at temperature T1
// Using equation 3.10.3 , we get
q = s*1*%pi*((D/2)^2)*(T1^4-T2^4); // [W]

printf("Rate at which heat is lost by radiation = %f W",q);
```

Scilab code Exa 3.13 `Rate of nitrogen evaporation`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
```

```
// Thermal Radiation

// Example 3.13
// Page 146
printf("Example 3.13, Page 146 \n\n")

D = 0.32 ; // [m]
D_s = 0.36 ; // [m]
e = 0.02 ; // emissivity
l = 201 ; // [kJ/kg]
rho = 800 ; // [kg/m^3]
s = 5.670*10^-8 ;

T2 = 303 ; // [K]
T1 = 77 ; // [K]

// From equation 3.10.1
q1 = s*4*%pi*((D/2)^2)*(T1^4-T2^4)/[1/e+((D/D_s)^2)*(1/e-1)]; // [W]

evap = abs(q1)*3600*24/(l*1000); // [kg/day]
mass = 4/3*%pi*((D/2)^3)*rho;
boiloff = evap/mass*100 ; // percent

T_drop = (abs(q1))/(4*%pi*((D/2)^2))*(1/100); // [C]

printf("Rate at which nitrogen evaporates = %f kg/day \n",evap)
printf("Boil-off rate = %f percent \n",boiloff);
printf("Temperature drop between liquid Nitrogen and inner surface = %f C",T_drop);
```

Scilab code Exa 3.14 Rate of energy loss from satellite

```
clear;
```

```
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.14
// Page 147
printf("Example 3.14, Page 147 \n\n")

D = 1 ; // [m]
r = 6250 ; // [km]
D_surf = 300 ; // [km]
s = 5.670*10^-8;
e = 0.3 ;
Tc = -18+273 ; // [K]
T_surf = 27+273 ; // [K]

// Rate of emissino of radiant energy from the two faces of satellite disc
r_emission = 2*e*%pi*((D/2)^2)*s*Tc^4; // [W]

// A2*F21 = A1*F12
sina = (r/(r+D_surf));
F12 = sina^2;

// Rate at which the satellite receives and absorbs energy coming from earth
r_receive = e*s*(%pi*((D/2)^2))*F12*T_surf^4; // [W]

r_loss = r_emission - r_receive; // [W]

printf("Net Rate at which energy is leaving the satellite = %f W",r_loss);
```

Scilab code Exa 3.15 `Net radiative heat transfer`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 3
// Thermal Radiation

// Example 3.15
// Page 151
printf(”Example 3.15, Page 151 \n\n”)

// From example 3.10
F12 = 0.0363;
F11 = 0;
F13 = 1-F11-F12;
// Similarly
F21 = 0.0363;
F22 = 0;
F23 = 0.9637;

// Now, F31 = A1/A3*F13
F31 = 2/24*F13;
// Therefore
F32 = F31;
F33 = 1-F31-F32;

// Substituting into equation 3.11.6, 3.11.7, 3.11.8, we have f(1),f(2),f(3)

function[f]=flux(B)
    f(1)= B(1) - 0.4*0.0363*B(2) - 0.4*0.9637*B(3) - 0.6*(473^4)*(5.670*10^-8);
    f(2)= -0.4*0.0363*B(1) + B(2) - 0.4*0.9637*B(3) - 0.6*(5.670*10^-8)*(373^4);
    f(3)= 0.0803*B(1) + 0.0803*B(2) - 0.1606*B(3);
    funcprot(0);
```

```
endfunction

B = [0 0 0];
y = fsolve(B,flux);
printf(”\n B1 = %.1f W/mˆ2”,y(1));
printf(”\n B2 = %.1f W/mˆ2”,y(2));
printf(”\n B3 = %.1f W/mˆ2 \n”,y(3));

// Therefore
H1 = 0.0363*y(2) + 0.9637*y(3) ; // [W/mˆ2]
// and
q1 = 2*(y(1) - H1) ; // [W]

printf(”Net radiative heat transfer = %f W”,q1);
```

Chapter 4

Principles of Fluid Flow

Scilab code Exa 4.1 `Pressure drop in smooth pipe`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 4
// Principles of Fluid Flow

// Example 4.1
// Page 172
printf("Example 4.1, Page 172 \n\n");

L = 3 ; // Length, [m]
D = 0.01 ; // ID, [m]
V = 0.2 ; // Average Velocity, [m/s]

// From Table A.1 at 10 degree C
rho=999.7 ; // [kg/m^3]
v=1.306 * 10^-6 ; // [m^2/s]

Re_D=0.2*0.01/(1.306*10^-6) ;

```

```
// this value is less than the transition Reynolds
    number 2300.
// Hence flow is laminar. From eqn 4.4.19
f = 16/Re_D;

// from eqn 4.4.17
delta_p = 4*f*(L/D)*(rho*V^2)/2;

// since flow is laminar
V_max = 2*V;

printf("Pressure drop is %f Pa \n",delta_p);
printf(" Maximum velocity is %f m/s",V_max);
```

Scilab code Exa 4.2.a `Pressure drop and maximum velocity calculation`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 4
// Principles of Fluid Flow

// Example 4.2(a)
// Page 180
printf("Example 4.2(a), Page 180 \n\n")

L = 3 ; //[m]
D = 0.01 ; //[m]
V = 0.2 ; //[m/s]

// (a)
printf("(a) If the temperature of water is increased
     to 80 degree C \n");

```

```

// Properties of water at 80 degree C
rho = 971.8 ; // [kg/m^3]
v = 0.365 * 10^-6 ; // [m^2/s]

Re_D = D*V/v;

// flow is turbilent , so from eqn 4.6.4a

f=0.079*(Re_D)^(-0.25);
delta_p = (4*f*L*rho*V^2)/(D*2); // [Pa]
printf("Pressure drop is %f Pa \n",delta_p);

// from eqn 4.4.16

// x = (T_w/p)^0.5 = ((f/2)^0.5)*V ;
x = ((f/2)^0.5)*V ;
y_plus = 0.005*x/(0.365*10^-6);

// from eqn 4.6.1c & 4.6.2

V_max = x*(2.5* log(y_plus) + 5.5) ; // [m/s]
ratio = V_max/V;
printf("V_max = %f m/s \n",V_max);
printf("V_max/V_bar = %f \n\n",ratio);
```

Scilab code Exa 4.2.b `Pressure drop and maximum velocity calculation`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 4
// Principles of Fluid Flow

```

```
// Example 4.2(b)
// Page 180
printf("Example 4.2(b), Page 180 \n\n")

L = 3 ; //[m]
D = 0.01 ; //[m]
V = 0.2 ; //[m/s]

// (b)

V1=0.7;
v1 = 1.306 * 10^-6 ; // [m^2/s]

printf("(b) If the velocity is increased to 0.7 \n");
// if velocity of water is 0.7 m/s
V1=0.7; // [m/s]
Re_D1=V1*D/(1.306*10^-6);
printf("Reynolds no is %f \n",Re_D1);

// flow is again turbulent
f1 = 0.079*(Re_D1)^(-0.25);

delta_p1 = (4*f1*L*999.7*0.7^2)/(0.01*2); // [Pa]
printf("Pressure drop is %f Pa \n",delta_p1);

// x1 = (T_w/p)^0.5 = ((f1/2)^0.5)*V ;
x1 = ((f1/2)^0.5)*V1 ;

y1_plus = 0.005*x1/(v1);
printf("y+ at centre line = %f \n",y1_plus);

V_max1 = x1*(2.5* log(y1_plus) + 5.5) ; // [m/s]
printf("V_max is %f m/s \n",V_max1);

ratio1 = V_max1/V1;
printf("Vmax/Vbar = %f ",ratio1);
```

Scilab code Exa 4.3 `Pressure drop and power needed`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 4
// Principles of Fluid Flow

// Example 4.3
// Page 181
printf("Example 4.3, Page 181 \n\n")
P = 80 * 10^3 ; // [Pa]
L = 10 ; // [m]
V_bar = 1.9 ; // [m/s]
l = 0.25 ; // [m]
b = 0.15 ; // [m]

// Fully developed flow

// From Table A.2, for air at ! atm pressure and 25
   degree C
rho = 1.185 ; // [kg/m^3]
mew = 18.35 * 10^-6 ; // [kg/m s]

// At 80 kPa and 25 degree C
rho1 = rho*(80/101.3) ; // [kg/m^3]

// For given duct r=(b/a)
r = b/l;

D_e = (4*l/2*b/2)/(l/2 + b/2); // [m]

// From eqn 4.6.7
```

```

D_l = [2/3 + 11/24*0.6*(2-0.6)]*D_e ; // [m]

// Reynolds no based on D_l

Re = rho1*D_l*V_bar/mew;
printf("Reynolds no = %f \n",Re);

f = 0.079*(Re^-0.25) ;
printf("f = %f \n",f);

// From eqn 4.4.17

delta_P = 4*f*(L/D_l)*(rho1*(V_bar^2)/2);
printf("Pressure drop = %f Pa \n",delta_P);

power = delta_P*(V_bar*l*b)
printf("Power required = %f W",power);
```

Scilab code Exa 4.4 `Thickness of velocity boundary layer`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 4
// Principles of Fluid Flow

// Example 4.4
// Page 189
printf("Example 4.4, Page 189 \n\n")

l = 2 ; // [m]
b = 1 ; // [m]
V = 1 ; // [m/s]
```

```

// From table A.2

rho = 1.060 ; // [kg/m^3]
v = 18.97 * 10^-6 ; // [m^2/s]

// At x = 1.5m
x = 1.5 ; // [m]
Re = V*x/v; // Reynolds number

// From eqn 4.8.12

d = 5*x/(Re^(1/2))*1000 ; // [mm]
printf("Thickness of Boundary layer at x = 1.5 is %f
    mm \n",d)

Re_l = V*l/v;

// From eqn 4.8.19 and 4.8.16

c_f = 1.328*Re_l^-(1/2); // drag coefficient
printf("Drag Coefficient c_f = %f \n",c_f);

F_d = 0.00409*(1/2)*rho*(2*l*b)*1^2;
printf("Drag Force F_D = %f N",F_d);
```

Scilab code Exa 4.5 Drag coefficient and drag force

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 4
// Principles of Fluid Flow

```

```
// Example 4.5
// Page 195
printf("Example 4.5, Page 195 \n\n");

l = 2 ; // [m]
v = 4 ; // [m/s]

// From Table A.2

mew = 18.1*10^-6; // [N s/m^2]
rho = 1.205*1.5; // [kg/m^3]

Re_l = rho*v*l/mew;
// Boundary layer is partly laminar and partly
   turbulent , we shall use eqn 4.10.4
Cf = 0.074*(7.989*10^5)^(-0.2) - 1050/Re_l ;
printf("Drag coefficieent is %f \n",Cf)

D_f= Cf*1/2*rho*l*v^2;
printf("Drag force per meter width = %f N \n",D_f);

//from eqn 4.10.1

x = 3*10^5 * (18.1*10^-6)/(1.808*4);
printf("Value of x_c is %f m",x);
```

Chapter 5

Heat Transfer by Forced Convection

Scilab code Exa 5.1.a `Local heat transfer coefficient`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.1(a)
// Page 209
printf("Example 5.1(a) \n\n")

D = 0.015 ; // [m]
Q = 0.05 ; // [m^3/h]
H = 1000 ; // [W/m^2]
T_b = 40 ; // [degree C]

// From table A.1, properties at 40 degree C
k = 0.634 ; // [W/m K]
```

```
v = 0.659*10^-6 ; // [m^2/s]

V_bar = 4*Q/((%pi)*D^2);

Re_D = V_bar*D/v;

// Therefore, Laminar Flow, from eqn 5.2.8

h = 4.364*k/D; // [W/m^2 K]

printf("(a) Local heat transfer coefficient is %f W/
   m^2 K \n",h);
```

Scilab code Exa 5.1.b `Wall temperature`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.1(b)
// Page 209
printf("Example 5.1(b) \n\n")

D = 0.015 ; // [m]
Q = 0.05 ; // [m^3/h]
H = 1000 ; // [W/m^2]
T_b = 40 ; // [degree C]

// From table A.1, properties at 40 degree C
k = 0.634 ; // [W/m K]
v = 0.659*10^-6 ; // [m^2/s]
```

```

V_bar = 4*Q/((%pi)*D^2);

Re_D = V_bar*D/v;

// Therefore, Laminar Flow, from eqn 5.2.8

h = 4.364*k/D;

// From the definition of h in eqn 5.2.3, the local wal to bulk mean temperature difference is given by

T_w = H/h + T_b;

printf(”(b) Wall Temperature Tw = %f degree C”,T_w);
```

Scilab code Exa 5.2 `ratio of thermal entrance length to entrance length`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.2
// Page 213
printf(”Example 5.2,Page 213 \n\n”)

// From eqn 5.2.12 and 4.4.20
// Let r = Lth/Le
// r = 0.04305*Pr/0.0575;

```

```
function[T]=r(Pr)
    T = 0.04305*Pr/0.0575
endfunction

// For Pr = 0.01
r1 = r(0.01);
// For Pr = 0.1
r2 = r(1);
// For Pr = 100
r3 = r(100);

printf("Lth/Le at Pr = 0.01 is %f \n",r1);
printf("Lth/Le at Pr = 1 is %f \n",r2);
printf("Lth/Le at Pr = 100 is %f",r3);
```

Scilab code Exa 5.3.i `Length of tube`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.3(i)
// Page 215
printf("Example 5.3(i), Page 215 \n\n")

D = 0.015 ; // [m]
V = 1 ; // [m/s]
Tw = 90 ; // [degree C]
Tmi = 50 ; // [degree C]
Tmo = 65 ; // [degree C]

```

```
// (i)

// From Table A.1
k = 0.656 ; // [W/m K]
rho = 984.4 ; // [kg/m^3]
v = 0.497 * 10^-6 ; // [m^2/s]
Cp = 4178 ; // [J/kg K]
Pr = 3.12 ;
rho_in = 988.1 ; // [kg/m^3]

m_dot = %pi*(D^2)*rho_in*V/4 ; // [kg/s]

Re = 4*m_dot/(%pi*D*rho*v) ;

// Using eqn 5.3.2 and 4.6.4a
f = 0.079*(Re)^-0.25 ;

Nu = (f/2)*(Re-1000)*Pr/[1+12.7*(f/2)^(1/2)*((Pr^(2/3))-1)];
h = Nu*k/D; // [W/m^2 K]

// From the energy equation, extracting the value of L
L = m_dot*Cp*(Tmo-Tmi)*[log((Tw-Tmi)/(Tw-Tmo))]/[((Tw-Tmi)-(Tw-Tmo))*h*D*%pi]; // [m]

printf("The length of tube if the exit water temperature is 65 degree C = %f m\n",L);
```

Scilab code Exa 5.3.ii Exit water temperature

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
```

```
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.3(i)
// Page 215
printf("Example 5.3(ii), Page 215 \n\n")

D = 0.015 ; // [m]
V = 1 ; // [m/s]
Tw = 90 ; // [degree C]
Tmi = 50 ; // [degree C]
Tmo = 65 ; // [degree C]

// From Table A.1
k = 0.656 ; // [W/m K]
rho = 984.4 ; // [kg/m^3]
v = 0.497 * 10^-6 ; // [m^2/s]
Cp = 4178 ; // [J/kg K]
Pr = 3.12 ;
rho_in = 988.1 ; // [kg/m^3]

m_dot = %pi*(D^2)*rho_in*V/4 ; // [kg/s]

Re = 4*m_dot/(%pi*D*rho*v) ;

// Using eqn 5.3.2 and 4.6.4a
f = 0.079*(Re)^-0.25 ;

Nu = (f/2)*(Re-1000)*Pr/[1+12.7*(f/2)^(1/2)*((Pr
   ^(2/3))-1)];
h = Nu*k/D; // [W/m^2 K]

// From the energy equation, extracting the value of
   L
L = m_dot*Cp*(Tmo-Tmi)*[log((Tw-Tmi)/(Tw-Tmo))]/[((
   Tw-Tmi)-(Tw-Tmo))*h*D*%pi]; // [m]

```

```
// (ii)
printf("\nTrial and error method \n");

// Trial 1
printf("Trial 1\n");
printf("Assumed value of Tmo = 70 degree C\n");
T_mo = 70 ; // [degree C]
T_b = 60 ; // [degree C]

k1 = 0.659 ; // [W/m K]
rho1 = 983.2 ; // [kg/m^3]
v1 = 0.478 * 10^-6 ; // [m^2/s]
Cp1 = 4179 ; // [J/kg K]
Pr1 = 2.98 ;

Re1 = 4*m_dot/(%pi*D*rho1*v1);

// From Blasius eqn (4.6.4a), we get
f1 = 0.005928;

// Substituting this value into the Gnielinski Eqn
Nu_d = 154.97;
h = Nu_d*k1/D ; // [W/m^2 K]

// from eqn 5.3.3, we get
Tmo1 = 73.4 ;  // [degree C]
printf("Value of Tmo obtained = 73.4 degree C\n");

// Trial 2
printf("Trial 2\n");
printf("Assume Tmo = 73.4 degree C\n");
printf("Value of Tmo obtained = 73.6 degree C which
   is in reasonably close agreement with assumed
   value.\n")
```

Scilab code Exa 5.4 `Length of tube over which temperature rise occurs`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.4
// Page 219
printf("Example 5.4, Page 219 \n\n")

D_i = 0.05 ; // [m]
m = 300 ; // [kg/min]
m1 = m/60 ; // [kg/sec]
rho = 846.7 ; // [kg/m^3]
k = 68.34 ; // [W/m K]
c = 1274; //  [J/kg K]
v = 0.2937*10^-6 ; // [m^2/s]
Pr = 0.00468 ;

Re_D = 4*m1/(%pi*D_i*rho*v);

// Assuming both temperature and velocity profile
   are fully developed over the length of tube
// using eqn 5.3.6
Nu_D = 6.3 + 0.0167*(Re_D^0.85)*(Pr^0.93);

h = Nu_D*k/D_i;

// Equating the heat transferred through the wall of
    the tube to the change of enthalpy pf sodium
L = 300/60*1274*(500-400)/(h*%pi*D_i*30)

printf("Length of tube over which the temperature
   rise occurs = %f m",L)
```

Scilab code Exa 5.5 Rate of heat transfer to the plate

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.5
// Page 231
printf("Example 5.5, Page 231 \n")

V = 15 ; // [m/s]
s=0.2 ; // [m]
T_m = (20+60)/2; // [degree C]
// Properties at mean temp = 40 degree C
v = 16.96*10^-6; //  [m^2/s]
rho = 1.128 ; // [kg/m^3]
k = 0.0276; // [W/m K]
Pr = 0.699;
A=s^2;
Re_L = V*0.2/v;
// This is less than 3*10^5, hence the boundary layer may be assumed to be laminar over the entire length.
// from eqn 4.8.19

Cf = 1.328/(Re_L)^0.5
Fd = 2*Cf*1/2*rho*A*V^2;

// From eqn 5.5.10
Nu_l = 0.664*(Pr^(1/3))*(Re_L^(1/2));
```

```

h = Nu_l*k/s;
// Therefore rate of heat transfer q is
q = 2*A*h*(60-20);// [W]

// With a turbulent boundary layer from leading edge , the drag coefficient is given by eqn 4.10.4
Cf1 = 0.074*(Re_L)^(-0.2);
Fd1 = 2*Cf1*1/2*rho*A*V^2; // [N]

// from eqn 5.8.3 with C1 = 0
Nu_l1 = 0.0366*(0.699^(1/3))*(Re_L^(0.8));

h1 = Nu_l1*k/s; // [W/m^2 K]
q1 = 2*A*h1*(60-20);

printf("For Laminar Boundary Layer \n");
printf("Rate of Heat transfer = %f W\n",q);
printf("Drag force = %f N \n \n",Fd)

printf("For Turbulent Boundary Layer from the leading edge \n");
printf("Rate of Heat transfer = %f W\n",q1);
printf("Drag force = %f N\n",Fd1)
```

Scilab code Exa 5.6.i `Heat transfer rate`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


```

```
// Example 5.6(i)
// Page 235
printf("Example 5.6(i), Page 235 \n\n")

D = 0.075 ; // [m]
V = 1.2 ; // [m/s]
T_air = 20 ; // [degree C]
T_surface = 100 ; // [degree C]
T_m = (T_air+T_surface)/2;

v = 18.97*10^-6 ; // [m^2/s]
k = 0.0290 ; // [W/m K]
Pr = 0.696 ;

Re_D = V*D/v;

Nu = 0.3 + [(0.62*(Re_D^(1/2))*(Pr^(1/3)))/[(1+((0.4/Pr)^(2/3)))^(1/4)]]*([1+((Re_D/282000)^(5/8))]^(4/5)) ;

h = Nu*k/D ; // [W/m^2 K]

flux = h*(T_surface - T_air); // [W/m^2]
q = flux*%pi*D*1; // [W/m]

printf("Heat transfer rate per unit length = %f W/m\n",q);
```

Scilab code Exa 5.6.ii Average wall tempeature

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
```

```
// Heat Transfer by Forced Convection


// Example 5.6(ii)
// Page 235
printf(”Example 5.6(ii), Page 235 \n\n”)

D = 0.075 ; // [m]
V = 1.2 ; // [m/s]
T_air = 20 ; // [degree C]
T_surface = 100 ; // [degree C]
T_m = (T_air+T_surface)/2;

v = 18.97*10^-6 ; // [m^2/s]
k = 0.0290 ; // [W/m K]
Pr = 0.696 ;

Re_D = V*D/v;
Nu = 0.3 + [(0.62*(Re_D^0.5)*(Pr^(1/3)))/[(1+((0.4/Pr)^(2/3)))^(1/4)]]*[1+(Re_D/282000)^(5/8)]^(5/8) ;
h = Nu*k/D ; // [W/m^2 K]
flux = h*(T_surface - T_air); // [W/m^2]

// (ii) Using Trial and error method
T_avg = 1500/flux*(T_surface - T_air);

T_assumd = 130 ; // [degree C]
Tm= 75 ; // [degree C]

v1 = 20.56*10^-6 ; // [m^2/s]
k1 = 0.0301 ; // [W/m K]
Pr1 = 0.693 ;

Re_D1 = V*D/v1;


// Using eqn 5.9.8
```

```
Nu1 = 33.99;
h = Nu1*k1/D;
// Therefore
T_diff = 1500/h; // [degree C]
T_avg_calc = 129.9 ; // [degree C]
printf("Assumed average wall temperature = %f degree
    C\n",T_assumd);
printf("Calculated average wall Temperature = %f
    degree C\n",T_avg_calc);
printf("Hence, Average wall Temperature = %f degree C
    ",T_avg_calc);
```

Scilab code Exa 5.7.i `Pressure drop`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.7(i)
// Page 241
printf("Example 5.7(i), Page 241 \n \n");

// Given data
D = 0.0125 ; // [m]
ST = 1.5*D ;
SL = 1.5*D ;
V_inf = 2 ; // [m/s]

N = 5;
Tw = 70; // [degree C]
Tmi = 30; // [degree C]
```

```
L = 1; // [m]
// Properties of air at 30 degree C
rho = 1.165 ; // [kg/m^3]
v = 16.00 *10^-6 ; // [m^2/s]
Cp = 1.005 ; // [kJ/kg K]
k = 0.0267 ; // [W/m K]
Pr = 0.701;

// From eqn 5.10.2
Vmax = ST/(SL-D)*V_inf ; // [m/s]
Re = Vmax*D/v ;

// From fig 5.15
f = 0.37/4;
// Also, tube arrangement is square
X = 1;
// From eqn 5.10.6
delta_P = 4*f*N*X*(rho*Vmax^2)/2 ; // [N/m^2]

printf("(i) Pressure drop of air across the bank is
    %f N/m^2 \n",delta_P);
```

Scilab code Exa 5.7.ii Exit temperature of air

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.7(ii)
// Page 241
printf("Example 5.7(ii), Page 241 \n \n");
```

```

D = 0.0125 ; // [m]
ST = 1.5*D ;
SL = 1.5*D ;
V_inf = 2 ; // [m/s]
N = 5;
Tw = 70; // [degree C]
Tmi = 30; // [degree C]
L = 1; // [m]

rho = 1.165 ; // [kg/m^3]
v = 16.00 *10^-6 ; // [m^2/s]
Cp = 1.005*1000 ; // [J/kg K]
k = 0.0267 ; // [W/m K]
Pr = 0.701;

// From eqn 5.10.2
Vmax = ST/(SL-D)*V_inf ; // [m/s]
Re = Vmax*D/v ;

// From fig 5.15
f = 0.37/4;
// Also, tube arrangement is square
X = 1;
// From eqn 5.10.6
delta_P = 4*f*N*X*(rho*Vmax^2)/2 ; // [N/m^2]

// At 70 degree C
Pr1 = 0.694 ;
// From table 5.4 and 5.5

C1 = 0.27;
m = 0.63;
C2 = 0.93;

// Substituting in Eqn 5.10.5
Nu = C1*C2*(Re^m)*(Pr^0.36)*(Pr/Pr1)^(1/4);
h = Nu*k/D; // [W/m^2 K]
```

```

// For 1 m long tube
m_dot = rho*(10*1.5*D*1)*2; // [kg/s]

// Substituting m_dot in 5.3.4 and solving, we get
function[f]=temp(Tmo)
    f(1) = h*(%pi*D*L)*50*[(Tw-Tmi)-(Tw-Tmo(1))]/[log((Tw-Tmi)/(Tw-Tmo(1)))]-m_dot*Cp*(Tmo(1)-Tmi) ;
    // h*(%pi*D*L)*N*((Tw-Tmi)-(Tw-Tmo))/log[(Tw-Tmi)/(Tw-Tmo)] - m_dot*Cp*(Tmo - Tmi);
    funcprot(0);
endfunction

Tmo = 40; // Initial assumed value for fsolve function
y = fsolve(Tmo,temp);
printf("Tmo = %f \n",y);

printf("(ii) Exit temperature of air = %f degree C \n",y);
```

Scilab code Exa 5.7.iii Heat transfer rate

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 5
// Heat Transfer by Forced Convection


// Example 5.7(iii)
// Page 241
printf("Example 5.7(iii), Page 241 \n \n");
```

```
D = 0.0125 ; // [m]
ST = 1.5*D ;
SL = 1.5*D ;
V_inf = 2 ; // [m/s]
N = 5;
Tw = 70; // [degree C]
Tmi = 30; // [degree C]
L = 1; // [m]

rho = 1.165 ; // [kg/m^3]
v = 16.00 *10^-6 ; // [m^2/s]
Cp = 1.005*1000 ; // [J/kg K]
k = 0.0267 ; // [W/m K]
Pr = 0.701;

// From eqn 5.10.2
Vmax = ST/(SL-D)*V_inf ; // [m/s]
Re = Vmax*D/v ;

// From fig 5.15
f = 0.37/4;
// Also, tube arrangement is square
X = 1;
// From eqn 5.10.6
delta_P = 4*f*N*X*(rho*Vmax^2)/2 ; // [N/m^2]

// At 70 degree C
Pr1 = 0.694 ;
// From table 5.4 and 5.5

C1 = 0.27;
m = 0.63;
C2 = 0.93;

// Substituting in Eqn 5.10.5
Nu = C1*C2*(Re^m)*(Pr^0.36)*(Pr/Pr1)^(1/4);
h = Nu*k/D; // [W/m^2 K]
```

```

// For 1 m long tube
m_dot = rho*(10*1.5*D*1)*2; // [kg/s]

// Substituting m_dot in 5.3.4 and solving, we get
function[f]=temp(Tmo)
    f(1) = h*(%pi*D*L)*50*[(Tw-Tmi)-(Tw-Tmo(1))]/[log((Tw-Tmi)/(Tw-Tmo(1)))]-m_dot*Cp*(Tmo(1)-Tmi) ;
    // h*(%pi*D*L)*N*((Tw-Tmi)-(Tw-Tmo))/log[(Tw-Tmi)/(Tw-Tmo)] - m_dot*Cp*(Tmo - Tmi);
    funcprot(0);
endfunction

Tmo = 40; // Initial assumed value for fsolve function
y = fsolve(Tmo,temp);

// Heat transfer rate q
q = h*(%pi*D*L)*50*((Tw-Tmi)-(Tw-y))/(log((Tw-Tmi)/(Tw-y)));

printf(”(iii) Heat transfer rate per unit length to air = %f W”,q);
```

Chapter 6

Heat Transfer by Natural convection

Scilab code Exa 6.1 `Average nusselt number`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 6
// Heat Transfer by Natural Convection


// Example 6.1
// Page 258
printf("Example 6.1, Page 258 \n \n");

H = 0.5 ; // [m]
T_h = 100; // [degree C]
T_l = 40; // [degree C]

v = 20.02*10^-6 ; // [m/s]
Pr = 0.694;
k = 0.0297; // [W/m K]
```

```
T = (T_h+T_l)/2 + 273 ; // [K]
printf("Mean film temperature = %f K \n",T);
B = 1/T;

Gr = 9.81*B*((T_h-T_l)*H^3)/(v^2);
Ra = Gr*Pr;

// (a)
// Exact analysis - Equation 6.2.17
disp("(a)");
printf("Exact analysis\n");
Nu_a = 0.64*(Gr^(1/4))*(Pr^0.5)*((0.861+Pr)^(-1/4));
printf("Nu_L = %f \n",Nu_a);

// (b)
// Integral method - Equation 6.2.29
disp("(b)");
printf("Integral method \n");
Nu_b = 0.68*(Gr^(1/4))*(Pr^0.5)*((0.952+Pr)^(-1/4));
printf("Nu_L = %f \n",Nu_b);

// (c)
// McAdams correlation - Equation 6.2.30
disp("(c)");
printf("McAdams correlation \n");
Nu_c = 0.59*(Ra)^(1/4);
printf("Nu_L = %f \n",Nu_c);

// (d)
// Churchill and Chu correlation - Equation 6.2.31
disp("(d)")
printf("Churchill and Chu correlation\n");
Nu_d = 0.68 + 0.670*(Ra^(1/4))/[1+(0.492/Pr)^(9/16)
   ]^(4/9);
printf("Nu_L = %f \n",Nu_d);
```

Scilab code Exa 6.2 `Reduce the equation`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 6
// Heat Transfer by Natural Convection


// Example 6.2
// Page 259
printf("Example 6.2, Page 259 \n \n");

Tm = 150 ; // [degree C]
// From table A.2
v = 28.95*10^-6 ; // [m^2/s]
Pr = 0.683;
k = 0.0357 ; // [W/m K]

B = 1/(273+Tm); // [K^-1]

// from eqn 6.2.30
printf("Equation 6.2.30 \n h = k/L*0.59*[9.81*B*(Tw-
    Tinf)*(L^3)*0.683/(v^2)]^(1/4)\n")
// h = k/L*0.59*[9.81*B*(Tw-Tinf)*(L^3)*0.683/(v^2)
    ]^(1/4);
// simplifying we get
// h = 1.38*[(Tw-Tinf)/L]^(1/4)
printf("Reduces to h = 1.38*[(Tw-Tinf)/L]^(1/4) \n")


// From eqn 6.2.33
// h*L/k = 0.10*[9.81*B*(Tw-Tinf)*(L^3)*0.683/(v^2)
```

```
]^(1/3);
printf("Equation 6.2.33 \n h*L/k = 0.10*[9.81*B*(Tw-Tinf)*(L^3)*0.683/(v^2)]^(1/3) \n");
// simplifying
// h = 0.95*(Tw-Tinf)^1/3
printf("Reduces to h = 0.95*(Tw-Tinf)^1/3 \n");

printf("where h is expressed in W/m^2 K, (Tw-Tinf) in C and L in metres \n");
```

Scilab code Exa 6.3 `Time for cooling of plate`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 6
// Heat Transfer by Natural Convection


// Example 6.3
// Page 260
printf("Example 6.3, Page 260 \n \n");

s = 0.2 ; // [m]
d = 0.005 ; // [m]
rho = 7900 ; // [kg/m^3]
Cp = 460 ; // [J/kg K]

T_air = 20 ; // [C]
// For 430 C to 330 C
T_avg = 380 ; // [C]
Tm = (T_avg + T_air)/2 ; // [C]


```

```
v = 34.85*10^-6 ; // [m^2/s]
Pr = 0.680 ;
k = 0.0393 ; // [W/m K]

Re = 9.81*1/(273+Tm)*(T_avg-T_air)*(s^3)/(v^2)*Pr;

// From eqn 6.2.31
Nu = 0.68 + 0.670*(Re^(1/4))/[1+(0.492/Pr)^(4/9)]^(4/9);

h = Nu*k/s; // [W/m^2 K]
t1 = rho*s*s*d*Cp/((s^2)*2*h)*log((430-T_air)/(330-T_air)); // [s]
printf(”Time required for the plate to cool from 430 C to 330 C is %f s\n”,t1);

// for 330 to 230
h2 = 7.348 ; // [W/m^2 K]
t2 = rho*s*s*d*Cp/((s^2)*2*h2)*log((330-T_air)/(230-T_air)); // [s]
printf(”Time required for the plate to cool from 330 C to 230 C is %f s\n”,t2);

// for 230 to 130
h3 = 6.780; // [W/m^2 K]
t3 = rho*s*s*d*Cp/((s^2)*2*h3)*log((230-T_air)/(130-T_air)); // [s]
printf(”Time required for the plate to cool from 230 C to 130 C is %f s\n”,t3);

// Total time

time = t1+t2+t3;
minute = time/60;
printf(”Hence, time required for the plate to cool from 430 C to 130 C \n = %f s\n = %f min”,time,minute);
```

Scilab code Exa 6.4 `True air temperature`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 6
// Heat Transfer by Natural Convection


// Example 6.4
// Page 264
printf("Example 6.4, Page 264 \n \n");

D = 0.006 ; // [m]
e = 0.1 ;
Ti = 800 ; // [C]
Ta = 1000 ; // [C]
// Rate at which heat gained = net radiant heat,
   gives h*(Ta-800) = 1306.0 ; // [W/m^2]

// Using trial and error method
// Trial 1
printf("Trial 1 \n");
// Let Ta = 1000 degree C
printf("Let Ta = 10000 C \n");

Tm = (Ta+Ti)/2;
// From table A.2
v = 155.1*10^-6 ; // [m^2/s]
k = 0.0763 ; // [W/m K]
Pr = 0.717 ;

Gr = 9.81*1/1173*(200*D^3)/(v^2);
```

```
Ra = Gr*Pr ;

// From eqn 6.3.2
Nu = 0.36 + 0.518*(Ra^(1/4))/[1+(0.559/Pr)^(9/16)
   ]^(4/9);
h = Nu*k/D;
x = h*(Ta-Ti); // [W/m^2]
printf("Value of h(Ta-800) = %f W/m^2, which is much
    larger than the required value of 1306 W/m^2 \n"
   ,x);

// Trial 2
printf("\nTrial 2 \n");
// Let Ta = 900
printf("Let Ta = 900 C \n");
Ra2 = 6.42 ;
Nu2 = 0.9841 ;
h2 = 12.15 ;
x2 = h2*(900-800);
printf("Value of h(Ta-800) = %f W/m^2, which is a
   little less than the required value of 1306 W/m^2
    \n",x2);

// Trial 3
printf("\nTrial 3 \n");
// Let Ta = 910
printf("Let Ta = 910 C \n");
Ra3 = 6.93 ;
Nu3 = 0.9963 ;
h3 = 12.33 ;
x3 = h3*(910-800);
printf("Value of h(Ta-800) = %f W/m^2 \nThis value
   is little more than the required value of 1306 W/
   m^2 \n",x3);
// Interpolation
T = 900 + (910-900)*(1306-x2)/(x3-x2);
printf("\nThe correct value of Ta obtained by
   interpolation is %f C",T);
```

Scilab code Exa 6.5 `Rate of heat flow by natural convection`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 6
// Heat Transfer by Natural Convection


// Example 6.5
// Page 269
printf("Example 6.5, Page 269 \n \n");

T_p = 75 ; // Temperature of absorber plate , degree C
T_c = 55 ; // Temperature of glass cover , degree C
L = 0.025 ; // [m]

H = 2 ; // [m]
Y = 70 ; // degree

a = 19/180*%pi ; // [Radians]

r = H/L ;

T_avg = (T_p+T_c)/2+273 ; // [K]
// Properties at 65 degree C
k = 0.0294 ; // [W/m K]
v = 19.50*10^-6 ; // [m^2/s]
Pr = 0.695 ;

Ra = 9.81*(1/T_avg)*(T_p-T_c)*(L^3)/(v^2)*Pr*cos(a);

```

```
// From eqn 6.4.3
Nu = 0.229*(Ra)^0.252;

h = Nu*k/L ; // [W/m^2 K]

Rate = h*2*1*(T_p-T_c); // [W]

printf("Heat transfer rate = %f W",Rate);
```

Scilab code Exa 6.6 `Average Heat transfer coefffficient`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 6
// Heat Transfer by Natural Convection


// Example 6.6
// Page 270
printf("Example 6.6, Page 270 \n \n");

T_air = 30 ; // [C]
D = 0.04 ; // [m]
T_s = 70 ; // surface temperature, [C]
V = 0.3 ; // [m/s]

Tm = (T_air + T_s)/2 ; // [C]
// Properties at Tm
v = 17.95*10^-6 ; // [m^2/s]
Pr = 0.698 ;
k = 0.0283 ; // [W/m K]

Gr = 9.81*1/323*(T_s-T_air)*(D^3)/v^2;
```

```
Re = V*D/v ;
X = Gr/Re^2 ;
printf("Since Gr/Re^2 = %f is > 0.2, we have a combined convection situation. \n\n",X);

// From Eqn 5.9.8
Nu_forced = 0.3 + 0.62*(Re^0.5)*(Pr^(1/3))/[[1+(0.4/Pr)^(2/3)]^(1/4)]*[1+(Re/282000)^(5/8)]^(4/5);

// Substituting in Eqn 6.5.1
Nu = Nu_forced*[1+6.275*(X)^(7/4)]^(1/7);
h = Nu*(k/D);
printf("The Average heat transfer coefficient = %f W/m^2 K",h);
```

Chapter 7

Heat Exchangers

Scilab code Exa 7.1 `Heat transfer coeffficient`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 7
// Heat Exchangers


// Example 7.1
// Page 285
printf("Example 7.1, Page 285 \n \n");

h = 2000 ; // [W/m^2 K]
// From Table 7.1
U_f = 0.0001 ; // fouling factor, m^2K/W
h_f = 1/[1/h+U_f];
printf("Heat transfer coefficient including the effect of foulung = %f W/m^2 K \n",h_f);

p = (h-h_f)/h*100;
printf("Percentage reduction = %f  \n",p);
```

Scilab code Exa 7.2 `Area of heat exchanger`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 7
// Heat Exchangers


// Example 7.2
// Page 294
printf(”Example 7.2, Page 294 \n \n”);

m = 1000 ; // [kg/h]
Thi = 50 ; // [C]
The = 40 ; // [C]
Tci = 35 ; // [C]
Tce = 40 ; // [C]
U = 1000 ; // OHTC, W/m^2 K

// Using Eqn 7.5.25
q = m/3600*4174*(Thi-The) ; // [W]

delta_T = ((Thi-Tce)-(The-Tci))/log((Thi-Tce)/(The-Tci)); // [C]
printf(”delta T = %f \n\n”,delta_T);

// T1 = Th and T2 = Tc
R = (Thi-The)/(Tce-Tci) ;
S = (Tce-Tci)/(Thi-Tci) ;
// From fig 7.15,
F =0.91 ;

```

```
printf("Taking T1 = Th and T2 = Tc \n")
printf("R = %f, S = %f \n",R,S);
printf("Hence, F = %f \n \n",F);

// Alternatively, taking T1 = Tc and T2 = Th
R = (Tci-Tce)/(The-Thi);
S = (The-Thi)/(Tci-Thi);

// Again from fig 7.15,
F =0.91 ;

printf("Taking T1 = Tc and T2 = Th \n")
printf("R = %f, S = %f \n",R,S);
printf("Hence, F = %f \n",F);

A = q/(U*F*delta_T);
printf("\nArea = %f m^2",A);
```

Scilab code Exa 7.3 `Mean temperature difference`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 7
// Heat Exchangers


// Example 7.3
// Page 295
printf("Example 7.3, Page 295 \n \n");

// Because of change of phase, Thi = The
Thi = 100 ; // [C], Saturated steam
The = 100 ; // [C], Condensed steam
```

```
Tci = 30 ; // [C], Cooling water inlet
Tce = 70 ; // [C], cooling water outlet

R = (Thi-The)/(Tce-Tci) ;
S = (Tce-Tci)/(Thi-Tci) ;

// From fig 7.16
F = 1;

// For counter flow arrangement
Tm_counter = ((Thi-Tce)-(The-Tci))/log((Thi-Tce)/(The-Tci)); // [C]
// Therefore
Tm = F*Tm_counter ;
printf("Mean Temperaature Difference = %f C",Tm)
```

Scilab code Exa 7.4.a `Area of heat exchanger`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 7
// Heat Exchangers


// Example 7.4(a)
// Page 302
printf("Example 7.4(a), Page 302 \n \n");

// (a)
printf("(a) \n");
// Using Mean Temperature Difference approach
m_hot = 10 ; // [kg/min]
m_cold = 25 ; // [kg/min]
```

```
hh = 1600 ; // [W/m^2 K], Heat transfer coefficient on hot side
hc = 1600 ; // [W/m^2 K], Heat transfer coefficient on cold side

Thi = 70 ; // [C]
Tci = 25 ; // [C]
The = 50 ; // [C]

// Heat Transfer Rate, q
q = m_hot/60*4179*(Thi-The); // [W]

// Heat gained by cold water = heat lost by the hot water
Tce = 25 + q*1/(m_cold/60*4174); // [C]

// Using equation 7.5.13
Tm = ((Thi-Tci)-(The-Tce))/log((Thi-Tci)/(The-Tce)); // [C]
printf("Mean Temperature Difference = %f C \n",Tm);

U = 1/(1/hh + 1/hc); // [W/m^2 K]
A = q/(U*Tm); // Area, [m^2]
printf("Area of Heat Exchanger = %f m^2 \n",A);
```

Scilab code Exa 7.4.b Exit temperature of hot and cold streams

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 7
// Heat Exchangers


```

```
// Example 7.4(b)
// Page 302
printf(”Example 7.4(b), Page 302 \n \n”);

// Using Mean Temperature Difference approach
m_hot = 10 ; // [kg/min]
m_cold = 25 ; // [kg/min]
hh = 1600 ; // [W/m^2 K], Heat transfer coefficient on hot side
hc = 1600 ; // [W/m^2 K], Heat transfer coefficient on cold side

Thi = 70 ; // [C]
Tci = 25 ; // [C]
The = 50 ; // [C]

// Heat Transfer Rate, q
q = m_hot/60*4179*(Thi-The); // [W]

// Heat gained by cold water = heat lost by the hot water
Tce = 25 + q*1/(m_cold/60*4174); // [C]

// Using equation 7.5.13
Tm = ((Thi-Tci)-(The-Tce))/log((Thi-Tci)/(The-Tce)); // [C]
U = 1/(1/hh + 1/hc); // [W/m^2 K]
A = q/(U*Tm); // Area, [m^2]

m_hot = 20 ; // [kg/min]
// Flow rate on hot side i.e. 'hh' is doubled
hh = 1600*2^0.8 ; // [W/m^2 K]
U = 1/(1/hh + 1/hc); // [W/m^2 K]
m_hC_ph =  m_hot/60*4179 ; // [W/K]
m_cC_pc = m_cold/60*4174 ; // [W/K]
// Therefore
C = m_hC_ph/m_cC_pc ;
NTU = U*A/m_hC_ph ;
```

```
printf(”NTU = %f \n”,NTU);

// From equation 7.6.8
e = [1 - exp(-(1+C)*NTU)]/(1+C) ;

// Therefore (Thi - The)/(Thi - Tci) = e , we get
The = Thi - e*(Thi - Tci); // [C]

// Equating the heat lost by water to heat gained by cold water , we get
Tce = Tci + [m_hC_ph*(Thi-The)]/m_cC_pc;
printf(”Exit temperature of cold and hot stream are %f C and %f C respectively.”,Tce,The);
```

Scilab code Exa 7.5 `Exit Temperature`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 7
// Heat Exchangers


// Example 7.5
// Page 304
printf(”Example 7.5, Page 304 \n \n”);

mc = 2000 ; // [kg/h]
Tce = 40 ; // [C]
Tci = 15 ; // [C]
Thi = 80 ; // [C]
U = 50 ; // OHTC, [W/m^2 K]
A = 10 ; // Area, [m^2]

```

```
// Using effective NTU method
// Assuming m_c*C_pc = (m*C_p)s
NTU = U*A/(mc*1005/3600);
e = (Tce-Tci)/(Thi-Tci);
// From fig 7.23, no value of C is found corresponding to the above values, hence assumption was wrong.
// So, m_h*C_ph must be equal to (m*C_p)s, proceeding by trail and error method


printf("m_h(kg/h      NTU      C         e      T_he(C)         T_he(C) (Heat Balance)");

mh = rand(1:5);
NTU = rand(1:5);
The = rand(1:5);
The2 = rand(1:5);

mh(1) = 200
NTU(1) = U*A/(mh(1)*1.161);
//Corresponding Values of C and e from fig 7.23
C = .416;
e = .78;
//From Equation 7.6.2 Page 297
The(1) = Thi - e*(Thi-Tci)
//From Heat Balance
The2(1) = Thi -  mc*1005/3600*(Tce-Tci)/(mh(1)*1.161);
printf("\n\n   %i      %.3f        %.3f        %.3f      %.2f         %.2f",mh(1),NTU(1),C,e,The(1),The2(1));

mh(2) = 250
NTU(2) = U*A/(mh(2)*1.161);
//Corresponding Values of C and e from fig 7.23
C = .520;
e = .69;
```

```
//From Equation 7.6.2 Page 297
The(2) = Thi - e*(Thi-Tci)
//From Heat Balance
The2(2) = Thi -  mc*1005/3600*(Tce-Tci)/(mh(2)*1.161);
printf("\n\n    %i      %.3f        %.3f        %.3f        %.2f        %.2f",mh(2),NTU(2),C,e,The(2),The2(2));

mh(3) = 300
NTU(3) = U*A/(mh(3)*1.161);
//Corresponding Values of C and e from fig 7.23
C = .624;
e = .625;
//From Equation 7.6.2 Page 297
The(3) = Thi - e*(Thi-Tci)
//From Heat Balance
The2(3) = Thi -  mc*1005/3600*(Tce-Tci)/(mh(3)*1.161);
printf("\n\n    %i      %.3f        %.3f        %.3f        %.2f        %.2f",mh(3),NTU(3),C,e,The(3),The2(3));

mh(4) = 350
NTU(4) = U*A/(mh(4)*1.161);
//Corresponding Values of C and e from fig 7.23
C = .728;
e = .57;
//From Equation 7.6.2 Page 297
The(4) = Thi - e*(Thi-Tci)
//From Heat Balance
The2(4) = Thi -  mc*1005/3600*(Tce-Tci)/(mh(4)*1.161);
printf("\n\n    %i      %.3f        %.3f        %.3f        %.2f        %.2f",mh(4),NTU(4),C,e,The(4),The2(4));

mh(5) = 400
```

```
NTU(5) = U*A/(mh(5)*1.161);
//Corresponding Values of C and e from fig 7.23
C = .832;
e = .51;
//From Equation 7.6.2 Page 297
The(5) = Thi - e*(Thi-Tci)
//From Heat Balance
The2(5) = Thi -  mc*1005/3600*(Tce-Tci)/(mh(5)*1.161);
printf("\n\n    %i      %.3f        %.3f        %.3f       %.2f        %.2f",mh(5),NTU(5),C,e,The(5),The2(5));

clf();
plot(mh,The,mh,The2,[295 295 200],[0 39.2 39.2])
xtitle('The vs mh','mh (kg/hr)','The (C)');
printf("\n\n From the plot, value of mh = 295 kg/hr and correspondingly The = 39.2 C")
```

Chapter 8

Condensation and boiling

Scilab code Exa 8.1 `Average Heat Transfer Coefficient`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 8
// Condensation and Boiling


// Example 8.1
// Page 318
printf("Example 8.1, Page 318 \n \n");
Ts = 80 ; // [C]
Tw = 70 ; // [C]
L = 1 ; // [m]
g = 9.8 ; // [m/s^2]

// Assuming condensate film is laminar and Re < 30
Tm = (Ts + Tw)/2 ;
// From table A.1
rho = 978.8 ; // [kg/m^3]
k = 0.672 ; // [W/m K]
```

```
u = 381 *10^-6 ; // [kg/m s]
v = u/rho ;
// At 80 C,
lambda = 2309 ; // [kJ/kg]
// Substituting in eqn 8.3.9, we get
h = 0.943*[(lambda*1000*(rho^2)*g*(k^3))/((Ts-Tw)*u*L)]^(1/4); // [W/m^2 K]

rate = h*L*(Ts-Tw)/(lambda*1000); // [kg/m s]
Re = 4*rate/u;
printf("Assuming condensate film is laminar and Re < 30 \n");
printf("h = %f W/m^2 K\n",h);
printf("Re_L = %f \n",Re);
printf("Initial assumption was wrong, Now considering the effect of ripples, we get\n");

// Substituting h = Re*(lambda*1000)*u/(4*L*(Ts-Tw)), in eqn 8.3.12
Re = [[[4*L*(Ts-Tw)*k/(lambda*1000*u)*(g/(v^2))^(1/3)]+5.2]/1.08]^(1/1.22);
// From eqn 8.3.12
h = [Re/(1.08*(Re^1.22)-5.2)]*k*((g/v^2)^(1/3)); // [W/m^2 K]
m = h*L*10/(lambda*1000); // rate of condensation , [kg/m s]

printf("Re = %f \n",Re);
printf("Heat Transfer Cofficient = %f W/m^2 K \n",h);
printf("Rate of condensation = %f kg/m s",m);
```

Scilab code Exa 8.2 `Average heat transfer coefficient and film Reynolds number`

```
clear;
```

```
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 8
// Condensation and Boiling


// Example 8.2
// Page 321
printf("Example 8.2, Page 321 \n \n");

Ts = 262 ; // [K]
D = 0.022 ; // [m]
Tw = 258 ; // [K]

Tm = (Ts+Tw)/2;
// Properties at Tm
rho = 1324 ; // [kg/m^3]
k = 0.1008 ; // [W/m K]
v = 1.90*10^-7 // [m^2/s];
lambda = 215.1*10^3 ; // [J/kg]
g = 9.81 ; // [m/s^2]
u = v*rho ; // Viscosity

// From eqn 8.4.1
h = 0.725*[lambda*(rho^2)*g*(k^3)/((Ts-Tw)*u*D)]^(1/4);

rate = h*%pi*D*(Ts-Tw) /lambda ; // [kg/s m]
Re = 4*rate/u ;

printf("Heat transfer coefficient = %f W/m^2 K\n",h);
printf("Condensation rate per unit length = %f kg/s m \n",rate);
printf("Film Reynolds number = %f \n",Re);
```

Scilab code Exa 8.3 `Length of the tube`

```
clear;
clc;

// A TeTwtbook on HEAT TRANSFER by S P SUKHATME
// Chapter 8
// Condensation and Boiling


// ETwample 8.3
// Page 322
printf("Example 8.3, Page 322 \n \n");

m = 25/60 ; // [kg/sec]
ID = 0.025 ; // [m]
OD = 0.029 ; // [m]
Tci = 30 ; // [C]
Tce = 70 ; // [C]
g = 9.8 ; // [m/s^2]

Ts = 100 ; // [C]
// Assuming 5.3.2 is valid, properties at 50 C
// Properties at Tm
rho = 988.1 ; // [kg/m^3]
k = 0.648 ; // [W/m K]
v = 0.556*10^-6 // [m^2/s];
Pr = 3.54 ;
Re = 4*m/(%pi*ID*rho*v);
// From eqn 4.6.4a
f = 0.005635;
// From eqn 5.3.2
Nu = 198.39 ;
h = Nu*k/ID ;
```

```

// Assuming average wall temperature = 90 C
Tw = 90 ; // [C]
Tm = (Tw+Ts)/2;
// Properties at Tm
// Properties at Tm
rho = 961.9 ; // [kg/m^3]
k = 0.682 ; // [W/m K]
u = 298.6*10^-6 ; // [kg/m s]
lambda = 2257*10^3 ; // [J/kg]

h = 0.725*[lambda*(rho^2)*g*(k^3)/((Ts-Tw)*u*OD)]^(1/4);
// Equating the heat flow from the condensing steam to the tube wall, to the heat flow from the tube wall to the flowing water.
// Solving the simplified equation
function[f] =temp(Tw)
    f=(100-Tw)^(3/4)-8.3096/[log((Tw-Tci)/(Tw-Tce))];
    funcprot(0);
endfunction

T=fsolve(Tw,temp);
printf("Temperature obtained from trial and error = %f C \n",T);

// Therefore
hc = 21338.77/(100-T)^(1/4); // [W/m^2 K]
printf("h_c = %f W/m^2 K \n",hc);

// Now, equating the heat flowing from the condensing steam to the tube wall to the heat gained by the water, we have
function[g] =lngth(l)
    g=hc*(%pi*OD*l)*(100-T)-m*4174*(Tce-Tci);
    funcprot(0);
endfunction
```

```
l = 0; // (initial guess, assumed value for fsolve
    function)
L = fsolve(l,lngth);
printf("\nLength of the tube = %f m \n",L);
```

Scilab code Exa 8.4 boiling regions

```
clear;
clc;

//Properties at (Tw+Ts)/2 = 100.5 degree celsius
deltaT1 = 1;                    //in degree celsius
p1 = 7.55e-4;             //[K^(-1) p1 is coefficient
    of cubical expansion
v1 = 0.294e-6;                  //[m^2/sec]  viscosity
     at 100.5 degree celsius
k1 = 0.683;                    //[W/m-k]thermal
    conductivity
Pr1 = 1.74;                    //Prandtl number
g = 9.81;                      //acceleration due to
    gravity
L = 0.14e-2;                   //diameter in meters
//Properties at (Tw+Ts)/2 =102.5
deltaT2 = 5;                   //in degree celsius
p2 = 7.66e-4;              //[K^(-1) p1 is coefficient
     of cubical expansion
v2 = 0.289e-6;              //[m^2/sec]  viscosity at
    102.5 degree celsius
k2 = 0.684;                   //[W/m-k]thermal
    conductivity
Pr2 = 1.71;                    //Prandtl number
//Properties at (Tw+Ts)/2 =105
deltaT3 = 10;                   //in degree celsius
p3 = 7.80e-4;              //[K^(-1) p1 is coefficient
```

```
    of cubical expansion
v3 = 0.284e-6;             //[m^2/sec]  viscosity at
   105 degree celsius
k3 = 0.684;                  //[W/m-k]thermal
   conductivity
Pr3 = 1.68;                   //Prandtl number

function[Ra]=Rayleigh_no(p,deltaT,v,Pr)
     Ra = [(p*g*deltaT*L^3)/(v^2)]*Pr;
     funcprot(0);
endfunction

function[q] = flux(k,deltaT,Rai,v)
    q=(k/L)*(deltaT)*{0.36+(0.518*Rai^(1/4))
       /[1+(0.559/v)^(9/16)]^(4/9)};
    funcprot(0);
endfunction

Ra = Rayleigh_no(p1,deltaT1,v1,Pr1);
q1 = flux(k1,deltaT1,Ra,Pr1);
printf("\n q/A = %.1f W/m^2 at (Tw-Ts)=1",q1);
Ra = Rayleigh_no(p2,deltaT2,v2,Pr2);
q2 = flux(k2,deltaT2,Ra,Pr2);
printf("\n q/A = %.1f W/m^2 at (Tw-Ts)=5",q2);
Ra = Rayleigh_no(p3,deltaT3,v3,Pr3);
q3 = flux(k3,deltaT3,Ra,Pr3);
printf("\n q/A = %.1f W/m^2 at (Tw-Ts)=10",q3);

//At 100 degree celsius
Cpl = 4.220;          //[kJ/kg]
lamda = 2257;         //[kJ/kg]
ul = 282.4e-6;        //viscosity is in kg/m-sec
sigma = 589e-4;       //Surface tension is in N/m
pl = 958.4;           //density in kg/m^3
pv =0.598;            //density of vapour in kg/m^3
deltap = pl-pv;
Prl = 1.75;           //Prandtl no. of liquid
Ksf = 0.013;
```

```
function[q1]=heat_flux(deltaT)
    q1=141.32*deltaT^3;
    funcprot(0);
endfunction

printf("\n q/A at deltaT = 5 degree celsius = %.1f W/m^2",heat_flux(5));
printf("\nq/A at deltaT = 10 degree celsius = %.1f W/m^2",heat_flux(10));
printf("\n q/A at deltaT =20 degree celsius = %.1f W/m^2",heat_flux(20));
//qi = [heat_flux(5),heat_flux(10),heat_flux(20)];
q = [q1 q2 q3];
i=1;
while i<=10
    T(i)=i;
    ql(i) = heat_flux(i);
    i=i+1;
end
plot2d([1 5 10],q);
plot2d(T,ql);
xtitle("Boiling curve","(Tw - Ts)degree celsius","Heat flux,(q/A)W/m^2");
L1 = (L/2)*[g*(pl-pv)/sigma]^(1/2);
printf("\n Peak heat flux L = %.3f ",L1);
f_L = 0.89+2.27*exp(-3.44*L1^(0.5));
printf("\n f(l) = %.4f",f_L);
q2 = f_L*{(%pi/24)*lamda*10^(3)*pv^(0.5)*[sigma*g*(pl-pv)]^(0.25)};
printf("\n q/A = %.3e W/m^2",q2);

Tn = poly([0],'Tn');
Tn1 = roots(141.32*Tn^3 - q2);
printf("\n Tw-Ts = %.1f degree celsius",Tn1(3));



printf("\n\n Minimum heat flux");
```

```
q3 = 0.09*lamda*10^3*pv*[sigma*g*(pl-pv)/(pl+pv)^(2)]^(0.25);
printf("\n q/A = %d W/m^2",q3);
printf("\n\n Stable film boiling");
Ts1 = 140;           //surface temperature in degree celsius
Ts2 = 200;           //surface temperature in degree celsius
Ts3 = 600;           //surface temperature in degree celsius
Twm1 = (140+100)/2;  //Mean film temperature
//properties of steam at 120 degree celsius and 1.013 bar
kv = 0.02558;        //thermal conductivity in W/mK
pv1 = 0.5654;        //vapor density in kg/m^3
uv=13.185*10^(-6);   //viscosity of vapour in kg/m sec
lamda1 = (2716.1-419.1)*10^(3);//Latent heat of fusion in J/kg
hc = 0.62*[(kv^3)*pv*(pl-pv)*g*lamda1/(L*uv*(140-100))]^(0.25);
printf("\n hc = %.2f W/m^2",hc);
qrad = 5.67*10^(-8)*(413^4 - 373^4)/[(1/0.9)+1-1];
printf("\n q/A due to radiation = %.2f W/m^2",qrad);
hr = qrad/(413-373);
printf("\n hr = %.2f W/m^2 K ",hr);

printf("\n Since hr<hc ");
printf("\n The total heat transfer coefficient ");
h = hc + 0.75*hr;
printf(" h = %.2f W/m^2 K",h);
printf("\n Total heat flux = %.3f W/m^2 K",h*(140-100));

hc_200 = 0.62*[(kv^3)*pv*(pl-pv)*g*lamda1/(L*uv*(200-100))]^(0.25);
qrad1 = 5.67*10^(-8)*(473^4 - 373^4)/[(1/0.9)+1-1];
hr_200 = qrad1/(200-100);
```

```
printf(”\n\n hc = %.2f W/mˆ2”,hc_200);
printf(”\n hr = %.2f W/mˆ2 K”,hr_200);
printf(”\n q/A due to radiation = %.2f W/mˆ2”,qrad1);
h_200 = hc_200 +0.75*hr_200;
printf(”\n Total heat flux = %d W/mˆ2”,h_200*100);
hc_600 = 0.62*[(kvˆ3)*pv*(pl-pv)*g*lamda1/(L*uv*(600-100))]ˆ(0.25);
qrad2 = 5.67*10ˆ(-8)*(873ˆ4 - 373ˆ4)/[(1/0.9)+1-1];
hr_600 = qrad1/(600-100)
printf(”\n\n hc = %.2f W/mˆ2”,hc_600);
printf(”\n hr = %.2f W/mˆ2 K”,hr_600);
printf(”\n q/A due to radiation = %.2f W/mˆ2”,qrad2);
```

Scilab code Exa 8.5 `Initial heat transfer rate`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 8
// Condensation and Boiling


// Example 8.5
// Page 337
printf(”Example 8.5, Page 337 \n \n”);

D = 0.02 ; // [m]
l = 0.15 ; // [m]
T = 500+273 ; // [K]
Tc = -196+273 ; // [K]
e = 0.4;
s = 5.670*10ˆ-8;
```

```
// Film boiling will occur, hence eqn 8.7.9 is applicable
Tm = (T+Tc)/2;

// Properties
k = 0.0349 ; // [W/m K]
rho = 0.80 ; // [kg/m^3]
u = 23*10^-6 ; // [kg/m s]

Cp_avg = 1.048 ; // [kJ/kg J]
rho_liq = 800 ; // [kg/m^3]
latent = 201*10^3 ; // [J/kg]

lambda = [latent + Cp_avg*(Tm-Tc)*1000]; // [J/kg]
h_c = 0.62*[((k^3)*rho*799.2*9.81*lambda)/(D*u*(T-Tc))]^(1/4); // [W/m^2 K]

// Taking the emissivity of liquid surface to be unity and using equation 3.9.1, the exchange of radiant heat flux
flux = s*(T^4-Tc^4)/(1/e+1/1-1); // [W/m^2]
h_r = flux/(T-Tc);

// Since h_r < h_c, total heat transfer coefficient is determined from eqn 8.7.11
h = h_c+3/4*h_r; // [W/m^2 K]

flux_i = h*(T-Tc);
Rate = flux_i*%pi*D*l; // [W]

printf("Initial heat flux = %f W/m^2 \n",flux_i);
printf("Initial heat transfer rate = %f W",Rate);
```

Chapter 9

Mass Transfer

Scilab code Exa 9.1 `Composition on molar basis`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.1
// Page 349
printf(”Example 9.1, Page 349 \n \n”);

w_a = 0.76 ;
w_b = 0.24 ;
m_a = 28 ; // [kg/kg mole]
m_b = 32 ; // [kg/kg mole]

x_a = (w_a/m_a)/(w_a/m_a+w_b/m_b);
x_b = (w_b/m_b)/(w_a/m_a+w_b/m_b);
printf(”The molar fractions are given by \n”);
printf(”x_a = %f\n”,x_a);
```

```
printf("x_b = %f",x_b);
```

Scilab code Exa 9.2 `Diffusion coefficient of napthalene`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.2
// Page 350
printf("Example 9.2, Page 350 \n \n");

// From Table 9.1 at 1 atm and 25 C
Dab = 0.62*10^-5 ; // [m^2/s]
// Therefore at 2 atm and 50 C
Dab2 = Dab*(1/2)*(323/298)^1.5 ;
printf("Dab at 2 atm & 50 C = %e m^2/s",Dab2);
```

Scilab code Exa 9.3.a `Rate of hydrogen diffusion`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.3(a)
```

```
// Page 352
printf(”Example 9.3(a), Page 352 \n \n”);

t = 0.04 ; // [m]
A = 2 ; // [m^2]
rho1 = 0.10 ;
rho2 = 0.01 ;
D_400 = 1.6*10^-11 ; // at 400K [m^2/s]

// Mass Diffusion in solid solution, assuming Ficks law is valid & steady state and one dimensional diffusion

// Subtituting the values in eqn 9.3.3 , At 400 K

m_400 = A*D_400*(rho1-rho2)/t; // [kg/s]
printf(”Rate of diffusion of Hydrogen at 400 K = %e kg/s \n”,m_400);
```

Scilab code Exa 9.3.b `Rate of hydrogen diffusion`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.3(b)
// Page 352
printf(”Example 9.3(b), Page 352 \n \n”);

t = 0.04 ; // [m]
A = 2 ; // [m^2]
```

```
rho1 = 0.10 ;
rho2 = 0.01 ;
D_1200 = 3.5*10^-8 ; // at 1200k [m^2/s]

// Mass Diffusion in solid solution, assuming Ficks law is valid & steady state and one dimensional diffusion

// At 1200 K
// From eqn 9.3.3

m_1200 = A*D_1200*(rho1-rho2)/t ;
printf(”(b) Rate of diffusion of Hydrogen at 1200 K = %e kg/s \n”,m_1200);
```

Scilab code Exa 9.4.a `Rate of loss of ammonia`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.4(a)
// Page 356
printf(”Example 9.4(a), Page 356 \n \n”);

L = 1 ; // [m]
D = 0.005 ; // [m]
Pa1 = 1 ; // [atm]
Pa2 = 0 ;
R = 8314 ;
T = 298 ; // [K]
```

```

// Assuming Equimolal counter diffusion
// From Table 9.1
Dab = 2.80*10^-5 ; // [m^2/s]
// Substituing in eqn 9.4.12
Na = -[Dab/(R*T)*(Pa2-Pa1)*(1.014*10^5)/L]*(%pi*(D/2)^2);
R_NH3 = Na*17 ; // [kg/s]

printf(”Na = -Nb = %e (kg mole)/m^2 s\n”,Na);
printf(”Rate at which ammonia is lost through the tube = %e kg/s \n”,R_NH3);
```

Scilab code Exa 9.4.b Rate at which air enters the tank

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.4(b)
// Page 356
printf(”Example 9.4(b), Page 356 \n \n”);

L = 1 ; // [m]
D = 0.005 ; // [m]
Pa1 = 1 ; // [atm]
Pa2 = 0 ;
R = 8314 ;
T = 298 ; // [K]

// Since the tank is large and the pressure and
```

```
   temperature at the two ends of the same tube are
   same, we are assuming Equimolal counter diffusion
// From Table 9.1
Dab = 2.80*10^-5 ; // [m^2/s]
// Substituing in eqn 9.4.12
Na = -[Dab/(R*T)*(Pa2-Pa1)*(1.014*10^5)/L]*(%pi*(D
   /2)^2);

// Since equimolal counter diffusion is taking place
Nb = - Na ;
// therefore rate at which air enters the tank
R_air = abs(Nb)*29 ; // [kg/s]

printf("Rate at which air enters the tank = %e kg/s"
   ,R_air);
```

Scilab code Exa 9.5 `Rate of evaporation`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.5
// Page 359
printf("Example 9.5, Page 359 \n \n");

// Evaporation of water, one dimensional
T_w = 20+273 ; // [K]
D = 0.04 ; // [m]
h = 0.20 ; // [m]
h_w = 0.03 ; // [m]
```

```
P = 1.014*10^5; // [Pa]
R = 8314 ; // [J/kg mole K]
P_sat = 0.02339 ; // [bar]
x_a1 = P_sat/1.014 ; // mole fraction at liq-vap interface
x_a2 = 0 ; // mole fraction at open top
c = P/(R*T_w);
// From Table 9.2
Dab = 2.422*10^-5 ; // [m^2/s]

// Substituting above values in eqn 9.4.18
flux = 0.041626*Dab/0.17*log((1-0)/(1-x_a1)); // [kg mole/m^2 s]
rate = flux*18*(%pi/4)*(D^2);

printf(”Rate of evaporation of water = %e kg/s”,rate);
```

Scilab code Exa 9.6 `Rate of evaporation`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.6
// Page 364
printf(”Example 9.6, Page 364 \n \n”);

l = 1; // length, [m]
w = 0.25; // width, [m]
```

```
T = 293 ; // Temperature, [K]
rho_infinity = 0; // [kg/m^3]
R = 8314; // [J/ kg K]

// From Table A.2
v = 15.06*10^-6; // [m^2/s]
// From Table 9.2
Dab = 2.4224*10^-5; // [m^2/s]
Re = 2.5/v;
Sc = v/Dab;
// Since Re > 3*10^5, we may assume laminar boundary
   layer
Sh = 0.664*Sc^(1/3)*Re^(1/2); // Sherwood number
h = Sh*Dab;

p_aw = 2339; // Saturation pressure of water at 20
   degree C. [N/m^2]
rho_aw = p_aw/(R/18*T); // [kg/m^3]
rho_a_inf = 0 ; // since air in the free stream is
   dry
m_h = h*(2*l*w)*(rho_aw-rho_infinity);
printf("Rate of evaporation from plate = %e kg/s",
   m_h);
```

Scilab code Exa 9.7.a Mass transfer coefficient Colburn anology

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.7(a)
```

```
// Page 366
printf(”Example 9.7(a), Page 366 \n \n”);

D = 0.04 ; // [m]
V = 1.9 ; // [m/s]

// (a) Colburn anology and Gnielinski equation
// Properties of air at 27 degree C
v = 15.718*10^-6 ; // [m^2/s]
rho = 1.177 ; // [kg/m^3]
Pr = 0.7015 ;
Cp = 1005 ; // [J/kg K]
k = 0.02646 ; // [W/m K]
// From Table 9.2
Dab = 2.54 * 10^-5 ; // [m^2/s]
Sc = v/Dab ;
Re = V*D/v;
// The flow is turbulent and eqn 9.6.5 may be applied
// let r = h/h_m
r = rho*Cp*((Sc/Pr)^(2/3));
// From Blasius equation 4.6.4a
f = 0.079*Re^(-0.25);
// Substituting this value into Gnielinski equation 5.3.2
Nu = [(f/2)*(Re-1000)*Pr]/[1+12.7*((f/2)^(1/2))*((Pr^(2/3))-1)];
h = Nu*k/D;
h_m = h/r; // [m/s]

printf(”h_m using Colburn anology and Gnielinski equation = %f \n”,h_m);
```

Scilab code Exa 9.7.b `Mass transfer coefficient Gnielinski equation`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.7(b)
// Page 366
printf("Example 9.7(b), Page 366 \n \n");

D = 0.04 ; // [m]
V = 1.9 ; // [m/s]

// (b) mess transfer correlation equivalent to the Gleilinski equation

// Properties of air at 27 degree C
v = 15.718*10^-6 ; // [m^2/s]
rho = 1.177 ; // [kg/m^3]
Pr = 0.7015 ;
Cp = 1005 ; // [J/kg K]
k = 0.02646 ; // [W/m K]
// From Table 9.2
Dab = 2.54 * 10^-5 ; // [m^2/s]
Sc = v/Dab ;
Re = V*D/v;

// From Blasius equation 4.6.4a
f = 0.079*Re^(-0.25);

// Substituting in eqn 9.6.7
Sh_D = [(f/2)*(Re-1000)*Sc]/[1+12.7*((f/2))*((Sc^(2/3))-1)];
h_m1 = Sh_D*Dab/D;

printf("(b) h_m = %f \n",h_m1);
```

Scilab code Exa 9.7.c `To show mass flux of water vapour is small`

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.7(c)
// Page 366
printf(”Example 9.7(c), Page 366 \n \n”);

D = 0.04 ; // [m]
V = 1.9 ; // [m/s]

// (c) To show that mass flux of water is very small compared to the mass flux of air flowing in the pipe
// Properties of air at 27 degree C
v = 15.718*10^-6 ; // [m^2/s]
rho = 1.177 ; // [kg/m^3]
Pr = 0.7015 ;
Cp = 1005 ; // [J/kg K]
k = 0.02646 ; // [W/m K]
// From Table 9.2
Dab = 2.54 * 10^-5 ; // [m^2/s]
Sc = v/Dab ;
Re = V*D/v;
// The flow is turbulent and eqn 9.6.5 may be applied
// let r = h/h_m
r = rho*Cp*((Sc/Pr)^(2/3));
```

```
// From Blasius equation 4.6.4a
f = 0.079*Re^(-0.25);

// From steam table
rho_aw = 1/38.77 ; // [kg/m^3]
// let X = (m_a/A)_max
X = f*rho_aw; // [kg/m^2 s]

// let Y = mass flux of air in pipe = (m/A)
Y = rho*V ; // [kg/m^2 s]
ratio = X/Y ;
percent = ratio*100;

printf("(c) (m_a/A)_max/(m_a/A) = %f percent Thus, mass flux of water is very small compared to the mass flux of air flowing in the pipe. ",percent );
```

Scilab code Exa 9.8 Mass fraction

```
clear;
clc;

// A Textbook on HEAT TRANSFER by S P SUKHATME
// Chapter 9
// Mass Transfer


// Example 9.8
// Page 369
printf("Example 9.8, Page 369 \n \n");

V = 0.5 ; // [m/s]
T_h = 30 ; // [C]
T_c = 26 ; // [C]
```

```
Tm = (T_h+T_c)/2;
// From table A.2
rho = 1.173 ; // [kg/m^3]
Cp = 1005 ; // [J/kg K]
k = 0.02654 ; // [W/m K]

alpha = k/(rho*Cp); // [m^2/s]

// From Table 9.2 at 301 K
Dab = 2.5584*10^-5 ; // [m^2/s]
lambda = 2439.2*10^3 ; // [J/kg]

// Substituting in equation 9.7.5
// let difference = rho_aw-rho_a infinity
difference = rho*Cp*((alpha/Dab)^(2/3))*(T_h-T_c)/
    lambda;

// From steam table
Psat = 3363;
rho_aw = Psat/(8314/18*299);
rho_inf = rho_aw - difference;
x = rho_inf/rho; // mole fraction of water vapour in
     air stream

PP = rho_inf*8314/18*303; // Partial pressure of
    water vapour in air stream
// From steam table partial pressure of water vapour
     at 30 C
PP_30 = 4246 ; // [N/m^2]

rel_H = PP/PP_30;
percent = rel_H*100;

printf("Relative humidity = %f i.e. %f percent ",
    rel_H,percent);
```