

Scilab Textbook Companion for
Heat Transfer
by K. A. Gavhane¹

Created by
Deepak
Bachelor of Technology
Chemical Engineering
DCRUST, Murthal
College Teacher
Ms. Sunanda
Cross-Checked by
Lavitha Pereira

November 22, 2013

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

Book Description

Title: Heat Transfer

Author: K. A. Gavhane

Publisher: Nirali Prakashan, Pune

Edition: 10

Year: 2010

ISBN: 8190639617

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes	4
2 Conduction	9
3 Convection	52
4 Radiation	100
5 Heat Exchangers	118
6 Evaporation	153

List of Scilab Codes

Exa 2.1	Thickness of insulation	9
Exa 2.2	Heat loss per metre	10
Exa 2.3	Heat Loss	11
Exa 2.4	Heat loss	12
Exa 2.5	Heat loss	13
Exa 2.7	Heat loss	14
Exa 2.8	Loss per area	14
Exa 2.9	Heat loss	15
Exa 2.10	Heat Passed	16
Exa 2.11	Insulated pipe	17
Exa 2.12	Composite brick	18
Exa 2.13	Heat flow in a pipe	19
Exa 2.15	Thickness of insulation	20
Exa 2.16	Reduction in heat loss in insulated pipe	21
Exa 2.17	Heat loss in a pipe	22
Exa 2.18	Arrangements for heat loss	23
Exa 2.19	Insulation thickness	24
Exa 2.20	Heat loss in furnace	24
Exa 2.21	Rate of heat loss in pipe	25
Exa 2.22	Heat loss from insulated steel pipe	26
Exa 2.23	Heat loss from furnace	27
Exa 2.24	Rate of heat loss	28
Exa 2.25	Thickness of insulation	29
Exa 2.26	Heat loss per metre	29
Exa 2.27	Mineral wool insulation	30
Exa 2.28	Furnace wall	31
Exa 2.29	Thickness of insulating brick	32
Exa 2.30	Heat flow through furnace wall	33

Exa 2.31	Heat loss in pipe	34
Exa 2.32	Heat flux through layers	35
Exa 2.33	Conductive conductance furnace wall	35
Exa 2.34	Critical radius of insulation	37
Exa 2.36	Critical radius of pipe	38
Exa 2.37	Time required for steel ball	38
Exa 2.38	Steel ball quenched	39
Exa 2.39	Ball plunged in a medium	40
Exa 2.40	Slab temperature suddenly lowered	41
Exa 2.41	Flow over a flat plate	42
Exa 2.42	Stainless steel rod immersed in water	42
Exa 2.43	Chromel alumel thermocouple	43
Exa 2.44	Thermocouple junction	44
Exa 2.45	Batch reactor	45
Exa 2.46	Heat dissipation by aluminium rod	46
Exa 2.47	Aluminium fin efficiency	47
Exa 2.49	Pin fins	48
Exa 2.50	Metallic wall surrounded by oil and water	49
Exa 2.51	Brass wall	50
Exa 3.1	Boundary layer thickness	52
Exa 3.2	Boundary layer thickness of plate	53
Exa 3.3	Thickness of hydrodynamic boundary layer	54
Exa 3.4	Flat plate boundary layer	54
Exa 3.5	Rate of heat removed from plate	55
Exa 3.6	Heat removed from plate	56
Exa 3.7	Local heat transfer coefficient	57
Exa 3.8	Width of plate	58
Exa 3.9	Heat transferred in flat plate	59
Exa 3.10	Rate of heat transferred in turbulent flow	60
Exa 3.11	Heat transfer from plate in unit direction	61
Exa 3.12	Heat lost by sphere	62
Exa 3.13	Heat lost by sphere	62
Exa 3.14	Percent power lost in bulb	63
Exa 3.15	Heat lost by cylinder	64
Exa 3.16	Heat transfer in tube	64
Exa 3.17	Heat transfer coefficient	65
Exa 3.18	Heat transfer coefficient in heated tube	66
Exa 3.19	h of water flowing in tube	67

Exa 3.20	Overall heat transfer coefficient	68
Exa 3.21	Number of tubes in exchanger	69
Exa 3.22	Convective film coefficient	71
Exa 3.23	Length of tube	72
Exa 3.24	Cooling coil	73
Exa 3.25	Outlet temperature of water	73
Exa 3.26	Inside heat transfer coefficient	74
Exa 3.27	Film heat transfer coefficient	75
Exa 3.28	Area of exchanger	76
Exa 3.29	Natural and forced convection	77
Exa 3.30	Natural convection	78
Exa 3.31	Free convection in vertical pipe	79
Exa 3.32	Heat loss per unit length	80
Exa 3.33	Free convection in pipe	81
Exa 3.34	Free convection in plate	81
Exa 3.35	Heat transfer from disc	82
Exa 3.36	Rate of heat input to plate	83
Exa 3.37	Two cases in disc	84
Exa 3.38	Total heat loss in a pipe	86
Exa 3.39	Heat loss by free convection	87
Exa 3.40	Heat loss from cube	87
Exa 3.41	Plate exposed to heat	88
Exa 3.42	Nucleate poolboiling	90
Exa 3.43	Peak Heat flux	90
Exa 3.44	Stable film pool boiling	91
Exa 3.45	Heat transfer in tube	92
Exa 3.46	Nucleat boiling and heat flux	93
Exa 3.47	Dry steam condensate	93
Exa 3.48	Laminar Condensate film	94
Exa 3.49	Saturated vapour condensate in array	95
Exa 3.50	Mass rate of steam condensation	96
Exa 3.51	Saturated tube condensate in a wall	97
Exa 3.52	Condensation rate	98
Exa 3.53	Condensation on vertical plate	98
Exa 4.1	Heat loss by radiaiton	100
Exa 4.2	Radiation from unlagged steam pipe	100
Exa 4.3	Interchange of radiation energy	101
Exa 4.4	Heat loss in unlagged steam pipe	101

Exa 4.5	Loss from horizontal pipe	102
Exa 4.6	Heat loss by radiation in tube	103
Exa 4.7	Net radiant interchange	103
Exa 4.8	Radiant interchange between plates	104
Exa 4.9	Heat loss from thermflask	104
Exa 4.10	Diwar flask	105
Exa 4.11	Heat flow due to radiation	106
Exa 4.12	Heat exchange between concentric shell	106
Exa 4.13	Evaporation in concenric vessels	107
Exa 4.15	infinitely long plates	108
Exa 4.16	Heat exchange between parallel plates	109
Exa 4.17	Thermal radiation in pipe	110
Exa 4.18	Heat transfer in concentric tube	111
Exa 4.19	Heat exchange between black plates	111
Exa 4.20	Radiation shield	112
Exa 4.21	Heat transfer with radiaiton shield	113
Exa 4.22	Radiaiton shape factor	114
Exa 4.23	Radiation loss in plates	115
Exa 4.24	Concentric tube	116
Exa 5.1	Harpin exchanger	118
Exa 5.2	Length of pipe	120
Exa 5.3	Double pipe heat exchanger	121
Exa 5.4	Parallel flow arrangement	123
Exa 5.5	Counter flow exchanger	124
Exa 5.6	LMTD approach	125
Exa 5.7	Shell and tube exchanger	126
Exa 5.8	Order of Scale resistance	127
Exa 5.9	Length of tube required	128
Exa 5.10	Suitability of Exchanger	130
Exa 5.11	Number of tubes required	131
Exa 5.12	Shell and tube heat exchanger	133
Exa 5.13	Length of pipe in Exchanger	135
Exa 5.14	Dirt factor	136
Exa 5.15	Heat transfer area	139
Exa 5.16	Oil Cooler	140
Exa 5.17	Countercurrent flow heat exchanger	141
Exa 5.18	Vertical Exchanger	142
Exa 5.19	Countercurrent Heat Exchanger	144

Exa 5.20	Number of tube side pass	145
Exa 5.21	Number of tubes passes	146
Exa 5.22	Outlet temperature for hot and cold fluids	147
Exa 5.23	Counterflow concentric heat exchanger	149
Exa 5.24	Number of tubes required	150
Exa 5.25	Parallel and Countercurrent flow	151
Exa 6.1	Boiling point Elevation	153
Exa 6.2	Capacity of evaporator	153
Exa 6.3	Economy of Evaporator	154
Exa 6.4	Steam economy	155
Exa 6.5	Evaporator economy	157
Exa 6.6	Single effect Evaporator	158
Exa 6.7	Single effect evaporator reduced pressure	159
Exa 6.8	Mass flow rate	160
Exa 6.9	Heat load in single effect evaporator	160
Exa 6.10	Triple effect evaporator	161
Exa 6.11	Double effect evaporator	162
Exa 6.12	lye in Triple effect evaporator	165
Exa 6.13	Triple effect unit	168
Exa 6.14	Quadruple effect evaporator	169
Exa 6.15	Single effect Calendria	171
Exa 6.16	Single effect evaporator	173

Chapter 2

Conduction

Scilab code Exa 2.1 Thickness of insulation

```
1  clc;
2  clear;
3  printf("Example 2.1 \n Page no. 2.18\n Part-(a)")
4  A=1; //sq metre
5  printf("Area of heat transfer ,A=%f m^2\n",A)
6  Q=450; // W/ sq mtre
7  printf("Rate of heat loss/unit area=%f W/m^2\n",Q)
8  dT=400; // K
9  printf("Temperature difference across insulation
        layer\t,dT=%f K\n",dT)
10 k=0.11 //W/(m.K)
11 printf("For asbestos ,k=%f\n",k)
12 //Q=(k* A*dT)/x
13 x=(k*A*dT)/Q
14 X=x*1000;
15
16 //for fire clay insulation
17 k=0.84; // W/(m.K)
18 printf("For fire clay insulation ,k=%f W/(m.K)\n",k);
19 x=(k*A*dT)/Q;
20 X=x*1000;
```

```

21 printf("Ans.(A). Thickness of asbestos is: %f m=%f mm
    \n",x,X)
22 printf("Ans.(B) Thickness of fire clay insulation is:
    %f m =%f mm\n",x,X)

```

Scilab code Exa 2.2 Heat loss per metre

```

1
2 clc;
3 printf("Example 2.2,\nPage no.2.18\n");
4 L=1 // m
5 printf("Length of ppipe ,L = %f m\n",L);
6 r1=(50/2) // in mm
7 r1=r1/1000 // in m
8 printf(" r1=%f m\n",r1);
9 r2=(25+3)/1000 // m
10 printf(" r2=%f m\n",r2)
11 rm1=(r2-r1)/log(r2/r1);
12 printf(" rm1=%f m\n",rm1)
13 k1=45 //W/(m.K)
14 R1=(r2-r1)/(k1*(2*pi*rm1*L)) // K/W
15 printf(" Thermal resistance of wall pipe=R1=%f K/W\n"
    ,R1);
16 printf(" For inner lagging:\n") ;
17 k2=0.08 //W/(m.K)
18 ri1=0.028 //m
19 ri2=(ri1+r1) // m
20 rmi1=(ri2-ri1)/log(ri2/ri1)
21 R2=(ri2-ri1)/(k2*2*pi*rmi1*L)
22 printf(" Thermal resistance of inner lagging=R2=%f K/
    W",R2);
23 printf(" For outer lagging:\n") ;
24 k3=0.04 //W/(m.K)
25 ro1=0.053 //m
26 ro2=(ro1+0.04) // m

```

```

27 rmo1=(ro2-ro1)/log(ro2/ro1)
28 R3=(ro2-ro1)/(k3*2*pi*rmo1*L)
29 printf("Thermal resistance of inner lagging=R2=%f K/
      W\n",R3);
30 R=R1+R2+R3
31 Ti=550 //K //inside
32 To=330 //K // outside
33 dT=Ti-To; //Temperature difference
34 Q=dT/R
35 printf("Rate of heat loss per metre of pipe ,Q=%f W/m
      ",Q)

```

Scilab code Exa 2.3 Heat Loss

```

1 clear;
2 clc;
3 printf("Example 2.3")
4 //Given
5 r1=44 // [mm]
6 r1=r1/1000 // [m]
7 r2=0.094 // [m]
8 r3=0.124 // [m]
9 T1=623 //Temperature at outer surface of wall in [K]
10 T3=313 //Temperature at outer surface of outer
      insulation [K]
11 k1=0.087 //Thermal conductivity of insulation layer
      1..in [W/m.K]
12 k2=0.064 //Thermal conductivity of insulation layer
      2 [W/m.K]
13 l=1 // Length of pipe [m]
14 rm1=(r2-r1)/log(r2/r1) //log mean radius of
      insulation layer 1 [m]
15 rm2=(r3-r2)/log(r3/r2) //log mean radius of
      insulation layer 2[m]
16 //Putting values in following eqn:

```

```

17 Q= (T1-T3)/((r2-r1)/(k1*2*%pi*rm1*1)+(r3-r2)/(k2*2*
    %pi*rm2*1));
18 printf("Heat loss per meter pipe is %f W/m",Q)

```

Scilab code Exa 2.4 Heat loss

```

1  clc;
2  clear;
3  //Example 2.4
4  printf("Example 2.4")
5  //Given
6  A=1 //Heat transfer area [sq m]
7  x1=0.229 // thickness of fire brick in [m]
8  x2=0.115 // thickness of insulating brick in [m]
9  x3=0.229 // thickness of building brick in [m]
10 k1=6.05 //thermal conductivity of fir brick [W/(m.
    K)]
11 k2=0.581 //thermal conductivity of insulating brick
    [W/m.K]
12 k3=2.33 //thermal conductivity of building brick
    [W/m.K]
13 T1=1223 // inside temperature [K]
14 T2=323 // Outside temperature [K]
15 dT=T1-T2 //Overall temp drop [K]
16 R1=(x1/k1*A) //thermal resistance 1
17 R2=(x2/k2*A) // Thermal resistance 2
18 R3=(x3/k3*A) //Thermal resistance 3
19 Q=dT/(R1+R2+R3) //w/SQ m
20 Ta=-((Q*R1)-T1) //from Q1=Q=(T1-Ta)/(x1/k1*A)
21 //Similarly
22 Tb=(Q*R3)+T2;
23 printf("Interface temperature:\n i-Between FB-IB=%f
    K \n ii-Between IB-PB=%fK",Ta,Tb);

```

Scilab code Exa 2.5 Heat loss

```
1  clc ;
2  clear ;
3  //Example 2.5
4  printf("Example 2.5\nPage 2.23")
5  //Given
6  A=1; //let [sq m]
7  x1=0.23; //thickness of fir brick layer [m]
8  x2=0.115; // [m]
9  x3=0.23; //[m]
10 T1=1213; //Temperature of furnace [K]
11 T2=318; //Temperature of furnace [K]
12 dT=T1-T2; //[K]
13 k1=6.047; //W/(m.K) (fire brick)
14 k2=0.581; //W/(m.K) (insulating brick)
15 k3=2.33; //W/(m.K) (building brick)
16 Q_by_A=dT/((x1/k1)+(x2/k2)+(x3/k3)) //Heat lost per
    unit Area in Watt
17
18 R1=(x1/k1) //Thermal resistance
19 R1=0.04 //Approximate
20 R2=(x2/k2)
21 R2=0.2025 //Approximate
22 R3=(x3/k3)
23 R3=0.1 //Approximate
24 Ta=T1-((dT*R1)/(R1+R2+R3))
25 Tb=((dT*R3)/(R1+R2+R3))+T2
26 Tb=565 //Approximation
27 printf("Answer:Heat loss per unit area is %f W=%f
    J/s\n",Q_by_A,Q_by_A);
28 printf("Answer:\n Ta=%f K =Temperature at the
    interface between fire brick and insulating brick
    \n Tb=%d K Temperature at the interface between
```

insulating and building brick\n",Ta,Tb)

Scilab code Exa 2.7 Heat loss

```
1  clc
2  printf("Example 2.7,Page no 2/26 \n");
3  printf("Part -(a)\n");
4  A=1; // sq metre
5  x1=114 // mm
6  x1=x1/1000 // metre
7  k1=0.138 // W/(m.K)
8  R1= x1/(k1*A)
9  x2=229 //mm
10 x2= x2/1000 // metre
11 k2=1.38 // W/m.K
12 R2=x2/(k2*A)
13 dT=1033-349
14 //Heat loss
15 Q=dT/(R1+R2)
16 printf("ANSWER:Heat loss from 1 sq metre wall=%f W"
    ,Q);
17 printf("Part (b)\n");
18 //contact resistance=cr
19 cr=0.09 //K/W
20 R=R1+R2+cr
21 Q=dT/R
22 printf("ANSWER:Heat loss from 1 sq metre when
    resistance present=%f W",Q);
```

Scilab code Exa 2.8 Loss per area

```
1  clear;
2  clc;
```

```

3 //Example 2.8
4 printf("Example 2.8 \n")
5 //Given:
6 x1=0.02 // [m]
7 x2=0.01 // [m]
8 x3=0.02 // [m]
9 k1=0.105 //W/(m.k)
10 k3=k1 //W/(m.K)
11 k2=0.041 //W/(m.K)
12 T1=303
13 T2=263
14 dT=T1-T2 // [K]
15 Q_by_A=dT/((x1/k1)+(x2/k2)+(x3/k3))
16 R=0.625 //K/W
17 Tx=293 //K
18 Rx=0.9524 //K/W
19 x=R*(T1-Tx)/(dT*Rx)
20 x=x*100 //mm
21 printf("The temperature of 293 K will be reached at
    point %f mm from the outermost wall surface of
    the ice-box",x)

```

Scilab code Exa 2.9 Heat loss

```

1 clc
2 printf("Example 2.9, Page 2.28\n");
3 //Given
4 ID=50 //mm;
5 dT=(573-303);
6 printf("Internal diameter ,ID=%f mm",ID);
7 r1=ID/2 //mm
8 r1=r1/1000 // metres
9 OD=150 // mm
10 printf("Outer diameter ,OD=%f mm",OD);
11 r2=OD/2 // mm

```



```

12 r2=75/1000 // m
13 //Thermal conductivity
14 k=17.45 // W/(m.K)
15 //Solution
16 printf("Q/A=dT/(r2-r1)/k\n");
17 A1=4*pi*(r1^2);
18 A2=4*pi*(r2^2);
19 A=sqrt(A1*A2)
20 Q=(A*k*dT)/(r2-r1)
21 printf("ANSWER:\nHeat loss=Q=%f W",Q);

```

Scilab code Exa 2.10 Heat Passed

```

1 clear;
2 clc;
3 //Example 2.10
4 printf("Example 2.10")
5 A= 1 //sq m
6 x1=0.15
7 x2=0.01
8 x4=0.15
9 T1=973 // [K]
10 T2=288 // [K]
11 dT=T1-T2 // [K]
12 //Thermal conductivities
13 k1=1.75
14 k2=16.86
15 k3=0.033
16 k4=5.23
17 //in absence of air gap,sum of thermal resistances
18 sR=(x1/k1*A)+(x2/k2*A)+(x4/k4*A)
19 Q= dT/sR
20 printf("Heat lost per sq meter is %d W/sq m",Q);
21 //When heat loss ,Q=1163,then new resistance =sR1
22 Q1=1163 // [W/sq m]

```

```

23 sR1=dT/Q1
24 //width of air gap be w then
25 w=(sR1-sR)*k3*A // [m]
26 w=w*1000 //in [mm]
27 printf("Width of air gap is %f mm",w);

```

Scilab code Exa 2.11 Insulated pipe

```

1 clear;
2 clc;
3 //Example 2.11
4 printf("Example 2.11");
5 d1=300 // [mm]
6 r1=d1/2 // [mm]
7 r1=r1/1000 // [m]
8 r2=r1+0.05 // [m]
9 r3= r2+0.04 // [m]
10 x1=0.05 // [m]
11 x2=0.04 // [m]
12 k1=0.105 //W/(m.K)
13 k2=0.07 //W/(m.K)
14 rm1=(r2-r1)/log(r2/r1); // [m]
15 rm2=(r3-r2)/log(r3/r2); // [m]
16 L=1 //let
17 A1=%pi*rm1*L //let L=1
18 R1=x1/(k1*A1);
19 A2=%pi*rm2*L
20 R2=x2/(k2*A2)
21 T1=623 // [K]
22 T2=323 // [K]
23 dT=T1-T2 // [K]
24 //Part a
25 Q_by_L= dT/(R1+R2) //Heat loss
26 printf("Heat loss is %f W/m",Q_by_L);
27 //Part b:

```

```

28 P=2*%pi*(r1+x1+x2) // [m]
29 Q_by_L_peri=Q_by_L/P // [W/sq m]
30
31 printf("Heat lost per sq meter of outer insulation
    is %f W/sq m",Q_by_L_peri);
32 R1=x1/(k1*A1)
33 sR=0.871+0.827
34 dT1=dT*R1/sR
35 printf("Temperature between two layers of insulation
    =%f K", (T1-dT1) );

```

Scilab code Exa 2.12 Composite brick

```

1 //Example 2.12
2 clear;
3 clc;
4 printf("Example 2.12\n")
5 //Given
6 x1=0.01 // [m]
7 x2=0.15 // [m]
8 x3=0.15 // [m]
9 T1=973 // [K]
10 T2=423 // [K]
11 dT=T1-T2;
12 //Thermal conductivities
13 k1=16.86 // [W/m.K]
14 k2=1.75 // [W/m.K]
15 k3=5.23 // [W/m.K]
16 k_air=0.0337 // [W/m.K]
17 A=1 // [sq m]
18 sigma_R=(x1/(k1*A)+x2/(k2*A)+x3/(k3*A))
19 Q=dT/sigma_R //Heat flow in [W]
20 Tm= Q*x3/k3 //Temperature drop in magnesite brick
21 //Interface temperature=iT
22 iT=T2+Tm // [K]

```

```

23 sigma_xbyk= A*dT/1163 //with air gap for reducing
    heat loss to 1163 per sq m
24 x_by_k=sigma_xbyk-sigma_R //x/k for air
25 t=x_by_k*k_air
26 t=t*1000;
27 printf("Width of the air gap is %f mm",t);

```

Scilab code Exa 2.13 Heat flow in a pipe

```

1 //Example 2.13
2 printf("Example 2.13 \n");
3
4 L=1 //assume [m]
5
6 k1=43.03 // [W/(m.K)]
7
8 k2=0.07 // (W/m.K)
9
10 T1=423 //inside temperature [K]
11
12 T2=305 // [K]
13
14 r1=0.0525 // [mm]
15
16 r2=0.0575; // [m]
17
18 r3=0.1075 // [m]
19 //r3=r3/1000; // [m]
20 Q=(2*pi*L*(T1-T2))/(((log(r2/r1))/k1)+((log(r3/r2))
    /k2)); //Heat loss per metre
21
22 printf("Heat flow per metre of pipe is %f W/m",Q);
23
24 printf("Part 2\n");
25 //T=Temperature of outer surface

```

```

26 T=T1-(Q*log(r2/r1))/(k1*2*%pi*L);
27
28 printf("Temperature at outer surface of steel pipe:
    %f K",T);
29
30 printf("\nPart iii\n");
31 id=0.105 //inside diametre in [m]
32
33 A=%pi*id*1 //inside area in [sq m]
34
35 C=Q/(A*(T1-T2)); //conductance per length
36
37 printf("Conductance per m length based on inside
    area is %f W/K",C)

```

Scilab code Exa 2.15 Thickness of insulation

```

1 //Example 2.15
2 printf("Example 2.15 \n")
3 A=1 // [sq m]
4 x1=0.1 //m
5 x2=0.04
6 k1=0.7
7 k2=0.48
8 sigma=x1/(k1*A)+x2/(k2*A) //K/W
9 //Q=4.42*dT
10 //Q=dT/sigma
11 //with rockwool insulation added , Q_dash=0.75*Q
12 k3=0.065 // W/(m.K)
13 //Q_dash=dT/sigma+x3/k3*A
14 //On solving Q and Q_dash we get
15 x3=((1/(0.75*4.42))-sigma)*k3 // [m]
16 x3=x3*1000 // [mm]
17 printf("Thickness of rockwool insulation required=
    %f mm",x3)

```

Scilab code Exa 2.16 Reduction in heat loss in insulated pipe

```
1  clc;
2  clear;
3  //Example 2.16,Page no 2.36
4  d1=40; // Diameter of pipe [mm]
5  r1=(d1/2)/1000 //Outside radius in [m]
6  t1=20; //Insulation 1 thickness in [mm]
7  t1=t1/1000 // [m]
8  t2=t1; //Insulation 2 thickness in [m]
9  r2=r1+t1; //radius after 1st insulation in [m]
10 r3=r2+t2; //Radius after second insulation in [m
    ]
11
12 //Since Scilab does not handles symbolic constants ,
    we will assume some values:
13 //(1)
14 printf("Let the layer M-1 be nearer to the surface")
15 L=1; // [m]
16 T1=10; //Temperature of inner surface of pipe
    [K]
17 T2=5; //Temperature of outer surface of
    insulation [K]
18 k=1; //Thermal conductivity
19 k1=k; //For M-1 material
20 k2=3*k; //For material M-2
21 Q1=(T1-T2)/(log(r2/r1)/(2*pi*L*k1)+log(r3/r2)/(2*
    pi*L*k2))
22
23 //(2)
24 printf("Let the layer of material M-2 be nearer to
    the surface");
25 Q2=(T1-T2)/(log(r2/r1)/(2*pi*L*k2)+log(r3/r2)/(2*
    pi*L*k1))
```

```

26 printf("Q1=%f and Q2= %f \n For dummy variables
    unity...\nFor any value of k,T1 and T2,Q1 is
    always less than Q2",Q1,Q2);
27 printf("\n So,M-1 near the surface is advisable(i.e
    Arrangement one will result i ,ess heat loss\n)")
    ;
28 per_red=(Q2-Q1)*100/Q2
29 printf("Percent reduction in heat loss is %f percent
    ",per_red)
30 printf("\nNOTE: Slight variation in answers due to
    less precise calculation in book.If performed
    manually ,this answer stands to be correct")

```

Scilab code Exa 2.17 Heat loss in a pipe

```

1 //Example2.17
2 T1=523 // [K]
3 T2=323 // [K]
4 r1=0.05 // [m]
5 r2=0.055 // [m]
6 r3=0.105 // [m]
7 r4=0.155 // [m]
8 k1=50 // [W/(m.K) ]
9 k2=0.06 // [W/(m.K) ]
10 k3=0.12 //W/(m.K)
11 //CASE 1
12 Q_by_L1=2*pi*(T1-T2)/((log(r2/r1))/k1+(log(r3/r2)
    )/k2+(log(r4/r3))/k3) // [W/m]
13 printf("Heat loss=%f W/m", Q_by_L1)
14 //Case 2
15 Q_by_L2=2*pi*(T1-T2)/((log(r2/r1))/k1+(log(r3/r2)
    )/k3+(log(r4/r3))/k2)
16 perct=(Q_by_L2-Q_by_L1)*100/Q_by_L1
17 printf("If order is changed then heat loss=%f W/m"
    ,Q_by_L2)

```

```

18  printf("\n loss of heat is increased by %f percent
      by putting material with higher thermal
      conductivity near the pipe surface",perct)

```

Scilab code Exa 2.18 Arrangements for heat loss

```

1
2  clc;
3  clear;
4  //Example 2.18,Page no 2.38
5  //Given
6  //Assume:
7  L=1      // [m]
8  r1=0.10  // [m]      Outside radius od pipe
9  ia=0.025 //inner insulaiton [m]
10
11 r2=r1+ia  //Outer radius of inner insulation
12 r3=r2+ia  //Outer radius of outer insulation
13 //CASE 1:'a' near the pipe surface
14 //let k1=1
15 k1=1;     //Thermal conductivity of A[W/m.K]
16 //and k2=3k1=3
17 k2=3;     //Thermal conductivity of B[W/m.K]
18 //Let dT=1
19 dT=1
20 Q1=dT/(log(r2/r1)/(2*%pi*k1*L)+log(r3/r2)/(2*%pi*k2*
      L))
21 Q1=22.12  // Approximate
22 //CASE 2:'b' near the pipe surface
23 Q2=dT/(log(r2/r1)/(2*%pi*k2*L)+log(r3/r2)/(2*%pi*k1*
      L))
24 Q2=24.39  // Approximation
25 printf("ANSWER-(i)\nQ1=%f W \nQ2= %f W \nQ1 is less
      than Q2.i.e arrangement A near the pipe surface
      and B as outer layer gives less heat loss\n",Q1,

```



```

    Q2);
26 percent=(Q2-Q1)*100/Q1;    //percent reduction in
    heat loss
27 printf("\nANSWER-(ii) \nPercent reduction in heat
    loss (with near the pipe surface)=%f percent",
    percent);

```

Scilab code Exa 2.19 Insulation thickness

```

1  clc
2  clear
3  printf("Example 2.19. Page no.2.39")
4  //Given
5  x1=0.224 // m
6  k1=1.3 // W/(m.K)
7  k2=0.346 // W/(m.K)
8  T1=1588 // K
9  T2= 299 // K
10 QA=1830 // W/ sq metre //heat loss
11 //solution
12 printf("Q/A=(T1-T2)/x1/k1+x2/k2");
13 x2=k2*((T1-T2)*1/(QA)-(x1/k1))
14 x2=x2*1000;
15 printf("Thickness of insulation=%f mm",x2)

```

Scilab code Exa 2.20 Heat loss in furnace

```

1 //Example 2.20
2 //Given
3 //for clay
4 k1=0.533 // [W/(m.K)]
5 //for red brick
6 k2=0.7 // [W/m.K]

```

```

7 //Case 1
8 A=1 //Area
9 x1=0.125 // [m]
10 x2=0.5 // [m]
11 // Resistances
12 r1=x1/(k1*A) //Res of fire clay [K/W]
13 r2=x2/(k2*A) //Res of red brick [K/W]
14 r=r1+r2
15 //Temperatures
16 T1=1373 // [K]
17 T2=323 // [K]
18 Q=(T1-T2)/r // [W/sq m]
19 Tdash=T1-Q*r1 // [K]
20 //Case2
21 // Heat loss must remain unchanged, Thickness of
red brick also reduces to its half
22 x3=x2/2 // [m]
23 r3=x3/(k2*A) // [K/W]
24 Tdd= T2+(Q*r3) // [K]
25 //Thickness of diatomite be x2, km be mean
conductivity
26 Tm=(Tdash+Tdd)/2 // [K]
27 km=0.113+(0.00016*Tm) // [W/(m.K)]
28 x2=km*A*(Tdash-Tdd)/Q // [m]
29 x2=x2*1000 // [mm]
30 printf("Thickness of diatomite layer=%f mm",x2)

```

Scilab code Exa 2.21 Rate of heat loss in pipe

```

1 //Exaample2.21
2 //Given
3 k1=0.7 //common brick W/((m.K)
4 k2=0.48 //gypsum layer [W/(m.K)
5 k3=0.065 //Rockwool [W/m.K]
6 //Heat loss with insulatiob will be 20% of without

```

```

        insulation
7  A=1          //sq m
8  x1=0.1      //[m]
9  x2=0.04    //[m]
10 R1=x1/(k1*A) //K/W
11 R2=x2/(k2*A) //K/W
12 R=R1+R2     //K/W
13 //R3=x3/(k3*A)
14 QbyQd=0.2
15 sigRbyRd=QbyQd
16 x3=(R/QbyQd-R)/15.4 //m
17 x3=x3*1000    //[mm]
18 printf("Thickness of rockwool insulation =%f mm",x3)

```

Scilab code Exa 2.22 Heat loss from insulated steel pipe

```

1  clc;
2  clear;
3  //Example 2.22
4  Ts=451; //Steam temperature in [K]
5  Ta=294; //Air temperature in [K]
6  Di=25; //Internal diameter of pipe [mm]
7  Di=Di/1000; // [m]
8  od=33; //Outer diameter of pipe [mm]
9  od=od/1000; // [m]
10 hi=5678; //Inside heat transfer coefficient [W/(m^2.
    K)]
11 ho=11.36; //Outsideheat transfer coefficient [W/(sq
    m.K)]
12 xw=(od-Di)/2; //Thickness of steel pipe [m]
13 k2=44.97; //k for steel in W/(m.K)
14 k3=0.175; //k for rockwool in W/(m.K)
15 ti=38/1000; //thickness of insulation in [
    m]
16 r1=Di/2; // [m]

```

```

17 r2=od/2; // [m]
18 rm1=(r2-r1)/log(r2/r1); // [m]
19 r3=r2+ti; // [m]
20 rm2=(r3-r2)/log(r3/r2); // [m]
21 Dm1=2*rm1; // [m]
22 Dm2=2*rm2; // [m]
23 //Rate of heat loss = dT/(sigma_R)
24 L=1; // [m]
25 R1=1/(hi*%pi*Di*L); // [K/W]
26 R2=xw/(k2*%pi*Dm1*L);
27 R3=(r3-r2)/(k3*%pi*Dm2*L);
28 Do=(od+2*ti) ; // [mm]
29 R4=1/(ho*%pi*Do*L); // [m]
30 sigma_R=R1+R2+R3+R4;
31 //Heat loss
32 dT=Ts-Ta; // [K]
33 Q=dT/sigma_R; //Heat loss [W/m]
34 printf("\nAns:Rate of heat loss is %f W/m",Q);
35 printf("\n NOTE:Slight variation in final answer due
to lack of precision in calculation of R1,R2,R3
and R4.In book an approximate values of these is
taken\n ")

```

Scilab code Exa 2.23 Heat loss from furnace

```

1 clc;
2 //Example 2.23
3 T1=913 // [K]
4 T=513 // [K]
5 T2=313 // [K]
6 //Q=(T1-T)/(x/(k*A))
7 //Q=(T-T2)/(1/(h*A))
8 //x=2k/h
9 //Q=(T1-T2)/(x/(kA)+1/(h*A))
10 // Therefore ,Q=hA/3*(T1-T2)

```

```

11 //With increase in thickness(100%)
12 //x1=4*k/h
13 //Q2=(T1-T2)/(x1/k*A+1/(h*A))
14 //Q2=(h*A)/5)*(T1-T2)
15 //Now
16 h=1; //Assume
17 A=1; //Assume for calculation
18 Q1=(h*A/3)*(T1-T2)
19 Q2=((h*A)/5)*(T1-T2)
20 percent=(Q1-Q2)*100/Q1 //Percent reduction in
    heat loss
21 printf("\nTherefore, Percentage reduction in heat
    loss is %d percent",percent);

```

Scilab code Exa 2.24 Rate of heat loss

```

1 clc;
2 clear;
3 printf("Example 2.24\n Page no. 2.47");
4 //given
5 L=1//m
6 thp=2//Thickness of pipe; in mm
7 thi=10//Thickness of insulation; in mm
8 T1=373//K
9 T2=298//K
10 id=30//mm
11 r1=id/2//mm
12 r2=r1+thp//mm
13 r3=r2+thi//mm
14 //In S.I units
15 r1=r1/1000 //m
16 r2=r2/1000 //m
17 r3=r3/1000 //m
18 k1=17.44//W/(m.K)
19 k2=0.58//W/(m.K)

```

```

20 hi=11.63//W/(sq m.K)
21 ho=11.63//W/(sq m.K)
22 //Solution
23 Q=(2*%pi*L*(T1-T2))/(1/(r1*hi)+(log(r2/r1))/k1+((log
    (r3/r2))/k2)+(1/(0.02*ho)))
24 printf("ANSWER: \n Rate of heat loss ,Q=%f W",Q);

```

Scilab code Exa 2.25 Thickness of insulation

```

1  clc;
2  clear;
3  //Exampler 2.25
4  h=8.5 ;           // [W/sq m.K]
5  dT=175 ;         // [K]
6  r2=0.0167;       // [m]
7  Q_by_1=h*2*%pi*r2*dT // [W/m]
8  k=0.07 ;         //For insulating material in [W/m.
    K]
9  //for insulated pipe--50% reduction in heat loss
10 Q_by_11=0.5*Q_by_1 // [w/m]
11 def(' [x]=f(r3)', 'x=Q_by_11-dT/((log(r3/r2))/(2*%pi*
    k)+1/(2*%pi*r3*h))')
12
13 //by trial and error method we get:
14 r3=fsolve(0.05,f)
15 t=r3-r2 //thickness of insulation in [m]
16 printf('\n Hence, required thickness of insulation is
    %f m=%f mm or %d m",t,t*1000,round(t*1000));

```

Scilab code Exa 2.26 Heat loss per metre

```

1
2 //Example 2.26

```

```

3 //Calculate heat loss per metre length
4 //Given
5 id=0.1 //internal diameter in[m]
6 od=0.12 //outer diameter in [m]
7 T1=358 //Temperature of fluid [K]
8 T2=298 //Temperature of surrounding [K]
9 t=0.03 //thickness of insulation [m]
10 k1=58 // [W/m.K]
11 k2=0.2 //W/(m.K) insulating material
12 h1=720 //inside heat transfer coeff [W/sq m .K]
13 h2=9 //W/sq m.K
14 r1=id/2 // [m]
15 r2=od/2 // [m]
16 r3=r2+t // [m]
17 //Heat loss per meter=Q_by_L
18 Q_by_L=(T1-T2)/(1/(2*pi*r1*h1)+log(r2/r1)/(2*pi*k1
    )+log(r3/r2)/(2*pi*k2)+1/(2*pi*r3*h2)); //W/m
19 printf("Heat loss per metre length of pipe=%f W",
    Q_by_L)

```

Scilab code Exa 2.27 Mineral wool insulation

```

1
2 clc;
3 clear;
4 //Example 2.26
5 //Given:
6 T1=573; // [K]
7 T2=323; // [K]
8 T3=298; // [K]
9 h1=29; // Outside heat transfer
    coefficients [W/sq m.K]
10 h2=12; // [W/sq m.K]
11 r1=0.047; //Internal radius [m]
12 r2=0.05; //Outer radius [m]

```

```

13 k1=58 ; // [W/m.K]
14 k2=0.052; // [W/m.K]
15 //Q=(T1-T2)/(1/(r1*h1)+log(r2/r1)/k1+log(r3/r2)/k2)
    =(T2-T3)/(1/(r3*h2))
16 def f(' [x]=f(r3)', 'x=(T1-T2)/(1/(r1*h1)+log(r2/r1)/k1
    +log(r3/r2)/k2)-(T2-T3)/(1/(r3*h2))')
17 //by trial and error method :
18 r3=fsolve(0.05,f)
19 t=r3-r2 //Thickness of insulation in [m]
20 //Q=h2*2*%pi*r3*L*(T2-T3)
21 Q_by_1=h2*2*%pi*r3*(T2-T3) // [W/m]
22 printf("\n Thicknesss of insulation is %d mm \n Rate
    of heat loss per unit length is %f W/m",round(t
    *1000),Q_by_1);

```

Scilab code Exa 2.28 Furnace wall

```

1
2 clc;
3 clear;
4 //Example 2.28
5 //Calculate heat loss per sq m and temperature of
    outside surface
6 //Given
7 A=1 //assume [sq m]
8 x1=0.006 // [m]
9 x2=0.075 // [m]
10 x3=0.2 // [m]
11 k1=39 // [W/m.K]
12 k2=1.1 // [W/m.K]
13 k3=0.66 // [W/m.K]
14 h0=65 //W/sq m .K
15 T1=900 //K
16 T2=300 //K
17 sigma_R=(x1/(k1*A)+x2/(k2*A)+x3/(k3*A)+1/(h0*A));

```



```

18 //To calculate heat loss/sq m area
19 Q=(T1-T2)/sigma_R // [W/sq m]
20 printf("Heat loss per sq metre area is: %f W/sq m",Q
    );
21 //Q/A=T-T2/(1/h0), where T=Temp of outside surface
22 //So, T=T2+Q/(A*h0)
23 T=Q/(A*h0)+T2 // [K]
24 printf("Temperature of outside surface of furnace is:
    %f K (%f degree C)",T,T-273)

```

Scilab code Exa 2.29 Thickness of insulating brick

```

1
2 clear;
3 clc;
4 //Example 2.29
5 //Determine necessary thickness of insulation brick
6 //Given
7 A=1 //Assume [sq m]
8 x1=0.003 // [m]
9 x3=0.008 // [m]
10 k1=30 // [W/m.K]
11 k2=0.7 // [W/m.K]
12 k3=40 // [W/m.K]
13 T1=363 // [K]
14 T=333 // [K]
15 T2=300 // [K]
16 h0=10 //W/sq m.K
17 //Q=(T1-T2)/(x1/(k1*A)+x2/(k2*A)+x3/(k3*A)+1/(h0*A))
18 //Also ,Q=(T-T2)/(1/(h0*A))
19 //So, (T1-T2)/((x1/(k1*A)+x2/(k2*A)+x3/(k3*A)+1/(h0*
    A))=(T-T2)/(1/(h0*A))
20 //or ,x2=k2*A*((T1-T2)/((T-T2)*h0*A)-1/(h0*A)-x1/(k1*A)
    )-x3/(k3*A))
21 x2=k2*A*((T1-T2)/((T-T2)*h0*A)-1/(h0*A)-x1/(k1*A)-x3

```

```

        /(k3*A)); // [m]
22 printf("Thickness of insulating brick required is %f
        mm", x2*1000);

```

Scilab code Exa 2.30 Heat flow through furnace wall

```

1
2 clear;
3 clc;
4 //Example 2.30
5 //Given
6 hi=75 // [W/sq m.K]
7 x1=0.2 //m
8 x2=0.1 // [m]
9 x3=0.1 // [m]
10 T1=1943 // [K]
11 k1=1.25 //W/m.K
12 k2=0.074 //W/m.K
13 k3=0.555 //W/m.K
14 T2=343 //K
15 A=1 //assume [sq m]
16 sigma_R=1/(hi*A)+x1/(k1*A)+x2/(k2*A)+x3/(k3*A);
17 //Heat loss per i sq m
18 Q=(T1-T2)/sigma_R // [W]
19 //if T=temperature between chrome brick and koalin
    brick then
20 //Q=(T1-T)/(1/(hi*A)+x1/(k1*A))
21 //or T=T1-(Q*(1/(hi*A)+x1/(k1*A)))
22 T=T1-(Q*(1/(hi*A)+x1/(k1*A))); // [K]
23 printf("Temperature at inner surface of middle layer
        =%f K(%f degree C)", T, T-273);
24 //if Tdash=temperature at the outer surface of
    middel layer, then
25 //Q=(Tdash-T2)/(x3/(k1*A))
26 //or Tdash=T2+(Q*x3/(k3*A))

```

```

27 Tdash=T2+(Q*x3/(k3*A))    // [K]
28 printf("Temperature at outer surface of middle layer
    =%f K (%f degree C)",Tdash ,Tdash-273);

```

Scilab code Exa 2.31 Heat loss in pipe

```

1
2 clear;
3 clc;
4 //Example 2.31
5 //Calculate:(a) Heat loss per unit length
6 //(b)Reduction in heat loss
7 //Given
8 hi=10    //W/sq m.K
9 h0=hi    //W/sq.m.K
10 r1=0.09 //m
11 r2=0.12 //m
12 t=0.05  //thickness of insulation [m]
13 k1=40    //W/m.K
14 k2=0.05  //W/m.K
15 T1=473   //K
16 T2=373   //K
17 Q_by_L=2*pi*(T1-T2)/(1/(r1*hi)+log(r2/r1)/k1+1/(r2*
    h0)); //W/m
18 printf("Ans (a) Heat loss=%f W/m ",Q_by_L)
19 //After addition of insulation:
20 r3=r2+t; //radius of outer surface of insulaiton
21 Q_by_L1=2*pi*(T1-T2)/(1/(r1*hi)+log(r2/r2)/k1+log(
    r3/r2)/k2+1/(r3*h0)); // W
22 Red=Q_by_L-Q_by_L1 //Reduciton in heat loss in [W
    /m]
23 percent_red=(Red/Q_by_L)*100 // % Reduction in
    heat loss
24 printf("Ans (b) Percent reduction in heat loss is %f
    percent",percent_red)

```

Scilab code Exa 2.32 Heat flux through layers

```
1
2 clear;
3 clc;
4 //Example 2.32
5 //Determine: i-Heat flux across the layers and
6 //ii-Interfacial temperature between the layers
7
8 //Given
9 T1=798 //K
10 T2=298 //K
11 x1=0.02 //m
12 x2=x1 //m
13 k1=60 //W/m.K
14 k2=0.1 //W/m.K
15 hi=100 //W/sq m.K
16 h0=25 //W/sq m.K
17 Q_by_A=(T1-T2)/(1/hi+x1/k1+x2/k2+1/h0); //W/sq m
18 printf("Ans (i)- Heat flux across the layers is %f W
19 /sq m",Q_by_A);
20 //If Tis the interfacial temperature between steel
21 //plate and insulating material
22 //Q_by_A=(T-T2)/(x2/k2+1/h0)
23 T=Q_by_A*(x2/k2+1/h0)+T2
24 printf("Ans-(ii)-Interfacial temperature between
25 layers is %f K (%f degree C)",T,T-273);
```

Scilab code Exa 2.33 Conductive conductance furnace wall

1

```

2
3 clc;
4 clear;
5 //Example 2.33
6 //Determine Temperature at the outer surface of wall
   and convective conductance on the outer wall
7 //Temperature of hot gas:
8 T1=2273 //K
9 //Ambient air temperature:
10 T4=318 //K
11 //Heat flow by radiation from gases to inside
   surface of wall:
12 Qr1_by_A=23260 // [W/sq m]
13 //Heat transfer coefficient on inside wall:
14 hi=11.63 //W/sq m.K
15 //Thermal conductivity of wall:
16 K=58 //W.sq m/K
17 //Heat flow by radiation from external surface
   to ambient:
18 Qr4_by_A=9300 //W/sq m.
19 //Inside Wall temperature:
20 T2=1273 //K
21
22 Qr1=Qr1_by_A //W for
23 A=1 //sq m
24
25 Qc1_by_A=hi*(T1-T2) //W/sq m
26 Qc1=Qc1_by_A //for A=1 sq m
27 //Thermal resistance:
28 R=1/K //K/W per sq m
29 //Now Q=(T2-T3)/R, i.e
30 //External wall temp T3=T2-Q*R
31 //Q entering wall=
32 Q_enter=Qr1+Qc1 //W
33 T3=T2-Q_enter*R //K
34 T3=673 //Approximate
35 //Heat loss due to convection:
36 Qc4_by_A=Q_enter-Qr4_by_A //W/sq m

```

```

37 //Qc4_by_A=h0*(T3-T4)
38 //or h0=Qc4_by_A/(T3-T4)
39 h0=Qc4_by_A/(T3-T4) //W/sq m.K
40 //Result
41 printf("Convective conductance is: %f W/sq m.K",h0)

```

Scilab code Exa 2.34 Critical radius of insulation

```

1  clc;
2  clear;
3  //Example 2.34
4  //Given
5  T1=473 // [K]
6  T2=293 // [K]
7  k=0.17 //W/(m.K)
8  h=3 //W/(sq m.K)
9  h0=h //W/sq m.K
10 rc=k/h //m
11 r1=0.025 //Inside radius of insulaiton [mm]
12 q_by_l1=2*%pi*(T1-T2)/(log(rc/r1)/k+1/(rc*h0)) //
    Heat transfer with insulation in W/m
13 //Without insulation:
14 q_by_l2=h*2*%pi*r1*(T1-T2) //W/m
15 inc=(q_by_l1-q_by_l2)*100/q_by_l2 //Increase of
    heat transfer
16 printf("When covered with insulation ,\n heat loss=%f
    W \n When without insulation ,heat loss= %f W \n
    percent increase =%f percent",q_by_l1,q_by_l2,inc
    );
17 k=0.04 //Fibre glass insulaiton W/(sq m.K)
18 rc=k/h //Critical radius of insulaiton
19 printf("In this case the avlue of rc=%f m is less
    than the outside radius of pipe (%f),\n So
    additon of any fibre glass would cause a decrease
    in the heat transfer \n",rc,r1)

```

Scilab code Exa 2.36 Critical radius of pipe

```
1
2 clear;
3 clc;
4 //Example 2.36
5 //Calculate the heat loss per metre of pipe and
  outer surface temperature
6 //Given
7 k=1 //Thermal conductivity in [W/sq m.K]
8 h=8 //Het transfer coeff in W/sq m.K
9 rc=k/h //Critical radius in m
10 T1=473 //K
11 T2=293 //K
12 r1=0.055 //Outer radius =inner radius in [m]
13 Q_by_L=2*%pi*(T1-T2)/(log(rc/r1)/k+1/(rc*h))
14 printf("Heat loss per meter of pipe is %f W/m",
  Q_by_L)
15 //For outer surface
16 //Q_by_L=2*%pi*(T-T2)/(1/rc*h)
17 // implies that , T=T2+Q_by_L/(rc*2*%pi)
18 T=T2+Q_by_L/(rc*2*%pi*h) //K
19 printf("Outer surface temperature is: %f K(%f degree
  C)",T,T-273)
```

Scilab code Exa 2.37 Time required for steel ball

```
1
2 clc;
3 clear;
4 //Example 2.37
```

```

5 //Calculate the time required for a ball to attain a
   temperature of 423 K
6 //Given
7 k_steel=35 //W/m.K
8 Cp_steel=0.46 //kJ/(kg*K)
9 Cp_steel=Cp_steel*1000 //J/(kg*K)
10 h=10 //W/sq m.K
11 rho_steel=7800 //kg/cubic m
12 dia=50 //mm
13 dia=dia/1000 //m
14 R=dia/2 //radius in m
15 A=4*pi*R^2 //Area in sq m
16 V=A*R/3 //Volume in cubic meter
17 Nbi=h*(V/A)/k_steel
18 //As Nbi<0.10,internal temp gradient is negligible
19 T=423 //K
20 T0=723 //K
21 T_inf=373 //K
22 //(T-T_inf)/(T0-T_inf)=e^(-h*At/rho*Cp*V)
23 t=-rho_steel*Cp_steel*R*log((T-T_inf)/(T0-T_inf))
   /(3*h); //s
24 printf("Time required for a ball to attain a
   temperature of 423 K is %f s= %f h",t,t/(3600))

```

Scilab code Exa 2.38 Steel ball quenched

```

1
2 clc;
3 clear;
4 //Example 2.38
5 //Given
6 dia=50 //mm
7 dia=dia/1000 //m
8 r=dia/2 //radius in m
9 h=115 //W/sq m.K

```



```

10 rho=8000 //kg/cubic m
11 Cp=0.42 //kJ/kg.K
12 Cp=Cp*1000 //J/(kg*K)
13 A=4*%pi*r^2 //Area in sq m
14 V=A*r/3 //Volume in cubic m
15 T=423 //K
16 T_inf=363 //K
17 T0=723 //K
18 //(T-T_inf)/(T0-T_inf)=e^(-3ht/(rho*Cp*r))
19 t=-rho*Cp*r*log((T-T_inf)/(T0-T_inf))/(3*h); //
    Time in seconds
20 printf("Time taken by centre of ball to reach a
    temperature of 423 K is %f s (= %f minutes",t,t
    /60);

```

Scilab code Exa 2.39 Ball plunged in a medium

```

1
2 clc;
3 clear;
4 //Example 2.39
5 //Given
6 h=11.36 //W/sq m.K
7 k=43.3 //w/(m.K)
8 r=25.4 //radius in mm
9 r=r/1000 // radius in m
10 A=4*%pi*r^2 //Area of sphere [sq m]
11 V=A*r/3 //Volume in [cubic m]
12 rho=7849 //kg/cubic m
13 Cp=0.4606*10^3 //J/kg.K
14 t=1 //hour
15 t=t*3600 //seconds
16 T_inf=394.3 // [K]
17 T0=700 // [K]
18 // (T-T_inf)/(T0-T_inf)=e^(-3*h*t/rho*Cp*V)

```

```

19 T=T_inf+(T0-T_inf)*(%e^((-h*A*t)/(rho*Cp*V)));
20 printf("Temperature of ball after 1 h= %f K (%f
    degree C)",T,T-273)

```

Scilab code Exa 2.40 Slab temperature suddenly lowered

```

1  clc;
2  clear;
3  //Example 2.40
4  //Given
5  rho=9000; //kg/cubic m
6  Cp=0.38; //kJ/(kg.K)
7  Cp=Cp*1000 //J/(kg.K)
8  k=370; //W/m.K
9  h=90; //W/sq m.K
10 l=400; //mm
11 l=l/1000 ; //length of copper slab
12 t=5/1000 ; //thickness in [m]
13 A=2*l^2 //Area of slab
14 V=t*l^2 //Volume in [cubic m]
15 L_dash=V/A // [m]
16 //for slab of thickness 2x
17 //L_dash=x
18 L_dash=0.025 ; // [m]
19 Nbi=h*L_dash/k // < 0.10
20 var=h*A/(rho*Cp*V)
21 //As Nbi<0.10,we can apply lumped capacity analysis
22 T=363 // [K]
23 T_inf=303 // [K]
24 T0=523 // [K]
25 t=-(log((T-T_inf)/(T0-T_inf)))/var
26 printf("Time at which slab temperature becomes 363 K
    is %f s",t)
27 printf("CALCULATION MISTAKE IN BOOK IN LAST LINE")

```

Scilab code Exa 2.41 Flow over a flat plate

```
1
2 clc;
3 clear;
4 //Example 2.41
5 //Given
6 rho=9000 //kg/cubic meter
7 Cp=0.38 //kJ/(kg.K)
8 Cp=Cp*1000 //J/kg.K
9 k=370 //W/(m.K)
10 T0=483 //K
11 T_inf=373 //K
12 delta_T=40 //K
13 T=T0-delta_T //K
14 t=5 //time in [minutes]
15 t=t*60 //[seconds]
16 //A=2A.....Two faces
17 //V=A.2x
18 //2x=thickness of slab=30 mm=0.03 m
19 x=0.015 //[m]
20 th=2*x //thickness of slab
21 h=-rho*Cp*x*log((T-T_inf)/(T0-T_inf))/t
22 printf("Heat transfer coefficient is: %f W/(sq m.K)"
, h)
```

Scilab code Exa 2.42 Stainless steel rod immersed in water

```
1
2 clear;
3 clc;
4 //Example 2.42
```

```

5 //Given
6 rho=7800 // [kg per cubic m]
7 h=100 //W/(sq m.K) Convective heat transfer
  coeff
8 Cp=460 //J/(kg.K)
9 k=40 //W/(m.K)
10 L=1 // [m] length ofrod
11 D=10 //mm
12 D=D/1000 //diameter in [m]
13 R=D/2 //raidus in [m]
14 //For cylindrical rod:
15 A=2*pi*R*L //Area in [sq m]
16 V=%pi*R^2*L //Volume in [cubic m]
17 L_dash=V/A // [m]
18 Nbi=h*L_dash/k //Biot number
19 //N_bi<0.10,Hence lumped heat capavity is possible
20 T=473 // [K]
21 T_inf=393 // [K]
22 T0=593 // [K]
23 t=-rho*Cp*V*log((T-T_inf)/(T0-T_inf))/(h*A)
24 printf("Time required to reach temperature %f is %f
  s",T,t);

```

Scilab code Exa 2.43 Chromel alumel thermocouple

```

1
2 clear;
3 clc;
4 //Example 2.43
5 //Given
6 rho=8600 // [kg/cubic m]
7 Cp=0.42 //kJ/(kg.K)
8 Cp=Cp*1000 //J/(kg.K)
9 dia=0.71 // [mm]
10 dia=dia/1000 // [dia in m]

```

```

11 R=dia/2      //radius [m]
12 h=600      //convective coeff W/(sq m.K)
13 //Let length =L=1
14 L=1        //[m]
15 A=2*%pi*R*L;
16 V=%pi*(R^2)*L;
17 tao=(rho*Cp*V)/(h*A);
18 printf("Time constant of the thermocouple is %f s",
        tao);
19 //at
20 t=tao
21 //From  $(T-T_{inf})/(T_0-T_{inf})=e^{(-t/tao)}$ 
22 ratio=%e^(-t/tao) //Ratio of thermocouple
        difference to initial temperature difference
23 printf("At the end of the time period t=tao=%f s ,
        Temperature difference b/n the thermocouple and
        the gas stream would be %f of the initial
        temperature difference",tao,ratio);
24 printf("\n It should be reordered after %f s",4*tao)
        ;

```

Scilab code Exa 2.44 Thermocouple junction

```

1
2 clc;
3 clear;
4 //Example 2.44
5 rho=8000 //kg/cubic m
6 Cp=420 //J/(kg.K)
7 h_hot=60 // for hot stream W/(sq m.K)
8 dia=4 // [mm]
9 t=10;
10 r=dia/(2*1000) //radius in [m]
11 //For sphere
12 V=(4/3)*%pi*r^3 //Volume in [cubic m]

```

```

13 A=4*%pi*r^2           //Volume in [sq m]
14 tao=rho*Cp*V/(h_hot*A) // Time constant in [s]
15 ratio=%e^(-t/tao)     // %e^(-t/tao)=(T-T-inf)/(T0-
    T_inf)
16 T_inf=573            // [K]
17 T0=313              // [K]
18 T=T_inf+ratio*(T0-T_inf)
19 //ANS-[i]
20 printf("\n Answer: Time constant of thermocouple is
    %f s",tao);
21
22 //IN STILL AIR:
23 h_air=10            //W/(sq m .K)
24 tao_air=rho*Cp*V/(h_air*A) // [s]
25 t_air=20           // [s]
26 ratio_air=%e^(-t_air/tao_air)
27 T_inf_air=303      // [K]
28 T0_air=T;
29 T_air=T_inf_air+ratio_air*(T0_air-T_inf_air)
30 //ANS-[ii]
31 printf("Temperature attained by junction 20 s after
    removing from the hot air stream is:%d K",round(
    T_air))

```

Scilab code Exa 2.45 Batch reactor

```

1 clc;
2 clear;
3 //Example 2.45
4 T_inf=390;           // [K]
5 U=600;              // [W/sq m.K]
6 Ac=1;               // [sq m]
7 Av=10               // Vessel area in [sq m]
8 m=1000;             // [kg]
9 Cp=3.8*10^3;        // [J/kg.K]

```

```

10 To=290; // [K]
11 T=360; // [K]
12 h=8.5 // [W/sq m.K]
13 //Heat gained from the steam=Rate of increase of
    internal energy
14 //U*A*(T_inf-T)=m*Cp*dT
15 deff(' [x]=f(t)', 'x=log((T_inf-To)/(T_inf-T))-U*Ac*t
    /(m*Cp)');
16 t=fsolve(1,f); // [in s]
17 t=round(t) // [in s]
18 Ts=290;
19 printf("\nTime taken to heat the reactants over the
    same temperature range is %f h",t);
20 function t1=g(T),t1=m*Cp/(U*Ac*(T_inf-T)-h*Av*(T-Ts)
    ),endfunction
21 t1=intg(To,T,g);
22 deff(' [m]=fx(Tmax)', 'm=U*Ac*(T_inf-Tmax)-h*Av*(Tmax-
    Ts)')
23 T_max=fsolve(1,fx)
24 printf("\nANS: In CASE 1\nTime taken to heat the
    reactants = %f s .ie %f h\n",t,t/3600);
25 printf("\nANS: In CASE 2 \n Time taken to heat the
    reactants = %f s\n",t1);
26 printf("\nANS.: Maximum temperature at which
    temperature can be raised is %f K\n",T_max);

```

Scilab code Exa 2.46 Heat dissipation by aluminium rod

```

1
2 clc;
3 clear;
4 //Example 2.46
5 dia=3 // [mm]
6 dia=dia/1000 // [m]
7 r=dia/2 //radius in [m]

```

```

8 k=150      //W/(m.K)
9 h=300      //W/(sq m.K)
10 T0=413    // [K]
11 T_inf=288  // [K]
12 A=%pi*(r^2) //Area in [sq m]
13 P=%pi*dia // [W/sq m.K]
14 Q=(T0-T_inf)*sqrt(h*P*k*A) //Heat dissipated in [W
]
15 printf("Heat dissipated by the rod is %f W",Q)

```

Scilab code Exa 2.47 Aluminium fin efficiency

```

1
2 clc;
3 clear;
4 //Example 2.47
5 //Given
6 k=200      //W/(m.K)
7 h=15       //W/(sq m.K)
8 T0=523    // [K]
9 T_inf=288  // [K]
10 theta_0=T0-T_inf
11 dia=25     //diameter [mm]
12 dia=dia/1000 //diameter [m]
13 r=dia/2    //radius in [m]
14 P=%pi*dia // [m]
15 A=%pi*r^2  // [sq m]
16 //For insulated fin:
17 m=sqrt(h*P/(k*A))
18 L=100      //length of rod in [mm]
19 L=L/1000   //length of rod in [m]
20 Q=theta_0*tanh(m*L)*sqrt(h*P*k*A) //Heat loss
21 //ANSWER-1
22 printf("Heat loss by the insulated rod is %f W \n",Q
)

```



```

23 nf=tanh(m*L)/(m*L)    //Fin efficiency for
    insulated fin
24 //ANSWER-2
25 printf("Fin efficiency is %f percent \n",nf*100)
26 //At the end of the fin: theta/theta_0=(cosh [m(L-x)
    ]/ cosh(mL))
27 //at x=L, theta/theta_0=1/(cosh(mL)
28 T=T_inf+(T0-T_inf)*(1/cosh(m*L))    // [K]
29 //ANSWER-3
30 printf("Temperature at the end of the fin is %f K \n
    ",T)

```

Scilab code Exa 2.49 Pin fins

```

1
2 clc;
3 clear;
4 //Example 2.49
5 //Given
6 k=300    //W/(m.K)
7 h=20    //W.(sq m.K)
8 P=0.05    // [m]
9 A=2    // [sq cm]
10 A=A/10000    // [sq m]
11 T0=503    // [K]
12 T_inf=303    // [K]
13 theta_0=T0-T_inf    // [K]
14 m=sqrt(h*P/(k*A))
15 //CASE 1: 6 Fins of 100 mm length
16 L1=0.1    //Length of fin in [m]
17 Q=sqrt(h*P*k*A)*theta_0*tanh(m*L1)    // [W]
18 //For 6 fins
19 Q=Q*6    //for 6 fins [W]
20 //CASE 2: 10 fins of 60 mm length
21 L2=60    // [mm]

```

```

22 L2=L2/1000      // [m]
23 Q2=sqrt(h*P*k*A)*theta_0*tanh(m*L2);      // [W]
24 Q2=Q2*10      //For 10 fins
25 printf("As,Q for 10 fins of 60 mm length( %f W) is
        more than Q for 6 fins of 100 mm length (%f W).\n
        The agreement-->10 fins of 60 mm length is more
        effective",Q2,Q);

```

Scilab code Exa 2.50 Metallic wall surrounded by oil and water

```

1  clc;
2  clear;
3  //Example 2.50
4  //Given
5  h_oil=180      //W/(sq m.K)
6  h_air=15      //W/(sq m.K)
7  T_oil=353     // [K]
8  T_air=293     // [K]
9  delta_T=T_oil-T_air;      // [K]
10 k=80          // Conductivity in [W/(m.K)]
11 for_section=11*10^-3     // [m]
12 L=25          // [mm]
13 L=L/1000     // [m]
14 W=1          // [m] Width,.. let
15 t=1          // [mm]
16 t=t/1000    // [m]
17 A=W*t       // [m]
18 P=2*t
19 Af=2*L*W    //sq m
20 N=1
21 Ab=for_section-A      // [sq m]
22 //CASE 1: Fin on oil side only
23 m=sqrt(h_oil*P/(k*A))
24 nf_oil=tanh(m*L)/(m*L)
25 Ae_oil=Ab+nf_oil*Af*N      // [sq m]

```

```

26 Q=delta_T/(1/(h_oil*Ae_oil)+1/(h_air*for_section))
    // [W]
27 printf("In oil side ,Q=%f W\n",Q);
28 //CASE 2: Fin on air side only
29 m=sqrt(h_air*P/(k*A))
30 nf_air=tanh(m*L)/(m*L)
31 nf_air=0.928 //Approximation
32 Ae_air=Ab+nf_air*Af*N // [sq m]
33 Q=delta_T/(1/(h_oil*for_section)+1/(h_air*Ae_air))
    // [W]
34 printf("In air side ,Q=%f W",Q);
35 printf("\n From above results we see that more heat
    transfer takes place if fins are provided on the
    air side");

```

Scilab code Exa 2.51 Brass wall

```

1
2 clc;
3 clear;
4 //Example 2.51
5 //Given
6 k=75 //Thermal conductivity [W/(m.K)]
7 T_water=363 // [K]
8 T_air=303 // [K]
9 dT=T_water-T_air //delta T
10 h1=150 // for water [W/(sq m.K)]
11 h2=15 //for air [W/(sq m.K)]
12 W=0.5 //Width of wall [m]
13 L=0.025 // [m]
14 Area=W^2 //Base Area [sq m]
15 t=1 // [mm]
16 t=t/1000 // [m]
17 pitch=10 // [mm]
18 pitch=pitch/1000 // [m]

```

```

19 N=W/pitch // [No of fins]
20 // Calculations
21 A=N*W*t // Total cross-sectional area of fins in
    [sq m]
22 Ab=Area-A // [sq m]
23 Af=2*W*L // Surface area of fins [sq m]
24
25 //CASE 1: HEAT TRANSFER WITHOUT FINS
26 A1=Area // [sq m]
27 A2=A1 // [sq m]
28 Q=dT/(1/(h1*A1)+1/(h2*A2)); // [W]
29 printf("\n Without fins ,Q=%f W\n",Q);
30 //CASE 2: Fins on the water side
31 P=2*(t+W);
32 A=0.5*10^-3;
33 m=sqrt(h1*P/(k*A))
34 nfw=tanh(m*L)/(m*L) // Efficiency on water side
35 Aew=Ab+nfw*Af*N // Effective area on the water
    side [sq m]
36 Q=dT/(1/(h1*Aew)+1/(h2*A2)); // [W]
37 printf("\n With fins on water side ,Q=%f W\n",Q);
38 //CASE 3: FINS ON THE AIR SIDE
39 m=sqrt(h2*P/(k*A))
40 nf_air=tanh(m*L)/(m*L) // Efficiency
41 Aea=Ab+nf_air*Af*N // Effective area on air side
42 Q=dT/(1/(h1*A1)+1/(h2*Aea)); // [W]
43 printf("\n With Fins on Air side ,Q=%f W\n",Q)
44 //BOTH SIDE:
45 Q=dT/(1/(h1*Aew)+1/(h2*Aea)); // [W]
46 printf("\n With Fins on both side ,Q=%f W\n",Q);

```

Chapter 3

Convection

Scilab code Exa 3.1 Boundary layer thickness

```
1  clc ;
2  clear ;
3  //Example 3.1
4  mu=10^-3           //N.s/m^2
5  //At distance y from surface
6  //ux=a+by+cy^2+dy^3
7  //At y=0,ux=0 therefore a=0
8  //i.e tao=0
9  //At edge of boundary layer ,ie y=del
10 //ux=u_inf
11 //At y=0, c=0
12 //At y=del ,ux=b*del+d*del^3
13
14 //Therefore , b=-3*d*del^3
15 //d=-u_inf/(2*del^2)
16 //b=3*u_inf/(2*del)
17
18 //For velocity profile ,we have:
19 //del/x=4.64*(Nre_x)^(-1/2)
20
21 //Evaluate N re_x
```

```

22
23 x=75;           // [mm]
24 x=x/1000;      // [m]
25 u_inf=3;       // [m/s]
26 rho=1000       // [kg/m^3] for air
27 Nre_x=u_inf*rho*x/mu //Reynold number
28 //Substituting the value ,we get
29 del=x*4.64*(Nre_x^(-1/2)) // [m]
30 printf("\nBoundary layer thickness is del=%f m or %f
      mm",del,del*1000);
31 printf("\nWrong units in answer of book ,m and mm are
      wrongly interchanged");

```

Scilab code Exa 3.2 Boundary layer thickness of plate

```

1  clc;
2  clear;
3  //Example3.2
4  //Given
5  mu=15*10^-6 //sq m /s
6  v=2 //m/s
7  L=2 // [m] length of plate
8  Nre_x=3*10^5
9  xc=Nre_x*mu/v //critical length at whihc the
      transition takes place
10 //Since xc is less than 2 m. Therefore the flow is
      laminar
11 //at any distance x,.it is calculated from
12 //del/x=4.64/(sqrt(NRe,x))
13 //At x=L=2 m
14 Nre_l=v*L/mu
15 del_l=4.64*L/sqrt(Nre_l)
16 del_l=del_l*1000 // [mm]
17 printf("Boundary layerthickness at the trailing edge
      is %f mm",del_l);

```

Scilab code Exa 3.3 Thickness of hydrodynamic boundary layer

```
1 clc;
2 clear;
3 //Example 3.3
4 //Given
5 mu=15*10^-6 //Kinematic viscosity in [sq m /s]
6 x=0.4 // [m]
7 u_inf=3 // [m/s]
8 //At x=0.4 m,
9 Nre_x=u_inf*x/mu ;
10 printf("Since Nre,x (%f) is Less than 3*10^5,..the
    boundary layer is laminar",Nre_x);
11 del=4.64*x/sqrt(Nre_x) // [m]
12 del=del*1000 // [mm]
13 printf("\nThickness of boundary layer at x=%f m =%f
    mm\n",x,del);
14 Cf_x=0.664/sqrt(Nre_x);
15 printf("Local skin friction coefficient is :%f",Cf_x
    );
```

Scilab code Exa 3.4 Flat plate boundary layer

```
1 clc;
2 clear;
3 //Example 3.4
4 mu=1.85*10^-5 // [kg/(m.s)]
5 P=101.325; //Pressure in [kPa]
6 M_avg=29; //Avg molecular wt of air
7 R=8.31451; //Gas constant
8 T=300; // [K]
```

```

9 rho=P*M_avg/(R*T)    //[kg/m^3]
10 u_inf=2              //Viscosity in [m/s]
11 //At x=20 cm =0.2 m
12 x=0.2;              //[m]
13 Nre_x=rho*u_inf*x/mu  //[Reynolds number]
14 del_by_x=4.64/sqrt(Nre_x)  //[Boundary layer]
15 del=del_by_x*x      //[m]
16 //del=del*1000      //[mm]
17
18 //At
19 x=0.4 ;             //[m]
20 Nre_x=(rho*u_inf*x)/mu  //<3*10^5
21 //Boundary layer is laminar
22 del_by_x=4.64/sqrt(Nre_x)
23 del1=del_by_x*x     //[m]
24 //del1=del1*1000    //[mm]
25 d=del1-del          //Del
26 function m_dot=f(y),m_dot=u_inf*(1.5*(y/d)-0.5*(y/d)
    ^3)*rho, endfunction
27 m_dot=intg(0,d,f)
28 printf("\nBoundary layer thickness at distance 20 cm
    from leading edge is %f m=%f mm\n",del,del*1000)
    ;
29 printf("\nBoundary layer thickness at distance 40 cm
    from leading edge is %f m=%f mm\n",del1,del1
    *1000);
30 printf("\nThus,Mass flow rate entering the boundary
    layer is %f kg/s",m_dot);

```

Scilab code Exa 3.5 Rate of heat removed from plate

```

1
2 clc;
3 clear;
4 //Example 3.5

```



```

5 //Given
6 mu=3.9*10^-4 //Kinematic viscosity in sq m/s
7 k=36.4*10^-3 //Thermal conductivity in W/(m.K)
8 Npr=0.69
9 u_inf=8 // [m/s]
10 L=1 //Lenght of plate in [m]
11 Nre_l=u_inf*L/mu
12 //Since Nre_l is less than 3*10^5 ,the flow is
    laminar over the entire length of plate
13 Nnu=0.664*sqrt(Nre_l)*Npr^(1.0/3.0) //hL/k
14
15 h=k*Nnu/L //w/sq m.K
16 h=3.06 //Approximation [W/sq m.K]
17 T_inf=523 // [K]
18 Tw=351 // [K]
19 W=0.3 //Width of plate [m]
20 A=W*L //Area in [sq m]
21 Q=h*A*(T_inf-Tw) // Rate of heat removal from one
    side in [W]
22 printf("\nRate of heat removal is %f W\n",Q)
23 //from two side:
24 Q=2*Q // [W]
25 printf("\n %f W heat should be removed continously
    from the plate",Q);

```

Scilab code Exa 3.6 Heat removed from plate

```

1
2 clc;
3 clear;
4 //Example 3.6
5 P1=101.325 //Pressure in [kPa]
6 mu1=30.8*10^-6 //Kinematic viscosity in [sq m /s]
7 k=36.4*10^-3 // [W/(m.K)]
8 Npr=0.69

```

```

 9 u_inf=8      //Velocity in [m/s]
10 Cp=1.08     //kJ/(kg.K)
11 L=1.5       //Length of plate in [m]
12 W=0.3       //Width in [m]
13 A=L*W       //Area in [sq m]
14 //At constant temperature: mu1/mu2=P2/P1
15 P2=8        //[kPa]
16 mu2=mu1*P1/P2 //Kinematic viscosity at P2 in [sq
    m/s]
17 Nre_l=u_inf*L/mu2 //Reynold's no.
18 //Since this is less than 3*10^5
19 Nnu=0.664*sqrt(Nre_l)*(Npr^(1.0/3.0))
20 h=Nnu*k/L // Heat transfer coefficient in [W/sq m.
    K]
21 h=2.5       //Approximation in [W/sq m.K]
22 T_inf=523   //[K]
23 Tw=353     //[K]
24 Q=2*h*A*(T_inf-Tw) //Heat removed from both sides
    in [W]
25 printf("Rate of heat removed from both sides of
    plate is %f W",Q);

```

Scilab code Exa 3.7 Local heat transfer coefficient

```

1 clc;
2 clear;
3 //Example 3.7
4 rho=0.998 //kg/cubic m
5 v=20.76*10^-6 //[sq m/s]
6 Cp=1.009 // [kJ/kg.K]
7 k=0.03 // [W/m.K]
8 u_inf=3 // [m/s]
9 x=0.4 // [m]
10 w=1.5 // [m]
11 Nre_x=u_inf*x/v //Reynolds no at x=0.4 m

```

```

12 //Since this is less than  $3 \times 10^5$ .The flow is laminar
    upto x=0.4 m
13 mu=rho*v // [kg/(m. s)]
14
15 Cp=1.009 // [kJ/kg.K]
16 Cp=Cp*1000 // [J/kg.K]
17 k=0.03 //W/(m.K)
18 Npr=Cp*mu/k
19 Nnu_x=0.332*(sqrt(Nre_x))*(Npr^(1.0/3.0))
20 hx=Nnu_x*k/x // [W/(m.K)]
21 //Average value is twice this value
22 h=2*hx // [W/(m.K)]
23 h=10.6 //Approximation
24 A=x*w //Area in [sq m]
25 Tw=407 // [k]
26 T_inf=293 // [K]
27 Q=h*A*(Tw-T_inf) // [W]
28 //From both sides of the plate:
29 Q=2*Q // [W]
30 printf("The heat transferred from both sides of the
    plate is %d W",round(Q));

```

Scilab code Exa 3.8 Width of plate

```

1 clc;
2 clear;
3 //Example 3.8
4 rho=0.998 // [kg/cubic m]
5 v=20.76*10^-6 // [sq m/s]
6 k=0.03 // [W/m.K]
7 Npr=0.697
8 x=0.4 // [m] from leading edge of the plate
9 u_inf=3 // [m/s]
10 Nre_x=u_inf*x/v //Reynold numebr at x=0.40 m
11 //Since this is less than  $3 \times 10^5$ 

```

```

12 //therefore flow is laminar and
13 Nnu_x=0.332*sqrt(Nre_x)*(Npr^(1.0/3.0));
14 hx=Nnu_x*k/x // [W/sq m.K]
15 //Average heat tarnser coefficient is twice this
    value
16 h=2*hx // [W/sq m.K]
17 //Given:
18 Q=1450 // [W]
19 Tw=407 // [K]
20 T_inf=293 // [K]
21 L=0.4 // [m]
22 //Q=h*w*L*(Tw-T_inf)
23 //L=Q/(h*w*(Tw-T_inf))
24 w=Q/(h*L*(Tw-T_inf)) // [m]
25 printf("\n Width of plate is %f m",w);

```

Scilab code Exa 3.9 Heat transferred in flat plate

```

1
2 clc;
3 clear;
4 //Example 3.9
5 v=17.36*10^-6 //Viscosity for air [sq m./s]
6 k=0.0275 //for air ..[W/(m.K)]
7 Cp=1.006 // [kJ/(kg.K)]
8 Npr=0.7 //for air
9 u_inf=2 // [m/s]
10 x=0.2 // [m]
11 Nre_x=u_inf*x/v //Reynolds number at x=0.2 m
12 //Since this is less than 3*10^5
13 Nnu_x=0.332*sqrt(Nre_x)*(Npr^(1.0/3.0))
14 hx=Nnu_x*k/x // [W/(sq m.K)]
15 //Average value of heat transfer coeff is twice this
    value
16 h=2*hx // [W/sq m.K)]

```

```

17 h=12.3      //Approximation
18 w=1        //width in [m]
19 A=x*w      //[sq m] Area of plate
20 Tw=333     //[K]
21 T_inf=300  //[K]
22 Q=h*A*(Tw-T_inf) //Heat flow in [W]
23 printf("\nANSWER:\nHeat flow is :%f W\n",Q)
24 //From both sides of plate:
25 Q=2*Q      //[W]
26 printf("\nANSWER\n Heat flow from both sides of
    plate is %f W",Q);

```

Scilab code Exa 3.10 Rate of heat transferred in turbulent flow

```

1  clc;
2  clear;
3  //Example 3.10
4  v=16.96*10^-6 // [sq m./s]
5  rho=1.128     //[kg/cubic m]
6  Npr=0.699    //Prandtl number
7  k=0.0276    //[W/m.K]
8  u_inf=15     //[m/s]
9  L=0.2       //[m]
10 Nre_l=L*u_inf/v //Reynold's number
11 //Since this is less than 3*10^5,the boundary layer
    is laminar over entire length
12 Nnu=0.664*sqrt(Nre_l)*(Npr^(1.0/3.0))
13 h=Nnu*k/L    //[W/sq m.K]
14 A=L^2       //[Area in [sq m]
15 Tw=293     //[K]
16 T_inf=333  //[K]
17 //Rate of heat transfer from BOTH sides is:
18 Q=2*h*A*(T_inf-Tw) // [W]
19 printf("Rate of heat transfer from both sides of
    plate is %f W\n",Q);

```

```

20 //ii-With turbulent boundary layer from the leading
    edge:
21 h=k*0.0366*(Nre_l^(0.8))*(Npr^(1.0/3.0))/L      //[
    W/(sq m.K)]
22 //Heat transfer from both sides is :
23 Q=2*h*A*(T_inf-Tw)                          //[W]
24 printf("\nThese calculations sho that the that
    transfer rate is approximately doubled if
    boundary layer is turbulent from the leading edge
    \n");

```

Scilab code Exa 3.11 Heat transfer from plate in unit direction

```

1  clc;
2  clear;
3  //Example 3.11
4  mu=1.906*10^-5      //[kg/(m.s)]
5  k=0.02723          //[W/m.K]
6  Cp=1.007           //[kJ/(kg.K)]
7  rho=1.129          //[kg/cubic m]
8  Npr=0.70
9  Mavg=29
10 u_inf=35           //[m/s]
11 L=0.75             //[m]
12 Tm=313             //[K]
13 P=101.325          //[kPa]
14 Nre_l=rho*u_inf*L/mu //Reynold's number >5*10^5
15 Nnu=0.0366*Nre_l^(0.8)*Npr^(1.0/3.0);
16 h=Nnu*k/L          //[W/s m.K]
17 A=1*L              //[sq m]
18 Tw=333             //[K]
19 T_inf=293           //[K]
20 Q=h*A*(Tw-T_inf);  //[W]
21 printf("Heat transfer from the plate is %f W",Q);

```

Scilab code Exa 3.12 Heat lost by sphere

```
1  clc ;
2  clear ;
3  //Example 3.12
4  v=18.23*10^-6 //sq m/s
5  k=0.02814 // [W/m.K]
6  D=0.012 // [m]
7  r=0.006 // [m]
8  u_inf=4 // [m/s]
9  Nre=D*u_inf/v //Reynold's number
10 Nnu=0.37*Nre^(0.6) ;
11 h=Nnu*(k/D)
12 A=4*pi*r^2 //Area of sphere in [sq m]
13 Tw=350 // [K]
14 T_inf=300 // [K]
15 Q=h*A*(Tw-T_inf) //Heat lost by sphere in [W]
16 printf(" \n Heat lost by sphere is %f W",Q);
```

Scilab code Exa 3.13 Heat lost by sphere

```
1  clc ;
2  clear ;
3  //Exmample 3.13
4  v=15.69*10^-6 // [sq m./s]
5  k=0.02624 // [W/m.K]
6  Npr=0.708 // Prandtl number
7  mu=2.075*10^-5 //kg/m.s
8  u_inf=4 // [m/s]
9  mu_inf=1.8462*10^-5 // [m/s] velocity
10 Tw=350 // [K]
11 T_inf=300 // [K]
```

```

12 D=0.012      // [m]
13 r=D/2        // Radius in [m]
14 Nre=u_inf*D/v // Reynold's numbe
15 Nnu=2+(0.4*Nre^(1.0/2.0)+0.06*Nre^(2.0/3.0))*Npr
    ^ (0.4)*(mu_inf/mu)^(1.0/4.0)
16 h=Nnu*k/D    // [W/sq m.K]
17
18 A=4*%pi*r^2  // Area in [sq m]
19 Q=h*A*(Tw-T_inf);
20 printf("\n Heat lost by the sphere is %f W",Q);

```

Scilab code Exa 3.14 Percent power lost in bulb

```

1  clc;
2  clear;
3  //Example 3.14
4  v=2.08*10^-5 // [sq m/s]
5  k=0.03       //W/(m.K)
6  Npr=0.697    // Prandtl number
7  D=0.06       // [m]
8  u_inf=0.3    // [m/s]
9  Nre=D*u_inf/v // Reynolds number
10 //Average nusselt number is given by:
11 Nnu=0.37*(Nre^0.6);
12 h=Nnu*k/D    //W/sq m.K
13 Tw=400      // [K]
14 T_inf=300   // [K]
15 D=0.06     // [m]
16 r=0.03     // [m]
17 A=4*%pi*r^2 // Area in [sq m]
18 Q=h*A*(Tw-T_inf) // [W]
19 per=Q*100/100 //Percent of heat lost by forced
    convection
20 printf("Heat transfer rate is %f W,And percentage of
    power lost by convectio is: %f percent ",Q,per);

```

Scilab code Exa 3.15 Heat lost by cylinder

```
1
2 clc;
3 clear;
4 //Example 3.15
5 u_inf=50 //velocity in [m/s]
6 mu=2.14*10^-5 // [kg/(m.s)]
7 rho=0.966 // [kg/cubic m]
8 k=0.0312 // [W/(m.K)]
9 Npr=0.695 //Prandtl number
10 D=0.05 //Diameter in [m]
11 Nre=D*u_inf*rho/mu ; //Reynold's number
12 printf(" %f",Nre)
13 Nnu=0.0266*Nre^0.805*Npr^(1/3);
14 h=Nnu*k/D ; //W/sq m.K
15 h=171.7 //Approximation
16 printf(" \n%f",h)
17 Tw=423 // [K]
18 T_inf=308 // [K]
19 //Heat loss per unit length is :
20 Q_by_l=h*pi*D*(Tw-T_inf); // [W]
21 printf("Heat lost per unit length of cylinder is %f
    W(approx)",round(Q_by_l));
```

Scilab code Exa 3.16 Heat transfer in tube

```
1
2 clc;
3 clear;
4 //Example 3.16
```

```

5 v=20.92*10^-6 //sq m/s
6 k=3*10^-2 //W/(m.K)
7 Npr=0.7
8 u_inf=25 // [m/s]
9 d=50 // [mm]
10 d=d/1000 // [m]
11 Nre=u_inf*d/v //Reynold's number
12 Tw=397 // [K]
13 T_inf=303 // [K]
14
15 //Case 1: Circular tube
16
17 Nnu=0.0266*Nre^(0.805)*Npr^(1.0/3.0);
18 h=Nnu*k/d // [W/sq m.K]
19 A=%pi*d //Area in [sq m]
20 Q=h*A*(Tw-T_inf) // [W]
21 Q_by_l1=h*%pi*d*(Tw-T_inf) // [W/m]
22
23 //Case 2:Square tube
24 A=50*50 //Area in [sq mm]
25 P=2*(50+50) //Perimeter [mm]
26 l=4*A/P // [mm]
27 l=l/1000 // [m]
28 Nnu=0.102*(Nre^0.675)*(Npr^(1.0/3.0))
29 h=Nnu*k/d //W/(sq m.K)
30 A=4*l*l // [sq m]
31
32 Q=h*A*(Tw-T_inf)
33 Q_by_l2=Q/l // [W/m]
34 printf("\nRate of heat flow from the square pipe=%f
W/m \n which is more than that from the circular
pipe which is equal to %f W/m",Q_by_l2,Q_by_l1);

```

Scilab code Exa 3.17 Heat transfer coefficient

```

1
2 clc;
3 clear;
4 //Example 3.17
5 mu=0.8 //Viscosity of flowing fluid [N.s/sq m]
6 rho=1.1 //Density of flowinf fluid [g/cubic cm]
7 rho=rho*1000 //Density in [kg/cubic m]
8 Cp=1.26 //Specific heat [kJ/kg.K]
9 Cp=Cp*10^3 // in [J/(kg.K)]
10 k=0.384 // [W/(m.K)]
11 mu_w=1 //Viscosity at wall temperature [N.s/sq m]
12 L=5 // [m]
13 vfr=300 //Volumetric flow rate in [cubic cm/s]
14 vfr=vfr*10^-6 // [cubic m/s]
15 mfr=vfr*rho //Mass flow rate of flowinf fluid [kg
/s]
16 Di=20 //Inside diameter in [mm]
17 Di=Di/1000 // [m]
18 Area=(%pi/4)*Di^2 //Area of cross-section [sq m]
19 u=vfr/Area //Veloctiy in [m/s]
20 Nre=Di*u*rho/mu //Reynold's number
21 //As reynold's number is less than 2100,he flow is
laminar
22 Npr=Cp*mu/k //Prandtl number
23 Nnu=1.86*(Nre*Npr*Di/L)^(1.0/3.0)*(mu/mu_w)^(0.14)
24 hi=Nnu*k/Di //inside heat transfer coefficient [W
/sq m.K]
25 printf("Inside heat transfer coefficient is %f W/(sq
m.K)",hi);
26 //Note:
27 printf("\\n The answer given in book..ie 1225 is
wrong.please redo the calculation of last line
manually to check\\n");

```

Scilab code Exa 3.18 Heat transfer coefficient in heated tube

```

1  clc;
2  clear;
3  //Example 3.18
4  m=5500    //Mass flow rate in [kg/h]
5  m=m/3600  //[kg/s]
6  rho=1.07  //Density of fluid in [g/cm^3]
7  rho=rho*1000  //[kg/m^3]
8  vfr=m/rho  //Volumetric flow rate in [m^3/s]
9  Di=40     //Diameter of tube [mm]
10 Di=Di/1000  //[m]
11 A=(%pi/4)*Di^2  //Area of cross-section in [sq m]
12 u=vfr/A  //Velocity of flowing fluid [m/s]
13 rho=1070  //Density in [kg/m^3]
14 mu=0.004  //Viscosity in [kg/m.s]
15 Nre=Di*u*rho/mu
16 Nre=12198  //Approx
17 //Since this reynold's number is less than 10000,the
    flow is turbulent
18 Cp=2.72  //Specific heat in [kJ/kg.K]
19 Cp=Cp*10^3  //Specific heat in [J/kg.K]
20 k=0.256  //thermal conductivity in [W/m.K]
21 Npr=Cp*mu/k  //Prandtl number
22 Nnu=0.023*(Nre^0.8)*(Npr^0.4)  //Nusselt number
23 hi=k*Nnu/Di  //Inside heat transfer coefficient in
    [W/m^2.K]
24 printf("Inside heat transfer coefficient is %f W/sq
    m.K",hi);

```

Scilab code Exa 3.19 h of water flowing in tube

```

1
2  clc;
3  clear;
4  //Example 3.19
5

```

```

6 //DATA:
7 rho=984.1 //Density of water [kg/m^3]
8 Cp=4187 //Specific heat in [J/kg.K]
9 mu=485*10^-6 //Viscosity at 331 K[Pa.s]
10 k=0.657 // [W/(m.K)]
11 mu_w=920*10^-6 //Viscosity at 297 K [Pa.s]
12 //Solution
13 D=16 //Diameter in [mm]
14 D=D/1000 //Diameter in [m]
15 u=3 //Velocity in [m/s]
16 rho=984.1 // [kg/m^3]
17 Nre=D*u*rho/mu //Reynolds number
18 Nre=round(Nre)
19 Npr=Cp*mu/k //Prandtl number
20
21 //Dittus-Boelter equation (i)
22 Nnu=0.023*(Nre^0.8)*(Npr^0.3) //nusselt number
23 h=k*Nnu/D //Heat transfer coefficient [W/m^2.K]
24 printf("\nANSWER-(i) \nBy Dittus-Boelter equation we
get h=%f W/sq m.K\n\n",h);
25
26 //sieder-tate equation (ii)
27 Nnu=0.023*(Nre^0.8)*(Npr^(1.0/3.0))*((mu/mu_w)^0.14)
//Nusselt number
28 h=k*Nnu/D //Heat transfer coefficient in [W/sq m.
K]
29 printf("\nAnswer-(ii)\n-By Sieder-Tate equation we
get h=%f W/sq m.K\n",h);
30 printf("\nNOTE: Calculation mistake in book in part 2
ie sieder tate eqn\n")

```

Scilab code Exa 3.20 Overall heat transfer coefficient

```

1 clc;
2 clear;

```

```

3 //Example 3.20
4 m_dot=2250 //Mass flow arte in [kg/h]
5 Cp=3.35 //Specific heat in [kJ/(kg.K)]
6 dT=316-288.5 //Temperature drop for oil [K]
7 Q=Cp*m_dot*dT //Rate of heat transfer in [kJ/h]
8 Q=round(Q*1000/3600) //[J/s] or [W]
9 Di=0.04 //Inside diameter [m]
10 Do=0.048 //Outside diamter in [m]
11 hi=4070 //for steam [W/sq m.K]
12 ho=18.26 //For oil [W/sq m.K]
13 Rdo=0.123 //[sq m.K/W]
14 Rdi=0.215 //[sq m.K/W]
15 Uo=1/(1/ho+Do/(hi*Di)+Rdo+Rdi*(Do/Di)) // [W/m^2.K]
    ]
16 Uo=2.3
17 dT1=373-288.5 // [K]
18 dT2=373-316 // [K]
19 dTm=(dT1-dT2)/log(dT1/dT2) // [K]
20 Ao=Q/(Uo*dTm) //Heat transfer area in [m^2]
21 printf("Heatr transfer area is:%f m^2",Ao);

```

Scilab code Exa 3.21 Number of tubes in exchanger

```

1
2 clc;
3 clear;
4 //Example 3.21
5 k_tube=111.65 // [W/m.K]
6 W=4500 // [kg/h]
7 rho=995.7 // [kg/sq m]
8 Cp=4.174 // [kJ/(kg.K)]
9 k=0.617 // [W/(m.K)]
10 v=0.659*10^-6 //Kinematic viscosity [sq m/s]
11 m_dot=1720 //kg/h
12 T1=293 //Initial temperature in [K]

```

```

13 T2=318 //Final temperature in [K]
14 dT=T2-T1 // [K]
15 Q=m_dot*Cp*dT //Heat transfer rate in [kJ/h]
16 Q=Q*1000/3600 // [J/s] or [W]
17 Di=0.0225 // [m]
18 u=1.2 // [m/s]
19 //Nre=Di*u*rho/mu or
20 Nre=Di*u/v //Reynolds number
21 //As Nre is greater than 10000,Dittus Boelter
    equation is applicable
22 Cp=Cp*10^3 //J/(kg.K)
23 mu=v*rho // [kg/(m.s)]
24 Npr=Cp*mu/k //Prandtl number
25 //Dittus-Boelter equation for heating is
26 Nnu=0.023*(Nre^0.8)*(Npr^0.4)
27 hi=k*Nnu/Di //Heat transfer coefficient [W/(sq m
    .K)]
28 Do=0.025 // [m]
29 Dw=(Do-Di)/log(Do/Di) //Log mean diameter in [m]
30 ho=4650 // [W/sq m.K]
31 k=111.65 // [W/m.K]
32 xw=(Do-Di)/2 // [m]
33 Uo=1/(1/ho+Do/(hi*Di)+xw*Do/(k*Dw)) // Overall
    heat transfer coefficient in W/(m^2.K)
34 T_steam=373 //Temperature of condensing steam in
    [K]
35 dT1=T_steam-T1+10 // [K]
36 dT2=T_steam-T2+10 // [K]
37 dTm=(dT1-dT2)/log(dT1/dT2) // [K]
38 Ao=Q/(Uo*dTm)//Area in [m^2]
39 L=4 //length of tube [m]
40 n=Ao/(%pi*Do*L) //number of tubes
41 printf("No. of tubes required=%d\n",round(n));
42 printf("\n NOTE: there is an error in book in
    calculation of dT1 and dT2,\n 373-293 is written
    as 90,instead of 80...similarly in dT2,\nSo,in
    compliance with the book,10 is added to both of
    them")

```

Scilab code Exa 3.22 Convective film coefficient

```
1  clc;
2  clear;
3  //Example 3.22
4  m_dot=25000 //massflow rate of water [kg/h]
5  rho=992.2 // [kg/m^3]
6  k=0.634 // [W/m.K]
7  vfr=m_dot/rho // [m^3/h]
8  Npr=4.31 //Prandtl numberl
9  Di=50 // [mm]
10 Di=0.05 // [m]
11 dT=10 // [K] as the wall is at a temperature of 10
    K above the bulk temperature
12 u=(vfr/3600)/(%pi*(Di/2)^2) //Velocity of water
    in [m/s]
13 u=3.56 //Approximation
14 //Nre=Di*u*rho/mu=Di*u/v as v=mu/rho
15 v=0.659*10^-6; // [m^2/s]
16 Nre=Di*u/v //Reynolds number
17 //As it is less than 10000,the flow is in the
    turbulent region for heat transfer and Dittus
    Boelter eqn is used
18 Nnu=0.023*(Nre^0.8)*(Npr^0.4); //Nusselt number
19 hi=Nnu*k/Di //Heat transfer coefficiet in [W/sq m
    .K]
20 q_by_l=hi*%pi*Di*dT //Heat transfer per unit
    length [kW/m]
21 printf("Average value of convective film coefficient
    is hi= %d W/sq m.K \nHeat transferred per unit
    length is Q/L=%f kW/m",round(hi),q_by_l/1000);
```

Scilab code Exa 3.23 Length of tube

```

1  clc;
2  clear;
3  //Example 3.23
4  vfr=1200 ; //Water flow rate in [l/h]
5  rho=0.98 ; //Density of water in g/[cubic cm]
6  m_dot=vfr*rho //Mass flow rate of water [kg/h]
7  m_dot2=m_dot/3600 // [kg/s]
8  Cp=4.187*10^3 ; // [J/kg.K]
9  Di=0.025 ; //Diameter in [m]
10 mu=0.0006 ; // [kg/(m.s)]
11 Ai=%pi*((Di/2)^2) //Area of cross-section in [m
    ^2]
12 Nre=(Di/mu)*(m_dot2/Ai) //Reynolds number
13 k=0.63 ; //for metal wall in [W/(m.K)]
14 Npr=Cp*mu/k; //Prandtl number
15 //Since Nre>10000
16 //therefore ,Dittus boelter eqn for heating is
17 Nnu=0.023*(Nre^(0.8))*(Npr^(0.4))
18 ho=5800 ; //Film heat coefficientW/(m^2.K)
19 hi=Nnu*k/Di //Heat transfer coeffcient in [W/(sq
    m.K)]
20 Do=0.028 ; // [m]
21 Di=0.025 ; // [m]
22 xw=(Do-Di)/2; // [m]
23 Dw=(Do-Di)/log(Do/Di); // [m]
24 k=50 ; //for metal wall in [W/(m.K)]
25 Uo=1/(1/ho+Do/(hi*Di)+xw*Do/(k*Dw)); //in [W/sq m
    .K]
26 dT=343-303 ; // [K]
27 dT1=393-303 ; // [K]
28 dT2=393-343 ; // [K]
29 dTm=(dT1-dT2)/log(dT1/dT2); // [K]
30 Cp=Cp/1000; // [in [kJ/kg.K]]
31 Q=m_dot*Cp*dT; //Rate of heat transfer in [kJ/h]
32 Q=Q*1000/3600; // [J/s] or [W]
33 Ao=Q/(Uo*dTm); //Heat transfer area in [sq m]

```

```

34 // Also ,...Ao=%pi*Do*L    ..implies that
35 L=Ao/(%pi*Do)           // [m]
36 printf("Length of tube required is %f m",round(L));

```

Scilab code Exa 3.24 Cooling coil

```

1  clc;
2  clear;
3  //Example 3.24
4  //1.For initial conditions:
5  T=360;           // [K]
6  T1=280;          // [K]
7  T2=320;          // [K]
8  dT1=T-T1;        // [K]
9  dT2=T-T2;        // [K]
10 //Q1=m1_dot*Cp1*(T2-T1)
11 Cp1=4.187         //Heat capacity
12 dT1m=(dT1-dT2)/log(dT1/dT2) // [K]
13 m1_by_UA=dT1m/(Cp1*(T2-T1))
14 //For final conditions :
15 //m2_dot=m1_dot
16 //U2=U1
17 //A2=5*A1
18 def('x=f(t)', 'x=m1_by_UA*Cp1*(t-T1) -5*((dT1-(T-t))/log(dT1/(T-t)))')
19 T=fsolve(350.5,f)
20 printf("\nOutlet temperature of water is %f K\n",T);

```

Scilab code Exa 3.25 Outlet temperature of water

```

1  clc;
2  clear;
3  //Example 3.25

```

```

4 mo_dot=60 //Mass flow rate of oilin [g/s]
5 mo_dot=6*10^-2 // [kg/s]
6 Cpo=2.0 //Specific heat of oil in [kJ/(kg.K)]
7 T1=420 // [K]
8 T2=320 // [K]
9 Q=mo_dot*Cpo*(T1-T2) //Rate of heat flow in [kJ/s
]
10 mw_dot=mo_dot //Mass flow rate of water //kg/s
11 t1=290 // [K]
12 Cpw=4.18 // [kJ/(kg.K)]
13 //For finding outlet temperature of water
14 t2=t1+Q/(mw_dot*Cpw) // [K]
15 dT1=T1-t2 // [K]
16 dT2=T2-t1 // [K]
17 dTm=(dT1-dT2)/log(dT1/dT2) // [K]
18 ho=1.6 //Oil side heat transfer coefficient in [kW
/(sq m.K)]
19 hi=3.6 //Water side heat transfer coeff in [kW/(sq
m.K)]
20 //Overall heat transfer coefficient is:
21 U=1/(1/ho+1/hi) // [kW/(m^2.K)]
22
23 A=Q/(U*dTm) // [sq m]
24 Do=25 // [mm]
25 Do=Do/1000 // [m]
26 L=A/(%pi*Do) //Length of tube in [m]
27 printf("\nOutlet temperature of water is %f K \n",
round(t2));
28 printf("Area of heat transfer required is %f sq m\n"
,A);
29 printf("Length of tube required is %f m",L)

```

Scilab code Exa 3.26 Inside heat transfer coefficient

1

```

2  clc;
3  clear;
4  //Example 3.26
5  k=0.14 // for oil [W/m.K]
6  Cp=2.1 // for oil [kJ/kg.K]
7  Cp=Cp*10^3 //J/kg.K
8  mu=154 // [mN.s/sq m]
9  mu_w=87 //(mm.s/sq m)
10 L=1.5 // [m]
11 m_dot=0.5 //Mass flow rate of oil [kg/s]
12 Di=0.019 //Diameter of tube [m]
13 mean_T=319 //Mean temperature of oil [K]
14 mu=mu*10^-3 //[N.s/sq m] or [kg/(m.s)]
15 A=%pi*(Di/2)^2 //[sq m]
16 G=m_dot/A //Mass velocity in [kg/sq m.s]
17 Nre=Di*G/mu //Reynolds number
18 //As Nre<2100,the flow is laminar
19 mu_w=mu_w*10^-3 //[N.s/sq m] or kg/(m.s)
20 //The siedler tate equation is
21 hi=(k*(2.0*((m_dot*Cp)/(k*L))^(1.0/3.0)*(mu/mu_w)
    ^ (0.14)))/Di //Heat transfer coeff in [W/sq m.K
    ]
22 printf(" \n The inside heat transfer coefficient is
    %f W/(m^2.K) ",hi);
23
24 printf(' \nNOTE: Calculation mistake in last line.ie
    in the calculation of hi in book,please perform
    the calculation manually to check the answer\n")

```

Scilab code Exa 3.27 Film heat transfer coefficient

```

1  clc;
2  clear;
3  //Example 3.27
4

```

```

5 m_dot=0.217 //Water flow rate in [kg/s]
6 Do=19 //Outside diameter in [mm]
7 rho=1000 //Density
8 t=1.6 //Wall thickness in [mm]
9 Di=Do-2*t //i.d of tube in [mm]
10 Di=Di/1000 // [m]
11 Do=Do/1000 // [m]
12 Ai=%pi*(Di/2)^2 //Cross-sectional area in sq m
13 u=m_dot/(rho*Ai) //Water velocity through tube [m/
    s]
14 u=1.12 //approx in book
15 Di=0.0157 //apprx in book
16 T1=301 //Inlet temperature of water in [K]
17 T2=315 //Outlet temperature of water in [K]
18 T=(T1+T2)/2 // [K]
19 hi=(1063*(1+0.00293*T)*(u^0.8))/(Di^0.20) //Inside
    heat transfer coefficient W/(sq m.K)
20 hi=5084 //Approximation
21 printf("%f",hi);
22 hio=hi*(Di/Do) //Inside heat transfer coeff based
    on outside diameter in W/(sq m.K)
23 printf("%f",hio);
24 printf("Based on outside temperature,Inside heat
    transfer coefficient is %d W/(m^2.K) or %f kW/(m
    ^2.K)",round(hio),round(hio)/1000);

```

Scilab code Exa 3.28 Area of exchanger

```

1 clc;
2 clear;
3 //Example 3.28
4 mair_dot=0.90 // [kg/s]
5 T1=283 // [K]
6 T2=366 // [K]
7 dT=(T1+T2)/2 // [K]

```

```

8 Di=12 // [mm]
9 Di=Di/1000 // [m]
10 G=19.9 // [kg/(sq m. s)]
11 mu=0.0198 // [mN. s/(sq m)]
12 mu=mu*10^-3 // [N. s/sq m] or [kg/(m. s)]
13 Nre=Di*G/mu // Reynolds number
14 // It is greater than 10^4
15 k=0.029 //W/(m.K)
16 Cp=1 // [kJ/kg.K]
17 Cp1=Cp*10^3 // [J/kg.K]
18 Npr=Cp1*mu/k // Prandtl number
19 // Dittus-Boelter equation is
20 hi=0.023*(Nre^0.8)*(Npr^0.4)*k/Di // [W/sq m.K]
21 ho=232 //W/sq m.K
22 U=1/(1/hi+1/ho) // Overall heat transfer coefficient
    [W/m^2.K]
23 Q=mair_dot*Cp*(T2-T1) //kJ/s
24 Q=Q*10^3 // [J/s] or [W]
25 T=700 // [K]
26 dT1=T-T2 // [K]
27 dT2=T2-T1 // [K]
28 dTm=(dT1-dT2)/log(dT1/dT2) // [K]
29 //Q=U*A*dTm
30 A=Q/(U*dTm) // Area in sq m
31 printf("Heat transfer area of equipment is %f sq m",
    A);

```

Scilab code Exa 3.29 Natural and forced convection

```

1 clc;
2 clear;
3 //Example 3.29
4 v=18.41*10^-6 // [sq m./s]
5 k=28.15*10^-3 // [W/m.K]
6 Npr=0.7 // Prandtl number

```

```

7 Beta=3.077*10^-3 //K^-1
8 g=9.81 //m/s^2
9 Tw=350 // [K]
10 T_inf=300 // [K]
11 dT=Tw-T_inf // [K]
12 L=0.3 // [m]
13 //1.Free Convection
14 Ngr=(g*Beta*dT*L^3)/(v^2) //Grashof number
15 Npr=0.7 //Prandtl number
16
17 Nnu=0.59*(Ngr*Npr)^(1.0/4.0) //Nusselt number
18 h=Nnu*k/L //Average heat transfer coefficient [W/
sq m K]
19 printf("\n In free convection ,heat transfer coeff ,h=
%f W/(sq m.K)\n",h)
20 //2.Forced Convection
21 u_inf=4 // [m/s]
22 Nre_l=u_inf*L/v
23 Nnu=0.664*(Nre_l^(1/2))*(Npr^(1.0/3.0)) //
Nusselt number
24 h=Nnu*k/L // [W/sq m.K]
25 printf("\n In forced convection ,heat transfer coeff ,
h=%f W/(sq m.K)\n",h)
26 printf("\n From above it is clear that heat transfer
coefficient in forced convection is much larger
than that in free convection \n ");

```

Scilab code Exa 3.30 Natural convection

```

1
2 clc;
3 clear;
4 //Example 3.30
5 k=0.02685 //W/(m.K)
6 v=16.5*10^-6 //kg/(m.s)

```

```

7 Npr=0.7 //Prandtl number
8 Beta=3.25*10^-3 //K^-1
9 g=9.81 //m/(s^2)
10 Tw=333; //[k]
11 T_inf=283 // [K]
12 dT=Tw-T_inf // [K]
13 L=4 //Length/height of plate [m]
14 Ngr=(g*Beta*dT*(L^3))/(v^2) //Grashoff number
15 //Let const=Ngr*Npr
16 const=Ngr*Npr
17 //Sice it is >10^9
18 Nnu=0.10*(const^(1.0/3.0)) //Nusselt number
19 h=Nnu*k/L //W/(sq m.K)
20 h=4.3 //Approx in book
21 W=7 //width in [m]
22 A=L*W //Area of heat transfer in [sq m]
23 Q=h*A*dT // [W]
24 printf("\nHeat transferred is %d W\n",Q)

```

Scilab code Exa 3.31 Free convection in vertical pipe

```

1 clc;
2 clear;
3 //Example 3.31
4 v=18.97*10^-6 //m^2/s
5 k=28.96*10^-3 //W/(m.K)
6 Npr=0.696
7 D=100 //Outer diameter [mm]
8 D=D/1000 // [m]
9 Tf=333 //Film temperature in [K]
10 Tw=373 // [K]
11 T_inf=293 // [K]
12 dT=Tw-T_inf // [K]
13 Beta=1/Tf // [K^-1]
14 g=9.81 // [m/s^2]

```



```

15 L=3 //Length of pipe [m]
16 Ngr=(g*Beta*dT*(L^3))/(v^2) //Grashof number
17 Nra=Ngr*Npr
18 Nnu=0.10*(Ngr*Npr)^(1.0/3.0) //nusselt number for
    vertical cylinder
19 h=Nnu*k/L //W/(sq m.K)
20 Q_by_l=h*pi*D*dT //Heat loss per metre length [W/
    m]
21 printf("\n Hence, Heat loss per metre length is %f W/
    m \n", Q_by_l);

```

Scilab code Exa 3.32 Heat loss per unit length

```

1 clc;
2 clear;
3 //Example 3.32
4 k=0.630 //W/(m.K)
5 Beta=3.04*10^-4 //K^-1
6 rho=1000 //kg/m^3
7 mu=8.0*10^-4 // [kg/(m.s)]
8 Cp=4.187 //kJ/(kg.K)
9 g=9.81 // [m/(s^2)]
10 Tw=313 // [K]
11 T_inf=298 // [K]
12 dT=Tw-T_inf // [K]
13 D=20 // [mm]
14 D=D/1000 // [m]
15 Ngr=9.81*(rho^2)*Beta*dT*(D^3)/(mu^2) //Grashoff
    number
16 Cp1=Cp*1000 // [J/kg.K]
17 Npr=Cp1*mu/k //Prandtl number
18 //Average nusselt number is
19 Nnu=0.53*(Ngr*Npr)^(1.0/4.0)
20 h=Nnu*k/D // [W/ sqm.K]
21 Q_by_l=h*pi*D*dT //Heat loss per unit length [W/m]

```

```

]
22 printf("\nHeat loss per unit length of the heater is
    %f W/m", Q_by_1);

```

Scilab code Exa 3.33 Free convection in pipe

```

1  clc;
2  clear;
3  //Example 3.33
4  k=0.03406 // [W/(m/K)]
5  Beta=2.47*10^-3 //K^-1
6  Npr=0.687 //Prandtl number
7  v=26.54*10^-6 //m^2/s
8  g=9.81 // [m/s^2]
9  Tw=523 // [K]
10 T_inf=288 // [K]
11 dT=Tw-T_inf // [K]
12 D=0.3048 // [m]
13 Ngr=(g*Beta*dT*(D^3))/(v^2) //Grashof number
14 Nra=Ngr*Npr
15 //For Nra less than 10^9,we have for horizontal
    cylinder
16 Nnu=0.53*(Nra^(1.0/4.0)) //Nusselt number
17 h=Nnu*k/D // [W/sq m.K]
18 Q_by_1=h*pi*D*dT; //W/m
19 printf("Heat loss of heat transfer per meter lengh
    is %f W/m", Q_by_1);

```

Scilab code Exa 3.34 Free convection in plate

```

1
2  clc;
3  clear;

```

```

4 //Example 3.34
5 rho=960.63 //Density in [kg/m^3]
6 Cp=4.216*10^3 //Specific heat in [J/(kg.K)]
7 D=16 //Diameter in [cm]
8 D=D/100 // [m]
9 k=0.68 //Thermal conductivity in [W/m.K]
10 A=(%pi*(D/2)^2)
11 L=A/(%pi*D) //Length=A/P in [m]
12 Beta=0.75*10^-3 // [K^-1]
13 alpha=1.68*10^-7 // [m^2/s]
14 g=9.81 // [m/s^2]
15 Tw=403 // [K]
16 T_inf=343 // [K]
17 dT=Tw-T_inf // [K]
18 v=0.294*10^-6 // [m^2/s]
19 Nra=(g*Beta*(L^3)*dT)/(v*alpha)
20
21 //1.For Top surface
22 Nnu=0.15*(Nra)^(1.0/3.0) //Nusselt number
23 ht=Nnu*k/L //Heat transfer coeff for top surface in
    W/(m^2.K)
24 ht=round(ht)
25 //2.For bottom surface
26 Nnu=0.27*Nra^(1.0/4.0) //Nusselt number
27 hb=Nnu*k/L // [W/sq m.K]
28 hb=round(hb)
29 Q=(ht+hb)*A*dT; // [W]
30 printf("The rate of heat input is %f W",Q)

```

Scilab code Exa 3.35 Heat transfer from disc

```

1 clc;
2 clear;
3 //Example 3.35
4 v=2*10^-5 // [m^2/s]

```

```

5 Npr=0.7 //Prandtl number
6 k=0.03 // [W/m.K]
7 D=0.25 //Diameter in [m]
8 L=0.90*D // Characteristic length ,let [m]
9 T1=298 // [K]
10 T2=403 // [K]
11 dT=T2-T1 // [K]
12 Tf=(T1+T2)/2 // [K]
13 Beta=1/Tf // [K^-1]
14 A=%pi*(D/2)^2 //Area in [sq m]
15 g=9.81 // [m/s^2]
16
17 //Case 1: Hot surface facing up
18 Ngr=g*Beta*dT*(L^3)/(v^2) //Grashoff number
19 Nnu=0.15*((Ngr*Npr)^(1.0/3.0)) //Nusselt number
20 h=Nnu*k/L // [W/sq m.K]
21 Q=h*A*dT // [W]
22 printf("\n Heat transferred when hot surface is
    facing up is %f W\n",Q);
23
24
25 //Case 2:For hot surface facing down
26 Nnu=0.27*(Ngr*Npr)^(1.0/4.0); //Grashof Number
27 h=Nnu*k/L // [W/sqm.K]
28 Q=h*A*dT // [W]
29 printf("\n Heat transferred when hot surface is
    facing down is %f W\n",Q);

```

Scilab code Exa 3.36 Rate of heat input to plate

```

1
2
3 clc;
4 clear;
5 //Example 3.36

```

```

6 rho=960 // [kg/m^3]
7 Beta=0.75*10^-3 // [K^-1]
8 k=0.68 // [W/m.K]
9 alpha=1.68*10^-7 // [m^2/s]
10 v=2.94*10^-7 // [m^2/s]
11 Cp=4.216 // [kJ/kg.K]
12 Tw=403 // [K]
13 T_inf=343 // [K]
14 dT=Tw-T_inf // [K]
15 g=9.81 // [m/s^2]
16 l=0.8 // [m]
17 W=0.08 // [m]
18 A=l*W //Area in [m^2]
19 P=2*(0.8+0.08) //Perimeter in [m]
20 L=A/P //Characteristic dimension/length,L in [m]
21 Nra=g*Beta*L^3*dT/(v*alpha)
22
23 //(i) for natural convection,heat transfer from top/
    upper surface heated
24 Nnu=0.15*(Nra^(1.0/3.0)) //Nusselt number
25 ht=Nnu*k/L // [W/m^2.K]
26 ht=2115.3 //Approximation in book,If done manually
    then answer diff
27 //(ii)For the bottom/lower surface of the heated
    plate
28 Nnu=0.27*(Nra^(1.0/4.0)) //Nusselt number
29 hb=Nnu*k/L // [W/(m^2.K)]
30 hb=round(hb)
31 //Rate of heat input is equal to rate of heat
    dissipation from the upper and lower surfaces of
    the plate
32 Q=(ht+hb)*A*(Tw-T_inf) // [W]
33 printf("\n Rate of heat input is equal to heat
    dissipation =%f W",Q);

```

Scilab code Exa 3.37 Two cases in disc

```

1  clc;
2  clear;
3  //Example 3.37
4  k=0.03 //W/(m.K)
5  Npr=0.697 //Prandtl number
6  v=2.076*10^-6 //m^2/s
7  Beta=0.002915 //K^-1
8  D=25 ; // [Diameter in cm]
9  D=D/100 // [m]
10 Tf=343 //Film temperature in [K]
11 A=%pi*(D/2)^2 //Area in [m^2]
12 P=%pi*D //Perimeter [m]
13 T1=293 // [K]
14 T2=393 // [K]
15 g=9.81 // [m/s^2]
16
17 //Case (i) HOT SURFACE FACING UPWARD
18 L=A/P //Characteristic length in [m]
19 Beta=1/Tf; // [K^-1]
20 dT=T2-T1 // [K]
21 Ngr=(g*Beta*dT*(L^3))/(v^2) //Grashoff number
22 Nra=Ngr*Npr
23 Nnu=0.15*(Nra^(1.0/3.0)) //Nusselt number
24 h=Nnu*k/L // [W/m^2.K]
25 Q=h*A*dT // [W]
26 printf("\\nHeat transferred when disc is horizontal
    with hot surface facing upward is %f W\\n",Q);
27
28 //Case-(ii) HOT FACE FACING DOWNWARD
29 Nnu=0.27*(Nra^(1/4)) //Nusselt number
30 h=Nnu*k/L //W/(m^2.K)
31 Q=h*A*dT // [W]
32 printf("\\nHeat transferred when disc is horizontal
    with hot surface facing downward is %f W\\n",Q);
33
34

```

```

35 //Case-(iii)-For disc vertical
36 L=0.25 //Characteristic length [m]
37 D=L //dia [m]
38 A=%pi*((D/2)^2) //[sq m]
39 Ngr=(g*Beta*dT*(L^3))/(v^2) //Grashoff number
40 Npr=0.697
41 Nra=Ngr*Npr
42 Nnu=0.10*(Nra^(1/3)) //Nusselt number
43 h=Nnu*k/D // [W/(m^2.K)]
44 Q=h*A*dT // [W]
45 printf("For vertical disc ,heat transferred is %f W",
Q);

```

Scilab code Exa 3.38 Total heat loss in a pipe

```

1 clc;
2 clear;
3 //Example 3.38
4 v=23.13*10^-6 ; // [m^2/s]
5 k=0.0321 ; // [W/m.K]
6 Beta=2.68*10^-3; // [K^-1]
7 Tw=443 ;// [K]
8 T_inf=303 ; // [K]
9 dT=Tw-T_inf; // [K]
10 g=9.81 ; // [m/s^2]
11 Npr=0.688; //Prandtl number
12 D=100 ; //Diameter [mm]
13 D=D/1000 //Diameter [m]
14 Nra=(g*Beta*dT*(D^3)*Npr)/(v^2)
15 Nnu=0.53*(Nra^(1.0/4.0)) //Nusselt number
16 h=Nnu*k/D // [W/(m^2.K)]
17 h=7.93 //Approximation
18 e=0.90; //Emissivity
19 sigma=5.67*10^-8 ;
20 //Q=Q_conv+Q_rad //Total heat loss

```

```

21 //for total heat loss per meter length
22 Q_by_l=h*%pi*D*dT+sigma*e*%pi*D*(Tw^4-T_inf^4) // [W
    /m]
23 printf("Total heat loss per metre length of pipe is
    %f W/m",Q_by_l)

```

Scilab code Exa 3.39 Heat loss by free convection

```

1  clc;
2  clear;
3  //Example 3.39
4  k=0.035; // [W/(m.K) ]
5  Npr=0.684 ;//Prandtl number
6  Beta=2.42*10^-3; // [K^-1]
7  v=27.8*10^-6; // [m^2/s ]
8  Tw=533; // [K]
9  T_inf=363 ; // [K]
10 dT=Tw-T_inf // [K]
11 D=0.01 ;// [m]
12 g=9.81; // [m/s ^2]
13 Nra=(g*Beta*dT*(D^3))/(v^2)
14 //For this <10^5,we have for sphere
15 A=4*%pi*(D/2)^2 //Area of sphere in [m^2]
16 Nnu=(2+0.43*Nra^(1.0/4.0))//Nusslet number
17 h=Nnu*k/D //W/(m^2.K)
18 Q=h*A*dT // [W]
19 printf("\nRate of heat loss is %f W",Q)

```

Scilab code Exa 3.40 Heat loss from cube

```

1
2  clc;
3  clear;

```



```

4 //Exampe 3.40
5 v=17.95*10^-6 // [m^2/s]
6 dT=353-293 // [K]
7 k=0.0283 // [W/m.K]
8 g=9.81 // [m/s^2]
9 Npr=0.698 // Prandtl number
10 Cp=1005 //J/(kg.K)
11 Tf=323 //Film temperature in [K]
12 Beta=1/Tf // [K^-1]
13 l=1 // [m]
14 Nra=(g*Beta*dT*(l^3)*Npr)/(v^2)
15
16 //In textbook result of above statement is wrongly
    calculated ,So
17 Nra=3.95*10^8
18 //For Nra <10^9,for a vertical plate ,the average
    nusselt number is
19 Nnu=0.59*Nra^(1.0/4.0) //Nusselt number
20 h=Nnu*k/l // [W/m^2.K]
21 h=2.35 //Approx in book
22 A=l^2 //Area [m^2]
23 //Heat loss form 4 vertical faces of lm*lm is
24 Q1=4*(h*A*dT) // [W]
25 //For top surface
26 P=4*l //Perimeter in [m]
27 L=A/P // [m]
28 Nra=(Npr*g*Beta*dT*(L^3))/(v^2)
29 Nnu=0.15*Nra^(1.0/3.0) //Nusselt number
30 h=Nnu*k/L // [W/m^2.K]
31 h=6.7 //Approx
32 Q2=h*A*dT // [W]
33 Q_total=Q1+Q2 //Total heat loss [W]
34 printf("\n Therefore total heat loss is %d W",
    Q_total);

```

Scilab code Exa 3.41 Plate exposed to heat

```
1  clc ;
2  clear ;
3  //Example 3.41
4  rho=0.910;           //Density in [kg/m^3]
5  Cp=1.009*1000;      // [J/kg.K]
6  k=0.0331;           // [W/m.K]
7  mu=22.65*10^-6;     // [N.s/m^2]
8  //Let a=smaller side
9  //b=bigger side
10 //Qa=ha*A*dT
11 //Qb=hb*A*dT
12 //Qa=1.14*Qb
13 //Given a*b=15*10^-4
14 //On solving we get:
15 a=0.03;             // [m]
16 b=0.05;             // [m]
17 A=a*b               //Area in [sq m]
18 Tf=388;             // [K]
19 Beta=1/Tf           // [K^-1]
20 T1=303;             // [K]
21 T2=473;             // [K]
22 dT=T2-T1           // [K]
23 v=mu/rho
24 g=9.81              //m/s^2[acceleration due to gravity ]
25 hb=0.59*(((g*Beta*dT*(b^3))/(v^2))*Cp*mu/k)^(1/4)*(k
    /b)                // [W/sq m.K]
26 Qb=hb*A*(dT)       // [W]
27
28 Qa=1.14*Qb         // [W]
29 printf(" \nDimensions of the plate are %fx%f m\n",a,b
    );
30 printf(" \nHeat transfer when the bigger side held
    vertical is %f W\n",Qb);
31 printf(" \nHeat transfer when the small side held
    vertical is %f W\n",Qa);
```

Scilab code Exa 3.42 Nucleate poolboiling

```
1  clc ;
2  clear ;
3  //Example 3.42
4  Ts=373 // [K]
5  rho_l=957.9 //rho*l [kg/m^3]
6  Cpl=4217 // [J/kg.K]
7  mu_l=27.9*10^-5 // [kg/(m.s)]
8  rho_v=0.5955 // [kg/m^3]
9  Csf=0.013
10 sigma=5.89*10^-2 // [N/m]
11 Nprl=1.76
12 lambda=2257 // [kJ/kg]
13 lambda=lambda*1000 //in [J/kg]
14 n=1 //for water
15 m_dot=30 //Mass flow rate [kg/h]
16 m_dot=m_dot/3600 // [kg/s]
17 D=30 //Diameter of pan [cm]
18 D=D/100 // [m]
19 g=9.81 // [m/s^2]
20 A=%pi*(D/2)^2 //Area in [sq m]
21 Q_by_A=m_dot*lambda/A // [W/sq m]
22 //For nucleate boiling point we have:
23 dT=(lambda/Cpl)*Csf*(((Q_by_A)/(mu_l*lambda))*sqrt(
    sigma/(g*(rho_l-rho_v))))^(1.0/3.0)*(Nprl^n) // [K
    ]
24 Tw=Ts+dT // [K]
25 printf("\\n Temperature of the bottom surface of the
    pan is %f W/(sq m)",Tw);
```

Scilab code Exa 3.43 Peak Heat flux

```

1  clc;
2  clear;
3  //Example 3.4
4  lambda=2257 // [kJ/kg]
5  lambda=lambda*1000 //in [J/kg]
6  rho_l=957.9 //rho*l [kg/m^3]
7  rho_v=0.5955 // [kg/m^3]
8  sigma=5.89*10^-2 // [N/m]
9  g=9.81 // [m/s^2]
10 //Peak heat flux is given by
11 Q_by_A_max=(%pi/24)*(lambda*rho_v^0.5*(sigma*g*(
    rho_l-rho_v))^(1/4)) //W/m^2
12 Q_by_A_max=Q_by_A_max/(10^6) //MW/(sq m)
13 printf("\n Peak heat flux is %f MW/sq m", Q_by_A_max)
    ;

```

Scilab code Exa 3.44 Stable film pool boiling

```

1  clc;
2  clear;
3  //Example 3.44
4  rho_l=957.9 // [kg/m^3]
5  lambda=2257 // [kJ/kg]
6  lambda=lambda*10^3 // [J/kg]
7  rho_v=31.54 // [kg/m^3]
8  Cpv=4.64 // [kJ/kg.K]
9  Cpv=Cpv*10^3 // [J/kg.K]
10 kv=58.3*10^-3 // [W/(m.K)]
11 g=9.81 // [m/s^2]
12 mu_v=18.6*10^-6 // [kg/(m.s)]
13 e=1.0 // Emissivity
14 sigma=5.67*10^-8;
15 Ts=373 // [K]
16 Tw=628 // [K]
17 dT=Tw-Ts // [K]

```

```

18 D=1.6*10^-3 // [m]
19 T=(Tw+Ts)/2 // [K]
20 hc=0.62*((kv^3)*rho_v*(rho_l-rho_v)*g*(lambda+0.40*
    Cpv*dT)/(D*mu_v*dT))^(1.0/4.0) // Convective heat
    transfer coeff [W/sq m.K]
21 hr=e*sigma*(Tw^4-Ts^4)/(Tw-Ts) // Radiation heat
    transfer coeff in [W/sq m.K]
22 h=hc+(3/4)*hr // Total heat transfer coefficient W
    /(sq m.K)
23 Q_by_l=h*%pi*D*dT // Heat dissipation rate per unit
    length in [kW/m]
24 printf("\n Stable film boiling point heat transfer
    coefficient is %f W/(sq m.K)",h);
25 Q_by_l=Q_by_l/1000 // [kW/m]
26 printf("\n Heat dissipated per unit length of the
    heater is %f kW/m",Q_by_l);

```

Scilab code Exa 3.45 Heat transfer in tube

```

1 clc;
2 clear;
3 //Exmample 3.45
4 dT=10 // [K]
5 P=506.625 // [kPa]
6 P=P/10^3 // [Mpa]
7 D=25.4 // Diameter [mm]
8 D=D/1000 // [m]
9 h=2.54*(dT^3)*(%e^(P/1.551)) // [W/sq m.K]
10 //Q=h*%pi*D*L*dT
11 //Heat transfer rate per meter length of tube is
12 Q_by_l=h*%pi*D*dT // [W/m]
13 printf("\n Rate of heat transfer per 1m length of
    tube is %f W/m",round(Q_by_l));

```

Scilab code Exa 3.46 Nucleat boiling and heat flux

```
1  clc ;
2  clear ;
3  //Example 3.46
4  dT=8      // [K]
5  P=0.17    // [Mpa]
6  P=P*1000  // [kPa]
7  h1=2847   // [W/(sq m.K)]
8  P1=101.325 // [kPa]
9  h=5.56*(dT^3) // [W/sq m.K]
10 Q_by_A=h*dT // [W/sq m]
11 hp=h*(P/P1)^(0.4) // [W/sq m.K]
12 //Correponding heat flux is :
13 Q_by_A1=hp*dT // [W/sq m]
14 per=(Q_by_A1-Q_by_A)*100/Q_by_A //Percent increase
    in heat flux
15 printf("\nHeat flux when pressure is 101.325 kPa is
    %f W/sq m\n",Q_by_A);
16 printf("\n Percent increase in heat flux is %f
    percent",round(per));
```

Scilab code Exa 3.47 Dry steam condensate

```
1  clc ;
2  clear ;
3  //Example 3.47
4  mu=306*10^-6 // [N. s/m^2]
5  k=0.668 // [W/m.K]
6  rho=974 // [kg/m^3]
7  lambda=2225 // [kJ/kg]
8  lambda=lambda*10^3 // [J/kg.K]
```

```

 9 g=9.81 // [m/s ^2]
10 Ts=373 // [K]
11 Tw=357 // [K]
12 dT=Ts-Tw // [K]
13 Do=25 // [mm]
14 Do=Do/1000 // [m]
15 h=0.725*((rho^2*g*lambda*k^3)/(mu*Do*dT))^(1.0/4.0)
    // [W/sq m.K]
16 Q_by_l=h*pi*Do*dT // [W/m]
17 m_dot_byl=(Q_by_l/lambda) // [kg/s]
18 m_dot_byl=m_dot_byl*3600 // [kg/h]
19
20 printf("\nMean heat transfer coefficient is %f W/(sq
    m.K)\n",h);
21 printf("\nHeat transfer per unit length is %f W/m\n"
    ,Q_by_l);
22 printf("\nCondensate rate per unit length is %f kg/h
    ",m_dot_byl);

```

Scilab code Exa 3.48 Laminar Condensate film

```

1 clc;
2 clear;
3 //Example 3.48
4 rho=960 // [kg/m^3]
5 mu=2.82*10^-4 // [kg/(m.s)]
6 k=0.68 // [W/(m.K)]
7 lambda=2255 // [kJ/kg]
8 lambda=lambda*10^3 // [J/kg]
9 Ts=373 // Saturation temperature of steam [K]
10 Tw=371 // [K]
11 dT=Ts-Tw // [K]
12 L=0.3 // Dimension [m]
13 g=9.81 // [m/s ^2]
14 h=0.943*(rho^2*g*lambda*k^3/(L*mu*dT))^(1/4) //W/

```

```

    sq m.K
15 A=L^2    //[sq m]
16 Q=h*A*(Ts-Tw)    //[W]=[J/s]
17 m_dot=Q/lambda    //Condensate rate [kg/s]
18 m_dot=m_dot*3600    //[kg/h]
19 printf("\n Average heat transfer coefficient is %f W
    /(sq m.K)\n",h);
20 printf("\nHeat transfer rate is %f J/kg\n",Q);
21 printf("\n Steam condensate rate per hour is %f kg/h
    \n",m_dot);

```

Scilab code Exa 3.49 Saturated vapour condensate in array

```

1
2 clc;
3 clear;
4 //EXample 3.49
5 rho=1174    //[kg/m^3]
6 k=0.069    //[W/(m.K)]
7 mu=2.5*10^-4    //[N.s/m^2]
8 lambda=132*10^3    //[J/kg]
9 g=9.81    //[m/s^2]
10 Ts=323    //[K]
11 Tw=313    //[K]
12 dT=Ts-Tw    //[K]
13 //For square array,n=4
14 n=4 //number of tubes
15 Do=12    //[mm]
16 Do=Do/1000    //[m]
17 h=0.725*(rho^2*lambda*g*k^3/(n*Do*mu*dT))^(1/4) //W
    /(sq m.K)
18 //For heat transfer area calcaultion ,n=16
19 A=n*pi*Do    //[sq m]
20 A=0.603
21 Q=h*A*dT    //[W/m]

```



```

22 m_dot=Q/lambda // [kg/s]
23 m_dot=0.049 // Appriximation in book
24 m_dot=m_dot*3600 // [kg/h]
25 printf("\n Rate of condensation per unit length is
%f kg/h",m_dot);

```

Scilab code Exa 3.50 Mass rate of steam condensation

```

1
2 clc;
3 clear;
4 //Example 3.50
5 rho=960 // [kg/m^3]
6 k=0.68 // [W/m.K]
7 mu=282*10^-6 // [kg/(m.s)]
8 Tw=371 //Tube wall temperature [K]
9 Ts=373 //Saturation temperature in [K]
10 dT=Ts-Tw // [K]
11 lambda=2256.9 // [kJ/kg]
12 lambda=lambda*10^3 // [J/kg]
13 //Fora square array with 100tubes ,n=10
14 Do=0.0125 // [m]
15 g=9.81 // [m/s^2]
16 n=10
17 h=0.725*(((rho^2)*g*lambda*(k^3)/(mu*n*Do*dT))
^ (1.0/4.0)) //W/(sq m.K)
18
19 L=1 // [m]
20 //n=100
21 n=100;
22 A=n*%pi*Do*L // [m^2/m length]
23 Q=h*A*dT //Heat transfer rate in [W/m]
24 ms_dot=Q/lambda // [kg/s]
25 ms_dot=ms_dot*3600 // [kg/h]
26 printf("\n Mass rate of steam condensation is %d kg/

```

```

    h\n",round(ms_dot));
27
28 printf("\n NOTE:ERROR in Solution in book.Do is
    wrongly taken as 0.012 in lines 17 and 22 of the
    book,Also A is wrongly calculated\n")

```

Scilab code Exa 3.51 Saturated tube condensate in a wall

```

1
2 clc;
3 clear;
4 //Example 3.51
5 rho=975 // [kg/m^3]
6 k=0.871 // [W/m.K]
7 dT=10 // [K]
8 mu=380.5*10^-6 // [N.s/m^2]
9 lambda=2300 // [kJ/kg]
10 lambda=lambda*1000 // Latent heat of condensation [
    J/kg]
11 Do=100 //Outer diameter [mm]
12 Do=Do/1000 // [m]
13 g=9.81 // [m/s^2]
14 //for horizontal tube
15 h1=0.725*((rho^2*lambda*g*k^3)/(mu*Do*dT))^(1/4)
    //Average heat transfer coefficient
16 //for vertical tube
17 //h2=0.943*((rho^2*lambda*g*k^3)/(mu*L*dT))^(1/4)
    //Average heat transfer coefficient
18 h2=h1 //For vertical tube
19 //implies that
20 L=(0.943*((rho^2*lambda*g*k^3)^(1/4))/(h1*((mu*dT)
    ^((1/4))))^4 // [m]
21 L=0.29 //Approximate in book
22 h=0.943*((rho^2*lambda*g*k^3)/(mu*L*dT))^(1/4) // [W
    /(sq m.K)]

```

```

23 A=%pi*Do*L //Area in [m^2]
24 Q=h*A*dT //Heat transfer rate [W]
25 mc_dot=Q/lambda //[Rate of condensation]in [kg/s]
26 mc_dot=mc_dot*3600 //[kg/h]
27 printf("\n Tube length is %f m\n",L);
28 printf("\n Rate of condensation per hour is %f kg/h"
,mc_dot);

```

Scilab code Exa 3.52 Condensation rate

```

1 clc;
2 clear;
3 //Example 3.52
4 m1_dot=50 // For horizontal position [kg/h]
5 Do=10 // [mm]
6 Do=Do/1000 // [m]
7 L=1 // [m]
8 //For 100 tubes n=10
9 n=10;
10 //We know that
11 //m_dot=Q/lambda=h*A*dT/lambda
12 //m_dot is proportional to h
13 //m1_dot prop to h1
14 //m2_dot propn to h2
15 //m1_dot/m2_dot=h1/h2
16 //or :
17 m2_dot=m1_dot/((0.725/0.943)*(L/(n*Do))^(1/4)) // [
kg/h]
18 printf("\n For vertical position ,Rate of
condensationis %f kg/h",m2_dot);

```

Scilab code Exa 3.53 Condensation on vertical plate

```

1  clc;
2  clear;
3  rho=975 // [kg/m^3]
4  k=0.671 // [W/(m.K)]
5  mu=3.8*10^-4 // [N.s/m^2]
6  dT=10 // [K]
7  lambda=2300*10^3 // [J/kg]
8  L=1 // [m]
9  g=9.81 // [m/s^2]
10 h=0.943*((rho^2*lambda*g*k^3)/(mu*L*dT))^(1/4) //W
    // (sq m.K) // [W/sq m.K]
11
12 printf("\\n (i)- Average heat transfer coefficient is
    %d W/(m^2.K)\\n", round(h));
13
14 //Local heat transfer coefficient
15 //at x=0.5 // [m]
16 x=0.5 // [m]
17 h=((rho^2*lambda*g*k^3)/(4*mu*dT*x))^(1/4) // [W/sq
    m.K]
18 printf("\\n (ii)-Local heat transfer coefficient at
    0.5 m height is %d W/(sq m.K)\\n", round(h));
19 delta=((4*mu*dT*k*x)/(lambda*rho^2*g))^(1/4) // [m
    ]
20 delta=delta*10^3 // [mm]
21 printf("\\n (iii)-Film thickness is %f mm",delta);

```

Chapter 4

Radiation

Scilab code Exa 4.1 Heat loss by radiation

```
1 clc;
2 clear;
3 //Example 4.1
4 e=0.9 // [ Emissivity ]
5 sigma=5.67*10^-8 // [W/m^2.K^4]
6 T1=377 // [K]
7 T2=283 // [K]
8 Qr_by_a=e*sigma*(T1^4-T2^4) // [W/sq m]
9 printf("Heat loss by radiation is %d W/sq m", round(
    Qr_by_a));
```

Scilab code Exa 4.2 Radiation from unlagged steam pipe

```
1 clc;
2 clear;
3 //Example 4.2
4 e=0.9 // Emissivity
5 T1=393 // [K]
```

```

6 T2=293 // [K]
7 sigma=5.67*10^-8 // [W/sq m.K]
8 Qr_by_a=e*sigma*(T1^4-T2^4) //W/sq m
9 printf("\n Rate of heat transfer by radiation is %f
    W/sq m",Qr_by_a);

```

Scilab code Exa 4.3 Interchange of radiation energy

```

1
2 clc;
3 clear;
4 //Example 4.3
5 L=1; // [m]
6 e=0.8 ; // Emissivity
7 sigma=5.67*10^-8 ; // [m^2.K^4]
8 T1=423; // [K]
9 T2=300; // [K]
10 Do=60; // [mm]
11 Do=Do/1000; // [m]
12 A=%pi*Do*L // [sq m]
13 A=0.189 //Approx in book [m^2]
14 Qr=e*sigma*A*(T1^4-T2^4) // [W/m]
15 printf("\n Net radiaiton rate per 1 metre length of
    pipe is %d W/m",round(Qr));

```

Scilab code Exa 4.4 Heat loss in unlagged steam pipe

```

1 clc;
2 clear;
3 //Example 4.4
4 e=0.9 // Emissivity
5 L=1 // [m]
6 Do=50 // [mm]

```

```

7 Do=Do/1000 // [m]
8 sigma=5.67*10^-8 // [W/(m^2.K^4)]
9 T1=415 // [K]
10 T2=290 // [K]
11 dT=T1-T2 // [K]
12 hc=1.18*(dT/Do)^(0.25) // [W/sq m.K]
13 A=%pi*Do*L // Area in [sq m]
14 Qc=hc*A*dT // Heat loss by convection W/m
15 Qr=e*sigma*A*(T1^4-T2^4) // Heat loss by radiation
    per length W/m
16 Qt=Qc+Qr // Total heat loss in [W/m]
17 printf("\n Total heat loss by convection is %f W/m",
    Qt);

```

Scilab code Exa 4.5 Loss from horizontal pipe

```

1 clc;
2 clear;
3 //Example 4.5
4 e=0.85
5 sigma=5.67*10^-8 // [W/sq m.K]
6 T1=443 // [K]
7 T2=290 // [K]
8 dT=T1-T2 // [K]
9 hc=1.64*dT^0.25 //W/sq m.K
10 Do=60 // [mm]
11 Do=Do/1000 // [m]
12 L=6 //Length [m]
13 A=%pi*Do*L //Surface area of pipe in [sq m]
14 Qr=e*sigma*A*(T1^4-T2^4) // Rate of heat loss by
    radiaiton W
15 Qc=hc*A*(T1-T2) // Rate of heat loss by convection [
    W]
16 Qt=Qr+Qc //Total heat loss [W]
17 printf("\n Total heat loss is %d W", round(Qt))

```

Scilab code Exa 4.6 Heat loss by radiation in tube

```
1
2 clc;
3 clear;
4 //EXample 4.6
5 sigma=5.67*10^-8 // [W/m^2.K^4]
6 e1=0.79;
7 e2=0.93;
8 T1=500 ; // [K]
9 T2=300 ; // [K]
10 D=70 // [mm]
11 D=D/1000 // [m]
12 L=3 // [m]
13 W=0.3 //Side of conduit [m]
14 A1=%pi*D*L // [sq m]
15 A1=0.659 //Approximate calculation in book in
    [m^2]
16 A2=4*(L*W) // [sq m]
17 Q=sigma*A1*(T1^4-T2^4)/(1/e1+((A1/A2)*(1/e2-1)))
    // [W]
18 printf("\\n Heat lost by radiation is %f W",Q);
```

Scilab code Exa 4.7 Net radiant interchange

```
1 clc;
2 clear;
3 //Example 4.7
4 sigma=5.67*10^-8 // [W/sq m.K^4]
5 T1=703 // [K]
6 T2=513 // [K]
```



```

7 e1=0.85
8 e2=0.75
9 Q_by_Ar=sigma*(T1^4-T2^4)/(1/e1+1/e2-1) // [W/sq m]
10 printf("\n Net radiant interchange per square metre
      is %d W/sq m",round(Q_by_Ar));

```

Scilab code Exa 4.8 Radiant interchange between plates

```

1 clc;
2 clear;
3 //Example 4.8
4 L=3 ;// [m]
5 A=L^2 //Area in [sq m]
6 sigma=5.67*10^-8; // [W/sq m.K^4]
7 T1=373; // [K]
8 T2=313; // [K]
9 e1=0.736;
10 e2=e1;
11 F12=1/((1/e1)+(1/e2)-1)
12 Q=sigma*A*F12*(T1^4-T2^4) // [W]
13 printf("\n Net radiant interchange is %d W",round(Q)
      );

```

Scilab code Exa 4.9 Heat loss from thermflask

```

1 clc;
2 clear;
3 sigma=5.67*10^-8 // [W/sq m.K^4]
4 e1=0.05
5 e2=0.05
6 //A1=A2=1 (let)
7 A1=1;
8 A2=A1;

```

```

 9 F12=1/(1/e1+(A1/A2)*(1/e2-1))
10 T1=368 // [K]
11 T2=293 // [K]
12 Q_by_A=sigma*F12*(T1^4-T2^4) //Heat loss per unit
    Area [W/sq m]
13 printf("\nRate of heat loss when of silvered surface
    is %f W/sq m",Q_by_A);
14 //When both the surfaces are black
15 e1=1;
16 e2=1;
17 F12=1/(1/e1+(A1/A2)*(1/e2-1))
18 Q_by_A=sigma*F12*(T1^4-T2^4) // [W/sq m]
19 printf("\n When both surfaces are black ,Rate of heat
    loss is %d W/sq m",round(Q_by_A));

```

Scilab code Exa 4.10 Diwar flask

```

1 clc;
2 clear;
3 //Example 4.10
4 e1=0.05
5 e2=e1
6 A1=0.6944;
7 A2=1;
8 T1=293 // [K]
9 T2=90 // [K]
10 sigma=5.67*10^-8 // [W/m^2.K^4]
11 D=0.3 //Diameter in [m]
12
13 F12=1/(1/e1+(A1/A2)*(1/e2-1))
14 Q_by_A=sigma*F12*(T1^4-T2^4) // [W/sq m]
15 Q=Q_by_A*pi*(D^2) // [kJ/h]
16 Q=Q*3600/1000 // [kJ/h]
17 lambda=21.44 //Latent heat in [kJ/kg]
18 m_dot=Q/lambda //kg/h

```

```
19 printf("\n The liquid oxygen will evaporate at %f kg
    /h", m_dot);
```

Scilab code Exa 4.11 Heat flow due to radiation

```
1  clc;
2  clear;
3  //Example4.11
4  sigma=5.67*10^-8    //W/(m^2.K^4)
5  e1=0.3;
6  e2=e1;
7  D1=0.3    // [m]
8  D2=0.5    // [m]
9  T1=90    // [K]
10 T2=313    // [K]
11 A1=%pi*D1^2 //Area in [sq m]
12 A2=%pi*D2^2 //Area in [sq m]
13 Q1=sigma*A1*(T1^4-T2^4)/(1/e1+(A1/A2)*(1/e2-1)) // [
    W]
14 Q1=abs(Q1); //Absolute value in [W]
15 printf("\n Rate of heat flow due to radiation is %f
    W", Q1);
16 //When Aluminium is used
17 e1=0.05
18 e2=0.5
19 Q2=sigma*A1*(T1^4-T2^4)/(1/e1+(A1/A2)*(1/0.3-1)) //
    [W]
20 Q2=abs(Q2) //Absolute value in [W]
21 Red=(Q1-Q2)*100/Q1 //Percent reduction
22 printf("\n Reduction in heat flow will be %f percent
    ", Red);
```

Scilab code Exa 4.12 Heat exchange between concentric shell

```

1
2 clc;
3 clear;
4 //Example 4.12
5 sigma=5.67*10^-8 // [W/sq m.K^4]
6 T1=77 // [K]
7 T2=303 // [K]
8 D1=32 //cm
9 D1=D1/100 // [m]
10 D2=36 // [cm]
11 D2=D2/100 // [m]
12 A1=%pi*D1^2 // [sq m]
13 A2=%pi*D2^2 // [sq m]
14 e1=0.03;
15 e2=e1;
16 Q=sigma*A1*(T1^4-T2^4)/(1/e1+(A1/A2)*(1/e2-1)) // [W
]
17 Q=Q*3600/1000 // [kJ/h]
18 Q=abs(Q); // [kJ/h]
19 lambda=201 //kJ/kg
20 m_dot=Q/lambda //Evaporation rate in [kg/h]
21 printf("\n Nitrogen evaporates at %f kg/h",m_dot);

```

Scilab code Exa 4.13 Evaporation in concentric vessels

```

1
2 clc;
3 clear;
4 //Example 4.13
5 D1=250 //Inner sphere idiameter [mm]
6 D1=D1/1000 //Outer diameter [m]
7 D2=350 // [mm]
8 D2=D2/1000 // [m]
9 sigma=5.67*10^-8 //W/(sq m.K^4)
10 A1=%pi*D1^2 // [sq m]

```

```

11 A2=%pi*D2^2 // [sq m]
12 T1=76 // [K]
13 T2=300 // [K]
14 e1=0.04;
15 e2=e1;
16 Q=sigma*A1*(T1^4-T2^4)/((1/e1)+(A1/A2)*((1/e2)-1))
    // [W]
17 Q=-2.45 // Approximate
18 Q=abs(Q) // [W]
19 Q=Q*3600/1000 // [kJ/h]
20 lambda=200 //kJ/kg
21 Rate=Q/lambda // [kg/h]
22 printf("\n Rate of evaporation is %f kg/h(approx)",
    Rate);

```

Scilab code Exa 4.15 infinitely long plates

```

1 clc;
2 clear;
3 //Example 4.15
4 sigma=5.67*10^-8 // [W/(m^2.K^4)]
5 e1=0.4
6 e3=0.2
7 T1=473 // [K]
8 T3=303 // [K]
9 Q_by_a=sigma*(T1^4-T3^4)/((1/e1)+(1/e3)-1) // [W/sq
    m]
10 //Q1_by_a=sigma*(T1^4-T2^4)/((1/e1)+(1/e2)-1)=sigma*
    A*(T2^4-T3^4)/((1/e2)+(1/e3)-1) // [W/sq m]
11 e2=0.5
12 //Solving we get
13 T2=((6/9.5)*((3.5/6)*T3^4+T1^4))^(1/4) // [K]
14 Q1_by_a=sigma*(T1^4-T2^4)/((1/e1)+(1/e2)-1) // [W/sq
    m]
15 red=(Q_by_a-Q1_by_a)*100/Q_by_a

```

```

16 printf("\nHeat transfer rate per unit area(WITHOUT
    SHIELD) due to radiation is %f W/sq m\n",Q_by_a);
17 printf("\nHeat transfer rate per unit area(WITH
    SHIELD) due to radiation is %f W/sq m\n",Q1_by_a)
    ;
18 printf("\nReduction in heat loss is %f percent",red)
    ;

```

Scilab code Exa 4.16 Heat exchange between parallel plates

```

1  clc;
2  clear;
3  //Example 4.16
4  //In steady state ,we can write:
5  //Qcd=Qdb
6  //sigma(Tc^4-Td^4)/(1/ec+1/ed-1)=sigma(Td^4-Tb^4)
    /(1/ed+1/eb-1)
7  // i.e Td^4=0.5*(Tc^4-Tb^4)
8  //Given:
9  Ta=600 // [K]
10 eA=0.8;
11 eC=0.5;
12 eD=0.4;
13 sigma=5.67*10^-8 //For air
14 //(600^4-Tc^4)/2.25=(Tc^4-Td^4)/3.5
15 //1.56*(600^4-Tc^4)=Tc^4-Td^4
16 //Putting value of Td in terms of Tc
17 //1.56*(600^4-Tc^4)=Tc^4-0.5*(Tc^4-300^4)
18 function y=f(Tc)
19     y=1.56*(600^4-Tc^4)-Tc^4+0.5*(Tc^4-300^4)
20 endfunction
21 Tc=fsolve(500,f); // [K]
22 //or
23 Tc=560.94 // [K] Approximate after solving
24 Td=sqrt(sqrt(0.5*(Tc^4-300^4))) // [K]

```

```

25 Q_by_a=sigma*(Ta^4-Tc^4)/(1/eA+1/eC-1)           // [W/sq
    m]
26 printf("\nRate of heat exchange per unit area=%f W/m
    ^2",Q_by_a);
27 printf("\nSteady state temperatures ,Tc=%f K,and Td=
    %f K",Tc,Td);

```

Scilab code Exa 4.17 Thermal radiation in pipe

```

1  clc;
2  clear;
3  //Example 4.17
4  sigma=5.67*10^-8           // [W/(sq m.K^4)]
5  e=0.8
6  T1=673; // [K]
7  T2=303; // [K]
8  Do=200 // [mm]
9  Do=Do/1000 // [m]
10 L=1 //Let [m]
11 A1=%pi*Do*L // [m^2/m]
12 //Case 1: Pipe to surroundings
13
14 Q1=e*A1*sigma*(T1^4-T2^4) // [W/m]
15 Q1=5600 //Approximated
16 //Q1=5600 // [W/m] approximated in book for
    calculation purpose
17 //Concentric cylinders
18 e1=0.8;
19 e2=0.91;
20 D1=0.2 // [m]
21 D2=0.4 // [m]
22 Q2=sigma*0.628*(T1^4-T2^4)/((1/e1)+(D1/D2)*((1/e2)
    -1)) // [W/m] length
23 Red=Q1-Q2 //Reduction in heat loss
24

```

```

25 printf("\nDue to thermal radiaiton ,Loss of heat to
    surrounding is %d W/m\n",round(Q1));
26 printf("\nWhen pipe is enclosed in 1 400 mm diameter
    brick conduit ,Loss of heat is %d W/m\n",round(Q2
    ));
27 printf("\n Reduction in heat loss is %d W/m\n",round
    (Red));

```

Scilab code Exa 4.18 Heat transfer in concentric tube

```

1
2
3 clc;
4 clear;
5 //Example 4.18
6
7
8 sigma=5.67*10^-8 ; // [W/(sq m.K^4)]
9 T1=813; // [K]
10 T2=473; // [K]
11 e1=0.87;
12 e2=0.26;
13 D1=0.25 ;//[m]
14 D2=0.3; //[m]
15 Q_by_a1=sigma*(T1^4-T2^4)/(1/e1+(D1/D2)*(1/e2-1))
    // [W/ sqm]
16 printf("\n Heat transfer by radiaiton is %d W/sq m",
    Q_by_a1);

```

Scilab code Exa 4.19 Heat exchange between black plates

```

1
2 clc;

```



```

3 clear;
4 //Example 4.19
5 sigma=5.67*10^-8 // [W/sq m.K^4]
6 A1=0.5*1 // [sq m]
7 F12=0.285
8 T1=1273 // [K]
9 T2=773 // [K]
10 Q=sigma*A1*F12*(T1^4-T2^4) // [W]
11 printf("\n Net radiant heat exchange between plates
    is %d W",Q);

```

Scilab code Exa 4.20 Radiation shield

```

1 clc;
2 clear;
3 //Example 4.20
4 sigma=5.67*10^-8 // [W/sq m.K^4]
5 T1=750 // [K]
6 T2=500 // [K]
7 e1=0.75;
8 e2=0.5;
9 //Heat transfer without shield :
10
11 Q_by_a=sigma*(T1^4-T2^4)/((1/e1)+(1/e2)-1) // [W/sq
    m]
12
13 //Heat transfer with shield:
14 R1=(1-e1)/e1 //Resistance 1
15
16 F13=1;
17 R2=1/F13 //Resistance 2
18
19 e3=0.05
20 R3=(1-e3)/e3 //Resistance 3
21

```

```

22 R4=(1-e3)/e3      //Resistance 4
23
24 F32=1;
25 R5=1/F32          //Resistance 5
26
27 R6=(1-e2)/e2      //Resistance 6
28
29 Total_R=R1+R2+R3+R4+R5+R6 //Total resistance
30
31 Q_by_as=sigma*(T1^4-T2^4)/Total_R // [W/sq m]
32
33 Red=(Q_by_a-Q_by_as)*100/Q_by_a //Reducton in
    heat tranfer due to shield
34
35 printf("\n Reduction in heat transfer rate as a
    result of radiaiotn shield is %f percent",Red);

```

Scilab code Exa 4.21 Heat transfer with radiaiton shield

```

1  clc;
2  clear;
3  //Example 4.21
4  e1=0.3
5  e2=0.8
6  //Let sigma*(T1^4-T2^4)=z=1(const)
7  z=1; //Let
8  Q_by_A=z/(1/e1+1/e2-1) //W/sq m
9
10 //Heat transfer with radiation shield
11 e3=0.04
12 F13=1;
13 F32=1;
14 //The resistances are:
15 R1=(1-e1)/e1
16 R2=1/F13

```

```

17 R3=(1-e3)/e3
18 R4=R3
19 R5=1/F32
20 R6=(1-e2)/e2
21 R=R1+R2+R3+R4+R5+R6 //Total resistance
22 Q_by_As=z/R //where z=sigma*(T1^4-T2^4) //W/sq m
23 red=(Q_by_A-Q_by_As)*100/Q_by_A //Percent
    reduction in heat transfer
24 printf("\n The heat transfer is reduced by %f
    percent due to shield",red)

```

Scilab code Exa 4.22 Radiation shape factor

```

1
2 clc;
3 clear;
4 //Example 4.22
5 sigma=5.67*10^-8;
6 T1=1273 // [K]
7 T2=773 // [K]
8 T3=300 // [K]
9 A1=0.5 // [sq m]
10 A2=A1; // [sq m]
11 F12=0.285;
12 F21=F12;
13 F13=1-F12;
14 F23=1-F21;
15 e1=0.2;
16 e2=0.5;
17 //Resistance in the network are calculated as:
18 R1=1-e1/(e1*A1)
19 R2=1-e2/(e2*A2)
20 R3=1/(A1*F12)
21 R4=1/(A1*F13)
22 R5=1/(A2*F23)

```

```

23 R6=0      // Given (1-e3)/e3*A3=0
24 // Also
25 Eb1=sigma*T1^4 // [W/sq m]
26 Eb2=sigma*T2^4 // [W/sq m]
27 Eb3=sigma*T3^4 // [W/sq m]
28
29 // Equations are:
30 // (Eb1-J1)/2+(J2-J1)/7.018+(Eb3-J1)/2.797=0
31 // (J1-J2)/7.018+(Eb3-J2)/2.797+(Eb2-J2)/2=0
32
33 // On solving we get:
34 J1=33515 // [W/sq m]
35 J2=15048 // [W/sqm]
36 J3=Eb3 // [W/sq m]
37 Q1=(Eb1-J1)/((1-e1)/(e1*A1)) // [W/sq m]
38 Q2=(Eb2-J2)/((1-e2)/(e2*A2)) // [W/sq m]
39 Q3=(J1-J3)/(1/(A1*F13))+(J2-J3)/(1/(A2*F23)) // [W
    /sq m]
40 printf("\n Total heat lost by plate 1 is %f W/sq m\n
    ",Q1);
41 printf("\n Total heat lost by plate 2 is %f W/sq m\n
    ",Q2);
42 printf("\nThe net energy lost by both plates must be
    absorbed by the room,\n %f=%f",Q3,Q1+Q2)

```

Scilab code Exa 4.23 Radiation loss in plates

```

1  clc;
2  clear;
3  // Example 4.23
4  sigma=5.67*10^-8 // [W/sq m.K^4]
5  e1=0.7;
6  e2=0.7;
7  T1=866.5 // [K]
8  T2=588.8 // [K]

```

```

9  Q_by_A=sigma*(T1^4-T2^4)/((1/e1)+(1/e2)-1)    // [W/sq
    m]
10 e1=0.7;
11 e2=e1;
12 e3=e1;
13 e4=e1;
14 e=e1;
15 //Q with n shells =1/(n+1)
16 n=2
17 Q_shield=1/(n+1);
18 es1=e1;
19 es2=e1;
20 Q_by_A=sigma*(T1^4-T2^4)/((1/e1)+(1/e2)+2*(1/es1+1/
    es2)-(n+1))    // [W/sq m]
21 printf("\n New Radiaiton loss is %f W/sq m",Q_by_A);

```

Scilab code Exa 4.24 Concentric tube

```

1  clc;
2  clear;
3  //Example 4.24
4  //1.WITHOUT SHIELD
5  sigma=5.67*10^-8
6  e1=0.12;
7  e2=0.15;
8  T1=100    // [K]
9  T2=300    // [K]
10 r1=0.015    // [m]
11 r2=0.045    // [m]
12 L=1    // [m]
13 A1=2*pi*r1*L    // [sq m]
14 Q_by_L=2*pi*r1*sigma*(T1^4-T2^4)/((1/e1+(r1/r2)*(1/
    e2-1))    // [W/m]
15 // -ve saign indicates that the net heat flow is in
    the radial inward direction

```

```

16 // 2.WITH CYLINDRICAL RADIATION SHIELD
17 e3=0.10;
18 e4=0.05;
19 r3=0.0225 // [m]
20 Qs_by_L=2*%pi*r1*sigma*(T1^4-T2^4)/((1/e1+r1/r2*(1/e2
    -1)+(r1/r3)*(1/e3+1/e4-1)) // [W/sq m]
21 red=(abs(Q_by_L)-abs(Qs_by_L))*100/abs(Q_by_L) //
    percent reduction in heat gain
22
23 // Radiation network approach
24 A3=2*%pi*r3 // [sq m]
25 A2=2*%pi*r2 // [sq m]
26 F13=1;
27 F32=1;
28 R1=(1-e1)/(e1*A1)
29 R2=1/(A1*F13)
30 R3=(1-e3)/(e3*A3)
31 R4=(1-e4)/(e4*A3)
32 R5=1/(A3*F32)
33 R6=(1-e2)/(e2*A2)
34
35 Qs=sigma*(T1^4-T2^4)/(((1-e1)/(e1*A1)+1/(A1*F13)+(1-
    e3)/(e3*A3)+(1-e4)/(e4*A3)+1/(A3*F32)+(1-e2)/(e2*
    A2))
36 printf("\n With cylindrical radiaiton shield Heat
    gained by fluid per 1 m lengh of tube is %f W/m\n
    ",Qs_by_L);
37 printf("\nPercent reduction in heat gain is %f
    percent\n",red);
38 printf("\nWith radiaiton network approach %f W/sqm "
    ,Qs);

```

Chapter 5

Heat Exchangers

Scilab code Exa 5.1 Harpin exchanger

```
1
2
3 clc;
4 clear;
5 //Example 5.1
6 Di=35 // [mm]
7 Di=Di/1000 // [m]
8 Do=42 // [mm]
9 Do=Do/1000 // [m]
10 //for benzene
11 mb_dot=4450 // [kg/h]
12 Cpb=1.779 // [kJ/(kg.K)]
13 t2=322 // [K]
14 t1=300 // [K]
15 Q=mb_dot*Cpb*(t2-t1) //for benzene in [kJ/h]
16
17 //For toluene
18 T1=344 // [K]
19 T2=311 // [K]
20 Cpt=1.842 // [kJ/kg.K]
21 mt_dot=Q/(Cpt*(T1-T2)) // [kg/h]
```

```

22 Q=Q*1000/3600           // [W]
23 //Hot fluid (toluene)
24 //Cold fluid (benzene)
25 dT1=22                 // [K]
26 dT2=11                 // [K]
27 dTlm=(dT1-dT2)/(log(dT1/dT2)) // [K]
28
29 //Clod fluid:Inner pipe , benzene
30 Di=0.035               // [m]
31 Ai=(%pi/4)*Di^2        //Flow area [sq m]
32 Gi=mb_dot/Ai           //Mass velocity [kg/m^2.h]
33 Gi=Gi/3600             // [kg/m^2.s]
34 mu=4.09*10^-4          // [kg/(m.s)]
35 Nre=Di*Gi/mu           //Reynolds number
36
37 Cp=Cpb*10^3            // [J/(kg.K)]
38 k=0.147                // [W/m.K]
39 Npr=Cp*mu/k            //Prandtl number
40 hi=(k/Di)*0.023*(Nre^0.8)*(Npr^0.4) // [W/sq m.K]
41 hio=hi*Di/Do           // [W/sq m.K]
42 D1=0.042               //Outside dia of inside pipe
   [mm]
43 D2=0.0525              //Inside dia of outside pipe [m]
44 De=(D2^2-D1^2)/D1     // [m]
45 De=0.0236              //Approximated
46 aa=%pi*(D2^2-D1^2)/4  //Flow area [sq m]
47 Ga=mt_dot/aa          //Mass velocity in [kg/m^2.h]
48 Ga=Ga/3600            // [kg/m^2.s]
49 mu=5.01*10^-4         // [kg/(m.s)]
50 Nre=De*Ga/mu          //Reynolds number
51 Npr=Cp*mu/k           //Prandtl number
52 ho=(k/De)*0.023*(Nre^0.8)*(Npr^0.3) // [W/sq m.K]
53 Uc=1/(1/ho+1/hio)     // [W/sq m.K]
54 Rdi=1.6*10^-4         //Fouling factor [m^2.K/W]
55 Rdo=1.6*10^-4         //Fouling factor [m^2.K/W]
56 Rd=Rdi+Rdo            // (m^2.K/W)
57 Ud=1/(1/Uc+Rd)        // [W/sq m.K]
58 A=Q/(Ud*dTlm)         //sq m

```



```

59 ex=0.136 // [sq m]
60 l=A/ex //m
61 t1=12 //Total length of one harpin of 6m [m]
62 printf("b%f",l);
63 printf("\n\Required surface is fulfilled by
connecting %d(three) 6m harpins in series\n",
round(1/t1))

```

Scilab code Exa 5.2 Length of pipe

```

1 clc;
2 clear;
3 //Example 5.2
4 ma_dot=300*1000/24 //Mass flow rate of acid
in [kg/h]
5 mw_dot=500*1000/24 //Mass flow rate of
water in [kg/h]
6 Cp1=1.465 // [kJ/kg.K]
7 T1=333 // [K]
8 T2=313 // [K]
9 Q=ma_dot*Cp1*(T1-T2) // [kJ/h]
10 Q=Q*1000/3600 // [W]
11 Cp2=4.187 // [kJ/kg.K]
12 t1=288 // [K]
13 t2=(Q/(mw_dot*Cp2))+t1 // [K]
14 dT1=T1-t2 // [K]
15 dT2=T2-t1 // [K]
16 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
17 dTlm=32.26 //Approximation in [K]
18 //Inner pipe
19 m_dot=12500 // [kg/h]
20 Di=0.075 // [m]
21 Ai=(%pi/4)*Di^2 // [sq m]
22 G=ma_dot/Ai // [kg/m^2.h]
23 G=G/3600 // [kg/m^2.s]

```

```

24 mu=0.0112 // [kg/m.s]
25 k=0.302 //W/(m.K)
26 Nre=Di*G/mu //Reynold number
27 Npr=Cp1*10^3*mu/k //Prandtl number
28 hi=(k/Di)*0.023*(Nre^0.8)*(Npr^0.3) //W/sq m.K
29 Do=0.1 // [m]
30 hio=hi*Di/Do //W/sq m.K
31 D1=0.1 // [m]
32 D2=0.125 // [m]
33 De=(D2^2-D1^2)/D1 // [m]
34 Aa=(%pi/4)*(D2^2-D1^2) // [sq m]
35 Ga=mw_dot/Aa // [kg/m^2.h]
36 Ga=Ga/3600 // [kg/sq m.s]
37 mu=0.0011 // [kg/m.s]
38 Nre=De*Ga/mu // Reynolds number
39 k=0.669 //for water
40 Npr=Cp2*10^3*mu/k //Prandtl number
41 ho=(k/De)*0.023*(Nre^0.8)*Npr^0.4 // [W/sq m.K]
42 xw=(Do-Di)/2 // [m]
43 Dw=(Do-Di)/log(Do/Di) // [m]
44 kw=46.52 //thermal conductivity
    of wall in [W/m.K]
45 Uc=1/(1/ho+1/hio+xw*Do/(kw*Dw)) // [W/sq m.K]
46 Ud=Uc //As dirt factor values
    are not given
47 Ud=195.32 //Approximation
48 A=Q/(Ud*dTlm) // [sq m]
49
50 L=A/(%pi*Do) // [sq m]
51 printf("\nArea =%f m^2,\nLength fo pipe required =%f
    m(approx)",A,L)

```

Scilab code Exa 5.3 Double pipe heat exchanger

```
1 clc;
```

```

2 clear;
3 //Example 5.3
4 me_dot=5500 ; // [kg/h]
5 me_dot1=me_dot/3600 // [kg/s]
6 Di=0.037 ; //I.D of inner pipe in [m]
7 Ai=(%pi/4)*Di^2 // [sq m]
8 G=me_dot1/Ai // [kg/sq m.s]
9 mu=3.4*10^-3 ; // [Pa.s] or [kg/(m.s)]
10 Nre=Di*G/mu // Reynolds number
11 Cp=2.68 ; // [kJ/kg.K]
12 Cp1=Cp*10^3 // [J/kgK]
13 k=0.248 ; // [W/m.K]
14 Npr=Cp1*mu/k // Prandtl number
15 //Nre is greater than 10,000,Use Dittus-Boelter eqn:
16 Nnu=0.023*(Nre^0.8)*(Npr^0.3) // Nusselt number
17 hi=k*Nnu/Di // [W/sq m.K]
18 T2=358 // [K]
19 T1=341 // [K]
20 Cp2=1.80 // [kJ/kg.K]
21 t2=335 // [K]
22 t1=303 // [K]
23 mt_dot=me_dot*Cp*(T2-T1)/(Cp2*(t2-t1)) // [kg/h]
24 mt_dot=mt_dot/3600 // [kg/s]
25 D1=0.043 // [m]
26 D2=0.064 // Inside dia of outer pipe
27 De=(D2^2-D1^2)/D1 //Equivalent diameter [m]
28 ]
29 Aa=%pi/4*(D2^2-D1^2) // [sq m]
30 Ga=mt_dot/Aa //kg/(sq m.s)
31 mu2=4.4*10^-4 // Viscosity of
    toluene Pa.s
32 k2=0.146 //For toluene [W/m.K]
33 Cp2=1.8*10^3 //J/kg.K
34 Nre=De*Ga/mu2 // Reynolds number
35 Npr=Cp2*mu2/k2 // Prandtl number
36 Nnu=0.023*Nre^0.8*Npr^0.4 // Nusselt number
37 ho=k2*Nnu/De //W/(sq m.K)
38 Dw=(D1-Di)/log(D1/Di) // [m]

```

```

38 x=0.003 //Wall thickness in [m]
39 Uo=1/(1/ho+(1/hi)*(D1/Di)+(x*D1/(46.52*Dw))) // [
    W/sq m.K]
40 dT1=38 // [K]
41 dT2=23 // [K]
42 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
43 Q=me_dot1*Cp*(T2-T1) // [kJ/s]
44 Q=Q*1000 // [J/s]
45 L=Q/(Uo*pi*D1*dTlm) // [m]
46 printf("\nTotal lenggth of double pipe heat
    exchanger is %f m",L)

```

Scilab code Exa 5.4 Parallel flow arrangement

```

1 clc;
2 clear;
3 //Example 5.4
4 mc_dot=1000 // [kg/h]
5 mc_dot=mc_dot/3600 // [kg/s]
6 mh_dot=250 // [kg/h]
7 mh_dot=mh_dot/3600 // [kg/s]
8 Cpc=4187 // [J/(kg.K)]
9 Cph=3350 // [W/K]
10 w=mc_dot*Cpc // [W/K]
11 l=mh_dot*Cph // [W/K]
12 C=mh_dot*Cph/(mc_dot*Cpc)
13 U=1160 // [W/sq m.K]
14 A=0.25 //Heat transfer surface for exchanger in
    [sq m]
15 ntu=U*A/(mh_dot*Cph) //
16 E=(1-%e^(-ntu*(1+C)))/(1+C) // Effectiveness of
    heat exchanger
17 T1=393 // Inlet temperature in [K]
18 t1=283 // Cooling water [K]
19 T2=T1-E*(T1-t1) // Outlet T of hot liquid

```

```

20
21 t2=C*(T1-T2)+t1 // [K]
22 printf("\n\nEffectiveness of heat exchanger is %f\n",
    ,E);
23 printf("\nOutlet temperature of hot liquid is %f\n",
    T2);
24 printf("\nOutlet temperature of water is %f\n",t2)

```

Scilab code Exa 5.5 Counter flow exchanger

```

1
2 clc;
3 clear;
4 //Example 5.5
5 Cpc=4187 // Specific heat of water in
    [J/(kg.K)]
6 Cph=2000 //Sp heat of oil in [J/(kg.K
    )]
7 mc_dot=1300/3600 // [kg/s]
8 mh_dot=550/3600 // [kg/s]
9 w=mc_dot*Cpc // [W/K]
10 o=mh_dot*Cph // [W/K]
11 //Heat capacity of rate of hot fluid is smaller than
    water
12 U=1075 // [W/sq m.K]
13 A=1 // [sq m]
14 ntu=(U*A)/(mh_dot*Cph)
15 C=mh_dot*Cph/(mc_dot*Cpc)
16 E=(1-%e^(-ntu*(1-C)))/(1-C*%e^(-ntu*(1-C))) //
    Efficiency
17 T1=367 // [K]
18 t1=288 // [K]
19 T2=T1-E*(T1-t1) //Outlet temperature
    [K]
20 T2=291.83 //Approximated in book

```

```

    without precise calculation
21 t2=C*(T1-T2)+t1 // [K]
22 Q=mh_dot*Cph*(T1-T2) // [W]
23 printf("\n\nEffectiveness of exchanger is %f\n",E);
24 printf("\nOutlet temperature of oil is %f K\n",T2);
25 printf("\nOutlet temperature of water is %f K\n",t2)
    ;
26 printf("\nRate of heat transfer is %f W",Q);

```

Scilab code Exa 5.6 LMTD approach

```

1  clc;
2  clear;
3  //Example 5.6
4  printf("\nLMTD Approach\n")
5
6  Cph=4187 // [J/(kg.K)]
7  mh_dot=600/3600 //Hot side flow rate [kg/s]
8  mc_dot=1500/3600 // [kg/s]
9  Cpc=Cph // [J/kg.K]
10 T1=343 // [K]
11 T2=323 // [K]
12 Q=mh_dot*Cph*(T1-T2) // [W]
13 t1=298 // [K]
14 t2=(mh_dot*Cph*(T1-T2))/(mc_dot*Cpc)+t1 // [K]
15 dT1=45 // [K]
16 dT2=17 // [K]
17 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
18 hi=1600 //Heat transfer coeff in [W/sq m.K]
19 ho=hi // [W/sq m.K]
20 U=1/(1/hi+1/ho) // [W/sq m.K]
21 A=Q/(U*dTlm) // [sq m]
22
23 printf("\nEffectiveness -NTU approach\n");
24

```

```

25 //hot water:
26 h=mh_dot*Cph // [W/K]
27 c=mc_dot*Cpc // [W/K]
28 //Heat capacity rate of hot fluid is small
29 C=mh_dot*Cph/(mc_dot*Cpc) //
30 E=(T1-T2)/(T1-t1) // Effectiveness
31 //for paralell flow:
32 ntu=-log(1-E*(1+C))/(1+C)
33 A2=(ntu*mh_dot*Cph)/U // [sq m]
34 t2=C*(T1-T2)+t1 // [K]
35 printf("\n By LMTD approach area of heat exchanger
    is %f sq m\n",A);
36 printf("\nBy Ntu approach Area of heat exchanger is
    %f sq m\n",A);
37 printf("\n Outlet temperature of cold water=%f K\n",
    t2)

```

Scilab code Exa 5.7 Shell and tube exchanger

```

1 clc;
2 clear;
3 //Example 5.7
4 mw_dot=10 // [kg/s]
5 Cpw=4.187 // [kJ/(kg.K)]
6 t2=318 // [K]
7 t1=295 // [K]
8 Q=mw_dot*Cpw*(t2-t1) // [kJ/s]
9 Q=Q*1000 //W
10 dT1=98 // [K]
11 dT2=75 // [K]
12 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
13 hi=850 // [W/sq m.K]
14 id=0.027 // Inside dia [m]
15 od=0.031 // Outside dia [m]
16 hio=hi*id/od // [W/sq m.K]

```

```

17 ho=6000 //Heat transfer coefficients [W/sq m.K]
18 Uo=1/(1/ho+1/hio) // [W/sq m.K]
19 Ao=Q/(Uo*dTlm) // [sq m]
20 L=4 //Length [m]
21 n=Ao/(%pi*od*L) // [No. of tubes]
22 printf("\n Number of tubes required = %d\n",round(n)
);

```

Scilab code Exa 5.8 Order of Scale resistance

```

1 clc;
2 clear;
3 //Example 5.8
4 mdot=7250; //Nitrobenzene in shell in [kg/
    h]
5 Cp=2.387; // [kJ/(kg.K)]
6 mu=7*10^-4 ; //Pa.s
7 k=0.151; // [W/m.K]
8 T1=400; // [K]
9 T2=317; // [K]
10 t1=305; // [K]
11 t2=345; // [K]
12 dT1=T1-t2 // [K]
13 dT2=T2-t1 // [K]
14 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
15 Q=mdot*Cp*(T1-T2) // [kJ/h]
16 Q=Q*1000/3600 // [W]
17 n=166; //no of tubes
18 L=5; // [m]
19 Do=0.019; // [m]
20 Di=0.015 // [m]
21 Ao=n*%pi*Do*L // [sq m]
22 Uo=Q/(Ao*dTlm) // [W/sq m.K]
23 Ud=Uo
24 //Shell side heat transfer coefficient

```



```

25 Pt=0.025 // [m]
26 C_dash=Pt-(0.5*Do+0.5*Do)
27
28 //Shell side crossflow area
29 B=0.15 // [m]
30 id=0.45 // [m]
31 as=id*C_dash*B/Pt // [sq m]
32 //As there are two shell passes, area per pass is :
33 as_dash=as/2 // [sq m]
34
35 //Equivalent diameter of shell
36 De=4*(Pt^2-(%pi/4)*Do^2)/(%pi*Do) // [m]
37
38 //Mass velocity on shell side
39 Gs=mdot/as_dash // [kg/m^2.h]
40 Gs=Gs/3600 // [kg/m^2.s]
41 mu=7*10^-4 //Pa.s
42 Cp=Cp*1000 //J/kg.K
43 Nre=De*Gs/mu //Reynold number
44 Npr=Cp*mu/k //Prandtl's number
45 Nnu=0.36*Nre^0.55*Npr^(1.0/3.0) //Nusselts number
46 hi=1050 // [W/sq m .K]
47 ho=Nnu*k/De // [W/sq m.K]
48 Uo=1/(1/ho+(1/hi*(Do/Di))) // [W/sq m K]
49 Uc=Uo
50 Rd=(Uc-Ud)/(Uc*Ud) //m^2.K/W
51 printf("\n Fouling factor=Sclae resistance=%f m^2.K/
W\n", Rd);

```

Scilab code Exa 5.9 Length of tube required

```

1 clc;
2 clear;
3 //Example 5.9
4 k=0.628 //W/(m.K)

```

```

5 rho=980          // [kg/m^3]
6 mu=6*10^-4      // kg/(m.s)
7 Cpw=4.187       //kJ/(kg.K)
8 Cp=Cpw*10^3     //J/(kg.K)
9 Di=25           // [mm]
10 Di=Di/1000     // [m]
11 mw_dot=1200*10^-3*rho //Mass flow rate of
    water [kg/h]
12 mw_dot=mw_dot/3600 // [kg/s]
13 Ai=(%pi*Di^2)/4 //Inside area of tube
    in sq m
14 G=mw_dot/Ai     //kg/m^2.s
15 Nre=Di*G/mu     //Reynolds number
16 Npr=Cp*mu/k     //Prandtl number
17 //Inside heat transfer coefficient
18 Nnu=0.023*Nre^0.8*Npr^0.4 //Nusselt number
19 hi=Nnu*k/Di     // [W/sq m.K]
20 ho=6000        // [W//sq m.K]
21 Do=0.028       // [m]
22 Dw=(Do-Di)/log(Do/Di) // [m]
23 x=(Do-Di)/2    // [m]
24 k2=348.9       //thermal conductivity of metal
    in [W/m.K]
25 Uo=1/((1/ho)+(1/hi)*(Do/Di)+(x/k2)*(Do/Dw)) // [W/sq
    m.K]
26 t1=303         // [K]
27 t2=343         // [K]
28 Q=mw_dot*Cpw*(t2-t1) // [kJ/h]
29 Q=Q*1000       // [W]
30 Ts=393         // [K]
31
32 dT1=Ts-t1      // [K]
33 dT2=Ts-t2      // [K]
34 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
35 Ao=Q/(Uo*dTlm) // [sq m]
36 L=Ao/(%pi*Do) //Length
37 printf("\n therefore length of tube required is %f m
    \n",L);

```

Scilab code Exa 5.10 Suitability of Exchanger

```
1
2 clc;
3 clear;
4 //Example 5.10
5 m_dot=7250           // [kg/h] of nitrobenzene
6 Cp=2.387;           // [kJ/kg.K]
7 mu=7*10^-4;         // [kg/m.s]
8 k=0.151;            // [W/m.K]
9 vis=1;
10 Ft=0.9;             //LMTD correction factor
11 T1=400              // [K]
12 T2=317              // [K]
13 t1=333              // [K]
14 t2=300              // [K]
15 dT1=T1-t1           // [K]
16 dT2=T2-t2           // [K]
17 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
18 //For nitrobenzene
19 Q=m_dot*Cp*(T1-T2) // [kJ/h]
20 Q=Q*1000/3600       // [W]
21 n=170               //No. of tubes
22 L=5                 // [m]
23 Do=0.019            // [m]
24 Di=0.015            // [m]
25 Ao=n*pi*Do*L        // [sq m]
26 Uo=Q/(Ao*Ft*dTlm) // [W/sq m.K]
27 Ud=Uo               // [W/sq m.K]
28 B=0.15              // Baffle spacing [m]
29 Pt=0.025            // Tube pitch in [m]
30 C_dash=Pt-Do        // Clearance in [m]
31 id=0.45             // [m]
32
```

```

33 //Shell side cross flow area
34 as=id*C_dash*B/Pt // [sq m]
35
36 //Equivalent diameter of shell
37 De=4*(Pt^2-(%pi/4)*(Do^2))/(%pi*Do) // [m]
38
39 //Mass velocity on shell side
40 Gs=m_dot/as // [kg/(m.h)]
41 Gs=Gs/3600 // [kg/m^2.s]
42 mu=7*10^-4 // [kg/m.s]
43 Cp=Cp*1000 // [J/kg.K]
44 Nre=De*Gs/mu // Reynolds number
45 Npr=Cp*mu/k // Prandtl number
46
47 //From empirical eqn:
48 mu_w=mu //
49 Nnu=0.36*Nre^0.55*Npr^(1/3)
50 ho=Nnu*k/De // [W/sq m.K]
51 hi=1050 // Given [W/sq m.K]
52 Uo=1/(1/ho+(1/hi)*(Do/Di)) // [W/sq m.K]
53 Uc=Uo //W/sq m.K
54
55 //Suitability of heat exchanger
56 Rd_given=9*10^-4 // [W/sq m.K]
57 Rd=(Uc-Ud)/(Uc*Ud) // [W/sq m.K]
58 printf("\n Rd calculated (%f W/m^2.K) is mazimum
allowable scale resistance\n",Rd);
59 printf("\n\nAs Rd calculated (%f W/sq m.K)(OR
1.1*10^-3) is more than Rd given (%f W/sq m,K), the
given heat exchanger is suitable\n",Rd,Rd_given)
;

```

Scilab code Exa 5.11 Number of tubes required

```
1 clc;
```

```

2 clear;
3 //Example 5.11
4 mw_dot=1720;           //water in [kg/h]
5 t1=293 ;              // [K]
6 t2=318 ;              // [K]
7 Cpw=4.28;             // [kJ/kg.K]
8 Q=mw_dot*Cpw*(t2-t1)  // [kJ/h]
9 Q=Q*1000/3600         //W
10 lambda=2230;         // [kJ/kg]
11 dT1=90 ;             // [K]
12 dT2=65 ;             // [K]
13 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
14
15 // Calculation of inside heat transfer coefficient
16 Di=0.0225;           // [m]
17 u=1.2 ;              // [m/s]\
18 rho=995.7 ;          // [kg/m^3]
19 v=0.659*10^-6       // [m/s]
20 mu=v*rho             // [kg/m.s]
21 Nre=Di*u*rho/mu      //reynolds number
22 Cp=Cpw*1000          // [J/kg.K]
23 k=2.54 ;             // [kJ/h.m.K]
24 k=k*1000/3600        // [W/m.K]
25 Npr=Cp*mu/k          //Prandtl number
26 Nnu=0.023*Nre^0.8*Npr^0.4 //Nusselt number
27 hi=k*Nnu/Di          // [W/sq m.K]
28 ho=19200             // [kJ/h.m^2.K]
29 ho=ho*1000/3600     // [W/m^2.K]
30 Do=0.025            // [m]
31 Dw=(Do-Di)/log(Do/Di) // [m]
32 x=(Do-Di)/2         // [m]
33 kt=460               //For tube wall material [kJ/h.m.K]
34 kt=kt*1000/3600     // [W/m.K]
35 Uo=1/(1/ho+(1/hi)*(Do/Di)+(x/kt)*(Do/Dw)) // [
    W/sq m.K]
36 //Q=Uo*Ao*dTlm
37 Ao=Q/(Uo*dTlm)      // [sq m]
38 L=4                  //Tube length in [m]

```

```

39 n=Ao/(%pi*Do*L)           //[Number of tubes]
40 n=round(n)                //Approximate
41 printf("\n Number of tubes reuired= %d",n);

```

Scilab code Exa 5.12 Shell and tube heat exchanger

```

1  clc;
2  clear;
3  //Example 5.12
4  t1=290                    //Inlet temperature of cooling water
   [K]
5  ho=2250                  //Heat transfer coefficient based on
   inside area in [W/sq m.K]
6  lambda=400              //[kJ/kg] LAtent heat of benzene
7  mb_dot=14.4             //[t/h] Condensation rate of benzene
   vapour
8  Cpw=4.187               //Specific heat
9  //With no Scale
10
11 Q=mb_dot*1000*lambda     //Heat duty of condenser in
   [kJ/h]
12 Q=(Q/3600)*1000        //[W]
13 //Shell and tube type of heat exchanger is used as a
   single pass surface condenser
14 Di=0.022                //I.D of tube [m]
15 L=2.5                   //Length of each tube in [m]
16 n=120                   //Number of tubes
17 A=%pi*Di*L             //Area of heat transfer per metre
   length in [m^2/m]
18 A=n*A                   //Total area of heat transfer in [m
   ^2]
19 Ai=(%pi/4)*Di^2        //Cross-sectional area of each tube
   in [m^2]
20 Ai=n*Ai                //Total area of flow in [m^2]
21 u=0.75                  //Velocty of water [ms^-1]

```

```

22 V=u*Ai           //Volumetric flow of water
23 rho=1000         //[Density of water in [kg/m^3]]
24 mw_dot=V*rho     //Mass flow rate of water in [kg/s]
25
26 //Heat balance
27
28 //Q=mw_dot*Cpw*(t2-t1)
29 t2=Q/(mw_dot*Cpw*1000)+t1           //[K]
30 T=350           //Condensing benzene temperature in [K]
31 dT1=T-t1       //[K]
32 dT2=T-t2       //[K]
33 dTlm=(dT1-dT2)/log(dT1/dT2)         //LMTD
34 U=Q/(A*dTlm)   //[W/m^2.K]
35 U=round(U)
36 //Neglecting resistance ,we have:
37 hi=1/(1/U-1/ho)           //[W/m^2.K]
38 //hi is proportional to u^0.8
39 C=hi/(u^0.8)             //Constant
40
41 //With Scale
42
43 Rd=2.5*10^-4             //[m^2 K./W]
44 //1/U=1/hi+1/ho+Rd
45 //U=hi/(1+3.38*u^0.8)
46 //mw_dot=rho*u*Ai       //[kg/s]
47 //Let t2 be the outlet temperature of water
48 //Q=mw_dot*Cpw*(t2-t1)
49 //t2=Q/(mw_dot*Cpw)+t1
50 dT1=60
51 //dT2=T-(t1+8.373/u)
52 //dTlm=8.373/(u*log(60*u/(60*u-8.373)))
53 //Q=U*A*dTlm
54 //1.89=((u^-0.2)/(1+3.38*u^0.8))*(1/log((60*u)/60*u
    -8.373))
55 //If we assume values of u greater than 0.75 m/s
56 //For u=3.8             //[ms^-1]
57 u=3.8                   //[ms^-1]
58 printf("\nWater velocity must be 3.88 ms^-1");

```

Scilab code Exa 5.13 Length of pipe in Exchanger

```
1  clc;
2  clear;
3  //Example 5.13
4  mh_dot=1.25           // [kg/s]
5  Cpw=4.187*10^3       // Heat capacity of water in [J/
   kg.K]
6  lambda=315           // [kJ/kg]
7  Q=mh_dot*lambda      // Rate of heat transfer from
   vapour [kJ/s]
8  Q=Q*10^3            // [W]
9  Ts=345              // Temperature of condensing
   vapour [K]
10 t1=290              // Inlet temperature of water [K]
11 t2=310              // Outlet temperature of water [K]
12 dT1=Ts-t1          // [K]
13 dT2=Ts-t2          // [K]
14 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
15 //Heat removed from vapour = Heat gained
16 mw_dot=Q/(Cpw*(t2-t1)) // [kg/s]
17 hi=2.5             // [kW/sq m.K]
18 hi=hi*1000         // [W/sq m.K]
19 Do=0.025           // [m]
20 Di=0.020           // [m]
21 hio=hi*(Di/Do)     // Inside heat transfer
   coefficient referred to outside dia in [W/sq m.K]
22 ho=0.8             // Outside heat transfer
   coefficient in [kW/sq m.K]
23 ho=ho*1000         // [W/sq m.K]
24 Uo=1/(1/ho+1/hio) // [W/sq m.K]
25 //Ud is 80% of Uc
26 Ud=(80/100)*Uo     // [W/sq m.K]
27 Ao=Q/(Ud*dTlm)     // [sq m]
```



```

28 L=1 // [m]
29 A=%pi*Do*L // Outside area of pipe
    per m length of pipe
30 len=Ao/A // Total length of piping
    required.
31 rho=1000 // [kg/m^3]
32 V=mw_dot/rho // [m^3/s]
33 v=0.6 // [m/s]
34 a=V/v // Cross-sectional area
    for flow pass [sq m]
35 a1=(%pi*Di^2)/4 // [sq m]
36 //for single pass on tube side fluid(water)
37 n=round(a/a1) //No. of tubes
    per pass
38 l=len/n //Length of each tube in
    [m]
39 //For two passes on water side:
40 tn=2*n //Total no of tubes
41 l2=len/tn //Length of each tube in
    [m]
42 //For four passes on water side/tube side
43 tn2=4*n //Total no. of tubes
44 l3=len/tn2 //Length of each tube in
    [m]
45
46 printf("\nNo. of tubes=%d ,\nLength of tube=%f m",
    tn2 , l3);

```

Scilab code Exa 5.14 Dirt factor

```

1
2 clc
3 clear
4 //Example 5.14
5 //Properties of crude oil:

```

```

6 Cp2=1.986 ; // [kJ/(kg.K)]
7 mu1=2.9*10^-3; // [N.s/sq m]
8 k1=0.136 ; // [W/m.K]
9
10 rho1=824 ; // [kg/m^3]
11
12 // Properties of bottom product:
13 Cp2=2.202 ; // [kJ/kg.K]
14 rho2=867 ; // [kg/m^3]
15 mu2=5.2*10^-3 ; // [N.s/sq m]
16 k2=0.119 ; // [W/sq m.K]
17
18 mc_dot=135000 ; //Basis: cruid oil flow
    rate in [kg/h]
19 m_dot=106000 ; //Bottom product flow
    rate inn [kg/h]
20 t1=295 ; // [K]
21 t2=330 ; // [K]
22 T1=420 ; // [K]
23 T2=380 ; // [K]
24 dT1=T1-t2 // [K]
25 dT2=T2-t1 // [K]
26 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
27 Q=mc_dot*Cpc*(t2-t1) //kJ/h
28 Q=Q*1000/3600 // [W]
29
30 //Shell side calculations:
31 Pt=25 ; // [mm]
32 Pt=Pt/1000 ; // [m]
33 B=0.23 ; // [m]
34 Do=0.019 ; // [m] Outside
    diameter for square pitch
35 c_dash=Pt-Do // Clearance in [
    m]
36 id=0.6 ; // [m]
37 as=id*c_dash*B/Pt // Cross flow
    area of shell [sq m]
38 //since there is a Calculaiton mistake ,we take:

```

```

39 as=0.0353;
40 Gs=m_dot/as //Shell side
    mass velocity in [kg/sq m.h]
41 Gs=Gs/3600; // [kg/sq m.s]
42 De=4*(Pt^2-(%pi/4)*Do^2)/(%pi*Do) // [m]
43 Nre=De*Gs/mu2 //Reynolds
    number
44 Npr=Cp2*1000*mu2/k2 //Prandtl
    number
45 muw=mu2 //Since mu/muw=1
46 Nnu=0.36*(Nre^0.55)*Npr^(1.0/3.0)*(mu2/muw)^(0.14)
    //Nusselt number
47 ho=Nnu*k2/De // [W/sq m.K]
48
49 //Tube side heat transfer coefficient:
50 n=324 ; //No. of tubes
51 n_p=324/2 ; //No.of tubes per pass
52 t=2.1 ; //Thickness in [mm]
53 t=t/1000 ; // [m]
54 Di=Do-2*t //I.d of tube in [m]
55 A=(%pi/4)*(Di^2) //Cross-sectional area of
    one tube in [sq m]
56 A_p=n_p*A //Total area for flow per
    pass in [sq m]
57 G=mc_dot/A_p // [kg/sq m h]
58 G=G/3600 // [kg/sq m.s]
59 Nre=Di*G/mu1 //Reynolds number
60 Npr=42.35 ; //Prandtl number
61 Nnu=0.023*(Nre^0.8)*(Npr^0.4) //Nusselt number
62 hi=Nnu*k1/Di // [W/sq m.K]
63 hio=hi*Di/Do // [W/sq m.K]
64 Uo=1/(1/ho+1/hio) // [W/sq m.K]
65 Uc=Uo
66 L=4.88 ; //Length of
    tube in [m]
67 Ao=n*%pi*Do*L // [sq m]
68 Ud=Q/(Ao*dTlm) // [W/sq m.K]
69 Rd=(Uc-Ud)/(Uc*Ud) // [m^2.K/W]

```

```

70 printf("\n The calculation of line no.36 to
    calculated as is wrongly done in Book by printing
    0.0353 , ,..which is wrong\n");
71 printf("\nRd=%f K/w,or 7.34*10^-4 which is less than
    the provided ,so this if installed will not give
    required temperarues without frequent cleaning\n\
n" ,Rd);

```

Scilab code Exa 5.15 Heat transfer area

```

1  clc;
2  clear;
3  //Example 5.15
4
5  //CASE I:
6  Cp=4*10^3;           // [J/kg.K]
7  t1=295;             // [K]
8  t2=375;             // [K]
9  sp=1.1;             // Specific gravity of liquid
10 v1=1.75*10^-4;      //Flow of liquid in [m^3/s]
11 rho=sp*1000         // [kg/m^3]
12 m_dot=v1*rho        // [kg/s]
13 Q=m_dot*Cp*(t2-t1) // [W]
14 T=395;              // [K]
15 dT1=T-t1           // [K]
16 dT2=T-t2           // [K]
17 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
18 U1A=Q/dTlm         // [W/K]
19
20 //CASE-II
21 v2=3.25*10^-4      //Flow in [m^3/s]
22 T2=370             // [K]
23 m_dot=v2*rho       // [kg/s]
24 Q=m_dot*Cp*(T2-t1) // [W]
25 dT1=T-t1           // [K]

```

```

26 dT2=T-T2 // [K]
27 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
28 U2A=Q/dTlm // [W/K]
29 //since u is propn to v
30 //hi =C*v^0.8
31
32 U2_by_U1=U2A/U1A
33
34 ho=3400 //Heat transfer
    coeff for condensing steam in [W/sq m.K]
35 C=poly(0,"C")
36 //Let C=1 and v=v1
37 //C=1;
38 v=v1; // =1.75*10^-4 m^3/s
39 hi=C*v^0.8
40 U1=1/(1/ho+1/hi) //
41
42 //When v=v2
43 v=v2;
44 hi=C*v^0.8
45 U2=1/(1/ho+1/hi) //
46
47 //Since U2=1.6U1
48 //On solving we get:
49 C=142497
50 v=v1
51 hi=C*v^0.8
52 U1=1/(1/ho+1/hi) //
53 A=U1A/U1 //Heat transfer area in [sq
    m]
54 printf("\n Overall heat transfer coefficient is %f W
    /sq m.K and\n\nHeat transfer area is %f sq m",U1,
    A);

```

Scilab code Exa 5.16 Oil Cooler

```

1  clc;
2  clear;
3  //Example 5.16
4  mo_dot=6*10^-2           // [kg/s]
5  Cpo=2*10^3              // Specific heat of
   oil in [J/kg.K]
6  Cpw=4.18*10^3          // Specific heat of water
   in [J/kg.K]
7  T1=420                  // [K]
8  T2=320                  // [K]
9  T=290                   // [K] Water entering
   temperature
10 Q=mo_dot*Cpo*(T1-T2)   // [J/s]=[W]
11 //Heat given out =Heat gained
12 t2=Q/(mo_dot*Cpw)+T    // [K]
13 dT1=T1-t2              // [K]
14 dT2=T2-T               // [K]
15 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
16 hi=1.6*1000            // [W/sq m.K]
17 ho=3.6*1000            // [W/sq m.K]
18 U=1/(1/ho+1/hi)       // [W/sq m.K]
19 A=Q/(U*dTlm)           // [sq m]
20 D=0.025                // [m]
21 L=A/(%pi*D)            // [m]
22 printf("\n Length of tube required = %f m",L);

```

Scilab code Exa 5.17 Countercurrent flow heat exchanger

```

1  clc;
2  clear;
3  //Example 5.17
4  mb_dot=1.25             //Benzene in [kg/s]
5  Cpb=1.9*10^3           //For benzene in [J/kg.K]
6  Cpw=4.187*10^3        //in [J/kg.K]
7  T1=350                 // [K]

```

```

8 T2=300 // [K]
9 Q=mb_dot*Cpb*(T1-T2) // [W]
10 t1=290 // [K]
11 t2=320 // [K]
12 dT1=T1-t2 // [K]
13 dT2=T2-t1 // [K]
14 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
15 mw_dot=Q/(Cpw*(t2-t1)) //Minimum flow rate of
    water in [kg/s]
16 hi=850 // [W/sq m.K]
17 ho=1700 // [W/sq m.K]
18 Do=0.025 // [m]
19 Di=0.022 // [m]
20 x=(Do-Di) /2 //Thickness in [m]
21 hio=hi*(Di/Do) // [W/sq m.K]
22 Dw=(Do-Di)/log(Do/Di) // [m]
23 k=45 // [W/m.K]
24 Uo=1/((1/ho)+(1/hio)+(x/k)*(Do/Dw)) // [W/sq m.K]
25 Ao=Q/(Uo*dTlm) // [sq m]
26 L=1 //Length in [m]
27 area=%pi*Do*L // Outside surface
    area of tube per i m length
28 Tl=Ao/area //Total length of
    tubing required in [m]
29 printf("\nTotal length of tubing required=%d m",
    round(Tl));

```

Scilab code Exa 5.18 Vertical Exchanger

```

1 clc;
2 clear;
3 //Example 5.18
4 m_dot=4500 //Benzene condensation rate in [
    kg/h]
5 lambda=394 //Latent heat of condensation of

```

```

        benzene in [kJ/kg]
6  Q=m_dot*lambda      // [kJ/h]
7  Q=Q*1000/3600      // [W]
8  Cpw=4.18           // [kJ/kg.K]
9  t1=295              // [K]
10 t2=300              // [K]
11 //For water :
12 mw_dot=Q/(Cpw*1000*(t2-t1)) // [kg/s]
13 rho=1000            // [kg/m^3]
14 V=mw_dot/rho        // Volumetric flow rate in [m
    ^3/s]
15 u=1.05              // [m/s]
16 A=V/u               // Cross-sectional area
    required in [sq m]
17
18 //For tube:
19 x=1.6                // thickness in [mm]
20 x=x/1000            // [m]
21 Do=0.025            // [m]
22 Di=Do-2*x           // [m]
23 A1=(%pi*Di^2)/4     // Of one tube [sq m]
24 n=A/A1              // No. of tubes reuired
25 n=round(n)
26 L=2.5               // Length of tube in [m]
27 Ao=n*pi*Do*L        // Surface area for heat transfer
    in [sq m]
28 Ts=353              // Condensing temp of benzene in
    [K]
29 T1=295              // Inlet temperature in [K]
30 T2=300              // Outlet temperature in [K]
31 dT1=Ts-T1          // [K]
32 dT2=Ts-T2          // [K]
33 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
34 Uo=Q/(Ao*dTlm)     // [W/sq mK]
35 Ud=Uo               // [W/sq m.K]
36
37 //OVERALL HEAT TRANSFER COEFFICIENT:
38 //Inside side:

```



```

39 T=(T2+T1)/2 // [K]
40
41 hi=1063*((1+0.00293*T)*u^0.8)/(Di^0.2) // [W/
    sq m.K]
42 hio=hi*(Di/Do) // [W
    /sq m.K]
43 Dw=(Do-Di)/log(Do/Di) // [m
    ]
44 k=45 //For tube in [W/(m.) ]
45
46 //Outside of tube:
47 mdot_dash=1.25/n // [kg/s ]
48 M=mdot_dash/(%pi*Do) // [kg/(m. s )]
49 k=0.15 // [W/(m.K) ]
50 rho=880 // [kg/m^3 ]
51 mu=0.35*10^-3 // [N. s/sq m]
52 g=9.81 // [m/s ^2]
    Acceleration due to gravity
53 hm=(1.47*((4*mdot_dash)/mu)^(-1/3))/(mu^2/(k^3*rho
    ^2*g))^(1/3) // [W/sq m.K]
54 ho=hm // [W/sq m.K]
55 k=45 // [W/m]
56 Uo=1/(1/ho+1/hio+(x*Do)/(k*Dw))
57 //Uo=1/(1/ho+1/hio+(x*Do)/(k*Dw)) // Overall heat
    transfer coefficient in [W/sq m.K]
58 Uc=Uo // [W/sq m.K]
59
60 Rd=(Uc-Ud)/(Uc*Ud) //Maximum
    allowable sclae resistance in [K/W]
61 printf("\n Uc(%f) is in excess of Ud(%f), therefore
    we allow for reasonable scale resistance ,\nRd=%f
    K/W\n",Uc,Ud,Rd);
62 printf("\n No. of tubes = %d ",n)

```

Scilab code Exa 5.19 Countercurrent Heat Exchanger

```

1  clc;
2  clear;
3  //Example 5.19
4  mw_dot=5;           //Water flow rate in [kg/s]
5  Cpw=4.18;          //Heat capacity of water [kJ/kg
   .K]
6  t1=303;            // [K]
7  t2=343;            // [K]
8  Q=mw_dot*Cpw*(t2-t1) // [kJ/s]
9  Q=Q*1000;          // [W]
10 T1=413;            // [K]
11 T2=373;            // [K]
12 dT1=T1-t2          // [K]
13 dT2=T2-t1          // [K]
14 dTlm=dT1           // [K]
15 hi=1000;           // [W/sq m.K]
16 ho=2500;           // [W/sq m.K]
17 Rd=1/(0.714*1000) // Fouling factor [m^2.K/KW]
18 U=1/(1/hi+1/ho+Rd) // [W/sq m.K]
19 A=Q/(U*dTlm)       // [sq m]
20 printf("\\nHeat transfer area is %f sq m",A);

```

Scilab code Exa 5.20 Number of tube side pass

```

1  clc;
2  clear;
3  //Example 5.20
4  Cpo=1.9             //Heat capacity for oil [kJ/kg.K]
5  Cps=1.86            //Heat capacity for steam [kJ/kg.K]
6  ms_dot=5.2          //Mass flow rate in [kg/s]
7  T1=403              // [K]
8  T2=383              // [K]
9
10 Q=ms_dot*Cps*(T1-T2) // [kJ/s]
11 Q=Q*1000            // [W]

```

```

12 t1=288; // [K]
13 t2=358; // [K]
14 dT2=T1-t2 // [K]
15 dT1=T2-t1 // [K]
16 dTlm=(dT1-dT2)/log(dT1/dT2) //LMTD in [K]
17 U=275 ; //Overall heat
    transfer coeffcient in [W//sq m.K]
18 Ft=0.97 //LMTD correction
    factor
19 A=Q/(U*Ft*dTlm) // [sq m]
20 printf("\nHeat exchanger surface area is %f sq m",A)
    ;

```

Scilab code Exa 5.21 Number of tubes passes

```

1 clc;
2 clear;
3 //Example 5.21
4 mc_dot=3.783; //Cold water flow rate [kg/s]
5 mh_dot=1.892; //Hot water flow rate [kg/s]
6 Cpc=4.18; //Sp heat of cold water [kJ/(kg.
    K)]
7 T1=367; // [K]
8 t2=328; // [K]
9 t1=311; // [K]
10 Cph=4.18; //Specific heat of hot water [kJ
    /(kg.K)]
11 rho=1000; //Density [kg/m^3]
12 D=0.019; //Diameter of tube in [m]
13 U=1450 ; //Overal heat transfer
    coefficient in [W/sq m.K]
14 T2=T1-mc_dot*Cpc*(t2-t1)/(mh_dot*Cph) // [K]
15 Q=mc_dot*Cpc*(t2-t1) // [kJ/s]
16 Q=Q*1000 // [W]
17 //For counterflow heat exchanger

```

```

18 dT1=T1-t2 // [K]
19 dT2=17; // [K]
20 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
21 lmtd=dTlm //LMTD
22 Ft=0.88 //LMTD correction factor
23 A=Q/(U*dTlm) // [sq m]
24 u=0.366; //Velocity through tubes
    [ms^-1]
25 Ai=mc_dot/(rho*u) //Total flow Area in [sq
    m]
26 n=Ai/((%pi/4)*(D^2)) //No. of tubes
27 L=1 //Per m length [m]
28 sa=%pi*D*L //S.S per tube per 1 m
    length
29 L=A/(n*%pi*D) //Length of tubes in [m]
30 printf("\nThe length is more than allowable 2.44 m
    length ,so we must use more than one tube \n");
31
32 //For 2passes on the tube side
33 A=Q/(U*Ft*lmtd) //[sq m]
34 L=A/(2*n*%pi*D) //Length in [m]
35 printf("\n This length is within 2.44 m requirement ,
    so the design choice is \n\n");
36 printf("\nType of heat exchanger : 1-2 Shell and
    tube heat exchanger\n");
37 printf("\nNo of tubes per pass= %d\n",round(n));
38 printf("\nLength of tube per pass=%f m\n ",L);

```

Scilab code Exa 5.22 Outlet temperature for hot and cold fluids

```

1 clc;
2 clear;
3 //Example 5.22
4 mh_dot=16.67; //Mass flow rate of hot fluid in
    [kg/s]

```

```

5 mc_dot=20;           //Mass flow rate of cold fluid in
   [kg/s]
6 Cph=3.6;            //Sp heat of hot fluid in [kJ/kg
   .K]
7 Cph=Cph*1000;      //Sp heat of hot fluid in [J/kg
   .K]
8 Cpc=4.2;           //Sp heat of cold fluid in [kJ/(
   kg.K) ]
9 Cpc=Cpc*1000;      //Sp heat of cold fluid in [J/(
   kg.K) ]
10 U=400;             //Overall heat transfer
   coefficient in [W/sq m.K]
11 A=100;             //Surface area in [sq m]
12 mCp_h=mh_dot*Cph  //[J/s] or [W/K]
13 mCp_c=mc_dot*Cpc  //[J/s] or [W/K]
14 mCp_small=mCp_h   //[W/K]
15 C=mCp_small/mCp_c //Capacity ratio
16 ntu=U*A/mCp_small //NTU
17 T1=973;           //Hot fluid inlet temperature in
   [K]
18 t1=373;           //Cold fluid inlet temperature
   in [K]
19 //Case 1:Countercurrent flow arrangement
20 E=(1-%e^(-(1-C)*ntu))/(1-C*%e^(-(1-C)*ntu)) //
   Effectiveness
21 //W=T1-T2/(T1-t1) therefore:
22 T2=T1-E*(T1-t1)   //[K]
23 printf("\nExit temperature of hot fluid is %d K",
   round(T2));
24 t2=mCp_h*(T1-T2)/(mCp_c)+t1 // [From energy
   balance eqn in ][K]
25 printf("\nExit temperature of cold fluid is %d K(%d
   C)\n",round(t2),round(t2-273));
26
27 //Case 2:Parallel flow arrangement
28 E1=(1-%e^(-(1+C)*ntu))/(1+C)
29 //In the textbok here is a calculation mistake ,and
   the value of E is takne as E=0.97

```

```

30
31 T2=T1-E1*(T1-t1)           //[K]
32 t2=mCp_h*(T1-T2)/(mCp_c)+t1      //[From energy
    balance eqn in ][K]
33 printf("\nExit temperature of Hot water=%f K\n",T2);
34 printf("\nExit temperature of cold water=%f K\n",t2)
    ;

```

Scilab code Exa 5.23 Counterflow concentric heat exchanger

```

1  clc;
2  clear;
3  //Example 5.23
4  Cpo=2131;           //Sp heat of oil in [J/kg.K]
5  Cpw=4187;           //Sp heat of water in [J/kg.K]
6  mo_dot=0.10;        //Oil flow rate in [kg/s]
7  mw_dot=0.20;        //Water flow rate in [kg/s]
8  U=380;              //Overall heat transfer coeff in [W/
    sq m.K]
9  T1=373;             //Initial temp of oil [K]
10 T2=333;             //Final temperature of oil [K]
11 t1=303;             //Water enter temperature in [K]
12 t2=t1+mo_dot*Cpo*(T1-T2)/(mw_dot*Cpw)    //[K]
13 // 1.LMTD method
14 dT1=T1-t2           //[K]
15 dT2=T2-t1           //[K]
16 dTlm=(dT1-dT2)/log(dT1/dT2)             //[K]
17 lmtd=dTlm;         //[K]
18 Q=mo_dot*Cpo*(T1-T2)           //[J/s]
19 A=Q/(U*dTlm)                 //[sq m]
20 Do=0.025;                    //Inner tube diameter [
    m]
21 L=A/(%pi*Do)                 //Length in [m]
22
23 // 2.NTU method

```

```

24 mCp_c=mw_dot*Cpw           // [W/K]
25 mCp_h=mo_dot*Cpo           // [W/K]
26 printf("\n In textbook this value of mCp_h is
    wrongly calculated as 231.1 so we will take this
    only for calculation\n");
27 mCp_h=231.1;               // [W/K]
28 //mCp_h is smaller
29 C=mCp_h/mCp_c
30 E=(T1-T2)/(T1-t1)          // Efficiency
31 //For countercurrent flow
32 deff(' [x]=f(ntu)', 'x=E-(1-%e^(-(1-C)*ntu))/(1-C*%e
    ^(-(1-C)*ntu))')
33 ntu=fsolve(1,f)
34 A=ntu*mCp_h/U              // [sq m]
35 A=0.56                      // Approximately
36 L1=A/(%pi*Do)              // Length in [m]
37 printf("\nFrom LMTD approach:\n length=%f m\n",L);
38 printf("\nFrom NTU method:\n length=%f m\n",L1);

```

Scilab code Exa 5.24 Number of tubes required

```

1
2 clc;
3 clear;
4 //Example 5.24
5 ho=200;           // [W/sq m.K]
6 hi=1500;         // [W/sq m.K]
7 Cpw=4.2;         //Sp heat of Water in [kJ/(kg.K)]
8 Cpo=2.1;         //Sp heat of Oil in [kJ/(kg.K)]
9 E=0.8;           // Effectiveness
10 k=46;           // [W/m.K]
11 m_dot=0.167;    // [kg/s]
12
13 mCp_oil=2*m_dot*Cpo*1000 //For oil [W/K]
14 //mCp_oil is wrongly calculated as 710.4

```

```

15 mCp_water=m_dot*Cpw*1000      //For water [W/K]
16 //mCp_oil is wrongly calculated as 710.4
17 //NOTE:The above two values are wrongly calculated
    in book as 710.4
18 //so we take here:
19 mCp_small=710.4      //[W/K]
20 //Since both mCp_water and mCp_oil are equal ,
    therefore:
21 C=1;
22
23 deff (' [x]=f( ntu) ', 'x=E-(ntu/(1+ntu)) ');
24 ntu=fsolve(1,f)
25 id=20;      //Internal diameter in [mm]
26 od=25;      //External diameter in [mm]
27 hio=hi*id/od      //[W/sq m.K]
28 Dw=(od-id)/log(od/id)      //[mm]
29 Dw=Dw/1000      //[m]
30 x=(od-id)/2      //[mm]
31 x=x/1000      //[m]
32 Do=0.025      //External dia in [m]
33 L=2.5;      //Length of tube in [m]
34 Uo=1/(1/ho+1/hio+(x/k)*(Dw/Do))      //[W/sq m.K]
35 A=ntu*mCp_small/Uo      //Heat transfer area in [sq m]
36 n=A/(%pi*Do*L)      //No of tubes
37 printf("\nNo. of tubes required = %d",round(n+1));

```

Scilab code Exa 5.25 Parallel and Countercurrent flow

```

1 clc;
2 clear;
3 //Example 5.25
4
5 //(i) Parallel flow
6 T1=633;      //[K]
7 t2=303;      //[K]

```



```

8 T2=573;          // [K]
9 t1=400;          // [K]
10 dT1=T1-t2;     // [K]
11 dT2=T2-t1;     // [K]
12 mh_dot=1.2;    // [kg/s]
13 U=500;          // Overall heat transfer coefficient in [
    W/sqm.K]
14 Cp=2083;        // Sp.heat of oil J/kg.K
15 dTlm=(dT1-dT2)/log(dT1/dT2) // [K]
16 Q=mh_dot*Cp*(T1-T2) // [W]
17 A=Q/(U*dTlm)   // [sq m]
18
19 //(ii)Counter current flow
20 dT1=T1-t1;     // [K]
21 dT2=T2-t2;     // [K]
22 dTlm=(dT2-dT1)/log(dT2/dT1) // [K]
23 A1=Q/(U*dTlm) // [sq m]
24 printf("\nFor parallel flow ,Area = %f sq m \n For
    countercurrent flow ,Area=%f sq m\n",A,A1);
25 printf("\n\nFor the same terminal temperatures of
    the fluid ,the surface area for the counterflow
    arrangement\n is less than the required for the
    parallel flow\n")

```

Chapter 6

Evaporation

Scilab code Exa 6.1 Boiling point Elevation

```
1  clc;
2  clear;
3  //Example6.1
4  T=380          //B.P of solution [K]
5  T_dash=373    //B.P of water [K]
6  BPE=T-T_dash  //Boiling point elevation in [K]
7  Ts=399        //Saturating temperature in [K]
8  DF=Ts-T       //Driving force in [K]
9  printf("\nBoiling point of elevation of the solution
         is %d K \n",BPE);
10 printf("\nDriving forve for heat transfer is %d K \n
         ",DF)
```

Scilab code Exa 6.2 Capacity of evaporator

```
1  clc;
2  clear;
3  //Example 6.2
```

```

4 m_dot=10000 //Weak liquor entering in [kg/h]
5 fr_in=0.04 //Fraciton of caustic soda IN i.e 4
  %
6 fr_out=0.25 //Fraciton of caustic soda OUT i.e 25
  %
7 //Let mdash_dot be the kg/h of thick liquor leaving
8 mdash_dot=fr_in*m_dot/fr_out // [kg/h]
9
10 //Overall material balance
11 //kg/h of feed=kg/h of water evaporated +kg/h of
  thick liquor
12 //we=water evaporated in kg/h
13 //Therefore
14 we=m_dot-mdash_dot // [kg/h]
15 printf("\n Capacity of evaporator is %d kg/h",we);

```

Scilab code Exa 6.3 Economy of Evaporator

```

1 clc;
2 clear;
3 //Exmample 6.3
4 ic=0.05 //Initial concentration (5%)
5 fc=0.2 //Final concentration (20%)
6 T_dash=373 //B.P of water in [K]
7 bpe=5 //Boiling point elevation [K]
8 mf_dot=5000 // [Basis] feed to evaporator in [
  kg/h]
9 //Material balance of solute
10 mdash_dot=ic*mf_dot/fc // [kg/h]
11 //Overall material balance
12 mv_dot=mf_dot-mdash_dot //Water evaporated [kg/
  h]
13 lambda_s=2185 //Latent heat of condensation
  of steam [kJ/kg]
14 lambda_v=2257 //Latent heat of vaporisation of

```

```

        water [kJ/kg]
15 lambda=lambda_v // [kJ/kg]
16 T=T_dash+bpe //Temperature of thick liquor [K]
17 Tf=298 //Temperature of feed [K]
18 Cpf=4.187 //Sp. heat of feed in [kJ/kg.K]
19 //Heat balance over evaporator=ms_dot
20 ms_dot=(mf_dot*Cpf*(T-Tf)+mv_dot*lambda)/lambda_s
    //Steam consumption [kg/h]
21 Eco=mv_dot/ms_dot //Economy of evaporator
22 Ts=399 //Saturation temperature of steam in [K]
23 dT=Ts-T //Temperature driving force [K]
24 U=2350 // [W/sq m.K]
25 Q=ms_dot*lambda_s //Rate of heat transfer in [
    kJ/kg]
26 Q=Q*1000/3600 // [J/s]=[W]
27 A=Q/(U*dT) //Heat transfer area in [sq
    m]
28 printf("\nANSWER Economoy pf evaporator is %f \n",
    Eco);
29 printf("\nHeat tarnsfer area to be provided = %f sq
    m\n",A);

```

Scilab code Exa 6.4 Steam economy

```

1
2 clc;
3 clear;
4 //Example 6.4
5 Cpf=3.98 //Specific heat of feed in kJ/(kg.K)
6 lambda_s=2202 //Latent heat of conds of heat at
    0.2MPa in [kJ/kg]
7 lambda=2383 //Latent heat of vaporisation of
    water aty 323 [kJ/kg]
8 ic=0.1 //Initial concentration of soilds in
    [%]

```

```

9 fc=0.5 //Final concentration
10 m_dot=30000 //Feed to evaporator in [kg/h]
11 mdash_dot=ic* m_dot/fc //Mass flow rate of thick
    liquor in [kg/h]
12 mv_dot=m_dot-mdash_dot //Water evaporated in [
    kg/h]
13
14 //Case 1: Feed at 293K
15 mf_dot=30000 // [kg/h]
16 mv_dot=24000 // [kg/h]
17 Cpf=3.98 // [kJ/(kg.K)]
18 Ts=393 //Saturation temperature of steam in [K]
19 T=323 //Boiling point of solution [K]
20 lambda_s=2202 //Latent heat of condensation [
    kJ/kg]
21 lambda=2383 //Latent heat of vaporisation [kJ/kg]
22 Tf=293 //Feed temperature
23 //Enthalpy balance over the evaporator:
24 ms_dot=(mf_dot*Cpf*(T-Tf)+mv_dot*lambda)/lambda_s
    //Steam consumption [kg/h]
25 eco=(mv_dot/ms_dot) //Steam economy
26 printf("\nWhen Feed introduced at 293 K ,Steam
    economy is %f\n",eco);
27 dT=Ts-T // [K]
28 U=2900 // [W/sq m.K]
29 Q=ms_dot*lambda_s //Heat load =Rate of
    heat transfer in [kJ/h]
30 Q=Q*1000/3600 // [J/s]
31 A=Q/(U*dT) //Heat transfer area
    required [sq m]
32 printf("\n ANSWER-(i)\n\n At 293 K,Heat transfer
    area required is %f sq m\n",A);
33
34 //Case2: Feed at 308K
35 Tf=308 // [Feed temperature] [K]
36 ms_dot=(mf_dot*Cpf*(T-Tf)+mv_dot*lambda)/lambda_s
    //Steam consumption in [kg/h]
37 eco=mv_dot/ms_dot //Economy of

```

```

    evaporator
38 printf("\n ANSWER-(ii)\n\n When T=308 K \nEconomy of
    evaporator is %f\n",eco);
39 Q=ms_dot*lambda_s // [kJ/h]
40 Q=Q*1000/3600 // [J/s]
41 A=Q/(U*dT) //Heat transfer area
    required [sq m]
42 printf('\nANSWER-(iii) \n When T=308 K,\nHeat
    transfer Area required is %f sq m\n",A);

```

Scilab code Exa 6.5 Evaporator economy

```

1  clc;
2  clear;
3  //Example 6.5
4  m_dot=5000 //Feed to the evaporator [kg/h]
5  Cpf=4.187 //Cp of feed in [kJ/kg.K]
6  ic=0.10 //Initial concentration
7  fc=0.4 //Final concentration
8  mdash_dot=m_dot*ic/fc // [kg/h] of thick
    liquor
9  mv_dot=m_dot-mdash_dot //Water evaporated
    in [kg/h]
10 lambda_s=2162 //Latent heat of condensing
    steam [kJ/kg]
11 P=101.325 //Pressure in the evaporator [kPa]
12 bp=373 // [K]
13 Hv=2676 //Enthalpy of water vapor [kJ/kg]
14 H_dash=419 // [kJ/kg]
15 Hf=170 // [kJ/kg]
16 ms_dot=(mv_dot*Hv+mdash_dot*H_dash-m_dot*Hf)/
    lambda_s //Steam consumption in [kg/h]
17 eco=mv_dot/ms_dot //Steam economy of
    evaporator
18 Q=ms_dot*lambda_s // [kJ/h]

```

```

19 U=1750 // [W/sq m.K]
20 dT=34 // [K]
21 Q=Q*1000/3600 // [J/s]
22 A=Q/(U*dT) // [sq m]
23 printf("\n Heat transfer area to be provided is %f
sq m",A);

```

Scilab code Exa 6.6 Single effect Evaporator

```

1 clc;
2 clear;
3 //Example 6.6
4 mf_dot=5000 // [kg/h]
5 ic=0.01 //Initial concentration [kg/h]
6 fc=0.02 //Final concentration [kg/h]
7 T=373 //Boiling pt of saturation in [K]
8 ]
9 Ts=383 //Saturation temperature of
steam in [K]
10 mdash_dot=ic*mf_dot/fc // [kg/h]
11 mv_dot=mf_dot-mdash_dot //Water evaporated in [
kg/h]
12 Hf=125.79 // [kJ/kg]
13 Hdash=419.04 // [kJ/kg]
14 Hv=2676.1 // [kJ/kg]
15 lambda_s=2230.2 // [kJ/kg]
16 ms_dot=(mdash_dot*Hdash+mv_dot*Hv-mf_dot*Hf)/
lambda_s //Steam flow rate in [kg/h]
17 eco=mv_dot/ms_dot //Steam economy
18 Q=ms_dot*lambda_s //Rate of heat transfer
in [kJ/h]
19 Q=Q*1000/3600 // [J/s]
20 dT=Ts-T // [K]
21 A=69 //Heating area of evaporator in [sq

```

```

    m]
22 U=Q/(A*dT)           //Overall heat transfer coeff in [W/
    sq m.K]
23 printf("\nSteam economy is %f\n",eco);
24 printf("\n\nOverall heat transfer coefficient is %d
    W/sq m.K",round(U));

```

Scilab code Exa 6.7 Single effect evaporator reduced pressure

```

1  clc;
2  clear;
3  //Example 6.7
4  //From previous example:
5  mf_dot=5000           //[kg/h]
6  Hf=125.79           //[kJ/kg]
7  lambda_s=2230.2     //[kJ/kg]
8  mdash_dot=2500      //[kg/h]
9  Hdash=313.93        //[kJ/kg]
10 mv_dot=2500         //[kg/h]
11 Hv=2635.3          //[kJ/kg]
12 ms_dot=(mdash_dot*Hdash+mv_dot*Hv-mf_dot*Hf)/
    lambda_s           //Steam flow rate in [kg/h]
13 Q=ms_dot*lambda_s   //[kJ/h]
14 Q=Q*1000/3600       //[W]
15 U=2862              //[W/sq m.K]
16 dT=35              //[K]
17 A=Q/(U*dT)         //[sq m]
18 printf("\n The heat transfer area in this case is %f
    sq m\n",A);
19 printf("\n\nNOTE :There is a calculation mistake in
    the book at the line12 of this code,ms_dot value
    is written as 2320.18,which is wrong\n\n");

```

Scilab code Exa 6.8 Mass flow rate

```
1  clc;
2  clear;
3  //Example 6.8
4  mf_dot=6000           //Feed rate in [kg/h]
5  //Taking the given values from previous example(6.6)
6  Hf=125.79           //[kJ/kg]
7  ms_dot=3187.56      //[kg/h]
8  lambda_s=2230.2     //[kJ/kg]
9  Hdash=419.04        //[kJ/kg]
10 Hv=2676.1           //[kJ/kg]
11 mv_dot=(mf_dot*Hf+ms_dot*lambda_s-6000*Hdash)/(Hv-
    Hdash) //Water evaporated in [kg/h]
12 mdash_dot=6000-mv_dot //Mass flow rate of
    product [kg/h]
13 x=(0.01*mf_dot)*100/mdash_dot //Wt % of solute
    in products
14 printf("\nMass flow rate of product is %f kg/h\n\n",
    mdash_dot);
15 printf("\n\nThe product concentration is %f percent
    by weight \n\n",x);
```

Scilab code Exa 6.9 Heat load in single effect evaporator

```
1  clc;
2  clear;
3  //Example 6.9
4  Tf=298           //Feed temperature in [K]
5  T_dash=373       //[K]
6  Cpf=4           //[kJ/kg.K]
7  fc=0.2          //Final concentration of salt
8  ic=0.05         //Initial concentration
9  mf_dot=20000     //[kg/h] Feed to evaporator
10 mdash_dot=ic*mf_dot/fc //Thick liquor [kg/h]
```

```

11 mv_dot=mf_dot-mdash_dot      //Water evaporated in [
    kg/h]
12 lambda_s=2185                //[kJ/kg]
13 lambda=2257                  //[kJ/kg]
14 bpr=7                        //Boiling point rise [K]
15 T=T_dash+bpr                //Boiling point of solution in [K]
16 Ts=39                        //Temperature of condensing steam in [K]
17 ms_dot=(mf_dot*Cpf*(T-Tf)+mv_dot*lambda)/lambda_s
    //Steam consumption in [kg/h]
18 eco=mv_dot/ms_dot            //Economy of evaporator
19 Q=ms_dot*lambda_s            //[kJ/h]
20 Q=Q*1000/3600                //[J/s]
21 printf("\nHeat load is %d W or J/s",round(Q));
22 printf("\n\nEconomy of evaporator is %f ",eco);
23
24 printf("\n\nNOTE:Again there is a calcualtion
    mistake in book at line 19 of code,it is written
    as 4041507.1 instead of 40415071 \n\n");

```

Scilab code Exa 6.10 Triple effect evaporator

```

1  clc;
2  clear;
3  //Example 6.10
4  Ts=381.3                      //[K]
5  dT=56.6;                      //[K]
6  U1=2800; //Overall heat transfer coeff in first
    effect
7  U2=2200; //Overall heat transfer coeff in first
    effect
8  U3=1100; //Overall heat transfer coeff in first
    effect
9  dT1=dT/(1+(U1/U2)+(U1/U3))    ///[K]
10 dT2=dT/(1+(U2/U1)+(U2/U3))    ///[K]
11 dT3=dT-(dT1+dT2)              //[K]

```

```

12 //dT1=Ts-T1_dash      //[K]
13 T1_dash=Ts-dT1
14 //dT2=T1_dash-T2_dash      //[K]
15 T2_dash=T1_dash-dT2      //[K]
16 printf("\n\nBoiling point of solution in first
    effect =%f K\n\n",T1_dash);
17 printf("\n\nBoiling point of solution in second
    effect =%f K\n\n",T2_dash);

```

Scilab code Exa 6.11 Double effect evaporator

```

1  clc;
2  clear;
3  //Example 6.11
4  mf_dot=10000      //[kg/h] of feed
5  ic=0.09          //Initial concentration
6  fc=0.47          //Final concentration
7  m1dot_dash=ic*mf_dot/fc      //[kg/h]
8  Ps=686.616      //Steam pressure [kPa.g]
9  Ps=Ps+101.325    //[kPa]
10 Ts=442.7         //Saturation temperature in [K]
11 P2=86.660        //Vacuum in second effect in [kPa]
12 U1=2326          //Overall heat transfer in first effect
    [W/sq m.K]
13 U2=1744.5        //Overall heat transfer in 2nd effect [W
    /sqm.K]
14 P2_abs=101.325-P2 //Absolute pressure in second
    effect [kPa]
15 T2=326.3         //Temperature in 2nd effect in [K]
16 dT=Ts-T2         // [K]
17 Tf=309           //Feed temperature in [K]
18 T=273            // [K]
19 Cpf=3.77         //kJ/kg.K Specific heat for all
    caustic streams
20 //Q1=Q2

```

```

21 //U1*A1*dT1=U2*A2*dT2
22 dT2=dT/1.75 // [K]
23 dT1=(U2/U1)*dT2 // [K]
24 //Since there is no B.P.R
25 Tv1=Ts-dT1 //Temperature in vapor space of
    first effect in [K]
26 Tv2=Tv1-dT2 //Second effect [K]
27 Hf=Cpf*(Tf-T) //Feed enthalpy [kJ/kg]
28 H1dash=Cpf*(Tv1-T) //Enthalpy of final product [
    kJ/kg]
29 H2dash=Cpf*(Tv2-T) //kJ/kg
30 //For steam at 442.7 K
31 lambda_s=2048.7 // [kJ/kg]
32 //For vapour at 392.8 K
33 Hv1=2705.22 // [kJ/kg]
34 lambda_v1=2202.8 // [kJ/kg]
35 //for vapour at 326.3 K:
36 Hv2=2597.61 // [kJ/kg]
37 lambda_v2=2377.8 // [kJ/kg]
38
39 //Overall material balance:
40 mv_dot=mf_dot-m1dot_dash // [kg/h]
41
42 //Equation 4 becomes:
43 //mv1_dot*lambda_v1+mf_dot*Hf=(mv_dot-mv1_dot)*Hv2+(
    mf_dot-mv2_dot)*H2_dash
44 mv1_dot=(H2dash*(mf_dot-mv_dot)-mf_dot*Hf+mv_dot*Hv2
    )/(Hv2+lambda_v1-H2dash)
45 mv2_dot=mv_dot-mv1_dot // [kg/h]
46
47 //From equation 2
48
49 m2dot_dash=m1dot_dash+mv1_dot //First
    effect material balance [kg/h]
50 ms_dot=(mv1_dot*Hv1+m1dot_dash*H1dash-m2dot_dash*
    H2dash)/lambda_s // [kg/h]
51
52

```

```

53 //Heat transfer Area
54 //First effect
55 A1=ms_dot*lambda_s*(10^3)/(3600*U1*dT1) // [sq m]
56
57 //Second effect
58 lambda_v1=lambda_v1*(10^3/3600)
59 A2=mv1_dot*lambda_v1/(U2*dT2) // [sq m]
60
61 //Since A1 not= A2
62
63 //SECOND TRIAL
64 Aavg=(A1+A2)/2 // [sq m]
65 dT1_dash=dT1*A1/Aavg // [K]
66 dT2_dash=dT-dT1 // [K]
67
68 //Temperature distribution
69 Tv1=Ts-dT1_dash // [K]
70 Tv2=Tv1-dT2_dash // [K]
71 Hf=135.66 // [kJ/kg]
72 H1dash=Cpf*(Tv1-T) // [kJ/kg]
73 H2dash=200.83 // [kJ/kg]
74
75 //Vapour at 388.5 K
76 Hv1=2699.8 // [kJ/kg]
77 lambda_v1=2214.92 // [kJ/kg]
78 mv1_dot=(H2dash*(mf_dot-mv_dot)-mf_dot*Hf+mv_dot*Hv2
    )/(Hv2+lambda_v1-H2dash)
79 mv2_dot=mv_dot-mv1_dot // [kg/h]
80
81 //First effect Energy balance
82 ms_dot=((mv1_dot*Hv1+m1dot_dash*H1dash)-(mf_dot-
    mv2_dot)*H2dash)/lambda_s // [kg/h]
83
84 //Area of heat transfer
85 lambda_s=lambda_s*1000/3600
86 A1=ms_dot*lambda_s/(U1*dT1_dash) // [sq m]
87
88 //Second effect:

```

```

89 A2=mv1_dot*lambda_v1*1000/(3600*U2*dT2_dash)
    // [sq m]
90
91 printf("\nA1(%f)=A2(%f),So the area in each effect
    can be %f sq m\n",A1,A2,A2);
92 printf("\nHeat transfer surface in each effect is %f
    sq m\n",A2);
93 printf("\nSteam consumption=%d kg/h\n",round(ms_dot)
    );
94 printf("\nEvaporation in the first effect is %d kg/h
    \n",round(mv1_dot));
95 printf("\nEvaporation in 2nd effect is %d kg/h\n",
    round(mv2_dot));

```

Scilab code Exa 6.12 lye in Triple effect evaporator

```

1
2 clc;
3 clear;
4 //Example 6.12
5 Tf=353;           // [K]
6 T=273             // [K]
7 mf_dot=10000;    //Feed [kg/h]
8 ic=0.07;         //Initial conc of glycerine
9 fc=0.4;          //FinaL CONC OF GLYCERINE
10 //Overall glycerine balance
11 m3dot_dash=(ic/fc)*mf_dot           // [kg/h]
12 mv_dot=mf_dot-m3dot_dash           ///[kg/h]
13 P=313;           //Steam pressure [kPa]
14 Ts=408;         // [from steam table][K]
15 P1=15.74;       // [Pressure in last effect][kPa]
16 Tv3=328;        // [Vapour temperature]
17 dT=Ts-Tv3       // Overall apparent [K]
18 bpr1=10 ;       // [K]
19 bpr2=bpr1;

```

```

20 bpr3=bpr2;
21 sum_bpr=bpr1+bpr2+bpr3 // [K]
22 dT=dT-sum_bpr // True_Overall
23 dT1=14.5; // [K]
24 dT2=16; // [K]
25 dT3=19.5; // [K]
26 Cpf=3.768 // [kJ/(kg.K)]
27 //Enthalpies of various streams
28 Hf=Cpf*(Tf-T) // [kJ/kg]
29 H1=Cpf*(393.5-T) // [kJ/kg]
30 H2=Cpf*(367.5-T) // [kJ/kg]
31 H3=Cpf*(338-T) // [kJ/kg]
32 //For steam at 40K
33 lambda_s=2160 // [kJ/kg]
34 Hv1=2692 // [kJ/kg]
35 lambda_v1=2228.3 // [kJ/kg]
36 Hv2=2650.8 // [kJ/kg]
37 lambda_v2=2297.4 // [kJ/kg]
38 Hv3=2600.5 // [kJ/kg]
39 lambda_v3=2370 // [kJ/kg]
40
41 //MATERIAL AND EBERGY BALANCES
42 //First effect
43 //Material balance
44
45 //m1dot_dash=mf_dot-mv1_dot
46 //m1dot_dash=1750+mv2_dot+mv3_dot
47
48 //Energy balance
49 //ms_dot*lambda_s+mf_Dot*hf=mv1_dot*Hv1+m1dot_dash*
H1
50 //2160*ms_dot+2238*(mv2_dot+mv3_dot)=19800500
51
52 //Second effect
53 //Energy balance:
54 //mv3_dot=8709.54-2.076*mv2_dot
55
56 //Third effect:

```

```

57 //m2dot_dash=mv3_dot+m3dot_dash
58 //m2dot_dash=mv3_dot+1750
59 //From eqn 8 we get
60 mv2_dot
    =(8709.54*2600.5+1750*244.92-8790.54*356.1-356.1*1750)
    /(-2.076*356.1+2297.4+2600.5*2.076)
61 //From eqn 8:
62 mv3_dot=8709.54-2.076*mv2_dot // [kg/h]
63 mv1_dot=mv_dot-(mv2_dot+mv3_dot) // [kg/h]
64 //From equation 4:
65 //m1dot_dash=mf_dot-mv1_dot
66 //ms_dot=(mv1_dot*Hv1+m1dot_dash*H1-mf_dot*Hf)/
    lambda_s // [kg/h]
67 ms_dot=(19800500-2238*(mv2_dot+mv3_dot))/2160
    // [kg/h]

68
69 //Heat transfer Area is
70 U1=710 // [W/sq m.K]
71 U2=490 // [W/sq m.K]
72 U3=454 // [W/sq m.K]
73 A1=ms_dot*lambda_s*1000/(3600*U1*dT1) // [sq m]
74 A2=mv1_dot*lambda_v1*1000/(3600*U2*dT2) // [sq m]
75 A3=mv2_dot*lambda_v2*1000/(3600*U3*dT3) // [sq m]
76 //The deviaiton is within +-10%
77 //Hence maximum A1 area can be recommended
78
79 eco=(mv_dot/ms_dot) // [Steam economy]
80
81 Qc=mv3_dot*lambda_v3 // [kJ/h]
82 dT=25 //Rise in water temperature
83 Cp=4.187
84 mw_dot=Qc/(Cp*dT)
85 printf("\nANSWER\n Area in each effect%f sq m\n",A1)
    ;
86 printf("\nANSWER \n Steam economy is%f\n",eco);
87 printf("\nANSWER Cooling water rate is %f t/h",
    mw_dot/1000)

```


Scilab code Exa 6.13 Triple effect unit

```
1  clc ;
2  clear ;
3  //Example 6.13
4  Cpf=4.18           // [kJ/kg.K]
5  dT1=18            // [K]
6  dT2=17            // [K]
7  dT3=34            // [K]
8  mf_dot=4          // [kg/s]
9  Ts=394            // [K]
10 bp=325            //Bp of water at 13.172 kPa [K]
11 dT=Ts-bp          // [K]
12 lambda_s=2200     // [kJ/kg]
13 T1=Ts-dT1         // [K]
14 lambda1=2249      // [kJ/kg]
15 lambda_v1=lambda1 // [kJ/kg]
16
17 T2=T1-dT2         // [K]
18 lambda2=2293      // [kJ/kg]
19 lambda_v2=lambda2 // [kJ/kg]
20
21 T3=T2-dT3         // [K]
22 lambda3=2377      // [kJ/kg]
23 lambda_v3=lambda3 // [kJ/kg]
24
25 ic=0.1            //Initial conc of solids
26 fc=0.5            //Final conc of solids
27 m3dot_dash=(ic/fc)*mf_dot // [kg/s]
28 mv_dot=mf_dot-m3dot_dash //Total evaporation
    in [kg/s]
29 //Material balance over first effect
30 //mf_dot=mv1_dot+m1dot_dash
31 //Energy balance:
```

```

32 //ms_dot*lambda_s=mf_dot*(Cpf*(T1-Tf)+mv1_dot*
    lambda_v1)
33
34 //Material balance over second effect
35 //m1dot_dash=mv2_dot+m2dot_dash
36 //Enthalpy balance:
37 //mv1_dot*lambda_v1+m1dot_dash(cp*(T1-T2)=mv2_dot*
    lambda_v2)
38
39 //Material balance over third effect
40 //m2dot_dash=mv3_dot+m3dot+dash
41
42 //Enthalpy balance:
43 //mv2_lambda_v2+m2dot_dash*cp*(T2-T3)=mv3_dot*
    lambda_v3
44 294
45 mv2_dot=3.2795/3.079 // [kg/s]
46 mv1_dot=1.053*mv2_dot-0.1305 // [kg/s]
47 mv3_dot=1.026*mv2_dot+0.051 // [kg/s]
48 ms_dot=(mf_dot*Cpf*(T1-294)+mv1_dot*lambda_v1)/
    lambda_s // [kg/s]
49 eco=mv_dot/ms_dot //Steam economy
50 eco=round(eco)
51 printf("\nSteam economy is %d\n",eco);
52 U1=3.10 // [kW/sq m.K]
53 U2=2 // [kW/sq m.K]
54 U3=1.10 // [kW/sq m.K]
55 //First effect:
56 A1=ms_dot*lambda_s/(U1*dT1) // [sq m]
57 A2=mv1_dot*lambda_v1/(U2*dT2) // [sq m]
58 A3=mv2_dot*lambda_v2/(U3*dT3) // [sq m]
59 //Areas are calculated witha deviation of +-10%
60 printf("\nArea pf heat transfer in each effect is %f
    sq m\n",A3)

```

Scilab code Exa 6.14 Quadruple effect evaporator

```
1  clc ;
2  clear ;
3  //Example 6.14
4  mf_dot=1060      // [kg/h]
5  ic=0.04         // Initial concentration
6  fc=0.25         // Final concentration
7  m4dot_dash=(ic/fc)*mf_dot      // [kg/h]
8  //Total evaporation=
9  mv_dot=mf_dot-m4dot_dash      // [kg/h]
10
11 //Fromsteam table :
12 P1=370          // [kPa.g]
13 T1=422.6       // [K]
14 lambda1=2114.4 // [kJ/kg]
15
16 P2=235          // [kPa.g]
17 T2=410.5       // [K]
18 lambda2=2151.5 // [kJ/kg]
19
20 P3=80           // [kPa.g]
21 T3=390.2       // [K]
22 lambda3=2210.2 // [kJ/kg]
23
24 P4=50.66        // [kPa.g]
25 T4=354.7       // [K]
26 lambda4=2304.6 // [kJ/kg]
27
28 P=700           // Latent heat of steam [kPa .g]
29 lambda_s=2046.3 // [kJ/kg]
30
31 //FIRST EFFECT
32 // Enthalpy balance :
33 //ms_dot=mf_dot*Cpf*(T1-Tf)+mv1_dot*lambda1
34 //ms_dot=1345.3-1.033*m1dot_dash
35
36 //SECOND EFFECT
```

```

37 //m1dot_dash=m2dot_dash+mdot_v2
38 //Enthalpy balance:
39 //m1dot_dash=531.38+0.510*m2dot_dash
40
41 //THIRD EFFECT
42 //Material balance:
43 //m2dot_dash-m3dot_dash+mv3_dot
44
45 //FOURTH EFFECT
46 //m3dot_dash=m4dot_dash+mv4_dot
47 mv4dot_dash=169.6 // [kg/h]
48 m3dot_dash=416.7 // [kg/h]
49
50 //From eq n 4:
51 m2dot_dash=-176.84+1.98*m3dot_dash // [kg/h]
52
53 //From eqn 2:
54 m1dot_dash=531.38+0.510*m2dot_dash // [kg/h]
55
56 //From eqn 1:
57 ms_dot=1345.3-1.033*m1dot_dash
58 eco=mv_dot/ms_dot // [kg evaporation /kg
    steam]
59 printf("\nSteam economy is %f evaporation/kg steam",
    eco);

```

Scilab code Exa 6.15 Single effect Calendria

```

1 clc;
2 clear;
3 //Example 6.15
4 m1_dot=5000 // [kg/h]
5 ic=0.1 // Initial concentration
6 fc=0.5 // Final concentration
7 mf_dot=(fc/ic)*m1_dot // [kg/h]

```

```

8  mv_dot=mf_dot-m1_dot           //Water evaporated [kg/h]
9  P=357                          //Steam pressure [kN/sq m]
10 Ts=412                         // [K]
11 H=2732                         // [kJ/kg]
12 lambda=2143                    // [kJ/kg]
13 bpr=18.5                       // [K]
14 T_dash=352+bpr                 // [K]
15 Hf=138                         // [kJ/kg]
16 lambda_s=2143                  // [kJ/kg]
17 Hv=2659                       // [kJ/kg]
18 H1=568                         // [kJ/kg]
19 ms_dot=(mv_dot*Hv+m1_dot*H1-mf_dot*Hf)/lambda_s
                                //Steam consumption in kg/h
20 printf("\nSteam consumption is %f kg/h\n",ms_dot);
21 printf("\nCapacity is %f kg/h\n",mv_dot);
22 eco=mv_dot/ms_dot              //Economy
23 printf("\nSteam economy is %f\n",eco);
24 dT=Ts-T_dash                  // [K]
25 hi=4500                       // [W/sq m.K]
26 ho=9000                       // [W/sq m.K]
27 Do=0.032                      // [m]
28 Di=0.028                      // [m]
29 x1=(Do-Di)/2                  // [m]
30 Dw=(Do-Di)/log(32/28)        // [m]
31 x2=0.25*10^-3                // [m]
32 L=2.5                          //Length [m]
33 hio=hi*(Di/Do)                // [W/sq m.K]
34 printf("\n NOTE:In textbook this value of hio is
    wrongly calculated as 3975.5..So we will take
    this\n\n");
35 hio=3975.5
36 k1=45                          //Tube material in [W/sq m.K]
37 k2=2.25                        //For scale [W/m.K]
38 Uo=1/(1/ho+1/hio+(x1*Dw)/(k1*Do)+(x2/k2)) //
    Overall heat transfer coeff in W/sq m.K
39 Q=ms_dot*lambda_s             // [kJ/h]
40 Q=Q*1000/3600                // [W]
41

```

```

42 A=Q/(Uo*dT)           // [sq m]
43 n=A/(%pi*Do*L)       //from A=n*%pi*Do*L
44 printf("\n No. of tubes required is %d",round(n));

```

Scilab code Exa 6.16 Single effect evaporator

```

1  clc;
2  clear;
3  //Example 6.16
4  bpr=40.6;           // [K]
5  Cpf=1.88;          // [kJ/kg.K]
6  Hf=214;            // [kJ/kg]
7  H1=505;            // [kJ/kg]
8  mf_dot=4536;       // [kg/h] of feed solution
9  ic=0.2;            // Initial conc
10 fc=0.5;            // Final concentration
11 m1dot_dash=(ic/fc)*mf_dot // Thisck liquor flow
    arte [kg/h]
12 mv_dot=mf_dot-m1dot_dash // [kg/H]
13 Ts=388.5;          // Saturation temperature of steam
    in [K]
14 bp=362.5           //b.P of solution in [K]
15 lambda_s=2214;     // [kJ/kg]
16 P=21.7;            // Vapor space in [kPa]
17 Hv=2590.3;         // [kJ/kg]
18
19 //Enthalpy balance over evaporator
20 ms_dot=(m1dot_dash*H1+mv_dot*Hv-mf_dot*Hf)/lambda_s
    // [kg/h]
21 printf("\nSteam consumption is %f kg/h\n",ms_dot);
22 dT=Ts-bp           // [K]
23 U=1560              // [W/sq m.K]
24 Q=ms_dot*lambda_s   // [kJ/h]
25 Q=Q*1000/3600       // [W]
26 A=Q/(U*dT)         // [sq m]

```

```

27 printf("\nHeat transfer area is %f sq m\n",A);
28
29 //Calculations considering enthalpy of superheated
    vapour
30
31 Hv=Hv+Cpf*bpr // [kJ/kg]
32 ms_dot=(m1dot_dash*H1+mv_dot*Hv-mf_dot*Hf)/lambda_s
    // [kg/h]
33 printf("\n Now, Steam consumption is %f kg/h\n",
    ms_dot);
34 eco=mv_dot/ms_dot //Steam economy
35 printf("\nEconomy of evaporator %f\n",eco);
36 Q=ms_dot*lambda_s // [kJ/h]
37 Q=Q*1000/3600 // [w]
38 A2=Q/(U*dT) //Area
39 printf("\nNow, Area is %f\n",A);
40 perc=(A2-A)*100/A // %error in the heat
    transfer area
41 printf("\n If enthalpy of water vapour Hv were based
    on the saturated vapour at the pressure\nthe
    error introduced is only %f percent\n",perc);

```
